



PENGEMBANGAN APLIKASI KESEHATAN MENTAL BERBASIS KOTLIN: STUDI KASUS APLIKASI HUGME

Antonius Sigid Priharsanto ¹, Abdul Azzam Ajhari ²

¹ Department of Informatics, Universitas Siber Asia, Jakarta, 12550, Indonesia

² Department of Informatics, Universitas Siber Asia, Jakarta, 12550, Indonesia

<p>INFORMASI ARTIKEL</p>	<p>A B S T R A K</p>
<p>Received: July 22, 2024 Revised: August 03, 2024 Available online: Nopember 07, 2024</p>	<p>Akses terbatas terhadap layanan kesehatan mental dan kurangnya kesadaran tentang pentingnya kesehatan mental merupakan masalah yang dihadapi masyarakat saat ini. Pengembangan aplikasi mobile menjadi salah satu solusi yang menjanjikan untuk memberikan akses yang lebih mudah dan cepat terhadap layanan kesehatan mental. Penelitian ini mengembangkan aplikasi kesehatan mental berbasis Android menggunakan bahasa pemrograman Kotlin. Kotlin, sebagai bahasa pemrograman yang populer dan kuat untuk pengembangan aplikasi Android, menawarkan fitur-fitur modern yang dapat mempercepat pengembangan aplikasi dan meningkatkan kualitasnya. Aplikasi memanfaatkan teknologi <i>deep learning</i> dengan arsitektur <i>Bidirectional Long Short-Term Memory (Bi-LSTM)</i> dan <i>flatten layer</i> untuk menganalisis catatan pengguna. Dataset yang digunakan terdiri dari 2.432.229 data, dibagi menjadi 2.383.584 (98%) untuk data <i>training</i> dan 48.645 (2%) untuk data <i>validation</i>. Dari proses <i>training</i> model, diperoleh akurasi <i>training</i> mencapai 88.38% dan akurasi validasi 85.34%. Aplikasi ini dirancang untuk memberikan dukungan kesehatan mental dengan fitur-fitur utama seperti analisis catatan pengguna dengan memanfaatkan teknologi <i>deep learning</i>, rekomendasi tempat <i>healing</i> terdekat, <i>chat</i> anonim, dan riwayat catatan. Aplikasi ini diharapkan dapat meningkatkan akses layanan kesehatan mental, mengurangi stigma negatif, dan menguatkan kesadaran kesehatan mental masyarakat. Namun, masih terdapat beberapa kekurangan yaitu pada integrasi layanan <i>deep learning</i> dan API peta, yang dapat di optimalkan dalam penelitian selanjutnya.</p> <p>Kata kunci— <i>bidirectional long short-term memory</i>, <i>chat</i> anonim, <i>flatten layer</i>, kesadaran kesehatan mental, kesehatan mental</p>
<p>CORRESPONDENCE</p>	<p>A B S T R A C T</p>
<p>E-mail: ¹ underfighterkkh@gmail.com</p>	<p><i>Limited access to mental health services and lack of awareness regarding the importance of mental health are significant issues in today's society. The development of mobile applications has emerged as a promising solution to provide easier and quicker access to mental health services. This research specifically focuses on creating an Android-based mental health application using the Kotlin programming language. Kotlin, a popular and powerful language for Android app development, offers modern features that can expedite app development and enhance its quality. The application leverages deep learning technology with a Bidirectional Long Short-Term Memory (Bi-LSTM) and a flattened layer architecture to analyze user notes. A dataset of 2,432,229 data points is used, with 98% (2,383,584) for training data and 2% (48,645) for validation data. The model achieved a training accuracy of 88.38% and a validation accuracy of 85.34% during the training process. The application is designed to provide mental health support with key features such as user note analysis using deep learning, recommendations for nearby healing places, anonymous chat, and note history. It is expected that this application will improve access to mental health services, reduce negative stigma, and enhance public awareness of mental health. However, there are still some shortcomings, such as the integration of deep learning services and map APIs, which can be optimized in future research.</i></p> <p><i>Keywords— bidirectional long short-term memory, anonymous chat, flattened layer, mental health awareness, mental health</i></p>

I. PENDAHULUAN

Dalam era global yang semakin kompleks dan dinamis, kesehatan mental menjadi isu yang mendesak untuk

diperhatikan. Hasil survei terbaru dari Indonesia *National Adolescent Mental Health Survey (I-NAMHS)* pada tahun 2022, menunjukkan 5,5 persen remaja usia 10-17 tahun didiagnosis memiliki gangguan mental dalam 12 bulan terakhir. Sekitar 34,9 persen setidaknya memiliki satu

masalah kesehatan mental atau masalah kejiwaan [1]. Gangguan mental yang dapat dialami oleh seseorang antara lain adalah gangguan kecemasan, bipolar, depresi, *post-traumatic stress disorder* (PTSD), *schizophrenia*, hingga gangguan makan [2]. Kebanyakan orang dengan gangguan mental tidak memiliki akses ke perawatan yang efektif. Selain itu, banyak orang mengalami stigma, diskriminasi dan pelanggaran Hak Asasi Manusia (HAM) [3].

Pandemi global dan tekanan hidup modern telah meningkatkan kesadaran akan pentingnya kesehatan mental. Namun, masih banyak individu yang mengalami stres, kecemasan, depresi, dan berbagai masalah kesehatan mental lainnya [4]. Solusi teknologi, khususnya aplikasi *mobile*, telah menjadi sarana yang populer dan mudah diakses bagi individu untuk mendapatkan dukungan dan sumber daya kesehatan mental. Kotlin [5], sebagai bahasa pemrograman yang populer dan kuat untuk pengembangan aplikasi Android [6], [7], menawarkan fitur-fitur *modern* yang dapat mempercepat pengembangan aplikasi dan meningkatkan kualitasnya.

II. STUDI LITERATURE

Geona dan Cyntia (2023) dalam penelitiannya mengungkapkan bahwa orang dengan gangguan mental seringkali tidak memiliki teman yang peduli, atau teman namun cenderung menghakimi. Dengan masalah tersebut maka perlu dilakukan pengembangan aplikasi *mobile* berbasis Android dengan pendekatan *design thinking* sebagai ruang bagi orang dengan gangguan mental untuk bebas bercerita tanpa perlu khawatir dengan privasi. Fitur-fitur yang dikembangkan antara lain *open sharing*, *call centre*, *medical appointment*, *daily quotes*, dan *music playlist* [8].

Hanna dkk (2021) membahas pengembangan aplikasi kesehatan mental berbasis *mobile* dilengkapi fitur terapi dengan konsep psikologi Islam, yang diadaptasi dan diimplementasikan dalam bentuk terapi berbasis *audio* dan *video*. Selain fitur terapi, aplikasi ini mengimplementasikan fitur artikel dan konseling [9].

Imron (2021) dalam penelitiannya merancang aplikasi kesehatan mental dengan metode *Rapid Application Development* (RAD). Perancangan aplikasi dimaksudkan untuk mengatasi kesehatan mental secara *online* dengan menyediakan fitur konsultasi bersama psikolog [10].

Imbalo dan Suendri (2023) dalam penelitiannya mengimplementasikan *hybrid app framework* dalam membangun aplikasi kesehatan mental berbasis *mobile*. Fitur yang tersedia antara lain fitur *voyage*, *random quotes*, dan *journal* [11].

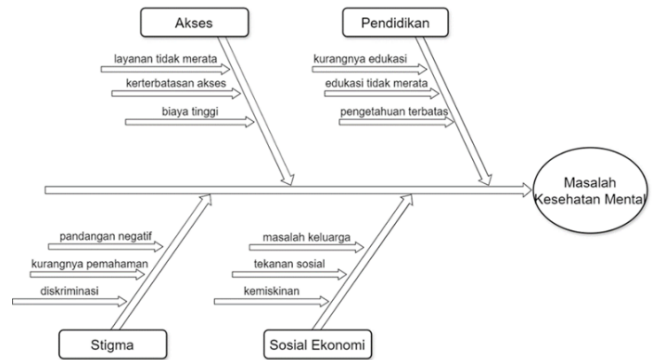
Abukhair dkk (2022) melakukan penelitian kesehatan mental dengan fokus pengembangan aplikasi CURCOL, diperuntukan bagi remaja yang mengalami gangguan kesehatan mental. Fitur utama aplikasi yaitu curhat *online*, didukung fitur *music playlist* dan fitur motivasi [12].

Berdasarkan penelitian sebelumnya, solusi yang sudah ada antara lain fitur konsultasi *online*, motivasi, artikel, *open sharing*, *call centre*, serta fitur terapi *audio* dan *video*. Sehingga, diusulkan sebuah inovasi yaitu aplikasi HugMe yang mengimplementasikan fitur baru, yaitu rekomendasi tempat hiburan terdekat untuk healing [13] berdasarkan analisis catatan dengan analisis *deep learning* [14].

III. METODOLOGI PENELITIAN

A. Fishbone Diagram

Analisis fishbone diagram (diagram tulang ikan) sering disebut juga diagram Ishikawa atau *cause-and-effect diagrams* (diagram sebab-akibat). Fishbone diagram adalah alat untuk mengidentifikasi berbagai sebab potensial dari satu efek atau masalah, dan menganalisis masalah tersebut melalui sesi *brainstorming* [15]. Berikut adalah analisis masalah kesehatan mental dengan metode fishbone diagram:



Gambar 1. Fishbone Diagram Masalah Kesehatan Mental

Gambar 1 menunjukkan diagram fishbone membagi penyebab permasalahan utama kesehatan mental ke dalam beberapa kategori:

1. Akses

Masalah utama yang dihadapi dalam akses terhadap layanan kesehatan mental di Indonesia adalah distribusi layanan yang tidak merata, akses terbatas, dan biaya yang mahal. Layanan kesehatan mental mayoritas terpusat di kota besar, sedangkan daerah lain sangat kekurangan.

Solusi yang diusulkan melalui aplikasi kesehatan mental berbasis Android mencakup fitur yang memungkinkan pengguna untuk berkonsultasi secara online. Selain itu, aplikasi ini akan memberikan rekomendasi tempat hiburan terdekat sebagai salah satu solusi untuk mengurai masalah kesehatan mental.

2. Stigma

Banyak orang memandang masalah kesehatan mental secara negatif, sehingga penderita gangguan kesehatan mental mendapat perlakuan diskriminasi. Masyarakat menganggap kesehatan mental tidak bisa disembuhkan sehingga penderitanya justru dikucilkan dari masyarakat.

Sebagai solusi, aplikasi kesehatan mental berbasis Android menawarkan fitur diskusi forum anonim dimana pengguna dapat berbagi pengalaman dan mendapatkan dukungan tanpa takut dikucilkan.

3. Pendidikan

Kurangnya edukasi dan pengetahuan tentang kesehatan mental di masyarakat menjadi masalah utama yang menghambat penanganan secara efektif. Masyarakat tidak tahu cara mengidentifikasi gejala gangguan mental atau bagaimana mencari bantuan.

Aplikasi kesehatan mental berbasis Android memberikan solusi untuk meningkatkan pendidikan dan kesadaran dengan menyediakan berbagai materi edukasi,

termasuk artikel, *video*, dan infografis yang mudah diakses.

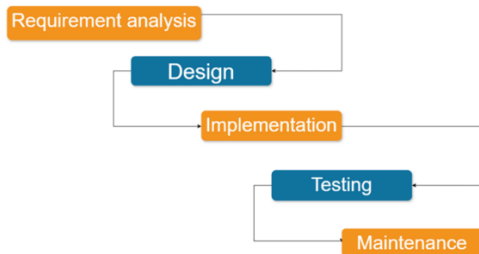
4. Sosial-ekonomi.

Kemiskinan, pengangguran, dan tekanan sosial seringkali memperburuk kesehatan mental masyarakat Indonesia. Individu yang mengalami stres tidak mendapat akses terhadap layanan kesehatan mental sehingga keadaannya semakin memburuk.

Aplikasi yang dibangun menawarkan layanan dengan biaya gratis sehingga bisa dijangkau semua lapisan masyarakat

B. Metode Waterfall

Penelitian ini menggunakan metode Waterfall, yaitu model penelitian yang terstruktur [16], [17], sistematis [18] dan berurutan [19]. Metode penelitian yang dilakukan mengadopsi dari [20] dibagi dalam 5 tahapan seperti pada Gambar 2.



Gambar 2. Metode Waterfall

1. Requirement analysis

Pada tahap analisa kebutuhan, kebutuhan pengguna dan sistem dikumpulkan dan dianalisis secara mendalam. Informasi tersebut akan digunakan untuk memahami masalah yang ingin diatasi dan fitur yang diperlukan dalam aplikasi.

2. Design (desain)

Pada tahap desain, dilakukan analisa dan memodelkan sistem yang akan dibangun sesuai dengan kebutuhan yang diminta [21].

3. Implementation

Proses penulisan code ada ditahap ini. Pembuatan software akan dipecah menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap selanjutnya. Dalam tahap ini juga akan dilakukan pemeriksaan lebih dalam terhadap modul yang sudah dibuat, apakah sudah memenuhi fungsi yang diinginkan atau belum [22].

4. Testing (pengujian)

Pengujian fokus pada perangkat lunak dari segi *logic* dan fungsional serta memastikan bahwa semua bagian sudah diuji sehingga keluaran yg dihasilkan sesuai dengan yg diinginkan [23]. Tahap pengujian menggunakan metode *blackbox testing* untuk memastikan aplikasi berjalan sesuai dengan fungsinya.

5. Maintenance (pemeliharaan)

Setelah aplikasi diluncurkan, pemeliharaan dilakukan untuk memperbaiki *bug*, memperbarui

aplikasi dengan fitur baru, dan memastikan kinerja yang optimal.

IV. HASIL DAN PEMBAHASAN

A. Requirement Analysis

Requirement analysis (analisis kebutuhan) merupakan proses mengidentifikasi, mendokumentasikan, dan mengelola kebutuhan yang terkait dengan suatu sistem, baik perangkat lunak, produk, atau layanan. Analisis kebutuhan dilakukan untuk memastikan bahwa hasil akhir dari proyek memenuhi harapan dan kebutuhan pengguna.

1. Pengumpulan Informasi

Wawancara dan diskusi dilakukan bersama pengguna potensial untuk mengidentifikasi kebutuhan dan harapan terhadap aplikasi. Analisis kompetitor dilakukan dengan mengidentifikasi kekurangan dan kelebihan dari aplikasi yang sudah ada, serta tinjauan pustaka penelitian terkait.

2. Analisis Kebutuhan Pengguna

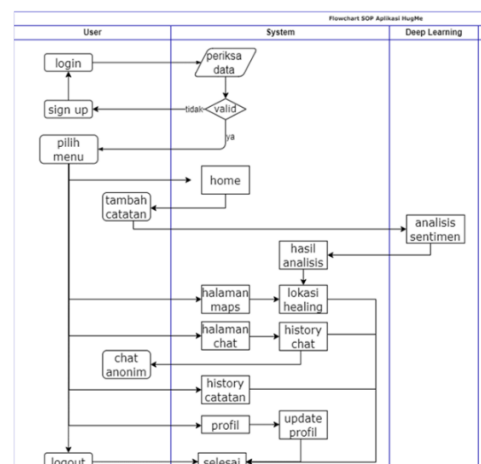
- Pengguna membutuhkan fitur untuk melacak suasana hati mereka setiap hari dengan pilihan untuk menambahkan catatan. Analisis catatan mengimplementasikan *deep learning* jenis *Recurrent Neural Network* (RNN).
- Tempat aman dan nyaman untuk curhat.
- Rekomendasi tempat hiburan untuk healing.

3. Analisis Kebutuhan Sistem

- Antarmuka pengguna harus mudah digunakan dan menarik [24].
- Data pengguna harus dilindungi dengan enkripsi dan kebijakan privasi yang ketat.
- Sistem harus dapat berintegrasi dengan *API backend* untuk penyimpanan dan pengambilan data, serta harus dirancang untuk mampu menangani peningkatan jumlah pengguna.

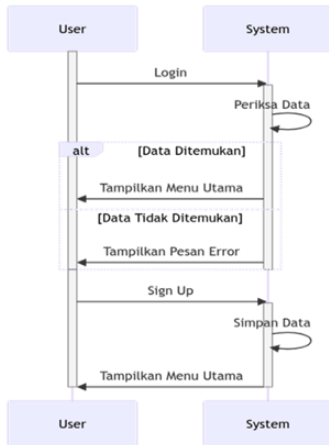
B. Design

Perancangan pengembangan aplikasi dapat dilakukan dengan menggunakan *Standard Operating Procedure* (SOP) *flowchart* dan *sequence diagram*, untuk menggambarkan fungsi utama aplikasi dan urutan interaksi yang terjadi. SOP *flowchart* perancangan aplikasi HugMe ditunjukkan dengan Gambar 3.



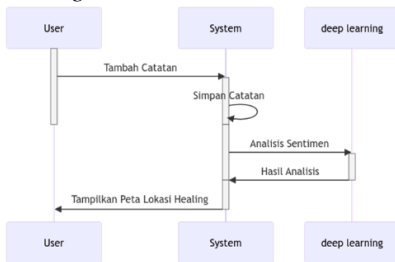
Gambar 3. SOP Flowchart Aplikasi HugMe

Sequence diagram menggambarkan alur interaksi antar komponen dalam aplikasi HugMe. Dari Gambar 4, interaksi dimulai dengan user melakukan *login* atau *sign up*. Sistem kemudian memeriksa data yang dimasukkan. Jika valid, *user* akan diarahkan ke menu utama (*home*), jika tidak maka pesan *error* ditampilkan.

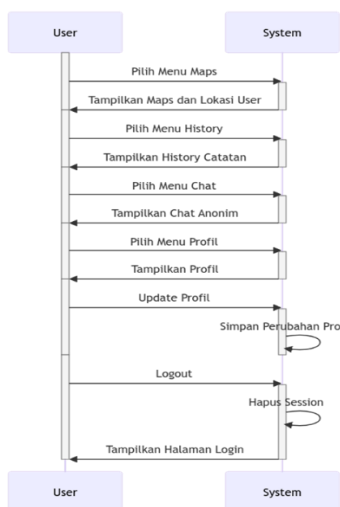


Gambar 4. Sequence Diagram Login dan Sign Up

Pada Gambar 5, *user* bisa menambahkan catatan pengalamannya. Sistem menyimpan catatan dan mengirimkannya ke *deep learning* untuk analisis sentimen. Deep learning menganalisis sentimen dan mengirimkan hasil analisis kembali ke sistem, kemudian menampilkan peta lokasi *healing*.



Gambar 5. Sequence diagram analisis catatan



Gambar 6. Sequence diagram akses menu lain

Dari Gambar 6, dapat dijelaskan bahwa *user* juga memiliki akses ke menu lain seperti *history*, *chat*, dan profil. Masing-masing menu untuk melihat catatan, berinteraksi dengan pengguna lain, dan melakukan *update* profil. Jika

user memilih *logout*, sistem akan menghapus *session user* dan menampilkan halaman *login*.

C. Implementation

1. Pengembangan deep learning

Pada tahap implementasi, dilakukan pengembangan kode aplikasi dan integrasi komponen. Pengembangan *deep learning* yang dilakukan dalam penelitian ini menggunakan dataset model *dataframe*. *Dataframe* adalah struktur data dua dimensi yang menyimpan data dalam bentuk tabel dengan baris dan kolom. *Dataframe* memungkinkan penyimpanan berbagai tipe data dalam setiap kolom, sehingga fleksibilitas data lebih tinggi. Latih model dilakukan dengan menggabungkan bidirectional LSTM [25], [26] dan *flatten layer* [27] seperti ditunjukkan Gambar 7.

```

model = keras.Sequential([
    keras.layers.Embedding(max_words, embedding_dim, input_length=max_length),
    keras.layers.Bidirectional(keras.layers.LSTM(max_length)),
    keras.layers.Flatten(),
    keras.layers.Dense(256, activation = "relu"),
    keras.layers.Dropout(0.1),
    keras.layers.Dense(256, activation = "relu"),
    keras.layers.Dropout(0.1),
    keras.layers.Dense(256, activation = "relu"),
    keras.layers.Dropout(0.1),
])

# Assuming output classes are 3 (neg, neu, pos)
keras.layers.Dense(3, activation='softmax')

optimizer = tf.keras.optimizers.Adam(learning_rate=0.001)

# Compile the model
model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])
    
```

Gambar 7. Implementasi bi-LSTM dan *flatten layer*

- Embedding* digunakan untuk mengubah input kata-kata menjadi representasi *vector* berdimensi tetap
- Bidirectional* berfungsi menangkap informasi secara dua arah *input* (*forward* dan *backward*) menggunakan LSTM.
- Flatten layer* mengubah *input* dari bentuk multi-dimensi menjadi bentuk satu dimensi supaya bisa diteruskan ke *dense layer*.
- Dense layer*, disebut juga *hidden layer*, membantu model untuk menangani non-linearitas.
- Dropout layer* mencegah *overfitting* dengan mengabaikan (*drop*) 10%-unit selama pelatihan secara acak.
- Output layer* terdapat 3 *unit* yang masing-masing mewakili kelas output negatif, positif, netral, dengan aktivasi softmax.
- Optimizer* adam menggunakan *learning rate* 0.001
- Compile model* berfungsi menyusun model dengan *optimizer* adam, *loss function* *categorical cross-entropy*, dan metrik akurasi.

Jumlah total *dataset* yang digunakan 2.432.229, dibagi menjadi 2.383.584 (98%) untuk data pelatihan dan 48.645 (2%) untuk data validasi. Proporsi data validasi diberikan 2% dengan alasan eksperimen untuk mempersingkat waktu training model.

```
total_data = len(kalimat_)
jumlah_training = len(train_kalimat)
jumlah_validation = len(val_kalimat)

persen_training = (jumlah_training / total_data) * 100
persen_validation = (jumlah_validation / total_data) * 100

print(f"Total data: {total_data}")
print(f"Jumlah data training: {jumlah_training} ({persen_training:.2f}%)")
print(f"Jumlah data validasi: {jumlah_validation} ({persen_validation:.2f}%)")
```

Total data: 2432229
 Jumlah data training: 2383584 (98.00%)
 Jumlah data validasi: 48645 (2.00%)

Gambar 8. Proporsi jumlah dataset

```
Epoch 1/10  
1084/1084 [=====] - 555s 509ms/step - loss: 0.4330 - accuracy: 0.8044 - val_loss: 0.3629 - val_accu  
acy: 0.8432  
Epoch 2/10  
1084/1084 [=====] - 554s 511ms/step - loss: 0.3622 - accuracy: 0.8517 - val_loss: 0.3512 - val_accu  
acy: 0.8488  
Epoch 3/10  
1084/1084 [=====] - 555s 512ms/step - loss: 0.3168 - accuracy: 0.8645 - val_loss: 0.3458 - val_accu  
acy: 0.8524  
Epoch 4/10  
1084/1084 [=====] - 554s 511ms/step - loss: 0.2999 - accuracy: 0.8720 - val_loss: 0.3468 - val_accu  
acy: 0.8532  
Epoch 5/10  
1084/1084 [=====] - 554s 511ms/step - loss: 0.2861 - accuracy: 0.8783 - val_loss: 0.3495 - val_accu  
acy: 0.8534  
Epoch 6/10  
1084/1084 [=====] - 554s 511ms/step - loss: 0.2737 - accuracy: 0.8838 - val_loss: 0.3655 - val_accu  
acy: 0.8534  
Epoch 7/10  
58/1084 [>.....] - ETA: 8:38 - loss: 0.2498 - accuracy: 0.8964
```

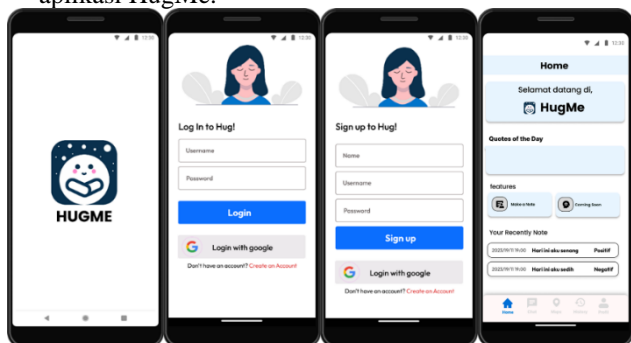
Gambar 9. Hasil training model dengan bi-LSTM dan flatten layer

Berdasarkan Gambar 9, proses pelatihan berlangsung selama 10 *epoch*. Model mencapai *loss* sebesar 0.4330 dengan akurasi 80.44% pada *epoch* 1. Akurasi pada data pelatihan meningkat dari 80.44% di *epoch* 1 hingga mencapai 88.38% di *epoch* 6. Pada data validasi, akurasi menunjukkan peningkatan dari 84.32% di *epoch* 1 menjadi 85.38% di *epoch* 5. Pada *epoch* 6, akurasi tetap stabil di 85.34%.

Pada *epoch* 7, pelatihan belum selesai, tetapi pada langkah ke-58 dari 1084, model menunjukkan performa awal yang menjanjikan dengan *loss* 0.2498 dan akurasi 89.64%. Secara keseluruhan, model mencapai hasil yang baik dengan akurasi pelatihan akhir sebesar 88.38% dan akurasi validasi sebesar 85.34%. Namun, dengan proporsi data *validation* hanya 2%, maka model ini memiliki kekurangan rawan terjadinya *overfitting*.

2. Pengembangan antarmuka pengguna

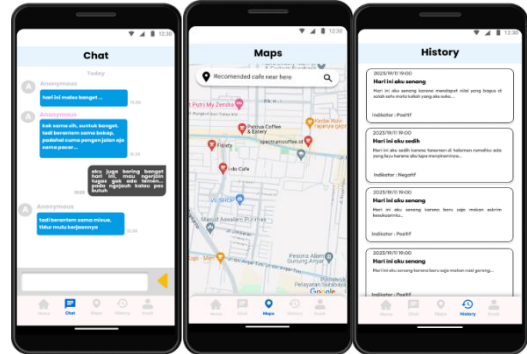
Pengembangan antarmuka pengguna dilakukan dengan mengimplementasikan *layout* xml. Berikut adalah tampilan fitur utama dalam pengembangan aplikasi HugMe.



Gambar 10. Halaman splash screen, login dan sign up, home

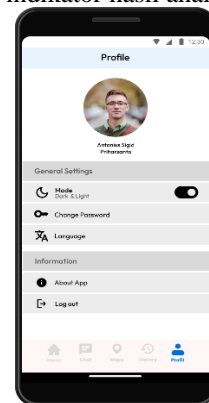
Halaman *splash screen* yang ditampilkan pada Gambar 10, digunakan untuk menyambut pengguna saat pertama kali menjalankan aplikasi. Pada halaman *login* dan *sign up*, fokus utama adalah memberikan pengalaman yang mudah bagi pengguna untuk masuk atau mendaftar ke dalam aplikasi.

Desain halaman home memiliki prioritas tampilan yang bersih dan teratur, dengan penekanan pada konten yang relevan seperti informasi penting, notifikasi, atau pilihan navigasi. Pengguna harus dengan mudah menemukan apa yang mereka cari dan diarahkan ke fitur-fitur utama aplikasi.



Gambar 11. Halaman chat anonym, maps, history

Keamanan dan privasi pengguna menjadi prioritas utama. Desain fitur *chat* seperti pada Gambar 11 memastikan identitas pengguna terlindungi, namun tetap menyediakan pengalaman komunikasi yang lancar dan mudah. Fokus utama halaman *maps* adalah menyajikan data geografis dengan cara yang mudah dimengerti dan digunakan. Mengutamakan keterbacaan peta, navigasi, serta fitur-fitur interaktif seperti zoom dan pencarian lokasi. Halaman *history* bertujuan untuk memberikan akses cepat dan mudah ke riwayat catatan sebelumnya, dengan menampilkan seluruh informasi catatan beserta indikator hasil analisisnya.



Gambar 12. Halaman Profil

Pengguna dapat mengelola informasi pribadi dan menyesuaikan pengaturan aplikasi sesuai kebutuhan pada halaman profil seperti pada Gambar 12. Pengguna dapat mengatur tampilan mode tema, mengganti *password* dan foto, serta melihat halaman credit aplikasi.

D. Testing

Pengujian aplikasi menggunakan metode *blackbox testing* dilakukan dengan pengujian fungsionalitas, untuk memastikan semua fitur bekerja sesuai dengan spesifikasi dan kebutuhan pengguna. Dibutuhkan perangkat Android dengan Software Development Kit (SDK) minimum 21 (Lollipop, Android 5.0). Pengujian aplikasi dilakukan menggunakan perangkat Infinix Mobility Limited X688B

(Android 11, API level 30). Hasil pengujian menggunakan metode blackbox testing dapat dilihat pada Tabel 1.

Tabel 1. Hasil blackbox testing

Skenario	Hasil Diharapkan	Hasil Pengujian	Status
user jalankan aplikasi dengan klik icon aplikasi	animasi <i>splash screen</i> sebelum <i>login/sign up</i>	animasi <i>splash screen</i> sebelum <i>login/sign up</i>	sukses
1. user klik menu <i>sign up</i>	1. tampilkan form pendaftaran	1. tampilkan form pendaftaran	sukses
2. memasukkan <i>email</i> atau <i>password</i> tidak sesuai	2. pesan <i>error</i> <i>email/password</i> tidak sesuai	2. pesan <i>error</i> <i>email/password</i> tidak sesuai	sukses
3. kosongkan kolom <i>email</i> atau <i>password</i>	3. pesan <i>error</i> <i>email/password</i> tidak boleh kosong	3. pesan <i>error</i> <i>email/password</i> tidak boleh kosong	sukses
4. mengisi pendaftaran sesuai petunjuk	4. pesan pendaftaran berhasil	4. pesan pendaftaran berhasil	sukses
1. user klik menu <i>login</i>	1. menampilkan form login	1. menampilkan form login	sukses
2. memasukkan <i>email</i> atau <i>password</i> tidak sesuai	2. pesan <i>error</i> <i>email/password</i> tidak sesuai	2. pesan <i>error</i> <i>email/password</i> tidak sesuai	sukses
3. kosongkan kolom <i>email</i> atau <i>password</i>	3. pesan <i>error</i> <i>email/password</i> tidak boleh kosong	3. pesan <i>error</i> <i>email/password</i> tidak boleh kosong	sukses
4. mengisi form <i>login</i> sesuai petunjuk	4. pesan login berhasil dan diarahkan ke halaman <i>home</i>	4. pesan login berhasil dan diarahkan ke halaman <i>home</i>	sukses
5. login dengan akun Google	5. login dengan akun Google dan diarahkan ke halaman <i>home</i>	5. <i>login</i> gagal dan tetap berada di halaman <i>login</i>	gagal
klik catatan di halaman <i>home</i>	1. menampilkan form tambah catatan 2. menampilkan hasil analisis	1. menampilkan form tambah catatan 2. menampilkan hasil analisis	sukses
	3. menampilkan maps sebagai rekomendasi lokasi <i>healing</i>	3. menampilkan maps sebagai rekomendasi lokasi <i>healing</i>	gagal
klik menu chat	tampil halaman chat anonim	tampil halaman kosong	gagal
klik menu maps	tampil halaman maps dengan titik lokasi user	tampil halaman maps dengan titik lokasi user	gagal
klik menu history	tampil riwayat catatan user	tampil halaman kosong	gagal
klik menu profil	menampilkan halaman profil	menampilkan halaman profil	sukses
klik menu logout dari halaman profil	hapus session user, tampilkan halaman login	hapus session user, tampilkan halaman login	sukses

Dari Tabel 1, terdapat beberapa skenario pengujian yang gagal. Adapun penjelasan dari hasil pengujian tersebut adalah sebagai berikut:

- 1 Implementasi autentikasi Google belum berhasil dilakukan karena keterbatasan waktu yang diberikan untuk penelitian

- 2 Deep learning belum terintegrasi dengan kode Kotlin karena keterbatasan waktu yang diberikan, sehingga hasil analisis catatan tidak bisa ditampilkan.
- 3 Halaman maps belum dapat menampilkan peta karena belum terintegrasi dengan API peta
- 4 Rekomendasi tempat *healing* belum dapat ditampilkan karena catatan belum bisa dianalisis oleh *deep learning* dan API peta belum terintegrasi.
- 5 Halaman history menampilkan halaman kosong karena belum bisa menyimpan catatan dan hasil analisisnya.
- 6 Fitur yang gagal pengujian dengan skenario, belum dilakukan perbaikan karena keterbatasan waktu yang diberikan untuk penelitian dan service berlangganan Google Cloud yang sudah berakhir.

E. Maintenance

Pada tahap pemeliharaan, pengembang aplikasi bertanggung jawab untuk memastikan aplikasi tetap berfungsi dengan baik, aman, dan relevan bagi pengguna. Berikut adalah aktivitas utama yang dilakukan pada tahap pemeliharaan:

- 1 Pemantauan dan analisis *log* aplikasi untuk mengidentifikasi dan memperbaiki masalah kinerja.
- 2 Mengumpulkan laporan *bug* dari pengguna dan melakukan *resolving* dengan prioritas berdasarkan dampak dan tingkat keparahan.
- 3 Pembaharuan terhadap *library* yang digunakan pada aplikasi.
- 4 Pembaharuan dataset untuk meningkatkan relevansi data pengguna.

V. PENUTUP

A. Kesimpulan

Aplikasi HugMe menyediakan fitur-fitur seperti analisis catatan pengguna dengan *deep learning*, rekomendasi tempat *healing*, *chat* anonim, dan riwayat catatan. Implementasi model bidirectional LSTM dan *flatten layer* untuk analisis sentimen menunjukkan hasil akurasi training mencapai 88.38% dan akurasi validasi 85.34%, menunjukkan model memiliki kemampuan yang baik dalam menganalisis sentimen catatan pengguna. Meskipun model menunjukkan akurasi yang baik, jumlah data yang diuji masih relatif sedikit. Hal ini dapat mempengaruhi keakuratan model secara keseluruhan karena model mungkin belum menemukan pola yang beragam dan mungkin kurang mampu menangani variasi dalam data yang lebih besar atau data yang berbeda dari yang diuji. Pengujian aplikasi menggunakan metode *blackbox testing* sukses pada skenario fungsionalitas utama, namun terdapat beberapa fitur yang memerlukan perbaikan dan pengembangan lebih lanjut.

B. Saran

Berdasarkan penelitian yang dilakukan, terdapat beberapa hal yang memerlukan perbaikan dan pengembangan sebagai berikut:

- 1 Penambahan porsi *data validation* dan *data testing* dalam pelatihan model *deep learning*.

- 2 Perbaikan pada fitur tambah catatan dan analisis, perlu mengintegrasikan model *deep learning* ke dalam kode Kotlin.
- 3 Perbaikan pengaturan API pada fitur chat anonim yang masih menampilkan halaman kosong.
- 4 Perbaikan pada fitur maps, perlu mengintegrasikan API untuk membaca peta, misalnya dengan Google Maps API.
- 5 Perbaikan pada fitur *history*, riwayat catatan yang berhasil di analisis oleh *deep learning* disimpan dan ditampilkan di menu *history*.
- 6 Menambahkan autentikasi Google untuk *login* dan *sign up*, serta optimasi performa *backend* untuk menangani peningkatan jumlah user.

REFERENSI

- [1] Arif, "Krisis Kesehatan Mental Menghantui Generasi Z Indonesia," <https://www.kompas.id/baca/humaniora/2023/07/09/krisis-kesehatan-mental-menghantui-generasi-z-indonesia>.
- [2] WHO, "Mental disorders," <https://www.who.int/news-room/fact-sheets/detail/mental-disorders>.
- [3] WHO, "Mental health," <https://www.who.int/news-room/fact-sheets/detail/mental-health-strengthening-our-response>.
- [4] Kementerian Kesehatan, "Pengertian Kesehatan Mental," <https://ayosehat.kemkes.go.id/pengertian-kesehatan-mental>.
- [5] Google for Developers, "Ringkasan Kotlin. Android Developers," <https://developer.android.com/kotlin/overview?hl=id>.
- [6] Putra, "PENGERTIAN ANDROID: Sejarah, Kelebihan & Versi Sistem Operasi," <https://salamadian.com/pengertian-android/>.
- [7] D. A. Verdianto, "Android - Pengertian, Fungsi, Kelebihan dan Kekurangannya," <https://teknogram.id/kamus/android/>.
- [8] G. Theophillia and C. Chandra, "Perancangan Aplikasi Kesehatan Mental berbasis Mobile Menggunakan Metode Design Thinking," 2023.
- [9] H. Oktasya Ross *et al.*, "Perancangan Aplikasi Kesehatan Mental 'NAFS' (Islamic Psycho Spiritual Therapy) Berbasis Android Menggunakan Metode PDCA (Plan-Do-Check-Action)," *Jurnal Informatika Universitas Pamulang*, vol. 6, no. 4, pp. 849–856, 2021, doi: 10.32493/informatika.v6i4.14421.
- [10] I. Khoiri, "Perancangan Aplikasi Kesehatan Mental Online Menggunakan Metode Rapid Application Development(Rad)," 2021.
- [11] M. I. Z. Hasibuan and S. Suendri, "Implementasi Hybrid App Framework dalam Membangun Aplikasi Kesehatan Mental berbasis Mobile," *Edumatic: Jurnal Pendidikan Informatika*, vol. 7, no. 1, pp. 141–150, Jun. 2023, doi: 10.29408/edumatic.v7i1.17478.
- [12] A. Abukhair, N. Herawati, S. Solihat, Y. Nur, and A. Pratiwi, "Perancangan aplikasi curhat online untuk membantu dalam menyelesaikan gangguan kesehatan mental remaja," 2023, doi: 10.17509/v4i1.xx.
- [13] A. Arsandi, "Pengertian dan Cara Self-Healing yang Tepat Menurut Psikolog," <https://www.klikdokter.com/psikologi/kesehatan-mental/pengertian-dan-cara-self-healing-yang-tepat-menurut-psikolog>.
- [14] F. Amalya, D. D. Laras, and A. Hidayat, "APLIKASI JURNALING PENDETEKSI KEBAHAGIAAN MENGGUNAKAN NATURAL LANGUAGE PROCESSING BERBASIS ANDROID," 2023. Accessed: Jul. 11, 2024. [Online]. Available: <https://ejournal.gunadarma.ac.id/index.php/ugjournal/article/viewFile/8719/2851>
- [15] Y. A. Sujarwo and A. Ratnasari, "Aplikasi Reservasi Parkir Inap Menggunakan Metode Fishbone Diagram dan QR-Code," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 9, no. 3, pp. 302–309, Aug. 2020, doi: 10.32736/sisfokom.v9i3.808.
- [16] M. Mailasari, "SISTEM INFORMASI PERPUSTAKAAN MENGGUNAKAN METODE WATERFALL," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 8, no. 2, pp. 207–214, Aug. 2019, doi: 10.32736/sisfokom.v8i2.657.
- [17] G. Wirosasmito, "Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal," *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, vol. 2, no. 1, pp. 6–12, 2017, [Online]. Available: <http://www.tegalkab.go.id>,
- [18] E. Listiyan and E. R. Subhiyakti, "Rancang Bangun Sistem Inventory Gudang Menggunakan Metode Waterfall (Studi Kasus Di CV. Aqualux Duspha Abadi Kudus Jawa Tengah)."
- [19] M. Susilo and R. Kurniati, "RANCANG BANGUN WEBSITE TOKO ONLINE MENGGUNAKAN METODE WATERFALL," 2018. doi: <https://doi.org/10.30743/infotekjar.v2i2.171>.
- [20] N. Supiana, "PENGEMBANGAN APLIKASI GEOLOCATION UNTUK MONITORING LOKASI MAHASISWA SELAMA PANDEMI BERBASIS ANDROID MENGGUNAKAN METODE WATERFALL (STUDI KASUS: STMIK INSAN PEMBANGUNAN)," *Jurnal Khatulistiwa Informatika*, vol. 10, no. 1, pp. 74–80, 2022, [Online]. Available: <https://stmik.ipem.ac.id/>
- [21] A. Reservasi, L. Futsal, B. Web, T. Ardiansah, and D. Hidayatullah, "Penerapan Metode Waterfall Pada," *Journal of Information Technology, Software Engineering, and Computer Science (ITSECS)*, vol. 1, no. 1, 2023.
- [22] V. Olindo and A. Syaripudin, "Perancangan Sistem Informasi Absensi Pegawai Berbasis Web Dengan Metode Waterfall (Studi Kasus : Kantor Dbpr Tangerang Selatan)," *OKTAL : Jurnal Ilmu Komputer dan Science*, vol. 1, no. 01, 2022.
- [23] M. Badrul, "PENERAPAN METODE WATERFALL UNTUK PERANCANGAN SISTEM INFORMASI INVENTORY PADA TOKO KERAMIK BINTANG TERANG," vol. 8, no. 2, 2021.
- [24] L. M. A. A. Gandhi, R. S. Sianturi, and A. P. Kharisma, "Perancangan User Experience Aplikasi Mobile Online Therapy untuk Mengatasi Depresi bagi Remaja menggunakan Metode Human-Centered Design," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, pp. 3161–3171, 2022.
- [25] Y. Karyadi and H. Santoso, "Prediksi Kualitas Udara Dengan Metoda LSTM, Bidirectional LSTM, dan GRU," *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, vol. 9, no. 1, pp. 671–684, 2022, Accessed: May 26, 2024. [Online]. Available: <https://doi.org/10.35957/jatisi.v9i1.1588>
- [26] Anishnama, "Understanding Bidirectional LSTM for Sequential Data Processing," <https://medium.com/@anishnama20/understanding-bidirectional-lstm-for-sequential-data-processing-b83d6283befc>.
- [27] J.P. Srivatsavaya, "Flatten Layer — Implementation, Advantage and Disadvantages," <https://medium.com/@prudhviraju.srivatsavaya/flatten-layer-implementation-advantage-and-disadvantages-0f8c4ecf5ac5>.