

LAMPIRAN

Lampiran 1. Surat Kesepakatan Bimbingan Skripsi

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Irfan Triadi Saputra
NIM : 20090017
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Slamet Wiyono, S.Pd., M.Eng.
Status : Dosen
NIDN : 0626059001
Jabatan Fungsional : Lektor
Pangkat/Golongan : Penata Tk.I

Pada hari ini 14 Maret 2024 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan syarat Pihak Pertama wajib melakukan bimbingan Skripsi minimal 2 minggu sekali kepada Pihak Kedua. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.
Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

Tegal, 14 Maret 2024

Pihak Pertama



Irfan Triadi Saputra

Pihak Kedua



Slamet Wiyono, S.Pd., M.Eng.

Mengetahui
Ketua Program Studi Sarjana Terapan Teknik Informatika



Dyah Aprianti, S.P., M.Kom.
NIPY. 09.015.225

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Irfan Triadi Saputra
NIM : 20090017
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Dwi Intan Af'idah, M.Kom.
Status : Dosen
NIDN : 0620089303
Jabatan Fungsional : Asisten Ahli
Pangkat/Golongan : Penata Muda Tk.I/III

Pada hari ini 13 Maret 2024 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Skripsi Pihak Pertama dengan syarat Pihak Pertama wajib adanya kemajuan pengerjaan skripsi yang dipresentasikan setiap dua minggu kepada Pihak Kedua. Adapun waktu dan tempat pelaksanaan disepakati antar pihak. Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

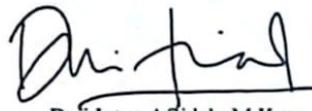
Tegal, 14 Maret 2024

Pihak Pertama



Irfan Triadi Saputra

Pihak Kedua



Dwi Intan Af'idah, M.Kom.

Mengetahui
Ketua Program Studi Sarjana Terapan Teknik Informatika



Dyah Apriliani, S.T., M.Kom.
NIPY. 09.015.225

Lampiran 2. Surat Pernyataan Pengajuan HKI

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

1. Nama : Irfan Triadi Saputra
Kewarganegaraan : Indonesia
Alamat : Dusun 06 – Bandelan, RT/RW 006/007, Desa Taman, Kecamatan Taman, Kabupaten Pemalang, Provinsi Jawa Tengah
2. Nama : Slamet Wiyono, S.Pd., M.Eng
Kewarganegaraan : Indonesia
Alamat : Cibelok 02, RT/RW 004/002, Desa Cibelok, Kecamatan Taman, Kabupaten Pemalang, Provinsi Jawa Tengah
3. Nama : Dwi Intan Af'idah, M.Kom
Kewarganegaraan : Indonesia
Alamat : Desa Grinting RT/RW 003/001, Kecamatan Bulakamba, Kabupaten Brebes, Propinsi Jawa Tengah

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:
Berupa : Program Komputer
Berjudul : BaikMu: Self Care Virtual Assistant Kesehatan Mental
 - Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);
 - Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
 - Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
 - Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
 - Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
 - Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.
2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh ciptaan yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.
3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan.
4. Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami melanggar, maka saya / kami bersedia secara sukarela bahwa:

- a. permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
- b. Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia R.I dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.
- c. Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam berperkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

Demikian Surat pernyataan ini saya/kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Tegal, 28 Mei 2024



(Irfan Triadi Saputra)

(Slamet Wiyono, S.Pd., M.Eng)

(Dwi Intan Af'idah, M.Kom)

Lampiran 3. Surat Pengalihan HKI

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

1. Nama : Irfan Triadi Saputra
Kewarganegaraan : Indonesia
Alamat : Dusun 06 – Bandelan, RT/RW 006/007, Desa Taman, Kecamatan Taman, Kabupaten Pemalang, Provinsi Jawa Tengah
2. Nama : Slamet Wiyono, S.Pd., M.Eng
Kewarganegaraan : Indonesia
Alamat : Cibelok 02, RT/RW 004/002, Desa Cibelok, Kecamatan Taman, Kabupaten Pemalang, Provinsi Jawa Tengah
3. Nama : Dwi Intan Af'idah, M.Kom
Kewarganegaraan : Indonesia
Alamat : Desa Grinting RT/RW 003/001, Kecamatan Bulakamba, Kabupaten Brebes, Propinsi Jawa Tengah

Adalah **Pihak I** selaku pencipta, dengan ini menyerahkan karya ciptaan saya kepada :

Nama : Pusat Penelitian dan Pengabdian Masyarakat (P3M)
Alamat : Jl. Mataram No. 9 Pesurungan Lor Kota Tegal

Adalah **Pihak II** selaku Pemegang Hak Cipta berupa Program Komputer dengan judul "BaikMu: Self Care Virtual Assistant Kesehatan Mental", untuk didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

Demikianlah surat pengalihan hak ini kami buat, agar dapat dipergunakan sebagaimana mestinya.

Tegal, 28 Mei 2024

Pencipta

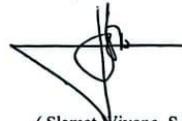
Pemegang Hak Cipta
Ketua P3M



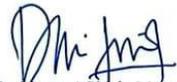
(Dr. Aldi Budi Riyanta, S.Si., M.T)



(Irfan Triadi Saputra)



(Slamet Wiyono, S.Pd., M.Eng)



(Dwi Intan Af'idah, M.Kom)



Daftar Isi

Daftar Isi.....	1
Pendahuluan.....	2
Spesifikasi Teknis.....	6
Source Code.....	7
1. Import Library.....	7
2. Kofigurasi Database.....	8
3. Route Register.....	9
4. Route Login.....	11
5. Route Edit Pengguna.....	13
6. Route Edit Password.....	14
7. Route Menampilkan Artikel.....	15
8. Route Menambahkan Artikel.....	16
9. Route Menampilkan Landing Page.....	18
10. Api Service Flutter to Flask-API.....	18

Pendahuluan

Selamat datang di BaikMu Self Care, sebuah aplikasi virtual assistant kesehatan mental yang dirancang khusus untuk membantu Anda menjaga dan meningkatkan kesehatan mental Anda dengan lebih baik. Dalam era modern yang penuh dengan tantangan dan tekanan, kesehatan mental menjadi aspek yang sangat penting untuk diperhatikan. BaikMu Self Care hadir untuk memberikan dukungan yang Anda butuhkan melalui berbagai fitur yang mendukung self-care serta layanan konsultasi kesehatan mental secara virtual.

Latar Belakang dari aplikasi ini adalah Kesehatan mental merupakan kondisi dimana individu memiliki kesejahteraan yang tampak dari dirinya yang mampu menyadari potensinya sendiri, memiliki kemampuan untuk mengatasi tekanan hidup normal pada berbagai situasi dalam kehidupan, mampu bekerja secara produktif dan menghasilkan, serta mampu memberikan kontribusi kepada komunitasnya. Mengutip dari jargon yang digunakan oleh WHO, “*there is no health without mental health*” menandakan bahwa kesehatan mental perlu dipandang sebagai sesuatu yang penting sama seperti kesehatan fisik. Kesehatan mental sekarang ini sangat diperlukan sebagai upaya mengurangi angka bunuh diri yang tinggi pada remaja. Kesehatan mental dapat terganggu karena adanya masalah psikiatri, salah satunya adalah stres. Lebih dari 23 juta orang di Amerika Serikat mengalaminya setiap tahun.

Di Indonesia, menurut *National Adolescent Mental Health Survey (I-NAMHS)*, survei kesehatan mental nasional pertama yang mengukur angka kejadian gangguan mental pada remaja 10 – 17 tahun di Indonesia, menunjukkan bahwa satu dari tiga remaja Indonesia memiliki masalah kesehatan mental sementara satu dari dua puluh remaja Indonesia memiliki gangguan mental dalam 12 bulan terakhir. Angka ini setara dengan 15,5 juta dan 2,45 juta remaja. Remaja dalam kelompok ini adalah remaja yang terdiagnosis dengan gangguan mental sesuai dengan panduan *Diagnostic and Statistical Manual of Mental Disorders Edisi Kelima (DSM-5)* yang menjadi panduan penegakan diagnosis gangguan mental di Indonesia.

Perkembangan teknologi yang dinamis memberikan pengaruh signifikan pada kesehatan mental generasi Z. Perkembangan tersebutlah yang akhirnya mengubah pandangan dan pengalaman hidup generasi Z. Generasi ini memiliki kekhawatiran

berlebih di berbagai faktor, seperti perubahan sosial, ekonomi, dan lingkungan. Hal inilah yang kemudian menimbulkan permasalahan kesehatan mental. Selain hal-hal yang disebutkan di atas, generasi Z juga menghadapi tekanan untuk sukses dalam hidup karena mereka tumbuh di zaman di mana pendidikan dan karier sangat penting. Tekanan ini dapat menyebabkan kecemasan dan stres. Generasi Z mungkin merasa terdorong untuk memenuhi harapan di lingkungan di mana kemajuan akademik dan profesional dianggap penting. Ketidakpastian dan tekanan yang berlebihan dapat disebabkan oleh pilihan pendidikan, ekspektasi sosial, dan persaingan di tempat kerja. Akibatnya, generasi berikutnya menghadapi tantangan yang signifikan untuk mencapai tingkat kesuksesan yang diharapkan. Generasi Z dapat mengalami dampak negatif dari tekanan kesuksesan ini.

Salah satu ciri khas dari generasi ini adalah banyaknya penggunaan teknologi, cenderung lebih kreatif, dan lebih peduli terhadap lingkungan. Mayoritas populasi generasi Z saat ini masih menempuh pendidikan, baik di sekolah menengah maupun perguruan tinggi. Sangat perlu adanya bimbingan orang tua maupun psikolog untuk mendukung kesehatan mental mereka para generasi Z tetap terjaga.

Beberapa penelitian sebelumnya juga pernah dilakukan dengan mengidentifikasi masalah kesehatan mental pada remaja dan generasi Z. Sebagian besar penelitian sebelumnya cenderung berfokus pada analisis masalah dan faktor risiko yang terlibat dalam gangguan mental remaja. Oleh karena itu, penelitian ini bergerak satu langkah lebih maju dengan menghadirkan solusi yang konkret dalam bentuk *mobile* aplikasi.

Dalam menghadapi masalah kesehatan mental pada remaja dan generasi Z, dibutuhkan pendekatan inovatif. Oleh karena itu, pengembangan aplikasi virtual asisten kesehatan mental berbasis *mobile* menjadi langkah yang relevan dan progresif mengingat generasi Z banyak menggunakan perangkat teknologi. Aplikasi ini dihadirkan dengan tujuan memberikan dukungan yang lebih personal dan responsif dalam memahami serta mengatasi tantangan kesehatan mental yang dihadapi oleh pengguna. Melalui penerapan teknologi *Natural Language Processing* (NLP), aplikasi ini dapat memberikan pengalaman interaktif seperti berbicara dengan seorang konselor secara langsung, menciptakan lingkungan yang lebih intim dan membantu mengatasi stigma terkait kesehatan mental.

Penggunaan NLP dalam aplikasi ini memiliki keunggulan dalam memahami dan merespon teks manusia dengan cara yang mirip dengan komunikasi antarmanusia. NLP memungkinkan aplikasi untuk menganalisis dan merespon percakapan pengguna dengan lebih baik, memahami nuansa emosional, serta memberikan saran atau informasi yang relevan secara kontekstual. Keputusan untuk menggunakan NLP dalam aplikasi ini bertujuan untuk menciptakan pengalaman pengguna yang lebih alami dan memperkuat daya dukungnya terhadap kesehatan mental. Dengan demikian, melalui penggabungan aplikasi virtual asisten dan NLP, diharapkan dapat memberikan solusi yang lebih efektif dan terjangkau dalam mendukung kesehatan mental generasi Z.

Pengujian sistem dilakukan untuk memastikan apakah sistem sudah bekerja dengan semestinya, pengujian ini dilakukan dengan cara menguji keakuratan antara pertanyaan yang diberikan oleh user dengan jawaban yang akan ditampilkan oleh sistem pada virtual assistant serta melakukan pengujian untuk keakuratan prediksi yang ditampilkan pada fitur prediksi kesehatan mental. Pengujian machine learning dilakukan dengan menguji keakuratan yang diterapkan dalam fitur prediksi kesehatan mental.

Pertanyaan	Hasil Sistem	Tingkat Kesesuaian
Pertanyaan pengguna mengalami trauma.	Depresi Berat	Sesuai
Pertanyaan pengguna mengalami sedikit masalah pada dirinya.	Depresi Ringan	Sesuai
Pertanyaan pengguna tidak mengalami masalah pada dirinya	Normal	Sesuai

Pengujian chatbot dilakukan dengan menguji keakuratan respon antara pertanyaan dan jawaban:

Pertanyaan	Hasil Sistem	Tingkat Kesesuaian
Pertanyaan seputar sapaan	Menampilkan sapaan	Sesuai
Pertanyaan seputar kesehatan mental	Menampilkan jawaban seputar kesehatan mental	Sesuai

Pengujian ini juga dilakukan pada keseluruhan fitur yang ada untuk memastikan semua fungsi berjalan dengan baik, pengujian aplikasi ini dilakukan menggunakan dua metode, yang pertama Blackbox Testing, pengujian ini berfokus pada sisi fungsionalitas dari aplikasi untuk menyesuaikan antara input dan output pada aplikasi ini. Kemudian Usability Testing, pengujian ini digunakan untuk pengujian pada User Experience dari aplikasi dengan cara melakukan pengujian terhadap suatu kelompok pengguna.

Evaluasi sistem dilakukan setelah melakukan pengujian. Evaluasi ini dilakukan apabila terdapat ketidakakuratan pada sistem yang berjalan pada saat tahap pengujian.

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Meningkatkan kesadaran tentang kesehatan mental di kalangan generasi Z melalui informasi yang mudah diakses.
2. Memberikan kemudahan kepada pengguna untuk mengakses beragam konten edukatif yang interaktif, yang bertujuan untuk meningkatkan pemahaman mereka tentang gangguan mental dan membantu mereka mengembangkan strategi penanganan yang efektif dan sehat.
3. Layanan ini memungkinkan pengguna untuk dengan mudah mengidentifikasi psikiater terdekat, memastikan ketersediaan akses yang cepat dan efisien ke bantuan profesional di lingkungan sekitar mereka.

Spesifikasi Teknis

Spesifikasi teknis meliputi:

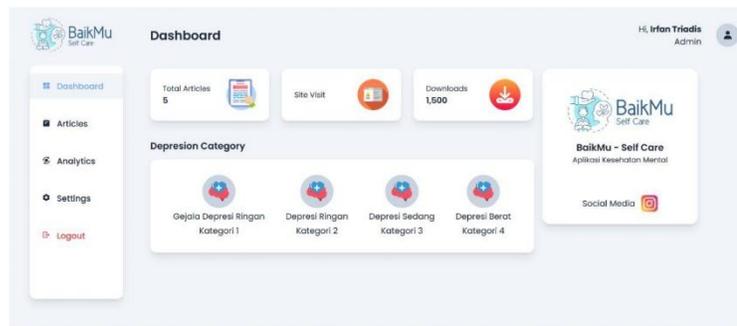
1. Modul Pengguna
2. Source Code

Berikut uraian spesifikasi untuk pembangunan aplikasi:

1. Windows 11 Ram 8gb
2. Visual Studio Code
3. Web Browser
4. Smartphone

Berikut uraian spesifikasi modul:

1. Modul untuk developer



Bagian ini hanya dapat diakses oleh developer, bertujuan untuk memudahkan melakukan perbaharuan isi konten dari aplikasi berupa artikel yang ada didalamnya.

2. Modul untuk pengguna



Bagian ini adalah aplikasi BaikMu: Self Care bagian yang akan ditampilkan untuk pengguna berserta beragam fitur yang disediakan oleh aplikasi berupa Tes Kesehatan Mental, Penanganan, Mencari Psikiater, Catatan, Artikel dan Chatbot

Source Code

Source code back-end dari aplikasi BaikMu: Self Care mengguna framework Flask-Api yang digunakan sebagai route untuk melakukan login, register, edit pengguna, edit password dan menambahkan serta menampilkan artikel, back-end ini juga dijadikan sebagai website untuk menampilkan landing page pada Baikmu.com serta halaman developer.

1. Import Library

```
import sqlite3
import subprocess
from flask import Flask, make_response, jsonify, render_template,
session, request, redirect, url_for, send_file
from flask_restx import Resource, Api, reqparse
from flask_cors import CORS
from flask_sqlalchemy import SQLAlchemy
from werkzeug.security import generate_password_hash, check_password_hash
from datetime import datetime, timedelta
import jwt, os, random
from flask_mail import Mail, Message
```

Script tersebut adalah sebuah program Python yang menggunakan berbagai pustaka dan kerangka kerja seperti Flask, Flask-Restx, Flask-CORS, Flask-SQLAlchemy, dan Flask-Mail untuk membuat aplikasi web. Berikut adalah penjelasan singkat mengenai bagian-bagian dari script tersebut:

1. `import sqlite3`: Mengimpor modul `sqlite3` yang memungkinkan interaksi dengan database SQLite.
2. `import subprocess`: Digunakan untuk mengeksekusi perintah shell atau subproses dalam Python.
3. `from flask import ...`: Mengimpor berbagai modul dan fungsi dari Flask yang diperlukan untuk membuat aplikasi web, seperti pembuatan respon, rendering template HTML, session, dan lain-lain.
4. `from flask_restx import Resource, Api, reqparse`: Mengimpor modul Flask-Restx yang memudahkan pembuatan API RESTful dalam aplikasi Flask, termasuk definisi sumber daya (Resource), Api, dan `reqparse` untuk parsing argumen permintaan.

5. `from flask_cors import CORS`: Mengimpor modul Flask-CORS yang memungkinkan lintas sumber daya bersama antara aplikasi Flask dan domain yang berbeda.
6. `from flask_sqlalchemy import SQLAlchemy`: Mengimpor modul Flask-SQLAlchemy untuk interaksi dengan database SQL menggunakan SQLAlchemy, yang merupakan ORM (Object Relational Mapper) untuk Python.
7. `from werkzeug.security import generate_password_hash, check_password_hash`: Mengimpor fungsi-fungsi dari Werkzeug untuk menghasilkan dan memeriksa hash password.
8. `from datetime import datetime, timedelta`: Mengimpor modul datetime yang digunakan untuk manipulasi tanggal dan waktu dalam Python.
9. `import jwt, os, random`: Mengimpor modul-modul yang digunakan untuk melakukan operasi terkait JWT (JSON Web Token), operasi sistem, dan operasi acak.
10. `from flask_mail import Mail, Message`: Mengimpor modul Flask-Mail yang memungkinkan aplikasi Flask untuk mengirim email.

Setelah mengimpor semua modul yang diperlukan, script kemudian mendefinisikan dan mengkonfigurasi aplikasi Flask beserta berbagai fitur seperti CORS, SQLAlchemy untuk koneksi database, dan Mail untuk pengiriman email. Script ini kemungkinan besar akan berisi juga definisi rute-rute (routes) yang menangani permintaan dari klien, termasuk operasi-operasi seperti otentikasi, manipulasi data dalam database, dan pengiriman email.

2. Kofigurasi Database

```
app = Flask(__name__)
api = Api(app)
CORS(app)
app.config["SQLALCHEMY_DATABASE_URI"] =
"mysql://root:@127.0.0.1:3306/appkesmental"
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
app.config['SECRET_KEY'] = 'whateveryouwant'
# mail env config
app.config['MAIL_SERVER'] = 'smtp.gmail.com'
app.config['MAIL_PORT'] = 465
app.config['MAIL_USE_SSL'] = True
app.config['MAIL_USERNAME'] = "appkesihatanmental@gmail.com"
app.config['MAIL_PASSWORD'] = "kaqbvmemgmvr1er"
```

```

app.config['MAIL_DEFAULT_SENDER'] = app.config['MAIL_USERNAME']
mail = Mail(app)
# mail env config
db = SQLAlchemy(app)

```

Script di atas membuat objek aplikasi Flask (app) dan objek API Flask-Restx (api). Konfigurasi tambahan yang dilakukan termasuk:

1. CORS(app): Mengaktifkan Cross-Origin Resource Sharing (CORS) untuk memungkinkan permintaan lintas domain ke aplikasi Flask.
2. Konfigurasi database menggunakan SQLAlchemy:
 - o app.config["SQLALCHEMY_DATABASE_URI"]: Mendefinisikan URI database MySQL yang akan digunakan oleh aplikasi.
 - o app.config["SQLALCHEMY_TRACK_MODIFICATIONS"]: Menonaktifkan fitur pelacakan perubahan objek SQLAlchemy untuk meningkatkan performa.
3. app.config["SECRET_KEY"]: Mendefinisikan kunci rahasia yang digunakan oleh Flask untuk membuat sesi yang aman.
4. Konfigurasi pengiriman email menggunakan Flask-Mail:
 - o app.config["MAIL_SERVER"],app.config["MAIL_PORT"], app.config["MAIL_USE_SSL"]: Mendefinisikan pengaturan server SMTP Gmail dan penggunaan SSL untuk koneksi yang aman.
 - o app.config["MAIL_USERNAME"],app.config["MAIL_PASSWORD"]: Mendefinisikan kredensial pengguna Gmail yang akan digunakan untuk mengirim email.
 - o app.config["MAIL_DEFAULT_SENDER"]: Mendefinisikan pengirim default untuk email.

Setelah konfigurasi, objek Flask-Mail (mail) juga dibuat dengan menggunakan konfigurasi aplikasi (app), dan objek SQLAlchemy (db) juga dibuat dengan menggunakan konfigurasi aplikasi (app). Dengan konfigurasi ini, aplikasi Flask telah siap digunakan dengan koneksi database MySQL dan pengiriman email yang terkonfigurasi.

3. Route Register

```

@api.route('/register')
class Registration(Resource):
    @api.expect(regParser)
    def post(self):

```

```

# BEGIN: Get request parameters.
args      = regParser.parse_args()
nama      = args['nama']
email     = args['email']
password  = args['password']
password2 = args['confirm_password']
is_verified = True

# cek confirm password
if password != password2:
    return {
        'message': 'Password tidak cocok'
    }, 400

#cek email sudah terdaftar
user =
db.session.execute(db.select(Users).filter_by(email=email)).first()
if user:
    return "Email sudah terpakai silahkan coba lagi menggunakan
email lain"
user      = Users()
user.nama = nama
user.email = email
user.password = generate_password_hash(password)
user.is_verified = is_verified
db.session.add(user)
db.session.commit()
return {'message':
        'Registrasi Berhasil. Silahkan login.'}, 200

```

Dalam bagian kode tersebut, terdapat sebuah endpoint API /register yang digunakan untuk mendaftarkan pengguna baru ke dalam sistem. Ketika sebuah permintaan POST diterima ke endpoint ini, data yang diperlukan seperti nama, email, dan password diambil dari permintaan tersebut. Selanjutnya, dilakukan pengecekan apakah password yang dimasukkan oleh pengguna cocok dengan konfirmasi password yang juga dimasukkan. Jika tidak cocok, pesan kesalahan akan dikirimkan kembali ke pengguna dengan status kode 400.

Setelah itu, dilakukan pengecekan apakah email yang dimasukkan oleh pengguna sudah terdaftar sebelumnya dalam database. Jika sudah terdaftar, pesan kesalahan akan dikirimkan kembali ke pengguna.

Jika semua pengecekan telah berhasil, data pengguna baru akan dimasukkan ke dalam database. Password yang dimasukkan akan di-hash terlebih dahulu menggunakan fungsi

generate_password_hash dari Werkzeug untuk keamanan. Selanjutnya, pengguna akan ditandai sebagai terverifikasi (`is_verified = True`), karena mereka baru saja mendaftar. Data pengguna akan disimpan ke dalam tabel `Users` dalam database menggunakan SQLAlchemy. Transaksi kemudian di-commit untuk menyimpan perubahan ke dalam database.

Jika semua langkah berhasil dilakukan, respons sukses akan dikirimkan kembali ke pengguna dengan pesan bahwa registrasi telah berhasil dan mereka dapat melakukan login. Respons ini memiliki status kode 200 yang menunjukkan bahwa permintaan telah berhasil diproses tanpa ada kesalahan.

4. Route Login

```
@api.route('/login')
class LogIn(Resource):
    @api.expect(logParser)
    def post(self):
        args = logParser.parse_args()
        email = args['email']
        password = args['password']
        # cek jika kolom email dan password tidak terisi
        if not email or not password:
            return {
                'message': 'Email Dan Password Harus Diisi'
            }, 400
        #cek email sudah ada
        user = db.session.execute(
            db.select(Users).filter_by(email=email)).first()
        if not user:
            return {
                'message': 'Email / Password Salah'
            }, 400
        else:
            user = user[0]
        #cek password
        if check_password_hash(user.password, password):
            if user.is_verified == True:
                token= jwt.encode({
                    "user_id":user.id,
                    "user_email":user.email,
                    "exp": datetime.utcnow() + timedelta(hours= 1)
                },app.config['SECRET_KEY'],algorithm="HS256")
            # Set session variables
            session['user_id'] = user.id
```

```

        session['user_email'] = user.email
        session['token'] = token
        tup = email, ":", password
        #toString
        data = functools.reduce(operator.add, tup)
        byte_msg = data.encode('ascii')
        base64_val = base64.b64encode(byte_msg)
        code = base64_val.decode('ascii')
        msg = Message(subject='Verification OTP',
recipients=[user.email])
        session['email'] = user.email
        session['token'] = str(token)
        msg.html = render_template('verify_email.html',
token=token)
        mail.send(msg)
        return {'message' : 'Login Berhasil',
                'token' : token,
                'code' : code
                }, 200
    else:
        return {'message' : 'Email Belum Diverifikasi ,Silahkan
verifikasikan terlebih dahulu '}, 401
    else:
        return {
            'message': 'Email / Password Salah'
        }, 400

```

Pada endpoint /login, aplikasi mengizinkan pengguna untuk masuk ke sistem. Saat menerima permintaan POST, data yang dibutuhkan seperti alamat email dan kata sandi diambil dari permintaan tersebut. Dilakukan pemeriksaan terlebih dahulu apakah alamat email dan kata sandi sudah diisi oleh pengguna. Jika salah satu atau kedua kolom tersebut kosong, pesan kesalahan akan dikirimkan kembali ke pengguna dengan status kode 400. Setelah itu, dilakukan pencarian pengguna berdasarkan alamat email yang diberikan dalam database. Jika tidak ada pengguna yang terdaftar dengan alamat email yang diberikan, pesan kesalahan akan dikirimkan kembali ke pengguna dengan status kode 400.

Jika alamat email ditemukan dalam database, dilakukan verifikasi kata sandi yang dimasukkan oleh pengguna dengan menggunakan fungsi `check_password_hash`. Jika kata sandi sesuai dengan yang disimpan dalam database, dan pengguna sudah terverifikasi, maka akan di-generate JWT (JSON Web Token) yang berisi informasi pengguna seperti

ID pengguna, alamat email, dan waktu kedaluwarsa token. Token ini kemudian disimpan dalam sesi menggunakan session Flask.

Setelah itu, sebuah email verifikasi dikirimkan ke alamat email pengguna dengan mengandung token JWT sebagai bagian dari pesan HTML. Pesan tersebut dikirim menggunakan modul Flask-Mail yang telah dikonfigurasi sebelumnya. Respons sukses dikirimkan kembali ke pengguna dengan pesan bahwa login berhasil, bersamaan dengan token JWT dan kode verifikasi yang telah dienkripsi menggunakan base64. Jika pengguna belum terverifikasi, pesan kesalahan akan dikirimkan kembali ke pengguna dengan status kode 401. Jika kata sandi yang dimasukkan tidak sesuai, pesan kesalahan akan dikirimkan kembali ke pengguna dengan status kode 400.

5. Route Edit Pengguna

```
@api.route('/edit-user')
class EditUser(Resource):
    @api.expect(editParser)
    def put(self):
        args = editParser.parse_args()
        bearerAuth = args['Authorization']
        nama = args['nama']
        datenow = datetime.today().strftime('%Y-%m-%d %H:%M:%S')
        try:
            jwtToken = bearerAuth[7:]
            token = decodetoken(jwtToken)
            user =
Users.query.filter_by(email=token.get('user_email')).first()
            user.nama = nama
            user.updatedAt = datenow
            db.session.commit()
        except:
            return {
                'message' : 'Token Tidak valid,Silahkan Login Terlebih
Dahulu!'
            }, 400
        return {'message' : 'Update User Sukses'}, 200
```

Pada endpoint /edit-user, aplikasi memungkinkan pengguna untuk mengubah informasi profil mereka. Saat menerima permintaan PUT, data yang diperlukan seperti token autentikasi Bearer dan informasi yang ingin diubah (misalnya, nama pengguna baru) diambil dari permintaan tersebut.

Pertama-tama, token autentikasi Bearer diproses untuk mendapatkan JWT (JSON Web Token) yang dikirimkan oleh pengguna. Token tersebut kemudian di-dekode untuk memperoleh informasi pengguna, seperti alamat email pengguna yang diambil dari token.

Setelah mendapatkan alamat email pengguna dari token, dilakukan pencarian pengguna dalam database berdasarkan alamat email tersebut menggunakan SQLAlchemy. Jika pengguna ditemukan, informasi yang ingin diubah (dalam contoh ini, nama pengguna) akan diperbarui dalam entri database pengguna. Timestamp untuk pembaruan (updatedAt) juga diperbarui dengan tanggal dan waktu saat ini.

Jika terjadi kesalahan selama proses ini, misalnya token tidak valid atau ada kesalahan lain dalam proses pembaruan, pesan kesalahan akan dikirimkan kembali ke pengguna dengan status kode 400. Jika pembaruan berhasil dilakukan, respons sukses akan dikirimkan kembali ke pengguna dengan pesan bahwa pembaruan profil pengguna telah berhasil, bersamaan dengan status kode 200.

6. Route Edit Password

```
@api.route('/edit-password')
class Password(Resource):
    @api.expect(authParser, editPasswordParser)
    def put(self):
        args = editPasswordParser.parse_args()
        argss = authParser.parse_args()
        bearerAuth = argss['Authorization']
        cu_password = args['current_password']
        newpassword = args['new_password']
        try:
            jwtToken = bearerAuth[7:]
            token = decodetoken(jwtToken)
            user = Users.query.filter_by(id=token.get('user_id')).first()
            if check_password_hash(user.password, cu_password):
                user.password = generate_password_hash(newpassword)
                db.session.commit()
            else:
                return {'message': 'Password Lama Salah'}, 400
        except:
            return {
                'message': 'Token Tidak valid! Silahkan, Sign in!'
            }, 401
        return {'message': 'Password Berhasil Diubah'}, 200
```

Pada endpoint `/edit-password`, aplikasi memungkinkan pengguna untuk mengubah kata sandi mereka. Saat menerima permintaan PUT, data yang diperlukan seperti token autentikasi Bearer, kata sandi saat ini, dan kata sandi baru diambil dari permintaan tersebut.

Pertama-tama, token autentikasi Bearer dan informasi lainnya yang dibutuhkan untuk mengubah kata sandi diproses. Token Bearer di-dekode untuk mendapatkan informasi pengguna, khususnya ID pengguna yang kemudian digunakan untuk mencari pengguna dalam database.

Setelah mendapatkan informasi pengguna dari token, dilakukan pencarian pengguna dalam database berdasarkan ID pengguna tersebut menggunakan SQLAlchemy. Jika pengguna ditemukan, dilakukan pengecekan apakah kata sandi saat ini yang dimasukkan oleh pengguna cocok dengan kata sandi yang disimpan dalam database. Jika cocok, kata sandi baru akan di-hash menggunakan fungsi `generate_password_hash` dari Werkzeug, dan kata sandi pengguna akan diperbarui dalam entri database pengguna. Jika tidak cocok, pesan kesalahan akan dikirimkan kembali ke pengguna dengan status kode 400.

Jika terjadi kesalahan selama proses ini, misalnya token tidak valid atau ada kesalahan lain dalam proses pembaruan kata sandi, pesan kesalahan akan dikirimkan kembali ke pengguna dengan status kode 401. Jika pembaruan kata sandi berhasil dilakukan, respons sukses akan dikirimkan kembali ke pengguna dengan pesan bahwa kata sandi telah berhasil diubah, bersamaan dengan status kode 200.

7. Route Menampilkan Artikel

```
@app.route('/articlesmobile', methods=['GET'])
def get_articles():
    conn = pymysql.connect(
        host='localhost',
        port=3306,
        user='root',
        password='',
        database='appkesmental',
        charset='utf8mb4',
        cursorclass=pymysql.cursors.DictCursor
    )
    cursor = conn.cursor()
```

```

cursor.execute("SELECT * FROM articles")
articles = cursor.fetchall()

# Convert the fetched data into a list of dictionaries (to match the
current data structure)
article_list = [{'id': row['id'], 'title': row['title'], 'content':
row['content'], 'category': row['category'], 'link': row['link']} for row
in articles]

cursor.close()
conn.close()

return jsonify(article_list)

```

Endpoint `/articlesmobile` digunakan untuk mengambil artikel dari database dan mengirimkan data dalam format JSON. Saat menerima permintaan GET, koneksi ke database MySQL dibuat menggunakan modul PyMySQL. Setelah koneksi berhasil, sebuah kursor dibuat untuk menjalankan kueri SQL yang akan mengambil semua data dari tabel `articles`.

Setelah kueri dieksekusi, data artikel yang dihasilkan dari kueri tersebut diambil menggunakan metode `fetchall()`. Data tersebut kemudian diubah menjadi sebuah daftar (list) dari kamus (dictionary) di mana setiap kamus mewakili satu artikel, dengan kunci-kunci yang sesuai dengan nama kolom dalam tabel.

Setelah semua data diambil, kursor dan koneksi ditutup untuk mencegah kebocoran sumber daya. Akhirnya, data artikel yang telah diubah dimasukkan ke dalam format JSON menggunakan fungsi `jsonify()` dari Flask untuk dikirimkan kembali sebagai respons dari permintaan GET.

8. Route Menambahkan Artikel

```

@app.route('/post_article', methods=['POST'])
def post_article():
    # Mendapatkan data dari formulir
    title = request.form.get('title')
    content = request.form.get('content')
    category = request.form.get('category')
    link = request.form.get('link')

    # Validasi data
    if not title or not content:

```

```

        return jsonify({'message': 'Judul dan konten artikel
diperlukan'}), 400

    conn = pymysql.connect(
        host='localhost',
        port=3306,
        user='root',
        password='',
        database='appkesmental',
        charset='utf8mb4',
        cursorclass=pymysql.cursors.DictCursor
    )
    cursor = conn.cursor()

    try:
        # Menyimpan artikel baru ke database
        cursor.execute("INSERT INTO articles (title, content, category,
link) VALUES (%s, %s, %s, %s)",
                    (title, content, category, link))
        conn.commit()

        # Menutup kursor dan koneksi
        cursor.close()
        conn.close()

        # Mengarahkan kembali ke halaman HTML indeks setelah posting
berhasil
        return redirect(url_for('articles'))
    except Exception as e:
        return jsonify({'message': f'Gagal memposting artikel:
{str(e)}'}), 500

```

Endpoint `/post_article` digunakan untuk menambahkan artikel baru ke dalam database. Saat menerima permintaan POST, data artikel seperti judul, konten, kategori, dan tautan (link) diambil dari formulir yang dikirimkan.

Setelah data artikel berhasil diambil, dilakukan validasi untuk memastikan bahwa judul dan konten artikel tidak kosong. Jika salah satu atau kedua atribut tersebut kosong, pesan kesalahan akan dikirimkan kembali dalam format JSON dengan status kode 400.

Setelah validasi berhasil dilakukan, koneksi ke database MySQL dibuat menggunakan modul PyMySQL. Setelah koneksi berhasil, sebuah kursor dibuat untuk menjalankan kueri SQL yang akan menyimpan artikel baru ke dalam tabel `articles` dengan menggunakan pernyataan `INSERT`.

Jika penyimpanan artikel berhasil, transaksi di-commit untuk menyimpan perubahan ke dalam database. Cursor dan koneksi ditutup untuk mencegah kebocoran sumber daya.

Jika terjadi kesalahan selama proses ini, seperti kesalahan server atau kesalahan lainnya, pesan kesalahan akan dikirimkan kembali dalam format JSON dengan status kode 500.

Jika penyimpanan berhasil, pengguna akan diarahkan kembali ke halaman indeks menggunakan fungsi `redirect()` dan `url_for()` Flask.

9. Route Menampilkan Landing Page

```
@app.route('/landing')
def landing():
    return render_template('landing.html')
```

Endpoint `/landing` digunakan untuk menampilkan halaman landing HTML kepada pengguna. Saat pengguna membuat permintaan GET ke endpoint ini, fungsi `landing()` akan dijalankan.

Fungsi ini menggunakan fungsi `render_template()` dari Flask untuk merender halaman HTML dengan nama `landing.html`. Halaman tersebut kemungkinan berisi konten yang menjelaskan tentang aplikasi atau layanan yang ditawarkan, serta mungkin juga memiliki tautan atau tombol untuk menavigasi ke halaman lain dalam aplikasi. Pengguna akan menerima respons berupa halaman HTML yang di-render saat mereka mengakses endpoint `/landing`.

Untuk bagian Mobile Aplikasi menggunakan Flutter dengan bahasa pemrograman dart untuk menghasilkan tampilan dan kebutuhan lainya yang akan membantu mengkoneksikan antara Flutter dan Flask-API.

10. Api Service Flutter to Flask-API

```
class ApiService {
    static String getBaseUrl() {
        return 'https://baikmu.com';
    }

    static String getLoginEndpoint() {
        return '${getBaseUrl()}/loginmobile';
    }
}
```

```

}

static String getLogoutEndpoint() {
    return '${getBaseUrl()}/logout';
}

static String getRegistEndpoint() {
    return '${getBaseUrl()}/register';
}

static String getCodeEndpoint() {
    return '${getBaseUrl()}/bearer-auth';
}

static String getEditUserEndpoint() {
    return '${getBaseUrl()}/edit-user';
}

static String getEditPasswordEndpoint() {
    return '${getBaseUrl()}/edit-password';
}

static String getChatEndpoint() {
    return '${getBaseUrl()}/chatbot';
}

static String getArticlesEndpoint() {
    return '${getBaseUrl()}/articlesmobile';
}

static String getNewArticlesEndpoint() {
    return '${getBaseUrl()}/articlesmobilebaru';
}
}

```

Kelas ApiService ini berfungsi sebagai utilitas untuk mendapatkan URL berbagai endpoint dalam aplikasi. Berikut adalah penjelasan singkat tentang setiap metode dalam kelas ini:

1. `getBaseUrl()`: Metode ini mengembalikan URL dasar dari aplikasi, dalam hal ini `'https://baikmu.com'`.
2. `getLoginEndpoint()`: Metode ini mengembalikan URL untuk endpoint login, yang merupakan gabungan dari URL dasar dan `'/loginmobile'`.

3. `getLogoutEndpoint()`: Metode ini mengembalikan URL untuk endpoint logout, yang merupakan gabungan dari URL dasar dan `'/logout'`.
4. `getRegistEndpoint()`: Metode ini mengembalikan URL untuk endpoint registrasi, yang merupakan gabungan dari URL dasar dan `'/register'`.
5. `getCodeEndpoint()`: Metode ini mengembalikan URL untuk endpoint kode autentikasi, yang merupakan gabungan dari URL dasar dan `'/bearer-auth'`.
6. `getEditUserEndpoint()`: Metode ini mengembalikan URL untuk endpoint edit pengguna, yang merupakan gabungan dari URL dasar dan `'/edit-user'`.
7. `getEditPasswordEndpoint()`: Metode ini mengembalikan URL untuk endpoint edit kata sandi pengguna, yang merupakan gabungan dari URL dasar dan `'/edit-password'`.
8. `getChatEndpoint()`: Metode ini mengembalikan URL untuk endpoint chatbot, yang merupakan gabungan dari URL dasar dan `'/chatbot'`.
9. `getArticlesEndpoint()`: Metode ini mengembalikan URL untuk endpoint artikel, yang merupakan gabungan dari URL dasar dan `'/articlesmobile'`.
10. `getNewArticlesEndpoint()`: Metode ini mengembalikan URL untuk endpoint artikel baru, yang merupakan gabungan dari URL dasar dan `'/articlesmobilebaru'`.

Dengan menggunakan kelas `ApiService` ini, Anda dapat dengan mudah mengakses URL berbagai endpoint dalam aplikasi tanpa perlu menuliskan URL secara langsung di setiap bagian kode. Ini memudahkan dalam pemeliharaan dan perubahan URL, serta membuat kode menjadi lebih mudah dibaca dan dimengerti.

BaikMu

Self Care



BaikMu Self Care

MANUAL BOOK

VIRTUAL ASSISTANT KESEHATAN MENTAL

Irfan Triadi Saputra
Slamet Wiyono, S.Pd., M.Eng
Dwi Intan Af'idah, M.Kom

Daftar Isi

Daftar Isi	1
Daftar Gambar	2
Pendahuluan.....	3
Instalasi Aplikasi	4
Tampilan Awal	5
Penggunaan Aplikasi	6
1. Registrasi dan Login	6
2. Dashboard.....	8
3. Tes Kesehatan Mental.....	9
4. Penanganan.....	10
5. Mencari Psikiater	14
6. Chatbot.....	15
7. Profil.....	16
8. Artikel.....	17
9. Logout.....	18
Pemecahan Masalah	19
1. Tidak Bisa Login	19
2. Aplikasi Tidak Merespon.....	19

Daftar Gambar

Gambar 1. Logo BaikMu.....	3
Gambar 2. Cara Installasi	4
Gambar 3. Splash Screen.....	5
Gambar 4. Tampilan Awal	5
Gambar 5. Register.....	6
Gambar 6. Login.....	7
Gambar 7. Dashboard.....	8
Gambar 8. Tes Kesehatan Mental	9
Gambar 9. Penanganan.....	10
Gambar 10. Meditasi	11
Gambar 11. Musik Relaksasi	11
Gambar 12. Pengingat	12
Gambar 13. Fitur Bercecerita.....	13
Gambar 14. Fitur Cari Psikater.....	14
Gambar 15. Chatbot.....	15
Gambar 16. Profile	16
Gambar 17. Edit Profile.....	16
Gambar 18. Fitur Artikel	17
Gambar 19. Isi Artikel	17
Gambar 20. Logout.....	18

Pendahuluan

Selamat datang di BaikMu Self Care, sebuah aplikasi virtual assistant kesehatan mental yang dirancang khusus untuk membantu Anda menjaga dan meningkatkan kesehatan mental Anda dengan lebih baik. Dalam era modern yang penuh dengan tantangan dan tekanan, kesehatan mental menjadi aspek yang sangat penting untuk diperhatikan. BaikMu Self Care hadir untuk memberikan dukungan yang Anda butuhkan melalui berbagai fitur yang mendukung self-care serta layanan konsultasi kesehatan mental secara virtual.

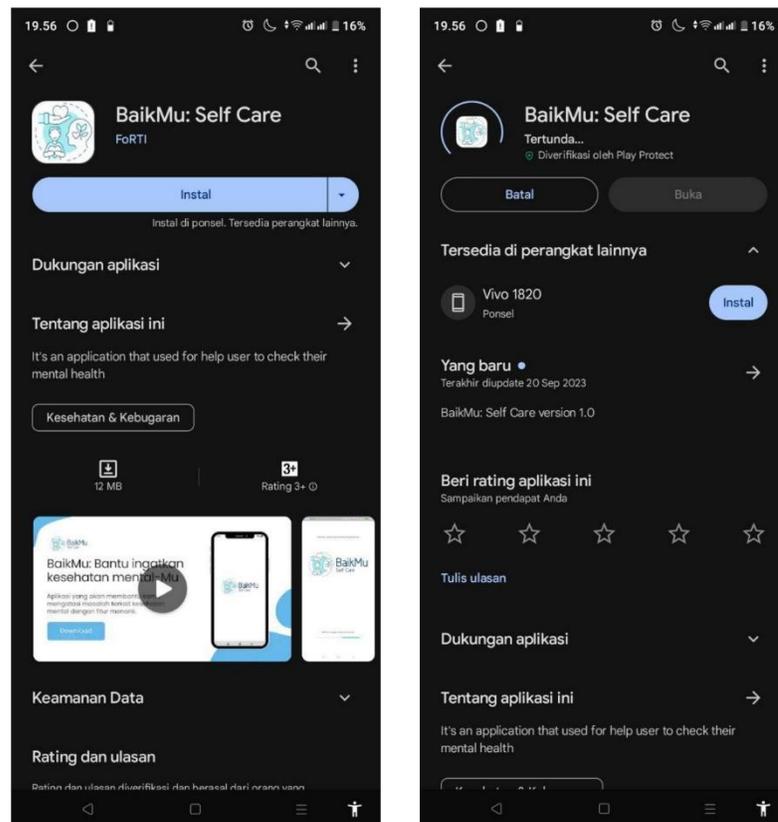
BaikMu Self Care dirancang dengan mempertimbangkan kebutuhan pengguna dari berbagai latar belakang. Aplikasi ini menawarkan pendekatan yang holistik dan personal dalam menangani isu-isu kesehatan mental. Dengan menggunakan teknologi canggih, BaikMu Self Care berkomitmen untuk menyediakan layanan yang berkualitas tinggi dan mudah diakses oleh siapa saja, kapan saja.



Gambar 1. Logo BaikMu

Instalasi Aplikasi

1. Buka Google Play Store.
2. Cari "BaikMu Self Care".
3. Klik tombol "Install" untuk mengunduh dan menginstal aplikasi.



Gambar 2. Cara Instalasi

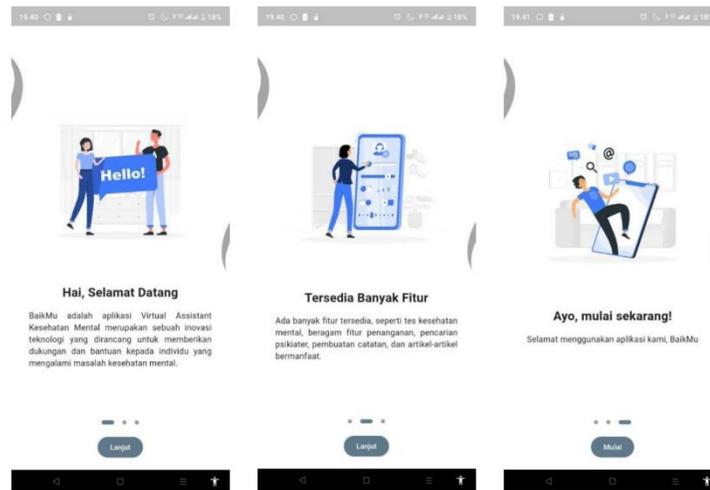
Tampilan Awal

Saat Anda pertama kali membuka aplikasi BaikMu Self Care, Anda akan melihat splash screen seperti yang ditunjukkan pada gambar berikut:



Gambar 3. Splash Screen

Pada splash screen ini, Anda akan melihat logo aplikasi, nama aplikasi "BaikMu Self Care", dan tagline "Virtual Assistant Kesehatan Mental". Setelah beberapa saat, aplikasi akan menuju ke halaman sapaan untuk pengguna.



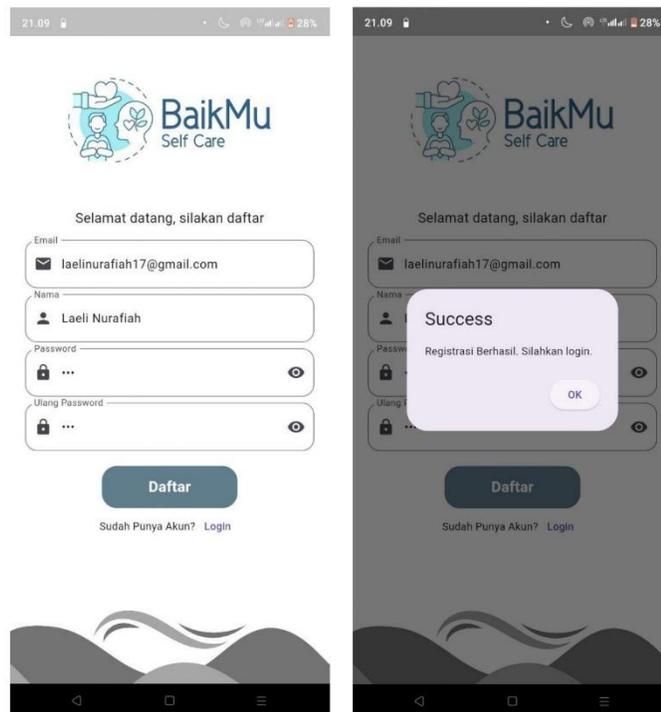
Gambar 4. Tampilan Awal

Penggunaan Aplikasi

1. Registrasi dan Login

a. Registrasi

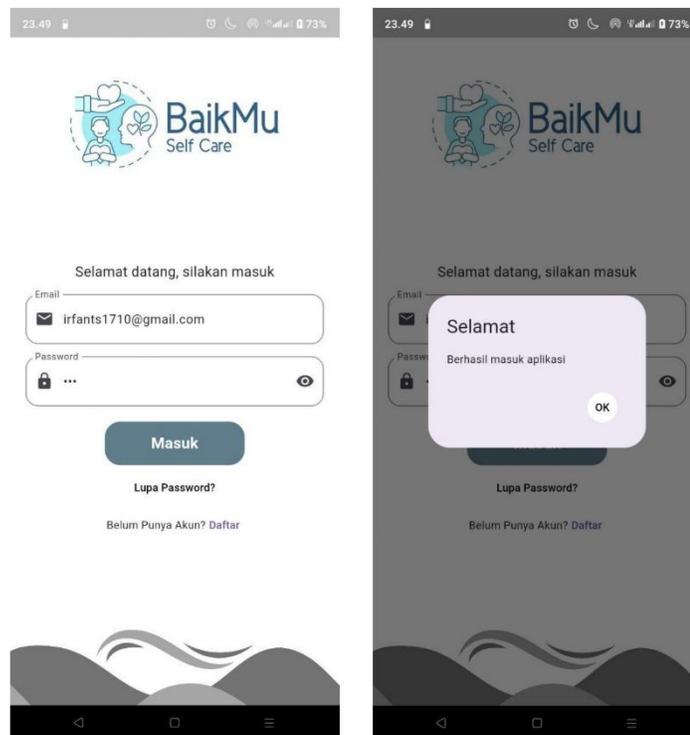
- 1) Buka aplikasi BaikMu Self Care.
- 2) Pilih "Daftar" untuk membuat akun baru.
- 3) Isi formulir pendaftaran dengan informasi yang diminta, seperti, email, nama dan password.
- 4) Klik "Daftar" untuk menyelesaikan proses pendaftaran.



Gambar 5. Register

b. Login

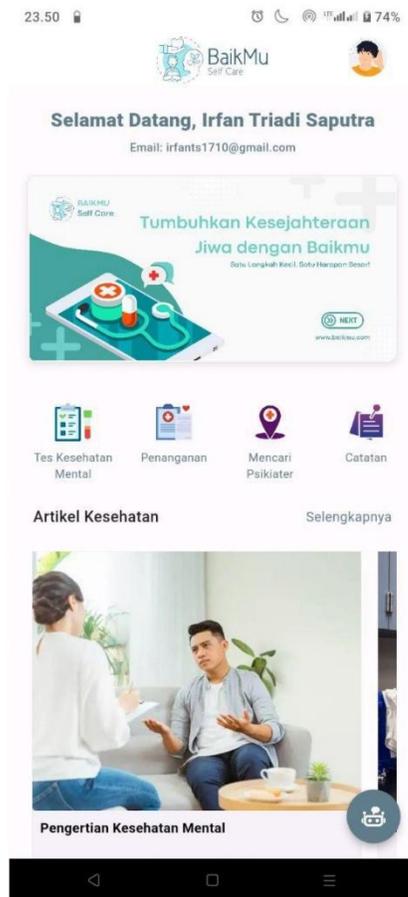
- 1) Buka aplikasi BaikMu Self Care.
- 2) Pilih "Login".
- 3) Masukkan email dan kata sandi yang telah Anda daftarkan.
- 4) Klik "Login" untuk masuk ke akun Anda.



Gambar 6. Login

2. Dashboard

Setelah berhasil login, Anda akan diarahkan ke dashboard. Di sini Anda dapat mengakses berbagai fitur utama aplikasi, termasuk tes kesehatan mental, penanganan, mencari psikiater, chatbot, profil, artikel, dan logout.

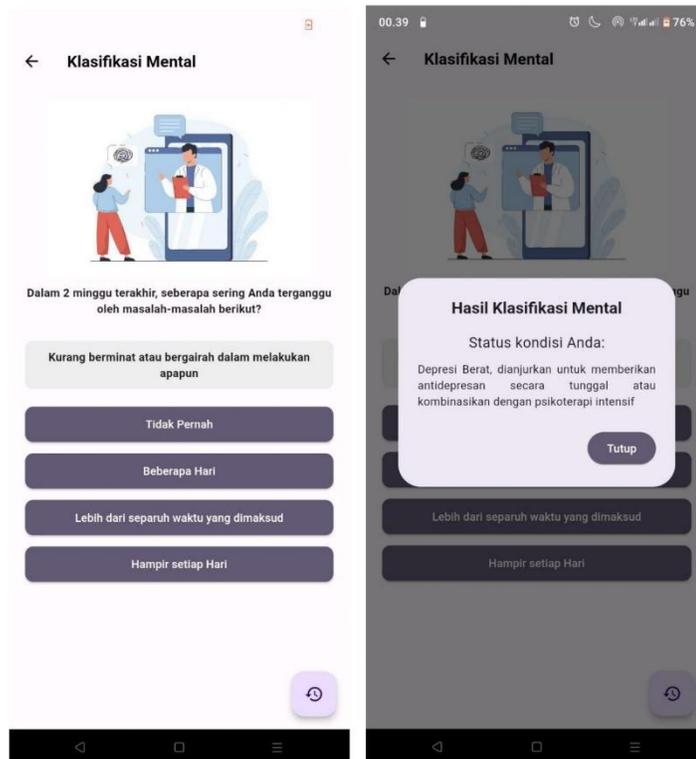


Gambar 7. Dashboard

3. Tes Kesehatan Mental

Aplikasi ini menyediakan tes kesehatan mental dengan beberapa pertanyaan yang dirancang untuk membantu Anda memahami kondisi mental Anda. Berikut cara mengaksesnya:

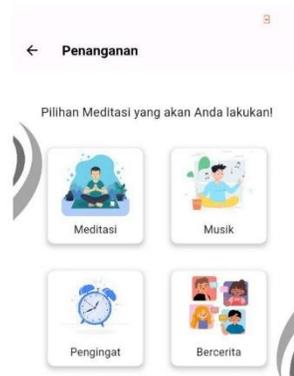
- Pilih menu "Tes Kesehatan Mental" di dashboard.
- Jawab setiap pertanyaan dengan jujur.
- Setelah selesai, Anda akan mendapatkan hasil tes dan rekomendasi berdasarkan jawaban Anda.



Gambar 8. Tes Kesehatan Mental

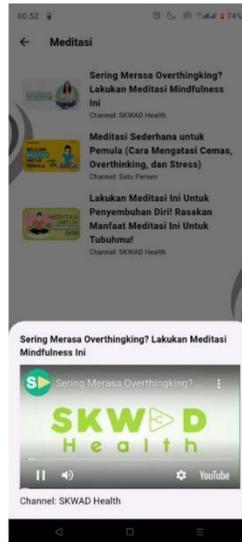
4. Penanganan

Fitur penanganan menawarkan berbagai metode untuk membantu Anda mengatasi masalah kesehatan mental, yaitu meditasi, music relaksaksi, pengingat dan bercerita:



Gambar 9. Penanganan

- a. Meditasi: Berbagai sesi meditasi yang dapat membantu Anda merasa lebih tenang dan rileks.



Gambar 10. Meditasi

- b. Musik Relaksasi: Pilihan musik yang dirancang untuk membantu Anda mengurangi stres dan meningkatkan suasana hati.



Gambar 11. Musik Relaksasi

c. Pengingat: Fitur untuk mengatur pengingat untuk melakukan aktivitas self-care.



Gambar 12. Pengingat

d. Bercerita: Ruang untuk menulis dan mencatat perasaan Anda, membantu Anda dalam proses refleksi diri.

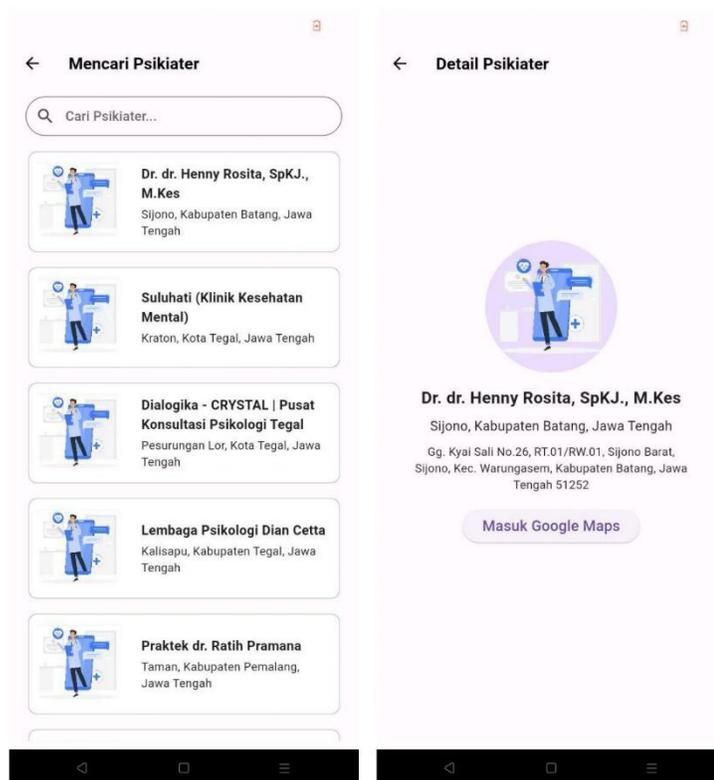


Gambar 13. Fitur Ber cerita

5. Mencari Psikiater

Jika Anda memerlukan bantuan profesional lebih lanjut, Anda dapat mencari dan menjadwalkan sesi dengan psikiater melalui aplikasi:

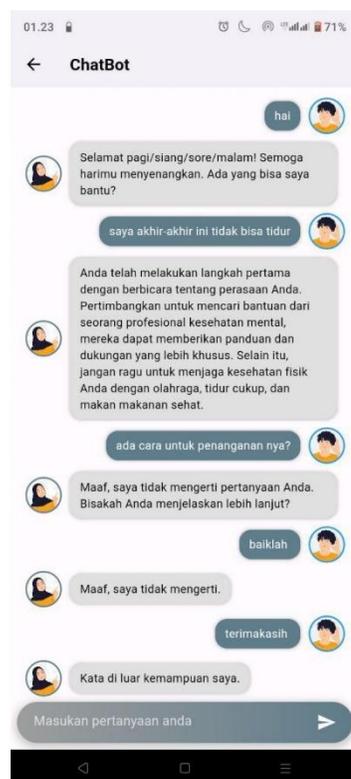
- a. Pilih menu "Mencari Psikiater".
- b. Telusuri daftar psikiater yang tersedia.
- c. Pilih psikiater yang Anda inginkan dan jadwalkan sesi konsultasi.



Gambar 14. Fitur Cari Psikiater

6. Chatbot

Fitur chatbot menyediakan dukungan instan dan jawaban atas pertanyaan umum seputar kesehatan mental. Anda bisa berinteraksi dengan chatbot kapan saja untuk mendapatkan informasi dan saran.



Gambar 15. Chatbot

7. Profil

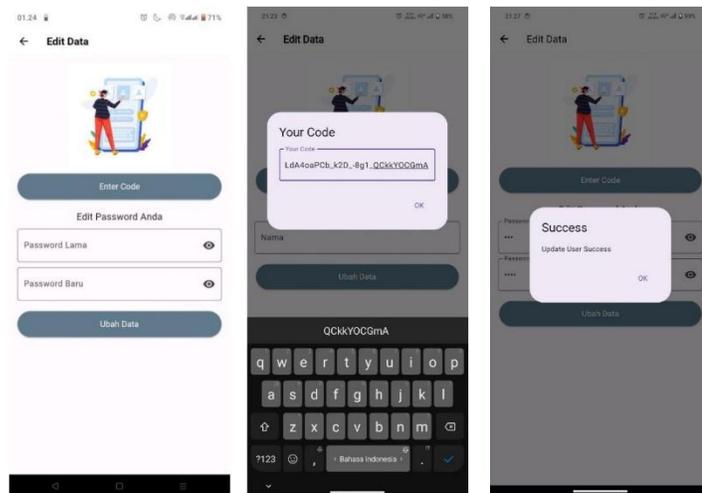
Anda dapat mengakses dan mengatur informasi akun Anda di menu profil:

- a. Pilih menu "Profil".



Gambar 16. Profile

- b. Edit informasi seperti nama, email, dan preferensi lainnya.
- c. Simpan perubahan untuk memperbarui profil Anda.



Gambar 17. Edit Profile

8. Artikel

Aplikasi ini menyediakan berbagai artikel yang berfokus pada kesehatan mental dan self-care:

- a. Pilih "Artikel" pada halaman dashboard.



Gambar 18. Fitur Artikel

- b. Telusuri dan baca artikel yang tersedia.
- c. Artikel ini mencakup topik-topik seperti manajemen stres, teknik relaksasi, dan tips untuk meningkatkan kesejahteraan mental.

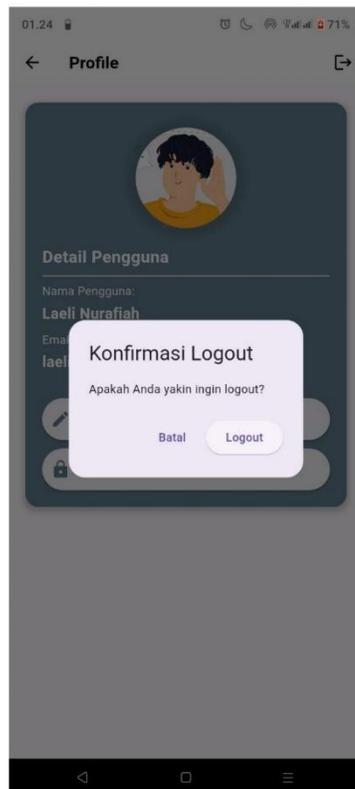


Gambar 19. Isi Artikel

9. Logout

Untuk keluar dari aplikasi:

- a. Pilih menu "Logout" di dashboard.
- b. Konfirmasi untuk keluar dari akun Anda.



Gambar 20. Logout

Pemecahan Masalah

1. Tidak Bisa Login

- a. Periksa Koneksi Internet: Pastikan perangkat Anda terhubung ke internet.
- b. Cek Email dan Kata Sandi: Pastikan Anda memasukkan email dan kata sandi yang benar.

2. Aplikasi Tidak Merespon

- a. Tutup dan Buka Kembali Aplikasi: Coba tutup aplikasi dan buka kembali.
- b. Update Aplikasi: Pastikan Anda menggunakan versi terbaru dari aplikasi.

Lampiran 5. Sertifikat HKI


REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202446246, 6 Juni 2024

Pencipta

Nama : **Irfan Triadi Saputra, Slamet Wiyono, S.Pd., M.Eng dkk**
Alamat : Dusun 06 AgAA Bandelan, RT RW 006-007, Desa Taman, Kecamatan Taman, Kabupaten Pemalang, Propinsi Jawa Tengah 52361, Taman, Pemalang, Jawa Tengah, 52361
Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : **Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik Harapan Bersama**
Alamat : Jalan Mataram No. 9, Pesurungan Lor, Kecamatan Margadana 52142, Margadana, Tegal, Jawa Tengah 52142
Kewarganegaraan : Indonesia

Jenis Ciptaan : **Program Komputer**
Judul Ciptaan : **BalkMu: Self Care Virtual Assistant Kesehatan Mental**
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 6 Juni 2024, di Tegal
Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor pencatatan : 000621601

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL
a.b
Direktur Hak Cipta dan Desain Industri


IGNATIUS M.T. SILALAH
NIP. 196812301996031001



Disclaimer:
Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.

Lampiran 6. Lembar Bimbingan



**SARJANA TERAPAN TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA**

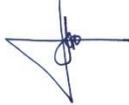
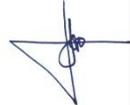
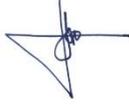
LEMBAR BIMBINGAN SKRIPSI

Nama : Irfan Triadi Saputra
Nim : 20090017
No. Ponsel : 082324282477
Judul TA : BAIKMU SELF CARE: Virtual Assistant Kesehatan Mental Untuk Generasi Z Berbasis Mobile
Dosen Pembimbing I : Slamet Wiyono, S.Pd., M.Eng

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1.	22 Maret 2024	Konsep Aplikasi	<ul style="list-style-type: none"> - Memfokuskan pada fitur utama - Mendemokan aplikasi pada bimbingan selanjutnya 	
2.	5 April 2024	User Interface Aplikasi	<ul style="list-style-type: none"> - Menambahkan Lupa Password - Memperbaiki tampilan artikel jangan pakai headline tapi buat vertikal saja agar tidak terkesan seperti berita - Memperbaiki Icon dan Penamaan agar lebih pada fitur agar lebih familiar karena targetnya masyarakat luas - Memberikan opening untuk mengenalkan aplikasinya dan mengarahkan ke fitur aplikasinya 	
3.	19 April 2024	Interface Artikel dan Lupa Password	<ul style="list-style-type: none"> - Melanjutkan eror pada lupa password yang masih belum terselesaikan - Menambahkan tampilan sambutan pada home, agar aplikasi lebih interaktif 	
4.	3 Mei 2024	Kelengkapan Website dan Rencana Pengujian Aplikasi Mobile	<ul style="list-style-type: none"> - Menambahkan kelengkapan yang akan ditampilkan pada website - Membuat perancangan Uji Pengguna terkait 	



**SARJANA TERAPAN TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA**

5.	20 Mei 2024	Menampilkan hasil perancangan uji pengguna dan menampilkan hasil pengujian fungsional	dengan kebermanfaatan kepada pengguna - Membuat Uji Fungsional Aplikasi - Melakukan implementasi pengujian pengguna - Memperbaiki bug pada website aplikasi	
6.	31 Mei 2024	Menampilkan hasil pengujian	- Memperbaiki hasil pengujian agar lebih realistis - Memperbaiki bug pada aplikasi	
7.	14 Juni 2024	Progres memperbaiki bug aplikasi dan hasil pengujian aplikasi	- Menambahkan grafik pada hasil pengujian pre test dan post test - Menyelesaikan laporan	
8.	28 Juni 2024	Laporan Skripsi	- Acc Laporan	

Tegal, 8 Juli 2024
Dosen Pembimbing I



Slamet Wiyono, S.Pd., M.Eng
NIPY. 08.015.222



**SARJANA TERAPAN TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA**

LEMBAR BIMBINGAN SKRIPSI

Nama : Irfan Triadi Saputra
Nim : 20090017
No. Ponsel : 082324282477
Judul TA : BAIKMU SELF CARE: Virtual Assistant Kesehatan Mental Untuk Generasi Z Berbasis Mobile
Dosen Pembimbing II : Dwi Intan Af'idah, M.Kom

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1.	18 Maret 2024	Konsep produk skripsi	<ul style="list-style-type: none">- Tidak perlu psikiater terdekat- Fitur meditasi diperluas menjadi fitur perawatan/pengobatan- Fitur reminder	
2.	4 April 2024	Update Konsep dan Fitur penanganan, Rencana Luaran (HKI)	<ul style="list-style-type: none">- Menyatukan Fitur pengingat tidur dan minum obat menjadi satu dengan nama fitur Pengingat- Berikan kategori pilihan pengingat tidur dan minum obat dan berikan default yang sesuai	
3.	24 April 2024	Update semua fitur telah berhasil dibuat	<ul style="list-style-type: none">- Masuk ke pengujian fungsional	
4.	17 Mei 2024	Hasil Pengujian fungsional aplikasi	<ul style="list-style-type: none">- Masuk ke pengujian pengguna- Mempersiapkan dokumen untuk HKI	



SARJANA TERAPAN TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA

5.	31 Juni 2024	<ul style="list-style-type: none">- Menampilkan hasil pengujian- Menampilkan dokumen teknis dan manual book	<ul style="list-style-type: none">- Melanjutkan ke Laporan	
6.	28 Juni 2024	Laporan Skripsi	<ul style="list-style-type: none">- Memperbaiki Latar belakang- Memperbaiki Data yang dibutuhkan- Memperbaiki Bab 2	
7.	5 Juli 2024	Laporan Skripsi	<ul style="list-style-type: none">- Memperbaiki Kesimpulan dan Saran	
8.	8 Juli 2024	Laporan Skripsi	<ul style="list-style-type: none">- Acc Laporan	

Tegal, 8 Juli 2024
Dosen Pembimbing II

Dwi Intan Af'idah, M.Kom
NIPY. 11.020.470