

**INTEGRASI *MACHINE LEARNING* DENGAN *KANSEI*  
*ENGINEERING* DAN *KANO* SEBAGAI UPAYA MENINGKATKAN  
KEPUASAN PELANGGAN APLIKASI *E-GROCERY*  
(STUDI KASUS: APLIKASI ALFAGIFT)**

**TUGAS AKHIR**

**Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Strata-1  
Program Studi Teknik Industri - Fakultas Teknologi Industri  
Universitas Islam Indonesia**



Nama : Miranda Alya Leadi  
No. Mahasiswa : 19522392

**PROGRAM STUDI TEKNIK INDUSTRI PROGRAM SARJANA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA  
2023**

## PERNYATAAN KEASLIAN

### PERNYATAAN KEASLIAN

Saya mengakui bahwa tugas akhir ini adalah hasil karya saya sendiri kecuali kutipan dan ringkasan yang seluruhnya sudah saya jelaskan sumbernya. Jika dikemudian hari ternyata terbukti pengakuan saya ini tidak benar dan melanggar peraturan yang sah maka saya bersedia ijazah yang telah saya terima ditarik kembali oleh Universitas Islam Indonesia.

Yogyakarta, 25 Oktober 2023



Miranda Alya Leadi  
19522392

## SURAT BUKTI PENELITIAN



FAKULTAS  
TEKNOLOGI INDUSTRI

Gedung KH. Mas Mansur  
Kampus Terpadu Universitas Islam Indonesia  
Jl. Kaliurang km 14,5 Yogyakarta 55584  
T. (0274) 898444 ext. 4110, 4100  
F. (0274) 895007  
E. fti@uii.ac.id  
W. fti.uii.ac.id

### SURAT KETERANGAN PENELITIAN

Nomor : 170/Ka.lab SIMANTI/20/Lab.SIMANTI/X/2023

*Assalamu 'alaikum Warohmatullahi Wabarokaatuh*

Dengan hormat,

Yang bertanda tangan dibawah ini, menerangkan bahwa:

Nama : Miranda Alya Leadi  
Nim : 19522392  
Jurusan : Teknik Industri  
Dosen Pembimbing : Ir. Abdullah 'Azzam S.T., M.T., IPM

Menyatakan bahwa mahasiswa tersebut diatas telah melaksanakan penelitian tugas akhir dengan judul **"INTEGRASI MACHINE LEARNING DENGAN KANSEI ENGINEERING DAN KANO SEBAGAI UPAYA MENINGKATKAN KEPUASAN PELANGGAN APLIKASI E-GROCERY (STUDI KASUS: APLIKASI ALFAGIFT)"** mulai pelaksanaan penelitian 19 Mei 2023 sampai 25 Oktober 2023.

Demikian surat keterangan penelitian ini kami buat. Atas perhatiannya dan kerja samanya kami mengucapkan terima kasih.

*Wassalamu 'alaikum Warohmatullahi Wabarokaatuh*

Yogyakarta, 26 Oktober 2023  
Kepala Laboratorium  
Sistem Manufaktur Terintegrasi

  
Putri Dwi Annisa, S.T., M.Sc.

## LEMBAR PENGESAHAN PEMBIMBING

### LEMBAR PENGESAHAN PEMBIMBING

**Integrasi *Machine Learning* dengan *Kansei Engineering* dan *Kano* Sebagai Upaya  
Meningkatkan Kepuasan Pelanggan Aplikasi *E-Grocery*  
(Studi Kasus: Aplikasi Alfagift)**



**Disusun Oleh :**

**Nama : Miranda Alya Leadi**

**No. Mahasiswa : 19522392**

**Yogyakarta, 26 Oktober 2023**

**Dosen Pembimbing**

A handwritten signature in black ink, appearing to be 'Abdullah 'Azzam', written over a white background.

**Ir. Abdullah 'Azzam, S.T., M.T., IPM**

**LEMBAR PENGESAHAN DOSEN PENGUJI**

**INTEGRASI *MACHINE LEARNING* DENGAN *KANSEI ENGINEERING* DAN  
*KANO* SEBAGAI UPAYA MENINGKATKAN KEPUASAN PELANGGAN  
APLIKASI *E-GROCERY* (STUDI KASUS: APLIKASI ALFAGIFT)**

**TUGAS AKHIR**

**Disusun Oleh :**

**Nama : Miranda Alya Leadi**

**No. Mahasiswa : 19 522 392**

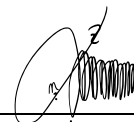
**Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk  
memperoleh gelar Sarjana Strata-1 Teknik Industri Fakultas Teknologi Industri  
Universitas Islam Indonesia**

**Yogyakarta, 21 - November - 2023**

**Tim Penguji**

Ir. Abdullah 'Azzam, S.T., M.T., IPM

Ketua



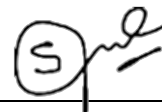
Suci Miranda, S.T., M.Sc.

Anggota I



Wahyudhi Sutrisno, S.T., M.M., M.T

Anggota II

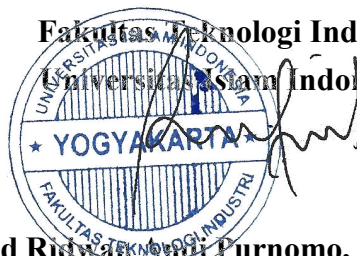


**Mengetahui,**

**Ketua Program Studi Teknik Industri Program Sarjana**

**Fakultas Teknologi Industri**

**Universitas Islam Indonesia**



**Ir. Muhammad Ridwan Purnomo, S.T., M.Sc., Ph.D., IPM**

**015220101**

## HALAMAN PERSEMBAHAN

*Dengan Menyebut Nama Allah yang Maha Pengasih lagi Maha Penyayang.  
Puji Syukur atas Kehadirat Allah SWT atas nikmat yang telah diberikan, Tugas Akhir  
ini saya persembahkan kepada:*

*Kedua orang tua saya yang tercinta, Bapak Yeri Leadi dan Ibu Kurniati (almh) yang tiada henti mendo'akan, mendidik, memberikan kasih sayang yang tiada henti serta semangat dari kecil hingga saat ini. Terima kasih untuk selalu menjadi sosok utama dan segalanya dalam hidup saya yang tidak pernah tergantikan. Terima kasih atas pengorbanan dan segala hal yang tidak bisa diungkapkan kata dan tak akan pernah bisa dibalas dengan apapun.*

*Kakak saya tersayang, Shafira Alyssa Leadi dan keluarga besar lainnya, terima kasih atas do'a, dukungan, dan motivasi yang selalu diberikan sampai saat ini dan seterusnya.*

*Serta kerabat, sahabat, dan teman-teman saya yang selalu mendukung, menemani dan membantu saya setiap hari dari awal hingga akhir masa perkuliahan ini*

## **MOTTO**

*“Maka sesungguhnya sesudah kesulitan itu ada kemudahan. Sesungguhnya sesudah kesulitan itu ada kemudahan”*

(QS. Al-Insyirah:5-6)

*“Barangsiapa menempuh jalan untuk mendapatkan ilmu, Allah akan memudahkan baginya jalan menuju surga.”*

(H.R. Muslim)

## KATA PENGANTAR

*Assalamu'alaikum Warahmatullahi Wabarakatuh*

Alhamdulillah, puji syukur penulis panjatkan kehadirat Allah SWT. Karena izin, karunia, petunjuk, dan rahmat-Nya, penulis dapat menyelesaikan Tugas Akhir yang berjudul **“Integrasi *Machine Learning* dengan *Kansei Engineering* dan *Kano* Sebagai Upaya Meningkatkan Kepuasan Pelanggan Aplikasi *E-Grocery* (Studi Kasus: Aplikasi *Alfagift*)”**. Shalawat serta salam senantiasa tercurahkan kepada Nabi Muhammad SAW beserta keluarga, sahabat serta para pengikut beliau hingga akhir zaman.

Penulisan Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat yang harus dipenuhi dalam menyelesaikan jenjang sarjana strata-1 (S-1) Teknik Industri, Fakultas Teknologi Industri, Universitas Islam Indonesia. Dalam proses menyelesaikan Tugas Akhir ini, penulis menyadari bahwa terdapat banyak bimbingan, bantuan, serta dukungan dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Bapak Prof. Dr. Ir. Hari Purnomo, M.T., IPU., ASEAN.Eng., selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
2. Bapak Dr. Drs. Imam Djati Widodo, M.Eng.Sc., selaku Ketua Jurusan Teknik Industri Fakultas Teknologi Industri, Universitas Islam Indonesia.
3. Bapak Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM., selaku Ketua Program Studi Teknik Industri Program Sarjana Universitas Islam Indonesia.
4. Bapak Ir. Abdullah ‘Azzam, S.T., M.T., IPM., selaku Dosen Pembimbing Tugas Akhir yang senantiasa meluangkan waktu untuk memberikan bimbingan, saran, nasehat, masukan, dan berbagai informasi yang berkaitan dengan pelaksanaan penelitian Tugas Akhir ini.
5. Bapak Yeri Leadi dan Ibu Kurniati (almh) yang tiada henti mendo’akan, mendidik, memberikan kasih sayang yang tiada henti serta semangat dari kecil hingga saat ini.
6. Kakak Shafira Alyssa Leadi selaku saudara kandung penulis yang juga turut memberikan motivasi, semangat, dan doa.
7. Teman-teman Teknik Industri UII Angkatan 2019 yang selalu mendukung, menemani dan membantu proses penyelesaian Tugas Akhir ini.
8. Semua pihak yang belum disebutkan melainkan terlibat dari awal hingga akhir dalam pembuatan tugas akhir.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna. Oleh karena itu, saran dan kritik yang membangun sangat diharapkan oleh penulis demi perbaikan di masa mendatang. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat khususnya dalam memberikan pengetahuan kepada pembaca.

*Wassalamu'alaikum Warahmatullahi Wabarakatuh*



Yogyakarta, 25 Oktober 2023

Penulis

A handwritten signature in black ink, appearing to read 'Miranda Alya Leadi', written in a cursive style.

Miranda Alya Leadi

## ABSTRAK

Indonesia menduduki posisi enam besar negara dengan pengguna internet yang melakukan pembelian produk atau jasa secara *online* setiap minggunya dengan persentase sebesar 62,6%. Di Indonesia sudah banyak ritel turut serta meluncurkan aplikasi untuk memudahkan konsumen membeli produk secara *online*, salah satunya merupakan Alfamart yang meluncurkan aplikasi Alfagift. Namun, berdasarkan perolehan *rating* dan *review* pengguna, tingkat kepuasan pengguna aplikasi ini masih tergolong rendah. Dalam kurun waktu setahun terakhir, aplikasi ini mengalami penurunan *rating* secara kontinu. Oleh karena itu, perlu dilakukan evaluasi pada aplikasi Alfagift untuk meningkatkan kepuasan pelanggan. Penelitian ini mengintegrasikan *Machine Learning* dengan *Kansei Engineering* dan *Kano*. Digunakan data ulasan pengguna aplikasi yang didapatkan pada situs *Google Play* dan *AppStore*. Penelitian ini bertujuan untuk mengetahui hasil integrasi *Machine Learning* dengan *Kansei Engineering* dan *Kano* dalam pengembangan aplikasi Alfagift guna meningkatkan kepuasan pelanggan. Berdasarkan hasil klasifikasi data, algoritma *Support Vector Machine* memperoleh nilai akurasi tertinggi sebesar 92,23%. Berdasarkan hasil analisis data ulasan menggunakan *Machine Learning* dengan kelompok sentimen negatif didapatkan 15 *kansei words* yaitu “lambat”, “bagus”, “mudah”, “rugi”, “kosong”, “kecewa”, “eror”, “tidak bisa”, “masuk”, “lama”, “tidak kirim”, “tidak masuk”, “salah”, “kurang”, dan “kecewa”. Kemudian dari hasil *machine learning* yang diintegrasikan dengan *Kansei Engineering* dan *Kano*, terdapat 10 atribut yang diprioritaskan untuk dilakukan peningkatan kualitas sistem, informasi, dan layanan karena berdampak besar pada kepuasan pengguna.

Kata Kunci: *Machine Learning*, Klasifikasi, *Kansei Engineering*, *Kano*, *E-Grocery*

## DAFTAR ISI

<b>PERNYATAAN KEASLIAN.....</b>	<b>ii</b>
<b>SURAT BUKTI PENELITIAN .....</b>	<b>iii</b>
<b>LEMBAR PENGESAHAN PEMBIMBING .....</b>	<b>iv</b>
<b>LEMBAR PENGESAHAN DOSEN PENGUJI.....</b>	<b>v</b>
<b>HALAMAN PERSEMBAHAN.....</b>	<b>vi</b>
<b>MOTTO.....</b>	<b>vii</b>
<b>KATA PENGANTAR .....</b>	<b>viii</b>
<b>ABSTRAK .....</b>	<b>x</b>
<b>DAFTAR ISI.....</b>	<b>xi</b>
<b>DAFTAR TABEL.....</b>	<b>xiii</b>
<b>DAFTAR GAMBAR.....</b>	<b>xv</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	10
1.3 Tujuan Penelitian .....	11
1.4 Manfaat Penelitian .....	11
1.5 Batasan Penelitian .....	11
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>13</b>
2.1 Kajian Literatur .....	13
2.2 Landasan Teori.....	27
2.2.1 Kansei Engineering.....	27
2.2.2 Web Scraping .....	31
2.2.3 Text Mining.....	31
2.2.4 Analisis Sentimen .....	32
2.2.5 Klasifikasi .....	33
2.2.6 Machine Learning.....	33
2.2.7 Support Vector Machine .....	35

2.2.8	K-Nearest Neighbor.....	36
2.2.9	Decision Tree .....	38
2.2.10	Naïve Bayes Classifier (NBC) .....	39
2.2.11	Term Weighting (Pembobotan).....	39
2.2.12	Uji Validitas .....	40
2.2.13	Uji Reliabilitas.....	41
2.2.14	Kepuasan Pelanggan.....	42
2.2.15	Kano.....	43
<b>BAB III METODE PENELITIAN .....</b>		<b>48</b>
3.1	Objek Penelitian.....	48
3.2	Jenis dan Sumber Data .....	48
3.2.1	Data Primer .....	48
3.2.2	Data Sekunder .....	48
3.3	Metode Pengumpulan Data.....	48
3.4	Tahapan Penelitian .....	48
<b>BAB IV PENGUMPULAN DAN PENGOLAHAN DATA.....</b>		<b>53</b>
4.1	Pemilihan <i>Service Domain</i> dan Pengumpulan Data Ulasan.....	53
4.2	Perentangan Ruang Semantik .....	55
4.3	Perentangan Ruang Properti .....	84
4.4	Sintesis.....	88
<b>BAB V ANALISIS DAN PEMBAHASAN .....</b>		<b>100</b>
5.1	Pemilihan <i>Domain</i> .....	100
5.2	Perentangan Ruang Semantik .....	100
5.3	Perentangan Ruang Properti .....	102
5.4	Sintesis.....	102
5.5	Rekomendasi Perbaikan .....	103
<b>BAB VI PENUTUP.....</b>		<b>114</b>
6.1	Kesimpulan.....	114
6.2	Saran.....	115
<b>DAFTAR PUSTAKA .....</b>		<b>116</b>
<b>LAMPIRAN.....</b>		<b>A-1</b>

## DAFTAR TABEL

Tabel 2. 1 Kajian Induktif.....	22
Tabel 2. 2 Kategori kelas Sentimen .....	32
Tabel 2. 3 <i>Kano Model Categorization</i> .....	45
Tabel 4. 1 Proses <i>Data Cleaning</i> dan <i>Case Folding</i> .....	55
Tabel 4. 2 <i>Pseudocode</i> dari Proses <i>Data Cleaning</i> dan <i>Case Folding</i> .....	56
Tabel 4. 3 Proses <i>Tokenizing</i> .....	57
Tabel 4. 4 <i>Pseudocode</i> dari Proses <i>Tokenizing</i> .....	57
Tabel 4. 5 Proses <i>Stopword Removal</i> .....	58
Tabel 4. 6 <i>Pseudocode</i> dari Proses <i>Stopword</i> .....	58
Tabel 4. 7 Proses Normalisasi.....	63
Tabel 4. 8 <i>Pseudocode</i> dari Proses Normalisasi .....	63
Tabel 4. 9 Proses <i>Stemming</i> .....	64
Tabel 4. 10 <i>Pseudocode</i> dari Proses <i>Stemming</i> .....	64
Tabel 4. 11 <i>Pseudocode</i> dari Pembobotan.....	65
Tabel 4. 12 Hasil Perhitungan Skor Sentimen .....	66
Tabel 4. 13 <i>Pseudocode</i> dari Proses Perhitungan Skor Sentimen.....	67
Tabel 4. 14 Jumlah Ulasan Kelas Sentimen .....	69
Tabel 4. 15 <i>Pseudocode</i> dari Proses Pelabelan Skor Sentimen .....	69
Tabel 4. 16 Pembagian Data Skenario 1.....	70
Tabel 4. 17 Pembagian Data Skenario 2.....	70
Tabel 4. 18 Pembagian Data Skenario 3.....	71
Tabel 4. 19 Pembagian Data Skenario 4.....	71
Tabel 4. 20 <i>Pseudocode</i> dari Proses Perhitungan Skor Sentimen.....	72
Tabel 4. 21 <i>Confusion Matrix</i> .....	74
Tabel 4. 22 <i>Pseudocode</i> dari Klasifikasi Setiap Algoritma .....	74
Tabel 4. 23 Hasil <i>Association Rules</i> .....	80
Tabel 4. 24 <i>Pseudocode</i> dari <i>Association Rules</i> .....	81
Tabel 4. 25 Ringkasan Asosiasi Terpilih.....	81
Tabel 4. 26 <i>Kansei Words</i> Terpilih .....	82
Tabel 4. 27 Pengelompokkan Jenis Kata pada <i>Kansei Word</i> Terpilih .....	83
Tabel 4. 28 <i>Kansei Word</i> Positif .....	84
Tabel 4. 29 Atribut dengan <i>Kansei Word</i> Negatif.....	85
Tabel 4. 30 Atribut dengan <i>Kansei Word</i> Positif.....	85
Tabel 4. 31 Atribut .....	85
Tabel 4. 32 Kuesioner <i>Kano</i> Fungsional .....	87
Tabel 4. 33 Kuesioner <i>Kano</i> Disfungsional.....	87
Tabel 4. 34 Hasil Uji Validitas Kuesioner <i>Kano</i> .....	87
Tabel 4. 35 Hasil Uji Reliabilitas .....	88

Tabel 4. 36 Klasifikasi kategori kano dari data kuesioner.....	89
Tabel 4. 37 Evaluasi Model Kano .....	91
Tabel 4. 38 Hasil Perhitungan Nilai <i>Better</i> dan <i>Worse</i> .....	93
Tabel 4. 39 Kategori Kano pada Setiap Atribut dengan Metode <i>Better-Worse</i> .....	94
Tabel 4. 40 Deskripsi Kategori Kano dan Tindakan Peningkatan .....	95
Tabel 4. 41 Konsep Perbaikan dan Urutan Prioritas Kategori Kano pada Aplikasi Alfagift .....	97

## DAFTAR GAMBAR

Gambar 1. 1 Kontribusi Perdagangan Besar dan Eceran, Reparasi, dan Sepeda Motor Dalam PDB Triwulan Tahun 2020-2022.....	2
Gambar 1. 2 Jumlah Ritel di Indonesia .....	2
Gambar 1. 3 Pembelian <i>Online</i> Mingguan .....	3
Gambar 1. 4 Aktivitas Belanja <i>Online</i> Mingguan.....	3
Gambar 1. 5 Faktor Pendorong Belanja <i>Online</i> .....	4
Gambar 1. 6 <i>Rating</i> Alfagift di <i>Google Play</i> .....	6
Gambar 1. 7 <i>Rating</i> Alfagift di <i>App Store</i> .....	6
Gambar 2. 1 Alur Proses <i>Kansei Engineering</i> .....	27
Gambar 2. 2 Perentangan Ruang Semantik .....	28
Gambar 2. 3 Tahapan Perentangan Ruang Properti .....	29
Gambar 2. 4 Fase Sintesis.....	29
Gambar 2. 5 Klasifikasi .....	33
Gambar 2. 6 Algoritma <i>Machine Learning</i> .....	34
Gambar 2. 7 <i>Hyperlane</i> pada <i>Support Vector Machine</i> .....	35
Gambar 2. 8 <i>K-Nearest Neighbor</i> .....	37
Gambar 2. 9 <i>Decision Tree</i> .....	38
Gambar 2. 10 <i>Kano Model Graph</i> .....	44
Gambar 3. 1 Alur Penelitian .....	49
Gambar 4. 1 <i>Interface AppFollow</i> .....	54
Gambar 4. 2 Ulasan Pengguna Aplikasi Alfagift.....	54
Gambar 4. 3 Hasil <i>Scraping Data</i> .....	55
Gambar 4. 4 Hasil Pembobotan TF-IDF .....	65
Gambar 4. 5 Hasil Akurasi Masing-Masing Algoritma .....	73
Gambar 4. 6 Persebaran Kata Kelas Sentimen Negatif.....	78
Gambar 4. 7 Persebaran Kata Kelas Sentimen Positif .....	79
Gambar 4. 8 Contoh Data Ulasan dengan Kelas Sentimen Positif .....	80
Gambar 4. 9 Diagram Kano Atribut Layanan Aplikasi Alfagift.....	94
Gambar 5. 1 Ulasan Pengguna Alfagift pada Atribut X3.....	104
Gambar 5. 2 Ulasan Pengguna Alfagift pada Atribut X3.....	104
Gambar 5. 3 Fitur <i>A-Poin</i> , <i>A-Voucher</i> , <i>Alfastamp</i> , dan <i>Alfatar</i> pada Aplikasi Alfagift .....	105
Gambar 5. 4 Ulasan Pengguna Alfagift pada Atribut X4.....	105
Gambar 5. 5 Ulasan Pengguna Alfagift pada Atribut X6.....	106
Gambar 5. 6 Ulasan Pengguna Alfagift pada Atribut X6.....	106
Gambar 5. 7 Ulasan Pengguna Alfagift pada Atribut X9.....	107
Gambar 5. 8 Tampilan Opsi Pengambilan Pesanan pada Aplikasi Alfagift .....	107
Gambar 5. 9 Ulasan Pengguna Alfagift pada Atribut X11 .....	108

Gambar 5. 10 Tampilan Struk Elektronik pada Aplikasi Alfagift .....	109
Gambar 5. 11 Ulasan Pengguna Alfagift pada Atribut X12 .....	109
Gambar 5. 12 Tampilan Ketersediaan Stok Barang pada Aplikasi Alfagift.....	111
Gambar 5. 13 Ulasan Pengguna Alfagift pada Atribut X14.....	111
Gambar 5. 14 Ulasan Pengguna Alfagift pada Atribut X15.....	112
Gambar 5. 15 Tampilan Metode Pembayaran <i>Cash on Delivery</i> pada Aplikasi Alfagift .....	113



## **BAB I**

### **PENDAHULUAN**

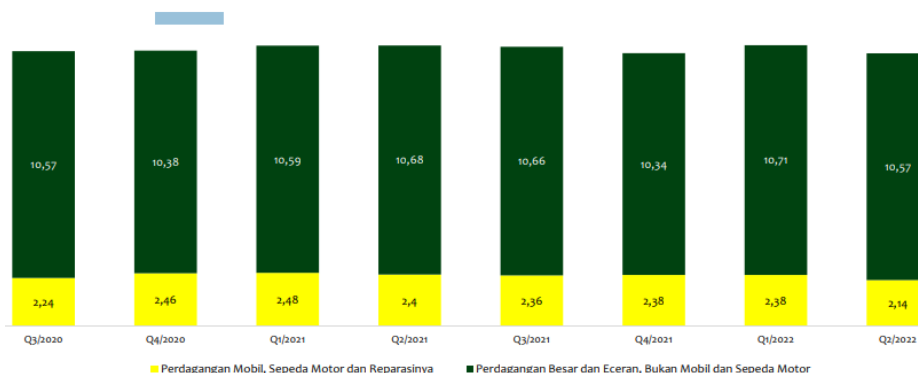
#### **1.1 Latar Belakang**

Bisnis ritel merupakan kegiatan usaha yang melibatkan penjualan produk ataupun jasa secara langsung dengan sasaran konsumen akhir untuk penggunaan pribadi, bukan untuk bisnis (M. Kamal Fathoni, 2021). Ritel merupakan mata rantai yang penting dalam proses distribusi barang dan merupakan mata rantai terakhir dalam suatu proses distribusi (Soliha, 2008). Dalam saluran pemasaran, ritel berperan sebagai perantara antara produsen dan konsumen, sehingga memberi kemudahan kepada konsumen akhir untuk mendapatkan produk yang diinginkan dalam jumlah kecil (ecer) tanpa harus membeli langsung ke produsen. Hal ini menunjukkan pentingnya peranan ritel sebagai perantara dagang (*middleman*) (Utomo, 2009).

Di Indonesia, bisnis ritel semakin berkembang pesat dibuktikan dengan sudah sngat menjamurnya ritel di Indonesia, baik di kota-kota besar maupun daerah terpencil. Para pengusaha ritel baik lokal maupun asing berlomba-lomba mendirikan gerai ritel seperti Carrefour, Hypermart, Indomaret, Alfamart, Matahari, dll. Banyaknya ritel asing yang masuk ke Indonesia dikarenakan industri ritel Indonesia yang terbukti mempunyai daya tahan yang kuat dalam menghadapi krisis. Selain itu, Indonesia merupakan negara yang menduduki urutan keempat dengan jumlah penduduk terbesar di dunia yang menjadi daya tarik bagi para pengusaha ritel (Jayanti & Santoso, 2019).

Keberadaan ritel memiliki peran penting terhadap pendapatan negara. Ritel termasuk dalam jenis perdagangan besar dan eceran yang memiliki kontribusi sebesar 10,57% terhadap Produk Bruto Domestik (PDB) pada triwulan II tahun 2022 (Badan Pusat Statistik, 2022). Berikut merupakan grafik yang menunjukkan kontribusi ritel terhadap PDB.

**Gambar 20. Kontribusi Perdagangan Besar dan Eceran; Reparasi Mobil dan Sepeda Motor Dalam PDB Triwulanan Tahun 2020-2022 (persen)**



**Gambar 1. 1 Kontribusi Perdagangan Besar dan Eceran, Reparasi, dan Sepeda Motor Dalam PDB Triwulan Tahun 2020-2022**

Sumber: (Badan Pusat Statistik, 2022)

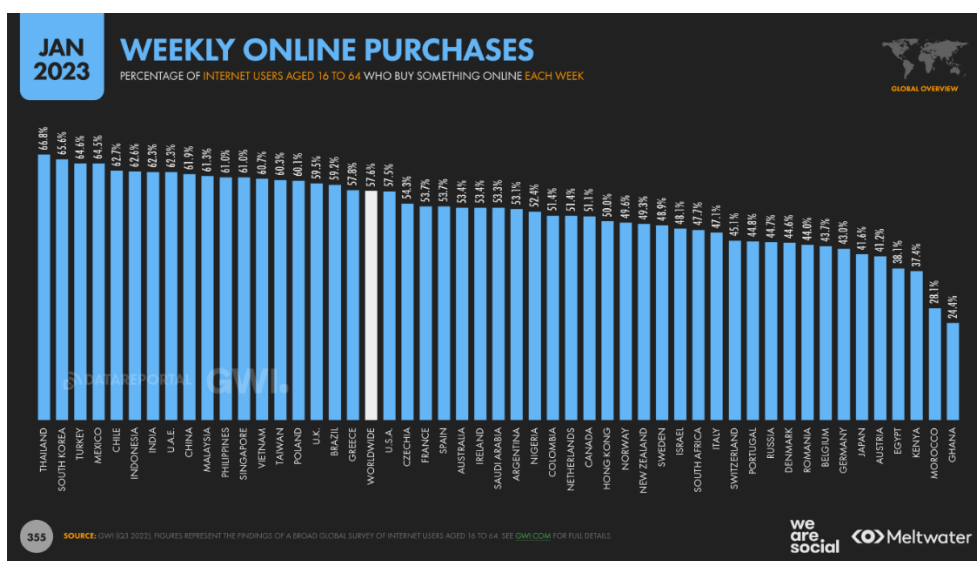
Namun, tak sedikit ritel yang tersebar di Indonesia menutup gerainya. Berdasarkan data pada Gambar 1. 2, selama tahun 2017-2021 jumlah gerai ritel di Indonesia semakin berkurang (Euromonitor International, 2022). Puncaknya pada tahun 2021, tercatat jumlah ritel di Indonesia mengalami penurunan sebanyak 11,85%. Hal ini dikarenakan kebijakan Pembatasan Sosial Berskala Besar (PSBB) di masa pandemi COVID-19 yang membuat gerai-gerai ritel sepi pengunjung dan berujung menutup gerai. Selain itu, pergeseran pola perilaku konsumen yang memilih berbelanja secara *online* terutama di kota-kota besar dan semakin ketatnya persaingan usaha menjadi penyebab perguncangan pada sektor ritel (Utami, 2018).



**Gambar 1. 2 Jumlah Ritel di Indonesia**

Sumber: (Euromonitor International, 2022)

Berdasarkan data yang diperoleh dari *We Are Social*, Indonesia menduduki posisi enam besar negara dengan pengguna internet yang melakukan pembelian produk atau jasa secara *online* setiap minggunya dengan persentase sebesar 62,6% yang meningkat sebesar 3,3% dibandingkan tahun sebelumnya. Belanja kebutuhan sehari-hari juga meningkat sebesar 5,8% dari tahun sebelumnya menjadi 38,1%. Adapun alasan terbesar yang mendorong pembelian secara *online* yaitu pengiriman gratis, pemberian kupon dan diskon, ulasan dari konsumen lain, dan *checkout online* yang mudah dan cepat.



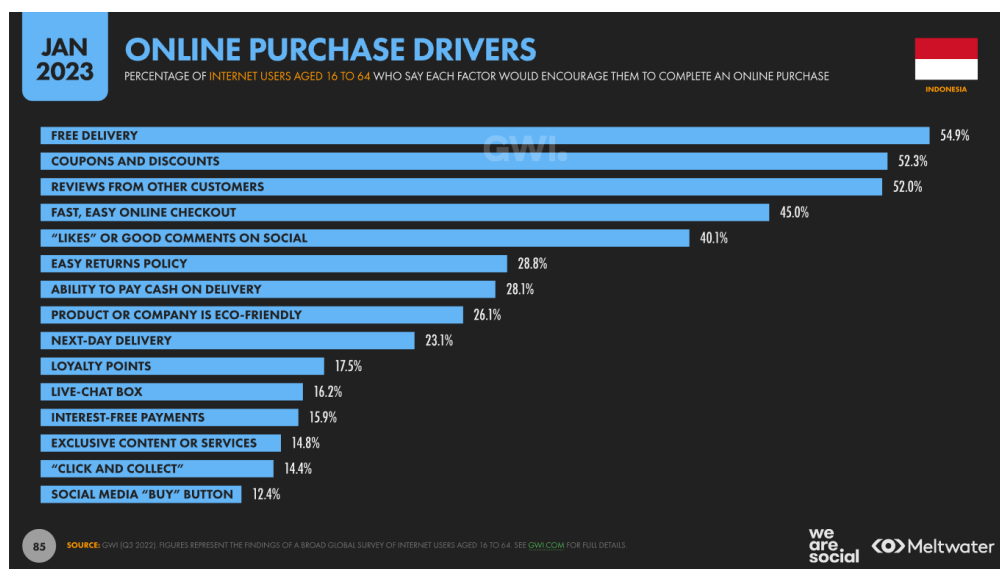
Gambar 1. 3 Pembelian *Online* Mingguan

Sumber: (We Are Social, 2023)



Gambar 1. 4 Aktivitas Belanja *Online* Mingguan

Sumber: (We Are Social, 2023)



Gambar 1. 5 Faktor Pendorong Belanja *Online*

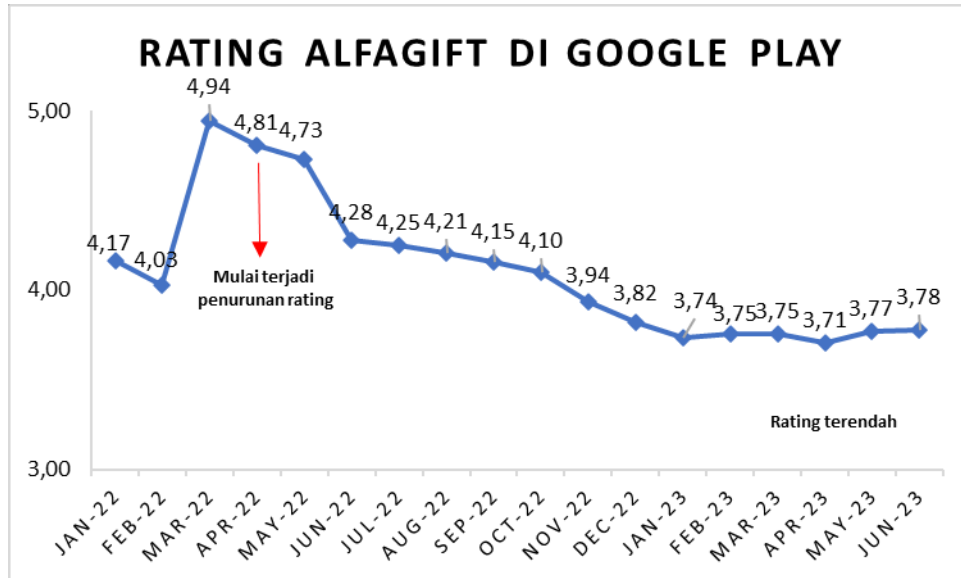
Sumber: (We Are Social, 2023)

Berdasarkan data tersebut menunjukkan pentingnya bagi ritel untuk mengadopsi aplikasi berbelanja *online* untuk dapat menjangkau konsumen lebih luas lagi. Di Indonesia sudah banyak ritel turut serta meluncurkan aplikasi untuk memudahkan konsumen membeli produk secara *online* seperti Indomaret, Alfamart, Hypermart, Alfamidi, Superindo, dll. *Alfamart* merupakan salah satu ritel yang telah meluncurkan aplikasi berbelanja *online* bernama *Alfagift*. *Alfagift* menawarkan produk-produk kebutuhan sehari-hari yang dijual di ritel *Alfamart* seperti, makanan, minuman, bahan masakan, produk kecantikan dan kesehatan, hingga pembelian pulsa dan paket data.

Namun, berdasarkan perolehan *rating* dan *review* pengguna, tingkat kepuasan pengguna aplikasi ini masih tergolong rendah. *Rating* merupakan representasi dari kepuasan konsumen, dimana semakin tinggi *rating* merefleksikan tingkat kepuasan terhadap produk atau layanan. *Review* merupakan komentar secara tekstual konsumen yang menggambarkan pengalaman atau kepuasannya terhadap suatu produk atau layanan (Kamisa et al., 2022). Kepuasan pelanggan menurut Kotler merupakan perasaan senang atau kecewa seseorang yang muncul karena membandingkan kinerja nyata yang dipersepsikan terhadap produk, dibandingkan ekspektasi pelanggan sebelum menggunakan produk (Nurhalimah & Nurhayati, 2019). Kepuasan pelanggan merupakan

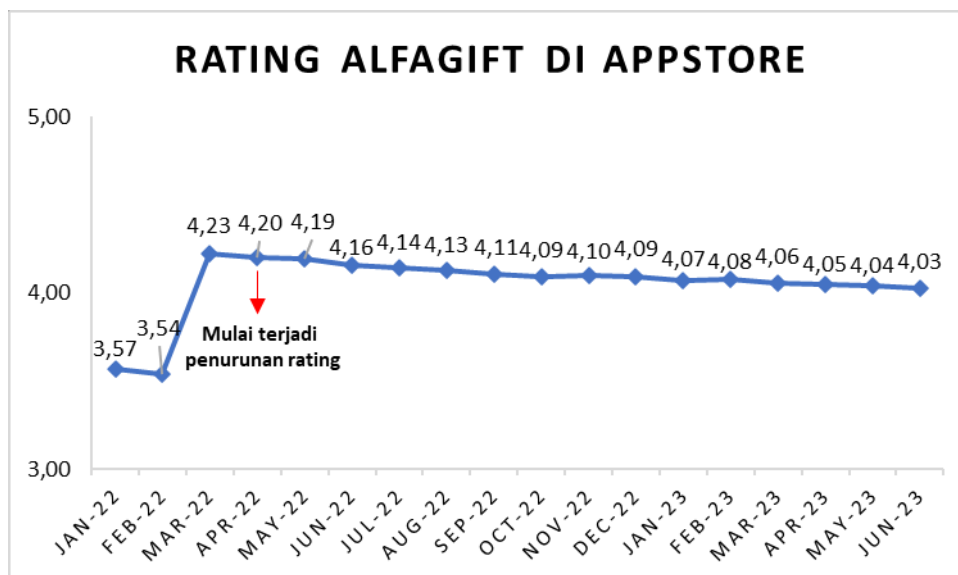
hal penting bagi eksistensi suatu bisnis serta, memuaskan pelanggan juga dapat meningkatkan kualitas bisnis dalam bersaing dengan bisnis-bisnis sejenis (Nurhalimah & Nurhayati, 2019). Menurut Kotler dan Armstrong, tingkat kepuasan pelanggan atau konsumen dapat diukur berdasarkan seberapa baik kinerja produk atau jasa dalam memenuhi harapan konsumen (Maramis et al., 2018). DeLone dan McLean yang melakukan pengembangan model konseptual untuk mengukur keberhasilan situs B2C berteori bahwa kualitas informasi, kualitas sistem, dan kualitas pelayanan merupakan faktor mendasar yang menentukan kepuasan pelanggan (Rinaldi & Santoso, 2018).

Pada situs *Google Play*, aplikasi ini hanya mendapat *rating* 3,7 dari skala 5 dibandingkan aplikasi sejenis lainnya seperti Klik Indomaret, My Superindo, Midi Kriing dengan *rating* di atas 4. Sedangkan pada situs *AppStore*, aplikasi ini mendapat *rating* yang sedikit lebih tinggi yaitu sebesar 4,0. Berdasarkan data *rating* pada Gambar 1.6 dan 1.7 yang diperoleh dari situs *App Follow* dalam kurun waktu setahun terakhir, aplikasi ini mengalami penurunan *rating* secara kontinu. *Rating* menggambarkan baik atau buruknya kualitas atau kinerja suatu produk (Harli & Mutasowifin, 2021). Hal ini menunjukkan bahwa, terjadi penurunan kualitas Aplikasi *Alfagift* yang dirasakan oleh pengguna baik dari segi kualitas sistem, informasi maupun layanan. Adapun beberapa permasalahan yang dirasakan oleh pengguna aplikasi ini yaitu tidak tersedia fitur *tracking* kurir, pengiriman lama, aplikasi lemot, pesanan tidak lengkap, *voucher* tidak bisa digunakan dan sistem aplikasi bermasalah. Selain itu transaksi sering kali gagal dan masalah teknis seperti *bug* (Octaviani et al., 2022).



Gambar 1. 6 Rating Alfagift di Google Play

Sumber: (App Follow, 2023)



Gambar 1. 7 Rating Alfagift di App Store

Sumber: (App Follow, 2023)

Oleh karena itu, perlu dilakukan evaluasi pada aplikasi *Alfagift* dalam upaya meningkatkan kepuasan pelanggan yaitu dengan meningkatkan kualitas baik dari segi sistem, informasi maupun layanan kepada pengguna sehingga dapat meningkatkan kepuasan pelanggan. Putri et al. (2022) dalam hasil penelitiannya mengenai pengaruh kualitas layanan elektronik *Alfagift* menunjukkan bahwa terdapat pengaruh *E-Service*

*Quality* terhadap kepuasan pelanggan. Sama halnya dengan kualitas informasi, semakin tinggi kualitas informasi yang dihasilkan suatu sistem informasi, maka kepuasan pengguna juga semakin meningkat (DeLone & Mclean, 1992). Berdasarkan hasil penelitian yang dilakukan oleh Octaviani et al. (2022), kualitas sistem dan kualitas informasi memiliki pengaruh terhadap kepuasan pelanggan.

Dalam upaya meningkatkan kepuasan pelanggan, metode *kansei engineering* digunakan dalam pengembangan produk untuk memperoleh kepuasan pelanggan, yaitu dengan menganalisis perasaan dan emosi manusia serta menghubungkan emosi dan perasaan tersebut ke dalam produk desain (Ginting & Hadiana, 2018). *Kansei engineering* dinilai memiliki keunggulan dibandingkan metode lain yang sejenis, dikarenakan kemampuannya menafsirkan kebutuhan emosional konsumen ke dalam parameter desain. Mitsuo Namagachi yang pertama kali mengenalkan *Kansei Engineering*, menyadari bahwa perasaan pengguna adalah bagian penting yang menjadi pertimbangan dalam mengembangkan produk (Faisal et al., 2021). Sehingga penggunaan *kansei engineering* ini untuk dapat memberikan rekomendasi dalam meningkatkan kualitas sistem, informasi, dan layanan berdasarkan kebutuhan emosional dari pelanggan.

*Kansei engineering* umumnya menggunakan kuesioner, survei, wawancara, dan *focus group meetings* untuk mengumpulkan, mengukur, dan mengevaluasi respons emosional pelanggan terhadap produk. Tetapi terdapat beberapa tantangan dalam pengumpulan data menggunakan metode *kansei engineering* tradisional tersebut yaitu:

1. Membutuhkan banyak waktu dan tenaga. Selain itu membutuhkan partisipasi aktif dari responden dan peneliti. Keterbatasan jumlah subjek dan produk menyulitkan untuk memperoleh umpan balik emosional yang komprehensif dan objektif (Liu et al., 2023).
2. Pertanyaan survei metode tradisional dirancang berdasarkan pemikiran ahli dan bukan berdasarkan pemikiran pelanggan sehingga responden hanya mampu secara pasif menanggapi pertanyaan. Selain itu responden mungkin tidak menjadi konsumen produk sasaran (Wang et al., 2018).
3. Hasil penelitian sulit diperbarui dikarenakan tingginya biaya tenaga kerja dan waktu. Saat produk baru memasuki pasar dengan cepat, kebutuhan dan preferensi konsumen terus berubah (Liu et al., 2023).
4. Tidak cocok untuk dilakukan secara *real-time* (Wang et al., 2018).

Oleh karena itu, pada penelitian ini pengumpulan data emosional pelanggan menggunakan metode *kansei engineering* memanfaatkan data ulasan pelanggan. Sebagian besar situs belanja *online* mengizinkan pelanggan untuk memberikan ulasan produk mereka setelah membeli suatu produk, dimana data tersebut langsung, *realtime*, dan terverifikasi dari sudut pandang pelanggan (Liu et al., 2023). Pada *platform* yang mendistribusikan aplikasi Alfabeta yaitu *Play Store (Google Play)* dan *App Store* juga menyediakan kolom ulasan *online*. Ulasan dari pengguna dapat digunakan sebagai alat untuk memperoleh informasi kelebihan dan kekurangan pada suatu produk atau jasa secara efisien dan efektif yang berguna bagi perusahaan untuk melakukan pengembangan. Namun, Banyaknya ulasan yang diberikan akan menyulitkan pembaca mendapatkan informasi tersebut. Membaca ulasan secara keseluruhan akan memakan waktu yang lama tetapi jika sedikit ulasan yang dibaca, evaluasi akan bias (Aryanti et al., 2019). Oleh karena itu, dibutuhkan metode untuk mengekstrak dan mengklasifikasikan ulasan menjadi informasi yang berguna bagi pembaca ulasan (Tsani et al., 2020).

Metode *text mining* merupakan metode yang dapat digunakan untuk mengatasi permasalahan tersebut. *Text mining* merupakan bagian dari *data mining* yang akan mengeksplor dan menganalisis data atau teks maupun dokumen dengan jumlah yang sangat besar (Munthe et al., 2022). Salah satu pendekatan yang digunakan pada *Text Mining* adalah *Machine Learning*. *Machine learning* merupakan salah satu cabang dalam *Artificial Intelligence* yang memiliki kemampuan untuk melakukan olah data berjumlah besar menjadi informasi yang dapat membantu dalam proses pengambilan keputusan (Diamendia & Setyowati, 2021). *Machine learning* menggunakan *training set* dan *test set* dalam melakukan klasifikasi (Akbari et al., 2017). Yudianto et al. (2022) dalam hasil penelitiannya membandingkan analisis sentimen menggunakan algoritma *Machine Learning* dan *Lexicon Based* menunjukkan bahwa algoritma *Machine Learning* memiliki akurasi jauh lebih baik dibandingkan pendekatan berbasis *Lexicon Based*.

Klasifikasi dalam *Machine Learning* adalah memberikan label pada sekumpulan data untuk mengklasifikasikan kelas yang tidak dikenal. Klasifikasi pada *Machine Learning* termasuk ke dalam kategori *Supervised Learning*. Beberapa algoritma klasifikasi yang ada pada *Machine Learning* yaitu *Decision Tree*, *Support Vector Machine*, *Random Forest*, *Naïve Bayes Classifier*, dan *K-Nearest Neighbor*. Pada penelitian ini menggunakan algoritma *Naïve Bayes Classifier*, *Support Vector Machine*, *Decision Tree*, dan *K-Nearest Neighbor*. Algoritma *Naïve Bayes Classifier* dipilih karena metode ini



sangat fleksibel jika ada perubahan pada data latih dan hasil dari klasifikasi metode ini cukup akurat (Sandi et al., 2019). Pattiha & Hendry (2022) dalam hasil penelitiannya analisis sentimen pada salah satu Badan Usaha Milik Negara (BUMN) dengan membandingkan metode K-NN, *Naïve Bayes Classifier*, dan *Decision Tree* menunjukkan metode *Naïve Bayes Classifier* memiliki tingkat akurasi lebih tinggi daripada kedua metode lainnya. Namun pada penelitian lain yang dilakukan Puspita & Widodo (2021) dalam membandingkan metode K-NN, *Decision Tree*, dan *Naïve Bayes* terhadap analisis sentimen pengguna Layanan BPJS dihasilkan tingkat akurasi tertinggi pada algoritma *Decision Tree*. *Decision Tree* memiliki kelebihan yaitu sangat cepat dalam mempelajari kumpulan data (Bhavitha et al., 2017). Kemudian Hendrayana et al. (2023) dalam penelitiannya membandingkan metode SVM, K-NN dan NBC pada analisis sentimen Wisata Alam Jatiluwih menunjukkan bahwa metode K-NN menghasilkan nilai akurasi dan presisi lebih unggul dibandingkan SVM dan NBC. K-NN dipilih karena mampu bekerja maksimal terhadap data yang mempunyai kesalahan acak (*noise*). Sehingga ketiga algoritma baik *Naïve Bayes Classifier*, *Decision Tree*, maupun *K-Nearest Neighbor* digunakan pada penelitian ini.

Algoritma *Support Vector Machine* (SVM) juga akan digunakan pada penelitian ini. Algoritma *Support Vector Machine* dipilih karena kemampuan menemukan *hyperlane* terbaik sebagai pemisah yang berfungsi untuk memisahkan dua kelas data berbeda seperti positif dan negative (Ubaidillah et al., 2019), dan meminimalkan *error* dalam data *training* (Ulfah & Anam, 2020). pada penelitiannya melakukan perbandingan algoritma *Naïve Bayes*, SVM, dan K-NN pada analisis sentimen *gadget* menunjukkan bahwa algoritma SVM menghasilkan akurasi lebih baik di antara ketiganya.

Integrasi *Machine Learning* dengan *Kansei Engineering* saja tidak cukup untuk mengetahui atribut mana yang memiliki dampak yang besar pada kepuasan pelanggan, sehingga dibutuhkan metode kano. Metode *Kano* digunakan untuk mengklasifikasikan kata-kata yang sudah disusun berdasarkan metode *Kansei Engineering* (Stiyono et al., 2022). Ada banyak pendekatan untuk memuaskan pelanggan tetapi model Kano adalah metode paling efektif yang saat ini digunakan oleh sebagian besar praktisi dan peneliti dalam berbagai aspek kepuasan pelanggan (Robert, 2016). *Kano* berfokus pada kebutuhan dan keinginan konsumen terhadap produk ataupun jasa yang akan dikembangkan, sementara *Kansei Engineering* merupakan sistem yang menerjemahkan kesan ke dalam parameter produk (Prabowo, 2019). Dengan mengintegrasikan *Machine*

*Learning* dengan metode *Kansei Engineering* dan *Kano* diharapkan dapat mengidentifikasi kebutuhan pengguna terhadap aplikasi sehingga dapat menjadi pendukung untuk mengembangkan aplikasi guna meningkatkan kepuasan pelanggan pada aplikasi *Alfagift*.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah pada penelitian ini sebagai berikut:

1. Aplikasi belanja *online* sangat memudahkan konsumen membeli produk secara *online*.

*Alfagift* merupakan salah satu aplikasi belanja *online* yang menawarkan produk-produk kebutuhan sehari-hari yang dijual di ritel *Alfamart* seperti, makanan, minuman, bahan masakan, produk kecantikan dan kesehatan, hingga pembelian pulsa dan paket data. Namun, berdasarkan perolehan *rating* dan *review* pengguna, tingkat kepuasan pengguna aplikasi ini masih tergolong rendah. Hal ini menunjukkan bahwa, terjadi penurunan kualitas Aplikasi *Alfagift* yang dirasakan oleh pengguna baik dari segi kualitas sistem, informasi maupun layanan. Dalam upaya meningkatkan kepuasan pelanggan akan menggunakan metode *Kansei Engineering* untuk dapat memberikan rekomendasi dalam meningkatkan kualitas sistem, informasi, dan layanan berdasarkan kebutuhan emosional dari pelanggan. Pengumpulan data emosional pelanggan pada tahapan *Kansei Engineering* menggunakan metode *Text Mining*. Metode *Text Mining* dapat mengeksplor dan menganalisis data atau teks maupun dokumen dengan jumlah yang sangat besar yang kemudian diklasifikasikan menggunakan algoritma *Machine Learning*. Klasifikasi menggunakan *Machine Learning* bertujuan memberikan label pada sekumpulan data untuk mengklasifikasikan kelas yang tidak dikenal. Penggunaan *Kansei Engineering* saja tidak cukup untuk mengetahui atribut mana yang memiliki dampak yang besar terhadap kepuasan pelanggan Sehingga dibutuhkan metode *Kano* untuk mengklasifikasikan atribut ke dalam kategori *Kano* agar dapat mengetahui atribut-atribut mana yang berdampak besar terhadap kepuasan pelanggan yang akan diprioritaskan untuk dilakukan peningkatan.

Pertanyaan: Bagaimana hasil integrasi *Machine Learning* dengan *Kansei Engineering* dan *Kano* dalam upaya meningkatkan kepuasan pelanggan Aplikasi *Alfagift*?

### 1.3 Tujuan Penelitian

Menganalisis hasil integrasi *Machine Learning* dengan *Kansei Engineering* dan *Kano* dalam upaya meningkatkan kepuasan pelanggan Aplikasi Alfagift.

### 1.4 Manfaat Penelitian

Manfaat pada penelitian ini adalah sebagai berikut:

#### 1. Bagi Perusahaan

- a. Hasil penerapan *Text Mining* dan *Machine Learning* dapat membantu perusahaan untuk mengetahui tanggapan pelanggan mengenai aplikasi secara cepat dan informatif.
- b. Hasil penerapan *Kansei Engineering* dapat membantu untuk mengidentifikasi kebutuhan pelanggan dengan menerjemahkan perasaan pelanggan terhadap aplikasi.
- c. Hasil penerapan *Kano* dapat membantu perusahaan mengidentifikasi atribut yang memiliki dampak yang besar terhadap kepuasan pelanggan yang diprioritaskan untuk dilakukan peningkatan.

#### 2. Bagi Masyarakat

Penelitian ini diharapkan dapat menjadi referensi untuk perbaikan kualitas sistem, informasi, dan layanan pada aplikasi belanja *online*, sehingga dapat meningkatkan kepuasan pelanggan dan masyarakat yang tertarik menggunakan aplikasi belanja *online* semakin bertambah.

#### 3. Bagi Akademisi

- a. Dapat digunakan sebagai referensi selanjutnya terkait dengan upaya peningkatan kepuasan pelanggan dengan integrasi *Machine Learning*, *Kansei Engineering*, dan *Kano*.

### 1.5 Batasan Penelitian

Adapun batasan masalah pada penelitian ini, yaitu sebagai berikut:

1. Data ulasan pengguna aplikasi Alfagift diambil pada situs *Google Play* dan *AppStore*.
2. Data ulasan aplikasi *Alfagift* yang diambil dari tanggal 6 April 2022 – 30 April 2023 dengan menggunakan versi 4.9.0 hingga 4.17.0
3. Pengolahan data ulasan menggunakan metode *Text Mining* dengan algoritma *Machine Learning*.

4. Kriteria responden kuesioner *kano* yaitu pernah menggunakan aplikasi Alfagift dan pernah berbelanja melalui aplikasi Alfagift.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Kajian Literatur

Puspita & Widodo (2021) melakukan penelitian berjudul “Perbandingan Metode *K-NN*, *Decision Tree*, dan *Naïve Bayes* terhadap Analisis Sentimen Pengguna Layanan BPJS”. Penelitian ini bertujuan untuk mengetahui keluhan masyarakat Indonesia terkait layanan BPJS dengan analisis sentimen. Metode yang digunakan pada penelitian ini yaitu *K-NN*, *Decision Tree*, dan *Naïve Bayes*. Hasil dari penelitian ini menunjukkan bahwa metode *Decision Tree* memiliki tingkat akurasi lebih tinggi dibandingkan *K-NN* dan *Naïve Bayes* dengan tingkat akurasi sebesar 96,13%.

Iskandar & Nataliani (2021) dalam penelitiannya berjudul “Perbandingan *Naïve Bayes*, *SVM*, dan *K-NN* untuk Analisis Sentimen Gadget Berbasis Aspek” bertujuan mengetahui sentimen dari sebuah produk *gadget* Samsung Galaxy Z Flip 3. Metode yang digunakan yaitu *Naïve Bayes*, *Support Vector Machine* dan *K-NN*. Hasil dari penelitian ini menunjukkan bahwa model klasifikasi *Support Vector Machine* memiliki hasil terbaik di antara kedua metode lainnya dengan rata-rata akurasi *SVM* sebesar 96,43% dilihat dari empat aspek, yaitu aspek desain sebesar 94,40%, aspek harga sebesar 97,44%, aspek spesifikasi sebesar 96,22%, dan aspek citra *merk* sebesar 97,63%.

Restuputri et al. (2021) melakukan penelitian berjudul “*The Effect of Logistic Service Quality on Customer Satisfaction and Loyalty Using Kansei Engineering during the COVID 19 Pandemic*”. Penelitian ini mengidentifikasi pengaruh kualitas layanan logistik di Indonesia terhadap kepuasan dan loyalitas pelanggan selama pandemi COVID-19. Metode yang digunakan yaitu *Kansei Engineering* dan *Structural Equation Model* (SEM). Hasil dari penelitian menunjukkan bahwa kualitas pelayanan operasional berpengaruh signifikan terhadap kepuasan pelanggan sehingga peningkatan kualitas layanan operasional sangat diperlukan. Dalam penelitian ini dapat dibuktikan bahwa metode *Kansei Engineering* dapat menggambarkan keinginan konsumen sehingga dapat

mengidentifikasi ekspektasi konsumen terhadap kualitas layanan logistik di masa pandemi COVID-19.

Pattiiha & Hendry, (2022) dalam penelitiannya berjudul “Perbandingan Metode *K-NN*, *Naïve Bayes*, *Decision Tree* untuk Analisis Sentimen *Tweet Twitter* Terkait Opini Terhadap PT PAL Indonesia” melakukan analisis sentimen terhadap opini dari masyarakat mengenai kinerja dan pelayanan yang diberikan oleh PT PAL Indonesia pada media sosial Twitter. Metode yang digunakan yaitu *Naïve Bayes*, *K-NN*, dan *Decision Tree*. Hasil dari penelitian menunjukkan bahwa metode *Naïve Bayes* memiliki tingkat akurasi paling tinggi di antara kedua metode lainnya dengan tingkat akurasi sebesar 84,08%.

Barrios-Ipenza et al. (2021) melakukan penelitian berjudul “*Quality Evaluation of Health Services Using the Kano Model in Two Hospitals in Peru*”. Penelitian ini bertujuan untuk mengevaluasi kualitas pelayanan kesehatan di dua Rumah Sakit PPP di Peru, mengidentifikasi dan mengklasifikasikan atribut yang menentukan kualitas layanan, dan menggunakan klasifikasi tersebut untuk memandu tindakan perbaikan dalam manajemen rumah sakit. Metode yang digunakan yaitu *Kano Model*. Hasil dari penelitian menunjukkan bahwa keberadaan sebageian besar atribut yang dievaluasi relevan untuk menjaga tingkat kepuasan pengguna dan tidak adanya atribut ini menimbulkan ketidakpuasan pada pengguna. Kemudian hasil evaluasi pengguna terhadap layanan kesehatan bersifat *multidimensional*.

Putu et al. (2021) dalam penelitiannya berjudul “Analisis Sentimen dan Pemodelan Topik Pariwisata Lombok Menggunakan Algoritma *Naïve Bayes* dan *Latent Dirichlet Allocation*” melakukan klasifikasi sentimen opini-opini wisatawan menjadi kelas positif dan negatif serta melakukan pemodelan topik pada kedua kelas tersebut. Metode yang digunakan yaitu *Naïve Bayes* dan *Latent Dirichlet Allocation* (LDA). Hasil penelitian ini didapatkan kinerja model algoritma *Naïve Bayes* dengan nilai akurasi, presisi, recall, dan spesifisitas masing-masing sebesar 92%, 100%, 83,84%, dan 100%. Penerapan algoritma *Naïve Bayes* dan LDA dinilai efektif untuk analisis sentimen serta pemodelan topik untuk pariwisata lombok.

Andriani et al. (2021) dalam penelitiannya berjudul “*Improving Quality Using the Kano Model in Overcoming Competition in The Service Industry*” yang bertujuan untuk mengidentifikasi ketidakpuasan pelanggan terhadap kualitas pelayanan hotel dan memberikan usulan desain untuk meningkatkan kualitas pelayanan. Metode yang

digunakan yaitu *Kano Model*. Hasil dari penelitian yaitu diperoleh 21 atribut pada kategori *must be*, 8 atribut pada kategori *one dimensional*, 2 pada kategori *attractive*, dan 2 pada kategori *indifferent* dimana atribut yang dikategorikan *one dimensional*, *attractive*, dan *must be* merupakan atribut yang diprioritaskan untuk meningkatkan kualitas pelayanan. Terdapat 18 atribut sebagai usulan desain dalam peningkatan kualitas pelayanan.

Nguyen & Nguyen (2021) melakukan penelitian berjudul “*Improvement of Service Quality in the Supply Chain of Commercial Banks—A Case Study in Vietnam*”. Penelitian ini dilakukan untuk meningkatkan kualitas layanan rantai pasok di Bank Umum di Vietnam dengan mengintegrasikan *SERVQUAL*, *Binary Logistic model* dan *Grey Forecasting Model*. Hasil penelitian ini yaitu penerapan ketiga model yang digunakan menunjukkan faktor-faktor yang mempengaruhi kepuasan nasabah yang melakukan deposito tabungan di bank umum serta faktor-faktor yang mempengaruhi probabilitas penarikan modal lebih awal dari tabungan umum.

Indrawati et al. (2020) dalam penelitiannya berjudul “*Lean Concept Development in Fast Food Industry Using Integration of Six Sigma and TRIZ Method*” melakukan pengembangan untuk meningkatkan sistem pelayanan pada restoran cepat saji. Metode yang digunakan adalah integrasi antara *Six Sigma* dan *Theory of Inventive Problem Solving* (TRIZ). Hasil dari penelitian yaitu terjadi peningkatan waktu sebesar 90% dengan mengimplementasikan strategi yang telah didapatkan.

Hameed et al. (2022) dalam penelitiannya berjudul “*Sustainable Product Development Using FMEA ECQFD TRIZ and Fuzzy TOPSIS*” melakukan pengembangan produk yang berkelanjutan dengan menggabungkan metode *Failure Modes and Effects Analysis* (FMEA), *Quality Function Deployment* (QFD), *Theory of Inventive Problem Solving* (TRIZ), *Life Cycle Evaluation* (LCA), dan *Fuzzy TOPSIS*. Hasil dari penelitian ini yaitu kerangka kerja yang dikembangkan untuk pengembangan produk berkelanjutan berhasil mengintegrasikan FMEA, ECQFD, TRIZ, dan TOPSIS Fuzzy. Penelitian ini juga membuktikan kelayakan model kerangka kerja yang diusulkan, yang mencakup berbagai aspek berkelanjutan untuk mendukung pengembangan produk inovatif.

Laksono et al. (2019) melakukan penelitian berjudul “*Sentiment Analysis of Restaurant Customer Reviews on TripAdvisor using Naive Bayes*”. Penelitian bertujuan untuk meningkatkan layanan restoran kepada konsumen secara berkelanjutan dengan menganalisis ulasan konsumen di 10 restoran terbaik di Surabaya dari situs *TripAdvisor*.

Metode yang digunakan yaitu analisis sentimen dengan membandingkan algoritma *Naïve Bayes* dan *TextBlob*. Hasil dari penelitian ini menunjukkan bahwa kedua metode tersebut mendapatkan respon pelanggan secara akurat dimana metode *Naïve Bayes* lebih akurat daripada *TextBlob* dengan perbedaan akurasi sebesar 2,9%.

Priyantina & Sarno (2019) melakukan penelitian berjudul “*Sentiment Analysis of Hotel Reviews Using Latent Dirichlet Allocation, Semantic Similarity and LSTM*”. Metode yang digunakan yaitu Analisis Sentimen, *Latent Dirichlet Allocation* (LDA), *Semantic Similarity*, dan *Long-Short Term Memory* (LSTM). Penelitian ini bertujuan untuk mengetahui kinerja terbaik pada masing-masing metode analisis dan menganalisis ulasan pelanggan untuk mengetahui aspek hotel mana yang membuat pelanggan puas dan tidak puas berdasarkan 5 aspek hotel yang telah ditentukan sehingga perusahaan dapat merencanakan perubahan, mengatur ulang bisnis, dan merevisi kesalahan. Hasil dari penelitian menunjukkan bahwa metode yang diusulkan dapat mengklasifikasikan ulasan ke dalam lima aspek hotel dimana kinerja kategorisasi aspek tertinggi diperoleh dengan menggunakan LDA+TF-ICF 100% + *Semantic Similarity* yang mencapai 85% dan sentimen kinerja klasifikasi untuk analisis sentimen berbasis aspek tertinggi diperoleh dengan menggunakan *Word Embedding* + LSTM yang mencapai 93%.

Swarnakar et al. (2020) melakukan penelitian berjudul “*A QFD-TISM Approach for Service Quality Improvement in Polytechnic Education Institutes: A Case Study*” yang bertujuan untuk meningkatkan kualitas layanan *Polytechnic Education Institutions* (PEI) yang memungkinkan pembuatan kebijakan untuk mengatasi masalah rendahnya tingkat partisipasi dan pengangguran siswa. Metode yang digunakan dalam penelitian ini adalah *Quality Function Deployment* (QFD) dan *Total Interpretive Structural Modelling* (TISM). Hasil dari penelitian yaitu dari kesembilan parameter kualitas, terbentuknya visi, misi dan tujuan lembaga pendidikan dengan baik (DC1) dan mempertahankan sumber daya keuangan dan dukungan kelembagaan (DC9) merupakan parameter kualitas yang paling penting.

Altuntas & Kansu (2019) melakukan penelitian berjudul “*An Innovative and Integrated Approach Based on SERVQUAL, QFD and FMEA for Service Quality Improvement*”. Penelitian ini mengusulkan pendekatan inovatif dan terintegrasi berdasarkan *Service Quality Measurement* (SERVQUAL), *Quality Function Deployment* (QFD), dan *Failure Modes and Effects Analysis* (FMEA) untuk meningkatkan kualitas layanan pada sebuah rumah sakit di Turki dengan total responden penelitian sebanyak



390 pasien. Hasil penelitian menunjukkan bahwa pendekatan yang diusulkan dapat digunakan secara efektif untuk menilai kualitas layanan dalam praktiknya dimana ditemukan bahwa kualitas layanan pada tiga item penting (item 12, 26, dan 28) melebihi harapan pasien, penampilan fisik dan peralatan medis yang modern memiliki nilai SERVQUAL (gap) tertinggi yaitu 1,149, dan pada karakteristik layanan 31 yaitu memberikan pelatihan hak pasien untuk dokter dan staf kesehatan memiliki tingkat kepentingan relatif tertinggi di antara 37 karakteristik layanan. Selain itu, ditemukan bahwa mode kegagalan 2, 7 dan 11 memiliki prioritas tertinggi risiko di antara dua belas penyebab kegagalan potensial.

Muafi & Sanjaya (2018) melakukan penelitian yang berjudul “*Incorporating Kansei Engineering into Service Quality Tools to Improve the Airline Services*” untuk mengembangkan strategi *Airline X* dalam rangka meningkatkan kualitas pelayanan. Penelitian ini menggunakan mengintegrasikan metode *Kansei Engineering* dengan model SERVQUAL, Kano, dan *Quality Function Deployment (QFD)*. Data diambil dengan menyebarkan kuesioner kepada konsumen yang pernah menggunakan *Airline X* setelah bergabung dengan aliansi *Sky Team* dan responden juga merupakan konsumen yang menggunakan *Airline X* di 5 bandara internasional di Indonesia yaitu 119 responden dari Bandara Internasional Soekarno-Hatta, Jakarta, 36 responden dari Bandara Internasional Juanda Surabaya, 31 responden dari Bandara Internasional Ngurah Rai Denpasar, 20 responden Bandara Internasional Sultan Hassanudin Makassar, dan 17 responden dari Bandara Internasional Kuala Namu Medan. Hasil dari penelitian ini adalah rencana aksi untuk meningkatkan kualitas pelayanan *Airline X* seperti aliansi maskapai, kelas kursi, *brand identity*, sistem informasi modern, dan *expert consultation*.

Qi (2021) melakukan penelitian yang berjudul “*Research on The Design of E-Education Application Interface Based on Kansei Engineering*” yang bertujuan untuk menganalisis faktor-faktor diperlukan dalam perancangan antarmuka *Wastu Mobile*. Metode yang digunakan pada penelitian ini yaitu *Kansei Engineering*. Hasil dari penelitian ini yaitu usulan desain antarmuka aplikasi dari keseluruhan perasaan, pencocokan warna, tata letak antarmuka, dan pengalaman pengguna.

Suhaimi & Lokman (2018) melakukan penelitian yang berjudul “*A Kansei Approach to Investigate Mobile Apps Requirements for Muslim Women Tourist*” yang bertujuan untuk mengetahui kebutuhan aplikasi *mobile* untuk turis Muslimah. Metode yang digunakan pada penelitian ini yaitu *Kansei Engineering*. Hasilnya dapat dijadikan konsep

desain dalam menghasilkan aplikasi yang dapat membantu meningkatkan pengalaman emosional dan kepuasan wisatawan muslimah selama berwisata.

Putra & Suzianti (2022) melakukan penelitian yang berjudul “*Design of a Food Sharing App Using Kansei Engineering and Fuzzy Linguistic Methods*”. Tujuan dari penelitian ini adalah membuat antarmuka aplikasi *food sharing* dengan pendekatan *Kansei Engineering* untuk memperoleh emosi pengguna terhadap rancangan yang diusulkan, yang kemudian dilengkapi dengan *Fuzzy Linguistics* untuk memperjelas kebutuhan fitur pengguna sebagai bentuk dukungan terhadap terciptanya pengembangan keberlanjutan. Berdasarkan hasil pengujian *prototype*, disarankan aplikasi dengan “konsep keren” karena memiliki efisiensi lebih tinggi dan pengalaman pengguna lebih baik.

Sanja et al. (2022) melakukan penelitian yang berjudul “*Designing The BOC Bank App Using Kansei Engineering*” menyajikan bagaimana *Kansei Engineering* dapat diterapkan untuk *mobile banking* meningkatkan kegunaan, pengalaman pengguna, visualisasi emosional dan psikologis pada antarmuka pengguna. Hasil penelitian ini mampu menentukan bahwa KE merupakan salah satu alat penting yang terampil dengan kemampuan memilih elemen desain dan mengubahnya sehubungan dengan berbagai faktor kunci yang hidup berdampingan.

Nugroho et al. (2019) melakukan penelitian yang berjudul “*Design of Wastu Mobile Interface Using Kansei Engineering*”. Tujuan dari penelitian ini adalah untuk menganalisis faktor-faktor tersebut diperlukan dalam perancangan antarmuka Wastu Mobile dengan pendekatan *Kansei Engineering*, penerapan metode AHP dan SAW dalam pemilihan spesimen. Metode yang digunakan pada penelitian ini yaitu *Kansei Engineering*, *Analytical Hierarchy Process (AHP)* and the *Simple Additive Weighting (SAW)*. Penelitian ini menghasilkan rekomendasi perancangan antarmuka *Wastu Mobile* dengan pendekatan *Kansei Engineering* yaitu berupa matriks rekomendasi elemen desain berdasarkan kombinasi konsep emosi “menarik” dan “keren”, kombinasi emosi “mudah” dan “keren”. gunakan" dan "sederhana".

Malinka et al. (2022) dalam penelitiannya yang berjudul “*Prioritization of Quality Principles for Health Apps Using the Kano Model: Survey Study*” bertujuan untuk membantu memanfaatkan sumber daya yang terbatas (misalnya waktu) dengan lebih baik dengan membantu pengambilan keputusan mengenai prinsip-prinsip mutu tertentu yang patut diprioritaskan dalam praktik medis sehari-hari dan prinsip-prinsip yang dapat

diberikan prioritas lebih rendah, bahkan dalam kasus di mana prinsip-prinsip keseluruhan dinilai sama. Metode yang digunakan pada penelitian ini yaitu metode Kano. Hasil penelitian ini menunjukkan metodologi yang disajikan di sini merupakan pilihan yang menarik untuk menentukan prioritas, dan memungkinkan fokus pada kriteria yang paling penting, sehingga menghemat waktu yang berharga ketika meninjau aplikasi untuk digunakan dalam bidang medis, bahkan dengan hasil kategorisasi yang sebagian besar serupa.

Kim et al. (2018) melakukan Penelitian yang berjudul “*The Study on the e-Service Quality Factors in m-Shopping Mall App based on the Kano Model*”. Penelitian ini bertujuan untuk mengklasifikasikan dimensi kualitas layanan aplikasi belanja seluler menggunakan model Kano. Hasil dari penelitian ini yaitu koefisien kepuasan menunjukkan kesan baik, respon hasil cepat, pengiriman cepat, dan koefisien kurang memuaskan menunjukkan lebih banyak ketertarikan terhadap informasi pribadi seperti keamanan metode pembayaran, dan keamanan transaksi. Metode yang digunakan yaitu metode Kano. Dalam kualitas informasi, informasi yang berlebihan diklasifikasikan sebagai komponen kualitas yang apatis, sedangkan penyediaan informasi terkait termasuk dalam komponen kualitas yang menarik. Dalam kualitas keandalan, penyediaan layanan yang disesuaikan diklasifikasikan sebagai komponen kualitas yang menarik. Pada konektivitas instan, kualitas sambungan selama pengangkutan tergolong komponen kualitas yang menarik. Dalam kualitas kenyamanan, akses terhadap informasi produk diklasifikasikan sebagai komponen kualitas satu arah. Seluruh komponen mutu desain tergolong dalam komponen mutu menarik, dan pada mutu keamanan seluruh komponennya tergolong dalam satu komponen mutu. Terakhir, dalam layanan pelanggan, semua komponen diklasifikasikan sebagai satu komponen kualitas.

R. A. S. Putra & Priyanto (2021) melakukan penelitian yang berjudul “*Kano Model Analysis of Android Apps Quality from End User’s Preferences*”. Penelitian ini bertujuan untuk mengeksplorasi preferensi pengguna akhir mengenai fitur-fitur aplikasi android dengan mengklasifikasikannya menggunakan Model Kano. Hasil dari penelitian ini adalah terdapat 17 fitur yang tergolong dalam kategori kebutuhan *Attractive*, enam fitur yang tergolong dalam kategori *One Dimensional*, satu fitur yang tergolong dalam kategori *Must-be*, dan lima fitur yang tergolong dalam kategori *Indifferent*. Secara keseluruhan, tampaknya banyak fitur yang berpotensi meningkatkan kepuasan pengguna akhir.

Barutçu & Elif (2018) melakukan penelitian yang berjudul “*Expectations from Personal mHealth Apps through Kano's Model*”. Tujuan utama dari penelitian ini adalah untuk berkontribusi pada pengembangan aplikasi *mHealth* secara lebih efektif dengan Model Kano, yang mendefinisikan ekspektasi pengguna dengan lebih baik dan mengukur dampak ekspektasi tersebut terhadap kepuasan. Penelitian ini menghasilkan 22 karakteristik desain untuk aplikasi *mHealth* yang efisien diidentifikasi beserta kepentingan relatifnya bagi pengguna dengan menggunakan Model Kano yang memprioritaskan pengguna harapan. Terdapat 9 atribut kategori *One Dimensional*, 6 kategori *Attractive*, dan 7 atribut kategori *Indifferent*. Sehingga 15 atribut yang masuk ke dalam kategori *One Dimensional* dan *Attractive* ini harus dipertimbangkan oleh para desainer saat ini mengembangkan aplikasi *mHealth*.

Ho & Tzeng (2021) melakukan penelitian yang berjudul “*Using The Kano Model to Analyze The User Interface Needs of Middle-Aged and Older Adults in Mobile Reading*” meneliti mengenai 14 aplikasi membaca seluler yang menargetkan pengguna di Taiwan dan mengevaluasi kebutuhan antarmuka pengguna bagi orang dewasa paruh baya dan lanjut usia menggunakan metode Kano. Hasil penelitian ini mengungkapkan bahwa gaya *font* merupakan atribut antarmuka usability *One Dimensional* untuk orang dewasa yang lebih tua, yang lebih menyukai desain antarmuka tombol dan *list*.

Cai et al. (2023) melakukan penelitian yang berjudul “*Integrated Framework of Kansei Engineering and Kano Model Applied to Service Design*”. Penelitian ini mengusulkan kerangka desain layanan berdasarkan model *Kansei Engineering* (KE) dan *Kano* untuk merancang layanan yang berorientasi pada persepsi pelanggan dengan studi kasus layanan penerbangan maskapai penerbangan Tiongkok. Hasilnya mengungkapkan adanya hubungan antara persepsi pelanggan dan karakteristik layanan sehingga direkomendasikan agar maskapai penerbangan memasukkan metode ini ke dalam proses desain layanan mereka.

Hartono (2020) dalam penelitiannya melakukan pendekatan berbasis *Kansei Engineering* yang diintegrasikan dengan *TRIZ* dan *Kano Model* untuk memahami dan memenuhi kebutuhan emosional pelanggan. Penelitian ini menghasilkan panduan bagi manajemen dan staf bandara internasional untuk memprioritaskan atribut layanan yang akan terus ditingkatkan dan ditingkatkan sesuai *kansei* pelanggan, sumber daya tersedia, dan strategi inovatif.

Ginting & Hadiana (2018) melakukan penelitian yang berjudul “*Interface and Service Analysis on Student Website Using Kansei Engineering and Kano*”. Penelitian ini bertujuan untuk membuat rekomendasi desain antarmuka *website* mahasiswa dan saran

layanan yang perlu ditingkatkan dan dipertahankan pada *website* mahasiswa menggunakan metode *Kansei Engineering* dan *Kano*. Penelitian ini menghasilkan satu faktor utama yang mempengaruhi antarmuka yaitu “fresh” dan untuk atribut layanan prioritas pertama yang perlu ditingkatkan terdapat 1 atribut kegunaan, kualitas informasi dan dimensi interaksi layanan, atribut prioritas kedua yang perlu ditingkatkan 3 atribut kualitas informasi, 1 atribut dari interaksi layanan. Atribut prioritas ketiga yang perlu ditingkatkan satu atribut kegunaan dan satu atribut kualitas informasi. Prioritas pertama yang harus dipertahankan 3 atribut kualitas informasi dan interaksi layanan. Prioritas kedua adalah menjaga 7 atribut kegunaan, 4 atribut kegunaan interaksi layanan.

Kuo et al. (2020) melakukan penelitian berjudul “*Developing a Hand Sizing System For a Hand Exoskeleton Device Based on The Kansei Engineering Method*” menggunakan metode *Kansei Engineering* dan *Kano*. Penelitian ini bertujuan untuk mengidentifikasi kebutuhan pelanggan terhadap perangkat exoskeleton rehabilitasi tangan dan mengusulkan perbaikan. Hasil penelitian menunjukkan tiga faktor menarik yang sesuai dengan atribut desain hasil RUPSLB: sesuai (ukuran), fleksibel (dapat dikendalikan), dan ringan (berat)

Oboudi et al. (2020) melakukan penelitian berjudul “*Using Mixed Technique of Kansei Engineering, Kano Model, and Taguchi-based Experiment Design to Improve Satisfaction and Participation of Football Spectators at Stadiums*”. Penelitian ini bertujuan untuk mengetahui kebutuhan penonton di Liga Premier Sepak Bola Iran dengan menggunakan metode *Kansei Engineering*, *Kano*, dan *Taguchi*. Temuan menunjukkan bahwa otoritas olahraga Iran, khususnya Federasi Sepak Bola, dapat meningkatkan kepuasan dan kehadiran penonton di stadion jika mereka meningkatkan kondisi fisik dan keamanan stadion sepak bola dan menghilangkan hambatan yang ada termasuk kehadiran perempuan dan kehadiran keluarga di stadion. Klub sepak bola juga dapat membantu mencapai tujuan ini dengan mempekerjakan pemain dan pelatih yang berkualitas dan populer.



	al., 2021)												
6	(Putu et al., 2021)	Pariwi sata	√	√			√			√			
7	(Andr iani et al., 2021)	Ruma h Sakit		√		√							
8	(Ngu yen & Nguy en, 2021)	Bank Kome rsial		√								√	√
9	(Indr awati et al., 2020)	Indust ri Maka nan								√		√	
10	(Ham eed et al., 2022)	<i>Sustai nable Produ ct</i>			√		√			√	√	√	
11	(Laks ono et al., 2019)	Restor an	√	√					√			√	
12	(Priy antin a & Samo , 2019)	Hotel	√	√			√	√	√				
13	(Swar nakar	Institu si		√			√						

14	et al., 2020)	Pendi dikan Ruma h & Sakit Kans u, 2019)	√	√	√			
15	(Muan fi & Sanja ya, 2018)	Indust ri Pener banga n	√	√	√	√		
16	(Qi, 2021)	Aplik asi Belaja r			√			
17	(Suha imi & Lokm an, 2018)	Aplik asi Travel			√			√
18	(P. S. Putra & Suzia nti, 2022)	Aplik asi <i>Food</i> <i>Sharin</i> <i>g</i>			√			
19	(Sanj a et al., 2022)	Aplik asi Bank			√			
20	(Nugr oho	Aplik asi			√			√ √



	et al., 2019)	Unive rsitas		
2	(Mali nka	Aplik asi	√	
1	et al., 2022)	Keseh atan		
2	(Kim et al., 2018)	Aplik asi <i>Mobil e Shopp ing Mall</i>	√	
2	(R. A. S. Putra & Priya nto, 2021)	Aplik asi Andro & id	√	
2	(Baru tçu & Elif, 2018)	Aplik asi Keseh atan	√	
4	(Ho & Tzen g, 2021)	Aplik asi Memb aca	√	
2	(Cai et al., 2023)	Layan an maska pai	√	√
6				

27	(Hartono, 2020)	Layanan bandara			√	√							
28	(Ginting & Hadiana, 2018)				√	√							√
29	(Kuo et al., 2020)	Hand Exoskeleton Device			√	√							
30	(Oboudi et al., 2020)	Stadium Sepak Bola			√	√							
31	(Leadhi, 2023)	Aplikasi E-Grocery	√	√	√	√		√	√	√	√		

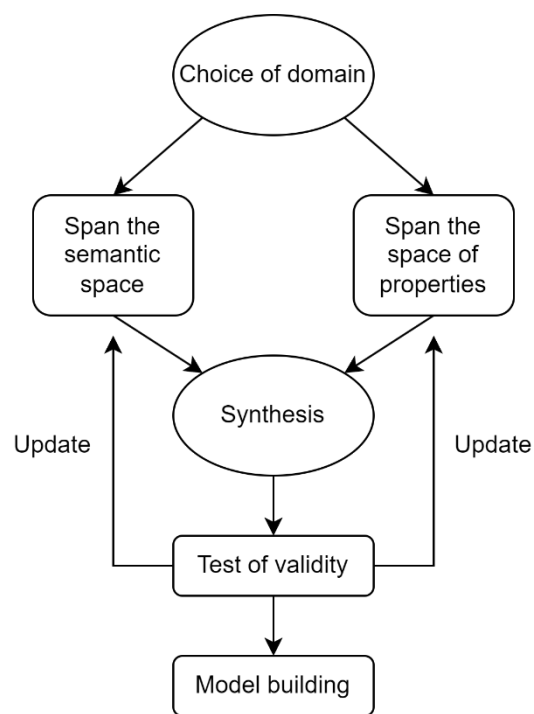
---

Berdasarkan kajian yang ditunjukkan pada Tabel 2. 1, beberapa penelitian terdahulu telah menggunakan metode *Kansei Engineering* dan *Kano* dalam meneliti objek penelitian. Pada penelitian ini terdapat perbedaan yaitu menambahkan metode *Text Mining* untuk pengumpulan data pada tahapan *Kansei Engineering*. Dalam klasifikasi data ulasan yang telah diekstraksi di tahapan *Text Mining*, menggunakan empat algoritma *Machine Learning* yaitu *Naïve Bayes*, *Support Vector Machine*, *K-Nearest Neighbor*, dan *Decision Tree*.

## 2.2 Landasan Teori

### 2.2.1 *Kansei Engineering*

Chen et al. (2015) dalam (Hartono et al., 2018) mendefinisikan *Kansei Engineering* sebagai sistem yang menerjemahkan pengguna layanan (*user's images*) dan perasaan ke dalam spesifikasi desain yang sesungguhnya. *Kansei* merupakan gambar dan perasaan psikologikal pengguna yang mendeskripsikan impresi pengguna secara umum dan emosi terhadap suatu layanan yang kemudian ditafsirkan ke dalam karakteristik layanan (Schütte et al., 2004). Hubungan antara emosi konsumen dan elemen layanan dapat menjadi acuan yang baik bagi penyedia jasa dan layanan untuk memberikan atribut layanan yang diinginkan konsumen dan memfasilitasi niat atau tujuan penggunaan layanan dari pelanggan (Hartono et al., 2018). Terdapat beberapa tahapan pada *Kansei Engineering* yang ditunjukkan pada gambar 2.6. Berikut merupakan alur proses dari *Kansei Engineering* (Schütte, 2005).



Gambar 2. 1 Alur Proses *Kansei Engineering*

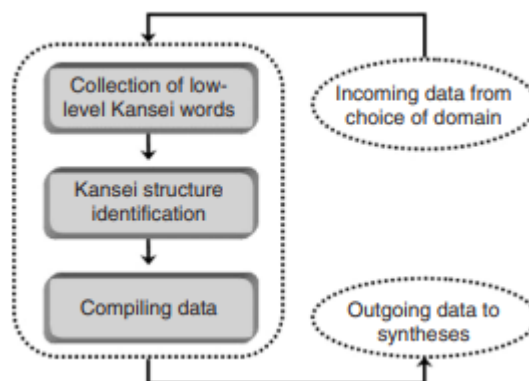
Sumber: (Schütte, 2005)

#### 1. Memilih *Domain*

Pada tahapan ini, domain menentukan kelompok pengguna, ceruk pasar, serta spesifikasi dari produk atau jasa baru.

## 2. Perentangan Ruang Semantik

Pada perentangan ruang semantik terdapat tiga langkah seperti yang disajikan pada Gambar 2.2. Menggunakan domain yang diinginkan sebagai titik awal, kansei tingkat rendah, biasanya kata sifat, dikumpulkan, mendeskripsikan produk yang dipertimbangkan secara semantik. Dari rangkaian deskripsi semantik ini, Kansei tingkat tinggi diidentifikasi dalam 'Identifikasi Struktur *Kansei*'. Dalam literatur Teknik *Kansei*, *Kansei* tingkat tinggi ini terkadang juga disebut sebagai '*Kansei Words*' atau '*Kansei Engineering Words*'. *Kansei words* merupakan representasi perasaan atau emosi yang sebenarnya dirasakan pelanggan ketika menggunakan suatu layanan tertentu. Dalam identifikasi *kansei words* dapat dilakukan melalui wawancara terhadap pelanggan, analisis perilaku dan reaksi pelanggan ataupun referensi penelitian lain pada bidang spesifik tertentu (Hartono et al., 2018). Terakhir, data disusun dengan cara yang terstandarisasi untuk memudahkan tahap sintesis berikutnya. Jika kata-kata *Kansei* penting terlewatkan dalam langkah ini, validitas hasilnya mungkin sangat terbatas. Oleh karena itu, lebih baik memilih beberapa kata lebih banyak dari yang diperlukan.

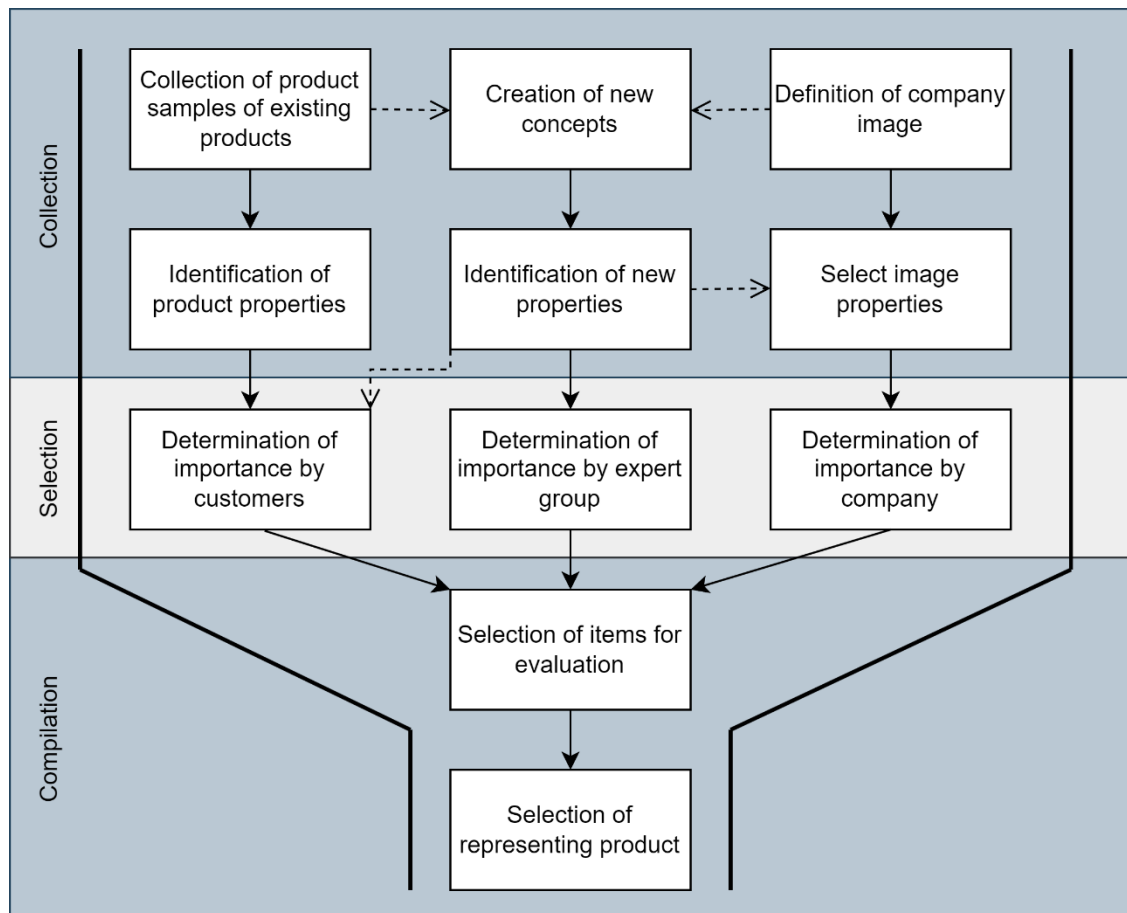


Gambar 2. 2 Perentangan Ruang Semantik

Sumber: (Schütte S. , 2005)

## 3. Perentangan Ruang Properti

Tahapan ini terdiri dari beberapa langkah yaitu i) mengumpulkan properti produk; ii) mengidentifikasi sifat-sifat penting dari kumpulan properti produk sebelumnya; iii) pemilihan akhir properti dan produk representatif.

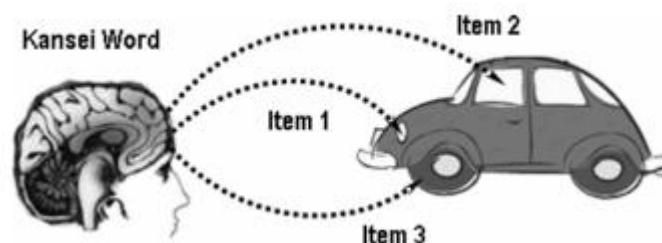


Gambar 2. 3 Tahapan Perentangan Ruang Properti

Sumber: (Schütte S. , 2005)

#### 4. Sintesis

Pada tahapan ini, ruang semantik dan ruang properti dihubungkan bersama seperti pada gambar 2.4. Untuk setiap kata Kansei, ditemukan properti produk yang memengaruhi kata Kansei. Ukuran dampak afektif dari properti produk pada setiap Kansei juga dihitung.



Gambar 2. 4 Fase Sintesis

Sumber: (Schütte S. , Eklund, Ishihara, & Nagamachi, 2008)

## 5. Pembuatan Model dan Uji Validitas

Model matematis atau non-matematis dibangun tergantung pada metode sintesis yang dipilih. Namun, sebelum menggunakan model sebagai model prediksi untuk produk masa depan, harus divalidasi. Saat ini, tidak hanya metode validasi untuk Diferensial Semantik yang tersedia, tetapi diperlukan konsep validasi yang lebih integratif.

*Kansei engineering* memiliki beberapa tipe. Adapun tipe-tipe dari *kansei engineering* yaitu sebagai berikut (Nagamachi & Lokman, 2011).

### 1. *Kansei Engineering Type I: a Category Classification*

*Kansei Engineering Type I* memecah kategori *kansei* dari suatu produk yang ditargetkan menjadi konsep yang lebih detail dan memperluas ke beberapa tingkatan, lalu diterjemahkan ke dalam karakteristik desain produk. Dalam melakukan identifikasi elemen desain produk yang akan dikembangkan, ditafsirkan melalui perasaan dan citra pengguna.

### 2. *Kansei Engineering Type II: Kansei Engineering Computer System*

*Kansei Engineering Type II* merupakan sistem rekayasa *kansei* berbantuan komputer. *Kansei Engineering System* (KES) adalah sistem terkomputerisasi dengan sistem pakar untuk menerjemahkan perasaan dan emosi pengguna ke dalam detail desain.

### 3. *Kansei Engineering Type III: Kansei Engineering Modelling*

Pada *Kansei Engineering Type III*, model matematika dibangun dalam menghubungkan konsumen dan elemen desain *kansei*. Contohnya yaitu Sanyo Co. dimana Fukushima dan rekannya di Sanyo melakukan penerapan *kansei engineering* pada *printer* berwarna. Dengan menggunakan *Fuzzy Kansei Logic*, mereka berhasil membangun *printer* warna cerdas yang memungkinkan perubahan warna asli menjadi lebih indah.

### 4. *Kansei Engineering Type IV: Hybrid Kansei Engineering*

*Kansei engineering type IV* terdiri dari dua metode yaitu *kansei engineering forward* dan *backward*. Pada *Kansei engineering forward*, konsumen menentukan produk berdasarkan *kansei word*, kemudian menggunakan computer diubah menjadi desain yang lebih sesuai. *Backward kansei engineering* berkaitan dengan tahapan mendesain dan mendownload di computer untuk menampilkan *kansei word*.

### 5. *Kansei Engineering Type V: Virtual Kansei Engineering*

*Kansei engineering type V* dalam mengumpulkan data menerapkan *virtual reality* yang merupakan teknologi yang memposisikan pengguna dalam lingkungan *virtual 3D*.

### 2.2.2 *Web Scraping*

*Web scraping* merupakan proses untuk mengekstraksi dokumen semi-terstruktur dari situs *web* untuk diambil data tertentu (Turland, 2010). *Web scraping* disebut juga “*web harvesting*”, “*web data extraction*” atau “*web data mining*”, dapat didefinisikan sebagai “konstruksi agen untuk mengunduh, mengurai, dan mengatur data dari *web* dalam sebuah cara otomatis.”. Alih-alih pengguna akhir mengklik *browser web* dan menyalin-menempelkan bagian-bagian yang menarik ke dalam, katakanlah, *spreadsheet*, *web scraping* memindahkan tugas ini ke program komputer yang dapat menjalankannya lebih cepat, dan lebih tepat, dibandingkan manusia (Broucke & Baesens, 2018).

### 2.2.3 *Text Mining*

*Text mining* adalah pemrosesan teks otomatis atau sebagian otomatis. ini melibatkan penerapan struktur pada teks dan mengekstraksi informasi yang relevan dari teks (Miller, 2005). *Text mining* memberikan sebuah set metodologi dan *tool* untuk menemukan (*discovering*), memvisualisasikan (*presenting*), mengevaluasi pengetahuan dari kumpulan teks dokumen yang besar (Purbo, 2019). *Text mining* disebut juga sebagai *data text mining*, yang kira-kira sebanding dengan analisis teks, merupakan proses untuk mendapatkan informasi berkualitas tinggi dari teks. ‘Kualitas tinggi’ pada *text mining* umumnya mengacu pada beberapa kombinasi dari relevansi, kebaruan, dan kemenarikan. *Text mining* bertujuan mengeksplorasi dan menganalisis data teks tidak terstruktur dalam jumlah besar dengan menggunakan perangkat lunak dengan kemampuan mengidentifikasi pola, konsep, topik, kata kunci, dan atribut lain dalam data. *Text mining* mencakup kategorisasi teks, *clustering* teks, ekstraksi konsep/ entitas, analisis sentimen, merangkum dokumen, dan pemodelan hubungan entitas (Purbo, 2019). Dalam memproses data dengan berjumlah sangat besar akan membutuhkan sumber daya yang tidak sedikit jumlahnya dan kaitannya dengan pengolahan data tersebut. Oleh karena itu diperlukan tahap pemrosesan awal atau *preprocessing* data agar data yang sebelumnya tidak terstruktur menjadi terstruktur sebelum melakukan tahapan *text mining*. Terdapat beberapa tahapan dalam *preprocessing data* yaitu (Feldman & Sanger, 2007):

1. *Data Cleaning* dan *Case Folding*

Proses *data cleaning* bertujuan untuk membersihkan ulasan dari *noise* yang tidak dibutuhkan dalam proses klasifikasi. Sementara proses *case folding*, semua kata-kata dikonversi menjadi huruf-huruf kecil.

2. *Tokenizing*

*Tokenizing* merupakan tahapan memisahkan teks, kalimat, atau paragraf menjadi token-token atau bagian-bagian tertentu.

3. *Stopword Removal*

Tahapan *stopword removal* bertujuan untuk menghilangkan kata-kata yang sering muncul dan tidak diperlukan.

4. *Normalization*

Tahapan normalisasi bertujuan untuk mengoreksi kata-kata yang disingkat ataupun salah penulisan kata.

5. *Stemming*

*Stemming* bertujuan untuk mengubah kata menjadi bentuk dasarnya dengan menghilangkan imbuhan pada suatu kata

#### 2.2.4 Analisis Sentimen

Analisis sentimen berkaitan dengan kategorisasi atau klasifikasi opini yang diungkapkan dalam dokumen tekstual. Seringkali unit teks diklasifikasikan ke dalam beberapa kategori seperti positif, negatif, atau netral berdasarkan valensi opini yang diungkapkan dalam unit tersebut. Dengan analisis sentimen, perusahaan dapat mengidentifikasi tanggapan, sentimen atau emosi pelanggan terhadap suatu produk atau jasa. Analisis sentimen bekerja pada teks opini sedangkan *text mining* bagus untuk teks faktual. Terdapat 3 kategori kelas sentimen, dimana kelas sentimen ini ditentukan berdasarkan hasil perhitungan skor sentimen yaitu (Singh et al., 2019):

Tabel 2. 2 Kategori kelas Sentimen

Kelas Sentimen	Skor Sentimen
Positif	Skor > 0
Negatif	Skor < 0
Netral	Skor = 0

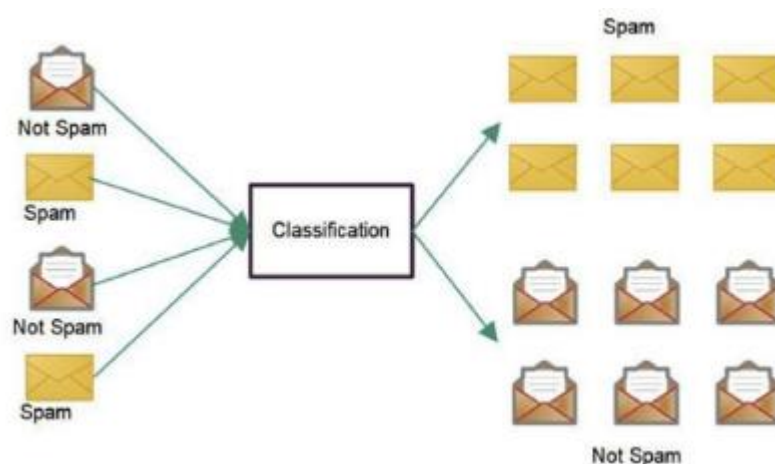


Adapun rumus dari perhitungan skor sentimen adalah sebagai berikut(Singh et al., 2019):

$$Skor = \Sigma \text{ kata positif} - \Sigma \text{ kata negatif} \quad (2.1)$$

### 2.2.5 Klasifikasi

Klasifikasi merupakan proses mengelompokkan *output* ke dalam kelas berbeda berdasarkan satu atau lebih variabel *input* (Hiran et al., 2021). Klasifikasi tergolong dalam pembelajaran jenis *supervised learning*. Jika algoritma mencoba untuk mengklasifikasikan variabel masukan ke dalam 2 kelas berbeda disebut klasifikasi *binary*. Contohnya seperti “spam” atau “tidak spam” dan “ya” atau “tidak”. Jika algoritma mencoba untuk mengklasifikasikan variabel masukan ke dalam lebih dari 2 kelas disebut klasifikasi *multiclass* (Hiran et al., 2021). Contohnya seperti “positif” atau “negatif” atau “netral”.

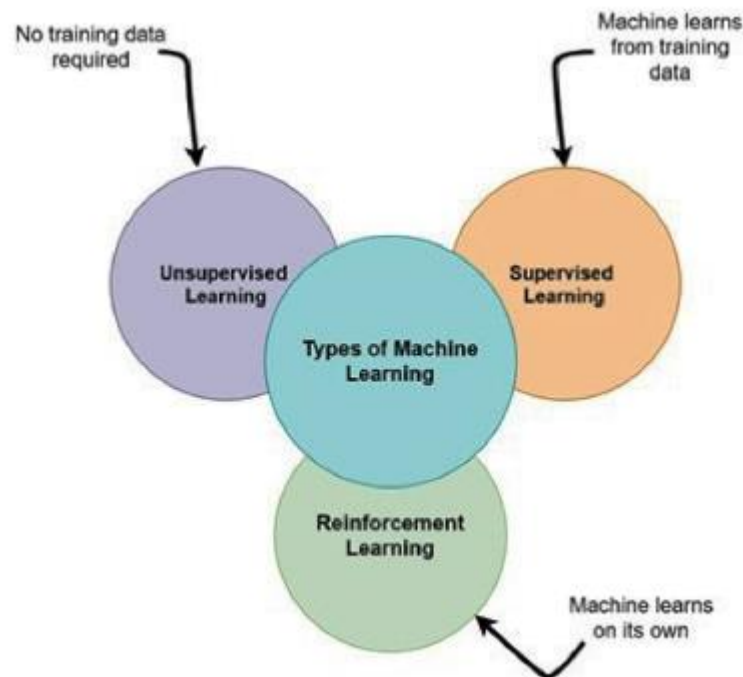


Gambar 2. 5 Klasifikasi

Sumber: (Hiran et al., 2021)

### 2.2.6 Machine Learning

*Machine learning* merupakan salah satu cabang dalam *artificial intelligence*. *Machine Learning* bertujuan untuk memahami struktur dari suatu data dan untuk mencocokkan data tersebut ke dalam model-model yang dapat dipahami dan digunakan oleh manusia (Hiran et al., 2021).



Gambar 2. 6 Algoritma *Machine Learning*

Sumber: (Hiran et al., 2021)

*Machine learning* diklasifikasikan menjadi tiga kategori, yaitu (Hiran et al., 2021):

1. *Supervised Learning*

*Supervised learning* merupakan algoritma yang didesain untuk belajar melalui contoh. Disebut dengan *supervised learning* karena proses pembelajaran algoritma dari data latih dianggap sebagai guru dari proses *supervised learning* (Hiran et al., 2021). Beberapa contoh pada algoritma *supervised machine learning* yaitu *linear regression* dan *random forest* pada permasalahan regresi dan *support vector machine* pada permasalahan klasifikasi (Brownlee, 2016).

2. *Unsupervised Learning*

*Unsupervised learning* berhubungan dengan data yang tidak berlabel berarti di sini kita memiliki data masukan dan tidak ada variabel keluaran yang sesuai. Algoritma ini berkebalikan dengan *supervised learning*. Pada *unsupervised learning*, *user* tidak butuh diajarkan dengan model dimana tidak ada keluaran yang benar. Algoritma ini belajar sendiri dari data masukan dan menemukan polanya dan informasi dari data untuk belajar dan mengelompokkan data berdasarkan kemiripannya (Hiran et al., 2021). Beberapa

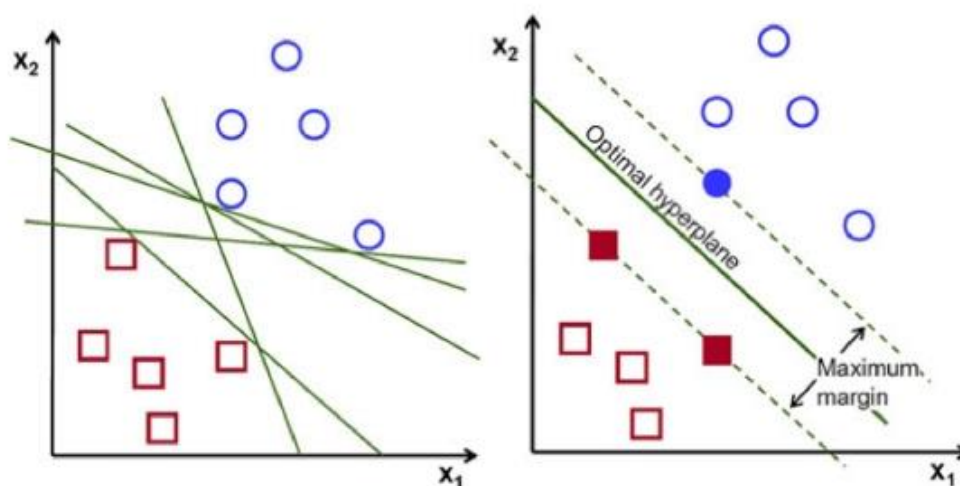
contoh pada algoritma *unsupervised learning* yaitu *k-means* pada permasalahan *clustering* dan algoritma *apriori* pada permasalahan *association rule learning* (Brownlee, 2016).

### 3. *Reinforcement Learning*

*Reinforcement learning* merupakan bagian dari *artificial intelligence* yang melatih algoritma dengan pendekatan *trial and error* untuk mencapai tujuan. Jika agen melalui langkah yang berhasil maka akan mendapatkan *reward*, sebaliknya jika gagal agen mendapat penalti. Tujuannya untuk memaksimalkan *reward* keseluruhan. Algoritma ini belajar menggunakan *feedback* tanpa data yang diberi label. *Reinforcement learning* berinteraksi dengan lingkungannya dan, setelah mengamati konsekuensi dari tindakannya, dapat belajar untuk mengubah perilakunya sendiri tanggapan atas imbalan yang diterima (Sutton & Barto, 1988).

#### 2.2.7 *Support Vector Machine*

*Support Vector Machine* merupakan teknik yang digunakan untuk memprediksi baik pada kasus regresi maupun klasifikasi (Santosa, 2007). SVM tergolong ke dalam *supervised learning*. SVM menghasilkan akurasi yang signifikan dengan menggunakan daya komputasi lebih sedikit. Tujuan algoritma SVM adalah menemukan sebuah *hyperplane* di ruang N-dimensional (N- angka dari fitur) yang mengategorikan poin data secara jelas.



Gambar 2. 7 *Hyperlane* pada *Support Vector Machine*

Sumber: (Hiran et al., 2021)

Ada banyak *hyperplane* yang dapat dipilih untuk memisahkan dua kelas titik data. Tujuannya untuk menemukan *plane* dengan margin terbesar, atau jarak terjauh antar titik data dari kedua kelas seperti pada Gambar 2. 7. memaksimalkan jarak margin memberikan beberapa penguatan, sehingga lebih mudah untuk mengklasifikasikan titik data di masa depan. SVM memiliki beberapa jenis yaitu:

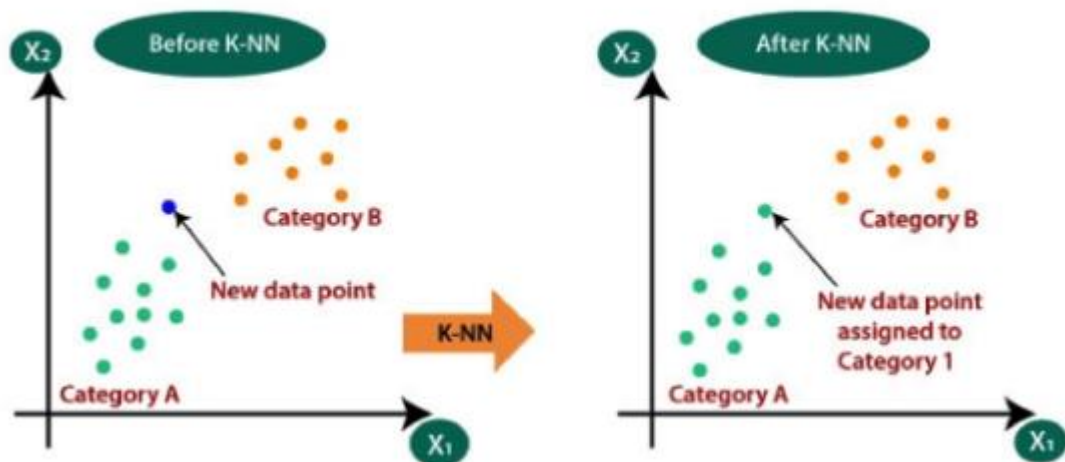
1) *Linear SVM*, digunakan untuk data yang dapat dipisahkan secara linier, artinya jika suatu dataset dapat diklasifikasikan menjadi dua kelas hanya dengan menggunakan satu garis lurus maka disebut data yang dapat dipisahkan secara linier, dan pengklasifikasi yang digunakan disebut Linear SVM

2) *Non-linear SVM*, digunakan untuk mengklasifikasikan data yang terpisah secara non-linier, artinya jika suatu dataset tidak dapat diklasifikasikan menggunakan garis lurus, maka diklasifikasikan sebagai data non-linier, dan pengklasifikasi yang digunakan adalah pengklasifikasi SVM non-linier

Terdapat beberapa kelebihan SVM yaitu efektif pada ruang dimensi yang lebih tinggi, sangat akurat dan bekerja dengan baik di lingkungan berdimensi tinggi. Selain itu dalam menentukan jarak menggunakan *support vector* yang menjadikan proses komputasi menjadi cepat (Cortes & Vapnik, 1995).

### **2.2.8 *K-Nearest Neighbor***

*K-Nearest Neighbor* merupakan algoritma *supervised learning* yang jelas dan simpel yang bisa digunakan untuk memecahkan masalah regresi dan klasifikasi (Hiran et al., 2021). Algoritma ini mengasumsikan bahwa kasus baru dan kasus yang ada adalah mirip dan menempatkan kasus baru ke dalam kategori yang paling mirip ke dalam kategori yang ada. Berikut merupakan gambaran kategori atau kelas dari sebuah dataset dengan bantuan K-NN.



Gambar 2. 8 *K-Nearest Neighbor*

Sumber: (Hiran et al., 2021)

*Nearest Neighbor* secara sederhana diartikan sebagai tetangga terdekat, sementara  $k$  merupakan jumlah tetangga terdekatnya. Maksudnya yaitu suatu data baru (data uji) akan dinilai berdasarkan tetangga terdekatnya (Saputra & Kristiyanti, 2021). Dikarenakan algoritma ini mengandalkan jarak, oleh karena itu perlu jarak perlu diukur terlebih dahulu menggunakan salah satu metode. *Euclidean distance* merupakan metode paling umum digunakan untuk menghitung jarak pada  $K$ -NN. Berikut merupakan persamaan *euclidean distance* (Saputra & Kristiyanti, 2021):

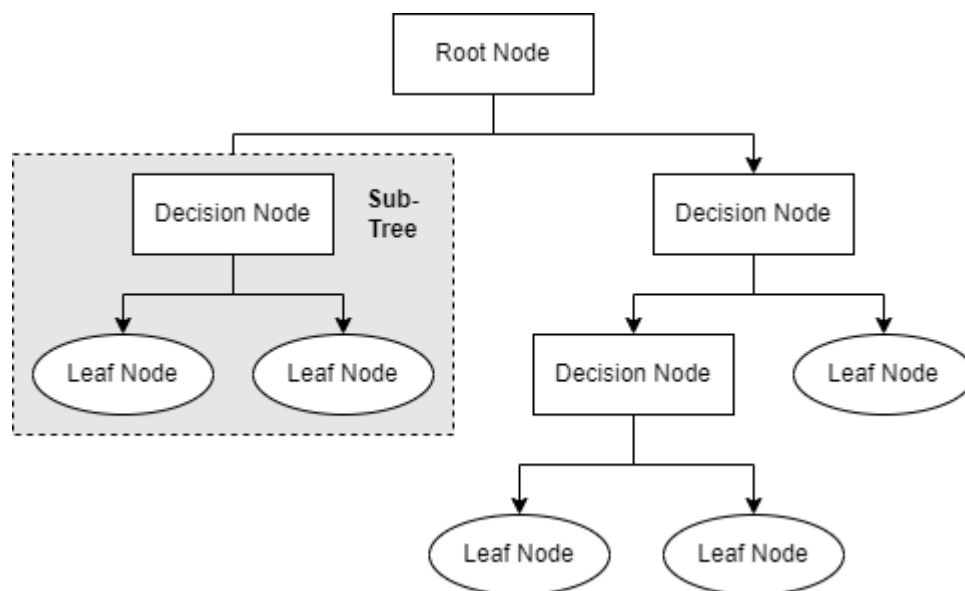
$$d(P, Q) = \|P_i - Q_i\|_0 = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (2.2)$$

Algoritma *K-Nearest Neighbor* memiliki beberapa kelebihan yaitu mudah dimengerti dan diaplikasikan, bekerja lebih efektif pada data latih yang besar, dan menghasilkan data yang lebih akurat. Namun algoritma ini juga memiliki kekurangan yaitu perlu menetapkan nilai  $k$  yang paling optimal yang menyatakan jumlah tetangga terdekat, data yang hilang tidak ditangani secara implisit, dan sensitif terhadap *outlier* (Saputra & Kristiyanti, 2021).

### 2.2.9 Decision Tree

*Decision tree* (pohon keputusan) merupakan algoritma dengan kategori *supervised learning* yang dapat diterapkan pada permasalahan klasifikasi dan regresi (Hiran et al., 2021). Algoritma ini bekerja dengan mencari akar yang menjadi fitur dengan pengaruh tertinggi pada kasus (Saputra & Kristiyanti, 2021). Posisi fitur lainnya terdapat di bawah fitur akar berperan menjadi cabang hingga sampai daun. Pada posisi ini, tidak dapat lagi dilakukan perhitungan dikarenakan akhir dari struktur pohon (Saputra & Kristiyanti, 2021). Algoritma *decision tree* mencoba untuk meningkatkan akurasi dengan menghapus cabang-cabang pohon yang mencerminkan *noise* dalam sebuah data (Suyanto, 2017).

Model *Decision Tree* digambarkan sebagai sebuah pohon (*tree*) yang terdiri atas node-node yang merepresentasikan atribut untuk memprediksi hasil dari sebuah permasalahan. Adapun struktur dari *decision tree* ditunjukkan pada Gambar 2. 9



Gambar 2. 9 Decision Tree

Sumber: (Saputra & Kristiyanti, 2021)

Pada gambar tersebut, *root node* atau disebut juga *node induk* merupakan *node* pertama yang fitur tempati yang memiliki nilai informasi tertinggi. Pada *decision node*, *sub node* dipisah menjadi lebih banyak *sub node*. Pada *leaf node* disebut juga *node terminal* karena tidak bisa dipisahkan lagi. *Sub-Tree* merupakan bagian pohon dari pohon keseluruhan (Saputra & Kristiyanti, 2021).

### 2.2.10 *Naïve Bayes Classifier (NBC)*

*Naïve Bayes Classifier* merupakan algoritma klasifikasi yang berdasarkan Theorema Bayes. Algoritma ini menggunakan probabilitas atau kemungkinan dalam menyelesaikan permasalahannya. Algoritma ini dibentuk dari kata Naïve dan bayes. Disebut Naïve karena mengasumsikan kejadian dari satu fitur tidak berhubungan dengan kejadian yang lainnya. Dikenal sebagai bayes karena berdasarkan pada Theorema Bayes (Hiran et al., 2021). Algoritma ini melakukan perhitungan nilai berdasarkan kemungkinan kemunculan kategori berdasarkan kelas targetnya (Saputra & Kristiyanti, 2021). Adapun rumusnya yaitu:

$$P(c|X) = \frac{P(x|c)P(c)}{P(x)} \quad (2.3)$$

Keterangan:

X = data yang belum diketahui kelasnya

c = hipotesa data merupakan suatu kelas yang spesifik

$P(c|X)$  = Probabilitas hipotesis berdasarkan kondisi

$P(x|c)$  = Probabilitas berdasarkan kondisi pada hipotesis

$P(c)$  = Probabilitas hipotesis

$P(x)$  = Probabilitas c

Algoritma ini tergolong ke dalam pembelajaran *supervised*, oleh karena itu pada proses pembelajaran dibutuhkan data awal berupa data latih untuk dapat mengambil keputusan. *Naïve Bayes Classifier* memiliki beberapa kelebihan yaitu tidak membutuhkan *data training* yang banyak dan jumlah data yang banyak, apabila terdapat nilai yang hilang dapat diabaikan pada perhitungan, dan perhitungan yang cepat dan efisien (Saputra & Kristiyanti, 2021).

### 2.2.11 *Term Weighting (Pembobotan)*

Pembobotan merupakan proses penghitungan dan pemberian bobot setiap kata sesuai tingkat kepentingannya. Istilah frekuensi dan bobot TFIDF adalah skema pembobotan kata yang populer. Istilah bobot dapat digunakan sebagai nilai atribut dalam pengkodean teks menjadi vektor numerik atau disebut juga sebagai vektorisasi (Jo, 2019). Kita dapat menggunakan *term frequency* yang merupakan kemunculan setiap kata yang diberikan

teks sebagai skema pembobotan kata. Diasumsikan bahwa *stopword* yang paling sering muncul dalam teks telah dihapus seluruhnya terlebih dahulu. Kata-kata tersebut diberi bobot dengan menghitung kemunculannya dalam teks yang diberikan dalam skema ini. Ada dua jenis *term frequency*: *term frequency* absolut sebagai kemunculan kata-kata dan *term frequency* relatif sebagai rasio kemunculannya terhadap frekuensi maksimal. *Term frequency* relatif lebih disukai daripada *term frequency* absolut, untuk menghindari perkiraan yang terlalu tinggi dan terlalu rendah karena panjang teks (Jo, 2019).

TF-IDF (*Term Frequency-Inverse Document Frequency*) merupakan metode pembobotan kata yang paling populer di bidang pencarian informasi tetapi memerlukan referensi ke seluruh kumpulan teks yang disebut korpus. Bobot kata yang dihitung dengan skema TF-IDF sebanding dengan kemunculan pada teks tertentu, namun berbanding terbalik dengan kemunculan pada teks lainnya. Adapun rumus dari TF-IDF adalah sebagai berikut (Purbo, 2019):

$$TF\ IDF = TF \times IDF = TF \times \log_e \frac{|D|}{DF} \quad (2.4)$$

Dimana,

TF = *term frequency*, banyaknya sebuah kata atau istilah muncul

DF = *document frequency*, jumlah dokumen di mana kata atau istilah tersebut muncul.

Minimal 1 dokumen

Apabila TF tinggi maka TF IDF akan naik yang berarti istilah atau kata tersebut penting.

Namun, jika DF tinggi maka TF IDF menjadi rendah (Purbo, 2019).

### 2.2.12 Uji Validitas

Uji validitas adalah uji yang digunakan untuk mengetahui dan menguji ketepatan dan ketetapan dari kuesioner yang digunakan sebagai instrumen penelitian sehingga dapat dikatakan instrumen tersebut *valid*. Uji validitas suatu kuesioner dinyatakan *valid* apabila setiap pertanyaan dalam kuesioner dapat digunakan menjadi perantara dalam mengungkapkan dan mengetahui sesuatu yang akan diukur oleh kuesioner tersebut



(Ghozali, 2015). Apabila korelasi antara skor setiap *item* pertanyaan dengan total skor memiliki tingkat signifikansi  $> 0,05$ , maka kriteria yang digunakan valid. Uji validitas instrumen dapat menggunakan rumus korelasi melalui koefisien korelasi *Product Moment*. Berikut merupakan rumus korelasi berdasarkan *Pearson Product Moment* (Hidayat, 2021):

$$r_{hitung} = \frac{n. (\Sigma XY) - (\Sigma X). (\Sigma Y)}{\sqrt{(n\Sigma X^2 - (\Sigma X)^2) . (n\Sigma Y^2 - (\Sigma Y)^2)}} \quad (2.5)$$

Keterangan:

$r_{hitung}$	=	koefisien korelasi ( $r_{hitung}$ )
$n$	=	jumlah responden
$X$	=	skor setiap <i>item</i>
$Y$	=	skor total
$\Sigma X$	=	jumlah skor <i>item</i>
$\Sigma Y$	=	jumlah skor total

Dalam pengolahan data menggunakan SPSS, *Bivariate Pearson (Pearson Product Moment)* dan *Corrected Item-Total Correlation* merupakan uji validitas yang sering digunakan. apabila  $r_{hitung} > r_{tabel}$  maka item pertanyaan berkorelasi signifikansi terhadap skor total dan dinyatakan valid. Jika  $r_{hitung} < r_{tabel}$  maka item pertanyaan tidak berkorelasi secara signifikansi terhadap skor total dan dinyatakan tidak valid (Hidayat, 2021).

### 2.2.13 Uji Reliabilitas

Uji reliabilitas merupakan uji yang digunakan untuk mengetahui apakah data yang dihasilkan dapat diandalkan atau bersifat tangguh. Uji ini mengukur variabel yang digunakan melalui pernyataan atau pertanyaan yang digunakan dengan membandingkan nilai *Cronbach's Alpha* dengan tingkat signifikansi yang digunakan (Darma, 2021). Pengukuran reliabilitas hanya dilakukan dengan cara *one shot* atau pengukuran sekali saja dibantu dengan SPSS uji statistik *Cronbach Alpha* ( $\alpha$ ), dimana suatu variabel atau konstruk dinyatakan reliabel apabila nilai *Cronbach Alpha*  $> 0,60$  (Ghozali, 2013). Adapun rumus *Alpha* yaitu sebagai berikut:

$$r_{it} = \left[ \frac{k}{k-1} \right] \left[ 1 - \frac{\sum Si^2}{\sum St^2} \right] \quad (2.6)$$

Keterangan:

$r_{it}$  = koefisien reliabilitas

$k$  = banyaknya butir pertanyaan

$\sum Si^2$  = jumlah varians butir

$\sum St^2$  = varians total

Dasar pengambilan keputusan dalam uji reliabilitas yaitu sebagai berikut (Darma, 2021):

- Apabila nilai *Cronbach's Alpha* > tingkat signifikan, maka instrumen dikatakan reliabel
- Apabila nilai *Cronbach's Alpha* < tingkat signifikan, maka instrumen dikatakan tidak reliabel

#### 2.2.14 Kepuasan Pelanggan

Kepuasan pelanggan merupakan tanggapan pelanggan terhadap ketidaksesuaian antara tingkat kepentingan sebelumnya dan kinerja nyata yang pelanggan rasakan setelah penggunaan (Rangkuti, 2008). Menurut Kotler (1997), kepuasan pelanggan adalah perasaan senang atau kecewa seseorang akibat membandingkan kinerja (atau hasil) yang diterima suatu produk dibandingkan dengan harapan orang tersebut. Menurut Gerson (1993), pelanggan yang puas akan membeli lebih banyak sepanjang waktu dan membeli lebih sering. Selin itu, bila pelanggan puas maka pelanggan akan merekomendasikan kepada orang lain. Menurut (Kotler, 1997) dalam (Wijaya, 2011) terdapat empat metode untuk mengetahui kepuasan pelanggan:

##### 1) Sistem Keluhan dan Saran

Perusahaan yang berorientasi pada pelanggan (*Customer-Oriented*) menyediakan kesempatan seluas – luasnya bagi para pelanggannya untuk menyampaikan saran, kritik dan keluhan mereka.

##### 2) Survei kepuasan Pelanggan

Dengan melakukan survei kepuasan pelanggan, perusahaan secara langsung akan mendapatkan respons dan umpan balik dari pelanggan dan juga menunjukkan kesan positif bahwa perusahaan menaruh perhatian kepada para pelanggannya.

### 3) *Ghost Shopping*

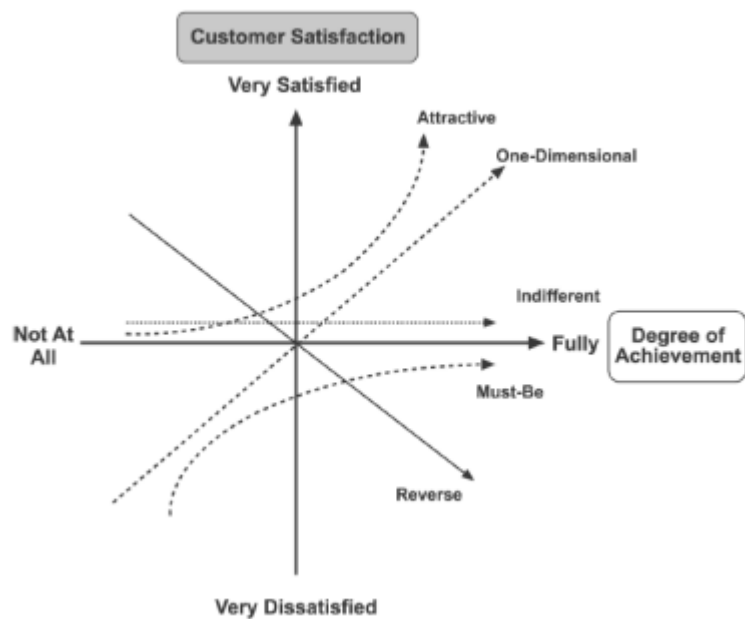
*Ghost shopping* yaitu mengumpulkan beberapa orang yang akan dipekerjakan sebagai pelanggan atau pembeli potensial produk perusahaan dan kompetitor berdasarkan pengalamannya dalam pembelian produk-produk tersebut. Selanjutnya melaporkan hal-hal mengenai kekurangan dan kekuatan produk, mencermati dan menilai cara penanganan yang lebih baik.

### 4) Analisis Kehilangan Konsumen (*Lost Customer Analysis*)

Perusahaan mencoba menghubungi para pelanggannya yang sudah berhenti berlangganan atau berpindah ke perusahaan lain. Perusahaan mencoba mengamati penyebab pelanggan bisa beralih ke produk lain.

#### **2.2.15 Kano**

Model Kano dikembangkan oleh Noriaki Kano (Kano et al., 1984) dalam yang bertujuan untuk mengategorikan atribut-atribut dari produk atau pun jasa berdasarkan seberapa baik produk atau jasa dapat memuaskan kebutuhan pelanggan (Wijaya, 2011). Model *Kano* adalah metode yang baik untuk mengeksplorasi karakteristik kebutuhan pelanggan. Ketidaktahuan terhadap kategori dari atribut layanan dapat berdampak negatif bagi pihak perusahaan. Pemberi produk atau jasa tidak menyadari bahwa keinginan pelanggan bervariasi (Wijaya, 2011). Dengan menggunakan model *Kano*, seseorang dapat menentukan kebutuhan pelanggan dengan mengklasifikasikannya ke dalam kategori *Kano* dan dengan menempatkan kebutuhan setiap atribut pada grafik 2. 11.



Gambar 2. 10 Kano Model Graph

Sumber: (Kano et al., 1984)

Menurut Kano et al. (1984) dalam Matzler et al. (2004) pandangan tradisional yang linear antara tingkat kinerja layanan yang diberikan dan peningkatan kepuasan pelanggan tidak selalu benar. Terkadang kepuasan pelanggan dapat menunjukkan bentuk non-linier dan non-linier inilah kualitas pelayanan yang diberikan tidak selalu sesuai dengan harapan atau dengan kata lain tidak selalu juga memperoleh kepuasan. Menurut Kano et al. (1984) dalam (Hartono et al., 2018) terdapat 3 jenis tingkat atribut layanan yang berdampak pada kepuasan konsumen yaitu:

1. *Attractive* (Menarik)

Layanan pada kategori ini tidak diharapkan dan disampaikan langsung oleh konsumen, tetapi jika layanan tersebut disediakan akan berdampak pada tingkat kepuasan konsumen yang sangat tinggi.

2. *One Dimensional* (Satu Dimensi)

Layanan pada kategori ini dituntut konsumen untuk disediakan. Tingkat kepuasan konsumen berbanding lurus dengan kinerja layanan. Apabila kinerja layanan semakin baik, maka tingkat kepuasan pelanggan juga semakin tinggi, dan sebaliknya.

3. *Must-Be* (Harus Ada)

Layanan pada kategori ini apabila tidak terpenuhi akan menimbulkan kekecewaan pelanggan dan kemungkinan besar yang terjadi pelanggan akan berhenti menggunakan layanan tersebut kembali. Tetapi, apabila layanan dipenuhi juga tidak selalu berdampak pada kepuasan konsumen.

Adapun respon konsumen terhadap kebutuhan dari layanan yaitu:

1. *Indifferent*

Pada kategori ini, tersedia atau tidak tersedianya layanan tidak memiliki pengaruh terhadap kepuasan konsumen

2. *Reverse*

Pada kategori ini, apabila layanan tidak disediakan maka konsumen akan lebih puas. Begitu juga sebaliknya, jika layanan tersebut disediakan maka tingkat kepuasan konsumen akan menurun

3. *Questionable*

Pada kategori ini memiliki ketidakpastian pada tingkat kepuasan konsumen, dimana konsumen akan puas atau tidak puas jika layanan disediakan atau pun tidak disediakan.

Pada penyusunan kuesioner *kano*, masing-masing atribut layanan disusun menjadi pernyataan positif (fungsional) dan pernyataan negatif (disfungsional) sehingga dapat diketahui pengaruh dari kepuasan pelanggan jika layanan disediakan atau tidak disediakan. Jawaban dari kuesioner kemudian dikombinasikan ke dalam tabel evaluasi model kano yang tercantum pada Tabel 2.3.

Tabel 2. 3 *Kano Model Categorization*

Kebutuhan Konsumen	Disfungsional				
	1. Suka	2. Harap	3. Netral	4. Toleransi	5. Tidak Suka
Fungsional	1. Suka	Q	A	A	O
	2. Harap	R	I	I	M
	3. Netral	R	I	I	M
	4. Toleransi	R	I	I	M

Kebutuhan Konsumen	Disfungsional				
	1. Suka	2. Harap	3. Netral	4. Toleransi	5. Tidak Suka
5. Tidak Suka	R	R	R	R	Q

Hasil evaluasi model Kano selanjutnya dianalisis dengan menggunakan metode *better-worse* atau *if-then* untuk memberikan gambaran yang lebih lengkap. Peran masing-masing komponen atau atribut pelayanan akan berdampak pada tingkat kepuasan oleh karena itu harus diukur dengan nilai *better* dan nilai *worse*. Nilai *better* didapatkan dengan menjumlahkan atribut A dan O lalu membaginya dengan total nilai respon A, O, M dan I. Nilai *worse* didapatkan dengan menjumlahkan atribut O dan M dan membaginya dengan total nilai respon. dari atribut A, O, M, dan I. Berikut merupakan rumus perhitungannya (Hartono et al., 2018):

$$Better (Y) = \frac{A + O}{A + O + M + I} \quad (2.6)$$

$$Worse (X) = \frac{O + M}{(A + O + M + I)(-1)} \quad (2.7)$$

Nilai *better* menggambarkan seberapa besar peningkatan kepuasan pelanggan jika atribut pelayanan disediakan yang ditandai dengan tanda positif, sementara nilai *worse* menggambarkan seberapa besar penurunan kepuasan jika atribut pelayanan tidak disediakan yang ditandai dengan tanda negatif. Sementara itu, metode *if-then* digunakan berdasarkan *Formula Blauth* (Hartono et al., 2018):

1. Jika  $(one-dimensional + attractive + must-be) > (indifferent + reverse + questionable)$ , maka kategori yang didapatkan yaitu yang paling tinggi dari  $(one-dimensional + attractive + must-be)$ .
2. Jika  $(one-dimensional + attractive + must-be) < (indifferent + reverse + questionable)$ , maka kategori yang didapatkan yaitu yang paling tinggi dari  $(indifferent + reverse + questionable)$ .

3. Jika (*one-dimensional + attractive + must-be*) = (*indifferent + reverse + questionable*)  
maka kategori yang didapatkan yaitu yang paling tinggi di antara seluruh kategori  
*Kano*

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Objek Penelitian**

Objek pada penelitian ini adalah persepsi pengguna aplikasi *Alfagift* melalui *review* pengguna pada situs *Google Play* dan *AppStore*. Adapun subjek dalam penelitian ini merupakan pengguna aplikasi *Alfagift*.

#### **3.2 Jenis dan Sumber Data**

##### *3.2.1 Data Primer*

Data primer pada penelitian ini yaitu ulasan pengguna aplikasi *Alfagift* yang diperoleh melalui situs *Google Play* dan *AppStore* dengan metode *scraping*. Data ulasan aplikasi *Alfagift* yang diambil dari tanggal 6 April 2022 – 30 April 2023 dengan menggunakan versi 4.9.0 hingga 4.17.0. Versi tersebut merupakan versi terakhir aplikasi sebelum dilakukan pembaruan.

##### *3.2.2 Data Sekunder*

Data sekunder pada penelitian ini diperoleh dari penelitian terdahulu melalui jurnal-jurnal, artikel-artikel maupun buku-buku.

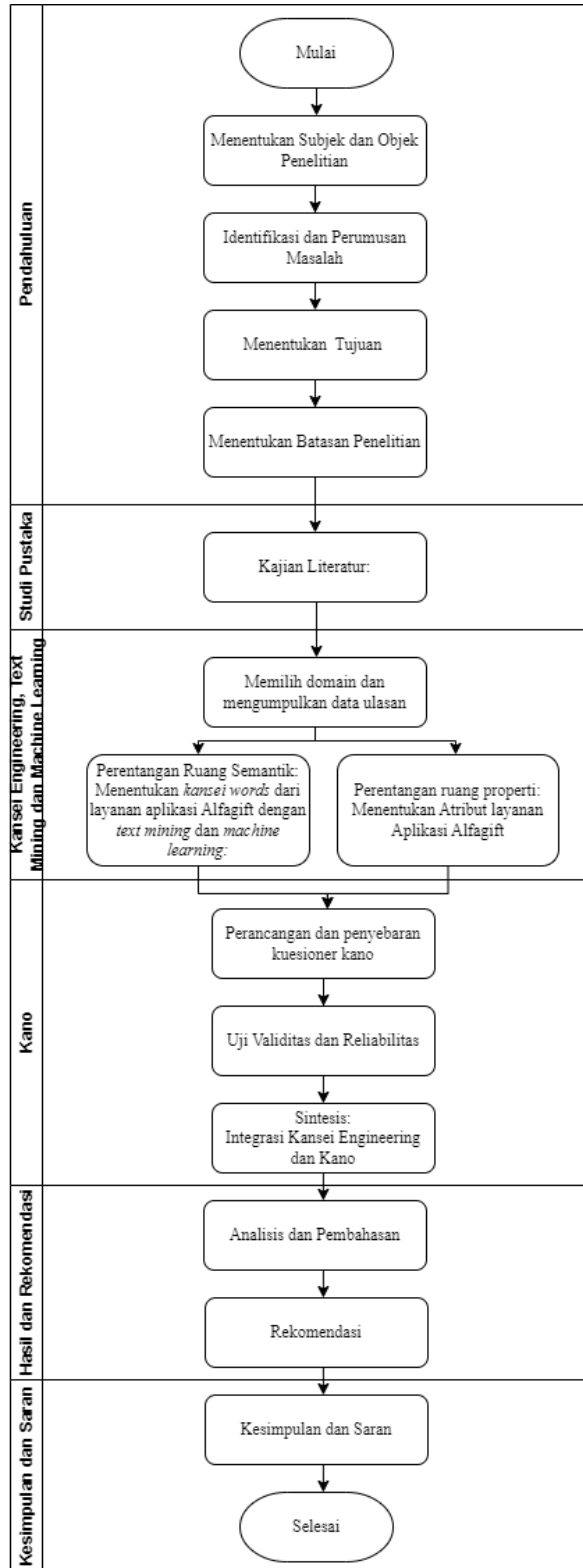
#### **3.3 Metode Pengumpulan Data**

Pengumpulan data pada penelitian ini dilakukan dengan teknik *scraping* data dengan bantuan *website AppFollow*. *AppFollow* merupakan sebuah *website* yang menyediakan fasilitas untuk melakukan *scraping data*. Selain itu, dilakukan pengambilan data melalui kuesioner yang ditujukan pada pengguna aplikasi *Alfagift*.

#### **3.4 Tahapan Penelitian**

Tahapan penelitian ini mengacu pada alur penelitian (Hsiao et al., 2017) dan (Mujibulloh & Jakaria, 2022). Berikut tahapan-tahapan dalam penelitian ini ditunjukkan dalam *flowchart* penelitian yang terdapat pada Gambar 3.1.





Gambar 3. 1 Alur Penelitian

1. Mulai
2. Menentukan Objek dan Subjek Penelitian

Penelitian dimulai dengan menentukan objek dan subjek dari penelitian. Objek dari penelitian ini merupakan persepsi pengguna aplikasi *Alfagift* yang diperoleh dari ulasan pada *website Google Play* dan *AppStore*. Sementara subjek dari penelitian ini yaitu pengguna aplikasi *Alfagift*.

### 3. Identifikasi dan Perumusan Masalah

Pada tahapan ini, identifikasi masalah dilakukan dengan melihat situasi dan kondisi saat ini. Berdasarkan hasil identifikasi, diperoleh informasi bahwa di tengah pergeseran perilaku konsumen yang memilih berbelanja secara *online*, aplikasi *Alfagift* menjadi media bagi ritel *Alfamart* untuk menarik konsumen lebih luas lagi dengan memberikan informasi promo, kemudahan transaksi, dan pengantaran produk gratis ongkir. Namun, berdasarkan *rating* dan *review* yang diberikan oleh pengguna aplikasi menunjukkan bahwa perlu dilakukan evaluasi dan pengembangan pada aplikasi. Setelah itu dilakukan perumusan masalah yang menjadi sasaran dari penelitian. Adapun rumusan masalah dalam penelitian ini yaitu bagaimana hasil integrasi *Machine Learning* dengan *Kansei Engineering* dan *Kano* dalam mengembangkan aplikasi *Alfagift* guna meningkatkan kepuasan pelanggan.

### 4. Menentukan Tujuan

Pada tahapan ini dilakukan penentuan tujuan penelitian. Tujuan penelitian menjelaskan mengenai apa yang ingin dihasilkan atau dicapai dalam penelitian ini. Adapun tujuan dari penelitian ini yaitu mengetahui persepsi pengguna mengenai aplikasi *Alfagift* yang diperoleh dengan ekstraksi data ulasan, sehingga diharapkan aplikasi *Alfagift* dapat dikembangkan sesuai dengan kebutuhan dan keinginan pengguna aplikasi *Alfagift* guna meningkatkan kepuasan pelanggan.

### 5. Menentukan Batasan Penelitian

Agar penelitian tetap terarah dan pokok masalah tidak melebar ke hal yang tidak perlu, maka dibutuhkan batasan penelitian. Adapun batasan dari penelitian ini yaitu:

- 1) Data ulasan pengguna aplikasi *Alfagift* diambil pada situs *Google Play* dan *AppStore*
- 2) Data ulasan pengguna aplikasi diambil dari tanggal 6 April 2022 – 30 April 2023 dengan menggunakan versi 4.9.0 hingga 4.17.0
- 3) Pengolahan data ulasan menggunakan metode *Text Mining* dengan algoritma *Machine Learning*.

### 6. Kajian Literatur

Pada tahap kajian literatur terdapat dua jenis yaitu kajian induktif dan kajian deduktif. Kajian induktif berisi tentang hasil penelitian-penelitian terdahulu yang sebelumnya telah dilakukan sebelumnya. Sementara kajian deduktif membahas teori-teori yang berkaitan dengan topik penelitian, dimana topik yang digunakan pada penelitian ini yaitu *Text Mining*, *Machine Learning*, *Kansei Engineering*, *Kano*, dan lain-lain.

#### 7. Memilih *Domain* dan Mengumpulkan Data Ulasan

Pada tahapan ini, domain menentukan kelompok pengguna, ceruk pasar, serta spesifikasi dari produk atau jasa baru. Pada penelitian ini ditentukan berupa aplikasi belanja *online* yang mudah digunakan dan minim permasalahan, serta baik dari segi sistem, informasi, maupun layanan yang dapat memuaskan penggunanya. Kemudian pada tahapan pengumpulan data dilakukan dengan cara mengekstraksi ulasan pengguna secara *online* dengan teknik *scraping* data dengan bantuan *website AppFollow*. Data ulasan pengguna aplikasi *Alfagift* diperoleh melalui *website Google Play* dan *AppStore* yang diambil dari tanggal 6 April 2022 – 30 April 2023 dengan versi 4.9.0 hingga 4.17.0. Pemilihan rentang waktu tersebut dikarenakan terjadi penurunan *rating* secara kontinu yang disebabkan oleh banyaknya pengguna yang memberikan *rating* rendah pada aplikasi tersebut seperti pada Gambar 1.6 dan 1.7. Kemudian, dilakukan pengambilan data melalui kuesioner yang ditujukan pada pengguna aplikasi *Alfagift*.

#### 8. Perentangan Ruang Semantik

Pada tahapan ini dilakukan identifikasi *kansei words* dengan metode *text mining*. *Text mining* diterapkan untuk mengekstrak data ulasan pengguna aplikasi *Alfagift*. Kemudian data ulasan diklasifikasi dengan algoritma *machine learning*. *Kansei words* diambil dari hasil asosiasi kata berdasarkan hasil klasifikasi menggunakan algoritma *machine learning*.

#### 9. Perentangan Ruang Properti

Tahapan ini bertujuan untuk mengetahui apakah sebuah elemen produk atau puluhan elemen dapat berpengaruh besar terhadap *kansei*, maka dibutuhkan untuk mengidentifikasi semua atribut produk atau layanan dan hanya dipilih yang memiliki dampak terbesar pada *kansei* tertentu (Nagamachi & Lokman, 2011).

#### 10. Perancangan dan Penyebaran Kuesioner Kano

Pada tahapan ini dilakukan penyebaran kuesioner *kano* untuk mengukur fungsional dan disfungsional suatu produk dan layanan. Kuesioner ditujukan kepada 50 responden yang

memenuhi kriteria yaitu pernah menggunakan aplikasi *Alfagift* dan pernah belanja menggunakan aplikasi *Alfagift*.

#### 11. Uji Validitas dan Reliabilitas

Uji validitas dilakukan untuk mengetahui dan menguji ketepatan dan ketetapan dari kuesioner yang digunakan sebagai instrumen penelitian sehingga dapat dikatakan instrumen tersebut *valid*. Sedangkan uji reliabilitas dilakukan untuk mengetahui sejauh mana suatu instrumen dapat dipercaya.

#### 12. Sintesis

Pada tahapan ini, ruang semantik dan ruang properti dihubungkan bersama. Untuk setiap *kansei word* yaitu dengan menghubungkan *kansei words* dan atribut produk atau layanan. Pada Atribut produk atau layanan yang ditentukan berdasarkan *kansei words* dikategorikan menggunakan metode *Kano*

#### 13. Analisis dan Pembahasan

Analisis dan pembahasa merupakan tahapan mengidentifikasi hasil dari pengolahan data yang telah dilakukan. Pada tahapan ini akan melakukan pembahasan mengenai hasil pengolahan data

#### 14. Rekomendasi

Pada tahapan ini berisi rekomendasi perbaikan yang dapat diberikan untuk mengatasi permasalahan pada aplikasi *Alfagift* berdasarkan hasil dari integrasi *Kansei* dan *Kano*.

#### 15. Kesimpulan dan Saran

Pada tahapan ini dijabarkan kesimpulan dan saran penelitian. Pada bagian kesimpulan membahas hasil-hasil dari penelitian secara singkat. Sementara itu, saran penelitian berisi usulan berdasarkan hasil temuan penelitian sebagai bahan evaluasi agar penelitian selanjutnya menjadi lebih baik.

#### 16. Selesai

## BAB IV

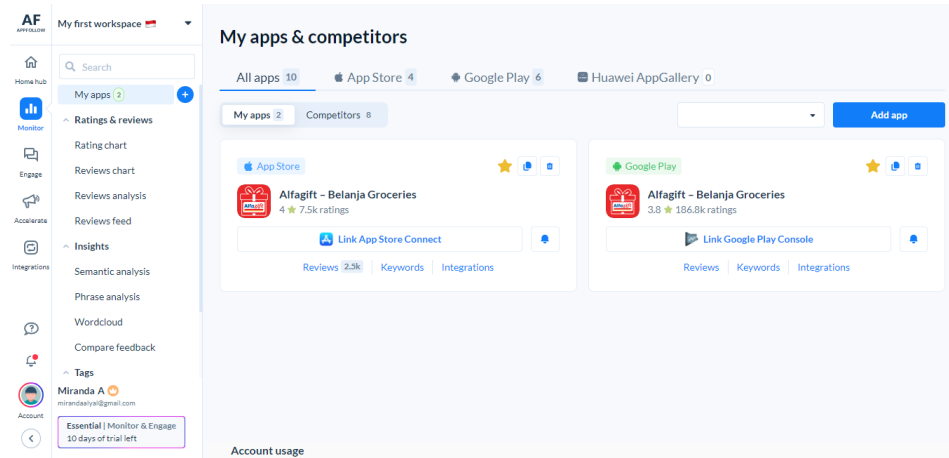
### PENGUMPULAN DAN PENGOLAHAN DATA

#### 4.1 Pemilihan *Service Domain* dan Pengumpulan Data Ulasan

Aplikasi secara umum dibagi menjadi tiga *platform* yaitu *desktop*, *web*, dan *mobile*. Aplikasi terdiri dari beberapa jenis, yaitu hiburan dan berita, perjalanan, *gaming*, *social media*, *fintech*, utilitas, kesehatan dan kebugaran, dan *e-commerce*. Aplikasi *Alfagift* dikategorikan sebagai aplikasi *e-commerce* berbasis *mobile* dan *web*. Aplikasi *mobile* merupakan teknologi yang paling banyak digunakan, hal ini didukung dengan perkembangan *smartphone* yang semakin canggih (Yusril et al., 2021). Sehingga *domain* yang dipilih dalam penelitian ini yaitu aplikasi *e-commerce* berbasis *mobile*.

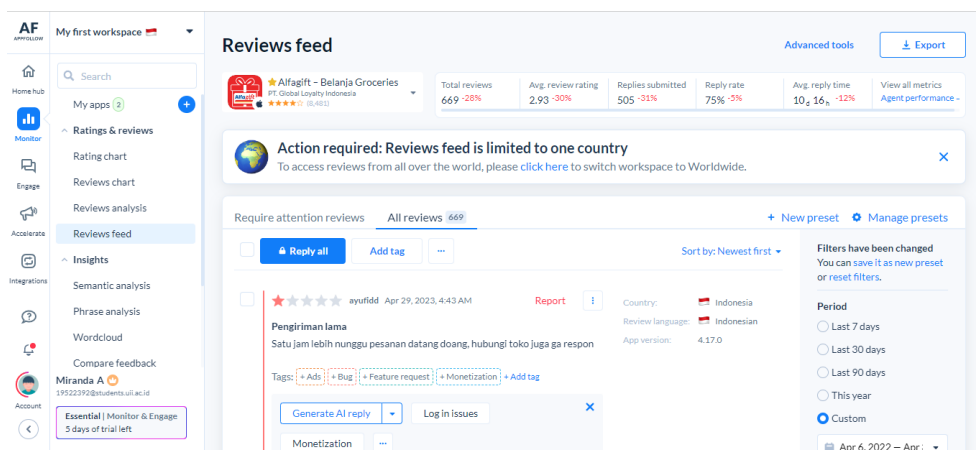
Data yang diambil pada penelitian ini yaitu berupa data ulasan pengguna aplikasi *Alfagift* yang dapat diakses pada situs *Google Play* dengan alamat *domain*: <https://play.google.com/store/apps/details?id=com.alfamart.alfagift&hl=id> dan situs *AppStore* dengan alamat *domain*: <https://apps.apple.com/id/app/alfagift-belanja-groceries/id1013717463?l=id>. Pengumpulan data pada penelitian ini dilakukan dengan teknik *scraping data* menggunakan *tools scraping*. Salah satu *tools* yang digunakan untuk melakukan *scraping data* yaitu *AppFollow*. *AppFollow* merupakan sebuah *website* yang menyediakan fasilitas untuk melakukan ekstraksi data secara gratis. *AppFollow* dapat diakses pada alamat *domain*: <https://appfollow.io/> yang digunakan untuk mengambil data ulasan yang kemudian akan diekspor ke dalam bentuk *file xlsx* atau *CSV*.

Tahapan pertama yang harus dilakukan adalah membuka situs *AppFollow*. Pada Gambar 4.1 merupakan *interface* pada situs *AppFollow*. Untuk menambahkan aplikasi yang akan dilakukan *scraping data*, klik “*add app*” yang berada di bagian kanan atas. Pada penelitian ini, mengambil dua data ulasan aplikasi *Alfagift* yaitu dari *AppStore* dan *Google Play*.



Gambar 4. 1 Interface AppFollow

Kemudian untuk melihat *review* yang diberikan pengguna aplikasi, klik pada bagian “*reviews feed*”. Ulasan yang ditampilkan dapat dilakukan filterisasi tanggal diberikan ulasan dan bahasa yang digunakan seperti pada Gambar 4.2. Dalam penelitian ini, data ulasan yang diambil dari tanggal 6 April 2022 – 30 April 2023 dengan versi 4.9.0 hingga 4.17.0. Data yang diambil sebanyak 16.291 ulasan yang terdiri dari 15.622 ulasan pada situs *Google Play* dan 669 ulasan pada situs *AppStore*. Setelah data difilterisasi, selanjutnya klik “*export*” yang berada di bagian kanan atas untuk menyimpan data ke dalam bentuk *file xlsx* atau *csv* dengan hasil seperti pada Gambar 4.3.



Gambar 4. 2 Ulasan Pengguna Aplikasi Alfagift

Submissic	Publicatio	AppID	AppName	Country	Review La	Version	Author	Rating	Title	Review	Translatec	Translatec	Reply	Dati	Reply	Dell	Developpe	User	Tags	Categorie	Updated	Semal
1	2022-04-0	2023-03-1	com.alfamart.alfagift	id	4.9.0	Edita Mah	5	Goodat														
2	2022-04-0	2023-03-1	com.alfamart.alfagift	id	4.9.0	Fariz Musi	1	Registrasi gagal terus														
3	2022-04-0	2023-03-1	com.alfamart.alfagift	id	4.9.0	Imas Setiy	5	Enak banget...														
4	2022-04-0	2023-03-1	com.alfamart.alfagift	id	4.9.0	Tika Irok	3	Terlalu sering harus di perbarui														
5	2022-04-0	2023-03-1	com.alfamart.alfagift	id	4.9.0	Asyifa Haz	2	Dulu suka sekali belanja alfagift														
6	2022-04-0	2023-03-1	com.alfamart.alfagift	id	4.9.0	Handayan	1	Order dari jam 8.15 pagi sampe														
7	2022-04-0	2023-03-1	com.alfamart.alfagift	id	4.9.0	way Yudi	1	Hapus aja aplikasi gx mutu...														
8	2022-04-0	2023-03-1	com.alfamart.alfagift	id	4.9.0	Hendrik A	1	Susah untuk daftar														
9	2022-04-0	2023-03-1	com.alfamart.alfagift	id	4.9.0	Rahma Yai	3	Udah 2 Minggu kenapa ga bisa														
10	2022-04-0	2023-03-1	com.alfamart.alfagift	id	4.9.0	Irma Indru	5	Sangat senang gak perlu keluar														
11	2022-04-0	2023-03-1	com.alfamart.alfagift	id	4.9.0	No Face N	1	Alfagift kenapa jadi begini sete														
12	2022-04-0	2023-03-1	com.alfamart.alfagift	id	4.9.0	Indri Kurn	5	Aku kasih bintang 5 karena sud														
13	2022-04-0	2023-03-1	com.alfamart.alfagift	id	4.9.0	Husnul K	3	Sangat membantu														
14	2022-04-0	2023-03-1	com.alfamart.alfagift	id	4.9.0	ä-ä-ä-ä-ä	1	Maaf saya ralat penilaian saya														
15	2022-04-0	2023-03-1	com.alfamart.alfagift	id	4.9.0	rakmat if	5	Mantap... apalagi ada cashback														
16	2022-04-0	2023-03-1	com.alfamart.alfagift	id	4.9.0	Dewi Sri V	4	Konapa kok gk bisa di buka ?														
17	2022-04-0	2023-03-1	com.alfamart.alfagift	id	4.9.0	Refan Insi	5	Baik pelayanan dzn juga ramah														
18	2022-04-0	2023-03-1	com.alfamart.alfagift	id	4.9.0	Risma Pra	3	Dari kapan tau daftar dji memb														
19	2022-04-0	2023-03-1	com.alfamart.alfagift	id	4.9.0																	

Gambar 4. 3 Hasil *Scraping Data*

## 4.2 Perentangan Ruang Semantik

### 4.2.1 Ekstraksi Data Ulasan

#### 4.2.2.1 Pre-Processing Data

Dalam memproses data dengan berjumlah sangat besar akan memerlukan sumber daya yang jumlahnya tidak sedikit kaitannya dengan pengolahan data tersebut sehingga tahapan pemrosesan awal atau *preprocessing data* perlu dilakukan agar data data yang sebelumnya tidak terstruktur menjadi terstruktur. Tahapan ini dilakukan dengan bantuan *tools python*. Adapun tahapan dari *preprocessing data* adalah sebagai berikut:

#### 1. Data Cleaning dan Case Folding

Proses *data cleaning* bertujuan untuk membersihkan ulasan dari *noise* yang tidak dibutuhkan dalam proses klasifikasi. Adapun atribut yang tidak diperlukan yaitu berupa tag HTML, URL, *username* (@), tanda baca (.,), simbol (#%&'?"~\$!) atau lainnya yang bukan huruf dan karakter numerik. Dalam proses *data cleaning* dibantu menggunakan *library regex*. *Regular expression (regex)* digunakan untuk pencarian teks dengan menggunakan pola (*pattern*). *Library* ini dapat memudahkan untuk mencari *string* tertentu dari teks dengan jumlah banyak (Mahawardana et al., 2022). Sementara proses *case folding*, semua kata-kata dikonversi menjadi huruf-huruf kecil menggunakan fungsi *lower ()*. Hasil dari tahapan ini dapat dilihat pada Tabel 4.1.

Tabel 4. 1 Proses *Data Cleaning* dan *Case Folding*

Sebelum	Sesudah
Aplikasi yg sangat membantu	aplikasi yg sangat membantu

Sebelum	Sesudah
Minuman kaleng diskon 50%, pesan 6, yg datang 3.... Kalo ngga diskon jgn bilang diskon, itu namanya penipuan.	minuman kaleng diskon pesan yg datang kalo ngga diskon jgn bilang diskon itu namanya penipuan

Berikut merupakan *pseudocode* dari proses *data cleaning* dan *case folding*.

Tabel 4. 2 *Pseudocode* dari Proses *Data Cleaning* dan *Case Folding*

<i>Pseudocode</i>	Fungsi
def cleansing(Review):	
Review = re.sub('[^a-zA-Z]', ' ', Review).lower()	Mengubah teks menjadi huruf kecil
Review = Review.translate(str.maketrans("", "" ,string.punctuation))	Menghapus tanda baca
Review = Review.encode('ascii', 'replace').decode('ascii')	Menghapus karakter <i>nonASCII</i> ( <i>emoticon, chinese word, dll.</i> )
Review = ' '.join(re.sub("([@#][A-Za-z0-9]+) (\w+:\V\S+)", " ", Review).split())	Menghilangkan <i>mention, link, hashtag</i>
Review = re.sub(r"(?:\@ http?:// https?:// www)\S+", "", Review)	Menghapus <i>https</i> dan <i>http</i>
Review = re.sub(r'\d+', "", Review)	Menghapus angka
Review = re.sub(r'\b[a-zA-Z]\b', "", Review)	Menghilangkan karakter tunggal
Review = re.sub("\n", " ", Review)	Mengganti <i>line</i> baru dengan spasi
Review = re.sub("\s+', ' ', Review)	Menghilangkan <i>multiple whitespace</i> menjadi <i>single whitespace</i>
return Review	



## 2. *Tokenizing*

*Tokenizing* merupakan tahapan memisahkan teks, kalimat, atau paragraf menjadi token-token atau bagian-bagian tertentu. *Library nltk.tokenize* dengan fungsi *word\_tokenize* digunakan untuk membagi kalimat menjadi beberapa bagian yang disebut *token*. Kata-kata tersebut dipisahkan menggunakan tanda koma (,). Hasil dari tahapan proses *tokenizing* dapat dilihat pada Tabel 4.3.

Tabel 4. 3 Proses *Tokenizing*

Sebelum	Sesudah
aplikasi yg sangat membantu	[aplikasi, yg, sangat, membantu]
minuman kaleng diskon pesan yg datang kalo ngga diskon jgn bilang diskon itu namanya penipuan	[minuman, kaleng, diskon, pesan, yg, datang, kalo, ngga, diskon, jgn, bilang, diskon, itu, namanya, penipuan]

Berikut merupakan *pseudocode* dari proses *tokenizing*.

Tabel 4. 4 *Pseudocode* dari Proses *Tokenizing*

<i>Pseudocode</i>	Fungsi
def tokenize (txt): word_token = word_tokenize (txt) return word_token	Memisahkan kalimat menjadi bagian-bagian tertentu menggunakan tanda koma (,).

## 3. *Stopword Removal*

Tahapan *stopword removal* bertujuan untuk menghilangkan kata-kata yang sering muncul dan tidak diperlukan seperti konjungsi, kata tanya, *timestamps*, dan kata ganti orang. *Stopword removal* bertujuan untuk menyimpan kata-kata penting pada suatu kalimat (Setyawati et al., 2021). Proses ini menggunakan *library sastrawi* yaitu *StopwordRemoverFactory ()*. *Library* ini telah mendukung pemrosesan dalam Bahasa Indonesia, dimana daftar *stopwords* dalam Bahasa Indonesia sudah tersedia (Gunawan & Santoso, 2021). Selain *list stopwords* Indonesia yang disediakan oleh *library Sastrawi*, juga ditambahkan *list* kata yang tidak diperlukan dengan menambahkan secara langsung

kata pada *more\_stopword* agar dapat dihapus. Penghapusan *stopword* pada *preprocessing* dapat meningkatkan kinerja klasifikasi. Apabila *stopwords* tidak dihapus akan menambah fitur yang tidak bermakna dan performa model yang dihasilkan menurun (Arifiyanti et al., 2022). Hasil dari tahapan proses *stopword removal* dapat dilihat pada Tabel 4.5.

Tabel 4. 5 Proses *Stopword Removal*

Sebelum	Sesudah
[aplikasi, yg, sangat, membantu]	[aplikasi, sangat, membantu]
[email, bantuan, alpagift, pergantian, nomor, dan, email, gak, di, respon]	[email, bantuan, alpagift, pergantian, nomor, email, gak, respon]

Berikut merupakan *pseudocode* dari proses *stopword*.

Tabel 4. 6 *Pseudocode* dari Proses *Stopword*

<i>Pseudocode</i>	Fungsi
<code>stop_factory = StopWordRemoverFactory()</code>	Create
<code>more_stopword = ['muluuuu', 'agszblj', 'yang', 'tetepa', 'hufft', 'kagak', 'nya', 'ya', 'saja', 'aja', 'padahal', 'kokk', 'loh', 'di', 'banget', 'dengan', 'deh', 'emang', 'kalau', 'sih', 'lautze', 'men', 'tuh', 'itu', 'tu', 'ituu', 'dong', 'donk', 'kali', 'yaaahhh', 'yaaaah', 'hadehhh', 'iyya', 'yaa', 'yah', 'nih', 'sayah', 'haduhhhhhh', 'biat', 'kaa', 'qu', 'nha', 'yuuuu', 'tadinya', 'aku', 'kak', 'diky', 'maulana', 'enggak', 'yuuu', 'yya', 'sik', 'mba', 'koyo', 'tk', 'gw', 'ajh', 'yg', 'yng', 'sing', 'trm', 'ksh', 'je', 'jooooossss', 'anjir', 'nb', 'gb', 'cuma', 'gedek', 'wkwkwk', 'mah', 'refind', 'ayolah', 'behhhhhhh', 'bpnya', 'mrsa', 'kesni', 'anjirrrrr', 'lhoh', 'cok', 'cht', 'jm', 'utuk', 'tewur', 'aggqxd', 'agtmsgn', 'lagih', 'tikarriss', 'kecust', 'jug', 'qk', 'ho', 'hufft', 'du', 'bualazisokhi', 'ndruru', 'ayo', 'tahi', 'lho', 'padahak', 'sampr', 'sekaliiii', 'dansering', 'gww', 'aduh', 'kabr', 'deh', 'kadang', 'mau', 'ini', 'jadi', 'kalau', 'mending', 'dari', 'mana', 'malah', 'masa', 'kayak', 'mending', 'payah', 'kalaw', 'agwxlby', 'agszpjz', 'ffit', 'rrtr', 'rf', 'gc', 'lllllllllllllllllllllllllllllllllllllll', 'nah', 'jajanannya', 'yamana', 'hmmmmmmmm', 'peh', 'su', 'kenap', 'hpku', 'drmh',</code>	 factor y + ambil stopw ord bawaa n Meng gabun gan stopw ord

<i>Pseudocode</i>	<b>Fungs i</b>
'getiiiiingggg', 'lbihbyk', 'gi', 'loch', 'bedug', 'trss', 'rbu', 'melulu', 'josss', 'bor', 'jhooossss', 'uuu', 'woy', 'nian', 'sat', 'shg', 'maksh', 'puk', 'ag', 'alafa', 'agtylfq', 'broo', 'oesa', 'nisa', 'riazl', 'choi', 'xoxo', 'widih', 'ca', 'suruhkomplain', 'qq', 'uuuuwhh', 'agxjjmc', 'pv', 'hehehe', 'gang', 'kok', 'pripriitt', 'pilhan', 'yuk', 'wkkkkk', 'mq', 'kgk', 'dsb', 'yivni', 'cobaik', 'hariii', 'wkwkw', 'hai', 'hahahhah', 'bye', 'bwa', 'wkwk', 'tq', 'is', 'satsetsatset', 'teruwu', 'uwu', 'yoyo', 'satset', 'siii', 'itupun', 'sskarang', 'ceunahhhh', 'agstkft', 'urusannsaya', 'cetho', 'iki', 'oy', 'yuhu', 'nybselalu', 'my', 'lingk', 'ii', 'ajinggg', 'bqbi', 'thank', 'apa', 'wok', 'lu', 'wi', 'cuakkssss', 'mnit', 'ndk', 'abcd', 'yowesben', 'cnuh', 'pihat', 'sory', 'dksh', 'hmm', 'maknyossssss', 'terrrruncccccchhh', 'emangg', 'aakan', 'kan', 'distruk', 'disemoilin', 'se', 'gue', 'ah', 'gakelas', 'entahlah', 'dall', 'lah', 'pula', 'hhhhh', 'sarankn', 'masih', 'trmksh', 'jw', 'klari', 'hemm', 'bk', 'jj', 'sukakmu', 'ajalaahhhh', 'lol', 'sipp', 'lgy', 'jozz', 'kapook', 'lsng', 'bangeetttt', 'hadeuhhhhhh', 'jdnya', 'brkali', 'ach', 'trmkasih', 'ogah', 'nying', 'gr', 'agnwlnx', 'gd', 'tokok', 'syag', 'udanawu', 'strukyng', 'mughni', 'hehee', 'tos', 'pukil', 'uan', 'cux', 'sangaat', 'ilha', 'iyahh', 'ng', 'chuakss', 'hahahah', 'idh', 'ckckckc', 'pulaaak', 'hadeeeh', 'ddrrrrrds', 'zzuuu', 'xx', 'seei', 'langkaplancar', 'hahahahahahahaha', 'ci', 'woiiii', 'pokokny', 'vc', 'xd', 'io', 'op', 'ig', 'tlongperbaiki', 'setegh', 'kapokkk', 'di', 'dannn', 'ntb', 'udsh', 'stampny', 'seharusx', 'dtoko', 'haduhh', 'efkuasi', 'mntuuul', 'pisann', 'kaannn', 'zzzzz', 'pesanankita', 'siniii', 'bangetttttttt', 'dd', 'sss', 'did', 'chfigog', 'hemmmm', 'pffft', 'uh', 'ff', 'ooo', 'hahahha', 'wkwkkwkwk', 'lmyn', 'xxxx', 'hm', 'agzzryh', 'eeeeeeuuuuuhhhh', 'haliaangsss', 'indoxxx', 'llahh', 'haduhhhh', 'deeehhh', 'mulugiliran', 'bcndaaaa', 'anjingg', 'trussssssss', 'mantullllll', 'mantapppppppp', 'okedahhh', 'ntapss', 'mueheheh', 'deeeh', 'tapiiiiii', 'sumpaaaah', 'hahahahahhah', 'mantaaaabbb', 'mantjaapp', 'sxxx', 'suiiiippp', 'ggg', 'hhh', 'siiip', 'mndg', 'serri', 'kocaggg', 'siiip', 'berxx', 'alfao', 'membantunperencanaan', 'mantapppp', 'alflfagif', 'mantaaaaaapppppp', 'debbie', 'sangatt', 'agxzqrs', 'bia', 'oyeeee', 'adhityofirdausputra', 'mantb', 'mamtap', 'benja', 'maszeeeh', 'maszeeeh',	Meng hilang kan kata- kata yang terdap at pada <i>list</i> <i>stopw</i> <i>ord</i>

<i>Pseudocode</i>	<b>Fungs</b> <b>i</b>
<p>'membantuuuuuu', 'ge', 'siiiipp', 'qqqqq', 'manttttapppp', 'teruuusss',  'mantaapp', 'teruuussss', 'hadeeehh', 'www', 'fg', 'xvi', 'okkkk', 'laahhhhh',  'hellooo', 'sekaleeeeeeee', 'qn', 'agvkncq', 'waduuhhh', 'pnx', 'klopun', 'tg',  'ekwkekwk', 'aaaaa', 'kenak', 'mengsad', 'tmpa', 'alfaaa', 'semangaaat',  'terosssss', 'naon', 'ajg', 'hh', 'barkot', 'pqapku', 'msknstam', 'bcanya',  'huhhh', 'kenp', 'terussss', 'malzz', 'wkwkwkkwk', 'by', 'sla',  'mantapppppppp', 'mantapmantap', 'pokoknyaa', 'mantapsssss',  'semuapastidapat', 'joss', 'gandos', 'coyyyyy', 'maauk', 'okhe', 'josssss',  'ntabbss', 'lokotreee', 'kuwokkk', 'josssmantabbbb', 'pdahla', 'mntaaap',  'downloadagi', 'pollllll', 'mlahan', 'keapliksinya', 'cex', 'hilih', 'aph',  'hadeeehh', 'appah', 'sekarank', 'wkwkwkwkwk', 'au', 'qa', 'lllllllll', 'brow',  'mampusss', 'sukeeee', 'heloowww', 'siti', 'nurbayaaaaa',  'adannghfbbfbbfngngn', 'aamma',  'mmmxmmdmkkdkdnrjfyhcjcnfnfmn',  'ncnfncjdjjdhxhbbfncncnfjfnfnngngmcmnmctjnnfmmgmmhmmhmmh  hmmhmmmmmbmbmgkjkkfkfkckcmfmmfmmcmccmcmcmffmjfn  o', 'dek', 'nxnxbhxbxbxbxbxnxnxbxnxnsnnxnsnnxnjnznxbxbxnxn',  'iccx dx', 'xz', 'jadj', 'nurul', 'fikri', 'nf', 'bangeeetttt', 'teruuuss',  'terooosssss', 'lov', 'uu', 'yuuuuu', 'slurr', 'chas', 'selaluada', 'mucel',  'bangatt', 'pp', 'okt', 'ret', 'maav', 'hemmmmm', 'beknya', 'awwas', 'sken',  'maneh', 'sono', 'kon', 'iku', 'gtau', 'oleng', 'gaskeun', 'seriiiiing', 'ditnya',  'awq', 'mngnai', 'selaluuuu', 'matap', 'ai', 'terpa', 'woy',  'wooooooyyyyyyy', 'anjim', 'agxldqz', 'virgo', 'sexan', 'wadidaw', 'nbb',  'mmjju', 'uy', 'gnji', 'himne', 'gen', 'jin', 'gam', 'cumak', 'sekalidg', 'ieu',  'klkk', 'dupeeeer', 'hariau', 'transksi', 'paaassss', 'mpah', 'madesuuu', 'pnah',  'ttoooooopppp', 'ngontek', 'apaam', 'hahah', 'three', 'alahhh', 'lngsungpun',  'recom', 'oks', 'menemptn', 'nebgamvil', 'kobtak', 'jmen', 'kkkkk',  'grgrgrgrgrgr', 'cuss', 'sukseme', 'thaknk', 'dllnya', 'pulaaa', 'terusyaaa',  'oi', 'taee', 'daahhh', 'begimana', 'beberapakali', 'idn', 'ba', 'uraaa', 'waahhh',  'eeeist', 'taaadaa', 'laahh', 'lnua', 'gasspoll', 'pokonyaa', 'komplein', 'oh',  'aaaa', 'pluss', 'omooo', 'gj', 'sni', 'aplagi', 'swaktu', 'mntaapp', 'tya',</p>	

<i>Pseudocode</i>	<b>Fungs</b> <b>i</b>
<p>'murahmeriahmerakyat', 'yampon', 'prt', 'dddddd', 'kaapook', 'daebak',  'alfaambil', 'selali', 'termantulityyy', 'duanyaaa', 'yaampunn', 'alfanyaa',  'hayuuu', 'mantulllll', 'warrrrr', 'hehehhe', 'shari', 'jdhie', 'mskih', 'pdal',  'paharah', 'terlopelope', 'eeh', 'hadeeh', 'bro', 'hahaha', 'denga', 'sampk',  'cuz', 'duh', 'ataw', 'bkr', 'bangeetxzzz', 'thnkyou', 'soa', 'aaa', 'mallaem',  'ditelfongakk', 'ata', 'gercepp', 'teback', 'bkl', 'arigattou', 'monangis',  'siscakhoh', 'agqlmzk', 'setelh', 'akhhhh', 'asyukaaa', 'hadeuuuuuh', 'wawan',  'prikkkkkkkk', 'prik', 'haiyahhh', 'anying', 'ntahlah', 'yeh', 'dog',  'gimanaaaaa', 'lunya', 'goblogg', 'goblokk', 'tolloll', 'pastinyaaa',  'gerceptanpabatas', 'omo', 'npa', 'pakememberalfadiskonnyagede', 'sx',  'xxxxxx', 'terussssssss', 'af', 'capcus', 'lebin', 'mon', 'nasir', 'rotunnadillah',  'ulul', 'semantap', 'pr', 'pe', 'ws', 'are', 'wweew', 'esa', 'as', 'we', 'wwf',  'wewdwew', 'sdhs', 'alah', 'joo', 'bgst', 'pkonyaaa', 'hmmmm', 'pliss', 'uat',  'sorr', 'ditl', 'yaudahlah', 'yeayyy', 'bhkan', 'apll', 'jir', 'abu', 'achmad',  'omnya', 'ir', 'iii', 'cii', 'bangsattt', 'kenapaaaaaa', 'yeay', 'mantaappp', 'ibuu',  'mommmmy', 'ean', 'thnk', 'bangeettttt', 'hadehhhhh', 'njing', 'hampi', 'ygdi',  'bhh', 'mbok', 'hihi', 'gmb', 'ksmi', 'ssw', 'kmk', 'terussssss', 'vibri',  'remadhani', 'yakkk', 'dongok', 'drtd', 'terooooos', 'buuangeeet', 'hutuo',  'limboto', 'snk', 'badahal', 'kalao', 'laah', 'duu', 'sekaraanng', 'seksrang',  'ihiiyy', 'rekom', 'goblox', 'hehhh', 'noh', 'capekanjiir', 'tutpny', 'trisna',  'tansa', 'thankyoy', 'eze', 'ddk', 'oo', 'thp', 'tersu', 'meuli', 'kamari', 'udud',  'ari', 'ayeuna', 'teu', 'sb', 'mlvvkmlv', 'mnpplvvp', 'mmmm', 'mpmp', 'lvn',  'vb', 'mvpvlmmkijj', 'nvymp', 'kin', 'vn', 'sedg', 'ses', 'sllau', 'mintibs',  'mboh', 'rifando', 'agustinus', 'eyyehhe', 'vangke', 'panindo', 'hba', 'iasa',  'anjenggggg', 'rko', 'pepk', 'saiia', 'jrngn', 'saepudln', 'kntl', 'otpt', 'hiegis',  'hiegis', 'ina', 'pasowr', 'gimanaaaa', 'mudaj', 'trhdp', 'siput', 'duli', 'akhm',  'dekay', 'spb', 'mhhak', 'jamal', 'hhhhhhh', 'ampunnnnnn', 'hoi', 'tetntang',  'pdjl', 'haduhhh', 'qlo', 'tj', 'uwahahaha', 'ta', 'gesss', 'hufffff', 'qqery',  'benereeer', 'untyk', 'mantaapp', 'mahhh', 'slebeww', 'mahhh', 'dch',  'dahhh', 'tksh', 'xixixi', 'ahaha', 'mantaabb', 'loohh', 'elama', 'kkikklik',  'setiao', 'hitu', 'tadaaaaa', 'soalnyaaa', 'arghhhhh', 'hheee', 'yeayy', 'guys',</p>	

<i>Pseudocode</i>	<b>Fungs</b> <b>i</b>
<pre> 'hheehe', 'khann', 'xixi', 'sooo', 'hahahahah', 'thanksss', 'terkma', 'xixix', 'dab', 'yey', 'wlopun', 'bey', 'yeei', 'tis', 'gays', 'hayyyy', 'gokilllll', 'polllll', 'yeeey', 'looohhh', 'wlw', 'susanto', 'uwuu', 'duwa', 'ey', 'uhuyy', 'hellow', 'daebakkk', 'ahahahahah', 'terlqu', 'banyq', 'sue', 'semog', 'ilon', 'jffjin', 'guufgh', 'mugbutfvyuf', 'fghygchb', 'rudi', 'leaflet', 'qw', 'nanyeaaaaa', 'teruuuuuuuus', 'akuuuu', 'yahud', 'toolol', 'agtdmdz', 'lhoe', 'njiing', 'kintil', 'geab', 'diman', 'jangkauannya', 'cokk', 'rebesss', 'bangeeeeeet', 'cabca', 'bm', 'aq sada', 'karn', 'ginimah', 'puooollll', 'xnya', 'suelah', 'jafi', 'hya', 'kmdn', 'kmvret', 'seing', 'woiiiiiiiiiiiiiii', 'cusssaa', 'manthulllllll', 'msan', 'gituuuu', 'aajja', 'apsih', 'ith', 'mueah', 'ndg', 'buuuufwy', 'hmmm', 'nbbbbgeuiyy', 'ychtsuw', 'titydz', 'urr', 'ru', 'yeihuu', 'agpxgww', 'atd', 'vktqau', 'ao', 'goblogggggggggggg', 'oon', 'mng', 'assurur', 'wlpn', 'xxdxa', 'sikh', 'sakasi', 'ume', 'ytta', 'wkw', 'ituloh', 'baaaay', 'hhmmm', 'redefdexwe', 'haggkk', 'hei', 'mbknya', 'hdh', 'hhehe', 'srapat', 'gess', 'bambang', 'gassspoll', 'memmem', 'ua', 'iqbal', 'nsp', 'aulia', 'rosiyani', 'eee', 'rahel', 'nja', 'cikakak', 'bngsd', 'kwe', 'huhh', 'laaahh', 'riw', 'taii', 'lu', 'oii', 'inv', 'fry', 'agyklxb', 'refreal', 'purw', 'jatiseeng', 'berlkasi'] datax = stop_factory.get_stop_words()+ more_stopword  dictionary = ArrayDictionary(datax) str = StopWordRemover(dictionary)  def stopw (txt_tokenized):     stop = [word for word in txt_tokenize             d if word not in datax]     return stop </pre>	

#### 4. Normalization

Tahapan normalisasi bertujuan untuk mengoreksi kata-kata yang disingkat ataupun salah penulisan kata. Contohnya seperti kata “cpt” menjadi “cepat”. Kata-kata tidak baku juga diubah menjadi bentuk baku yang sesuai dengan Ejaan Indonesia yang Disempurnakan

(EYD). Contohnya seperti kata “gak” menjadi “tidak”. Sebelum melakukan proses normalisasi perlu menyiapkan *dataset* untuk *mapping term* yang akan diseragamkan. Hasil dari tahapan normalisasi dapat dilihat pada Tabel 4.7.

Tabel 4. 7 Proses Normalisasi

Sebelum	Sesudah
[mau, nanya, min, mengganti, nomer, hp, bagaimana, caranya, aplikasi, alfagift]	[mau, tanya, admin, mengganti, nomor, hp, bagaimana, caranya, aplikasi, alfagift]
[aplikasi, bs, lihat, promo]	[aplikasi, bisa, lihat, promo]
[kalo, pesan, antar, barang, suka, lamaaa]	[kalau, pesan, antar, barang, suka, lama]

Berikut merupakan *pseudocode* dari proses *normalization*.

Tabel 4. 8 *Pseudocode* dari Proses Normalisasi

<i>Pseudocode</i>	Fungsi
def normalized_term(document): return [normalizad_word_dict[term] if term in normalizad_word_dict else term for term in document]	Mengoreksi kata-kata yang disingkat ataupun salah penulisan dan mengubah kata-kata tidak baku menjadi kata-kata baku sesuai dengan EYD

### 5. *Stemming*

Proses *stemming* bertujuan untuk mengubah kata menjadi bentuk dasarnya dengan menghilangkan imbuhan pada suatu kata sehingga mengurangi jumlah kata yang diproses dalam *text mining*. Sedangkan lematisasi merupakan proses megubah kata menjadi bentuk yang sesuai (lemma) agar kata dapat dikelompokkan dengan kata lain yang sama (Chrismanto & Lukito, 2016). Lemitisasi bertujuan mengubah *infinite tense* dan *noun* menjadi sebuah kata dalam Bahasa Inggris yang sama (Nugraha & Sebastian, 2018). Pada penelitian ini tidak dibutuhkan lematisasi dikarenakan kata-kata dalam Bahasa Indonesia tidak mempunyai bentuk-bentuk khusus (*infinite, tense, noun*) seperti kata dalam Bahasa

Inggris. Selain itu, belum terdapat *library* yang menyediakan lematisasi dalam Bahasa Indonesia. Pada proses ini menggunakan *library* Sastrawi yang merupakan *library* populer digunakan untuk *stemming* yang berfokus pada teks berbahasa Indonesia. Sastrawi merupakan *library python* sederhana yang memungkinkan untuk mereduksi kata-kata infleksi berbahasa Indonesia menjadi bentuk dasarnya (*stem*) (Gunawan & Santoso, 2021). *Library Sastrawi* digunakan karena sudah mendukung proses *stemming* dalam Bahasa Indonesia. *Library* ini didasarkan pada aturan morfologi Bahasa Indonesia dan dirancang secara khusus dalam mengatasi permasalahan *stemming* pada Bahasa Indonesia (Mualfah et al., 2023). Dalam penelitian yang dilakukan Nurirwan, menyimpulkan bahwa dengan proses normalisasi dan *stemming* dapat memperoleh akurasi yang lebih tinggi pada algoritma yang digunakan dibandingkan hanya melakukan proses normalisasi saja. Hasil dari proses *stemming* dapat dilihat pada Tabel 4.9.

Tabel 4. 9 Proses *Stemming*

Sebelum	Sesudah
[memudahkan, dalam, berbelanja, kebutuhan, bulanan]	[mudah, dalam, belanja, butuh, bulan]

Berikut merupakan *pseudocode* dari proses *stemming* dapat dilihat pada Tabel 4.10

Tabel 4. 10 *Pseudocode* dari Proses *Stemming*

<i>Pseudocode</i>	Fungsi
def get_stemmed_term(document): return [term_dict[term] for term in document]	Mengembalikan kata menjadi kata dasarnya dengan menghilangkan imbuhan pada suatu kata.

#### 4.2.2.2 Pembobotan TF-IDF

*Term Frequency-Inverse Document Frequency* (TF-IDF) bertujuan untuk melakukan pembobotan setiap kata yang sudah diekstrak. Frekuensi kemunculan kata pada dokumen yang diberikan menunjukkan seberapa penting kata tersebut di dalam dokumen tersebut. Semakin tinggi bobot TF-IDF sebuah kata (*term*) dalam sebuah dokumen, maka semakin penting kata tersebut dalam konteks dokumen tersebut dan semakin besar kontribusi



terhadap identifikasi topik atau makna dokumen (Kusuma Dewi et al., 2023). TF-IDF digunakan untuk memberikan nilai numerik pada setiap kata dalam dokumen agar memudahkan proses analisis dan klasifikasi yang disebut juga proses vektorisasi (Salam, et al., 2023). Adapun hasil dari pembobotan TF-IDF pada salah satu data ulasan dapat dilihat pada Gambar 4.4.

term	TF	TF-IDF
top	0.06666666666666667	0.31616940196432725
up	0.06666666666666667	0.32257037884696643
dana	0.13333333333333333	0.6107165699532552
ribu	0.06666666666666667	0.2913661278851059
salah	0.13333333333333333	0.5380543475906965
kasir	0.06666666666666667	0.255271411595493
kirim	0.06666666666666667	0.15803916990963587
nomor	0.06666666666666667	0.23694549686270158
sekarang	0.06666666666666667	0.2843097389007953
kagak	0.06666666666666667	0.39737989791885336
konfirmasi	0.06666666666666667	0.29889548863047893
bagaimana	0.06666666666666667	0.2400195869933286
lanjut	0.06666666666666667	0.4040521284893189

Gambar 4. 4 Hasil Pembobotan TF-IDF

Berdasarkan hasil pembobotan TF-IDF salah satu dokumen pada Gambar 4.4, menghasilkan nilai numerik yang disebut juga proses vektorisasi, agar data ulasan dapat diklasifikasikan menggunakan algoritma *machine learning*. Hasil pembobotan pada Gambar 4.4 menunjukkan kata “dana” memiliki nilai bobot TF-IDF tertinggi yang berarti kata “dana” merupakan kata terpenting pada dokumen tersebut. Berikut merupakan *pseudocode* dari proses perhitungan skor sentimen dapat dilihat pada Tabel 4.11

Tabel 4. 11 *Pseudocode* dari Pembobotan

<i>Pseudocode</i>	Fungsi
<pre>def calc_TF_IDF(TF):     TF_IDF_Dict = {}     #For each word in the review, we multiply its tf     and its idf.     for key in TF:         TF_IDF_Dict[key] = TF[key] * IDF[key]</pre>	<p>1. Memberi bobot pada setiap kata dalam sebuah kalimat ulasan.</p>

<i>Pseudocode</i>	<b>Fungsi</b>
return TF_IDF_Dict	

#### 4.2.2.3 Perhitungan Skor Sentimen

Pelabelan data dilakukan dengan cara menghitung skor sentimen oleh kamus *lexicon-based*. *Lexicon-based* merupakan metode yang menggunakan kamus sebagai sumber bahasa atau leksikal. Kamus *lexicon* berisi kumpulan dari kata beserta nilai sentimennya yang digunakan untuk mengetahui emosi pada satu kalimat (Fatra et al., 2020). *Lexicon-based* dalam melakukan klasifikasi sentimen dari setiap opini sehingga sebuah kalimat sentimen dapat dikelas-kelaskan sesuai kelas positif, negatif, dan netral. Perhitungan skor sentimen menggunakan Rumus 2. Berikut merupakan hasil perhitungan skor sentimen yang ditunjukkan pada Tabel 4. 12.

Tabel 4. 12 Hasil Perhitungan Skor Sentimen

<b>Ulasan</b>	<b>Kata Positif</b>	<b>Kata Negatif</b>
Selalu belanja	belanja (2), mudah	-
dialfagift mudah transaksi	(4), transaksi (4)	
<b>Jumlah</b>	<b>10</b>	<b>0</b>

Berdasarkan Tabel 4.12, terdapat 3 kata positif dan tidak terdapat kata negatif berdasarkan kamus *lexicon*. Kata positif yang terdeteksi yaitu “belanja”, “mudah”, dan “transaksi. Sehingga jumlah skor sentimen yang diperoleh adalah sebagai berikut:

$$\text{Skor} = 10 - 0 = 10$$

Berdasarkan kategori kelas sentimen pada Tabel 2.2, ulasan pada Tabel 4. 12 dikategorikan sebagai kelas sentimen positif karena hasil perhitungan skor sentimen lebih dari 0. Pada proses perhitungan skor sentimen dibantu menggunakan *library nltk*. Berikut merupakan *pseudocode* dari proses perhitungan skor sentimen dapat dilihat pada Tabel 4.13

Tabel 4. 13 *Pseudocode* dari Proses Perhitungan Skor Sentimen

<i>Pseudocode</i>	<b>Fungsi</b>
<pre> def found_word(ind,words,word,sen,sencol,sentiment,add):     if word in sencol:         sen[sencol.index(word)] += 1     else:         sencol.append(word)         sen.append(1)         add += 1     if (words[ind-1] in negasi):         sentiment += - lexicon['weight'][lexicon_word.index(word)]     else:         sentiment += lexicon['weight'][lexicon_word.index(word)]     return sen,sencol,sentiment,add for i in range(len(df)):     nsen = senrow.shape[0]     words = word_tokenize(df['data_join'][i])     sentiment = 0     add = 0     prev = [0 for ii in range(len(words))]     n_words = len(words)     if len(sencol)&gt;0:         sen =[0 for j in range(len(sencol))]     else:         sen =[]     for word in words:         ind = words.index(word)         sen,sencol,sentiment,add= found_word(ind,words,word,sen,sencol,sentiment,add)     else: </pre>	<p>2. Menuliskan sentimen kata jika ditemukan.</p> <p>3. Jika sudah masuk dalam matriks <i>bag of word</i>, maka tinggal naikkan saja nilainya</p> <p>4. Memeriksa setiap kata, apakah muncul dalam leksikon, lalu menghitung sentimennya jika muncul.</p> <p>5. Periksa apakah termasuk dalam leksikon, jika tidak maka periksa kata dasar, jika masih negatif coba cocokkan kombinasi kata dengan kata yang berdekatan.</p> <p>6. Jika terdapat kata baru yang ditemukan, maka perluas matriksnya</p>

<i>Pseudocode</i>	<b>Fungsi</b>
<pre> kata_dasar = stemmer.stem(word) if kata_dasar in lexicon_word:     sen,sencol,sentiment,add= found_word(ind,words,kata_dasar,sen,sencol,sentiment,add) elif(n_words&gt;1):     if ind-1&gt;-1:         back_1 = words[ind-1]+' '+word         if (back_1 in lexicon_word):             sen,sencol,sentiment,add= found_word(ind,words,back_1,sen,sencol,sentiment,add)         elif(ind-2&gt;-1):             back_2 = words[ind-2]+' '+back_1             if back_2 in lexicon_word:                 sen,sencol,sentiment,add= found_word(ind,words,back_2,sen,sencol,sentiment,add) if add&gt;0:     if i&gt;0:         if (nsen==0):             senrow = np.zeros([i,add],dtype=int)         elif(i!=nsen):             padding_h = np.zeros([nsen,add],dtype=int)             senrow = np.hstack((senrow,padding_h))             padding_v = np.zeros([(i- nsen),senrow.shape[1]],dtype=int)             senrow = np.vstack((senrow,padding_v))         else:             padding =np.zeros([nsen,add],dtype=int)             senrow = np.hstack((senrow,padding))             senrow = np.vstack((senrow,sen))     if i==0:         senrow = np.array(sen).reshape(1,len(sen)) elif(nsen&gt;0): </pre>	<p>7. Jika tidak ada, maka perbarui saja matriks lama.</p>

<i>Pseudocode</i>	<b>Fungsi</b>
<pre>senrow = np.vstack((senrow, sen))</pre>	
<pre>sentiment_list.append(sentiment)</pre>	

#### 4.2.2.4 Pelabelan Skor Sentimen

Pada penelitian ini hanya digunakan ulasan dengan kelas sentimen positif dan negatif saja dikarenakan ulasan dengan kelas sentimen netral dianggap tidak memiliki nilai atau sentimen sehingga kurang memberikan masukan pada aplikasi *Alfagift*. Berikut merupakan jumlah ulasan pada masing-masing kelas sentimen yang dapat dilihat pada Tabel 4.14

Tabel 4. 14 Jumlah Ulasan Kelas Sentimen

<b>Kelas Sentimen</b>	<b>Jumlah Ulasan</b>
Positif	11.772
Negatif	3.281
Netral	1.238
<b>Total</b>	<b>16.281</b>

Berdasarkan hasil perhitungan skor sentimen pada data ulasan di tahap 4.2.2.3, jumlah ulasan dengan kelas sentimen positif sebanyak 11.772 data, kelas sentimen negatif sebanyak 3.281 data, dan kelas sentimen netral sebanyak 1.238 dari keseluruhan total data ulasan sebanyak 16.281 data. Adapun *pseudocode* dari proses pelabelan skor sentimen adalah sebagai berikut.

Tabel 4. 15 *Pseudocode* dari Proses Pelabelan Skor Sentimen

<i>Pseudocode</i>	<b>Fungsi</b>
<pre>def analysis(score):     if score &lt; 0:         return 'Negative'     elif score == 0:         return 'Neutral'</pre>	<p>Melabelkan ulasan menjadi kelas sentimen positif, netral, dan negatif berdasarkan hasil perhitungan skor sentimen.</p>

<i>Pseudocode</i>	<i>Fungsi</i>
else: return 'Positive'	

#### 4.2.2.5 *Klasifikasi*

##### 1. *Pembagian Training Data dan Testing Data*

*Data training* digunakan untuk melatih algoritma dalam membangun sebuah model. Dalam penelitian ini menggunakan 4 skenario pembagian *data training* dan *data testing* yaitu 60:40, 70:30, 80:20, dan 90:10. Berikut merupakan perbandingan jumlah setiap data.

- 1) *Pembagian data training* sebesar 60% dan *data testing* sebesar 40%

Tabel 4. 16 *Pembagian Data Skenario 1*

<i>Klasifikasi</i>	<i>Jumlah</i>	<i>Data Training</i> (60%)	<i>Data Testing</i> (40%)
Positif	11.772	7.063	4.709
Negatif	3.281	1.969	1.312
Total	15.053	9.032	6.021

Berdasarkan pada Tabel 4.16 pada pembagian *data training* sebesar 60% dan *data testing* 40% dari total 15.053 data ulasan, terdapat 9.032 data ulasan yang digunakan sebagai *data training* dan 6.021 data ulasan yang digunakan sebagai *data testing*.

- 2) *Pembagian data training* sebesar 70% dan *data testing* sebesar 30%

Tabel 4. 17 *Pembagian Data Skenario 2*

<i>Klasifikasi</i>	<i>Jumlah</i>	<i>Data Training</i> (70%)	<i>Data Testing</i> (30%)
Positif	11.772	8.240	3532
Negatif	3.281	2.297	984
Total	15.053	10.537	4.516

Berdasarkan pada Tabel 4.17 pada pembagian *data training* sebesar 70% dan *data testing* 30% dari total 15.053 data ulasan, terdapat 10.537 data ulasan yang digunakan sebagai *data training* dan 4.516 data ulasan yang digunakan sebagai *data testing*.

3) Pembagian *data training* sebesar 80% dan *data testing* sebesar 20%

Tabel 4. 18 Pembagian Data Skenario 3

Klasifikasi	Jumlah	<i>Data Training</i> (80%)	<i>Data Testing</i> (20%)
Positif	11.772	9.418	2354
Negatif	3.281	2.625	656
Total	15.053	12.043	3.010

Berdasarkan pada Tabel 4.18 pada pembagian *data training* sebesar 80% dan *data testing* 20% dari total 15.053 data ulasan, terdapat 12.043 data ulasan yang digunakan sebagai *data training* dan 3.010 data ulasan yang digunakan sebagai *data testing*.

4) Pembagian *data training* sebesar 90% dan *data testing* sebesar 10%

Tabel 4. 19 Pembagian Data Skenario 4

Klasifikasi	Jumlah	<i>Data Training</i> (90%)	<i>Data Testing</i> (10%)
Positif	11.772	10.595	1.177
Negatif	3.281	2.953	328
Total	15.053	13.548	1.505

Berdasarkan pada Tabel 4.19 pada pembagian *data training* sebesar 90% dan *data testing* 10% dari total 15.053 data ulasan, terdapat 13.548 data ulasan yang digunakan sebagai *data training* dan 1.505 data ulasan yang digunakan sebagai *data testing*.

Berikut merupakan *pseudocode* dari proses perhitungan skor sentimen dapat dilihat pada Tabel 4.20

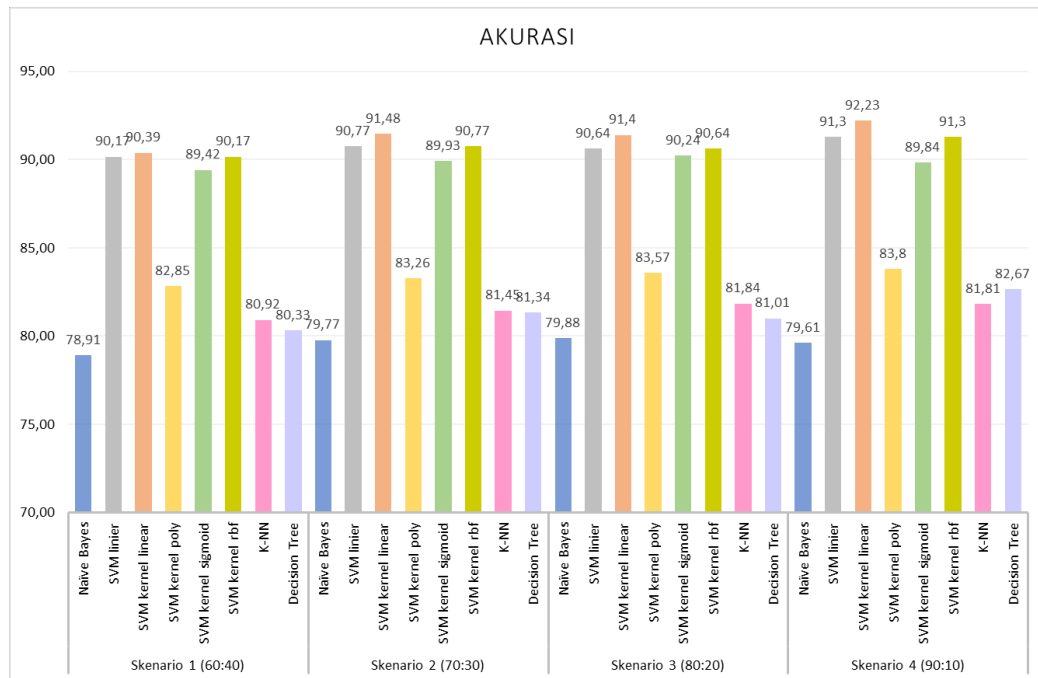
Tabel 4. 20 *Pseudocode* dari Proses Perhitungan Skor Sentimen

<i>Pseudocode</i>	<b>Fungsi</b>
<pre> from sklearn.model_selection import train_test_split  X_train, X_test, y_train, y_test = train_test_split(text_tf, datanew['analysis'], test_size=0.1, random_state= 1)  X_train, X_test, y_train, y_test = train_test_split(text_tf, datanew['analysis'], test_size=0.2, random_state= 1)  X_train, X_test, y_train, y_test = train_test_split(text_tf, datanew['analysis'], test_size=0.3, random_state= 1)  X_train, X_test, y_train, y_test = train_test_split(text_tf, datanew['analysis'], test_size=0.4, random_state= 1) </pre>	<p>Melakukan pembagian data latih dan data uji dengan perbandingan 90:10, 80:20, 70:30, dan 60:40.</p>

## 2. *Klasifikasi Algoritma Support Vector Machine, Naïve Bayes Classifier, K-Nearest Neighbor, dan Decision Tree*

Klasifikasi pada penelitian ini dilakukan menggunakan 4 algoritma yaitu *Support Vector Machine* (SVM), *Naïve Bayes Classifier* (NBC), *K-Nearest Neighbor* (K-NN), dan *Decision Tree* (DT). Pada algoritma SVM terdiri atas SVM *linear* dan SVM *non-linear*, dimana SVM *non-linear* menggunakan 4 kernel yaitu *Linear*, *Polynomial*, *Radial Basis Function* (RBF), dan *Sigmoid*. Adapun hasil klasifikasi dari setiap algoritma dilihat dari akurasi dapat dilihat pada Gambar 4.5.





Gambar 4. 5 Hasil Akurasi Masing-Masing Algoritma

Berdasarkan hasil pada Gambar 4.5, diperoleh nilai akurasi pada setiap algoritma. Terdapat 4 skenario yang digunakan yaitu skenario 1 dengan perbandingan data latih 60% dan data uji 40%, skenario 2 dengan perbandingan data latih 70% dan data uji 30%, skenario 3 dengan perbandingan data latih 80% dan data uji sebesar 20%, dan skenario 4 dengan perbandingan data latih 90% dan data uji sebesar 10%. Pada algoritma *SVM linear* diperoleh nilai akurasi skenario 1 sebesar 90,17%, skenario 2 sebesar 90,77%, skenario 3 sebesar 90,64%, dan skenario 4 sebesar 91,30%.

Pada algoritma *SVM kernel linear (nonlinear)* diperoleh nilai akurasi skenario 1 sebesar 90,39%, skenario 2 sebesar 91,48%, skenario 3 sebesar 91,40%, dan skenario 4 sebesar 92,23%. Pada algoritma *SVM kernel poly* diperoleh nilai akurasi skenario 1 sebesar 82,85%, skenario 2 sebesar 83,26%, skenario 3 sebesar 83,57%, dan skenario 4 sebesar 83,80%. Pada algoritma *SVM kernel sigmoid* diperoleh nilai akurasi skenario 1 sebesar 89,11%, skenario 2 sebesar 89,28%, skenario 3 sebesar 89,21%, dan skenario 4 sebesar 88,31%. Pada algoritma *SVM kernel rbf* diperoleh nilai akurasi skenario 1 sebesar 89,6%, skenario 2 sebesar 90,68%, skenario 3 sebesar 90,34%, dan skenario 4 sebesar 89,51%.

Pada algoritma *Naive Bayes* diperoleh nilai akurasi skenario 1 sebesar 78,76%, skenario 2 sebesar 79,19%, skenario 3 sebesar 79,67%, dan skenario 4 sebesar 78,55%.

Pada algoritma K-NN diperoleh nilai akurasi skenario 1 sebesar 81,75%, skenario 2 sebesar 83,02 %, skenario 3 sebesar 82,27%, dan skenario 4 sebesar 81,01%. Pada algoritma *Decision Tree* diperoleh nilai akurasi skenario 1 sebesar 81,07%, skenario 2 sebesar 81,09%, skenario 3 sebesar 80,17%, dan skenario 4 sebesar 79,35%.

Berdasarkan grafik akurasi, Algoritma *SVM kernel linier* dengan skenario data latih 90% dan data uji 10% memperoleh nilai akurasi tertinggi daripada algoritma lainnya yaitu sebesar 92,23%. Sehingga algoritma yang dipilih untuk klasifikasi. Adapun hasil *confusion matrix* dari algoritma yang dipilih yaitu sebagai berikut.

Tabel 4. 21 *Confusion Matrix*

Sentimen		Aktual	
		Positif	Negatif
Prediksi	Positif	<i>True Positive (TP)</i> 1090	<i>False Positive (FP)</i> 46
	Negatif	<i>False Negative (FN)</i> 71	<i>True Negative (TN)</i> 299

Berdasarkan hasil dari *confusion matrix*, diketahui jumlah ulasan positif yang diprediksi positif (*true positive*) yaitu sebanyak 1.090 ulasan dan jumlah ulasan negatif yang diprediksi negatif (*true negative*) yaitu sebanyak 299 ulasan. Sedangkan jumlah ulasan positif yang diprediksi negatif (*false negative*) sebanyak 71 ulasan dan jumlah ulasan negatif yang diprediksi positif (*false positive*) sebanyak 46 ulasan.

Berikut merupakan *pseudocode* dari klasifikasi setiap algoritma dapat dilihat pada Tabel 4.22.

Tabel 4. 22 *Pseudocode* dari Klasifikasi Setiap Algoritma

<i>Pseudocode</i>	Fungsi
NB = MultinomialNB().fit(X_train, y_train) predicted = NB.predict(X_test)	1. Membuat model

<i>Pseudocode</i>	<b>Fungsi</b>
<pre>print("MultinomialNB accuracy:" , accuracy_score(y_test,predicted)) print("MultinomialNB Precision:" , precision_score(y_test,predicted, average="binary", pos_label="Negative")) print("MultinomialNB Recall:" , recall_score(y_test,predicted, average="binary", pos_label="Negative")) print("MultinomialNB f1_score:" , f1_score(y_test,predicted, average="binary", pos_label="Negative")) print(f'confusion matrix:\n{confusion_matrix(y_test, predicted)}') print('=====\\n') print(classification_report(y_test, predicted, zero_division=0))</pre>	<p>algoritma Naïve Bayes</p>
<pre>SVM_linear = svm.SVC().fit(X_train, y_train) predicted = SVM_linear.predict(X_test) print("SVM accuracy:" , accuracy_score(y_test,predicted)) print("SVM Precision:" , precision_score(y_test,predicted, average="binary", pos_label="Negative")) print("SVM Recall:" , recall_score(y_test,predicted, average="binary", pos_label="Negative")) print("SVM f1_score:" , f1_score(y_test,predicted, average="binary", pos_label="Negative")) print(f'confusion matrix:\n{confusion_matrix(y_test, predicted)}') print('=====\\n') print(classification_report(y_test, predicted, zero_division=0))</pre>	<p>2. Membuat model algoritma SVM Linier</p>
<pre>SVM_kernel_linear = svm.SVC(kernel='linear').fit(X_train, y_train) predicted = SVM_kernel_linear.predict(X_test) print("SVM accuracy:" , accuracy_score(y_test,predicted)) print("SVM Precision:" , precision_score(y_test,predicted, average="binary", pos_label="Negative")) print("SVM Recall:" , recall_score(y_test,predicted, average="binary", pos_label="Negative"))</pre>	<p>3. Membuat model algoritma SVM Kernel Linear</p>

<i>Pseudocode</i>	<b>Fungsi</b>
<pre> print("SVM fl_score:" , fl_score(y_test,predicted, average="binary", pos_label="Negative")) print(f'confusion matrix:\n{confusion_matrix(y_test, predicted)}') print('=====\\n') print(classification_report(y_test, predicted, zero_division=0))  SVM_kernel_poly = svm.SVC(kernel='poly').fit(X_train, y_train) predicted = SVM_kernel_poly.predict(X_test) print("SVM accuracy:" , accuracy_score(y_test,predicted)) print("SVM Precision:" , precision_score(y_test,predicted, average="binary", pos_label="Negative")) print("SVM Recall:" , recall_score(y_test,predicted, average="binary", pos_label="Negative")) print("SVM fl_score:" , fl_score(y_test,predicted, average="binary", pos_label="Negative")) print(f'confusion matrix:\n{confusion_matrix(y_test, predicted)}') print('=====\\n') print(classification_report(y_test, predicted, zero_division=0))  SVM_kernel_sigmoid = svm.SVC(kernel='sigmoid').fit(X_train, y_train) predicted = SVM_kernel_sigmoid.predict(X_test) print("SVM accuracy:" , accuracy_score(y_test,predicted)) print("SVM Precision:" , precision_score(y_test,predicted, average="binary", pos_label="Negative")) print("SVM Recall:" , recall_score(y_test,predicted, average="binary", pos_label="Negative")) print("SVM fl_score:" , fl_score(y_test,predicted, average="binary", pos_label="Negative")) print(f'confusion matrix:\n{confusion_matrix(y_test, predicted)}') print('=====\\n') print(classification_report(y_test, predicted, zero_division=0)) </pre>	<p>4. Membuat model algoritma SVM Kernel Poly</p> <p>5. Membuat model algoritma SVM Kernel Sigmoid</p>

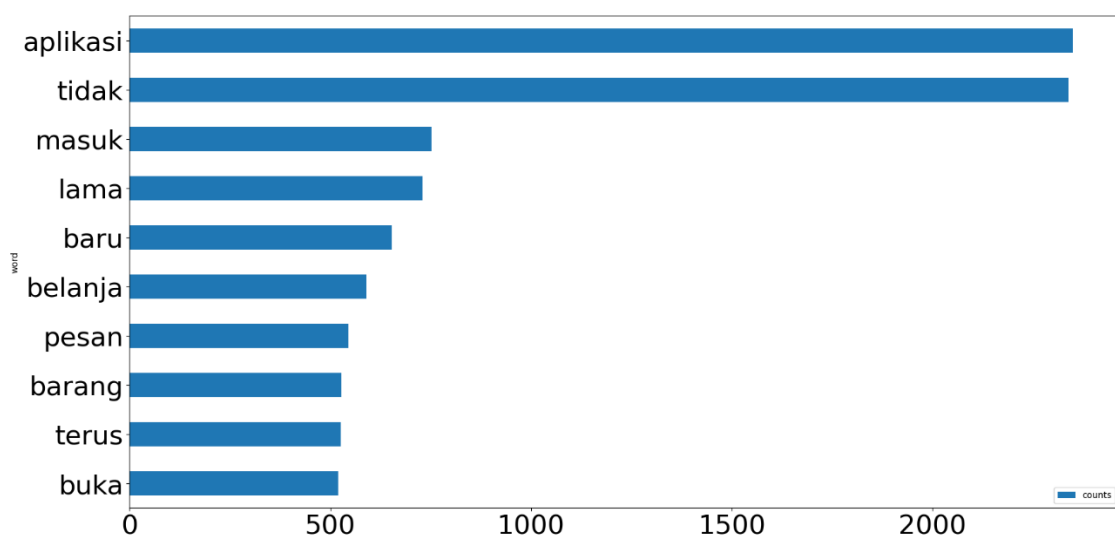
<i>Pseudocode</i>	<b>Fungsi</b>
<pre> SVM_kernel_rbf = svm.SVC(kernel='rbf').fit(X_train, y_train) predicted = SVM_kernel_rbf.predict(X_test) print("SVM accuracy:" , accuracy_score(y_test,predicted)) print("SVM Precision:" , precision_score(y_test,predicted, average="binary", pos_label="Negative")) print("SVM Recall:" , recall_score(y_test,predicted, average="binary", pos_label="Negative")) print("SVM f1_score:" , f1_score(y_test,predicted, average="binary", pos_label="Negative")) print(f'confusion matrix:\n{confusion_matrix(y_test, predicted)}') print('=====\\n') print(classification_report(y_test, predicted, zero_division=0))  KNN = KNeighborsClassifier(n_neighbors=3, weights= 'distance').fit(X_train, y_train) predicted = KNN.predict(X_test) print("KNN accuracy:" , accuracy_score(y_test,predicted)) print("KNN Precision:" , precision_score(y_test,predicted, average="binary", pos_label="Negative")) print("KNN Recall:" , recall_score(y_test,predicted, average="binary", pos_label="Negative")) print("KNN f1_score:" , f1_score(y_test,predicted, average="binary", pos_label="Negative")) print(f'confusion matrix:\n{confusion_matrix(y_test, predicted)}') print('=====\\n') print(classification_report(y_test, predicted, zero_division=0))  DT = tree.DecisionTreeClassifier().fit(X_train, y_train) predicted = DT.predict(X_test) print("Decision Tree accuracy:" , accuracy_score(y_test,predicted)) print("Decision Tree Precision:" , precision_score(y_test,predicted, average="binary", pos_label="Negative")) </pre>	<p>6. Membuat model algoritma SVM Kernel RBF</p> <p>7. Membuat model algoritma KNN</p>

<i>Pseudocode</i>	<b>Fungsi</b>
<pre>print("Decision Tree Recall:" , recall_score(y_test,predicted, average="binary", pos_label="Negative")) print("Decision Tree fl_score:" , fl_score(y_test,predicted, average="binary", pos_label="Negative")) print(f'confusion matrix:\n{confusion_matrix(y_test, predicted)}') print('=====\n') print(classification_report(y_test, predicted, zero_division=0))</pre>	<p>8. Membuat model Algoritma <i>Decision Tree</i></p>

#### 4.2.2.6 Visualisasi dan Asosiasi

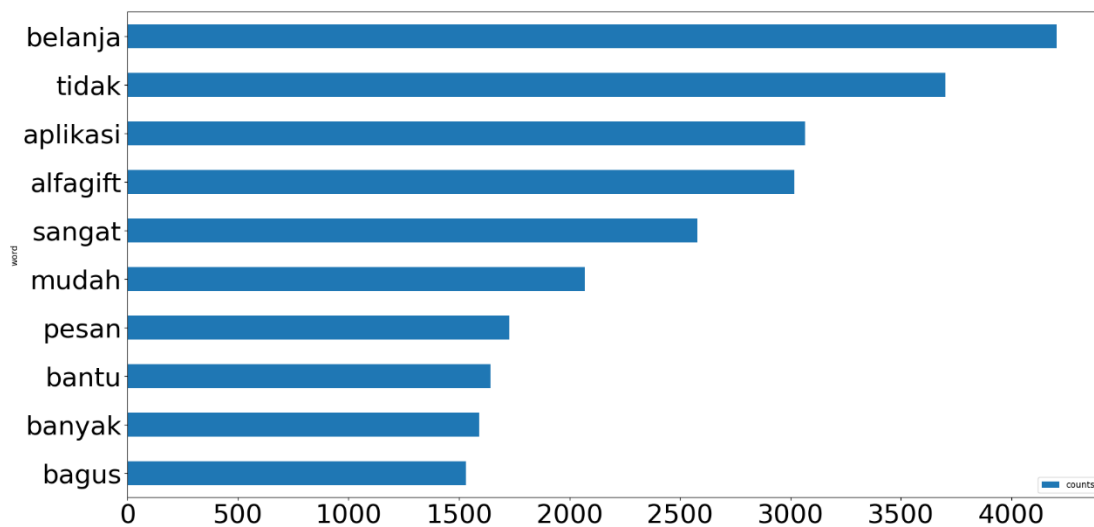
##### 1) Visualisasi Data

Visualisasi data dilakukan untuk menunjukkan persebaran kata yang paling sering dibahas oleh pengguna aplikasi Alfabeta. Visualisasi dilakukan pada masing-masing kelas sentimen baik positif maupun negatif. Berikut merupakan visualisasi 10 teratas kata pada kelas sentimen negatif:



Gambar 4. 6 Persebaran Kata Kelas Sentimen Negatif

Berdasarkan visualisasi persebaran kata kelas sentimen negatif, kata-kata yang paling sering dibahas adalah kata “aplikasi” sebanyak 2.349 kali, kata “tidak” sebanyak 2.338 kali, kata “masuk” sebanyak 752 kali, kata “lama” sebanyak 730 kali, kata “baru” sebanyak 654 kali, kata “belanja” sebanyak 589 kali, kata “pesan” sebanyak 545 kali, kata “barang” sebanyak 528 kali, kata “terus” sebanyak 526 kali, dan kata “buka” sebanyak 519 kali. Adapun visualisasi 10 teratas kata pada kelas sentimen positif:



Gambar 4. 7 Persebaran Kata Kelas Sentimen Positif

Berdasarkan visualisasi persebaran kata kelas sentimen positif, kata-kata yang paling sering dibahas adalah kata “belanja” sebanyak 4.205 kali, kata “tidak” sebanyak 3.701 kali, kata “aplikasi” sebanyak 3.066 kali, kata “alfagift” sebanyak 3.017 kali, kata “sangat” sebanyak 2.580 kali, kata “mudah” sebanyak 2.069 kali, kata “pesan” sebanyak 1.727 kali, kata “bantu” sebanyak 1.642 kali, kata “banyak” sebanyak 1.592 kali, dan kata “bagus” sebanyak 1.532 kali. Akan tetapi berdasarkan persebaran kata kelas sentimen positif terdapat kata “tidak” yang memiliki makna negatif. Hal ini dikarenakan dalam suatu data ulasan dengan kelas sentimen positif tidak selalu terdiri dari kata-kata positif saja, tetapi dapat terdiri dari kata negatif juga seperti yang ditunjukkan pada data ulasan ke-56 pada Gambar 4. 8. Dikarenakan data ulasan tersebut memiliki kelas sentimen positif, sehingga kata “tidak” pada data ulasan dapat muncul pada persebaran kata positif.

1	text awal	label
53	keren alfa gift pas kendala apa langsung respon kirim selalu tepat waktu tidak besok besok	Positive
54	aplikasi bisa lihat promo	Positive
55	good	Positive
56	enak unduh aplikasi tinggal duduk rumah lalu pilih pesan butuh bayar gampang tidak rumit pokok top deh alfa gift gratis ongkir belanja jadi gampang terimakasih alfa gift	Positive

Gambar 4. 8 Contoh Data Ulasan dengan Kelas Sentimen Positif

## 2) Asosiasi Kata

Tahapan asosiasi dilakukan untuk mengetahui kata yang sering muncul secara bersamaan. Berikut merupakan hasil *association rules* dapat dilihat pada Tabel 4. 23.

Tabel 4. 23 Hasil *Association Rules*

No	<i>Antecedents</i>	<i>Consequents</i>	<i>Support</i>	<i>Confidence</i>	<i>Lift</i>
1	Buka, lambat	aplikasi	0,012626	0,92	1,835465
2	Bagus	aplikasi	0,052703	0,799999	1,596056
3	mudah	aplikasi	0,015097	0,723684	1,443801
4	Rugi	aplikasi	0,011254	0,621212	1,239362
5	Barang, stok	kosong	0,010156	0,606557	1,712936
6	kecewa	aplikasi	0,021410	0,6	1,970427
7	Muat, lama	aplikasi	0,020861	0,589147	1,175390
8	buruk	aplikasi	0,015371	0,583333	1,163791
9	kosong, stok	barang	0,010156	0,569230	5,924879
10	Error, terus	aplikasi	0,012077	0,564102	1,125424
11	cod	tidak	0,014822	0,5625	1,275163
12	eror	aplikasi	0,039527	0,539325	1,075993
13	poin	masuk	0,020038	0,506944	3,211823
14	antar	lama	0,015646	0,419117	2,389429
15	Kirim, tidak	pesan	0,013175	0,328767	2,804914
16	struk	tidak, masuk	0,010156	0,324561	3,649312
17	Aplikasi, susah	buka	0,014273	0,315151	2,568449
18	salah	pesan	0,011528	0,313432	2,674088



No	<i>Antecedents</i>	<i>Consequents</i>	<i>Support</i>	<i>Confidence</i>	<i>Lift</i>
19	salah	barang	0,011528	0,313432	3,262388
20	kurang	barang	0,011528	0,311111	3,238222
21	kecewa	barang	0,010156	0,284615	2,962439

Berikut merupakan *pseudocode* dari *association rules* dapat dilihat pada Tabel 4.24.

Tabel 4. 24 *Pseudocode* dari *Association Rules*

<b>Pseudocode</b>	<b>Fungsi</b>
rules = association_rules(freq_items, metric="confidence", min_threshold=0.25)	Menjalankan <i>association rules</i>

Dalam tahapan *association rule* terdapat beberapa *output* yaitu *support*, *confidence*, dan *lift ratio*. *Support* merupakan persentase kombinasi item dalam data. *Confidence* (nilai kepastian) merupakan kuatnya hubungan antar-item dalam aturan asosiasi. Sedangkan *lift ratio* merupakan suatu parameter yang digunakan untuk melihat kuat atau tidaknya hubungan antar aturan asosiasi. Nilai *lift ratio* umumnya dijadikan sebagai penentu valid atau tidaknya aturan asosiasi. Jika nilai *lift ratio* > 1, maka *rules* bersifat kuat atau valid untuk digunakan sebagai acuan dalam memprediksi. Semakin tinggi nilai *lift ratio*, maka semakin besar kekuatan asosiasi. Adapun hasil asosiasi yang terpilih dapat dilihat pada Tabel 4.25. Diambil beberapa asosiasi pada dari Tabel 4. 24 dikarenakan terdapat asosiasi kata yang memiliki makna yang sama seperti asosiasi nomor 1 dan 7 pada Tabel 4. 24

Tabel 4. 25 Ringkasan Asosiasi Terpilih

No	Kata Awal	Kata lanjutan
1	<b>lambat</b>	aplikasi
2	<b>bagus</b>	aplikasi
3	<b>mudah</b>	aplikasi
4	<b>rugi</b>	aplikasi
5	stok	<b>kosong</b>

No	Kata Awal	Kata lanjutan
6	kecewa	aplikasi
7	eror	aplikasi
8	cod	<b>tidak (bisa)</b>
9	poin	<b>masuk</b>
10	antar	<b>lama</b>
11	<b>tidak kirim</b>	pesan
12	struk	<b>tidak masuk</b>
13	<b>salah</b>	barang
14	<b>kurang</b>	barang
15	<b>kecewa</b>	barang

Berdasarkan hasil asosiasi, diperoleh pasangan *kansei word* yang terpilih. Pada pasangan *kansei word* yang tercantum pada Tabel 4.25, diambil salah satu *kansei word* yang paling dominan mewakili emosi pelanggan contohnya seperti *kansei word* “kosong” diikuti kata yang menyertainya yaitu “stok”, dimana kata yang mewakili emosi pelanggan adalah kata “kosong”. Berikut merupakan *Kansei word* terpilih dapat dilihat pada Tabel 4.26.

Tabel 4. 26 *Kansei Words* Terpilih

<i>Kansei Words</i>
Lambat
Bagus
Mudah
Rugi
Kosong
Kecewa
Eror
Tidak bisa
Masuk
Lama
Tidak dikirim

<i>Kansei Words</i>
Tidak masuk
Salah
Kurang
Kecewa

*Kansei word* dapat berupa kata sifat, kata kerja, dan kata benda, tetapi umumnya kata sifat lebih sering digunakan (Schütte, 2005). Pengelompokan jenis kata pada *kansei word* dilakukan dengan bantuan Kamus Besar Bahasa Indonesia Daring yang tersedia pada laman (Kemendikbud, 2016). Berikut hasil pengelompokan jenis kata pada *kansei word* terpilih yang ditunjukkan pada Tabel 4. 27.

Tabel 4. 27 Pengelompokan Jenis Kata pada *Kansei Word* Terpilih

<i>Kansei Words</i>	Kata Sifat	Kata Kerja	Kata Benda
Lambat	✓		✓
Bagus	✓		
Mudah	✓		
Rugi	✓	✓	
Kosong	✓		
Kecewa	✓		
Eror			✓
Tidak bisa		✓	
Masuk		✓	
Lama	✓		
Tidak dikirim		✓	
Tidak masuk		✓	
Salah	✓		✓
Kurang	✓		
Kecewa	✓		

Pada penelitian ini menggunakan *kansei word* yang memiliki makna positif sehingga *kansei word* dengan makna negatif akan diubah dahulu menjadi makna positif.

Tabel 4. 28 *Kansei Word* Positif

<b>Lawan Kata <i>Kansei Word</i></b>	<b><i>Kansei Word</i> Positif</b>
Lambat >< Cepat	Cepat
Bagus >< Buruk	Bagus
Mudah >< Susah	Mudah
Rugi >< Untung	Untung
Kosong >< Lengkap	Lengkap
Kecewa >< Puas	Puas
Eror >< Tidak Eror	Eror
Tidak Bisa >< Bisa	Bisa
Masuk >< Tidak masuk	Masuk
Lama >< Cepat	Cepat
Tidak dikirim >< Dikirim	Dikirim
Tidak masuk >< Masuk	Masuk
Salah >< Benar	Benar
Kurang >< Cukup	Cukup
Kecewa >< Puas	Puas

Berdasarkan hasil pada table 4. 28 menunjukkan *kansei word* yang diubah menjadi makna positif. Contohnya seperti kata “lambat” menjadi “cepat” dan “kecewa” menjadi “puas”.

### 4.3 Perentangan Ruang Properti

Untuk mengetahui apakah sebuah elemen suatu produk atau jasa secara individual atau puluhan elemen dapat berpengaruh besar pada *kansei*, diperlukan identifikasi semua atribut produk (Nagamachi & Lokman, 2011). Atribut yang dipilih hanya yang memiliki dampak terbesar atau potensial terhadap peningkatan kualitas aplikasi Alfagift.

#### 4.3.1 Penentuan Atribut

Atribut pada penelitian ini yaitu sistem, informasi, dan pelayanan yang diberikan oleh Aplikasi *Alfagift* kepada pengguna aplikasi. Atribut pada penelitian ini ditentukan berdasarkan *kansei words* yang diperoleh pada proses perentangan ruang semantik sebelumnya. *Kansei words* mewakili masing-masing perasaan dan emosi konsumen terhadap aplikasi *Alfagift* baik dari segi sistem, informasi, maupun layanan. Pada penelitian ini Atribut ditentukan menggunakan *kansei word* yang memiliki makna positif sehingga *kansei word* dengan makna negatif akan diubah dahulu menjadi makna positif. Berikut merupakan salah contoh dari atribut yang menggunakan *kansei word* positif dapat dilihat pada Tabel 4. 29 dan Tabel 4. 30.

Tabel 4. 29 Atribut dengan *Kansei Word* Negatif

Atribut
Aplikasi <i>Alfagift</i> <b>lambat</b> dibuka

Tabel 4. 30 Atribut dengan *Kansei Word* Positif

Atribut
Aplikasi <i>Alfagift</i> <b>cepat</b> dibuka

Adapun atribut yang didapatkan setelah *kansei word* negatif diubah menjadi *kansei word* positif dapat dilihat pada Tabel 4.31

Tabel 4. 31 Atribut

Atribut
Aplikasi <i>Alfagift</i> <b>cepat</b> dibuka
Aplikasi dirancang dengan <b>bagus</b>
Aplikasi <b>mudah</b> digunakan
Belanja menggunakan aplikasi <b>menguntungkan</b>
Stok barang di aplikasi <b>lengkap</b>
Layanan aplikasi <b>memuaskan</b>
Aplikasi jarang terjadi <b>error</b>
Belanja menggunakan aplikasi <b>bisa</b> COD
Poin <b>masuk</b> ke dalam aplikasi

<b>Atribut</b>
Pengantaran barang / pesanan <b>cepat</b>
Pesanan <b>dikirim</b> ke lokasi pembeli
Struk <b>masuk</b> (tersimpan) ke dalam aplikasi
Barang yang datang <b>sesuai (benar)</b> pesanan
Barang yang datang jumlahnya <b>cukup</b>
Kondisi barang yang datang <b>memuaskan</b>

Berdasarkan hasil pada tabel 4. 31 menunjukkan atribut yang dihasilkan setelah *kansei word* yang memiliki makna negatif diubah menjadi *kansei word* bermakna positif.

#### 4.3.2 Pengambilan Data Kano

Model Kano mengklasifikasikan jawaban dari atribut yang diberikan kepada pelanggan berdasarkan bagaimana persepsi pelanggan terhadap layanan yang diberikan ke dalam kategori masing-masing; *must-be*, *one-dimensional*, *attractive*, *reverse*, dan *indifferent attributes* (Kano et al., 1984). Pengambilan data *kano* dilakukan menggunakan kuesioner kano untuk mengukur fungsional dan disfungsional layanan dari aplikasi. Atribut layanan pada Tabel 4. 30 digunakan sebagai pernyataan pada kuesioner kano. Kuesioner Kano menyusun atribut layanan sebagai pernyataan yang bersifat positif (fungsional) dan pernyataan yang bersifat negatif (disfungsional) sehingga dapat diketahui dampak kepuasan pelanggan jika layanan disediakan atau tidak disediakan. Kuesioner ini bersifat tertutup, dimana telah diberikan pilihan jawaban berdasarkan evaluasi model kano pada Tabel 2. 2 yaitu suka, harap, netral, toleransi, dan tidak suka. Kuesioner *kano* disebar kepada 50 responden. Menurut Roscoe dan Sugiyono, ukuran sampel yang layak dalam penelitian yaitu antara 30 – 500 orang. Sebagaimana yang dikemukakan oleh Bailey, ukuran paling minimum sampel yang menggunakan analisis data dengan statistik yaitu sebanyak 30 responden (Bailey, 1999). Responden dalam penelitian ini harus memenuhi kedua kriteria yaitu pernah menggunakan aplikasi *Alfagift* dan pernah belanja menggunakan aplikasi *Alfagift*. Kuesioner terdiri dari dua bagian yaitu bagian fungsional dan bagian disfungsional dapat dilihat pada Lampiran A-7. 1 dan Lampiran A-7. 2. Berikut merupakan pertanyaan-pertanyaan pada kuesioner *kano* dapat dilihat pada Tabel 4.32 dan Tabel 4.33.

Tabel 4. 32 Kuesioner *Kano* Fungsional

<b>Bagian Fungsional</b>				
Barang yang datang jumlahnya <b>cukup</b>				
Suka	Mengharapkan	Netral	Toleransi	Tidak Suka

Tabel 4. 33 Kuesioner *Kano* Disfungsional

<b>Bagian Disfungsional</b>				
Barang yang datang jumlahnya <b>kurang</b>				
Suka	Mengharapkan	Netral	Toleransi	Tidak Suka

#### 4.3.3 Uji Validitas

Pada tahapan ini dilakukan pengujian validitas dari hasil rekapitulasi kuesioner *kano*. Uji validitas dilakukan dengan bantuan *software* SPSS 25. Berikut merupakan hasil dari uji validitas dapat dilihat pada Tabel 4.34.

Tabel 4. 34 Hasil Uji Validitas Kuesioner *Kano*

Atribut	r Tabel	r Hitung Fungsional	r Hitung Disfungsional	Validitas (r hitung > r tabel)
X1	0,278	0,509	0,689	Valid
X2	0,278	0,451	0,782	Valid
X3	0,278	0,746	0,728	Valid
X4	0,278	0,640	0,482	Valid
X5	0,278	0,430	0,533	Valid
X6	0,278	0,614	0,731	Valid
X7	0,278	0,560	0,598	Valid
X8	0,278	0,398	0,297	Valid
X9	0,278	0,303	0,648	Valid
X10	0,278	0,556	0,642	Valid
X11	0,278	0,601	0,755	Valid
X12	0,278	0,351	0,475	Valid

Atribut	r Tabel	r Hitung Fungsional	r Hitung Disfungsional	Validitas (r hitung > r tabel)
X13	0,278	0,464	0,662	Valid
X14	0,278	0,503	0,697	Valid
X15	0,278	0,462	0,557	Valid

Berdasarkan hasil uji validitas kuesioner kano, diperoleh nilai  $r_{hitung}$  pada setiap pertanyaan lebih besar daripada  $r_{tabel}$  (0,278) dengan nilai signifikansi 0,05 (*two tailed*) dan N berjumlah 50 sampel. Maka dapat disimpulkan pertanyaan dinyatakan valid untuk dijadikan alat ukur penelitian ini.

#### 4.3.4 Uji Reliabilitas

Uji reliabilitas dilakukan untuk mengetahui sejauh mana suatu instrumen dapat dipercaya (reliabel). Pada uji ini dilakukan menggunakan SPSS 25 dengan melihat nilai *Cronbach's Alpha* dimana suatu variabel atau konstruk dinyatakan reliabel apabila nilai *Cronbach's Alpha* > 0,60 (Ghozali, 2013). Berikut merupakan hasil dari uji reliabilitas dapat dilihat pada Tabel 4.35.

Tabel 4. 35 Hasil Uji Reliabilitas

<i>Cronbach's Alpha</i>	
Kano Fungsional	Kano Disfungsional
0,781	0,875

Berdasarkan hasil di atas, diperoleh nilai *Cronbach's Alpha* > 0,60 yaitu sebesar 0,781 pada pertanyaan fungsional dan 0,875 pada pertanyaan disfungsional. Sehingga dapat disimpulkan semua butir pertanyaan dinyatakan reliabel.

#### 4.4 Sintesis

Pada tahapan ini, ruang semantik dan ruang properti dihubungkan bersama yaitu dengan menghubungkan *kansei words* dan atribut produk atau layanan. Atribut produk atau layanan yang ditentukan berdasarkan *kansei words* dikategorikan menggunakan metode *Kano* untuk mengetahui atribut manakah yang memiliki dampak terbesar terhadap pelanggan



#### 4.4.1 Menentukan Klasifikasi Tiap Atribut Berdasarkan Kategori Kano

Pada tahapan ini, data kuesioner kano yang telah disebar kepada 50 responden diklasifikasikan setiap atribut berdasarkan kategori kano. Pengelompokan dilihat berdasarkan nilai yang diberikan oleh responden pada pertanyaan fungsional dan pertanyaan disfungsional. Tabel evaluasi kano pada Tabel 2. 2 digunakan untuk mengetahui kategori setiap atribut-atribut dari setiap responden. Dalam menentukan kategori kano dari setiap nilai atribut yang diberikan responden, dilihat dari nilai titik temu atribut fungsional dan disfungsional. Sehingga diperoleh hasil klasifikasi kategori kano setiap atribut yang dapat dilihat pada Tabel 4.36.

Tabel 4. 36 Klasifikasi Kategori Kano dari Data Kuesioner

Responden	Atribut															
	X 1	X 2	X 3	X 4	X 5	X 6	X 7	X 8	X 9	X1 0	X1 1	X1 2	X1 3	X1 4	X1 5	
1	O	M	M	M	I	M	A	A	O	M	O	A	M	M	M	
2	I	I	I	M	M	M	M	O	M	O	O	O	O	O	O	
3	M	I	M	A	I	M	I	A	A	O	M	I	O	I	M	
4	I	I	I	O	I	I	M	I	I	I	A	I	I	M	I	
5	A	M	M	O	O	O	O	M	O	O	O	O	O	O	O	
6	O	M	M	O	I	I	O	O	O	I	O	O	O	O	M	
7	I	O	O	O	I	O	I	O	O	O	O	O	I	I	O	
8	M	M	I	M	A	M	A	I	M	I	M	O	O	O	O	
9	I	M	M	O	I	M	I	M	O	O	O	O	O	O	O	
10	I	I	I	O	I	I	I	O	O	I	O	O	O	O	O	
11	I	I	A	O	I	O	I	I	I	I	M	I	O	O	A	
12	I	I	I	I	I	I	I	A	A	I	I	A	A	A	I	
13	I	I	I	I	R	I	I	I	A	R	M	O	O	M	M	
14	M	O	O	O	I	O	I	O	O	I	O	O	O	O	O	
15	O	O	A	O	O	M	M	O	M	O	O	O	O	M	O	
16	O	O	O	O	O	O	O	R	O	O	O	O	O	O	Q	
17	A	I	M	A	I	I	O	I	M	I	O	O	A	O	A	

Responden	Atribut														
	X1	X1	X1	X1	X1	X1	X1	X1	X1	X1	X1	X1	X1	X1	X1
18	I	O	O	M	I	M	I	O	O	A	O	A	O	O	O
19	M	O	O	I	I	I	I	O	O	M	O	O	M	O	M
20	A	I	I	I	I	I	I	I	I	I	A	I	A	A	I
21	A	A	A	O	I	O	A	O	O	I	O	O	O	O	M
22	M	O	O	M	M	M	M	O	O	O	O	O	M	O	M
23	I	I	I	M	M	M	I	O	O	O	O	O	O	O	O
24	I	I	I	I	R	I	A	I	O	I	O	I	O	O	O
25	O	O	O	O	M	O	O	O	O	O	O	I	O	O	O
26	R	R	I	O	R	I	R	O	O	I	O	O	O	O	O
27	M	M	M	O	M	O	M	I	M	O	O	A	O	O	O
28	I	I	I	I	I	A	I	I	O	M	I	M	O	O	M
29	I	R	A	I	A	M	I	O	A	I	M	O	A	A	A
30	A	A	O	M	M	M	M	O	A	I	I	A	I	I	A
31	I	I	I	I	I	I	I	I	I	I	O	I	I	I	I
32	M	M	O	O	O	O	O	M	O	O	O	O	O	O	O
33	M	M	M	M	I	I	M	I	O	Q	O	O	O	M	O
34	I	M	O	O	I	I	M	O	I	I	O	O	O	O	O
35	M	M	M	I	I	M	M	I	O	I	O	O	O	O	O
36	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
37	M	M	M	I	M	M	O	I	I	O	M	I	O	O	O
38	I	I	I	M	M	I	I	I	I	I	M	M	M	M	M
39	I	M	I	M	A	I	I	I	I	I	I	I	M	I	M
40	I	I	O	O	I	O	O	O	O	O	O	O	A	A	A
41	I	I	O	O	I	M	I	O	O	O	O	O	O	O	M
42	M	M	M	I	M	M	I	O	O	I	O	O	O	O	O
43	O	M	O	O	I	O	O	O	O	O	O	O	O	O	O
44	R	I	R	M	I	I	R	O	O	I	O	O	O	O	O
45	I	M	M	M	M	M	M	I	M	O	O	M	M	M	M
46	I	I	I	I	I	I	I	I	M	I	I	I	I	I	I

Responden	Atribut															
	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	
47	I	I	I	A	A	A	I	A	I	A	A	A	I	I	A	
48	I	I	I	I	I	I	I	I	A	I	I	O	I	I	I	
49	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
50	I	I	I	A	I	I	I	I	A	I	A	M	A	A	A	

Berdasarkan hasil klasifikasi kategori kano pada Tabel 4. 35, pada salah satu responden yaitu responden 1 memberikan jawaban skala 1 (suka) untuk bagian fungsional kuesioner dan skala 5 (tidak suka) untuk bagian disfungsional kuesioner pada atribut X1, X9, dan X11. Sehingga berdasarkan kategorisasi model kano pada Tabel 2. 2, ketiga atribut tersebut dikategorikan sebagai *one-dimensional*. Kemudian pada atribut X2, X3, X4, X6, X10, X13, X14, dan X15, responden tersebut memberikan jawaban skala 2-5 (harap – tidak suka) dan 3-5 (netral – tidak suka) pada bagian fungsional-disfungsional kuesioner sehingga dikategorikan sebagai atribut *must-be*. Pada atribut X7, X8, dan X12, responden tersebut memberikan jawaban skala 1-2 (suka – harap), 1-3 (suka – netral), dan 1-4 (suka – toleransi) sehingga dikategorikan sebagai atribut *attractive*. Pada atribut X5, responden tersebut memberikan jawaban skala 3-4 (netral – toleransi) sehingga dikategorikan sebagai atribut *indifferent*.

#### 4.4.2 Evaluasi Model Kano

Setelah melakukan klasifikasi setiap atribut fungsional dan disfungsional, selanjutnya melakukan evaluasi model kano dengan merekapitulasi perhitungan jumlah masing-masing kategori kano dari penilaian yang diberikan setiap responden pada masing-masing atribut. Berikut merupakan hasil rekapitulasi jumlah kategori kano masing-masing atribut dapat dilihat pada Tabel 4. 37.

Tabel 4. 37 Evaluasi Model Kano

Atribut	Kategori Kano						Total
	A	M	O	R	Q	I	
X1	5	11	6	2	0	26	50

Atribut	Kategori Kano						Total
	A	M	O	R	Q	I	
X2	2	15	8	2	0	23	50
X3	4	12	13	1	0	20	50
X4	4	12	19	0	0	15	50
X5	2	10	6	3	0	29	50
X6	2	16	11	0	0	21	50
X7	4	10	9	2	0	25	50
X8	4	3	21	1	0	21	50
X9	7	7	25	0	0	11	50
X10	2	3	17	1	1	26	50
X11	4	7	31	0	0	8	50
X12	7	3	28	0	0	12	50
X13	6	6	29	0	0	9	50
X14	5	7	28	0	0	10	50
X15	7	12	22	0	1	8	50

Berdasarkan hasil evaluasi model kano pada Tabel 4. 36, sebagai contoh pada atribut 1 terdapat 5 responden yang memilih atribut tersebut sebagai kategori *attractive*, 11 responden yang memilih atribut tersebut sebagai kategori *must-be*, 6 responden yang memilih atribut tersebut sebagai kategori *one-dimensional*, 2 responden yang memilih atribut tersebut sebagai kategori *reverse*, 0 responden yang memilih atribut tersebut sebagai kategori *questionable*, 26 responden yang memilih atribut tersebut sebagai kategori *indifferent* dari total keseluruhan 50 responden.

#### 4.4.3 Analisis Hasil Evaluasi Model Kano

Hasil evaluasi model kano selanjutnya dianalisis dengan menggunakan metode *better-worse*. Pemilihan metode ini dipilih karena hasil analisisnya menghasilkan data yang ringkas tanpa menghilangkan ketepatannya, serta hasil yang diperoleh lebih akurat (Hartono et al., 2018). Analisis hasil evaluasi model kano dengan metode *better-worse* yaitu dengan menghitung nilai *better* dan *worse* menggunakan Rumus 2.6 dan 2.7. Nilai *better* mengindikasikan seberapa besar kenaikan kepuasan konsumen jika atribut disediakan (*Attractive* dan *One Dimensional*). Sedangkan nilai *worse* mengindikasikan

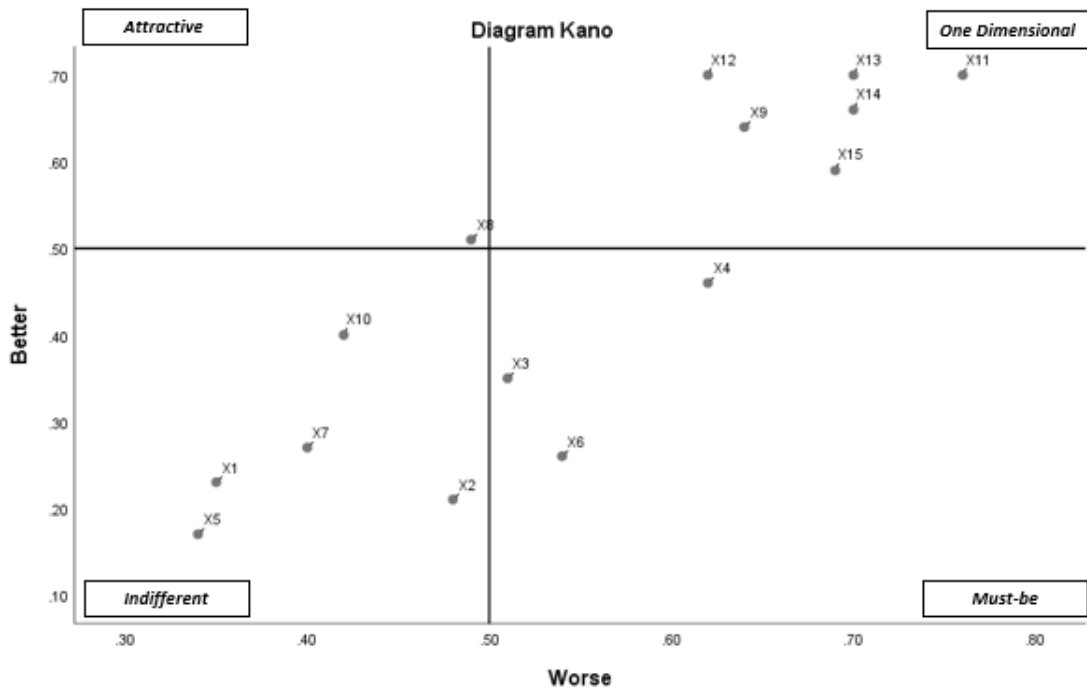
seberapa besar penurunan kepuasan konsumen jika atributnya tidak disediakan (*One Dimensional dan Must-be*) (Nofirza & Indrayani, 2011). Koefisien tingkat kepuasan nilai *better* berkisar antara 0 sampai 1, apabila nilai *better* semakin mendekati 1 maka semakin tinggi kepuasan pengguna. Koefisien penurunan kepuasan nilai *worse* berkisar antara 0 sampai -1, apabila nilai *worse* semakin mendekati -1 maka tingkat kekecewaan pengguna semakin kuat (Ramadhani & Suciandani, 2011). Berdasarkan rumus, tanda minus yang terdapat pada perhitungan *worse* untuk menegaskan pengaruh negatif dari kepuasan pengguna apabila kualitas jasa tidak dipenuhi. Berikut merupakan hasil perhitungan *better* dan *worse* pada masing-masing atribut yang ditunjukkan pada Tabel 4.38.

Tabel 4. 38 Hasil Perhitungan Nilai *Better* dan *Worse*

Atribut	Kategori Kano						Total	<i>Better</i>	<i>Worse</i>
	A	M	O	R	Q	I			
X1	5	11	6	2	0	26	50	0,23	-0,35
X2	2	15	8	2	0	23	50	0,21	-0,48
X3	4	12	13	1	0	20	50	0,35	-0,51
X4	4	12	19	0	0	15	50	0,46	-0,62
X5	2	10	6	3	0	29	50	0,17	-0,34
X6	2	16	11	0	0	21	50	0,26	-0,54
X7	4	10	9	2	0	25	50	0,27	-0,40
X8	4	3	21	1	0	21	50	0,51	-0,49
X9	7	7	25	0	0	11	50	0,64	-0,64
X10	2	3	17	1	1	26	50	0,40	-0,42
X11	4	7	31	0	0	8	50	0,70	-0,76
X12	7	3	28	0	0	12	50	0,70	-0,62
X13	6	6	29	0	0	9	50	0,70	-0,70
X14	5	7	28	0	0	10	50	0,66	-0,70
X15	7	12	22	0	1	8	50	0,59	-0,69

Setelah memperoleh hasil perhitungan nilai *better* dan *worse* pada Tabel 4.38, selanjutnya dilakukan pemetaan diagram kano berdasarkan nilai *better* dan *worse* menggunakan bantuan *software SPSS* versi 26.0. Nilai *better* digambarkan sebagai sumbu

Y dan nilai *worse* digambarkan sebagai sumbu X. Berikut merupakan diagram yang menentukan 4 kategori kano pada atribut layanan aplikasi Alfragift yang ditunjukkan pada Gambar 4. 8 dibawah ini.



Gambar 4. 9 Diagram Kano Atribut Layanan Aplikasi Alfragift

Berdasarkan hasil pemetaan diagram kano pada Gambar 4. 9, semua atribut layanan masuk ke dalam keempat kategori kano yaitu *attractive*, *one dimensional*, *must-be*, dan *indifferent*. Berikut hasil kategori kano setiap atribut yang ditunjukkan pada Tabel 4. 39.

Tabel 4. 39 Kategori Kano pada Setiap Atribut dengan Metode *Better-Worse*

Atribut	Kategori
X1	I
X2	I
X3	M
X4	M
X5	I
X6	M

Atribut	Kategori
X7	I
X8	A
X9	O
X10	I
X11	O
X12	O
X13	O
X14	O
X15	O

Berdasarkan hasil kategori kano menggunakan metode *better-worse* pada Tabel 4. 40, terdapat 3 atribut dengan kategori *must-be* yaitu atribut X3, X4, dan X6. Pada kategori *one dimensional* terdapat 6 atribut 6 atribut X9, X11, X12, X13, X14, dan X15. Terdapat 1 atribut dengan kategori *attractive* yaitu atribut X8 dan terdapat 5 atribut dengan kategori *indifferent* yaitu X1, X2, X5, X7, dan X10.

Setelah semua atribut dikategorikan, selanjutnya menentukan tindakan yang diambil untuk peningkatan. Menurut MacDonald et al (2006), berikut merupakan tindakan yang diambil untuk peningkatan atribut pada setiap kategori kano yang ditunjukkan pada Tabel 4. 40.

Tabel 4. 40 Deskripsi Kategori Kano dan Tindakan Peningkatan

Kategori	Deskripsi Pada Pengguna	Tindakan yang diambil untuk peningkatan
<i>Attractive</i>	Jika atributnya " <i>Attractive</i> ", berarti Atribut tersebut memberikan kepuasan ekstra bagi pelanggan ketika disediakan, melainkan pelayanan masih melakukan tugasnya	Berusaha untuk menyediakan atribut pada tingkat fungsi dasar, karena kehadiran fitur saja akan menimbulkan kepuasan.

---

	dengan sempurna ketika atributnya tidak disediakan.	
<i>One-dimensional</i>	Jika atributnya “ <i>One-dimensional</i> ”, artinya semakin banyak perhatian perusahaan atau organisasi terhadap atribut ini dalam desain produk/jasa, semakin puas pelanggan, terhadap produk/jasa tersebut, dan semakin baik	Berusaha untuk meningkatkan fungsionalitas atribut melebihi standar industri
<i>Must-be</i>	Jika atributnya “ <i>Must-be</i> ”, artinya tidak adanya atribut ini pasti akan menyebabkan ketidakpuasan, dan mungkin membuat layanan tidak berguna bagi pelanggan. Namun, tidak seperti kategori <i>One dimensional</i> , upaya perbaikan ekstra yang dilakukan untuk meningkatkan kategori <i>Must-be</i> tidak akan memberikan banyak perbedaan bagi pelanggan – kategori tersebut hanya perlu disertakan dan berfungsi secara normal.	Memastikan atribut disediakan dan berfungsi dengan baik
<i>Indifferent</i>	Jika atributnya “ <i>Indifferent</i> ”, artinya atribut tidak menimbulkan kepuasan dan ketidakpuasan pelanggan	Tidak berfokus pada atribut dengan kategori ini.
<i>Reverse</i>	Jika atributnya “Terbalik”, berarti atribut tersebut menyebabkan ketidakpuasan pelanggan dan mungkin	Memastikan tidak menyediakan atribut ini

---



membuat layanan kurang  
berguna bagi pelanggan

---

Tindakan perbaikan, secara umum panduan bagi organisasi dalam menentukan target untuk kategori kano adalah berupaya memenuhi semua atribut *must-be*, mempunyai kinerja lebih baik daripada kompetitor pada atribut *one dimensional*, dan memasukkan atribut *attractive* yang beda dengan kompetitor (Ramadhani & Suciandani, 2011). Berdasarkan tindakan yang diambil untuk peningkatan atribut menurut (MacDonald et al., 2006) pada Tabel 4. 40, Berikut merupakan konsep perbaikan atribut pada masing-masing kategori kano yang disajikan pada Tabel 4. 41. Adapun urutan prioritas kategori kano yang dilakukan perbaikan juga disajikan pada Tabel 4. 41 berdasarkan urutan prioritas kategori kano untuk dilakukan penanganan menurut (Ramadhan, Purnamasari, & Setyono, 2020) yaitu, *must-be* > *one dimensional* > *attractive* > *indifferent*.

Tabel 4. 41 Konsep Perbaikan dan Urutan Prioritas Kategori Kano pada Aplikasi Alfagift

<b>Prioritas</b>	<b>No</b>	<b>Atribut</b>	<b>Konsep Perbaikan</b>
Prioritas 1 ( <i>must-be</i> )	X3	Aplikasi mudah digunakan	Memenuhi semua atribut
	X4	Belanja menggunakan aplikasi menguntungkan	
	X6	Layanan aplikasi memuaskan	
Prioritas 2 ( <i>one dimensional</i> )	X9	Poin masuk ke dalam aplikasi	Peningkatan fungsional atribut
	X11	Pesanan dikirim ke lokasi pembeli	
	X12	Struk masuk (tersimpan) ke dalam aplikasi	
	X13	Barang yang datang sesuai (benar) pesanan	
	X14	Barang yang datang jumlahnya cukup	

<b>Prioritas</b>	<b>No</b>	<b>Atribut</b>	<b>Konsep Perbaikan</b>
	X15	Kondisi barang yang datang memuaskan	
Prioritas 3 ( <i>attractive</i> )	X8	Belanja menggunakan aplikasi bisa COD	Berusaha untuk menyediakan atribut pada tingkat fungsi dasar, karena kehadiran fitur saja akan menimbulkan kepuasan
Prioritas 4 ( <i>indifferent</i> )	X1	Aplikasi Alfagift cepat dibuka	Tidak berfokus pada atribut dengan kategori ini atau dengan kata lain tidak dilakukan perbaikan
	X2	Aplikasi dirancang dengan bagus	
	X5	Stok barang di aplikasi lengkap	
	X7	Aplikasi jarang terjadi eror	
	X10	Pengantaran barang / pesanan cepat	

Prioritas pertama kategori kano yang dilakukan perbaikan yaitu atribut dengan kategori *must-be* dengan perbaikan memenuhi semua atribut. Hal ini dikarenakan tidak adanya atribut ini pasti akan menyebabkan ketidakpuasan. Kemudian prioritas kedua kategori kano yang dilakukan perbaikan yaitu atribut dengan kategori *one dimensional* dengan perbaikan meningkatkan fungsionalitas atribut melebihi standar yang ada. Hal ini dikarenakan semakin banyak perhatian perusahaan atau organisasi terhadap atribut ini, maka semakin meningkatkan kepuasan pelanggan dan semakin baik. Prioritas ketiga kategori kano untuk dilakukan peningkatan kepuasan yaitu atribut dengan kategori *attractive* dengan berusaha untuk menyediakan atribut pada tingkat fungsi dasar, karena kehadiran fitur saja akan menimbulkan kepuasan. Sementara prioritas terakhir yaitu

kategori kano *indifferent* tidak dilakukan perbaikan dikarenakan atribut pada kategori ini tidak menimbulkan kepuasan dan ketidakpuasan pelanggan.

## BAB V

### ANALISIS DAN PEMBAHASAN

#### 5.1 Pemilihan *Domain*

Alfagift merupakan aplikasi berbelanja *online* sehingga masuk ke dalam jenis aplikasi *e-commerce*. Alfagift dapat dijalankan pada *platform web* maupun *mobile*. Aplikasi *mobile* merupakan teknologi yang paling banyak digunakan, hal ini didukung dengan perkembangan *smartphone* yang semakin canggih (Yusril et al., 2021). Sehingga pada penelitian ini, domain yang dipilih yaitu aplikasi *e-commerce* berbasis *mobile*.

#### 5.2 Perentangan Ruang Semantik

Pada tahapan ini bertujuan untuk mengidentifikasi *kansei words* dengan metode *text mining* yang diklasifikasikan menggunakan algoritma *machine learning*. Berdasarkan hasil perbandingan klasifikasi algoritma *Machine Learning* pada keempat algoritma dengan empat perbandingan data latih dan data uji yang ditunjukkan pada Gambar 4. 5, algoritma *Support Vector Machine* (SVM) kernel *linear* dengan perbandingan data latih 90% dan data uji 10% memperoleh hasil performansi tertinggi dengan hasil akurasi 92,23%. Semakin tinggi nilai akurasi, maka semakin baik model yang dihasilkan (Harahap, 2016). Sehingga dapat disimpulkan bahwa algoritma *Support Vector Machine* (SVM) kernel *linear* merupakan model yang paling baik dalam mengklasifikasikan data ulasan pada penelitian ini. Perolehan performansi tertinggi pada perbandingan data 90:10 tersebut dapat disebabkan oleh perbedaan jumlah data latih dan data uji yang berpengaruh pada akurasi algoritma. Semakin besar selisih persentase data latih dan data uji maka perolehan akurasi semakin tinggi. Hal ini dikarenakan jika semakin banyak data latih yang digunakan, maka model *classifier* yang dibentuk semakin baik dan semakin lengkap berdasarkan fakta dari data latih tersebut. Sehingga prediksi pada klasifikasi data uji atau data baru perolehan akurasi akan semakin baik atau tinggi (Rahman et al., 2018).

Algoritma SVM memiliki performansi tertinggi dikarenakan kelebihanannya mampu bekerja pada dataset yang berdimensi tinggi seperti data besar yang memiliki variabel lebih besar daripada banyaknya observasi (Setiawan & Mulyati, 2023). Algoritma SVM mempunyai keunggulan dapat mengolah data berdimensi tinggi, tanpa mengalami penurunan performa (Setiawan & Mulyati, 2023). Kemudian dapat meminimalkan *error* dalam data *training* (Ulfah & Anam, 2020) dan akurasi yang diperoleh tinggi bahkan dengan kumpulan data besar (Bhavitha et al., 2017). Sedangkan Algoritma *Naïve Bayes Classifier* (NBC) memiliki performansi terendah dalam klasifikasi data pada penelitian ini dibandingkan algoritma *machine learning* lainnya. Hal ini dikarenakan Algoritma *Naïve Bayes Classifier* menghasilkan performansi klasifikasi yang relatif rendah pada kumpulan data besar (Bhavitha et al., 2017). Selain itu terdapat kelemahan dari Algoritma *Naïve Bayes* dikenal sebagai prediktor yang buruk karena hasil prediksi adalah asumsi (Babu et al., 2019).

Berdasarkan hasil *confusion matrix* dari algoritma terpilih yang ditunjukkan pada Tabel 4. 21, menunjukkan kinerja model yang baik. *Confusion matrix* menggambarkan nilai akurasi prediksi positif benar, prediksi positif salah, prediksi negatif benar dan prediksi negatif salah. Algoritma ini mampu memprediksi data ulasan positif dengan benar (*true positive*) sebanyak 1.090 data dan memprediksi data ulasan negatif dengan benar (*true negative*) sebanyak 299 data. Sedangkan jumlah data ulasan positif yang diprediksi dengan salah yaitu sebanyak 46 data ulasan dan data ulasan negatif yang diprediksi dengan salah yaitu sebanyak 71 data. Terdapat 4 parameter dalam mengukur performansi model yaitu *accuracy*, *precision*, *recall*, dan *f1 score*. Berdasarkan hasil, pada kelas negatif dan positif memperoleh nilai *precision*, *recall*, dan *f1 score* yang tinggi. Semakin tinggi nilai dari *accuracy*, *precision*, dan *recall* maka semakin baik klasifikasi yang dilakukan oleh algoritma (Felix et al., 2019).

Setelah data diklasifikasi, pada tahapan *text mining* kemudian dilakukan proses asosiasi kata untuk mengetahui kata yang sering muncul secara bersamaan. Pada tahapan ini terdapat 15 *kansei words* pada Tabel 4. 26 yang didapatkan dari hasil asosiasi kata negatif yaitu “lambat”, “bagus”, “mudah”, “rugi”, “kosong”, “kecewa”, “eror”, “tidak bisa”, “masuk”, “lama”, “tidak dikirim”, “tidak masuk”, “salah”, “kurang”, “kecewa”. Pada tahapan ini masih terdapat kekurangan yaitu pada perhitungan skor sentimen pada *data training* menggunakan kamus *lexicon*. Kamus *lexicon* berisi kumpulan dari kata beserta nilai sentimennya memiliki keterbatasan dalam perhitungan skor sentimen yaitu

hanya menghitung skor sentimen kata yang terdapat pada kamus saja, sehingga jika terdapat kata-kata pada data ulasan yang tidak terdapat pada kamus *lexicon*, maka kata tersebut tidak memiliki skor.

### 5.3 Perentangan Ruang Properti

Pada tahapan ini dilakukan identifikasi atribut yang dihubungkan dengan *kansei words* yang diperoleh pada proses perentangan ruang semantik sebelumnya. Atribut ditentukan berdasarkan pasangan *kansei word* pada asosiasi kata negatif pada proses *text mining*. *Kansei word* yang memiliki makna negatif terlebih dahulu diubah menjadi makna negatif. Kemudian pasangan *kansei word* tersebut dikembangkan menjadi pernyataan atribut yang mewakili *kansei word*, sehingga diperoleh 15 atribut yang ditunjukkan pada Tabel 4. 31.

### 5.4 Sintesis

Setelah atribut didapatkan, dilakukan penyebaran kuesioner *Kano* pada 50 responden yang pernah menggunakan dan berbelanja di Aplikasi Alfagift. Berdasarkan hasil klasifikasi kategori *kano* pada Gambar 4. 8 dan Tabel 4. 39 hanya 4 kategori *kano* yang terdapat pada atribut-atribut yaitu *must-be*, *one dimensional*, *attractive* dan *indifferent*. Tidak ada atribut dengan kategori *reverse* dan *questionable* dikarenakan berdasarkan rekapitulasi jumlah kategori *kano* setiap atribut, hanya sedikit responden yang memberikan nilai atribut pada kedua kategori *kano* tersebut. Berikut merupakan atribut pada masing-masing kategori *kano*.

#### a) *Must-be*

Atribut dengan kategori *must-be* atau disebut juga *basic need* dianggap pelanggan sebagai suatu keharusan yang harus tersedia dalam suatu produk atau jasa. Adanya atribut *must-be* tidak akan meningkatkan kepuasan pelanggan, namun apabila kategori ini tidak terpenuhi akan menurunkan tingkat kepuasan pelanggan (Ramadhan et al., 2020). Terdapat 3 atribut pada kategori *must-be* yaitu layanan aplikasi memuaskan yaitu atribut X3, X4, dan X6.

#### b) *One Dimensional*

Atribut dengan kategori *one dimensional* merupakan atribut yang tergolong penting untuk dipenuhi. Hal ini dikarenakan pada kategori ini kepuasan konsumen berbanding lurus dengan performa kriteria produk atau layanan. Kepuasan konsumen akan meningkat apabila atribut pada kategori ini disediakan, tetapi konsumen akan menurun tingkat

kepuasannya bahkan merasa kecewa apabila atribut tidak disediakan. Terdapat 6 atribut dengan kategori *one dimensional* yaitu atribut X9, X11, X12, X13, X14, dan X15.

c) *Attractive*

Atribut dengan kategori *attractive* merupakan atribut yang jika tidak disediakan tidak akan menurunkan kepuasan pelanggan. Namun, jika atribut ini disediakan maka akan memberikan kepuasan ekstra bagi pelanggan. Terdapat 1 atribut dengan kategori *attractive* yaitu atribut X8

d) *Indifferent*

Atribut dengan kategori *indifferent* merupakan atribut yang berada pada kategori yang kurang diperhatikan oleh pengguna. Tersedia ataupun tidak disediakan atribut tidak akan mempengaruhi tingkat kepuasan konsumen. Atau dengan kata lain kategori *indifferent* mempunyai pengaruh kecil pada kepuasan dan ketidakpuasan pengguna (Ramadhan et al., 2020). Terdapat 5 atribut yang masuk dalam kategori *indifferent* yaitu atribut X1, X2, X5, X7, dan X10.

Setelah atribut dikategorisasi, selanjutnya ditentukan konsep perbaikan yang mengacu pada Tabel 4. 41. Prioritas pertama kategori kano yang dilakukan perbaikan yaitu atribut dengan kategori *must-be* dengan perbaikan memenuhi semua atribut X3, X4, dan X6. Hal ini dikarenakan tidak adanya atribut ini pasti akan menyebabkan ketidakpuasan. Kemudian prioritas kedua kategori kano yang dilakukan perbaikan yaitu atribut dengan kategori *one dimensional* yaitu atribut X9, X11, X12, X13, X14, dan X15 dengan perbaikan meningkatkan fungsionalitas atribut melebihi standar yang ada. Hal ini dikarenakan semakin banyak perhatian perusahaan atau organisasi terhadap atribut ini, maka semakin meningkatkan kepuasan pelanggan dan semakin baik. Prioritas ketiga kategori kano untuk dilakukan peningkatan kepuasan yaitu atribut X8 dengan kategori *attractive* dengan berusaha untuk menyediakan atribut pada tingkat fungsi dasar, karena kehadiran fitur saja akan menimbulkan kepuasan. Sementara prioritas terakhir yaitu atribut X1, X2, X5, X7, dan X10 dengan kategori kano *indifferent* tidak dilakukan perbaikan dikarenakan atribut pada kategori ini tidak menimbulkan kepuasan dan ketidakpuasan pelanggan.

## 5.5 Rekomendasi Perbaikan

Berdasarkan hasil kategori kano, atribut pada penelitian ini terdapat 4 kategori yaitu *must-be*, *one dimensional*, *attractive* dan *indifferent*. Rekomendasi perbaikan aplikasi *Alfagift*

diberikan hanya pada atribut dengan kategori *must-be*, *one dimensional*, dan *attractive* dikarenakan atribut dengan kategori *indifferent* tidak berdampak pada kepuasan pelanggan apabila atribut disediakan ataupun tidak disediakan.

#### 5.5.1 Rekomendasi Atribut dengan Kategori Must-be

Pada atribut X3 yaitu “Aplikasi mudah digunakan” berdasarkan klasifikasi model kano masuk dalam kategori *must-be*. Pada Gambar 5. 1 dan Gambar 5. 2, diketahui terdapat keluhan konsumen sulit menggunakan aplikasi dikarenakan terjadinya error pada aplikasi. Selain itu proses registrasi yang dinilai pengguna cukup sulit. Rekomendasi perbaikan yang dapat diberikan yaitu melakukan evaluasi dan perbaikan pada sistem aplikasi oleh pihak *developer* agar meminimalisir terjadinya error sehingga pengguna dapat menggunakan aplikasi dengan mudah. Kemudian perlu peninjauan kembali pada proses registrasi akun dengan dibuat lebih ringkas agar pengguna tidak mengalami kesulitan registrasi akun.



Gambar 5. 1 Ulasan Pengguna Alfagift pada Atribut X3

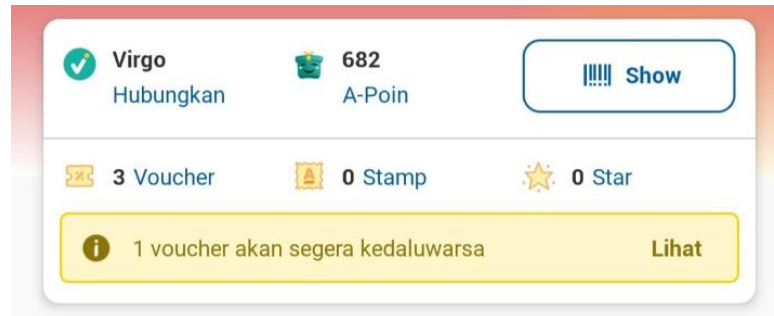


Gambar 5. 2 Ulasan Pengguna Alfagift pada Atribut X3

Pada atribut X4 yaitu “Belanja menggunakan aplikasi menguntungkan” berdasarkan klasifikasi model kano masuk dalam kategori *must-be*. aplikasi *Alfagift* menawarkan keuntungan yang bisa didapatkan bagi penggunanya seperti *A-Poin*, *A-Voucher*, *Alfastamp*, dan *Alfastar* yang dapat dilihat pada Gambar 5. 3. Tetapi masih terdapat keluhan pelanggan terkait penawaran-penawaran tersebut seperti *stamp* lama masuk,



*voucher* yang tidak bisa digunakan, dan poin yang tidak masuk yang ditunjukkan pada Gambar 5. 4. Adapun rekomendasi perbaikan yang dapat diberikan yaitu perbaikan sistem aplikasi oleh pihak *developer* untuk mengatasi permasalahan poin, *voucher*, *star* dan *stamp* yang tidak masuk dan tidak bisa digunakan.



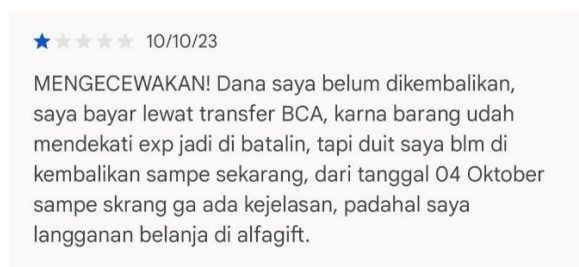
Gambar 5. 3 Fitur *A-Poin*, *A-Voucher*, *Alfastamp*, dan *Alfastar* pada Aplikasi Alfagift



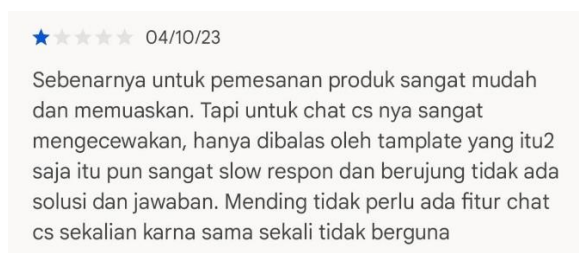
Gambar 5. 4 Ulasan Pengguna Alfagift pada Atribut X4

Pada atribut X6 yaitu “layanan aplikasi memuaskan” berdasarkan klasifikasi model kano masuk dalam kategori *must-be*. Beberapa pengguna *Alfagift* masih merasakan ketidakpuasan layanan aplikasi seperti proses *refund* yang lama dikembalikan dan juga

*customer service* yang tidak solutif dalam menangani keluhan konsumen yang dapat dilihat pada Gambar 5. 5 dan 5. 6. Pada permasalahan ini, rekomendasi yang dapat diberikan yaitu melakukan evaluasi kinerja pada pihak *customer service* dan memeriksa alur kerja dari *customer service* untuk mengidentifikasi alur yang memerlukan perbaikan untuk mempercepat respons.



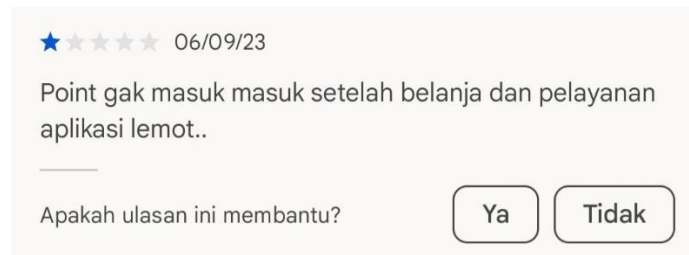
Gambar 5. 5 Ulasan Pengguna Alfagift pada Atribut X6



Gambar 5. 6 Ulasan Pengguna Alfagift pada Atribut X6

### 5.5.2 Rekomendasi Atribut dengan Kategori One Dimensional

Pada atribut X9 yaitu “Poin masuk ke dalam aplikasi” berdasarkan klasifikasi model kano masuk dalam kategori *one dimensional*. A-poin akan didapatkan oleh pelanggan setelah belanja di *Alfamart* melalui aplikasi *Alfagift*. Poin ini dapat menjadi *cashback* atau ditukarkan dengan *voucher* potongan harga/ tebus produk gratis. Namun masih terdapat keluhan terkait poin yang tidak masuk setelah belanja seperti pada Gambar 5. 7. Sehingga rekomendasi perbaikan yang diberikan yaitu perbaikan sistem aplikasi oleh pihak *developer* untuk mengatasi permasalahan poin yang tidak masuk ke pelanggan.

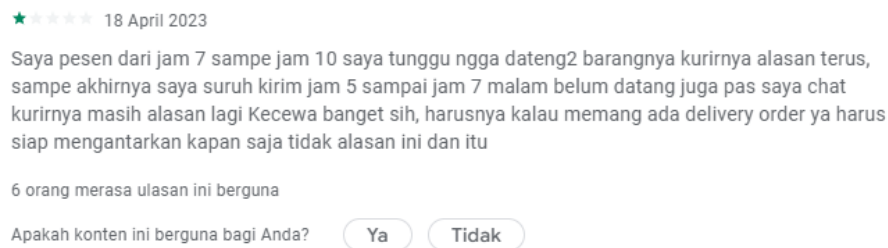


Gambar 5. 7 Ulasan Pengguna Alfagift pada Atribut X9

Pada atribut X11 yaitu “Pesanan dikirim ke lokasi pembeli” berdasarkan klasifikasi model kano masuk dalam kategori *one dimensional*. Pada aplikasi *Alfagift* sudah tersedia fitur untuk pengiriman pesanan ke lokasi pembeli pada Gambar 5. 8. Namun masih terdapat keluhan pelanggan terkait pengiriman pesanan seperti yang terlihat pada Gambar 5. 9. Sehingga dalam mengatasi keluhan konsumen terkait pengiriman, direkomendasikan agar menambah karyawan yang dikhususkan bertugas untuk melayani pelanggan yang berbelanja melalui aplikasi. Kemudian perlu dilakukan pengecekan pesanan secara berkala oleh karyawan agar pesanan konsumen dapat segera diproses hingga pesanan terkirim sampai ke lokasi pelanggan.



Gambar 5. 8 Tampilan Opsi Pengambilan Pesanan pada Aplikasi Alfagift



Gambar 5. 9 Ulasan Pengguna Alfacift pada Atribut X11

Pada atribut X12 yaitu “Struk masuk (tersimpan) ke dalam aplikasi” berdasarkan klasifikasi model kano masuk dalam kategori *one dimensional*. Pada aplikasi *Alfacift*, setelah proses pemesanan barang selesai maka akan diberikan struk yang dapat dilihat pada aplikasi seperti pada Gambar 5. 10. Tetapi terdapat keluhan dari pengguna mengenai struk dapat dilihat pada Gambar 5. 11. Sehingga berdasarkan permasalahan tersebut, direkomendasikan agar melakukan perbaikan sistem aplikasi oleh pihak *developer* untuk meminimalisir terjadinya struk yang tidak terkirim ke pelanggan.

**Alfamart**

Delivered at : 

Time : Friday, 06 October 2023  
07:00 - 21:00

Status Order : **Pengiriman Selesai**

---

Ref :

Indomie Mi Instan Soto Spesial 75 g	1	3,100	3,100
MILKU Susu Stroberi 200 ml	1	3,300	3,300
INDOMILK Seoul Susu UHT Banana 180 ml	1	5,400	5,400
Delfi Big Thunder Cookie Bar Chocolate 36 g	1	6,300	6,300
Nestle Pure Life Air Mineral 1500 ml	1	6,700	6,700
Sasa Tepung Bumbu Bakwan Spesial 225 g	1	7,400	7,400
Delfi Take-it Big Wafer Cokelat 2 x 33 g	1	9,700	9,700
Aice Histeria Ice Cream Vanila 70 ml	1	9,900	9,900
<b>Subtotal</b>	<b>8</b>		<b>51,800</b>
<b>Total Diskon</b>			<b>(2,000)</b>
<b>Biaya Pengiriman</b>			<b>0</b>
<b>Total</b>			<b>49,800</b>

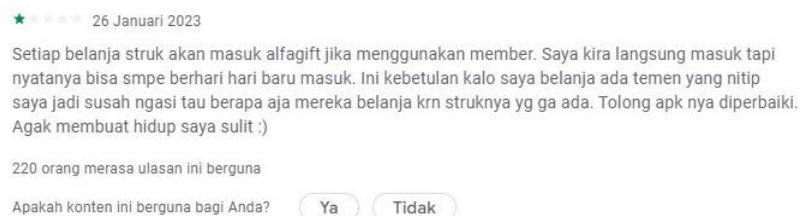
\*Harga yang tertera sudah termasuk PPN

L U N A S

Tgl. 06-10-2023 20:00:11

Kritik & Saran : 1500959, SMS : 0817111234

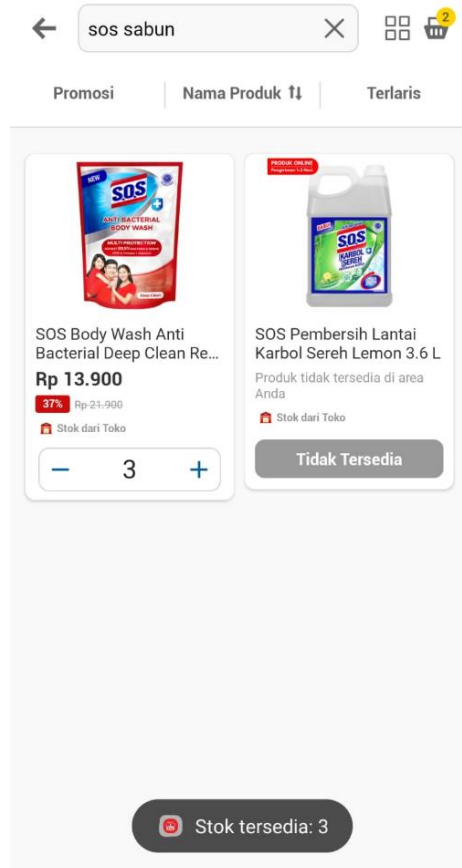
Gambar 5. 10 Tampilan Struk Elektronik pada Aplikasi Alfagift



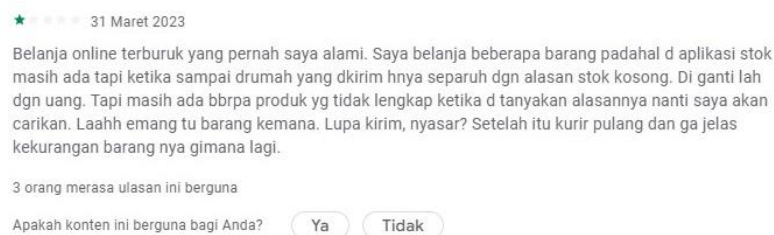
Gambar 5. 11 Ulasan Pengguna Alfagift pada Atribut X12

Pada atribut X13 yaitu “Barang yang datang sesuai pesanan” berdasarkan klasifikasi model kano masuk dalam kategori *one dimensional*. Dalam upaya meminimalisir kesalahan pengiriman barang, perlu dilakukan pelatihan karyawan mengenai prosedur untuk menjaga akurasi pesanan. Rekomendasi yang dapat diberikan yaitu mewajibkan karyawan untuk memeriksa kembali dan verifikasi pesanan sebelum dilakukan pengemasan barang untuk memastikan bahwa barang yang akan dikemas sesuai dengan pesanan pelanggan.

Pada atribut X14 yaitu “Barang yang datang jumlahnya cukup” berdasarkan klasifikasi model kano masuk dalam kategori *one dimensional*. Pada aplikasi terdapat fitur untuk mengetahui stok barang ditunjukkan pada Gambar 5. 12. Namun, masih terdapat keluhan pelanggan terkait stok barang yang terlihat pada Gambar 5. 13. Sehingga sangat direkomendasikan untuk melakukan *update* stok barang pada setiap laman produk agar tidak ada *miss* mengenai stok barang yang habis (R. O. Putri et al., 2017). Selain itu karyawan diwajibkan untuk memeriksa kembali dan verifikasi pesanan sebelum melakukan pengemasan barang untuk memastikan bahwa barang yang akan dikemas jumlahnya sudah sesuai dengan jumlah yang dipesan oleh pelanggan.



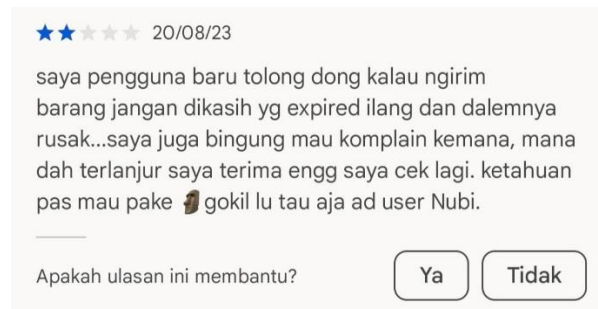
Gambar 5. 12 Tampilan Ketersediaan Stok Barang pada Aplikasi Alfagift



Gambar 5. 13 Ulasan Pengguna Alfagift pada Atribut X14

Pada atribut X15 yaitu “kondisi barang yang datang memuaskan” berdasarkan klasifikasi model kano masuk dalam kategori *one dimensional*. Pada atribut ini terdapat keluhan konsumen yang ditunjukkan pada Gambar 5. 14. dalam upaya agar kondisi barang yang diterima konsumen memuaskan yaitu dengan memerhatikan pengemasan barang agar barang yang dikirim terlindungi dengan baik dari hal-hal yang dapat merusak barang seperti benturan, tumpahan air, dll. Beberapa cara pengemasan barang agar aman yaitu membungkus barang dengan bungkus yang tidak mudah robek, tidak mudah rusak

atau pecah jika barangnya rapuh. Kemudian gunakan bungkus yang berfungsi sebagai bantalan (*padding*) untuk menahan getaran dan hentakan seperti *bubble wrap* (Barri et al., 2021). Selain itu melakukan pengecekan kondisi barang sebelum melakukan pengiriman seperti kondisi kemasan barang dan kadaluwarsa dari produk.




Gambar 5. 14 Ulasan Pengguna Alfagift pada Atribut X15

### 5.5.3 Rekomendasi Atribut dengan Kategori Attractive

Pada atribut X8 yaitu “Belanja menggunakan aplikasi bisa COD” berdasarkan klasifikasi model kano masuk dalam kategori *one dimensional*. Aplikasi *Alfagift* telah dilengkapi metode pembayaran *Cash on Delivery* (COD) yang ditunjukkan pada Gambar 5. 15. Namun, metode pembayaran *Cash on Delivery* (COD) baru bisa digunakan jika pelanggan melakukan pembelian minimal Rp 50.000,-. Sehingga rekomendasi yang dapat diberikan pada atribut ini yaitu pihak manajemen *Alfagift* melakukan peninjauan kembali minimal transaksi belanja yang ditetapkan untuk dapat menggunakan metode pembayaran COD. Karena dengan hadirnya metode pembayaran COD dengan minimal pembelian Rp 50.000,- saja akan menimbulkan kepuasan apalagi jika tidak memberlakukan minimal pembelian Rp 50.000 yang tentunya akan memberikan kepuasan ekstra bagi pelanggan.





The screenshot shows a payment selection interface. At the top left, there is a circular icon with a yellow border and a blue center containing the letters 'COD', followed by the text 'Cash On Delivery' in red. Below this is a red rounded rectangular button with the text 'Cash on Delivery' in white. Underneath the button, a line of text reads: 'Untuk kemudahan transaksi, silakan pilih uang tunai dibawah ini pada saat pembayaran pesanan anda:'. Below this text are five radio button options arranged in two columns. The first column contains three options: 'Rp. 8.000', 'Rp. 10.000', and 'Rp. 20.000'. The second column contains two options: 'Rp. 50.000' and 'Rp. 100.000'. At the bottom of the screen, there is a yellow rectangular box containing an information icon (an 'i' in a circle) followed by the text: 'Tingkatkan transaksi belanja minimal Rp 50.000 untuk menggunakan Cash on Delivery'.

Gambar 5. 15 Tampilan Metode Pembayaran *Cash on Delivery* pada Aplikasi Alfagift

## BAB VI

### PENUTUP

#### 6.1 Kesimpulan

Berdasarkan analisis dan pembahasan, maka kesimpulan yang diperoleh yaitu:

Pada penelitian ini ditentukan *domain* yaitu aplikasi *e-commerce* berbasis *mobile*. Pada proses perentangan ruang semantic menggunakan *text mining* yang diklasifikasikan menggunakan algoritma *machine learning* menunjukkan bahwa algoritma *Support Vector Machine* kernel *linear* dengan skenario perbandingan data latih 90% dan data uji 10% memiliki performansi tertinggi dibandingkan dengan ketiga algoritma *machine learning* lainnya. Berdasarkan hasil performansi model, algoritma ini memperoleh nilai *accuracy*, *precision*, *recall*, dan *f1 score* yang tinggi. Adapun nilai akurasi yang diperoleh yaitu sebesar 92,23%. Sehingga dapat disimpulkan algoritma *Support Vector Machine* kernel *linear* merupakan algoritma paling baik dalam mengklasifikasikan data ulasan pengguna aplikasi *Alfagift* pada penelitian ini. Kemudian berdasarkan hasil asosiasi kata dengan kelas negatif didapatkan 15 *kansei words* yaitu “lambat”, “bagus”, “mudah”, “rugi”, “kosong”, “kecewa”, “eror”, “tidak bisa”, “masuk”, “lama”, “tidak kirim”, “tidak masuk”, “salah”, “kurang”, dan “kecewa”.

Tahapan perentangan ruang properti, atribut diperoleh dari *kansei words*. Terdapat 15 atribut yang akan dilakukan klasifikasi kategori kano pada tahapan sintesis untuk mengetahui atribut yang memiliki pengaruh lebih terhadap kepuasan pengguna yang akan diprioritaskan untuk dilakukan peningkatan. Hasil klasifikasi kategori kano pada atribut-atribut menunjukkan terdapat 10 atribut yang diprioritaskan untuk dilakukan peningkatan kualitas sistem, informasi, dan layanan karena berdampak besar pada kepuasan pengguna. Rekomendasi yang diberikan untuk peningkatan kualitas sistem, informasi, dan layanan pada aplikasi *Alfagift* yaitu evaluasi kinerja *customer service*, evaluasi dan perbaikan sistem aplikasi oleh pihak *developer*, peninjauan kembali minimal transaksi belanja untuk metode pembayaran COD, pengecekan pesanan secara berkala oleh karyawan, mewajibkan karyawan untuk memeriksa kembali dan verifikasi pesanan sebelum

dilakukan pengemasan barang, membungkus barang dengan bungkus yang tidak mudah robek, tidak mudah rusak atau pecah jika barangnya rapuh, dan pengecekan kondisi barang sebelum melakukan pengiriman.

## 6.2 Saran

Berikut merupakan saran yang dapat diberikan berdasarkan hasil penelitian ini:

### 1. Bagi Pihak Aplikasi *Alfagift*

Hasil ekstraksi ulasan pengguna aplikasi *Alfagift* dapat dijadikan bahan evaluasi untuk melakukan upaya peningkatan kualitas sistem, informasi, dan layanan aplikasi. Kemudian dalam melakukan pengembangan aplikasi, pihak *Alfagift* dapat lebih memprioritaskan atribut yang memiliki pengaruh besar pada tingkat kepuasan pengguna aplikasi terutama pada atribut dengan kategori *must-be* dan *one dimensional*.

### 2. Bagi Penelitian Selanjutnya

1) Pada tahapan ini masih terdapat kekurangan yaitu pada perhitungan skor sentimen pada *data training* menggunakan kamus *lexicon*. Kamus *lexicon* berisi kumpulan dari kata beserta nilai sentimennya memiliki keterbatasan dalam perhitungan skor sentimen yaitu hanya menghitung skor sentimen kata yang terdapat pada kamus saja, sehingga jika terdapat kata-kata pada data ulasan yang tidak terdapat pada kamus *lexicon*, maka kata tersebut tidak memiliki skor. Oleh karena itu pada penelitian selanjutnya disarankan untuk menambahkan *tools* lainnya.

2) Pada penelitian selanjutnya, dapat menambahkan metode lain untuk upaya meningkatkan kepuasan pelanggan dikarenakan penelitian ini memiliki keterbatasan hanya memberikan rekomendasi perbaikan pada aplikasi tidak sampai melakukan pengembangan pada aplikasi.

## DAFTAR PUSTAKA

- Akbari, M. I. H. A. D., Novianty, A., & Setianingsih, C. (2017). *Analisis Sentimen Menggunakan Metode Learning Vector Quantization*.
- Altuntas, S., & Kansu, S. (2019). An innovative and integrated approach based on SERVQUAL, QFD and FMEA for service quality improvement. *Kybernetes*, 49(10), 2419–2453. <https://doi.org/10.1108/K-04-2019-0269>
- Andriani, M., Irawan, H., & Rizqa Asyura, N. (2021). Improving Quality Using The Kano Model in Overcoming Competition in The Service Industry. *International Journal of Engineering, Science and Information Technology*, 1(4), 13–18. <https://doi.org/10.52088/ijesty.v1i4.145>
- App Follow. (2023). *App Follow*.
- Arifiyanti, A. A., Pandji, M. F., & Utomo, B. (2022). Analisis Sentimen Ulasan Pengunjung Objek Wisata Gunung Bromo pada Situs Tripadvisor. *Explore: Jurnal Sistem Informasi Dan Telematika*, 13(1), 32. <https://doi.org/10.36448/jsit.v13i1.2539>
- Aryanti, R., Saepudin, A., Fitriani, E., Permana, R., & Saefudin, D. F. (2019). Komparasi Algoritma Naive Bayes Dengan Algoritma Genetika Pada Analisis Sentimen Pengguna Busway. *Jurnal Teknik Komputer*, 5(2), 227–234.
- Babu, D. R. K., Sree, C. U., Vidya, K. V., & Kumar, V. M. (2019). PREDICTING HEART DISEASE USING MACHINE LEARNING TECHNIQUES. *International Research Journal of Computer Science (IRJCS)*, 4(6).
- Badan Pusat Statistik. (2022). Kontribusi Perdagangan Besar dan Eceran; Reparasi Mobil dan Sepeda Motor Dalam PDB Triwulanan Tahun 2020-2022 (persen). In *Produk Domestik Bruto Indonesia Triwulanan 2018-2022* (p. 78). Badan Pusat Statistik.
- Bailey, D. K. (1999). *Methods of Social Research*. The free Press.
- Barri, M. H., Rizal, A., Cahyadi, W. A., Hidayat, I., Pramudita, B. A., & Prihatiningrum, N. (2021). PELATIHAN E-COMMERCE BAGI WARUNG SEDEKAH (WARKAH) UNTUK MENINGKATKAN PERTUMBUHAN EKONOMI RUMAH TANGGA. *KACANEGARA Jurnal Pengabdian Pada Masyarakat*, 4(1), 83. <https://doi.org/10.28989/kacanegara.v4i1.821>
- Barrios-Ipenza, F., Calvo-Mora, A., Criado-García, F., & Curioso, W. H. (2021). Quality Evaluation of Health Services Using the Kano Model in Two Hospitals in Peru. *International Journal of Environmental Research and Public Health*, 18(11), 6159. <https://doi.org/10.3390/ijerph18116159>
- Barutçu, S., & Elif, B. (2018). Expectations from personal mhealth apps through Kano's model. *Ournal of Internet Applications and Management*, 9(2), 35–48.

- Bhavitha, B. K., Rodrigues, A. P., & Chiplunkar, N. N. (2017). Comparative study of machine learning techniques in sentimental analysis. *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 216–221. <https://doi.org/10.1109/ICICCT.2017.7975191>
- Broucke, S. Vanden, & Baesens, B. (2018). *Practical Web Scraping for Data Science*. Apress. <https://doi.org/10.1007/978-1-4842-3582-9>
- Cai, M., Wu, M., Luo, X., Wang, Q., Zhang, Z., & Ji, Z. (2023). Integrated Framework of Kansei Engineering and Kano Model Applied to Service Design. *International Journal of Human–Computer Interaction*, 39(5), 1096–1110. <https://doi.org/10.1080/10447318.2022.2102301>
- Chen, M.-C., Chang, K.-C., Hsu, C.-L., & Xiao, J.-H. (2015). Applying a Kansei engineering-based logistics service design approach to developing international express services. *International Journal of Physical Distribution & Logistics Management*, 45(6), 618–646. <https://doi.org/10.1108/IJPDLM-10-2013-0251>
- Chrismanto, A. R., & Lukito, Y. (2016). Klasifikasi Sentimen Komentar Politik dari Facebook Page Menggunakan Naive Bayes. *Jurnal Informatika Dan Sistem Informasi*, 2–2.
- Darma, B. (2021). *Statistika Penelitian Menggunakan SPSS*. GUEPEDIA.
- DeLone, W. H., & Mclean, E. R. (1992). *Determinants of Success for Computer Usage in Small Business*. *MIS Quarterly*.
- Diamendia, T., & Setyowati, M. S. (2021). Analisis Kebijakan Compliance Risk Management Berbasis Machine Learning pada Direktorat Jenderal Pajak. *Indonesian Treasury Review*, 6(3), 289–298.
- Euromonitor International. (2022). *Euromonitor International website*.
- Faisal, D., Fathimahhayati, L. D., & Sitania, F. D. (2021). Penerapan Metode Kansei Engineering Sebagai Upaya Perancangan ulang Kemasan Takoyaki (Studi Kasus: Takoyakiku Samarinda). *TEKNO*, 18(1), 92–109.
- Fatra, A. H. D., Hayatin, N., & Aditya, C. S. K. (2020). Analisa Sentimen Tweet Berbahasa Indonesia dengan Menggunakan Metode Lexicon pada Topik Perpindahan Ibu Kota Indonesia. *REPOSITOR*, 2(7), 977–984.
- Feldman, R., & Sanger, J. (2007). *The Text Mining Handbook*. Cambridge University Press.
- Felix, Faisal, S., Butarbutar, T. F. M., & Sirait, P. (2019). Implementasi CNN dan SVM untuk Identifikasi Penyakit Tomat via Daun. *Jurnal SIFO Mikroskil*, 20(2).
- Gerson, R. F. (1993). *Mengukur Kepuasan Pelanggan*. PPM.
- Ghozali, I. (2013). *Aplikasi Analisis Multivariate dengan Program SPSS Edisi Kelima*. Badan Penerbit Universitas Diponegoro.
- Ghozali, I. (2015). *Metode Penelitian Kualitatif*. Ar-Ruzz.

- Ginting, S., & Hadiana, A. (2018). Interface and Service Analysis on Student Website Using Kansei Engineering and Kano. *IOP Conference Series: Materials Science and Engineering*, 407, 012170. <https://doi.org/10.1088/1757-899X/407/1/012170>
- Gunawan, K. I., & Santoso, J. (2021). Multilabel Text Classification Menggunakan SVM dan Doc2Vec Classification Pada Dokumen Berita Bahasa Indonesia. *Journal of Information System, Graphics, Hospitality and Technology*, 3(01), 29–38. <https://doi.org/10.37823/insight.v3i01.126>
- Hameed, A. Z., Kandasamy, J., Aravind Raj, S., Baghdadi, M. A., & Shahzad, M. A. (2022). Sustainable Product Development Using FMEA ECQFD TRIZ and Fuzzy TOPSIS. *Sustainability*, 14(21), 14345. <https://doi.org/10.3390/su142114345>
- Harahap, R. S. (2016). Komparasi Algoritma Klasifikasi Decision Tree, Naive Bayes dan Neural Network untuk Prediksi Penyakit Ginjal Kronis. *KNIT-2 Nusa Mandiri*.
- Harli, I. I., & Mutasowifin, A. (2021). Pengaruh Online Customer Review dan Rating terhadap Minat Beli Produk Kesehatan pada E-Marketplace Shopee Selama Masa Pandemi Covid-19. *INOBSIS*, 558–572.
- Hartono, M. (2020). The modified Kansei Engineering-based application for sustainable service design. *International Journal of Industrial Ergonomics*, 79, 102985. <https://doi.org/10.1016/j.ergon.2020.102985>
- Hartono, M., Santoso, A., Tanugraha, M. benita, Prayogo, D. N., & Kusumo, A. H. (2018). *Kansei Engineering, Kano, & TRIZ for Logistics Service Excellence*. Graha Ilmu.
- Hendrayana, I. G., Divayana, D. G. H., & Kesiman, M. W. A. (2023). KOMPARASI METODE SVM, K-NN DAN NBC PADA ANALISIS SENTIMEN. *Jurnal Indonesia: Manajemen Informatika Dan Komunikasi*, 4(1), 191–198. <https://doi.org/10.35870/jimik.v4i1.157>
- Hidayat, A. A. (2021). *Menyusun Instrumen Penelitian & Uji Validitas-Reliabilitas*. Health Books.
- Hiran, K. K., Jain, R. K., Lakhwani, K., & Doshi, R. (2021). *Machine Learning: Master Supervised and Unsupervised Learning Algorithms with Real Examples*. BPB Publications.
- Ho, H.-H., & Tzeng, S.-Y. (2021). Using the Kano model to analyze the user interface needs of middle-aged and older adults in mobile reading. *Computers in Human Behavior Reports*, 3, 100074. <https://doi.org/10.1016/j.chbr.2021.100074>
- Hsiao, Y.-H., Chen, M.-C., & Lin, M.-K. (2017). Kansei Engineering with Online Review Mining for Hotel Service Development. *2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, 29–34. <https://doi.org/10.1109/IIAI-AAI.2017.12>
- Indrawati, S., ‘Azzam, A., Adrianto, E., Miranda, S., & Prabaswari, A. D. (2020). Lean Concept Development in Fast Food Industry Using Integration of Six Sigma and

- TRIZ Method. *IOP Conference Series: Materials Science and Engineering*, 722(1), 012044. <https://doi.org/10.1088/1757-899X/722/1/012044>
- Iskandar, J. W., & Nataliani, Y. (2021). Perbandingan Naïve Bayes, SVM, dan k-NN untuk Analisis Sentimen Gadget Berbasis Aspek. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(6), 1120–1126. <https://doi.org/10.29207/resti.v5i6.3588>
- Jayanti, M., & Santoso, B. H. (2019). Analisis Pengaruh Likuiditas, Leverage, Profitabilitas, dan Rasio Pasar terhadap Harga Saham. *Jurnal Ilmu Dan Riset Manajemen*, 8(10), 1–18.
- Jo, T. (2019). *Text Mining* (Vol. 45). Springer International Publishing. <https://doi.org/10.1007/978-3-319-91815-0>
- Kamisa, N., P, A. D., & Novita, D. (2022). Pengaruh Online Customer Review dan Online Customer Rating terhadap Kepercayaan Konsumen. *Journal of Economic and Business Research*, 2(1), 21–29.
- Kano, N., Seraku, N., Takahashi, F., & Tsuji, S. (1984). Attractive quality and must-be quality. *The Journal of The Japanese Society for Quality Control*, 14(2), 39–48.
- Kemendikbud. (2016). *KBBI Daring*. <https://kbbi.kemdikbud.go.id/>.
- Kim, S.-O., Youn, S.-H., & Lee, M.-J. (2018). The Study on the e-Service Quality Factors in m-Shopping Mall App based on the Kano Model. *International Journal of Industrial Distribution & Business*, 9(12), 63–72.
- Kotler, P. (1997). *The Marketing of Nations, Strategic Approach to Building National Wealth*. The Free Press.
- Kuo, C.-C., Kung, H.-Y., Wu, H.-C., & Wang, M.-J. (2020). Developing a hand sizing system for a hand exoskeleton device based on the Kansei Engineering method. *Journal of Ambient Intelligence and Humanized Computing*. <https://doi.org/10.1007/s12652-020-02354-8>
- Kusuma Dewi, K., Kaniawulan, I., Dewi Lestari, C., Studi Teknik Informatika, P., & Tinggi Teknologi Wastukencana Purwakarta, S. (2023). *ANALISIS SENTIMEN PENGGUNA APLIKASI JAMSOSTEK MOBILE (JMO) PADA APP STORE MENGGUNAKAN METODE NAIVE BAYES*. 8(2).
- Laksono, R. A., Sungkono, K. R., Sarno, R., & Wahyuni, C. S. (2019). Sentiment Analysis of Restaurant Customer Reviews on TripAdvisor using Naïve Bayes. *2019 12th International Conference on Information & Communication Technology and System (ICTS)*, 49–54. <https://doi.org/10.1109/ICTS.2019.8850982>
- Liu, Z., Wu, J., Chen, Q., & Hu, T. (2023). An improved Kansei engineering method based on the mining of online product reviews. *Alexandria Engineering Journal*, 65, 797–808. <https://doi.org/10.1016/j.aej.2022.09.044>

- M. Kamal Fathoni. (2021). Transaksi Kerjasama Pembiayaan Produk Ritel Perspektif Hukum Ekonomi Syariah. *Istidlal: Jurnal Ekonomi Dan Hukum Islam*, 5(1), 47–61. <https://doi.org/10.35316/istidlal.v5i1.298>
- MacDonald, E., Backsell, M., Gonzalez, R., & Papalambros, P. (2006). The Kano Method's Imperfections, and Implications in Product Decision Theory. *International Design Research Symposium* .
- Mahawardana, P. P. O., Sasmita, G. A., & Pratama, I. P. A. E. (2022). Analisis Sentimen Berdasarkan Opini dari Media Sosial Twitter terhadap “Figure Pemimpin” Menggunakan Python. *JITTER*, 3(1).
- Malinka, C., von Jan, U., & Albrecht, U.-V. (2022). Prioritization of Quality Principles for Health Apps Using the Kano Model: Survey Study. *JMIR MHealth and UHealth*, 10(1), e26563. <https://doi.org/10.2196/26563>
- Maramis, F. S., Sepang, J. L., & Soegoto, A. S. (2018). Pengaruh Kualitas Produk, Harga, dan Kualitas Pelayanan terhadap Kepuasan Konsumen pada PT Air Manado. *EMBA*, 6(3).
- Matzler, K., Bailom, F., Hinterhuber, H. H., Renzl, B., & Pichler, J. (2004). The asymmetric relationship between attribute-level performance and overall customer satisfaction: a reconsideration of the importance–performance analysis. *Industrial Marketing Management*, 33(4), 271–277.
- Miller, T. W. (2005). *Data and Text Mining: A Business Applications Approach*. Pearson Education, Inc.
- Muafi, R. S., & Sanjaya, H. P. (2018). Incorporating Kansei Engineering into Service Quality Tools to Improve the Airline Services. *International Journal for Quality Research*, 12(2), 297–316.
- Mualfah, D., Ramadhoni, Gunawan, R., & Suratno, D. M. (2023). Analisis Sentimen Komentar YouTube TvOne Tentang Ustadz Abdul Somad Dideportasi Dari Singapura Menggunakan Algoritma SVM. *Jurnal FASILKOM*, 13(1), 72–80.
- Mujibulloh, A., & Jakaria, R. B. (2022). *Perancangan Desain Produk Sepatu Safety Dengan Menggunakan Metode Kansei Engineering Dan Model Kano (Studi Kasus Desain Sepatu Karyawan PT Kalam Leverage Mulia)*. 20(1), 33–39.
- Munthe, C. J. E., Hasibuan, N. A., & Hutabarat, H. (2022). Penerapan Algoritma Text Mining Dan TF-RF Dalam Menentukan Promo Produk Pada Marketplace. *Penerapan Algoritma Text Mining Dan TF-RF Dalam Menentukan Promo Produk Pada Marketplace*, 2(3), 110–115.
- Nagamachi, M., & Lokman, A. M. (2011). *Innovations of Kansei Enginnering*. CRC Press.
- Nguyen, H.-K., & Nguyen, T.-D. (2021). Improvement of Service Quality in the Supply Chain of Commercial Banks—A Case Study in Vietnam. *Journal of Risk and Financial Management*, 14(8), 357. <https://doi.org/10.3390/jrfm14080357>



- Nofirza, & Indrayani, K. (2011). Aplikasi Metode Kano dalam Analisis Indikator Kualitas Pelayanan di Rumah Sakit Arifin Ahmad Pekanbaru. *Jurnal Sains, Teknologi Dan Industri*, 9(1), 1–8.
- Nugraha, K. A., & Sebastian, D. (2018). Pembentukan Dataset Topik Kata Bahasa Indonesia pada Twitter Menggunakan TF-IDF & Cosine Similarity. *JUTISI*.
- Nugroho, I. M., Hadiana, A., & Singasatia, D. (2019). Design of Wastu Mobile Interface Using Kansei Engineering. *2019 Fourth International Conference on Informatics and Computing (ICIC)*, 1–8. <https://doi.org/10.1109/ICIC47613.2019.8985955>
- Nurhalimah, N., & Nurhayati, A. (2019). Pengaruh Kualitas Pelayanan dan Nilai Pelanggan terhadap Kepuasan Pelanggan. *Jurnal Ekonomi Dan Bisnis*, 6(2), 1–5.
- Oboudi, A., Akbari Yazdi, H., & abdolahi, soliemann. (2020). Using Mixed Technique of Kansei Engineering, Kano Model, and Taguchi-based Experiment Design to Improve Satisfaction and Participation of Football Spectators at Stadiums. *Research in Sport Management & Motor Behavior*, 10(20), 107–123. <https://doi.org/10.29252/JRSM.10.20.107>
- Octaviani, K., Komara, M. A., & Kurniawan, I. (2022). Analisis Kesuksesan Aplikasi Alfagift Menggunakan Model Delone dan Mclean Studi Kasus Alfa Express Rest Area KM 72B. *Jurnal Informatika Teknologi Dan Sains*, 4(3), 173–178.
- Pattiiha, F. S., & Hendry, H. (2022). Perbandingan Metode K-NN, Naïve Bayes, Decision Tree untuk Analisis Sentimen Tweet Twitter Terkait Opini Terhadap PT PAL Indonesia. *JURIKOM (Jurnal Riset Komputer)*, 9(2), 506. <https://doi.org/10.30865/jurikom.v9i2.4016>
- Prabowo, R. (2019). Penerapan Integrasi Kano dan Kansei Engineering untuk Perbaikan Kualitas Layanan (Studi Kasus: J&T Express Indonesia - Surabaya). *Prosiding SENDI*.
- Priyantina, R., & Sarno, R. (2019). Sentiment Analysis of Hotel Reviews Using Latent Dirichlet Allocation, Semantic Similarity and LSTM. *International Journal of Intelligent Engineering and Systems*, 12(4), 142–155. <https://doi.org/10.22266/ijies2019.0831.14>
- Purbo, O. W. (2019). *Text Mining*. ANDI.
- Puspita, R., & Widodo, A. (2021). Perbandingan Metode KNN, Decision Tree, dan Naïve Bayes Terhadap Analisis Sentimen Pengguna Layanan BPJS. *Jurnal Informatika Universitas Pamulang*, 5(4), 646. <https://doi.org/10.32493/informatika.v5i4.7622>
- Putra, P. S., & Suzianti, A. (2022). Design of a Food Sharing App Using Kansei Engineering and Fuzzy Linguistic Methods. *Proceedings of the 7th North American International Conference on Industrial Engineering and Operations Management*.
- Putra, R. A. S., & Priyanto. (2021). Kano Model Analysis of Android Apps Quality from End User's Preferences. *Journal of Physics: Conference Series*, 1737(1), 012016. <https://doi.org/10.1088/1742-6596/1737/1/012016>

- Putri, R. O., Wibawa, B. M., & Laksamana, T. (2017). Identifikasi Permasalahan Komplain pada E-Commerce Menggunakan Metode. *Jurnal Sains Dan Seni ITS*, 6(1).
- Putri, R. R. E., Istoningtyas, M., & Boerroek, M. R. (2022). Analysis of The Effect of Alfagift's Electronic Service Quality Based on Objectives Using The E-SERVQUAL Method. *The 1st ICOBIMA: Adaptability of Business in the New Economic Era*, 376–384.
- Putu, N. L. P. M., Ahmad Zuli Amrullah, & Ismarmiaty. (2021). Analisis Sentimen dan Pemodelan Topik Pariwisata Lombok Menggunakan Algoritma Naive Bayes dan Latent Dirichlet Allocation. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(1), 123–131. <https://doi.org/10.29207/resti.v5i1.2587>
- Qi, Z. (2021). Research on the Design of E-education Application Interface Based on Kansei Engineering. *8th International Conference*, 161–171. [https://doi.org/10.1007/978-3-030-77889-7\\_11](https://doi.org/10.1007/978-3-030-77889-7_11)
- Rahman, M. A., Hidayat, N., & Supianto, A. A. (2018). Komparasi Metode Data Mining K-Nearest Neighbor Dengan Naive Bayes Untuk Klasifikasi Kualitas Air Bersih (Studi Kasus PDAM Tirta Kencana Kabupaten Jombang). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(12), 6346–6353.
- Ramadhan, N. A., Purnamasari, W. D., & Setyono, D. A. (2020). Konsep Penataan Ruang Terbuka Publik berdasarkan Pola Aktivitas Pengguna (Studi Kasus Alun-alun Kota Bekasi). *Jurnal RUAS*, 18(1).
- Ramadhani, Y., & Suciandani, P. (2011). Analisis Kepuasan Pengunjung Wisata dengan Model Kano Berdasarkan Dimensi SERVQUAL. *Jurnal Teknologi Technoscienza*, 4(1).
- Rangkuti, F. (2008). *Measuring Customer Satisfaction*. PT Gramedia Pustaka Utama.
- Restuputri, D. P., Indriani, T. R., & Masudin, I. (2021). The effect of logistic service quality on customer satisfaction and loyalty using kansei engineering during the COVID-19 pandemic. *Cogent Business & Management*, 8(1). <https://doi.org/10.1080/23311975.2021.1906492>
- Rinaldi, A., & Santoso, S. B. (2018). Analisis Pengaruh Kualitas Informasi, Kualitas Sistem, dan Kualitas Pelayanan terhadap Minat Beli Ulang dengan Kepuasan Pelanggan sebagai Variabel Intervening (Studi pada Pelanggan Traveloka di Kota Semarang). *Diponegoro Journal of Management*, 7(2).
- Robert, U. (2016). THE USE OF KANO MODEL FOR THE CLASSIFICATION OF THE ELEMENTS OF PRODUCT QUALITY. *SYSTEMY WSPOMAGANIA W INŻYNIERII PRODUKCI*, 3(15), 117–126.
- Sandi, G. D. K., Syauqy, D., & Maulana, R. (2019). Sistem Pendeteksi Kesegaran Ikan Bandeng Berdasarkan Bau Dan Warna Daging Berbasis Sensor MQ135 Dan TCS3200 Dengan Metode Naive Bayes. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(10), 10110–10117.

- Sanja, A., Weerakoon, S., Vinsura, K., Ekanayake, U., Bak, V., Haa, S., Jkdt, G., Emuh, E., Dss, W., Hs, H., & Pb, P. (2022). *Designing The BOC Bank App Using Kansei Engineering 2022*. <https://www.researchgate.net/publication/361755127>
- Saputra, I., & Kristiyanti, D. A. (2021). *Machine Learning untuk Pemula*. Informatika.
- Schütte \*, S. T. W., Eklund, J., Axelsson, J. R. C., & Nagamachi, M. (2004). Concepts, methods and tools in Kansei engineering. *Theoretical Issues in Ergonomics Science*, 5(3), 214–231. <https://doi.org/10.1080/1463922021000049980>
- Schütte, S. (2005). *Engineering Emotional Values in Product Design*. Linköpings Universitet.
- Setiawan, A., & Mulyati, S. (2023). ANALISIS SENTIMEN PENGGUNA SHOPEEPAYLATER PADA TWITTER MENGGUNAKAN METODE SUPPORT VECTOR MACHINE (SVM). *Logic*, 1(2).
- Setyawati, K. S., Handojo, A., & Palit, H. N. (2021). Aplikasi Sentiment Analysis Terhadap Pelaksanaan Pembelajaran Jarak Jauh Universitas Kristen Petra Dengan Metode Naive Bayes Classifier. *Jurnal Infra*.
- Singh, N., Sharma, N., & Juneja, A. (2019). *Sentiment Score Analysis for Opinion Mining* (pp. 363–374). [https://doi.org/10.1007/978-981-13-0923-6\\_32](https://doi.org/10.1007/978-981-13-0923-6_32)
- Soliha, E. (2008). Analisis Industri Ritel di Indonesia. *Jurnal Bisnis Dan Ekonomi (JBE)*, 15(2), 128–142.
- Stiyono, A., Sujana, I., & Prawatya, Y. E. (2022). Rancang Bangun Alat Pengepres Kaleng Bekas dengan Menggunakan Metode Kansei Engineering dan Metode Kano. *INTEGRATE*, 6(1), 18–24.
- Suhaimi, A. N. M., & Lokman, A. M. (2018). A Kansei Approach to Investigate Mobile Apps Requirements for Muslim Women Tourist. *Proceedings of the 7th International Conference on Kansei Engineering and Emotion Research 2018: KEER 2018*, 113–120. [https://doi.org/10.1007/978-981-10-8612-0\\_13](https://doi.org/10.1007/978-981-10-8612-0_13)
- Swarnakar, V., Jain, R., Singh, A. R., & Kinker, P. (2020). A QFD-TISM approach for service quality improvement in polytechnic education institutes: a case study. *International Journal of Applied Systemic Studies*, 9(2), 85. <https://doi.org/10.1504/IJASS.2020.10034539>
- Tsani, M. R., Rupaka, A. P. G., Asmoro, L., & Pradana, B. (2020). Analisis Sentimen Review Transportasi Menggunakan Algoritma Support Vector Machine Berbasis Chi Square. *Jurnal Smart Comp*, 9(1), 2020.
- Turland, M. (2010). *php| architect's Guide to Web scraping*. Marco Tabini & Associates, Inc.
- Ubaidillah, M. J., Munadhif, I., & Rinanto, N. (2019). Klasifikasi Gelombang Otot Lengan Pada Robot Manipulator Menggunakan Support Vector Machine. *Rekayasa*, 12(2), 91–97. <https://doi.org/10.21107/rekayasa.v12i2.5406>

- Ulfah, A. N., & Anam, M. K. (2020). Analisis Sentimen Hate Speech Pada Portal Berita Online Menggunakan Support Vector Machine (SVM). *Jurnal Teknik Informatika Dan Sistem Informasi*, 7(10), 1–10.
- Utomo, T. J. (2009). Fungsi dan Peran Bisnis Ritel dalam Saluran Pemasaran. *Fokus Ekonomi*, 4(1), 44–55.
- Wang, W. M., Li, Z., Tian, Z. G., Wang, J. W., & Cheng, M. N. (2018). Extracting and summarizing affective features and responses from online product descriptions and reviews: A Kansei text mining approach. *Engineering Applications of Artificial Intelligence*, 73, 149–162. <https://doi.org/10.1016/j.engappai.2018.05.005>
- We Are Social. (2023, January). *We Are Social*.
- Wijaya, T. (2011). *Manajemen kualitas Jasa*. PT Indeks.
- Yudianto, M. R. A., Rahim, A., Sukmasetya, P., & Hasani, R. A. (2022). Perbandingan Metode Support Vector Machine dengan Metode Lexicon dalam Analisis Sentimen Bahasa Indonesia. *Jurnal Teknologi Informasi*, 6(1).
- Yusril, A. N., Larasati, I., & Zukri, P. Al. (2021). Systematic Literature Review Analisis Metode Agile dalam Pengembangan Aplikasi Mobile. *SISTEMASI*, 10(2), 369–380.

## LAMPIRAN

### A - 1 Hasil Klasifikasi *Machine Learning*

```
MultinomialNB accuracy: 0.7891416237755271
MultinomialNB Precision: 0.8527777777777777
MultinomialNB Recall: 0.20144356955380577
MultinomialNB f1_score: 0.32590233545647557
confusion matrix:
[[ 307 1217]
 [  53 4446]]
```

```
=====
                precision    recall  f1-score   support

   Negative         0.85         0.20         0.33         1524
   Positive         0.79         0.99         0.88         4499

   accuracy                   0.79         6023
  macro avg         0.82         0.59         0.60         6023
 weighted avg         0.80         0.79         0.74         6023
```

#### Lampiran A-1. 1 Hasil Klasifikasi Naïve Bayes 60:40

```
SVM accuracy: 0.9017101112402457
SVM Precision: 0.8629283489096573
SVM Recall: 0.7270341207349081
SVM f1_score: 0.7891737891737892
confusion matrix:
[[1108  416]
 [ 176 4323]]
```

```
=====
                precision    recall  f1-score   support

   Negative         0.86         0.73         0.79         1524
   Positive         0.91         0.96         0.94         4499

   accuracy                   0.90         6023
  macro avg         0.89         0.84         0.86         6023
 weighted avg         0.90         0.90         0.90         6023
```

#### Lampiran A-1. 2 Hasil Klasifikasi SVM Linear 60:40

```

SVM accuracy: 0.9038685040677403
SVM Precision: 0.8396836808051761
SVM Recall: 0.7664041994750657
SVM f1_score: 0.8013722126929674
confusion matrix:
[[1168 356]
 [ 223 4276]]
=====

```

	precision	recall	f1-score	support
Negative	0.84	0.77	0.80	1524
Positive	0.92	0.95	0.94	4499
accuracy			0.90	6023
macro avg	0.88	0.86	0.87	6023
weighted avg	0.90	0.90	0.90	6023

### Lampiran A-1. 3 Hasil Klasifikasi SVM Kernel Linear 60:40

```

SVM accuracy: 0.8284907853229287
SVM Precision: 0.8714069591527988
SVM Recall: 0.3779527559055118
SVM f1_score: 0.5272311212814645
confusion matrix:
[[ 576 948]
 [ 85 4414]]
=====

```

	precision	recall	f1-score	support
Negative	0.87	0.38	0.53	1524
Positive	0.82	0.98	0.90	4499
accuracy			0.83	6023
macro avg	0.85	0.68	0.71	6023
weighted avg	0.84	0.83	0.80	6023

### Lampiran A-1. 4 Hasil Klasifikasi SVM Kernel Poly 60:40

---

```

SVM accuracy: 0.8942387514527644
SVM Precision: 0.8152096659559346
SVM Recall: 0.7526246719160105
SVM f1_score: 0.7826680313886046
confusion matrix:
[[1147  377]
 [ 260 4239]]
=====

```

	precision	recall	f1-score	support
Negative	0.82	0.75	0.78	1524
Positive	0.92	0.94	0.93	4499
accuracy			0.89	6023
macro avg	0.87	0.85	0.86	6023
weighted avg	0.89	0.89	0.89	6023

#### Lampiran A-1. 5 Hasil Klasifikasi SVM Kernel Sigmoid 60:40

```

SVM accuracy: 0.9017101112402457
SVM Precision: 0.8629283489096573
SVM Recall: 0.7270341207349081
SVM f1_score: 0.7891737891737892
confusion matrix:
[[1108  416]
 [ 176 4323]]
=====

```

	precision	recall	f1-score	support
Negative	0.86	0.73	0.79	1524
Positive	0.91	0.96	0.94	4499
accuracy			0.90	6023
macro avg	0.89	0.84	0.86	6023
weighted avg	0.90	0.90	0.90	6023

#### Lampiran A-1. 6 Hasil Klasifikasi SVM Kernel RBF 60:40

```

KNN accuracy: 0.809231280092977
KNN Precision: 0.6195028680688337
KNN Recall: 0.6377952755905512
KNN f1_score: 0.6285160038797285
confusion matrix:
[[ 972  552]
 [ 597 3902]]
=====

```

	precision	recall	f1-score	support
Negative	0.62	0.64	0.63	1524
Positive	0.88	0.87	0.87	4499
accuracy			0.81	6023
macro avg	0.75	0.75	0.75	6023
weighted avg	0.81	0.81	0.81	6023

#### Lampiran A-1. 7 Hasil Klasifikasi K-NN 60:40

```

Decision Tree accuracy: 0.8022580109579943
Decision Tree Precision: 0.611371237458194
Decision Tree Recall: 0.5997375328083989
Decision Tree f1_score: 0.605498509440212
confusion matrix:
[[ 914  610]
 [ 581 3918]]
=====

```

	precision	recall	f1-score	support
Negative	0.61	0.60	0.61	1524
Positive	0.87	0.87	0.87	4499
accuracy			0.80	6023
macro avg	0.74	0.74	0.74	6023
weighted avg	0.80	0.80	0.80	6023

#### Lampiran A-1. 8 Hasil Klasifikasi Decision Tree 60:40

```

MultinomialNB accuracy: 0.79765330971884
MultinomialNB Precision: 0.8419354838709677
MultinomialNB Recall: 0.23179396092362345
MultinomialNB f1_score: 0.36350974930362123
confusion matrix:
[[ 261  865]
 [  49 3342]]
=====

```

	precision	recall	f1-score	support
Negative	0.84	0.23	0.36	1126
Positive	0.79	0.99	0.88	3391
accuracy			0.80	4517
macro avg	0.82	0.61	0.62	4517
weighted avg	0.81	0.80	0.75	4517

#### Lampiran A-1. 9 Hasil Klasifikasi Naive Bayes 70:30



```

SVM accuracy: 0.9076820898826655
SVM Precision: 0.8628454452405322
SVM Recall: 0.7486678507992895
SVM f1_score: 0.8017118402282453
confusion matrix:
[[ 843 283]
 [ 134 3257]]
=====

```

	precision	recall	f1-score	support
Negative	0.86	0.75	0.80	1126
Positive	0.92	0.96	0.94	3391
accuracy			0.91	4517
macro avg	0.89	0.85	0.87	4517
weighted avg	0.91	0.91	0.91	4517

#### Lampiran A-1. 10 Hasil Klasifikasi SVM Linear 70:30

```

SVM accuracy: 0.9147664379012619
SVM Precision: 0.8465855940130963
SVM Recall: 0.8037300177619894
SVM f1_score: 0.8246013667425968
confusion matrix:
[[ 905 221]
 [ 164 3227]]
=====

```

	precision	recall	f1-score	support
Negative	0.85	0.80	0.82	1126
Positive	0.94	0.95	0.94	3391
accuracy			0.91	4517
macro avg	0.89	0.88	0.88	4517
weighted avg	0.91	0.91	0.91	4517

#### Lampiran A-1. 11 Hasil Klasifikasi SVM Kernel Linear 70:30

```

SVM accuracy: 0.8326322780606598
SVM Precision: 0.8530534351145038
SVM Recall: 0.3969804618117229
SVM f1_score: 0.5418181818181818
confusion matrix:
[[ 447 679]
 [ 77 3314]]
=====

```

	precision	recall	f1-score	support
Negative	0.85	0.40	0.54	1126
Positive	0.83	0.98	0.90	3391
accuracy			0.83	4517
macro avg	0.84	0.69	0.72	4517
weighted avg	0.84	0.83	0.81	4517

#### Lampiran A-1. 12 Hasil Klasifikasi SVM Kernel Poly 70:30

```

SVM accuracy: 0.8992694266105823
SVM Precision: 0.8109360518999074
SVM Recall: 0.7770870337477798
SVM f1_score: 0.7936507936507936
confusion matrix:
[[ 875 251]
 [ 204 3187]]
=====

```

	precision	recall	f1-score	support
Negative	0.81	0.78	0.79	1126
Positive	0.93	0.94	0.93	3391
accuracy			0.90	4517
macro avg	0.87	0.86	0.86	4517
weighted avg	0.90	0.90	0.90	4517

Lampiran A-1. 13 Hasil Klasifikasi SVM Kernel Sigmoid 70:30

```

SVM accuracy: 0.9076820898826655
SVM Precision: 0.8628454452405322
SVM Recall: 0.7486678507992895
SVM f1_score: 0.8017118402282453
confusion matrix:
[[ 843 283]
 [ 134 3257]]
=====

```

	precision	recall	f1-score	support
Negative	0.86	0.75	0.80	1126
Positive	0.92	0.96	0.94	3391
accuracy			0.91	4517
macro avg	0.89	0.85	0.87	4517
weighted avg	0.91	0.91	0.91	4517

Lampiran A-1. 14 Hasil Klasifikasi SVM Kernel RBF 70:30

```

KNN accuracy: 0.8144786362630064
KNN Precision: 0.6230769230769231
KNN Recall: 0.6474245115452931
KNN f1_score: 0.6350174216027874
confusion matrix:
[[ 729 397]
 [ 441 2950]]
=====

```

	precision	recall	f1-score	support
Negative	0.62	0.65	0.64	1126
Positive	0.88	0.87	0.88	3391
accuracy			0.81	4517
macro avg	0.75	0.76	0.76	4517
weighted avg	0.82	0.81	0.82	4517

#### Lampiran A-1. 15 Hasil Klasifikasi K-NN 70:30

```

Decision Tree accuracy: 0.8100509187513837
Decision Tree Precision: 0.6245353159851301
Decision Tree Recall: 0.5968028419182948
Decision Tree f1_score: 0.6103542234332425
confusion matrix:
[[ 672 454]
 [ 404 2987]]
=====

```

	precision	recall	f1-score	support
Negative	0.62	0.60	0.61	1126
Positive	0.87	0.88	0.87	3391
accuracy			0.81	4517
macro avg	0.75	0.74	0.74	4517
weighted avg	0.81	0.81	0.81	4517

#### Lampiran A-1. 16 Hasil Klasifikasi Decision Tree 70:30

```

MultinomialNB accuracy: 0.798804780876494
MultinomialNB Precision: 0.7933884297520661
MultinomialNB Recall: 0.25668449197860965
MultinomialNB f1_score: 0.3878787878787879
confusion matrix:
[[ 192 556]
 [ 50 2214]]
=====

```

	precision	recall	f1-score	support
Negative	0.79	0.26	0.39	748
Positive	0.80	0.98	0.88	2264
accuracy			0.80	3012
macro avg	0.80	0.62	0.63	3012
weighted avg	0.80	0.80	0.76	3012

#### Lampiran A-1. 17 Hasil Klasifikasi Naive Bayes 80:20

```

SVM accuracy: 0.9063745019920318
SVM Precision: 0.8562691131498471
SVM Recall: 0.7486631016042781
SVM f1_score: 0.7988587731811697
confusion matrix:
[[ 560 188]
 [ 94 2170]]
=====

```

	precision	recall	f1-score	support
Negative	0.86	0.75	0.80	748
Positive	0.92	0.96	0.94	2264
accuracy			0.91	3012
macro avg	0.89	0.85	0.87	3012
weighted avg	0.90	0.91	0.90	3012

### Lampiran A-1. 18 Hasil Klasifikasi SVM Linear 80:20

```

SVM accuracy: 0.9140106241699867
SVM Precision: 0.8477951635846372
SVM Recall: 0.7967914438502673
SVM f1_score: 0.8215024121295659
confusion matrix:
[[ 596 152]
 [ 107 2157]]
=====

```

	precision	recall	f1-score	support
Negative	0.85	0.80	0.82	748
Positive	0.93	0.95	0.94	2264
accuracy			0.91	3012
macro avg	0.89	0.87	0.88	3012
weighted avg	0.91	0.91	0.91	3012

### Lampiran A-1. 19 Hasil Klasifikasi SVM Kernel Linear 80:20

```

SVM accuracy: 0.8356573705179283
SVM Precision: 0.850415512465374
SVM Recall: 0.410427807486631
SVM f1_score: 0.5536519386834987
confusion matrix:
[[ 307 441]
 [ 54 2210]]
=====

```

	precision	recall	f1-score	support
Negative	0.85	0.41	0.55	748
Positive	0.83	0.98	0.90	2264
accuracy			0.84	3012
macro avg	0.84	0.69	0.73	3012
weighted avg	0.84	0.84	0.81	3012

### Lampiran A-1. 20 Hasil Klasifikasi SVM Kernel Poly 80:20

```

SVM accuracy: 0.9023904382470119
SVM Precision: 0.8179271708683473
SVM Recall: 0.7807486631016043
SVM f1_score: 0.79890560875513
confusion matrix:
[[ 584 164]
 [ 130 2134]]
=====

```

	precision	recall	f1-score	support
Negative	0.82	0.78	0.80	748
Positive	0.93	0.94	0.94	2264
accuracy			0.90	3012
macro avg	0.87	0.86	0.87	3012
weighted avg	0.90	0.90	0.90	3012

#### Lampiran A-1. 21 Hasil Klasifikasi SVM Kernel Sigmoid 80:20

```

SVM accuracy: 0.9063745019920318
SVM Precision: 0.8562691131498471
SVM Recall: 0.7486631016042781
SVM f1_score: 0.7988587731811697
confusion matrix:
[[ 560 188]
 [ 94 2170]]
=====

```

	precision	recall	f1-score	support
Negative	0.86	0.75	0.80	748
Positive	0.92	0.96	0.94	2264
accuracy			0.91	3012
macro avg	0.89	0.85	0.87	3012
weighted avg	0.90	0.91	0.90	3012

#### Lampiran A-1. 22 Hasil Klasifikasi SVM Kernel RBF 80:20

```

KNN accuracy: 0.8183930942895087
KNN Precision: 0.627700127064803
KNN Recall: 0.660427807486631
KNN f1_score: 0.6436482084690553
confusion matrix:
[[ 494 254]
 [ 293 1971]]
=====

```

	precision	recall	f1-score	support
Negative	0.63	0.66	0.64	748
Positive	0.89	0.87	0.88	2264
accuracy			0.82	3012
macro avg	0.76	0.77	0.76	3012
weighted avg	0.82	0.82	0.82	3012

#### Lampiran A-1. 23 Hasil Klasifikasi K-NN 80:20

```

Decision Tree accuracy: 0.8157370517928287
Decision Tree Precision: 0.6331034482758621
Decision Tree Recall: 0.6136363636363636
Decision Tree f1_score: 0.6232179226069247
confusion matrix:
[[ 459 289]
 [ 266 1998]]
=====

```

	precision	recall	f1-score	support
Negative	0.63	0.61	0.62	748
Positive	0.87	0.88	0.88	2264
accuracy			0.82	3012
macro avg	0.75	0.75	0.75	3012
weighted avg	0.81	0.82	0.81	3012

#### Lampiran A-1. 24 Hasil Klasifikasi Decision Tree 80:20

```

MultinomialNB accuracy: 0.796148738379814
MultinomialNB Precision: 0.7480314960629921
MultinomialNB Recall: 0.25675675675675674
MultinomialNB f1_score: 0.38229376257545267
confusion matrix:
[[ 95 275]
 [ 32 1104]]
=====

```

	precision	recall	f1-score	support
Negative	0.75	0.26	0.38	370
Positive	0.80	0.97	0.88	1136
accuracy			0.80	1506
macro avg	0.77	0.61	0.63	1506
weighted avg	0.79	0.80	0.76	1506

#### Lampiran A-1. 25 Hasil Klasifikasi Naive Bayes 90:10

```

SVM accuracy: 0.9130146082337317
SVM Precision: 0.8610271903323263
SVM Recall: 0.7702702702702703
SVM f1_score: 0.8131241084165479
confusion matrix:
[[ 285 85]
 [ 46 1090]]
=====

```

	precision	recall	f1-score	support
Negative	0.86	0.77	0.81	370
Positive	0.93	0.96	0.94	1136
accuracy			0.91	1506
macro avg	0.89	0.86	0.88	1506
weighted avg	0.91	0.91	0.91	1506

#### Lampiran A-1. 26 Hasil Klasifikasi SVM Linear 90:10

```

SVM accuracy: 0.9223107569721115
SVM Precision: 0.8666666666666667
SVM Recall: 0.8081081081081081
SVM f1_score: 0.8363636363636363
confusion matrix:
[[ 299  71]
 [  46 1090]]
=====

```

	precision	recall	f1-score	support
Negative	0.87	0.81	0.84	370
Positive	0.94	0.96	0.95	1136
accuracy			0.92	1506
macro avg	0.90	0.88	0.89	1506
weighted avg	0.92	0.92	0.92	1506

#### Lampiran A-1. 27 Hasil Klasifikasi SVM Kernel Linear 90:10

```

SVM accuracy: 0.8379814077025233
SVM Precision: 0.8705882352941177
SVM Recall: 0.4
SVM f1_score: 0.5481481481481483
confusion matrix:
[[ 148  222]
 [  22 1114]]
=====

```

	precision	recall	f1-score	support
Negative	0.87	0.40	0.55	370
Positive	0.83	0.98	0.90	1136
accuracy			0.84	1506
macro avg	0.85	0.69	0.72	1506
weighted avg	0.84	0.84	0.81	1506

#### Lampiran A-1. 28 Hasil Klasifikasi SVM Kernel Poly 90:10

```

SVM accuracy: 0.898406374501992
SVM Precision: 0.8056338028169014
SVM Recall: 0.772972972972973
SVM f1_score: 0.7889655172413793
confusion matrix:
[[ 286  84]
 [  69 1067]]
=====

```

	precision	recall	f1-score	support
Negative	0.81	0.77	0.79	370
Positive	0.93	0.94	0.93	1136
accuracy			0.90	1506
macro avg	0.87	0.86	0.86	1506
weighted avg	0.90	0.90	0.90	1506

#### Lampiran A-1. 29 Hasil Klasifikasi SVM Kernel Sigmoid 90:10

```

SVM accuracy: 0.9130146082337317
SVM Precision: 0.8610271903323263
SVM Recall: 0.7702702702702703
SVM f1_score: 0.8131241084165479
confusion matrix:
[[ 285  85]
 [ 46 1090]]
=====

```

	precision	recall	f1-score	support
Negative	0.86	0.77	0.81	370
Positive	0.93	0.96	0.94	1136
accuracy			0.91	1506
macro avg	0.89	0.86	0.88	1506
weighted avg	0.91	0.91	0.91	1506

### Lampiran A-1. 30 Hasil Klasifikasi SVM Kernel RBF 90:10

```

KNN accuracy: 0.8180610889774237
KNN Precision: 0.6243523316062176
KNN Recall: 0.6513513513513514
KNN f1_score: 0.6375661375661376
confusion matrix:
[[241 129]
 [145 991]]
=====

```

	precision	recall	f1-score	support
Negative	0.62	0.65	0.64	370
Positive	0.88	0.87	0.88	1136
accuracy			0.82	1506
macro avg	0.75	0.76	0.76	1506
weighted avg	0.82	0.82	0.82	1506

### Lampiran A-1. 31 Hasil Klasifikasi K-NN 90:10

```

Decision Tree accuracy: 0.8193891102257637
Decision Tree Precision: 0.6310160427807486
Decision Tree Recall: 0.6378378378378379
Decision Tree f1_score: 0.6344086021505376
confusion matrix:
[[236 134]
 [138 998]]
=====

```

	precision	recall	f1-score	support
Negative	0.63	0.64	0.63	370
Positive	0.88	0.88	0.88	1136
accuracy			0.82	1506
macro avg	0.76	0.76	0.76	1506
weighted avg	0.82	0.82	0.82	1506

### Lampiran A-1. 32 Hasil Klasifikasi Decision Tree 90:10



### A - 2 Hasil Pembobotan

	term	rank			
			34	sekali	304.596455
7	bagus	1447.506230	46	suka	278.258997
10	baik	959.739305	14	pakai	277.348952
4	sangat	926.615202	9	sudah	272.093076
0	aplikasi	728.311158	40	banget	267.049987
8	bantu	711.324465	18	rumah	266.689574
2	belanja	667.199236	31	poin	263.932582
5	mudah	644.950097	22	alfamart	263.889356
1	tidak	623.742295	15	barang	260.989492
39	lambat	511.661602	38	guna	256.527401
3	alfajift	493.819785	23	bayar	241.689069
13	baru	417.000819	25	antar	221.258868
11	banyak	402.451078	37	selalu	215.606674
27	buka	401.441422	35	alfa	210.886516
12	kirim	382.445569	48	sering	204.584017
29	layan	359.833333	28	buat	200.840317
17	lama	353.659443	36	sekarang	195.988008
19	masuk	341.254126	32	beli	195.527885
6	pesan	330.433246	24	kalau	186.691948
26	cepat	328.476656	43	kasih	186.452536
16	terus	324.082918	44	jam	182.015447
20	promo	321.227187	30	toko	179.153334
21	lebih	319.698475	42	gratis	175.816481
			47	ongkir	169.450138
			41	keluar	166.279584
			33	sama	153.625760
			45	sampai	137.480110
			49	langsung	130.319161

Lampiran A-2. 1 Ranking Pembobotan

### A - 3 Hasil Asosiasi Kata

```
In [111]: rules = rules.sort_values(by='confidence', ascending=False)
rules
```

```
Out[111]:
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
359	(baik, buka)	(aplikasi)	0.010980	0.501235	0.010431	0.950000	1.895318	0.004927	9.975295	0.477628
347	(bisa)	(tidak)	0.047214	0.441120	0.043920	0.930233	2.108797	0.023093	8.010614	0.551851
468	(lambat, buka)	(aplikasi)	0.013725	0.501235	0.012627	0.920000	1.835465	0.005748	6.234559	0.461513
654	(bisa, aplikasi)	(tidak)	0.022783	0.441120	0.020587	0.903614	2.046455	0.010537	5.798380	0.523760
608	(terus, buka)	(aplikasi)	0.024979	0.501235	0.022234	0.890110	1.775833	0.009714	4.538759	0.448077
...	...	...	...	...	...	...	...	...	...	...
200	(barang)	(toko)	0.096075	0.066978	0.024156	0.251429	3.753911	0.017721	1.246404	0.811584
635	(daftar)	(aplikasi, tidak)	0.048037	0.223168	0.012078	0.251429	1.126635	0.001358	1.037753	0.118073
482	(aplikasi, belanja)	(alfagift)	0.073566	0.114741	0.018391	0.250000	2.178828	0.009950	1.180346	0.584000
356	(sama)	(sekali, tidak)	0.043920	0.023058	0.010980	0.250000	10.842262	0.009967	1.302589	0.949469
797	(sudah, aplikasi, tidak)	(belanja)	0.043920	0.130936	0.010980	0.250000	1.909329	0.005229	1.158752	0.498134

```
In [112]: rules.to_csv("Asosiasi.csv", index = False)
```

### Lampiran A-3. 1 Hasil Asosiasi

### A - 4 Pseudocode Keseluruhan

```
pip install PySastrawi
import string
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import nltk
import re
import sys
```

```
import csv
df = pd.read_csv('dataset.csv', delimiter = ';', encoding = 'utf-8')

#proses data cleaning dan case folding
import re
import string
```

```

def cleansing(Review):
    #Mengubah teks menjadi lower
    Review = re.sub('[^a-zA-Z]', '', Review).lower()
    #Menghapus tanda baca
    Review = Review.translate(str.maketrans("", "", string.punctuation))
    #Remove non ASCII (emoticon, chinese word, etc.)
    Review = Review.encode('ascii', 'replace').decode('ascii')
    #Remove mention, link, hashtag
    Review = ' '.join(re.sub("([@#][A-Za-z0-9]+)(\w+:\V\S+)", " ", Review).split())
    #Menghapus https dan http
    Review = re.sub(r"(?:\@|http?:\://|https?:\://|www)\S+", "", Review)
    #Menghapus angka
    Review = re.sub(r"d+", " ", Review)
    #Remove single char
    Review = re.sub(r"\b[a-zA-Z]\b", "", Review)
    #Mengganti line baru dengan spasi
    Review = re.sub("\n", " ", Review)
    #Remove multiple whitespace into single whitespace
    Review = re.sub("\s+', ' ', Review)
    return Review

df['Review_cleansing'] = df['Review'].apply(cleansing)

#tokenize
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize
def tokenize (txt):
    word_token = word_tokenize (txt)
    return word_token
df['Review_token'] = df['Review_cleansing'].apply(lambda x: tokenize (x))

# stop words sastrawi

```

```

from          Sastrawi.StopWordRemover.StopWordRemoverFactory          import
StopWordRemoverFactory, StopWordRemover, ArrayDictionary

#Create Factory + Ambil Stopword Bawaan
stop_factory = StopWordRemoverFactory()
more_stopword = ['muluuuu', 'agszblj', 'yang', 'tetepa', 'hufft', 'kagak', 'nya', 'ya', 'saja', 'aja',
'padahal', 'kokk', 'loh', 'di', 'banget', 'dengan', 'deh' 'emang', 'kalau', 'sih', 'lautze', 'men', 'tuh',
'itu', 'tu', 'ituu', 'dong', 'donk', 'kali', 'yaaahhh', 'yaaaah', 'hadehhh', 'iyya', 'yaa', 'yah', 'nih',
'sayah', 'haduhhhhh', 'biat', 'kaa', 'qu', 'nha', 'yuuuu', 'tadinya', 'aku', 'kak', 'diky', 'maulana',
'enggak', 'yuuu', 'yya', 'sik', 'mba', 'koyo', 'tks', 'gw', 'ajh', 'yg', 'yng', 'sing', 'trm', 'ksh', 'je',
'jooosssss', 'anjir', 'nb', 'gb', 'cuma', 'gedek', 'wkwkwk', 'mah', 'refind', 'ayolah', 'behhhhhh',
'bpnya', 'mrса', 'kesni', 'anjirrrrr', 'lhoh', 'cok', 'cht', 'jm', 'utuk', 'tewur', 'aggqgd', 'agtmsgn',
'lagih', 'tikarriss', 'kecust', 'jug', 'qk', 'ho', 'huft', 'du', 'bualazisokhi', 'ndruru', 'ayo', 'tahi', 'lho',
'padahak', 'sampr', 'sekalii', 'dansering', 'gww', 'aduh', 'kabr', 'deh', 'kadang', 'mau', 'ini', 'jadi',
'kalau', 'mending', 'dari', 'mana', 'malah', 'masa', 'kayak', 'mending' 'payah', 'kalaw', 'agwxlby',
'agszpjz', 'fft', 'rrtr', 'rf', 'gc', 'lllllllllllllllllllllllllllllllll', 'nah', 'jajanannya', 'yamana',
'hmmmmmmmm', 'peh', 'su', 'kenap', 'hpku', 'drmh', 'getiiiiingggg', 'lbihbyk', 'gi', 'loch', 'bedug',
'trss', 'rbu', 'melulu', 'josss', 'bor', 'jhooossss', 'uuu', 'woy', 'nian', 'sat', 'shg', 'maksh', 'puk', 'ag',
'alafa', 'agtylfq', 'broo', 'oesa', 'nisa', 'riazl', 'choi', 'xoxo', 'widih', 'ca', 'suruhkomplain', 'qq',
'uuuuwhh', 'agxjjmc', 'pv', 'hehehe', 'gang', 'kok', 'pripriitt', 'pilhan', 'yuk', 'wkkkkk', 'mq', 'kgk',
'dsb', 'yivni', 'cobaik', 'hariii', 'wkkw', 'hai', 'hahahhah', 'bye', 'bwa', 'wkwk', 'tq', 'is',
'satsetsatset', 'teruwu', 'uwu', 'yoyo', 'satset', 'siii', 'itupun', 'sskarang', 'ceunahhhh', 'agstkft',
'urusannsaya', 'cetho', 'iki', 'oy', 'yuhu', 'nybselalu', 'my', 'lingk', 'ii', 'ajinggg', 'bqbi', 'thank',
'apa', 'wok', 'lu', 'wi', 'cuakkssss', 'mmit', 'ndk', 'abcd', 'yowesben', 'cnuh', 'pihat', 'sory', 'dksh',
'hmm', 'maknyossssss', 'terrrrunccchhh', 'emangg', 'aakan', 'kan', 'distruk', 'disemoilin', 'se',
'gue', 'ah', 'gakelas', 'entahlah', 'dall', 'lah', 'pula', 'hhhhh', 'sarankn', 'masih', 'trmksh', 'jw',
'klari', 'hemm', 'bk', 'jj', 'sukakmu', 'ajalaahhhh', 'lol', 'sipp', 'lgy', 'jozz', 'kapook', 'lsng',
'bangeetttt', 'hadeuhhhhh', 'jdnya', 'brkali', 'ach', 'trmkasih', 'ogah', 'nying', 'gr', 'agnwlnx', 'gd',
'tokok', 'syag', 'udanawu', 'strukyng', 'mughni', 'hehee', 'tos', 'pukil', 'uan', 'cux', 'sangaat',
'ilha', 'iyahh', 'ng', 'chuakss', 'hahahah', 'idh', 'ckckckc', 'pulaaak', 'hadeeeh', 'ddrrrrrds', 'zzuuu',
'xx', 'seei', 'langkaplancar', 'hahahahahahahaha', 'ci', 'woiiii', 'pokokny', 'vc', 'xd', 'io', 'op', 'ig',
'tlongperbaiki', 'setegh', 'kapokkk', 'di', 'dannn', 'ntb', 'udsh', 'stampny', 'seharusx', 'dtoko',

```

'haduhh', 'efkuasi', 'mntuuul', 'pisann', 'kaannn', 'zzzzz', 'pesanankita', 'siniiii', 'bangettntttttt',  
 'dd', 'sss', 'did', 'chfigog', 'hemmmm', 'pffft', 'uh', 'ff', 'ooo', 'hahahha', 'wkwkwkwkw', 'lmyn',  
 'xxxx', 'hm', 'agzzryh', 'eeeeeeuuuuuuuhhhh', 'haliaangsss', 'indoxxx', 'llahh', 'haduhhhh',  
 'deeehhh', 'mulugiliran', 'bcndaaaa', 'anjingg', 'trusssssssss', 'mantullllll', 'mantappppppppp',  
 'okedahhh', 'ntapss', 'mueheheh', 'deeeh', 'tapiiiiiii', 'sumpaaaah', 'hahhahahahhah',  
 'mantaabbb', 'mantjaapp', 'sxxx', 'suiiiipp', 'ggg', 'hhh', 'siip', 'mndg', 'serri', 'kocaggg',  
 'siiiip', 'berxx', 'alfao', 'membantunperencanaan', 'mantapppp', 'alfifagif',  
 'mantaaaaaaapppppp', 'debbie', 'sangatt', 'agxzqrs', 'bia', 'oyeeee', 'adhityofirdausputra',  
 'mantb', 'mamtap', 'benja', 'maszeehh', 'maszeehh', 'membantuuuuuu', 'ge', 'siiiipp', 'qqqqq',  
 'manttttapppp', 'teruuusss', 'mantaapp', 'teruuusss', 'hadeeehh', 'www', 'fg', 'xvi', 'okkkk',  
 'laahhhh', 'hellooo', 'sekaleeeeeeee', 'qn', 'agvkncq', 'waduhhhh', 'pnx', 'klopun', 'tg',  
 'ekwkekww', 'aaaa', 'kenak', 'mengsad', 'tmpa', 'alfaaa', 'semangaaat', 'terosssss', 'naon', 'ajg',  
 'hh', 'barkot', 'pqapku', 'msknstam', 'bcanya', 'huhhh', 'kenp', 'terusssss', 'malzz',  
 'wkwkwkwkw', 'by', 'sla', 'mantapppppppp', 'mantapmantap', 'pokoknyaa', 'mantapsssss',  
 'semuapastidapat', 'joss', 'gandos', 'coyyyyy', 'maauk', 'okhe', 'josssss', 'ntabbss', 'lokotreee',  
 'kuwokkk', 'josssmantabbbb', 'pdahla', 'mmtaaap', 'downloadlagi', 'pollllll', 'mlahan',  
 'keapliksinya', 'cex', 'hilih', 'aph', 'hadeehhh', 'appah', 'sekarank', 'wkwkwkwkwkw', 'au', 'qa',  
 'lllllllll', 'brow', 'mampusss', 'sukeeee', 'helooowww', 'siti', 'nurbayaaaaa',  
 'adannghfbbfbfbfngngn', 'aamma', 'mmmxmmdmdmkkdkdnrjyfuyhcjcnfnfnn',  
 'ncnfnncjdjjdhhxhbbfncncnfjfnfnngngmcmnmnctjnnfmmgmmhmmhmmhmmhmmmmnbn  
 mbmgkjkkfkfkfkckcmfmmfmmcmccmcmcmffmjfno', 'dek',  
 'nxnxbhxbxbxbxbxnxnxbxnsnnxnsnnxnjznbxbxbnxnx', 'iccx', 'xz', 'jadj', 'nurul',  
 'fikri', 'nf', 'bangeeetttt', 'teruuuss', 'terooooosssss', 'lov', 'uu', 'yuuuuu', 'slurrr', 'chas',  
 'selaluada', 'mucel', 'bangatt', 'pp', 'okt', 'ret', 'maav', 'hemmmmm', 'beknya', 'awwas', 'sken',  
 'maneh', 'sono', 'kon', 'iku', 'gtau', 'oleng', 'gaskeun', 'seriiiiiiing', 'ditnya', 'awq', 'mngnai',  
 'selaluuuu', 'matap', 'ai', 'terpa', 'woy', 'wooooooyyyyyyy', 'anjim', 'agxldqz', 'virgo', 'sexan',  
 'wadidaw', 'nbb', 'mmjju', 'uy', 'gnji', 'himne', 'gen', 'jin', 'gam', 'cumak', 'sekaidg', 'ieu', 'klkk',  
 'dupeeeer', 'hariau', 'transksi', 'paaassss', 'mpah', 'madesuuu', 'pnah', 'ttooopppp', 'ngontek',  
 'apaam', 'hahah', 'three', 'alahhh', 'lngsungpun', 'recom', 'oks', 'menemptn', 'nebgamvil',  
 'kobtak', 'jmen', 'kkkkk', 'grgrgrgrgrgr', 'cuss', 'sukseme', 'thaknk', 'dllnya', 'pulaaa',  
 'terusyaaa', 'oi', 'taee', 'daahhh', 'begimana', 'beberapakali', 'idn', 'ba', 'uraaa', 'waahhh', 'eeeist',  
 'taaadaa', 'laahh', 'lnua', 'gasspoll', 'pokonyaa', 'komplein', 'oh', 'aaaa', 'pluss', 'omooo', 'gj',

'sni', 'aplagi', 'swaktu', 'mntaapp', 'tya', 'murahmeriahmerakyat', 'yampon', 'prt', 'dddddd', 'kaapooook', 'daebak', 'alfaambil', 'selali', 'termantulityyy', 'duanyaaa', 'yaampunn', 'alfanyaa', 'hayuuu', 'mantulllll', 'warrrrr', 'hehehhe', 'shari', 'jdhie', 'mskih', 'pdal', 'paharah', 'terlopelope', 'eeh', 'hadeeh', 'bro', 'hahaha', 'denga', 'sampk', 'cuz', 'duh', 'ataw', 'bkr', 'bangeetxzzz', 'thnkyou', 'soa', 'aaa', 'mallaem', 'ditelfongakk', 'ata', 'gercepp', 'tebaek', 'bkl', 'arigattou', 'monangis', 'siscakhol', 'agqlmzk', 'setelh', 'akhhhh', 'asyukaaa', 'hadeuuuuuh', 'wawan', 'prikkkkkkkkk', 'prik', 'haiyahhh', 'anying', 'ntahlah', 'yeh', 'dog', 'gimanaaaaa', 'lunya', 'goblogg', 'goblokk', 'tolloll', 'pastinyaaa', 'gerceptanpabatas', 'omo', 'npa', 'pakememberalfadiskonnyagede', 'sx', 'xxxxxx', 'terussssssss', 'af', 'capcus', 'lebin', 'mon', 'nasir', 'rotunnadillah', 'ulul', 'semantap', 'pr', 'pe', 'ws', 'are', 'wweew', 'esa', 'as', 'we', 'wwf', 'wewdwew', 'sdhs', 'alah', 'joo', 'bgst', 'pkonyaaa', 'hmmmm', 'plisss', 'uat', 'sorr', 'ditl', 'yudahlah', 'yeayyy', 'bhkan', 'apll', 'jir', 'abu', 'achmad', 'omnya', 'ir', 'iii', 'cii', 'bangsattt', 'kenapaaaaaa', 'yeay', 'mantaapp', 'ibuu', 'mommmmy', 'ean', 'thnk', 'bangeettttt', 'hadehhhhh', 'njing', 'hampi', 'ygdi', 'bhh', 'mbok', 'hihi', 'gmb', 'ksmi', 'ssw', 'kmk', 'terussssss', 'vibri', 'remadhani', 'yakkk', 'dongok', 'drtd', 'terooooo', 'buuangeet', 'hutuo', 'limboto', 'snk', 'badahal', 'kalao', 'laah', 'duu', 'sekaraanng', 'seksrang', 'ihiiyy', 'rekom', 'goblox', 'hehhh', 'noh', 'capekanjiir', 'tutpny', 'trisna', 'tansa', 'thankyoy', 'eze', 'ddk', 'oo', 'thp', 'tersu', 'meuli', 'kamari', 'udud', 'ari', 'ayeuna', 'teu', 'sb', 'mlvvkmlv', 'mnpvlvp', 'mmmm', 'mpmp', 'lvn', 'vb', 'mvpvlmmkijj', 'nvymp', 'kin', 'vn', 'sedg', 'ses', 'sllau', 'mintibs', 'mboh', 'rifando', 'agustinus', 'eyyehhe', 'vangke', 'panindo', 'hba', 'iasa', 'anjenggggg', 'rko', 'pepk', 'saiia', 'jrngn', 'saepudln', 'kntl', 'otpt', 'hiegis', 'hiegis', 'ina', 'pasowr', 'gimanaaaa', 'mudaj', 'trhdp', 'siput', 'duli', 'akhm', 'dekey', 'spb', 'mhhak', 'jamal', 'hhhhhhh', 'ampunnnnnn', 'hoi', 'tetntang', 'pdjl', 'haduhhh', 'qlo', 'tj', 'uwahahaha', 'ta', 'gesss', 'huffff', 'qqery', 'benereeer', 'untyk', 'mantaapp', 'mahhh', 'slebeww', 'mahhh', 'dch', 'dahhh', 'tksh', 'xixixi', 'ahaha', 'mantaabb', 'loohh', 'elama', 'kkikklik', 'setiao', 'hitu', 'tadaaaaa', 'soalnyaaa', 'arghhhhh', 'hheee', 'yeayy', 'guys', 'hheehe', 'khann', 'xixi', 'sooo', 'hahahahah', 'thanksss', 'terkma', 'xixix', 'dab', 'yey', 'wlopun', 'bey', 'yeei', 'tis', 'gays', 'hayyyy', 'gokilllll', 'polllll', 'yeey', 'loohhh', 'wlw', 'susanto', 'uwuu', 'duwa', 'ey', 'uhuyy', 'hellow', 'daebakkk', 'ahahahahah', 'terlqu', 'banyq', 'sue', 'semog', 'ilon', 'jffin', 'guufgh', 'mugbutfyuf', 'fghyghb', 'rudi', 'leaflet', 'qw', 'nanyeaaaaa', 'teruuuuuuuus', 'akuuuu', 'yahud', 'toolol', 'agtdmdz', 'lhoe', 'njiing', 'kintil', 'geab', 'diman', 'jangkauannya', 'cokk', 'rebesss', 'bangeeeeeet', 'cabca', 'bm', 'aq sada', 'karn', 'ginimah', 'puooollll', 'xnya', 'suelah', 'jafi', 'hya', 'kmdn', 'kmvret', 'seing', 'woiiiiiiiiiiiiii', 'cusssaa', 'manthulllllll', 'msan',

```
'gituuuu', 'aajja', 'apsih', 'ith', 'mueah', 'ndg', 'buuuufwy', 'hmmm', 'nbbbbgeuiyy', 'ychtsuw',
'tttydz', 'urr', 'ru', 'yeihuu', 'agpxgww', 'atd', 'vktqau', 'ao', 'gobloggggggggggggg', 'oon', 'mng',
'assurur', 'wlpn', 'xxdxaa', 'sihk', 'sakasi', 'ume', 'ytta', 'wkw', 'ituloh', 'baaaay', 'hhmmm',
'redefdexwe', 'haggkk', 'hei', 'mbknya', 'hdh', 'hhehe', 'srapat', 'gess', 'bambang', 'gasssspoll',
'memmem', 'ua', 'iqbal', 'nsp', 'aulia', 'rosiyani', 'eee', 'rahel', 'nja', 'cikakak', 'bngsd', 'kwe',
'huhh', 'laaahh', 'riw', 'taii', 'lu', 'oii', 'inv', 'fry', 'agyklxb', 'refreal', 'purw', 'jatiseeng', 'berlkasi']
```

```
#Merge Stopword
```

```
datax = stop_factory.get_stop_words()+more_stopword
```

```
dictionary = ArrayDictionary(datax)
```

```
str = StopWordRemover(dictionary)
```

```
def stopw (txt_tokenized):
```

```
    stop = [word for word in txt_tokenized if word not in datax]
```

```
    return stop
```

```
df['Review_stopword'] = df['Review_token'].apply(lambda x: stopw (x))
```

```
#normalization
```

```
normalizad_word = pd.read_excel("normalisasi.xlsx")
```

```
normalizad_word_dict = {}
```

```
for index, row in normalizad_word.iterrows():
```

```
    if row[0] not in normalizad_word_dict:
```

```
        normalizad_word_dict[row[0]] = row[1]
```

```
def normalized_term(document):
```

```
    return [normalizad_word_dict[term] if term in normalizad_word_dict else term for term in
document]
```

```
df['Review_normalized'] = df['Review_stopword'].apply(normalized_term)
```

```
conda install -c conda-forge swifter
#Stemming
#import sastrawi package
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import swifter

#create stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

#stemmed
def stemmed_wrapper(term):
    return stemmer.stem(term)

term_dict = {}

for document in df['Review_normalized']:
    for term in document:
        if term not in term_dict:
            term_dict[term] = ""

print(len(term_dict))
print("-----")

for term in term_dict:
    term_dict[term] = stemmed_wrapper(term)
    print(term,":", term_dict[term])

print(term_dict)
print("-----")
```



```
#apply stemmed term to dataframe
def get_stemmed_term(document):
    return [term_dict[term] for term in document]

df['Review_stemmed'] = df['Review_normalized'].swifter.apply(get_stemmed_term)
print(df['Review_stemmed'])

#mengembalikan data
def gabung(txt):
    data_gabung = ''.join([c for c in txt if c not in string.punctuation])
    return data_gabung

pd.options.display.max_colwidth = None
df['data_join'] = df['Review_stemmed'].apply(lambda x: gabung(x))

df.to_csv('data_cleaned.csv', index=False)
```

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
def calc_TF(document):
    # Counts the number of times the word appears in review
    TF_dict = {}
    for term in document:
        if term in TF_dict:
            TF_dict[term] += 1
        else:
            TF_dict[term] = 1
    # Computes tf for each word
    for term in TF_dict:
        TF_dict[term] = TF_dict[term] / len(document)
    return TF_dict
```

```

df["TF_dict"] = df['Review_stemmed'].apply(calc_TF)

# Check TF result
index = 16286

print('%20s' % "term", "\t", "TF\n")
for key in df["TF_dict"][index]:
    print('%20s' % key, "\t", df["TF_dict"][index][key])

#IDF
def calc_DF(tfDict):
    count_DF = {}
    # Run through each document's tf dictionary and increment countDict's (term, doc) pair
    for document in tfDict:
        for term in document:
            if term in count_DF:
                count_DF[term] += 1
            else:
                count_DF[term] = 1
    return count_DF

DF = calc_DF(df["TF_dict"])

n_document = len(df)

def calc_IDF(__n_document, __DF):
    IDF_Dict = {}
    for term in __DF:
        IDF_Dict[term] = np.log(__n_document / (__DF[term] + 1))
    return IDF_Dict

#Stores the idf dictionary

```

```

IDF = calc_IDF(n_document, DF)

#calc TF-IDF
def calc_TF_IDF(TF):
    TF_IDF_Dict = {}
    #For each word in the review, we multiply its tf and its idf.
    for key in TF:
        TF_IDF_Dict[key] = TF[key] * IDF[key]
    return TF_IDF_Dict

#Stores the TF-IDF Series
df["TF-IDF_dict"] = df["TF_dict"].apply(calc_TF_IDF)

# Check TF-IDF result
index = 16286

print('%20s' % "term", "\t", '%10s' % "TF", "\t", '%20s' % "TF-IDF\n")
for key in df["TF-IDF_dict"][index]:
    print('%20s' % key, "\t", df["TF_dict"][index][key], "\t", df["TF-IDF_dict"][index][key])

# sort descending by value for DF dictionary
sorted_DF = sorted(DF.items(), key=lambda kv: kv[1], reverse=True)[:50]

# Create a list of unique words from sorted dictionary `sorted_DF`
unique_term = [item[0] for item in sorted_DF]

def calc_TF_IDF_Vec(__TF_IDF_Dict):
    TF_IDF_vector = [0.0] * len(unique_term)

    # For each unique word, if it is in the review, store its TF-IDF value.
    for i, term in enumerate(unique_term):
        if term in __TF_IDF_Dict:

```

```

        TF_IDF_vector[i] = __TF_IDF_Dict[term]
    return TF_IDF_vector

df["TF_IDF_Vec"] = df["TF-IDF_dict"].apply(calc_TF_IDF_Vec)

print("print first row matrix TF_IDF_Vec Series\n")
print(df["TF_IDF_Vec"][0])

print("\nmatrix size : ", len(df["TF_IDF_Vec"][0]))

# Convert Series to List
TF_IDF_Vec_List = np.array(df["TF_IDF_Vec"].to_list())

# Sum element vector in axis=0
sums = TF_IDF_Vec_List.sum(axis=0)

data = []

for col, term in enumerate(unique_term):
    data.append((term, sums[col]))

ranking = pd.DataFrame(data, columns=['term', 'rank'])
ranking.sort_values('rank', ascending=False)

```

```

from nltk.tokenize import word_tokenize
word_dict = {}
for i in range(0, len(df['data_join'])):
    sentence = df['data_join'][i]
    word_token = word_tokenize(sentence)
    for j in word_token:
        if j not in word_dict:
            word_dict[j] = 1

```

```

else:
    word_dict[j] += 1

import pandas as pd
lexicon = pd.read_csv('full_lexicon.csv')
lexicon
negasi = ['bukan','tidak','ga','gk']
lexicon_word = lexicon['word'].to_list()
lexicon_number_word = lexicon['number_of_words']

import pandas as pd
negasi = ['bukan','tidak','ga','gk']
lexicon = pd.read_csv('full_lexicon.csv')
lexicon = lexicon.drop(lexicon[(lexicon['word'] == 'bukan')
                               |(lexicon['word'] == 'tidak')
                               |(lexicon['word'] == 'ga')|(lexicon['word'] == 'gk') ].index,axis=0)
lexicon = lexicon.reset_index(drop=True)

lexicon_word = lexicon['word'].to_list()
lexicon_num_words = lexicon['number_of_words']

#Checking if there is words in dictionary that does not included in the lexicon

ns_words = []
factory = StemmerFactory()
stemmer = factory.create_stemmer()
for word in word_dict.keys():
    if word not in lexicon_word:
        kata_dasar = stemmer.stem(word)
        if kata_dasar not in lexicon_word:
            ns_words.append(word)
len(ns_words)

```

```

len({k:v for (k,v) in word_dict.items() if ((k in ns_words)&(v>3)) })
ns_words_list = {k:v for (k,v) in word_dict.items() if ((k in ns_words)&(v>3))}

#Ternyata kata-kata yang tidak termasuk dalam leksikon adalah kata-kata yang tidak
mempunyai makna sentimen
sort_orders = sorted(ns_words_list.items(), key=lambda x: x[1], reverse=True)
sort_orders=sort_orders[0:20]
for i in sort_orders:
    print(i[0], i[1])

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import swifter

sencol=[]
senrow =np.array([])
nsen = 0
factory = StemmerFactory()
stemmer = factory.create_stemmer()
sentiment_list = []
# function to write the word's sentiment if it is founded
def found_word(ind,words,word,sen,sencol,sentiment,add):
    # if it is already included in the bag of words matrix, then just increase the value
    if word in sencol:
        sen[sencol.index(word)] += 1
    else:
        #if not, than add new word
        sencol.append(word)
        sen.append(1)
        add += 1
    #if there is a negation word before it, the sentiment would be the negation of it's sentiment
    if (words[ind-1] in negasi):
        sentiment += -lexicon['weight'][lexicon_word.index(word)]

```

```

else:
    sentiment += lexicon['weight'][lexicon_word.index(word)]

return sen,sencol,sentiment,add

# checking every words, if they are appear in the lexicon, and then calculate their sentiment
if they do
for i in range(len(df)):
    nsen = senrow.shape[0]
    words = word_tokenize(df['data_join'][i])
    sentiment = 0
    add = 0
    prev = [0 for ii in range(len(words))]
    n_words = len(words)
    if len(sencol)>0:
        sen =[0 for j in range(len(sencol))]
    else:
        sen =[]

    for word in words:
        ind = words.index(word)
        # check whether they are included in the lexicon
        if word in lexicon_word :
            sen,sencol,sentiment,add= found_word(ind,words,word,sen,sencol,sentiment,add)
        else:
            # if not, then check the root word
            kata_dasar = stemmer.stem(word)
            if kata_dasar in lexicon_word:
                sen,sencol,sentiment,add=
found_word(ind,words,kata_dasar,sen,sencol,sentiment,add)
            # if still negative, try to match the combination of words with the adjacent words
            elif(n_words>1):

```

```

    if ind-1>-1:
        back_1 = words[ind-1]+' '+word
        if (back_1 in lexicon_word):
            sen,sencol,sentiment,add=
found_word(ind,words,back_1,sen,sencol,sentiment,add)
        elif(ind-2>-1):
            back_2 = words[ind-2]+' '+back_1
            if back_2 in lexicon_word:
                sen,sencol,sentiment,add=
found_word(ind,words,back_2,sen,sencol,sentiment,add)
    # if there is new word founded, then expand the matrix
    if add>0:
        if i>0:
            if (nsen==0):
                senrow = np.zeros([i,add],dtype=int)
            elif(i!=nsen):
                padding_h = np.zeros([nsen,add],dtype=int)
                senrow = np.hstack((senrow,padding_h))
                padding_v = np.zeros([(i-nsen),senrow.shape[1]],dtype=int)
                senrow = np.vstack((senrow,padding_v))
            else:
                padding =np.zeros([nsen,add],dtype=int)
                senrow = np.hstack((senrow,padding))
                senrow = np.vstack((senrow,sen))
        if i==0:
            senrow = np.array(sen).reshape(1,len(sen))
    # if there isn't then just update the old matrix
    elif(nsen>0):
        senrow = np.vstack((senrow,sen))

    sentiment_list.append(sentiment)

```



```

sencol.append('sentiment')
sentiment_array = np.array(sentiment_list).reshape(senrow.shape[0],1)
sentiment_data = np.hstack((senrow,sentiment_array))
df_sen = pd.DataFrame(sentiment_data,columns = sencol)
df_sen.head(10)

df_sen.to_csv("bobot.csv")

```

```

cek_df = pd.DataFrame([])
cek_df['text'] = df['data_join'].copy()
cek_df['sentiment'] = df_sen['sentiment'].copy()
cek_df

# function to analyze the reviews
def analysis(score):
    if score < 0:
        return 'Negative'
    elif score == 0:
        return 'Neutral'
    else:
        return 'Positive'
cek_df['analysis'] = cek_df['sentiment'].apply(analysis)

#Results from determine sentiment score
print(cek_df['analysis'].value_counts())
cek_df = cek_df.astype({'analysis' : 'category'})
cek_df = cek_df.astype({'text' : 'string'})
cek_df.dtypes
cek_df.to_csv('sentiment_analysis.csv', index=False)

datanew = pd.read_csv('sentiment_analysis1.csv', delimiter = ';', encoding = 'utf-8')
datanew

```

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer

tf = TfidfVectorizer()
text_tf = tf.fit_transform(datanew['text'].astype('U'))
print(text_tf)

#splitting data
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(text_tf, datanew['analysis'], test_size=0.4,
random_state= 1)

print('Number transactions X_train dataset: ', X_train.shape)
print('Number transactions y_train dataset: ', y_train.shape)
print('Number transactions X_test dataset: ', X_test.shape)
print('Number transactions y_test dataset: ', y_test.shape)

#Klasifikasi menggunakan Algoritma Naive Bayes
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
NB = MultinomialNB().fit(X_train, y_train)
predicted = NB.predict(X_test)
print("MultinomialNB accuracy:" , accuracy_score(y_test,predicted))
print("MultinomialNB Precision:" , precision_score(y_test,predicted, average="binary",
pos_label="Negative"))
print("MultinomialNB Recall:" , recall_score(y_test,predicted, average="binary",
pos_label="Negative"))
print("MultinomialNB f1_score:" , f1_score(y_test,predicted, average="binary",
pos_label="Negative"))
```

```

print(f'confusion matrix:\n{confusion_matrix(y_test, predicted)}')
print('=====\n')
print(classification_report(y_test, predicted, zero_division=0))

#Klasifikasi menggunakan Algoritma SVM
from sklearn import svm
SVM_linear = svm.SVC().fit(X_train, y_train)
predicted = SVM_linear.predict(X_test)
print("SVM accuracy:" , accuracy_score(y_test,predicted))
print("SVM Precision:" , precision_score(y_test,predicted, average="binary",
pos_label="Negative"))
print("SVM Recall:" , recall_score(y_test,predicted, average="binary",
pos_label="Negative"))
print("SVM f1_score:" , f1_score(y_test,predicted, average="binary",
pos_label="Negative"))

print(f'confusion matrix:\n{confusion_matrix(y_test, predicted)}')
print('=====\n')
print(classification_report(y_test, predicted, zero_division=0))

#Klasifikasi menggunakan Algoritma SVM
from sklearn import svm
SVM_kernel_linear = svm.SVC(kernel='linear').fit(X_train, y_train)
predicted = SVM_kernel_linear.predict(X_test)
print("SVM accuracy:" , accuracy_score(y_test,predicted))
print("SVM Precision:" , precision_score(y_test,predicted, average="binary",
pos_label="Negative"))
print("SVM Recall:" , recall_score(y_test,predicted, average="binary",
pos_label="Negative"))
print("SVM f1_score:" , f1_score(y_test,predicted, average="binary",
pos_label="Negative"))

```

```

print(f'confusion matrix:\n{confusion_matrix(y_test, predicted)}')
print('=====\n')
print(classification_report(y_test, predicted, zero_division=0))

#Klasifikasi menggunakan Algoritma SVM
from sklearn import svm
SVM_kernel_poly = svm.SVC(kernel='poly').fit(X_train, y_train)
predicted = SVM_kernel_poly.predict(X_test)
print("SVM accuracy:" , accuracy_score(y_test,predicted))
print("SVM Precision:" , precision_score(y_test,predicted, average="binary",
pos_label="Negative"))
print("SVM Recall:" , recall_score(y_test,predicted, average="binary",
pos_label="Negative"))
print("SVM f1_score:" , f1_score(y_test,predicted, average="binary",
pos_label="Negative"))

print(f'confusion matrix:\n{confusion_matrix(y_test, predicted)}')
print('=====\n')
print(classification_report(y_test, predicted, zero_division=0))

#Klasifikasi menggunakan Algoritma SVM
from sklearn import svm
SVM_kernel_sigmoid = svm.SVC(kernel='sigmoid').fit(X_train, y_train)
predicted = SVM_kernel_sigmoid.predict(X_test)
print("SVM accuracy:" , accuracy_score(y_test,predicted))
print("SVM Precision:" , precision_score(y_test,predicted, average="binary",
pos_label="Negative"))
print("SVM Recall:" , recall_score(y_test,predicted, average="binary",
pos_label="Negative"))
print("SVM f1_score:" , f1_score(y_test,predicted, average="binary",
pos_label="Negative"))

```

```

print(f'confusion matrix:\n{confusion_matrix(y_test, predicted)}')
print('=====\\n')
print(classification_report(y_test, predicted, zero_division=0))

#Klasifikasi menggunakan Algoritma SVM
from sklearn import svm
SVM_kernel_rbf = svm.SVC(kernel='rbf').fit(X_train, y_train)
predicted = SVM_kernel_rbf.predict(X_test)
print("SVM accuracy:" , accuracy_score(y_test,predicted))
print("SVM Precision:" , precision_score(y_test,predicted, average="binary",
pos_label="Negative"))
print("SVM Recall:" , recall_score(y_test,predicted, average="binary",
pos_label="Negative"))
print("SVM f1_score:" , f1_score(y_test,predicted, average="binary",
pos_label="Negative"))

print(f'confusion matrix:\n{confusion_matrix(y_test, predicted)}')
print('=====\\n')
print(classification_report(y_test, predicted, zero_division=0))

#Klasifikasi menggunakan Algoritma KNN
from sklearn.neighbors import KNeighborsClassifier

KNN = KNeighborsClassifier(n_neighbors=3, weights= 'distance').fit(X_train, y_train)
predicted = KNN.predict(X_test)
print("KNN accuracy:" , accuracy_score(y_test,predicted))
print("KNN Precision:" , precision_score(y_test,predicted, average="binary",
pos_label="Negative"))
print("KNN Recall:" , recall_score(y_test,predicted, average="binary",
pos_label="Negative"))

```

```

print("KNN    fl_score:"      ,      fl_score(y_test,predicted,      average="binary",
pos_label="Negative"))

print(f'confusion matrix:\n{confusion_matrix(y_test, predicted)}')
print('=====\\n')
print(classification_report(y_test, predicted, zero_division=0))

#Klasifikasi menggunakan Algoritma Decision Tree
from sklearn import tree

DT = tree.DecisionTreeClassifier().fit(X_train, y_train)
predicted = DT.predict(X_test)
print("Decision Tree accuracy:" , accuracy_score(y_test,predicted))
print("Decision Tree Precision:" , precision_score(y_test,predicted, average="binary",
pos_label="Negative"))
print("Decision Tree Recall:" , recall_score(y_test,predicted, average="binary",
pos_label="Negative"))
print("Decision Tree fl_score:" , fl_score(y_test,predicted, average="binary",
pos_label="Negative"))

print(f'confusion matrix:\n{confusion_matrix(y_test, predicted)}')
print('=====\\n')
print(classification_report(y_test, predicted, zero_division=0))

#Bagian klasifikasi untuk pelabelan
import pickle
import collections, numpy
Data = datanew['text']
input_transformed = tf.transform(datanew['text'])
SVM_kernel_linear_prediction = SVM_kernel_linear.predict(input_transformed)
SVM_kernel_poly_prediction = SVM_kernel_poly.predict(input_transformed)
SVM_kernel_sigmoid_prediction = SVM_kernel_sigmoid.predict(input_transformed)

```

```

SVM_kernel_rbf_prediction = SVM_kernel_rbf.predict(input_transformed)
SVM_kernel_linear_prediction
Result = pd.DataFrame({"text awal" : datanew['analysis'], "label" :
SVM_kernel_linear_prediction})
Result
Result.to_csv("Hasil_SVM.csv", index = False)
df1 = pd.read_csv('Hasil_SVM.csv')
df1
Result1 = pd.DataFrame({"text awal" : datanew['text'], "label" :
SVM_kernel_linear_prediction})
Result1
Result1.to_csv("Hasil_SVM1.csv", index = False)

```

```

import nltk
nltk.download('punkt')
import matplotlib as plt
from nltk.tokenize import word_tokenize
from nltk.probability import FreqDist
visual = pd.DataFrame([])
visual['text'] = df1['label'].values.copy()
visual

word_dict = {}
for i in range(0, len(visual['text'])):
    sentence = visual['text'][i]
    word_token = word_tokenize(sentence)
    for j in word_token:
        if j not in word_dict:
            word_dict[j] = 1
        else:
            word_dict[j] += 1
Result1_top = pd.DataFrame((word_dict.items()))

```

```

Result1_top = Result1_top.rename(columns={0:'word', 1:'counts'})
Result1_top = Result1_top.sort_values(by='counts', ascending=False)
Result1_top.head(10)

import matplotlib.pyplot as plt
Result1_top.head(10).sort_values(by='counts',ascending=True).plot.barh(x='word',
y='counts', rot=0,figsize=(20,10),fontsize=30)
plt.savefig('grafik .png',bbox_inches='tight')

neg = Result1.loc[Result1['label'] == 'Negative']
pos = Result1.loc[Result1['label'] == 'Positive']
neg.to_csv('ulasan negatif.csv')
pos.to_csv('ulasan positif.csv')
Positif=pos.to_csv('ulasan positif.csv')
Negatif = pd.read_csv('ulasan negatif.csv')
positif = pos.reset_index(drop=True)
negatif = neg.reset_index(drop=True)

#visualisasi negatif
neg_dict = {}
for i in range(0, len(negatif['text awal'])):
    sentence = negatif['text awal'][i]
    tokenisasi = word_tokenize(sentence)
    for c in tokenisasi:
        if c not in neg_dict:
            neg_dict[c] = 1
        else:
            neg_dict[c] += 1
negatif_top = pd.DataFrame((neg_dict.items()))
negatif_top = negatif_top.rename(columns={0:'word', 1:'counts'})
negatif_top = negatif_top.sort_values(by='counts', ascending=False)
negatif_top

```



```
negatif_top.head(10).sort_values(by='counts',ascending=True).plot.barh(x='word',
y='counts', rot=0,figsize=(20,10),fontsize=30)
plt.savefig('grafik negatif .png',bbox_inches='tight')

#visualisasi positif
pos_dict = {}
for i in range(0, len(positif['text awal'])):
    sentence = positif['text awal'][i]
    token = word_tokenize(sentence)
    for b in token:
        if b not in pos_dict:
            pos_dict[b] = 1
        else:
            pos_dict[b] += 1
positif_top = pd.DataFrame((pos_dict.items()))
positif_top = positif_top.rename(columns={0:'word', 1:'counts'})
positif_top = positif_top.sort_values(by='counts', ascending=False)
positif_top.head(10)
positif_top.head(10).sort_values(by='counts',ascending=True).plot.barh(x='word',
y='counts', rot=0,figsize=(20,10),fontsize=30)
plt.savefig('grafik positif .png',bbox_inches='tight')

#Asosiasi
pip install mlxtend
import pandas as pd
import numpy as np
import seaborn as sns
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

```
data = pd.read_csv('negatif_rev.csv', delimiter = ';', encoding = 'utf-8')
data = data.join(data['text awal'].str.split(' ', expand=True).add_prefix(""))
print(data[['2']])
del data['text awal']
items = (data['0'].unique())
items

#OneHotEncoding

itemset = set(items)
encoded_vals = []
for index, row in data.iterrows():
    rowset = set(row)
    labels = {}
    uncommons = list(itemset - rowset)
    commons = list(itemset.intersection(rowset))
    for uc in uncommons:
        labels[uc] = 0
    for com in commons:
        labels[com] = 1
    encoded_vals.append(labels)
encoded_vals[0]
ohe_df = pd.DataFrame(encoded_vals)

freq_items = apriori(ohe_df, min_support=0.01, use_colnames=True)
freq_items.head(10)

rules = association_rules(freq_items, metric="confidence", min_threshold=0.25)
rules.head(20)

rules = rules.sort_values(by='support', ascending=False)
```

rules.head(10)

rules = rules.sort\_values(by='lift', ascending=False)

rules

rules = rules.sort\_values(by='confidence', ascending=False)

rules

rules.to\_csv("Asosiasi.csv", index = False)

### A - 5 Hasil Uji Validitas dan Reliabilitas

Correlations																	
		X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	Total
X1	Pearson Correlation	1	.311*	.425**	.376**	.235	.308*	.710**	.046	-.089	.129	.225	-.077	.032	.029	.014	.509**
	Sig. (2-tailed)		.028	.002	.007	.101	.029	.000	.752	.538	.372	.116	.594	.826	.840	.923	.000
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X2	Pearson Correlation	.311*	1	.618**	-.046	.311*	.308*	.442**	.020	.123	.173	.138	-.039	-.110	-.044	.014	.451**
	Sig. (2-tailed)	.028		.000	.750	.028	.029	.001	.889	.394	.230	.338	.790	.448	.761	.923	.001
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X3	Pearson Correlation	.425**	.618**	1	.397**	.289*	.531**	.418**	.341*	.210	.366**	.285*	.273	.092	.190	.215	.746**
	Sig. (2-tailed)	.002	.000		.004	.042	.000	.003	.015	.143	.009	.045	.055	.527	.186	.133	.000
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X4	Pearson Correlation	.376**	.046	.397**	1	.124	.349*	.270	.250	.166	.266	.524**	.218	.388**	.302*	.329*	.640**
	Sig. (2-tailed)	.007	.750	.004		.389	.013	.058	.080	.250	.062	.000	.129	.005	.033	.020	.000
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X5	Pearson Correlation	.235	.311*	.289*	.124	1	.430**	.244	-.050	-.290*	.304*	-.070	.054	.007	.019	.204	.430**
	Sig. (2-tailed)	.101	.028	.042	.389		.002	.087	.731	.041	.032	.629	.709	.964	.893	.156	.002
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X6	Pearson Correlation	.308*	.308*	.531**	.349*	.430**	1	.414**	.022	-.020	.449**	.150	-.014	.143	.178	.123	.614**
	Sig. (2-tailed)	.029	.029	.000	.013	.002		.003	.879	.890	.001	.298	.923	.320	.216	.393	.000
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X7	Pearson Correlation	.710**	.442**	.418**	.270	.244	.414**	1	-.096	-.018	.228	.177	-.006	.012	.111	.069	.560**
	Sig. (2-tailed)	.000	.001	.003	.058	.087	.003		.505	.903	.112	.219	.969	.937	.443	.635	.000
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X8	Pearson Correlation	.046	.020	.341*	.250	-.050	.022	-.096	1	.374**	.165	.298*	.365**	.120	.129	.114	.398**
	Sig. (2-tailed)	.752	.889	.015	.080	.731	.879	.505		.007	.251	.036	.009	.408	.373	.433	.004
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X9	Pearson Correlation	-.089	.123	.210	.166	-.290*	-.020	-.018	.374**	1	-.074	.417**	.430**	.283*	.226	.140	.303*
	Sig. (2-tailed)	.538	.394	.143	.250	.041	.890	.903	.007		.609	.003	.002	.046	.114	.331	.033
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X10	Pearson Correlation	.129	.173	.366**	-.266	.304*	.449**	.228	-.165	-.074	1	.203	-.036	.127	.224	.218	.556**
	Sig. (2-tailed)	.372	.230	.009	.062	.032	.001	.112	.251	.609		.157	.804	.380	.117	.129	.000
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X11	Pearson Correlation	.225	.138	.285**	.524**	-.070	.150	.177	.298*	.417**	.203	1	.346*	.487**	.488**	.355*	.601**
	Sig. (2-tailed)	.116	.338	.045	.000	.629	.298	.219	.036	.003	.157		.014	.000	.000	.011	.000
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X12	Pearson Correlation	-.077	-.039	.273	.218	.054	-.014	-.006	.365**	.430**	-.036	.346*	1	.150	.138	.273	.351*
	Sig. (2-tailed)	.594	.790	.055	.129	.709	.923	.969	.009	.002	.804	.014		.298	.338	.055	.013
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X13	Pearson Correlation	.032	-.110	.092	.388**	.007	.143	.012	.120	.283*	.127	.487**	.150	1	.862**	.345*	.464**
	Sig. (2-tailed)	.826	.448	.527	.005	.964	.320	.937	.408	.046	.380	.000	.298		.000	.014	.001
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X14	Pearson Correlation	.029	-.044	.190	.302*	.019	.178	.111	.129	.226	.224	.488**	.138	.862**	1	.298*	.503**
	Sig. (2-tailed)	.840	.761	.186	.033	.893	.216	.443	.373	.114	.117	.000	.338	.000		.036	.000
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X15	Pearson Correlation	.014	.014	.215	.329*	.204	.123	.069	.114	.140	.218	.355*	.273	.345*	.298*	1	.462**
	Sig. (2-tailed)	.923	.923	.133	.020	.156	.393	.635	.433	.331	.129	.011	.055	.014	.036		.001
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
Total	Pearson Correlation	.509**	.451**	.746**	.640**	.430**	.614**	.560**	.398**	.303*	.556**	.601**	.351*	.464**	.503**	.462**	1
	Sig. (2-tailed)	.000	.001	.000	.000	.002	.000	.000	.004	.033	.000	.000	.013	.001	.000	.001	
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50

\*. Correlation is significant at the 0.05 level (2-tailed).  
 \*\*. Correlation is significant at the 0.01 level (2-tailed).

### Lampiran A-5. 1 Hasil Uji Validitas Kuesioner Kano Fungsional

### Reliability Statistics

Cronbach's Alpha	N of Items
.781	15

### Lampiran A-5. 2 Hasil Uji Reliabilitas Kuesioner Kano Fungsional

		Correlations															Total
		X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	
X1	Pearson Correlation	1	.311*	.425**	.376**	.235	.308*	.710**	.046	-.089	.129	.225	-.077	.032	.029	.014	.509**
	Sig. (2-tailed)		.028	.002	.007	.101	.029	.000	.752	.538	.372	.116	.594	.826	.840	.923	.000
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X2	Pearson Correlation	.311*	1	.618**	.046	.311*	.308*	.442**	.020	.123	.173	.138	-.039	-.110	-.044	.014	.451**
	Sig. (2-tailed)	.028		.000	.750	.028	.029	.001	.889	.394	.230	.338	.790	.448	.761	.923	.001
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X3	Pearson Correlation	.425**	.618**	1	.397**	.289*	.531**	.418**	.341*	.210	.366**	.285*	.273	.092	.190	.215	.746**
	Sig. (2-tailed)	.002	.000		.004	.042	.000	.003	.015	.143	.009	.045	.055	.527	.186	.133	.000
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X4	Pearson Correlation	.376**	.046	.397**	1	.124	.349*	.270	.250	.166	.266	.524**	.218	.388**	.302*	.329*	.640**
	Sig. (2-tailed)	.007	.750	.004		.389	.013	.058	.080	.250	.062	.000	.129	.005	.033	.020	.000
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X5	Pearson Correlation	.235	.311*	.289*	.124	1	.430**	.244	-.050	-.290*	.304*	-.070	.054	.007	.019	.204	.430**
	Sig. (2-tailed)	.101	.028	.042	.389		.002	.087	.731	.041	.032	.629	.709	.964	.893	.156	.002
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X6	Pearson Correlation	.308*	.308*	.531**	.349*	.430**	1	.414**	.022	-.020	.449**	.150	-.014	.143	.178	.123	.614**
	Sig. (2-tailed)	.029	.029	.000	.013	.002		.003	.879	.890	.001	.298	.923	.320	.216	.393	.000
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X7	Pearson Correlation	.710**	.442**	.418**	.270	.244	.414**	1	-.096	-.018	.228	.177	-.006	.012	.111	.069	.560**
	Sig. (2-tailed)	.000	.001	.003	.058	.087	.003		.505	.903	.112	.219	.969	.937	.443	.635	.000
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X8	Pearson Correlation	.046	.020	.341*	.250	-.050	.022	-.096	1	.374**	.165	.298*	.365**	.120	.129	.114	.398**
	Sig. (2-tailed)	.752	.889	.015	.080	.731	.879	.505		.007	.251	.036	.009	.408	.373	.433	.004
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X9	Pearson Correlation	-.089	.123	.210	.166	-.290*	-.020	-.018	.374**	1	-.074	.417**	.430**	.283*	.226	.140	.303*
	Sig. (2-tailed)	.538	.394	.143	.250	.041	.890	.903	.007		.609	.003	.002	.046	.114	.331	.033
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X10	Pearson Correlation	.129	.173	.366**	.266	.304*	.449**	.228	.165	-.074	1	.203	-.036	.127	.224	.218	.556**
	Sig. (2-tailed)	.372	.230	.009	.062	.032	.001	.112	.251	.609		.157	.804	.380	.117	.129	.000
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X11	Pearson Correlation	.225	.138	.285*	.524**	-.070	.150	.177	.298*	.417**	.203	1	.346*	.487**	.488**	.355*	.601**
	Sig. (2-tailed)	.116	.338	.045	.000	.629	.298	.219	.036	.003	.157		.014	.000	.000	.011	.000
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X12	Pearson Correlation	-.077	-.039	.273	.218	.054	-.014	-.006	.365**	.430**	-.036	.346*	1	.150	.138	.273	.351*
	Sig. (2-tailed)	.594	.790	.055	.129	.709	.923	.969	.009	.002	.804	.014		.298	.338	.055	.013
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X13	Pearson Correlation	.032	-.110	.092	.388**	.007	.143	.012	.120	.283*	.127	.487**	.150	1	.862**	.345*	.464**
	Sig. (2-tailed)	.826	.448	.527	.005	.964	.320	.937	.408	.046	.380	.000	.298		.000	.014	.001
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X14	Pearson Correlation	.029	-.044	.190	.302*	.019	.178	.111	.129	.226	.224	.488**	.138	.862**	1	.298*	.503**
	Sig. (2-tailed)	.840	.761	.186	.033	.893	.216	.443	.373	.114	.117	.000	.338	.000		.036	.000
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
X15	Pearson Correlation	.014	.014	.215	.329*	.204	.123	.069	.114	.140	.218	.355*	.273	.345*	.298*	1	.462**
	Sig. (2-tailed)	.923	.923	.133	.020	.156	.393	.635	.433	.331	.129	.011	.055	.014	.036		.001
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
Total	Pearson Correlation	.509**	.451**	.746**	.640**	.430**	.614**	.560**	.398**	.303*	.556**	.601**	.351*	.464**	.503**	.462**	1
	Sig. (2-tailed)	.000	.001	.000	.000	.002	.000	.000	.004	.033	.000	.000	.013	.001	.000	.001	
	N	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50

\*. Correlation is significant at the 0.05 level (2-tailed).

\*\* . Correlation is significant at the 0.01 level (2-tailed).

### Lampiran A-5. 3 Hasil Uji Validitas Kuesioner Kano Disfungsional

### Reliability Statistics

Cronbach's Alpha	N of Items
.875	15

Lampiran A-5. 4 Hasil Uji Reliabilitas Kuesioner Kano Disfungsional

### A - 6 Penyebaran Kuesioner

The image shows a social media post on the left and a Google Forms survey link on the right. The social media post is a purple-to-blue gradient box containing text in Indonesian. The text asks for help with a research task and offers a reward. The Google Forms link is a blue box with a white background, also containing text in Indonesian and a link to the survey. Below the social media post is a grey comment box. To the right of the social media post is a screenshot of a mobile phone displaying the Google Forms survey link.

Halo kak, perkenalkan saya Miranda, mahasiswi Teknik Industri UII. Saat ini saya lagi melakukan penelitian untuk tugas akhir dengan objek Alfagift. Sebelumnya saya pernah liat komen kakak tentang Alfagift. Oleh karena itu, saya mohon kesedian kakak untuk mengisi kuesioner penelitian tugas akhir saya dengan link di bawah ini: <https://forms.gle/3ytvaNx7kVTtP8YW8>

Sebagai ucapan terima kasih atas kesediaanya mengisi kuesioner ini, peneliti menyediakan reward berupa gopay/ pulsa/ ovo/ dll bagi 5 responden terpilih. Atas kesediannya berpartisipasi dalam penelitian ini, saya ucapkan terima kasih 🙏

Yesterday 9:53 PM

Done ya kaaa, sukses tugas akhirnya

Halo kak, perkenalkan saya Miranda, mahasiswi Teknik Industri UII. Saat ini saya lagi melakukan penelitian untuk tugas akhir dengan objek Alfagift. Sebelumnya saya pernah liat tweet kakak tentang Alfagift. Oleh karena itu, saya mohon kesedian kakak untuk mengisi kuesioner penelitian tugas akhir saya dengan link di bawah ini: [forms.gle/3ytvaNx7kVTtP8...](https://forms.gle/3ytvaNx7kVTtP8...)

Sebagai ucapan terima kasih atas kesediaanya mengisi kuesioner ini, peneliti menyediakan reward berupa gopay/ pulsa/ ovo/ dll bagi 5 responden terpilih. Atas kesediannya berpartisipasi dalam penelitian ini, saya ucapkan terima kasih 🙏

6:54 PM

docs.google.com/fo

Kuesioner Kano Penelitian Tugas Akhir Layanan Aplikasi Alfagift

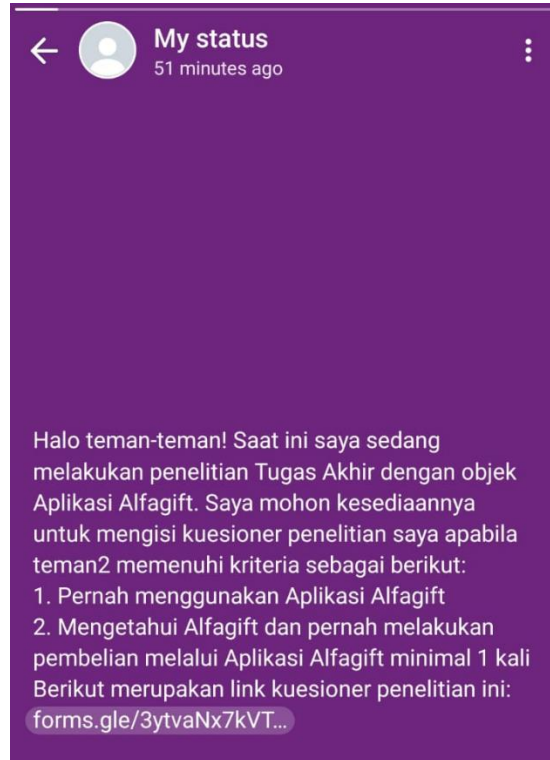
Your response has been recorded

[Submit another response](#)

This form was created inside of education. [Report Abuse](#)

Google Forms

Lampiran A-6. 1 Penyebaran Kuesioner melalui *Instagram* dan *Twitter*



Lampiran A-6. 2 Penyebaran Kuesioner melalui *Whatsapp*

## A - 7 Kuesioner Kano

### Bagian Fungsional (keberadaan suatu atribut)

Pada bagian ini responden diminta untuk memberi penilaian terhadap keberadaan suatu atribut dengan tujuan untuk memahami kebutuhan pelanggan terhadap atribut-atribut dari suatu produk atau layanan. Pilihlah salah satu jawaban yang Anda anggap sesuai.

Keterangan:

1 = Suka (Saya menyukai hal seperti itu)

2 = Mengharapkan (Saya mengharapkan hal seperti itu)

3 = Netral (Saya netral)

4 = Toleran (Saya tidak suka tapi saya dapat mentoleransi / menerima hal seperti itu)

5 = Tidak Suka (Saya tidak suka dan tidak dapat menerima hal seperti itu)

Aplikasi Alfagift cepat dibuka \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Aplikasi dirancang dengan bagus \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Aplikasi mudah digunakan \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Belanja menggunakan aplikasi menguntungkan \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Stok barang di aplikasi lengkap \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Layanan aplikasi memuaskan \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Aplikasi jarang terjadi eror \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka



Belanja menggunakan aplikasi bisa COD \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Poin masuk ke dalam aplikasi \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Pengantaran barang / pesanan cepat \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Pesanan dikirim ke lokasi pembeli \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Struk masuk (tersimpan) ke dalam aplikasi \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

---

Barang yang datang sesuai (benar) pesanan \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

---

Barang yang datang jumlahnya cukup \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

---

Kondisi barang yang datang memuaskan \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

## Lampiran A-7. 2 Kuesioner Kano Fungsional

**Bagian Disfungsional (tidak adanya suatu atribut)**

Pada bagian ini responden diminta untuk memberi penilaian terhadap tidak adanya suatu atribut

dengan tujuan untuk memahami apakah responden dapat menerima atau tidak dapat menerima jika atribut-atribut dari suatu produk atau layanan tidak tersedia. Pilihlah salah satu jawaban yang Anda anggap sesuai.

Keterangan:

1 = Suka (Saya menyukai hal seperti itu)

2 = Mengharapkan (Saya mengharapkan hal seperti itu)

3 = Netral (Saya netral)

4 = Toleran (Saya tidak suka tapi saya dapat mentoleransi / menerima hal seperti itu)

5 = Tidak Suka (Saya tidak suka dan tidak dapat menerima hal seperti itu)

Aplikasi Alfagift lambat (lama) dibuka \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Aplikasi dirancang dengan buruk \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Aplikasi susah digunakan \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Belanja menggunakan aplikasi merugikan \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Stok barang di aplikasi kosong (tidak lengkap) \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Layanan aplikasi mengecewakan \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Aplikasi sering terjadi eror \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Belanja menggunakan aplikasi tidak bisa COD \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Poin tidak masuk ke dalam aplikasi \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Pengantaran barang / pesanan lama \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Pesanan tidak dikirim ke lokasi pembeli \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Struk tidak masuk (tidak tersimpan) ke dalam aplikasi \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Barang yang datang salah (tidak sesuai) pesanan \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Barang yang datang jumlahnya kurang \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Kondisi barang yang datang mengecewakan \*

	1	2	3	4	5	
Suka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tidak Suka

Lampiran A-7. 3 Kuesioner Kano Disfungsional



Responde n	X 1	X 2	X 3	X 4	X 5	X 6	X 7	X 8	X 9	X1 0	X1 1	X1 2	X1 3	X1 4	X1 5
43	2	2	2	2	2	3	2	1	1	3	1	1	1	1	1
44	4	2	3	2	2	4	5	1	1	3	1	1	1	1	1
45	2	2	2	2	3	2	2	4	2	1	1	2	2	2	2
46	2	2	2	2	2	2	3	2	2	2	2	2	2	2	2
47	2	2	2	1	1	1	2	1	2	1	1	1	2	2	1
48	2	2	2	2	2	2	2	2	2	3	2	2	2	2	2
49	3	3	2	1	2	2	3	2	1	3	1	1	1	1	1
50	4	3	3	4	4	3	3	3	1	4	3	1	3	3	3

## Lampiran A-7. 5 Rekapitulasi Kuesioner Kano Disfungsional

Responde n	X 1	X 2	X 3	X 4	X 5	X 6	X 7	X 8	X 9	X1 0	X1 1	X1 2	X1 3	X1 4	X1 5
1	5	5	5	5	4	5	4	3	5	5	5	4	5	5	5
2	4	4	4	5	5	5	5	5	5	5	5	5	5	5	5
3	5	4	5	4	4	5	4	3	4	5	5	3	5	4	5
4	4	4	4	5	4	4	5	4	4	2	4	4	4	5	4
5	2	5	5	5	5	5	5	5	5	5	5	5	5	5	5
6	5	5	5	5	4	4	5	5	5	2	5	5	5	5	5
7	4	5	5	5	3	5	4	5	5	5	5	5	3	4	5
8	5	5	4	5	3	5	4	3	5	4	5	5	5	5	5
9	3	5	5	5	3	5	4	5	5	5	5	5	5	5	5
10	4	4	4	5	3	4	4	5	5	4	5	5	5	5	5
11	3	4	4	5	3	5	4	4	4	4	5	4	5	5	4
12	3	3	3	4	3	2	3	4	4	2	4	3	3	4	3
13	3	3	3	4	2	3	3	3	4	2	5	5	5	5	5
14	5	5	5	5	4	5	4	5	5	2	5	5	5	5	5
15	5	5	4	5	5	5	5	5	5	5	5	5	5	5	5
16	5	5	5	5	5	5	5	3	5	5	5	5	5	5	1
17	4	4	5	4	3	4	5	3	5	3	5	5	4	5	4
18	4	5	5	5	3	5	4	5	5	4	5	3	5	5	5
19	5	5	5	4	4	4	4	5	5	5	5	5	5	5	5
20	2	3	4	4	3	3	2	4	3	2	3	3	4	4	4
21	4	4	4	5	3	5	4	5	5	4	5	5	5	5	5
22	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
23	2	4	4	5	5	5	3	5	5	5	5	5	5	5	5
24	4	4	4	4	1	4	4	3	5	3	5	4	5	5	5
25	5	5	5	5	5	5	5	5	5	5	5	3	5	5	5
26	1	2	4	5	3	3	1	5	5	3	5	5	5	5	5
27	5	5	5	5	5	5	5	3	5	5	5	4	5	5	5
28	3	3	3	3	3	3	4	3	5	5	4	5	5	5	5
29	4	1	2	2	4	5	4	5	4	4	5	5	4	4	4
30	3	2	5	5	5	5	5	5	4	3	3	3	3	3	3



Responde n	X 1	X 2	X 3	X 4	X 5	X 6	X 7	X 8	X 9	X1 0	X1 1	X1 2	X1 3	X1 4	X1 5
31	3	3	3	4	4	4	4	4	4	4	5	4	4	4	4
32	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
33	5	5	5	5	4	4	5	3	5	5	5	5	5	5	5
34	4	5	5	5	4	4	5	5	4	4	5	5	5	5	5
35	5	5	5	3	3	5	5	3	5	3	5	5	5	5	5
36	3	3	3	4	4	4	4	4	4	4	4	4	4	4	4
37	5	5	5	3	5	5	5	3	3	5	5	3	5	5	5
38	4	4	4	5	5	4	4	2	4	4	5	5	5	5	5
39	5	5	5	5	4	5	5	5	5	5	5	5	5	5	5
40	3	5	4	5	3	4	3	2	3	4	4	3	5	4	5
41	4	4	5	5	4	5	5	5	5	5	5	5	4	4	4
42	3	4	5	5	3	5	4	5	5	5	5	5	5	5	5
43	5	5	5	2	5	5	4	5	5	3	5	5	5	5	5
44	1	3	1	5	3	3	1	5	5	3	5	5	5	5	5
45	4	5	5	5	5	5	5	4	5	5	5	5	5	5	5
46	3	3	3	4	3	4	3	4	5	3	4	4	4	4	3
47	2	3	3	3	3	4	4	4	3	3	3	2	3	3	2
48	4	4	4	4	3	4	4	4	4	3	4	4	4	4	4
49	2	3	3	4	4	4	4	4	4	4	4	4	4	4	4
50	2	3	3	3	3	3	4	3	4	3	3	5	3	3	3

