

RANCANG BANGUN APLIKASI EDUKASI KESEHATAN MENTAL MENGGUNAKAN ALGORITMA BERT BERBASIS ANDROID



AHMAD FAUZAN MANSUR

H071201001



**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR**

2024

**RANCANG BANGUN APLIKASI EDUKASI
KESEHATAN MENTAL MENGGUNAKAN
ALGORITMA BERT BERBASIS ANDROID**

**AHMAD FAUZAN MANSUR
H071201001**



**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR
2024**

**RANCANG BANGUN APLIKASI EDUKASI
KESEHATAN MENTAL MENGGUNAKAN
ALGORITMA BERT BERBASIS ANDROID**

AHMAD FAUZAN MANSUR
H071201001

Skripsi

sebagai salah satu syarat untuk mencapai gelar sarjana

Program Studi Sistem Informasi

pada

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR
2024**

SKRIPSI
RANCANG BANGUN APLIKASI
EDUKASI KESEHATAN MENTAL
MENGGUNAKAN ALGORITMA BERT BERBASIS ANDROID

AHMAD FAUZAN MANSUR

H071201001

Skripsi,

telah dipertahankan di depan Panitia Ujian Sarjana Sistem Informasi
pada 6 September 2024

dan dinyatakan telah memenuhi syarat kelulusan

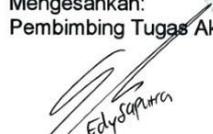
pada

Program Studi Sistem Informasi
Departemen Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Hasanuddin
Makassar



Mengesahkan:
Pembimbing Tugas Akhir,

Pembimbing Pendamping,


A. Muh. Amil Siddik, S.Si., M.Si
NIP. 19911003 201903 1 015


Edy Saputra Rusdi, S.Si., M.Si
NIP. 19910410 202005 3 001.

Mengetahui:
Ketua Program Studi


Prof. Drs. Jeffrey Kusuma, Ph.D
NIP. 19641112 198703 1 002

PERNYATAAN KEASLIAN SKRIPSI DAN PELIMPAHAN HAK CIPTA

Dengan ini saya menyatakan bahwa, skripsi berjudul "RANCANG BANGUN APLIKASI EDUKASI KESEHATAN MENTAL MENGGUNAKAN ALGORITMA BERTERBASIS ANDROID" adalah benar karya saya dengan arahan dari pembimbing A. Muh. Amil Siddik, S.Si.,M.Si sebagai Pembimbing Utama dan Edy Saputra Rusdi, S.Si., M.Si sebagai Pembimbing Pendamping. Karya ilmiah ini belum diajukan dan tidak sedang diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka skripsi ini. Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini adalah karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut berdasarkan aturan yang berlaku.

Dengan ini saya melimpahkan hak cipta (hak ekonomis) dari karya tulis saya berupa skripsi ini kepada Universitas Hasanuddin.

Makassar, 23 September 2024



Ahmad Fauzan Mansur
H071201001

UCAPAN TERIMA KASIH

Puji dan syukur penulis panjatkan kepada Allah Subhanahu Wa Ta'ala atas segala pertolongan, rahmat, dan kasih sayang-Nya yang senantiasa tercurah kepada penulis hingga saat ini. Shalawat serta salam semoga selalu tercurahkan kepada Nabi Muhammad Shallallahu 'Alaihi Wa Salam. Alhamdulillah, berkat nikmat dan pertolongan dari-Nya, penulis dapat menyelesaikan skripsi ini yang berjudul **“Rancang Bangun Aplikasi Edukasi Kesehatan Mental Menggunakan Algoritma BERT Berbasis Android”**.

Dalam menyusun skripsi ini, penulis tidak luput dari berbagai kesulitan dan hambatan, namun atas bantuan dan dorongan dari berbagai pihak akhirnya penulisan skripsi ini dapat terselesaikan.

Untuk itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya dan penghargaan yang setinggi-tingginya kepada semua pihak yang telah membantu serta mendukung penulis dalam menyusun dan menyelesaikan skripsi ini, yaitu kepada:

1. Rektor Universitas Hasanuddin, Bapak **Prof. Dr. Ir. Jamaluddin Jompa, M. Sc.**, beserta jajarannya.
2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam, Bapak **Dr. Eng. Amiruddin, S.Si., M.Si.**, beserta jajarannya.
3. Ketua Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam, Bapak **Dr. Firman, S.Si., M.Si.**, beserta jajarannya.
4. Ketua Program Studi Sistem Informasi Bapak **Prof. Drs. Jeffry Kusuma, Ph.D**, beserta jajarannya.
5. Dosen pembimbing utama Bapak **A. Muh. Amil Siddik, S.Si.,M.Si.** yang telah memberikan arahan, bimbingan, dan panduan selama proses penyusunan skripsi.
6. Dosen pembimbing kedua Bapak **Edy Saputra, S.Si., M.Si.** yang telah meluangkan waktu untuk memberikan konsultasi selama proses penyusunan skripsi.
7. Dosen penguji pertama Bapak **Muhammad Sadno, S.Si., M.Si.** yang bersedia meluangkan waktu untuk menguji, memberikan kritik dan saran yang bermanfaat dalam perbaikan skripsi ini.
8. Dosen penguji kedua sekaligus penasehat akademik Bapak **Dr. Hendra, S.Si, M.Kom.** yang telah meluangkan waktu untuk menguji, dan memberikan segala saran selama seminar hasil dan ujian sidang.
9. Keluarga penulis, yaitu Ayahanda **Mansur Makkasau** dan Ibunda **Rosmawati Achmad** yang telah membesarkan dan mendidik penulis serta memberikan dukungan baik secara psikis maupun materiil, bimbingan, dan motivasi selama proses penyusunan skripsi ini.
10. Kakak **Ahmad Farid Mansur** yang telah memberikan bantuan finansial selama pengerjaan skripsi ini.

11. Sahabat **Galaxy Developer** yaitu Mamet, Haerul, Azhar, Faizah, dan Iman yang telah menjadi tempat berbagi informasi, teman belajar, hingga tempat keluh kesah selama perkuliahan.
12. Teman-teman dari **Bangkit Academy 2023 Batch 1**, yaitu Ifan, Thariq, Gabriel, Tiffany, Daffa, dan Pandu yang memberikan motivasi dan ide kepada penulis mengenai skripsi ini.
13. Teman magang di **X-Camp**, yaitu Mas Yusuf, Vincent, Ishom, dan Tunjung yang secara tidak langsung memberikan bantuan emosional dan motivasi kepada penulis selama pengerjaan skripsi ini.
14. **Laboratorium Sistem Informasi** sebagai tempat berbagi ilmu dan informasi yang berharga selama masa perkuliahan.
15. Teman-teman **KKNT 111 ITTG Gowa 2 Posko Erelembang**, yaitu Jou, Ai, Nisa, Catherine, Yaya, dan Surya atas kerja samanya dan bantuannya selama pengerjaan skripsi ini.
16. Seluruh teman-teman **Sistem Informasi 2020** yang telah menjadi teman seperjuangan dari masa kuliah daring hingga akhir perkuliahan di Universitas Hasanuddin.

Penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna, oleh karena itu penulis ingin meminta maaf serta berikan segala kritik dan saran agar penulis bisa menjadi lebih baik lagi. Penulis berharap skripsi ini akan memberikan manfaat bagi penulis dan pembaca, Aamiin.

Penulis,
Ahmad Fauzan Mansur

ABSTRAK

AHMAD FAUZAN MANSUR. **Rancang Bangun Aplikasi Edukasi Kesehatan Mental Menggunakan Algoritma BERT Berbasis Android** (dibimbing oleh A. Muh. Amil Siddik dan Edy Saputra Rusdi).

Latar belakang. Kesehatan mental merupakan isu global dengan dampak signifikan, terutama di Indonesia, di mana gangguan mental emosional seperti depresi dan kecemasan semakin meningkat. Meskipun upaya penanganan telah dilakukan, masih dibutuhkan lebih banyak edukasi untuk meningkatkan kesadaran masyarakat. Penelitian ini bertujuan untuk merancang dan mengembangkan aplikasi Android yang memberikan edukasi kesehatan mental berbasis kecerdasan buatan. **Metode.** Penelitian ini menggunakan metode pengembangan air terjun, mencakup analisis kebutuhan, desain, pengkodean, integrasi, pengujian, dan pemeliharaan sistem. Aplikasi ini mengimplementasikan algoritma *Bidirectional Encoder Representations from Transformers* untuk analisis sentimen, memungkinkan deteksi emosi dan memberikan respons yang relevan. Pengembangan melibatkan pembuatan model machine learning di Google Colaboratory, backend dengan FastAPI dan ExpressJS, serta frontend di Android Studio. **Hasil.** Aplikasi berhasil memberikan edukasi kesehatan mental dengan fitur chatbot yang menganalisis sentimen pengguna menggunakan BERT. Model ini dioptimalkan melalui fine-tuning dan dievaluasi menggunakan metrik seperti akurasi, recall, precision, dan F1-score, menunjukkan kinerja yang memadai. **Kesimpulan.** Aplikasi ini diharapkan dapat meningkatkan pemahaman masyarakat tentang kesehatan mental dan berfungsi sebagai referensi untuk penelitian lanjutan.

Kata Kunci: Kesehatan Mental, BERT, Android, Edukasi, Kecerdasan Buatan.

ABSTRACT

AHMAD FAUZAN MANSUR. **Design and Development of a Mental Health Education Application Using BERT Algorithm on Android** (supervised by A. Muh Amil Siddik and Edy Saputra Rusdi).

Background. Mental health is a global issue with significant impact, especially in Indonesia, where emotional mental disorders such as depression and anxiety are on the rise. Although efforts have been made to address these issues, there is still a need for more education to raise public awareness. This study aims to design and develop an Android application that provides mental health education based on artificial intelligence. **Methods.** This research adopts the waterfall development method, covering requirement analysis, system design, coding, integration, testing, and system maintenance. The application implements the Bidirectional Encoder Representations from Transformers algorithm for sentiment analysis, allowing emotion detection and providing relevant responses. The development process involved creating a machine learning model on Google Colaboratory, backend development using FastAPI and ExpressJS, and frontend development on Android Studio. **Results.** The application successfully delivers mental health education with a chatbot feature that analyzes user sentiment using BERT. The model was optimized through fine-tuning and evaluated using metrics such as accuracy, recall, precision, and F1-score, showing satisfactory performance. **Conclusion.** This application is expected to enhance public understanding of mental health and serve as a reference for further research in the same field.

Keywords: Mental Health, BERT, Android, Education, Artificial Intelligence.

DAFTAR ISI

UCAPAN TERIMA KASIH	vi
ABSTRAK	viii
ABSTRACT	ix
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan dan Manfaat Penelitian	2
1.2.1 Rumusan Masalah	2
1.2.2 Tujuan	2
1.2.3 Manfaat	3
1.3 Landasan Teori	3
1.3.1 Rancang Bangun	3
1.3.2 Aplikasi Berbasis Android	3
1.3.3 Edukasi Kesehatan Mental	4
1.3.4 BERT (<i>Bidirectional Encoder Representations from Transformers</i>)	5
1.3.5 Analisis Sentimen	6
1.3.6 F1-Score	8
1.3.7 API (<i>Application Programming Interface</i>)	9
1.3.8 TypeScript	9
1.3.9 Python	10
1.3.10 Kotlin	10
1.3.11 FastAPI	11
1.3.12 Express	11
1.3.13 Google Colab	11
1.3.14 Visual Studio Code	12
1.3.15 Android Studio	12
1.3.16 Jetpack Compose	12
1.3.17 Model-View-ViewModel (MVVM)	13
1.3.18 Black Box Testing	14
1.3.19 Gemini API	14
1.3.20 Use Case Diagram	15
1.3.21 Activity Diagram	17
1.3.22 Architecture Diagram	17
BAB II METODE PENELITIAN	19
2.1 Tempat Penelitian	19
2.2 Metode Pengembangan	19
2.3 Waktu dan Jadwal Penelitian	20
2.4 Analisis Kebutuhan Sistem	21

2.4.1 Analisis Kebutuhan Perangkat Lunak	21
2.4.2 Analisis Kebutuhan Perangkat Keras	22
2.5 Desain dan Perancangan Sistem	22
2.5.1 Use Case Diagram	22
2.5.2 Activity Diagram	24
2.5.3 Perancangan Tampilan Antarmuka Aplikasi	31
2.5.4 Diagram Arsitektur Aplikasi	35
2.5.5 Perancangan Basis Data	36
2.5.6 Perancangan <i>Machine Learning</i>	37
BAB III HASIL DAN PEMBAHASAN	39
3.1 Implementasi Model Machine Learning	39
3.1.1 Data Collection	39
3.1.2 Data Pre-processing	39
3.1.3 Model Fine-tuning	40
3.1.4 Model Evaluation	43
3.2 Implementasi Backend FastAPI	45
3.2.1 Inisialisasi FastAPI	45
3.2.2 Integrasi Machine Learning	45
3.2.3 Integrasi Gemini API	46
3.2.4 Pembuatan Endpoint	47
3.2.5 Menjalankan Server FastAPI	49
3.3 Implementasi Backend Express	50
3.3.1 Inisialisasi Express	50
3.3.2 Integrasi Database PostgreSQL	51
3.3.3 Pembuatan Middleware	53
3.3.4 Pembuatan Endpoint	55
3.3.5 Menjalankan Server Express	57
3.4 Implementasi Frontend Mobile	57
3.4.1 Integrasi Dengan Backend	58
3.4.2 Implementasi Antarmuka Menggunakan Jetpack Compose	60
3.4.3 Hasil Implementasi	66
3.5 Pengujian Sistem	79
BAB IV KESIMPULAN DAN SARAN	81
4.1 Kesimpulan	81
4.2 Saran	81
DAFTAR PUSTAKA	82
LAMPIRAN	85

DAFTAR TABEL

Nomor Urut	Halaman
Tabel 1 Komponen activity diagram	17
Tabel 2 Jadwal penelitian	20
Tabel 3 Kebutuhan perangkat Lunak	21
Tabel 4 Spesifikasi komputer yang digunakan	22
Tabel 5 Spesifikasi Android yang digunakan	22
Tabel 6 Daftar <i>endpoint</i> pada <i>server</i> FastAPI	47
Tabel 7 Contoh <i>request</i> dan <i>response endpoint</i> FastAPI	49
Tabel 8 Daftar <i>endpoint</i> pada sistem <i>backend</i> Express	55
Tabel 9 Daftar <i>ViewModel</i> pada aplikasi	61
Tabel 10 Daftar <i>screen</i> pada aplikasi	63
Tabel 11 Hasil uji aplikasi	80

DAFTAR GAMBAR

Nomor Urut	Halaman
Gambar 1 BERT model <i>flowchart</i> (Ganesh et al., 2021)	5
Gambar 2 Proses analisis sentimen menggunakan BERT (Sadia & Basak, 2021) .	7
Gambar 3 <i>Confusion matrix</i> (Narkhede, 2018)	8
Gambar 4 Komponen MVVM (Wikipedia, 2021)	13
Gambar 5 Alur kerja Gemini API (Google Codelabs, 2024)	15
Gambar 6 Komponen use case diagram (Nicki, 2013)	16
Gambar 8 Contoh <i>architecture diagram</i> (Amazon, 2023)	18
Gambar 9 Metode <i>waterfall</i>	19
Gambar 10 <i>Use case diagram</i>	23
Gambar 11 <i>Activity diagram</i> registrasi	24
Gambar 12 <i>Activity diagram</i> log in	25
Gambar 13 <i>Activity diagram</i> membaca <i>news</i>	26
Gambar 14 <i>Activity diagram</i> membuat <i>thread</i>	27
Gambar 15 <i>Activity diagram</i> baca <i>thread</i>	28
Gambar 16 <i>Activity diagram</i> lihat profil	29
Gambar 17 <i>Activity diagram</i> bercakap dengan <i>chatbot</i>	30
Gambar 18 <i>Activity diagram</i> log out	31
Gambar 19 <i>Wireframe</i> aplikasi	32
Gambar 20 <i>Design system</i> aplikasi	32
Gambar 21 <i>Mockup</i> halaman <i>welcome</i> , <i>login</i> , dan <i>home</i>	33
Gambar 22 <i>Mockup</i> halaman <i>explore</i> , <i>profile</i> , dan <i>chat</i>	34
Gambar 23 Diagram arsitektur aplikasi	35
Gambar 24 Rancangan basis data	36
Gambar 25 Diagram alur implementasi <i>machine learning</i> pada aplikasi	38
Gambar 26 Isi <i>dataset</i>	39
Gambar 27 Insialisasi <i>function</i> pembersihan data	40
Gambar 28 Contoh metode <i>split data</i>	41
Gambar 29 <i>Model training parameter</i>	41
Gambar 30 <i>Function</i> untuk <i>custom metrics</i>	42
Gambar 31 Tahap <i>fine-tuning model</i>	43
Gambar 32 Evaluasi <i>model pre-trained</i>	44
Gambar 33 Evaluasi <i>model fine-tuned</i>	44
Gambar 34 Contoh kode inialisasi FastAPI	45
Gambar 35 Inialisasi fungsi memuat model dan cek <i>intent</i>	46
Gambar 36 Integrasi Gemini API	47
Gambar 37 Implementasi fungsi <i>send_chat</i>	48
Gambar 38 Menjalankan <i>server</i> FastAPI	49
Gambar 39 Daftar dependensi <i>backend</i> Express	50
Gambar 40 Contoh kode inialisasi Express	51
Gambar 41 Contoh <i>file</i> Dotenv untuk backend Express	51

Gambar 42	Inisialisasi <i>database backend</i> Express	52
Gambar 43	Contoh kode <i>models</i> untuk tabel <i>threads</i>	52
Gambar 44	Contoh <i>script</i> dari <i>generate</i> drizzle	53
Gambar 45	Contoh <i>requestValidator</i> untuk <i>endpoint register</i>	54
Gambar 46	Contoh kode <i>verifyToken</i>	54
Gambar 47	Contoh kode <i>wildcardRoute</i>	55
Gambar 48	Contoh kode <i>endpoint /threads</i>	56
Gambar 49	Contoh <i>middleware routes</i>	57
Gambar 50	Menjalankan <i>server</i> Express	57
Gambar 51	Pendefinisian <i>ApiService</i>	58
Gambar 52	Pendefinisian <i>ApiConfig</i>	59
Gambar 53	Contoh kode <i>repository</i>	60
Gambar 54	Contoh kode <i>viewmodel</i> untuk <i>ExploreViewModel</i>	62
Gambar 55	Contoh kode bagian <i>content</i> untuk <i>MoreNewsScreen</i>	64
Gambar 56	Contoh kode bagian <i>screen</i> untuk <i>MoreNewsScreen</i>	65
Gambar 57	Halaman <i>StartScreen</i>	66
Gambar 58	Halaman <i>SignUpScreen</i>	67
Gambar 59	Halaman <i>SignInScreen</i>	68
Gambar 60	Halaman <i>HomeScreen</i>	69
Gambar 61	Halaman <i>ExploreScreen</i>	70
Gambar 62	Halaman <i>ProfileScreen</i>	71
Gambar 63	Halaman <i>MoreNewsScreen</i>	72
Gambar 64	Halaman <i>MoreThreadsScreen</i>	73
Gambar 65	Halaman <i>NewsScreen</i>	74
Gambar 66	Halaman <i>ThreadsScreen</i>	75
Gambar 67	Halaman <i>ChatScreen</i>	76
Gambar 68	Halaman <i>ChatResultScreen</i>	77
Gambar 69	Halaman <i>SettingsScreen</i>	78
Gambar 70	Halaman <i>EditScreen</i>	79

BAB I PENDAHULUAN

1.1 Latar Belakang

Menurut WHO, sekitar 80% hingga 90% kasus bunuh diri disebabkan oleh depresi dan kecemasan. Sedangkan, data Riskesdas 2018 juga menunjukkan bahwa prevalensi gangguan mental emosional yang ditunjukkan dengan gejala-gejala depresi dan kecemasan untuk usia 15 tahun ke atas mencapai sekitar 6,1% dari jumlah penduduk Indonesia atau setara dengan 11 juta orang.

Kedua data tersebut menunjukkan bahwa kesehatan mental merupakan hal yang penting bagi kehidupan seorang individu. Kesejahteraan mental seseorang tidak hanya mempengaruhi kehidupan seseorang secara pribadi, tetapi juga memiliki dampak yang signifikan pada masyarakat secara keseluruhan. Di seluruh dunia, masalah kesehatan mental telah menjadi isu yang mendesak. Isu ini tidak dapat diabaikan karena dampaknya yang merambah ke berbagai aspek lain dalam masyarakat, termasuk aspek sosial dan ekonomi. Sayangnya, di negara-negara berkembang, termasuk Indonesia, pemecahan masalah penyakit mental belum mendapat prioritas yang seharusnya (Rahvy et al., 2020).

Pemerintah di Indonesia maupun seluruh dunia telah melakukan upaya untuk menangani masalah mental. Beberapa di antaranya penetapan Undang-Undang No 18 Tahun 2014 tentang Kesehatan Jiwa, yang mengatur upaya promotif, preventif, kuratif, dan rehabilitatif dalam penanganan kesehatan jiwa. Salah satu upaya rehabilitatif dari pemerintah pusat adalah pembangunan Rumah Sakit Jiwa di 7 provinsi yang belum memiliki rumah sakit jiwa, yaitu: Banten, Kalimantan Utara, Sulawesi Tenggara, Kepulauan Riau, Sulawesi Utara, Papua Barat dan Gorontalo.

Upaya saat ini difokuskan terhadap upaya promotif dan preventif. Individu yang sehat maupun dengan penyakit kronis menjadi fokus upaya preventif kesehatan jiwa. Upaya tersebut tidak hanya dilakukan oleh tenaga kesehatan namun juga menjadi tanggung jawab masyarakat. Maka dari itu perlu pemberdayaan masyarakat di mulai dari menanamkan pengetahuan, kesadaran, perilaku sehat jiwa dan kepedulian terhadap masalah-masalah kesehatan jiwa (Winahayu, et al., 2016).

Motivasi utama dalam penelitian ini berkaitan dengan upaya-upaya yang disebutkan sebelumnya. Penulis merasa bahwa sangat penting untuk memberikan pendidikan kepada masyarakat tentang kesehatan mental. Data sebelumnya juga menunjukkan bahwa kesehatan mental memiliki dampak signifikan pada kehidupan seseorang, dan jika tidak ditangani dengan baik, dapat berakibat fatal. Penulis juga mengakui urgensi memanfaatkan teknologi dalam memecahkan masalah ini. Mengutip dari Askarasoft, digambarkan bahwa penggunaan teknologi menawarkan sejumlah keunggulan dibandingkan metode konvensional. Keunggulan tersebut meliputi efisiensi waktu dan biaya, peningkatan aksesibilitas, serta memberikan peluang inovasi yang lebih besar. Dalam kasus ini, penulis dapat memanfaatkan teknologi sebagai media edukasi kesehatan mental.

Pada beberapa tahun belakangan ini, teknologi mengalami perkembangan yang sangat pesat. Salah satu contoh teknologi bakalan saat ini adalah kecerdasan buatan (*artificial intelligence*). Kecerdasan buatan merupakan bidang teknologi yang dirancang untuk meniru kemampuan intelektual manusia. Penerapan dari kecerdasan buatan sendiri sudah mencakup berbagai bidang seperti bidang manufaktur, hiburan, telekomunikasi, militer, pendidikan, kesehatan, dan lain-lain.

Dalam bidang kesehatan secara umumnya, kecerdasan buatan sudah sering diimplementasikan, salah satu contohnya dalam perlakuan diagnosis medis yang lebih akurat. Kecerdasan buatan dapat dimanfaatkan pada bidang kesehatan mental. Studi sebelumnya sudah menerapkan *Natural Language Processing* dalam bentuk *chatbot* pada aplikasi edukasi kesehatan mental berbasis web (Setiawan, 2023) dan perancangan aplikasi pendeteksian wajah berdasarkan kesehatan mental (Marsujitullah et al., 2022). Pada studi sebelumnya, kecerdasan buatan diimplementasikan dalam sebuah aplikasi berbasis web dan juga melakukan pendiagnosaan menggunakan metode *viola-jones*. Oleh karena itu, penulis memutuskan untuk membuat skripsi yang berjudul "Rancang Bangun Aplikasi Edukasi Kesehatan Mental Menggunakan Algoritma BERT Berbasis Android".

1.2 Tujuan dan Manfaat Penelitian

1.2.1 Rumusan Masalah

Rumusan masalah dari penelitian ini sebagai berikut.

- a. Bagaimana cara merancang aplikasi konsultasi kesehatan mental pada platform Android?
- b. Bagaimana cara membangun aplikasi konsultasi kesehatan mental pada platform Android?
- c. Bagaimana cara mengimplementasikan algoritma BERT pada aplikasi konsultasi kesehatan mental pada platform Android?
- d. Bagaimana kinerja dari model BERT pada aplikasi konsultasi kesehatan mental pada platform Android?

1.2.2 Batasan Masalah

Batasan masalah dari penelitian ini sebagai berikut.

- a. Aplikasi akan tersedia dalam bahasa Inggris.
- b. Aplikasi akan tersedia pada platform Android.

1.2.3 Tujuan

Adapun tujuan dari penelitian ini sebagai berikut.

- a. Merancang aplikasi konsultasi kesehatan mental pada platform Android.
- b. Membangun aplikasi konsultasi kesehatan mental pada platform Android.

- c. Mengimplementasikan algoritma BERT pada aplikasi konsultasi kesehatan mental pada platform Android.
- d. Mengukur kinerja dari model BERT pada aplikasi konsultasi kesehatan mental pada platform Android.

1.2.4 Manfaat

Manfaat dari penelitian adalah sebagai berikut.

- a. Diharapkan aplikasi konsultasi kesehatan mental berbasis Android ini akan meningkatkan kualitas kesehatan mental penggunanya.
- b. Menjadi referensi untuk penelitian berikutnya.

1.3 Landasan Teori

1.3.1 Rancang Bangun

Menurut Kamus Besar Bahasa Indonesia kata rancang berarti mengatur segala sesuatu sebelum bertindak mengerjakan atau melakukan sesuatu untuk merencanakan. Sedangkan kata bangun berarti sesuatu yang didirikan atau berdiri.

Rancang bangun adalah kegiatan menerjemahkan hasil analisis ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut ataupun memperbaiki sistem yang sudah ada (Gunawan et al., 2021). Berdasarkan uraian menurut ahli tersebut, dapat disimpulkan bahwa rancang bangun merupakan proses merencanakan, mengatur, dan menerjemahkan hasil analisis ke dalam bentuk perangkat lunak atau sistem yang akan dibangun atau diperbaiki. Proses ini melibatkan tahap perencanaan sebelum bertindak, termasuk pengaturan segala sesuatu yang diperlukan sebelum melaksanakan tugas pengembangan perangkat lunak atau sistem. Dengan demikian, rancang bangun menjadi landasan penting dalam menghasilkan solusi teknologi yang efektif dan efisien.

1.3.2 Aplikasi Berbasis Android

Aplikasi merupakan program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu (Yuhefizar, 2012). Sedangkan, aplikasi Android menyediakan platform secara terbuka bagi para pengguna, pengembang dalam menciptakan berbagai bentuk aplikasi yang mereka inginkan. Aplikasi ini bisa dalam bentuk pengetahuan, gim, pendidikan, agama, dan lain sebagainya (Murtiwyati & Lauren, 2013). Berdasarkan target penggunanya, aplikasi Android dapat diklasifikasikan menjadi:

- a. Aplikasi umum, yaitu aplikasi yang dapat digunakan oleh semua orang, seperti aplikasi *browser*, aplikasi peta, dan aplikasi kamera.

- b. Aplikasi khusus, yaitu aplikasi yang dirancang untuk memenuhi kebutuhan pengguna tertentu, seperti aplikasi pendidikan, aplikasi kesehatan, dan aplikasi bisnis.

Berdasarkan uraian ahli tersebut, penulis menyimpulkan bahwa aplikasi adalah program yang dirancang untuk memenuhi kebutuhan pengguna dalam menjalankan tugas-tugas tertentu. Aplikasi Android, khususnya, memberikan platform terbuka bagi pengguna dan pengembang untuk menciptakan berbagai jenis aplikasi sesuai dengan preferensi dan kebutuhan masing-masing. Aplikasi ini dapat berkisar dari aplikasi pengetahuan, gim, aplikasi pendidikan, aplikasi agama, dan beragam jenis lainnya. Dengan demikian, aplikasi Android menjadi wadah kreativitas yang luas, memungkinkan berbagai inovasi dan solusi untuk dapat diakses dan digunakan oleh banyak orang.

1.3.3 Edukasi Kesehatan Mental

Kesehatan mental adalah keadaan kesejahteraan mental yang memungkinkan orang untuk mengatasi tekanan hidup, menyadari kemampuan mereka, belajar dengan baik dan bekerja dengan baik, serta berkontribusi bagi komunitas mereka. Ini merupakan komponen integral dari kesehatan dan kesejahteraan yang mendukung kemampuan individu dan kolektif kita untuk membuat keputusan, membangun hubungan, dan membentuk dunia tempat kita tinggal. Kesehatan mental adalah hak asasi manusia dasar. Dan itu sangat penting untuk pengembangan pribadi, komunitas, dan sosial-ekonomi. (World Health Organization, 2022)

Edukasi kesehatan adalah usaha terencana untuk menyebarkan pengaruh terhadap kesehatan orang lain baik individu, kelompok, atau masyarakat, sehingga perilaku sasaran dapat sesuai dengan yang diharapkan oleh pemberi edukasi dan promosi kesehatan. Di dalam definisi ini juga terkandung poin-poin meliputi input (pendidik dan sasaran edukasi kesehatan), proses (rencana dan strategi), serta output (melakukan sesuai dengan yang diharapkan). Hasil yang diharapkan dari edukasi atau promosi kesehatan adalah peningkatan perilaku sehingga terpeliharanya kesehatan oleh sasaran dari edukasi kesehatan (Notoatmodjo, 2012). Jadi, edukasi kesehatan mental dapat didefinisikan sebagai upaya terencana untuk menyebarkan pengaruh positif terhadap kesehatan mental individu, kelompok, atau masyarakat secara keseluruhan.

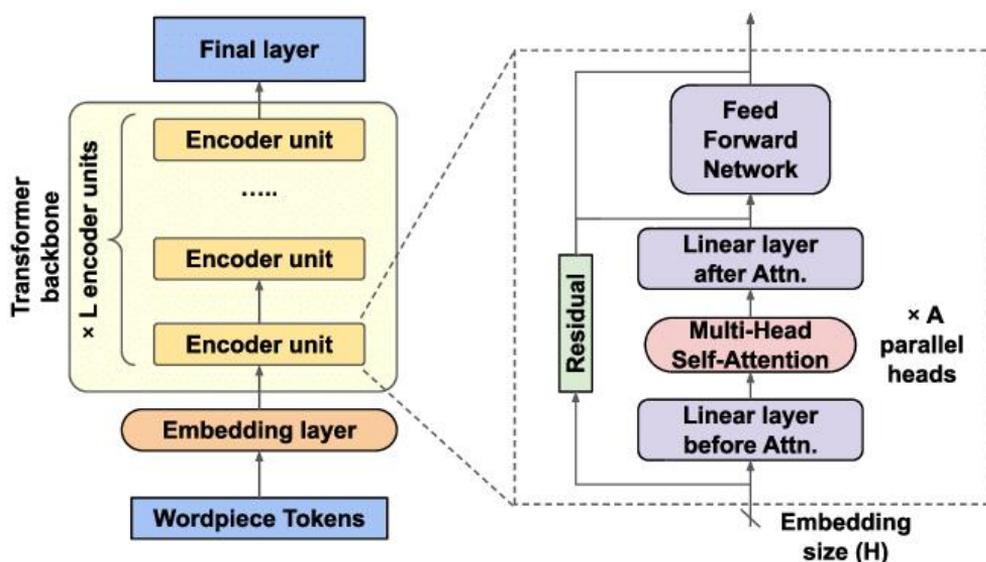
Edukasi kesehatan mental melibatkan pemberian informasi, pemahaman, dan keterampilan yang diperlukan untuk mengatasi tekanan hidup, meningkatkan kesadaran akan kesejahteraan mental, dan membantu individu dalam memahami kemampuan serta potensinya. Tujuannya adalah agar individu dapat belajar dengan baik, bekerja secara optimal, berkontribusi positif bagi komunitas mereka, dan menjalani kehidupan yang seimbang. Pada intinya, edukasi kesehatan mental

bertujuan untuk mempromosikan hak asasi manusia dasar terkait kesehatan mental serta mendukung perkembangan pribadi, sosial, dan ekonomi yang berkelanjutan.

1.3.4 BERT (*Bidirectional Encoder Representations from Transformers*)

Bidirectional Encoder Representation from Transformers atau disingkat BERT adalah model representasi bahasa terlatih yang dibuat pada tahun 2018 oleh peneliti Google AI Language. BERT dibuat menggunakan pembelajaran semi-diawasi, ELMo, ULMFiT, OpenAI Transformers, serta teknik *deep learning*. (Kurniawan, 2022)

Algoritma ini adalah menggunakan arsitektur *transformer* dimana model tersebut memproses sebuah kata pada kalimat berdasarkan ada atau tidaknya kaitan antara kata tersebut dengan kalimat secara keseluruhan. BERT akan memproses sebuah kata dengan mempelajari konteks daripada kata tersebut berdasarkan kata-kata yang ada (Sayla et al., 2023). BERT dilatih menggunakan 2500 juta kata dari Wikipedia dan 800 juta kata dari buku. Pelatihan BERT ini tergolong menjadi dua metode modeling yaitu *Masked Language Model* (MLM) dan *Next Sentence Prediction* (NSP). Pada MLM, sebuah kata pada *corpus* akan disembunyikan dan dilatih. Sementara pada NSP, BERT akan memprediksi kalimat selanjutnya yang memiliki konteks sesuai dengan kalimat sebelumnya. (Devlin, 2018)



Gambar 1 BERT model *flowchart* (Ganesh et al., 2021)

Pada tahap awal, setiap token diubah menjadi vektor *embedding* yang menggabungkan informasi *token*, segmen, dan posisi. Vektor ini kemudian diteruskan ke beberapa lapisan *encoder Transformer*. Setiap lapisan *encoder* terdiri

dari dua sub-unit utama: *multi-head, self-attention, dan feed-forward network* (FFN). *Sub-unit, multi-head, self-attention* memungkinkan model untuk memperhatikan berbagai bagian kalimat input secara paralel, membantu BERT memahami hubungan kontekstual yang kompleks. Sementara itu, *feed-forward network* menerapkan transformasi *non-linear* pada representasi vektor yang dihasilkan oleh *sub-unit self-attention*, memperkaya informasi yang disimpan dalam vektor tersebut. Dalam penerapannya, terdapat dua varian BERT (Ganesh et al., 2021).

- a. *BERT-base*, terdiri dari 12 lapisan *encoder* (transformer blocks), 768 ukuran hidden, 12 *attention heads*, dan total 110 juta parameter.
- b. *BERT-large*, memiliki 24 lapisan *encoder*, 1024 ukuran hidden, 16 kepala perhatian, dan total 340 juta parameter.

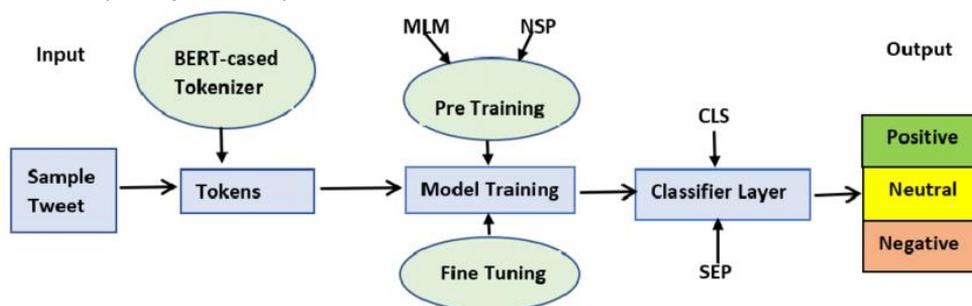
Terdapat dua tahap saat mengimplementasikan BERT: *pre-training* dan *fine-tuning*. Model dilatih menggunakan data yang tidak berlabel melalui berbagai tugas *pre-training* yang berbeda. Selanjutnya, dalam tahap *fine-tuning*, model BERT awalnya diinisialisasi dengan parameter yang telah di-*training* sebelumnya, dan seluruh parameter disesuaikan (*fine-tuned*) menggunakan data yang sudah berlabel dari tugas-tugas khusus yang berbeda, yang disebut sebagai *downstream tasks*. (Devlin, 2018)

1.3.5 Analisis Sentimen

Analisis sentimen adalah proses mengidentifikasi dan mengelompokan opini yang masih berbentuk teks ke dalam sentimen positif atau negatif. Dalam hal ini, penerapan analisis sentimen dapat berupa menganalisa pendapat, sentimen, evaluasi, emosi, penilaian, atau sikap pada suatu produk, tokoh, organisasi, layanan, isu, atau peristiwa yang terjadi di masyarakatnya sendiri. Selain itu, analisis sentimen akan selalu berhubungan dengan masyarakat karena sumber informasi yang didapat dari media sosial dimana masyarakat sebagai penggunanya (Rusdian & Rosiyadi, 2019).

Analisis sentimen merupakan salah satu contoh dari bidang *Natural Language Processing* (NLP) yang paling populer. *Natural Language Processing* (NLP) adalah bidang ilmiah yang membahas tentang bagaimana caranya agar komputer bisa bekerja dan berpikir seperti manusia. *Natural Language Processing* (NLP) merupakan bagian dari *Artificial Intelligence* atau kecerdasan buatan. Dalam perkembangan *data mining*, *Artificial Intelligence* (AI) merupakan salah satu dari empat cabang ilmu *data mining*, yaitu statistika, *database*, dan pencarian informasi. Dalam penerapannya, *Artificial Intelligence* (AI) juga memerlukan machine learning sebagai algoritma penyelesaian. Adanya *machine learning* digunakan untuk menggantikan manusia dalam mengambil keputusan. *Machine learning* tidak mempunyai perasaan seperti manusia sehingga keputusan yang diambil berdasarkan data yang sudah diolah (Saputra, 2022).

Analisis sentimen merupakan proses komputasi menggunakan teknik analisis teks dalam bentuk data tekstual dengan cara mengelola, memahami, dan mengelompokkan emosional baik positif atau negatif. Analisis sentimen banyak digunakan karena meningkatnya kebutuhan individu atau kelompok dalam mengetahui pendapat seseorang terhadap sesuatu. Analisis sentimen juga dipengaruhi oleh *dataset* yang digunakan akan mengalami penanganan yang berbeda (Widayat, 2021).



Gambar 2 Proses analisis sentimen menggunakan BERT (Sadia & Basak, 2021)

Proses analisis sentimen menggunakan BERT dapat dijelaskan sebagai berikut. Pertama, input seperti tweet diambil dan diubah menjadi token menggunakan *BERT-based tokenizer*. Semua *token* ini kemudian diteruskan ke tahap pelatihan model, yang terdiri dari *pre-training* dan *fine-tuning*. Pada tahap *pre-training*, BERT menggunakan strategi *Masked Language Model* (MLM) dan *Next Sentence Prediction* (NSP) untuk melatih model memahami konteks dan hubungan antar kalimat. Setelah *pre-training*, model BERT disesuaikan dengan data berlabel spesifik untuk tugas analisis sentimen melalui *fine-tuning*. Lapisan klasifikasi kemudian ditambahkan ke model BERT untuk melakukan klasifikasi sentimen. *Token* berupa CLS ditempatkan pada awal kalimat pertama dan token berupa SEP ditempatkan di akhir kalimat untuk menandai batas kalimat. Akhirnya, model memprediksi sentimen dari tweet sebagai positif, netral, atau negatif.

Berdasarkan pendapat ahli di atas, penulis menyimpulkan bahwa analisis sentimen merupakan proses identifikasi dan pengelompokan sentimen positif atau negatif dalam data teks yang berkaitan dengan pendapat, emosi, atau penilaian terhadap berbagai entitas. Pendekatan ini berkembang dalam konteks *Natural Language Processing* (NLP), yang merupakan subdomain dalam *Artificial Intelligence* (AI). Analisis sentimen digunakan untuk memahami pandangan individu atau kelompok terhadap berbagai subjek dan sangat dipengaruhi oleh data yang diolah serta digunakan dalam berbagai aplikasi.

1.3.6 F1-Score

F1-score adalah metrik evaluasi dalam kasus klasifikasi yang menggabungkan *precision* dan *recall* menjadi satu nilai yang merepresentasikan kualitas keseluruhan dari model klasifikasi. *F1-score* digunakan untuk mengukur performa model dalam memprediksi suatu kelas. Dalam metrik evaluasi ini, nilai berada pada rentan 0,0 hingga 1,0, dengan 1,0 merepresentasikan nilai terbaik. (Kundu, 2022)

F1-score suatu model dapat dihitung menggunakan rumus berdasar dari sebuah *confusion matrix*. *Confusion matrix* merupakan sebuah tabel yang digunakan untuk mengevaluasi performa dari sebuah model. *Confusion matrix* menunjukkan jumlah prediksi yang benar dan salah yang dibagi menjadi empat kategori, yaitu *true positive (TP)*, *false positive (FP)*, *true negative (TN)*, dan *false negative (FN)*. (Narkhede, 2018)

	Positive	FN
Actual value	TP	FN
Negative	FP	TN
	Positive	Negative
	Predicted value	

Gambar 3 *Confusion matrix* (Narkhede, 2018)

- True Negative (TN)*: Model memprediksi data ada di kelas negatif dan yang sebenarnya data memang ada di kelas negatif.
- True Positive (TP)*: Model memprediksi data ada di kelas positif dan yang sebenarnya data memang ada di kelas positif.
- False Negative (FN)*: Model memprediksi data ada di kelas negatif, namun yang sebenarnya data ada di kelas positif.
- False Positive (FP)*: Model memprediksi data ada di kelas positif, namun yang sebenarnya data ada di kelas negatif.

Nilai-nilai tersebut akan digunakan dalam kalkulasi *precision* dan *recall*.

Precision merupakan perbandingan antara *True Positive (TP)* dengan banyaknya data yang diprediksi positif, secara matematis dituliskan sebagai:

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

Sedangkan *recall* merupakan perbandingan antara *True Positive* (TP) dengan banyaknya data yang sebenarnya positif, secara matematis dituliskan sebagai:

$$recall = \frac{TP}{TP + FN} \quad (2)$$

Setelah memperoleh nilai *precision* dan *recall*, maka sudah dapat dilakukan kalkulasi *F1-score*. Rumus *F1-score* merupakan:

$$F1\ score = 2 * \frac{(precision * recall)}{(precision + recall)} \quad (3)$$

1.3.7 API (*Application Programming Interface*)

Berdasarkan situs IBM, API atau *Application Programming Interface* adalah sebuah protokol yang didefinisikan untuk memungkinkan berbagai aplikasi berbeda untuk saling berkomunikasi satu sama lain. API berfungsi sebagai lapisan perantara yang memproses data dan memungkinkan aplikasi untuk berinteraksi dengan sumber daya atau komponen perangkat lunak lainnya.

Selain itu, API memudahkan desain dan pengembangan aplikasi baru, serta memungkinkan developer untuk mengakses data dan fungsionalitas aplikasi melalui internet (Affrianto, 2022). Menurut situs Midtrans, API dapat dibagi menjadi beberapa jenis menurut arsitekturnya:

- a. REST API adalah jenis API yang paling umum digunakan dan memanfaatkan protokol HTTP untuk mengakses sumber daya atau data. Adapun turunan dari REST API merupakan RESTful API, yang merupakan jenis API layanan web yang menggunakan permintaan HTTP untuk melakukan operasi CRUD (Buat, Baca, Perbarui, Hapus) pada data.
- b. SOAP API adalah jenis API yang menggunakan protokol SOAP (*Simple Object Access Protocol*) untuk mengirim pesan XML antara aplikasi.
- c. Apache Kafka, adalah kumpulan API yang digunakan untuk mengakses dan memanipulasi data pada platform *streaming event*.
- d. *GraphQL* API adalah jenis API yang memungkinkan client untuk meminta data secara spesifik dan hanya mendapatkan data yang dibutuhkan/
- e. *Async* API merupakan sebuah spesifikasi yang digunakan untuk membangun dan memelihara arsitektur berbasis *event-driven*, dan bersifat asinkron sehingga dapat dijalankan bersamaan.
- f. RPC adalah arsitektur API yang memungkinkan aplikasi untuk memanggil fungsi atau prosedur yang berjalan pada server jarak jauh yang menggunakan mekanisme protokol seperti XML-RPC atau JSON-RPC.

1.3.8 TypeScript

TypeScript adalah bahasa pemrograman *open-source* yang dikembangkan dan dipelihara oleh Microsoft, yang pertama kali diperkenalkan pada tahun 2012. TypeScript merupakan *superset* dari JavaScript, yang berarti bahwa setiap kode

JavaScript yang valid juga merupakan kode TypeScript yang valid. Namun, TypeScript memperkenalkan fitur tambahan yang tidak tersedia di JavaScript, seperti tipe statis. Tipe statis memungkinkan pengembang untuk menentukan tipe variabel, fungsi, dan properti objek pada saat pengembangan, yang membantu dalam mendeteksi kesalahan lebih awal dan meningkatkan keterbacaan serta pemeliharaan kode.

Selain itu, TypeScript mendukung fitur-fitur modern JavaScript dan dapat dikompilasi menjadi JavaScript murni yang dapat dijalankan di berbagai lingkungan, termasuk browser *web* dan *server* (Microsoft, 2012). Dengan adopsi yang luas dan dukungan kuat dari komunitas, TypeScript telah menjadi alat penting bagi pengembang yang bekerja pada proyek-proyek besar dan kompleks.

1.3.9 Python

Python adalah bahasa pemrograman tingkat tinggi yang dikembangkan oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991 (Van Rossum & Drake, 2009). Python dikenal karena sintaksnya yang mudah dibaca dan ditulis, serta dukungannya terhadap berbagai paradigma pemrograman seperti pemrograman berorientasi objek, pemrograman fungsional, dan pemrograman imperatif (Lutz, 2013). Salah satu keunggulan utama Python adalah ekosistem pustakanya yang luas, mencakup berbagai bidang seperti pengembangan web, analisis data, kecerdasan buatan, dan komputasi ilmiah (Seiring, 2020). Bahasa ini juga dilengkapi dengan manajer paket yang memudahkan pengguna untuk menginstal dan mengelola pustaka pihak ketiga. Dengan komunitas yang aktif dan terus berkembang, Python telah menjadi salah satu bahasa pemrograman yang paling populer di dunia dan digunakan oleh perusahaan-perusahaan besar serta peneliti di berbagai disiplin ilmu (Pypl, 2023).

1.3.10 Kotlin

Kotlin adalah bahasa pemrograman modern yang dikembangkan oleh JetBrains dan pertama kali diumumkan pada tahun 2011. Kotlin dirancang untuk bekerja dengan mulus bersama Java, sehingga memungkinkan pengembang untuk menulis kode dalam kedua bahasa dalam proyek yang sama tanpa masalah kompatibilitas. Kotlin menawarkan sintaks yang lebih bersih dan lebih ringkas dibandingkan dengan Java, serta memperkenalkan fitur-fitur canggih seperti *null safety*, *data classes*, *extension functions*, dan *coroutines* untuk pemrograman asinkron.

Bahasa ini diadopsi secara resmi oleh *Google* sebagai bahasa pemrograman utama untuk pengembangan aplikasi Android pada tahun 2017, yang mendorong popularitas dan penggunaannya secara luas di kalangan pengembang aplikasi. Kotlin telah menjadi salah satu bahasa pilihan untuk pengembangan aplikasi modern di berbagai platform, termasuk Android, server, dan multiplatform (JetBrains, 2011).

1.3.11 FastAPI

FastAPI adalah kerangka kerja web modern dan cepat untuk membangun API dengan Python, yang pertama kali diperkenalkan oleh Sebastián Ramírez pada tahun 2018. FastAPI dirancang untuk memberikan performa tinggi, mendekati kecepatan Node.js dan Go, dengan efisiensi pengembangan yang tinggi. Salah satu fitur utama FastAPI adalah dukungan tipe anotasi Python 3.6+ yang digunakan untuk menghasilkan validasi data otomatis, dokumentasi interaktif, dan kontrol tipe yang ketat.

Fitur tersebut membuat pengembangan API menjadi lebih cepat, lebih mudah dipelihara, dan lebih dapat diandalkan. FastAPI dibangun di atas Starlette untuk web *microframework* dan Pydantic untuk validasi data, dan mendukung asynchronous programming dengan *async* dan *await*, yang memungkinkan penanganan beban kerja tinggi dengan respons cepat. (Ramírez, 2018).

1.3.12 Express

Express adalah kerangka kerja web minimalis dan fleksibel untuk Node.js yang pertama kali dirilis pada tahun 2010 oleh TJ Holowaychuk (Holowaychuk, 2010). Express menyediakan serangkaian fitur yang kuat untuk pengembangan aplikasi web dan API, seperti *middleware* yang mudah digunakan, sistem routing yang kuat, serta integrasi yang mulus dengan berbagai template *engines* dan *database* (Brown, 2014).

Meskipun sederhana, Express dapat diperluas dan dikustomisasi sesuai kebutuhan, memungkinkan pengembang untuk membangun aplikasi dari skala kecil hingga besar dengan cepat dan efisien (Cantelon et al., 2021). Karena berjalan di atas platform Node.js, Express memanfaatkan arsitektur *non-blocking* dan *event-driven*, yang membuatnya sangat cocok untuk aplikasi yang membutuhkan performa tinggi dan skalabilitas, seperti *real-time web applications*.

1.3.13 Google Colab

Google Colab, atau Google Colaboratory, adalah platform berbasis *cloud* yang disediakan oleh Google untuk membantu pengembang, peneliti, dan ilmuwan data dalam menulis dan menjalankan kode Python melalui notebook interaktif (Google, 2017).

Colab menawarkan *environment* yang mirip dengan Jupyter Notebook, tetapi dengan keunggulan tambahan karena terintegrasi dengan Google Drive, memungkinkan pengguna untuk menyimpan dan berbagi *notebook* mereka dengan mudah. Salah satu fitur paling menonjol dari Google Colab adalah akses gratis ke GPU dan TPU, yang sangat berguna untuk menjalankan tugas-tugas komputasi intensif seperti pelatihan model *machine learning* dan *deep learning*. Colab juga mendukung berbagai pustaka Python yang umum digunakan, seperti TensorFlow,

Keras, PyTorch, dan banyak lagi, serta menyediakan alat-alat kolaborasi yang memudahkan tim untuk bekerja bersama secara real-time. (Bisong, 2019)

1.3.14 Visual Studio Code

Visual Studio Code (VS Code) adalah editor kode sumber terbuka dan gratis yang dikembangkan oleh *Microsoft* dan pertama kali dirilis pada tahun 2015 (Microsoft, 2015). VS Code dirancang untuk mendukung berbagai bahasa pemrograman dan menyediakan fitur-fitur yang kuat seperti penyorotan sintaks, penyelesaian kode otomatis, *debugging*, dan integrasi kontrol versi.

Salah satu keunggulan utama VS Code adalah ekosistem ekstensi yang kaya, yang memungkinkan pengguna untuk menambahkan fungsionalitas tambahan sesuai kebutuhan mereka, mulai dari dukungan bahasa pemrograman hingga alat pengembangan dan framework spesifik. VS Code juga memiliki antarmuka yang intuitif dan ringan, dengan kemampuan kustomisasi yang tinggi, yang membuatnya populer di kalangan pengembang untuk berbagai jenis proyek pengembangan perangkat lunak. VS Code dapat dijalankan di Windows, MacOS, dan Linux, serta memiliki integrasi yang kuat dengan layanan cloud seperti GitHub dan Azure. (Taft, 2019).

1.3.15 Android Studio

Android Studio adalah lingkungan pengembangan terintegrasi (IDE) resmi yang disediakan oleh Google untuk pengembangan aplikasi *Android*, yang pertama kali dirilis pada tahun 2013 (Google, 2013). Android Studio dibangun di atas platform IntelliJ IDEA dari JetBrains dan menyediakan alat-alat yang komprehensif untuk seluruh siklus hidup pengembangan aplikasi Android. Fitur-fitur utamanya meliputi editor kode yang canggih dengan dukungan penyorotan sintaks dan penyelesaian otomatis, *emulator* Android untuk pengujian aplikasi, alat-alat *debugging* yang kuat, serta integrasi dengan sistem kontrol versi seperti Git. Selain itu, Android Studio menyediakan *Layout Editor* visual yang memudahkan pengembang dalam merancang antarmuka pengguna dengan metode *drag-and-drop*. IDE ini juga mendukung integrasi dengan Google Cloud Platform, yang memungkinkan pengembang untuk membangun aplikasi yang terhubung dengan layanan *cloud*.

1.3.16 Jetpack Compose

Jetpack Compose adalah *toolkit* modern untuk membangun antarmuka pengguna (UI) pada aplikasi *Android*, yang dirancang untuk menyederhanakan dan mempercepat proses pengembangan UI dengan pendekatan deklaratif. Dalam Jetpack Compose, UI didefinisikan menggunakan fungsi-fungsi *composable*, yang diberi anotasi `@Composable`, memungkinkan pengembang untuk mendesain elemen UI secara langsung dalam kode Kotlin. Konsep *state* sangat penting dalam

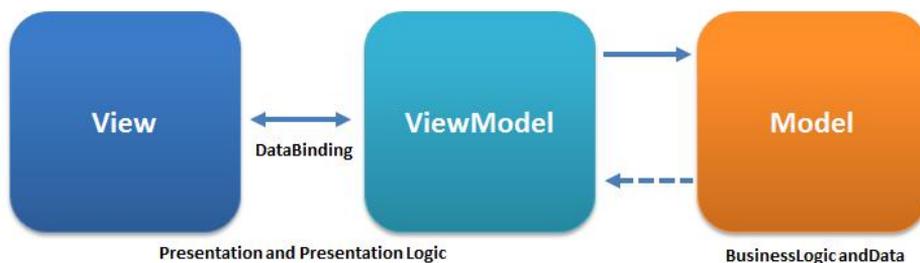
Jetpack Compose, di mana state mengacu pada data yang dapat berubah dan mempengaruhi tampilan UI. Jetpack Compose secara otomatis melakukan *recomposition*, yaitu proses memanggil ulang fungsi *composable* untuk memperbarui UI ketika *state* berubah, sehingga memastikan tampilan aplikasi selalu *up-to-date* (Google, 2020).

Modifier adalah alat dalam Jetpack Compose yang digunakan untuk memodifikasi atau menghias *composable* dengan berbagai cara seperti menambah *padding*, *margin*, dan *ukuran*. *Layouts* seperti *Column*, *Row*, dan *Box* digunakan untuk mengatur tata letak elemen UI dalam aplikasi. Selain itu, Jetpack Compose mendukung penyesuaian tema dan gaya melalui *MaterialTheme* yang memungkinkan konsistensi tampilan dan nuansa di seluruh aplikasi.

Navigasi antar layar dalam aplikasi didukung melalui *NavController* dan *NavHost*, yang memungkinkan pengelolaan rute dan tujuan dalam aplikasi dengan cara yang terstruktur. Kelebihan utama dari Jetpack Compose termasuk pendekatan deklaratif yang memudahkan pengembangan dan pemeliharaan UI, kemampuan untuk menggunakan kembali fungsi *composable* di berbagai bagian aplikasi, serta interoperabilitas dengan tampilan berbasis XML tradisional.

1.3.17 Model-View-ViewModel (MVVM)

Model-View-ViewModel (MVVM) adalah sebuah arsitektur perangkat lunak yang dirancang untuk memisahkan dan mengelola logika bisnis dari logika antarmuka pengguna dalam aplikasi. Arsitektur ini terdiri dari tiga komponen utama: *Model*, *View*, dan *ViewModel* (Microsoft, 2009).



Gambar 4 Komponen MVVM (Wikipedia, 2021)

- a. *Model* mewakili lapisan data aplikasi. Ini mencakup struktur data, logika bisnis, dan operasi basis data. *Model* bertanggung jawab untuk mengelola data yang digunakan oleh aplikasi, baik data tersebut berasal dari sumber lokal seperti *database* maupun dari sumber eksternal seperti API.
- b. *View* adalah komponen yang bertanggung jawab untuk menampilkan data kepada pengguna dan menangani interaksi pengguna. Dalam konteks aplikasi Android, *View* sering kali diimplementasikan sebagai *Activity*,

Fragment, atau *Composable* di Jetpack Compose. *View* hanya berinteraksi dengan *ViewModel* untuk mendapatkan data dan mengirimkan perintah pengguna.

- c. *ViewModel* bertindak sebagai penghubung antara *Model* dan *View*. *ViewModel* mengelola logika presentasi dan *state UI*, menyediakan data dari *Model* kepada *View* dan memperbarui *View* ketika data berubah. *ViewModel* memastikan bahwa logika bisnis tidak tercampur dengan kode antarmuka pengguna, yang membuat aplikasi lebih mudah untuk diuji dan dipelihara (Google, 2020).

Keuntungan utama dari arsitektur MVVM termasuk pemisahan yang jelas antara logika bisnis dan logika presentasi, yang memudahkan pengujian dan pemeliharaan kode. Dengan MVVM, perubahan pada UI tidak mempengaruhi logika bisnis, dan sebaliknya, perubahan pada logika bisnis tidak memerlukan perubahan pada UI. Ini meningkatkan modularitas dan memudahkan pengembangan fitur baru atau perubahan pada aplikasi (Google, 2020).

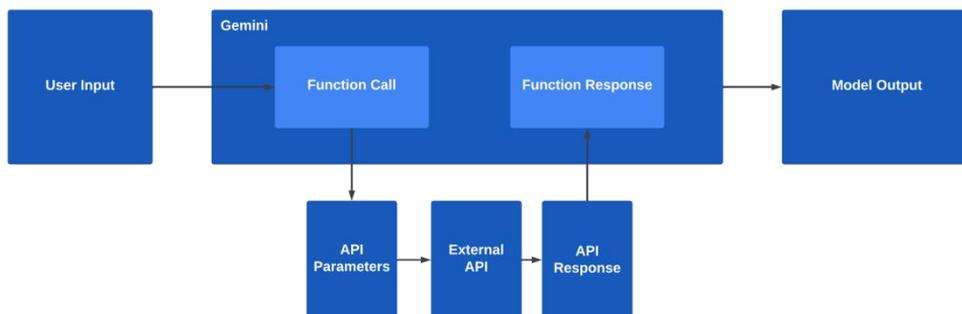
1.3.18 Black Box Testing

Black box testing adalah metode pengujian perangkat lunak yang berfokus pada pengujian fungsionalitas aplikasi tanpa memeriksa struktur internal atau kode sumbernya (Myers, 2004).

Dalam *black box testing*, penguji hanya berinteraksi dengan antarmuka aplikasi dan memverifikasi apakah output yang dihasilkan sesuai dengan input yang diberikan, berdasarkan spesifikasi dan persyaratan yang telah ditentukan. Metode ini digunakan untuk memastikan bahwa aplikasi berfungsi sesuai dengan yang diharapkan oleh pengguna akhir tanpa memperhatikan bagaimana fungsi tersebut diimplementasikan di dalam kode. *Black box testing* dapat diterapkan pada berbagai tingkat pengujian, termasuk *testing* unit, integrasi, sistem, dan penerimaan (Beizer, 1995).

1.3.19 Gemini API

Gemini API adalah antarmuka terprogram yang diekspos oleh Google AI yang memungkinkan pengembang untuk membangun aplikasi atau fitur yang didukung AI menggunakan model Gemini Google. Dengan API ini, pengembang dapat menggunakan model Gemini untuk fungsi seperti pembuatan konten, agen dialog, peringkasan, sistem klasifikasi, membangun mesin pengolah gambar, dan lain-lain (Adi, 2024).



Gambar 5 Alur kerja Gemini API (Google Codelabs, 2024)

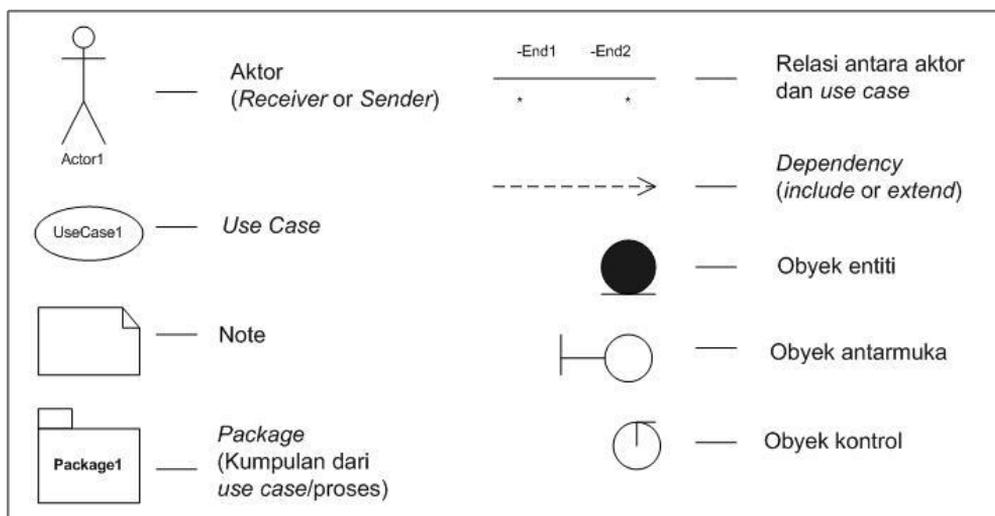
Struktur Gemini API terdiri dari dua komponen utama: sumber daya dan tipe data. Sumber daya mewakili entitas yang dapat berinteraksi dengan klien API, seperti model, operasi, dan model yang disesuaikan. Tipe data menggambarkan format standar data yang dipertukarkan antara klien dan API, memastikan cara data dikirim dan respons yang diterima sesuai dengan format yang diharapkan.

Fungsi utama yang didukung oleh Gemini API meliputi metode daftar yang memberikan daftar semua model yang tersedia, metode *get* untuk mengambil informasi model tertentu, metode *generateContent* untuk menghasilkan konten berdasarkan *prompt* yang diberikan, metode *countTokens* untuk menghitung jumlah token, dan metode *embedContent* untuk menghasilkan penyematan konten. Selain itu, ada juga metode *streamGenerateContent* yang memungkinkan pengumpulan respons dalam beberapa bagian.

Setiap sumber daya model di API ini memiliki properti seperti nama model, versi model, batas token input dan *output*, serta suhu yang menentukan derajat keacakan respon model. Dengan pemahaman yang mendalam tentang struktur dan fungsi API Gemini, pengembang dapat merancang masukan mereka dengan lebih efektif untuk mendapatkan tanggapan yang diinginkan dari model AI.

1.3.20 Use Case Diagram

Use case diagram adalah salah satu jenis diagram UML (Unified Modeling Language) yang digunakan untuk menggambarkan interaksi antara pengguna (aktor) dan sistem yang sedang dikembangkan. Diagram ini menunjukkan fungsionalitas sistem dari perspektif pengguna, termasuk berbagai kasus penggunaan (use cases) yang menggambarkan skenario di mana aktor berinteraksi dengan sistem (Dicoding, 2021). *Use case diagram* sendiri terdiri dari beberapa komponen.



Gambar 6 Komponen use case diagram (Nicki, 2013)

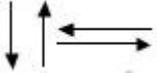
- a. Aktor adalah entitas eksternal yang berinteraksi dengan sistem, yang dapat berupa individu, kelompok, atau sistem lain. Aktor digambarkan sebagai simbol *stick figure*.
- b. *Use case* mewakili fungsi atau layanan yang disediakan oleh sistem kepada aktor. Setiap use case digambarkan dengan oval dan diberi nama yang menjelaskan fungsionalitas tersebut.
- c. Relasi antara Aktor dan *Use Case* berupa hubungan yang menunjukkan bagaimana aktor berinteraksi dengan *use case*. Digambarkan dengan garis lurus yang menghubungkan aktor dan *use case*.
- d. *Dependency* menggambarkan hubungan antara *use case*. Relasi *include* digunakan ketika sebuah use case selalu menyertakan use case lain, sedangkan *extend* digunakan ketika sebuah use case dapat memperluas fungsionalitas use case lain dalam kondisi tertentu. *Dependency* digambarkan dengan garis putus-putus dengan panah.
- e. Obyek Entiti, Antarmuka, dan Kontrol:
 - i. Obyek Entiti: Representasi objek yang menyimpan data dalam sistem, digambarkan sebagai lingkaran hitam penuh.
 - ii. Obyek Antarmuka: Antarmuka yang digunakan oleh aktor untuk berinteraksi dengan sistem, digambarkan sebagai lingkaran putih dengan garis tepi.
 - iii. Obyek Kontrol: Logika pengontrol dalam sistem, digambarkan sebagai lingkaran dengan garis tengah vertikal.
- f. Note berupa simbol berbentuk kertas yang digunakan untuk menambahkan catatan atau komentar pada diagram, memberikan informasi tambahan yang penting.

- g. *Package* digunakan untuk mengelompokkan beberapa *use case* atau proses yang terkait, digambarkan sebagai simbol folder.

1.3.21 Activity Diagram

Activity diagram pada dasarnya menggambarkan macam-macam alir aktifitas yang akan dirancang dalam sebuah sistem. Dimana masing-masing alir memiliki awal, decision yang mungkin terjadi pada sistem, dan akhir dalam sistem tersebut. Activity diagram pada dasarnya memiliki struktur yang hampir mirip dengan flowchart atau diagram alir dalam perancangan sistem secara terstruktur. Activity diagram ini dibuat berdasarkan sebuah *use case* atau beberapa *use case* dalam *use case diagram*. (Rahardjo, 2018)

Tabel 1 Komponen activity diagram

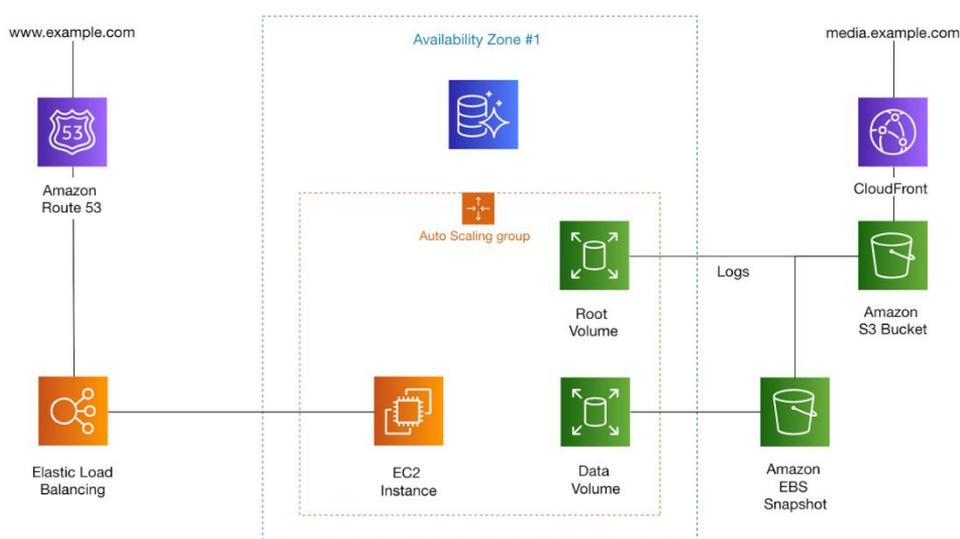
No.	Gambar	Nama	Fungsi
1		Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		Action	Merupakan <i>state</i> dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		Initial Node	Menunjukkan bagaimana objek dibentuk atau diawali.
4		Activity Final Node	Menunjukkan bagaimana objek dibentuk dan diakhiri.
5		Decision	Digunakan untuk menggambarkan suatu keputusan atau tindakan yang harus diambil pada kondisi tertentu.
6		Line Connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya.

1.3.22 Architecture Diagram

Diagram arsitektur adalah representasi visual dari arsitektur sistem yang menampilkan komponen-komponen utama sistem serta interaksi di antara komponen-komponen tersebut. Diagram ini digunakan untuk merencanakan, mendesain, dan mendokumentasikan arsitektur sistem yang kompleks, termasuk aplikasi, jaringan, dan infrastruktur TI. Dengan menggunakan diagram arsitektur, pengembang dan arsitek dapat memahami bagaimana elemen-elemen dalam sistem berhubungan dan berinteraksi, memastikan bahwa desain sistem sesuai

dengan kebutuhan bisnis dan teknis, serta memfasilitasi komunikasi yang efektif antara tim pengembang. (Amazon, 2023)

Diagram arsitektur memiliki berbagai manfaat penting dalam pengembangan sistem. Pertama, diagram ini membantu memvisualisasikan struktur dan komponen utama sistem secara keseluruhan, memungkinkan pengembang untuk melihat bagaimana bagian-bagian sistem saling berinteraksi. Kedua, diagram arsitektur digunakan untuk perencanaan dan desain, memastikan bahwa semua kebutuhan bisnis dan teknis terpenuhi sebelum implementasi. Selain itu, diagram ini berfungsi sebagai dokumentasi yang berguna untuk pemeliharaan dan peningkatan sistem di masa mendatang.



Gambar 8 Contoh *architecture diagram* (Amazon, 2023)

Terdapat beberapa jenis diagram arsitektur yang digunakan untuk menggambarkan berbagai aspek sistem. Pertama, diagram komponen menunjukkan komponen-komponen utama dari sistem dan bagaimana mereka saling berhubungan. Kedua, diagram deployment menggambarkan pengaturan fisik dari komponen perangkat keras dan perangkat lunak dalam sistem. Ketiga, diagram aliran data menampilkan aliran data di dalam sistem dan antara komponen-komponennya, membantu memahami bagaimana data diproses dan dikirim.