

LAMPIRAN

Lampiran 1. Surat Kesepakatan Bimbingan Skripsi

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini:

Pihak Pertama

Nama : Rafael Rinta Primadani
NIM : 21090078
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Dwi Intan Af'idah, S.T., M.Kom.
Status : Dosen
NIDN : 062008903
Jabatan Fungsional : Lektor
Pangkat/Golongan : Penata, III/c

Pada hari ini Kamis tanggal 24 Maret 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Skripsi Pihak Pertama dengan syarat adanya peningkatan progres dari proyek skripsi yang dilaporkan setiap satu pekan. Jika Pihak Pertama tidak memenuhi syarat tersebut, maka Pihak Kedua berhak memutuskan kesepakatan ini. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Tugas Akhir

Tegal, 24 Maret 2025

Pihak Pertama,

Pihak Kedua,



Rafael Rinta Primadani



Dwi Intan Af'idah, S.T., M.Kom.

Mengetahui,
Ka. Prodi Sarjana Terapan Teknik Informatika



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

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini:

Pihak Pertama

Nama : Rafael Rinta Primadani
NIM : 21090078
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Sharfina Febbi Handayani, S.Kom., M.Kom.
Status : Dosen
NIDN : 0617029201
Jabatan Fungsional : Asisten Ahli
Pangkat/Golongan : Penata, III/c

Pada hari ini Jum'at tanggal 07 Maret 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Tugas Akhir Pihak Pertama dengan syarat Pihak Pertama wajib melakukan bimbingan Tugas Akhir minimal 8 kali kepada Pihak Kedua. Adapun waktu dan tempat pelaksanaan disepakati antar pihak dilaksanakan setiap hari rabu atau kamis pukul 08.00-12.00 wib dengan menunjukan progres. Pihak pertama menyatakan bersedia untuk menyelesaikan skripsi tepat waktu dan apabila pihak pertama tidak menyelesaikan skripsi tepat waktu sesuai dengan kesepakatan maka pihak ke dua tidak akan memberikan rekomendasi untuk ujian skripsi.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Tugas Akhir

Tegal, 24 Maret 2025

Pihak Pertama,

Pihak Kedua,



Rafael Rinta Primadani



Sharfina Febbi Handayani, S.Kom., M.Kom.

Mengetahui,
Ka. Prodi Sarjana Terapan Teknik Informatika



Dyan Sudhan, 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. N a m a : Rafael Rinta Primadani
Kewarganegaraan : Indonesia
Alamat : Jl. Moh. Toha No.163 RT.002, RW.003, Kelurahan
Kaligangsa, Kota Tegal, Provinsi Jawa Tengah, 52147
2. N a m a : Dwi Intan Afidah, S.T., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Grinting RT003/RW002, Kecamatan Bulakamba, Kabupaten
Brebek, Provinsi Jawa Tengah, 52253
3. N a m a : Sharfina Febbi Handayani, S.Kom., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Desa Grobog Kulon RT.002, RW.006, Kecamatan Pangkah
Kabupaten Tegal, 52471

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:
Berupa : Program Komputer
Berjudul : Sistem Perawatan Kesehatan Mental Secara Mandiri
Berbasis Website
 - 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 langgar, 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 perkara 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, Juni 2025



(Rafael Rinta Primadani)



(Dwi Irtan Afidah, S.T., M.Kom.)



(Sharfina Febbi Handayani, S.Kom., M.Kom.)

Lampiran 3. Surat Pengalihan HKI

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

1. N a m a : Rafael Rinta Primadani
Kewarganegaraan : Indonesia
Alamat : Jl. Moh. Toha No.163 RT.002, RW.003, Kelurahan Kaligangsa,
Kecamatan Margadana, Kota Tegal, Provinsi Jawa Tengah, 52147
2. N a m a : Dwi Intan Afidah, S.T., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Grinting RT003/RW002, Kecamatan Bulakamba, Kabupaten Brebes,
Provinsi Jawa Tengah, 52253
3. N a m a : Sharfina Febbi Handayani, S.Kom., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Desa Grobog Kulon RT.002, RW.006, Kecamatan Pangkah,
Kabupaten Tegal, Provinsi Jawa Tengah, 52471

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

N a m a : Pusat Penelitian dan Pengabdian Masyarakat (P3M)
Politeknik Harapan Bersama
Alamat : Jl.Mataram No.9 Pesurungan Lor Kota Tegal

Adalah **Pihak II** selaku Pemegang Hak Cipta berupa Program Komputer dengan judul "Sistem Perawatan Kesehatan Mental Secara Mandiri Berbasis Website". 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.

Pemegang Hak Cipta
Kepala P3M

(Muhammad Fikri Hidayatullah, S.T., M.Kom.)

Tegal, Juni 2025
Pencipta



(Rafael Rinta Primadani)

(Dwi Intan Afidah, S.T., M.Kom.)

(Sharfina Febbi Handayani, S.Kom., M.Kom.)

Lampiran 4. Manual Book dan Dokumen Teknikal



DAFTAR ISI

DAFTAR ISI	2
PENDAHULUAN	3
AKSES APLIKASI	4
TAMPILAN AWAL	5
PENGUNAAN	6
1. <i>Register dan Login</i>	6
2. <i>Dashboard</i>	8
3. <i>Better Journal</i>	8
4. <i>Better Test</i>	9
5. <i>Better Breath</i>	10
6. <i>Better Music</i>	11
7. <i>Better Affirmation</i>	12
8. <i>Rekap</i>	12

PENDAHULUAN

Selamat datang di Feelbetter, teman setia Anda dalam perjalanan menuju kesehatan mental yang lebih baik. Dalam era di mana kesehatan mental menjadi prioritas yang semakin penting, Feelbetter menyediakan akses mudah dan terjangkau ke berbagai alat bantu. Feelbetter hadir sebagai aplikasi web inovatif yang dirancang khusus untuk mendampingi Anda dalam merawat kesehatan mental secara mandiri.

Feelbetter dirancang sebagai solusi komprehensif untuk kebutuhan perawatan kesehatan mental mandiri bagi pengguna dari segala kalangan. Aplikasi ini menawarkan pendekatan yang holistik dan personal, mengintegrasikan metode-metode terapi yang efektif dalam satu platform yang mudah digunakan. Dengan menggunakan teknologi berbasis web yang ringan, Feelbetter berkomitmen untuk menyediakan alat-alat terapi mandiri yang berkualitas tinggi, gratis, dan mudah diakses oleh siapa saja yang memiliki koneksi internet, kapan saja mereka membutuhkan bantuan.



Gambar 1. Logo Feelbetter

AKSES APLIKASI

Untuk menjalankan website Feelbetter akan dijelaskan sebagai berikut:

1. Siapkan *browser* yang tersedia pada perangkat kamu seperti(*Firefox, Google Chrome, Microsoft Edge*, dll).
2. Buka *website*, dengan memasukan *link* Feelbetter.
3. Tekan tombol *enter* pada *keyboard*.

TAMPILAN AWAL

Setelah Anda membuka *website* Feelbetter, Anda akan melihat *landing page website*. Halaman ini menampilkan beberapa *section* seperti navigasi untuk menuju ke halaman *login*, hero banner, dan *section* yang menjelaskan mengenai aplikasi Feelbetter. Berikut tampilan *landing page* bisa dilihat pada gambar 2.



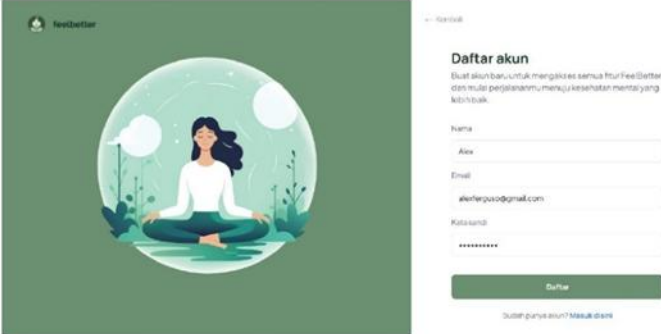
Gambar 2. Tampilan *landing page*

PENGUNAAN

1. *Register dan Login*

a. *Register*

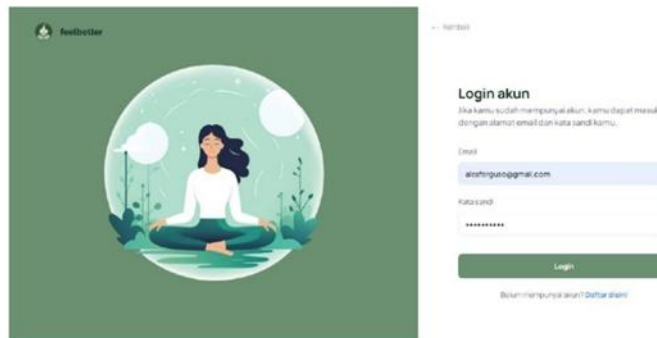
- 1) Pastikan sudah berada di *landing page*.
- 2) Tekan tombol “mulai” pada menu navigasi.
- 3) Tekan tombol “Daftar disini” untuk menuju ke halaman daftar.
- 4) Isi formulir yang diminta seperti nama, email, dan kata sandi.
- 5) Tekan tombol “Daftar” untuk menyelesaikan.



Gambar 3. Tampilan halaman *register*

b. *Login*

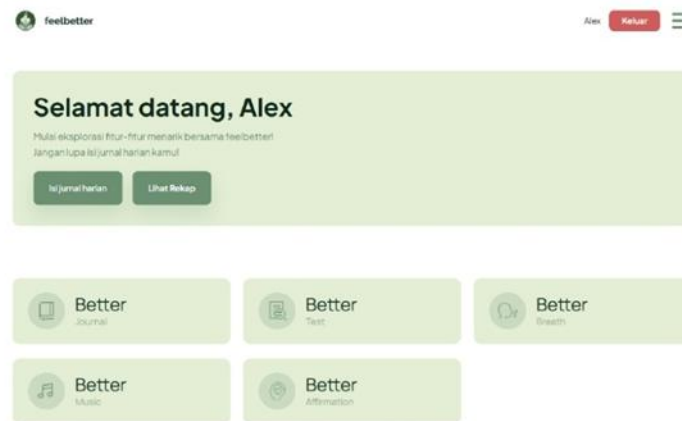
- 1) Tekan tombol “mulai” pada menu navigasi.
- 2) Isi formulir yang diminta seperti email dan kata sandi.
- 3) Tekan tombol “Login” untuk masuk.



Gambar 4. Tampilan halaman Login

2. *Dashboard*

Anda akan diarahkan ke *dashboard*, setelah berhasil *login*. Pada halaman *dashboard*, berisi fitur utama aplikasi, seperti *better journal*, *better tes*, *better breath*, *better music*, dan *better affirmation*. Berikut tampilan halaman *dashboard* bisa dilihat pada gambar 5.



Gambar 5. Tampilan halaman *dashboard*

3. *Better Journal*

Dalam fitur *Better Journal*, Anda dapat menulis catatan tentang pengalaman, perasaan, dan pemikiran Anda setiap hari yang membantu Anda untuk merefleksikan diri. Berikut cara mengaksesnya :

- Tekan card “*Better Journal*” pada *dashboard*.
- Tekan tombol “Selanjutnya”.
- Isi catatan Anda pada kolom yang disediakan.
- Tekan tombol “Selesai” jika Anda sudah selesai mengisi catatan.

Gambar 6. Tampilan halaman *Better Journal*

4. *Better Test*

Dalam fitur *Better Test*, terdapat beberapa pertanyaan yang membantu memahami kondisi mental Anda, pertanyaan ini menggunakan metode *Patient Health Questionnaire-9*(PHQ-9). Adapun cara mengaksesnya :

- Tekan card "*Better Test*" pada dashboard.
- Isi setiap pertanyaan sesuai dengan kondisi Anda.
- Tekan tombol "Selesai" jika Anda sudah mengisi semua pertanyaan.
- Anda akan mendapatkan hasil tes dan rekomendasi berdasarkan jawaban Anda.

Hasil Tes PHQ-9

Skor Total: 6

Interpretasi:
Depresi ringan

Rekomendasi:
Perhatikan perubahan suasana hati kamu. Cobalah fitur *Better Breath* untuk relaksasi pernapasan harian, dengan *Better Must* pada nomor 1 dan 2 untuk membantu meningkatkan suasana hati, dan praktikkan *Better Affirmation* pada nomor 3 dan 4 setiap pagi. Tindakan pencegahan pemrosesan kamu secara berkala di *Better Journal*.

[Mulai Ulang Tes](#)

Gambar 7. Tampilan halaman *Better Test*

5. *Better Breath*

Dalam fitur *Better Breath*, Anda dapat memilih 2 opsi menit untuk melakukan relaksasi pernapasan dengan teknik 4-7-8. Fitur ini untuk membantu Anda menenangkan pikiran dan tubuh melalui pengaturan pola pernapasan. Berikut cara mengaksesnya :

- a. Tekan *card* “*Better Breath*” pada dashboard.
- b. Pilih salah satu dari 2 opsi menit.
- c. Tekan tombol “Mulai”.
- d. Ikuti intruksi pernapasan pada grafik.

Manfaat dari relaksasi pernapasan meliputi:

1. Mengurangi Kecemasan dan Stres: Teknik pernapasan dalam menghasilkan respons relaksasi tubuh, menurunkan kadar hormon stres seperti kortisol. Ini membantu menenangkan pikiran yang gelisah dan mengurangi gejala fisik kecemasan seperti jantung berdebar dan tangan berkeringat.
2. Meningkatkan Fokus dan Konsentrasi: Dengan mempersiapkan aliran oksigen ke otak, relaksasi pernapasan dapat meningkatkan kejernihan mental dan kemampuan kognitif. Fokus terstruktur dapat membantu Anda fokus pada tugas-tugas penting dan meningkatkan produktivitas.
3. Meningkatkan Kualitas Tidur: Melakukan teknik pernapasan sebelum tidur dapat membantu menenangkan sistem saraf dan mempersiapkan tubuh untuk istirahat. Ini dapat mengurangi insomnia dan meningkatkan kualitas tidur yang lebih dalam dan istirahat.
4. Membantu Mengatasi Emosi Negatif: Pernapasan yang terkontrol dapat membantu Anda mengelola emosi yang intens seperti kemarahan, kesedihan, atau frustrasi. Dengan memberikan Anda dan ruang untuk bernapas, Anda dapat merespons situasi dengan lebih tenang dan bijaksana.
5. Meningkatkan Kesehatan Fisik Secara Keseluruhan: Praktik pernapasan yang teratur dapat membantu meningkatkan fungsi paru-paru, memperkuat sistem kekebalan tubuh, dan berkontribusi membantu dalam manajemen nyeri kronis. Ini juga dapat meningkatkan energi dan vitalitas secara keseluruhan.

ⓘ Jika ada peringatan, pastikan kamu sudah duduk atau berbaring dengan nyaman.

Silahkan pilih salah satu opsi dibawah ini

1,5 menit

3 menit

[Mulai](#)

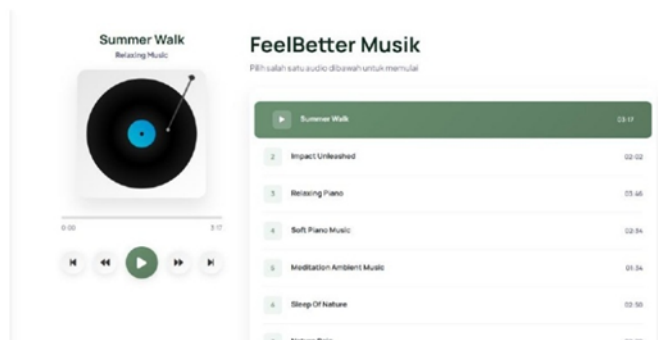


Gambar 8. Tampilan halaman *Better Breath*

6. *Better Music*

Dalam fitur *Better Music*, terdapat beberapa musik ketenangan untuk membantu mengurangi stress dan menciptakan perasaan bahagia. Berikut cara mengaksesnya :

- a. Tekan *card* “*Better Music*” pada *dashboard*.
- b. Pilih musik yang Anda inginkan.



Gambar 9. Tampilan halaman *Better Music*

7. *Better Affirmation*

Dalam fitur *Better Affirmation*, terdapat beberapa suara afirmasi positif untuk membantu anda menciptakan sikap yang lebih optimis dan meningkatkan kepercayaan diri dalam menghadapi stres. Berikut cara mengaksesnya :

- Tekan card “*Feel Affirmation*” pada *dashboard*.
- Pilih afirmasi yang Anda inginkan.



Gambar 10. Tampilan Halaman *Affirmation*

8. *Rekap*

Anda dapat melihat riwayat penggunaan beberapa fitur yang telah Anda lakukan, seperti *Better Journal* yang dilengkapi dengan analisis *Wordcloud*, *Better Test*, dan *Better Breath* untuk membantu Anda memantau progres kesehatan mentalmu dari waktu ke waktu. Berikut cara mengaksesnya :

- Tekan tombol “Lihat Rekap” pada *dashboard*.
- Pada tabel riwayat jurnal Anda dapat mengedit jika menginginkan pada tombol di kolom aksi. Selain tabel riwayat jurnal Anda hanya dapat menghapus.

Riwayat Jurnal Harian

motor

ayah

ibu

kalah

memberi

hari

sepeda

senang

hadiah

spesial

ulang

tahun

show

new

entries

search

#

Tanggal

isi

action

1

16/10/2025

Hari ini saya sangat senang, karena bu, saya mendapatkan yang sangat spesial di hari ulang tahun ...

✕

🔍

Previous

1

Next

Riwayat Tes Depresi

show

new

entries

search

#

Tanggal

Skor

Interpretasi

action

1

16/10/2025

14.000000

Depresi ringan

✕

🔍

Previous

1

Next

Riwayat Relaksasi Pernapasan

show

new

entries

search

#

Tanggal

Durasi

Siklus Pernapasan

Status

action

1

16/10/2025

1.0 menit

4 / 4

Selesai

✕

🔍

Previous

1

Next

← Kembali

Gambar 11. Tampilan halaman Rekap

D-13

Dokumen Teknikal



SISTEM PERAWATAN KESEHATAN MENTAL SECARA MANDIRI

Rafael Rinta Primadani
Dwi Intan Af'idah, S.T., M.Kom.
Sharfina Febbi Handayani, S.Kom., M.Kom.

DAFTAR ISI

DAFTAR ISI	2
1. Konfigurasi dan Inisialisasi Aplikasi	3
2. <i>Route Landing Page</i>	4
3. Sistem Autentifikasi	5
4. <i>Dashboard</i>	9
5. Fitur <i>Journaling</i>	9
6. Fitur Tes Depresi	13
7. Fitur Relaksasi Pernapasan	16
8. Fitur Musik Ketenangan.....	19
9. Fitur Suara Afirmasi Positif	20
10. Fitur Rekap Riwayat	20

1. Konfigurasi dan Inisialisasi Aplikasi

Feelbetter dibuat menggunakan *framework* Flask, yang merupakan *framework* web berbasis Python. Digunakan sebagai *back-end* untuk melakukan *login*, *register*, menampilkan *landing page*, menampilkan fitur-fitur dari aplikasi Feelbetter.

a. *Imporr Library*

```
from flask import Flask, render_template, request, redirect, url_for, flash, session, jsonify
from flask_mysql import MySQL
import MySQLdb.cursors
from werkzeug.security import generate_password_hash, check_password_hash
import re
import os
```

Beberapa *library* yang diimpor dalam kode ini meliputi:

- Flask: Untuk membuat aplikasi web dan menangani routing.
- render_template: Untuk merender file HTML sebagai respons.
- request: Untuk menangani data yang dikirimkan melalui *form*.
- redirect dan url_for: Untuk mengarahkan pengguna ke halaman lain.
- flash: Untuk menampilkan pesan kepada pengguna.
- session: Untuk menyimpan data sesi pengguna.
- jsonify: Untuk mengembalikan data dalam format JSON.
- flask_mysql: Untuk menghubungkan aplikasi dengan database MySQL
- MySQLdb.cursors: Untuk menggunakan *cursor* dalam *query database*.
- werkzeug.security: Untuk meng-hash dan memverifikasi password,
- re: Untuk melakukan validasi menggunakan *regular expressions*.
- os: Untuk mengakses fungsi sistem operasi, seperti menghasilkan *secret key*.

b. Konfigurasi Aplikasi

```
app = Flask(__name__)

# Konfigurasi secret key untuk sesi dan flash messages
app.secret_key = os.urandom(24)
```

- Inisialisasi Aplikasi: Baris ini membuat instance dari aplikasi Flask.
- Secret Key: `os.urandom(24)` digunakan untuk menghasilkan secret key yang acak dan aman. Secret key ini penting untuk mengamankan sesi pengguna dan flash messages, sehingga tidak dapat dipalsukan oleh pihak ketiga.

c. Konfigurasi Database

```
# Konfigurasi database MySQL dari environment variables
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ''
app.config['MYSQL_DB'] = 'feelbetter'

# Inisialisasi koneksi mysql
mysql = MySQL(app)
```

- `MYSQL_HOST`: Alamat host database, dalam hal ini adalah *localhost*.
- `MYSQL_USER`: Nama pengguna untuk mengakses database, di sini menggunakan *root*.
- `MYSQL_PASSWORD`: Password untuk pengguna database, di sini dibiarkan kosong (default).
- `MYSQL_DB`: Nama database yang digunakan, yaitu *feelbetter*.
- Inisialisasi Koneksi: `mysql = MySQL(app)` menginisialisasi koneksi ke database MySQL dengan menggunakan konfigurasi yang telah ditentukan.

2. Route Landing Page

a. Route Home

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

Route ini menangani permintaan ke halaman awal aplikasi. Ketika pengguna mengakses *root* URL (*/*), fungsi ini akan merender template *home.html*, yang berisi informasi tentang aplikasi dan tombol untuk memulai. Halaman ini berfungsi sebagai landing page yang memperkenalkan pengguna pada fitur-fitur yang tersedia.

3. Sistem Autentifikasi

a. Route Login

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    # Jika pengguna sudah login, redirect ke dashboard
    if 'logged_in' in session:
        return redirect(url_for('dashboard'))

    # Jika metodenya POST (form submission)
    if request.method == 'POST':
        # Ambil data dari form
        email = request.form['email']
        password = request.form['password']

        # Periksa apakah user ada di database
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute('SELECT * FROM users WHERE email = %s', (email,))
        user = cursor.fetchone()

        # Jika user ditemukan dan password benar
        if user and check_password_hash(user['password'], password):
            # Buat sesi baru
            session['logged_in'] = True
            session['id'] = user['id']
            session['name'] = user['name']
            session['email'] = user['email']

            # Redirect ke dashboard
            return redirect(url_for('dashboard'))
        else:
            # Jika login gagal
            flash('Email atau Password salah!', 'danger')

    # Jika metodenya GET, tampilkan halaman login
    return render_template('login.html')
```

- Pemeriksaan Sesi: Jika pengguna sudah login (terdapat *loggedin* dalam session), pengguna akan diarahkan ke halaman dashboard.
- Metode POST: Jika metode yang digunakan adalah POST (artinya pengguna mengirimkan form login), aplikasi akan mengambil email dan password dari form.
- Query Database: Menggunakan *cursor* untuk mengeksekusi *query* yang mencari pengguna berdasarkan email. Jika pengguna ditemukan, password yang di-hash akan diverifikasi menggunakan `check_password_hash()`.
- Membuat Sesi: Jika email dan password valid, aplikasi akan membuat sesi baru dengan menyimpan informasi pengguna (id, nama, email) dalam session.
- Pesan Error: Jika login gagal, aplikasi akan menampilkan pesan error menggunakan *flash*.
- Render Halaman Login: Jika metode GET, aplikasi akan merender halaman login.

b. *Route Register*

```

@app.route('/register', methods=['GET', 'POST'])
def register():
    # Jika pengguna sudah login, redirect ke dashboard
    if 'logged_in' in session:
        return redirect(url_for('dashboard'))

    if request.method == 'POST':
        # Ambil data dari form
        name = request.form['name']
        email = request.form['email']
        password = request.form['password']

        # Validasi input
        if not name or not email or not password:
            flash('Harap isi semua field!', 'danger')
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            flash('Email tidak valid!', 'danger')
        else:
            # Periksa apakah email sudah terdaftar
            cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
            cursor.execute('SELECT * FROM users WHERE email = %s', (email,))
            existing_user = cursor.fetchone()

            if existing_user:
                flash('Email sudah terdaftar!', 'danger')
            else:
                # Hash password untuk keamanan
                hashed_password = generate_password_hash(password)

                # Masukkan data user baru ke database
                cursor.execute('INSERT INTO users (name, email, password) VALUES (%s, %s, %s)',
                               (name, email, hashed_password))
                mysql.connection.commit()

                flash('Pendaftaran berhasil! Silakan login.', 'success')
                return redirect(url_for('login'))

    # Jika metodenya GET, tampilkan halaman pendaftaran
    return render_template('register.html')

```

- Pemeriksaan Sesi: Jika pengguna sudah login, mereka akan diarahkan ke dashboard.
- Metode POST: Jika metode yang digunakan adalah POST, aplikasi akan mengambil nama, email, dan password dari form pendaftaran.
- Validasi Input: Memastikan semua *field* terisi. Jika ada yang kosong, aplikasi akan menampilkan pesan error. Kemudian memvalidasi format email menggunakan regex. Jika format tidak valid, aplikasi akan menampilkan pesan error.
- Pemeriksaan Duplikasi Email: Menggunakan *cursor* untuk memeriksa apakah email sudah terdaftar di database. Jika sudah ada, aplikasi akan menampilkan pesan error.

- Hash Password: Jika email belum terdaftar, password akan di-hash menggunakan `generate_password_hash()` untuk keamanan.
- Insert Data: Data pengguna baru akan dimasukkan ke dalam tabel *users* di database.
- Pesan Sukses: Setelah berhasil, aplikasi akan menampilkan pesan sukses dan mengarahkan pengguna ke halaman login.
- Render Halaman Pendaftaran: Jika metode GET, aplikasi akan merender halaman pendaftaran.

c. *Route Logout*

```
@app.route('/logout')
def logout():
    # Hapus semua data sesi
    session.pop('logged_in', None)
    session.pop('id', None)
    session.pop('name', None)
    session.pop('email', None)

    # Redirect ke halaman login
    return redirect(url_for('login'))
```

- Menghapus Data Sesi: Menggunakan `session.pop()` untuk menghapus semua data sesi yang terkait dengan pengguna, termasuk status login, id, nama, dan email.
- Redirect ke Halaman Login: Setelah logout, pengguna akan diarahkan kembali ke halaman login.

4. *Dashboard*

```
@app.route('/dashboard')
def dashboard():
    # Periksa apakah pengguna sudah login
    if 'logged_in' in session:
        return render_template('dashboard.html', user=session)

    # Jika belum login, redirect ke halaman login
    return redirect(url_for('login'))
```

- Pemeriksaan Sesi: Jika pengguna sudah login, aplikasi akan merender halaman dashboard.html dan mengirimkan data pengguna melalui parameter *user*.
- Redirect ke Halaman Login: Jika pengguna belum login, mereka akan diarahkan ke halaman login.

5. *Fitur Journaling*

a. *Route Journal*

```
@app.route('/journal')
def journal():
    # Periksa apakah pengguna sudah login
    if 'logged_in' in session:
        return render_template('journal.html', user=session)

    # Jika belum login, redirect ke halaman login
    return redirect(url_for('login'))
```

- Pemeriksaan Sesi: Jika pengguna sudah login, aplikasi akan merender halaman journal.html dan mengirimkan data pengguna.
- Redirect ke Halaman Login: Jika pengguna belum login, mereka akan diarahkan ke halaman login.

b. *Route Journal Entry*

```

@app.route('/journal/entry', methods=['GET', 'POST'])
def journal_entry():
    # Periksa apakah pengguna sudah login
    if 'loggedin' not in session:
        return redirect(url_for('login'))

    # Format tanggal hari ini dalam bahasa Indonesia
    from datetime import datetime
    import locale

    # Mencoba set locale ke Indonesia
    try:
        locale.setlocale(locale.LC_TIME, 'id_ID.UTF-8')
    except:
        pass # Jika gagal, gunakan default locale

    current_date = datetime.now().strftime('%d %B %Y')

    # Jika metode POST, proses penyimpanan jurnal
    if request.method == 'POST':
        journal_content = request.form.get('journal_content')

        # Simpan ke database (kode ini perlu disesuaikan dengan struktur database Anda)
        if journal_content and journal_content.strip():
            cursor = mysql.connection.cursor()
            cursor.execute(
                'INSERT INTO journals (user_id, content, created_at) VALUES (%s, %s, NOW())',
                (session['id'], journal_content)
            )
            mysql.connection.commit()
            flash('Jurnal berhasil disimpan', 'success')
            return redirect(url_for('journal'))

    # Tampilkan halaman isi jurnal
    return render_template('journal_entry.html', user=session, current_date=current_date)

```

- Pemeriksaan Sesi: Jika pengguna belum login, pengguna akan diarahkan ke halaman login.
- Format Tanggal: Menggunakan *datetime* untuk mendapatkan tanggal saat ini dan *locale* untuk mengatur format tanggal ke bahasa Indonesia.
- Metode POST: Jika metode yang digunakan adalah POST, aplikasi akan mengambil konten jurnal dari *form*.
- Validasi Konten: Memastikan konten jurnal tidak kosong sebelum disimpan. Jika valid, konten jurnal akan disimpan ke dalam tabel *journals* di database dengan *timestamp* saat ini.
- Pesan Sukses: Setelah berhasil menyimpan, aplikasi akan menampilkan pesan sukses dan mengarahkan pengguna kembali ke halaman jurnal.

- Render Halaman Entri Jurnal: Jika metode GET, aplikasi akan merender halaman `journal_entry.html` dan menampilkan tanggal saat ini.

c. *Route Edit Journal.*

```
# Fungsi untuk mengedit jurnal
@app.route('/journal/edit/<int:id>', methods=['GET', 'POST'])
def edit_journal(id):
    # Periksa apakah pengguna sudah login
    if 'logged_in' not in session:
        return redirect(url_for('login'))

    # Ambil data jurnal yang akan diedit
    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    cursor.execute('SELECT * FROM journals WHERE id = %s AND user_id = %s', (id, session['id']))
    journal = cursor.fetchone()

    # Jika jurnal tidak ditemukan atau bukan milik user yang login
    if not journal:
        flash('Jurnal tidak ditemukan atau Anda tidak memiliki akses', 'danger')
        return redirect(url_for('recap'))

    # Format tanggal untuk ditampilkan
    from datetime import datetime
    import locale

    # Mencoba set locale ke Indonesia
    try:
        locale.setlocale(locale.LC_TIME, 'id_ID.UTF-8')
    except:
        pass # Jika gagal, gunakan default locale

    journal_date = journal['created_at'].strftime('%d %B %Y')

    # Jika metode POST, proses update jurnal
    if request.method == 'POST':
        journal_content = request.form.get('journal_content')

        # Update jurnal jika content tidak kosong
        if journal_content and journal_content.strip():
            cursor = mysql.connection.cursor()
            cursor.execute(
                'UPDATE journals SET content = %s WHERE id = %s AND user_id = %s',
                (journal_content, id, session['id'])
            )
            mysql.connection.commit()
            flash('Jurnal berhasil diperbarui', 'success')
            return redirect(url_for('recap'))

    # Tampilkan halaman edit jurnal
    return render_template("journal_edit.html", user=session, journal=journal, journal_date=journal_date)
```

- Pemeriksaan Sesi: Jika pengguna belum login, mereka akan diarahkan ke halaman login.
- Query Database: Mengambil data jurnal berdasarkan ID dan memastikan bahwa jurnal tersebut milik pengguna yang sedang login. Jika jurnal tidak ditemukan, aplikasi akan menampilkan pesan error dan mengarahkan pengguna kembali ke halaman recap.
- Format Tanggal: Menggunakan *locale* untuk mengatur format tanggal ke bahasa Indonesia.

- Metode POST: Jika metode yang digunakan adalah POST, aplikasi akan mengambil konten jurnal yang baru dari *form*.
- Validasi Konten: Memastikan konten jurnal tidak kosong sebelum diperbarui. Jika valid, konten jurnal akan diperbarui di database.
- Pesan Sukses: Setelah berhasil memperbarui, aplikasi akan menampilkan pesan sukses dan mengarahkan pengguna kembali ke halaman rekap.
- Render Halaman Edit Jurnal: Jika metode GET, aplikasi akan merender halaman `journal_edit.html` dan menampilkan data jurnal yang akan diedit.

d. *Route delete Journal*

```
# Fungsi untuk menghapus jurnal
@app.route('/journal/delete', methods=['POST'])
def delete_journal():
    # Periksa apakah pengguna sudah login
    if 'loggedin' not in session:
        return redirect(url_for('login'))

    # Ambil ID jurnal yang akan dihapus
    journal_id = request.form.get('journal_id')

    if journal_id:
        try:
            # Hapus jurnal dari database
            cursor = mysql.connection.cursor()
            cursor.execute('DELETE FROM journals WHERE id = %s AND user_id = %s', (journal_id, session['id']))
            mysql.connection.commit()

            if cursor.rowcount > 0:
                flash('Jurnal berhasil dihapus', 'success')
            else:
                flash('Jurnal tidak ditemukan atau Anda tidak memiliki akses', 'danger')
        except Exception as e:
            flash(f'Terjadi kesalahan: {str(e)}', 'danger')

    return redirect(url_for('recap'))
```

- Pemeriksaan Sesi: Jika pengguna belum login, mereka akan diarahkan ke halaman login.
- Mengambil ID Jurnal: ID jurnal yang akan dihapus diambil dari *form*.
- Query Database: Menggunakan *cursor* untuk mengeksekusi *query* penghapusan berdasarkan ID dan memastikan bahwa jurnal tersebut milik pengguna yang sedang login.

- Pesan Sukses atau Error: Jika jurnal berhasil dihapus, aplikasi akan menampilkan pesan sukses. Jika jurnal tidak ditemukan atau pengguna tidak memiliki akses, aplikasi akan menampilkan pesan error.
- Error Handling: Menggunakan try-except untuk menangani kemungkinan kesalahan saat menghapus jurnal.
- Redirect ke Halaman Rekap: Setelah proses penghapusan, pengguna akan diarahkan kembali ke halaman rekap.

e. *API Get Journals*

```
@app.route('/api/journals')
def get_journals_api():
    # Periksa apakah pengguna sudah login
    if 'logged_in' not in session:
        return jsonify({'error': 'Unauthorized'}), 401

    # Ambil data jurnal dari database untuk user yang sedang login
    cursor = mysql.connection.cursor(mysql.connector.cursor.DictCursor)
    cursor.execute('SELECT id, content, created_at FROM journals WHERE user_id = %s ORDER BY created_at DESC', (session['id'],))
    journals = cursor.fetchall()

    # Konversi objek datetime menjadi string agar bisa di-JSON-kan
    for journal in journals:
        if 'created_at' in journal and journal['created_at']:
            journal['created_at'] = journal['created_at'].strftime('%Y-%m-%d %H:%M:%S')

    return jsonify(journals)
```

- Pemeriksaan Sesi: Jika pengguna belum login, aplikasi akan mengembalikan response JSON dengan status 401 (Unauthorized).
- Query Database: Mengambil semua entri jurnal milik pengguna yang sedang login, diurutkan berdasarkan tanggal pembuatan.
- Format Tanggal: Mengonversi objek *datetime* menjadi *string* agar dapat dikembalikan dalam format JSON.
- Response JSON: Mengembalikan data jurnal dalam format JSON.

6. Fitur Tes Depresi

a. Fitur Tes Depresi

```

@app.route('/tes', methods=['GET', 'POST'])
def tes_depresi():
    # Periksa apakah pengguna sudah login
    if 'loggedin' not in session:
        return redirect(url_for('login'))

    # Jika metode POST, proses penyimpanan hasil tes
    if request.method == 'POST':
        # Ambil data dari form
        total_score = request.form.get('total_score')
        interpretation = request.form.get('interpretation')

        # Simpan hasil tes ke database
        if total_score and interpretation:
            cursor = mysql.connection.cursor()
            cursor.execute(
                'INSERT INTO depression_tests (user_id, score, interpretation, test_date) VALUES (%s, %s, %s, NOW())',
                (session['id'], total_score, interpretation)
            )
            mysql.connection.commit()
            flash('Hasil tes berhasil disimpan', 'success')

        # Kembali ke halaman tes depresi
        return redirect(url_for('tes_depresi'))

    # Tampilkan halaman tes depresi
    return render_template('tes_depresi.html', user=session)

```

- Pemeriksaan Sesi: Jika pengguna belum login, mereka akan diarahkan ke halaman login.
- Metode POST: Jika metode yang digunakan adalah POST, aplikasi akan mengambil total score dan interpretasi dari form yang diisi oleh pengguna.
- Validasi Input: Memastikan bahwa total *score* dan interpretasi tidak kosong sebelum disimpan. Jika valid, aplikasi akan menyimpan hasil tes ke dalam tabel *depression_tests* di database dengan *timestamp* saat ini.
- Pesan Sukses: Setelah berhasil menyimpan hasil tes, aplikasi akan menampilkan pesan sukses dan mengarahkan pengguna kembali ke halaman tes depresi.
- Render Halaman Tes Depresi: Jika metode GET, aplikasi akan merender halaman *tes_depresi.html* dan menampilkan data pengguna.

b. API *Get* Tes Depresi.

```

@app.route('/api/depression-tests')
def get_depression_tests_api():
    # Periksa apakah pengguna sudah login
    if 'loggedin' not in session:
        return jsonify({'error': 'Unauthorized'}), 401

    # Ambil data tes depresi dari database untuk user yang sedang login
    cursor = mysql.connection.cursor(mysql.connector.cursor.DictCursor)
    cursor.execute('SELECT id, score, interpretation, test_date FROM depression_tests WHERE user_id = %s ORDER BY test_date DESC', (session['id'],))
    tests = cursor.fetchall()

    # Konversi objek datetime menjadi string agar bisa di JSON kan
    for test in tests:
        if 'test_date' in test and test['test_date']:
            test['test_date'] = test['test_date'].strftime('%Y-%m-%d %H:%M:%S')

    return jsonify(tests)

```

- **Pemeriksaan Sesi:** Jika pengguna belum login, aplikasi akan mengembalikan *response* JSON dengan status 401 (Unauthorized).
- **Query Database:** Mengambil semua hasil tes depresi milik pengguna yang sedang login, diurutkan berdasarkan tanggal tes.
- **Format Tanggal:** Mengonversi objek *datetime* menjadi *string* agar dapat dikembalikan dalam format JSON.
- **Response JSON:** Mengembalikan data hasil tes dalam format JSON.

c. *Route delete* Tes Depresi

```

@app.route('/depression-test/delete', methods=['POST'])
def delete_depression_test():
    # Periksa apakah pengguna sudah login
    if 'loggedin' not in session:
        return redirect(url_for('login'))

    # Ambil ID tes yang akan dihapus
    test_id = request.form.get('test_id')

    if test_id:
        try:
            # Hapus tes dari database
            cursor = mysql.connection.cursor()
            cursor.execute('DELETE FROM depression_tests WHERE id = %s AND user_id = %s', (test_id, session['id']))
            mysql.connection.commit()

            if cursor.rowcount > 0:
                flash('Hasil tes berhasil dihapus', 'success')
            else:
                flash('Hasil tes tidak ditemukan atau Anda tidak memiliki akses', 'danger')
        except Exception as e:
            flash(f'Terjadi kesalahan: {str(e)}', 'danger')

    return redirect(url_for('recap'))

```

- **Pemeriksaan Sesi:** Jika pengguna belum login, mereka akan diarahkan ke halaman login.
- **Mengambil ID Tes:** ID tes yang akan dihapus diambil dari *form*.

- Query Database: Menggunakan *cursor* untuk mengeksekusi *query* penghapusan berdasarkan ID dan memastikan bahwa tes tersebut milik pengguna yang sedang login.
- Pesan Sukses atau Error: Jika tes berhasil dihapus, aplikasi akan menampilkan pesan sukses. Jika tes tidak ditemukan atau pengguna tidak memiliki akses, aplikasi akan menampilkan pesan error.
- Error Handling: Menggunakan try-except untuk menangani kemungkinan kesalahan saat menghapus tes.
- Redirect ke Halaman Rekap: Setelah proses penghapusan, pengguna akan diarahkan kembali ke halaman rekap.

7. Fitur Relaksasi Pernapasan

a. Route Breath

```
@app.route('/breath')
def breath():
    # Periksa apakah pengguna sudah login
    if 'logged_in' in session:
        return render_template('breath.html', user=session)

    # Jika belum login, redirect ke halaman login
    return redirect(url_for('login'))
```

- Pemeriksaan Sesi: Jika pengguna sudah *login*, aplikasi akan merender halaman *breath.html* dan mengirimkan data pengguna.
- Redirect ke Halaman Login: Jika pengguna belum *login*, mereka akan diarahkan ke halaman login.

b. Route Breath Exercise

```

@app.route('/breathing-exercise/save', methods=['POST'])
def save_breathing_exercise():
    # Periksa apakah pengguna sudah login
    if 'loggedin' not in session:
        return jsonify({'error': 'Unauthorized'}), 401

    # Ambil data dari request
    data = request.get_json()
    duration = data.get('duration')
    cycles_completed = data.get('cycles_completed')
    total_cycles = data.get('total_cycles')
    status = data.get('status')

    # Simpan ke database
    if duration and cycles_completed is not None and total_cycles and status:
        cursor = mysql.connection.cursor()
        cursor.execute(
            'INSERT INTO breathing_exercises (user_id, duration, cycles_completed, total_cycles, status, exercise_date) VALUES (%s, %s, %s, %s, %s, NOW())'
            % (session['id'], duration, cycles_completed, total_cycles, status)
        )
        mysql.connection.commit()
        return jsonify({'success': True, 'message': 'Latihan pernapasan berhasil disimpan'})
    return jsonify({'success': False, 'message': 'Data tidak lengkap'}), 400

```

- Pemeriksaan Sesi: Jika pengguna belum login, aplikasi akan mengembalikan response JSON dengan status 401 (Unauthorized).
- Mengambil Data JSON: Data latihan pernapasan diambil dari *request* JSON, termasuk durasi, siklus yang diselesaikan, total siklus, dan status.
- Validasi Input: Memastikan semua data yang diperlukan terisi sebelum disimpan. Jika valid, data akan disimpan ke dalam tabel *breathing_exercises* di database.
- Response JSON: Mengembalikan *response* JSON yang menunjukkan keberhasilan penyimpanan atau kesalahan jika data tidak lengkap.

e. *Route delete Breathing Exercise*

```

@app.route('/breathing-exercise/delete', methods=['POST'])
def delete_breathing_exercise():
    # Periksa apakah pengguna sudah login
    if 'loggedin' not in session:
        return redirect(url_for('login'))

    # Ambil ID exercise yang akan dihapus
    exercise_id = request.form.get('exercise_id')

    if exercise_id:
        try:
            # Hapus exercise dari database
            cursor = mysql.connection.cursor()
            cursor.execute('DELETE FROM breathing_exercises WHERE id = %s AND user_id = %s', (exercise_id, session['id']))
            mysql.connection.commit()

            if cursor.rowcount > 0:
                flash('Riwayat latihan pernapasan berhasil dihapus', 'success')
            else:
                flash('Riwayat tidak ditemukan atau Anda tidak memiliki akses', 'danger')
        except Exception as e:
            flash(f'Terjadi kesalahan: {str(e)}', 'danger')

    return redirect(url_for('recap'))

```

- Pemeriksaan Sesi: Jika pengguna belum *login*, mereka akan diarahkan ke halaman *login*.

- Mengambil ID Exercise: ID latihan pernapasan yang akan dihapus diambil dari *form*.
- Query Database: Menggunakan *cursor* untuk mengeksekusi *query* penghapusan berdasarkan ID dan memastikan bahwa latihan tersebut milik pengguna yang sedang login.
- Pesan Sukses atau Error: Jika latihan berhasil dihapus, aplikasi akan menampilkan pesan sukses. Jika latihan tidak ditemukan atau pengguna tidak memiliki akses, aplikasi akan menampilkan pesan error.
- Redirect ke Halaman Rekap: Setelah proses penghapusan, pengguna akan diarahkan kembali ke halaman rekap.

8. Fitur Musik Ketenangan

```
@app.route('/music')
def music():
    # Periksa apakah pengguna sudah login
    if 'logged_in' in session:
        return render_template('music.html', user=session)

    # Jika belum login, redirect ke halaman login
    return redirect(url_for('login'))
```

- Pemeriksaan Sesi: Jika pengguna sudah *login*, aplikasi akan merender halaman *music.html* dan mengirimkan data pengguna.
- Redirect ke Halaman *Login*: Jika pengguna belum *login*, mereka akan diarahkan ke halaman *login*.

9. Fitur Suara Afirmasi Positif

```
@app.route('/affirmation')
def affirmation():
    # Periksa apakah pengguna sudah login
    if 'loggedin' in session:
        return render_template('affirmation.html', user=session)

    # Jika belum login, redirect ke halaman login
    return redirect(url_for('login'))
```

- Pemeriksaan Sesi: Jika pengguna sudah *login*, aplikasi akan merender halaman `affirmation.html` dan mengirimkan data pengguna.
- Redirect ke Halaman *Login*: Jika pengguna belum *login*, mereka akan diarahkan ke halaman *login*.

10. Fitur Rekap Riwayat

```
@app.route('/recap')
def recap():
    # Periksa apakah pengguna sudah login
    if 'loggedin' not in session:
        return redirect(url_for('login'))

    # Ambil data jurnal dari database untuk user yang sedang login
    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    cursor.execute('SELECT * FROM journals WHERE user_id = %s ORDER BY created_at DESC', (session['id'],))
    journals = cursor.fetchall()

    # Tampilkan halaman rekap jurnal
    return render_template('recap.html', user=session, journals=journals)
```

- Pemeriksaan Sesi: Jika pengguna belum *login*, mereka akan diarahkan ke halaman *login*.
- Query Database: Mengambil semua entri jurnal milik pengguna yang sedang *login*, diurutkan berdasarkan tanggal pembuatan.
- Render Halaman Rekap: Aplikasi akan merender halaman `recap.html` dan mengirimkan data pengguna serta daftar jurnal yang diambil dari database.

Lampiran 5. Sertifikat HKI

 REPUBLIK INDONESIA KEMENTERIAN HUKUM	
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	: EC002025077966, 30 Juni 2025
Pencipta	
Nama	: Rafael Rinta Primadani, Dwi Intan Afidah, S.T., M.Kom. dkk
Alamat	: Jalan Moh Toha, RT/RW 02/03, Kelurahan Kaligangsa, Margadana, Kota Tegal, Jawa Tengah, 52147
Kewarganegaraan	: Indonesia
Pemegang Hak Cipta	
Nama	: Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik Harapan Bersama
Alamat	: Jalan Mataram No. 9, Pesurungan Lor, Kecamatan Margadana, Margadana, Kota Tegal, Jawa Tengah, 52142
Kewarganegaraan	: Indonesia
Jenis Ciptaan	: Program Komputer
Judul Ciptaan	: Sistem Perawatan Kesehatan Mental Secara Mandiri Berbasis Website
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	: 30 Juni 2025, di Kota Tegal
Jangka waktu perlindungan	: Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor Pencatatan	: 000918227
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 DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL u.b Direktur Hak Cipta dan Desain Industri Agung Damarsasongko.SH..MH. NIP. 196912261994031001
	Disclaimer: 1. Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan. 2. Surat Pencatatan ini telah disegel secara elektronik menggunakan segel elektronik yang diterbitkan oleh Balai Besar Sertifikasi Elektronik, Badan Siber dan Sandi Negara. 3. Surat Pencatatan ini dapat dibuktikan keasliannya dengan memindai kode QR pada dokumen ini dan informasi akan ditampilkan dalam browser.

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Rafael Rinta Primadani	Jalan Moh Toha, RT/RW 02/03, Kelurahan Kaligangsa Margadana, Kota Tegal
2	Dwi Intan Afidah, S.T., M.Kom.	RT03/RW02, Desa Grinting Bulakamba, Kab. Brebes
3	Sharfina Febbi Handayani, S.Kom., M.Kom.	RT.002, RW.006, Desa Grobog Kulon Pangkah, Kab. Tegal



Lampiran 6. Pengujian *Selenium*

Hasil Pengujian *Register User*

The screenshot displays the Selenium IDE interface for a project named "FeelBetter". The test suite contains 11 steps, with the "Register User" step highlighted. The steps are as follows:

Step	Command	Target	Value
1	open	http://127.0.0.1:5000/	
2	set window size	786x964	
3	click	linkText=Mulai	
4	click	linkText=Daftar disini	
5	click	id=name	
6	type	id=name	etheroo
7	click	id=email	
8	type	id=email	etheroo@gmail.com
9	click	id=password	
10	type	id=password	masuk12345
11	send keys	id=password	\$(KEY_ENTER)

The "Log" tab at the bottom shows the execution results for steps 7 through 11, all of which completed successfully:

- 7. click on id=email OK 15:37:45
- 8. type on id=email with value etheroo@gmail.com OK 15:37:46
- 9. click on id=password OK 15:37:46
- 10. type on id=password with value masuk12345 OK 15:37:46
- 11. sendKeys on id=password with value \$(KEY_ENTER) OK 15:37:47

The final status is: "Register User" completed successfully 15:37:47.

Hasil Pengujian *Login User*

Selenium IDE - FeatBetter*

Project: **FeatBetter***

Tests +

Search tests...

http://127.0.0.1:5000/

	Command	Target	Value
✓ Fitur Better Affirmation*			
✓ Fitur Better Breath*	1 ✓ open	http://127.0.0.1:5000/	
✓ Fitur Better Journal*	2 ✓ set window size	788x984	
✓ Fitur Better Music*	3 ✓ click	linkText=Mulai	
✓ Fitur Better Test*	4 ✓ click	id=email	
✓ Fitur Rangkap*	5 ✓ type	id=email	etheroo@gmail.com
✓ Login User*	6 ✓ click	id=password	
	7 ✓ type	id=password	masuk12345
✓ Register User*	8 ✓ send keys	id=password	\$(KEY_ENTER)

Command #

Target

Value

Description

Log Reference

4. click on id=email OK	15:39:10
5. type on id=email with value etheroo@gmail.com OK	15:39:10
6. click on id=password OK	15:39:10
7. type on id=password with value masuk12345 OK	15:39:11
8. sendKeys on id=password with value \$(KEY_ENTER) OK	15:39:11
Login User completed successfully	15:39:12

Hasil Pengujian Fitur *Better Journal*

The screenshot displays the Selenium IDE interface for a project named 'FeetBetter'. The 'Tests' panel on the left lists several test cases, with 'Fitur Better Journal*' selected. The main area shows a table of test steps with columns for Command, Target, and Value.

Command	Target	Value
1. ✓ open	http://127.0.0.1:5000/dashboard	
2. ✓ set window size	788x984	
3. ✓ click	css= row:nth-child(1) > col-md-4:nth-child(1) h2	
4. ✓ click	linkText=Selanjutnya →	
5. ✓ click	name=journal_content	
6. ✓ type	name=journal_content	Hari ini saya jogging pagi hari, suasana sangat ramai, saya sangat senang
7. ✓ click	css= btn-journal-submit	

Below the table, there are input fields for Command, Target, Value, and Description. The 'Log' panel at the bottom shows the execution results of the test steps, indicating that the test 'Fitur Better Journal' completed successfully.

Log Reference

- 3. click on css= row:nth-child(1) > col-md-4:nth-child(1) h2 OK 15:41:01
- 4. click on linkText=Selanjutnya → OK 15:41:02
- 5. click on name=journal_content OK 15:41:02
- 6. type on name=journal_content with value Hari ini saya jogging pagi hari, suasana sangat ramai, saya sangat senang OK 15:41:03
- 7. click on css= btn-journal-submit OK 15:41:03

'Fitur Better Journal' completed successfully 15:41:03

Hasil Pengujian Fitur *Better Test*

The screenshot displays the Selenium IDE interface for a project named 'Fitur Better'. The 'Tests' pane on the left lists several test cases, with 'Fitur Better Test' selected. The main area shows a table of test steps with columns for Command, Target, and Value. The steps include setting window size, clicking various elements, and submitting a form. The 'Log' pane at the bottom shows the execution results, indicating that the test completed successfully at 15:42:07.

Command	Target	Value
set window size		786x964
click	css= row:nth-child(1) > col-md-4:nth-child(2) .feature-icon	
click	css= phq9-question:nth-child(3) .phq9-option:nth-child(2) > label	
click	css= phq9-question:nth-child(4) .phq9-option:nth-child(1) > label	
click	css= phq9-question:nth-child(5) .phq9-option:nth-child(2) > label	
click	css= phq9-question:nth-child(6) .phq9-option:nth-child(3) > label	
click	css= phq9-question:nth-child(7) .phq9-option:nth-child(2) > label	
click	css= phq9-question:nth-child(8) .phq9-option:nth-child(3) > label	
click	css= phq9-question:nth-child(9) .phq9-option:nth-child(2) > label	
click	css= phq9-question:nth-child(10) .phq9-option:nth-child(2) > label	
click	css= phq9-question:nth-child(11) .phq9-option:nth-child(2) > label	
click	id=submitBtn	

Log

Step	Command	Target	Value
9	click on css= phq9-question:nth-child(8) .phq9-option:nth-child(3) > label		OK
10	click on css= phq9-question:nth-child(9) .phq9-option:nth-child(3) > label		OK
11	click on css= phq9-question:nth-child(10) .phq9-option:nth-child(2) > label		OK
12	click on css= phq9-question:nth-child(11) .phq9-option:nth-child(2) > label		OK
13	click on id=submitBtn		OK

'Fitur Better Test' completed successfully

Hasil Pengujian Fitur *Better Breath*

The screenshot displays the Selenium IDE interface for a project named 'FeelBetter'. The 'Tests' panel on the left lists several test cases, with 'Fitur Better Breath*' selected. The main area shows a table of commands and their results for this test case.

Command	Target	Value
1. ✓ open	http://127.0.0.1:5000/dashboard	
2. ✓ set window size	788x864	
3. ✓ click	css= col-md-4:nth-child(3).feature-item	
4. ✓ click	id=duration-short	
5. ✓ click	id=start-breathing	
6. ✓ click	id=startExercise	

Below the table, there are input fields for 'Command', 'Target', 'Value', and 'Description'. The 'Log' panel at the bottom shows the execution details for each step, indicating that all steps passed successfully.

Log | Reference

Step	Description	Time
2.	setWindowSize on 788x864 OK	15:43:39
3.	click on css= col-md-4:nth-child(3).feature-item OK	15:43:39
4.	click on id=duration-short OK	15:43:40
5.	click on id=start-breathing OK	15:43:41
6.	click on id=startExercise OK	15:43:41
'Fitur Better Breath' completed successfully		15:43:42

Hasil Pengujian Fitur *Better Music*

The screenshot displays the Selenium IDE interface for a project named 'FeelBetter'. The 'Tests' pane on the left lists several test cases, with 'Fitur Better Music' selected and highlighted in blue. The main area shows the details of this test case, which consists of four steps:

Step	Command	Target	Value
1	open	http://127.0.0.1:5000/dashboard	
2	set window size	786x964	
3	click	css= mt-1 > col-md-4:nth-child(1) h2	
4	click	css= playlist-item:nth-child(2) > track-number	

Below the table, there are input fields for 'Command', 'Target', 'Value', and 'Description'. At the bottom, the 'Log' pane shows the execution results of the test case:

```
Running "Fitur Better Music"
1. open on http://127.0.0.1:5000/dashboard OK
2. setWindowSize on 786x964 OK
3. click on css= mt-1 > col-md-4:nth-child(1) h2 OK
4. click on css= playlist-item:nth-child(2) > track-number OK
"Fitur Better Music" completed successfully
```

Hasil Pengujian Fitur *Better Affirmation*

The screenshot displays the Selenium IDE interface for a project named 'FeelBetter'. The 'Tests' pane on the left lists several test cases, with 'Fitur Better Affirmation' selected. The main area shows a table of commands and their targets for this test case.

Command	Target	Value
1. ✓ open	http://127.0.0.1:5000/dashboard	
2. ✓ set window size	786x864	
3. ✓ click	css= mt-1 > col-md-4:nth-child(2) feature-item	
4. ✓ click	css= playlist-item:nth-child(4) > track-number	

Below the table, there are input fields for 'Command', 'Target', 'Value', and 'Description'. The 'Log' pane at the bottom shows the execution details of the 'Fitur Better Affirmation' test case, indicating it completed successfully at 15:45:32.

Running 'Fitur Better Affirmation'

- 1. open on http://127.0.0.1:5000/dashboard OK 15:45:29
- 2. setWindowSize on 786x864 OK 15:45:30
- 3. click on css= mt-1 > col-md-4:nth-child(2) feature-item OK 15:45:30
- 4. click on css= playlist-item:nth-child(4) > track-number OK 15:45:31

'Fitur Better Affirmation' completed successfully 15:45:32

Hasil Pengujian Fitur Rekap

The screenshot displays the Selenium IDE interface for a project named 'FeelBetter'. The 'Tests' panel on the left lists several test cases, with 'Fitur Rekap*' selected. The main area shows a table of commands and their targets for the selected test case.

Command	Target	Value
1. ✓ open	http://127.0.0.1:5000/dashboard	
2. ✓ set window size	786x864	
3. ✓ click	linkText=Lihat Rekap	
4. ✓ run script	window.scrollTo(0,63.20000076293945)	

Below the table, there are input fields for Command, Target, Value, and Description. The 'Log' panel at the bottom shows the execution results:

```
Running 'Fitur Rekap'
1. open on http://127.0.0.1:5000/dashboard OK
2. setWindowSize on 786x864 OK
3. click on linkText=Lihat Rekap OK
4. runScript on window.scrollTo(0,63.20000076293945) OK
'Fitur Rekap' completed successfully
```

Lampiran 7. Lembar Bimbingan



D IV TEKNIK INFORMATIKA POLITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Rafael Rinta Primadani
 NIM : 21090078
 No. Ponsel : 081227717133
 Judul TA : Sistem Perawatan Kesehatan Mental Secara Mandiri Berbasis Website
 Dosen Pembimbing I : Dwi Intan Af'idah, S.T., M.Kom.

No.	Tanggal	Pemeriksaan	Perbaikan yang Perlu Dilakukan	Paraf Pembimbing
1.	9/5 - 2025	Aplikasi	- Perbaiki konsep wordcloud - Selesaikan fitur PHAG	
2.	16/5 - 2025	Aplikasi	lanjut pengembangan fitur lainnya	
3.	23/5 - 2025	Aplikasi	Perbaiki fitur hambatkan nafas, fitur history hasil PHAG, dan lanjutkan fitur lainnya	

4.	2/6-2025	Aplikasi	- Buat kan fitur grafik untuk melakukan pernafasan - Sediakan agar saat menggunakan HP tetap informatif	
5.	13/6-2025	Aplikasi	- Coba di ponsel - Siapkan dokumen pengajuan hak cipta	
6.	24/6-2025	Aplikasi	- Aplikasi sudah siap - Siapkan hak cipta - Siapkan laporan	
7.	7/7-2025	laporan	Perbaiki sesuai Catatan	
8.	8/7-2025	laporan	perbaiki sesuai Catatan	

9.	9/7-2015	laporan	Acc	
----	----------	---------	-----	---

Tegal, 9 Juli 2015

Dosen Pembimbing I



Dwi Intan Af'idah, S.T., M.Kom.

NIPY. 11.020.470



D IV TEKNIK INFORMATIKA POLITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Rafael Rinta Primadani
NIM : 21090078
No. Ponsel : 081227717133
Judul TA : Sistem Perawatan Kesehatan Mental Secara Mandiri Berbasis Website
Dosen Pembimbing II : Sharfina Febbi Handayani, S.Kom., M.Kom.

No.	Tanggal	Pemeriksaan	Perbaikan yang Perlu Dilakukan	Paraf Pembimbing
1.	24 April 2025	Product Requirement Document (PRD)	ditagut untuk membuat kode implementasi aplikasi	Sharfina
2.	21 Mei 2025	Progres Aplikasi Feel Better	Ditambahkan rekomendasi kontak psikolog / ahli jika hasil tes PHQ-9 cukup berat / berat	Sharfina
3.	28 Mei 2025	Progres Aplikasi Feel Better	Ditambahkan Informasi Seputar contoh relaksi, afirmasi music	Sharfina
4.	4 Juni 2025	Progres Aplikasi Feel Better	→ untuk interpretasi cukup berat mohon rekomendasi music dan afirmasi dapat disesuaikan	Sharfina

5	25 Juni 2025	Manual Book	» manualbook telah disesuaikan dengan aplikasi » lanjut Daftar HKI	Spafy
6	8 Juli 2025	Laporan BAB I dan BAB II	» latar belakang ditambahkan P429 terkait tes depresi » testing cepat diperbaiki dengan menambahkan Salza Y	Spafy
7	9 Juli 2025	Laporan BAB II dan BAB III	» hasil testing sudah diteliti di lampiran	Spafy
8	10 Juli 2025	ACC Laporan dan Aplikasi		Spafy

--	--	--	--	--

Tegal, 10 Juli 2025
Dosen Pembimbing II



Sharfina Febbi Handayani, S.Kom., M.Kom.
NIPY. 08.020.451