

S Integrasi Kansei Engineering Dan Kano Berbasis Natural Language Processing (NLP) Untuk Menunjang Pengembangan Produk Jasa Pada Pariwisata Candi Borobudur

Tugas Akhir

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Strata 1

Pada Jurusan Teknik Industri Fakultas Teknologi Industri

Universitas Islam Indonesia



Disusun Oleh:

N a m a : Nayoko Prasetyo Jati

NIM : 17522119

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

2021

SURAT KETERANGAN PENELITIAN



FAKULTAS
TEKNOLOGI INDUSTRI

Gedung KH. Mas Mansur
Kampus Terpadu Universitas Islam Indonesia
J. Kalikurang km 14,5 Yogyakarta 55584
T. (0274) 898444 ext. 4110, 4100
F. (0274) 956007
E. fti@uii.ac.id
W. fti.uii.ac.id

SURAT KETERANGAN PENELITIAN

Nomor : 171/Ka.lab SIMANTI/20/ Lab.SIMANTI/XII/2021

Assalamu'alaikum Warohmatullahi Wabarokaatuh

Dengan hormat,

Yang bertanda tangan dibawah ini, menerangkan bahwa:

Nama : Nayoko Prasetyo Jati

Nim : 17522119

Jurusan : Teknik Industri

Dosen Pembimbing : Abdullah 'Azzam, S.T, M.T.

Menyatakan bahwa mahasiswa tersebut diatas telah melaksanakan penelitian tugas akhir dengan judul " **INTEGRASI KANSEI ENGINEERING DAN KANO BERBASIS NATURAL LANGUAGE PROCESSING (NLP) UNTUK MENUNJANG PENGEMBANGAN PRODUK JASA PADA PARIWISATA CANDI BOROBUDUR**" mulai pelaksanaan penelitian 05 Agustus 2021 sampai 07 Desember 2021

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

Wassalamu'alaikum Warohmatullahi Wabarokaatuh

Yogyakarta, 03 Jumadil-Ula 1443 H
08 Desember 2021 M

Kepala Laboratorium
Sistem Manufaktur Terintegrasi

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

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

SURAT PERNYATAAN KEASLIAN

Demi Allah saya akui bahwa karya ini merupakan karya saya sendiri kecuali kutipan dan ringkasan yang setiap salah satunya telah dicantumkan sumbernya. Jika ditemukan di kemudian hari ternyata terbukti pengakuan saya ini tidak benar dan melanggar peraturan yang sah dalam karya tulis dan hak kekayaan intelektual maka saya bersedia ijazah yang saya terima untuk ditarik oleh Universitas Islam Indonesia.

Yogyakarta, 7 Desember 2021



Nayoko Prasetyo Jati

LEMBAR PENGESAHAN PEMBIMBING

**INTEGRASI *KANSEI ENGINEERING* DAN *KANO* BERBASIS
NATURAL LANGUAGE PROCESSING (NLP) UNTUK
MENUNJANG PENGEMBANGAN PRODUK JASA PADA
PARIWISATA CANDI BOROBUDUR**

TUGAS AKHIR

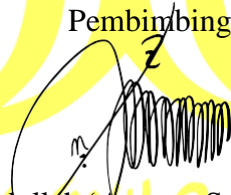
Disusun Oleh :

N a m a : Nayoko Prasetyo Jati

NIM : 17522119

Yogyakarta, 8 Desember 2021

Pembimbing



Abdullah Azzam, S.T., M.T.

الجامعة الإسلامية
الاستاذ الدكتور
الابن عبد العزيز

LEMBAR PENGESAHAN DOSEN PENGUJI

**INTEGRASI KANSEI ENGINEERING DAN KANO BERBASIS
NATURAL LANGUAGE PROCESSING (NLP) UNTUK
MENUNJANG PENGEMBANGAN PRODUK JASA PADA
PARIWISATA CANDI BOROBUDUR
TUGAS AKHIR**

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata-1 Teknik Industri

Ketua Penguji

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

Anggota 1

Harwati, S.T., M.T.

Anggota 2

Muchamad Sugarindra, S.T., M.T.

Mengetahui,

Ketua Program Studi Teknik Industri

Fakultas Teknologi Industri

Universitas Islam Indonesia



(*Immawan, S.T., M.M.*)

HALAMAN PERSEMBAHAN

Dengan Menyebut Nama Allah yang Maha Pengasih dan Maha Penyayang. Puji Syukur kehadiran Allah SWT atas nikmat yang telah diberikan. Skripsi ini saya persembahkan kepada:

Bapak Endro Dwiyatno dan Ibu Rini Erwindarti selaku kedua orang tua saya yang tiada henti mendo'akan, mendidik, dan memberi semangat dari kecil hingga saat ini. Beliau adalah tokoh utama dan segalanya dalam hidup saya yang tak akan tergantikan sampai kapanpun. Terimakasih atas pengorbanan dan segala hal yang tidak bisa diungkapkan kata dan tak akan pernah bisa dibalas oleh apapun.

Adik saya prames dan keluarga besar, terimakasih untuk semangat, dukungan, dan motivasi yang selalu diberikan sampai pada tahap ini.

Serta kerabat, sahabat, dan teman-teman saya yang selalu membantu dan menemani hari-hari dari awal sampai akhir masa perkuliahan ini.

HALAMAN MOTTO

"Barang siapa yang tidak mensyukuri yang sedikit, maka ia tidak akan mampu mensyukuri sesuatu yang banyak". (HR. Ahmad)

"Kerjakanlah urusan duniamu seakan-akan kamu hidup selamanya dan laksanakanlah urusan akhiratmu seakan-akan kamu akan mati besok". (HR. Ibnu Asakir)

"Sesungguhnya Allah tidak akan mengubah nasib suatu kaum sehingga mereka mengubah keadaan yang ada pada diri mereka sendiri." (Q.S Ar-Ra'd: 11)

KATA PENGANTAR



Assalamu 'ailaikum wa rahmatullahi wa barakatuh

Alhamdulillah rabbil'alamiin, segala puji dan syukur penulis sampaikan kepada Allah Subhanahu Wa Ta'ala atas berkat rahmat dan nikmat-Nya. sholawat dan salam senantiasa penulis panjatkan kepada Nabi besar Muhammad Shallallahu 'Alaihi Wa Sallam beserta keluarga, sahabat, serta para pengikutnya hingga akhir zaman yang telah berjuang dan membimbing kita dari zaman kegelapan menuju jalan yang terang benderang untuk menggapai ridho Allah Subhanahu Wa Ta'ala. Alhamdulillah puji syukur penulis dapat menyelesaikan penelitian Tugas Akhir berjudul "INTEGRASI KANSEI ENGINEERING DAN KANO BERBASIS NATURAL LANGUAGE PROCESSING (NLP) UNTUK MENUNJANG PENGEMBANGAN PRODUK JASA PADA PARIWISATA CANDI BOROBUDUR" penelitian ini merupakan salah satu persyaratan dalam menempuh gelar Sarjana Teknik (S.T) di Fakultas Teknologi Industri Universitas Islam Indonesia.

Adapun dalam penyusunan laporan Tugas Akhir ini penulis banyak mendapatkan bantuan, dukungan dan motivasi dari berbagai pihak. Untuk itu penulis ingin mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Ir. Hari Purnomo, M.T. selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
2. Bapak Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D selaku Ketua Jurusan Teknik Industri, Universitas Islam Indonesia.
3. Bapak Dr. Taufiq Immawan, S.T., M.M. selaku Ketua Program Studi S1 Teknik Industri Fakultas Teknologi Industri, Universitas Islam Indonesia.
4. Bapak Abdullah 'Azzam, S.T., M.T selaku dosen pembimbing tugas akhir yang telah memberikan bimbingan kepada penulis dalam penyelesaian penelitian serta memberikan masukan-masukan yang sangat bermanfaat bagi penulis dalam melaksanakan kegiatan penelitian tugas akhir.
5. Ayah, adik, dan saudara dan teman teman yang senantiasa memberikan doa dan kasih sayang terhadap penulis sehingga dapat termotivasi dalam mengerjakan tugas akhir dan terlaksana dengan lancar.
6. Rekan laboratorium lab sistem manufaktur, terutama pada laboran, teman angkatan 2018 dan 2019 yang selalu memberikan semangat dan dukungan selama saya mengerjakan tugas akhir.
7. Teman-teman Teknik Industri UII dan semua pihak yang telah mendoakan dan membantu proses penyelesaian laporan tugas akhir. Semoga kebaikan-kebaikan yang diberikan menjadi amal sholeh dan mendapat balasan yang berlipat ganda dari Allah Subhanahu Wa Ta'ala. Aamiin.

Penulis menyadari bahwa penulisan laporan tugas akhir ini masih jauh dari sempurna, untuk itu penulis mengharapkan kritik, saran dan masukan yang bersifat membangun demi kesempurnaan penulisan di masa yang akan datang. Akhir kata semoga laporan Laporan Akhir ini dapat digunakan sebagai mana mestinya serta berguna khususnya bagi penulis dan bagi para pembaca pada umumnya.

Yogyakarta, 8 Desember 2021



Nayoko Prasetyo Jati

NIM. 17522119

ABSTRAK

Sebagai salah satu destinasi “Program 10 Bali Baru”, Candi Borobudur menjadi ujung tombak pariwisata Indonesia yang akan dikenalkan kepada lingkup internasional. Maka dari itu, sudah menjadi kewajiban bagi pihak pengelola untuk memperhatikan kepuasan pengunjungnya. Penelitian menggunakan pendekatan leksikon dan *Natural Language Processing* sebelumnya telah digunakan untuk penyaringan gejala depresi pada pengguna *Twitter*, sedangkan penelitian ini bertujuan mengetahui integrasi antara *NLP* dan *Kansei Engineering Kano* pada pengembangan produk jasa Candi Borobudur. Digunakan data ulasan pengunjung yang diambil dari *tripadvisor* sebagai salah satu data primer yang digunakan. Dilakukan pelabelan dan klasifikasi kedalam bentuk sentimen positif dan negatif dengan menggunakan metode *Support vector machine (SVM)* dan *Naive Bayes Classifier (NBC)* kedua metode tersebut juga akan dibandingkan performasinya dalam mengklasifikasikan ulasan pariwisata. Hasil klasifikasi dilanjutkan dengan asosiasi untuk mendapatkan *Kansei words*. dari hasil tersebut akan diolah menggunakan *Kansei Engineering* tipe 1 kedalam kategorisasi kualitas jasa terpenting dengan bantuan metode *Kano*. Ekstraksi data dari integrasi metode *NLP* dengan menggunakan kelompok *Sentiment* negatif didapatkan *Kansei words* sejumlah delapan kata yaitu “Murah”, “Menikmati”, “Indah”, “Rileks”, “dekat”, “Sepi”, “Bagus”, “ Tersedia” dengan tingkat akurasi menggunakan metode *SVM* sebesar 86.79% dan hasil dari *NLP* yang diintegrasikan dengan *Kansei* dan *Kano* dianggap efektif karena didapatkan beberapa atribut jasa terdapat kemiripan pada pencarian populer pada *tripadvisor*, yaitu seperti *exit, morning, temple, sunrise, heat*. Serta beberapa atribut yang tidak tercantum namun dianggap penting berdasarkan hasil integrasi *Kansei* dan *kano* seperti harga tiket, pengelolaan kandang gajah, dan pamflet.

Kata kunci :Klasifikasi, Asosiasi Kata, *Natural Language Processing (NLP)*, *Support vector machine (SVM)*, *Naive Bayes Classifier (NBC)*, *Kansei Engineering*, *Kano*, *Candi Borobudur*

DAFTAR ISI

HALAMAN JUDUL.....	i
SURAT KETERANGAN PENELITIAN.....	ii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iii
LEMBAR PENGESAHAN DOSEN PENGUJI.....	v
HALAMAN PERSEMBAHAN.....	vi
HALAMAN MOTTO.....	vii
KATA PENGANTAR.....	viii
ABSTRAK.....	x
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xiv
BAB 1 PENDAHULUAN.....	1
1.1 LATAR BELAKANG	1
1.2 RUMUSAN MASALAH	4
1.3 BATASAN MASALAH	4
1.4 Tujuan Penilitan	4
1.5 Manfaat Penelitian	4
1.6 Statistika Penulisan	5
BAB 2 TINJAUAN PUSTAKA.....	7
2.1 Kajian Induktif	7
2.2 Kajian Deduktif.....	17
2.2.1 Candi Borobudur.....	17
2.2.2 <i>Big Data</i>	17
2.2.3 <i>Text mining</i>	18
2.2.4 <i>Web Scrapping</i>	19
2.2.5 <i>Klasifikasi</i>	19
2.2.6 <i>Support vector machine</i>	20
2.2.7 <i>Naive Bayes Classifier</i>	20
2.2.8 <i>Natural Language Processing</i>	21
2.2.9 <i>Kansei Engineering</i>	22
2.2.10 Uji Validitas	24
2.2.11 Uji Relibilitas.....	25
2.2.12 <i>Kano Model</i>	25
BAB 3 METODE PENELITIAN.....	27
3.1 Objek dan Subjek Penelitian	27
3.2 Populasi dan Sample Penelitian	27

3.3	Metode Pengumpulan Data	27
3.4	Tahap Penelitian.....	28
BAB 4	PENGUMPULAN DAN PENGOLAHAN DATA.....	33
4.1	Pemilihan Service Domain.....	33
4.2	Perentangan Ruang Semantik	33
4.2.1	Pengumpulan <i>data ulasan</i>	33
4.2.2	Ekstraksi data ulasan.....	35
4.3	Perentangan Ruang Atribut	55
4.3.1	Penentuan Atribut Jasa.....	55
4.3.2	Pengambilan Data Derajat Kepentingan dan Kepuasan	55
4.3.3	Pengambilan Data <i>Kano</i>	56
4.3.4	Uji <i>Validitas</i>	56
4.3.5	Uji Realibilitas	58
4.3.6	Pengukuran Kualitas Pelayanan Candi Borobudur.....	58
4.4	Integrasi <i>Kansei Engineering</i> Dengan Metode <i>Kano</i>	59
4.4.1	Pengolahan Kuisisioner <i>Kano Kano</i>	59
4.4.2	Menentukan Kategori <i>Kano</i> Untuk Setiap Atribut	60
BAB 5	ANALISIS DAN PEMBAHASAN.....	63
5.1	Integrasi <i>Natural Language Processing</i> dan <i>Kansei-Kano</i>	63
5.1.1	Hasil <i>Kano</i>	66
5.2	Rekomendasi Perbaikan Atribut Jasa.....	69
5.2.1	Rekomendasi 1	69
5.2.2	Rekomendasi 2.....	71
5.2.3	Rekomendasi 3.....	73
BAB 6	PENUTUP.....	76
6.1	Kesimpulan	76
6.2	Saran.....	77
	DAFTAR PUSTAKA.....	78
	LAMPIRAN.....	82

DAFTAR TABEL

Tabel 2. 1 Perbandingan Penelitian	12
Tabel 4. 1 <i>Input Output Normalization</i>	36
Tabel 4. 2 <i>Pseudocode Normalization</i>	36
Tabel 4. 3 <i>Input Output Casefolding</i>	37
Tabel 4. 4 <i>Pseudocode Casefolding</i>	37
Tabel 4. 5 <i>Input Output</i> Tokenizing.....	38
Tabel 4. 6 <i>Input Output</i> Filtering	38
Tabel 4. 7 <i>Pseudocode Filtering</i>	39
Tabel 4. 8 <i>Pseudocode</i> Pembobotan	39
Tabel 4. 9 Hasil Pembobotan	40
Tabel 4. 10 Perhitungan Skor Sentimen	41
Tabel 4. 11 <i>Pseudocode</i> Skor Sentimen	41
Tabel 4. 12 Jumlah Sentimen.....	44
Tabel 4. 13 Pembagian Skenario 1	45
Tabel 4. 14 Pembagian Skenario 2	45
Tabel 4. 15 skenario 3	46
Tabel 4. 16 <i>pseudocode</i> pembagian data uji dan latih	46
Tabel 4. 17 <i>Confusion Matrix</i>	48
Tabel 4. 18 <i>pseudocode</i> klasifikasi	48
Tabel 4. 19 Hasil Asosiasi	52
Tabel 4. 20 <i>pseudocode</i> asosiasi	53
Tabel 4. 21 Ringkasan Asosiasi Terpilih	54
Tabel 4. 22 <i>Kanseiwords</i> terpilih.....	54
Tabel 4. 23 Atribut Jasa	55
Tabel 4. 24 Kuisisioner <i>Kano</i>	56
Tabel 4. 25 Uji <i>Validitas Semantic Differential</i>	56
Tabel 4. 26 Uji <i>Validitas</i> kuisisioner <i>kano</i>	57
Tabel 4. 27 Uji Realibilitas	58
Tabel 4. 28 Pengolahan Kualitas Pelayanan	58
Tabel 4. 29 Aturan Evaluasi <i>Kano</i>	59
Tabel 4. 30 Perhitungan Kategori <i>Kano</i>	60
Tabel 4. 31 Penentuan kategori <i>kano</i>	61
Tabel 4. 32 Hasil Kategori <i>Kano</i>	61
Tabel 5. 1 Kategori <i>Must Be</i>	66
Tabel 5. 2 Kategori <i>One Dimensional</i>	67
Tabel 5. 3 Kategori <i>Indifferent</i>	68

DAFTAR GAMBAR

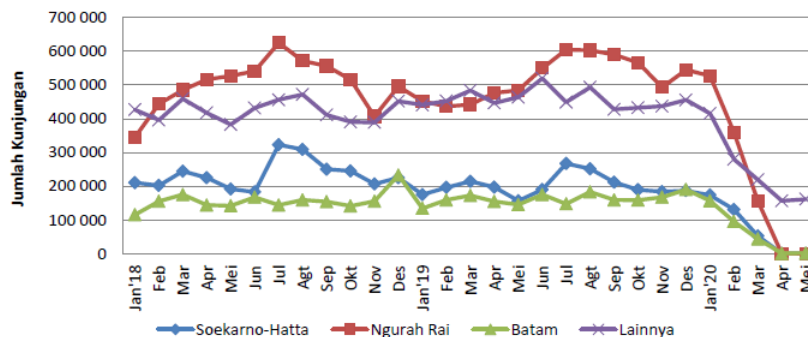
Gambar 1 1 Grafik kunjungan Wisata mancanegara	1
Gambar 3 1 Alur Penelitian	28
<i>Gambar 4. 1 Web Harvy</i>	34
Gambar 4. 2 Penggunaan <i>Web Harvy</i> 1	34
Gambar 4. 3 Penggunaan Web Harvy 2	
Gambar 4. 4 Grafik Performasi <i>SVM dan Naive Bayes</i>	
Gambar 4. 5 Frekuensi Muncul Label negatif	51
Gambar 4. 6 Frekuensi Muncul Label Positif	52
Gambar 5. 4 Panduan Pengunjung	70

BAB 1

PENDAHULUAN

1.1 LATAR BELAKANG

Pariwisata merupakan salah satu sektor yang tumbuh dan berkembang di berbagai wilayah Indonesia. Perkembangan pariwisata tentu saja memberikan pengaruh terhadap pertumbuhan perekonomian suatu wilayah dengan harapan bahwa sektor tersebut mampu meningkatkan pendapatan wilayah. Pariwisata mampu mendorong sektor-sektor lain untuk ikut tumbuh dan berkembang bersama-sama. Dari tahun 2020, Kerugian terhadap sektor pariwisata dapat dilihat pada penurunan jumlah kunjungan wisatawan mancanegara yaitu total kunjungan wisman pada Januari-Mei 2020 sebesar 2,9 juta menurun 53,36 persen dari tahun sebelumnya yang sebesar 6,3 juta kunjungan (BPS, 2020).



Gambar 1 1 Grafik kunjungan Wisata mancanegara

Untuk mengembalikan kondisi pariwisata Indonesia, pemerintah sedang mengembangkan program 10 bali baru untuk mempromosikan destinasi baru di Indonesia yang tidak kalah eksotis dari Bali (Bima et al., 2020.) Berdasarkan hasil wawancara dengan General Manager Taman Wisata Candi (TWC), jumlah pengunjung pada tahun 2019 mencapai 4,39 juta sedangkan tahun 2020 hanya dikunjungi sebanyak 996.000 pengunjung saja. Terjadi penurunan hingga 440% pengunjung dalam 1 tahun. Untuk mengetahui penyebab penurunan pengunjung, peneliti berupaya untuk melakukan pencarian sentimen negatif yang telah dirasakan oleh pengunjung serta melakukan

analisis terhadap permasalahan tersebut. Sehingga dari analisis tersebut dapat dijadikan pandangan terhadap pengembangan Candi Borobudur kedepannya.

Di era saat ini, banyak sekali situs yang menyediakan tempat untuk mengekspresikan emosi dan pendapat, *tripadvisor* adalah salah satu tempat bagi orang-orang untuk mengekspresikan apa yang telah dirasakan setelah melakukan perjalanan wisata di suatu tempat yang biasa disebut *knowledge sharing*. Selain itu, dalam penelitian yang dilakukan oleh Okazaki et al., (2016), dilakukan perbandingan terhadap nilai kepercayaan *review* wisata yang terdapat pada *tripadvisor* dan *facebook*, dikarenakan dalam *tripadvisor reviewer* benar-benar melakukan penilaian terhadap apa yang telah dirasakannya.

Pada tanggal 28 Agustus 2021, berdasarkan data yang didapatkan pada platform *tripadvisor*, diketahui ulasan pada Candi Borobudur berjumlah 3294 ulasan. Dikarenakan hasil *review* yang tersedia mencapai ribuan, maka dirasa sulit apabila ingin memahami *review* tersebut secara manual. Sehingga perlu adanya metode untuk memudahkan penerjemahan hasil *review* tersebut, digunakanlah *Text mining* dengan bantuan *Natural Language Processing (NLP)* sebagai salah satu tools untuk mengubah bahasa sehari-hari menjadi bahasa yang dimengerti oleh mesin. *NLP* merupakan cabang dari kecerdasan buatan yang berhubungan dengan interaksi antara komputer dan manusia menggunakan bahasa alami (Prasetyo et al., 2021). Penelitian menggunakan pendekatan leksikon dan *NLP* sebelumnya telah digunakan untuk penyaringan gejala depresi pada pengguna *Twitter*, hasilnya diperoleh bahwa pendekatan leksikon dan *NLP* memiliki akurasi lebih baik dari pada menggunakan pendekatan machine. Sedangkan dalam penelitian ini akan dititik beratkan sebagai dasar untuk mendapatkan ekstraksi data yang berhubungan dengan kepuasan pengunjung serta prioritas yang perlu ditingkatkan berdasarkan hasil klasifikasi yang didapatkan dari hasil integrasi *Kansei* dan *kano* terhadap fasilitas Candi Borobudur.

Metode klasifikasi yang digunakan pada *NLP* dalam penelitian ini yaitu *Support vector machine (SVM)* dikarenakan berdasarkan beberapa penelitian, hampir semua kasus penyelesaian menggunakan *SVM* lebih baik dibandingkan dengan metode klasifikasi yang lain. dilakukan penelitian terkait perbandingan ke tujuh metode klasifikasi, dan dihasilkan pengukuran bahwa *SVM* dan *Naive Bayes* berada pada posisi 2 teratas berdasarkan kriteria presisi, *recall*, *f-measure* serta dengan melakukan uji coba berbagai

simulasi pembagian data ujiannya menggunakan *crossvalidation* (Narayan, 2020). Sehingga dalam penelitian ini juga dilakukan perbandingan antara metode klasifikasi yang terbaik untuk diterapkan pada pengolahan data ulasan yang didapat dan diharapkan dapat memberikan usulan pengembangan desain produk jasa yang berguna untuk pelayanan borobudur

Untuk dapat memahami persepsi langsung dari *review* yang didapatkan secara emosional terhadap apa yang telah dirasakan, maka metode yang digunakan adalah *Kansei Engineering*, di mana *Kansei Engineering* adalah pendekatan ergonomis dari pengembangan produk/layanan berorientasi pelanggan dan dapat menerjemahkan subjektif pengguna users persepsi menjadi spesifikasi desain (Yeh & Chen, 2018). Sehingga diharapkan dengan menggunakan *Kansei Engineering* dapat memberikan rekomendasi berupa perancangan desain sesuai dengan keadaan emosional dan kebutuhan pengunjung.

Dalam penggunaan metode *Kansei Engineering* tidak dapat berdiri sendiri dan perlu melibatkan metode yang berbeda dari disiplin pemasaran, psikologi, dan statistik serta beberapa langkah dengan menggunakan kemungkinan alat yang berbeda pula. Dalam penelitian kali ini digunakanlah *kano* sebagai media dalam menentukan prioritas perbaikan yang dilakukan. *Kansei word* yang didapatkan digunakan untuk penentuan prioritas fasilitas yang perlu dikembangkan oleh Candi Borobudur. Konsep dari *Kano* berfokus kepada keinginan serta kebutuhan dari konsumen terhadap produk maupun jasa yang akan dikembangkan, sementara *Kansei Engineering* merupakan sistem yang menerjemahkan kesan ke dalam parameter produk (Prabowo, et al., 2019). Dengan di integrasikannya *NLP* dengan *Kansei Engineering* dan *kano* diharapkan mampu mengidentifikasi keinginan serta harapan dari konsumen secara penuh yang kemudian dapat dijadikan respon untuk perbaikan pada kualitas layanan yang ada di Candi Borobudur

1.2 RUMUSAN MASALAH

Berdasarkan uraian latar belakang diatas maka perumusan masalah pada penelitian ini adalah Bagaimana integrasi *NLP* dengan *Kansei Engineering* dan *kano* untuk pengembangan produk jasa pada Candi Borobudur?

1.3 BATASAN MASALAH

Supaya penelitian yang dilakukan lebih terarah dan topik yang dibahas tidak meluas, maka perlu dilakukan pembatasan lingkup penelitian. Adapun pembatasan lingkup penelitian ini adalah :

- a. Data yang digunakan merupakan data ulasan mengenai kunjungan Candi Borobudur pada situs *tripadvisor*.
- b. Data ulasan yang digunakan dalam penelitian yaitu dua ulasan mulai 1 januari 2019 – 1 September 2021.
- c. Ekstraksi data yang digunakan sebagai *Kansei words* hanya menggunakan hasil *NLP*
- d. Pengembangan produk menggunakan metode integrasi *Kansei* dengan metode *kano*.

1.4 Tujuan Penilitan

Penelitian ini bertujuan mengetahui integrasi antara *NLP* dan *Kansei Engineering Kano* pada pengembangan produk jasa Candi Borobudur.

1.5 Manfaat Penelitian

Diharapkan penelitian yang dilakukan dapat memberikan manfaat seperti berikut :

1. Pihak Akademisi
 - a. Memberikan informasi terkait gambaran umum persepsi pengunjung terhadap pariwisata Candi Borobudur.
 - b. Sebagai referensi untuk peneliti-peneliti berikutnya, khususnya yang berkaitan dengan mengekstraksi pengetahuan dari *review online* untuk mendukung desain produk menggunakan metode *NLP* dan *Kansei*.
2. Pihak Candi Borobudur
 - a. Hasil ekstraksi ulasan *online* dapat dijadikan sebagai bahan evaluasi kepada pihak pengelola pariwisata Candi Borobudur terhadap kepuasan pengunjung.

- b. Memudahkan pihak lain yang memiliki kepentingan untuk melihat informasi yang terdapat pada ulasan atau komentar yang sangat banyak, sehingga dapat melakukan penanganan dan evaluasi yang lebih baik.

1.6 Statistika Penulisan

Sistematika penulisan dibuat agar penulisan laporan dapat tersusun rapi dan terstruktur. Berikut merupakan sistematika penulisan pada penelitian ini :

BAB I PENDAHULUAN

Membahas latar belakang masalah terkait keadaan pariwisata Candi Borobudur saat ini dan penyelesaiannya berdasarkan penelitian yang dilakukan. Bab 1 juga berisi kajian singkat tentang perumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penelitian.

BAB II KAJIAN LITERATUR

Bab kajian pustaka berisi tentang kajian induktif dan deduktif yang digunakan dalam penelitian yang dilakukan. Kajian induktif membahas mengenai penelitian-penelitian yang terdahulu sebagai acuan dalam penelitian ini. Kajian induktif juga dilakukan untuk mencari kesempatan topik penelitian agar tidak mirip dengan penelitian sebelumnya. Kajian deduktif mengenai pengertian serta penjabaran keilmuan yang digunakan sebagai panduan penelitian. Kajian pustaka yang dilakukan pada penelitian ini berfokus kepada *Kansei Engineering*, klasifikasi, dan *Natural Language Processing*.

BAB III METODE PENELITIAN

Mengandung uraian mengenai populasi dan sampel penelitian, jenis dan sumber data, variabel penelitian, metode pengumpulan data, metode analisis data, dan diagram alir penelitian.

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Berisi tentang data yang didapatkan melalui proses *scrapping* selama penelitian dan proses menganalisa data tersebut. Data yang digunakan

pada penelitian ini merupakan data ulasan *online* pengunjung pariwisata Candi Borobudur. Hasil pengolahan data ditampilkan baik dalam tabel maupun grafik. Pada sub bab ini merupakan acuan dalam pembahasan hasil yang akan ditulis pada sub bab pembahasan hasil.

BAB V PEMBAHASAN

Melakukan pembahasan hasil yang diperoleh dalam penelitian, dan kesesuaian hasil dengan tujuan penelitian sehingga dapat menghasilkan sebuah rekomendasi melalui *Kansei Engineering*.

BAB VI PENUTUP

Berisi tentang kesimpulan terhadap analisis yang dibuat dan rekomendasi atau saran-saran atas hasil yang dicapai dan permasalahan yang *ditemukan* selama penelitian.

DAFTAR PUSTAKA

LAMPIRAN

BAB 2

TINJAUAN PUSTAKA

2.1 Kajian Induktif

Kajian induktif menjelaskan penelitian-penelitian sebelumnya yang telah dilakukan yang berkaitan dengan topik pada penelitian ini. Penelitian yang dilakukan oleh Fatyanosa & Bachtiar, (2017) yang berjudul “*Classification Method Comparison on Indonesian Social Media Sentiment Analysis*” berisi tentang analisis sentimen terkait klasifikasi media sosial yang ada di Indonesia terkait pemilihan gubernur Jakarta. Penelitian ini digunakan untuk membandingkan metode yang terbaik dari 5 metode yang digunakan, metode yang digunakan adalah *Summation Method*, *Average on Data*, *Average on Data with the threshold*, *Weighted Average*, dan *Naive Bayes*. Hasil dari penelitian ini menunjukkan bahwa analisis sentimen dengan menggunakan metode *Naive Bayes* berdasarkan pengukuran *precision*, *Recall*, dan *Accuracy*.

Fanissa et al., (2018) dalam penelitiannya yang berjudul “*Analisis Sentimen Pariwisata di Kota Malang Menggunakan Metode Naive Bayes dan Seleksi Fitur Query Expansion Ranking*” melakukan analisis sentimen terhadap ulasan dari masyarakat tentang pariwisata di Kota Malang yang diambil dari *website tripadvisor*. Penelitian ini menggunakan metode *Naive Bayes* dengan seleksi fitur *Query Expansion Ranking* untuk mengurangi komputasi pada proses klasifikasi. Hasil penelitian ini menunjukkan bahwa hasil seleksi fitur sebesar 75% dan memiliki akurasi terbaik sebesar 86.6%.

Zun, et al. (2018) dengan penelitian yang berjudul “*Comparison of Naive Bayes and Support Vector Machine Classifier on Document Classification*” melakukan klasifikasi pada *field* dari *research paper* IT serta melakukan komparasi terhadap 2 metode terbaik yang digunakan. Metode yang digunakan pada penelitian ini yaitu *Naive Bayes* dan *Support Vector Machine*. Penelitian ini menghasilkan perbandingan klasifikasi terhadap kebenaran dan keakuratan dari metode yang digunakan. Didapat *Support Vector Machine* lebih akurat dibandingkan dengan *Naive Bayes*.

Yeh & Chen, (2018) melakukan penelitian yang berjudul “ *Applying Kansei Engineering and Data Mining to Design Door-to-Door Delivery Service*” berisi tentang integrasi antara metode *kansei Engineering* dengan teknik *data mining* pada area industri jasa *Door to Door*. Tujuan dari penelitian ini yaitu menggunakan *customer’feelings* kedalam pengembangan desain proses jasa. Hasil yang dicapai dari penelitian ini yaitu kombinasi respons persepsi penting menghasilkan niat penggunaan positif dan kombinasi properti menghasilkan respons persepsi penting ini. Dengan demikian, penyedia layanan D2DD dapat meningkatkan atau membuat layanannya berdasarkan temuan penelitian.

Song et al., (2017) melakukan penelitian berjudul “A Novel Classification Approach Based on Naïve Bayes for *Twitter Sentiment Analysis*”. Berisi tentang tentang klasifikasi ulasan novel berdasarkan data ulasan pembaca di sosial media *Twitter*. Metode penelitian yang digunakan adalah Multinomial Naïve Bayes (MBC), Maximum Entropy, dan *Support vector machine (SVM)* dengan data sebanyak 1.600.000 Dataas. Terdapat banyak faktor yang mempengaruhi performansi, seperti koleksi data, banyaknya fitur, metode pra-proses, dan pemilihan algoritma klasifikasi. Hasil penelitian menunjukkan bahwa akurasi dipengaruhi oleh jumlah data dan pemilihan algoritma klasifikasi. Akurasi terbaik ditunjukkan oleh metode MBC dengan akurasi sebesar 85,33% dibandingkan dengan metode yang lain.

Azis et al., (2020) dalam penelitiannya yang berjudul “Analisis Perbandingan Performa Metode Klasifikasi pada *Dataset Multiclass* Citra Busur Panah” melakukan perbandingan performa pada tujuh metode klasifikasi yang digunakan dalam mengelola *dataset* yang memiliki lebih dari dua label (*multiclass*). *Dataset* yang digunakan dalam penelitian ini adalah *dataset* citra busur panah serta performa yang diukur berupa seluruh nilai *cross-validation* dari akurasi, presisi, *recall*, dan *f-measure*. Hasil dari penelitian ini menunjukkan bahwa metode *SVM* memiliki perofrma yang lebih baik dibandingkan enam metode lainnya pada kasus *dataset multiclass* citra busur panah.

Al-smadi et al., (2017) yang berjudul “Deep Recurrent Neural Network vs. *Support vector machine* for Aspect-based Sentiment Analysis of Arabic Hotels’ Reviews” dilakukan analisis sentimen pada data *review* hotel di Arab. Terdapat perbandingan antara dua metode pengklasifikasian, yaitu *Recurrent Neural Network (RNN)* dan *Support vector machine (SVM)* dengan *dataset* berjumlah 24.028 *review*. Hasil penelitian menunjukkan

bahwa metode *SVM* memiliki performa yang lebih baik dibandingkan *RNN* berdasarkan nilai F-1 dan akurasi dengan nilai sebesar 89,9% dan 95,4%.

Hartono, (2020) dengan penelitian berjudul “*The Modified Kansei Engineering-Based Application for Sustainable Service Design*” melakukan pendekatan terhadap integrasi *Kansei Engineering* dengan berbagai metode seperti *Kano* dan *TRIZ (Theory of Inventive Problem Solving)* dengan subjek penelitian pada *airport lounge*. Hasil dari penelitian ini menghasilkan *Kansei Engineering* pada *sustainable service development*. Sehingga hasil yang didapatkan dapat dijadikan sebagai referensi pengelola *airport lounge* untuk melakukan perbaikan kelanjutan.

Hartono, (2017) dalam penelitiannya yang berjudul “*How Kansei Engineering, Kano and QFD Improve Logistic Services*” melakukan pengembangan produk jasa pada pelayanan ekspedisi barang menggunakan metode integrasi *Kansei Engineering, Kano, dan Quality Function Development (QFD)*. Diharapkan dengan menggunakan ketiga metode tersebut dapat menambahkan variasi *improvement* pada produk. Data survey sebanyak 157 customer. Dihasilkan kategori *Kano* pada *Attractive (A)*, *QFD* berupa *critical improvement* pada bagian *modular system helmet*. Dan didapatkan *Kansei word* tertinggi pada kelompok *positive* yaitu “*helped*”.

(Cahigas, 2020) melakukan penelitian berjudul “*Kansei Engineering-Based Model and Online Content Assesment in Evaluating Service Design of Lazada Express*” Penelitian ini melakukan pengembangan salah satu layanan *Lazada* yaitu *Lazada Express* berdasarkan ulasan pengguna menggunakan metode *Kansei Engineering* serta melakukan pengolahan data menggunakan *Natural Language Processing*. Didapatkan hasil berupa 5.75% positif *Kansei word* dan 94.25% negatif *Kansei word*. Hasil tersebut dapat dijadikan referensi untuk pengembangan *lazada express* berikutnya.

Hong-bin & Ming, (2020) dengan penelitiannya yang berjudul “*An uncertain Kansei Engineering methodology for behavioral service design*” Penelitian ini dilakukan untuk mendapatkan desain pelayanan bidang logistik yang sesuai dengan kebiasaan dan perasaan konsumen. Metode yang digunakan yaitu *Kansei engineering* dengan diintegrasikan oleh *Kano*. hasil yang didapatkan berupa hasil *kano* terletak ada *Attractive*, dan penggunaan *KE* pada penelitian ini memfokuskan pada perilaku konsumen yang dilakukan pada *service design* dan disajikan rekomendasi desain yang dibuat.

Masudin et al., (2020) dengan penelitiannya yang berjudul “*The effect of humanitarian logistics service quality to customer loyalty using Kansei Engineering: Evidence from Indonesian logistics service providers*” penelitian ini dilakukan untuk mengidentifikasi pengaruh kualitas layanan logistik terhadap kepuasan pelanggan dan loyalitas pelanggan dengan kepercayaan pelanggan pelayanan logistik kemanusiaan menggunakan metode *service quality (SERVQUAL)* dan *Kansei Engineering*. Hasil yang didapatkan berupa Kualitas layanan personel berpengaruh signifikan terhadap kepuasan pelanggan, kepuasan pelanggan berpengaruh signifikan terhadap loyalitas pelanggan, dan kualitas layanan teknis berpengaruh signifikan terhadap loyalitas pelanggan. *Kansei Engineering* pada penelitian ini digunakan untuk memberikan perspektif yang berbeda terhadap kualitas layanan logistik kemanusiaan.

Jiao & Qu, (2019) melakukan penelitian yang berjudul “A proposal for *Kansei knowledge extraction method based on Natural Language Processing technology and online product reviews*” berisi tentang pengolahan ulasan *online* pada suatu produk untuk mengetahui data ekstraksi beserta saran yang dapat diberikan. Metode yang digunakan berupa *Kansei Engineering* untuk penentuan desain, serta *Natural Language Processing (NLP)* untuk melakukan pengolahan data. Didapatkan hasil pengolahan menggunakan *NLP* lebih unggul dari berbagai aspek dibandingkan dengan pengolahan tradisional, serta dihasilkan *Kansei word positive* pada produk *logitech G502*.

8	The Modified <i>Kansei Engineering</i> -Based Application for Sustainable Service Design	Airport	Internasional	v	v	v	v		
9	How <i>Kansei Engineering</i> , <i>Kano</i> and QFD Improve Logistic Services	jasa Logistik	internasional	v	v	v		v	
10	<i>Kansei Engineering</i> -Based Model and Online Content Assessment in Evaluating Service Design of Lazada Express	Lazada	internasional	v	v		v	v	v
11	An uncertain <i>Kansei Engineering</i> methodology for behavioral service design	Logistik	Internasional	v	v		v	v	v

2.2 Kajian Deduktif

2.2.1 Candi Borobudur

Candi Borobudur merupakan candi buddha yang terletak pada jl Badrawati, kec. Borobudur, Magelang, Jawa Tengah. Borobudur merupakan peninggalan bersejarah yang pernah dijadikan *cultural heritage* (warisan budaya) yang ditetapkan oleh *UNESCO* sebagai tujuh dari keajaiban dunia (Sulistyaningsih & Susila, 2019) Wisatawan yang berkunjung bervariasi dari wisatawan lokal maupun mancanegara.

Candi Borobudur termasuk kedalam program “10 bali baru” yang dibuat oleh pemerintah. Penetapan 10 destinasi pariwisata prioritas “Bali Baru” Indonesia merupakan amanat Presiden melalui surat Sekretariat/kabinet Nomor B 652/Seskab/Maritim/2015 tanggal 6 November 2015 perihal Arahan Presiden Republik Indonesia mengenai Pariwisata dan Arahan Presiden pada Sidang Kabinet Awal Tahun pada tanggal 4 Januari 2016. 10 “Bali Baru” Indonesia sendiri merupakan makna kiasan dan lebih untuk memberi tekanan agar “*Performance*” masing-masing daerah itu bisa menyamai Bali yang setahun bisa mendatangkan empat juta wisatawan mancanegara yang disepakati oleh Pemerintah melalui Kementerian Pariwisata sejak tahun 2016.

2.2.2 *Big Data*

Big Data atau Mahadata secara bahasa adalah segala himpunan data yang memiliki jumlah dan media penyimpanan yang sangat banyak. Supriyanto et al., (2021) menyatakan *Big Data* adalah sekumpulan data yang berukuran besar yang jumlahnya terus bertambah, terdiri dari berbagai macam jenis data, terbentuk dengan terus menerus dengan kecepatan tertentu dan harus diproses dengan kecepatan tertentu.

Big Data telah menjadi bantuan besar sebagai bagian dari revolusi industri 4.0 yang memudahkan dalam, pengelolaan, penyimpanan serta analisis data dari berbagai sumber dalam jumlah besar. (Qi, 2020) telah mengidentifikasi tiga karakteristik penting *Big Data* dalam membedakannya dari data lain yang biasa kita jumpai, yaitu ukuran, kompleksitas, dan teknologi. Cara lain untuk mengidentifikasi *Big Data* adalah dengan menggunakan atribut 5V, yang merupakan singkatan dari *volume* (data dalam jumlah besar dalam *petabyte* atau *terabyte*), variasi (berbagai jenis terstruktur, semi terstruktur, atau tidak

terstruktur). data), kecepatan (aliran kontinu dan analisis data dalam waktu dekat atau hampir waktu nyata), kejujuran (integritas data), dan nilai (informasi berharga di balik data).

2.2.3 *Text mining*

Text mining dapat didefinisikan sebagai suatu proses menggali informasi serta terdapat interaksi antara user dengan sekumpulan dokumen menggunakan tools analisis ada. Feldman dan Sanger (2006) dalam bukunya yang berjudul *The Text mining Handbook* mengartikan tujuan dari *Text mining* yaitu untuk mendapatkan informasi yang berguna dari sekumpulan dokumen. sumber data yang digunakan pada *Text mining* adalah kumpulan teks yang memiliki format yang tidak terstruktur atau minimal semi terstruktur. Adapun tugas khusus dari *Text mining* antara lain yaitu pengkategorisasian teks (*text categorization*) dan pengelompokan teks (*text clustering*). Menurut (Benjamín, 2020) secara umum *pre-processing* dilakukan dalam enam tahapan, yaitu:

1. *Cleansing*

Tahap *Cleansing* adalah tahap eliminasi aksara nonalfabetis untuk menurunkan noise. Aksara yang dihapus adalah tanda baca seperti titik (.), koma (,), tanda tanya (?), dan tanda seru (!), serta simbol-simbol seperti tanda '@' untuk *username*, *hashtag* (#), emotikon, dan alamat *website*.

2. *Case Folding*

Case Folding adalah tahap untuk mengonversi karakter alfabet yang telah melalui tahap *Cleansing* ke huruf kecil (*lower case*).

3. *Tokenizing*

Tahap ini berfungsi sebagai pemecah kalimat berdasarkan tiap kata yang menyusunnya, yang disebut *term* atau *token*. *Tokenizing* dipecah berdasarkan spasi.

4. *Normalization* atau konversi *Slangword*

Tahap ini dilakukan agar kata-kata yang disingkat atau diperpanjang menjadi kata-kata yang normal sesuai dengan Kamus Besar Bahasa Indonesia (KBBI). Konversi *slangword* adalah proses mengubah kata tidak baku menjadi kata baku. Tahap ini dilakukan dengan bantuan kamus *slangword* dalam kata-kata baku dan memeriksa kata tersebut terdapat

dalam kamus *slangword* atau tidak. Jika kata tidak baku terdapat dalam kamus, maka kata tidak baku diubah ke kata baku yang terdapat di dalam kamus.

5. *Filtering* atau *Stopword Removing*

Tahap ini memroses agar kata-kata yang tidak penting atau tidak bermakna dihapus untuk analisis sentimen. Contoh kata-kata tersebut adalah atau, yang, dengan, di, ke, dan tetapi.

6. *Stemming*

Tahap ini berfungsi mengubah kata berimbuhan pada tiap kata yang telah terseleksi menjadi kata dasar.

2.2.4 *Web Scrapping*

Webscrapping merupakan proses penggalian data tidak terstruktur dari internet, yang dapat diambil untuk membangun kumpulan data skala besar dari data terstruktur. *Web Scrapping* dapat dilakukan secara manual, melalui aplikasi atau ekstensi browser, atau melalui perangkat lunak. Salah satu cara termudah adalah dengan menyalin dan menempelkan materi secara langsung dari sebuah halaman, namun hal ini dapat memakan waktu yang cukup lama untuk jumlah informasi yang lebih besar (Lunn & Ross, 2020).

2.2.5 *Klasifikasi*

Klasifikasi adalah sebuah proses analisa data yang menghasilkan model-model untuk menggambarkan kelas-kelas yang terkandung di dalam data (Fahdia et al., 2021). Tujuan dari melakukan klasifikasi data adalah untuk memprediksi dan mengelompokkan kelas dari suatu objek sehingga terdapat suatu data yang dapat dengan mudah dipahami (Dwitri et al., 2020). Model klasifikasi terbagi menjadi dua yaitu, melatih data kemudian membedakan kedalam *class* dan melakukan evaluasi terhadap objek tidak dikenal (Abdi et al., 2020). Metode – metode umum yang digunakan oleh periset untuk menyelesaikan kasus klasifikasi antara lain sebagai berikut ini (Bidang et al., 2016):

1. Pohon Keputusan
2. *Naïve Bayes*
3. Jaringan saraf Tiruan
4. Analisis Statistik
5. Algoritma Genetik
6. *Rough Sets*
7. Pengklasifian *K-Nearest Neighbour*

8. Metode Berbasis Aturan
9. *Memory Based Reasoning*
10. *Support vector machine*

2.2.6 *Support vector machine*

Support vector machine merupakan suatu teknik untuk melakukan prediksi, baik dalam kasus klasifikasi maupun pada regresi. Didalam machine learning, *SVM* termasuk kedalam kelas *supervised learning*, dikarenakan dalam implementasinya perlu dilakukan tahap pelatihan menggunakan *sequential training SVM* dan disusul dengan tahapan pengujian (Rofiqoh et al., 2017).

Penggunaan algoritme *SVM* yang bertujuan untuk klasifikasi teks dengan menggunakan bobot *indeks term* sebagai fitur, dirintis oleh Thorsten Joachim. Pembelajaran *SVM* telah dipopulerkan sejak tahun 1992 oleh Boser, Guyon, dan Vapnik (Rofiqoh et al., 2017)

Konsep Klasifikasi yang terdapat pada *Support vector machine* berupa mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua kelas data. *SVM* dapat bekerja pada *dataset* yang berdimensi tinggi dengan menggunakan *kernel* trik. *Support vector machine* hanya menggunakan beberapa titik data terpilih yang berkontribusi (*support vector*) untuk membentuk model yang akan digunakan dalam proses klasifikasi.

2.2.7 *Naive Bayes Classifier*

Naive Bayes Classifier (NBC) adalah pengklasifikasian statistik yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu class. Teorema *Bayes* dikombinasikan dengan “*Naive*” yang berarti setiap atribut atau variabel bersifat bebas (*independent*). Bayesian classification terbukti memiliki akurasi dan kecepatan yang tinggi saat diaplikasikan ke dalam database dengan data yang besar (Annur, 2018). Algoritma *NBC* termasuk dalam pembelajaran *supervised*, sehingga pada tahapan pembelajaran dibutuhkan data awal berupa data pelatihan untuk dapat mengambil keputusan. Pada tahap pengklasifikasian akan dihitung nilai probabilitas dari masing-masing label kelas yang ada terhadap masukan yang diberikan. Label kelas yang memiliki nilai probabilitas paling besar akan dijadikan label kelas data masukan tersebut.

2.2.8 *Natural Language Processing*

Natural Language Processing (NLP) adalah kombinasi dari ilmu komputer dan bidang kecerdasan buatan yang terkait dengan linguistik, berkaitan dengan bagaimana mesin memahami bahasa manusia untuk saling berinteraksi (Prasetyo et al., 2021) Dengan adanya *NLP*, komputer dapat belajar dan memahami bahasa manusia, sehingga komputer dapat berkomunikasi dengan manusia. Bahasa manusia adalah hal yang unik karena dibuat khusus untuk menyampaikan suatu makna. Untuk membuat komputer dapat mengerti bahasa manusia adalah tugas yang sulit, karena bahasa manusia memiliki struktur yang kompleks. Selain itu, setiap bahasa memiliki keunikannya sendiri dan mungkin memiliki makna ganda (Alash & Al-sultany, 2019). Sebagai contoh dapat dilihat dari kalimat berikut, “*Look at the dog with one eye*”, di mana kalimat tersebut dapat memiliki arti “melihat anjing dengan satu mata” atau “melihat anjing yang mempunyai mata satu”. Tujuan dari *NLP* adalah memberikan jawaban atau respon yang sesuai berdasarkan pemahaman makna bahasa manusia yang dilakukan oleh mesin (Alamanda & Kania, 2016).

Dua teknik utama untuk memahami *NLP* adalah *syntactic analysis* (analisis sintaksis) dan *semantic analysis* (analisis semantik). Kedua teknik digunakan untuk memverifikasi struktur bahasa. Analisis sintaksis mengacu pada tata bahasa, sedangkan analisis semantik merujuk pada penafsiran suatu kalimat.

Syntactic analysis (analisis sintaksis) adalah teknik pengaturan pada suatu kalimat sehingga kalimat memiliki tata bahasa yang benar. Analisis sintaksis melibatkan penentuan stuktur kalimat seperti subjek, predikat, kata benda, kata kerja, kata ganti, dan sebagainya. Sistem akan dapat membaca *Input* kalimat, yang akan dipecah menjadi kata-kata, dan pada akhirnya menghasilkan deskripsi yang terstruktur. Teknik ini dapat digunakan untuk menyederhanakan kalimat untuk memudahkan pencarian informasi. Selain itu, penggunaan analisis sintaksis juga dapat membantu mendeteksi keberadaan kata atau kalimat baru atau tidak biasa ((Alamanda & Kania, 2016).

Sebuah kalimat dapat disebut sebagai kalimat, apabila paling tidak terdiri dari subjek dan predikat, sebagai contoh kalimat “*Andi eat*”. Dengan menggunakan teknik analisis sintaksis, komputer dapat membedakan mana yang termasuk subjek (“*Andi*”) dan predikat “*eat*”. Kalimat yang terbentuk mungkin saja tidak memiliki makna apapun,

karena analisis sintaksis hanya memastikan bahwa struktur dari sebuah kalimat sudah benar.

Analisis semantik adalah teknik yang digunakan untuk memahami makna dan interpretasi dari struktur bahasa. Seseorang bisa memahami perkataan orang lain berdasarkan intuisi dan pengetahuan dari bahasa itu sendiri. Komputer tidak memiliki intuisi dan pengetahuan semacam ini, sehingga mereka membutuhkan metode lain, yaitu semantik. Semantik adalah proses penting karena *Output* semantik yang diharapkan adalah makna yang terkandung pada sebuah *Input* (Darzhinova, 2019) Analisis semantik memproses teks untuk mengidentifikasi dan memahami topik yang dimaksud. Semantik juga mempelajari hubungan antara berbagai konsep dalam teks. Sebagai contoh, apabila sebuah teks terdapat kata “*money*” dan “*accounting*”, maka topik yang sedang dibahas berkaitan dengan “*economy*”.

2.2.9 Kansei Engineering

Kansei Engineering adalah subjek komprehensif yang mengintegrasikan ilmu desain, ergonomi, dan teknik. Hal ini membutuhkan desainer untuk membangun jembatan antara perasaan psikologis pengguna dan solusi desain produk (Jiao & Qu, 2019). *Kansei* adalah kata yang diambil dari bahasa Jepang yang digunakan untuk menggambarkan sentimen dan ekspresi pelanggan.

Dalam pekerjaan baru-baru ini, rekayasa *Kansei* dilakukan dengan menggunakan parameter produk yang berbeda. Menurut Nagamachi, (1995) empat poin pertama mengenai teknologi yang adalah bagaimana memahami pengetahuan *Kansei* pengguna tentang produk. Bagaimana mengukur, menemukan, dan menangkapnya secara akurat adalah masalah utama yang harus dipecahkan dalam desain emosional produk, dan juga premis untuk memastikan bahwa produk tersebut dapat diterima dan dipuaskan secara emosional oleh konsumen. Karena kebutuhan emosional pengguna akan produk bersifat implisit, tidak tepat, dan ambigu, bagaimana mengekspresikan pemahaman persepsi pengguna secara akurat tentang produk dan bagaimana mengintegrasikan kebutuhan emosional pengguna ke dalam desain produk, telah menjadi fokus dan kesulitan untuk penelitian akademis dan pengembangan produk (abdullah, 2020). Menurut Nagamachi dan lokman (2011), *Kansei Engineering* terbagi menjadi bermacam tipe sebagai berikut :

1. *Kansei Engineering* tipe 1

Kansei Engineering Tipe I adalah tipe yang termudah untuk dipahami serta untuk dikenalkan. Metode ini memecah konsep produk yang ditargetkan menjadi konsep yang lebih terperinci, serta mengembangkannya ke sejumlah tingkatan, hal tersebut dapat ditafsirkan menjadi hal karakteristik fisik dari desain produk.

2. *Kansei Engineering* tipe 2

Kansei Engineering tipe II adalah teknik untuk mengartikan citra atau *Kansei* berdasarkan produk yang ada dalam benak konsumen menjadi elemen desain produk nyata. Untuk membangun sistem seperti itu, yaitu, semacam sistem cerdas, *Kansei Engineering* Tipe II dan komputer digunakan. Dalam membangun sistem cerdas ini, teknik yang terkait dengan ilmu komputer seperti Sistem Pakar, Jaringan Saraf Tiruan, dan Algoritma Genetika (GA) digunakan. Jika perlu, *Fuzzy Logic* mungkin juga digunakan. Saat ini, rekayasa *Kansei* yang memiliki kemungkinan tertinggi untuk digunakan dalam teknologi komputer adalah rekayasa *Kansei* Tipe II.

3. *Kansei Engineering* tipe 3

Pada *Kansei Engineering* tipe III juga diawali dengan studi *Kansei* yang terlihat juga pada karakteristik desain fisik. Perbedaan dari *Kansei Engineering* tipe III ini adalah model matematika yang telah termediasi, dan hubungan dari *Input* ke *Output* (karakteristik fisik) ditetapkan dengan mencari nilai koefisien. *Kansei Engineering* tipe III adalah metode yang memungkinkan analisis *Kansei* untuk mengidentifikasi nilai-nilai koefisien dengan memperkenalkan beberapa model matematika.

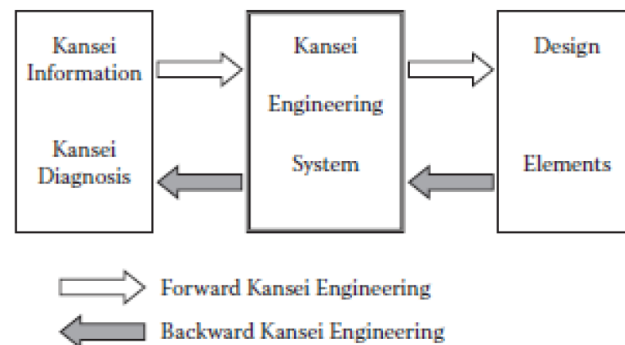
4. *Kansei Engineering* tipe 4

Kansei Engineering tipe IV ini disebut sebagai *Virtual Kansei Engineering*. *Virtual Kansei Engineering* merupakan teknologi yang mengintegrasikan *Virtual reality* dengan *Kansei Engineering*. Melalui *Virtual 15 Kansei Engineering*, dengan pengalaman *Virtual* maka seluruh simulasi yang ada dalam imajinasi kita, mampu untuk kita konfirmasi penyesuaian produk *Kansei* kita serta kegunaannya secara langsung. Ini merupakan teknologi yang melibatkan pelanggan dalam proses desain untuk mendapatkan kepuasan mereka secara langsung.

5. *Hybrid Kansei Engineering*

Pada penjelasan sebelumnya diketahui bahwa *Kansei Engineering* merupakan teknologi yang menerjemahkan *Kansei* atau gambar dalam pikiran konsumen tentang desain dari produk. *Kansei Engineering* dengan pendekatan tersebut disebut dengan *forward Kansei*

Engineering. Dalam *Hybrid Kansei Engineering* terdapat dua metode yaitu *forward* dan *backward*.



2.2.10 Uji Validitas

Menurut (Nazelina, 2020) *validitas* menyatakan tingkat kebenaran berdasarkan hasil yang diperoleh pengukuran dari kuesioner. Uji *validitas* yaitu pengukuran pada koefisien korelasi antara skor dari pertanyaan atau indikator yang dilakukan pengujian dengan skor keseluruhan pada variabelnya. Dalam menentukan kelayakan dari suatu *item* apakah dapat digunakan atau tidak yaitu melalui uji signifikansi koefisien korelasi dengan tingkat signifikansi yaitu 0,05 atau 5%, yang berarti *item* akan *valid* bila berkorelasi signifikan pada skor total *item*. Hasil *validitas* diketahui apabila r tabel lebih kecil dari r hitung sehingga dapat dinyatakan *valid*.

Berdasarkan Sujarweni (2007) hasil uji *validitas* dapat dinyatakan *valid* jika r hitung melebihi dari nilai r tabel. Nilai r tabel sendiri didapatkan berdasarkan pada tabel r dengan nilai $df = N-2$. Unaradjan (2019) menjelaskan bahwa untuk mendapatkan nilai r hitung dapat dilakukan dengan menggunakan rumus seperti pada persamaan :

$$r_{hitung} = \frac{n(\sum XY) - (\sum X) \cdot (\sum Y)}{\sqrt{\{n \cdot \sum X^2 - (\sum X)^2\} \cdot \{n \cdot \sum Y^2 - (\sum Y)^2\}}} \quad (2.1)$$

Keterangan :

r_{hitung} = Koefisien Korelasi

$\sum X$ = Jumlah Skor *item*

$\sum Y$ = Jumlah skort total (seluruh *item*)

n = Jumlah Responden

2.2.11 Uji Reliabilitas

Uji realibilitas adalah untuk mengetahui sejauh mana hasil pengukuran tetap konsisten apabila dilakukan pengukuran dua kali atau lebih terhadap gejala yang sama dengan menggunakan alat pengukur yang sama pula (Yusup et al., 2018). Pengujian realibilitas menggunakan program *SPSS*. Kriteria suatu instrumen penelitian dikatakan *reliable*, bila koefisien reliabilitas (r) $> 0,6$ (Hutomo et al., 2015) . Uji reliabilitas mampu dilakukan dengan cara eksternal atau dengan cara internal. Uji reliabilitas secara eksternal mampu dilakukan dengan test-retest, equivalent, serta gabungan dari keduanya. Rumus untuk uji reliabilitas yaitu seperti pada persamaan (Belajar et al., 2020) :

$$r = \left(\frac{k}{k-1} \right) \left(1 - \frac{\sum \sigma b^2}{\sigma t^2} \right) \quad (1.2)$$

Keterangan:

r = Reabilitas instrument

σt^2 = Varians Total

K = Jumlah variabel

$\sum \sigma b^2$ = Jumlah varian

2.2.12 Kano Model

Menurut *Kano et al.* (1984) dalam (Matzler et al., 2004), pandangan tradisional yang *linear* antara tingkat performa layanan yang diberikan dengan peningkatan kepuasan konsumen tidak bisa selamanya benar. Terkadang, kepuasan pelanggan dapat menunjukkan pola yang *non-linear* dan karena ketidak*linearan* inilah kualitas layanan yang diberikan tidak selalu sesuai dengan harapan atau dengan kata lain tidak selalu menghasilkan kepuasan. Sehingga *Kano* dikelompokkan menjadi tiga jenis dari persyaratan atau kebutuhan layanan yang mempengaruhi kepuasan pelanggan yaitu *mustbe*, *one-dimensional*, dan *attractive requirement*. Perlu diketahui bahwa atribut *attractive* memberi pengaruh terbesar terhadap kepuasan pelanggan.

Menurut *Kano et al.* (1984) 3 jenis tingkat atribut layanan yang berpengaruh terhadap kepuasan pelanggan yaitu:

1. *Attractive* (Menarik), merupakan kategori layanan yang tidak diharapkan dan disampaikan langsung oleh konsumen, namun ketika layanan ini disediakan maka akan menghasilkan tingkat kepuasan yang sangat tinggi atau melejit bagi konsumen.
2. *One-Dimensional* (Satu Dimensi), merupakan kategori layanan konsumen yang dituntut konsumen untuk dipenuhi. Tingkat kepuasan pelanggan pada tipe ini sebanding dengan kinerja suatu layanan. Semakin bagus kinerja layanan, semakin tinggi pula tingkat kepuasan pelanggan, dan sebaliknya.
3. *Must-Be* (Harus Ada), merupakan kategori layanan yang jika tidak dipenuhi, maka pelanggan akan sangat kecewa dan kemungkinan besar tidak akan mau untuk menggunakan layanan tersebut lagi. Namun, jika layanan ini dipenuhi juga tidak akan selalu berdampak pada kepuasan konsumen

Kemudian, respon konsumen terhadap persyaratan atau requirement dari layanan adalah sebagai berikut:

1. *Indifferent*, merupakan kategori dimana ada atau tidak adanya layanan tersebut tidak mempengaruhi kepuasan pelanggan.
2. *Reverse*, yaitu kategori dimana konsumen akan lebih puas ketika layanan kategori ini tidak diberikan dan kepuasan konsumen akan menurun ketika layanan tersebut diberikan.
3. *Questionable*, merupakan kategori dimana konsumen tidak konsisten atau ragu akan layanan yang diberikan, kadangkala konsumen puas/ tidak puas jika layanan itu diberikan atau tidak diberika

BAB 3

METODE PENELITIAN

3.1 Objek dan Subjek Penelitian

Objek penelitian ini adalah persepsi pengunjung pelayanan Pariwisata Candi Borobudur melalui *review* pada database *website tripadvisor*. Sedangkan subjek pada penelitian ini merupakan pengunjung Pariwisata Candi Borobudur.

3.2 Populasi dan Sample Penelitian

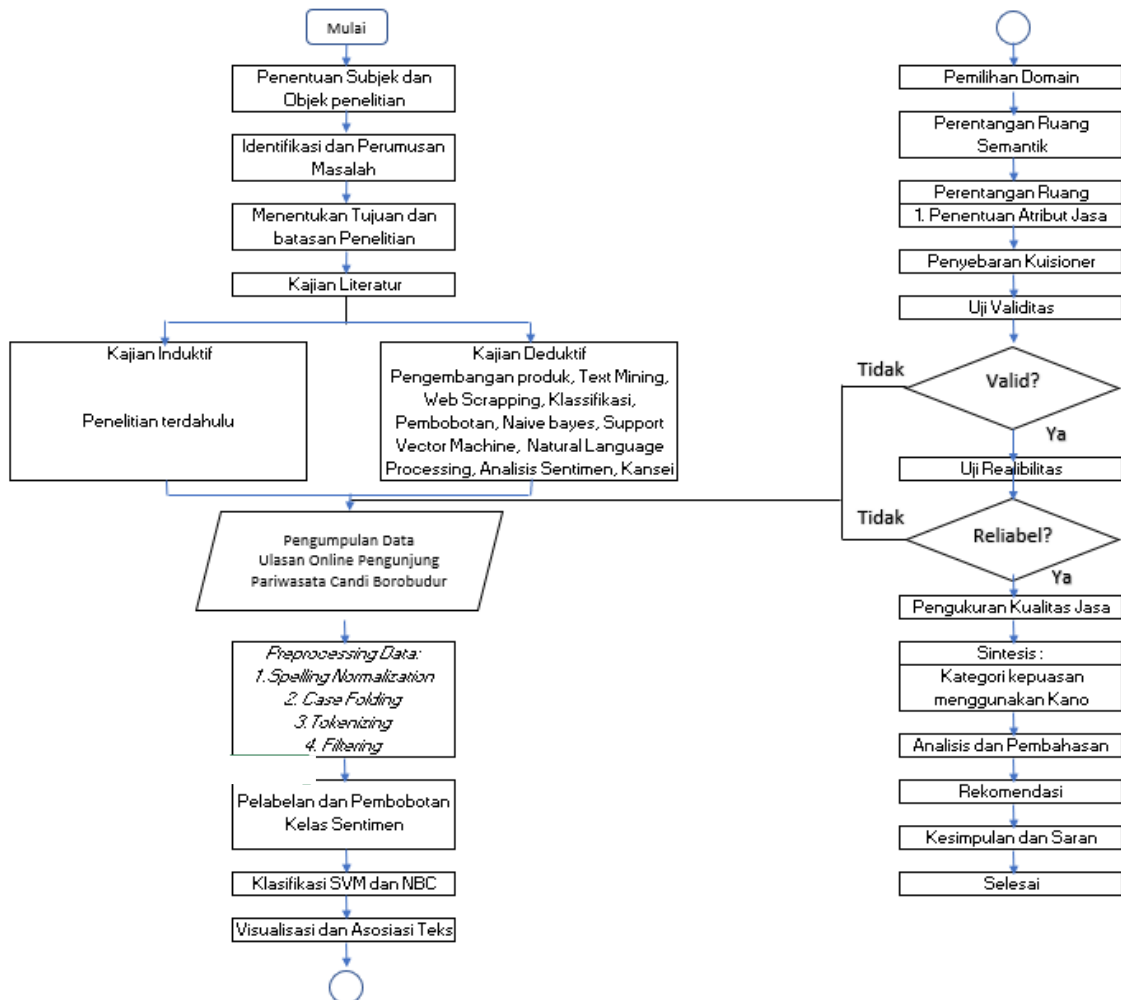
Populasi dalam penelitian ini adalah semua ulasan atau *review* pengunjung pariwisata Candi Borobudur dari *database website tripadvisor*. Sedangkan untuk sampel yang digunakan adalah ulasan pengunjung pada tanggal 1 Januari 2019 – 30 Agustus 2021.

3.3 Metode Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah data primer dan data sekunder. Data primer yang digunakan dalam penelitian ini berupa data yang didapatkan dari ulasan Candi Borobudur yang didapatkan dari *website tripadvisor* dengan bantuan aplikasi *WebHarvy*. Pengambilan Kuisisioner juga dilakukan dalam pengambilan data primer. Sedangkan untuk data sekunder didapatkan melalui studi literatur terhadap buku dan *journal* penelitian terdahulu.

3.4 Tahap Penelitian

Berikut merupakan alur penelitian yang menunjukkan tahapan penelitian dari awal hingga akhir dengan ditunjukkan oleh Gambar 3.1



Gambar 3.1 Alur Penelitian

Berdasarkan gambar alur penelitian, berikut merupakan langkah-langkah pada penelitian ini:

1. Mulai
2. Penentuan Objek dan Subjek Penelitian

Proses penelitian ini diawali dengan menentukan Objek dan Subjek penelitian yang akan dilakukan. Objek penelitian ini adalah persepsi pengunjung pelayanan Pariwisata Candi Borobudur melalui *review* pada database *website tripadvisor*. Sedangkan subjek pada penelitian ini merupakan pengunjung Pariwisata Candi Borobudur.

3. Identifikasi dan Perumusan Masalah

Pada tahap ini disajikan informasi terkait keadaan Candi Borobudur serta alasan dilakukan penelitian ini dan dilakukan perumusan masalah berupa bagaimana ekstraksi data *review online* untuk mendapatkan pengetahuan dan metode ekstraksi terbaik guna mendukung desain produk jasa dari pariwisata Candi Borobudur menggunakan metode *NLP* dan *Kansei*.

4. Menentukan Tujuan Penelitian

Pada tahap ini dilakukan penentuan tujuan penelitian. Tujuan penelitian ini menjelaskan hal-hal yang ingin dicapai dalam penelitian ini. Adapun tujuan dalam penelitian ini adalah mengetahui ekstraksi data *review* yang diberikan oleh pelanggan sehingga pariwisata Candi Borobudur dapat berkembang sesuai dengan kepuasan konsumen.

5. Menentukan Bahasan Penelitian

Supaya penelitian yang dilakukan lebih terarah dan topik yang dibahas tidak meluas, maka perlu dilakukan pembatasan lingkup penelitian. Adapun pembatasan lingkup penelitian ini adalah :

- a. Data yang digunakan merupakan data ulasan mengenai kunjungan Candi Borobudur pada situs *tripadvisor*.
- b. Data ulasan yang digunakan dalam penelitian yaitu dua ulasan mulai 1 Januari 2019 – 1 September 2021.
- c. Ekstraksi data yang digunakan sebagai *Kansei words* hanya menggunakan hasil *NLP*
- d. Pengembangan produk menggunakan metode integrasi *Kansei* dengan metode *kano*.

6. Kajian Literatur

Pada tahap ini dilakukan studi pustaka yang berkaitan dengan topik analisis sentimen untuk dapat mengetahui metode yang sesuai untuk penelitian ini. Terdapat beberapa topik yang digunakan dalam studi pustaka, seperti *Text mining*, *NLP*, *Kansei Engineering*, *Kano*.

7. Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data dengan melakukan pengambilan kuisioner dan esktraksi ulasan secara *online* dengan teknik *scraping data* dengan bantuan aplikasi

webharvy. Data yang digunakan dalam ekstraksi yaitu berupa ulasan pengunjung Pariwisata Candi Borobudur pada *website Trip Advisor*.

8. *Pre-processing* Data

Pre-processing data merupakan tahap awal dalam proses pengolahan data. Proses ini dilakukan untuk menyeleksi data dan mengubahnya menjadi data yang lebih terstruktur dengan bantuan *software python*. Terdapat beberapa proses yang dilakukan dalam tahap ini, seperti:

- a. *Spelling Normalization* : memperbaiki kata-kata yang disingkat maupun salah ejaan.
- b. *Case Folding* : tahap mengubah teks menjadi lowercase, menghapus angka, dan menghapus tanda baca
- c. *Tokenizing* : proses pemisahan teks ulasan menjadi kata per kata yang tidak saling berpengaruh (independent)
- d. *Filtering* : proses penyaringan atau pemilihan kata dalam dokumen.

9. Pelabelan dan Pembobotan Kelas Sentimen

Tahap ini diawali dengan menghitung jumlah skor setiap ulasan berdasarkan *lexicon*. Setelah itu akan dilakukan pelabelan kelas dengan ketentuan, apabila skor ulasan kurang dari 0 maka termasuk kelas negatif, sama dengan 0 maka termasuk kelas netral, dan lebih dari 0 maka termasuk kelas positif.

10. Klasifikasi *SVM* dan *NBC*

Data yang telah memiliki label kelas, akan digunakan untuk proses klasifikasi. Proses ini dilakukan untuk membentuk model yang akan digunakan untuk mengklasifikasikan data baru. Proses ini dilakukan dengan dua metode, yaitu *Support vector machine (SVM)* dan *Naïve Bayes Classifier (NBC)* dan dilakukan dengan bantuan *software python*.

11. Visualisasi dan Asosiasi Teks

Tahap ini dilakukan dengan mengekstraksi seluruh informasi sehingga dapat diketahui kata yang paling sering dibahas pada ulasan pengguna. *Output* dari tahap ini nantinya berupa hasil asosiasi dari kata yang muncul.

12. Pemilihan Domain

Tahap ini mencakup identifikasi kelompok sasaran, pasar, populasi tertentu dan spesifikasi dari produk atau jasa baru dan inovatif. Dalam penelitian ini ditentukan berupa Tempat pariwisata yang nyaman digunakan oleh pengunjung.

13. Perentangan Ruang Semantik.

Tahap ini dilakukan melalui tiga langkah utama yaitu i) mengumpulkan *Kansei word*, ii) memilih *Kansei word*, dan iii) menyatukan data. *Kansei word* dapat diperoleh dengan mengembangkan serangkaian metode pengukuran melalui penafsiran/interpretasi yang didapatkan dari pengolahan data menggunakan *NLP* sebelumnya.

14. Perentangan Ruang Properti

Tahap ini bertujuan untuk mengetahui apakah sebuah elemen produk secara individual atau puluhan elemen dapat memiliki pengaruh besar terhadap *Kansei* yang akan diterapkan, diperlukan identifikasi semua atribut produk dan hanya dipilih yang memberikan dampak terbesar/potensial pada pengembangan Candi Borobudur.

15. Penyebaran Kuisisioner

Tahap ini dilakukan penyebaran kuisisioner yang berisi tentang pengukuran kualitas jasa dan metode *kano*. dilakukan pengambilan data ke 50 responden dengan kriteria pernah mengunjungi Candi Borobudur.

16. Perhitungan *Service Quality*

Tahap ini dilakukan pengukuran kualitas jasa (Q) yang di berikan oleh pihak Candi Borobudur berdasarkan hasil Harapan dan Presepsi responden kepada Candi Borobudur dari kuisisioner yang telah disebarkan sebelumnya.

17. Sintesis

Pada tahap ini dilakukan penggabungan ruang semantik dan ruang properti. Untuk setiap *Kansei word*, sejumlah properti produk diidentifikasi. Dengan kata lain, *Kansei* adalah fungsi dari sifat dan kesan interaksi dengan produk atau jasa. KE tipe pertama (KE Identifikasi Kategori) dalam melakukan identifikasi kategori menggunakan metode *kano*.

18. Analisis dan Pembahasan

Pada tahap ini dilakukan proses pengolahan data tahap akhir berupa penentuan kategorisasi atribut jasa untuk keperluan pengembangan jasa yang ditawarkan oleh pihak Candi Borobudur berdasarkan hasil dari Integrasi *Kano* dan *Kansei*.

19. Rekomendasi

Pada tahap ini diberikan rekomendasi terhadap hasil dari kategori yang telah ditentukan. Rekomendasi yang diberikan hanya atribut yang termasuk kedalam kategori *Must Be* dan *One Dimensional*.

20. Kesimpulan dan Saran

Pada tahap ini berisi kesimpulan dan saran dari penelitian yang dilakukan. Kesimpulan penelitian ini akan menampilkan beberapa hasil penelitian dan jawaban dari rumusan masalah. Pada tahap ini juga ditampilkan perbaikan dan peningkatan terkait penelitian terkait pengembangan Candi Borobudur kedepannya.

21. Selesai.

BAB 4

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Pemilihan Service Domain

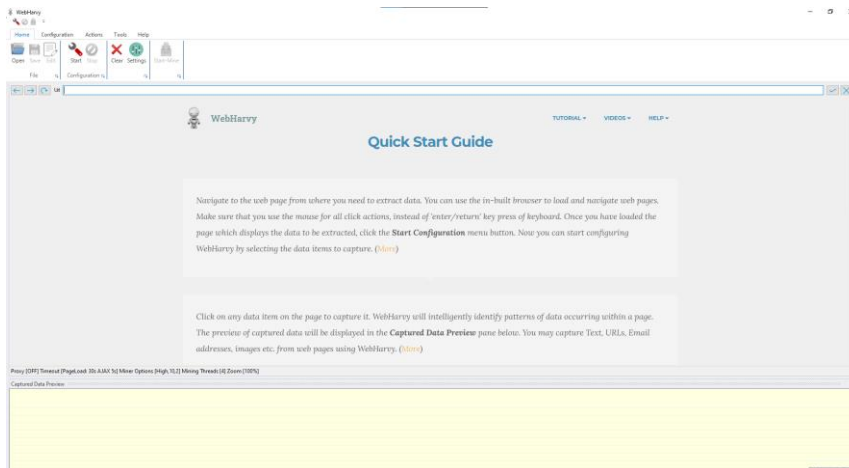
Berdasarkan jenisnya, pariwisata dapat dikelompokkan menjadi beberapa jenis, yaitu wisata budaya, wisata kesehatan, wisata olahraga, wisata komersial, wisata industri, wisata bahari, dll. (Pendit, 2012). Dikarenakan Candi Borobudur termasuk kedalam wisata budaya, maka domain yang dipilih dalam penelitian ini adalah kualitas pelayanan yang disajikan pada wisata budaya.

4.2 Perentangan Ruang Semantik

Perentangan ruang semantic dilakukan melalui tiga langkah utama, yaitu pengumpulan data ulasan, ekstraksi data ulasan, dan pemilihan *Kansei words*

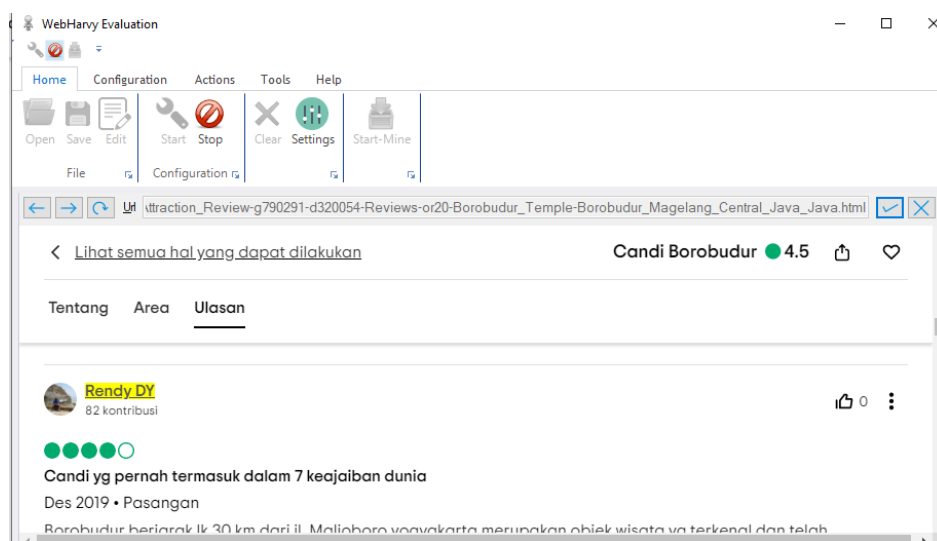
4.2.1 Pengumpulan data ulasan

Pengumpulan *Kansei words* dilakukan dengan cara *data mining* menggunakan metode *Natural Language Processing (NLP)* dengan menggunakan sentimen negatif. Sumber yang digunakan dalam proses *data mining* merupakan ulasan pengunjung pariwisata Candi Borobudur yang didapat dari situs *tripadvisor*. Pengambilan data dilakukan menggunakan teknik scraping dengan memanfaatkan aplikasi *Web Harvy*. *Web Harvy* merupakan aplikasi untuk melakukan scrapping yang dapat diunduh secara gratis. Pada penelitian ini digunakan *Web Harvy* versi 6.2.0.185 untuk mengambil data ulasan yang selanjutnya diimpor menjadi spreadsheet *Microsoft Excel* atau *CSV*.



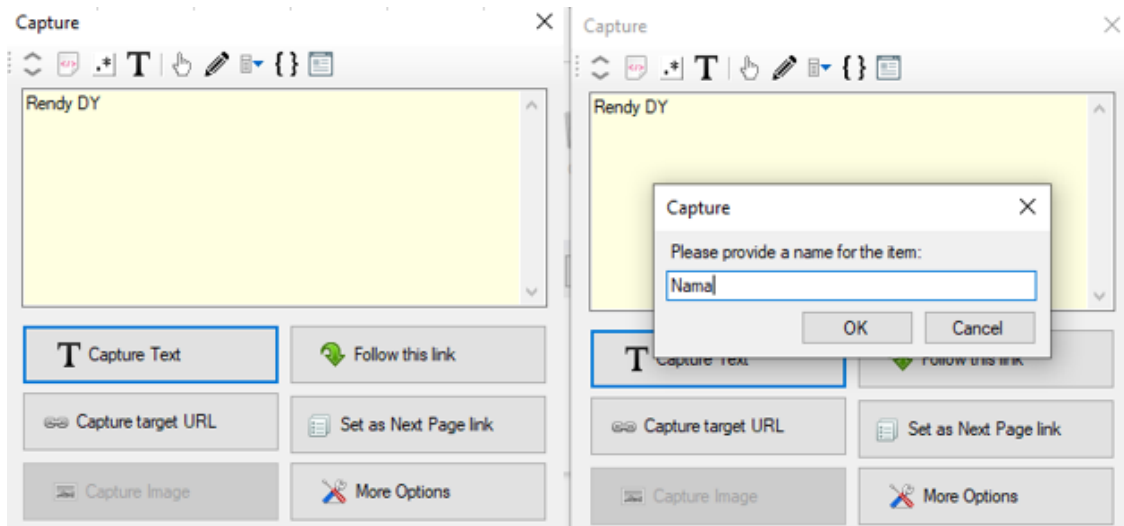
Gambar 4. 1 Web Harvy

Langkah awal yang dilakukan dalam pengambilan data yaitu membuka aplikasi *Web Harvy* yang dapat diunduh melalui <https://www.webharvy.com/>. Setelah itu masukkan alamat domain atau *website* yang ingin di scrapping, dalam hal ini yaitu *tripadvisor*.



Gambar 4. 2 Penggunaan *Web Harvy* 1

Langkah selanjutnya menentukan data apa saja yang diperlukan dalam penelitian, dalam penelitian ini digunakan data ulasan berupa Nama, tanggal, serta ulasan pengunjung. Untuk mendapatkannya, lakukan klik kiri pada nama pengunjung, maka akan muncul tampilan *window "capture"*



Gambar 4. 3 Penggunaan Web Harvy 2

Langkah terakhir melakukan *scrapping data*, dengan melakukan klik pada icon “*start mine*” maka akan muncul tampilan jendela baru, jumlah data yang akan dilakukan scrapping tergantung pada isian “*number of page*”. Dalam penelitian ini diambil data sebanyak 3294 ulasan

4.2.2 Ekstraksi data ulasan

Dalam melakukan ekstraksi data ulasan, dilakukan pengolahan data ulasan terlebih dahulu dengan melakukan *pre processing* data, klasifikasi data lalu asosiasi data.

1. *Pre Processing* Data

Tahap *Pre-processing* data, digunakan *dataset* berupa data ulasan yang diambil dari ulasan pengunjung Candi Borobudur pada *tripadvisor* yang memiliki berbagai macam gaya penulisan sehingga data yang diperoleh merupakan data yang tidak terstruktur. Sehingga diperlukan sebuah proses supaya data tersebut menjadi data yang lebih terstruktur sebelum dilakukan klasifikasi. Tahapan untuk mengubah data yang tidak terstruktur menjadi data yang lebih terstruktur disebut tahap *pre-processing*. Tujuan dilakukannya *pre-processing* ini untuk menyeragamkan kata dan mengurangi *volume* kata yang tidak perlu. Tahap *pre-processing* dilakukan menggunakan aplikasi *pyhton* versi 3.10. Adapun langkah-langkah *pre-processing* yang akan dilakukan adalah melakukan *Normalization*, *Casefolding*, *Tokenizing*, dan *Filtering*.

A. Normalization

Proses *spelling* normalization dilakukan untuk memperbaiki kata-kata yang disingkat maupun salah ejaan dengan bentuk berbeda namun memiliki maksud yang sama, sebagai contoh pada kata “sampai” memiliki beberapa bentuk penulisan, seperti “smp”, “sampe”, dan “smpe”. Berikut merupakan contoh proses *spelling* normalization:

Tabel 4. 1 *Input Output Normalization*

<i>Input</i>	<i>Output</i>
<p>tak ada bosenya wisata kesini dari jaman sd sampe sekarang perubahannya bagus toilet umum suda banyak dan persewaan payung juga tersedia harga tiket lokal hanya dan banyak foodcourt juga yang bisa ditemui di dlm area candi</p>	<p>tidak ada bosan wisata kesini dari zaman sd sampai sekarang perubahannya bagus toilet umum sudah banyak dan persewaan payung juga tersedia harga tiket lokal hanya dan banyak foodcourt juga yang bisa ditemui di dalam area candi</p>

Adapun *Pseudocode* pada proses *Tokenizing* seperti tabel 4.2 dibawah ini.

Tabel 4. 2 *Pseudocode* yang digunakan pada *Normalization*

<i>Pseudocode</i>	Fungsi
<pre>def normalized_term(document): return [normalizad_word_dict[term] if term in normalizad_word_dict else term for term in document]</pre>	<p>Menseragamkan berbagai macam kata baik singkatan dan non baku menjadi kata baku tanpa singkatan.</p>

B. Case Folding

Case Folding merupakan salah satu tahapan *pre-processing* yang akan dilakukan dengan tujuan untuk menyeragamkan bentuk huruf atau kata kedalam bentuk huruf kecil serta mengubah semua huruf dalam dokumen menjadi hanya huruf ‘a’ sampai dengan ‘z’ yang diterima. Karakter selain huruf dihilangkan dan dianggap delimiter seperti angka, tanda baca, dan *whitespace*. Berikut merupakan proses *Case Folding* yang dilakukan dapat dilihat pada tabel 4.3

Tabel 4. 3 *Input Output Casefolding*

<i>Input</i>	<i>Output</i>
<u>Lingkungan</u> di dalam candi benar bersih dan petugas juga sangat cekatan hanya masyarakat yang naik ke candi banyak sekali yang nakal dan duduk diatas candi padahal sudah jelas ada larangan duduk !.	<u>lingkungan</u> di dalam candi benar bersih dan petugas juga sangat cekatan hanya masyarakat yang naik ke candi banyak sekali yang nakal dan duduk diatas candi padahal sudah jelas ada larangan duduk

Adapun *Pseudocode* pada proses *CaseFolding* seperti tabel 4.4 dibawah ini.

Tabel 4. 4 *Pseudocode* yang digunakan pada *Casefolding*

<i>Pseudocode</i>	Fungsi
<pre>def Cleansing (data): data = data.lower() remove = string.punctuation translator = str.maketrans(remove, ' '*len(remove)) data = data. translate (translator) angber = ".join([x for x in data if not (x.isdigit()or x[0] == '-' and x[1:].isdigit())]) return angber</pre>	Mengganti huruf kapital menjadi huruf kecil dan menghilangkan tanda baca serta <i>whitespace</i> pada kalimat yang ada.

C. Tokenizing

Tokenizing metode untuk melakukan pemisahan kata dalam suatu kalimat dengan tujuan untuk proses analisis teks lebih lanjut. Kata- kata tersebut menggunakan tanda koma (,) sebagai pemisahannya. Berikut merupakan contoh penerapan *tokenizing* pada kalimat ulasan yang dapat dilihat pada tabel 4.5.

Tabel 4. 5 *Input Output Tokenizing*

<i>Input</i>	<i>Output</i>
Candi Borobudur indah bangetcandinya besarcukup jauh menuju candinya dari tempat parkir kendaraandisarankan untuk membaya payung jika ta ingin kepanasannyampe atas ngga bakal kecewa kokterbayar sudah bagusnya candi kyk apa	['candi', 'borobudur', 'indah', 'bangetcandinya', 'besarcukup', 'jauh', 'menuju', 'candinya', 'dari', 'tempat', 'parkir', 'kendaraandisarankan', 'untuk', 'membaya', 'payung', 'jika', 'ta', 'ingin', 'kepanasannyampe'. 'atas', 'ngga', 'bakal', 'kecewa', 'kokterbayar', 'sudah', 'bagusnya', 'candi', 'kyk', 'apa'.]

Adapun *Pseudocode* pada proses *Tokenizing* seperti tabel 4.4 dibawah ini.

Tabel 4. 6 *Pseudocode* yang digunakan pada *Tokenizing*

<i>Pseudocode</i>	Fungsi
<i>def tokenize (txt):</i> <i>word_token = word_tokenize (txt)</i> <i>return word_token</i>	Mengubah kalimat kedalam bentuk yang dipisah dengan tanda ' di setiap kata.

D. Filtering

Tahap *Filtering* merupakan tahapan untuk mengambil kata-kata penting dari hasil token.dalam tahap *pre-processing* yang dilakukan, menggunakan algoritma *stoplist* dan *wordstop*. Didalam *wordstop* sebelumnya diperlukan kamus *stopword* yang berisi tentang sejumlah kata yang sering digunakan dalam *stoplist*. sedangkan *Stoplist* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words*. Contoh *stopwords* adalah “ada”, “atau”,”dan”, dan seterusnya.

Tabel 4. 7 *Input Output Filtering*

<i>Input</i>	<i>Output</i>
['saya', 'berlibur', 'ke', 'jogja', 'selama', 'seminggu', 'dan', 'salah', 'satu', 'wisata', 'tujuan', 'saya', 'itu', 'borobudur', 'keren', 'banget', 'disini', 'seluruh', 'fasilitas',	['libur', 'keren','baik']

<i>Input</i>	<i>Output</i>
'sangat', 'baik', 'keren', 'banget', 'deh', 'pokoknya']	

Adapun *Pseudocode* pada proses *Tokenizing* seperti tabel 4.8 dibawah ini.

Tabel 4. 8 *Pseudocode Filtering*

<i>Pseudocode</i>	Fungsi
<i>list_stopwords</i> <i>stopwords.words('indonesian')</i>	= Menghapus kata yang ada didalam stoplist.
<i>list_stopwords.extend("yg", 'menjual', 'barang', "dg" dst)</i>	
<i>def stopwords_removal(words):</i> <i> return [word for word in words if word not</i> <i> in list_stopwords]</i>	

2. Pembobotan *TF IDF*

Data ulasan yang sudah melewati tahap *text pre-processing* masih berbentuk teks (kata), sedangkan dalam Algoritma pembobotan kata didasarkan bahwa teks perlu dikonversi menjadi angka sebelum dapat diterapkan ke dalam algoritma matematika dikenal sebagai *vectorization*. pada penelitian ini digunakan pembobotan *TF-IDF* sebagai teknik konversi teks menjadi angka. Proses pembobotan kata dengan *TF-IDF* dilakukan dengan *script* sebagai berikut dan menampilkan term yang memiliki bobot terbesar, hasil *output script* ditampilkan pada tabel berikut

Tabel 4. 9 *Pseudocode* Pembobotan

<i>Pseudocode</i>	Fungsi
<i>from sklearn.feature_extraction.text import</i> <i>CountVectorizer</i>	Memberi bobot pada setiap kata dalam sebuah kalimat ulasan.
<i>sklearn.feature_extraction.text</i> <i>TfidfTransformer</i> <i>cvec</i> =	
<i>CountVectorizer(stop_words='english',</i>	

<i>Pseudocode</i>	Fungsi
<pre> min_df=1, max_df=.5 , ngram_range=(1,2) transformer = TfidfTransformer() transformed_weights = transformer.fit_transform(cvec_counts) transformed_weights </pre>	

Berikut merupakan hasil pembobotan kata menggunakan *TF-IDF* berdasarkan lima kata (*term*) teratas :

Tabel 4. 10 Hasil Pembobotan

Term	TF	<i>TF-IDF</i>
candi	0.0338	0.2788
letak	0.0338	0.0987
cuaca	0.0169	0.0502
hujan	0.0169	0.0545
ramai	0.0169	0.0403

3. Perhitungan Skor Sentimen

Setelah dilakukan pembobotan di setiap kata, lalu dilakukan perhitungan skor dalam kalimat ulasan. Pada umumnya terdapat dua metode yang digunakan dalam menentukan suatu teks mengandung kalimat positif atau negatif. metode pertama yaitu dilakukab dengan manual berupa membaca isi teks secara keseluruhan, kemudian memberikan label. Metode ini memiliki presisi yang cukup tinggi namun sangat sulit diterapkan untuk data yang berjumlah sangat besar karena memakan banyak waktu, metode kedua yaitu dengan menerapkan algoritma yang dapat mengekstrak kalimat opini secara otomatis. Salah satu algoritma yang umum digunakan adalah *lexicon based*. *Lexicon based* dapat mengekstrak kalimat opini dengan presisi yang sangat tinggi (Azhar, 2018).

Cara kerja metode *Lexicon Based* yaitu dengan membuat kamus kata opini (*lexicon*) terlebih dahulu kedalam sebuah *file*. *File* tersebut berisi beberapa kata, bobot, dan jumlah kata yang terkandung. Sehingga dengan menggunakan *file* tersebut dapat

mengidentifikasi kata positif dan negatif pada suatu kalimat. Rumus perhitungan skor sentimen yang digunakan pada tahap ini adalah sebagai berikut ini:

$$Skor\ Sentimen = \sum Bobot\ Kata\ Positif + \sum Bobot\ Kata\ Negatif \quad (4.1)$$

Sebagai contoh perhitungan skor sentimen tertera pada tabel 4.11

Tabel 4. 11 Perhitungan Skor Sentimen

Ulasan	Kata Positif	Kata Negatif
Pemandangan indah , tempat nya bagus , bersejarah , bisa belajar bahasa asing juga, banyak object kunjungan, namun banyak sampah	“indah”(2), “bagus”(1), “bersejarah”(2), “belajar”(1).	“Sampah”(-3)
Jumlah	6	-3

Berdasarkan ulasan pada tabel 4. diketahui bahwa terdapat 4 kata positif dan 1 kata negatif berdasarkan kamus *lexicon*. Kata positif yang terdeteksi yaitu “indah”, “bagus”, “bersejarah”, “belajar” sedangkan kata negatif yaitu “sampah”. Oleh karena itu perhitungan skor sentimennya sebagai berikut:

$$Skor = 2+2+1+1 - 3 = 3 \quad (4.2)$$

Berdasarkan hasil diatas, maka kalimat tersebut termasuk kedalam ulasan positif dikarenakan skor yang didapatkan lebih dari 0. Dalam penelitian ini, perhitungan skor sentimen dilakukan dengan bantuan aplikasi *python* dengan bantuan library “*nltk*” yang umum digunakan dalam proses *text mining*. Adapun *pseudocode* yang digunakan dalam proses ini adalah sebagai berikut:

Tabel 4. 12 *Pseudocode* Skor Sentimen

<i>Pseudocode</i>	Fungsi
<i>def found_word(ind, words, word, sen, sencol, Sentiment, add):</i>	1. Membuat
<i>if word in sencol:</i>	model <i>lexicon</i>
<i>sen[sencol.index(word)] += 1</i>	2. Melakukan
<i>else:</i>	<i>crosscheck</i>

<i>Pseudocode</i>	Fungsi
<pre> 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(data)): nsen = senrow.shape[0] words = word_tokenize(data['data_kembali'][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) if word in lexicon_word : sen,sencol,Sentiment,add= found_word(ind,words,word,sen,sencol,Sentiment,add) else: 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>1): if ind-1>-1: </pre>	<p>terhadap kata dalam kamus <i>lexicon</i> dengan yang ada pada data ulasan yang tersedia</p> <p>3. Menghitung total skor yang didapatkan</p>

<i>Pseudocode</i>	Fungsi
<pre> 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 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)) elif(nsen>0): senrow = np.vstack((senrow,sen)) Sentiment_list.append(Sentiment) </pre>	

4. Pelabelan Kelas Sentimen

Pada dasarnya, proses pelabelan dibagi menjadi tiga kelas sentimen, yaitu sentimen positif, sentimen negatif dan sentimen netral. Dalam penilaian ulasan digunakan sentimen positif dan negatif karena dalam penggunaan sentimen netral dianggap tidak

mempengaruhi ulasan. Penentuan sentimen ditentukan dengan memanfaatkan hasil skor yang telah dilakukan sebelumnya, selanjutnya dilakukan pelabelan secara otomatis menggunakan bantuan aplikasi *python* dengan cara menghitung skor jumlah kata positif dikurangi dengan skor jumlah kata negatif dalam suatu kalimat ulasan. Jika suatu kalimat memiliki skor > 0 akan diklasifikasikan dalam kelas positif, jika kalimat memiliki skor $= 0$ akan diklasifikasikan dalam kelas netral, sedangkan jika kalimat memiliki skor < 0 diklasifikasikan dalam kelas negatif. Berikut merupakan hasil perbandingan jumlah data dari pelabelan kelas sentimen yang diperoleh:

Tabel 4. 13 Jumlah Sentimen

Jenis <i>Sentiment</i>	Jumlah Ulasan
Positif	2609
Negatif	393
Netral	292

Klasifikasi data pada penelitian ini dibagi menjadi sentimen positif, sentimen negatif dan sentimen netral, akan tetapi data yang digunakan hanya menggunakan data dengan sentimen positif dan Sentimen negatif. Hal ini dikira untuk kelas sentimen netral kurang memberikan masukan dan manfaat bagi pihak borobudur. Pada ulasan yang mengandung pernyataan positif seperti kata-kata pujian dan lain sebagainya dimasukkan kedalam kategori sentimen positif. Sedangkan untuk ulasan yang mengandung pernyataan negatif seperti kritikan, ketidakpuasan dan sebagainya dimasukkan kedalam kategori sentimen negatif.

5. Analisis Klasifikasi

Setelah melakukan pelabelan kelas sentimen, dilakukan proses pengolahan data berupa analisis klasifikasi. Hasil data pelabelan dibagi menjadi dua, yaitu data *training* dan data *testing*. Algoritma yang digunakan dalam proses klasifikasi yaitu *Support vector machine (SVM)* dan *Naïve Bayes Classifier (NBC)*.

6. Pembuatan Data Latih dan Data Uji

Data *training* digunakan oleh algoritma klasifikasi untuk membentuk sebuah model *classifier*, model ini merupakan representasi pengetahuan yang akan digunakan untuk prediksi kelas data baru yang belum pernah ada, semakin besar data latih yang digunakan,

maka akan semakin baik *machine* dalam memahami pola data. Data *testing* digunakan untuk mengukur sejauh mana *classifier* berhasil melakukan klasifikasi dengan benar. Data yang digunakan untuk data *training* dan data *testing* adalah data yang telah memiliki label kelas, dengan jumlah data *training* dan data *testing* memiliki perbandingan 80 : 20. Berdasarkan *Pareto Principle*, Rasio yang umum digunakan adalah 80:20 untuk *data sets training* dan *testing*. Namun tidak menutup kemungkinan dalam sebuah penelitian tidak hanya menggunakan perbandingan tersebut. Hal tersebut dikarenakan jumlah data *training* akan mempengaruhi akurasi. Semakin banyak data *training* maka model akan semakin banyak belajar sehingga ketelitian akan semakin baik (Romario et al., 2020) terdapat tiga scenario atau percobaan yang digunakan dalam penelitian yaitu 70:30, 80:20 dan 90:10. Berikut merupakan perbandingan jumlah data *training* dan data *testing*:

1. Perbandingan data *training* sebesar 70 % dan data *testing* 30%

Tabel 4. 14 Pembagian Skenario 1

Klasifikasi	Jumlah	Data <i>training</i> (70%)	Data <i>testing</i> (30%)
Positif	2609	1826	783
Negatif	393	275	118
Total	3002	2101	901

Berdasarkan Tabel 4.13, terlihat bahwa perbandingan data *training* dan data *testing* sebesar 70% : 30% dari total 3002 ulasan, terdapat sebanyak 2101 ulasan sebagai data *training* dan 901 ulasan sebagai data *testing*.

2. Perbandingan data *training* sebesar 80 % dan data *testing* 20%

Tabel 4. 15 Pembagian Skenario 2

Klasifikasi	Jumlah	Data <i>training</i> (80%)	Data <i>testing</i> (20%)
Positif	2609	2087	522
Negatif	393	314	79
Total	3002	2401	601

Berdasarkan Tabel 4.14, terlihat bahwa perbandingan data *training* dan data *testing* sebesar 80% : 20% dari total 3002 ulasan, terdapat sebanyak 2401 ulasan sebagai data *training* dan 601 ulasan sebagai data *testing*.

3. Perbandingan data *training* sebesar 90 % dan data *testing* 10%

Tabel 4. 16 Pembagian skenario 3

Klasifikasi	Jumlah	Data <i>training</i> (90%)	Data <i>testing</i> (10%)
Positif	2609	2348	261
Negatif	393	353	40
Total	3002	2701	301

Berdasarkan Tabel 4.15, terlihat bahwa perbandingan data *training* dan data *testing* sebesar 90% : 10% dari total 3002 ulasan, terdapat sebanyak 2701 ulasan sebagai data *training* dan 301 ulasan sebagai data *testing*.

Berikut merupakan *pseudocode* untuk menjalankan pembagian data *testing* dan *training* pada tabel 4.16

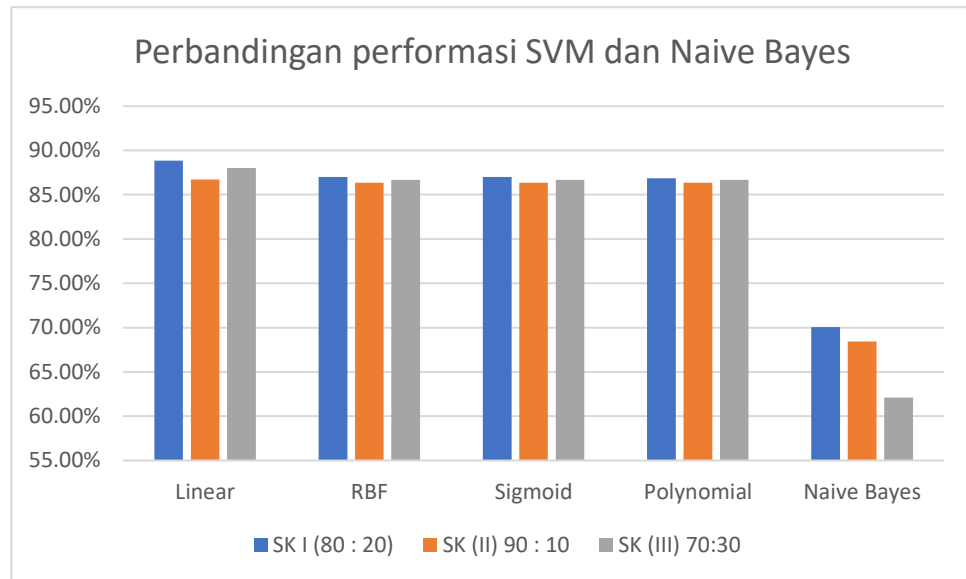
Tabel 4. 17 *pseudocode* pembagian data uji dan latih

<i>Pseudocode</i>	Fungsi
<code>from sklearn.model_selection import train_test_split</code>	Membagi data menjadi 80 : 20 (<i>training</i> : <i>testing</i>), 70:30, dan 90:10.
<code>X_train, X_test, y_train, y_test = train_test_split(x,y, test_size = 0.20, random_state = 0)</code>	
<code>X_train, X_test, y_train, y_test = train_test_split(x,y, test_size = 0.30, random_state = 0)</code>	
<code>X_train, X_test, y_train, y_test = train_test_split(x,y, test_size = 0.10, random_state = 0)</code>	

7. Klasifikasi Dengan Metode *Support vector machine* dan *Naive Bayes Classifier*

Setelah membagi *dataset* menjadi data *training* dan data *testing*, selanjutnya dilakukan klasifikasi menggunakan dua metode yang berbeda, yaitu *Support vector machine* (SVM) dan *Naive Bayes Classifier* (NBC). Dalam SVM, digunakan 4 jenis *kernel* yaitu *kernel*

Linear, Polynomial, Radial Basis Function (RBF), dan Sigmoid. Sedangkan dalam *NBC* hanya memiliki satu *kernel* saja. *Kernel* merupakan suatu fungsi yang memetakan data ke ruang dimensi yang lebih tinggi dengan harapan akan memiliki struktur yang lebih tinggi sehingga mudah dipisahkan (Jindal et al., 2016). Berikut merupakan hasil perbandingan dari keempat *kernel* yang telah dilakukan pengujian terhadap dua skenario yang telah dibuat:



Gambar 4. 4 Grafik Performansi *SVM dan Naive Bayes*

Berdasarkan hasil yang ditunjukkan pada Gambar 4.4 diatas, diketahui nilai akurasi untuk setiap algoritma *SVM* baik linier maupun non linier. Pada algoritma *SVM kernel linear* akurasi yang dihasilkan skenario 1 dengan data latih 80% dan data uji 20% sebesar 88.85%, skenario 2 dengan data latih 90 % dan data uji 10% sebesar 86.71% , skenario 3 dengan data latih 70% dan data uji 30% sebesar 88.01%. sedangkan pada *kernel* lain polynomial, sigmoid dan *RBF* memiliki nilai akurasi yang sama dengan nilai akurasi yaitu skenario 1 secara berturut turut adalah sebesar 86.85%;87.01%;87.01%, ketiga *kernel* pada skenario 2 dan 3 sebesar 86.37%, 86,68%. untuk klasifikasi *Naive Bayes* dengan satu jenis *kernel* yaitu secara berturut turut sebesar 70.05%, 68.43%, 62.12%.

Hasil perbandingan antara klasifikasi *SVM* dan *NBC* berdasarkan pada grafik 4. Menunjukkan bahwa performa *SVM* lebih unggul dibandingkan dengan *NBC*. Dengan akurasi terbesar yang didapatkan oleh *SVM* terletak pada *kernel linear* dengan hasil SK 1 sebesar 88.85%, dan *NBC* mempunyai hasil tertinggi pada Skenario 1 sebesar 70.05%.

dapat diketahui bahwa *kernel Linear* pada Skenario 1 (80:20) dengan data *testing* sebanyak memiliki akurasi tertinggi dibandingkan pengujian lainnya yaitu sebesar 88.85%. Oleh karena itu, untuk proses klasifikasi selanjutnya akan digunakan klasifikasi *SVM* dengan *kernel Linear* Skenario 80 % *data training* dan 20 % *data testing*.

Berikut merupakan hasil dari *confusion matrix*, dimana diketahui dengan pada kasus ini jumlah prediksi positif atau (*true positive*) sebesar 302 dan jumlah prediksi negatif yang sesuai (*true negative*) sebesar 232.

Tabel 4. 18 *Confusion Matrix*

		Aktual	
		Positif	Negatif
Prediksi	Positif	302	25
	Negatif	42	232
Akurasi		88.85%	

Confusion matrix menjelaskan seberapa besar presentase data yang diprediksi dengan benar oleh masing-masing percobaan pada data *testing*. Jumlah data observasi berkategori negatif yang mampu diprediksi negatif (diprediksi dengan benar) disebut *true* negatif. Jumlah data observasi berkategori positif yang mampu diprediksi negatif (diprediksi dengan benar) disebut *true* positif. Banyaknya data observasi yang berkategori positif akan tetapi diprediksi negatif disebut dengan *false* positif. Banyaknya data observasi yang berkategori negatif akan tetapi diprediksi negatif disebut dengan *false* negatif.

Berikut merupakan *pseudocode* untuk menjalankan *kernel* yang digunakan seperti pada tabel 4.19

Tabel 4. 19 *pseudocode* klasifikasi

<i>Pseudocode</i>	Fungsi
<i>SVM_Clasifier_kernel_linier = SVM.SVC(C=1.0, kernel='linear', degree=8, gamma='auto')</i>	1.Membuat model <i>kernel</i>
<i>SVM_Clasifier_kernel_linier.fit(features_train_transformed,y_train)</i>	<i>linear</i>

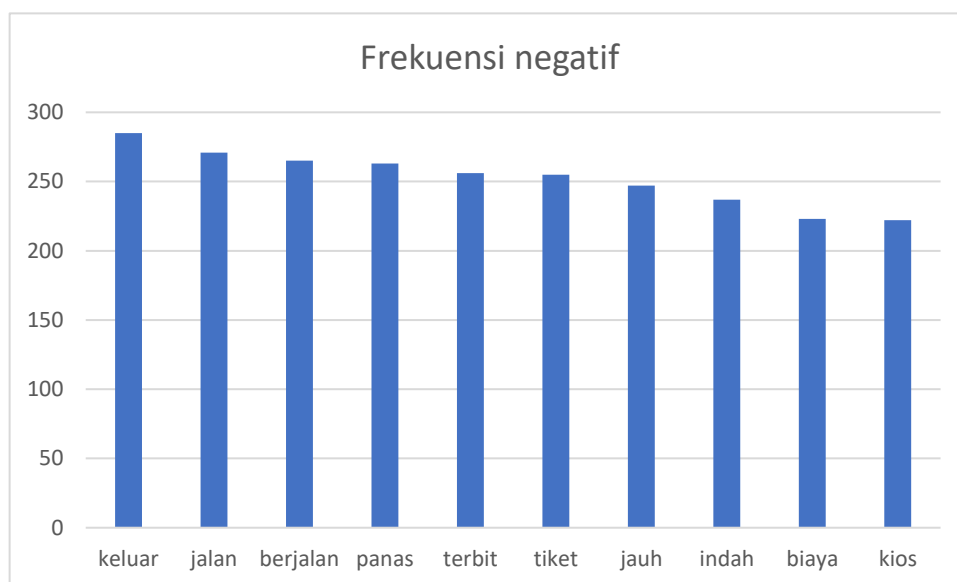
<i>Pseudocode</i>	<i>Fungsi</i>
<pre>SVM_Clasifier_kernel_poly = SVM.SVC(C=1.0, kernel='poly', degree=8,gamma='auto') SVM_Clasifier_kernel_poly.fit(features_train_transformed,y_train)</pre>	2.Membuat model kernel polynomial
<pre>SVM_Clasifier_kernel_sigmoid = SVM.SVC(C=1.0, kernel='sigmoid', degree=8, gamma='auto') SVM_Clasifier_kernel_sigmoid.fit(features_train_transformed,y_train)</pre>	3. Membuat model kernel sigmoid
<pre>SVM_Clasifier_RBF = SVM.SVC(C=1.0, kernel='RBF', degree=8, gamma='auto') SVM_Clasifier_RBF.fit(features_train_transformed,y_train)</pre>	4.Membuat model kernel RBF
<pre>from sklearn.metrics import classification_report,confusion_matrix,accuracy_score predictions_SVM = SVM_Clasifier.predict(features_test_transformer) print('linier') print(classification_report(y_test ,predictions_SVM)) print('Confusion \n',confusion_matrix(y_test,predictions_SVM)) print() print('Accuracy: ', accuracy_score(y_test,predictions_SVM)) from sklearn.metrics import classification_report,confusion_matrix,accuracy_score predictions_SVM SVM_Clasifier_kernel_linier.predict(features_test_transformer) print('kernel linier') print(classification_report(y_test ,predictions_SVM)) print('Confusion \n',confusion_matrix(y_test,predictions_SVM)) print() print('Accuracy: ', accuracy_score(y_test,predictions_SVM)) from sklearn.metrics import</pre>	5.Menjalankan fungsi kernel linear
	6.Menjalankan fungsi kernel polynomial
	7. Menjalankan fungsi kernel sigmoid
	8. Menjalankan fungsi kernel RBF
	9.Menjalankan fungsi kernel Naive Bayes.

<i>Pseudocode</i>	<i>Fungsi</i>
<pre> <i>predictions_SVM</i> = <i>SVM_Clasifier_kernel_poly.predict(features_test_transformer)</i> <i>print('polynomial')</i> <i>print(classification_report(y_test ,predictions_SVM))</i> <i>print('Confusion matrix:</i> \n',<i>confusion_matrix(y_test,predictions_SVM)</i>) <i>print()</i> <i>print('Accuracy: ', accuracy_score(y_test,predictions_SVM))</i> from <i>sklearn.metrics</i> import <i>classification_report,confusion_matrix,accuracy_score</i> <i>predictions_SVM</i> = <i>SVM_Clasifier_kernel_sigmoid.predict(features_test_transformer)</i> <i>print('sigmoid')</i> <i>print(classification_report(y_test ,predictions_SVM))</i> <i>print('Confusion matrix:</i> \n',<i>confusion_matrix(y_test,predictions_SVM)</i>) <i>print()</i> <i>print('Accuracy: ', accuracy_score(y_test,predictions_SVM))</i> from <i>sklearn.metrics</i> import <i>classification_report,confusion_matrix,accuracy_score</i> <i>predictions_SVM</i> = <i>SVM_Clasifier_RBF.predict(features_test_transformer)</i> <i>print('RBF')</i> <i>print(classification_report(y_test ,predictions_SVM))</i> <i>print('Confusion matrix:</i> \n',<i>confusion_matrix(y_test,predictions_SVM)</i>) <i>print()</i> <i>print('Accuracy: ', accuracy_score(y_test,predictions_SVM))</i> <i>print('Report:')</i> <i>print(classification_report(y_test, y_pred_nb))</i> <i>cmatrix_nb = pd.DataFrame(confusion_matrix(y_test, y_pred_nb),</i> </pre>	

<i>Pseudocode</i>	Fungsi
<pre> index = ['Actual {}'.format(i) for i in model_nb.classes_], columns = ['Predicted {}'.format(i) for i in model_nb.classes_],) print('\nConfusion matrix:') print(cmatrix_nb) </pre>	

8. Visualisasi Frekuensi Data

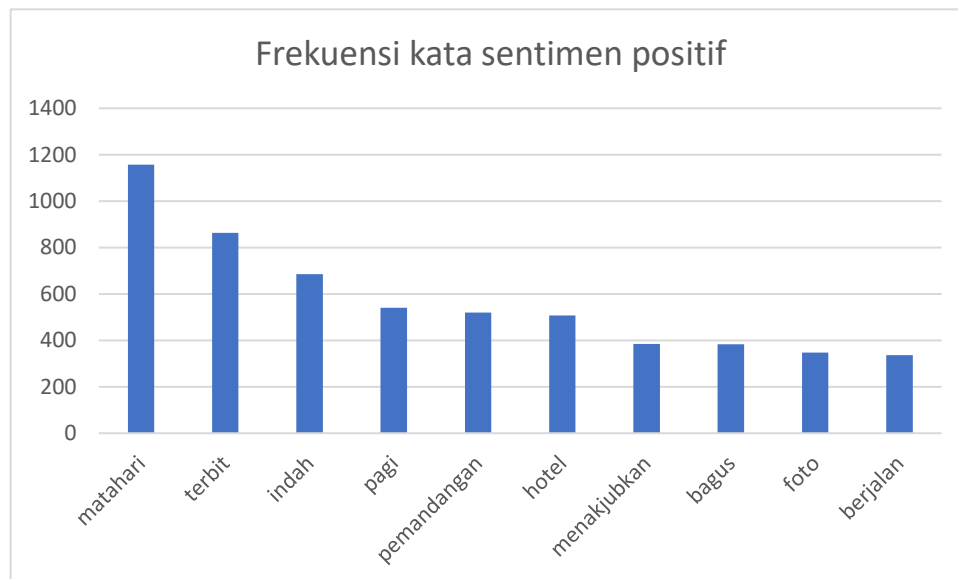
Dalam penelitian ini dilakukan visualisasi pada semua data yang ada baik itu masuk kelas sentimen negatif maupun positif hal ini bertujuan untuk mengekstrasi informasi secara keseluruhan tentang topik atau bahasan yang sering diulas oleh pengunjung Candi Borobudur. Selain itu juga dilakukan visualisasi pada masing-masing kelas sentimen untuk diambil informasi yang dianggap penting atau berguna serta dicari asosiasi antar kata pada masing-masing kelas *Sentiment* yang paling sering muncul secara bersamaan guna memperkuat pencarian informasi tersebut. Adapun visualisasi untuk sentimen negatif berdasarkan 10 kata teratas yaitu sebagai berikut:



Gambar 4. 5 Frekuensi Muncul Label negatif

Dapat dilihat pada gambar 4.5 pada hasil klasifikasi ulasan keseluruhan data sebanyak 393 data ulasan negatif, didapatkan beberapa kata yang paling banyak muncul yaitu kata “keluar” sebanyak 285 kali, kata “jalan” sebanyak 271 kali, kata “berjalan”

sebanyak 265 kali dan seterusnya. Sedangkan visualisasi untuk sentimen positif berdasarkan 10 kata teratas yaitu sebagai berikut :



Gambar 4. 6 Frekuensi Muncul Label Positif

Dapat dilihat pada gambar 4.6 pada hasil klasifikasi ulasan keseluruhan data sebanyak 2609 data didapatkan beberapa kata yang paling banyak muncul yaitu kata “matahari” sebanyak 1120 kali, “terbit” 810 kali, “indah” 630 kali, dan seterusnya.

9. Tahap Asosiasi

Selanjutnya dilakukan tahap *association rules*. Dimana tahap ini dilakukan untuk mengetahui kata apa yang sering muncul secara bersamaan serta mengetahui nilai *support* yang dimiliki. Berikut merupakan hasil *association rules* pada 17 kata teratas pada tabel 4.20

Tabel 4. 20 Hasil Asosiasi

No	Antecedents (awal)	Consequents (diikuti)	support	confidence	lift	conviction
1	frozenset({'biaya'})	frozenset({'mahal','turis'})	0.0108 7	1	61.3 3	inf
2	frozenset({'menikmati'})	frozenset({'terbit','gajah'})	0.0108 7	1	52.5 7	inf
3	frozenset({'pemandangan'})	frozenset({'indah'})	0.0108 7	1	40.8 9	inf
4	frozenset({'berjalan'})	frozenset({'melelahkan'})	0.0108 7	1	40.8 9	inf

No	Antecedents (awal)	Consequents (diikuti)	support	confidence	lift	conviction
5	frozenset({'bagus'})	frozenset({'berjalan'})	0.0108 7	1	30.6 7	inf
6	frozenset({'mobil', 'berjalan', 'jalan'})	frozenset({'membeli'})	0.0108 7	1	24.5 3	inf
7	frozenset({'bagus', 'terbit', 'jalan'})	frozenset({'berbeda'})	0.0108 7	1	16.7 3	inf
8	frozenset({'ramai'})	frozenset({'toko', 'jualan'})	0.0108 7	1	16.7 3	inf
9	frozenset({'parkir', 'mobil'})	frozenset({'toko', 'dekat'})	0.0108 7	1	14.7 2	inf
10	frozenset({'tidak', 'panduan'})	frozenset({'jalan'})	0.0108 7	1	14.7 2	inf
11	frozenset({'terbit', 'berbeda', 'berjalan'})	frozenset({'bagus'})	0.0108 7	1	14.7 2	inf
12	frozenset({'terbit', 'berbeda', 'berjalan', 'jalan'})	frozenset({'bagus'})	0.0108 7	1	14.7 2	inf
13	frozenset({'keluar', 'penjualan'})	frozenset({'ramai', 'jalan'})	0.0108 7	1	14.7 2	inf
14	frozenset({'pulau'})	frozenset({'jawa'})	0.0135 87	1	13.1 4	inf
15	frozenset({'terletak', 'jalan'})	frozenset({'jawa'})	0.0108 7	1	13.1 4	inf
16	frozenset({'puncak', 'patung', 'indah'})	frozenset({'mendaki'})	0.0108 7	1	12.6 9	inf
17	frozenset({'terbit', 'singkat'})	frozenset({'pagi'})	0.0108 7	1	10.5 1	inf

Adapun *pseudocode* yang digunakan dalam proses asosiasi yaitu tercantum pada tabel 4.21.

Tabel 4. 21 *pseudocode* yang digunakan pada asosiasi

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

Pada proses *association rule* menghasilkan beberapa *output* yaitu *support*, *confidence* dan *lift*. *Support* merupakan persentase kombinasi dari *item* yang ada dalam data yang digunakan. Lalu *Confidence* atau nilai kepastian adalah kuatnya hubungan antara *item* dalam aturan asosiasi. Sedangkan *lift ratio* adalah suatu ukuran untuk mengetahui kekuatan aturan asosiasi (*association rule*) yang telah terbentuk. Nilai *lift ratio* biasanya

digunakan sebagai penentu apakah aturan asosiasi *valid* atau tidak *valid*. Apabila nilai *lift* ratio >1 , maka menunjukkan bahwa *rules* bersifat kuat atau *valid* untuk digunakan sebagai acuan dalam memprediksi. Dan nilai *lift* menjauhi nilai 1 hingga *infinite* maka dianggap makin akurat. Sehingga didapatkan hasil seperti pada tabel 4.22.

Tabel 4. 22 Ringkasan Asosiasi Terpilih

No	Kata Awal	Kata lanjutan
1	Biaya	Mahal
2	Menikmati	Terbit (<i>Sunrise</i>)
	Menikmati	Gajah
3	Pemandangan	Indah
4	berjalan	melelahkan
5	bagus	jalan
6	Ramai	jualan
7	toko	parkir jauh
8	Panas	Berjalan
9	(ada)Panduan	jalan

Berikut merupakan *Kansei word* yang disajikan dalam tabel 4.23

Tabel 4. 23 *Kansei words* terpilih

<i>Kansei Word</i>
Mahal \times Murah
Menikmati \times Bosan
Indah \times Buruk
lelah \times Rileks
jauh \times dekat
Ramai \times Sepi
Bagus \times Jelek
Tersedia \times Tiada
Panas \times Dingin

4.3 Perentangan Ruang Atribut

4.3.1 Penentuan Atribut Jasa

Atribut jasa adalah sebuah pelayanan yang diberikan pihak Candi Borobudur kepada pengunjung. Dalam penelitian ini, inti dari penentuan atribut jasa berasal dari kata *Kansei word* yang didapatkan dari perentangan ruang semantik sebelumnya beserta kata yang menyertainya. Contohnya pada *Kansei words* pertama yaitu “mahal”, dikarenakan kata yang menyertainya yaitu biaya dan turis, maka dibuatlah dua kalimat atribut jasa yang mengandung kata “biaya” dan “turis” mengandung kata sifat “mahal”. Namun dalam atribut jasa yang dibuat menggunakan *Kansei word* yang memiliki kata positif, sehingga apabila terdapat kata *Kansei* negatif maka diubah terlebih dahulu menjadi positif seperti pada tabel 4.23. Sehingga dihasilkan beberapa atribut jasa seperti pada tabel 4.24.

Tabel 4. 24 Atribut Jasa

Atribut jasa
Biaya Borobudur murah untuk turis lokal
Biaya Borobudur murah untuk turis mancanegara
Pengunjung dapat menikmati Pemandangan Terbit
Kandang Gajah dirawat dengan baik, sehingga dapat di nikmati
Pemandangan Candi Borobudur Indah
Tempat Parkir berdekatan dengan Toko dan jauh harus memutar area pertokoan terlebih dahulu.
Berjalan di Candi Borobudur Melelahkan dan Panas
Kondisi Jalan Borobudur bagus
Jalan keluar candi ramai penjual
Tempat parkir yang dekat dengan kawasan pertokoan
Terdapat panduan pengunjung (buku petunjuk/pamflet)

4.3.2 Pengambilan Data Derajat Kepentingan dan Kepuasan

Dilakukan pengambilan data untuk melakukan pengukuran kepentingan dan kepuasan pelanggan untuk menentukan kualitas pelayanan Candi Borobudur. Pengambilan data diambil menggunakan penyebaran kuisisioner. Kuisisioner ini dibuat untuk mendapatkan penilaian terhadap seberapa penting atribut jasa tersebut dirasakan oleh pengguna serta seberapa puas wisatawan ketika mengunjungi Candi Borobudur. Pertanyaan diajukan dengan menggunakan 5 skala penilaian (*Semantic Differensial Scale*). Kuisisioner akan

diajukan kepada 50 responden. Semakin besar sample dari besarnya populasi yang ada adalah semakin baik, akan tetapi ada jumlah batas minimal yang harus diambil oleh peneliti yaitu sebanyak 30 sampel (Jindal et al., 2016).

4.3.3 Pengambilan Data *Kano*

Pengambilan data *kano* dilakukan melalui penyebaran kuisisioner. Terdapat dua macam pertanyaan yang diajukan dalam kuisisioner *kano*, yaitu *Functional* dan *DysFunctional*. Pertanyaan yang diajukan terlampir pada lampiran B-2 berikut merupakan contoh dari kuisisioner *kano* pada tabel 4.25

Tabel 4. 25 Kuisisioner *Kano*

Jalan keluar candi ramai penjual				
Suka	Mengharapkan	Netral	Toleransi	Tidak suka
Jalan keluar candi sepi penjual				
Suka	Mengharapkan	Netral	Toleransi	Tidak suka

4.3.4 Uji *Validitas*

Setelah mendapatkan hasil rekapitulasi dari kuisisioner kepentingan, kepuasan, dan *kano*. kemudian dilakukan pengujian *validitas* dengan menggunakan *software SPSS* 2agar memperoleh hasil yang akurat. Hasil pengujian *validitas* dapat dilihat pada table 4.26 dibawah ini.

Tabel 4. 26 Uji *Validitas Semantic Differential*

Atribut	r tabel	r hitung Derajat Kepentingan	r hitung Derajat Kepuasan	<i>Validitas</i> (r hitung > r tabel)
X1	0.278	0.374	0.455	<i>VALID</i>
X2	0.278	0.534	0.309	<i>VALID</i>
X3	0.278	0.632	0.665	<i>VALID</i>
X4	0.278	0.442	0.466	<i>VALID</i>
X5	0.278	0.797	0.787	<i>VALID</i>
X6	0.278	0.586	0.318	<i>VALID</i>
X7	0.278	0.782	0.488	<i>VALID</i>

Atribut	r tabel	r hitung Derajat Kepentingan	r hitung Derajat Kepuasan	Validitas (r hitung > r tabel)
X8	0.278	0.741	0.8	VALID
X9	0.278	0.438	0.757	VALID
X10	0.278	0.658	0.58	VALID
X11	0.278	0.541	0.467	VALID

Dari uji *validitas* kuisioner derajat kepentingan dan derajat kepuasan dapat dilihat nilai *Corrected Item-Total Correlation* atau nilai r hitung lebih besar dari nilai r tabel (0.278), sehingga pertanyaan tersebut dikatakan *valid* dan dapat digunakan sebagai instrumen penelitian. (Yamin & Kurniawan, 2009).

Sedangkan untuk hasil uji *validitas* pada kuisioner *kano* dapat dilihat pada tabel 4.27

Tabel 4. 27 Uji *Validitas* kuisioner *kano*

Atribut	r tabel	r hitung <i>Functional</i>	r hitung <i>Disfunctional</i>	Validitas (r hitung > r tabel)
X1	0.278	0.485	0.309	VALID
X2	0.278	0.354	0.287	VALID
X3	0.278	0.294	0.289	VALID
X4	0.278	0.325	0.366	VALID
X5	0.278	0.286	0.687	VALID
X6	0.278	0.331	0.318	VALID
X7	0.278	0.412	0.478	VALID
X8	0.278	0.589	0.64	VALID
X9	0.278	0.279	0.543	VALID
X10	0.278	0.568	0.512	VALID
X11	0.278	0.414	0.451	VALID

Dari uji *validitas* kuisioner *kano* dapat dilihat nilai *Corrected Item-Total Correlation* atau nilai r hitung lebih besar dari nilai r tabel (0.278), sehingga pertanyaan tersebut dikatakan *valid* dan dapat digunakan sebagai instrumen penelitian.

4.3.5 Uji Realibilitas

Setelah dilakukan uji *validitas* maka selanjutnya adalah melakukan uji reliabilitas yang bertujuan untuk menentukan sejauh mana atribut yang digunakan dapat handal sesuai dengan kepuasan pengunjung. Uji reliabilitas dilakukan dengan menggunakan bantuan *software SPSS* selanjutnya hasil dari uji reliabilitas dapat dilihat pada tabel 4.28

Tabel 4. 28 Uji Realibilitas

Cronbach's Alpha			
Harapan	Kenyataan	<i>Kano Functional</i>	<i>kano Disfunctional</i>
0.772	0.772	0.732	0.705

Berdasarkan uji reliabilitas dengan menggunakan *software SPSS* didapatkan nilai *Cronbach's Alpha* derajat kepentingan dan kepuasan yaitu sebesar 0.772 dan kuisisioner *kano Functional* dan *Disfunctional* yaitu sebesar 0.732 dan 0.705. Hal ini menunjukkan bahwa nilai Cronbach's Alpha > 0.7 jadi data tersebut dikatakan handal

4.3.6 Pengukuran Kualitas Pelayanan Candi Borobudur

Pengukuran Kualitas Pelayanan bertujuan untuk mengetahui atribut-atribut pada dimensi mana yang diberikan kontribusi dari yang terbesar sampai yang terkecil, maka dilakukan perhitungan harapan konsumen dan kinerja perusahaan. Untuk menganalisis kualitas yang telah dinilai, digunakan rumus Q yang sudah dijelaskan yaitu jika $Q \geq 1$, maka kualitas pelayanan dikatakan baik. Pada Tabel 4.29 menunjukkan kualitas pelayanan pada Candi Borobudur.

Tabel 4. 29 Pengolahan Kualitas Pelayanan

No	Atribut <i>Kansei</i>	Kenyataan (P)	Harapan (H)	Gap P-H	Kualitas (Q) =P/E
X2	Biaya Borobudur murah untuk turis mancanegara	1.5	3.98	-2.48	0.376884
X11	Terdapat panduan pengunjung (buku petunjuk/pamflet)	2.2	4.02	-1.82	0.547264
X6	Tempat Parkir berdekatan dengan Toko dan jauh harus memutari area pertokoan terlebih dahulu.	2.3	4.06	-1.76	0.566502

X7	Berjalan di Candi Borobudur Tidak Melelahkan dan Panas	2.96	4.42	-1.46	0.669683
X4	Kandang Gajah dirawat dengan baik, sehingga dapat di nikmati	2.36	3.34	-0.98	0.706587
X1	Biaya Borobudur murah untuk turis lokal	2.44	2.9	-0.46	0.841379
X3	Pengunjung dapat menikmati Pemandangan Terbit	3.44	3.94	-0.5	0.873096
X10	Tempat Parkir berdekatan dengan Toko dan jauh harus memutari area pertokoan terlebih dahulu.	3.44	3.92	-0.48	0.877551
X5	Pemandangan <i>Sunrise</i> Candi Borobudur Indah	4.56	4.38	0.18	1.041096
X8	Kondisi Jalan Borobudur bagus	4.6	4.4	0.2	1.045455
X9	Jalan keluar candi ramai penjual	4.64	2.76	1.88	1.681159

4.4 Integrasi *Kansei Engineering* Dengan Metode *Kano*

Dilakukan tahap sintesis untuk melakukan kategorisasi *Kansei words* sesuai dengan atribut jasa yang telah ditentukan bobotnya. dengan metode *kano*, kategorisasi yang dilakukan menjadi berdasarkan pada seberapa baik produk/jasa tersebut mampu memuaskan keperluan pelanggan.

4.4.1 Pengolahan Kuisisioner *Kano Kano*

Langkah yang dilakukan untuk mengkategorikan atribut-atribut kedalam metode *kano* yaitu dengan cara menentukan jawaban keseluruhan atas kuisisioner yang dibagikan terhadap 50 responden berdasarkan pengelompokan pertanyaan *Functional* dan *Disfunctional*. Tabel evaluasi *kano* digunakan untuk mengelompokkan suatu atribut-atribut dari setiap responden. Berikut tabel 4.30 evaluasi *kano* :

Tabel 4. 30 Aturan Evaluasi *Kano*

Kebutuhan Konsumen		Disfungsional				
		Suka	Seharusnya	Netral	Dapat ditoleransi	Tidak Suka
Fungsional	Suka	Q	A	A	A	O
	Seharusnya	R	I	I	I	M

Netral	R	I	I	I	M
Dapat ditoleransi	R	I	I	I	M
Tidak Suka	R	R	R	R	Q

Untuk melakukan penentuan kategori *kano*, dilakukan pencocokan antara jawaban *Functional* dengan *Disfunctional*. Apabila pada pertanyaan *Functional* responden menjawab *suka* dan pada pertanyaan *Disfunctional* responden menjawab *Netral*, maka akan diperoleh hasil berupa A yang berarti *Attractive*. selanjutnya dilakukan pengelompokan dan perhitungan dari hasil klasifikasi yang sudah didapatkan sehingga jumlah kategori *kano* disetiap atribut dapat terlihat. Hasil dari kuisioner dan pengolahan *Kano* dapat dilihat pada lampiran B-3. Dibawah ini hasil penentuan dalam klasifikasi masing – masing atribut ke kategori *Kano* berupa jumlah kategori yang didapatkan pada setiap atribut dapat dilihat tabel 4.31 sebagai berikut :

Tabel 4. 31 Perhitungan Kategori *Kano*

Kategori	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11
A	3	0	0	0	9	5	0	0	12	18	14
O	2	29	0	0	11	5	0	0	13	9	23
M	4	0	0	39	21	26	0	3	19	2	4
I	41	21	50	11	9	14	50	47	6	21	9
R	0	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0

4.4.2 Menentukan Kategori *Kano* Untuk Setiap Atribut

Setelah dilakukan penentuan dalam kebutuhan pelanggan sesuai dengan kategori *Kano*, selanjutnya menentukan kategori *Kano* tiap masing–masing atribut layanan dengan menggunakan *Blauth's Formula* sebagai berikut:

1. Jika $(one\ dimensional + attractive + must-be) > (indifferent + reverse + questionable)$, maka grade diperoleh dari ujung yang paling maksimum dari $(one - dimensional, attractive, must-be)$.
2. Jika $(one -dimensional + attractive + must-be) < (indifferent + reverse + questionable)$, maka grade diperoleh dari ujung yang paling maksimum dari $(indifferent, reverse, questionable)$.

3. Jika (*one – dimensional + attractive + must-be*) = (*indifferent + reverse + questionable*) maka grade diperoleh yang paling maksimum diantara semua kategori kano yaitu (*one – dimensional, attractive, must-be, indifferent, reverse, questionable*).

Berikut merupakan hasil pengolahan kategori kano pada tabel 4.32

Tabel 4. 32 Penentuan kategori kano

Atribut	Kategori Kano						Kategori
	A	O	M	Q	R	I	
X1	3	2	4	0	0	41	I
X2	0	29	0	0	0	21	O
X3	0	0	0	0	0	50	I
X4	0	0	39	0	0	11	M
X5	9	11	21	0	0	9	M
X6	5	5	26	0	0	14	M
X7	0	0	0	0	0	50	I
X8	0	0	3	0	0	47	I
X9	12	13	19	0	0	6	M
X10	18	9	2	0	0	21	A
X11	14	23	4	0	0	9	O

Setelah diperoleh hasil akhir dalam tabulasi diatas selanjutnya menentukan atribut layanan berdasarkan metode *Kano* dan selanjutnya dilakukan analisis pada atribut dengan *Kansei* yang telah di tentukan. Berikut tabel dibawah ini Tabel 4. Merupakan tingkatan atribut jasa kategori *Kano*.

Tabel 4. 33 Hasil Kategori Kano

No	Atribut <i>Kansei</i>	Gap P-H	Kualitas (Q) =P/E	<i>Kano</i>
X1	s	-0.46	0.841379	I
X2	Biaya Borobudur murah untuk turis mancanegara	-2.48	0.376884	O
X3	Pengunjung dapat menikmati Pemandangan Terbit	-0.5	0.873096	I
X4	Kandang Gajah dirawat dengan baik, sehingga dapat di nikmati	-0.98	0.706587	M

No	Atribut <i>Kansei</i>	Gap P-H	Kualitas (Q) =P/E	<i>Kano</i>
X5	Pemandangan Candi Borobudur Indah	0.18	1.041096	M
X6	Tempat Parkir berdekatan dengan Toko dan jauh harus memutari area pertokoan terlebih dahulu.	-1.76	0.566502	M
X7	Berjalan di Candi Borobudur Melelahkan dan Panas	-1.46	0.669683	I
X8	Kondisi Jalan Borobudur bagus	0.2	1.045455	I
X9	Jalan keluar candi ramai penjual	1.88	1.681159	M
X10	Tempat Parkir berdekatan dengan Toko dan jauh harus memutari area pertokoan terlebih dahulu.	-0.48	0.877551	I
X11	Terdapat panduan pengunjung (buku petunjuk/pamflet)	-1.82	0.547264	O

BAB 5

ANALISIS DAN PEMBAHASAN

5.1 Integrasi *Natural Language Processing* dan *Kansei-Kano*

Dalam mengintegrasikan *Natural Language Processing* dan *Kansei-Kano* dengan Candi Borobudur sebagai objek eksperimen, penelitian ini dilakukan metode ekstraksi pengetahuan *Kansei* dari ulasan online menggunakan *NLP*, dan selanjutnya mengusulkan model analisis yang dapat menghasilkan hubungan dari *Kansei word* yang didapatkan dengan evaluasi pengembangan berdasarkan *voice of customer*. Metode mulai dari semantik yang dalam dan struktur sintaksis mengekstrak pengetahuan *Kansei* murni melalui *NLP* yang hasil ekstraksinya disajikan di bentuk distribusi frekuensi dan asosiasi. Selanjutnya dilakukan metode *kano* untuk mengetahui kepentingan *Kansei word* berdasarkan kepentingannya. Alur integrasi yang digunakan dapat dilihat pada gambar 5.1.



Gambar 5. 1 Integrasi *NLP* dan *Kansei kano*

Dalam melakukan ekstraksi data *review* online, dilakukan klasifikasi untuk mengelompokkan sentimen negatif dan positif. Dalam klasifikasi terdapat beberapa metode yang sering digunakan peneliti lain yang memiliki rata-rata akurasi tiga pembagian data yang berbeda, didapatkan nilai akurasi metode *SVM* lebih tinggi dibandingkan metode *NBC* pada semua perbandingan data *training* dan *testing*. Nilai rata-rata total akurasi dari metode *SVM* menunjukkan error lebih dari 12%, yang mana nilai ini harusnya dapat lebih diminimalkan. Nilai error ini dapat disebabkan oleh jenis pelabelan kelas sentimen yang digunakan. Pendeteksian sentimen antar kata pada penelitian ini menggunakan kamus *lexicon* atau *bag of words*. Pada sistem ini dipelajari kata-kata yang terdapat dalam suatu sumber tanpa adanya pengkategorian kata berdasarkan struktur penyusun kalimat, seperti kata benda, kata kerja, kata sifat, dan sebagainya. Hal tersebut menjadikan kerancuan pada saat sistem melakukan pelabelan kata. Sebagai contoh kata “tidak suka”, susunan kalimat ini memiliki kata “tidak” yang memiliki sentimen negatif sedangkan kata “suka” bersentimen positif, dimana keduanya bertolak belakang. Dalam ekstraksi data selanjutnya, digunakan sentimen negatif sebagai acuan untuk melakukan pengembangan, dikarenakan sentimen positif dianggap pengunjung sudah puas akan pelayanan yang diberikan.

Kata-kata yang paling sering muncul dalam setiap kelas sentimen ini perlu dilakukan analisis asosiasi kata. Hal tersebut bertujuan agar peneliti mendapatkan informasi dan kesimpulan yang jelas atau tidak rancu Hasil asosiasi kata pada klasifikasi kelas sentimen positif seperti pada tabel 4. menunjukkan hasil asosiasi kata dengan beberapa performa. Performa pertama adalah “*support*” merupakan bobot penilaian terhadap seberapa sering asosiasi tersebut muncul, semakin besar nilai *support* maka asosiasi tersebut sering muncul dalam sebuah *dataset*. “*Confidence*” merupakan seberapa kuat hubungan antar kata, “*lift*” digunakan untuk melihat kuat tidaknya aturan asosiasi dalam pengolahan data, sehingga dapat ditentukan sebagai penentu apakah asosiasi tersebut *valid* atau tidak, sedangkan untuk nilai *conviction* menunjukkan nilai akurasi dalam sebuah asosiasi. Dalam penelitian ini, dilakukan pengambilan *Kansei word* dari hasil asosiasi yang telah dibuat, dengan kualifikasi nilai *lift* >1 dan nilai *conviction* dengan nilai *inf* (*infinite*). Dengan hasil asosiasi sentimen negatif sebanyak 560 asosiasi, didapatkan 9 *Kansei word* yang dapat diambil dari hasil pengolahan menggunakan *NLP*. Hal tersebut dikarenakan karena banyaknya asosiasi yang dirasa kurang sesuai apabila dijadikan sebagai *Kansei word* seperti kata “jawa”, “keajaiban”, “situs”, “mendaki ukiran”, dan lain lain. Kerancuan

asosiasi tersebut dapat terjadi karena banyaknya kata sambung yang dihilangkan pada saat proses “*filtering*” di tahap *pre-processing*, dan banyak juga hasil asosiasi yang tidak memenuhi kriteria berupa nilai *lift* <1.

Kansei Engineering yang digunakan pada penelitian ini adalah *Kansei Engineering* Tipe 1, yaitu berupa kategorisasi menjadi lebih terperinci dan menghubungkannya ke sebuah tingkatan. Sehingga hasil yang akan didapatkan dalam penelitian ini berupa tingkatan prioritas atribut jasa yang dibutuhkan oleh wisatawan serta detail skor kualitas jasa yang didapatkan.

Tahapan melakukan *Kansei Engineering* tipe 1 secara garis besar yaitu penentuan *domain* penelitian, perentangan ruang semantik, perentangan ruang properti, dan sintesis. *Domain* penelitian pada penelitian ini yaitu kualitas atribut jasa yang terdapat pada pariwisata Candi Borobudur, dikarenakan Candi Borobudur termasuk kedalam program “10 bali baru” yang ditetapkan oleh pemerintah. Sehingga dengan alasan tersebut, peneliti menentukannya sebagai *domain* penelitian.

Perentangan ruang semantik, dilakukan pencarian *Kansei words* yang didapatkan dari hasil pengolahan asosiasi kata menggunakan *Natural Language Processing (NLP)*. Didapatkan 8 *Kansei word* yang dapat mewakili isi hati dari keluhan maupun pengalaman yang dialami pengunjung selama mengunjungi *Candi Borobudur* yang telah memberikan ulasannya di internet khususnya pada *tripadvisor*. *Kansei words* yang didapatkan bisa dilihat pada tabel 4.23.

Selanjutnya, dilakukan perentangan ruang properti, berupa atribut jasa yang terdapat hubungan dengan *Kansei words*. Atribut jasa yang digunakan berasal dari pasangan *Kansei word* pada asosiasi yang telah di dapatkan sebelumnya. Dari pasangan tersebut lalu dikembangkan sebuah kalimat yang dapat mewakili *Kansei words*, sehingga didapatkan 11 atribut jasa yang tercantum dalam tabel 4.24. dalam atribut jasa tersebut, terdapat *Kansei word* yang terbagi menjadi dua atribut jasa, yaitu pada *Kansei words* “murah”, “menikmati”. Pada *Kansei words* “murah” peneliti membuat dua jenis atribut dikarenakan harga tiket masuk *Candi Borobudur* yang berbeda untuk turis lokal dan mancanegara. Sedangkan untuk *Kansei words* “menikmati” dikarenakan kata yang mengiringi “menikmati” dari hasil asosiasi berupa “gajah” dan “terbit”, maka peneliti menetapkan 2 atribut yaitu seberapa besar pengunjung dapat menikmati suasana terbit dan suasana saat melihat fasilitas “kandhang gajah” yang ada di Candi Borobudur.

Untuk mengetahui bobot atribut jasa yang telah ditentukan, dilakukan pembobotan yang ada pada metode *service quality (SERVQUAL)*, dengan menentukan seberapa besar kualitas (Q) yang telah diberikan oleh pihak *Candi Borobudur* kepada pengunjung. Nilai Q dapat didapatkan dari hasil bagi antara bobot kenyataan (P) dan bobot harapan (H) yang didapatkan dari pengambilan kuisioner sebelumnya. Bobot kenyataan (P) adalah keadaan sesungguhnya yang dirasakan oleh pengunjung, sedangkan untuk bobot harapan (H) merupakan persepsi awal sebelum mendatangi *Candi Borobudur*. dilakukan pembobotan untuk harapan dan kenyataan pada setiap atribut. Semakin kecil selisih kenyataan dan harapan, maka semakin bagus, karena apa yang dibayangkan dan diharapkan oleh pengunjung ternyata sesuai dengan kenyataan yang dirasakan setelah mengunjungi *Candi Borobudur*. Nilai tersebut terlihat dari nilai *GAP*. Sedangkan untuk nilai kualitas (Q) apabila kualitas yang didapatkan $Q \geq 1$ maka kualitas pelayanan di *Candi Borobudur* dapat dikatakan baik.

Selanjutnya dilakukan tahap sintesis yaitu berupa penyatuan ruang semantik dan ruang properti serta dihasilkan pengkategorian terhadap atribut jasa yang telah diberikan bobot. Dalam melakukan kategorisasi, pada penelitian ini menggunakan metode *Kano* untuk menentukan kategori terhadap atribut mana yang paling penting berdasarkan persepsi pelanggan.

5.1.1 Hasil *Kano*

Pada model *kano* dilakukan kategorisasi atribut-atribut berdasarkan hubungan antara tingkat kepuasan pelanggan atas atribut pelayanan dan tingkat pemenuhan atribut pelayanan tersebut. metode *kano* digunakan untuk mengetahui seberapa besar pengaruh masing-masing atribut dalam memenuhi kepuasan pelanggan. Hal ini dilakukan untuk memudahkan pihak pengelola *Candi Borobudur* untuk melihat fasilitas mana yang perlu ditingkatkan.. Terdapat tiga kategori yang didapatkan dari atribut jasa *Candi Borobudur*, yaitu *Must Be*, *One Dimensional*, dan *Indifferent*. Berdasarkan hasil dari Kuesioner *Kano* didapatkan hasil pada tabel 5.1, 5.2 dan 5.3:

- a. Terdapat 4 atribut jasa yang termasuk kedalam kategori *Must Be (M)*, yaitu :

Tabel 5. 1 Kategori *Must Be*

No	Atribut Jasa	Gap P-H	Kualitas (Q) =P/E	Kano	Kansei Words
X6	Tempat Parkir berdekatan dengan Toko dan jauh harus memutari area pertokoan terlebih dahulu.	-1.76	0.566	M	Dekat
X4	Kandang Gajah dirawat dengan baik, sehingga dapat di nikmati	-0.98	0.706	M	Menikmati
X5	Pemandangan Candi Borobudur Indah	0.18	1.041	M	Indah
X9	Jalan keluar candi ramai penjual	1.88	1.681	M	Ramai

Kategori *Must Be* dipandang sebagai syarat mutlak bagi konsumen, karena mereka menganggap kategori ini sudah semestinya ada sehingga jika kategori ini tidak ada maka tidak memuaskan mereka sehingga pihak Candi Borobudur harus memenuhi atribut yang termasuk kedalam kategori *Must Be*. Atribut X6 memiliki nilai kualitas sebesar 0.566, sehingga hal ini mengindikasikan bahwa jarak antar tempat / fasilitas yang ada di candi borobudur jauh, sehingga apabila diabaikan dapat mengakibatkan pengunjung mengalami kelelahan. Atribut X4 memiliki nilai kualitas sebesar 0.706, menandakan bahwa pengunjung kurang bisa menikmati Kandang Gajah yang dipunyai oleh Candi Borobudur. Sedangkan untuk atribut jasa X5 dan X9 nilai kualitasnya ≤ 1 menandakan bahwa pelanggan sudah puas akan pemandangan candi borobudur dan jalan keluar candi sudah sepi penjual. Hasil yang didapatkan dari X9 menandakan bahwa kondisi jalan candi borobudur saat ini sudah tidak ramai akan penjual.

b. Terdapat 2 atribut jasa yang termasuk kedalam kategori *One Dimensional (O)*, yaitu :

Tabel 5. 2 Kategori *One Dimensional*

No	Atribut Jasa	Gap P-H	Kualitas (Q) =P/E	Kano	Kansei Words
X2	Biaya Borobudur murah untuk turis mancanegara	-2.48	0.376	O	Murah
X11	Terdapat panduan pengunjung (buku petunjuk/pamflet)	-1.82	0.547	O	Ada

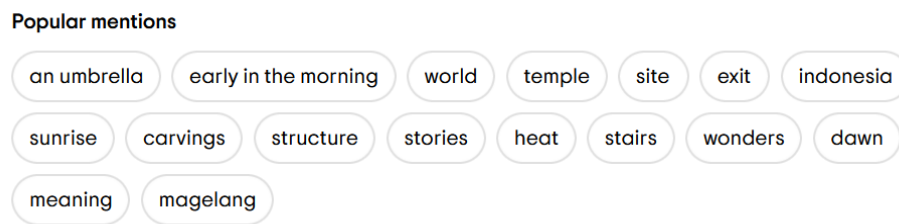
Atribut one-dimensional dirasa penting karena bersifat *linear*, jika atribut ini dipenuhi maka kepuasan konsumen akan meningkat namun, apabila tidak dipenuhi maka

kepuasan konsumen menurun atau konsumen akan merasa kecewa. Maka dari itu, pihak penyedia jasa disarankan memperhatikan atribut dalam kategori ini karena apabila atribut ini tidak terpenuhi maka konsumen akan merasa kecewa sehingga harus dilakukan perbaikan. Atribut X2 memiliki nilai kualitas (Q) sebesar 0.547 menandakan bahwa pengunjung merasa harga yang ditetapkan oleh pihak candi borobudur tidaklah murah, diketahui Harga Tiket Masuk (HTM) yang ditetapkan kepada turis mancanegara yaitu sebesar \$25 atau sebesar 350 ribu jika dibandingkan dengan harga turis lokal, maka terpaut jauh tujuh kali lipat. Sedangkan untuk atribut jasa X11 memiliki nilai kualitas sebesar 0.54 yang menandakan bahwa tidak adanya panduan pengunjung yang biasa digunakan sebagai petunjuk saat menikmati Candi Borobudur. Isi dari panduannya bisa berupa peta serta penjelasan baik fasilitas dan sejarah yang ada di Candi Borobudur. Dikarenakan atribut X11 masuk kedalam *One Dimensional*, maka apabila pihak Candi Borobudur tidak memenuhinya maka pengunjung dapat merasa kecewa. Terdapat 5 atribut jasa yang termasuk dalam kategori *Indifferent*, yaitu :

Tabel 5. 3 Kategori *Indifferent*

No	Atribut jasa	Gap P-H	Kualitas (Q) =P/E	Kano	Kansei Words
X7	Berjalan di Candi Borobudur Melelahkan dan Panas	-1.46	0.669	I	Lelah, Panas
X1	Biaya Borobudur murah untuk turis lokal	-0.46	0.841	I	Murah
X3	Pengunjung dapat menikmati pemandangan terbit	-0.5	0.873	I	Menikmati
X10	Tempat Parkir berdekatan dengan Toko dan jauh harus memutari area pertokoan terlebih dahulu.	-0.48	0.877	I	Dekat
X8	Kondisi jalan Borobudur bagus	0.2	1.045	I	Bagus

Kepuasan konsumen tidak dipengaruhi oleh sifat produk yang fungsional atau tidak fungsional. Baik buruknya atribut jasa yang disediakan dalam kategori *Indifferent* tidaklah berpengaruh kepada kepuasan pengunjung. Terdapat 5 atribut jasa yang termasuk kedalam kategori *Indifferent*.



Gambar 5. 2 kata kunci populer *tripadvisor*

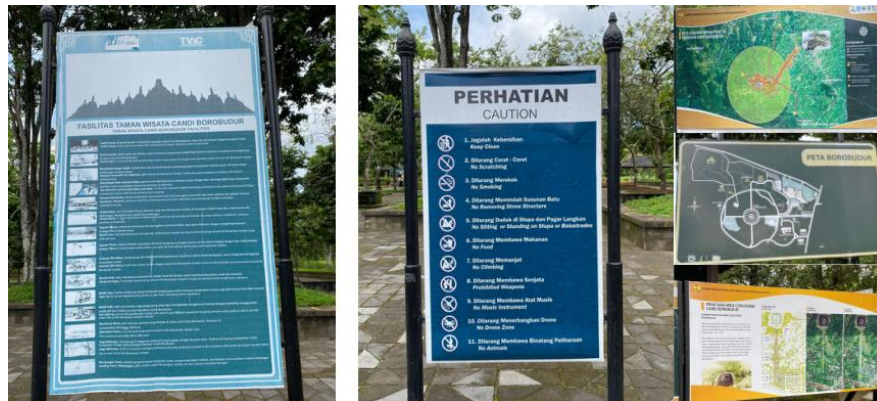
Hasil dari *NLP* yang diintegrasikan dengan *Kansei* dan *Kano* didapatkan beberapa atribut jasa terdapat kemiripan seperti yang terdapat pada pencarian populer pada *tripadvisor*, yaitu seperti *exit*, *morning*, *temple*, *Sunrise*, *heat*. Namun beberapa kata kunci lain tidak tercantum kedalam hasil penelitian, dikarenakan dalam penelitian ini dilakukan pengambilan ekstraksi data berupa sentimen negatif, sehingga kata yang tidak tercantum dapat termasuk kedalam sentimen positif maupun netral, yang tereliminasi dalam tahap *preprocessing*.

5.2 Rekomendasi Perbaikan Atribut Jasa

Rekomendasi yang diberikan kepada pengelola *Candi Borobudur* berdasarkan atribut jasa yang termasuk kedalam kategori *Kano Must Be* dan *One dimensional*. Kategori *Indifferent* tidak diberikan rekomendasi karena tercapai atau tidaknya atribut jasa tidak mempengaruhi kepuasan pengunjung saat menikmati Candi Borobudur. Dalam penelitian ini hanya terdapat 3 jenis kategori *kano*, sedangkan untuk kategori lain seperti *Attractive (A)*, *Reverse (R)*, dan *Questionable (Q)* tidak ada berdasarkan hasil perhitungan *kano* sebelumnya. Rekomendasi yang dapat diberikan berdasarkan studi literatur yaitu pada atribut jasa X4 pada kategori M, dan atribut jasa X2, X11 pada kategori O.

5.2.1 Rekomendasi 1

Pada atribut X11 yaitu “Terdapat panduan pengunjung (buku petunjuk/pamflet)” termasuk kedalam kategori *One Dimensional* dan memiliki kualitas jasa 0.547.



Gambar 5. 3 Panduan Pengunjung

Gambar 5.5 menunjukkan panduan yang dapat dilihat oleh pengunjung. Walaupun panduan pengunjung sudah disediakan, namun masih terdapat keluhan terkait panduan, sehingga disarankan untuk memberikan *handbook*/pamflet sebagai panduan saat mengunjungi Candi Borobudur tidak hanya pada pengunjung mancanegara, namun lokal juga. Dikarenakan pengunjung dapat lupa atau tidak menyadari panduan yang sudah disediakan disekitar pintu masuk. Dengan menambahkan buku panduan yang berisi edukasi terkait borobudur juga dapat menambah wawasan dan daya tarik pengunjung terhadap sejarah yang ada di Candi Borobudur. Atribut X11 apabila tidak terpenuhi maka akan menyebabkan ketidakpuasan pelanggan sehingga dianjurkan pihak Candi Borobudur untuk memenuhi atribut ini.



Gambar 5. 4 Contoh denah pengunjung

Rekomendasi yang dapat diberikan yaitu memberikan *handbook* berupa peta mini untuk memudahkan pengunjung untuk berkeliling kawasan Candi Borobudur. Gambar

5.6 diatas menunjukkan contoh denah yang diberikan oleh pihak gembira loka zoo kepada para pengunjung. Dengan adanya peta tersebut, pengunjung dapat mengetahui objek apa saja yang ada disekitarnya serta pihak pengelola juga dapat memberikan info terkait event apa saja yang terjadi pada hari tertentu, sehingga pengunjung dapat mengetahui info yang tersedia pada hari itu.

5.2.2 Rekomendasi 2

Pada atribut jasa X4, yaitu “Menikmati "Kandang Gajah" Candi Borobudur” termasuk kedalam kategori Must Be dan memiliki kualitas jasa 0.706.



Gambar 5. 5 Kondisi gajah di Taman Wisata Candi Borobudur

berdasarkan gambar 5.7 dapat dilihat kondisi gajah yang berada di kandang sedang terlihat tidak sehat, dan berdasarkan berita yang dikutip dari *borobudurnew.com* wahana “Kandang Gajah” nyaris ditutup karena pihak Candi Borobudur merantai kaki gajah, sehingga menuai protes dari pihak *Animal Friends Jogja (AFJ)* untuk menutup wahana tersebut. Bahkan berdasarkan gambar 5.7 terdapat pengunjung yang mengeluhkan kondisi gajah yang berada dikandang, perawatan gajah tersebut dianggap tidak mencerminkan ajaran hindu dimana Candi Borobudur merupakan Candi hindu terbesar di dunia.



Escape370681
1 contribution



Elephant mistreatment

Oct 2019

Seeing the chained up elephants at the end of the trip ruined the entire day. We would not have entered the park had we known these elephants would be there. Why are these elephants needed at the park? Do the right thing and hand them over to animal rescue. How can you teach Buddhism and treat these animals like this? You should be ashamed.

Written September 5, 2020

Sehingga Candi Borobudur perlu melakukan perbaikan terhadap wahana kandang gajah supaya pengunjung dapat nyaman saat menikmatinya, hal ini wajib karena termasuk kedalam kategori Must Be. Berikut merupakan beberapa penelitian yang telah dilakukan untuk meningkatkan kualitas memelihara gajah.

a. Pengelolaan kandang

Hal yang perlu dipertimbangkan dalam aspek perkandangan diantaranya jenis, , ukuran, , peralatan dan daya dukung kandang. Selain itu perlu juga dikaji suhu dan kelembaban kandang serta pemeliharaan kandang. Biaza, (2006) yang dikutip dalam Flying et al., (2017) menyatakan bahwa luas minimal kandang terbuka untuk setiap individu gajah jantan 500 m² dan betina 200 m², selain itu menurut Clubb & Mason, (2002) setidaknya untuk tiga gajah membutuhkan 400 m² dan 100 m² setiap penambahan gajah.

Prahara (1999) yang dikutip dalam Flying et al., (2017) menyatakan bahwa minimal 70% dari kandang harus merupakan ruang terbuka dan dapat ditembus oleh sinar matahari. Konstruksi atap yang terbuat dari seng tidak sesuai untuk kandang gajah, karena seng merupakan akumulator panas sehingga akan menyebabkan kenaikan suhu udara, atap yang terbuat dari genteng merupakan pilihan karena tidak mengumpulkan panas dan memiliki daya pakai panjang (Flying et al., 2017). Serta hindari genangan air pada kandang, karena berdasarkan penelitian oleh Vanitha et al., (2011) genangan air pada kandang akan menyebabkan masalah pada kaki gajah diantaranya kontaminasi pathogen.

Gajah merupakan satwa yang cepat mengalami pemanasan tubuh sehingga keberadaan air untuk minum dan membasahi tubuhnya sangat diperlukan untuk mempertahankan kelembaban tubuhnya. Gajah membutuhkan air sebanyak 120 liter per hari (Flying et al., 2017). Suhu udara yang sesuai dengan gajah berkisar

antara 26 - 37 *celcius* sehingga suhu pada lokasi masih sesuai dengan suhu yang diperlukan oleh gajah (Kowalski et al., 2010).

b. Pengelolaan Pakan

Pemindahan lokasi penggembalaan gajah dilakukan dua kali pada pagi dan sore hari. Pemberian pakan tambahan diberikan apabila ada kegiatan wisata dan patrol sehingga tidak rutin dan bersifat tentatif. Pemberian suplemen ini dilakukan 2 minggu sekali dan diberikan apabila ada kegiatan tambahan atau kondisi khusus. Waktu untuk pemberian suplemen ini tidak ditentukan karena keterbatasan informasi dari pustaka yang berkaitan dengan jumlah kebutuhan nutrisi harian gajah yang belum banyak diteliti (Kowalski et al., 2010)

c. Pengelolan Kesehatan

Pemeliharaan gajah di *WWF* Taman Nasional Riau yaitu dengan memandikan gajah minimal 1 kali setiap hari. Pemantauan kesehatan dilakukan setiap hari dengan memperhatikan nafsu makan dan minum, jumlah dan kondisi kotoran, luka atau pincang, warna selaput lendir, aktivitas tidak normal dan birahi. Pemeriksaan oleh dokter dilakukan enam bulan sekali dan penanganan dokter dilakukan apabila gajah mengalami sakit yang sangat parah, selain itu pemeriksaan darah dilakukan oleh dokter hewan satu tahun sekali. Pemeriksaan darah ini sesuai dengan anjuran (Flying et al., 2017). Sedangkan Penyakit yang sering diderita oleh gajah ialah cacangan, gondokan, penumpukan gas (blood-red) dan luka-luka. Penyakit cacangan memang merupakan penyakit yang sering menjangkiti gajah di penangkaran (Miller et al., 2015)

5.2.3 Rekomendasi 3

Pada atribut X2 yaitu “Biaya Borobudur murah untuk turis mancanegara” termasuk kedalam kategori One Dimensional dan memiliki kualitas jasa 0.37.



Diketahui harga tiket untuk pengunjung lokal seharga 50.000 rupiah pada hari senin-jumat, sedangkan untuk harga yang ditetapkan pada turis mancanegara yaitu sebesar 350.000 rupiah. Selisih harga antara lokal dengan mancanegara mencapai 7 kali lipat. Sehingga banyak turis asing yang merasa harga yang ditetapkan terlalu mahal dan dapat memicu diurungkannya niat untuk turis asing menungguni Candi Borobudur. Tidak sedikit juga turis asing yang telah melakukan komplain kepada pihak pengelola.

Berdasarkan wawancara yang dilakukan oleh suarajawatengah.id, Pelaksana Tugas General Manager TWC Borobudur, Jamaludin Mawardi, perbedaan harga yang diterapkan pada pengunjung candi borobudur salah satunya untuk mengapresiasi wisatawan lokal untuk berkunjung ke situs peninggalan bersejarah, yaitu dengan mensubsidi harga tiket masuk untuk turis lokal. Apabila dibandingkan dengan situs peninggalan dunia (*world heritage*) lainnya, harga yang ditetapkan tidak jauh berbeda, diketahui harga tiket masuk ke *Angkor Archeological Park* sebesar USD 37 (Rp 530 ribu). Dengan harga tiket candi borobudur sebesar Rp. 350.000, pengunjung mendapatkan perlakuan yang berbeda dibandingkan dengan turis lokal, yaitu tersedianya *lounge* (ruang santai khusus) serta *welcome drink* yang diberikan untuk pengunjung. Serta pada awal pembelian tiket pengunjung juga diberikan edukasi terkait sejarah Candi Borobudur.

Namun, dikarenakan atribut X2 berada dalam kategori “O” maka pihak pengelola disarankan untuk meninjau kembali harga yang ditetapkan kepada turis mancanegara, karena jika tidak pengunjung dapat merasa tidak puas. Berdasarkan penelitian Sekar dkk (2013) yang menyatakan untuk mengetahui tingkat kepuasan terhadap harga yang

ditetapkan, dapat menggunakan konsep *Willingness To Pay* (WTP), yaitu merupakan nilai preferensi responden yang berbasis kepada perilaku, serta kepentingannya, apakah bersedia membayar sejumlah uang untuk biaya ganti rugi, menghindari kerusakan atau hilangnya lingkungan ekowisata, berkontribusi terhadap konservasi dan perbaikan atas kualitas lingkungan, serta perlindungan jasa lingkungan wisata. Nilai ini ditentukan oleh bersedia atau tidaknya wisatawan mempertimbangkan untung rugi (trade-offs) dan sanggup membayar harga komoditas lingkungan berupa spek sumber daya alam, yang termaksud di dalamnya yaitu keindahan, keberadaan, serta upaya konservasi dan pemeliharaannya.

BAB 6

PENUTUP

6.1 Kesimpulan

Integrasi *NLP* dengan *Kansei Engineering* dan *Kano* dilakukan dengan berbagai tahap yaitu dengan melakukan perentangan ruang semantik, berisi pengolahan data sentimen yang didapatkan dengan melakukan tahap preprocessing dan klasifikasi menggunakan *SVM* dan *Naive Bayes*. Setelah itu dilakukan penentuan ruang atribut dengan melakukan pengukuran kualitas layanan serta mengintegrasikan *Kansei* dengan *kano* untuk mendapatkan prioritas mana yang memiliki tingkat kepuasan tertinggi berdasarkan klasifikasi yang didapatkan dari model *kano*. Ekstraksi data dari pengolahan menggunakan klasifikasi metode *Natural Language Processing* dengan menggunakan kelompok *Sentiment* negatif didapatkan *Kansei words* sejumlah delapan kata yaitu “Murah”, “Menikmati”, “Indah”, “Rileks”, “dekat”, “Sepi”, “Bagus”, “Tersedia”.

Metode *SVM* memiliki kemampuan yang lebih baik dalam mengklasifikasikan ulasan pengunjung Candi Borobudur dibandingkan metode *NBC*. Dengan hasil rata-rata total akurasi pada keempat *kernel SVM* lebih tinggi dengan nilai 86.97% dibandingkan dengan metode *NBC* yang memiliki nilai jika dirata rata yaitu sebesar 66.87%.

Hasil dari *NLP* yang diintegrasikan dengan *Kansei* dan *Kano* dianggap efektif karena didapatkan beberapa atribut jasa terdapat kemiripan pada pencarian populer pada *tripadvisor*, yaitu seperti *exit*, *morning*, *temple*, *Sunrise*, *heat*. Serta beberapa atribut yang tidak tercantum namun dianggap penting berdasarkan hasil integrasi *Kansei* dan *kano* seperti harga tiket, pengelolaan kandang gajah, dan *pamflet*.

Dihasilkan rekomendasi untuk pengembangan jasa pariwisata Candi Borobudur berupa pemberian pamflet pada pengunjung, peningkatan pemeliharaan gajah baik dari pakan, kandang, dan kesehatan gajah, serta peninjauan kembali harga tiket masuk menggunakan metode *willingness to pay*.

6.2 Saran

Berdasarkan hasil analisis, pembahasan, dan kesimpulan yang telah dilakukan, terdapat beberapa saran yang bisa diberikan yaitu :

1. Bagi Pihak Candi Borobudur

Hasil ekstraksi informasi dari ulasan-ulasan yang telah diberikan oleh pengunjung yang telah diolah menjadi atribut jasa, dapat dijadikan sebagai bahan evaluasi dalam upaya peningkatan kepuasan pelanggan dan memberikan pelayanan semaksimal mungkin, terutama pada kategori *Must Be* dan *One Dimensional*, karena kedua kategori tersebut sangat mempengaruhi kepuasan pengunjung.

2. Bagi Penelitian Selanjutnya

Untuk penelitian selanjutnya, sebaiknya algoritma klasifikasi yang digunakan dapat lebih bervariasi lagi supaya mendapatkan hasil yang lebih baik saat dibandingkan, memperbanyak jumlah kombinasi , serta periode pengumpulan data yang lebih panjang agar informasi yang diperoleh bisa lebih mendalam dikarenakan penelitian sekarang sentimen negatif yang didapatkan sedikit. Sehingga mempengaruhi hasil *Kansei word* yang didapatkan, dan membuat penelitian menjadi terbatas terutama pada variasi atribut yang diukur.

DAFTAR PUSTAKA

- Abdi, J. K., Kartika, M., Prodi, S., Keperawatan, S., Karya, S., & Kediri, H. (2020). *Sosialisasi tentang Pencegahan Covid-19 di Kalangan Siswa Sekolah Dasar di SD Minggiran 2 Kecamatan Papar Kabupaten Kediri*.
- Al-smadi, M., Qawasmeh, O., Al-ayyoub, M., Jararweh, Y., & Gupta, B. (2017). Accepted. *Journal of Computational Science*. <https://doi.org/10.1016/j.jocs.2017.11.006>
- Alamanda, D. T., & Kania, I. (n.d.). *Sentiment Analysis Using Text mining of Indonesia Tourism Reviews via Social Media*. 5(2), 72–82.
- Alash, H. M., & Al-sultany, G. A. (2019). *Automated Bahasa Indonesia essay evaluation with latent semantic analysis Automated Bahasa Indonesia essay evaluation with latent semantic analysis*. <https://doi.org/10.1088/1742-6596/1235/1/012100>
- Annur, H. (2018). KLASIFIKASI MASYARAKAT MISKIN MENGGUNAKAN METODE NAÏVE BAYES. In *Agustus* (Vol. 10, Issue 2).
- Azhar, Y. (2018). Metode *Lexicon-Learning Based* Untuk Identifikasi Tweet Opini Berbahasa Indonesia. *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, 6(3), 237. <https://doi.org/10.23887/janapati.v6i3.11739>
- Azis, H., Admojo, F. T., Susanti, E., Informatika, T., Komputer, F. I., Indonesia, U. M., Informatika, T., Industri, F. T., & Classifier, N. B. (2020). *Analisis Perbandingan Performa Metode Klasifikasi pada Dataset Multiclass Citra Busur Panah*. 19(3), 286–294.
- Belajar, M., Saat, S., Christy, N. J., & Oktavianti, R. (2020). *Pengaruh Komunikasi Persuasif Guru terhadap*. 187–193.
- Benjamín, G. F., & Benjamín, G. F. (2020). *Data preprocessing techniques in supervised machine learning : handling of noise and selection of characteristics . by Técnicas de preprocesamiento de datos en aprendizaje automático supervisado : tratamiento de ruido y selección de características . September*.
- Bidang, P., Sains, K., Mardi, Y., Gajah, J., No, M., & Barat, S. (2016). *Jurnal Edik Informatika Data Mining : Klasifikasi Menggunakan Algoritma C4 . 5 Data mining merupakan bagian dari tahapan proses Knowledge Discovery in Database (KDD) . Jurnal Edik Informatika*.
- Bima, A. P., Jofari, H. A., Candrawidodo, E. P., & Malang, U. M. (n.d.). *TANTANGAN INDONESIA DALAM PENATAAN PARIWISATA SUPER*. 1551–1570.
- Cahigas, M. M. L. (2020). *Kansei Engineering-based Model and Online Content Assessment in Evaluating Service Design of Lazada Express*. 49–55.

- Clubb, R., & Mason, G. (2002). *A Review of the Welfare of Zoo Elephants A Review of the Welfare of Zoo Elephants in Europe A report commissioned by the RSPCA. Animal Behaviour*, 1–303.
- Dwitri, N., Tampubolon, J. A., Prayoga, S., & Zer, P. P. P. A. N. W. F. I. R. H. (2020). *PENERAPAN ALGORITMA K-MEANS DALAM MENENTUKAN TINGKAT PENYEBARAN PANDEMI COVID-19 DI INDONESIA*. 4(1), 128–132.
- Engineering, J. I. (2020). *THE EFFECT OF BRAND IMAGE , PRICE AND SERVICE QUALITY ON CONSUMER DECISIONS*. 1(3), 135–147.
- Fahdia, M. R., Riana, D., Amsury, F., & Saputra, I. (2021). *Komparasi Algoritma Klasifikasi untuk Orientasi Minat Mahasiswa dalam Penuntasan Studi*. 2(7), 970–1007.
- Fanissa, S., Fauzi, M. A., & Adinugroho, S. (2018). *Analisis Sentimen Pariwisata di Kota Malang Menggunakan Metode Naive Bayes dan Seleksi Fitur Query Expansion Ranking*. 2(8), 2766–2770.
- Fatyanosa, T. N., & Bachtiar, F. A. (n.d.). *Classification Method Comparison on Indonesian Social Media Sentiment Analysis*.
- Flying, D. I., Wwf, S., Nasional, T., Ohir, R. I. K. T., Aris, A. B. H., Dan, M. U., & Asy, B. U. M. (2017). *PENGELOLAAN DAN TINGKAT KESEJAHTERAAN GAJAH SUMATERA (Elephas maximus sumatranus Temminck, 1847) DI FLYING SQUAD WWF TAMAN NASIONAL TESSO NILO RIAU. PENGELOLAAN DAN TINGKAT KESEJAHTERAAN GAJAH SUMATERA (Elephas Maximus Sumatranus Temminck, 1847) DI FLYING SQUAD WWF TAMAN NASIONAL TESSO NILO RIAU*, 21(2), 152–158. <https://doi.org/10.29244/medkon.21.2.152-158>
- Hartono, M. (2017). *Department of Industrial Engineering, Faculty of Engineering, University of Surabaya, Surabaya 60293, Indonesia*. 1070–1081.
- Hartono, M. (2020). *International Journal of Industrial Ergonomics The modified Kansei Engineering-based application for sustainable service design. International Journal of Industrial Ergonomics*, 79(January 2019), 102985. <https://doi.org/10.1016/j.ergon.2020.102985>
- Hong-bin, Y., & Ming, L. (2020). *pt us cr t. IISE Transactions*, 0(0), 000. <https://doi.org/10.1080/24725854.2020.1766727>
- Hutomo, M., Setiawan, R., Bisnis, P. M., Manajemen, P. S., Petra, U. K., & Siwalankerto, J. (2015). *KINERJA KARYAWAN PADA PT LESTARI PURNAMA PERKASA*. 3(2), 60–69.
- Jiao, Y., & Qu, Q. (2019). *Computers in Industry A proposal for Kansei knowledge extraction method based on Natural Language Processing technology and online product reviews. Computers in Industry*, 108, 1–11. <https://doi.org/10.1016/j.compind.2019.02.011>
- Jindal, A., Dua, A., Kaur, K., Singh, M., Kumar, N., & Mishra, S. (2016). *Decision Tree*

- and *SVM-Based Data Analytics for Theft Detection in Smart Grid*. *IEEE Transactions on Industrial Informatics*, 12(3), 1005–1016. <https://doi.org/10.1109/TII.2016.2543145>
- Kowalski, N. L., Dale, R. H. I., & Mazur, C. L. H. (2010). A survey of the management and development of captive african elephant (*Loxodonta africana*) calves: Birth to three months of age. *Zoo Biology*, 29(2), 104–119. <https://doi.org/10.1002/zoo.20195>
- Lunn, S., & Ross, M. (n.d.). *Utilizing Web Scraping and Natural Language Processing to Better Inform Pedagogical Practice*.
- Masudin, I., Safitri, N. T., Restuputri, D. P., Wisnu, R., & Amallynda, I. (2020). Cogent Business & Management The effect of humanitarian logistics service quality to customer loyalty using *Kansei Engineering*: Evidence from Indonesian logistics service providers The effect of humanitarian logistics service quality to customer loyalty . *Cogent Business & Management*, 7(1). <https://doi.org/10.1080/23311975.2020.1826718>
- Matzler, K., Fuchs, M., & Schubert, A. K. (2004). Employee satisfaction: Does *Kano's* model apply? *Total Quality Management and Business Excellence*, 15(9–10), 1179–1198. <https://doi.org/10.1080/1478336042000255569>
- Miller, D., Jackson, B., Riddle, H. S., Stremme, C., Schmitt, D., & Miller, T. (2015). Elephant (*Elephas maximus*) health and management in Asia: Variations in veterinary perspectives. *Veterinary Medicine International*, 2015. <https://doi.org/10.1155/2015/614690>
- Nagamachi, M. (1995). *Kansei Engineering: A new ergonomic consumer-oriented technology for product development*. 15, 3–11.
- Narayan, Y. (2020). Materials Today : Proceedings Comparative analysis of *SVM* and *Naive Bayes* classifier for the SEMG signal classification. *Materials Today: Proceedings*, xxxx. <https://doi.org/10.1016/j.matpr.2020.09.093>
- Okazaki, S., Andreu, L., & Campo, S. (2016). *Knowledge Sharing Among Tourists via Social Media: A Comparison Between Facebook and tripadvisor*. <https://doi.org/10.1002/jtr>
- Prabowo, R., & Industri, J. T. (2019). *PENERAPAN INTEGRASI KANO DAN KANSEI ENGINEERING UNTUK*. 978–979.
- Prasetyo, V. R., Benarkah, N., & Chrisintha, V. J. (2021). *Implementasi Natural Language Processing Dalam Pembuatan Chatbot Pada Program Information Technology Universitas Surabaya Implementation of Natural Language Processing in Creating Chatbots at Information Technology Program , University of Surabaya*. 10(2), 114–121. <https://doi.org/10.34148/teknika.v10i2.370>
- Qi, C. (2020). *Big Data management in the mining industry*. 27(2).
- Rofiqoh, U., Perdana, R. S., & Fauzi, M. A. (2017). *Analisis Sentimen Tingkat Kepuasan*

- Pengguna Penyedia Layanan Telekomunikasi Seluler Indonesia Pada Twitter Dengan Metode Support vector machine dan Lexicon Based Features. 1(12), 1725–1732.*
- Romario, M. H., Ihsanto, E., & Kadarina, T. M. (2020). Sistem Hitung dan Klasifikasi Objek dengan Metode Convolutional Neural Network. *Jurnal Teknologi Elektro, 11(2)*, 108. <https://doi.org/10.22441/jte.2020.v11i2.007>
- Song, J., Kim, K. T., Lee, B., Kim, S., & Youn, H. Y. (2017). *A novel classification approach based on Naïve Bayes for Twitter Sentiment analysis. 11(6), 2996–3011.*
- Sulistyaningsih, P., & Susila, A. (n.d.). *STRATEGI UNIT PAMOBVIT POLRES MAGELANG DALAM The 9 th University Research Colloquium 2019 Universitas Muhammadiyah Purworejo.*
- Supriyanto, E. E., Bakti, I. S., & Furqon, M. (2021). *THE ROLE OF BIG DATA IN THE IMPLEMENTATION OF DISTANCE. 6356(4), 61–68.*
- Vanitha, V., Thiyagesan, K., & Baskaran, N. (2011). Social life of captive Asian elephants (*Elephas maximus*) in Southern India: Implications for elephant welfare. *Journal of Applied Animal Welfare Science, 14(1), 42–58.* <https://doi.org/10.1080/10888705.2011.527603>
- Yamin, S. dan, & Kurniawan, H. (2009). statistik SPSS complete: tEknik anaLisis statistik tErLEngkaP dEngan software SPSS struCTuraL EquatiOn MOdELing: BELajar LEBih Mudah tEknik anaLisis data kuEsiOnEr dEngan LisrEL–PLs. *Analisis Korespondensi Bab Analisis Diskriminan.*
- Yeh, C., & Chen, M. (2018). Applying *Kansei Engineering* and data mining to design door-to-door delivery service. *Computers & Industrial Engineering, May.* <https://doi.org/10.1016/j.cie.2018.05.011>
- Yusup, F., Studi, P., Biologi, T., Islam, U., & Antasari, N. (2018). *UJI VALIDITAS DAN RELIABILITAS. 7(1), 17–23.*

LAMPIRAN

A-1 Hasil Klasifikasi *Naive Bayes* dan *Support Vector Machine*

Report:				
	precision	recall	f1-score	support
negatif	0.16	0.25	0.20	88
positif	0.86	0.78	0.82	513
accuracy			0.70	601
macro avg	0.51	0.51	0.51	601
weighted avg	0.76	0.70	0.73	601
Confusion Matrix:				
	Predicted negatif	Predicted positif		
Actual negatif	22	66		
Actual positif	114	399		

Lampiran A-1. 1 Hasil klasifikasi *Naive Bayes* 80:20

Report:				
	precision	recall	f1-score	support
negatif	0.80	0.14	0.23	88
positif	0.87	0.99	0.93	513
accuracy			0.87	601
macro avg	0.84	0.57	0.58	601
weighted avg	0.86	0.87	0.83	601
Confusion Matrix:				
	Predicted negatif	Predicted positif		
Actual negatif	12	76		
Actual positif	3	510		

Lampiran A-1. 2 Hasil Klasifikasi *SVM Linear* 80:20

Report:				
	precision	recall	f1-score	support
negatif	0.00	0.00	0.00	88
positif	0.85	1.00	0.92	513
accuracy			0.85	601
macro avg	0.43	0.50	0.46	601
weighted avg	0.73	0.85	0.79	601

Confusion Matrix:		
	Predicted negatif	Predicted positif
Actual negatif	0	88
Actual positif	0	513

Lampiran A-1. 3 Hasil Klasifikasi *SVM RBF* 80 :20

Report:				
	precision	recall	f1-score	support
negatif	0.82	0.10	0.18	88
positif	0.87	1.00	0.93	513
accuracy			0.87	601
macro avg	0.84	0.55	0.55	601
weighted avg	0.86	0.87	0.82	601

Confusion Matrix:		
	Predicted negatif	Predicted positif
Actual negatif	9	79
Actual positif	2	511

Lampiran A-1. 4 Hasil Klasifikasi *SVM Sigmoid* 80:20

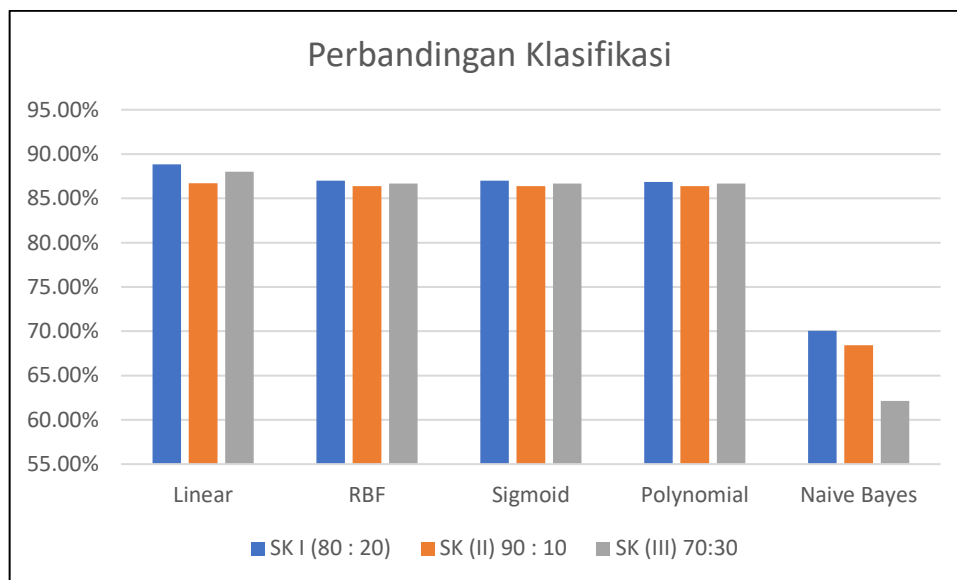
Report:				
	precision	recall	f1-score	support
negatif	0.00	0.00	0.00	88
positif	0.85	1.00	0.92	513
accuracy			0.85	601
macro avg	0.43	0.50	0.46	601
weighted avg	0.73	0.85	0.79	601

Confusion Matrix:		
	Predicted negatif	Predicted positif
Actual negatif	0	88
Actual positif	0	513

Lampiran A-1. 5 Hasil Klasifikasi *SVM Polynomial* 80:20

Report:				
	precision	recall	f1-score	support
negatif	0.80	0.18	0.29	45
positif	0.87	0.99	0.93	256
accuracy			0.87	301
macro avg	0.84	0.58	0.61	301
weighted avg	0.86	0.87	0.83	301

Confusion Matrix:		
	Predicted negatif	Predicted positif
Actual negatif	8	37
Actual positif	2	254

Lampiran A-1. 6 Hasil Klasifikasi *SVM linear* 90:10Lampiran A-1. 7 Grafik Perbandingan Klasifikasi *SVM* dan NB

Lampiran A-2 Pseudocode keseluruhan

Lampiran A-2. 1 Instal *Library* dan *mengimport* yang dibutuhkan

```

pip install Sastrawi
import string
pip install wordcloud
import pandas as pd
import numpy as np
import re
import nltk
from nltk.corpus import stopwords

```



```

from nltk.tokenize import word_tokenize
import string
from wordcloud import WordCloud
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from datetime import datetime, timedelta
import pytz
import itertools
import matplotlib.pyplot as plt
import seaborn as sns
from nltk.tokenize import word_tokenize

```

Lampiran A-2. 2 Preprocessing Data

```

data = pd.read_csv("dataset_final.csv")
data.head()
print(data['komentar'])
data=data.dropna()
data
#Cleaning Data
import string, re
def Cleansing (data):
    data = data.lower()
    remove = string.punctuation
    translator = str.maketrans(remove, ' '*len(remove))
    data = data. translate (translator)
    angber = ".join([x for x in data if not (x.isdigit()or x[0] == '-' and x[1:].isdigit())])
    return angber
stopwords = pd.read_csv("stopwordbudur.csv", header=None)[0].tolist()
stopwords[:10]
def tokenize (txt):

```

```

word_token = word_tokenize (txt)

return word_token

#StopWord

from Sastrawi.StopWordRemover.StopWordRemoverFactory import
StopWordRemoverFactory

factory = StopWordRemoverFactory()

def stopw (txt_tokenized):

    stopw = [word for word in txt_tokenized if word not in stopwords]

    return stopw

def kembali(txt):

    data_kembali = ' '.join([c for c in txt if c not in string.punctuation])

    return data_kembali

pd.options.display.max_colwidth = None

data['data_lower'] = data['komentar'].apply (lambda x: Cleansing(x))

data['data_token'] = data['data_lower'].apply (lambda x: tokenize (x))

data['data_stopword'] = data['data_token'].apply (lambda x: stopw (x))

data['data_kembali'] = data['data_stopword'].apply(lambda x: kembali (x))

data

data.to_csv("testing_prepros.csv")

```

```

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

DATA1["TF_dict"] = DATA1['Dataa_list'].apply(calc_TF)

DATA1["TF_dict"].head()

# Check TF result
index = 90

print('%20s' % "term", "\t", "TF\n")
for key in DATA1["TF_dict"][index]:
    print('%20s' % key, "\t", DATA1["TF_dict"][index][key])
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(DATA1["TF_dict"])
n_document = len(DATA1)

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
DATA1["TF-IDF_dict"] = DATA1["TF_dict"].apply(calc_TF_IDF)
# Check TF-IDF result
index = 90

print('%20s' % "term", "\t", '%10s' % "TF", "\t", '%20s' % "TF-IDF\n")
for key in DATA1["TF-IDF_dict"][index]:
    print('%20s' % key, "\t", DATA1["TF_dict"][index][key] , "\t" , DATA1["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

DATA1["TF_IDF_Vec"] = DATA1["TF-IDF_dict"].apply(calc_TF_IDF_Vec)

print("print first row matrix TF_IDF_Vec Series\n")
print(DATA1["TF_IDF_Vec"][0])

print("\nmatrix size : ", len(DATA1["TF_IDF_Vec"][0]))

# Convert Series to List
TF_IDF_Vec_List = np.array(DATA1["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=True)

```

```
lexicon= pd.read_csv('LexiconIndo.csv').reset_index(drop=True)
```

```

negasi = ['bukan','tidak','ga','gk']
lexicon_word = lexicon['word'].to_list()
lexicon_number_word = lexicon['number_of_word']
data['data_kembali'] = data['data_kembali'].astype(str)
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize

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(data)):
    nsen = senrow.shape[0]
    words = word_tokenize(data['data_kembali'][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.to_csv("bobot_ke3.csv")
cek_df = pd.DataFrame([])
cek_df['text'] = data['data_kembali'].copy()
cek_df['Sentiment'] = df_sen['Sentiment'].copy()
cek_df.head(5)
cek_df.to_csv("bobot_lengkap1.csv")
```

term	TF	TF-IDF
borobudur	0.03389830508474576	0.027889420856475108
candi	0.03389830508474576	0.023825461746548315
buddha	0.01694915254237288	0.03485973754907803
letak	0.03389830508474576	0.09876667346361248
magelang	0.01694915254237288	0.04991139042298921
java	0.01694915254237288	0.04533302160538917
indonesia	0.01694915254237288	0.03107764345900667
karena	0.05084745762711865	0.20157537966858205
belah	0.05084745762711865	0.23093039822967476
barat	0.05084745762711865	0.21031352832586983
daya	0.01694915254237288	0.0701045094419566
semarang	0.01694915254237288	0.08116086852737507
surakarta	0.01694915254237288	0.10534928133483516
laut	0.01694915254237288	0.08332753584110193
yogyakarta	0.01694915254237288	0.036479578742049205
cuaca	0.01694915254237288	0.05023630425532345
hujan	0.01694915254237288	0.05457415805425905
renyairenyai	0.01694915254237288	0.12396982860039636
sekejpa	0.01694915254237288	0.12396982860039636
lebt	0.01694915254237288	0.12396982860039636
iring	0.01694915254237288	0.10534928133483516
langkah	0.01694915254237288	0.06470698857554076
pagi	0.01694915254237288	0.029144211996975353
ramai	0.01694915254237288	0.04039575528467783
pemuduk	0.01694915254237288	0.12396982860039636
jual	0.01694915254237288	0.03717350061285193
payung	0.01694915254237288	0.04574744695901908
tegas	0.01694915254237288	0.12396982860039636
mahu	0.01694915254237288	0.11709753863246136
beli	0.01694915254237288	0.042066341166975364
rana	0.01694915254237288	0.11222157130277016
susah	0.01694915254237288	0.0909883006503063
nak	0.01694915254237288	0.11709753863246136
bawa	0.01694915254237288	0.033220252354865296
ajar	0.01694915254237288	0.04991139042298921
sekolah	0.01694915254237288	0.05604892715577821
tempat	0.01694915254237288	0.06102792916473012
lawat	0.01694915254237288	0.11709753863246136
sungguh	0.01694915254237288	0.0677971844534044
riah	0.01694915254237288	0.098476999136690019
kawasan	0.01694915254237288	0.06689701792265951
luas	0.01694915254237288	0.0547159927944373
perlu	0.01694915254237288	0.11222157130277016
semangat	0.01694915254237288	0.07594926344690117
daki	0.01694915254237288	0.04980445535191693
hingg	0.01694915254237288	0.12396982860039636
ake	0.01694915254237288	0.12396982860039636
puncak	0.01694915254237288	0.043621102786933735
lelah	0.01694915254237288	0.060024727954724515
la	0.01694915254237288	0.11709753863246136

Lampiran A-2. 3Hasil Pembobotan

Out [21]:

	term	rank
41	pintu	21.180261
44	biaya	22.860316
46	wisatawan	24.669687
43	bayar	25.123600
45	situs	25.239753
35	tinggal	25.282731
39	tur	25.668161
40	jual	26.389050
47	tarik	27.384263
32	tiket	27.503718
31	bawa	27.746461
36	foto	28.722042
48	ramai	29.041340
42	kali	29.639625
49	ajaib	29.951131
37	nikmat	30.033966
38	yogyakarta	30.346346
28	alam	30.351781
20	harga	30.502504
19	ambil	30.848819
14	jam	31.445637
17	milik	31.475576
29	hotel	31.505465
33	layak	32.226808
12	masuk	32.749720
34	buddha	32.974247
27	wisata	33.973086
30	panas	35.229111
26	bangun	35.740693
22	indonesia	35.813618
24	pandu	36.830473

Lampiran A-2. 4 Ranking Pembobotan

Lampiran A-2. 5 Pelabelan *Lexicon*

```
lexicon= pd.read_csv('LexiconIndo.csv').reset_index(drop=True)
```

```

negasi = ['bukan','tidak','ga','gk']
lexicon_word = lexicon['word'].to_list()
lexicon_number_word = lexicon['number_of_word']
data['data_kembali'] = data['data_kembali'].astype(str)
len(lexicon_word)
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize
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(data)):

    nsen = senrow.shape[0]

    words = word_tokenize(data['data_kembali'] [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.to_csv("bobot_ke3.csv")
cek_df = pd.DataFrame([])
cek_df['text'] = data['data_kembali'].copy()
cek_df['Sentiment'] = df_sen['Sentiment'].copy()
cek_df.head(5)
cek_df.to_csv("bobot_lengkap1.csv")

import pandas as pd
from nltk import word_tokenize
from string import punctuation
from sklearn.base import TransformerMixin
from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import FunctionTransformer
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.SVM import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix

data2 = pd.read_csv('bobot_lengkap1.csv', encoding="ISO-8859-1")
data = pd.read_csv('bobot_lengkap1.csv', encoding="ISO-8859-1")
data= data.astype(str)
X = data['content']

```

```
y = data['label']
```

Lampiran A-2. 6 Klasifikasi

```
bi_vectorizer = CountVectorizer(ngram_range=(2,2), analyzer='word')
neg_data = data.query('label == "negatif"')
bi_sparse_matrix = bi_vectorizer.fit_transform(neg_data['content'].apply(lambda x:
x.lower()))
bi_frequencies = sum(bi_sparse_matrix).toarray()[0]

bigram = pd.DataFrame(bi_frequencies,
                      index=bi_vectorizer.get_feature_names(),
                      columns=['frequency'])
bigram = bigram.sort_values(by='frequency', ascending=False)
bigram.head(10)

bi_vectorizer = CountVectorizer(ngram_range=(2,2), analyzer='word')
pos_data = data.query('label == "positif"')
bi_sparse_matrix = bi_vectorizer.fit_transform(pos_data['content'].apply(lambda x:
x.lower()))
bi_frequencies = sum(bi_sparse_matrix).toarray()[0]
bigram = pd.DataFrame(bi_frequencies,
                      index=bi_vectorizer.get_feature_names(),
                      columns=['frequency'])
bigram = bigram.sort_values(by='frequency', ascending=False)
bigram.head(10)

tri_vectorizer = CountVectorizer(ngram_range=(3,3), analyzer='word')
neg_data = data.query('label == "negatif"')
tri_sparse_matrix = tri_vectorizer.fit_transform(neg_data['content'].apply(lambda x:
x.lower()))
tri_frequencies = sum(tri_sparse_matrix).toarray()[0]
```



```

trigram = pd.DataFrame(tri_frequencies,
                        index=tri_vectorizer.get_feature_names(),
                        columns=['frequency'])

trigram = trigram.sort_values(by='frequency', ascending=False)

trigram.head(50)

tri_vectorizer = CountVectorizer(ngram_range=(3,3), analyzer='word')

pos_data = data.query('label == "positif"')

tri_sparse_matrix = tri_vectorizer.fit_transform(pos_data['content'].apply(lambda x:
x.lower()))

tri_frequencies = sum(tri_sparse_matrix).toarray()[0]

trigram = pd.DataFrame(tri_frequencies,
                        index=tri_vectorizer.get_feature_names(),
                        columns=['frequency'])

trigram = trigram.sort_values(by='frequency', ascending=False)

trigram.head(10)

X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size = 0.2,
                                                    random_state = 42)

class DenseTransformer(TransformerMixin):
    def fit(self, X, y=None, **fit_params):
        return self
    def transform(self, X, y=None, **fit_params):
        return X.todense()

def text_preprocessing(col: pd.Series) -> pd.Series:
    global stopwords
    result = []
    for text in col:
        # lowercase
        text_lower = str(text).lower()

```



```

        columns = ['Predicted {}'.format(i)
                   for i in model_nb.classes_]
print('\nConfusion Matrix:')
print(cmatrix_nb)
estimators_SVM = [
    ('preprocessor', FunctionTransformer(text_preprocessing)),
    ('vectorizer', TfidfVectorizer()),
    ('denser', DenseTransformer()),
    ('classifier', SVC(kernel='RBF'))
]
model_SVM = Pipeline(estimators_SVM)
model_SVM.fit(X_train, y_train)
y_pred_SVM = model_SVM.predict(X_test)
print('Report:')
print(classification_report(y_test, y_pred_SVM))
cmatrix_SVM = pd.DataFrame(confusion_matrix(y_test, y_pred_SVM),
                           index = ['Actual {}'.format(i)
                                     for i in model_SVM.classes_],
                           columns = ['Predicted {}'.format(i)
                                       for i in model_SVM.classes_])
print('\nConfusion Matrix:')
print(cmatrix_SVM)

```

Lampiran A-2. 7 Visualisasi dan Asosiasi

```

import pickle
import collections, numpy
Data=data3['content']
#bagian klasifikasi untuk melabelkan.
input_transformed = vectorizer.transform(Data)

```

```

SVM_linier_prediction = SVM_Clasifier.predict(input_transformed)
SVM_kernel_linier_prediction = SVM_Clasifier_kernel_linier.predict(input_transformed)
SVM_poly_prediction = SVM_Clasifier_kernel_poly.predict(input_transformed)
SVM_sigmoid_prediction = SVM_Clasifier_kernel_sigmoid.predict(input_transformed)
SVM_RBF_prediction = SVM_Clasifier_RBF.predict(input_transformed)
SVM_linier_prediction
#jika keluar banyak row benar, harusnya keluarnya positif negatif.
result = pd.DataFrame({"text awal" : Data,"label" : SVM_linier_prediction})
result.to_csv("HasilSVM.csv", index = False)
df = pd.read_csv('HasilSVM.csv')
df
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize
from nltk.probability import FreqDist
visual = pd.DataFrame([])
visual['teks'] = df['label'].values.copy()
visual
#Keseluruhan
word_dict = {}
for i in range(0,len(visual['teks'])):
    sentence = visual['teks'][i]
    word_token = word_tokenize(str(sentence))
    for j in word_token:
        if j not in word_dict:
            word_dict[j] = 1
        else:
            word_dict[j] += 1

```

```

df_top = pd.DataFrame((word_dict.items()))
df_top = df_top.rename(columns={0:'word', 1:'counts'})
df_top = df_top.sort_values(by='counts', ascending=False)
df_top.head(10)

import matplotlib.pyplot as plt

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

pos_dict = { }
for i in range(0, len(positif['text awal'])):
    sentence = positif['text awal'][i]
    token = word_tokenize(str(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.to_csv('frekuensipositif.csv')

positif_top.head(60).sort_values(by='counts',ascending=True).plot.barh(x='word',
y='counts', rot=0,figsize=(20,20),fontsize=20)

plt.savefig('grafik positif .png',bbox_inches='tight')

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.to_csv('frekuensinegatif.csv')

negatif_top.head(60).sort_values(by='counts',ascending=True).plot.barh(x='word',
y='counts', rot=0,figsize=(20,20),fontsize=20)

plt.savefig('grafik negatif .png',bbox_inches='tight')

pos_word = str(pos['text awal'])

neg_word = str(neg['text awal'])

pip install mlxtend

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

```

```
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
warnings.filterwarnings('ignore')
data = pd.read_csv('komentar_positif_lat.csv', encoding= 'unicode_escape')
data = data.join(data['komentar'].str.split(' ', expand=True).add_prefix(""))
print(data[['2']])
del data['komentar']
items=(data['0'].unique())
items
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
rules = association_rules(freq_items, metric="confidence", min_threshold=0.25)
rules
rules.to_csv('Hasilassosiasipost.csv')
data = pd.read_csv('komentar_neg_lat.csv', encoding= 'unicode_escape')
```

```

data = data.join(data['komentar'].str.split(' ', expand=True).add_prefix(""))
print(data[['2']])
del data['komentar']
items=(data['0'].unique())
items
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)
rules = association_rules(freq_items, metric="confidence", min_threshold=0.25)
rules.head(10)
rules.to_csv('Hasilassosiasineg.csv')
rules.sort_values(
    by="lift",
    ascending=False
)
rules.sort_values(

```



```

by="support",
ascending=False
)
rules.sort_values(
by="confidence",
ascending=False
)

```

	antecedents	consequents	antecedent support	consequent support	...	confidence	lift	leverage	conviction
0	(sibuk)	(terbit)	0.034169	0.255885	...	0.400000	1.563205	0.004924	1.240192
1	(sibuk)	(indah)	0.034169	0.277904	...	0.400000	1.439344	0.004172	1.203493
2	(sibuk)	(foto)	0.034169	0.131739	...	0.366667	2.783285	0.008027	1.370939
3	(sibuk)	(berjalan)	0.034169	0.163250	...	0.311111	1.905736	0.005052	1.214637
4	(menginap)	(terbit)	0.023918	0.255885	...	0.523810	2.047054	0.006408	1.562642
...
1363	(pemandangan, manohara)	(terbit, hotel)	0.014047	0.080486	...	0.837838	10.409740	0.010639	5.670337
1364	(pemandangan, hotel)	(terbit, manohara)	0.045558	0.037206	...	0.258333	6.943367	0.010074	1.298150
1365	(terbit, manohara)	(pemandangan, hotel)	0.037206	0.045558	...	0.316327	6.943367	0.010074	1.396049
1366	(manohara, hotel)	(pemandangan, terbit)	0.039484	0.082384	...	0.298077	3.618132	0.008516	1.307288
1367	(manohara)	(pemandangan, terbit, hotel)	0.045558	0.029233	...	0.258333	8.837013	0.010437	1.308899

Lampiran A-2. 8 Asosiasi Positif

	antecedents	consequents	antecedent support	consequent support	...	confidence	lift	leverage	conviction
280	(bangun, menunggu)	(terbit)	0.010870	0.114130	...	1.00	8.761905	0.009629	inf
261	(pengalaman, pagi)	(terbit)	0.016304	0.114130	...	1.00	8.761905	0.014444	inf
197	(membeli, mobil)	(jalan)	0.010870	0.141304	...	1.00	7.076923	0.009334	inf
510	(terbit, berbeda, berjalan)	(bagus)	0.010870	0.067935	...	1.00	14.720000	0.010131	inf
509	(bagus, berbeda, berjalan)	(terbit)	0.010870	0.114130	...	1.00	8.761905	0.009629	inf
...
205	(mobil)	(berjalan, parkir)	0.043478	0.019022	...	0.25	13.142857	0.010043	1.307971
66	(jawa)	(terletak)	0.076087	0.038043	...	0.25	6.571429	0.016127	1.282609
58	(hotel)	(indah)	0.065217	0.127717	...	0.25	1.957447	0.007975	1.163043
57	(jawa)	(indah)	0.076087	0.127717	...	0.25	1.957447	0.009304	1.163043
0	(mobil)	(sunrise)	0.043478	0.065217	...	0.25	3.833333	0.008034	1.246377

560 rows × 9 columns

Lampiran A-2. 9 Asosiasi Negatif

Lampiran B-1 Olah Data

Lampiran B-1 1 Rekap data form harapan

Responden	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	Total
1	4	5	5	3	5	4	5	5	4	5	4	49
2	3	3	4	3	4	4	4	4	3	4	4	40

Responde n	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	Tota l
3	5	5	3	3	5	4	5	5	3	3	4	45
4	1	5	4	3	4	4	4	4	4	4	4	41
5	2	5	5	4	5	4	5	5	3	5	4	47
6	2	5	4	3	4	4	4	4	3	4	4	41
7	5	3	3	3	5	4	5	5	4	3	4	44
8	2	5	4	3	4	4	4	4	4	4	4	42
9	1	5	5	3	5	4	5	5	3	5	4	45
10	3	4	4	3	4	4	4	4	2	4	4	40
11	1	4	3	4	5	5	5	5	4	3	5	44
12	4	5	4	3	4	4	4	4	3	4	4	43
13	4	5	5	3	5	4	5	5	3	5	4	48
14	1	4	4	4	4	4	4	4	4	4	3	40
15	1	5	3	4	5	4	5	5	4	3	3	42
16	4	5	4	4	4	4	4	4	3	4	4	44
17	3	5	5	4	5	4	5	5	2	5	5	48
18	4	5	4	3	4	4	4	4	2	4	5	43
19	4	5	3	4	5	4	5	5	2	3	4	44
20	3	4	4	4	4	3	4	4	3	4	5	42
21	3	4	5	2	5	5	5	5	4	5	4	47
22	2	4	4	4	4	3	4	4	4	4	4	41
23	2	4	3	4	5	5	5	5	3	3	3	42
24	3	4	4	2	4	3	4	4	2	4	4	38
25	2	5	5	4	5	5	5	5	3	5	5	49
26	3	4	4	4	4	3	4	4	2	4	3	39
27	3	4	3	2	5	5	5	5	3	3	3	41
28	3	3	4	2	4	3	4	4	4	4	5	40
29	4	4	5	4	5	5	5	5	4	5	3	49
30	5	4	4	5	4	3	4	4	3	4	5	45
31	4	4	3	2	5	5	5	5	2	3	3	41
32	3	2	4	5	4	3	4	4	2	4	3	38
33	5	3	5	5	5	5	5	5	2	5	5	50
34	2	3	4	2	4	3	4	4	3	4	5	38
35	5	3	3	5	5	5	5	5	4	3	5	48
36	1	3	4	2	4	3	4	4	4	4	5	38
37	2	3	5	5	5	5	5	5	4	5	5	49
38	4	4	4	5	4	5	4	5	3	4	3	45
39	1	5	3	2	5	5	5	4	2	3	3	38
40	1	5	4	5	4	5	4	5	2	4	3	42
41	3	5	5	1	5	4	5	4	3	5	5	45
42	2	3	4	5	4	5	4	5	3	4	5	44
43	5	4	3	5	5	3	5	3	2	3	5	43
44	2	4	4	2	4	5	4	5	1	4	5	40
45	5	2	5	2	5	3	5	4	1	3	3	38

Responden	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	Total
46	5	3	4	1	4	5	4	5	1	5	5	42
47	2	4	3	5	3	3	5	4	1	3	5	38
48	1	2	4	1	4	5	4	5	1	4	3	34
49	1	1	1	1	1	1	1	1	1	1	1	11
50	4	4	4	5	4	5	4	3	1	4	3	41
Total	145	199	197	167	219	203	221	220	138	196	201	2106
Rata rata	2.9	3.98	3.94	3.34	4.38	4.06	4.42	4.4	2.76	3.92	4.02	

Lampiran B-1 2 Rekap data kenyataan / persepsi pelanggan

Responden	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	Total
1	2	1	3	2	5	3	2	5	4	3	2	32
2	3	2	4	2	5	2	3	5	4	3	2	35
3	2	1	3	3	4	3	4	4	4	3	2	33
4	3	2	4	2	5	2	3	5	4	3	2	35
5	2	1	3	2	5	3	2	5	4	3	2	32
6	3	2	4	3	4	2	3	4	5	4	2	36
7	2	1	3	2	5	3	4	5	5	4	2	36
8	3	2	4	2	5	2	3	5	5	4	2	37
9	2	1	3	3	4	3	2	4	5	4	2	33
10	3	2	4	2	4	2	3	5	5	5	2	37
11	2	1	3	2	5	3	4	5	5	5	2	37
12	3	2	4	3	4	2	3	4	5	4	2	36
13	2	1	3	2	5	3	2	5	5	4	2	34
14	3	2	4	2	5	2	3	5	4	4	2	36
15	2	1	3	2	4	3	4	4	5	4	2	34
16	3	2	4	3	5	2	3	5	4	4	2	37
17	2	1	3	2	5	3	2	5	4	3	3	33
18	3	2	4	2	4	2	3	4	4	3	3	34
19	2	1	3	3	5	3	4	5	5	3	3	37
20	3	2	4	2	5	2	3	5	5	3	3	37
21	2	1	3	2	4	3	2	4	5	3	2	31
22	3	2	4	3	5	2	3	5	5	3	2	37
23	2	1	3	2	5	2	4	5	5	3	2	34
24	3	2	4	2	4	3	3	4	5	3	2	35
25	2	1	3	3	5	3	2	5	5	3	2	34
26	3	2	4	2	5	2	3	5	4	3	2	35
27	2	1	3	2	4	2	4	4	4	3	2	31
28	3	2	4	2	5	2	3	5	4	4	2	36
29	2	1	3	3	5	3	2	5	5	4	2	35
30	3	2	4	2	4	2	3	4	5	3	2	34

31	2	1	3	2	5	3	4	5	5	3	2	35
32	2	2	4	3	5	2	3	5	5	3	2	36
33	3	1	3	3	4	3	2	4	5	3	3	34
34	2	2	4	3	5	2	3	5	5	3	3	37
35	3	1	3	3	5	3	4	5	5	4	3	39
36	2	2	4	2	4	2	3	4	5	4	2	34
37	3	1	3	2	5	3	2	5	4	4	2	34
38	2	2	4	3	5	2	3	5	5	4	2	37
39	3	1	3	2	4	1	4	4	4	4	2	32
40	2	2	4	3	5	2	3	5	5	3	3	37
41	3	1	3	2	5	2	2	5	5	3	3	34
42	2	2	4	3	4	1	3	4	5	3	3	34
43	3	1	3	3	5	4	4	5	5	4	3	40
44	2	2	4	2	5	1	3	5	5	4	2	35
45	3	1	3	3	4	1	2	4	5	4	2	32
46	2	2	4	2	5	5	3	5	5	3	2	38
47	3	1	3	3	4	1	4	4	5	4	2	34
48	2	2	4	2	4	1	3	5	5	3	2	33
49	1	1	1	1	1	1	1	1	1	1	1	11
50	2	2	3	2	5	1	3	5	5	3	2	33
Rata rata	2.44	1.5	3.44	2.36	4.56	2.3	2.96	4.6	4.64	3.44	2.2	

Lampiran B-1 3 Lampiran Pengolahan *Kano*

Respon de n	X1	Eva luas i	X2	Eva luas i	X3	Eva luas i	X4	Eva luas i	X5	Eva luas i	X6	Eva luas i	X7	Eva luas i	X8	Eva luas i	X9	Eva luas i	X1 0	Eva luas i	X1 0	Eva luas i
1	2 3	I	2 5	I	4 4	I	3 5	M	1 3	A	2 4	I	2 4	I	3 4	I	2 4	I	2 4	I	1 5	O
2	2 3	I	2 5	I	4 4	I	3 5	M	1 3	A	2 4	I	2 4	I	3 4	I	2 5	M	2 4	I	1 5	O
3	2 3	I	2 5	I	4 4	I	3 5	M	1 3	A	2 4	I	1 5	O	3 3	i	2 5	M	2 4	I	1 5	O
4	2 3	I	2 5	I	4 4	I	3 5	M	2 3	I	2 4	I	2 4	I	3 4	I	2 5	M	2 4	I	1 5	O
5	2 3	I	2 5	I	4 4	I	3 5	M	1 4	A	2 4	I	1 5	O	2 4	I	2 5	M	2 4	I	1 5	O
6	2 3	I	3 5	I	4 4	I	3 4	M	1 4	A	3 4	I	2 4	I	3 4	I	2 5	M	2 4	I	1 5	O
7	2 4	I	3 5	I	4 4	I	2 4	M	1 5	O	3 4	I	1 5	O	3 4	I	2 5	M	2 4	I	1 5	O
8	2 4	I	3 5	I	2 4	I	2 4	M	1 4	A	3 4	I	2 4	I	3 5	M	2 5	M	2 4	I	1 5	O
9	2 4	I	3 5	I	2 4	I	2 5	M	1 4	A	3 5	M	2 4	I	2 4	I	2 5	M	2 4	I	1 5	O
10	2 4	I	2 5	I	2 4	I	2 5	M	2 5	M	3 5	M	1 5	O	2 4	I	1 4	A	2 4	I	1 5	O
11	2 3	I	2 5	I	2 4	I	2 5	M	1 5	O	4 5	M	2 4	I	3 4	I	1 4	A	1 5	O	1 5	O
12	2 4	I	2 5	I	2 4	I	4 5	M	2 5	M	3 5	M	1 5	O	3 5	M	1 4	A	1 5	O	1 5	O
13	2 4	I	2 5	I	4 4	I	3 5	M	1 5	O	3 5	M	2 4	I	3 4	I	1 4	A	1 5	O	1 5	O
14	2 4	i	2 5	I	4 4	I	3 5	M	2 5	M	3 5	M	2 4	I	3 4	I	1 4	A	1 4	A	1 5	O
15	2 4	I	2 5	I	2 4	I	3 5	M	1 5	O	2 5	M	1 5	O	3 4	I	1 5	O	1 4	A	1 5	O
16	2 4	I	2 5	I	4 4	I	3 5	M	2 4	I	2 5	M	1 5	O	3 5	M	1 5	O	1 4	A	1 5	O
17	2 4	I	2 5	I	4 4	I	3 5	M	2 4	I	1 5	O	1 5	O	2 3	I	1 5	O	1 4	A	1 5	O
18	2 4	I	2 5	I	4 4	I	3 5	M	2 3	I	2 5	M	1 5	O	2 4	I	1 5	O	1 4	A	1 5	O
19	2 5	M	3 5	I	4 3	I	2 5	M	1 5	O	1 4	A	2 4	I	3 4	I	1 5	O	1 4	A	1 4	A
20	1 5	O	3 5	I	4 3	I	2 5	M	2 5	M	2 4	I	2 4	I	3 4	I	1 5	O	1 4	A	1 4	A
21	1 5	O	3 5	I	4 4	I	2 5	M	1 5	O	1 4	A	1 5	O	3 3	I	1 5	O	1 4	A	1 4	A
22	2 5	M	3 5	I	4 4	I	2 5	M	2 5	M	2 4	I	1 5	O	3 4	I	2 5	o	1 4	A	1 4	A
23	2 5	M	3 5	I	4 4	I	2 4	I	2 4	I	2 5	M	2 4	I	3 3	I	2 4	I	1 4	A	1 4	A

24	3	5	M	3	5	I	4	4	I	2	4	I	2	5	M	2	5	M	1	5	O	3	4	I	2	4	I	3	5	M	3	4	I
25	2	4	I	3	5	I	4	3	I	2	4	I	1	5	O	3	5	M	1	5	O	3	3	I	1	4	A	2	5	M	2	5	M
26	2	4	I	2	5	I	2	3	I	2	5	M	2	5	M	2	5	M	1	5	O	3	4	I	1	4	A	2	4	I	2	5	M
27	2	4	I	2	5	I	2	3	I	2	5	M	3	5	M	2	5	M	1	5	O	3	3	I	2	4	I	2	4	I	2	4	I
28	2	3	I	2	5	I	3	3	I	2	5	M	3	5	M	1	5	O	1	5	O	3	4	I	2	5	M	2	4	I	2	4	I
29	2	3	I	2	5	I	3	3	I	2	5	M	2	5	M	3	5	M	2	4	I	2	3	I	1	5	O	1	5	O	1	4	A
30	1	3	A	2	5	I	3	3	I	3	5	M	1	4	O	3	4	I	1	5	O	2	3	I	2	5	M	1	5	O	1	4	A
31	1	3	A	2	5	I	3	3	I	3	5	M	2	4	I	3	5	M	1	5	O	3	3	I	2	5	M	1	4	A	1	5	O
32	1	3	A	2	5	I	3	3	I	2	4	I	1	3	A	2	5	M	1	5	O	3	3	I	2	5	M	1	4	A	1	4	A
33	2	4	I	4	5	I	3	3	I	2	4	I	2	5	M	2	4	I	1	5	O	3	4	I	1	5	O	1	4	A	1	4	A
34	2	4	I	4	5	I	3	3	I	2	4	I	1	5	O	1	4	A	2	4	I	3	3	I	1	5	O	1	4	A	1	4	A
35	2	4	I	4	5	I	4	3	I	2	4	I	2	5	M	2	5	M	1	5	O	2	4	I	1	5	O	1	4	A	1	4	A
36	2	4	I	2	5	I	2	3	I	2	4	I	2	5	M	1	4	A	2	4	I	3	4	I	1	4	A	4	4	I	3	4	I
37	2	4	I	2	5	I	4	3	I	2	5	M	2	5	M	2	4	I	2	4	I	3	3	I	1	4	A	4	4	I	2	4	I
38	2	4	I	2	5	I	2	3	I	2	5	M	1	5	O	1	4	A	1	5	O	3	4	I	1	4	A	2	4	I	2	4	I
39	2	4	I	2	5	I	4	3	I	2	5	M	2	5	M	2	5	M	1	5	O	2	4	I	1	4	A	2	4	I	2	4	I
40	2	4	I	2	5	I	2	3	I	2	5	M	1	5	O	2	5	M	1	5	O	2	4	I	1	4	A	2	4	I	2	4	I
41	2	4	I	2	5	I	3	3	I	3	5	M	2	5	M	2	5	M	1	5	O	3	4	I	2	5	M	2	4	I	2	3	I
42	2	4	I	2	5	I	2	3	I	3	5	M	2	5	M	3	5	M	2	4	I	3	4	I	2	5	M	1	4	A	1	5	O
43	2	4	I	2	5	I	3	3	I	3	5	M	2	4	I	2	5	M	2	4	I	3	3	I	2	5	M	1	5	O	1	3	A
44	2	4	I	2	5	I	3	3	I	3	5	M	1	4	A	3	5	M	1	5	O	3	4	I	2	5	M	1	5	O	1	5	O
45	2	4	I	2	5	I	3	3	I	3	5	M	2	3	I	2	5	M	1	5	O	3	4	I	1	5	O	1	5	O	1	3	A
46	2	4	I	2	3	I	3	3	I	2	4	I	3	5	M	2	4	I	2	4	I	3	4	I	2	5	M	1	5	O	1	5	O
47	2	4	I	2	5	I	3	3	I	2	4	I	2	5	M	1	5	O	1	5	O	2	3	I	2	5	M	1	4	A	1	3	A
48	2	4	I	2	5	I	3	3	I	2	4	I	3	5	M	1	5	O	1	5	O	3	4	I	2	5	M	1	4	A	1	5	O
49	2	4	I	2	3	I	3	3	I	2	5	M	3	5	M	2	5	M	2	4	I	3	3	I	2	4	I	3	4	I	3	5	M
50	2	4	I	3	5	I	3	3	I	2	5	M	2	4	I	1	5	O	2	4	I	3	4	I	2	4	I	3	4	I	3	5	M

A	3	0	0	0	9	5	0	0	12	18	14
O	2	0	0	0	11	5	29	0	13	9	23
M	4	0	0	39	21	26	0	3	19	2	4
I	41	50	50	11	9	14	21	47	6	21	9
R	0	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0
Kepu tusan	41	50	50	39	21	26	29	47	19	18	23

Biaya Borobudur murah untuk turis lokal

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

Biaya Borobudur murah untuk turis lokal

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

Biaya Borobudur murah untuk turis mancanegara

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

Biaya Borobudur murah untuk turis mancanegara

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

Pengunjung dapat menikmati Pemandangan Terbit

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

Pengunjung dapat menikmati Pemandangan Terbit

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

Menikmati "Kandang Gajah" Candi Borobudur

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

Menikmati "Kandang Gajah" Candi Borobudur

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

Pemandangan Candi Borobudur Indah

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

Pemandangan Candi Borobudur Indah

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

jarak antar tempat di candi borobudur berdekatan

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

jarak antar tempat di candi borobudur berdekatan

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

Berjalan di Candi Borobudur Rileks (Tidak melelahkan)

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

Berjalan di Candi Borobudur Rileks (Tidak melelahkan)

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

Kondisi Jalan Borobudur bagus

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

Kondisi Jalan Borobudur bagus

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

Jalan keluar candi sepi penjual

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

Jalan keluar candi sepi penjual

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

Tempat parkir yang dekat dengan kawasan pertokoan

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

Tempat parkir yang dekat dengan kawasan pertokoan

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

Terdapat panduan pengunjung (buku petunjuk/pamflet)

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

Terdapat panduan pengunjung (buku petunjuk/pamflet)

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

Submit

Clear form