

PENGEMBANGAN APLIKASI JURNAL EMOSI BERBASIS PROGRESSIVE WEB APP



Disusun Oleh:

N a m a : Rafif Wicaksana Putra
NIM : 18523056

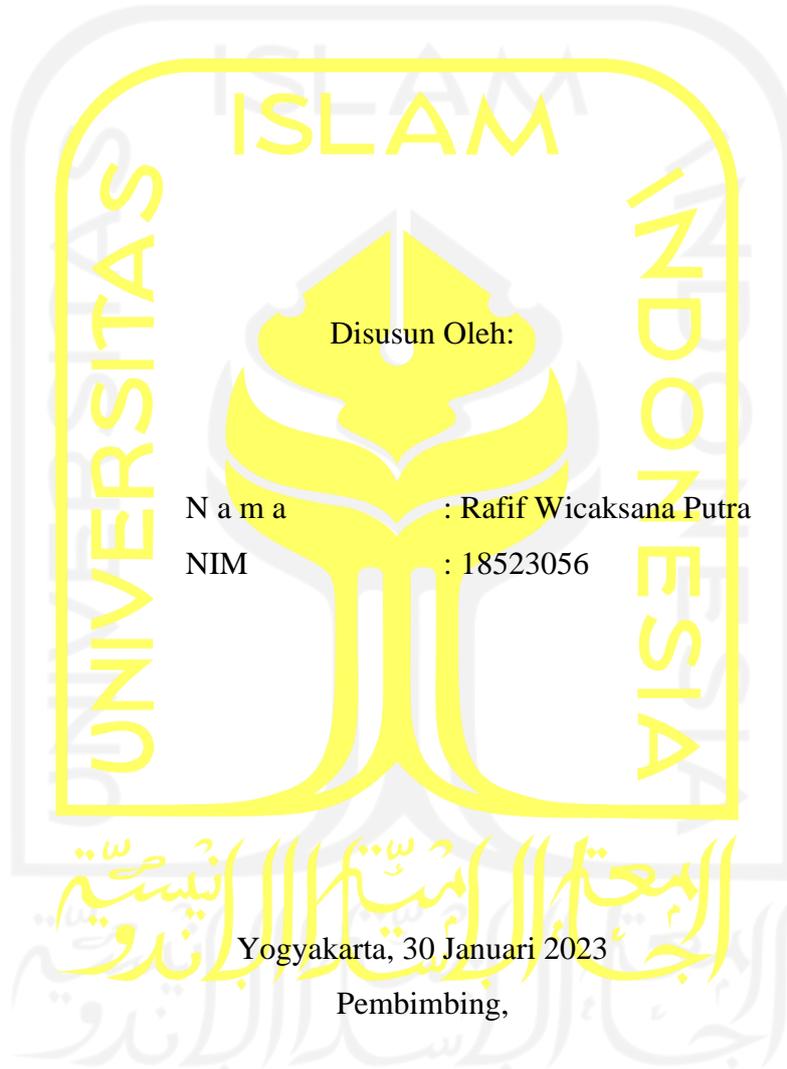
**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2022

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**PENGEMBANGAN APLIKASI JURNAL EMOSI BERBASIS
PROGRESSIVE WEB APP**

TUGAS AKHIR




(Arrie Kurniawardhani, S.Si., M.Kom.)

HALAMAN PENGESAHAN DOSEN PENGUJI

**PENGEMBANGAN APLIKASI JURNAL EMOSI BERBASIS
PROGRESSIVE WEB APP**

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 30 Januari 2023

Tim Penguji

Arrie Kurniawardhani, S.Si., M.Kom.



Anggota 1

Dhomas Hatta Fudholi, S.T., M.Eng.,
Ph.D.



Anggota 2

Sheila Nurul Huda, S.Kom., M.Cs.



Mengetahui,

Ketua Program Studi Informatika – Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia



(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Rafif Wicaksana Putra

NIM : 18523056

Tugas akhir dengan judul:

**PENGEMBANGAN APLIKASI JURNAL EMOSI BERBASIS
PROGRESSIVE WEB APP**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 30 Januari 2023



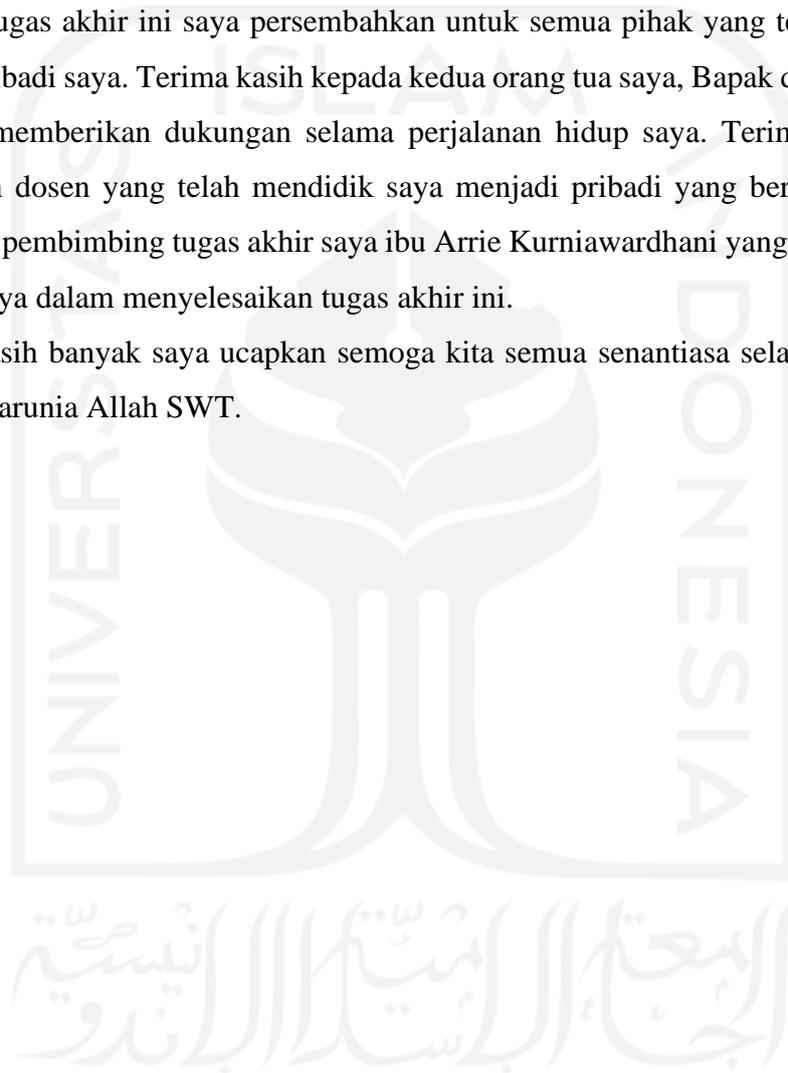
(Rafif Wicaksana Putra)

HALAMAN PERSEMBAHAN

Alhamdulillah Robbil 'Alamin, segala puja dan puji syukur bagi Allah *Subhanahu Wa Ta'ala* atas segala rahmat dan karunia Nya penulis dapat menyelesaikan laporan skripsi ini hingga akhir. Sholawat dan salam saya ucapkan kepada junjungan besar kita Nabi Muhammad *Shallallahu 'alaihi Wassalam* yang telah membimbing kita menuju jalan yang lurus dan benar, serta menjadi teladan bagi kita semua.

Laporan tugas akhir ini saya persembahkan untuk semua pihak yang telah mendukung terbentuknya pribadi saya. Terima kasih kepada kedua orang tua saya, Bapak dan Ibu, yang tak henti-hentinya memberikan dukungan selama perjalanan hidup saya. Terima kasih kepada semua guru dan dosen yang telah mendidik saya menjadi pribadi yang berilmu dan baik, khususnya pada pembimbing tugas akhir saya ibu Arrie Kurniawardhani yang telah membantu membimbing saya dalam menyelesaikan tugas akhir ini.

Terima kasih banyak saya ucapkan semoga kita semua senantiasa selalu berada dalam lindungan dan karunia Allah SWT.



HALAMAN MOTO

“Everything that happens happens as it should, and if you observe carefully, you will find this to be so.” - Marcus Aurelius

*“saara 'ala darbi washala, Siapa yang menapaki jalannya akan sampai pada tujuan”
-Pepatah Arab*



KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh, Alhamdulillah, atas rahmat dan karunia Allah *Subhanahu Wata'ala* sehingga penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul **“Pengembangan Aplikasi Jurnal Emosi Berbasis Progressive Web App”**

Penyusunan laporan tugas akhir ini ditujukan untuk memenuhi sebagian syarat dalam menempuh gelar sarjana pada Program Studi Informatika Fakultas Teknologi Industri, Universitas Islam Indonesia. Dalam penyusunan laporan tugas akhir ini banyak hambatan dan tantangan yang dihadapi yang tidak akan mencapai keberhasilan tanpa dukungan, bimbingan dan doa dari berbagai pihak yang membantu saya. Maka dari itu saya sebagai penulis ingin menyampaikan terima kasih kepada semua pihak yang memberikan dukungan dan bantuannya dalam pelaksanaan tugas akhir ini., terutama kepada:

1. Allah SWT yang selalu memberikan karunia berupa kesehatan dan rezeki sehingga penulis mampu menyelesaikan tugas akhir ini.
2. Kedua orang tua yaitu Bapak dan Ibu saya yang telah membesarkan dan mendidik saya dalam seluruh hidup saya.
3. Segenap dosen informatika UII yang telah memberikan ilmu berharga dan bermanfaat selama masa perkuliahan.
4. Ibu Arrie Kurniawardhani S.Si. M.Kom. selaku dosen pembimbing saya yang telah meluangkan waktu, tenaga dan ilmu dalam membimbing saya hingga dapat menyelesaikan tugas akhir ini.
5. Seluruh pihak yang telah membantu saya selama masa perkuliahan dan penyelesaian tugas akhir.

Penulis menyadari bahwa masih terdapat banyak kekurangan dalam penulisan dan pembuatan aplikasi Jurnal Emosi pada tugas akhir ini. Untuk itu penulis siap menerima kritik dan saran untuk menyempurnakan laporan ini menjadi lebih baik lagi. Akhir kata, penulis berharap agar Laporan Tugas Akhir ini dapat bermanfaat bagi semua pihak. Amin

Yogyakarta, 30 Januari 2023



(Rafif Wicaksana Putra)

SARI

Dalam kehidupan setiap individu tidak akan terlepas dari tekanan dan tuntutan. Tuntutan dapat berupa hal-hal yang telah terjadi, ataupun hal-hal baru yang mungkin akan terjadi, namun apabila tekanan dan tuntutan tersebut tidak teratasi dengan baik, maka akan membuat individu tertekan sehingga mengalami gangguan kesehatan mental berupa stres. Dalam mengatasi stres dapat dilakukan upaya menangani masalah penyebabnya dengan berfokus dalam menyelesaikan masalah tersebut atau yang disebut juga dengan koping. Salah satu metode koping yang dapat memberikan dampak positif dalam mempengaruhi kondisi emosi, pikiran, dan tindakan adalah dengan menulis buku harian. Melihat hal tersebut maka dibutuhkan sebuah aplikasi yang dapat menjadi alat koping penanganan stres untuk mengelola emosi dan suasana hati yaitu aplikasi Jurnal Emosi. Aplikasi Jurnal Emosi diciptakan untuk membantu mencatat jurnal harian, dan melacak perubahan emosi pengguna sehari-hari. Aplikasi Jurnal Emosi dikembangkan dengan memanfaatkan teknologi *Progressive Web App* (PWA) yang memungkinkan pengembangan aplikasi Jurnal Emosi untuk dapat diakses di berbagai macam perangkat. Dalam proses pengembangannya digunakan metodologi *Rapid Application Development* dengan tools pendukung berupa *framework* VueJS untuk membuat *frontend* aplikasi. Tujuan penelitian ini untuk mengetahui bagaimana menerapkan *Progressive Web App* serta melakukan uji coba menggunakan metode *black box*, *lighthouse testing*, dan *system usability scale (SUS)*. Berdasarkan penelitian yang telah dilakukan penerapan *Progressive Web App* berhasil memberikan aplikasi Jurnal Emosi kemampuan untuk diakses di perangkat *desktop* dan *smartphone* yang juga tetap dapat digunakan secara *offline* dengan performa yang baik. Diharapkan dengan adanya aplikasi ini dapat membantu pengguna dalam mencatat dan mengelola perubahan kondisi emosi sehari-hari.

Kata kunci: Jurnal Emosi, *Progressive Web App*, *Rapid Application Development*.

GLOSARIUM

- Backend* Bagian dari pengembangan perangkat lunak yang bertanggung jawab pada operasi yang terjadi di belakang layar yang tidak dapat dilihat langsung oleh pengguna.
- Cross-Platform* Pendekatan pengembangan aplikasi yang berfokus untuk membangun aplikasi yang dapat diakses menggunakan berbagai device yang berbeda.
- Framework* Kerangka kerja pemrograman yang digunakan untuk mengembangkan aplikasi.
- Frontend* Bagian dari pengembangan perangkat lunak yang bertanggung jawab pada tampilan dan antarmuka aplikasi yang dapat berinteraksi dengan pengguna.
- Input* Data atau perintah yang berasal dari masukan pengguna aplikasi.
- Native* Metode pengembangan aplikasi *mobile* yang dibangun dengan bahasa yang khusus tergantung sistem operasi perangkat tersebut.
- Package Manager* Program yang memudahkan dalam mengelola paket yang dibutuhkan dalam pengembangan perangkat lunak.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI	viii
GLOSARIUM.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Metodologi Pengembangan.....	3
1.7 Sistematika Penulisan.....	3
BAB II LANDASAN TEORI.....	5
2.1 Jurnal Emosi.....	5
2.2 Kajian Pustaka Fitur Aplikasi Jurnal Emosi	6
2.3 <i>Progressive Web App</i> (PWA)	8
2.3.1 Struktur Aplikasi <i>Progressive Web App</i>	9
2.3.2 <i>Manifest File</i>	9
2.3.3 <i>Service Worker</i>	9
2.3.4 <i>Lighthouse Testing</i>	10
2.4 <i>Local Storage</i>	10
2.5 <i>Vue JS</i>	10
2.6 <i>Single Page Application</i>	11
2.7 <i>Firebase</i>	11
2.8 <i>Rapid Application Development</i> (RAD)	12
2.9 <i>UML (Unified Modeling Language)</i>	12
2.9.1 <i>Activity Diagram</i>	13
2.9.2 <i>Use Case Diagram</i>	13
2.9.3 <i>Class Diagram</i>	14
2.10 <i>Black Box Testing</i>	14
2.11 <i>System Usability Scale</i> (SUS)	15
2.12 Tinjauan Pustaka Penelitian Sejenis	17
BAB III METODOLOGI PENGEMBANGAN	19
3.1 Metode Pengembangan Aplikasi.....	19
3.2 Fase Perancangan Kebutuhan (<i>Requirements Planning</i>)	20
3.3 Fase Perancangan Desain Sistem (<i>Design System</i>).....	21
3.3.1 Alur Proses Bisnis Aplikasi	21
3.3.2 Perancangan UML	24
3.3.3 Perancangan Antarmuka	35

3.3.4	Perancangan Basis Data	39
3.4	Fase Implementasi (<i>Implementation</i>).....	40
3.4.1	Memulai Proyek.....	40
3.4.2	Implementasi <i>Progressive Web App</i> (PWA).....	41
3.4.3	Konfigurasi <i>Manifest File</i>	42
3.4.4	Implementasi <i>Vue Router</i>	43
3.4.5	Implementasi <i>Firebase</i>	44
3.4.6	Implementasi Aplikasi	45
3.4.7	Implementasi <i>Hosting</i> Aplikasi.....	45
3.5	Pengujian Aplikasi	46
3.5.1	<i>Black Box Testing</i>	46
3.5.2	<i>Lighthouse Testing</i>	46
3.5.3	<i>System Usability Scale</i> (SUS)	46
3.5.4	Prosedur Pengujian <i>Black Box Testing</i> dan <i>Lighthouse Testing</i>	46
BAB IV HASIL DAN PEMBAHASAN		48
4.1	Hasil Implementasi Aplikasi	48
4.1.1	Tampilan Halaman Catat Emosi	48
4.1.2	Tampilan Halaman Instruksi Penggunaan Aplikasi.....	49
4.1.3	Tampilan Halaman Tambah Kegiatan	49
4.1.4	Tampilan Halaman Daftar Catatan Emosi	50
4.1.5	Tampilan Halaman Kelola Catatan Emosi.....	51
4.1.6	Tampilan Halaman Visualisasi Perubahan Emosi	51
4.1.7	Tampilan Halaman <i>Backup</i>	52
4.1.8	Tampilan Halaman <i>Login</i>	53
4.2	Hasil Pengujian Aplikasi.....	54
4.2.1	Pengujian <i>Black box</i>	54
4.2.2	Pengujian <i>Lighthouse Testing</i>	57
4.2.3	<i>Pengujian System Usability Scale</i> (SUS).....	60
BAB V PENUTUP		62
5.1	Kesimpulan	62
5.2	Saran.....	62
DAFTAR PUSTAKA		63

DAFTAR TABEL

Tabel 2.1 Delapan label tipe emosi dalam empat kategori yang diukur oleh PAM	6
Tabel 2.2 Tipologi tipe dan kategori emosi dalam aplikasi Jurnal Emosi	6
Tabel 2.3 Hasil kajian pustaka kebutuhan aplikasi Jurnal Emosi	7
Tabel 2.4 Notasi <i>Activity Digram</i>	13
Tabel 2.5 Notasi <i>Use Case Diagram</i>	14
Tabel 2.6 Pertanyaan Pengujian <i>System Usability Scale (SUS)</i>	15
Tabel 2.7 Pilihan Jawaban Pengujian <i>System Usability Scale (SUS)</i>	15
Tabel 2.8 Tinjauan Pustaka.....	18
Tabel 3.1 Daftar kebutuhan fungsional aplikasi Jurnal Emosi	20
Tabel 3.2 Identifikasi <i>Use Case</i>	24
Tabel 3.3 Skenario <i>Use Case</i> Lihat Instruksi.....	26
Tabel 3.4 Skenario <i>Use Case</i> Catat Emosi	26
Tabel 3.5 Skenario <i>Use Case</i> Tambah Kegiatan	27
Tabel 3.6 Skenario <i>Use Case</i> Lihat Daftar Catatan Emosi	27
Tabel 3.7 Skenario <i>Use Case</i> Mengelola Catatan Emosi	28
Tabel 3.8 Skenario <i>Use Case</i> Lihat Visualisasi Perubahan Emosi	28
Tabel 3.9 Skenario <i>Use Case Login</i>	29
Tabel 3.10 Skenario <i>Use Case Backup</i> Data Catatan Emosi	30
Tabel 3.11 Skenario <i>Use Case Logout</i>	30
Tabel 4.1 Hasil Pengujian <i>Black Box</i> Fungsionalitas Aplikasi Jurnal Emosi.....	54
Tabel 4.2 Hasil Pengujian <i>Black Box</i> Antarmuka Aplikasi Jurnal Emosi	55
Tabel 4.3 Penjelasan Hasil Pengujian <i>Lighthouse</i>	59
Tabel 4.4 Tabel Rekap Pengujian Kuesioner SUS	60
Tabel 4.5 Hasil Perhitungan Pengujian SUS	61

DAFTAR GAMBAR

Gambar 2.1 Delapan tipe dalam empat kategori emosi	5
Gambar 2.2 Tahapan Pendekatan RAD	12
Gambar 2.3 Skala Penilaian <i>System Usability Scale</i>	16
Gambar 3.1 Tahapan RAD Pengembangan Aplikasi Jurnal Emosi	19
Gambar 3.2 Alur Aplikasi Jurnal Emosi.....	23
Gambar 3.3 Sub Alur Fitur Aplikasi Jurnal Emosi.....	23
Gambar 3.4 Diagram <i>Use Case</i> Aplikasi Jurnal Emosi.....	25
Gambar 3.5 <i>Activity Diagram</i> Catat Emosi	31
Gambar 3.6 <i>Activity Diagram</i> Tambah Kegiatan	32
Gambar 3.7 <i>Activity Diagram</i> Mengelola Catatan Emosi	32
Gambar 3.8 <i>Activity Diagram Login</i>	33
Gambar 3.9 <i>Activity Diagram Backup</i> Catatan Emosi.....	34
Gambar 3.10 <i>Class Diagram</i> Aplikasi Jurnal Emosi.....	35
Gambar 3.11 Rancangan Halaman Catat Emosi.....	36
Gambar 3.12 Rancangan Tambah Kegiatan	36
Gambar 3.13 Rancangan Halaman Daftar Catatan Emosi.....	37
Gambar 3.14 Rancangan Halaman Kelola Catatan Emosi	38
Gambar 3.15 Rancangan Halaman Visualisasi Perubahan Emosi.....	38
Gambar 3.16 Rancangan Halaman Backup	39
Gambar 3.17 Struktur Basis Data Local Storage	40
Gambar 3.18 Struktur Basis Data <i>Firebase Firestore</i>	40
Gambar 3.19 Perintah instalasi <i>VueJS</i> ke dalam komputer.	41
Gambar 3.20 Perintah memulai proyek <i>VueJS</i>	41
Gambar 3.21 Perintah memasang PWA pada proyek <i>VueJS</i>	41
Gambar 3.22 Isi file <i>registerServiceWorker.js</i>	42
Gambar 3.23 Konfigurasi <i>Manifest File</i> PWA pada proyek aplikasi Jurnal Emosi	42
Gambar 3.24 Perintah memasang <i>plugin vue-router</i> pada proyek <i>VueJS</i>	43
Gambar 3.25 Konfigurasi <i>vue-router</i> pada aplikasi Jurnal Emosi.....	44
Gambar 3.26 Isi file <i>main.js</i> dalam proyek setelah menambahkan <i>Vue-Router</i>	44
Gambar 3.27 Konfigurasi SDK <i>Firebase</i>	44
Gambar 3.28 Perintah instalasi <i>plugin firebase</i> pada proyek aplikasi.	45
Gambar 3.29 Perintah <i>build</i> aplikasi <i>VueJS</i>	45

Gambar 3.30 Perintah <i>deploy</i> aplikasi <i>VueJS</i>	45
Gambar 4.1 Tampilan Halaman Catat Emosi	48
Gambar 4.2 Tampilan Halaman Instruksi Penggunaan Aplikasi.....	49
Gambar 4.3 Tampilan Halaman Tambah Kegiatan	50
Gambar 4.4 Tampilan Halaman Daftar Catatan Emosi	50
Gambar 4.5 Tampilan Halaman Kelola Catatan Emosi.....	51
Gambar 4.6 Tampilan Halaman Visualisasi Perubahan Emosi	52
Gambar 4.7 Tampilan Halaman <i>Backup</i> Data	53
Gambar 4.8 Tampilan Halaman Login	53
Gambar 4.9 Hasil Pengujian <i>Lighthouse</i>	58



BAB I

PENDAHULUAN

1.1 Latar Belakang

Peran teknologi di masa yang modern ini memiliki pengaruh besar pada kehidupan sehari-hari. Perkembangan kemajuan teknologi dapat membantu manusia dalam belajar, bekerja, dan bersosialisasi. Salah satu aspek yang dapat terbantu dengan adanya teknologi informasi adalah kesehatan mental (Chan et al., 2015).

Dalam kehidupan setiap individu tidak akan terlepas dari tekanan dan tuntutan. Tuntutan dapat berupa hal-hal yang telah terjadi, ataupun hal-hal baru yang mungkin akan terjadi, namun apabila tekanan dan tuntutan tersebut tidak teratasi dengan baik, maka akan membuat individu tertekan sehingga mengalami gangguan kesehatan mental berupa stres (Muslim, 2020). Menurut Kamus Besar Bahasa Indonesia stres dapat diartikan sebagai gangguan atau kekacauan mental dan emosional yang disebabkan oleh faktor luar; ketegangan. (KBBI <https://kbbi.kemdikbud.go.id/entri/stres>). Stres dapat mempengaruhi kondisi fisik, mental, dan emosi seseorang. Oleh karena itu penting bagi setiap orang untuk dapat mengatasi stres (Muslim, 2020).

Dalam mengatasi stres dapat dilakukan upaya menangani masalah penyebabnya dengan berfokus dalam menyelesaikan masalah tersebut atau yang disebut juga dengan koping (Baqutayan, 2015). Fungsi lain koping adalah untuk mengatur emosi. Setiap individu memiliki cara mengatasi stres yang berbeda-beda dimana cara tersebut perlahan dapat berubah menjadi kebiasaan jika dilakukan secara rutin terus menerus (Muslim, 2020). Ada dua jenis cara yang digunakan untuk mengatasi stres yaitu dengan cara yang baik dan buruk. Strategi yang baik seperti membuat manajemen waktu, meditasi, dan refleksi diri. Dan strategi yang buruk seperti menyakiti diri sendiri hingga mengonsumsi alkohol atau narkoba (Ballesteros & Janis, 2009). Salah satu metode yang dapat memberikan dampak positif dalam mempengaruhi kondisi emosi, pikiran, dan tindakan adalah dengan menulis buku harian (Rahmandani, 2017). Melihat hal tersebut maka diperlukan sebuah aplikasi yang dapat menjadi alat koping penanganan stres dengan mengatur kondisi emosi. Aplikasi Jurnal Emosi diciptakan untuk membantu mencatat jurnal harian dan melacak perubahan kondisi emosi pengguna sehari-hari.

Dalam penerapan aplikasi Jurnal Emosi sebagai alat pembantu koping, aplikasi perlu digunakan secara rutin agar bisa menjadi suatu kebiasaan baru pengguna supaya dapat memberikan hasil positif, berdasarkan kebutuhan tersebut aplikasi Jurnal Emosi diciptakan

untuk memberikan kemudahan pengguna dalam mencatat dan mengelola emosinya dengan cara mengembangkan aplikasi yang dapat diakses dimana saja dan kapan saja. Di era digital ini aplikasi Jurnal Emosi perlu dikembangkan agar dapat diakses di berbagai perangkat seperti *smartphone* dan *desktop* komputer untuk memberikan kemudahan *user experience* guna mendorong insentif pengguna dalam mencatat emosinya sehari-hari. Kebutuhan ini menimbulkan tantangan baru dalam mengembangkan aplikasi dikarenakan adanya berbagai macam perangkat dan *platform* yang beredar. Salah satu solusi atas permasalahan tersebut adalah dengan mengembangkan aplikasi Jurnal Emosi dengan pendekatan *cross-platform* berbasis *web* yang memanfaatkan teknologi *Progressive Web App* (PWA).

Pengembangan aplikasi *cross-platform* memungkinkan aplikasi untuk dapat diakses di berbagai perangkat. Teknologi *Progressive Web App* (PWA) memungkinkan pengembang untuk membuat aplikasi berbasis web yang dapat di-instal dalam *smartphone* layaknya aplikasi *native*. Pengembangan dengan PWA juga memberikan aplikasi kemampuan untuk dijalankan secara *offline*, hal ini dapat meningkatkan pengalaman pengguna dimana pengguna tetap dapat menggunakan aplikasi tanpa terkoneksi ke internet.

Berdasarkan permasalahan dari uraian sebelumnya, maka penulis tertarik untuk melakukan penelitian dengan mengembangkan aplikasi Jurnal Emosi dengan pendekatan *cross-platform* menggunakan teknologi *Progressive Web App*. Adapun judul penelitian ini adalah “**Pengembangan Aplikasi Jurnal Emosi Berbasis *Progressive Web App***”.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang telah dijelaskan pada latar belakang, maka rumusan yang diangkat pada laporan ini antara lain

- a. Bagaimana membangun aplikasi Jurnal Emosi yang dapat mencatat dan melacak perubahan emosi menggunakan *Progressive Web App*?
- b. Apakah dengan mengembangkan aplikasi Jurnal Emosi menggunakan teknologi *Progressive Web App*, aplikasi dapat dijalankan di berbagai perangkat dengan kemampuan *offline* dan memiliki performa yang baik?

1.3 Batasan Masalah

Agar penelitian lebih berfokus pada penyelesaian masalah dan tidak meluas dari pembahasan, maka penelitian ini membataskan ruang lingkup penelitian sebagai berikut:

- a. Aplikasi dikembangkan berbasis *web* menggunakan *framework front-end* vuejs.

- b. Penelitian tidak menguji aplikasi dalam perspektif bidang medis atau psikologi.
- c. Penelitian ini hanya berfokus pada pengembangan fitur pencatatan dan pelacakan perubahan emosi pengguna.
- d. Aplikasi Jurnal Emosi hanya menampilkan perubahan emosi pengguna dalam rentan waktu dua minggu terakhir.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah menghasilkan aplikasi Jurnal Emosi yang dapat mencatat dan melacak perubahan emosi berbasis *Progressive Web App*.

1.5 Manfaat Penelitian

Manfaat yang didapatkan dari pengembangan aplikasi ini adalah dapat menjadi solusi dalam membantu pengguna aplikasi dalam mencatat dan melacak perubahan emosi.

1.6 Metodologi Pengembangan

Metode yang digunakan dalam pengembangan aplikasi Jurnal Emosi adalah metode *Rapid Application Development (RAD)* yang memiliki beberapa tahapan sebagai berikut:

- a. Tahap Perancangan Kebutuhan (*Requirements Planning*)
Pada tahap ini dilakukan identifikasi kebutuhan, tujuan dan masalah yang mungkin muncul selama pengembangan aplikasi.
- b. Tahap Desain Sistem (*Design System*)
Pada tahap ini dilakukan perancangan proses sistem dan basis data berdasarkan hasil identifikasi kebutuhan yang dilakukan pada tahap sebelumnya.
- c. Tahap Implementasi (*Implementation*)
Pada tahap ini dilakukan pengembangan aplikasi berdasarkan hasil rancangan pada tahap sebelumnya. Hasil aplikasi akan diuji untuk mengetahui kesesuaian fitur, kinerja, dan performa aplikasi berdasarkan kebutuhan yang diharapkan.

1.7 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penyusunan tugas akhir ini adalah sebagai berikut:

a. **BAB I PENDAHULUAN**

Bab ini menjelaskan latar belakang pengembangan aplikasi, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian, dan sistematika penulisan terkait penerapan teknologi *progressive web app* pada pengembangan aplikasi Jurnal Emosi.

b. **BAB II LANDASAN TEORI**

Bab ini menjelaskan teori-teori yang berkaitan dengan topik penelitian dan menjadi landasan dalam pengembangan aplikasi Jurnal Emosi.

c. **BAB III METODOLOGI PENGEMBANGAN**

Bab ini menjelaskan tentang metode yang digunakan dalam penelitian beserta deskripsi dan tahapan-tahapan yang dilakukan yang meliputi fase perencanaan kebutuhan, fase desain sistem dan fase implementasi.

d. **BAB IV HASIL DAN PEMBAHASAN**

Bab ini berisi pembahasan mengenai hasil implementasi dan pengujian dari pengembangan aplikasi Jurnal Emosi.

e. **BAB V PENUTUP**

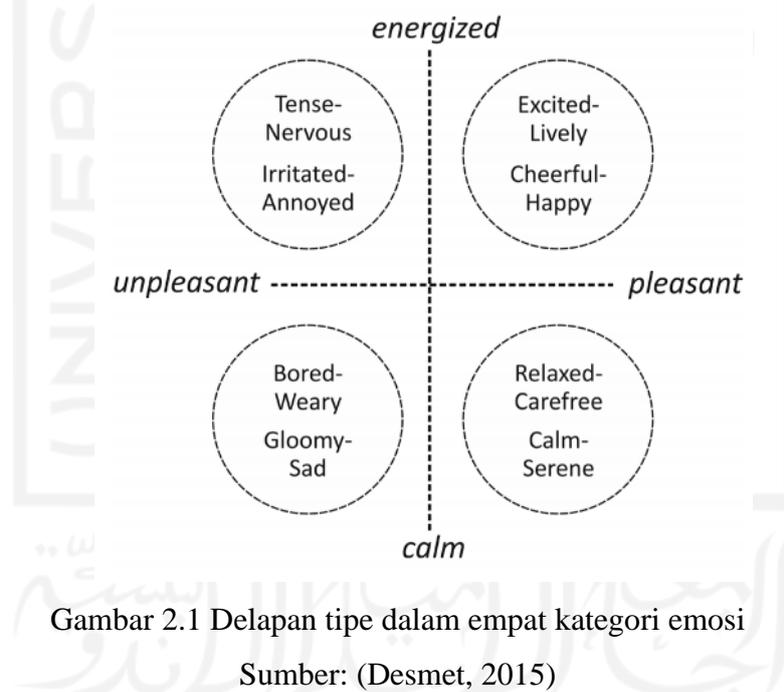
Bab ini berisi kesimpulan jawaban dari rumusan masalah pengembangan yang telah dilakukan, serta saran untuk pengembangan dan perbaikan di masa yang akan datang.

BAB II LANDASAN TEORI

2.1 Jurnal Emosi

Jurnal harian emosi adalah sebuah alat yang dapat membantu pengguna mencatat dan memantau perubahan kondisi emosi/suasana hati sehari-hari. Pada penelitian (Pridmore et al., 2022) menggunakan buku harian emosi sebagai alat untuk memberikan gambaran perubahan emosi atau *mood* seorang subjek dari waktu ke waktu, penerapan ini berbasis dari pertanyaan standar kesehatan yang sering digunakan yaitu “Bagaimana perasaan anda hari ini?”.

Dalam mencatat perubahan emosi, aplikasi Jurnal Emosi menggunakan empat kategori dan delapan tipe emosi sebagai indikator *input* emosi pengguna berdasarkan model klasifikasi emosi *Pick-A-Mood* (PAM) (Desmet et al., 2012), yang ditunjukkan pada Gambar 2.1 berikut.



Dapat dilihat pada Gambar 2.1 kategori *energized-pleasant* mempresentasikan kondisi suasana hati dimana seseorang merasa antusias, aktif dan gembira, yang mana memiliki kebalikan *calm-unpleasant* yaitu kondisi dimana seseorang merasa sedih dan lesu. Kategori *energized-unpleasant* mempresentasikan kondisi suasana hati dimana seseorang merasa tegang, cemas dan marah atau jengkel, yang mana memiliki kebalikan *calm-pleasant* yaitu kondisi dimana seseorang merasa tenang dan damai (Desmet et al., 2012). *Pick-A-Mood* (PAM)

mengukur dan memilih delapan label tipe emosi berdasarkan empat kategori yang disebutkan sebelumnya. Pengelompokan klasifikasi PAM dapat dilihat pada Tabel 2.1 berikut.

Tabel 2.1 Delapan label tipe emosi dalam empat kategori yang diukur oleh PAM

	<i>Pleasant</i>	<i>Unpleasant</i>
<i>Energized</i>	1. <i>Excited</i> 2. <i>Cheerful</i>	3. <i>Tense</i> 4. <i>Irritated</i>
<i>Calm</i>	7. <i>Calm</i> 8. <i>Relaxed</i>	5. <i>Sad</i> 6. <i>Bored</i>

Dalam aplikasi Jurnal Emosi penulis menggunakan tipologi yang dapat dilihat pada Tabel 2.2 untuk mewakili kategori dan tipe emosi yang merupakan hasil dari terjemahan 1:1 kata-kata tersebut ke bahasa Indonesia. Penggunaan kata “Berenergi-Senang” untuk mewakili kategori *energized-pleasant*, kata “Berenergi-Sedih” untuk mewakili kategori *energized-unpleasant*, kata “Tenang-Sedih” untuk mewakili kategori *calm-unpleasant*, dan kata “Tenang-Senang” untuk mewakili kategori *calm-pleasant*.

Tabel 2.2 Tipologi tipe dan kategori emosi dalam aplikasi Jurnal Emosi

No	Kategori	Tipe
1	Berenergi-Senang	1. Bersemangat 2. Gembira
2	Berenergi-Sedih	3. Cemas 4. Kesal
3	Tenang-Sedih	5. Sedih 6. Jenuh
4	Tenang-Senang	7. Tenang 8. Santai

2.2 Kajian Pustaka Fitur Aplikasi Jurnal Emosi

Dalam menentukan fitur yang diperlukan aplikasi Jurnal Emosi penulis melakukan kajian terhadap penelitian yang sudah menganalisis fitur aplikasi *mood tracking* yang tersedia di pasaran. Hasil dari kajian tersebut disajikan dalam bentuk tabel yang ditunjukkan pada Tabel 2.3 yang akan dianalisis dalam perancangan kebutuhan aplikasi Jurnal Emosi.

Tabel 2.3 Hasil kajian pustaka kebutuhan aplikasi Jurnal Emosi

No	Penelitian	Pembahasan
1	<p><i>Mobile Apps for Mood Tracking: An Analysis of Features and User Reviews</i> (Caldeira et al., 2017)</p>	<p>Dalam penelitian ini dilakukan analisis terhadap 32 aplikasi <i>mood tracking</i> yang tersedia di pasar penyedia aplikasi di tahun 2017. Hasil penelitian ini mengkategorikan fitur aplikasi <i>mood tracking</i> ke dalam 4 tahapan, yaitu <i>preparation</i>, <i>collection</i>, <i>reflection</i> dan <i>action</i>. Pada tahap <i>preparation</i> aplikasi menyediakan informasi dan instruksi dalam mencatat mood. Pada tahap <i>collection</i> aplikasi menyediakan antarmuka berupa teks, gambar, <i>emoji</i>, dan lain-lain sebagai input pencatatan <i>mood</i>. Pada tahap <i>reflection</i> aplikasi menyediakan visualisasi perubahan <i>mood</i> berupa tabel, <i>list</i>, dan <i>graph</i>. Pada tahap <i>action</i>, aplikasi menyediakan instruksi meningkatkan mood berdasarkan data yang telah dikoleksi, selain itu juga menyediakan <i>backup</i> data catatan <i>mood</i> pengguna. penelitian ini juga mengkaji <i>user reviews</i> pada aplikasi untuk mencari motivasi pengguna dalam menggunakan aplikasi <i>mood tracking</i>. Dari kajian tersebut ditemukan beberapa alasan penggunaan aplikasi, yaitu umumnya pengguna menggunakan aplikasi untuk mengetahui dan mengelola perubahan mood mereka.</p> <p>Hasil dari penelitian ini menyatakan bahwa walaupun banyak aplikasi <i>mood tracking</i> telah berhasil menyediakan fitur <i>collection</i> dan <i>reflection</i>, sebagian besar belum menyediakan fitur dalam tahapan <i>preparation</i> dan <i>action</i>.</p>
2	<p><i>An Interaction Design Analysis of Mood Trackers</i> (Branco et al., 2020)</p>	<p>Dalam penelitian ini dilakukan analisis terhadap 26 aplikasi <i>mood tracking</i> yang tersedia di pasar penyedia aplikasi di tahun 2020.</p> <p>Hasil dari penelitian ini menyatakan bahwa ditemukan kebanyakan aplikasi memiliki beberapa isu dari sudut pandang medis, dimana kebanyakan</p>

		<p>aplikasi hanya menggunakan indikator berupa <i>emoji</i> dalam menggambarkan emosi. Hal ini menimbulkan pertanyaan apakah pengguna dapat mendeskripsikan emosinya secara mendalam pada sesi terapi. Peneliti berpendapat memberikan ruang menulis narasi atau <i>journaling</i> dalam aplikasi dapat meningkatkan refleksi kondisi emosi pengguna.</p>
--	--	---

2.3 Progressive Web App (PWA)

Progressive Web App adalah sebuah cara pembuatan aplikasi berbasis web menggunakan beberapa teknologi modern yang memungkinkan pengembangan aplikasi dengan pengalaman pengguna menyerupai aplikasi *native* pada perangkat *mobile* (Sheppard, 2017), Aplikasi PWA sepenuhnya adalah *website* yang menerapkan beberapa strategi pengembangan dan API yang membuat aplikasi dapat dijalankan secara *offline* dengan menambahkannya ke halaman *home-screen* perangkat *mobile* (Majchrzak et al., 2018). Namun berbeda dengan aplikasi *web* biasa, dimana pengguna diharuskan membuka *browser*, mengetikkan URL dan lalu menunggu semua aset selesai diunduh pada setiap kunjungan yang tidak memungkinkan penggunaan aplikasi secara *offline*, PWA hanya membutuhkan hal tersebut pada kunjungan pertama kali. Setelah instalasi PWA ke home screen semua aset seperti HTML, CSS, Javascript, gambar dan *font* yang dibutuhkan untuk menjalankan aplikasi telah tersimpan dalam perangkat pengguna dan siap untuk digunakan secara *offline* (Majchrzak et al., 2018). Berikut karakteristik dan keunggulan *Progressive Web App* (MDN contributors, 2022):

- a. *Discoverable*, dimana konten aplikasi PWA dapat ditemukan oleh mesin pencari dengan lebih baik, dikarenakan adanya *metadata* dan *Manifest File* yang berisi informasi yang mendeskripsikan aplikasi.
- b. *Installable*, dimana aplikasi dapat di-instal di perangkat pengguna dan *icon* tertampil di halaman *homescreen* atau *app launcher* yang memudahkan pengguna untuk membuka kembali aplikasi tanpa harus mengaksesnya kembali melalui *browser*.
- c. *Linkable*, dimana aplikasi dapat mudah dibagikan dengan menyebarkan sebuah link URL.
- d. *Network independent*, dimana aplikasi dapat tetap diakses meskipun dalam kondisi jaringan buruk atau tidak tersambung ke internet.

- e. *Progressively enhanced*, dimana fitur dasar aplikasi tetap dapat digunakan di *browser* ataupun perangkat generasi lama, dan memiliki seluruh fungsionalitas di generasi terbaru.
- f. *Re-engageable*, dimana aplikasi dapat mengirim notifikasi ketika ada pembaruan baru.
- g. *Responsively designed*, dimana aplikasi dapat memberikan pengalaman pengguna yang baik di berbagai perangkat yang digunakan seperti *smartphone*, *laptop*, *tablet* dan lain-lain.
- h. *Secure*, dimana aplikasi aman digunakan karena dalam penggunaan *service worker* aplikasi harus berjalan di protokol HTTPS.

2.3.1 Struktur Aplikasi *Progressive Web App*

Dalam *me-render* sebuah *website* terdapat 2 pendekatan utama yaitu pada sisi *server* atau sisi *client* (MDN contributors, 2022).

- a. *Server-side rendering* (SSR) dimana komponen *website* dibentuk dalam *server* yang lalu diserahkan ke perangkat pengguna, metode ini dapat berjalan baik di berbagai *browser* namun membutuhkan pengunduhan halaman HTML baru dalam navigasinya. Memuat setiap halaman memerlukan perjalanan bolak-balik ke *server*.
- b. *Client-side rendering* (CSR) memungkinkan *website* untuk memuat komponen secara instan dalam navigasi halaman, namun metode ini membutuhkan unduhan yang lebih besar dan *rendering* yang lebih berat ketika *website* diakses pertama kali.

Dalam pengembangannya aplikasi Jurnal Emosi menggunakan pendekatan *Client-side rendering* dimana semua aset aplikasi termasuk data pencatatan jurnal emosi disimpan di dalam perangkat pengguna menggunakan *Local Storage* dan hanya akan berhubungan dengan *server* ketika menyimpan dan mengambil *backup* data, hal ini memungkinkan aplikasi dijalankan dengan cepat dan dapat digunakan sepenuhnya secara *offline*.

2.3.2 *Manifest File*

Web Manifest atau *Manifest File* adalah sebuah *file* JSON yang berisi *metadata* PWA berupa informasi *icon*, nama, tema, *splash screen*, dan deskripsi aplikasi yang akan di-instal. *Manifest File* memberitahu *browser* bagaimana aplikasi berperilaku ketika dijalankan (LePage et al., 2022). *Manifest File* memungkinkan untuk menginstal aplikasi ke perangkat pengguna dengan menyediakan *icon* pada *home screen* yang ketika ditekan dapat menjalankan aplikasi secara langsung tanpa harus membuka *browser* dan mengetik URL, layaknya aplikasi *native*. Hal tersebut dapat memudahkan pengguna dalam mengakses dan berinteraksi dengan aplikasi.

2.3.3 *Service Worker*

Service Worker adalah sebuah *script Javascript* yang berjalan di *background browser* pengguna. *Service worker* memungkinkan pengembang untuk menyimpan aset dan data yang

diperlukan aplikasi ke dalam *cache*, mengelola *push notification*, dan memberikan aplikasi kemampuan menggunakan *hardware* sebuah *device*. *Service worker* adalah komponen utama dalam sebuah PWA, tanpa *service worker browser* tidak dapat meng-*install* aplikasi untuk dijalankan secara *offline*.

2.3.4 *Lighthouse Testing*

Lighthouse adalah alat audit *open-source* yang disediakan oleh *browser Google Chrome* untuk menguji dan mengukur performa, aksesibilitas, SEO dan kelayakan instalasi PWA terhadap sebuah aplikasi *web*.

2.4 *Local Storage*

Local Storage adalah tempat penyimpanan pada sisi *browser* dimana sistem dapat menyimpan data perubahan yang dilakukan pengguna sehingga data tidak akan hilang ketika *browser* ditutup dan perangkat dimatikan. Pengembang dapat memanfaatkan *local storage* untuk menyimpan konfigurasi yang diperlukan aplikasi.

2.5 *Vue JS*

Vue adalah sebuah *framework Javascript open source* untuk membangun antarmuka pengguna yang bersifat interaktif pada aplikasi *web*. Tidak seperti beberapa *framework* monolitik yang lain, *Vue* dirancang dengan berfokus pada lapisan tampilan saja sehingga dapat dengan mudah diintegrasikan dengan proyek-proyek yang sudah ada. Fungsi lain *Vue JS* adalah membuat *Single Page Application (SPA)* (Rojas, 2020). *Vue JS* memiliki fitur dan komponen utama yang mendukung pengembangan aplikasi, antara lain:

a. *Component*

Component adalah sebuah blok elemen antarmuka yang dapat dikustomisasi, dan dapat digunakan berulang kali dalam pengembangan aplikasi.

b. *Virtual DOM*

Virtual DOM adalah sebuah *javascript object* (replika) yang mempresentasikan DOM HTML sebenarnya (*real DOM*) dimana perubahan pada *virtual DOM* tidak langsung mempengaruhi DOM asli yang berada dalam *browser*.

c. *Template*

Vue JS menyediakan format antarmuka berbasis HTML yang mengikat DOM dengan setiap elemen *vue*.

d. *Routing*

Vue menyediakan alat berupa *vue-router* yang memudahkan navigasi antar halaman dimana aplikasi dapat mengubah tampilan halaman tanpa harus memuat HTML baru. *Vue router* memungkinkan pembuatan *Single Page Application* (SPA).

e. *Data Binding*

Vue menyediakan fitur untuk mengikat data antara *model* dengan *view* yang melakukan sinkronisasi secara otomatis setiap terjadi perubahan data.

2.6 *Single Page Application*

Single Page Application (SPA) merupakan sebuah aplikasi yang menggunakan satu halaman HTML dimana semua *component* dapat diperbarui atau diganti tanpa perlu memuat ulang halaman. Penerapan SPA bertujuan untuk memudahkan pengguna melakukan navigasi dan meningkatkan *user experience*.

2.7 *Firebase*

Firebase adalah suatu layanan *Backend as a Service* (BaaS) yang disediakan oleh *Google* untuk memudahkan *developer* dalam mengembangkan suatu aplikasi. *Firebase* mengurus segala hal mengenai sistem *backend* aplikasi seperti penyimpanan, *database*, autentikasi, *hosting* dan lainnya. Beberapa fitur menarik yang disediakan oleh *firebase* adalah sebagai berikut (<https://firebase.google.com/>):

a. *Firebase Firestore*

Firebase Cloud Firestore adalah layanan penyimpanan database *NoSQL* yang bersifat fleksibel dan terstruktur. Dalam menyimpan data, *Firebase* menggunakan struktur berupa *collection* dan *document* untuk mengelompokkan data ke *database*. *Firebase Firestore* memungkinkan *developer* dalam membuat aplikasi dimana pengguna dapat mengakses dan menyimpan perubahan datanya dari *device* yang berbeda.

b. *Firebase Authentication*

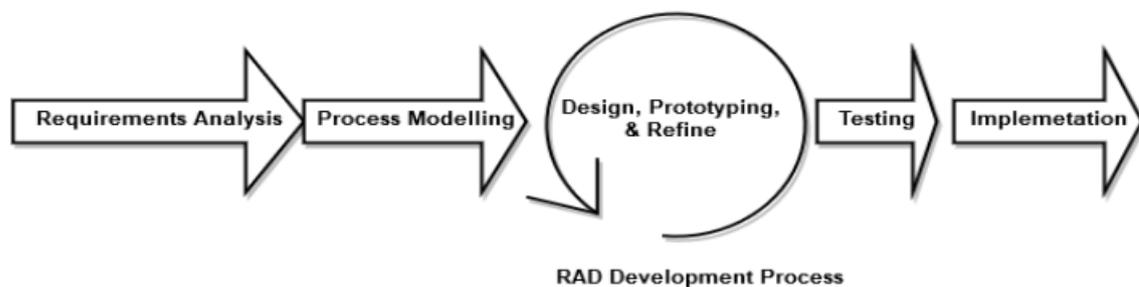
Firebase Authentication menyediakan layanan autentikasi *sign-in* pengguna menggunakan akun *email* dan sandi atau melalui akun sosial media seperti *Google*, *Facebook*, *twitter*, *Github* dan lain-lain. *Google* mengelola semua data login pengguna dimana data pribadi seperti *password* disimpan dalam bentuk *hash* yang sudah terenkripsi, sehingga aplikasi dapat menjamin keamanan data pengguna.

c. *Firebase Hosting*

Google menyediakan layanan *hosting* konten aplikasi web secara cepat dan aman. *Firestore hosting* dapat menampung *file frontend* aplikasi dan merilis aplikasi tersebut ke internet melalui *link domain* yang telah disediakan *firebase*.

2.8 Rapid Application Development (RAD)

Pengembangan aplikasi Jurnal Emosi menggunakan salah satu metode pengembangan software yang merupakan bagian dari *Software Development Life Cycle (SDLC)* yaitu *Rapid Application Development (RAD)*. Metode ini diciptakan sebagai alternatif *model* metode *waterfall* dimana secara umum pendekatan RAD kurang menekankan perencanaan dan lebih menekankan proses implementasi (Chrismanto et al., 2019). Metode RAD umumnya memiliki siklus pengembangan yang singkat antara 30 – 90 dimana cocok untuk mengembangkan aplikasi dalam skala kecil dan cepat. Dalam penerapannya metode RAD memiliki beberapa fase yang terlihat pada Gambar 2.2 diantaranya meliputi fase perencanaan kebutuhan, fase desain sistem, dan fase implementasi.



Gambar 2.2 Tahapan Pendekatan RAD

Sumber: (Chrismanto et al., 2019)

2.9 UML (Unified Modeling Language)

UML adalah suatu bahasa standar visualisasi, perancangan, dan pendokumentasian sistem yang digunakan untuk menggambarkan desain dari pengembangan sistem perangkat lunak (Dicoding Intern, 2021).

UML terbagi menjadi berbagai jenis diagram yang masing-masing memiliki tujuan yang berbeda. Dua jenis kategori paling luas yang umumnya digunakan dalam pengembangan perangkat lunak adalah diagram *Behavioral UML* dan *Structural UML*.

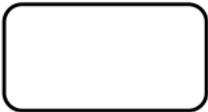
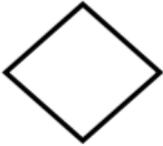
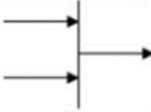
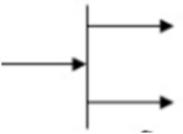
Diagram *behavioral* menunjukkan bagaimana sistem harus berjalan dan menggambarkan bagaimana objek berinteraksi satu sama lain untuk menciptakan sistem yang berfungsi

sebagaimana semestinya (Anisa, 2022). Diagram *structural* menggambarkan struktur objek pada sistem yang dimodelkan, diagram ini umumnya digunakan dalam mendokumentasikan arsitektur sistem dalam pengembangan perangkat lunak (Tallyfy, 2018).

2.9.1 Activity Diagram

Activity diagram merupakan diagram UML yang menggambarkan alur dari serangkaian aktivitas dan bagaimana aktivitas tersebut dapat berhubungan dari awal hingga akhir untuk memodelkan alur kerja pengoperasian sistem. Notasi yang digunakan dalam *activity diagram* dijelaskan pada Tabel 2.4.

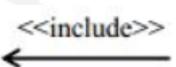
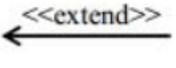
Tabel 2.4 Notasi Activity Diagram

No	Gambar	Nama	Fungsi
1		Aktivitas	Aktivitas yang dilakukan dalam sistem. Biasanya diawali dengan kata kerja.
2		Percabangan / <i>Decision</i>	Percabangan dimana ada pilihan aktivitas lebih dari satu. Percabangan diperlukan ketika sistem memiliki beberapa kemungkinan atau jalan alternatif.
3		Titik Mulai	Menandakan titik awal pada sebuah <i>activity diagram</i> .
4		Status Akhir	Menandakan berakhirnya proses aktivitas yang dilakukan.
5		Penggabungan / <i>Join</i>	Menggabungkan lebih dari satu aktivitas.
6		Percabangan / <i>Fork</i>	Memecah aktivitas secara paralel.

2.9.2 Use Case Diagram

Use Case Diagram menggambarkan hubungan interaksi antara aktor dengan sistem. Diagram ini dapat mendeskripsikan keseluruhan bentuk sistem dalam pengembangan perangkat lunak. Adapun fungsi dari use case diagram adalah sebagai jembatan antara pengembang dengan konsumen untuk menggambarkan proses bisnis sebuah sistem (Dicoding, 2021). Notasi yang digunakan dalam *use case diagram* dijelaskan pada Tabel 2.5.

Tabel 2.5 Notasi *Use Case Diagram*

No	Gambar	Nama	Fungsi
1		Aktor	Aktor mewakili peran orang, sistem, atau alat ketika berinteraksi dengan <i>use case</i> .
2		<i>Use Case</i>	Mendesripsikan suatu aktivitas yang dilakukan aktor dalam sistem.
3		<i>Association</i>	Menunjukkan hubungan antara aktor dengan <i>use case</i>
4		Generalisasi	Menunjukkan spesialisasi aktor untuk dapat berinteraksi dengan sistem.
5		<i>Include</i>	Menunjukkan suatu <i>use case</i> merupakan bagian fungsionalitas dari satu <i>use case</i> utama.
6		<i>Extend</i>	Menunjukkan suatu <i>use case</i> merupakan fungsi tambahan dari <i>use case</i> lainnya.

2.9.3 *Class Diagram*

Class Diagram menggambarkan dan menjelaskan hubungan antar kelas dari suatu sistem. *Class Diagram* memperlihatkan aturan-aturan dan tanggung jawab semua kelas yang menentukan perilaku sistem (Sulistyorini, 2009).

2.10 *Black Box Testing*

Black box testing adalah metode pengujian spesifikasi fungsionalitas dari perangkat lunak yang tidak mengacu pada struktur kode pemrogramannya (Febiharsa et al., 2018). Pengujian dilakukan dengan memberikan masukan sebagai pengujian spesifikasi fungsional aplikasi untuk mengetahui apakah perangkat lunak sudah berjalan seperti yang diharapkan. Penggunaan *black box testing* diperuntukan untuk menemukan beberapa kategori *bug* antara lain:

- a. Fungsionalitas yang hilang atau salah
- b. Kesalahan pada desain antarmuka tampilan perangkat lunak.
- c. Kesalahan pada struktur basis data.
- d. Kesalahan pada kinerja perangkat lunak.

2.11 System Usability Scale (SUS)

System Usability Scale (SUS) merupakan suatu metode pengujian untuk mengukur kinerja suatu sistem. *System Usability Scale* akan diberikan kepada pengguna dalam bentuk kuesioner yang berisikan 10 pertanyaan dan 5 pilihan jawaban yang memiliki skala 1-5 level yang memiliki arti dari sangat tidak setuju, tidak setuju, ragu-ragu, setuju, dan sangat setuju (Susilo, 2019). Pertanyaan memiliki dua tipe yaitu pertanyaan bernada positif untuk pertanyaan bernomor ganjil (1,3,5,7,9) dan pertanyaan bernada negatif untuk pertanyaan bernomor genap (2,4,6,8,10), Bentuk pertanyaan pengujian *System Usability Scale* ditunjukkan pada tabel 2.6.

Tabel 2.6 Pertanyaan Pengujian *System Usability Scale* (SUS)

No	Pertanyaan
1	Saya berpikir akan menggunakan sistem ini lagi
2	Saya merasa sistem ini rumit untuk digunakan
3	Saya merasa sistem ini mudah digunakan
4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini
5	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya
6	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi dengan sistem ini)
7	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat
8	Saya merasa sistem ini membingungkan
9	Saya merasa tidak ada hambatan dalam menggunakan sistem ini
10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini

Adapun pilihan jawaban pertanyaan pengujian dan skala nilai pertanyaan SUS ditunjukkan pada Tabel 2.7.

Tabel 2.7 Pilihan Jawaban Pengujian *System Usability Scale* (SUS)

Jawaban	Skor Nilai
Sangat Tidak Setuju (STS)	1
Tidak Setuju (TS)	2
Ragu-ragu (RG)	3
Setuju (S)	4
Sangat Setuju (SS)	5

Untuk menentukan kesimpulan penilaian kinerja sistem, metode ini menggunakan perhitungan skor berdasarkan rumus yang ditetapkan oleh *System Usability Scale*. Untuk pertanyaan bernomor ganjil, skor dihitung dengan cara skor nilai yang diberikan responden (X) dikurangi dengan 1, Ditunjukkan pada persamaan (2.1).

$$\text{Skor Pertanyaan Ganjil} = X - 1 \quad (2.1)$$

Sedangkan untuk pertanyaan bernomor genap, skor dihitung dengan cara nilai 5 dikurangi dengan skor nilai yang diberikan oleh responden (X). Ditunjukkan pada persamaan (2.2).

$$\text{Skor Pertanyaan Genap} = 5 - X \quad (2.2)$$

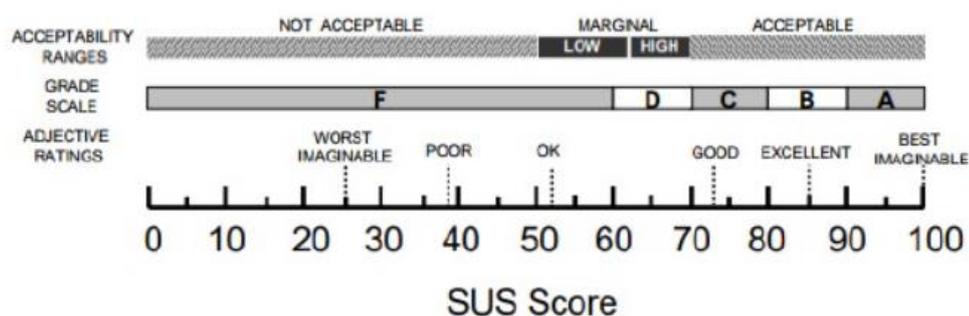
Dari hasil tersebut dilakukan perhitungan sub total yang didapatkan dari hasil penjumlahan skor setiap soal pertanyaan ganjil dan genap. Ditunjukkan pada persamaan (2.3).

$$\text{Skor sub total} = \text{Total Skor Ganjil} + \text{Total Skor Genap} \quad (2.3)$$

Setelah mendapatkan skor sub total, dilakukan perhitungan skor akhir responden dengan cara mengalikan skor sub total dengan nilai 2,5. Ditunjukkan pada persamaan (2.4).

$$\text{Skor akhir responden} = \text{Sub Total} \times 2,5 \quad (2.4)$$

Setelah menghitung semua skor akhir tiap responden, dilakukan perhitungan skor akhir *System Usability Scale* dengan cara mencari rata-rata dari penjumlahan seluruh skor akhir tiap responden dan membaginya dengan jumlah responden. Dari skor akhir SUS maka didapatkan kesimpulan kinerja sistem dengan menyesuaikan skor akhir dengan kategori penilaian SUS yang ditunjukkan pada Gambar 2.3.



Gambar 2.3 Skala Penilaian *System Usability Scale*

Sumber: (Susilo, 2019)

2.12 Tinjauan Pustaka Penelitian Sejenis

Penulis menggunakan beberapa penelitian sejenis sebagai referensi dalam mempelajari dan memperluas wawasan dalam melakukan penelitian ini. Penelitian tentang implementasi dan pengembangan aplikasi menggunakan teknologi *Progressive Web App* sudah pernah dilaksanakan sebelumnya. Kasus yang diangkat pada penelitian yang sudah dilakukan ada berbagai macam, seperti penelitian yang dilakukan (Workneh, 2019) dengan judul “*Evaluation of Progressive Web Application to develop an Offline-First Task Management App*”, di dalam penelitian tersebut peneliti menerapkan PWA dalam pembuatan aplikasi manajemen tugas/kegiatan, aplikasi dikembangkan dengan pendekatan *offline-first* dimana sistem menyimpan data pengguna ke dalam penyimpanan lokal *IndexedDB* sehingga aplikasi dapat digunakan sepenuhnya tanpa koneksi internet. Pada penelitian tersebut, aplikasi diuji menggunakan *lighthouse testing* untuk mencari tahu hasil penerapan PWA pada aplikasi.

Pada penelitian yang dilakukan (Aripin & Somantri, 2021) dengan judul “Implementasi *Progressive Web Apps* (PWA) pada *Repository* E-Portofolio Mahasiswa”, peneliti menerapkan PWA dalam pembuatan aplikasi E-Portofolio mahasiswa, adapun hasil dari penerapan PWA adalah aplikasi dapat diakses secara *offline* dan berjalan *responsive* pada berbagai perangkat. Aplikasi diuji menggunakan *black box testing*, *lighthouse testing*, dan *usability testing*.

Pada penelitian yang dilakukan (Dewi et al., 2020) dengan judul “Pemanfaatan *Progressive Web Apps* Pada *Web* Akuntansi”, peneliti menerapkan PWA dalam pembuatan aplikasi akuntansi, adapun hasil dari penelitian menyatakan PWA dapat membantu kecepatan *load* sebuah *website*. Aplikasi diuji menggunakan *lighthouse testing* dan *GTMetrix*.

Pada penelitian yang dilakukan (Darfiansyah, 2022) dengan judul “Perancangan Sistem Informasi Penyewaan Perangkat Elektronik Berbasis *Website* Dengan Menggunakan *Teknologi Progressive Web App*”, peneliti menerapkan PWA dalam pembuatan aplikasi penyewaan perangkat elektronik, adapun hasil dari penelitian menyatakan penerapan PWA dapat memberikan pengalaman pengguna seperti aplikasi *native*. Aplikasi diuji menggunakan *black box testing*.

Pada penelitian yang dilakukan (Kusnawan, 2021) dengan judul “Implementasi *Progressive Web App* pada Aplikasi Manajemen Data *Dropship*”, peneliti menerapkan PWA dalam pembuatan aplikasi manajemen data dropship, adapun hasil dari penelitian ini menyatakan PWA dapat digunakan sebagai alternatif pembuatan aplikasi *mobile*. Aplikasi diuji menggunakan *lighthouse testing* dan *black box testing*. Hasil dari tinjauan pustaka penelitian terkait dapat dilihat pada Tabel 2.8.

Tabel 2.8 Tinjauan Pustaka

No	Penelitian	Teknologi	Metode Pengujian	Hasil Penerapan PWA
1	<i>Evaluation of Progressive Web Application to develop an Offline-First Task Management App</i> (Workneh, 2019)	<i>Progressive Web App, Framework ReactJS</i>	<i>Lighthouse testing</i>	Aplikasi dapat digunakan sepenuhnya secara <i>offline</i> karena menerapkan PWA dengan basis data <i>IndexedDB</i> .
2	Implementasi <i>Progressive Web Apps</i> (PWA) pada <i>Repository E-Portofolio Mahasiswa</i> (Aripin & Somantri, 2021)	<i>Progressive Web App, Framework Laravel</i>	<i>Lighthouse testing, Black box testing, Usability testing</i>	Aplikasi dapat diakses secara <i>offline</i> dan berjalan <i>responsive</i> pada berbagai perangkat.
3	Pemanfaatan <i>Progressive Web Apps</i> Pada <i>Web Akuntansi</i> (Dewi et al., 2020)	<i>Progressive Web App, Framework Quasar</i>	<i>Performance Testing, Lighthouse Testing</i>	PWA dapat membantu kecepatan <i>load</i> sebuah <i>website</i> .
4	Perancangan Sistem Informasi Penyewaan Perangkat Elektronik Berbasis <i>Website</i> Dengan Menggunakan <i>Teknologi Progressive Web App</i> (Darfiansyah, 2022)	<i>Progressive Web App, Library ReactJS</i>	<i>Black box testing</i>	PWA dapat memberikan pengalaman pengguna seperti aplikasi <i>native</i> .
5	Implementasi <i>Progressive Web App</i> pada Aplikasi Manajemen Data <i>Dropship</i> (Kusnawan, 2021)	<i>Progressive Web App, Framework VueJS</i>	<i>Lighthouse testing, Black box testing</i>	PWA dapat digunakan sebagai alternatif pembuatan aplikasi <i>mobile</i> .

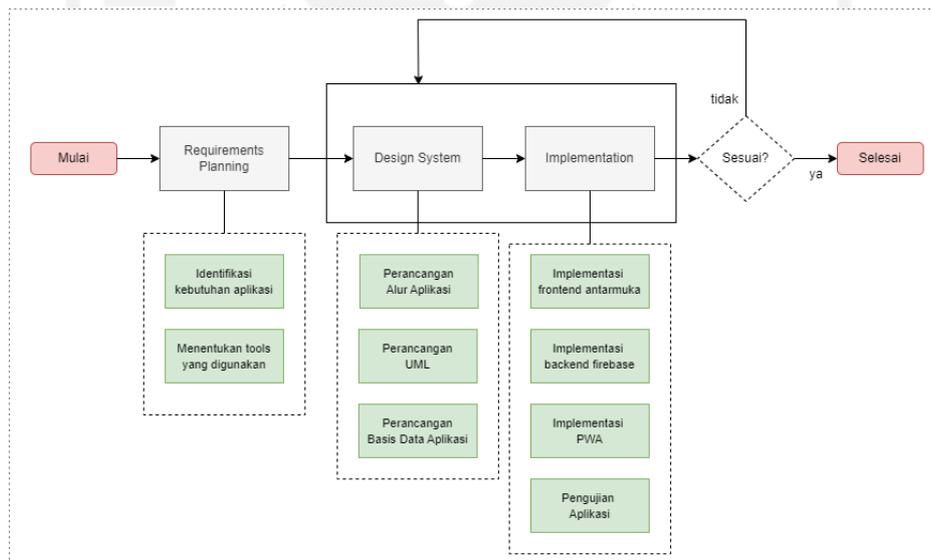
BAB III

METODOLOGI PENGEMBANGAN

3.1 Metode Pengembangan Aplikasi

Metode pengembangan aplikasi adalah sebuah aturan dan pedoman yang digunakan dalam menganalisis kebutuhan aplikasi, merancang desain sistem, pengembangan dan pengujian aplikasi. Dalam pengembangan aplikasi Jurnal Emosi penulis menggunakan metode *Rapid Application Development (RAD)* karena memiliki alur pengembangan yang mudah dan singkat, sehingga tepat digunakan untuk keperluan menguji hasil penerapan *Progressive Web App* dalam pengembangan aplikasi Jurnal Emosi.

Tahapan metodologi RAD pada pengembangan aplikasi Jurnal Emosi ditunjukkan pada Gambar 3.1.



Gambar 3.1 Tahapan RAD Pengembangan Aplikasi Jurnal Emosi

Pada Gambar 3.1 memberikan gambaran tahapan-tahapan metodologi RAD pada pengembangan aplikasi Jurnal Emosi. Tahapan dimulai dengan perancangan kebutuhan dimana dilakukan identifikasi kebutuhan fungsional dan non-fungsional aplikasi. Setelah itu dilakukan tahapan desain sistem untuk merancang struktur sistem dan alur proses bisnis aplikasi. Selanjutnya dilakukan tahapan implementasi untuk mengembangkan aplikasi Jurnal Emosi sesuai dengan rancangan yang sudah dibuat pada tahap sebelumnya dan diuji untuk menentukan apakah aplikasi yang sudah dibuat sesuai dengan kebutuhan yang diharapkan.

3.2 Fase Perancangan Kebutuhan (*Requirements Planning*)

Pada proses perancangan kebutuhan dilakukan kajian pustaka terhadap penelitian yang sudah menganalisis kebutuhan fitur berbagai aplikasi *mood tracking* yang tersedia di pasaran. Berdasarkan hasil kajian penelitian yang ditunjukkan pada Tabel 2.3 di bab sebelumnya, ditemukan inti permasalahan antara lain:

- a. Aplikasi ditunjukkan untuk calon pengguna yang ingin mengetahui dan mengelola perubahan emosi mereka.
- b. Masih kurangnya penerapan fitur *preparation* dan *action* dalam kebanyakan aplikasi jurnal emosi yang beredar dipasaran.
- c. Penggunaan emoji sebagai indikator kurang dapat menggambarkan suasana emosi pengguna aplikasi jurnal emosi sehingga memerlukan *input* tambahan lain berupa ruang menulis narasi atau *journaling*.

Berdasarkan hasil kajian pustaka dan inti permasalahan tersebut penulis menyimpulkan kebutuhan fungsional pengembangan aplikasi Jurnal Emosi dan usulan penyelesaiannya yang ditunjukkan pada Tabel 3.1.

Tabel 3.1 Daftar kebutuhan fungsional aplikasi Jurnal Emosi

No	Kebutuhan	Penyelesaian
1	Fitur <i>preparation</i> yang menyediakan informasi dan instruksi pencatatan emosi	Menambahkan <i>button</i> di halaman catat Jurnal Emosi yang ketika diklik akan menampilkan <i>popup</i> berisi instruksi penggunaan aplikasi Jurnal Emosi.
2	Fitur <i>collection</i> untuk mencatat dan menyimpan data pencatatan emosi pengguna	Membuat satu halaman fitur pencatatan emosi dimana pengguna dapat memasukkan tanggal dan kondisi emosi yang dirasakan saat itu.
3	Fitur <i>reflection</i> untuk menyediakan visualisasi perubahan emosi pengguna	Membuat satu halaman fitur visualisasi kondisi emosi pengguna dalam rentang waktu berdasarkan data dari fitur <i>collection</i> .
4	Fitur <i>action</i> untuk membuat <i>backup</i> data ke penyimpanan <i>cloud</i>	Membuat satu halaman <i>backup</i> data, dimana pengguna dapat menyimpan data catatan emosi ke penyimpanan <i>cloud</i> .
5	Input tambahan untuk mendeskripsikan emosi pengguna secara lebih dalam	Membuat sebuah <i>input</i> penulisan narasi dan <i>input</i> pilihan kegiatan pada halaman pencatatan emosi dimana pengguna dapat memberikan <i>input</i> tambahan berupa narasi jurnal dan kegiatan untuk memberikan

		konteks yang mempengaruhi kondisi emosinya. <i>Input</i> ini bersifat opsional.
--	--	---

Adapun kebutuhan non-fungsional berupa *tools* dan teknologi yang digunakan dalam pengembangan aplikasi Jurnal Emosi diantaranya:

1. Browser *google chrome*
2. Perangkat *android*
3. *Yarn* sebagai *package manager*
4. *Visual studio code* sebagai *text editor*
5. *VueJS* sebagai *framework frontend* aplikasi
6. *Bootstrap* sebagai *framework* pembuatan komponen antarmuka aplikasi.
7. *Firebase* sebagai *backend* aplikasi
8. *Progressive web app* sebagai alternatif model pengembangan *mobile* pada aplikasi web.

3.3 Fase Perancangan Desain Sistem (*Design System*)

Pada tahapan ini dilakukan perancangan desain aplikasi Jurnal Emosi berupa struktur sistem, alur proses bisnis aplikasi, purwarupa tampilan aplikasi dan skema *database* berdasarkan kebutuhan pada fase sebelumnya.

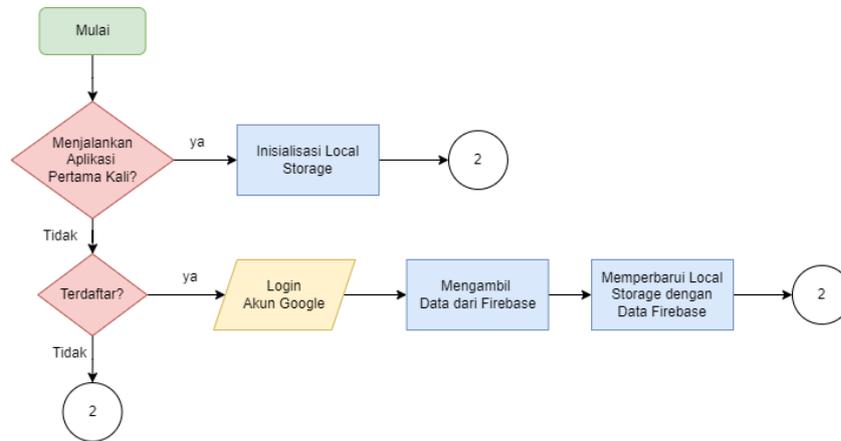
3.3.1 Alur Proses Bisnis Aplikasi

Sesuai hasil identifikasi kebutuhan sistem, aplikasi Jurnal Emosi diharapkan dapat menyediakan fitur pencatatan emosi berupa *preparation*, *collection*, *reflection* dan *action* untuk membantu pengguna dalam melacak dan mengelola perubahan suasana emosi. Untuk membuat fitur-fitur tersebut aplikasi menerapkan fungsi dasar *create*, *read*, *update*, *delete* (CRUD) yang mana memiliki alur sistem sebagai berikut:

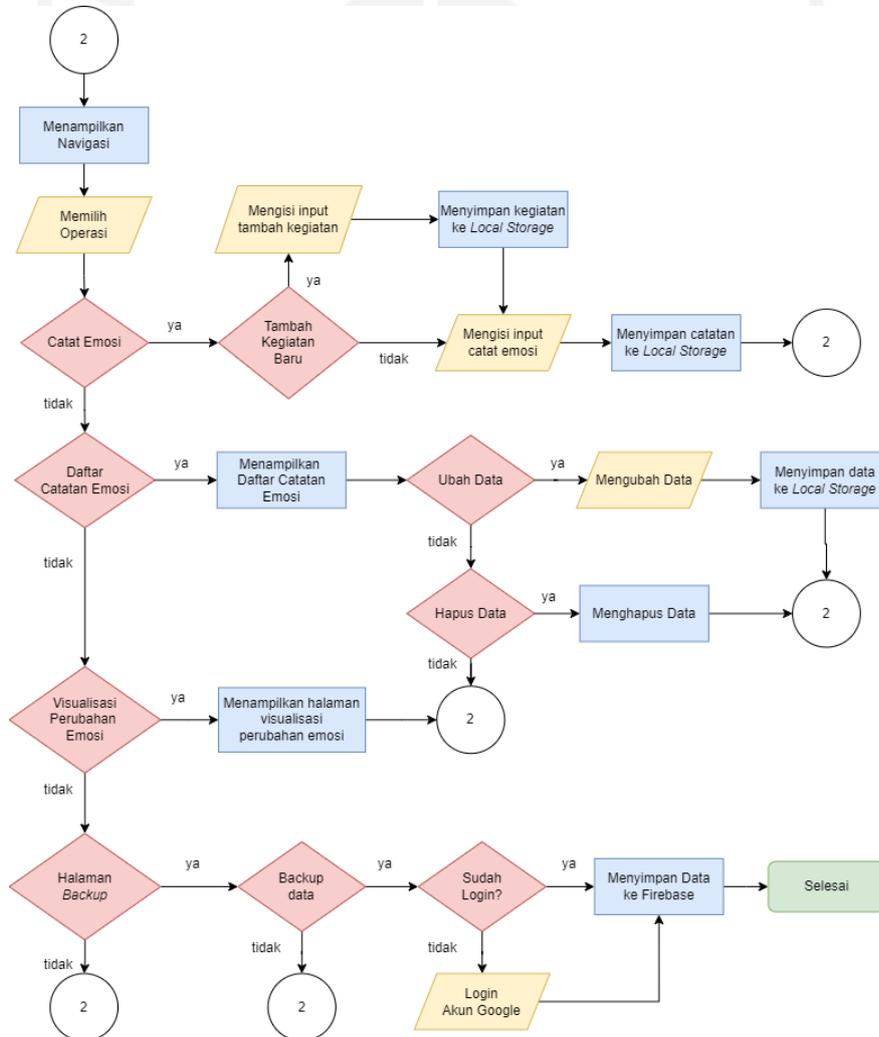
1. Pengguna sebagai aktor tunggal aplikasi.
2. Sistem melakukan inisiasi basis data *local storage* dalam *browser* ketika aplikasi dijalankan pertama kali.
3. Pengguna dapat *login* ke dalam aplikasi untuk mengambil data catatan emosi yang pernah dilakukan. Tahapan ini bersifat opsional dimana aplikasi dapat tetap dapat digunakan tanpa harus melakukan *login*.
4. Sistem menampilkan halaman pencatatan emosi (*collection*) sebagai halaman pertama aplikasi.

5. Sistem menyediakan *button* untuk menampilkan halaman instruksi penggunaan aplikasi Jurnal Emosi (*preparation*) di pojok kanan atas halaman pencatatan emosi.
6. Pengguna melakukan pencatatan emosi dengan memasukkan *input* berupa indikator emosi yang dirasakan, tanggal pencatatan, narasi jurnal dan daftar kegiatan yang dilakukan.
7. Sistem menerima input pengguna dan menyimpan informasi tersebut ke penyimpanan *local storage* dalam browser (*collection*).
8. Sistem menyediakan halaman yang menampilkan daftar seluruh data pencatatan emosi pengguna diurutkan berdasarkan tanggal pencatatan (*reflection*).
9. Pengguna dapat melihat dan mengubah data pencatatan yang telah dilakukan.
10. Pengguna dapat menghapus data pencatatan emosi.
11. Sistem menyediakan halaman yang menampilkan visualisasi perubahan emosi pengguna dari waktu ke waktu dalam bentuk *chart* (*reflection*).
12. Sistem menyediakan halaman *backup* data dimana pengguna dapat menyimpan data pencatatan emosi ke dalam penyimpanan *cloud firebase* (*action*).
13. Pengguna melakukan *login* menggunakan akun google.
14. Sistem menampilkan halaman *login* yang terintegrasi oleh *firebase*.
15. *Firestore* menyimpan data *login* ke dalam *server* dimana seluruh data autentikasi pengguna hanya dimiliki oleh *Google*.
16. Pengguna dapat mem-*backup* pembaruan data pencatatan emosi kapan saja dengan menekan tombol "*backup*" di halaman *backup* setelah melakukan *login*.
17. Pengguna dapat mengambil data pencatatan yang tersimpan di dalam penyimpanan *cloud* dengan melakukan *login* pada perangkat yang berbeda.

Tahapan alur sistem divisualisasikan pada Gambar 3.2 dan Gambar 3.3.



Gambar 3.2 Alur Aplikasi Jurnal Emosi



Gambar 3.3 Sub Alur Fitur Aplikasi Jurnal Emosi

Pada Gambar 3.2 menjelaskan alur proses bisnis aplikasi Jurnal Emosi ketika pertama kali dijalankan dimana sistem akan menginisiasi basis data pada *local storage* browser, pengguna juga dapat melakukan login untuk mengambil data pencatatan jika sudah pernah menggunakan aplikasi sebelumnya di perangkat yang berbeda.

Pada Gambar 3.3 menjelaskan alur proses bisnis fitur-fitur utama aplikasi Jurnal Emosi, dan menggambarkan bagaimana pengguna dapat menjalankan operasi mencatat, melihat, dan menyimpan catatan emosi.

3.3.2 Perancangan UML

Berdasarkan rancangan alur sistem aplikasi Jurnal Emosi yang dibuat sebelumnya selanjutnya dilakukan perancangan *Unified Modelling Language* (UML) untuk menggambarkan seluruh fungsi sistem.

a. Identifikasi Aktor

Sesuai dengan rancangan alur sistem, Aplikasi Jurnal Emosi memiliki satu aktor tunggal yaitu pengguna. Pengguna dapat melakukan pencatatan emosi, pengelolaan catatan, melihat perubahan emosi dan mem-*backup* data pencatatan ke penyimpanan *cloud*.

b. Identifikasi Use Case

Untuk menjelaskan berbagai proses yang berlangsung pada aplikasi Jurnal Emosi dilakukan identifikasi tiap fungsi aplikasi ke dalam beberapa *use case* dan deskripsinya yang ditunjukkan pada Tabel 3.2 sebagai berikut.

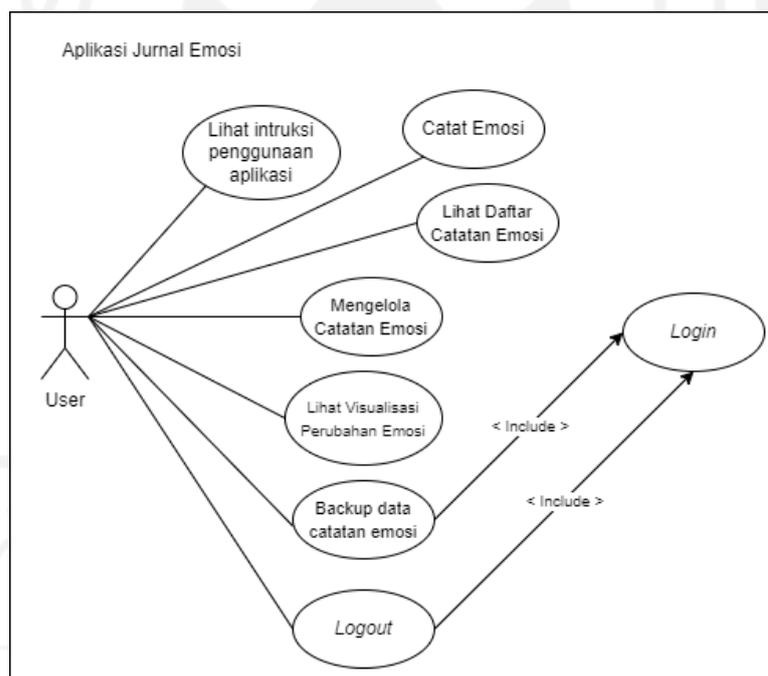
Tabel 3.2 Identifikasi *Use Case*

No	Aktor	Use Case	Deskripsi
1	User	Lihat instruksi penggunaan aplikasi	Menjelaskan aktivitas aktor dalam mengetahui instruksi penggunaan emosi
2	User	Catat emosi	Menjelaskan aktivitas aktor dalam melakukan pencatatan emosi.
3	User	Tambah kegiatan baru	Menjelaskan aktivitas aktor dalam menambah daftar kegiatan baru.
4	User	Lihat daftar catatan emosi	Menjelaskan aktivitas aktor dalam melihat daftar emosi yang sudah dibuat
5	User	Mengelola catatan emosi	Menjelaskan bagaimana aktor dapat mengubah dan menghapus data pencatatan emosi.

6	User	Lihat visualisasi perubahan emosi	Menjelaskan bagaimana aktor dapat melihat visualisasi perubahan emosi berdasarkan data catatan emosi yang telah dibuat
7	User	Backup data catatan emosi	Menjelaskan aktivitas aktor dalam melakukan <i>backup</i> data ke penyimpanan <i>cloud firebase</i> .
8	User	Login	Menjelaskan aktivitas aktor dalam mendapatkan autentikasi untuk dapat melakukan <i>backup</i> data.
9	User	Logout	Menjelaskan aktivitas aktor keluar dari aplikasi dan menghapus data <i>session</i> dari aplikasi.

c. Diagram Use Case

Untuk menggambarkan hubungan interaksi antara pengguna dengan sistem dibuat sebuah diagram *use case* yang ditunjukkan pada Gambar 3.4 yang menjelaskan rancangan aplikasi berdasarkan identifikasi kebutuhan dan *use case* pada tahap sebelumnya.



Gambar 3.4 Diagram *Use Case* Aplikasi Jurnal Emosi

d. Skenario Use Case

Untuk menjelaskan alur aktivitas yang dilakukan pengguna pada setiap *use case* dibuat sebuah skenario penggunaan yang ditunjukkan pada Tabel 3.3 - Tabel 3.11.

1. Use Case Lihat Instruksi Penggunaan Aplikasi

Skenario *use case* lihat instruksi ditunjukkan pada Tabel 3.3 dimana kondisi akhir berupa sistem yang dapat menampilkan *popup* halaman instruksi penggunaan aplikasi.

Tabel 3.3 Skenario Use Case Lihat Instruksi

<i>Use Case Lihat Instruksi</i>	
Aktor	<i>User</i>
Deskripsi	Menjelaskan aktivitas aktor dalam mengetahui instruksi penggunaan emosi
Skenario Utama	
Kondisi awal	Halaman catat emosi ditampilkan
Aksi Aktor	Reaksi Sistem
1. Menekan tombol lihat instruksi “?”	2. Menampilkan <i>popup</i> halaman instruksi penggunaan aplikasi
Kondisi Akhir	Sistem menampilkan <i>popup</i> halaman instruksi penggunaan aplikasi

2. Use Case Catat Emosi

Skenario *use case* catat emosi ditunjukkan pada Tabel 3.4 dimana kondisi akhir berupa sistem yang dapat menyimpan data catatan emosi ke dalam *local storage*.

Tabel 3.4 Skenario Use Case Catat Emosi

<i>Use Case Catat Emosi</i>	
Aktor	<i>User</i>
Deskripsi	Menjelaskan aktivitas aktor dalam melakukan pencatatan emosi
Skenario Utama	
Kondisi awal	Halaman catat emosi ditampilkan
Aksi Aktor	Reaksi Sistem
1. Menekan menu catat emosi pada <i>navbar</i>	2. Menampilkan halaman catat emosi
3. Mengisi <i>form</i> catat emosi	
4. Memilih tanggal pencatatan emosi	
5. Memilih kategori emosi	6. Menampilkan <i>input</i> tipe emosi sesuai kategori yang dipilih
7. Memilih tipe emosi	
8. Mengisi jurnal narasi	
9. Memilih kegiatan	
10. Menekan tombol “Simpan”	11. Menyimpan data catatan emosi ke dalam <i>Local Storage</i>
Kondisi Akhir	Data catatan emosi tersimpan di dalam <i>Local Storage</i>

3. Use Case Tambah Kegiatan

Skenario use case tambah kegiatan ditunjukkan pada Tabel 3.5 dimana kondisi akhir berupa sistem yang dapat menyimpan data kegiatan baru ke dalam *local storage*.

Tabel 3.5 Skenario Use Case Tambah Kegiatan

Use Case Tambah Kegiatan	
Aktor	User
Deskripsi	Menjelaskan aktivitas aktor dalam menambah daftar kegiatan baru.
Skenario Utama	
Kondisi awal	Halaman catat emosi ditampilkan
Aksi Aktor	Reaksi Sistem
1. Menekan tombol “+” dalam <i>box input</i> daftar kegiatan	2. Menampilkan <i>popup input</i> tambah kegiatan dan daftar kegiatan
3. Mengisi <i>input</i> nama kegiatan	
4. Menekan tombol “+” tambah kegiatan	5. Menyimpan kegiatan baru ke <i>Local Storage</i>
6. Memilih dan menekan salah satu kegiatan di daftar kegiatan	7. Menghapus kegiatan yang dipilih
Kondisi Akhir	Data kegiatan baru tersimpan di dalam <i>Local Storage</i>

4. Use Case Lihat Daftar Catatan Emosi

Skenario use case lihat daftar catatan emosi ditunjukkan pada Tabel 3.6 dimana kondisi akhir berupa sistem yang dapat menampilkan halaman daftar catatan emosi yang datanya diambil dari *local storage*.

Tabel 3.6 Skenario Use Case Lihat Daftar Catatan Emosi

Use Case Lihat Daftar Catatan Emosi	
Aktor	User
Deskripsi	Menjelaskan aktivitas aktor dalam melihat daftar emosi yang sudah dibuat
Skenario Utama	
Kondisi awal	Aktor sudah membuat data catatan emosi
Aksi Aktor	Reaksi Sistem
1. Aktor menuju halaman daftar emosi	2. Mengambil data catatan emosi dari <i>local storage</i> .
	3. Menampilkan data catatan diurutkan berdasarkan tanggal pencatatan.
Kondisi Akhir	Sistem menampilkan halaman daftar catatan emosi

5. Use Case Mengelola Catatan Emosi

Skenario *use case* mengelola catatan emosi ditunjukkan pada Tabel 3.7 dimana kondisi akhir berupa sistem yang dapat menyimpan perubahan data pencatatan emosi dan menyimpannya ke dalam *local storage*.

Tabel 3.7 Skenario Use Case Mengelola Catatan Emosi

Use Case Mengelola Catatan Emosi	
Aktor	<i>User</i>
Deskripsi	Menjelaskan bagaimana aktor dapat mengubah dan menghapus data pencatatan emosi.
Skenario Utama	
Kondisi awal	-Aktor sudah membuat data catatan emosi -Halaman daftar catatan ditampilkan
Aksi Aktor	Reaksi Sistem
1. Menekan salah satu catatan emosi	2. Menampilkan <i>popup</i> yang berisi kolom formulir data catatan emosi yang dapat diubah
2. Mengubah data catatan emosi	
3. Menekan tombol “simpan “	4. Menyimpan perubahan catatan ke <i>local storage</i>
5. Menekan tombol “hapus”	6. Menghapus catatan emosi dari <i>local storage</i>
Kondisi Akhir	Perubahan data catatan emosi tersimpan di dalam <i>Local Storage</i>

6. Use Case Lihat Visualisasi Perubahan Emosi

Skenario *use case* lihat visualisasi perubahan emosi ditunjukkan pada Tabel 3.8 dimana kondisi akhir berupa sistem yang dapat menampilkan halaman visualisasi perubahan emosi berdasarkan pencatatan yang sudah dilakukan pengguna.

Tabel 3.8 Skenario Use Case Lihat Visualisasi Perubahan Emosi

Use Case Lihat Visualisasi Perubahan Emosi	
Aktor	<i>User</i>
Deskripsi	Menjelaskan bagaimana aktor dapat melihat visualisasi perubahan emosi berdasarkan data catatan emosi yang telah dibuat
Skenario Utama	
Kondisi awal	Aktor sudah membuat data catatan emosi
Aksi Aktor	Reaksi Sistem
1. Aktor menuju halaman visualisasi perubahan emosi	2. Mengambil data catatan emosi dari <i>local storage</i> .
	3. Menampilkan <i>bar chart</i> jumlah tiap emosi pengguna yang dipilih selama seminggu terakhir

	4. Menampilkan <i>bar chart</i> jumlah tiap emosi pengguna yang dipilih pada minggu lalu
	5. Menampilkan <i>pie chart</i> yang menunjukkan jumlah tiap kategori emosi yang dipilih pengguna
	6. Menampilkan <i>chart pie</i> yang menunjukkan jumlah tiap kegiatan yang dipilih pengguna.
Kondisi Akhir	Pengguna dapat melihat visualisasi perubahan emosinya berdasarkan catatan yang sudah dilakukan

7. Use Case Login

Skenario *use case login* ditunjukkan pada Tabel 3.9 dimana kondisi akhir berupa pengguna berhasil masuk ke dalam aplikasi.

Tabel 3.9 Skenario *Use Case Login*

<i>Use Case Login</i>	
Aktor	<i>User</i>
Deskripsi	Menjelaskan aktivitas aktor dalam mendapatkan autentikasi untuk dapat melakukan backup data.
Skenario Utama	
Kondisi awal	Aktor memiliki akun <i>Google</i>
Aksi Aktor	Reaksi Sistem
1. Aktor menuju halaman <i>backup</i>	
2. Menekan tombol “ <i>Sign in with Google</i> ”	3. Menampilkan <i>popup</i> halaman login melalui akun <i>Google</i>
3. Melakukan <i>login</i> dengan akun <i>Google</i>	4. Menyimpan data autentikasi pengguna ke <i>firebase</i>
	5. Menyimpan data <i>session</i> pengguna ke dalam <i>browser</i>
	6. Mengambil data catatan emosi dari <i>firebase</i> jika pengguna sudah pernah <i>login</i> sebelumnya.
	7. Menyimpan data catatan emosi ke <i>local storage</i>
	8. Memuat ulang halaman <i>backup</i>
Kondisi Akhir	Aktor berhasil <i>login</i>

8. Use Case Backup Catatan Emosi

Skenario *use case backup* catatan emosi ditunjukkan pada Tabel 3.10 dimana kondisi akhir berupa sistem dapat menyimpan data catatan emosi pengguna ke dalam penyimpanan *cloud firebase firestore*.

Tabel 3.10 Skenario *Use Case Backup Data Catatan Emosi*

Use Case Backup Data Catatan Emosi	
Aktor	<i>User</i>
Deskripsi	Menjelaskan aktivitas aktor dalam melakukan <i>backup</i> data ke penyimpanan <i>cloud firebase</i> .
Skenario Utama	
Kondisi awal	Aktor telah melakukan login
Aksi Aktor	Reaksi Sistem
1. Aktor menuju halaman <i>backup</i>	
2. Menekan tombol " <i>Backup</i> "	3. Melakukan enkripsi pada data catatan emosi di <i>local storage</i>
	4. Menyimpan data catatan emosi yang sudah terenkripsi ke penyimpanan <i>cloud firebase</i> .
	5. Menampilkan teks konfirmasi backup telah berhasil di bawah tombol " <i>Backup</i> "
Kondisi Akhir	Data catatan emosi pengguna tersimpan di dalam penyimpanan <i>cloud firebase firestore</i> .

9. *Use Case Logout*

Skenario *use case logout* ditunjukkan pada Tabel 3.11 dimana kondisi akhir berupa pengguna dapat berhasil keluar dari aplikasi.

Tabel 3.11 Skenario *Use Case Logout*

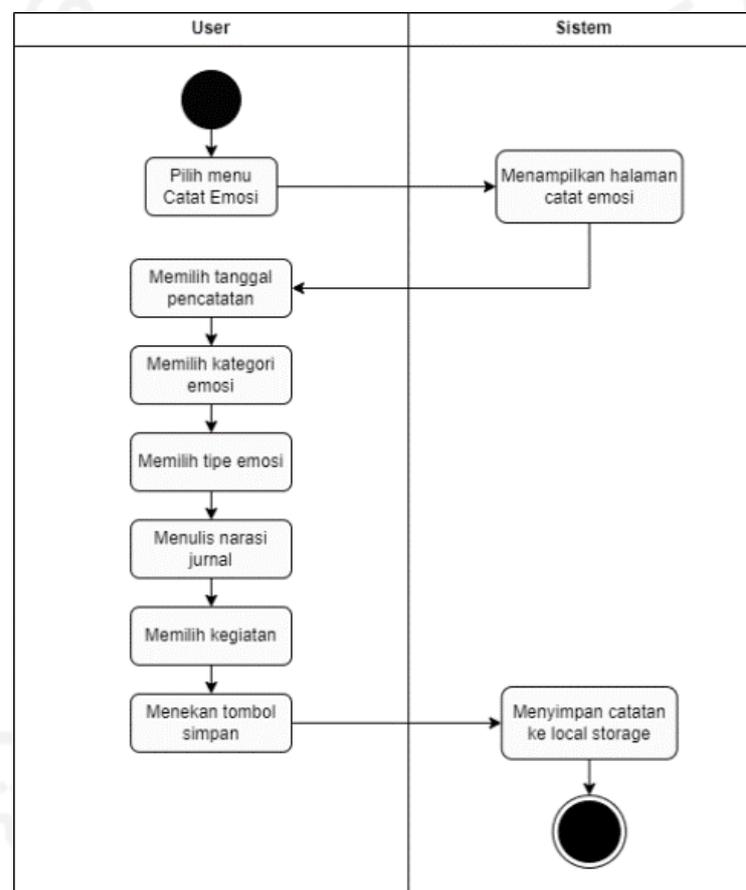
Use Case Logout	
Aktor	<i>User</i>
Deskripsi	Menjelaskan aktivitas aktor keluar dari aplikasi dan menghapus data <i>session</i> dari aplikasi.
Skenario Utama	
Kondisi awal	Aktor telah melakukan login
Aksi Aktor	Reaksi Sistem
1. Aktor menuju halaman <i>backup</i>	
2. Menekan tombol " <i>Logout</i> "	3. Menghapus data <i>session</i> pengguna dari <i>browser</i>
	4. Menghapus data catatan emosi pengguna dari <i>local storage</i>
	5. Mengeluarkan pengguna dari aplikasi
	6. Memuat ulang halaman <i>backup</i>
Kondisi Akhir	Aktor berhasil keluar dari aplikasi

e. Activity Diagram

Untuk menggambarkan lebih lanjut alur aktivitas pengguna pada skenario *use case* catat emosi, tambah kegiatan, mengelola catatan emosi, login, dan backup catatan emosi, dibuat sebuah activity diagram yang ditunjukkan pada Gambar 3.5 - Gambar 3.9.

1. Activity Diagram Use Case Catat Emosi

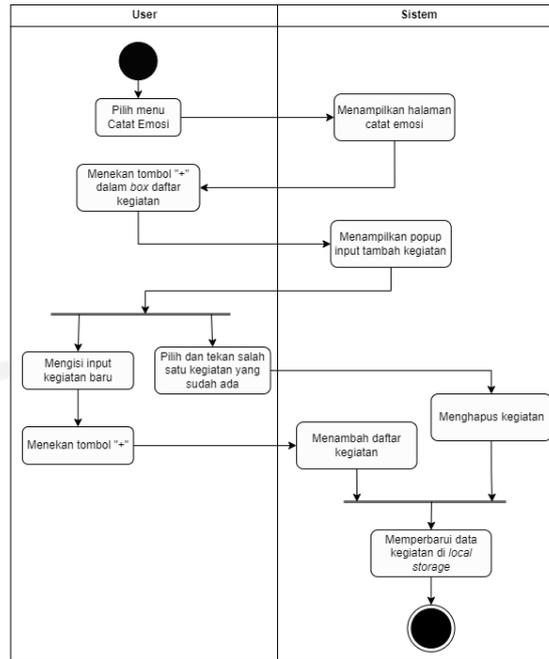
Pada bagian ini pengguna dapat mengisi formulir pencatatan emosi, *input* berupa tanggal, kategori emosi, tipe emosi, narasi, dan kegiatan. Setelah pengguna berhasil mengisi formulir sistem akan menyimpan data tersebut ke dalam *local storage*. *Activity diagram use case* catat emosi ditunjukkan pada Gambar 3.5.



Gambar 3.5 Activity Diagram Catat Emosi

2. Activity Diagram Use Case Tambah Kegiatan

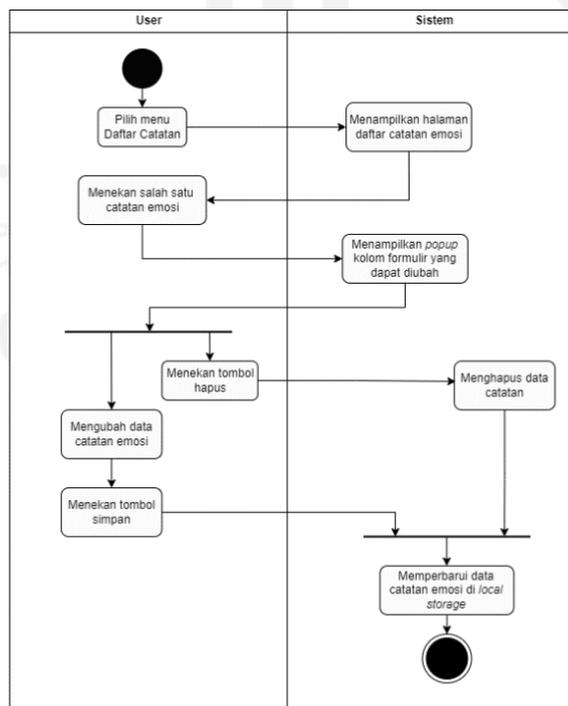
Tambah kegiatan adalah salah satu menu yang dapat dibuka melalui halaman catat emosi. Pengguna dapat menambah kegiatan baru dan menghapus kegiatan yang. Semua perubahan dari pengguna otomatis tersimpan ke penyimpanan *local storage* oleh sistem. *Activity diagram use case* tambah kegiatan ditunjukkan pada Gambar 3.6.



Gambar 3.6 Activity Diagram Tambah Kegiatan

3. Activity Diagram Use Case Mengelola Catatan Emosi

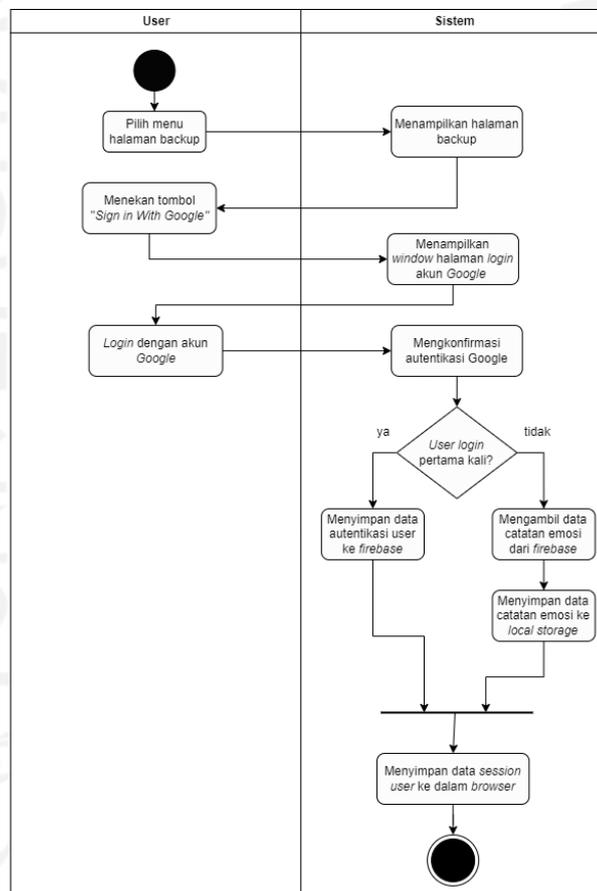
Menu mengelola catatan emosi adalah menu yang dapat dibuka dengan menekan salah satu catatan emosi pada fitur lihat daftar catatan emosi. Pengguna dapat mengubah data pencatatan dan menghapus catatan. Semua perubahan otomatis diperbarui ke dalam penyimpanan *local storage* oleh sistem. Activity diagram use case mengelola catatan emosi ditunjukkan pada Gambar 3.7.



Gambar 3.7 Activity Diagram Mengelola Catatan Emosi

4. Activity Diagram Use Case Login

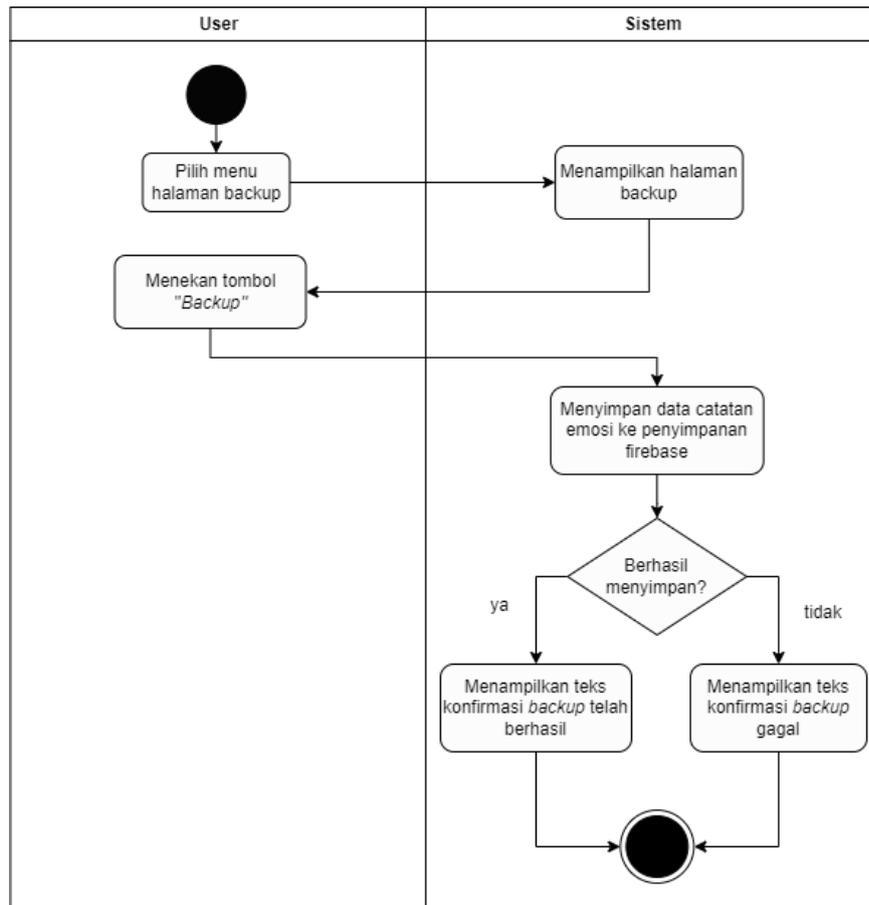
Untuk menggunakan fitur *backup* catatan emosi pengguna diharuskan masuk ke dalam aplikasi dengan melakukan *login* menggunakan akun google. Sistem akan menampilkan halaman *login* yang terintegrasi dengan *firebase*, setelah pengguna berhasil *login* sistem akan mengecek apakah pengguna sudah pernah masuk atau tidak. Jika pengguna baru pertama kali melakukan *login* sistem akan menyimpan data autentikasi pengguna ke penyimpanan *cloud firebase*, dan jika pengguna sudah pernah login sebelumnya sistem akan mengambil data catatan emosi dan menyimpan salinannya ke dalam penyimpanan *local storage* di *browser* pengguna. Activity diagram login ditunjukkan pada Gambar 3.8.



Gambar 3.8 Activity Diagram Login

5. Activity Diagram Use Case Backup Catatan Emosi

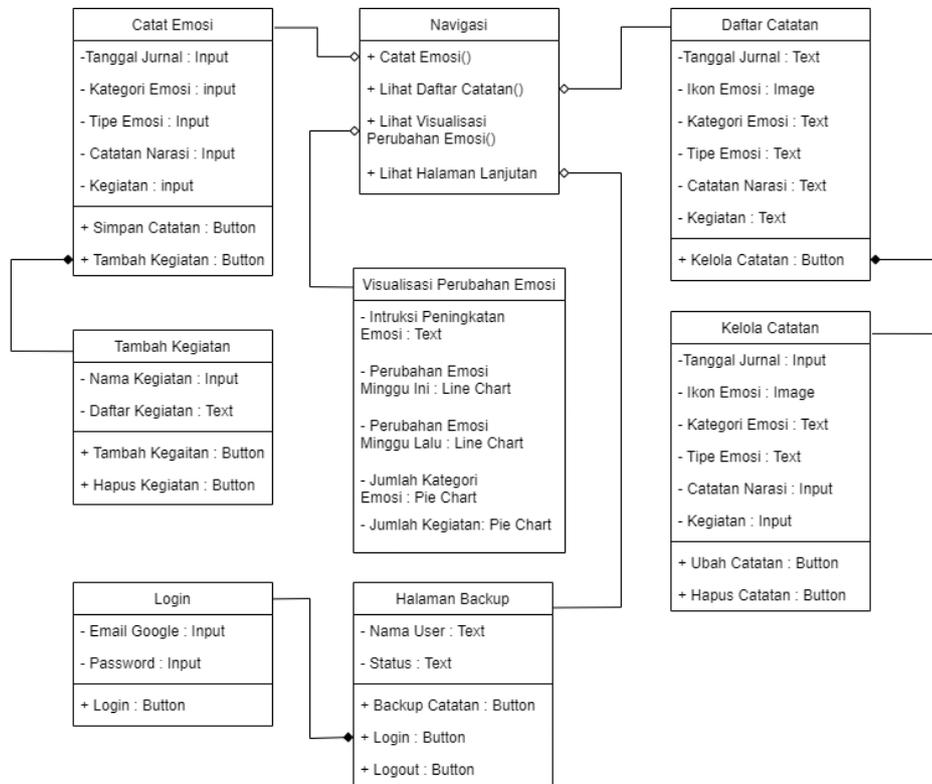
Setelah berhasil *login* pengguna dapat menggunakan fitur *backup* catatan emosi. Setelah menekan tombol “*backup*” sistem akan menyimpan data catatan emosi ke penyimpanan *cloud firebase*. Jika berhasil menyimpan sistem akan menampilkan teks konfirmasi berhasil, dan jika gagal maka sistem akan menampilkan konfirmasi *backup* catatan gagal. *Activity diagram backup* catatan emosi ditunjukkan pada Gambar 3.9.



Gambar 3.9 Activity Diagram Backup Catatan Emosi

f. Class Diagram

Pada bagian ini dibuat sebuah *class diagram* yang menggambarkan hubungan antar objek yang terdapat dalam setiap halaman aplikasi Jurnal Emosi. Diagram struktur kelas aplikasi ditunjukkan pada Gambar 3.10.



Gambar 3.10 *Class Diagram* Aplikasi Jurnal Emosi

3.3.3 Perancangan Antarmuka

Pada bagian ini akan dijelaskan rancangan tampilan antarmuka aplikasi Jurnal Emosi.

a. Rancangan halaman catat emosi

Halaman catat emosi adalah halaman yang digunakan pengguna untuk menambahkan catatan emosi. Halaman ini menampilkan formulir pencatatan emosi yang memiliki input berupa kategori emosi yang direpresentasikan menggunakan *icon emoji*, *input* tipe emosi, *input* tanggal pencatatan, *input* teks catatan narasi, dan *input* pemilihan kegiatan. Rancangan halaman catat emosi ditunjukkan pada Gambar 3.11

Gambar 3.11 Rancangan Halaman Catat Emosi

b. Rancangan halaman tambah kegiatan

Halaman tambah kegiatan adalah halaman *popup* yang muncul ketika pengguna menekan tombol “tambah kegiatan” pada halaman catat emosi. Halaman ini menampilkan *input* tambah kegiatan, dan daftar kegiatan yang sudah dibuat, pengguna dapat menghapus kegiatan dengan menekan salah *item* daftar kegiatan. Rancangan halaman tambah kegiatan ditunjukkan pada Gambar 3.12.

Gambar 3.12 Rancangan Tambah Kegiatan

c. Rancangan halaman daftar catatan emosi

Halaman daftar catatan emosi adalah halaman yang menampilkan daftar catatan emosi yang telah ditambahkan pengguna dalam bentuk *card*. *Card* setiap catatan emosi dipisahkan berdasarkan tanggal pencatatan, di dalam *card* terdapat detail data pencatatan emosi berupa kategori dan tipe emosi, catatan narasi, waktu pencatatan, dan kegiatan yang dipilih. Rancangan halaman daftar catatan emosi ditunjukkan pada Gambar 3.13.



Gambar 3.13 Rancangan Halaman Daftar Catatan Emosi

d. Rancangan halaman kelola catatan emosi

Halaman kelola catatan emosi adalah halaman popup yang muncul ketika pengguna menekan salah satu daftar catatan emosi. Halaman ini menampilkan formulir berisi detail catatan emosi yang dapat diubah oleh pengguna. Halaman ini juga menampilkan tombol *toggle* ubah kegiatan yang ketika ditekan akan memunculkan atau menyembunyikan *dropdown* daftar kegiatan yang dapat dipilih pengguna. Selain itu halaman ini juga menyediakan tombol "X" yang mengeluarkan pengguna dari halaman, tombol "hapus" yang akan menghapus catatan emosi, dan tombol "simpan" yang akan menyimpan perubahan pengguna. Rancangan halaman kelola catatan emosi ditunjukkan pada Gambar 3.14.

Gambar 3.14 Rancangan Halaman Kelola Catatan Emosi

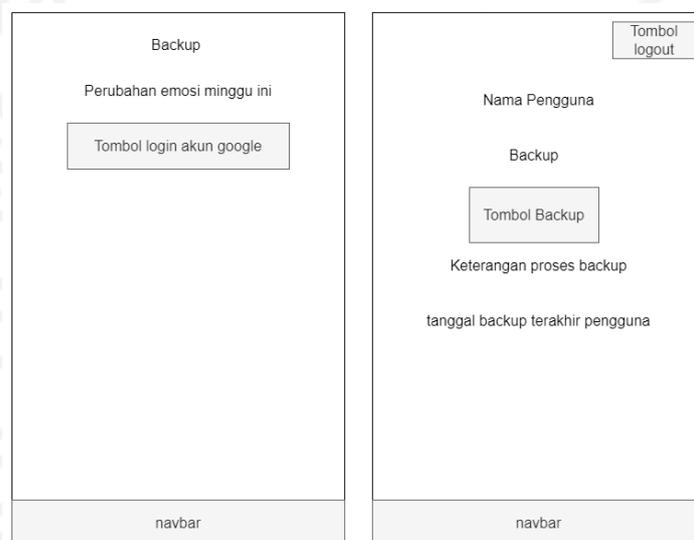
e. Rancangan halaman visualisasi perubahan emosi

Halaman visualisasi perubahan emosi adalah halaman yang menampilkan gambaran perubahan emosi pengguna dalam sebuah *bar chart* dan *pie chart*. Terdapat dua *bar chart* yang menggambarkan perubahan emosi di minggu ini dan minggu lalu. Selain itu juga terdapat dua *pie chart* yang menggambarkan jumlah pencatatan emosi yang telah dibuat pengguna, berupa jumlah kategori dan kegiatan yang dipilih. Rancangan halaman visualisasi perubahan emosi ditunjukkan pada Gambar 3.15.

Gambar 3.15 Rancangan Halaman Visualisasi Perubahan Emosi

f. Rancangan halaman *backup*

Halaman *backup* adalah halaman dimana pengguna dapat melakukan *backup* data catatan emosi yang sudah dilakukan. Halaman ini menampilkan tombol *login* menggunakan akun *Google* ketika pengguna belum masuk ke dalam aplikasi. Ketika pengguna sudah melakukan login, halaman akan menampilkan tombol “*backup*” yang akan menyimpan data catatan emosi pengguna ke penyimpanan *cloud*, halaman juga menampilkan tombol “*logout*” yang akan mengeluarkan pengguna dari aplikasi. Selain itu halaman ini juga menampilkan teks nama pengguna, status berhasil atau gagalnya *backup* data, dan tanggal *backup* terakhir pengguna. Rancangan halaman backup ditunjukkan pada Gambar 3.16.



Gambar 3.16 Rancangan Halaman Backup

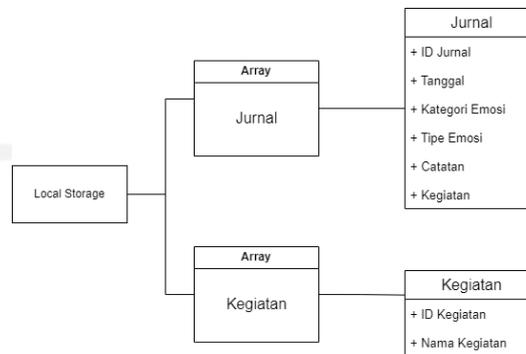
3.3.4 Perancangan Basis Data

Pada pengembangan aplikasi Jurnal Emosi digunakan dua basis data yaitu *Local Storage* dan *Firebase Firestore*. *Local storage* digunakan untuk menyimpan data catatan emosi sementara sebelum pengguna melakukan *login*, sedangkan *database Firestore* digunakan untuk menyimpan data yang berada di *local storage* ke penyimpanan *cloud* setelah pengguna melakukan *login*. Hal ini memungkinkan aplikasi dapat tetap digunakan tanpa harus melakukan *login* yang membuat aplikasi dapat dijalankan sepenuhnya secara *offline*.

a. Struktur Basis Data *Local Storage*

Pada basis data *local storage* data disimpan dalam bentuk *array* yang berisi objek catatan emosi. Data disimpan berdasarkan format penulisan JSON yang direpresentasikan dengan pasangan *key* dan *value* seperti penulisan objek pada *Javascript*. Data catatan emosi disimpan dalam *array* “Jurnal” yang berisi objek yang memiliki nilai *id jurnal*, *tanggal*, *kategori emosi*,

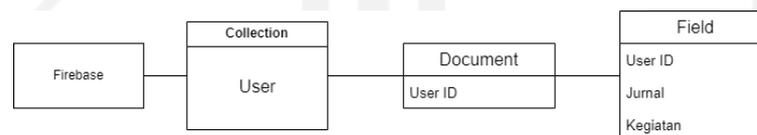
tipe emosi, catatan narasi, dan kegiatan. Sedangkan data daftar kegiatan disimpan dalam *array* “Kegiatan” yang berisi objek yang memiliki nilai id kegiatan dan nama kegiatan. Struktur basis data pada *local storage* digambarkan menggunakan diagram pohon yang ditunjukkan pada Gambar 3.17 sebagai berikut.



Gambar 3.17 Struktur Basis Data Local Storage

b. Struktur Basis Data *Firebase Firestore*

Pada basis data *firebase firestore* dilakukan pengambilan data jurnal dan kegiatan dari *local storage*. Dilakukan enkripsi pada array “Jurnal” dan “Kegiatan” dalam *local storage* menjadi bentuk *string* yang lalu disimpan sebagai nilai pada *field* dalam *document* pengguna ke dalam *collection* “User” di penyimpanan *firestore*. Struktur basis data pada *firebase firestore* digambarkan menggunakan diagram pohon yang ditunjukkan pada Gambar 3.18 sebagai berikut.



Gambar 3.18 Struktur Basis Data *Firebase Firestore*

3.4 Fase Implementasi (*Implementation*)

Pada tahap ini dilakukan pengembangan aplikasi berdasarkan hasil rancangan pada tahap sebelumnya. Hasil implementasi akan diuji untuk mengetahui kesesuaian fitur, kinerja, dan performa aplikasi berdasarkan kebutuhan yang diharapkan.

3.4.1 Memulai Proyek

Aplikasi Jurnal Emosi dibuat menggunakan *framework VueJS* dengan memanfaatkan teknologi *Progressive Web App (PWA)*. Dalam mengembangkan aplikasi diperlukan penginstalan *VueJS* terlebih dahulu dengan memanggil perintah instalasi menggunakan

package manager Yarn pada *terminal* atau *command line* dalam komputer. Perintah yang digunakan untuk menginstal *VueJS* ke dalam komputer ditunjukkan pada Gambar 3.19 berikut.

```
yarn global add @vue/cli
```

Gambar 3.19 Perintah instalasi *VueJS* ke dalam komputer.

Setelah *VueJS* berhasil diinstal tahap selanjutnya adalah memulai proyek *vue*. Perintah yang digunakan untuk memulai proyek *vue* ditunjukkan pada Gambar 3.20 berikut.

```
vue create jurnal-emosi-vue-pwa
cd jurnal-emosi-vue-pwa
```

Gambar 3.20 Perintah memulai proyek *VueJS*.

3.4.2 Implementasi *Progressive Web App (PWA)*

Langkah pengembangan selanjutnya adalah menerapkan *PWA* ke dalam aplikasi. *VueJS* menyediakan *command* untuk menginstal *file* dan konfigurasi yang dibutuhkan dalam mengembangkan aplikasi *PWA*, yang ditunjukkan pada Gambar 3.21 berikut.

```
vue add pwa
```

Gambar 3.21 Perintah memasang *PWA* pada proyek *VueJS*.

Setelah berhasil memasang *PWA*, *vue* secara otomatis membuat *file* baru pada direktori */src* yang bernama *registerServiceWorker.js*, *File* ini lalu diimpor ke dalam file utama aplikasi *main.js* yang akan *generate* konfigurasi *service worker* ketika dilakukan proses *build*.

Isi *file registerServiceWorker.js* ditunjukkan pada Gambar 3.22 berikut.

```
/* eslint-disable no-console */
import { register } from 'register-service-worker'

if (process.env.NODE_ENV === 'production') {
  register(`${process.env.BASE_URL}service-worker.js`, {
    ready () {
      console.log(
        'App is being served from cache by a service worker.\n' +
        'For more details, visit https://goo.gl/AFskqB'
      )
    },
    registered () {
      console.log('Service worker has been registered.')
    },
    cached () {
      console.log('Content has been cached for offline use.')
    },
    updatefound () {
      console.log('New content is downloading.')
    },
  })
}
```

```

updated () {
  console.log('New content is available; please refresh.')
},
offline () {
  console.log('No internet connection found. App is running in offline
mode.')
},
error (error) {
  console.error('Error during service worker registration:', error)
}
})
}

```

Gambar 3.22 Isi file *registerServiceWorker.js*

Secara *default* implementasi file tersebut telah memberikan *browser* kemampuan untuk mengunduh aplikasi Jurnal Emosi agar dapat digunakan secara *offline*.

3.4.3 Konfigurasi *Manifest File*

Dilakukan konfigurasi *Manifest File* untuk mendefinisikan perilaku aplikasi PWA ketika di-*install* ke dalam perangkat. Penulisan konfigurasi dilakukan pada file *vue.config.js* yang berada di *root* folder proyek. Isi dari file tersebut ditunjukkan pada Gambar 3.23 berikut.

```

const { defineConfig } = require('@vue/cli-service')
module.exports = defineConfig({
  transpileDependencies: true,
  pwa: {
    name: 'Jurnal Emosi PWA',
    short_name: 'Jurnal Emosi',
    theme_color: '#f8d888',
    icons: [
      {
        src: './src/assets/diary1.png',
        type: "image/png",
        sizes: "192x192",
        purpose: "maskable any"
      },
      {
        src: './src/assets/diary2.png',
        type: "image/png",
        sizes: "512x512",
        purpose: "maskable any"
      },
    ],
    start_url: "/",
    display: "standalone",
    scope: "/",
  }
})

```

Gambar 3.23 Konfigurasi *Manifest File* PWA pada proyek aplikasi Jurnal Emosi

Penjelasan tiap objek konfigurasi antara lain:

- a. *Name*: Mendefinisikan nama aplikasi PWA pada saat instalasi ke dalam perangkat.

- b. *Short-name*: Mendefinisikan singkatan nama aplikasi PWA.
- c. *Theme_color*: Mendefinisikan tema warna aplikasi PWA.
- d. *Icons*: Mendefinisikan *icon* yang digunakan aplikasi PWA
- e. *Start_url*: Mendefinisikan halaman awal yang ditampilkan pertama kali.
- f. *Display*: Mendefinisikan bagaimana aplikasi ditampilkan.

3.4.4 Implementasi *Vue Router*

Langkah selanjutnya adalah menambahkan fungsi *vue-router* untuk keperluan navigasi aplikasi. Pertama dilakukan instalasi *plugin vue-router* dengan menggunakan perintah yang ditunjukkan pada Gambar 3.24 berikut.

```
Yarn add vue-router
```

Gambar 3.24 Perintah memasang *plugin vue-router* pada proyek *VueJS*.

Setelah *plugin vue-router* berhasil diinstal tahap selanjutnya adalah membuat folder konfigurasi *router* pada direktori */src* yang berisi *file* dengan nama *index.js*. Pada *file index.js* dilakukan impor fungsi *createRouter* dan *createWebHistory* yang berasal dari *plugin vue-router*. Isi kodingan *file index.js* ditunjukkan pada Gambar 3.25 berikut.

```
import { createRouter, createWebHistory } from 'vue-router'

import Home from '@/views/Home.vue'
import Entries from '@/views/Entries.vue'
import User from '@/views/User.vue'
import Tables from '@/views/Tables.vue'

const routes = [
  {
    path: '/',
    name: 'home',
    component: Home
  },
  {
    path: '/entries',
    name: 'entries',
    component: Entries,
  },
  {
    path: '/user',
    name: 'user',
    component: User,
  },
  {
    path: '/tables',
    name: 'tables',
    component: Tables,
  },
]
```

```
const router = createRouter({
  history: createWebHistory(process.env.BASE_URL),
  routes
})
export default router
```

Gambar 3.25 Konfigurasi *vue-router* pada aplikasi Jurnal Emosi.

Tahap selanjutnya adalah memperbarui kodingan pada *file main.js* dengan melakukan impor *file* konfigurasi *vue-router* yang sudah dibuat. Isi *file main.js* setelah dilakukan penerapan *vue-router* ditunjukkan pada Gambar 3.26 berikut.

```
// main.js
import { createApp } from 'vue'
import App from './App.vue'
import './registerServiceWorker'
import router from './router'
createApp(App).use(router).mount('#app')
```

Gambar 3.26 Isi *file main.js* dalam proyek setelah menambahkan *Vue-Router*.

3.4.5 Implementasi *Firebase*

Dalam pengembangan aplikasi Jurnal Emosi digunakan *Firebase* sebagai *backend* aplikasi yang mengelola autentikasi *login* pengguna dan menyimpan data catatan emosi ke penyimpanan *cloud*. Penerapan *firebase* pada aplikasi Jurnal Emosi memungkinkan pengguna untuk menyimpan perubahan yang dilakukan dan mengakses data tersebut di perangkat yang berbeda.

Untuk menggunakan fitur-fitur yang disediakan oleh *firebase*, pengembang perlu melakukan inisiasi proyek aplikasi pada *website firebase*. Setelah berhasil mendaftarkan proyek, *firebase* akan memberikan konfigurasi SDK untuk ditambahkan pada proyek pengembangan. Contoh konfigurasi ditunjukkan pada Gambar 3.27 berikut.

Then, initialize Firebase and begin using the SDKs for the products you'd like to use.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import { getAnalytics } from "firebase/analytics";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyD3y5pC6zidN...",
  authDomain: "jurnal-emosi.firebaseio.com",
  projectId: "jurnal-emosi",
  storageBucket: "jurnal-emosi.appspot.com",
  messagingSenderId: "25673...",
  appId: "1:25673...:web:4205e4...",
  measurementId: "G-09..."
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const analytics = getAnalytics(app);
```

Gambar 3.27 Konfigurasi SDK *Firebase*.

Setelah menambahkan konfigurasi SDK ke dalam proyek, pengembang dapat menggunakan layanan-layanan yang disediakan *firebase* dengan mereferensi *file* konfigurasi *firebase* pada setiap kodingan fitur yang digunakan.

Untuk menggunakan layanan *hosting*, perlu dilakukan instalasi *plugin firebase* ke dalam proyek aplikasi dengan memanggil perintah yang ditunjukkan pada Gambar 3.28 berikut.

```
Yarn add firebase
Firebase init hosting
```

Gambar 3.28 Perintah instalasi *plugin firebase* pada proyek aplikasi.

3.4.6 Implementasi Aplikasi

Setelah berhasil menerapkan seluruh fungsi pada tahapan-tahapan preparasi sebelumnya, proyek PWA siap untuk menambahkan fitur lanjutan yang diperlukan aplikasi Jurnal Emosi berdasarkan hasil analisis kebutuhan dan perancangan sistem yang telah dilakukan pada fase *requirements planning* dan *design system*. Adapun hasil dari tahapan ini akan dibahas dalam bab selanjutnya.

3.4.7 Implementasi *Hosting* Aplikasi

Aplikasi Jurnal Emosi yang sudah dikembangkan perlu di-*deploy* ke internet agar bisa diakses oleh pengguna. Pada tahap ini dilakukan proses *build* untuk menyatukan seluruh *file* dan *modul* aplikasi menjadi satu versi ringan yang dapat didistribusi ke internet. Hasil dari proses ini berupa folder bernama “*dist*” pada *root* folder proyek. Perintah untuk melakukan *build* aplikasi ditunjukkan pada Gambar 3.29 berikut.

```
Yarn build
```

Gambar 3.29 Perintah *build* aplikasi *VueJS*.

Setelah folder *dist* dibuat langkah selanjutnya adalah men-*deploy* aplikasi dengan menggunakan layanan *firebase hosting*. Perintah untuk melakukan *deploy* aplikasi ditunjukkan pada Gambar 3.30 berikut.

```
Firestore deploy dist
```

Gambar 3.30 Perintah *deploy* aplikasi *VueJS*.

Setelah menjalankan perintah *deploy* *firebase* akan memberikan *link* *hosting* aplikasi yang dapat dibuka dan dijalankan. Aplikasi Jurnal Emosi dapat dibuka melalui link <https://jurnal-emosi.web.app/>.

3.5 Pengujian Aplikasi

Pada tahap ini dilakukan pengujian terhadap aplikasi yang sudah dikembangkan untuk memastikan apakah sistem dapat berjalan sesuai yang diharapkan. Pengujian dilakukan menggunakan metode *Blackbox Testing* untuk menguji fungsi aplikasi dan *Lighthouse Testing* yang disediakan oleh *browser Google Chrome* untuk mengukur performa aplikasi.

3.5.1 *Black Box Testing*

Pengujian *Black box* adalah pengujian yang berfokus pada spesifikasi fungsional aplikasi tanpa mengetahui struktur kode pembuatannya. Pada penelitian ini dilakukan uji coba terhadap setiap fungsi aplikasi Jurnal Emosi.

3.5.2 *Lighthouse Testing*

Digunakan pengujian *Lighthouse Testing* untuk mengukur performa dan kelayakan aplikasi dalam menerapkan fungsi PWA. *Lighthouse Testing* akan memberikan hasil pengujian berupa *rating* performa, aksesibilitas, SEO, dan *Progressive Web App* pada aplikasi.

3.5.3 *System Usability Scale (SUS)*

Digunakan pengujian *System Usability Scale (SUS)* untuk mengetahui apakah sistem sudah berjalan dengan baik sesuai harapan. Pengujian dilakukan dengan menyebarkan kuesioner yang berisikan pertanyaan terkait kegunaan sistem dan kenyamanan pengguna dalam menggunakan sistem.

3.5.4 *Prosedur Pengujian Black Box Testing dan Lighthouse Testing*

Pengujian aplikasi Jurnal Emosi dilakukan dengan persiapan sebagai berikut.

a. Perangkat Keras

Perangkat keras berupa *laptop* dengan spesifikasi sebagai berikut:

1. *Processor Intel Core i5-9300H*
2. *16GB RAM*
3. *512GB SSD*
4. *GPU Nvidia GTX 1650*

Perangkat keras lain berupa *smartphone android* dengan spesifikasi sebagai berikut:

1. *Processor Qualcomm Snapdragon 730*
2. *GPU Adreno 618*
3. *6GB RAM*

b. Perangkat Lunak

Perangkat lunak yang digunakan dalam perangkat *laptop* antara lain:

1. *Browser Google Chrome Dev Versi 110.0.5464.2*

2. Sistem operasi *Windows 11*

Perangkat lunak yang digunakan dalam perangkat *smartphone android* antara lain:

1. Sistem operasi *Android 10*
2. *Browser Google Chrome Android* Versi 100.0.4896.79



BAB IV

HASIL DAN PEMBAHASAN

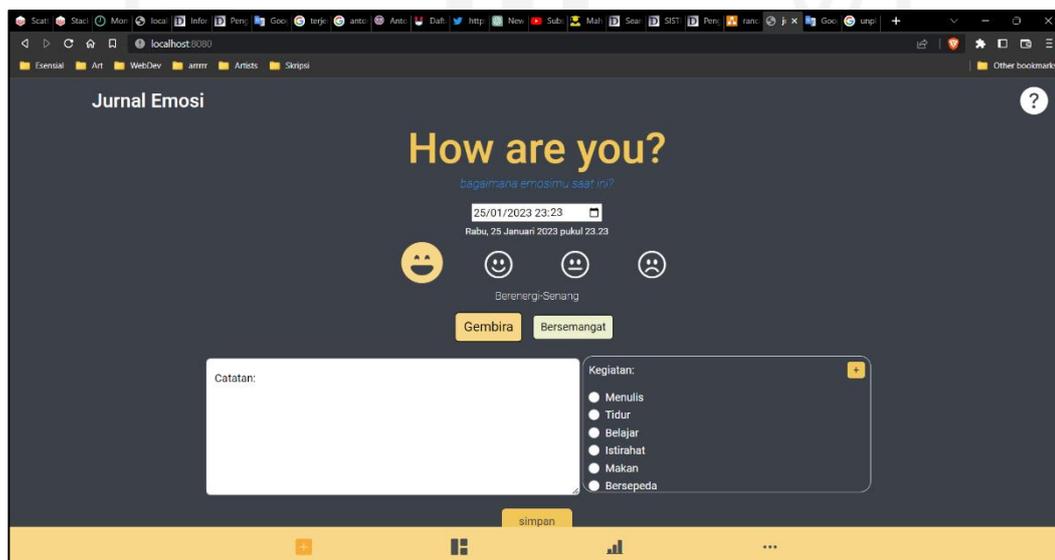
4.1 Hasil Implementasi Aplikasi

Pada bagian ini diuraikan hasil implementasi fitur aplikasi Jurnal Emosi yang dibuat berdasarkan perancangan kebutuhan dan desain sistem yang sudah dilakukan.

Hasil implementasi ditunjukkan pada tangkapan layar tampilan setiap halaman beserta penjelasan fungsinya sebagai berikut.

4.1.1 Tampilan Halaman Catat Emosi

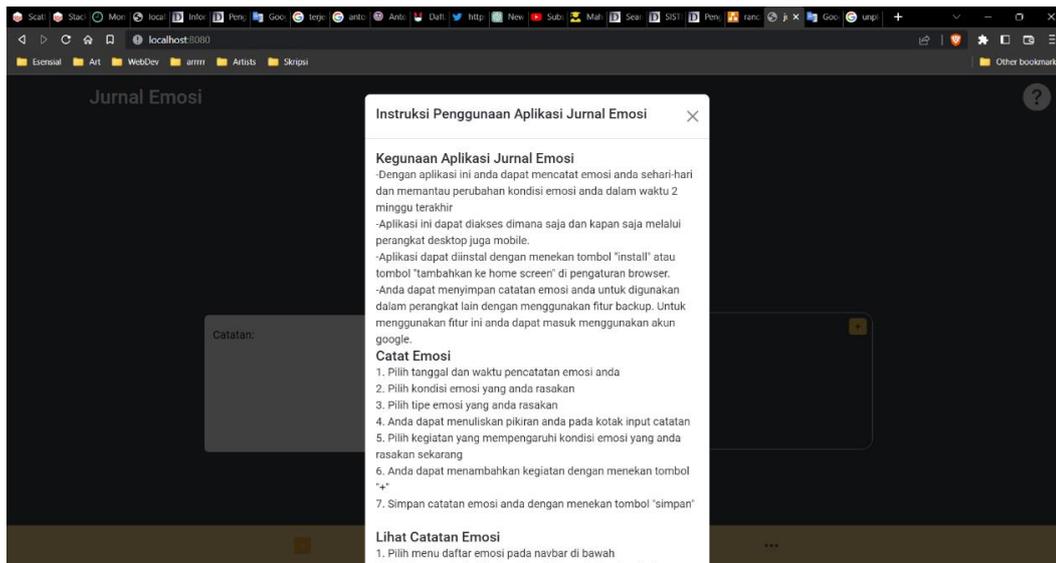
Halaman catat emosi merupakan halaman fitur *collection*. Pada halaman ini pengguna dapat melakukan pencatatan emosinya dengan mengisi *input* formulir pencatatan berupa tanggal pencatatan, kategori emosi, tipe emosi yang dirasakan, narasi jurnal dan kegiatan. Penerapan kategori dan tipe emosi didasari oleh klasifikasi emosi *Pick-A-Mood* yang mengkategorikan emosi ke dalam empat kategori dan delapan tipe yang berbeda. Setelah mengisi formulir pengguna perlu mengirimkan data pencatatan emosi dengan menekan tombol “simpan” dimana sistem akan menyimpan informasi tersebut ke penyimpanan *local storage*. Pada halaman ini pengguna juga dapat melihat instruksi penggunaan aplikasi dan menambah daftar kegiatan. Adapun hasil tampilan halaman ini ditunjukkan pada Gambar 4.1.



Gambar 4.1 Tampilan Halaman Catat Emosi

4.1.2 Tampilan Halaman Instruksi Penggunaan Aplikasi

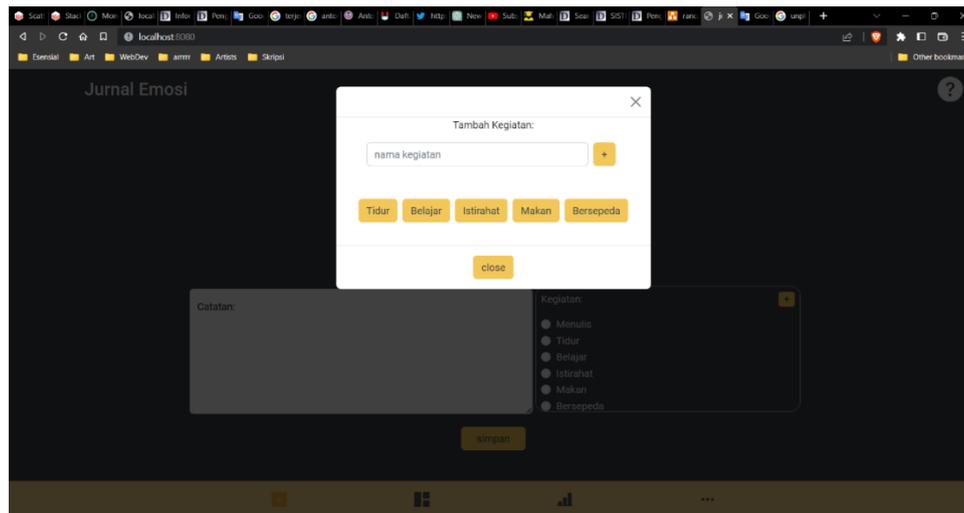
Halaman instruksi penggunaan aplikasi merupakan halaman fitur *preparation*. Setelah pengguna menekan tombol “?” pada halaman catat emosi, sistem akan menampilkan *popup* halaman yang berisi instruksi penggunaan aplikasi. Pengguna dapat menutup halaman ini dengan menekan tombol “X” atau tombol “close”. Adapun hasil tampilan halaman ini ditunjukkan pada Gambar 4.2



Gambar 4.2 Tampilan Halaman Instruksi Penggunaan Aplikasi

4.1.3 Tampilan Halaman Tambah Kegiatan

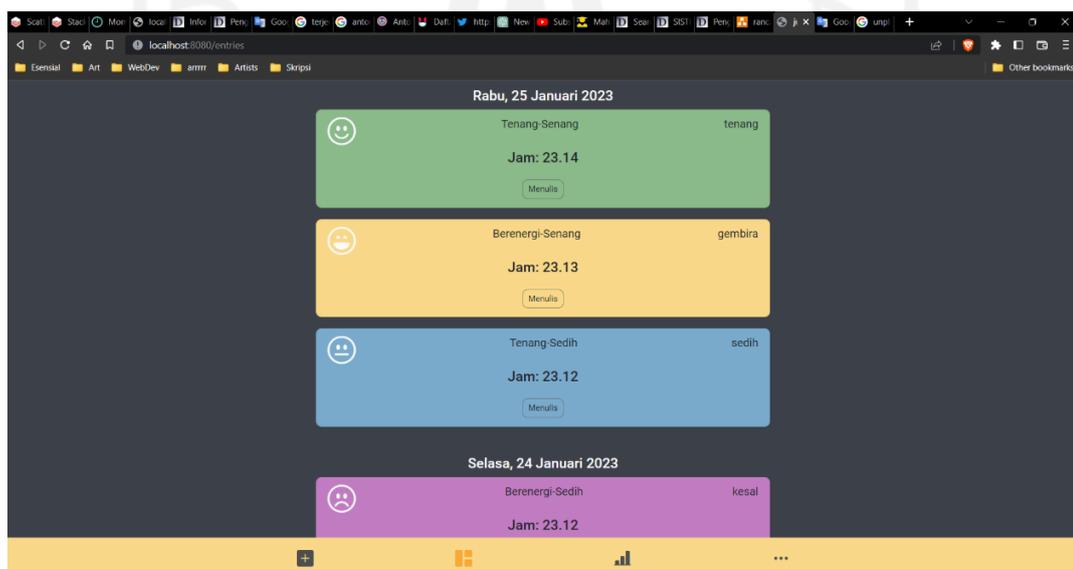
Setelah pengguna menekan tombol tambah kegiatan pada halaman catat emosi, sistem akan menampilkan *popup* halaman tambah kegiatan. Pada halaman ini pengguna dapat menambahkan kegiatan baru dengan mengisi *input* nama kegiatan dan menekan tombol “+” di samping kolom *input*, dimana sistem akan menyimpan kegiatan baru ke dalam *local storage*. Pengguna juga dapat menghapus kegiatan yang telah ditambahkan dengan menekan salah satu dari daftar kegiatan yang ditampilkan di bawah kolom *input*. Pengguna dapat menutup halaman ini dengan menekan tombol “X” atau tombol “close”. Adapun hasil tampilan halaman ini ditunjukkan pada Gambar 4.3.



Gambar 4.3 Tampilan Halaman Tambah Kegiatan

4.1.4 Tampilan Halaman Daftar Catatan Emosi

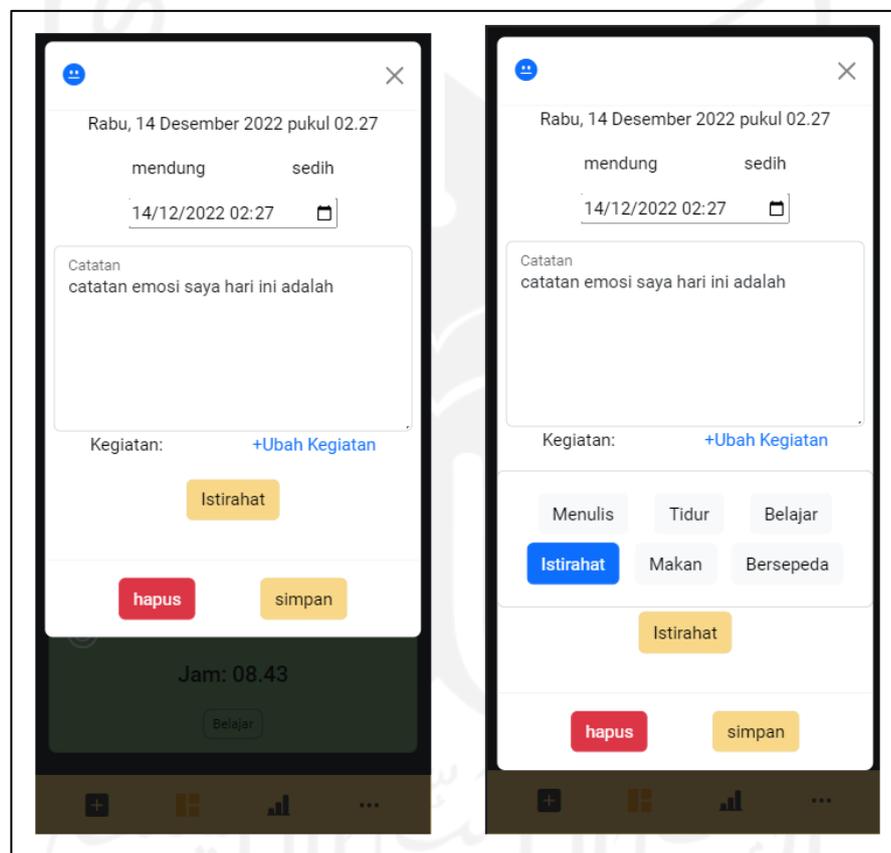
Halaman ini merupakan salah satu halaman fitur *reflection*. Pada halaman ini sistem akan menampilkan daftar catatan emosi yang telah dibuat pengguna. Catatan emosi dibedakan berdasarkan tanggal dilakukannya pencatatan dan diurutkan secara menurun. Catatan emosi ditampilkan dalam bentuk kartu yang berisi tentang informasi pencatatan emosi dan diwarnai berdasarkan kategori emosi yang dipilih pengguna. Pengguna juga dapat mengubah atau menghapus data pencatatan dengan menekan salah satu kartu catatan emosi yang akan menampilkan *popup* halaman detail pencatatan. Adapun hasil tampilan halaman ini ditunjukkan pada Gambar 4.4.



Gambar 4.4 Tampilan Halaman Daftar Catatan Emosi

4.1.5 Tampilan Halaman Kelola Catatan Emosi

Pada halaman ini pengguna dapat melihat detail pencatatan emosi yang telah dilakukan dan melakukan perubahan secara langsung dengan mengubah informasi pada *form* input tanggal dan narasi jurnal. Pada halaman ini pengguna dapat mengubah kegiatan yang dilakukan dengan menekan tombol “+Ubah Kegiatan” dimana sistem akan menampilkan atau menyembunyikan *dropdown* menu pilihan kegiatan. Pengguna juga dapat menghapus catatan emosi dengan menekan tombol “hapus” berwarna merah. Semua perubahan yang dilakukan akan disimpan ke dalam *local storage*. Adapun hasil tampilan halaman ini ditunjukkan pada Gambar 4.5.

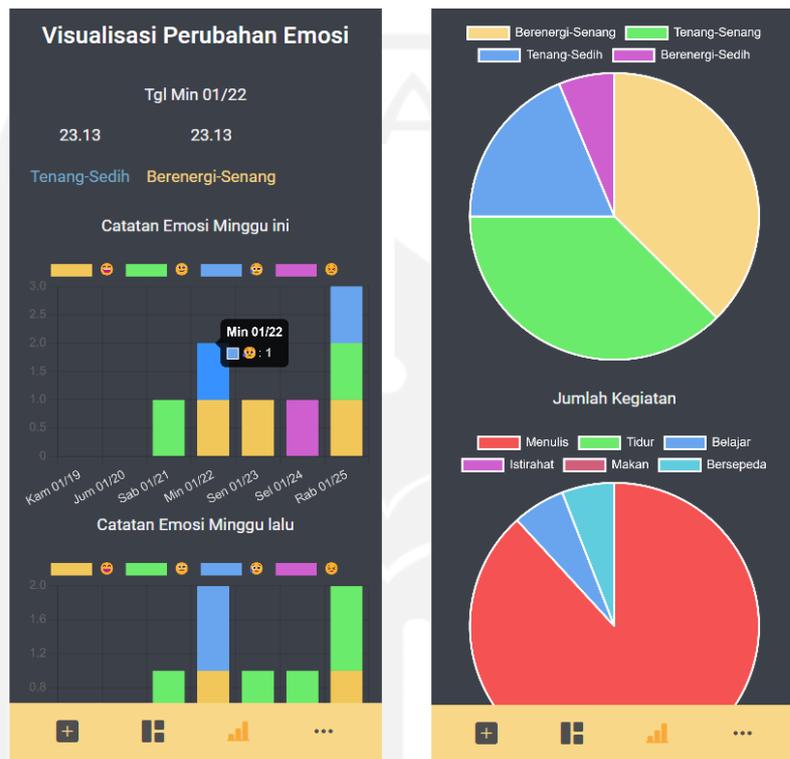


Gambar 4.5 Tampilan Halaman Kelola Catatan Emosi

4.1.6 Tampilan Halaman Visualisasi Perubahan Emosi

Halaman ini merupakan salah satu halaman fitur *reflection*. Pada halaman ini sistem menampilkan visualisasi perubahan emosi ke dalam bentuk *bar chart* dan *pie chart*. *Bar chart* digunakan untuk menggambarkan perubahan kondisi emosi pengguna dimana aksis horizontal mempresentasikan tanggal dan aksis vertikal mempresentasikan jumlah kategori emosi yang dipilih pengguna pada tanggal tersebut, pengguna juga dapat melihat detail waktu pencatatan

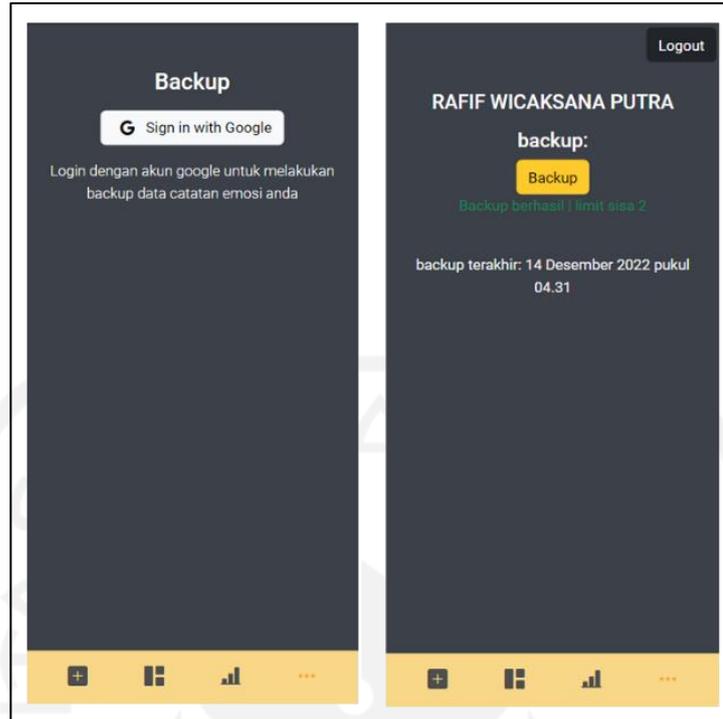
dengan menekan salah satu *bar* pada *chart* tersebut. Pada halaman ini sistem menampilkan dua *bar chart* perubahan emosi dalam rentang waktu dua minggu terakhir untuk memudahkan pengguna dalam membandingkan kondisi emosinya pada minggu ini dengan minggu yang lalu. *Pie chart* digunakan untuk menggambarkan jumlah tiap kategori emosi dan kegiatan berdasarkan data pencatatan yang telah dilakukan pengguna. Adapun hasil tampilan halaman ini ditunjukkan pada Gambar 4.6



Gambar 4.6 Tampilan Halaman Visualisasi Perubahan Emosi

4.1.7 Tampilan Halaman Backup

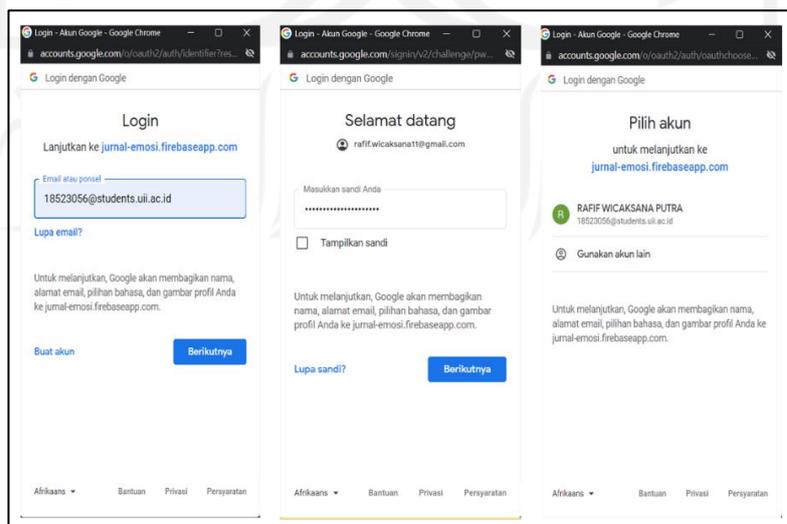
Halaman ini merupakan halaman fitur *action*. Pada halaman *backup* pengguna dapat menyimpan data catatan emosi ke penyimpanan *cloud firebase* dengan menekan tombol “*Backup*” setelah berhasil masuk menggunakan akun *Google*. Pengguna dapat keluar dari aplikasi dengan menekan tombol “*Logout*” dimana sistem akan menghapus data *session* pengguna dan memuat ulang halaman. Adapun hasil tampilan halaman ini ditunjukkan pada Gambar 4.7.



Gambar 4.7 Tampilan Halaman *Backup* Data

4.1.8 Tampilan Halaman *Login*

Setelah pengguna menekan tombol “*Sign in with Google*” pada halaman *backup* data, sistem akan menampilkan jendela baru berupa halaman *login* yang terintegrasi dengan layanan autentikasi *firebase* milik *Google*. Pada kasus pengguna pertama kali masuk ke aplikasi, sistem akan menyimpan data catatan dan autentikasi pengguna ke *firebase*. Selain itu sistem akan mengambil data catatan dari penyimpanan *cloud* dan menyimpannya ke *local storage*. Adapun hasil tampilan halaman ini ditunjukkan pada Gambar 4.8.



Gambar 4.8 Tampilan Halaman *Login*

4.2 Hasil Pengujian Aplikasi

4.2.1 Pengujian *Black box*

Dilakukan pengujian *black box* pada aplikasi yang sudah dikembangkan untuk memastikan bahwa fungsionalitas dan tampilan aplikasi dapat dijalankan dan bekerja dengan baik sesuai dengan hasil yang diharapkan. Hasil dari pengujian fungsionalitas aplikasi menggunakan *black box* disajikan dalam Tabel 4.1 dan hasil pengujian antarmuka aplikasi ditunjukkan pada Tabel 4.2.

Tabel 4.1 Hasil Pengujian *Black Box* Fungsionalitas Aplikasi Jurnal Emosi

No	Kegiatan	Tahapan Kegiatan	Hasil yang Diharapkan	Hasil
1	Lihat instruksi penggunaan aplikasi	1. Pengguna menekan tombol lihat instruksi pada halaman catatan emosi	Sistem menampilkan Halaman instruksi penggunaan aplikasi Jurnal Emosi	Berhasil
2	Catat Emosi	1. Pengguna menuju halaman catat emosi 3. Mengisi formulir catat emosi 4. Memilih tanggal pencatatan emosi 5. Memilih kategori emosi 6. Memilih tipe emosi 7. Mengisi jurnal narasi 8. Memilih kegiatan 9. Menekan tombol “simpan”	Sistem menyimpan data catatan emosi ke <i>local storage</i> .	Berhasil
3	Tambah Kegiatan	1. Menekan tombol “+” dalam <i>box input</i> daftar kegiatan 2. Mengisi <i>input</i> nama kegiatan 3. Menekan tombol “+” tambah kegiatan	Sistem menyimpan data kegiatan baru ke <i>local storage</i>	Berhasil
4	Lihat daftar Catatan Emosi	1. Pengguna menuju halaman catatan emosi	Sistem menampilkan halaman daftar catatan emosi	Berhasil
5	Mengelola catatan emosi	1. Menekan salah satu catatan emosi 2. Mengubah data catatan emosi 3. Menekan tombol “simpan “ 4. Menekan tombol “hapus”	Sistem Menyimpan perubahan catatan emosi ke <i>local storage</i>	Berhasil
6.	Lihat Visualisasi Perubahan Emosi	1. Pengguna menuju halaman visualisasi perubahan emosi	Sistem menampilkan halaman visualisasi perubahan emosi berupa <i>bar chart</i> dan <i>pie chart</i> berdasarkan data pencatatan yang telah dilakukan	Berhasil
7	Login	1. Pengguna menuju halaman <i>backup</i> data	Sistem menyimpan data catatan emosi dan	Berhasil

		2. Menekan tombol “ <i>Sign in with Google</i> ” 3. Melakukan <i>login</i> dengan akun <i>Google</i>	autentikasi pengguna ke <i>firebase</i>	
8	<i>Backup data catatan emosi</i>	1. Pengguna menuju halaman backup data 2. Menekan tombol “Backup”	Sistem melakukan enkripsi dan menyimpan data catatan emosi ke penyimpanan <i>firebase firestore</i>	Berhasil
9	<i>Logout</i>	1. Pengguna menuju halaman backup data 2. Menekan tombol “Logout”	Sistem menghapus data <i>session</i> pengguna dan mengeluarkan pengguna dari aplikasi	Berhasil
10	<i>Offline</i>	1. Mematikan koneksi internet perangkat 2. Melakukan uji coba kegiatan no 1 – 6	Aplikasi tetap dapat dijalankan secara <i>offline</i>	Berhasil
11	<i>Offline setelah login</i>	1. Melakukan <i>login</i> ke aplikasi 2. Mematikan koneksi internet 3. Melakukan uji coba kegiatan no 1 – 6 4. Menyalakan koneksi internet kembali	Aplikasi tetap dapat dijalankan secara <i>offline</i> dan sistem mengirimkan perubahan pengguna ke penyimpanan <i>firebase</i>	Berhasil

Tabel 4.2 Hasil Pengujian *Black Box* Antarmuka Aplikasi Jurnal Emosi

No	Halaman	Kegiatan	Hasil yang Diharapkan	Hasil
1	Catat Emosi	Menekan <i>input</i> tanggal	Sistem menampilkan antarmuka pemilihan tanggal dan waktu	Berhasil
		Menekan tombol “?” instruksi penggunaan aplikasi	Sistem menampilkan halaman instruksi penggunaan aplikasi	Berhasil
		Menekan <i>icon emoji</i> kategori emosi	Sistem memperbesar dan memberi warna icon emoji sesuai kategori yang dipilih dan menampilkan teks nama kategori. Sistem juga menampilkan dua pilihan tipe emosi yang dapat dipilih pengguna	Berhasil
		Menekan pilihan tipe emosi	Sistem memperbesar dan memberi warna pilihan sesuai kategori yang dipilih	Berhasil
		Menekan <i>input text area</i> catatan narasi	Sistem memberi fokus pada input dan pengguna dapat menuliskan narasi	Berhasil

		Menekan salah satu kegiatan pada pilihan <i>radio button</i>	Sistem memberi fokus pada kegiatan yang dipilih	Berhasil
		Menekan tombol “+” tambah kegiatan	Sistem menampilkan <i>popup</i> halaman tambah kegiatan	Berhasil
		Menekan tombol “simpan”	Sistem menyimpan data catatan dan mengalihkan pengguna ke halaman daftar catatan emosi	Berhasil
2	Instruksi penggunaan aplikasi	Melihat instruksi penggunaan aplikasi	Sistem menampilkan teks instruksi penggunaan aplikasi	Berhasil
		Menekan tombol “X” keluar aplikasi	Sistem menutup halaman instruksi	Berhasil
		Menekan tombol “close” keluar aplikasi	Sistem menutup halaman instruksi	Berhasil
3	Daftar catatan emosi	Melihat daftar catatan emosi yang sudah dilakukan	Sistem menampilkan daftar catatan emosi yang dipisahkan berdasarkan tanggal pencatatan	Berhasil
		Melihat detail catatan emosi	Sistem menampilkan <i>card</i> yang berisi icon kategori emosi, teks tipe emosi, waktu pencatatan, dan kegiatan	Berhasil
		Menekan salah satu <i>card</i> catatan emosi	Sistem menampilkan halaman kelola catatan emosi	Berhasil
4	Kelola catatan emosi	Menekan <i>input</i> tanggal	Sistem menampilkan antarmuka pemilihan tanggal dan waktu	Berhasil
		Menekan <i>input text area</i> catatan narasi	Sistem memberi fokus pada input dan pengguna dapat menuliskan narasi	Berhasil
		Menekan tombol “+Ubah Kegiatan”	Sistem membuka <i>dropdown</i> daftar kegiatan yang dapat dipilih pengguna	Berhasil
			Sistem menutup <i>dropdown</i> daftar kegiatan	Gagal
		Menekan salah satu pilihan daftar kegiatan	Sistem mengubah warna kegiatan yang dipilih menjadi warna biru yang artinya kegiatan berhasil dipilih	Berhasil
		Menekan tombol “hapus”	Sistem menghapus catatan emosi dan menutup halaman kelola catatan emosi.	Berhasil

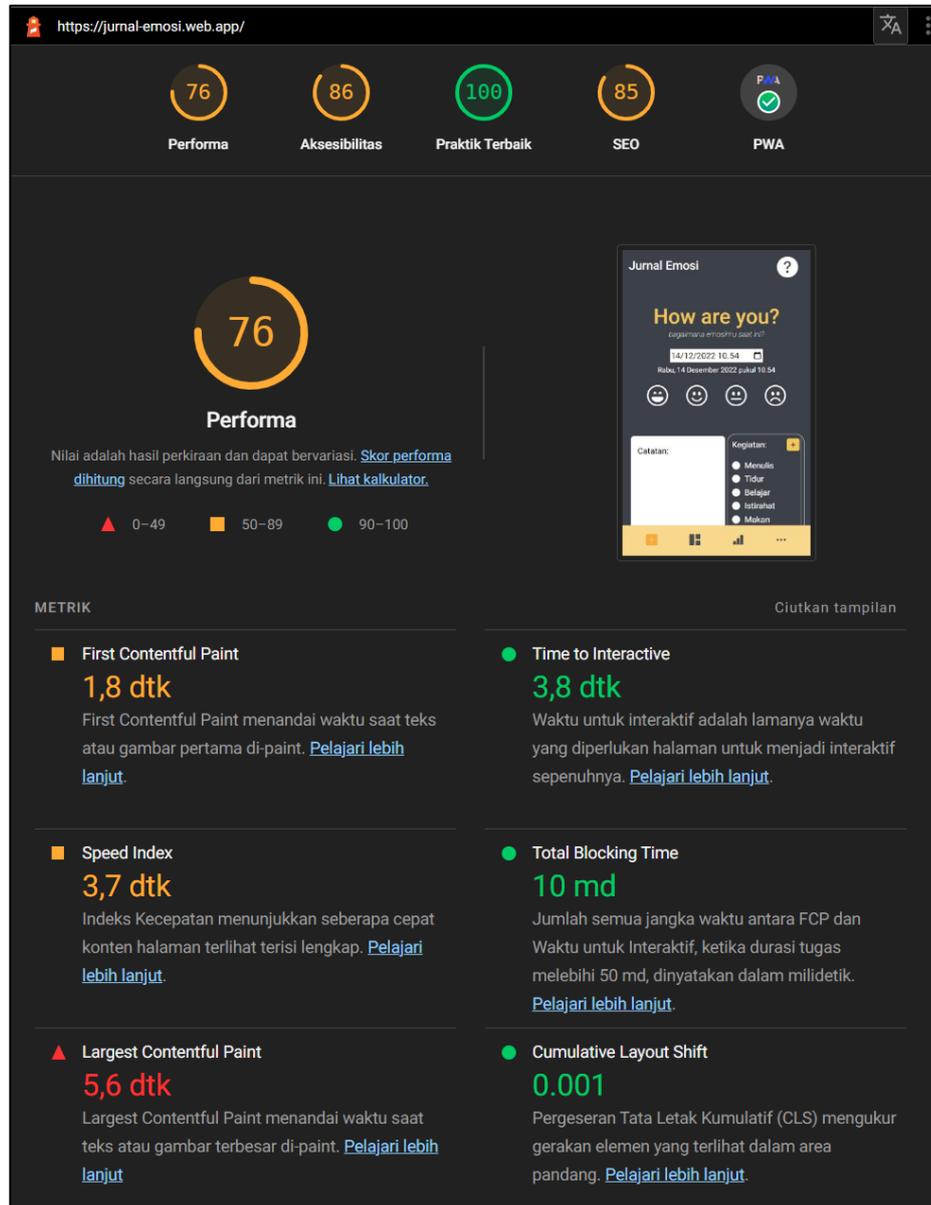
		Menekan tombol “close”	Sistem menutup halaman kelola catatan emosi.	Berhasil
5	Visualisasi perubahan emosi	Melihat perubahan emosi	Sistem menampilkan <i>bar chart</i> jumlah pencatatan emosi pengguna minggu ini	Berhasil
			Sistem menampilkan <i>bar chart</i> jumlah pencatatan emosi pengguna minggu lalu	Berhasil
			Sistem menampilkan <i>pie chart</i> jumlah kategori emosi	Berhasil
			Sistem menampilkan <i>pie chart</i> jumlah kegiatan	Berhasil
		Menekan salah satu <i>bar</i> catatan emosi dalam chart	Sistem menampilkan detail waktu pencatatan pada tanggal tersebut dan emosi yang dipilih.	Berhasil
6	Backup	Menekan tombol “Sign in with Google”	Sistem menampilkan halaman login akun google	Berhasil
		Menekan tombol “backup”	Sistem menampilkan teks keterangan backup berhasil	Berhasil
		Menekan tombol “logout”	Sistem mengeluarkan pengguna dari aplikasi dan mengubah tampilan halaman <i>backup</i> menjadi yang semula sebelum dilakukannya <i>login</i> .	Berhasil

Pada pengujian *black box* antarmuka aplikasi Jurnal Emosi terlihat bahwa semua antarmuka berfungsi dengan baik kecuali pada kegiatan menutup *dropdown* daftar kegiatan pada halaman kelola catatan emosi, hal ini disebabkan karena tidak sempurnanya integrasi *framework bootstrap* yang digunakan untuk membuat komponen antarmuka dengan *framework VueJS* yang digunakan pada pengembangan, sehingga fungsi *javascript* yang berperan untuk menutup *dropdown* tidak dapat berjalan.

4.2.2 Pengujian *Lighthouse Testing*

Dilakukan pengujian terhadap aplikasi Jurnal Emosi yang sudah dikembangkan menggunakan tools *Lighthouse Testing* yang disediakan oleh *Google Chrome Devtools*. *Lighthouse* memberikan laporan data performa, aksesibilitas, *best practice*, SEO dan *Progressive Web App* yang dipresentasikan dengan skala nilai 0 – 100, dimana skor bernilai 0

– 49 menyatakan aplikasi memiliki performa yang masih kurang, skor bernilai 50-89 menyatakan aplikasi memiliki performa yang cukup baik dan skor bernilai 90 – 100 menyatakan bahwa aplikasi memiliki performa yang sangat baik. Adapun hasil pengujian *lighthouse* ditunjukkan pada Gambar 4.9 dan penjelasan tiap metrik disajikan dalam Tabel 4.3 berikut.



Gambar 4.9 Hasil Pengujian *Lighthouse*

Tabel 4.3 Penjelasan Hasil Pengujian *Lighthouse*

Metrik		Hasil	Penjelasan
Performa		76/100	Mempresentasikan performa kecepatan aplikasi ketika dijalankan. Hasil dari metrik ini menyatakan aplikasi Jurnal Emosi memiliki performa yang cukup baik dengan nilai 76/100.
1	<i>First Contentful Paint</i>	1.8 detik	Mengukur waktu yang diperlukan aplikasi untuk me-render konten DOM (<i>Document Object Model</i>) pada layar <i>browser</i> pengguna. Sub-metrik ini memiliki bobot penilaian sebesar 10%, dimana disarankan hasil dari skor ini tidak lebih dari 1.8 detik.
2	<i>Time to Interactive</i>	3.8 detik	Mengukur waktu yang diperlukan aplikasi untuk mengisi konten awal (<i>first contentful paint</i>) pada halaman <i>browser</i> . Sub-metrik ini memiliki bobot penilaian sebesar 10%, dimana disarankan hasil dari skor ini tidak lebih dari 7.1 detik.
3	<i>Speed Index</i>	3.7 detik	Mengukur waktu yang diperlukan aplikasi untuk menampilkan keseluruhan konten pada layar <i>browser</i> pengguna. Sub-metrik ini memiliki bobot penilaian sebesar 10%, dimana disarankan hasil dari skor ini tidak lebih dari 4.3 detik.
4	<i>Total Blocking Time</i>	100 milidetik	Mengukur beberapa lama aplikasi tidak dapat merespon interaksi pengguna. Sub-metrik ini memiliki bobot penilaian yang paling besar yaitu 30%, dimana disarankan hasil dari skor ini tidak lebih dari 600 milidetik.
5	<i>Largest Contentful Paint</i>	5.6 detik	Mengukur waktu yang diperlukan aplikasi dalam memuat konten utama pada halaman. Sub-metrik ini memiliki bobot penilaian sebesar 25%, dimana disarankan hasil dari skor ini tidak lebih dari 4 detik.
6	<i>Cumulative Layout Shift</i>	0.001	Mengukur pergerakan elemen yang ditampilkan pada halaman. Sub-metrik ini memiliki bobot penilaian sebesar 25%, dimana disarankan hasil dari skor ini tidak lebih dari 0.1
Aksesibilitas		86/100	Mempresentasikan seberapa baik aplikasi dalam membantu interaksi pengguna yang memiliki kemampuan interaksi yang terbatas atau disabilitas.
<i>Best Practice</i>		100/100	Mengukur seberapa baik aplikasi dalam menerapkan panduan standar agar aplikasi <i>website</i> terhindar dari beberapa masalah umum.
SEO		85/100	Mengukur seberapa baik aplikasi dapat ditemukan dan dicari mesin pencarian.
<i>Progressive Web App</i>		8/8	Mengukur seberapa baik aplikasi dalam memenuhi aspek yang diperlukan dalam penerapan <i>Progressive Web App</i> . Adapun hasil dari pengujian metrik ini menyatakan aplikasi dapat di-instal dan digunakan secara <i>offline</i> .

Berdasarkan hasil pengujian *lighthouse testing* pada aplikasi Jurnal Emosi yang ditunjukkan oleh Gambar 4.9 dan Tabel 4.3 terlihat bahwa aplikasi Jurnal Emosi telah berhasil mengimplementasikan teknologi *Progressive Web App* (PWA) dengan performa yang cukup baik yang diukur melalui enam metrik pengujian *lighthouse*, dimana didapatkan skor akhir performa sebesar 76/100.

4.2.3 Pengujian System Usability Scale (SUS)

Pada bagian ini dilakukan pengujian dengan metode System Usability Scale. Pengujian dilakukan dengan memberikan *link hosting* aplikasi Jurnal Emosi kepada pengguna, dan memberikan kuesioner berupa *Google Form* yang terdiri dari 10 pertanyaan dengan jawaban yang memiliki skala Likert 1-5 yang berkaitan dengan kegunaan sistem dan kenyamanan pengalaman dalam menggunakannya. Pertanyaan kuesioner ditunjukkan pada Tabel 2.6 di bab II. Kuesioner diisi oleh 8 responden dengan hasil penilaian responden terhadap pertanyaan pengujian ditunjukkan pada Tabel 4.4.

Tabel 4.4 Tabel Rekap Pengujian Kuesioner SUS

No	Nama	Pekerjaan	Pertanyaan									
			1	2	3	4	5	6	7	8	9	10
P1	Rizki	Mahasiswa	3	2	4	1	3	2	4	4	4	4
P2	Bernas Cakra Sakti Harisna	Mahasiswa	5	1	5	1	5	1	5	1	5	1
P3	Andhika Putra Ramadhan	Mahasiswa	4	1	5	1	5	1	3	1	4	5
P4	Yusuf Syafi Ekananda Trisna	Mahasiswa	4	1	5	1	4	2	5	1	1	2
P5	Muhammad Daffa	Mahasiswa	4	2	4	1	3	3	4	1	4	3
P6	Indra Sukma Adhitama	Mahasiswa	4	1	5	1	5	1	5	1	5	4
P7	Muhammad Rizhan Ridha	Mahasiswa	5	2	4	3	4	2	4	2	3	3
P8	Anesty Dikra Kelana	Mahasiswa	4	1	5	1	5	1	5	1	5	2

Berdasarkan hasil pada Tabel 4.4 dilakukan proses perhitungan SUS menggunakan persamaan (2.1), (2.2), (2.3), dan (2.4) yang sudah dijelaskan pada bab II, hasil dari perhitungan tersebut lalu digunakan untuk menentukan kesimpulan pengujian SUS. Hasil dari perhitungan metode SUS dapat dilihat pada Tabel 4.5.

Tabel 4.5 Hasil Perhitungan Pengujian SUS

No	Nama	Pekerjaan	Pertanyaan										Sub Total	Sub Total x 2.5
			1	2	3	4	5	6	7	8	9	10		
P1	Rizki	Mahasiswa	3	2	4	1	3	2	4	4	4	4	25	62.5
P2	Bernas Cakra Sakti Harisna	Mahasiswa	5	1	5	1	5	1	5	1	5	1	40	100
P3	Andhika Putra Ramadhan	Mahasiswa	4	1	5	1	5	1	3	1	4	5	32	80
P4	Yusuf Syafi Ekananda	Mahasiswa	4	1	5	1	4	2	5	1	1	2	32	80
P5	Muhammad Daffa	Mahasiswa	4	2	4	1	3	3	4	1	4	3	29	72.5
P6	Indra Sukma Adhitama	Mahasiswa	4	1	5	1	5	1	5	1	5	4	36	90
P7	Muhammad Rizhan Ridha	Mahasiswa	5	2	4	3	4	2	4	2	3	3	28	70
P8	Anesty Dikra Kelana	Mahasiswa	4	1	5	1	5	1	5	1	5	2	38	95
	Rata-rata													88.25

Berdasarkan hasil perhitungan SUS didapatkan hasil akhir secara keseluruhan sebesar 88.25% dimana berdasarkan *acceptability ranges* masuk ke dalam kategori *acceptable*, dan berdasarkan *adjective ratings* masuk ke dalam kategori *excellent*. Dengan mendapatkan skor akhir B. Pada pertanyaan bernomor negatif untuk pertanyaan bernomor genap 8, salah satu pengguna masih merasa kebingungan dalam menjalankan sistem dengan penilaian tertinggi yaitu nilai 4. Kemudian pada pertanyaan bernomor genap 10, kebanyakan pengguna masih perlu membiasakan diri dalam menggunakan aplikasi dengan penilaian tertinggi yaitu nilai 5. Dari masukan yang diterima secara informal lisan, salah satu calon pengguna mengutarakan bahwa tampilan dan tata letak antarmuka aplikasi tidak seperti kebanyakan aplikasi *mobile* pada umumnya dimana calon pengguna tersebut merasa kurang terbiasa. Selain itu salah satu calon pengguna juga mengutarakan bahwa tampilan aplikasi Jurnal Emosi terlihat terlalu baku dan tidak modern. Berdasarkan penilaian SUS dan masukan dari calon pengguna dapat disimpulkan bahwa aplikasi Jurnal Emosi masih membutuhkan perbaikan dalam sisi tampilan antarmuka agar memberikan pengalaman pengguna yang baik untuk kedepannya, yang dapat dilakukan dengan mengubah tata letak dan tampilan agar lebih memberikan kesan modern menyerupai aplikasi yang banyak ditemukan di perangkat *mobile* pada umumnya.

Dengan hasil tersebut dapat disimpulkan bahwa sistem aplikasi Jurnal Emosi dapat diterima dengan cukup baik dan cukup memenuhi ekspektasi dari tujuan penelitian ini.

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian dan penjelasannya pada bab sebelumnya, penulis dapat menarik kesimpulan terkait penelitian implementasi *Progressive Web App* pada aplikasi Jurnal Emosi sebagai berikut:

- a. Aplikasi Jurnal Emosi dikembangkan menggunakan metode *Rapid Application Development* dengan menerapkan teknologi *Progressive Web App*, adapun hasil dari pengembangan ini berupa aplikasi Jurnal Emosi yang dapat mencatat dan melacak perubahan emosi pengguna.
- b. Berdasarkan hasil pengujian aplikasi, dapat disimpulkan bahwa pengembangan aplikasi Jurnal Emosi menggunakan teknologi *Progressive Web App* berhasil memberikan aplikasi kemampuan dijalankan di perangkat *desktop* dan *mobile*, dan juga dapat digunakan secara *offline* dengan performa yang cukup baik. Dimana aplikasi juga dapat di-instal dalam perangkat *mobile* layaknya aplikasi *native*.

5.2 Saran

Berdasarkan implementasi dan pengujian yang sudah dilakukan, pengembangan aplikasi ini masih kekurangan dari sisi pembuatan aplikasi. Oleh karena itu dibutuhkan saran yang dapat dipertimbangkan untuk penelitian sejenis selanjutnya, sebagai berikut:

- a. Menggunakan *framework* pembuatan komponen antarmuka yang secara pasti dapat diterapkan dengan baik pada pengembangan yang menggunakan *framework* VueJS.
- b. Menambahkan fitur yang dapat menampilkan perubahan emosi dalam rentan waktu yang dapat ditentukan.
- c. Membuat tampilan antarmuka aplikasi yang lebih baik untuk memberikan kesan modern dan lebih menyerupai aplikasi *mobile* pada umumnya.

DAFTAR PUSTAKA

- Anisa, N. (2022, April 22). *Diagram Apa Saja sih yang Termasuk ke dalam Behavioral UML Diagram? – School of Information Systems*. <https://sis.binus.ac.id/2020/04/22/diagram-apa-saja-sih-yang-termasuk-ke-dalam-behavioral-uml-diagram/>
- Aripin, S., & Somantri, S. (2021). Implementasi Progressive Web Apps (PWA) pada Repository E-Portofolio Mahasiswa. *Jurnal Eksplora Informatika*, 10(2), 148–158. <https://doi.org/10.30864/eksplora.v10i2.486>
- Ballesteros, D., & Janis, W. (2009). Coping: Stress management strategies. *Connell Research Program On Self-Injury and Recovery*.
- Baqutayan, S. M. S. (2015). Stress and Coping Mechanisms: A Historical Overview. *Mediterranean Journal of Social Sciences*. <https://doi.org/10.5901/mjss.2015.v6n2s1p479>
- Branco, R., Neves, M., 2, , 2, P. N., & Casais, M. (2020). *An Interaction Design Analysis of Mood Trackers*.
- Caldeira, C., Chen, Y., Chan, L., Pham, V., Chen, Y., & Zheng, K. (2017). *Mobile apps for mood tracking: an analysis of features and user reviews*.
- Chan, S., Torous, J., Hinton, L., & Yellowlees, P. (2015). Towards a Framework for Evaluating Mobile Mental Health Apps. *Telemedicine and E-Health*, 21(12), 1038–1041.
- Chrisyanto, A. R., Delima, R., Santoso, H. B., Wibowo, A., & Kristiawan, R. A. (2019). Developing agriculture land mapping using Rapid Application Development (RAD): A case study from Indonesia. *International Journal of Advanced Computer Science and Applications*, 10(10), 232–241. <https://doi.org/10.14569/ijacsa.2019.0101033>
- Darfiansyah, M. (2022). *Perancangan Sistem Informasi Penyewaan Perangkat Elektronik Berbasis Website Dengan Menggunakan Teknologi Progressive Web App*. <https://journals.upi-yai.ac.id/index.php/ikraith-informatika/issue/archive>
- Desmet, P. (2015). Design for Mood: Twenty Activity-Based Opportunities to Design for Mood Regulation Motives to Design for Mood Regulation. *International Journal of Design*, 9(2), 1–19. www.ijdesign.org
- Desmet, P., Romero, N., & Vastenburger, M. (2012). PICK-A-MOOD DEVELOPMENT AND APPLICATION OF A PICTORIAL MOOD-REPORTING INSTRUMENT. *8th International Design and Emotion Conference*.

- Dewi, G. L., Tjandra, S., & Ricardo. (2020). Pemanfaatan Progressive Web Apps Pada Web Akuntansi. *Teknika*, 9(1), 38–47. <https://doi.org/10.34148/teknika.v9i1.252>
- Dicoding. (2021). *Contoh Use Case Diagram Lengkap dengan Penjelasannya - Dicoding Blog*. <https://www.dicoding.com/blog/contoh-use-case-diagram/>
- Dicoding Intern. (2021). *Apa itu UML? Beserta Pengertian dan Contohnya - Dicoding Blog*. <https://www.dicoding.com/blog/apa-itu-uml/>
- Febiharsa, D., Made Sudana, I., & Hudallah, N. (2018). UJI FUNGSIONALITAS (BLACKBOX TESTING) SISTEM INFORMASI LEMBAGA SERTIFIKASI PROFESI (SILSP) BATIK DENGAN APPPERFECT WEB TEST DAN UJI PENGGUNA. In *Blackbox Texting) Sistem Informasi Lembaga Sertifikasi...* / (Vol. 117, Issue 2).
- Kusnawan, R. (2021). *Implementasi Progressive Web App pada Aplikasi Manajemen Data Dropship*.
- LePage, P., Beaufort, F., & Steomer. Thomas. (2022, September 14). *Add a web app manifest*. <https://web.dev/add-manifest/>
- Majchrzak, T. A., Biørn-Hansen, A., & Grønli, T.-M. (2018). Progressive Web Apps: the Definite Approach to Cross-Platform Development? *Hawaii International Conference on System Sciences*. <http://hdl.handle.net/10125/50607>
- MDN contributors. (2022, September 14). *Progressive web app structure - Progressive web apps (PWAs) / MDN*. https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/App_structure
- Muslim, Moh. (2020). Manajemen Stress Pada Masa Pandemi COVID-19. *ESENSI: Jurnal Manajemen Bisnis*, 23(2).
- Pridmore, S., Turnier-Shea, Y., Morey, R., Fasnacht, M., & Pridmore, W. (2022). Daily Mood Diary Daily Mood Diary (DMD): a proposal for tracking change in subjective mood. *Dynamics of Human Health*. http://journalofhealth.co.nz/?page_id=2691
- Rahmandani, A. (2017). Buku Harian Positif bagi Orang dengan Gangguan Suasana Hati Bipolar : Studi Pendahuluan. *Seminar Nasional Positive Psychology 2016, Surabaya*.
- Rojas, C. (2020). *Building Progressive Web Applications with Vue.js*. Apress. <https://doi.org/10.1007/978-1-4842-5334-2>
- Sheppard, D. (2017). *Beginning Progressive Web App Development*. Apress. <https://doi.org/10.1007/978-1-4842-3090-9>
- Sulistiyorini, P. (2009). Pemodelan Visual dengan Menggunakan UML dan Rational Rose. *Jurnal Teknologi Informasi DINAMIK*, 14, 23–29.

- Susilo, E. (2019). *Cara Menggunakan System Usability Scale (SUS) Pada Evaluasi Usability*.
<https://www.edisusilo.com/cara-menggunakan-system-usability-scale/>
- Tallyfy. (2018). *All You Need to Know About UML Diagrams: Types and 5+ Examples*.
<https://tallyfy.com/uml-diagram/>
- Workneh, T. (2019). *Evaluation of Progressive Web Application to develop an Offline-First Task Management App*. Aalto University.

