

Rancang Bangun Sistem *Skrining* Awal Kesehatan Mental Berbasis Web Menggunakan *LSTM* dan Skala *DASS-21*

PROYEK AKHIR

Disusun Oleh:

Zidan Muhammad Ikvan

3312311122

Disusun untuk memenuhi salah satu syarat memperoleh gelar Ahli Madya Teknik Informatika



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

POLITEKNIK NEGERI BATAM

TAHUN 2026

KATA PENGANTAR

Segala puji syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat, taufik, serta hidayah-Nya sehingga penulis dapat menyelesaikan Proyek Akhir dengan judul “Rancang Bangun Sistem *Skrining* Awal Kesehatan Mental Berbasis Web Menggunakan *LSTM* dan Skala *DASS-21*” dengan baik.

Proyek Akhir ini diajukan untuk memenuhi salah-satu persyaratan akademik dalam menyelesaikan pendidikan program studi Teknik Informatika pada Jurusan Teknik Informatika, Politeknik Negeri Batam. Proyek Akhir ini bertujuan untuk merancang dan mengembangkan sebuah sistem berbasis Web yang dapat membantu proses skrining awal kesehatan mental, khususnya pada aspek stres, depresi, dan kecemasan, dengan memanfaatkan instrumen DASS-21 serta metode *Long Short-Term Memory* (LSTM) untuk analisis teks.

Dalam penyusunan Proyek Akhir ini, penulis memperoleh banyak bimbingan, dukungan, dan bantuan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Kementerian Pendidikan Tinggi, Sains, dan Teknologi.
2. Politeknik Negeri Batam, yang telah menjadi tempat bagi penulis untuk menimba ilmu dan berkembang selama menempuh pendidikan pada Program Studi Diploma III Teknik Informatika.
3. Ahmad Hamim Thohari, S.S.T., M.T. selaku Ketua Jurusan Teknik Informatika Politeknik Negeri Batam.
4. Yeni Rokhayati, S.Si., M.Sc. selaku Ketua Program Studi Teknik Informatika Politeknik Negeri Batam.
5. Muhammad Idris, S.Tr., M.Tr.Kom., selaku Dosen Pembimbing Proyek Akhir sekaligus Dosen Wali, yang telah dengan sabar memberikan bimbingan, arahan, dan motivasi kepada penulis hingga Proyek Akhir ini dapat terselesaikan.
6. Staf Tata Usaha Jurusan Teknik Informatika Politeknik Negeri Batam, yang telah memberikan dukungan penuh dalam pengurusan berbagai keperluan administrasi.

7. Yelia Mesra, selaku ibu penulis, yang selalu memberikan doa, kasih sayang, serta pengorbanan tanpa batas dalam mendukung penulis, baik secara moral maupun material, selama proses pendidikan.

Penulis menyadari bahwa Proyek Akhir ini masih jauh dari sempurna. Oleh karena itu, penulis dengan lapang dada menerima kritik dan saran yang membangun demi kesempurnaan karya ini di masa mendatang. Besar harapan penulis semoga Proyek Akhir ini dapat memberikan manfaat, khususnya dalam pengembangan sistem berbasis kecerdasan buatan di bidang kesehatan mental.

Batam, 9 Mei 2026

Zidan Muhammad Ikvan

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	iii
DAFTAR TABEL	vi
DAFTAR GAMBAR.....	vii
DAFTAR LAMPIRAN	ix
ABSTRAK	x
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah	2
1.3. Tujuan.....	3
1.4. Batasan Masalah.....	3
1.5. Manfaat.....	4
1.6. Kolaborasi dan Sinergi dalam Pelaksanaan Proyek	5
BAB II TINJAUAN PUSTAKA.....	8
2.1. Tinjauan Solusi dan Sistem Terkait.....	8
2.2. Landasan Konsep dan Teknologi	12
2.2.1. Kesehatan Mental.....	12
2.2.2. Stres, Depresi, dan Kecemasan	13
2.2.3. Skala DASS-21 (<i>Depression Anxiety Stress Scale</i>)	13
2.2.4. <i>Long Short-Term Memory</i> (LSTM)	15
2.2.5. Pengembangan Sistem Berbasis Web	17
2.3. Metode Pengembangan Produk.....	17
2.3.1. Definisi Metode Agile.....	17
2.3.2. Alasan Pemilihan Metode Agile	18
2.3.3. Tahapan Pengembangan dengan Metode Agile.....	19
BAB III ANALISIS DAN PERANCANGAN.....	25
3.1. Analisis Kebutuhan	25
3.1.1. Proses Analisis Berjalan	25
3.1.2. Gambaran Umum Sistem.....	27
3.2. Perancangan.....	28

3.2.1.	Kebutuhan Fungsional	29
3.2.2.	Kebutuhan Non-Fungsional	29
3.2.3.	Diagram <i>Use Case</i>	30
3.2.4.	Skenario <i>Use Case</i>	31
3.2.5.	<i>Activity Diagram</i>	42
3.2.6.	ER Diagram.....	49
3.2.7.	Perancangan Antarmuka (<i>Wireframe</i>)	50
3.2.8.	Pengumpulan Data	53
3.2.9.	<i>Preprocessing</i> Data	54
3.2.10.	Pemodelan (LSTM).....	58
3.2.11.	Logika Kalkulasi DASS-21	60
3.2.12.	Analisis Curhat.....	62
3.2.13.	Sistem Rekomendasi Berbasis LLM (Gemini AI).....	63
BAB IV IMPLEMENTASI DAN PENGUJIAN		66
4.1.	Hasil Implementasi API dan <i>Machine Learning</i>	66
4.1.1.	Evaluasi Model.....	66
4.1.2.	Artefak Model untuk Deployment	68
4.1.3.	Model Inference Testing	69
4.1.4.	Integrasi Sistem dan Antarmuka Pemrograman Aplikasi (API).....	71
4.2.	Hasil Implementasi Aplikasi Web (Mentalku-Web).....	74
4.2.1.	Konfigurasi Lingkungan Sistem	74
4.2.2.	Fitur Registrasi	75
4.2.3.	Fitur <i>Login</i>	76
4.2.4.	Fitur <i>Dashboard</i>	76
4.2.5.	Fitur Evaluasi Diri.....	77
4.2.6.	Fitur Profil.....	82
4.2.7.	Fitur Kelola Profil	83
4.2.8.	Fitur Riwayat Aktivitas	84
4.2.9.	Fitur Edukasi	86
4.2.10.	Alur Penggunaan.....	87
4.3.	Pengujian <i>User Acceptance Testing</i> (UAT)	88
BAB V KESIMPULAN		102

5.1. Kesimpulan.....	102
5.2. Saran.....	103
DAFTAR PUSTAKA.....	104
DAFTAR LAMPIRAN	106

DAFTAR TABEL

Tabel 1. Pembagian Peran dan Tanggung Jawab.....	5
Tabel 2. Contoh Tinjauan Aplikasi Sejenis (Pengembangan Perangkat Lunak)	8
Tabel 3. Tabel dan butir DASS-21 (Hakim & Aristawati, 2023)	14
Tabel 4. Timeline Pengembangan Sistem Mentalku (Januari – April 2026).....	22
Tabel 5. <i>Functional Requirement</i>	29
Tabel 6. <i>Non-Functional Requirement</i>	29
Tabel 7. <i>Use Case</i> Scenario Registrasi Akun.....	31
Tabel 8. <i>Use Case</i> Scenario Login	32
Tabel 9. <i>Use Case</i> Scenario Logout	34
Tabel 10. <i>Use Case</i> Scenario Mengelola Profil	35
Tabel 11. <i>Use Case</i> Scenario Mengisi Form Skrining Evaluasi Diri.....	36
Tabel 12. <i>Use Case</i> Scenario Melihat Hasil Skrining Evaluasi Diri	38
Tabel 13. <i>Use Case</i> Scenario Riwayat Skrining Evaluasi Diri	39
Tabel 14. <i>Use Case</i> Scenario Edukasi Kesehatan Mental.....	40
Tabel 15. Pengujian.....	89

DAFTAR GAMBAR

Gambar 1. Metode Agile.....	17
Gambar 2. Gambaran Umum Sistem	28
Gambar 3. <i>Use Case</i>	30
Gambar 4. <i>Activity Diagram</i> Registrasi Akun	42
Gambar 5. <i>Activity Diagram Login</i>	43
Gambar 6. <i>Activity Diagram Logout</i>	44
Gambar 7. <i>Activity Diagram</i> Mengisi <i>Form</i> Skrining Evaluasi Diri & Melihat Hasil Skrining Evaluasi Diri	45
Gambar 8. <i>Activity Diagram</i> Mengelola Profil	46
Gambar 9. <i>Activity Diagram</i> Mengakses Riwayat Skrining Evaluasi Diri.....	47
Gambar 10. <i>Activity Diagram</i> Mengakses Edukasi Kesehatan Mental	48
Gambar 11. <i>Entity Relationship Diagram</i>	49
Gambar 12. Wireframe Halaman Awal <i>Form</i> Skrining Evaluasi diri	50
Gambar 13. Wireframe Kuisisioner <i>Form</i> Skrining Evaluasi Diri	50
Gambar 14. Wireframe Curhat <i>Form</i> Skrining Evaluasi Diri.....	51
Gambar 15. Wireframe Hasil Skrining Evaluasi Diri	51
Gambar 16. Wireframe Informasi Riwayat Skrining Evaluasi Diri dan Profil.....	52
Gambar 17. Wireframe Detail Riwayat Skrining Evaluasi Diri	52
Gambar 18. Wireframe Edukasi Kesehatan Mental.....	53
Gambar 19. Distribusi Label per Dataset	54
Gambar 20. <i>Merge Dataset</i>	55
Gambar 21. <i>Split Dataset</i>	55
Gambar 22. <i>Cleaning Text</i>	56
Gambar 23. <i>Tokenization</i> dan <i>Sequence Padding</i>	57
Gambar 24. <i>Label Encoding</i>	57
Gambar 25. <i>Model Summary</i>	58
Gambar 26. Proses Pelatihan Model	59
Gambar 27. Hasil Pelatihan	60
Gambar 28. Proses Perhitungan Skor Kuisisioner menggunakan instrumen Dass-21.....	61

Gambar 29. Proses Analisis Teks Curhat menggunakan <i>models</i> yang telah dikembangkan	63
Gambar 30. Rekomendasi Personalisasi menggunakan <i>Gemini 3 Flash</i>	64
Gambar 31. Hasil Evaluasi Model	67
Gambar 32. Ekstraksi dan Penyimpanan <i>Models</i>	68
Gambar 33. Proses <i>Model Inference Testing</i>	70
Gambar 34. Output <i>Model Inference Testing</i>	71
Gambar 35. Pengujian Fungsionalitas <i>Endpoint API</i> Evaluasi Diri menggunakan <i>Postman</i>	72
Gambar 36. Tampilan Antarmuka Modal Registrasi	75
Gambar 37. Tampilan Antarmuka Modal <i>Login</i>	76
Gambar 38. Tampilan Antarmuka Halaman Beranda.....	77
Gambar 39. Tampilan Antarmuka Halaman <i>Cover</i> Evaluasi Diri	78
Gambar 40. Tampilan Antarmuka Halaman Kuisioner <i>Dass-21</i>	79
Gambar 41. Tampilan Antarmuka Halaman Curhat	80
Gambar 42. Tampilan Antarmuka Halaman Hasil Evaluasi Diri	81
Gambar 43. Tampilan Antarmuka Halaman Profil	82
Gambar 44. Tampilan Antarmuka Halaman Edit Profil	83
Gambar 45. Tampilan Antarmuka Halaman Riwayat Aktivitas Evaluasi Diri.....	85
Gambar 46. Tampilan Antarmuka Halaman Edukasi	86
Gambar 47. Alur Penggunaan Web <i>Mentalku</i>	87
Gambar 48. <i>User Acceptance Testing</i> bersama Eka Fitri	106
Gambar 49. <i>User Acceptance Testing</i> bersama Masdika.....	107
Gambar 50. <i>User Acceptance Testing</i> bersama Davin.....	107
Gambar 51. <i>User Acceptance Testing</i> bersama Febby.....	107

DAFTAR LAMPIRAN

Lampiran A. Dokumen Proses Pengumpulan <i>Requirement</i>	106
Lampiran B. <i>Link</i> Produk.....	106
Lampiran C. Dokumen Pengujian.....	106

ABSTRAK

Kesehatan mental merupakan aspek penting dalam kualitas hidup individu, namun masih banyak masyarakat yang menghadapi kesulitan dalam mengenali maupun menangani gangguan psikologis. Kondisi ini diperburuk dengan kecenderungan melakukan *self*-diagnosis melalui internet atau media sosial yang tidak selalu akurat, serta rendahnya minat untuk mencari bantuan profesional akibat stigma sosial. Oleh karena itu, diperlukan sebuah sistem yang dapat membantu individu, khususnya dewasa awal, dalam melakukan skrining awal kesehatan mental secara mandiri, terpercaya, dan berbasis ilmiah. Proyek akhir ini bertujuan untuk merancang dan membangun sistem skrining awal kesehatan mental berbasis Web dengan memanfaatkan instrumen DASS-21, model *Long Short-Term Memory* (LSTM), serta integrasi Gemini AI. Pengembangan sistem ini menggunakan metode Agile dengan kerangka kerja Scrum, sehingga proses perancangan, implementasi, dan evaluasi dapat dilakukan secara iteratif dan adaptif sesuai kebutuhan pengguna. Selain itu, sistem ini mengimplementasikan pendekatan skrining hibrida yang unik, yaitu menggabungkan instrumen psikometri terstandar *Depression Anxiety Stress Scales 21 (DASS-21)* dengan model *Deep Learning LSTM* untuk menganalisis sentimen dari narasi teks yang diberikan pengguna. Sistem yang dikembangkan memungkinkan pengguna untuk melakukan registrasi, *Login*, *Logout*, mengelola profil, mengisi form evaluasi diri, serta memperoleh hasil skrining berupa skor kesehatan mental, rekomendasi tindak lanjut yang dipersonalisasi menggunakan Gemini AI, Edukasi terkait, dan riwayat skrining sebelumnya. Hasil implementasi menunjukkan bahwa sistem ini dapat berfungsi sebagai sarana skrining awal yang responsif, ramah pengguna, serta mampu memberikan gambaran kondisi psikologis secara cepat dan informatif tanpa menggantikan peran diagnosis medis dari tenaga profesional. Dengan adanya sistem ini, diharapkan kesadaran masyarakat terhadap pentingnya kesehatan mental semakin meningkat, akses terhadap evaluasi diri menjadi lebih mudah, serta menjadi dasar pengembangan solusi teknologi kesehatan mental yang lebih komprehensif di masa mendatang.

Kata Kunci: kesehatan mental, *DASS-21*, *LSTM*, Agile, skrining awal

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kesehatan mental merupakan salah satu aspek penting dalam kualitas hidup individu, sejajar dengan kesehatan fisik. Gangguan kesehatan mental seperti stres, depresi, dan kecemasan dapat memengaruhi produktivitas, hubungan sosial, serta kesejahteraan secara menyeluruh. Menurut Organisasi Kesehatan Dunia (WHO), sekitar 1 dari 8 orang di dunia hidup dengan gangguan mental, dan prevalensinya semakin meningkat pascapandemi COVID-19 akibat perubahan gaya hidup, isolasi sosial, serta tekanan ekonomi. Kondisi ini menunjukkan bahwa isu kesehatan mental bukanlah masalah individu semata, melainkan tantangan global yang berdampak luas terhadap masyarakat.

Di Indonesia, masalah kesehatan mental juga menjadi perhatian serius. Hasil Indonesia-National Adolescent Mental Health Survey (I-NAMHS) 2022 menunjukkan bahwa sekitar 34,9% remaja mengalami masalah kesehatan mental, namun hanya 2,6% yang mendapatkan layanan profesional. Angka ini menggambarkan adanya kesenjangan besar antara kebutuhan dan ketersediaan layanan kesehatan mental formal. Selain itu, data Riset Kesehatan Dasar (Riskesdas) juga menunjukkan peningkatan kasus depresi dan kecemasan pada usia produktif dalam beberapa tahun terakhir. Kondisi ini semakin menegaskan pentingnya deteksi dini dan upaya preventif untuk menjaga kesehatan mental masyarakat Indonesia.

Kelompok dewasa awal (20–40 tahun) merupakan salah satu kelompok yang paling rentan. Pada fase ini, individu menghadapi berbagai tantangan seperti tuntutan akademik, kompetisi di dunia kerja, tanggung jawab keluarga, hingga tekanan sosial. Beban tersebut seringkali memicu stres berkepanjangan, rasa cemas, bahkan gejala depresi. Studi psikologi perkembangan menunjukkan bahwa dewasa awal adalah fase transisi kritis yang menentukan pola hidup jangka panjang, sehingga gangguan mental pada fase ini dapat berdampak serius pada kualitas hidup, keberhasilan karier, dan relasi sosial. Sayangnya, masih banyak masyarakat

Indonesia yang enggan mencari bantuan psikologis karena adanya stigma negatif, rasa malu, atau anggapan bahwa masalah psikologis hanyalah hal wajar yang bisa diatasi sendiri tanpa intervensi profesional.

Kondisi tersebut membuat sebagian orang lebih memilih melakukan *self*-diagnosis dengan mencari informasi melalui internet atau media sosial. Walaupun akses informasi yang luas dapat membantu, cara ini berisiko menimbulkan kesalahpahaman atau bahkan memperburuk kondisi psikologis individu. Banyak kasus menunjukkan bahwa gejala awal yang diabaikan atau ditangani secara keliru dapat berkembang menjadi gangguan mental yang lebih serius, bahkan berujung pada perilaku menyakiti diri sendiri atau bunuh diri. Hal ini menegaskan perlunya sistem skrining awal yang aman, terpercaya, dan berbasis bukti ilmiah untuk membantu individu memahami kondisi mereka sejak dini.

Pemerintah Indonesia juga semakin memberi perhatian terhadap isu kesehatan mental. Kementerian Kesehatan telah meluncurkan berbagai program, seperti Gerakan Nasional Peduli Kesehatan Jiwa (GERMAS Jiwa) dan integrasi layanan kesehatan mental dalam program Indonesia Sehat dengan Pendekatan Keluarga (PIS-PK). Selain itu, kampanye nasional mengenai pentingnya deteksi dini kesehatan mental juga terus digalakkan, khususnya untuk remaja dan dewasa muda. Kehadiran platform digital yang dapat membantu skrining awal seperti ini sejalan dengan upaya pemerintah dalam meningkatkan literasi kesehatan mental, memperluas akses layanan, serta mendorong masyarakat agar lebih peduli terhadap kondisi psikologisnya.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan sebelumnya, maka dapat dirumuskan beberapa permasalahan utama yang menjadi fokus dalam proyek ini, yaitu:

1. Bagaimana merancang solusi digital berbasis Web untuk skrining awal kesehatan mental pada dewasa awal di Indonesia, yang aman, privat, dan terarah, sehingga dapat menjadi alternatif yang lebih baik dibandingkan praktik *self*-diagnosis yang berisiko?

2. Bagaimana mengembangkan sistem skrining awal yang valid dan terukur dengan mengintegrasikan instrumen psikologis DASS-21 serta inovasi analisis teks berbasis LSTM, sehingga hasil evaluasi lebih komprehensif dan sesuai dengan kondisi nyata pengguna?
3. Bagaimana mengimplementasikan sistem skrining awal kesehatan mental dalam bentuk platform Web yang responsif, mudah digunakan di berbagai perangkat (smartphone maupun laptop), serta dapat diakses secara luas oleh masyarakat Indonesia?

1.3. Tujuan

Tujuan dari sistem ini adalah:

1. Merancang dan membangun sistem skrining awal kesehatan mental berbasis Web yang membantu individu dewasa awal melakukan evaluasi mandiri secara tepat, terpercaya, dan mengurangi kecenderungan *self*-diagnosis yang tidak akurat.
2. Mengembangkan metode skrining yang menggabungkan DASS-21 dengan model LSTM sebagai inovasi utama untuk menghasilkan evaluasi kesehatan mental yang lebih akurat, terukur, dan informatif.
3. Membangun platform Web skrining kesehatan mental berbasis DASS-21 dan LSTM yang responsif, *user-friendly*, dan inklusif, sehingga dapat digunakan oleh masyarakat maupun mahasiswa sebagai sarana skrining awal kesehatan mental.

1.4. Batasan Masalah

Untuk menjaga agar pengembangan sistem ini tetap terarah dan sesuai dengan tujuan, maka ruang lingkup proyek akhir dibatasi pada hal-hal berikut:

1. Lingkup Deteksi: Sistem hanya mencakup tiga aspek kesehatan mental, yaitu depresi, kecemasan, dan stres.
2. Tujuan Sistem: Sistem yang dikembangkan ditujukan sebagai skrining awal (*early screening*), bukan diagnosis medis. Hasil yang diberikan hanya bersifat informatif untuk meningkatkan kesadaran pengguna, serta mendorong langkah preventif dan proaktif terkait kesehatan mental.
3. Metode dan Data:

- Data utama berasal dari kuesioner DASS-21 yang diisi pengguna untuk menghasilkan skor depresi, kecemasan, dan stres.
 - Analisis tambahan menggunakan data teks bebas (curhat) yang diproses dengan model *Long Short-Term Memory* (LSTM) guna mengidentifikasi kecenderungan emosi pengguna.
 - Dataset untuk pelatihan model LSTM diperoleh dari sumber terbuka seperti Kaggle dan HuggingFace, sehingga hasil model sangat dipengaruhi oleh kualitas dan distribusi data tersebut.
4. Integrasi Layanan: Sistem belum terhubung dengan tenaga profesional seperti psikolog atau konselor, dan juga belum menyediakan forum diskusi pengguna.

1.5. Manfaat

Sistem ini diharapkan dapat memberikan manfaat bagi berbagai pihak, di antaranya sebagai berikut:

1. Memberikan kontribusi nyata terhadap pencapaian **Tujuan Pembangunan Berkelanjutan atau *Sustainable Development Goals* (SDGs) poin ke-3 mengenai Kehidupan Sehat dan Sejahtera**, khususnya pada target 3.4 dalam mempromosikan kesehatan mental dan kesejahteraan melalui penyediaan platform identifikasi dini yang mandiri, objektif, dan ilmiah menggunakan instrumen DASS-21.
2. Memberikan edukasi yang dipersonalisasi kepada pengguna melalui integrasi sistem berbasis *Generative AI* (Gemini), sehingga pengguna mendapatkan pemahaman yang tepat mengenai kondisi kesehatan mentalnya.
3. Menyediakan media bagi pengguna untuk mengekspresikan emosi secara bebas dan aman melalui teks "curhatan" yang dianalisis secara otomatis menggunakan teknologi *Machine Learning* berbasis LSTM.
4. Mengurangi risiko kesalahan *self-diagnosis* di masyarakat dengan menyediakan alat skrining yang memiliki landasan klinis dan analisis data yang akurat.

5. Menjadi jembatan awal bagi individu yang merasa ragu untuk mencari bantuan profesional, sehingga meningkatkan kesadaran untuk melakukan konsultasi medis lebih lanjut jika diperlukan.
6. Mendukung akselerasi **SDGs poin ke-9 terkait Industri, Inovasi, dan Infrastruktur** melalui pemanfaatan inovasi teknologi *Health-Tech* berbasis kecerdasan buatan (*Artificial Intelligence*) guna menciptakan infrastruktur layanan kesehatan mental digital yang inklusif, terjangkau, dan mudah diakses oleh kelompok masyarakat maupun mahasiswa.

1.6. Kolaborasi dan Sinergi dalam Pelaksanaan Proyek

Meskipun proyek ini merupakan tugas mandiri sebagai bagian dari penyelesaian Tugas Akhir, pelaksanaannya tetap melibatkan kolaborasi dan sinergi dengan berbagai pihak guna memastikan sistem yang dikembangkan memenuhi standar akademis dan kebutuhan fungsional pengguna. Sinergi ini mencakup arahan strategis dari pembimbing, pengembangan teknis secara menyeluruh oleh peneliti, hingga tahap pengujian oleh pengguna.

Rincian kolaborasi dan tanggung jawab setiap pihak yang terlibat dalam pengembangan sistem Mentalku disajikan dalam Tabel 1 berikut:

Tabel 1. Pembagian Peran dan Tanggung Jawab

No	Nama	Role	Ruang Lingkup Tanggung Jawab
1.	Zidan Muhammad Ikvan	<i>Developer</i>	- Melakukan riset kebutuhan sistem, analisis fungsional, serta perancangan basis data secara menyeluruh. - Melakukan implementasi kode program (Frontend & Backend) serta integrasi teknologi kecerdasan buatan ke dalam infrastruktur aplikasi. - Melakukan manajemen data dan pengolahan dataset

			(<i>Preprocessing</i>) untuk keperluan pelatihan, pengujian, serta evaluasi performa model <i>Machine Learning</i>. • Menyusun dan menyelesaikan laporan penelitian Proyek Akhir secara sistematis serta memastikan keselarasan antara implementasi teknis dengan substansi karya tulis ilmiah. • Melaksanakan seluruh tahapan pengerjaan Rancang Bangun Sistem Skrining Awal Kesehatan Mental Berbasis Web Menggunakan LSTM dan Skala DASS-21 (Mentalku) secara end-to-end.
2.	Muhammad Idris, S.Tr., M.Tr.Kom	<i>Project Manager</i>	• Memberikan arahan strategis dan bimbingan teknis mulai dari tahapan analisis kebutuhan, perancangan, pengembangan, hingga integrasi sistem. • Melakukan pengawasan intensif terhadap milestone dan progres pengerjaan proyek untuk memastikan metode pengembangan berjalan sesuai dengan standar akademis. • Memberikan saran, kritik, serta evaluasi menyeluruh terhadap aspek fungsionalitas sistem maupun substansi laporan proyek.

			Menjamin kualitas luaran proyek dengan melakukan identifikasi potensi risiko teknis guna menjaga stabilitas dan validitas pengembangan sistem secara keseluruhan.
3.	Masdika	<i>UAT Participant</i>	Melakukan pengujian fungsionalitas sistem (UAT) dan memberikan umpan balik terkait pengalaman pengguna.
4.	Febby	<i>UAT Participant</i>	Melakukan pengujian fungsionalitas sistem (UAT) dan memberikan umpan balik terkait pengalaman pengguna.
5.	Eka Putri	<i>UAT Participant</i>	Melakukan pengujian fungsionalitas sistem (UAT) dan memberikan umpan balik terkait pengalaman pengguna.
6.	Davin	<i>UAT Participant</i>	Melakukan pengujian fungsionalitas sistem (UAT) dan memberikan umpan balik terkait pengalaman pengguna.

BAB II TINJAUAN PUSTAKA

2.1. Tinjauan Solusi dan Sistem Terkait

Untuk Proyek Pengembangan Perangkat Lunak, kajian difokuskan pada aplikasi-aplikasi sejenis yang telah ada. Contoh penyajian seperti Tabel 2.

Tabel 2. Contoh Tinjauan Aplikasi Sejenis (Pengembangan Perangkat Lunak)

Pengembang/ Tahun	Nama Aplikasi	Fitur Utama	Teknologi	Kelebihan	Keterbatasan
Sutanto, Dadang, Amiruddin, & Gagan Nugraha (2022)	Rancang Bangun Aplikasi Skrining Keseha- tan Mental Remaja Berbasis Web di RSUD Dr. Dradjat Prawira- negara dengan Menggun- akan Metode Analytic Hierar- chy	Laporan Kuisio- ner	MySQL, PHP	Mengelua- rkan hasil secara real time dan Memberi- kan informasi tentang kesehatan jiwa.	Ruang Lingkup yang Sangat Terbatas, Validitas Instrumen yang Dipertanya- kan, Hasil Skrining yang Kurang Informatif, Tidak Adanya Fitur Pemantauan (Riwayat).

	Process (AHP)				
Hafidz, Fais Al Huda, & Agi Putra Kharisma (2024)	Pengem- bangan Aplikasi Edukasi Keseha- tan Mental Berbasis Progre- ssive Web App	Materi dan kuis menge- nai keseha- tan mental, mana- jemen materi keseha- tan mental, konsul- tasi peng- guna.	Laravel, MySQL, Next.js , REST API	Proggre- sive Web Apps, fungsiona- litas menun- jukkan tingkat keberhasi- lan 100%.	Fitur konsultasi masih pakai sistem chat biasa, belum real time.
Tarisa Fitriani Hamenda, Sumiyati Moo, Ramdan Hunowu, & Ns. Yuniar M. Soeli, M.Kep.,Sp. Kep.J (2022)	Pengem- bangan Deteksi Dini Ganggu- an Keseha- tan Jiwa Dengan Mental Health	deteksi dini masalah Keseha- tan Jiwa berda- sarkan 7 klasifi- kasi diagno- sa gang-	Android	Aplikasi android Mental Health Screening dapat digunakan secara Offline atau tanpa perlu	Ruang Lingkup yang Sangat Terbatas, Validitas Instrumen yang Dipertanya- kan, Tidak Adanya Fitur

	Screening Berbasis Aplikasi Android Untuk Masyarakat Pesisir Pantai Desa Ponelo	guan jiwa dalam kepe-rawatan		menggunakan jaringan internet untuk menjalankannya.	Pemantauan (Riwayat).
Antonius Sigid Priharsanto, & Abdul Azzam Ajhari (2024)	Pengembangan Aplikasi Kesehatan Mental Berbasis Kotlin: Studi Kasus Aplikasi HugMe	Analisis catatan pengguna Berbasis AI, forum diskusi anonim	Kotlin, Long Short Term Memory (LSTM)	Analisis catatan pengguna Berbasis AI (LSTM), forum diskusi anonim, rekomendasi tempat healing.	Mengharuskan pengguna untuk mengunduh dan menginstal aplikasi, Banyak fitur yang belum selesai, Validitas Ilmiah yang Rendah, Fungsionalitas analisis berbasis <i>Deep Learning</i> belum terintegrasi sepenuhnya

					ke dalam aplikasi.
Proyek ini	Rancang Bangun Sistem Skrining Awal Kesehatan Mental Berbasis Web Menggunakan LSTM dan Skala DASS-21	Evaluasi Diri & Edukasi	Laravel, Python, Fast API, & LSTM.	Akses layanan gratis, Tampilan antarmuka (UI/UX) yang menarik dan ramah pengguna, Pengolahan hasil gabungan dari DASS-21 dan LSTM, Sumber data yang jelas dan valid, Mudah diakses melalui handphone atau laptop, rekomendasi tindak lanjut	Sistem belum terhubung dengan tenaga profesional seperti psikolog atau konselor, dan juga belum menyediakan forum diskusi pengguna. Ketergantungan pada koneksi internet.

				menggunakan Gemini AI.	
--	--	--	--	------------------------	--

2.2. Landasan Konsep dan Teknologi

Landasan Konsep dan Teknologi merupakan kerangka dasar yang digunakan untuk mendukung seluruh proses penelitian dan pembangunan sistem. Pada bagian ini, akan diuraikan teori-teori serta teknologi yang relevan dengan pengembangan "Rancang Bangun Sistem Skrining Awal Kesehatan Mental Berbasis Web Menggunakan LSTM dan Skala DASS-21". Pembahasan ini mencakup dua pilar utama, yaitu aspek konseptual terkait kesehatan mental dan psikometri, serta aspek teknis terkait infrastruktur teknologi yang diimplementasikan. Integrasi antara landasan konsep dan teknologi ini diharapkan dapat memberikan pemahaman yang komprehensif, sehingga setiap tahapan perancangan hingga implementasi sistem memiliki pijakan ilmiah yang kuat serta validitas teknis yang dapat dipertanggungjawabkan.

2.2.1. Kesehatan Mental

Kesehatan mental merupakan kondisi kesejahteraan individu yang memungkinkan seseorang untuk menyadari potensi dirinya, mengatasi tekanan hidup sehari-hari, bekerja secara produktif, serta mampu berkontribusi pada komunitas. Menurut (World Health, 2022), kesehatan mental bukan sekadar ketiadaan gangguan jiwa, melainkan kondisi yang optimal dari aspek emosional, psikologis, dan sosial. Pada dewasa awal (20–40 tahun), tekanan hidup seperti tuntutan pekerjaan, hubungan sosial, maupun tanggung jawab keluarga seringkali memicu stres, kecemasan, dan depresi. Dilihat dari aspek perkembangan biologis dan sosial, usia dewasa awal (*emerging adults*) atau rentang usia 20-40 tahun adalah usia yang paling rentan mengalami depresi, kecemasan, dan stres (DARIYO, 2003). Oleh karena itu, diperlukan mekanisme skrining awal untuk mendeteksi gejala-gejala tersebut agar dapat diambil langkah preventif.

2.2.2. Stres, Depresi, dan Kecemasan

Stres, depresi, dan kecemasan termasuk isu kesehatan mental yang paling banyak ditemui, khususnya pada kelompok dewasa awal. Hal ini sejalan dengan data global dari Organisasi Kesehatan Dunia (World Health, 2022) yang mengidentifikasi depresi dan kecemasan sebagai gangguan mental paling umum.

- Stres adalah respon tubuh terhadap tekanan atau tuntutan tertentu.
- Kecemasan adalah rasa khawatir berlebihan yang disertai gejala fisik maupun psikologis.
- Depresi merupakan gangguan suasana hati yang ditandai rasa sedih mendalam, hilang minat, dan menurunnya produktivitas.

Ketiganya tidak selalu berarti gangguan kejiwaan berat, tetapi jika dibiarkan berlarut dapat memengaruhi kualitas hidup dan membutuhkan penanganan lebih lanjut.

2.2.3. Skala DASS-21 (*Depression Anxiety Stress Scale*)

DASS-21 (*Depression Anxiety Stress Scales 21*) adalah sebuah instrumen psikometri yang digunakan untuk mengukur tingkat keparahan gejala depresi, kecemasan, dan stres. Skala ini terdiri dari 21 pernyataan yang dibagi rata ke dalam tiga aspek (7 butir untuk depresi, 7 butir untuk kecemasan, dan 7 butir untuk stres) seperti yang ditunjukkan pada tabel 3. Instrumen ini banyak digunakan dalam penelitian maupun skrining awal karena relatif singkat dan mudah diisi. Kredibilitasnya telah teruji secara luas, dan secara khusus untuk konteks Indonesia, sebuah penelitian oleh (Hakim & Aristawati, 2023) telah membuktikan bahwa DASS-21 versi Bahasa Indonesia memiliki validitas dan reliabilitas konstruk yang baik untuk digunakan pada kelompok dewasa awal. Temuan ini menjadi landasan kuat yang memvalidasi pemilihan DASS-21 sebagai metode utama untuk kuantifikasi gejala psikologis dalam sistem yang dikembangkan.

Tabel 3. Tabel dan butir DASS-21 (Hakim & Aristawati, 2023)

Dimensi	Nomor Butir	Butir
Depresi	D1	Saya sama sekali tidak dapat merasakan perasaan positif (contoh: merasa gembira, bangga, dsb).
	D2	Saya merasa sulit berinisiatif melakukan sesuatu.
	D3	Saya merasa tidak ada lagi yang bisa saya harapkan.
	D4	Saya merasa sedih dan tertekan.
	D5	Saya tidak bisa merasa antusias terhadap hal apapun.
	D6	Saya merasa diri saya tidak berharga.
	D7	Saya merasa hidup ini tidak berarti.
Kecemasan	A1	Saya merasa rongga mulut saya kering.
	A2	Saya merasa kesulitan bernafas (misalnya seringkali terengah-engah atau tidak dapat bernapas padahal tidak melakukan aktivitas fisik sebelumnya).
	A3	Saya merasa gemetar (misalnya pada tangan).
	A4	Saya merasa khawatir dengan situasi dimana saya mungkin menjadi panik dan mempermalukan diri sendiri.
	A5	Saya merasa hampir panik.
	A6	Saya menyadari kondisi jantung saya (seperti meningkatnya atau melemahnya detak jantung) meskipun sedang tidak melakukan aktivitas fisik.
	A7	Saya merasa ketakutan tanpa alasan yang jelas.
Stres	S1	Saya merasa sulit untuk beristirahat.
	S2	Saya cenderung menunjukkan reaksi berlebihan terhadap suatu situasi.
	S3	Saya merasa energi saya terkuras karena terlalu cemas.
	S4	Saya merasa gelisah.
	S5	Saya merasa sulit untuk merasa tenang.

	S6	Saya sulit untuk bersabar dalam menghadapi gangguan yang terjadi ketika sedang melakukan sesuatu.
	S7	Perasaan saya mudah tergugah atau tersentuh.

Proses penilaian skor mengikuti standar yang ada, di mana setiap item dinilai pada skala 0 ("Tidak pernah") hingga 3 ("Sangat sering atau hampir selalu"). Total skor mentah untuk setiap sub-skala dihitung dengan menjumlahkan skor dari 7 item terkait, lalu dikalikan dua untuk menyetarakan hasilnya dengan versi asli DASS-42. Hasil akhir kemudian dibandingkan dengan tabel standar untuk menentukan tingkat keparahan, yang diklasifikasikan menjadi Normal, Ringan, Sedang, Parah, dan Sangat Parah (Hakim & Aristawati, 2023). Sistem ini akan mengotomatisasi keseluruhan proses perhitungan dan interpretasi tersebut untuk memberikan hasil yang cepat dan akurat kepada pengguna.

2.2.4. *Long Short-Term Memory (LSTM)*

Long Short-Term Memory (LSTM) merupakan salah satu arsitektur *Recurrent Neural Network (RNN)* yang canggih, dirancang secara spesifik untuk mengatasi kelemahan fundamental RNN dalam mengingat dependensi jangka panjang atau yang dikenal dengan permasalahan *vanishing gradient* (GeeksforGeeks, 2025). Untuk mencapai hal ini, LSTM tidak hanya memiliki alur perulangan biasa, tetapi juga dilengkapi struktur internal berupa sel memori (*memory cell*) yang dapat menyimpan, membaca, dan menulis informasi dalam rentang waktu yang lebih lama.

Mekanisme pengelolaan informasi ini diatur oleh tiga komponen utama yang disebut gerbang (*gates*), yang secara cerdas mengontrol aliran data. Seperti yang dijelaskan oleh (Olah, 2015), ketiga gerbang tersebut adalah:

- *Forget Gate*, yang secara selektif menentukan informasi mana dari status sel sebelumnya yang tidak lagi relevan dan harus dibuang.
- *Input Gate*, yang mengatur informasi baru mana yang dianggap penting untuk diperbarui dan disimpan ke dalam sel memori.

- *Output Gate*, yang mengontrol informasi apa dari status sel saat ini yang akan dikeluarkan sebagai output, berdasarkan konteks yang telah dipelajari.

Dengan mekanisme ini, LSTM terbukti efektif dalam memproses data sekuensial seperti teks, karena mampu menangkap hubungan antar kata dalam kalimat panjang serta memahami konteks secara utuh.

Pemilihan arsitektur LSTM untuk proyek ini didasarkan pada kemampuannya untuk melakukan analisis yang lebih mendalam dibandingkan metode analisis teks sederhana. Saat pengguna menuliskan curhatan atau narasi bebas, makna dan sentimen yang terkandung tidak hanya ditentukan oleh kata-kata individual, tetapi oleh urutan, struktur kalimat, dan konteks yang melingkupinya. Kemampuan LSTM untuk mengenali pola emosi dari teks menjadikannya sangat relevan untuk mendeteksi indikasi masalah psikologis dengan beberapa alasan utama:

1. Pemahaman Kontekstual: Berbeda dengan metode yang hanya mendeteksi kata kunci (misalnya, "sedih" atau "cemas"), LSTM mampu memahami nuansa. Contohnya, LSTM dapat membedakan antara kalimat "saya tidak sedih" dengan "saya sangat sedih", meskipun keduanya mengandung kata kunci yang sama.
2. Menangkap Dependensi Jangka Panjang: Dalam sebuah narasi, emosi inti mungkin baru terungkap di akhir kalimat atau paragraf. LSTM dapat mengingat informasi dari awal teks untuk menginterpretasikan bagian akhir secara akurat.
3. Memberikan Wawasan Kualitatif: Kuesioner DASS-21 memberikan data kuantitatif ("seberapa sering"). LSTM memberikan data kualitatif ("bagaimana perasaan itu dideskripsikan"), yang memberikan lapisan pemahaman baru yang tidak bisa ditangkap oleh skala angka.

Hasil analisis LSTM dalam sistem ini akan diarahkan pada tiga kategori utama yang selaras dengan DASS-21, yaitu stres, kecemasan, dan depresi. Dengan demikian, model LSTM berfungsi sebagai pelengkap yang kuat untuk

hasil kuesioner DASS-21, sehingga proses skrining kondisi mental pengguna menjadi lebih komprehensif, akurat, dan informatif.

2.2.5. Pengembangan Sistem Berbasis Web

Sistem ini dikembangkan dalam bentuk aplikasi berbasis Web agar dapat diakses secara luas melalui laptop maupun perangkat seluler. *Framework* Laravel (PHP) digunakan untuk pengembangan sisi back-end, sedangkan Tailwind CSS dipakai untuk membangun antarmuka yang responsif, modern, dan ramah pengguna. MySQL digunakan sebagai basis data. Model *Long Short-Term Memory* (LSTM) dan logika kalkulasi DASS-21 diintegrasikan melalui API berbasis Python, sedangkan rekomendasi tambahan diperoleh dari API Gemini.

2.3. Metode Pengembangan Produk

2.3.1. Definisi Metode Agile



Gambar 1. Metode Agile

Agile adalah pendekatan pengembangan perangkat lunak yang menekankan iterasi singkat, kolaborasi, adaptasi terhadap perubahan, dan kepuasan pengguna. Menurut (Alsaqqa, Sawalha, & Abdel-Nabi, 2020), Agile adalah pendekatan pengembangan perangkat lunak yang menekankan iterasi pendek, kolaborasi tim, adaptasi terhadap perubahan kebutuhan, dan pengiriman produk secara bertahap. Sementara menurut (Schwaber & Sutherland, 2020), Scrum adalah kerangka kerja Agile yang menyediakan

struktur berupa event Sprint, *Sprint Planning*, *Daily Scrum*, *Sprint Review*, dan *Sprint Retrospective* sebagai mekanisme untuk inspeksi dan adaptasi. Agile memecah pengembangan ke dalam siklus pendek yang disebut iterasi atau sprint, di mana setiap sprint menghasilkan bagian kecil dari produk yang dapat diuji dan dievaluasi.

Berbeda dengan metode tradisional seperti *Waterfall* yang linear dan kaku, Agile memberikan fleksibilitas tinggi dalam meninjau kembali prioritas dan arah pengembangan di setiap akhir sprint. Dalam penelitian ini, Agile diadopsi dengan kerangka kerja Scrum sebagai panduan utama.

2.3.2. Alasan Pemilihan Metode Agile

Metode pengembangan produk yang digunakan dalam penelitian ini adalah Agile, yang dipilih karena sifatnya yang fleksibel, iteratif, dan adaptif terhadap perubahan kebutuhan. Dalam konteks pengembangan Rancang Bangun Sistem Skrining Awal Kesehatan Mental Berbasis Web Menggunakan LSTM dan Skala DASS-21, metode ini memberikan ruang untuk evaluasi berkelanjutan serta penyempurnaan produk secara bertahap. Pemilihan Agile (dengan adaptasi Scrum) didasarkan pada karakteristik proyek:

1. Kompleksitas Ganda (Web & AI)

Proyek ini melibatkan dua komponen utama yang berbeda:

- Pengembangan aplikasi Web (Laravel + Tailwind CSS).
- Pengembangan model *Deep Learning* LSTM (Python).

Agile mendukung pengembangan paralel pada kedua domain, sehingga progres salah satu bagian tidak harus menunggu bagian lainnya selesai sepenuhnya.

2. Sifat Eksperimental Model AI

Pengembangan model LSTM bersifat eksperimental dan berulang: *training* → evaluasi → *tuning* → *retraining*. Pendekatan iteratif Agile sangat sesuai untuk siklus ini karena memungkinkan perbaikan model secara bertahap di setiap sprint.

3. Fokus pada Pengalaman Pengguna (UI/UX)

Salah satu tujuan sistem ini adalah antarmuka harus menarik, ramah pengguna, dan informatif. Agile memungkinkan penulis untuk membangun prototipe fitur, mendapatkan masukan (misalnya dari dosen pembimbing atau rekan), dan langsung memperbaikinya pada iterasi berikutnya, memastikan hasil akhir benar-benar nyaman digunakan.

4. Adaptasi Kebutuhan Fitur

Selama pengembangan, mungkin akan muncul ide-ide baru untuk meningkatkan fungsionalitas sistem. Agile memudahkan penambahan atau modifikasi fitur seperti ini tanpa mengganggu seluruh rencana proyek.

2.3.3. Tahapan Pengembangan dengan Metode Agile

Secara umum, tahapan pengembangan produk dengan pendekatan Agile dapat dijelaskan melalui enam fase utama, yaitu:

1. **Requirements:** Mengidentifikasi serta merumuskan kebutuhan sistem, baik dari sisi fungsional maupun non-fungsional.
2. **Design:** Menyusun rancangan arsitektur sistem, alur proses, serta rancangan antarmuka pengguna yang sesuai dengan kebutuhan.
3. **Development:** Mengimplementasikan fitur-fitur yang telah dirancang dengan menggunakan teknologi yang dipilih.
4. **Testing:** Melakukan pengujian terhadap sistem untuk memastikan setiap fitur berjalan sesuai dengan kriteria yang telah ditentukan.
5. **Deployment:** Mengimplementasikan sistem pada server yang telah disiapkan agar dapat diakses oleh pengguna.
6. **Review:** Mengevaluasi kinerja sistem, mengumpulkan umpan balik, serta merencanakan perbaikan pada iterasi selanjutnya.

Tahapan pengembangan produk mengikuti kerangka kerja Scrum yang disesuaikan dengan proyek individu:

1. Perencanaan (*Planning*)

- Visi Produk: Menetapkan tujuan utama sistem skrining kesehatan mental berbasis Web.
- *Product backlog* (Daftar Fitur Utama):
 - Registrasi & *Login* Pengguna
 - Pengelolaan Profil
 - Form Evaluasi Diri (Skala DASS-21 & input teks curhat)
 - Halaman Hasil Skrining (Perhitungan skor, rekomendasi AI via Gemini, Edukasi)
 - Riwayat Skrining Pengguna
 - Edukasi Kesehatan Mental

2. Iterasi / *Sprint*

Secara teori, iterasi atau *sprint* adalah sebuah siklus kerja dengan durasi tetap (*time-boxed*), yang umumnya berlangsung antara satu hingga empat minggu. Pada fase ini, tim pengembang bekerja untuk menyelesaikan sejumlah item dari *product backlog* dan menghasilkan potongan perangkat lunak yang fungsional (Fowler, 2019). Setiap *sprint* meliputi:

- *Sprint Planning*: Memilih fitur dari backlog untuk dikerjakan (misalnya *Sprint 1*: registrasi/*Login*, *Sprint 2*: form DASS-21). Tahapan ini selaras dengan penjelasan dalam Scrum Guide (Schwaber & Sutherland, 2020), yang menyatakan bahwa *Sprint Planning* adalah proses di mana tim memilih item-item dari *product backlog* untuk diselesaikan dalam *sprint*.
- Pengembangan & Implementasi: Coding frontend (Tailwind), backend (Laravel), serta model AI (Python, LSTM).
- *Testing*: Memastikan fitur yang baru dibuat berjalan dengan baik, bebas dari bug, dan sesuai dengan kebutuhan. Menguji secara menyeluruh menggunakan *Black-box Testing* untuk memastikan semua fitur pada aplikasi dapat dijalankan sesuai dengan kriteria yang telah ditentukan.

- Tinjauan Sprint (*Sprint Review*): Di akhir setiap sprint, hasil kerja yang sudah menjadi bagian fungsional dari aplikasi akan ditinjau. Hal ini sesuai dengan penjelasan dalam Scrum Guide (Schwaber & Sutherland, 2020), bahwa *Sprint Review* bertujuan untuk memeriksa hasil sprint dan mendapatkan umpan balik langsung dari stakeholder. Tahap ini penting untuk mendapatkan masukan dan memastikan proyek berjalan di jalur yang benar.
- *Sprint Retrospective*: Mengevaluasi proses pengembangan sprint sebelumnya untuk meningkatkan efektivitas sprint berikutnya.

3. Timeline dan Aktivitas Sprint

Sebelum siklus sprint dimulai, proyek Mentalku diawali dengan tahap pengumpulan kebutuhan dan analisis sistem sebagai fondasi penyusunan *Product backlog*. Kebutuhan sistem diidentifikasi melalui studi literatur terkait kesehatan mental, analisis terhadap solusi sejenis yang telah ada, serta diskusi dan konfirmasi kebutuhan bersama dosen pembimbing. Proses ini menghasilkan daftar fitur yang diprioritaskan berdasarkan nilai manfaat bagi pengguna dan kompleksitas implementasinya. *Product backlog* yang dihasilkan bersifat dinamis, sehingga prioritas fitur dapat disesuaikan kembali di setiap awal sprint seiring dengan berkembangnya pemahaman terhadap kebutuhan sistem.

Pengembangan inti sistem dilaksanakan dalam empat Sprint yang terjadwal dari Januari hingga April 2026. Setiap Sprint memiliki durasi empat minggu dan menghasilkan Sprint Goal berupa luaran fungsional yang dapat diverifikasi. Timeline lengkap dan rincian aktivitas tiap Sprint disajikan pada tabel 4 berikut.

Tabel 4. Timeline Pengembangan Sistem Mentalku (Januari – April 2026)

Sprint	Periode	Aktivitas Utama	Teknologi	Sprint Goal
Pra-sprint (Analisis) Kebutuhan	September- Oktober 2025	Studi literatur kesehatan mental, analisis solusi sejenis, identifikasi kebutuhan sistem, diskusi dan konfirmasi kebutuhan bersama dosen pembimbing, penyusunan <i>Product backlog</i> awal	-	<i>Product backlog</i> tervalidasi berbasis data riset
Sprint 1	Januari 2026	Finalisasi <i>Product backlog</i> berbasis data riset; perancangan arsitektur sistem (ERD, <i>Use Case</i> Diagram, <i>Wireframe</i>); setup environment Laravel & FastAPI; konfigurasi repositori dan struktur proyek	Laravel, FastAPI, Git	Dokumen arsitektur sistem final, Backlog terprioritas, lingkungan pengembangan siap digunakan
Sprint 2	Februari 2026	Pengembangan inti aplikasi: autentikasi pengguna (registrasi/ <i>Login</i>), form evaluasi DASS-21, pengumpulan data latih model LSTM, pelatihan	Laravel, Python, LSTM, FastAPI	Modul DASS-21 berjalan fungsional, model LSTM terlatih dengan akurasi awal

		model awal, dan ekspos endpoint prediksi via FastAPI		terukur, endpoint prediksi aktif
Sprint 3	Maret 2026	Integrasi Gemini API sebagai mesin rekomendasi berbasis AI; penyempurnaan desain prompt untuk menghasilkan respons yang empatik dan kontekstual; pengembangan modul edukasi kesehatan mental; integrasi seluruh komponen sistem	Gemini API, Laravel, FastAPI	Sistem terintegrasi penuh dengan rekomendasi AI yang empatik dan modul edukasi aktif
Sprint 4	April 2026	User Acceptance <i>Testing</i> (UAT) dengan pengguna nyata; identifikasi dan perbaikan bug; pengujian Black-box <i>Testing</i> menyeluruh; <i>Sprint Review</i> final bersama dosen pembimbing; finalisasi dokumentasi laporan	Laravel, LSTM, Gemini API, FastAPI	Sistem Mentalku final, bebas bug, lulus UAT, siap didemonstrasikan pada sidang Proyek Akhir

4. Product Release

Setelah melalui beberapa siklus sprint dan semua fitur inti telah selesai dikembangkan, diuji, dan terintegrasi dengan baik, produk akhir akan siap untuk dirilis. Dalam konteks Proyek Akhir, "rilis" berarti sistem telah siap sepenuhnya untuk didemonstrasikan dan dievaluasi pada sidang akhir.

Dengan menggunakan metode Agile, proses pengembangan sistem ini diharapkan menjadi lebih terstruktur, adaptif, dan mampu menghasilkan produk akhir yang optimal dan sesuai dengan tujuan yang telah ditetapkan.

BAB III

ANALISIS DAN PERANCANGAN

3.1. Analisis Kebutuhan

Analisis kebutuhan merupakan tahapan fundamental yang dilakukan untuk memahami kondisi nyata terkait permasalahan kesehatan mental di masyarakat, menggambarkan proses bisnis sebelum adanya sistem yang diusulkan, serta menentukan fungsionalitas yang harus dimiliki oleh sistem. Hasil dari analisis ini menjadi dasar dalam merancang sistem skrining awal kesehatan mental berbasis Web menggunakan LSTM dan skala DASS-21, sehingga sistem yang dikembangkan benar-benar dapat menjawab permasalahan yang dihadapi oleh target pengguna.

3.1.1. Proses Analisis Berjalan

Saat ini, ketika seorang individu, khususnya dari kalangan dewasa awal, mulai merasakan gejala seperti depresi, kecemasan, atau stres berlebih, mereka cenderung melalui beberapa tahapan yang tidak terstruktur dan berisiko. Proses yang umum terjadi adalah sebagai berikut:

1. Tahap Pencarian Informasi Mandiri

Langkah pertama yang paling umum dilakukan oleh individu yang mengalami masalah kesehatan mental adalah mencari informasi secara mandiri melalui mesin pencari seperti Google. Kata kunci yang digunakan biasanya bersifat umum, seperti "cara mengatasi stres" atau "gejala cemas berlebihan". Sayangnya, informasi yang diperoleh sering kali terlalu luas, tidak terpersonalisasi, dan tidak terverifikasi kebenarannya. Kondisi ini berisiko mengarahkan pada *self-diagnosis* yang keliru dan tidak sesuai dengan kondisi psikologis sebenarnya. Di sisi lain, akses untuk melakukan skrining awal yang terpercaya dan mudah dijangkau masih sangat terbatas di masyarakat.

2. Tahap Pencarian Validasi di Media Sosial

Setelah pencarian informasi mandiri, banyak individu beralih ke media sosial seperti Instagram, TikTok, atau forum online untuk mendapatkan validasi atau rasa dimengerti. Mereka mungkin menuliskan curhatan secara anonim maupun terbuka, atau mengonsumsi konten-konten singkat terkait kesehatan mental. Namun, informasi yang beredar di platform ini umumnya tidak memiliki dasar ilmiah yang jelas. Respons dari komunitas daring juga bisa sangat beragam, mulai dari simpati hingga komentar negatif, candaan, atau pengabaian. Alih-alih merasa lebih baik, banyak individu justru semakin merasa tidak dipahami kemudian memilih untuk memendam perasaan dan menarik diri dari upaya pencarian bantuan.

3. Tahap Pertimbangan Bantuan Profesional

Sebagian kecil individu mulai menyadari pentingnya bantuan profesional dan mulai mempertimbangkan untuk berkonsultasi dengan psikolog, psikiater, atau konselor. Namun, langkah ini sering muncul setelah berbagai upaya mandiri dan pencarian validasi sosial gagal memberikan solusi yang memadai. Kesadaran ini penting, tetapi masih belum menjadi pilihan utama bagi kebanyakan orang karena berbagai faktor eksternal dan internal yang memengaruhi pengambilan keputusan mereka.

4. Tahap Menghadapi Hambatan

Keinginan untuk mencari bantuan profesional sering kali terhambat oleh berbagai kendala. Stigma sosial menjadi penghalang utama, banyak orang masih menganggap bahwa pergi ke psikolog berarti “gila” atau “lemah”. Selain itu, faktor biaya, rasa takut terhadap penilaian negatif, dan ketidaktahuan mengenai cara mengakses layanan juga menjadi hambatan signifikan. Akibatnya, meskipun sudah menyadari pentingnya bantuan profesional, banyak individu tetap urung mencari pertolongan.

5. Hasil Akhir

Proses pencarian yang tidak terarah, ditambah dengan hambatan sosial dan struktural, menyebabkan mayoritas individu akhirnya memilih untuk memendam semua perasaan dan gejala yang mereka alami seorang diri. Tidak adanya pendampingan profesional dan minimnya sistem skrining awal yang terpercaya membuat masalah kesehatan mental yang seharusnya bisa ditangani sejak dini justru dibiarkan memburuk. Hal ini menciptakan siklus masalah yang terus berulang dan semakin sulit diatasi seiring waktu.

Permasalahan utama dari proses berjalan saat ini adalah:

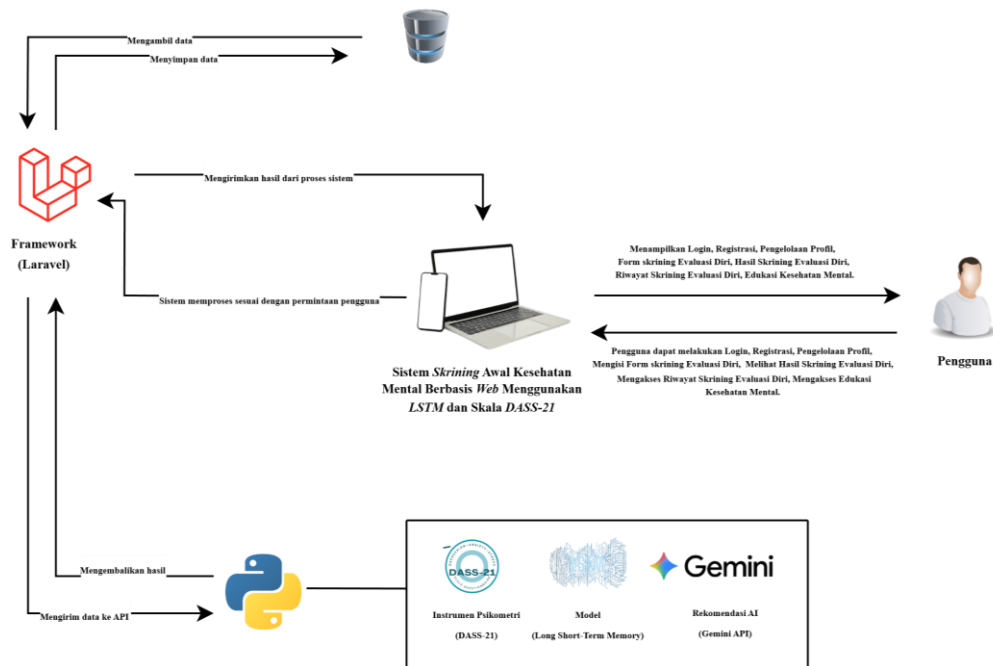
1. Minimnya sistem skrining awal yang terpercaya untuk membantu individu menilai kondisi kesehatan mental mereka.
2. Informasi di internet dan media sosial tidak terverifikasi sehingga dapat menyesatkan pengguna.
3. Keterbatasan akses ke tenaga profesional karena stigma sosial dan rasa takut.
4. Minimnya ruang privat untuk refleksi membuat individu tidak memiliki sarana yang aman dan privat untuk melakukan refleksi awal terhadap kondisinya sebelum memutuskan untuk berbicara dengan orang lain atau profesional.
5. Minimnya sistem terintegrasi yang bisa memberikan hasil analisis sekaligus rekomendasi langkah awal secara objektif.

Dengan kondisi tersebut, kebutuhan akan sebuah sistem skrining awal kesehatan mental berbasis Web menjadi sangat relevan. Sistem ini diharapkan dapat memberikan hasil yang lebih objektif, terukur, dan terpercaya, sekaligus menjadi sarana awal untuk membantu individu memahami kondisi mental mereka.

3.1.2. Gambaran Umum Sistem

Bagian ini berisi uraian singkat serta gambaran visual mengenai sistem yang akan dikembangkan sebagai solusi dari permasalahan pada proses yang

sedang berjalan. Tujuannya adalah agar pembaca mendapatkan pemahaman awal mengenai arsitektur dan alur kerja sistem yang akan dibangun sebelum masuk ke detail perancangan. Berikut adalah Gambaran Umum Sistem dari Rancang Bangun Sistem *Skrining* Awal Kesehatan Mental Berbasis Web Menggunakan *LSTM* dan Skala *DASS-21*:



Gambar 2. Gambaran Umum Sistem

Sebagaimana diilustrasikan pada Gambar 2, Pengguna adalah aktor utama yang berinteraksi langsung dengan sistem melalui antarmuka aplikasi Web responsif yang dapat diakses dari berbagai perangkat seperti laptop maupun ponsel. Melalui platform ini, pengguna dapat memanfaatkan seluruh fungsionalitas inti yang disediakan, mulai dari registrasi akun, *Login*, mengisi form evaluasi diri untuk skrining, melihat hasil skrining, hingga mengakses kembali riwayat hasil dan konten edukasi kesehatan mental.

3.2. Perancangan

Berdasarkan hasil analisis kebutuhan yang telah diuraikan pada sub-bab sebelumnya, bagian ini akan merinci perancangan teknis dari sistem yang akan dibangun. Perancangan ini mencakup pendefinisian kebutuhan fungsional dan non-fungsional, pemodelan interaksi pengguna melalui Diagram *Use Case* beserta skenarionya, penggambaran alur kerja proses dengan *Activity Diagram*,

perancangan struktur basis data menggunakan Entity-Relationship Diagram (ERD), hingga visualisasi perancangan antarmuka pengguna (UI/UX).

3.2.1. Kebutuhan Fungsional

Kebutuhan fungsional yang terdapat pada Rancang Bangun Sistem Skrining Awal Kesehatan Mental Berbasis Web Menggunakan LSTM dan Skala DASS-21 dapat dilihat pada tabel 5 berikut.

Tabel 5. Functional Requirement

Kode	Kebutuhan Fungsional
FR-01	Pengguna dapat melakukan Registrasi untuk membuat akun baru.
FR-02	Pengguna dapat melakukan <i>Login</i> dengan akun yang sudah didaftarkan.
FR-03	Pengguna dapat melakukan <i>Logout</i> .
FR-04	Pengguna dapat melakukan Pengelolaan Profil.
FR-05	Pengguna dapat mengisi Form Skrining Evaluasi Diri.
FR-06	Pengguna dapat melihat Hasil Skrining Evaluasi Diri.
FR-07	Pengguna dapat mengakses Riwayat Skrining Evaluasi Diri.
FR-08	Pengguna dapat mengakses Edukasi Kesehatan Mental.

3.2.2. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional yang terdapat pada Rancang Bangun Sistem Skrining Awal Kesehatan Mental Berbasis Web Menggunakan LSTM dan Skala DASS-21 dapat dilihat pada tabel 6 berikut.

Tabel 6. Non-Functional Requirement

Kode	Kebutuhan Non-Fungsional
NFR-01	Sistem harus responsif dan dapat diakses baik melalui laptop maupun perangkat seluler.
NFR-02	Sistem dapat dijalankan pada Web browser seperti Google Chrome, Mozilla Firefox, dan Internet Explorer.

NFR-03	Bahasa yang digunakan pada sistem adalah Bahasa Indonesia.
NFR-04	Sistem wajib melindungi kredensial otentikasi akun pengguna dengan cara mengenkripsi kata sandi (<i>password</i>) di dalam basis data menggunakan algoritma kriptografi <i>hashing</i> Bcrypt.
NFR-05	Sistem harus menyertakan peringatan medis dan legal (<i>disclaimer</i>) secara eksplisit di setiap halaman hasil evaluasi, yang menyatakan bahwa sistem hanya bertindak sebagai instrumen skrining awal dan bukan alat diagnosis klinis, sesuai dengan Undang-Undang Kesehatan dan Kode Etik Psikologi.

3.2.3. Diagram Use Case

Use Case Diagram dari Rancang Bangun Sistem Skrining Awal Kesehatan Mental Berbasis Web Menggunakan LSTM dan Skala DASS-21 dapat dilihat pada Gambar 3 berikut:



Gambar 3. Use Case

Berdasarkan Gambar 3, *Use Case Diagram* untuk sistem skrining kesehatan mental ini menggambarkan interaksi yang dapat dilakukan oleh satu aktor utama, yaitu Pengguna. Secara umum, Pengguna dapat Melakukan Registrasi untuk membuat akun baru dan Melakukan *Login* untuk masuk ke dalam sistem. Setelah berhasil *Login*, yang merupakan syarat utama untuk mengakses fitur-fitur inti, Pengguna dapat Mengelola Profil pribadinya, Mengisi Form Skrining Evaluasi Diri, Melihat Hasil Skrining Evaluasi Diri secara langsung, Mengakses Riwayat Skrining Evaluasi Diri dari tes-tes sebelumnya, serta Mengakses Edukasi Kesehatan Mental. Selain itu, sistem juga menyediakan fungsionalitas Melakukan *Logout* sebagai opsi tambahan setelah Pengguna selesai menggunakan sistem.

3.2.4. Skenario *Use Case*

Proses registrasi akun yang dilakukan oleh pengguna untuk membuat akun Sistem Skrining Awal Kesehatan Mental Berbasis Web Menggunakan LSTM dan Skala DASS-21 dijelaskan melalui serangkaian langkah dalam Main Success Scenario. Rincian langkah-langkah tersebut ditunjukkan pada Tabel 7.

Tabel 7. Use Case Scenario Registrasi Akun

Use Case Scenario 1: Melakukan Registrasi Akun	
Nomor	UC-01
Nama	Registrasi Akun
Deskripsi	Pengguna dapat melakukan Registrasi untuk membuat akun baru agar dapat <i>Login</i> ke dalam aplikasi.
<i>Primary Aktor</i>	Pengguna
<i>Preconditions</i>	Pengguna belum memiliki akun untuk <i>Login</i> ke dalam aplikasi.
<i>Postconditions</i>	Pengguna berhasil membuat akun baru.

<i>Main Success Scenario</i>	1. Pengguna berada di halaman beranda (guest). 2. Pengguna meng-klik button “Daftar”. 3. Sistem Menampilkan modal “Registrasi Akun”. 4. Pengguna mengisi form “Registrasi Akun”. 5. Pengguna meng-klik button “Daftar”. 6. Sistem memvalidasi format dan duplikasi data. 7. Sistem menyimpan data pengguna ke dalam database. 8. Pengguna berhasil melakukan “Registrasi Akun” dan akan memunculkan pop-up berhasil. 9. Sistem mengarahkan pengguna ke halaman “Login”.
<i>Alternative Flow (Extension)</i>	-

Proses *Login* yang dilakukan oleh pengguna untuk mengakses Sistem Skrining Awal Kesehatan Mental Berbasis Web Menggunakan LSTM dan Skala DASS-21 dijelaskan melalui serangkaian langkah dalam Main Success Scenario. Rincian langkah-langkah tersebut ditunjukkan pada Tabel 8.

Tabel 8. Use Case Scenario Login

Use Case Scenario 2: Melakukan Login	
Nomor	UC-02
Nama	<i>Login</i>
Deskripsi	Pengguna harus melakukan <i>Login</i> untuk menggunakan fitur-fitur yang tersedia pada aplikasi.

<i>Primary Aktor</i>	Pengguna
<i>Preconditions</i>	Pengguna berada di halaman beranda (guest) dan belum <i>Login</i> ke dalam aplikasi.
<i>Postconditions</i>	Pengguna berhasil <i>Login</i> ke dalam aplikasi dan dapat mengakses fitur-fitur yang tersedia.
<i>Main Success Scenario</i>	1. Pengguna berada di halaman beranda (guest) aplikasi. 2. Pengguna meng-klik button “Masuk”. 3. Sistem Menampilkan modal “Masuk”. 4. Pengguna mengisi Email dan Kata Sandi. 5. Pengguna meng-klik button “Masuk”. 6. Sistem memvalidasi informasi Masuk. 7. Jika informasi benar, sistem akan akan memunculkan pop-up berhasil, mengaktifkan user session, dan mengarahkan ke halaman beranda (user). 8. Jika informasi salah, sistem akan akan memunculkan pop-up gagal yang berisi pesan kesalahan.
<i>Alternative Flow (Extension)</i>	-

Proses *Logout* yang dilakukan oleh pengguna untuk mengakhiri sesi pengguna pada Sistem Skrining Awal Kesehatan Mental Berbasis Web Menggunakan LSTM dan Skala DASS-21 dijelaskan melalui serangkaian

langkah dalam Main Success Scenario. Rincian langkah-langkah tersebut ditunjukkan pada Tabel 9.

Tabel 9. Use Case Scenario Logout

Use Case Scenario 3: Melakukan Logout	
Nomor	UC-03
Nama	<i>Logout</i>
Deskripsi	Pengguna dapat melakukan <i>Logout</i> dari aplikasi.
Primary Aktor	Pengguna
Preconditions	Pengguna sudah <i>Login</i> ke dalam aplikasi dan dapat mengakses fitur-fitur yang tersedia.
Postconditions	Pengguna berhasil <i>Logout</i> dari aplikasi.
Main Success Scenario	1. Pengguna sudah <i>Login</i>. 2. Pengguna meng-klik button “Profil”. 3. Sistem menampilkan halaman “Profil”. 4. Pengguna meng-klik opsi “Keluar”. 5. Sistem menonaktifkan user session dan mengarahkan pengguna ke halaman beranda (guest).
Alternative Flow (Extension)	-

Proses yang dilakukan oleh pengguna untuk mengelola profil pada Sistem Skrining Awal Kesehatan Mental Berbasis Web Menggunakan LSTM dan Skala DASS-21 dijelaskan melalui serangkaian langkah dalam Main Success Scenario. Rincian langkah-langkah tersebut ditunjukkan pada Tabel 10.

Tabel 10. Use Case Scenario Mengelola Profil

Use Case Scenario 4: Mengelola Profil	
Nomor	UC-04
Nama	Mengelola Profil
Deskripsi	Pengguna dapat menambah, menghapus, mengedit, atau melihat data profil.
Primary Aktor	Pengguna
Preconditions	Pengguna sudah <i>Login</i> ke dalam aplikasi dan dapat mengakses fitur-fitur yang tersedia.
Postconditions	Pengguna berhasil menambah, menghapus, mengedit, atau melihat data profil, serta profil pengguna disimpan dan diperbarui pada database.
Main Success Scenario	1. Pengguna sudah <i>Login</i>. 2. Pengguna meng-klik button “Profil”. 3. Sistem menampilkan halaman “Profil”. 4. Pengguna meng-klik “Edit Profil”. 5. Pengguna melihat/mengisi/mengubah/menghapus data “Profil”. 6. Sistem melakukan validasi data. 7. Jika benar, Sistem akan memunculkan pop up berhasil, menyimpan data terbaru ke database dan memuat ulang tampilan halaman “Profil”.

	8. Jika salah, Sistem akan memunculkan pop up gagal yang berisi pesan kesalahan.
<i>Alternative Flow (Extension)</i>	-

Proses yang dilakukan oleh pengguna untuk Mengisi Form Skrining Evaluasi Diri pada Sistem Skrining Awal Kesehatan Mental Berbasis Web Menggunakan LSTM dan Skala DASS-21 dijelaskan melalui serangkaian langkah dalam Main Success Scenario. Rincian langkah-langkah tersebut ditunjukkan pada Tabel 11.

Tabel 11. Use Case Scenario Mengisi Form Skrining Evaluasi Diri

Use Case Scenario 5: Mengisi Form Skrining Evaluasi Diri	
Nomor	UC-05
Nama	Mengisi Form Skrining Evaluasi Diri
Deskripsi	Pengguna dapat Mengisi Form Skrining Evaluasi Diri
<i>Primary Aktor</i>	Pengguna
<i>Preconditions</i>	Pengguna sudah <i>Login</i> ke dalam aplikasi dan dapat mengakses fitur-fitur yang tersedia.
<i>Postconditions</i>	Pengguna berhasil mengisi dan submit Form Skrining Evaluasi Diri.
<i>Main Success Scenario</i>	1. Pengguna sudah <i>Login</i>. 2. Pengguna meng-klik button “Mulai Evaluasi Diri”. 3. Sistem menampilkan halaman awal form skrining evaluasi diri. 4. Pengguna membaca dan memahami informasi yang diberikan pada halaman awal form skrining evaluasi diri.

	5. Pengguna meng-klik button “Mulai”. 6. Sistem menampilkan form kuisioner dass-21. 7. Pengguna mengisi form kuisioner dass-21. 8. Pengguna meng-klik button “Selanjutnya”. 9. Sistem melakukan validasi data. 10. Sistem menampilkan halaman “Curhat” 11. Pengguna dapat mengisi kolom “Curhat” yang tersedia dengan teks. 12. Pengguna meng-klik buttton “Selesai”. 13. Sistem melakukan validasi data. 14. Sistem kirim request ke mentalku-ml-api. 15. Sistem menampilkan loading yang berarti sedang memproses data form skrining evaluasi diri yang telah diisi oleh pengguna.
<i>Alternative Flow (Extension)</i>	-

Proses yang dilakukan oleh pengguna untuk Melihat Hasil Skrining Evaluasi Diri pada Sistem Skrining Awal Kesehatan Mental Berbasis Web Menggunakan LSTM dan Skala DASS-21 dijelaskan melalui serangkaian langkah dalam Main Success Scenario. Rincian langkah-langkah tersebut ditunjukkan pada Tabel 12.

Tabel 12. Use Case Scenario Melihat Hasil Skrining Evaluasi Diri

Use Case Scenario 6: Melihat Hasil Skrining Evaluasi Diri	
Nomor	UC-06
Nama	Melihat Hasil Skrining Evaluasi Diri
Deskripsi	Pengguna dapat Melihat Hasil Skrining Evaluasi Diri.
Primary Aktor	Pengguna
Preconditions	Pengguna sudah meng-klik button “Selesai” pada form skrining evaluasi diri.
Postconditions	Pengguna berhasil melihat hasil skrining evaluasi diri.
Main Success Scenario	16. Sistem menerima Response dari mentalku-ml-api. 17. Sistem menyimpan data ke database dan melakukan query edukasi sesuai dengan hasil response. 18. Sistem menampilkan halaman hasil skrining evaluasi diri. 19. Pengguna dapat membaca dan memahami hasil skrining evaluasi diri. 20. Pengguna meng-klik “Selengkapnya” pada salah satu edukasi (bacaan terkait) yang diberikan. 21. Sistem membuka tab baru dan mengarahkan pengguna ke halaman detail edukasi tersebut.
Alternative Flow (Extension)	-

Proses yang dilakukan oleh pengguna untuk Mengakses Riwayat Skrining Evaluasi Diri pada Sistem Skrining Awal Kesehatan Mental Berbasis Web Menggunakan LSTM dan Skala DASS-21 dijelaskan melalui serangkaian langkah dalam Main Success Scenario. Rincian langkah-langkah tersebut ditunjukkan pada Tabel 13.

Tabel 13. Use Case Scenario Riwayat Skrining Evaluasi Diri

Use Case Scenario 7: Riwayat Skrining Evaluasi Diri	
Nomor	UC-07
Nama	Mengakses Riwayat Skrining Evaluasi Diri
Deskripsi	Pengguna dapat melihat list dan detail riwayat skrining pengguna.
<i>Primary Aktor</i>	Pengguna
<i>Preconditions</i>	Pengguna sudah selesai mengisi form skrining evaluasi diri.
<i>Postconditions</i>	Pengguna berhasil melihat list dan detail riwayat skrining pengguna.
<i>Main Success Scenario</i>	1. Pengguna meng-klik button “Profil”. 2. Sistem menampilkan halaman “Profil”. 3. Pengguna dapat melihat list riwayat skrining evaluasi diri pada bagian bawah halaman “Profil”. 4. Pengguna menggunakan filtering yang tersedia untuk mempermudah melihat daftar riwayat skrining evaluasi diri.

	5. Sistem menampilkan daftar riwayat skrining evaluasi diri berdasarkan filtering yang dipilih. 6. Pengguna dapat menggunakan pagination yang tersedia untuk melihat daftar skrining evaluasi diri yang lainnya. 7. Sistem berpindah halaman untuk menampilkan data berikutnya. 8. Pengguna memilih salah satu entry riwayat skrining evaluasi diri dengan cara meng-klik button “detail/mata”. 9. Sistem menampilkan halaman detail riwayat skrining pengguna.
<i>Alternative Flow (Extension)</i>	-

Proses yang dilakukan oleh pengguna untuk Mengakses Edukasi Kesehatan Mental pada Sistem Skrining Awal Kesehatan Mental Berbasis Web Menggunakan LSTM dan Skala DASS-21 dijelaskan melalui serangkaian langkah dalam Main Success Scenario. Rincian langkah-langkah tersebut ditunjukkan pada Tabel 14.

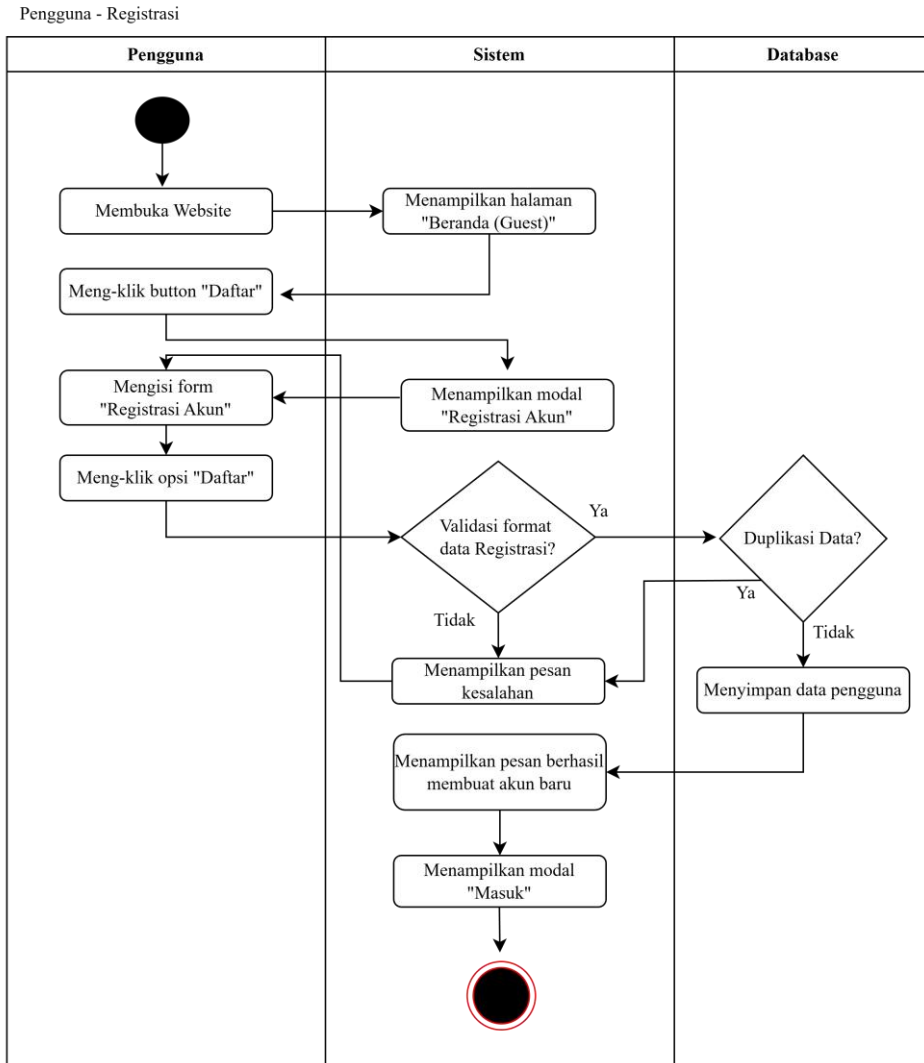
Tabel 14. Use Case Scenario Edukasi Kesehatan Mental

<i>Use Case Scenario 8: Edukasi Kesehatan Mental</i>	
Nomor	UC-08
Nama	Mengakses Edukasi Kesehatan Mental.
Deskripsi	Pengguna dapat melihat list dan detail edukasi kesehatan mental, serta dapat melakukan filter edukasi berdasarkan kategori.
<i>Primary Aktor</i>	Pengguna

<i>Preconditions</i>	Pengguna sudah <i>Login</i> ke dalam aplikasi dan dapat menggunakan fitur-fitur yang tersedia.
<i>Postconditions</i>	Pengguna berhasil melihat list dan detail edukasi kesehatan mental, serta dapat melakukan filter edukasi berdasarkan kategori.
<i>Main Success Scenario</i>	1. Pengguna meng-klik button “Edukasi” pada navbar aplikasi. 2. Sistem menampilkan list edukasi kesehatan mental. 3. Pengguna memilih kategori list edukasi kesehatan mental. 4. Sistem menampilkan list edukasi kesehatan mental sesuai dengan kategori yang dipilih oleh pengguna. 5. Pengguna dapat menggunakan pagination yang tersedia untuk melihat daftar skrining evaluasi diri yang lainnya. 6. Sistem berpindah halaman untuk menampilkan data berikutnya. 7. Pengguna meng-klik button “Selengkapnya” pada salah satu list edukasi kesehatan mental. 8. Sistem membuka tab baru dan mengarahkan pengguna ke <i>Link</i> Website yang dipilih. 9. Sistem menampilkan detail dari edukasi kesehatan mental.
<i>Alternative Flow (Extension)</i>	-

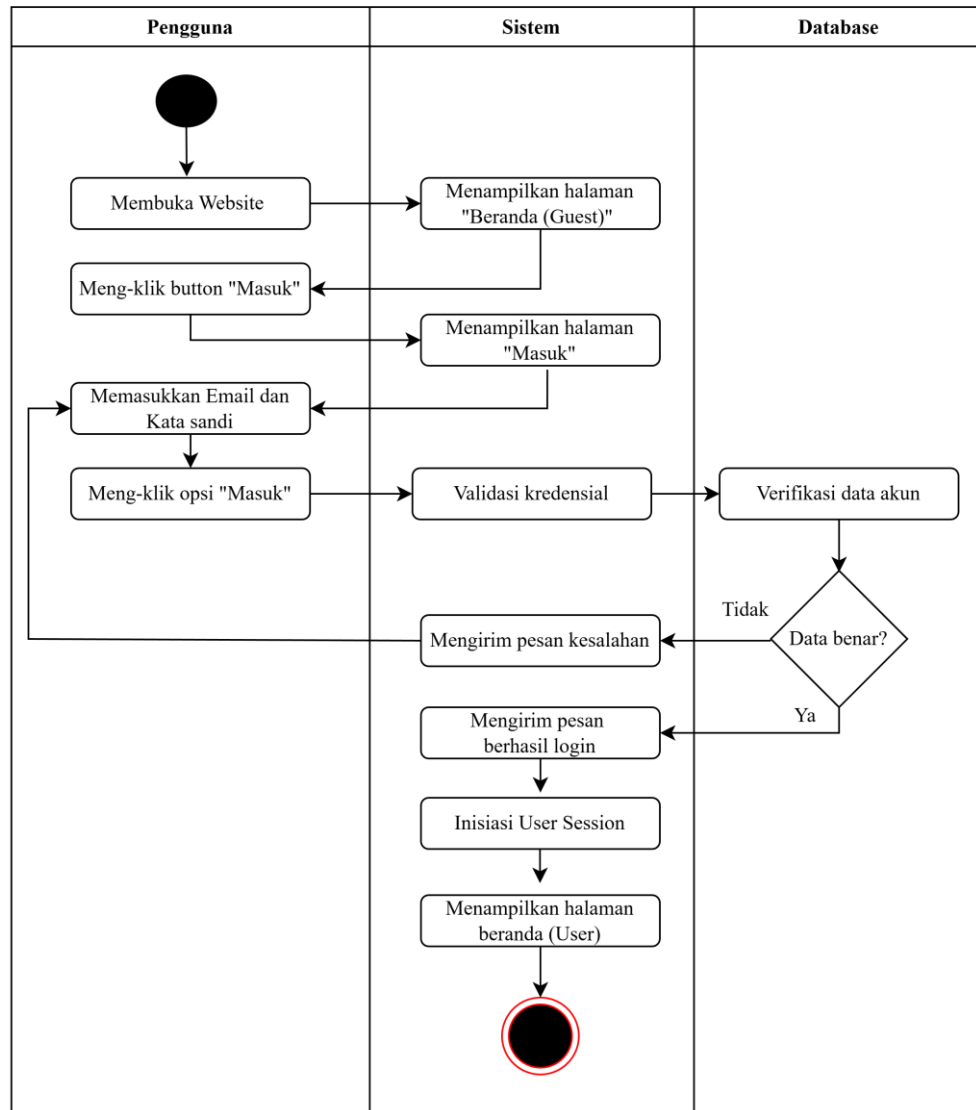
3.2.5. Activity Diagram

Setelah alur kerja dirinci secara naratif melalui skenario *Use Case*, *Activity Diagram* digunakan untuk memvisualisasikan alur tersebut secara lebih detail. Diagram-Diagram berikut akan memodelkan urutan aktivitas, titik keputusan (decision points), dan alur proses secara dinamis untuk fitur-fitur utama sistem yang telah dijelaskan sebelumnya.



Gambar 4. Activity Diagram Registrasi Akun

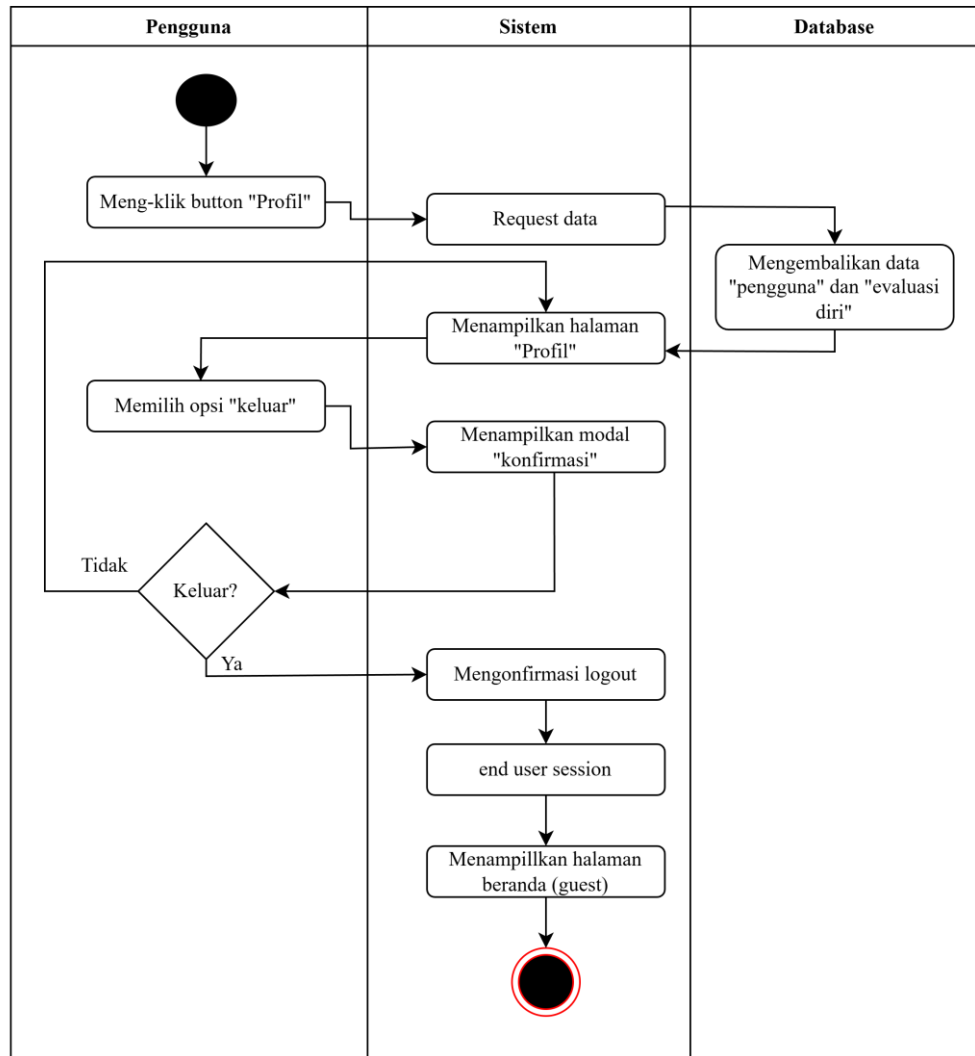
Gambar 4 menggambarkan *Activity Diagram* proses melakukan registrasi akun yang dapat dilakukan oleh aktor pengguna. Diagram ini menunjukkan alur sistem dari awal hingga akhir untuk proses melakukan registrasi akun.



Gambar 5. Activity Diagram Login

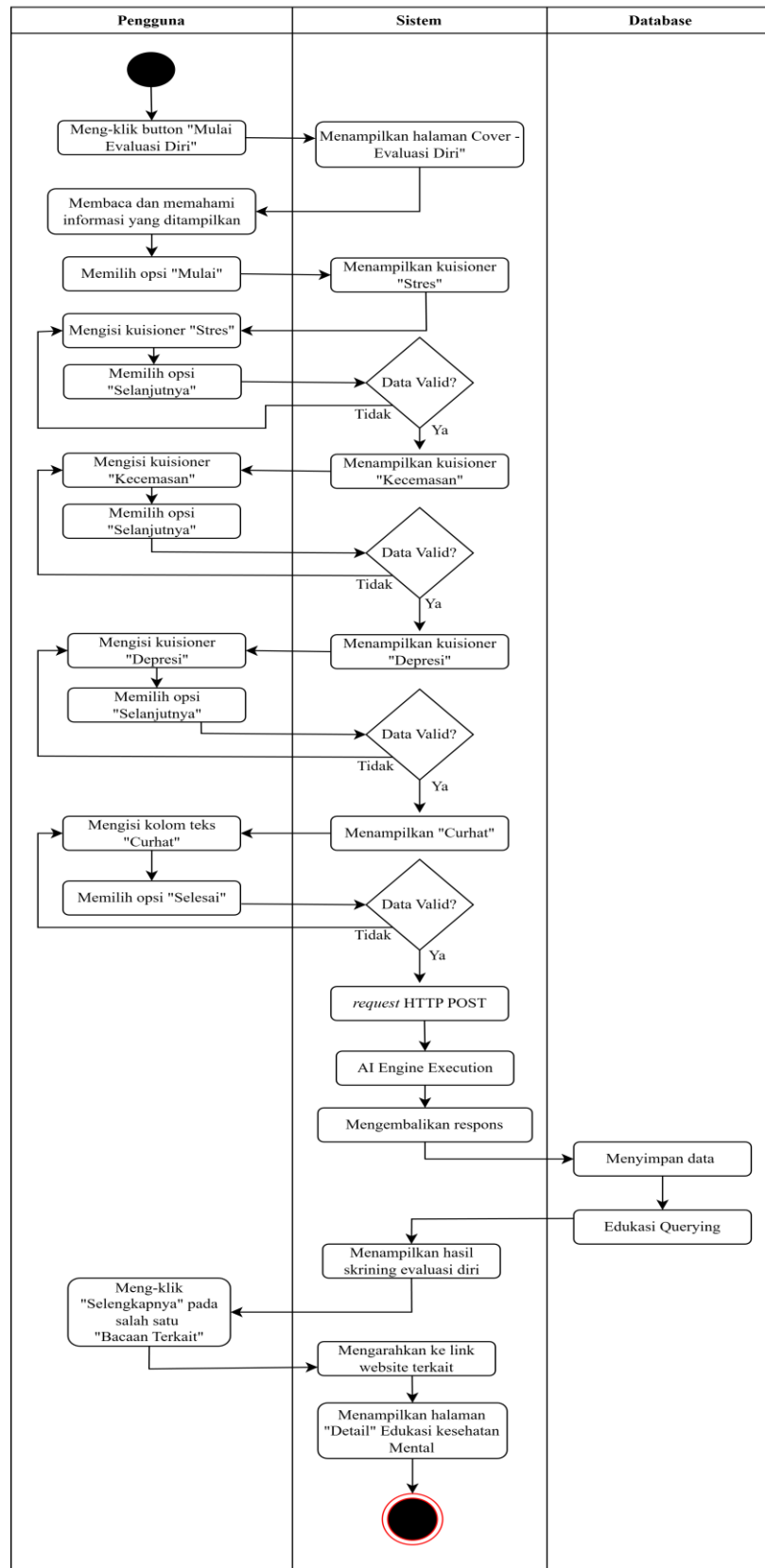
Gambar 5 menggambarkan Activity Diagram proses *Login* yang dapat dilakukan oleh satu aktor, yaitu Pengguna. Diagram ini menunjukkan alur yang dimulai dari tampilan form *Login*.

Pengguna - Logout



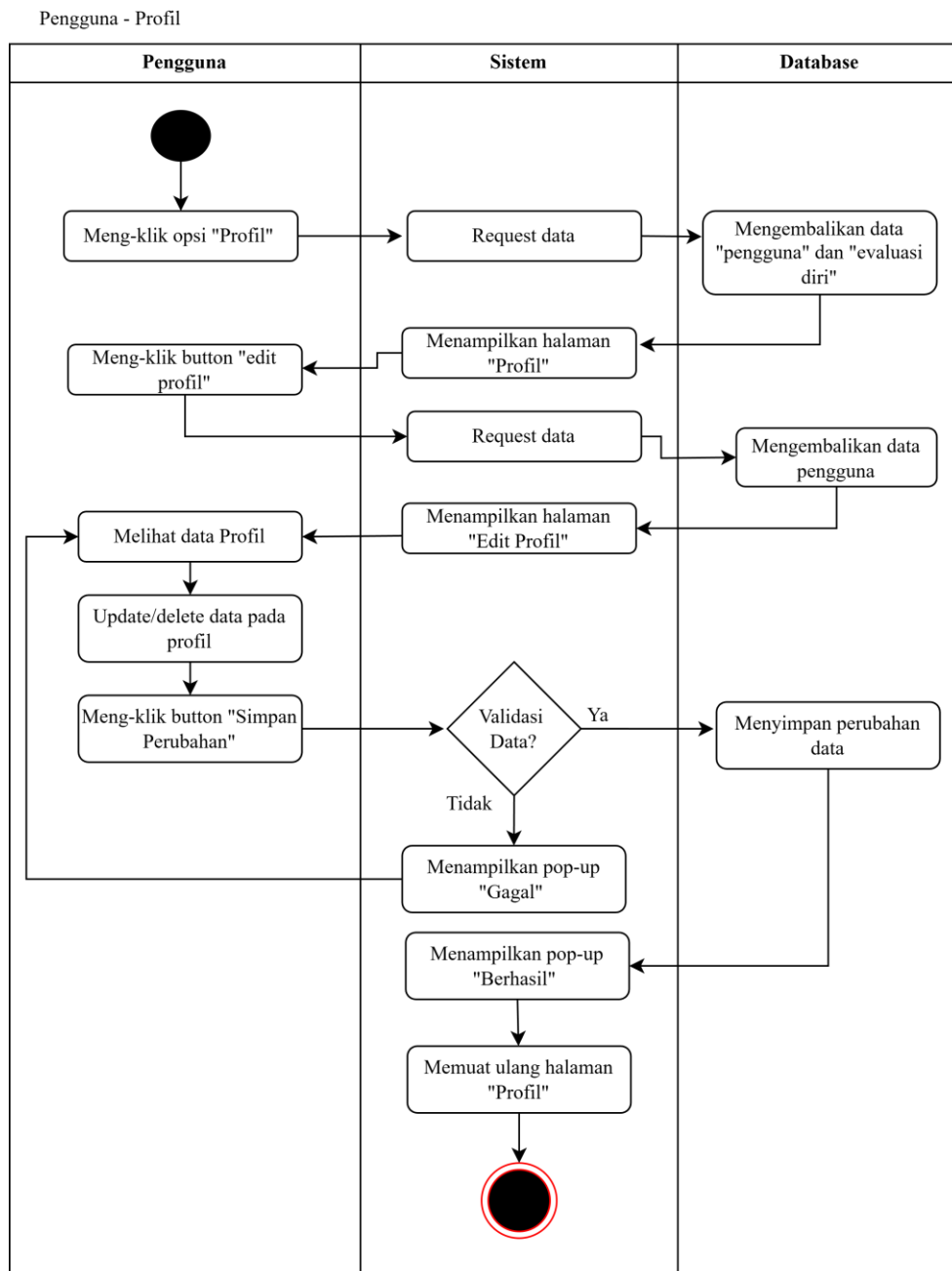
Gambar 6. Activity Diagram Logout

Gambar 6 menggambarkan Activity Diagram proses melakukan *Logout* yang dapat dilakukan oleh satu aktor, yaitu pengguna. Diagram ini menunjukkan alur sistem dari awal hingga akhir untuk proses melakukan *Logout*.



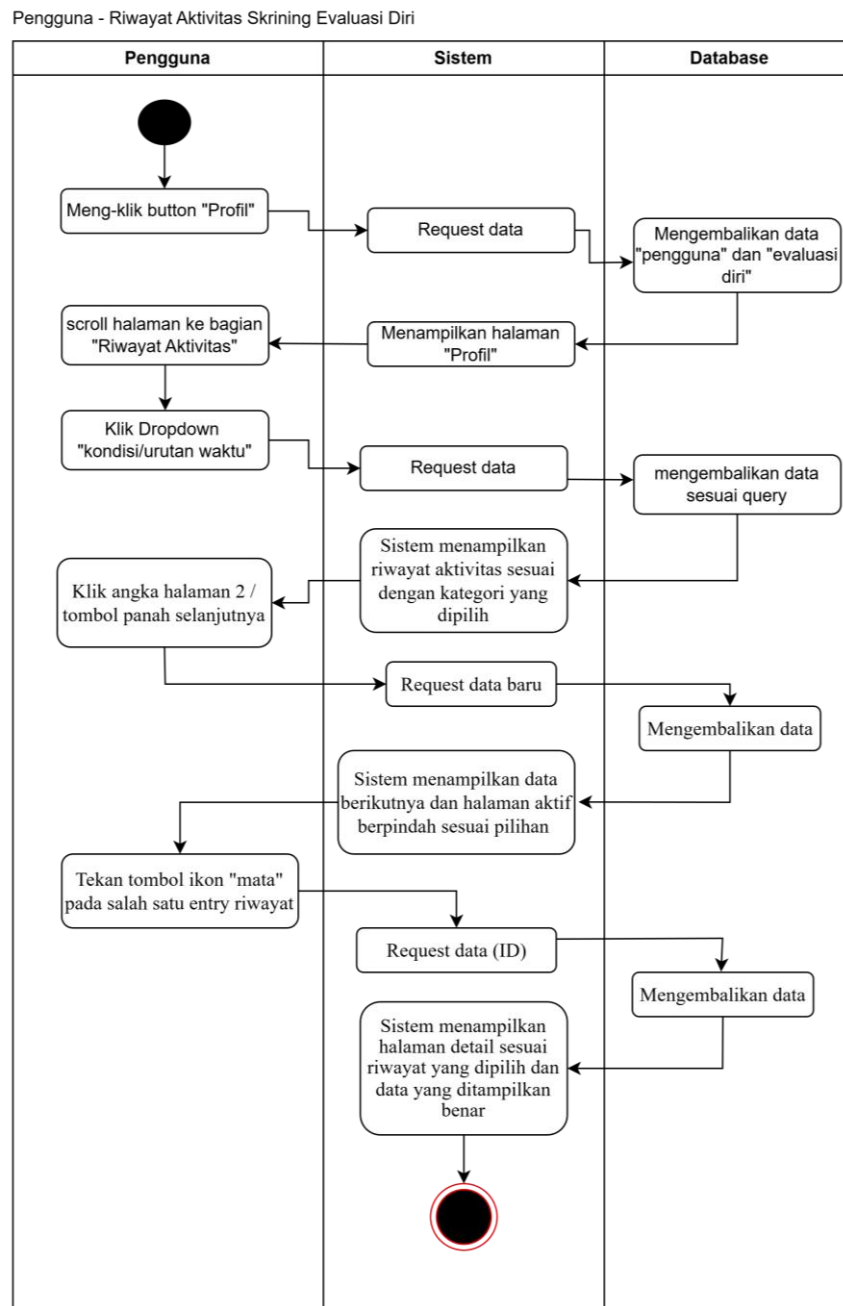
Gambar 7. Activity Diagram Mengisi Form Skrining Evaluasi Diri & Melihat Hasil Skrining Evaluasi Diri

Gambar 7 menggambarkan *Activity Diagram* proses mengisi form skrining evaluasi diri dan melihat hasil skrining evaluasi diri yang dapat dilakukan oleh satu aktor, yaitu pengguna. Diagram ini menunjukkan alur sistem dari awal hingga akhir untuk proses mengisi form skrining evaluasi diri dan melihat hasil skrining evaluasi diri.



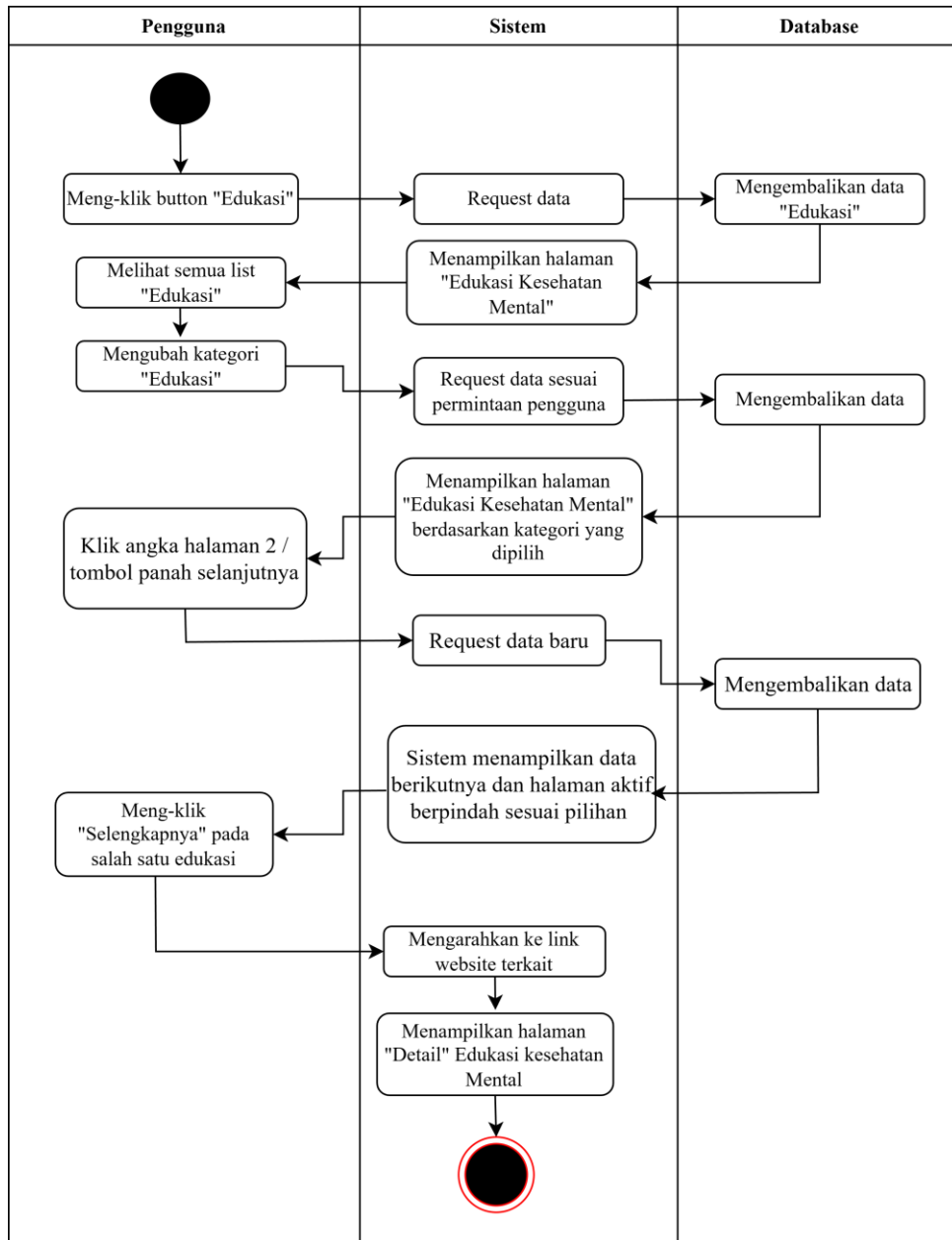
Gambar 8. Activity Diagram Mengelola Profil

Gambar 8 menggambarkan *Activity Diagram* proses mengelola profil yang dapat dilakukan oleh satu aktor, yaitu pengguna. Diagram ini menunjukkan alur sistem dari awal hingga akhir untuk proses mengelola profil.



Gambar 9. Activity Diagram Mengakses Riwayat Skrining Evaluasi Diri

Gambar 9 menggambarkan *Activity Diagram* proses mengakses riwayat skrining evaluasi diri yang dapat dilakukan oleh satu aktor, yaitu pengguna. Diagram ini menunjukkan alur sistem dari awal hingga akhir untuk proses mengakses riwayat skrining evaluasi diri.

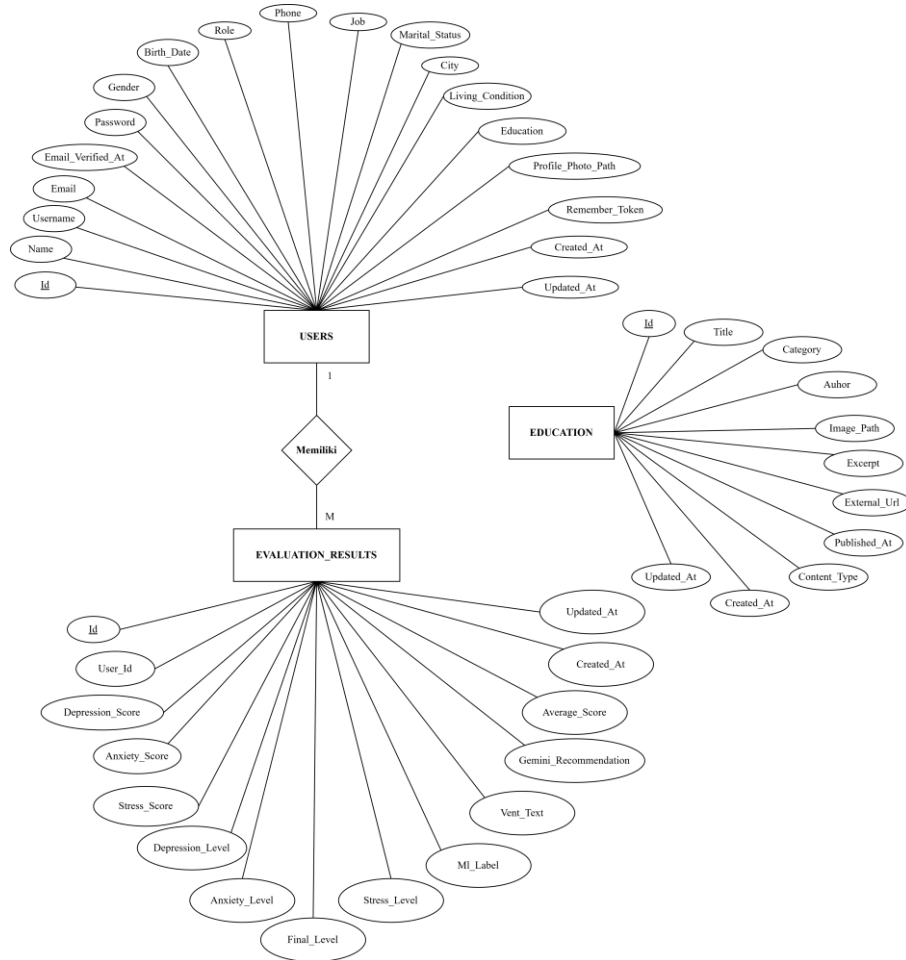


Gambar 10. Activity Diagram Mengakses Edukasi Kesehatan Mental

Gambar 10 menggambarkan Activity Diagram proses mengakses edukasi kesehatan mental yang dapat dilakukan oleh satu aktor, yaitu pengguna. Diagram ini menunjukkan alur sistem dari awal hingga akhir untuk proses mengakses edukasi kesehatan mental.

3.2.6. ER Diagram

Entity Relationship Diagram adalah sekumpulan cara untuk mendeskripsikan data-data yang dibuat berdasarkan dan berasal dari dunia nyata yang disebut dengan entitas (entity) serta hubungan (relationship) antar entitas tersebut menggunakan beberapa notasi. Entity Relationship Diagram (ERD) pada sistem ini diperlihatkan pada Gambar 11.

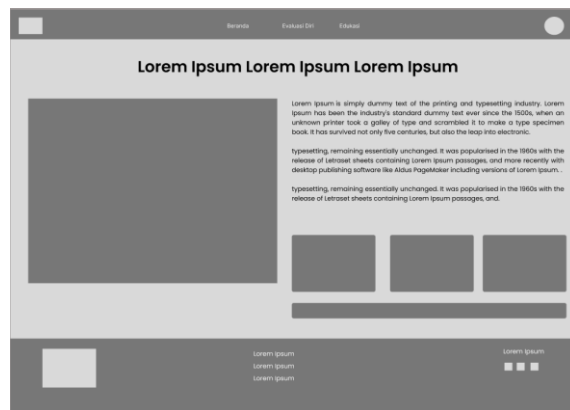


Gambar 11. Entity Relationship Diagram

ERD tersebut merancang struktur basis data dengan tiga entitas utama: USERS yang memiliki 19 kolom dengan Id sebagai primary key, EVALUATION_RESULTS yang memiliki 15 kolom dengan Id sebagai primary key, dan EDUKASI yang memiliki 11 kolom dengan Id sebagai primary key. Struktur ini mendefinisikan relasi one-to-many antara entitas USERS dan EVALUATION_RESULTS, yang berarti satu pengguna dapat memiliki banyak riwayat skrining evaluasi diri.

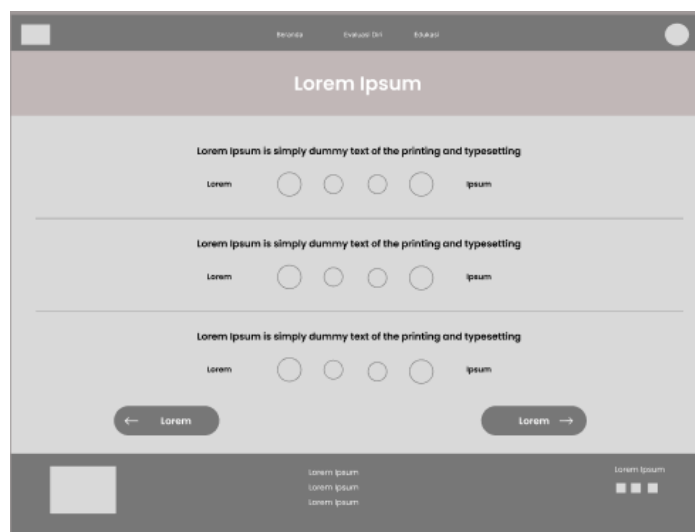
3.2.7. Perancangan Antarmuka (*Wireframe*)

Tahap perancangan antarmuka pengguna diawali dengan pembuatan *Wireframe* low-fidelity sebagai cetak biru visual sistem. *Wireframe* ini digunakan untuk mendefinisikan layout, hierarki informasi, dan alur navigasi sebelum pengembangan frontend. Dengan pendekatan minimalis tanpa elemen estetika, perancangan difokuskan pada fungsionalitas dan efisiensi interaksi pengguna.



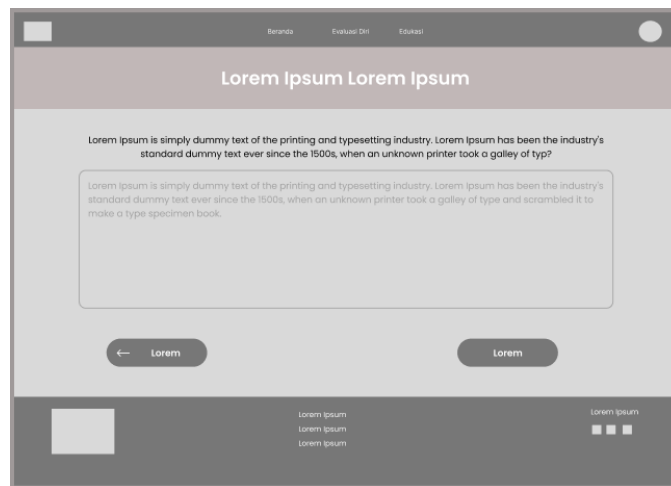
Gambar 12. Wireframe Halaman Awal Form Skrining Evaluasi diri

Gambar 12 menampilkan *Wireframe* low-fidelity halaman awal form pada fitur Skrining Evaluasi Diri. Rancangan ini menunjukkan struktur informasi awal, termasuk teks instruksi, ilustrasi, dan tombol aksi utama. Desain ini bertujuan memberikan konteks yang jelas sebelum pengguna memulai pengisian kuesioner.



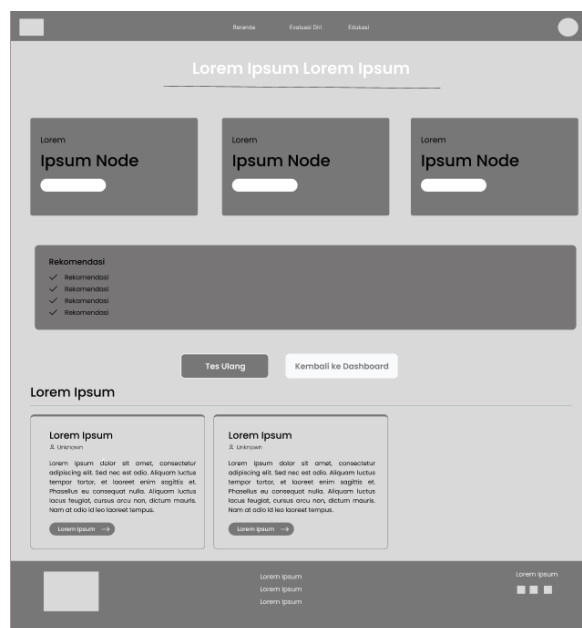
Gambar 13. Wireframe Kuisisioner Form Skrining Evaluasi Diri

Gambar 13 menampilkan *Wireframe* low-fidelity halaman kuesioner pada fitur Skrining Evaluasi Diri. Rancangan ini menggunakan alur bertahap (wizard) yang mencakup indikator progres, pertanyaan, dan pilihan jawaban berbasis radio button. Desain minimalis ini bertujuan menjaga fokus pengguna saat mengisi instrumen DASS-21.



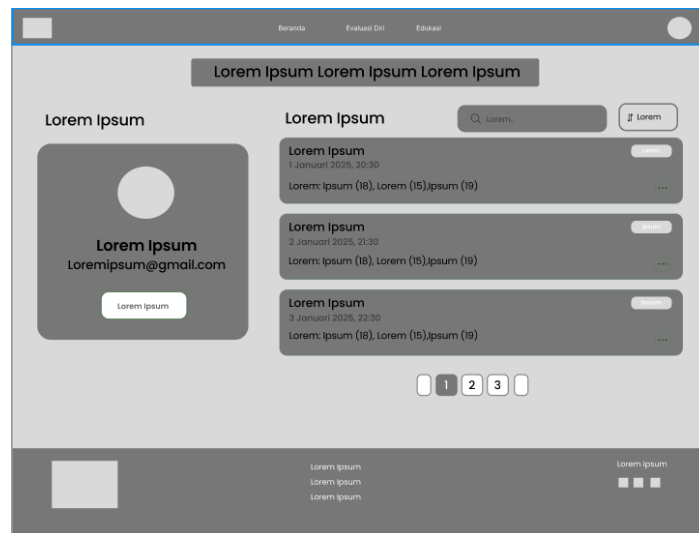
Gambar 14. Wireframe Curhat Form Skrining Evaluasi Diri

Gambar 14 menunjukkan *Wireframe* low-fidelity form Curhat (Ruang Bercerita) sebagai tahap akhir evaluasi. Halaman ini memuat text area dan tombol submit untuk mengatur input teks sebelum diproses model.



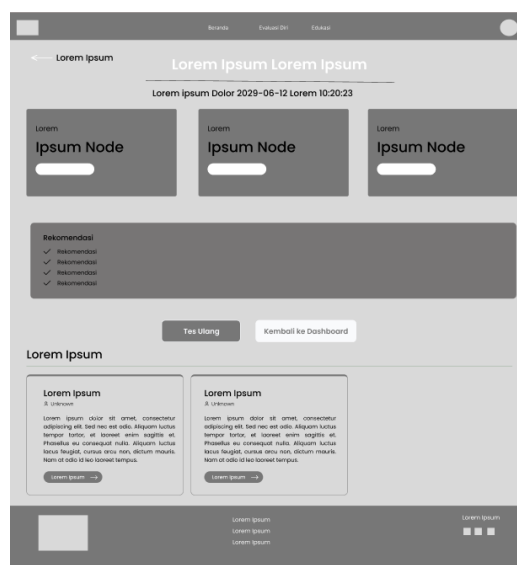
Gambar 15. Wireframe Hasil Skrining Evaluasi Diri

Gambar 15 menampilkan *Wireframe* low-fidelity halaman Hasil Skrining Evaluasi Diri. Rancangan ini menyajikan skor DASS-21 beserta tingkat keparahannya secara terstruktur. Selain itu, disediakan area untuk rekomendasi berbasis AI dan literatur edukasi sebagai tindak lanjut bagi pengguna.



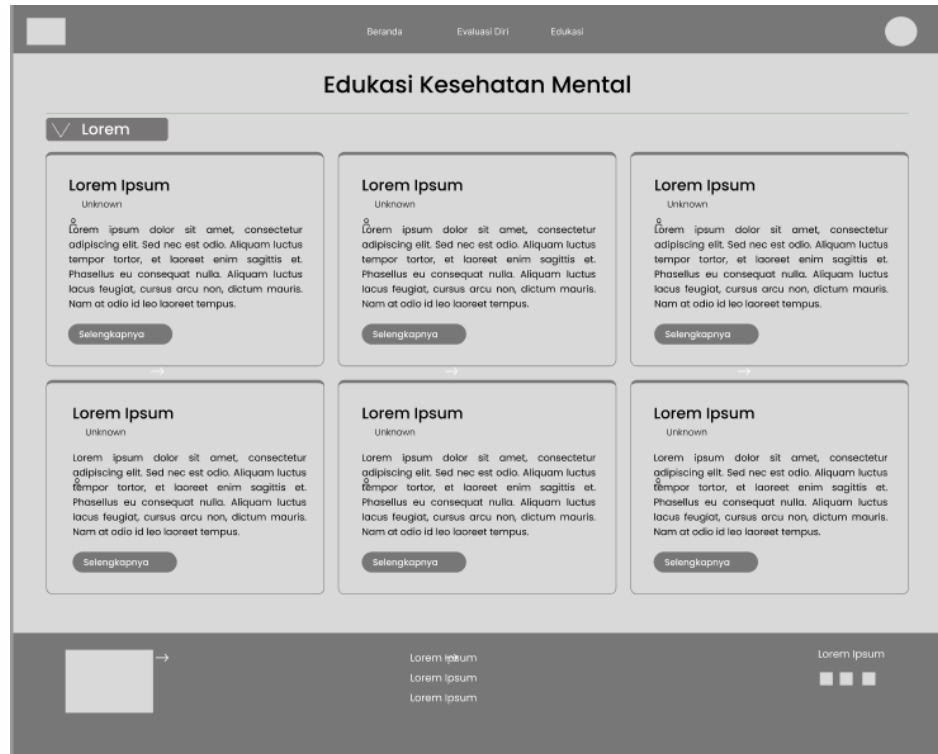
Gambar 16. Wireframe Informasi Riwayat Skrining Evaluasi Diri dan Profil.

Gambar 16 menunjukkan *Wireframe* low-fidelity halaman Profil dengan Riwayat Aktivitas. Halaman ini memuat identitas pengguna serta daftar hasil skrining, dilengkapi fitur filter, sorting, dan paginasi untuk mempermudah pengelolaan data.



Gambar 17. Wireframe Detail Riwayat Skrining Evaluasi Diri

Gambar 17 menunjukkan *Wireframe* low-fidelity halaman Detail Riwayat Skrining yang memuat skor DASS-21 dan rekomendasi dari hasil sebelumnya. Halaman ini digunakan untuk memantau riwayat dan perkembangan kondisi pengguna.



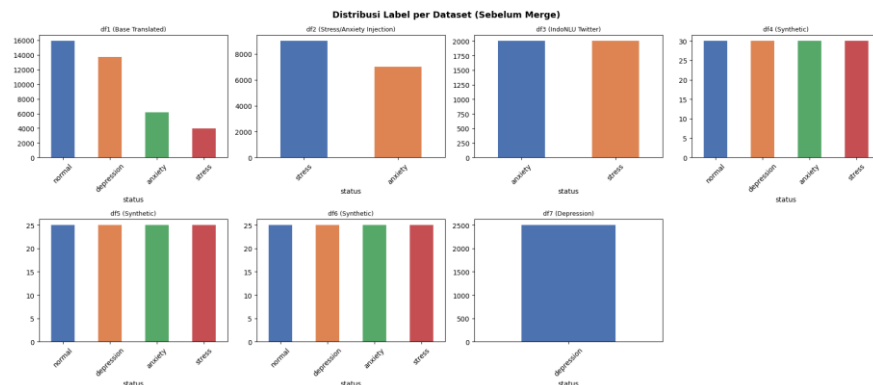
Gambar 18. Wireframe Edukasi Kesehatan Mental

Gambar 18 menampilkan *Wireframe* low-fidelity halaman Edukasi Kesehatan Mental. Halaman ini menyajikan edukasi dalam bentuk grid, dilengkapi fitur filter kategori dan paginasi. Rancangan ini bertujuan memudahkan pengguna dalam mengeksplorasi konten edukasi

3.2.8. Pengumpulan Data

Data yang digunakan dalam proyek ini merupakan gabungan dari tujuh dataset sekunder yang dikumpulkan melalui berbagai platform data terbuka, seperti Hugging Face dan Kaggle, serta diperkaya dengan pembuatan data sintetis. Karakteristik utama dari korpus data ini adalah tingginya keberagaman gaya bahasa yang digunakan. Teks yang dikumpulkan mencakup bahasa formal, teks tidak terstruktur dari forum anonim dan media sosial (seperti Reddit dan Twitter), hingga bahasa sehari-hari. Selain itu, penambahan data

sintetis menggunakan kecerdasan buatan (ChatGPT) dilakukan secara khusus untuk memperluas cakupan kosa kata slang atau "bahasa gaul" Indonesia. Pendekatan ini bertujuan agar model klasifikasi memiliki ketahanan (robustness) yang tinggi saat dihadapkan pada teks keluhan kesah pengguna asli di dunia nyata.



Gambar 19. Distribusi Label per Dataset

Secara keseluruhan, proses pengumpulan dan integrasi ini menghasilkan korpus data dengan total sebanyak 62.566 rekam data. Seluruh data tersebut diklasifikasikan ke dalam empat kategori kondisi mental utama. Evaluasi terhadap dataset akhir menunjukkan distribusi label yang proporsional dan seimbang, dengan rincian sebagai berikut:

1. Depression (Depresi): 16.278 data
2. Normal: 15.990 data
3. Anxiety (Kecemasan): 15.222 data
4. Stress (Stres): 15.076 data

Proporsi jumlah data yang merata antar kategori ini merupakan langkah prapemrosesan yang esensial untuk mencegah terjadinya bias kelas (*class imbalance*) pada saat proses pelatihan model *Deep Learning*. Dengan distribusi tersebut, model diharapkan mampu mempelajari dan mengenali pola linguistik dari masing-masing kondisi emosi secara adil dan presisi.

3.2.9. Preprocessing Data

Tahapan prapemrosesan data (*data Preprocessing*) bertujuan untuk membersihkan elemen teks yang tidak relevan dan mengubah teks tidak

terstruktur menjadi format numerik (*integer sequences*) yang dapat diproses oleh model LSTM. Rangkaian prapemrosesan ini dilakukan melalui beberapa tahap teknis:

1. **Penggabungan dan Pemisahan Data (*Data Splitting*):** Seluruh sumber data digabungkan, disaring untuk menghapus duplikasi (*drop duplicates*), dan diacak (*shuffle*) secara menyeluruh. Data kemudian dibagi menggunakan metode *stratified sampling* menjadi himpunan latih (*train*) sebesar 80%, himpunan validasi (*validation*) 10%, dan himpunan uji (*test*) 10%. Pendekatan ini sangat krusial untuk memastikan bahwa distribusi kelas emosi tetap seimbang dan proporsional di setiap himpunan data.

```
# Combine all the data
df = pd.concat([df1, df2, df3, df4, df5, df6, df7], ignore_index=True)

# Filter only the columns you need
df = df[['text', 'status']].copy()

# GLOBAL DROP DUPLICATES
df = df.drop_duplicates(subset='text').reset_index(drop=True)

# Shuffle it so the models don't memorize the order (SEED=42)
df = df.sample(frac=1, random_state=42).reset_index(drop=True)

# final verification
print("="*50)
print("GRAND MERGE SUMMARY")
print("="*50)
print(f"Total Baris Bersih: {len(df)}")
print("\nDistribusi Kelas:")
print(df['status'].value_counts())

=====
GRAND MERGE SUMMARY
=====
Total Baris Bersih: 60428

Distribusi Kelas:
status
depression    15929
normal        14938
anxiety       14809
stress        14752
Name: count, dtype: int64
```

Gambar 20. Merge Dataset

```
X = df['text']
y = df['status']

# Split 1: 80% Train, 20% Temp (for val and test)
X_train, X_temp, y_train, y_temp = train_test_split(
    X, y, test_size=0.2, stratify=y, random_state=42
)

# Split 2: Divide the score into 50% for the written val and 50% for the test (10% each of the total)
X_val, X_test, y_val, y_test = train_test_split(
    X_temp, y_temp, test_size=0.5, stratify=y_temp, random_state=42
)

print("="*50)
print("DATASET SPLIT SUMMARY")
print("="*50)
print(f"Train Set : {len(X_train)} baris")
print(f"Val Set   : {len(X_val)} baris")
print(f"Test Set  : {len(X_test)} baris")

=====
DATASET SPLIT SUMMARY
=====
Train Set : 48342 baris
Val Set   : 6043 baris
Test Set  : 6043 baris
```

Gambar 21. Split Dataset

2. **Pembersihan Teks (Text Cleaning):** Proses normalisasi awal dilakukan dengan menyeragamkan seluruh teks menjadi huruf kecil (*case folding*). Selanjutnya, ekspresi reguler (*Regex*) diaplikasikan untuk secara agresif mengeliminasi atribut noise yang tidak memiliki bobot semantik bagi analisis sentimen, seperti tautan *URL*, sebutan akun pengguna (*mentions*), tagar (*hashtags*), deretan angka, dan karakter spesial. Logika fungsi pembersihan ini juga dirancang untuk berjalan identik pada backend saat proses inferensi (FastAPI) demi mencegah bias format input.

```
# TEXT CLEANING & RE-SPLIT
# This function MUST be identical to the one in the FastAPI backend
def clean_text(text):
    text = str(text).lower()
    text = re.sub(r'http\S+|www\S+', '', text) # Remove URL
    text = re.sub(r'@\w+', '', text) # Remove Mention
    text = re.sub(r'#\w+', '', text) # Remove Hashtag
    text = re.sub(r'\d+', '', text) # Remove Digit
    text = re.sub(r'^\w/s]', '', text) # Remove Special char
    text = re.sub(r'\s+', ' ', text).strip() # Normalize Whitespace
    return text

# Apply cleaning ke dataframe induk
df['text'] = df['text'].apply(clean_text)

# Remove blank lines (if there is text that contains only links or symbols)
df = df[df['text'].str.len() > 0].reset_index(drop=True)

# RE-SPLIT (so that X_train, X_val, and X_test are clean versions)
X = df['text']
y = df['status']

X_train, X_temp, y_train, y_temp = train_test_split(
    X, y, test_size=0.2, stratify=y, random_state=42
)
X_val, X_test, y_val, y_test = train_test_split(
    X_temp, y_temp, test_size=0.5, stratify=y_temp, random_state=42
)

print(f"Post-Cleaning Train Set : {len(X_train)} baris")

Post-Cleaning Train Set : 48342 baris
```

Gambar 22. Cleaning Text

3. **Tokenisasi dan Sequence Padding:** Teks yang telah dibersihkan dipecah menjadi unit token dan direpresentasikan sebagai urutan angka menggunakan fungsi Tokenizer. Untuk mencegah kebocoran informasi (*data leakage*), proses fitting (pembuatan kamus kata) secara ketat hanya diterapkan pada data latih. Panjang urutan dibatasi secara dinamis menggunakan nilai `MAX_LEN` yang merepresentasikan persentil ke-95 dari distribusi panjang token seluruh data. Urutan yang lebih pendek dari `MAX_LEN` akan diseragamkan ukurannya menggunakan teknik post-padding agar kompatibel dengan dimensi input *Embedding layer*.

```

# Count the word lengths in the cleaned dataframe
lengths = df['text'].apply(lambda x: len(str(x).split()))

print(f"Mean : {lengths.mean():.0f}")
print(f"Median : {lengths.median():.0f}")
print(f"P90 : {np.percentile(lengths, 90):.0f}")
print(f"P95 : {np.percentile(lengths, 95):.0f}")
print(f"Max : {lengths.max():.0f}")

MAX_LEN = int(np.percentile(lengths, 95))
print(f"\n[MISSION CRITICAL] MAX_LEN di-set ke: {MAX_LEN}")

... Mean : 33
Median : 24
P90 : 78
P95 : 97
Max : 510

[MISSION CRITICAL] MAX_LEN di-set ke: 97

VOCAB_SIZE = 30000

# STRICT RULE: Fit ONLY on X_train
tokenizer = Tokenizer(num_words=VOCAB_SIZE, oov_token='<OOV>')
tokenizer.fit_on_texts(X_train)

# Transform text to integer sequence
X_train_seq = tokenizer.texts_to_sequences(X_train)
X_val_seq = tokenizer.texts_to_sequences(X_val)
X_test_seq = tokenizer.texts_to_sequences(X_test)

# Padding is placed at the end (post) so that it doesn't dominate the LSTM's initial memory
X_train_pad = pad_sequences(X_train_seq, maxlen=MAX_LEN, padding='post', truncating='post')
X_val_pad = pad_sequences(X_val_seq, maxlen=MAX_LEN, padding='post', truncating='post')
X_test_pad = pad_sequences(X_test_seq, maxlen=MAX_LEN, padding='post', truncating='post')

print(f"Shape X_train_pad : {X_train_pad.shape}")

... Shape X_train_pad : (48342, 97)

```

Gambar 23. Tokenization dan Sequence Padding

4. **Pengkodean Label (Label Encoding):** Tahap akhir prapemrosesan melibatkan transformasi data target yang awalnya berupa nilai kategorikal teks menjadi representasi vektor numerik (*one-hot encoding*). Proses ini menggunakan algoritma LabelEncoder dan `to_categorical` guna menyesuaikan format label dengan komputasi kerugian (*loss calculation*) dan fungsi aktivasi Softmax pada lapisan keluaran klasifikasi.

```

# Label Encoding
le = LabelEncoder()

# FIT MUST BE TRAINED
y_train_enc = to_categorical(le.fit_transform(y_train), num_classes=4)

# VAL & TEST: TRANSFORM ONLY
y_val_enc = to_categorical(le.transform(y_val), num_classes=4)
y_test_enc = to_categorical(le.transform(y_test), num_classes=4)

# Save the mapping for future reference
label_map = dict(enumerate(le.classes_))
print("Label mapping:", label_map)

... Label mapping: {0: 'anxiety', 1: 'depression', 2: 'normal', 3: 'stress'}

```

Gambar 24. Label Encoding

3.2.10. Pemodelan (LSTM)

Tahap pemodelan dalam penelitian ini berfokus pada pengembangan arsitektur *Deep Learning* berbasis *Long Short-Term Memory* (LSTM) untuk mengklasifikasikan kondisi kesehatan mental dari teks. Mengingat teks curhatan sering kali memiliki ketergantungan konteks yang panjang dan urutan kata yang kompleks, model dirancang menggunakan pendekatan Bidirectional agar sistem dapat mempelajari pola informasi baik dari arah depan maupun belakang secara simultan.

Model: "functional"			
Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 97)	0	-
embedding (Embedding)	(None, 97, 128)	3,840,000	input_layer[0][0]
not_equal (NotEqual)	(None, 97)	0	input_layer[0][0]
bidirectional (Bidirectional)	(None, 97, 256)	263,168	embedding[0][0], not_equal[0][0]
dropout (Dropout)	(None, 97, 256)	0	bidirectional[0]...
global_max_pooling... (GlobalMaxPooling1...	(None, 256)	0	dropout[0][0]
dense (Dense)	(None, 64)	16,448	global_max_pooli...
dropout_1 (Dropout)	(None, 64)	0	dense[0][0]
dense_1 (Dense)	(None, 4)	260	dropout_1[0][0]
Total params: 4,119,876 (15.72 MB)			
Trainable params: 4,119,876 (15.72 MB)			
Non-trainable params: 0 (0.00 B)			

Gambar 25. Model Summary

Arsitektur model dibangun menggunakan Keras 3 Functional API dengan rincian lapisan sebagai berikut:

1. **Lapisan *Embedding*:** Berfungsi untuk memetakan setiap token kata ke dalam ruang vektor berdimensi 128 (*embedding dimension*). Lapisan ini dilengkapi dengan parameter `mask_zero=True` yang sangat esensial untuk menginstruksikan model agar mengabaikan nilai *padding* (nol) dalam kalkulasi, sehingga proses komputasi hanya terfokus pada konten teks yang bermakna.
2. **Lapisan *Bidirectional LSTM*:** Inti dari pemodelan ini terdiri dari 128 unit LSTM yang bekerja secara dua arah. Penggunaan

Bidirectional memungkinkan model menangkap konteks kalimat secara lebih utuh, misalnya memahami bagaimana kata di akhir kalimat dapat memengaruhi makna kata di awal. Untuk menjaga stabilitas *training*, diterapkan *recurrent dropout* sebesar 0,1.

3. **Lapisan Global Max Pooling 1D:** Setelah melalui unit LSTM, lapisan ini bertugas mengekstraksi fitur paling dominan dari seluruh urutan teks. Teknik ini membantu model dalam mengidentifikasi sinyal emosi paling kuat (kata kunci emosional) yang tersebar di dalam dokumen.
4. **Lapisan Dense dan Dropout:** Model dilengkapi dengan satu lapisan tersembunyi (*fully connected*) sebanyak 64 neuron dengan aktivasi ReLU untuk mempelajari hubungan non-linear. Dua lapisan *dropout* (masing-masing 0,3 dan 0,5) disisipkan untuk meminimalisir risiko *overfitting* dengan cara menonaktifkan neuron secara acak selama pelatihan.
5. **Lapisan Output:** Menggunakan lapisan *Dense* dengan 4 unit yang mewakili kategori target (*Anxiety, Depression, Normal, Stress*) dan fungsi aktivasi *Softmax* untuk menghasilkan distribusi probabilitas kelas.

```
# TRAINING PROTOCOL

# Compile Model
model.compile(
    optimizer=Adam(learning_rate=1e-3),
    loss='categorical_crossentropy', # Since our labels are one-hot
    metrics=['accuracy']
)

# Setup Smart Callbacks
callbacks = [
    # If val_loss doesn't decrease over 5 epochs, stop. Don't waste time and resources.
    EarlyStopping(
        monitor='val_loss',
        patience=5,
        restore_best_weights=True,
        verbose=1
    ),
    # If the model starts to plateau at a local minimum (patience=3), cut the learning rate in half.
    ReduceLROnPlateau(
        monitor='val_loss',
        factor=0.5,
        patience=3,
        min_lr=1e-6,
        verbose=1
    ),
    ModelCheckpoint(
        filepath='mentalku_best_model.keras',
        monitor='val_loss',
        save_best_only=True,
        verbose=1
    )
]

# Execute Training
print("Mulai training... Bakal makan waktu, pantau val_loss-nya.")
history = model.fit(
    X_train_pad, y_train_enc,
    validation_data=(X_val_pad, y_val_enc),
    epochs=50,
    batch_size=64, # Sweet spot antara training speed & gradient stability
    callbacks=callbacks
)
```

Gambar 26. Proses Pelatihan Model

Proses pelatihan model dilakukan menggunakan *optimizer* Adam dengan parameter kerugian (*loss function*) berupa *categorical_crossentropy*. Pelatihan dikonfigurasi dengan *batch size* sebanyak 64 dan total 50 *epoch*. Namun, untuk menjaga integritas model dan mencegah penurunan performa pada data baru, diterapkan mekanisme *Early Stopping* dengan *patience* 5.

```
Epoch 1/50
756/756 - 0s 681ms/step - accuracy: 0.7366 - loss: 0.6935
Epoch 1: val_loss improved from None to 0.40250, saving model to mentalku_best_model.keras
Epoch 1: finished saving model to mentalku_best_model.keras
756/756 - 540s 700ms/step - accuracy: 0.8126 - loss: 0.5174 - val_accuracy: 0.8610 - val_loss: 0.4025 - learning_rate: 0.0010
Epoch 2/50
756/756 - 0s 683ms/step - accuracy: 0.8914 - loss: 0.3189
Epoch 2: val_loss improved from 0.40250 to 0.38324, saving model to mentalku_best_model.keras
Epoch 2: finished saving model to mentalku_best_model.keras
756/756 - 530s 700ms/step - accuracy: 0.8970 - loss: 0.2969 - val_accuracy: 0.8615 - val_loss: 0.3832 - learning_rate: 0.0010
Epoch 3/50
756/756 - 0s 686ms/step - accuracy: 0.9247 - loss: 0.2231
Epoch 3: val_loss did not improve from 0.38324
756/756 - 532s 704ms/step - accuracy: 0.9295 - loss: 0.2119 - val_accuracy: 0.8554 - val_loss: 0.4144 - learning_rate: 0.0010
Epoch 4/50
756/756 - 0s 691ms/step - accuracy: 0.9453 - loss: 0.1622
Epoch 4: val_loss did not improve from 0.38324
756/756 - 543s 718ms/step - accuracy: 0.9488 - loss: 0.1540 - val_accuracy: 0.8512 - val_loss: 0.4540 - learning_rate: 0.0010
Epoch 5/50
756/756 - 0s 693ms/step - accuracy: 0.9607 - loss: 0.1173
Epoch 5: ReduceLROnPlateau reducing learning rate to 0.00050000000237487257.
Epoch 5: val_loss did not improve from 0.38324
756/756 - 545s 721ms/step - accuracy: 0.9627 - loss: 0.1133 - val_accuracy: 0.8552 - val_loss: 0.4962 - learning_rate: 0.0010
Epoch 6/50
756/756 - 0s 690ms/step - accuracy: 0.9736 - loss: 0.0789
Epoch 6: val_loss did not improve from 0.38324
756/756 - 535s 700ms/step - accuracy: 0.9773 - loss: 0.0688 - val_accuracy: 0.8539 - val_loss: 0.5764 - learning_rate: 5.0000e-04
Epoch 7/50
756/756 - 0s 689ms/step - accuracy: 0.9851 - loss: 0.0496
Epoch 7: val_loss did not improve from 0.38324
756/756 - 534s 706ms/step - accuracy: 0.9860 - loss: 0.0453 - val_accuracy: 0.8512 - val_loss: 0.6609 - learning_rate: 5.0000e-04
Epoch 7: early stopping
Restoring model weights from the end of the best epoch: 2.
```

Gambar 27. Hasil Pelatihan

Berdasarkan hasil pelatihan, proses dihentikan secara otomatis pada *epoch* ke-7 dengan mengembalikan bobot terbaik dari *epoch* ke-2 untuk memastikan model tetap memiliki kemampuan generalisasi yang optimal.

3.2.11. Logika Kalkulasi DASS-21

Modul *dass21.py* dalam sistem ini dirancang untuk memproses data kognitif terstruktur yang dikirimkan melalui antarmuka pengguna. Berbeda dengan urutan butir soal pada instrumen DASS-21 standar yang bersifat acak, sistem ini mengimplementasikan pengelompokan soal berdasarkan kategori emosi pada setiap halaman formulir untuk meningkatkan pengalaman pengguna (User Experience).

Data jawaban yang diterima oleh API berupa sebuah array numerik tunggal berisi 21 elemen. Proses kalkulasi dilakukan dengan menggunakan teknik *array slicing* untuk memisahkan skor berdasarkan kelompok kategorinya:

1. **Stres (*Stress*):** Diambil dari indeks `answers[0:7]`, yang merepresentasikan jawaban dari soal nomor 1 hingga 7 pada halaman pertama antarmuka pengguna.
2. **Kecemasan (*Anxiety*):** Diambil dari indeks `answers[7:14]`, yang merepresentasikan jawaban dari soal nomor 8 hingga 14 pada halaman kedua.
3. **Depresi (*Depression*):** Diambil dari indeks `answers[14:21]`, yang merepresentasikan jawaban dari soal nomor 15 hingga 21 pada halaman ketiga.

```
def process_dass21(answers: List[int]):
    """
    Processes a list of 21 DASS-21 answers (0-3 scale).
    Returns a dictionary containing scores, levels for each dimension, average score, and final level.
    """
    if len(answers) != 21:
        raise ValueError("Jumlah jawaban DASS-21 harus tepat 21.")

    # Slice array & calculate scores (DASS-21 scores are multiplied by 2 to match DASS-42 scale)
    # UI input order: 1-7 (Stress), 8-14 (Anxiety), 15-21 (Depression)
    stress_score = sum(answers[0:7]) * 2
    anxiety_score = sum(answers[7:14]) * 2
    depresi_score = sum(answers[14:21]) * 2

    # Category determination logic based on DASS-42 scale
    dass21_scale = {
        "Depresi": [(0, 9, "Normal"), (10, 13, "Ringan"), (14, 20, "Sedang"), (21, 27, "Parah"), (28, 100, "Sangat Parah")],
        "Kecemasan": [(0, 7, "Normal"), (8, 9, "Ringan"), (10, 14, "Sedang"), (15, 19, "Parah"), (20, 100, "Sangat Parah")],
        "Stres": [(0, 14, "Normal"), (15, 18, "Ringan"), (19, 25, "Sedang"), (26, 33, "Parah"), (34, 100, "Sangat Parah")]
    }

    def get_level(score, category):
        for min_val, max_val, label in dass21_scale[category]:
            if min_val <= score <= max_val:
                return label
        return "Tidak Diketahui"

    # Calculate average and final overall status
    rata_rata = (depresi_score + anxiety_score + stress_score) / 3
    if rata_rata < 10:
        final_level = "Normal"
    elif rata_rata < 14:
        final_level = "Ringan"
    elif rata_rata < 20:
        final_level = "Sedang"
    elif rata_rata < 27:
        final_level = "Parah"
    else:
        final_level = "Sangat Parah"

    return {
        "scores": {
            "depression": depresi_score,
            "anxiety": anxiety_score,
            "stress": stress_score
        },
        "levels": {
            "depression": get_level(depresi_score, "Depresi"),
            "anxiety": get_level(anxiety_score, "Kecemasan"),
            "stress": get_level(stress_score, "Stres")
        },
        "average_score": round(rata_rata, 2),
        "final_level": final_level
    }
```

Gambar 28. Proses Perhitungan Skor Kuisioner menggunakan instrumen Dass-21

Setelah proses slicing dilakukan, sistem menghitung total skor pada masing-masing kategori dan menerapkan faktor pengali dua (x2). Langkah ini bersifat wajib guna menyetarakan hasil DASS-21 (versi singkat) dengan skala DASS-42 untuk menentukan tingkat keparahan (severity level) secara akurat sesuai dengan standar literatur psikologi. Skor akhir dari ketiga dimensi tersebut kemudian dibandingkan untuk menentukan variabel `dass_dominant`

sebagai indikator gangguan emosional yang paling menonjol sebelum diteruskan ke mesin rekomendasi berbasis AI.

3.2.12. Analisis Curhat

Sub-subbab ini menguraikan mekanisme internal modul `emotion.py` yang bertugas memproses input teks tidak terstruktur (curhatan) dari pengguna menjadi prediksi label kondisi mental secara *real-time*. Proses inferensi ini dirancang dengan prinsip konsistensi prapemrosesan guna menjamin integritas prediksi model Bidirectional LSTM yang telah dilatih sebelumnya. Alur kerja analisis teks ini mencakup empat tahapan utama:

1. **Normalisasi Input (*Inference Preprocessing*):** Teks mentah yang diterima melalui *endpoint* API terlebih dahulu melewati fungsi `clean_text`. Proses ini menerapkan ekspresi reguler (*Regex*) untuk mengeliminasi atribut non-semantik seperti URL, *mentions*, dan karakter spesial, serta melakukan *case folding*. Tahapan ini bersifat wajib untuk memastikan format teks input identik dengan karakteristik data pada fase pelatihan, sehingga model dapat mengenali fitur linguistik secara optimal.
2. **Vektorisasi Teks:** Setelah melalui tahap pembersihan, teks dikonversi menjadi urutan numerik (*integer sequences*) menggunakan objek *Tokenizer* yang telah dimuat ke dalam memori server. Proses ini memetakan setiap kata ke dalam indeks unik berdasarkan kosakata (*vocabulary*) yang telah dibentuk pada tahap *training*.
3. **Transformasi Dimensi (*Padding*):** Mengingat panjang curhatan pengguna bersifat variatif, sistem menerapkan fungsi `pad_sequences` dengan parameter `maxlen=MAX_LEN` dan teknik *post-padding*. Langkah ini bertujuan untuk menyeragamkan dimensi input teks agar sesuai dengan bentuk (*shape*) lapisan masukan (*Input Layer*) pada model LSTM, yaitu berupa tensor satu dimensi dengan panjang tetap.
4. **Inferensi Model dan Ekstraksi Probabilitas:** Data yang telah terformat kemudian diumpankan ke model untuk dilakukan proses

kalkulasi maju (*forward pass*). Model menghasilkan output berupa distribusi probabilitas untuk empat kelas kategori (*Anxiety, Depression, Normal, Stress*). Sistem menggunakan fungsi `np.argmax()` untuk menentukan indeks dengan nilai kepercayaan tertinggi, yang kemudian dikonversi kembali menjadi label teks menggunakan modul `LabelEncoder` (fungsi `inverse_transform`).

```
def clean_text_v1(text: str) -> str:
    """
    Zero-Destruction Preprocessing Protocol.
    100% identical to the clean_text function used during model training in Google Colab.
    """
    text = str(text).Lower()
    text = re.sub(r'http\S+|www\S+', '', text) # Remove URL
    text = re.sub(r'@\w+', '', text) # Remove Mention
    text = re.sub(r'#\w+', '', text) # Remove Hashtag
    text = re.sub(r'\d+', '', text) # Remove Digit
    text = re.sub(r'[\W\s]', '', text) # Remove Special char
    text = re.sub(r'\s+', ' ', text).strip() # Normalize Whitespace
    return text

def predict_emotion(text: str, model, tokenizer, label_encoder, max_sequence_length: int = 97) -> str:
    """
    Keras 3 Model Inference.
    Configured with MAX_LEN = 97, padding='post', and truncating='post'.
    """
    # Fallback for empty input
    if not text or not text.strip():
        return "normal"

    # 1. Text cleaning using training pipeline protocol
    cleaned_text = clean_text_v1(text)

    # 2. Tokenize text into sequence
    sequence = tokenizer.texts_to_sequences([cleaned_text])

    # 3. Padding (Must use 'post' as per architectural blueprint)
    padded = pad_sequences(
        sequence,
        maxlen=max_sequence_length,
        padding='post',
        truncating='post'
    )

    # 4. Keras 3 Prediction
    probabilities = model.predict(padded, verbose=0)[0]
    predicted_index = np.argmax(probabilities)

    # 5. Decode predicted index to string label (e.g., 'depression', 'stress')
    predicted_label = label_encoder.inverse_transform([predicted_index])[0]

    return predicted_label
```

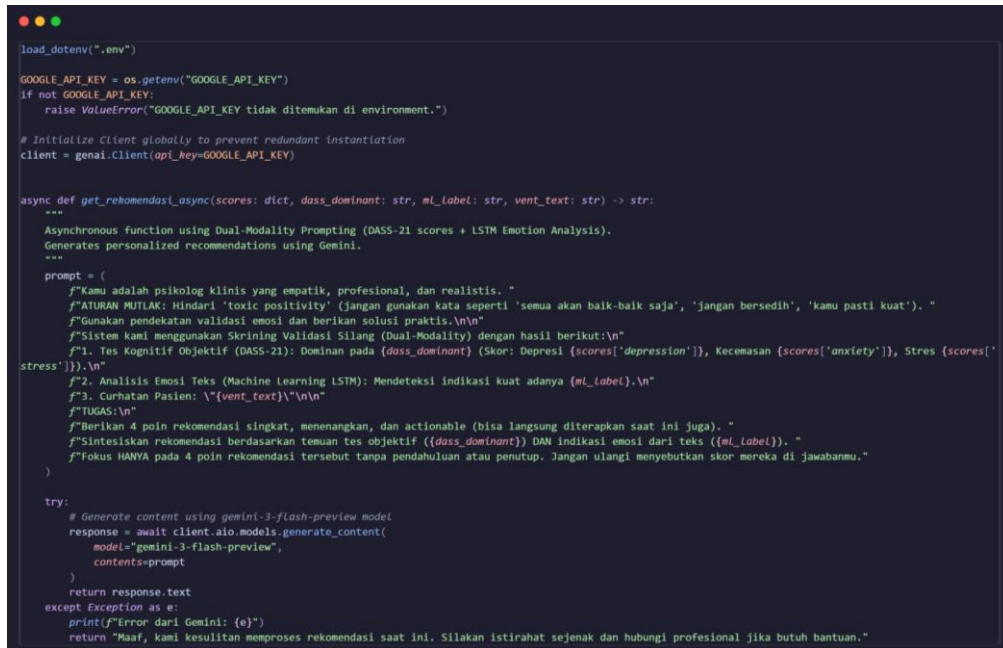
Gambar 29. Proses Analisis Teks Curhat menggunakan models yang telah dikembangkan

Hasil akhir dari tahap ini adalah sebuah label kategorikal (`ml_label`) dan data distribusi probabilitas yang memberikan gambaran mengenai tingkat keyakinan model terhadap prediksi tersebut. Output ini nantinya akan digabungkan dengan hasil tes DASS-21 untuk memberikan analisis kesehatan mental yang bersifat komprehensif.

3.2.13. Sistem Rekomendasi Berbasis LLM (Gemini AI)

Tahap akhir dari alur pemrosesan (*pipeline*) sistem Mentalku adalah pembangkitan solusi dan rekomendasi personal menggunakan *Large Language*

Model (LLM) dari Google Gemini. Modul rekomendasi.py bertindak sebagai mesin sintesis yang menggabungkan hasil evaluasi kognitif terstruktur (DASS-21) dengan hasil analisis sentimen tidak terstruktur (LSTM). Penggabungan dua modalitas ini (*Dual-Modality Screening*) memungkinkan AI untuk memahami kondisi pengguna secara holistik sebelum memberikan intervensi dasar.



```

load_dotenv(".env")

GOOGLE_API_KEY = os.getenv("GOOGLE_API_KEY")
if not GOOGLE_API_KEY:
    raise ValueError("GOOGLE_API_KEY tidak ditemukan di environment.")

# Initialize Client globally to prevent redundant instantiation
client = genai.Client(api_key=GOOGLE_API_KEY)

async def get_rekomendasi_async(scores: dict, dass_dominant: str, ml_label: str, vent_text: str) -> str:
    """
    Asynchronous function using Dual-Modality Prompting (DASS-21 scores + LSTM Emotion Analysis).
    Generates personalized recommendations using Gemini.
    """
    prompt = (
        f"Kamu adalah psikolog klinis yang empatik, profesional, dan realistis. "
        f"ATURAN MUTLAK: Hindari 'toxic positivity' (jangan gunakan kata seperti 'semua akan baik-baik saja', 'jangan bersedih', 'kamu pasti kuat'). "
        f"Gunakan pendekatan validasi emosi dan berikan solusi praktis.\n\n"
        f"Sistem kami menggunakan Skrining Validasi Silang (Dual-Modality) dengan hasil berikut:\n"
        f"1. Tes Kognitif Objektif (DASS-21): Dominan pada {dass_dominant} (Skor: Depresi {scores['depression']}, Kecemasan {scores['anxiety']}, Stres {scores['stress']}).\n\n"
        f"2. Analisis Emosi Teks (Machine Learning LSTM): Mendeteksi indikasi kuat adanya {ml_label}.\n"
        f"3. Curhatan Pasien: \"{vent_text}\".\n\n"
        f"TUGAS:\n"
        f"Berikan 4 poin rekomendasi singkat, menenangkan, dan actionable (bisa langsung diterapkan saat ini juga). "
        f"Sintesis rekomendasi berdasarkan temuan tes objektif ({dass_dominant}) DAN indikasi emosi dari teks ({ml_label}). "
        f"Fokus HANYA pada 4 poin rekomendasi tersebut tanpa pendahuluan atau penutup. Jangan ulangi menyebutkan skor mereka di jawabanmu."
    )

    try:
        # Generate content using gemini-3-flash-preview model
        response = await client.aio.models.generate_content(
            model="gemini-3-flash-preview",
            contents=prompt
        )
        return response.text
    except Exception as e:
        print(f"Error dari Gemini: {e}")
        return "Maaf, kami kesulitan memproses rekomendasi saat ini. Silakan istirahat sejenak dan hubungi profesional jika butuh bantuan."

```

Gambar 30. Rekomendasi Personalisasi menggunakan Gemini 3 Flash

Kinerja modul ini sangat bergantung pada teknik rekayasa *prompt* (*Prompt Engineering*) yang dirancang secara spesifik dengan parameter berikut:

1. **Penentuan Persona dan Gaya Bahasa:** LLM diinstruksikan untuk bertindak sebagai psikolog klinis dengan tingkat empati tinggi. Untuk membangun kedekatan emosional (*rapport*) dengan target demografi sistem, bahasa keluaran disesuaikan menjadi bahasa Indonesia kasual namun tetap profesional.
2. **Penerapan *Guardrails* (Batasan Keselamatan):** Sebuah instruksi kritis ditambahkan untuk melarang keras penggunaan kalimat *toxic positivity* (seperti "semua akan baik-baik saja" atau "jangan bersedih"). Pendekatan yang diwajibkan adalah validasi emosi,

sehingga pengguna merasa didengar dan diakui perasaannya tanpa dihakimi.

3. **Injeksi Konteks (*Context Injection*):** LLM tidak bekerja dari ruang hampa. Tiga elemen data utama disuntikkan ke dalam *prompt*: (a) label dominan dan skor absolut dari DASS-21, (b) label emosi hasil prediksi LSTM, dan (c) teks curhatan asli dari pengguna.
4. **Kendali Format Keluaran (*Output Formatting*):** Guna menjaga konsistensi antarmuka pengguna, LLM dibatasi untuk hanya menghasilkan tepat empat poin rekomendasi. Solusi yang diberikan harus bersifat *actionable* (dapat langsung diterapkan), singkat, dan menenangkan, tanpa menyertakan kalimat pembuka, penutup, maupun pengulangan skor teknis.

Secara teknis, pemanggilan API ke layanan Gemini dilakukan menggunakan metode asinkron (*async/await*). Implementasi ini sangat esensial pada arsitektur API (*FastAPI*) guna mencegah pemblokiran *thread (blocking)*, sehingga server tetap responsif dalam melayani permintaan (*requests*) dari pengguna lain secara bersamaan.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1. Hasil Implementasi API dan *Machine Learning*

4.1.1. Evaluasi Model

Setelah proses pelatihan selesai, performa model diuji menggunakan himpunan data uji (*test set*) yang belum pernah dilihat sebelumnya oleh model selama proses *training*. Tahap evaluasi ini menggunakan metrik obyektif berupa *Classification Report* yang mencakup nilai *Precision*, *Recall*, dan *F1-Score*, serta visualisasi *Confusion Matrix* untuk melihat distribusi prediksi pada setiap kelas.

Perhitungan performa model secara komprehensif didasarkan pada nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) yang diekstrak dari *confusion matrix*. Metrik utama yang digunakan adalah akurasi (*accuracy*), yang mengukur persentase ketepatan model dalam memprediksi keseluruhan kelas dengan membagi jumlah prediksi benar (diagonal utama pada matriks) dengan total seluruh observasi data uji. Secara matematis, persamaan akurasi dirumuskan sebagai berikut:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Selain akurasi, *classification report* juga diukur menggunakan tiga metrik evaluasi spesifik untuk menangani karakteristik klasifikasi multikelas. *Precision* digunakan untuk mengukur rasio ketepatan prediksi positif, *Recall* untuk mengukur sensitivitas model dalam menemukan kembali seluruh data yang relevan, serta *F1-Score* yang merupakan rata-rata harmonik dari keduanya untuk menunjukkan keseimbangan performa model. Ketiga metrik tersebut dihitung menggunakan formulasi matematis berikut:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Berdasarkan hasil pengujian pada data uji, model Bidirectional LSTM berhasil mencapai tingkat akurasi keseluruhan sebesar 86%. Capaian ini menunjukkan bahwa model memiliki kemampuan generalisasi yang sangat baik dalam mengenali pola linguistik terkait kondisi kesehatan mental. Rincian performa untuk setiap label emosi adalah sebagai berikut:

```

=====
CLASSIFICATION REPORT
=====

```

	precision	recall	f1-score	support
anxiety	0.85	0.85	0.85	1481
depression	0.90	0.82	0.86	1593
normal	0.87	0.93	0.90	1494
stress	0.81	0.84	0.82	1475
accuracy			0.86	6043
macro avg	0.86	0.86	0.86	6043
weighted avg	0.86	0.86	0.86	6043

```

=====
CONFUSION MATRIX
=====
[[1253  41  46 141]
 [ 70 1311 114  98]
 [ 16  35 1393  50]
 [ 131  63  49 1232]]

```

Gambar 31. Hasil Evaluasi Model

1. **Normal:** Kategori ini mendapatkan hasil tertinggi dengan *F1-Score* sebesar 0,90 dan *recall* 0,93. Hal ini menunjukkan model sangat efektif dalam mengidentifikasi kondisi sehat/normal tanpa banyak melakukan kesalahan prediksi.
2. **Depression (Depresi):** Kategori ini mencatatkan tingkat *precision* tertinggi sebesar 0,90. Artinya, ketika model memprediksi seseorang mengalami depresi, tingkat kebenaran predisinya sangat akurat, meskipun nilai *recall* (0,82) menunjukkan masih adanya ruang pengembangan untuk menangkap variasi teks depresi yang lebih luas.
3. **Anxiety (Kecemasan) & Stress (Stres):** Kedua kategori ini menunjukkan performa yang konsisten dengan *F1-Score* masing-masing sebesar 0,85 dan 0,83.

Keseimbangan antara nilai macro average dan weighted average yang sama-sama berada di angka 0,86 membuktikan bahwa model tidak mengalami bias terhadap salah satu kelas mayoritas. Hal ini memvalidasi efektivitas strategi penyeimbangan data (stratified split) yang dilakukan pada tahap awal. Pengujian inferensi pada data uji juga menunjukkan waktu pemrosesan yang efisien, yakni sekitar 146ms per batch, yang menandakan model siap untuk diintegrasikan ke dalam lingkungan produksi (API) dengan latensi yang rendah.

4.1.2. Artefak Model untuk Deployment

Setelah model *Bidirectional LSTM* selesai dievaluasi dan mencapai metrik performa yang ditargetkan, tahapan krusial selanjutnya adalah melakukan ekstraksi dan penyimpanan (*saving*) artefak pembelajaran mesin. Proses ini bertujuan agar model yang telah dilatih dapat diintegrasikan (*deployment*) ke dalam arsitektur *Application Programming Interface* (API) Mentalku.

```
# save the model in .keras
model.save('mentalku_best_model.keras')

# save Tokenizer (Mapping Teks -> Integer)
with open('tokenizer_lstm.pkl', 'wb') as f:
    pickle.dump(tokenizer, f)

# save Label Encoder (Mapping Integer -> Label)
with open('label_encoder.pkl', 'wb') as f:
    pickle.dump(le, f)

# save Config
with open('config.pkl', 'wb') as f:
    pickle.dump({'MAX_LEN': MAX_LEN, 'VOCAB_SIZE': VOCAB_SIZE}, f)

print("✅ MISSION ACCOMPLISHED. Semua artefak siap di-deploy:")
print("  - mentalku_best_model.keras")
print("  - tokenizer_lstm.pkl")
print("  - label_encoder.pkl")
print("  - config.pkl")

... ✅ MISSION ACCOMPLISHED. Semua artefak siap di-deploy:
    - mentalku_best_model.keras
    - tokenizer_lstm.pkl
    - label_encoder.pkl
    - config.pkl
```

Gambar 32. Ekstraksi dan Penyimpanan Models

Pada tahap ini, sistem tidak hanya menyimpan struktur *neural network*, tetapi juga menyimpan seluruh instrumen prapemrosesan (*preprocessing*) agar data teks baru yang masuk dari pengguna dapat diproses dengan standarisasi yang identik dengan fase pelatihan. Seluruh artefak tersebut diekspor menjadi empat komponen utama, yaitu:

1. Model Utama (`mentalku_best_model.keras`)

Merupakan representasi inti dari model *Deep Learning* yang berisi arsitektur jaringan *Bidirectional LSTM* beserta bobot (*weights*) terbaik yang didapatkan dari proses pelatihan. Format `.keras` digunakan sebagai format standar terbaru yang lebih efisien dalam menyimpan topologi model dan status *optimizer*.

2. Objek Tokenizer (`tokenizer_lstm.pkl`)

Artefak ini berfungsi untuk memetakan teks input baru ("curhatan" pengguna) menjadi sekuens angka (*integer*) yang dapat dipahami oleh model. Menyimpan objek *tokenizer* bersifat wajib agar tidak terjadi ketidaksesuaian indeks leksikon (*out-of-vocabulary*) antara kata yang dilatih dengan kata yang diinferensi di sistem API.

3. Objek Label Encoder (`label_encoder.pkl`)

Berfungsi sebagai mesin penerjemah *output* model. Karena model LSTM mengeluarkan prediksi dalam bentuk probabilitas numerik (misal: 0, 1, 2, 3), objek ini bertugas memetakan kembali angka tersebut menjadi label kelas emosi manusia yang dapat dibaca, seperti *Normal*, *Depression*, *Anxiety*, atau *Stress*.

4. Konfigurasi Eksekusi (`config.pkl`)

Menyimpan hiperparameter global yang bersifat statis, seperti nilai panjang maksimal sekuens teks (`MAX_LEN`) dan ukuran kosakata (`VOCAB_SIZE`). Hal ini memastikan logika pemotongan atau penambahan teks (*padding/truncating*) pada saat *deployment* selaras dengan dimensi vektor yang diharapkan oleh model.

4.1.3. Model Inference Testing

Setelah seluruh artefak model diekstraksi, tahap validasi akhir sebelum integrasi ke dalam API adalah melakukan pengujian inferensi (*inference testing*). Pengujian ini bertujuan untuk menyimulasikan alur kerja sistem secara *end-to-end* dalam merespons data teks mentah (kasus nyata) yang belum pernah diproses sebelumnya.

```

print("loading deployment artifacts...")

# 1. Load all exported artifacts
model_infer = load_model('mentaliku_best_model.keras')
tokenizer_infer = pickle.load(open('tokenizer_lstm.pkl', 'rb'))
le_infer = pickle.load(open('label_encoder.pkl', 'rb'))
config_infer = pickle.load(open('config.pkl', 'rb'))

MAX_LEN_INFER = config_infer['MAX_LEN']

# 2. Text Preprocessing Function (MUST BE STRICTLY IDENTICAL to Phase 2)
def clean_text_infer(text: str) -> str:
    text = str(text).lower()
    text = re.sub(r'http[s]?://', '', text)
    text = re.sub(r'@', '', text)
    text = re.sub(r'#', '', text)
    text = re.sub(r'\d+', '', text)
    text = re.sub(r'%', '', text)
    text = re.sub(r'$', '', text)
    text = re.sub(r'^', '', text)
    text = re.sub(r'$', '', text)
    return text

# 3. Core Inference Engine
def predict_mental_health(text: str) -> dict:
    # Pipeline: Clean -> Tokenize -> Pad Sequence
    cleaned = clean_text_infer(text)
    seq = tokenizer_infer.texts_to_sequences([cleaned])
    padded = pad_sequences(seq, maxlen=MAX_LEN_INFER, padding='post')

    # Execute prediction (returns an array of probabilities)
    probs = model_infer.predict(padded, verbose=0)[0]

    # Map predicted probabilities to class labels
    result = {}
    for label, prob in zip(le_infer.classes_, probs):
        result[label] = round(float(prob) * 100, 2)

    # Extract the class with the highest probability
    predicted = max(result, key=result.get)
    return {"predicted": predicted, "probabilities": result}

# 4. REAL-WORLD INFERENCE TESTING (Extreme Cases)
test_cases = [
    "Gue ngerasa kosong dan ga ada gunanya hidup lagi. Tiap hari rasanya berat banget buat bangun dari kasur, semuanya hancur.",
    "Jantung gue deg-degan kencang banget tiap mikirin masa depan. Dada rasanya sesak dan gue takut hal buruk bakal kejadian.",
    "Kerjaan numpuk ga ada habisnya, kepala gue mau meledak rasanya dikejer deadline tiap jam. Bener-bener mauk dan capeki!"
]

print("\n" + "="*50)
print("INFERENCE TEST RESULTS")
print("\n" + "="*50)

for text in test_cases:
    out = predict_mental_health(text)
    print(f"Text : {text}")
    print(f"Result : {out['predicted'].upper()}")
    print(f"Probs : {out['probabilities']}")

```

Gambar 33. Proses Model Inference Testing

Proses inferensi dirancang dengan arsitektur *pipeline* yang ketat guna memastikan konsistensi pemrosesan data. Terdapat tiga tahapan utama dalam *pipeline* inferensi ini:

1. **Pemuatan Artefak (Artifact Loading):** Menginisialisasi kembali model utama, objek *tokenizer*, *label encoder*, serta variabel konfigurasi dimensi secara luring dari *file* artefak yang telah diekspor.
2. **Prapemrosesan Teks Identik:** Teks masukan ("curhatan") melewati fungsi pembersihan teks (*cleaning*) yang secara ketat disamakan dengan fase pelatihan. Konsistensi ini krusial agar tidak terjadi pergeseran distribusi data (*data drift*) yang dapat merusak akurasi prediksi.
3. **Komputasi Probabilitas:** Teks yang telah dibersihkan dikonversi menjadi token matriks (*padding sequence*) dan diumpankan ke model LSTM untuk menghasilkan distribusi probabilitas (skala 0 hingga 100) terhadap empat kelas emosi (Normal, *Depresi*, *Anxiety*, dan *Stress*).

```

Loading deployment artifacts...

=====
INFERENCE TEST RESULTS
=====
Text : 'Gue ngerasa kosong dan ga ada gunanya hidup lagi. Tiap hari rasanya berat banget buat bangun dari kasur, semuanya hancur.'
Result : DEPRESSION
Probs : {'anxiety': 6.7, 'depression': 55.44, 'normal': 3.4, 'stress': 34.47}

Text : 'Jantung gue deg-degan kenceng banget tiap mikirin masa depan. Dada rasanya sesak dan gue takut hal buruk bakal kejadian.'
Result : ANXIETY
Probs : {'anxiety': 97.48, 'depression': 0.79, 'normal': 0.17, 'stress': 1.55}

Text : 'Kerjaan numpuk ga ada habisnya, kepala gue mau meledak rasanya dikejar deadline tiap jam. Bener-bener muak dan capek!'
Result : STRESS
Probs : {'anxiety': 4.77, 'depression': 2.33, 'normal': 0.35, 'stress': 92.55}

```

Gambar 34. Output Model Inference Testing

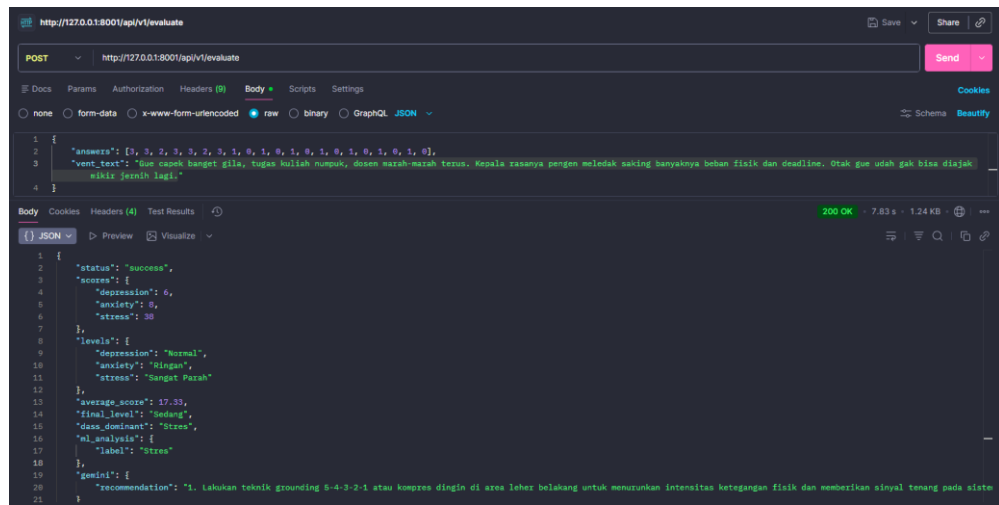
Untuk menguji keandalan sistem, pengujian dilakukan menggunakan tiga kasus (*test cases*) dengan hasil sebagai berikut:

- **Kasus Uji 1 (Kelelahan Emosional & Keputusanasaan):** Teks "*Gue ngerasa kosong dan ga ada gunanya hidup lagi...*" berhasil diprediksi oleh model sebagai **Depression** dengan tingkat keyakinan tertinggi sebesar 55,44%. Menariknya, model juga mendeteksi indikasi *Stress* sebesar 34,47%. Hal ini membuktikan bahwa model mampu menangkap nuansa komorbiditas (gejala yang saling beririsan) pada teks pengguna secara logis.
- **Kasus Uji 2 (Serangan Panik Fisiologis):** Teks "*Jantung gue deg-degan kenceng banget... Dada rasanya sesak...*" diprediksi secara sangat akurat sebagai **Anxiety** dengan probabilitas absolut sebesar 97,48%. Model terbukti sanggup mengenali leksikon yang berkaitan dengan respons fisik (psikosomatis) khas kecemasan.
- **Kasus Uji 3 (Tekanan Eksternal & *Burnout*):** Teks "*Kerjaan numpuk ga ada habisnya, kepala gue mau meledak...*" diklasifikasikan dengan probabilitas tinggi sebesar 92,55% sebagai **Stress**. Pemahaman model terhadap kata kunci tekanan tenggat waktu (*deadline*) dan beban kerja tervalidasi dengan baik.

4.1.4. Integrasi Sistem dan Antarmuka Pemrograman Aplikasi (API)

Tahapan integrasi merupakan proses krusial yang menghubungkan seluruh komponen logika pada backend Python dengan antarmuka pengguna pada *Framework* Laravel. Integrasi ini dibangun menggunakan arsitektur Representational State Transfer (REST) API dengan FastAPI sebagai penyedia

layanan karena keunggulannya dalam menangani permintaan asinkronus dan latensi yang rendah.



Gambar 35. Pengujian Fungsionalitas Endpoint API Evaluasi Diri menggunakan Postman

1. Arsitektur Endpoint FastAPI (app.py)

Layanan utama pada sisi Python dikemas dalam satu *endpoint* POST bernama `/predict`. Penggunaan metode POST sangat krusial karena sistem harus mengirimkan data yang bersifat sensitif dan kompleks, yaitu larik (*array*) jawaban DASS-21 dan teks curhatan pengguna. Untuk menjamin integritas data yang dikirim oleh Laravel, sistem menerapkan skema validasi menggunakan *Pydantic* melalui kelas *EvaluationRequest*. Skema ini memastikan bahwa setiap permintaan yang masuk wajib menyertakan:

- **answers:** Daftar 21 nilai integer dari kuesioner.
- **vent:** String teks berisi narasi curhatan pengguna.

Setiap kali *endpoint* ini dipanggil, sistem secara otomatis menjalankan *pipeline* berantai: kalkulasi DASS-21, inferensi model LSTM, hingga sintesis rekomendasi melalui Gemini AI, dan mengembalikannya dalam satu objek JSON terstruktur.

2. Konsumsi API pada Laravel (EvaluationController)

Pada sisi klien, Laravel berperan sebagai orkestrator yang mengelola alur data pengguna. Komunikasi dilakukan melalui *HTTP Client* Laravel yang mengirimkan *payload* JSON ke server Python. Proses integrasi pada *EvaluationController* dirancang dengan mekanisme *error handling* yang ketat guna menjaga stabilitas aplikasi:

- **Validasi Respons:** Sistem mengecek status kode HTTP dari API. Jika sukses (200 OK), data hasil analisis didekode dan disimpan ke dalam basis data MySQL melalui model *Result*.
- **Ketahanan Sistem (System Resilience):** Jika server Python dalam kondisi *offline* atau terjadi kegagalan koneksi, Laravel akan menangkap eksepsi (*catch exception*) dan memberikan pesan *feedback* yang informatif kepada pengguna, alih-alih menampilkan galat sistem yang fatal.

3. Struktur Request-Response JSON

Keberhasilan integrasi ini ditandai dengan pertukaran data yang bersih (*clean data exchange*). Berikut adalah abstraksi dari struktur data yang dikirim dan diterima:

- **Request Payload:** Berisi *answers* (integer array) dan *vent_text* (string).
- **Response Payload:** Mengembalikan metadata lengkap yang mencakup skor absolut DASS-21, level keparahan tiap kategori, label prediksi emosi dari LSTM beserta probabilitasnya, serta narasi rekomendasi final dari Gemini.

Pemisahan tanggung jawab (*separation of concerns*) antara Laravel untuk manajemen sesi/UI dan FastAPI untuk komputasi AI ini membuat sistem *Mentalku* menjadi lebih modular, mudah dipelihara, dan memiliki skalabilitas yang baik.

4.2. Hasil Implementasi Aplikasi Web (Mentalku-Web)

Subbab ini menguraikan hasil implementasi perangkat lunak pada sisi klien (*client-side*) dari sistem Mentalku. Aplikasi web ini dikembangkan menggunakan *Framework* Laravel 12 yang bertindak sebagai pusat pengelolaan antarmuka pengguna, manajemen sesi, dan operasi basis data relasional. Secara fungsional, aplikasi ini dirancang untuk mengakomodasi satu jenis peran pengguna (*single-user role*), yaitu pengguna akhir yang akan melakukan skrining kesehatan mental. Antarmuka sistem dibangun dengan prinsip desain responsif agar tata letak elemen dapat menyesuaikan secara otomatis dengan berbagai ukuran layar, baik pada perangkat komputer (desktop) maupun perangkat seluler (mobile). Selain itu, untuk memastikan aksesibilitas dan meminimalisasi bias pemahaman saat pengisian kuesioner, seluruh antarmuka sistem diimplementasikan menggunakan bahasa Indonesia. Penjabaran pada subbab ini difokuskan pada hasil antarmuka visual, logika otentikasi, dan mekanisme interaksi aplikasi dengan *Application Programming Interface* (API).

4.2.1. Konfigurasi Lingkungan Sistem

Tahap awal dalam pengembangan aplikasi web adalah melakukan pengaturan lingkungan kerja dan integrasi teknologi *frontend*. Sistem ini menggunakan ekosistem *Framework* Laravel yang dipadukan dengan basis data MySQL, serta memanfaatkan Vite sebagai alat pemaket aset (asset bundler) modern.

Untuk pengembangan antarmuka pengguna (UI), sistem mengimplementasikan Tailwind CSS yang dikompilasi secara terpusat melalui Vite. Konfigurasi ini tidak hanya mengoptimalkan ukuran berkas Cascading Style Sheets (CSS) dan JavaScript, tetapi juga mengaktifkan fitur hot module replacement guna mempercepat siklus iterasi desain selama masa pengembangan. Guna memastikan efisiensi dan konsistensi visual, desain antarmuka diperkaya dengan pustaka komponen UI, yakni DaisyUI. Selain itu, sistem menetapkan standarisasi tipografi secara menyeluruh dengan menggunakan rupa huruf (font) 'Instrument Sans' sebagai identitas visual utama aplikasi.

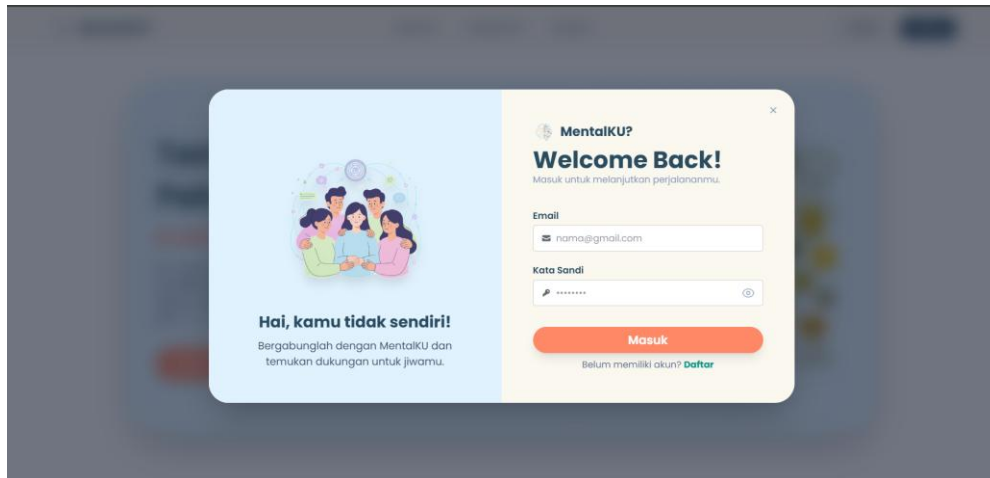
4.2.2. Fitur Registrasi

Gambar 36. Tampilan Antarmuka Modal Registrasi

Fitur registrasi merupakan titik akses awal bagi pengguna baru untuk dapat menggunakan layanan skrining awal kesehatan mental pada sistem Mentalku. Secara antarmuka, fitur ini diimplementasikan dalam bentuk modal (pop-up dialog), yang memungkinkan pengguna untuk melakukan pendaftaran tanpa harus berpindah dari halaman utama. Pendekatan desain ini dipilih untuk mempertahankan fokus pengguna dan memberikan pengalaman navigasi yang lebih efisien.

Pada formulir pendaftaran ini, sistem mengumpulkan sejumlah data identitas dan demografis dasar yang diperlukan untuk pembuatan profil dan personalisasi layanan ke depannya. Variabel data yang diminta mencakup nama lengkap, nama panggilan, jenis kelamin, tanggal lahir, dan alamat surel (email). Selain itu, sistem juga menyediakan kolom pembuatan kata sandi yang dilengkapi dengan fitur toggle visibilitas serta kolom konfirmasi kata sandi guna mencegah kesalahan input dari sisi pengguna. Setelah proses pengisian selesai dan pengguna menekan tombol "Daftar Sekarang", sistem pada sisi backend Laravel akan melakukan validasi data sebelum menyimpannya secara aman ke dalam basis data MySQL.

4.2.3. Fitur *Login*



Gambar 37. Tampilan Antarmuka Modal Login

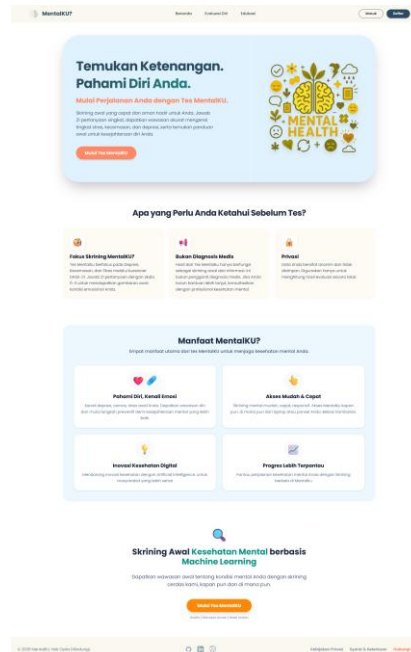
Fitur *Login* berfungsi sebagai gerbang keamanan utama bagi pengguna terdaftar untuk mengakses fungsionalitas inti dari sistem Mentalku. Senada dengan fitur registrasi, antarmuka *Login* juga diimplementasikan menggunakan elemen modal guna menjaga konsistensi desain dan meminimalisasi perpindahan halaman (page reload) yang tidak perlu. Antarmuka ini dirancang dengan tata letak terbagi (split-layout), di mana sisi kiri menampilkan elemen visual dan pesan empatik, sementara sisi kanan difokuskan pada formulir pengisian kredensial.

Untuk memvalidasi akses, pengguna diwajibkan memasukkan alamat surel (email) dan kata sandi yang telah didaftarkan sebelumnya. Guna meningkatkan pengalaman pengguna (UX), formulir ini dilengkapi dengan ikon indikator pada setiap kolom masukan, serta tombol toggle visibilitas pada kolom kata sandi. Ketika pengguna menekan tombol "Masuk", sistem backend akan melakukan pencocokan data kredensial dengan rekaman di dalam basis data. Jika otentikasi dinyatakan berhasil, sistem akan menginisiasi sesi pengguna (user session) dan secara otomatis mengarahkan pengguna (redirect) ke halaman dasbor utama aplikasi.

4.2.4. Fitur *Dashboard*

Halaman Beranda (*Dashboard*) merupakan pusat informasi sekaligus titik awal interaksi (*entry point*) pengguna setelah berhasil masuk ke dalam

sistem Mentalku. Secara fungsional, halaman ini tidak dirancang untuk menampilkan analitik data, melainkan difokuskan untuk memberikan edukasi pra-evaluasi dan mengarahkan pengguna menuju fitur inti aplikasi.



Gambar 38. Tampilan Antarmuka Halaman Beranda

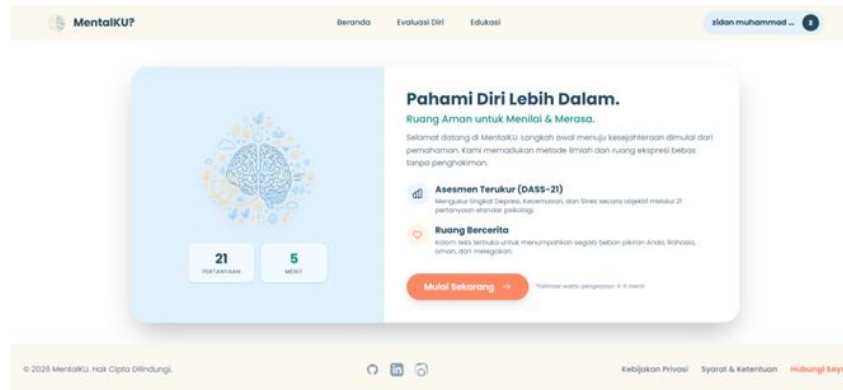
Pada halaman ini, sistem menyajikan informasi krusial terkait batasan layanan, di mana ditekankan bahwa tes yang disediakan murni merupakan skrining awal (berbasis skala DASS-21) dan bukan merupakan diagnosis medis pengganti profesional. Selain itu, sistem juga menjabarkan mekanisme privasi data serta manfaat penggunaan layanan. Sebagai fungsi navigasi utama, sistem menyediakan tombol *Call-to-Action* (CTA) "Mulai Tes Mentalku" yang ditempatkan pada beberapa bagian halaman strategis. Tombol ini berfungsi sebagai pemicu (*trigger*) yang akan mengarahkan pengguna secara langsung ke antarmuka pengisian kuesioner evaluasi diri.

4.2.5. Fitur Evaluasi Diri

Fitur Evaluasi Diri merupakan komponen fungsional utama (*core system*) dari aplikasi Mentalku. Sistem ini dirancang untuk memfasilitasi proses skrining kesehatan mental pengguna melalui pendekatan *hybrid*, yaitu penggabungan antara instrumen psikometri terstandarisasi dan pemrosesan bahasa alami (*Natural Language Processing*). Guna menjaga kenyamanan

pengguna dan menstrukturkan aliran data, proses evaluasi ini dipecah ke dalam empat tahapan sekuensial, yang dimulai dari halaman cover persiapan hingga penyajian hasil analisis akhir.

1. Evaluasi Diri – Cover



Gambar 39. Tampilan Antarmuka Halaman Cover Evaluasi Diri

Halaman cover berfungsi sebagai gerbang persiapan sebelum pengguna menginisiasi sesi pengisian data evaluasi. Secara sistem, halaman ini dirancang untuk mengelola ekspektasi pengguna dengan memberikan transparansi mengenai instrumen apa saja yang akan digunakan. Informasi yang disajikan mencakup penjelasan mengenai penggunaan 21 butir pertanyaan dari skala DASS-21 (pendekatan kuantitatif) serta penyediaan "Ruang Bercerita" sebagai media ekspresi teks bebas (pendekatan kualitatif). Antarmuka ini secara eksplisit menampilkan estimasi waktu pengerjaan dan total pertanyaan yang harus dijawab. Proses evaluasi akan resmi dimulai ketika pengguna menekan tombol "Mulai Sekarang", yang sekaligus memicu sistem untuk menyiapkan penampungan data sementara berbasis sesi (session-based storage) pada sisi server Laravel.

2. Evaluasi Diri – Kuisisioner Dass21

Setelah melewati halaman cover, pengguna diarahkan pada antarmuka pengisian kuesioner Depression, Anxiety, and Stress Scale (DASS-21). Mengingat instrumen ini terdiri dari 21 butir

pertanyaan, sistem mengimplementasikan pola navigasi wizard atau formulir multi-langkah (multi-step form).

The screenshot shows a web application interface for a mental health assessment. The header includes the logo 'MentaIKU?', navigation links 'Beranda', 'Evaluasi Diri', and 'Edukasi', and a user profile 'zidan muhammad'. The main content area is titled 'Bagian 1: Tingkat Stres' with a progress indicator '1 / 4 Bagian'. It contains three questions, each with four response buttons: 'Tidak Pernah', 'Kadang', 'Sering', and 'Selalu'. The first two questions have 'Kadang' selected, while the third has 'Sering' selected. A 'Lanjut ke Bagian 2' button is at the bottom right. The footer contains copyright information, social media icons, and links for 'Kebijakan Privasi', 'Syarat & Ketentuan', and 'Hubungi Saya'.

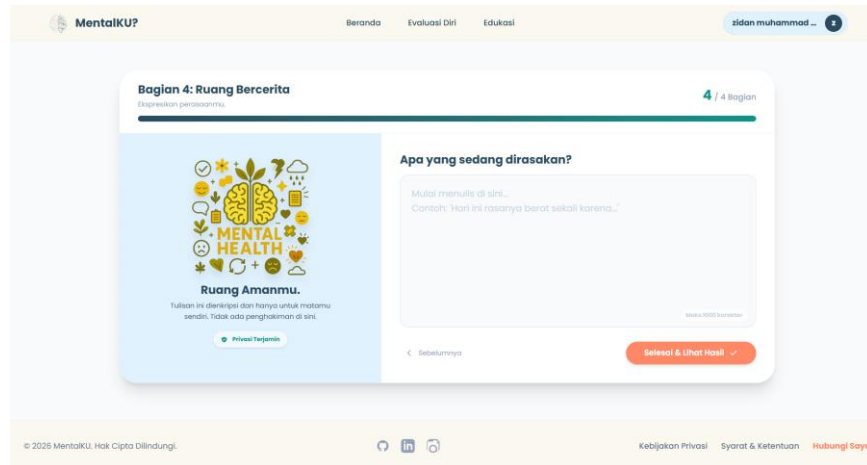
Gambar 40. Tampilan Antarmuka Halaman Kuisioner Dass-21

Kuesioner ini dipartisi ke dalam tiga halaman yang merepresentasikan tiga metrik utama evaluasi, yaitu Tingkat Stres (Bagian 1), Tingkat Kecemasan (Bagian 2), dan Tingkat Depresi (Bagian 3). Pada setiap halamannya, sistem merender tepat 7 butir pertanyaan yang berfokus pada satu kategori metrik tersebut. Untuk proses penjawaban, pengguna disediakan empat opsi skala Likert mulai dari "Tidak Pernah" hingga "Selalu". Guna memberikan umpan balik visual terkait proses pengisian, sistem menampilkan progress bar dinamis di bagian atas antarmuka. Selain itu, sistem juga menerapkan penyimpanan sementara berbasis sesi (session storage) pada setiap transisi halaman. Hal ini memastikan bahwa data jawaban yang telah diinput pada bagian sebelumnya tidak hilang ketika pengguna menekan tombol "Lanjut" maupun tombol "Kembali".

3. Fitur Evaluasi Diri – Curhat

Sebagai tahap penutup dari fase pengumpulan data evaluasi, sistem mengarahkan pengguna pada antarmuka "Ruang Bercerita" (Bagian 4). Berbeda dengan tahap sebelumnya yang bersifat kuantitatif, halaman ini berfokus pada ekstraksi data kualitatif tidak terstruktur melalui komponen text area. Pengguna diinstruksikan untuk mengekspresikan kondisi emosional mereka ke dalam bentuk

narasi teks bebas. Guna menjaga efisiensi komputasi pada pemrosesan Natural Language Processing (NLP) di sisi backend, sistem menerapkan batasan masukan maksimal sebanyak 1000 karakter.



Gambar 41. Tampilan Antarmuka Halaman Curhat

Antarmuka ini dilengkapi dengan tombol aksi final "Selesai & Lihat Hasil". Interaksi pada tombol tersebut berfungsi sebagai *trigger* utama untuk menyelesaikan sesi evaluasi. Secara prosedural, sistem Laravel akan mengagregasi seluruh data jawaban kuesioner DASS-21 yang sebelumnya tersimpan di dalam sesi, menggabungkannya dengan teks narasi (curhat) dari halaman ini, dan mengirimkannya sebagai payload terpadu ke endpoint API Python. Bersamaan dengan proses pengiriman data tersebut, sistem secara otomatis mengunci atau menghapus status sesi evaluasi yang sedang berjalan. Mekanisme ini diimplementasikan sebagai langkah preventif untuk mencegah terjadinya duplicate submission apabila pengguna mencoba kembali ke halaman ini menggunakan fitur navigasi mundur (browser back button).

4. Fitur Evaluasi Diri – Hasil

Tahap akhir dari alur evaluasi adalah penyajian antarmuka "Hasil Skrining", yang berfungsi untuk menampilkan luaran data (output) setelah respons dari endpoint API Python diterima dan diproses oleh sistem web. Antarmuka ini dirancang untuk

menyajikan informasi kompleks menjadi bentuk visual yang mudah dipahami oleh pengguna awam.



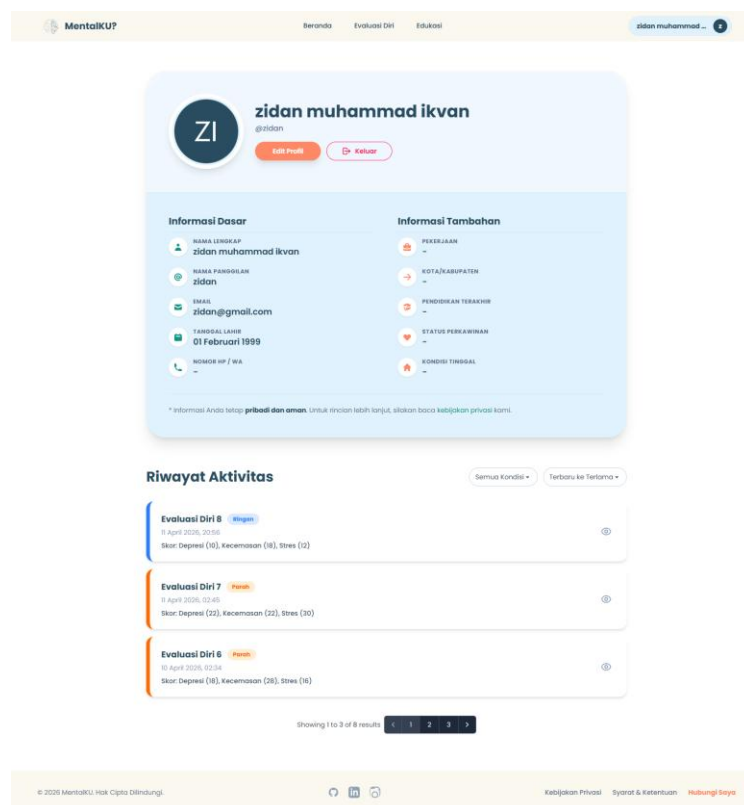
Gambar 42. Tampilan Antarmuka Halaman Hasil Evaluasi Diri

Bagian atas antarmuka menampilkan hasil kalkulasi instrumen DASS-21, yang dipecah menjadi tiga indikator utama (Depresi, Stres, dan Kecemasan) beserta skor numerik dan label tingkat keparahannya (misalnya: Normal, Ringan, Sedang, Parah, Sangat Parah). Tepat di bawahnya, sistem menampilkan blok "Rekomendasi Personalisasi". Teks pada bagian ini merupakan hasil generate dari model Large Language Model (Gemini AI) yang dikonfigurasi melalui prompt engineering pada sisi backend Python, menggunakan gabungan parameter skor kuesioner dan teks narasi pengguna.

Guna memitigasi risiko malpraktik digital dan mematuhi standar etika, sistem menyertakan blok peringatan medis secara eksplisit, yang merujuk pada Undang-Undang Kesehatan dan Kode Etik Psikologi Indonesia (HIMPPI). Peringatan ini menegaskan

bahwa luaran sistem murni merupakan skrining awal dan bukan diagnosis final. Sebagai bentuk intervensi edukatif lanjutan, halaman ini juga dilengkapi dengan modul "Bacaan Terkait" yang secara dinamis menyajikan artikel atau materi psikoedukasi yang relevan dengan kondisi dominan atau tingkat keparahan tertinggi yang dialami pengguna.

4.2.6. Fitur Profil



Gambar 43. Tampilan Antarmuka Halaman Profil

Fitur Profil berfungsi sebagai pusat informasi identitas dan data demografis pengguna di dalam sistem Mentalku. Secara fungsional, halaman ini bertugas melakukan penarikan data (data retrieval) dari basis data relasional MySQL dan merendernya secara dinamis berdasarkan identifikasi sesi pengguna (user session) yang sedang aktif.

Untuk memudahkan pembacaan arsitektur informasi, struktur data pada halaman ini dikategorisasikan ke dalam dua kelompok utama. Kelompok pertama adalah "Informasi Dasar", yang memuat atribut esensial seperti nama lengkap, nama panggilan, email, dan tanggal lahir yang sebelumnya

dikumpulkan melalui fase registrasi. Kelompok kedua adalah "Informasi Tambahan", yang mengakomodasi variabel demografis pendukung seperti status pekerjaan, pendidikan terakhir, kondisi tempat tinggal, dan beberapa lainnya. Selain berperan sebagai dasbor informasi personal, halaman ini juga diintegrasikan dengan dua fungsi kontrol utama (action triggers), yaitu tombol "Edit Profil" untuk menginisiasi proses pembaruan data (update), serta tombol "Keluar" yang secara langsung mengeksekusi logika terminasi sesi (session destroy) pada sisi backend Laravel untuk mencabut hak akses otentikasi pengguna.

4.2.7. Fitur Kelola Profil

The screenshot displays the 'Edit Profil' interface. At the top, there's a navigation bar with 'Mentalku?' and links for 'Beranda', 'Evaluasi Diri', and 'Edukasi'. The user's name 'Zidan Muhammad ...' is visible in the top right. The main content area is titled 'Edit Profil' and contains a profile card. The card has a circular profile picture placeholder with the initials 'ZI' and a 'Foto Profil' section with a 'Ganti Gambar' button and a 'Hapus' button. Below this, the 'Informasi Dasar' section includes fields for 'Nama Lengkap' (Zidan Muhammad Ivan), 'Nama Panggilan' (Zidan), 'Jenis Kelamin' (Laki-laki), 'Tanggal Lahir' (01/02/1999), 'Email' (zidan@gmail.com), and 'Nomor HP / WhatsApp' (Contoh: 08234567890). The 'Informasi Tambahan' section includes a 'Pekerjaan' field (Contoh: Mahasiswa, Karyawan Swasta), 'Status Perkawinan' (Pilih...), 'Kondisi Tinggal' (Pilih...), 'Kota / Kabupaten' (Contoh: Jakarta, Bandung, Batam), and 'Pendidikan Terakhir' (Pilih...). At the bottom of the form are 'Batal' and 'Simpan Perubahan' buttons. The footer contains copyright information, social media icons, and links for 'Kebijakan Privasi', 'Syarat & Ketentuan', and 'Hubungi Saya'.

Gambar 44. Tampilan Antarmuka Halaman Edit Profil

Fitur Kelola Profil merupakan implementasi dari fungsionalitas pembaruan data (update) dalam siklus manajemen basis data sistem Mentalku. Antarmuka ini dirancang untuk memberikan kendali kepada pengguna dalam memodifikasi informasi personal mereka, dengan tetap menerapkan protokol integritas data secara ketat.

Pada tingkat fungsionalitas manajemen berkas, antarmuka ini memfasilitasi pengguna untuk mengelola foto profil. Sistem menerapkan aturan validasi backend secara spesifik, di mana berkas yang diunggah harus berformat JPG atau PNG dengan batas ukuran maksimal 2 *Megabyte* (MB). Selain itu, tersedia pula fungsi untuk menghapus foto secara langsung dari *server storage*.

Dalam hal arsitektur pengisian formulir, sistem menerapkan kontrol akses modifikasi yang berbeda pada dua kelompok data. Pada blok "Informasi Dasar", sistem mengunci sebagian besar atribut identitas utama (seperti nama lengkap, nama panggilan, jenis kelamin, tanggal lahir, dan surel) dengan status *read-only* guna menjaga integritas rekam medis dan data sesi awal. Pada kelompok ini, pengguna hanya diizinkan untuk memodifikasi atribut nomor kontak (HP/WhatsApp). Sebaliknya, pada blok "Informasi Tambahan", sistem memberikan hak akses modifikasi penuh. Pengguna dapat memperbarui data demografis dinamis seperti status pekerjaan, pendidikan, dan kota asal secara berkelanjutan. Seluruh perubahan ini akan dieksekusi, divalidasi oleh controller Laravel, dan ditimpa ke dalam basis data MySQL hanya ketika pengguna menekan tombol "Simpan Perubahan".

4.2.8. Fitur Riwayat Aktivitas

Fitur Riwayat Aktivitas merupakan antarmuka perekaman historis (*historical record*) yang memungkinkan pengguna melacak rekam jejak evaluasi kesehatan mental mereka secara kronologis. Secara arsitektural, fitur ini diimplementasikan dalam dua lapis antarmuka: *list view* yang terintegrasi pada bagian bawah halaman profil, dan *detail view* pada halaman terpisah.

Pada *list view*, sistem melakukan penarikan data evaluasi masa lampau dari basis data. Guna menjaga kinerja pemuatan halaman dan mencegah query berlebih, sistem menerapkan teknik paginasi pada *server-side pagination* yang membatasi tampilan maksimal 3 entri riwayat per halaman. Fitur ini juga dibekali dengan 2 *Filtering*. Pengguna dapat mengurutkan data berdasarkan waktu (terbaru ke terlama, dan sebaliknya) serta menyaring data secara spesifik berdasarkan parameter *final_level* (Normal, Ringan, Sedang, Parah, Sangat

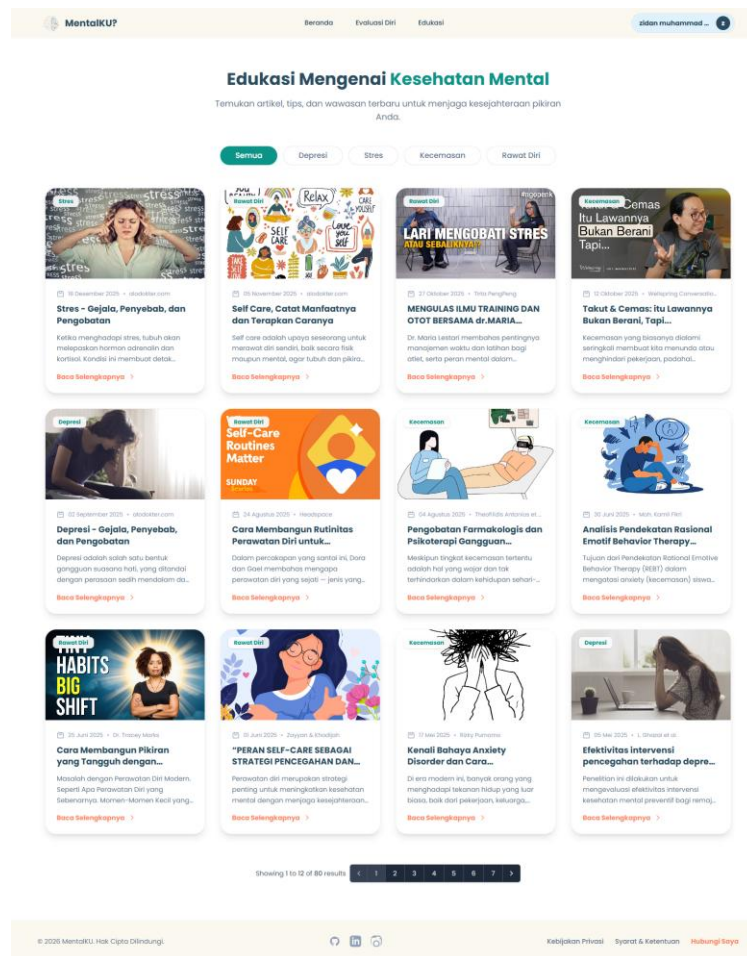
Parah). Sebagai penanda visual, sistem secara otomatis memberikan warna latar dan *border* yang dinamis pada setiap entri daftar, yang direpresentasikan dari tingkat keparahan evaluasi (contoh: warna indikator khusus untuk tingkat "Parah" dan "Ringan").



Gambar 45. Tampilan Antarmuka Halaman Riwayat Aktivitas Evaluasi Diri

Untuk melihat rekaman mendalam dari sebuah entri, pengguna dapat berinteraksi dengan tombol aksi (berupa ikon mata) pada daftar tersebut. Interaksi ini akan memicu permintaan menuju antarmuka detail. Halaman detail ini menyajikan rekaman statis dari luaran evaluasi yang bersangkutan, mencakup metrik persentase DASS-21 (Depresi, Stres, Kecemasan) serta blok "Rekomendasi Waktu Itu". Data rekomendasi yang ditampilkan merupakan data teks yang tersimpan secara permanen di dalam basis data pada saat evaluasi tersebut dieksekusi, guna memastikan pengguna mendapatkan konteks historis yang akurat tanpa terpengaruh oleh intervensi AI di sesi yang lebih baru.

4.2.9. Fitur Edukasi

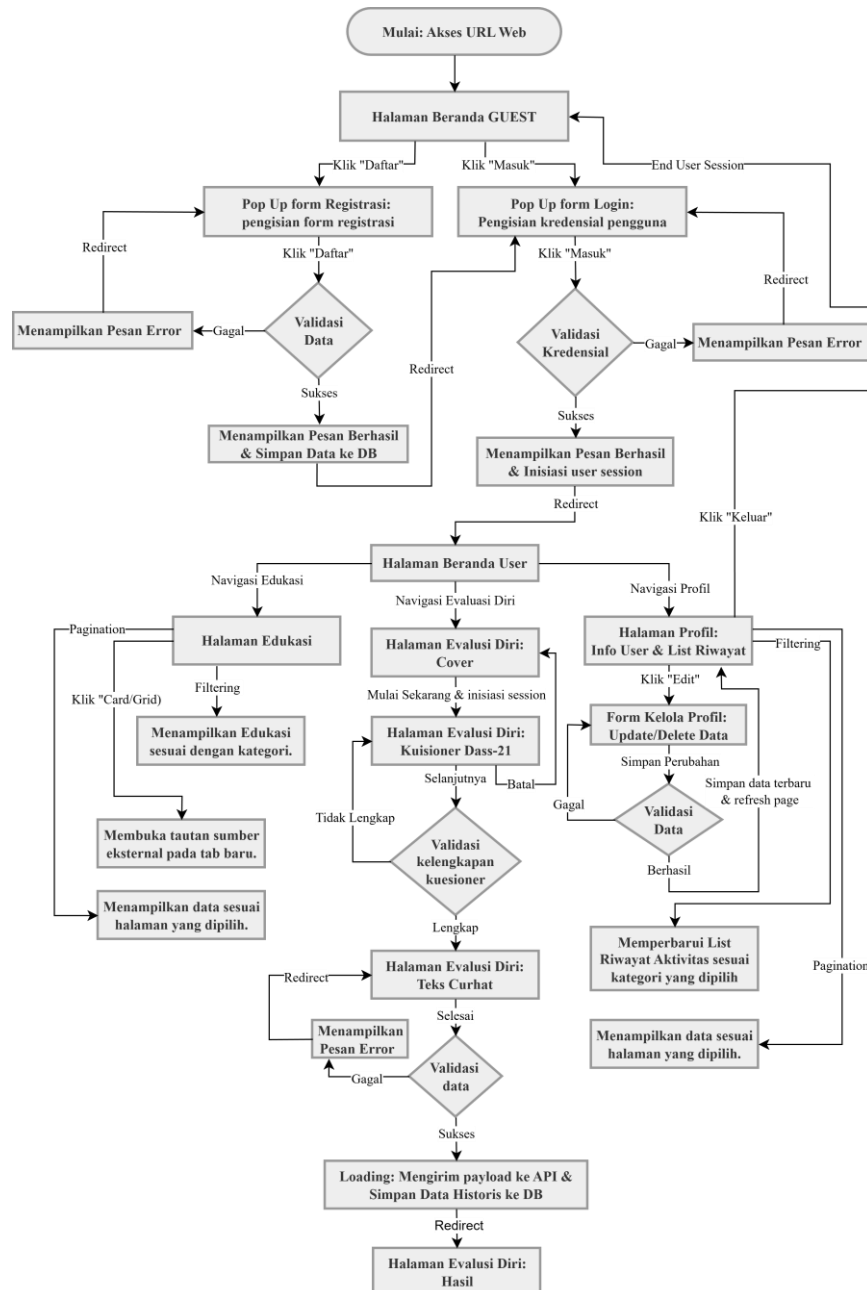


Gambar 46. Tampilan Antarmuka Halaman Edukasi

Fitur Edukasi berfungsi sebagai modul psikoedukasi yang melakukan penarikan data literatur kesehatan mental dari basis data secara dinamis. Data edukasi tersebut dihimpun dari berbagai sumber kredibel, termasuk artikel ilmiah, jurnal penelitian, media edukatif seperti YouTube, serta sumber terbuka lainnya. Untuk memfasilitasi penemuan informasi spesifik, sistem menyediakan fungsionalitas penyaringan (filtering) yang memungkinkan pengguna mengisolasi kumpulan artikel berdasarkan parameter kategori, seperti Depresi, Stres, Kecemasan, atau Rawat Diri. Mengingat besarnya volume data artikel yang tersedia, sistem mengimplementasikan teknik paginasi pada *server-side pagination* guna membatasi jumlah query basis data menjadi dua belas entri per halaman, sehingga performa pemuatan antarmuka tetap efisien dan stabil.

4.2.10. Alur Penggunaan

Sub-Subbab ini memaparkan alur penggunaan sistem (*user flow*) secara terintegrasi. Keseluruhan proses interaksi pengguna dengan aplikasi web Mentalku, mulai dari akses awal hingga eksekusi berbagai fungsionalitas utama sistem, diilustrasikan secara prosedural pada Gambar 47 berikut.



Gambar 47. Alur Penggunaan Web Mentalku

4.3. Pengujian *User Acceptance Testing* (UAT)

Pengujian *User Acceptance Testing* (UAT) dilakukan untuk memvalidasi bahwa aplikasi Mentalku telah memenuhi kebutuhan dan ekspektasi pengguna akhir sebelum aplikasi dinyatakan siap untuk digunakan. Pengujian ini melibatkan pengguna nyata yang diminta untuk mencoba secara langsung fungsionalitas yang tersedia pada aplikasi, sehingga hasil yang diperoleh mencerminkan pengalaman pengguna yang sesungguhnya.

Pengujian UAT dilaksanakan secara tatap muka dengan melibatkan empat orang pengguna yang memiliki latar belakang berbeda, yaitu sebagai berikut.

1. **Eka Fitri**, perempuan berusia 22 tahun, merupakan mahasiswi semester 8 Politeknik Negeri Batam yang sedang menjalani magang di PT XYZ sebagai *UI/UX Designer* sekaligus tengah mengerjakan tugas akhir.
2. **Masdika**, laki-laki berusia 23 tahun, merupakan mahasiswa semester 8 Politeknik Negeri Batam yang telah menyelesaikan tugas akhir dan sedang menjalani magang di PT XYZ sebagai *Machine Learning Engineer* sambil mempersiapkan diri memasuki dunia kerja.
3. **Davin**, laki-laki berusia 23 tahun, merupakan mahasiswa semester 8 Politeknik Caltex Riau yang telah menyelesaikan tugas akhir dan sedang menjalani magang di PT XYZ sebagai *Machine Learning Engineer*, sekaligus aktif belajar dan mengikuti tes bahasa Jepang dalam rangka mempersiapkan diri untuk bekerja di Jepang.
4. **Feby**, laki-laki berusia 28 tahun, merupakan karyawan PT XYZ yang bekerja sebagai *IT Support*.

Pengujian dilakukan dengan metode *blackbox Testing*, di mana setiap pengguna diminta untuk mencoba menggunakan aplikasi secara langsung tanpa mengetahui detail implementasi sistem di baliknya. Pengujian mencakup 8 kebutuhan fungsional utama dengan total 32 *test case*, meliputi fitur Registrasi (FR-01), *Login* (FR-02), *Logout* (FR-03), Pengelolaan Profil (FR-04), pengisian Form Skrining Evaluasi Diri (FR-05), melihat Hasil Skrining Evaluasi Diri (FR-06), mengakses Riwayat Skrining Evaluasi Diri (FR-07), serta mengakses Edukasi Kesehatan Mental (FR-08).

Berdasarkan hasil pengujian, seluruh 32 *test case* yang diujikan dinyatakan lulus (*passed*), yang berarti semua fitur berjalan sesuai dengan ekspektasi yang telah ditentukan. Selain memvalidasi fungsionalitas, pengguna juga memberikan masukan bahwa tampilan antarmuka aplikasi dapat dikembangkan lebih lanjut, khususnya dalam hal pemilihan warna dan estetika visual agar terasa lebih menarik bagi pengguna. Masukan tersebut menjadi catatan perbaikan yang dapat dipertimbangkan pada pengembangan aplikasi di iterasi berikutnya. Secara keseluruhan, hasil UAT menunjukkan bahwa aplikasi Mentalku telah memenuhi kebutuhan fungsional yang direncanakan dan siap untuk digunakan.

Tabel 15. Pengujian

Test Case Id	Kode Fitur	Deskripsi Pengujian	Kondisi Awal	Skenario Pengujian	Ekspektasi Hasil	Hasil Aktual	Status
TC001	FR-01	Pengujian fungsionalitas pendaftaran akun dengan data kredensial dan demografis yang valid secara menyeluruh.	Pengguna berstatus Guest berada di halaman Beranda dan modal Registrasi dalam keadaan terbuka.	1. Mengisi seluruh field input dengan data berformat valid. 2. Tekan tombol "Daftar".	Sistem menyimpan rekam data ke dalam database, menampilkan notifikasi pendaftaran berhasil, dan mengarahkan pengguna ke modal <i>Login</i> .	Sistem menyimpan rekam data ke dalam database, menampilkan notifikasi pendaftaran berhasil, dan mengarahkan pengguna ke modal <i>Login</i> .	Lulus
TC002	FR-01	Pengujian validasi penolakan sistem terhadap form yang kosong.	Modal Registrasi terbuka, seluruh/ salah satu field dalam keadaan kosong.	1. Membiarkan seluruh/ salah satu field input kosong. 2. Langsung tekan tombol "Daftar".	Sistem memblokir proses pengiriman data (submit) dan menampilkan pesan error validasi secara spesifik pada	Sistem memblokir proses pengiriman data (submit) dan menampilkan pesan error validasi secara spesifik pada	Lulus

					masing-masing field yang wajib diisi.	masing-masing field yang wajib diisi.	
TC003	FR-01	Pengujian validasi keunikan email pada database.	Modal Registrasi terbuka. Sistem telah memiliki satu akun terdaftar dengan surel "test@gmail.com".	1. Mengisi form dengan data valid. 2. Gunakan email "test@gmail.com" pada field email 3. Tekan tombol "Daftar".	Sistem menolak proses registrasi dan memberikan pop up pesan error bahwa email tersebut telah digunakan/ terdaftar.	Sistem menolak proses registrasi dan memberikan pop up pesan error bahwa email tersebut telah digunakan/ terdaftar.	Lulus
TC004	FR-01	Pengujian validasi pada field kata sandi (password mismatch).	Modal Registrasi terbuka.	1. Isi field kata sandi dengan kombinasi "Kunci123!". 2. Isi field Konfirmasi kata sandi dengan "KunciSalah456". 3. Tekan tombol "Daftar".	Sistem mendeteksi ketidakcocokan nilai string, menolak proses submit data, dan menampilkan pop up pesan peringatan bahwa konfirmasi kata sandi tidak sesuai.	Sistem mendeteksi ketidakcocokan nilai string, menolak proses submit data, dan menampilkan pop up pesan peringatan bahwa konfirmasi kata sandi tidak sesuai.	Lulus
TC005	FR-02	Pengujian fungsionalitas masuk	Pengguna berstatus Guest dan modal	1. Isi field Email dengan	Sistem memvalidasi data, berhasil menginisiasi	Sistem memvalidasi data, berhasil menginisiasi	Lulus

		(Login) menggunakan kredensial pengguna yang valid dan terdaftar.	Login dalam keadaan terbuka. Data pengguna telah ada di database.	email yang terdaftar. 2. Isi field kata sandi dengan kata sandi yang benar. 3. Tekan tombol "Masuk".	session otentikasi pengguna, dan mengarahkan pengguna ke halaman Beranda.	session otentikasi pengguna, dan mengarahkan pengguna ke halaman Beranda.	
TC006	FR-02	Pengujian validasi penolakan sistem terhadap upaya masuk dengan kata sandi yang salah.	Modal Login terbuka. email pengguna terdaftar di basis data.	1. Isi field Email dengan email yang terdaftar 2. Isi field kata sandi dengan kata sandi yang salah. 3. Tekan tombol "Masuk".	Sistem menolak upaya masuk dan menampilkan pop up pesan error "Email/Kata Sandi Salah".	Sistem menolak upaya masuk dan menampilkan pop up pesan error "Email/Kata Sandi Salah".	Lulus
TC007	FR-02	Pengujian penanganan terhadap email yang tidak terdaftar di dalam sistem.	Modal Login terbuka.	1. Isi field Email dengan email acak yang belum pernah didaftarkan. 2. Isi field kata sandi dengan kata sandi acak.	Sistem menolak upaya masuk dan menampilkan pop up pesan error "Email/Kata Sandi Salah"	Sistem menolak upaya masuk dan menampilkan pop up pesan error "Email/Kata Sandi Salah"	Lulus

				3. Tekan tombol "Masuk".			
TC008	FR-03	Pengujian fungsionalitas terminasi-sesi autentikasi melalui tombol Keluar.	Pengguna memiliki sesi aktif (logged-in) dan berada di antarmuka halaman Profil.	1. Tekan tombol "Keluar" yang terdapat pada blok identitas profil.	Sistem secara responsif mengeksekusi penghancuran sesi di backend dan mengarahkan pengguna ke halaman <i>Dashboard</i> (Guest). Tombol navigasi berbasis sesi (seperti Profil dan Evaluasi) tidak lagi dapat diakses.	Sistem secara responsif mengeksekusi penghancuran sesi di backend dan mengarahkan pengguna ke halaman <i>Dashboard</i> (Guest). Tombol navigasi berbasis sesi (seperti Profil dan Evaluasi) tidak lagi dapat diakses.	Lulus
TC009	FR-03	Memastikan pengguna tidak dapat mengakses kembali halaman setelah <i>Logout</i> , termasuk melalui tombol back atau cache browser.	Pengguna baru saja berhasil melakukan proses <i>Logout</i> dan saat ini berada di halaman Beranda.	1. Tekan tombol navigasi mundur pada web (browser).	Sistem menolak akses kembali ke halaman yang membutuhkan <i>Login</i> (seperti halaman Profil), memvalidasi ketiadaan sesi dan mengarahkan kembali pengguna ke halaman Beranda.	Sistem menolak akses kembali ke halaman yang membutuhkan <i>Login</i> (seperti halaman Profil), memvalidasi ketiadaan sesi dan mengarahkan kembali pengguna ke halaman Beranda.	Lulus

TC010	FR-04	Memastikan field Informasi Dasar tidak dapat diubah (read-only).	Pengguna berada di antarmuka Edit Profil.	1. Arahkan kursor dan coba ubah nilai teks pada field Nama Lengkap, Nama Panggilan, Jenis Kelamin, Tanggal Lahir, atau Email.	Sistem mengunci seluruh field tersebut (tidak dapat diklik/diketik) sehingga pengguna sama sekali tidak dapat memodifikasinya.	Sistem mengunci seluruh field tersebut (tidak dapat diklik/diketik) sehingga pengguna sama sekali tidak dapat memodifikasinya.	Lulus
TC011	FR-04	Memastikan pengguna dapat memperbarui Nomor HP dan Informasi Tambahan.	Pengguna berada di antarmuka Edit Profil.	1. Ubah nomor pada field Nomor HP/ WhatsApp. 2. Lakukan perubahan pada satu atau lebih opsi di blok Informasi Tambahan. 3. Tekan tombol "Simpan Perubahan.	Sistem menerima masukan, menimpa data lama di dalam basis data, menampilkan notifikasi berhasil, dan mengarahkan pengguna kembali ke halaman Profil utama dengan data yang telah diperbarui.	Sistem menerima masukan, menimpa data lama di dalam basis data, menampilkan notifikasi berhasil, dan mengarahkan pengguna kembali ke halaman Profil utama dengan data yang telah diperbarui.	Lulus
TC012	FR-04	Memastikan pengguna dapat mengunggah foto	Pengguna berada di antarmuka Edit Profil.	1. Tekan tombol "Ganti Gambar". 2. Pilih berkas	Sistem berhasil memproses berkas, menyimpannya ke dalam	Sistem berhasil memproses berkas, menyimpannya ke dalam	Lulus

		profil sesuai aturan sistem.		lokal berformat JPG/PNG dengan kapasitas di bawah 2 MB. 3. Tekan tombol "Simpan Perubahan".	direktori server, dan memperbarui tampilan avatar pengguna secara menyeluruh.	direktori server, dan memperbarui tampilan avatar pengguna secara menyeluruh.	
TC013	FR-04	Memastikan sistem menolak file dengan format dan ukuran yang tidak valid.	Pengguna berada di antarmuka Edit Profil.	1. Tekan "Ganti Gambar". 2. Pilih berkas dengan ekstensi non-gambar (contoh: .pdf atau .docx) ATAU gambar berukuran lebih dari 2 MB. 3. Tekan tombol "Simpan Perubahan".	Sistem gagal memproses pembaruan jika file tidak valid dan menampilkan pesan error sesuai format atau ukuran.	Sistem gagal memproses pembaruan jika file tidak valid dan menampilkan pesan error sesuai format atau ukuran.	Lulus
TC014	FR-04	Memastikan pengguna dapat	Pengguna berada di antarmuka Edit Profil	1. Tekan tombol "Hapus" yang	Sistem menghapus file foto profil dan	Sistem menghapus file foto profil dan	Lulus

		menghapus foto profil.	dan telah memiliki foto profil modifikasi sebelumnya.	berdekatan dengan foto profil. 2. Tekan tombol "Simpan Perubahan".	menampilkan avatar default (inisial).	menampilkan avatar default (inisial).	
TC015	FR-04	Memastikan perubahan dibatalkan saat pengguna menekan tombol Batal.	Pengguna berada di antarmuka Edit Profil.	1. Ketik atau ubah data pada beberapa field. 2. Tekan tombol "Batal".	Sistem tidak menyimpan perubahan dan pengguna kembali ke halaman Profil.	Sistem tidak menyimpan perubahan dan pengguna kembali ke halaman Profil.	Lulus
TC016	FR-05	Memastikan pengguna dapat mengisi kuesioner hingga selesai dan melihat hasil skrining.	Pengguna (logged in) berada di halaman cover Evaluasi Diri.	1. Klik "Mulai Sekarang". 2. Isi semua 21 pertanyaan DASS-21 pada Bagian 1 hingga 3. 3. Isi teks cerita pada Bagian 4. 4. Tekan "Selesai & Lihat Hasil".	Sistem menyimpan setiap jawaban ke dalam session, memproses gabungan payload ke API Python, dan berhasil mengarahkan pengguna ke halaman Hasil Skrining.	Sistem menyimpan setiap jawaban ke dalam session, memproses gabungan payload ke API Python, dan berhasil mengarahkan pengguna ke halaman Hasil Skrining.	Lulus
TC017	FR-05	Memastikan semua pertanyaan	Pengguna berada di salah satu	1. Biarkan satu atau lebih pertanyaan	Sistem menahan transisi halaman,	Sistem menahan transisi halaman,	Lulus

		kuesioner wajib diisi.	halaman kuesioner.	tidak dijawab. 2. Tekan tombol "Lanjut ke Bagian 2".	mencegah penyimpanan session, dan menampilkan peringatan visual bahwa semua pertanyaan wajib dijawab sebelum dapat melanjutkan.	mencegah penyimpanan session, dan menampilkan peringatan visual bahwa semua pertanyaan wajib dijawab sebelum dapat melanjutkan.	
TC018	FR-05	Memastikan jawaban tetap tersimpan saat pengguna kembali ke pertanyaan sebelumnya.	Pengguna telah mengisi Bagian 1 dan saat ini berada di Bagian 2.	1. Tekan tombol "Kembali" yang ada di antarmuka sistem. 2. Periksa isian pada Bagian 1.	Sistem merender kembali halaman Bagian 1 dengan opsi jawaban yang tetap terisi sesuai pilihan pengguna sebelumnya (data session tidak hilang).	Sistem merender kembali halaman Bagian 1 dengan opsi jawaban yang tetap terisi sesuai pilihan pengguna sebelumnya (data session tidak hilang).	Lulus
TC019	FR-05	Memastikan jumlah karakter pada Ruang Bercerita tidak melebihi batas yang ditentukan.	Pengguna berada di Bagian 4 (Ruang Bercerita).	1. Salin dan tempel teks (copy-paste) yang panjangnya lebih dari 1000 karakter ke dalam text area. 2. Tekan "Selesai &	Sistem menolak input lebih dari 1000 karakter dan menampilkan pesan error.	Sistem menolak input lebih dari 1000 karakter dan menampilkan pesan error.	Lulus

				Lihat Hasil".			
TC020	FR-06	Memastikan hasil evaluasi ditampilkan dengan benar berdasarkan data dari sistem.	Pengguna baru saja berhasil men-submit form evaluasi.	1. Amati halaman Hasil Skrining yang ditampilkan.	Sistem merender antarmuka hasil secara akurat: menampilkan tiga skor DASS-21 (kalkulasi benar), memuat rekomendasi teks dari Gemini AI, dan menampilkan artikel yang relevan dengan kondisi pengguna.	Sistem merender antarmuka hasil secara akurat: menampilkan tiga skor DASS-21 (kalkulasi benar), memuat rekomendasi teks dari Gemini AI, dan menampilkan artikel yang relevan dengan kondisi pengguna.	Lulus
TC021	FR-06	Memastikan pengguna tidak dapat kembali ke halaman Ruang Bercerita.	Pengguna sedang berada di halaman Hasil Skrining (setelah submit berhasil).	1. Tekan tombol Back pada browser untuk kembali ke halaman Ruang Bercerita.	Sistem mencegah akses kembali ke halaman Ruang Bercerita dan mengarahkan pengguna ke halaman yang sesuai.	Sistem mencegah akses kembali ke halaman Ruang Bercerita dan mengarahkan pengguna ke halaman yang sesuai.	Lulus
TC022	FR-06	Memastikan sistem menangani kegagalan koneksi	Pengguna berada di Bagian 4. (Kondisi disimulasi	1. Tekan "Selesai & Lihat Hasil".	Sistem menampilkan pesan bahwa layanan sedang	Sistem menampilkan pesan bahwa layanan sedang	Lulus

		API dengan menampilkan pesan yang sesuai.	kan: Server Python API dimatikan secara lokal).		bermasalah dan menyarankan pengguna mencoba lagi nanti.	bermasalah dan menyarankan pengguna mencoba lagi nanti.	
TC023	FR-07	Memastikan daftar Riwayat Aktivitas ditampilkan di halaman Profil.	Pengguna (logged in) berada di halaman Profil dan telah memiliki lebih dari 3 riwayat evaluasi sebelumnya.	1. Scroll ke bagian bawah halaman Profil pada blok "Riwayat Aktivitas". 2. Amati daftar yang dirender.	Sistem menampilkan daftar riwayat (judul, level, waktu, skor) maksimal 3 data per halaman.	Sistem menampilkan daftar riwayat (judul, level, waktu, skor) maksimal 3 data per halaman.	Lulus
TC024	FR-07	Memastikan pengguna dapat memfilter riwayat berdasarkan level.	Pengguna berada di blok Riwayat Aktivitas halaman Profil.	1. Klik dropdown "Semua Kondisi". 2. Pilih salah satu filter keparahan (misal: "Parah").	Sistem hanya menampilkan riwayat dengan level yang dipilih (misalnya "Parah").	Sistem hanya menampilkan riwayat dengan level yang dipilih (misalnya "Parah").	Lulus
TC025	FR-07	Memastikan pengguna dapat mengurutkan riwayat berdasarkan waktu.	Pengguna berada di blok Riwayat Aktivitas halaman Profil.	1. Klik dropdown pengurutan (misal: "Terbaru ke Terlama"). 2. Ubah opsi menjadi	Sistem menampilkan riwayat dari yang paling lama ke yang terbaru.	Sistem menampilkan riwayat dari yang paling lama ke yang terbaru.	Lulus

				"Terl lama ke Ter baru".			
TC026	FR-07	Memastikan pagination menampilkan data berikutnya.	Pengguna berada di blok Riwayat Aktivitas halaman Profil. Data riwayat berjumlah lebih dari 3 entri.	1. Scroll ke bagian navigasi paginasi di bawah daftar riwayat. 2. Klik angka halaman "2" (atau tombol panah selanjutnya).	Sistem menampilkan 3 data berikutnya dan halaman aktif berpindah sesuai pilihan.	Sistem menampilkan 3 data berikutnya dan halaman aktif berpindah sesuai pilihan.	Lulus
TC027	FR-07	Memastikan pengguna dapat membuka detail riwayat.	Pengguna berada di blok Riwayat Aktivitas halaman Profil.	1. Tekan tombol ikon "Mata" pada salah satu entri riwayat yang ada di daftar.	Sistem menampilkan halaman detail sesuai riwayat yang dipilih.	Sistem menampilkan halaman detail sesuai riwayat yang dipilih.	Lulus
TC028	FR-07	Memastikan data pada halaman detail riwayat tetap konsisten.	Pengguna telah berhasil masuk ke halaman detail riwayat	1. Amati skor metrik (Depresi, Stres, Kecemasan) dan teks blok "Rekomendasi	Sistem menampilkan data sesuai riwayat saat itu (skor dan rekomendasi tidak berubah).	Sistem menampilkan data sesuai riwayat saat itu (skor dan rekomendasi tidak berubah).	Lulus

				Waktu Itu".			
TC029	FR-08	Memastikan daftar edukasi ditampilkan saat halaman dibuka.	Pengguna mengakses halaman Edukasi. Filter aktif berada pada opsi "Semua".	1. Amati jumlah card edukasi yang ditampilkan pada halaman pertama. 2. Periksa teks indikator paginasi di bagian bawah (contoh: "Showing 1 to 12 of 80 results")	Sistem menampilkan list edukasi dalam bentuk grid, maksimal 12 edukasi per halaman, dan jumlah total data ditampilkan dengan benar.	Sistem menampilkan list edukasi dalam bentuk grid, maksimal 12 edukasi per halaman, dan jumlah total data ditampilkan dengan benar.	Lulus
TC030	FR-08	Memastikan pengguna dapat memfilter edukasi berdasarkan kategori.	Pengguna berada di halaman Edukasi.	1. Klik salah satu tombol filter spesifik (misal: tombol "Kecemasan"). 2. Amati daftar edukasi yang dirender setelah aksi klik.	Sistem hanya menampilkan list edukasi sesuai kategori yang dipilih (misalnya "Kecemasan") dan jumlah hasil diperbarui.	Sistem hanya menampilkan list edukasi sesuai kategori yang dipilih (misalnya "Kecemasan") dan jumlah hasil diperbarui.	Lulus

TC031	FR-08	Memastikan pagination menampilkan halaman berikutnya.	Pengguna berada di halaman 1 Edukasi.	1. Scroll ke bagian bawah dan klik angka "2" pada kontrol paginasi. 2. Amati perubahan URL dan data yang dirender.	Sistem menampilkan list edukasi selanjutnya dan halaman aktif berpindah ke halaman yang dipilih.	Sistem menampilkan list edukasi selanjutnya dan halaman aktif berpindah ke halaman yang dipilih.	Lulus
TC032	FR-08	Memastikan pengguna dapat membuka artikel/video dari sumber eksternal.	Pengguna berada di halaman Edukasi.	1. Klik tombol teks "Baca Selengkapnya " pada salah satu card edukasi.	Artikel/Video terbuka di tab baru tanpa menutup halaman aplikasi.	Artikel/Video terbuka di tab baru tanpa menutup halaman aplikasi.	Lulus

BAB V

KESIMPULAN

5.1. Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian aplikasi Mentalku, diperoleh kesimpulan sebagai berikut.

1. Aplikasi web Mentalku berhasil dibangun menggunakan Laravel 12 sebagai platform skrining awal kesehatan mental yang aman dan mandiri. Pengujian fungsionalitas melalui metode *blackbox Testing* dalam kerangka *User Acceptance Testing* (UAT) terhadap 32 *test case* oleh 4 pengguna akhir menunjukkan hasil 100% *passed*, membuktikan bahwa seluruh alur sistem mulai dari registrasi akun hingga akses modul edukasi kesehatan mental berjalan sesuai dengan kebutuhan fungsional yang telah ditetapkan. Platform ini terbukti mampu menjadi wadah evaluasi mandiri yang terstruktur sehingga mengurangi risiko *self-diagnosis* yang tidak tervalidasi.
2. Sistem berhasil mengintegrasikan instrumen psikometrik DASS-21 dengan model *Deep Learning* Bidirectional LSTM untuk analisis teks secara bersamaan dalam satu alur skrining. Model Bi-LSTM mencapai akurasi keseluruhan sebesar 86% dengan waktu inferensi 146ms per *batch* melalui layanan FastAPI, menunjukkan performa yang efisien untuk kebutuhan aplikasi *real-time*. Hasil gabungan kedua metode tersebut selanjutnya disintesis oleh Gemini AI untuk menghasilkan rekomendasi personal yang tervalidasi dan bebas dari *toxic positivity*.
3. Aplikasi Mentalku diimplementasikan secara responsif untuk mendukung akses lintas perangkat menggunakan Tailwind CSS. Arsitektur sistem dirancang dengan pendekatan REST API yang memisahkan lapisan komputasi AI (FastAPI/Python) dari lapisan manajemen antarmuka dan sesi pengguna (Laravel), sehingga menghasilkan sistem yang modular, skalabel, dan mudah dikembangkan lebih lanjut.

5.2. Saran

Berdasarkan hasil pengujian dan keterbatasan sistem yang telah diidentifikasi, dirumuskan beberapa saran sebagai berikut.

1. Model Bi-LSTM yang dikembangkan menunjukkan nilai recall sebesar 0,82 pada kategori depresi, yang mengindikasikan masih terdapat ruang peningkatan pada sensitivitas pendeteksian kelas tersebut. Pengembangan selanjutnya disarankan untuk memperkaya variasi dan volume data teks bermuatan depresi pada dataset pelatihan agar kemampuan generalisasi model terhadap ekspresi depresi yang beragam dapat meningkat.
2. Fitur sistem saat ini terbatas pada fungsi skrining awal. Pengembangan berikutnya sangat disarankan untuk mengintegrasikan platform dengan API layanan telekonseling profesional seperti layanan psikolog atau psikiater berlisensi, serta menyediakan fitur forum dukungan rekan sebaya (peer-support group) sebagai jembatan antara hasil skrining dan tindak lanjut yang tepat.
3. Berdasarkan masukan yang diperoleh selama proses UAT, antarmuka aplikasi disarankan untuk menjalani *revamp* pada palet warna dan estetika visual secara keseluruhan. Mengingat aplikasi ini berkaitan dengan kesehatan mental, desain antarmuka yang lebih menenangkan dan *engaging* secara psikologis dapat berkontribusi pada kenyamanan pengguna dalam menggunakan platform secara berkelanjutan.
4. Mengingat aplikasi Mentalku menangani data sensitif terkait kondisi psikologis pengguna, implementasi ke depan perlu mempertimbangkan kepatuhan terhadap regulasi perlindungan data pribadi yang berlaku, serta melibatkan tenaga profesional kesehatan mental dalam proses validasi klinis instrumen dan rekomendasi yang dihasilkan sistem.

DAFTAR PUSTAKA

- Alsaqqa, S., Sawalha, S., & Abdel-Nabi, H. (2020). Agile Software *Development: Methodologies and Trends. International Journal of Interactive Mobile Technologies* , 1-13.
- DARIYO, A. (2003). *Psikologi Perkembangan Dewasa Muda*. Jakarta: PT. Grasindo.
- Fowler, F. M. (2019). *Navigating Hybrid Scrum Environments: Understanding the Essentials, Avoiding the Pitfalls*. Sunnyvale: Apress Media LLC.
- GeeksforGeeks. (2025, August 13). What is LSTM - Long Short Term Memory? *GeeksforGeeks*, p. 1.
- Hakim, M. A., & Aristawati, N. V. (2023). Mengukur Depresi, Kecemasan, dan Stres pada Kelompok Dewasa Awal di Indonesia: Uji Validitas dan Reliabilitas Konstruk DASS-21. *Jurnal Psikologi Ulayat*, 1-17.
- Hamenda, T. F., Moo, S., Hunowu, R., & Soeli, Y. M. (2022). Pengembangan Deteksi Dini Gangguan Kesehatan Jiwa Dengan Mental Health Screening . *Jambura Nursing Journal* , 2-7.
- Kementerian, K. (2024). *Pentingnya Kesehatan Mental bagi Remaja dan Cara Menghadapinya*. Jakarta: Kementerian Kesehatan.
- Olah, C. (2015, August 27). Understanding LSTM Networks. *colah's blog*, p. 1.
- Priharsanto, A. S., & Ajhari, A. A. (2024). Pengembangan Aplikasi Kesehatan Mental Berbasis Kotlin: Studi Kasus Aplikasi HugMe. *JURNAL KAJIAN TEKNIK ELEKTRO* , 1-7.
- Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game*. California: Creative Commons.
- Situmorang, S., & Utami, W. S. (2024). Aplikasi Konsultasi Mental Health Berbasis Android (Studi Kasus UTY). *JURNAL SISFOTENIKA*, 1-9.
- Sutanto, Amiruddin, D., & Nugraha, G. (2022). RANCANG BANGUN APLIKASI SKRINING KESEHATAN MENTAL REMAJA BERBASIS WEB DI RSUD DR.DRADJAT PRAWIRANEGARA DENGAN MENGGUNAKAN METODE ANALYTIC HIERARCHY PROCESS (AHP). *Jurnal Innovation And Future Technology*, 1-10.
- Wahdi, A. E., Setyawan, A., Putri, Y. A., Wilopo, S. A., Erskine, H. E., Wallis, K., . . . Ramaiya, A. (2022, Oktober). *Indonesia – National Adolescent Mental Health Survey (I-NAMHS) Report*. Sleman: Pusat Kesehatan Reproduksi.

World Health, O. (2022). *World Mental Health Report: Transforming Mental Health for All*. Geneva: World Health Organization.

DAFTAR LAMPIRAN

Lampiran A. Dokumen Proses Pengumpulan Requirement

Folder berikut memuat dokumen-dokumen terkait proses pengumpulan kebutuhan sistem Mentalku, serta dokumentasi pendukung lainnya.

- https://polibatam.id/dokumen-proses-pengumpulan-Requirement_zidanmikvan

Lampiran B. Link Produk

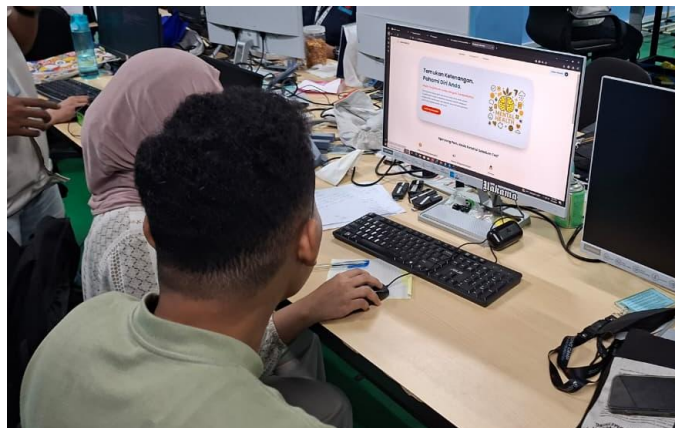
Kode sumber aplikasi Mentalku dapat diakses melalui repositori berikut.

- <https://github.com/zidanikvan22/mentalku-web>
- <https://github.com/zidanikvan22/mentalku-ml-api>

Lampiran C. Dokumen Pengujian

Folder berikut memuat dokumen-dokumen hasil pengujian sistem Mentalku, serta dokumentasi pendukung lainnya.

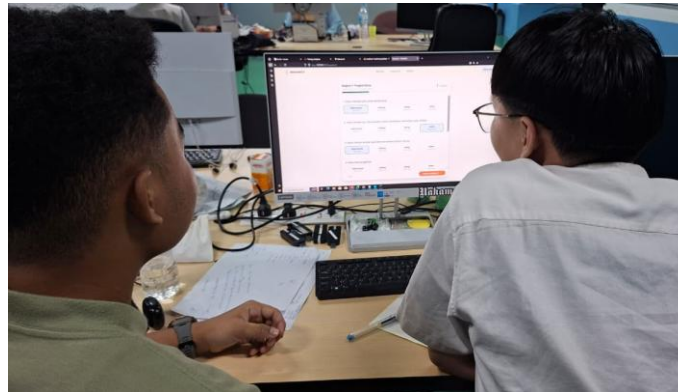
- https://polibatam.id/dokumen-pengujian_zidanmikvan



Gambar 48. User Acceptance Testing bersama Eka Fitri



Gambar 49. User Acceptance Testing bersama Masdika



Gambar 50. User Acceptance Testing bersama Davin



Gambar 51. User Acceptance Testing bersama Febby