

## LAMPIRAN

### A. SURAT KESEDIAAN MEMBIMBING

#### SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Saksono Bayu Ajie Sumantri  
NIM : 19098001  
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Sharfina Febbi Handayani, M.Kom  
Status : Dosen Tetap  
NIDN : 0617029201  
Jabatan Fungsional : Tenaga Pengajar  
Pangkat/Golongan : III/b

Pada hari ini Kamis tanggal 9 Maret 2023 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Skripsi Pihak Pertama dengan syarat Pihak Pertama Bersedia bimbingan secara rutin untuk menyelesaikan Skripsi Minimal 3 kali dalam sebulan, apabila dalam waktu 30 hari pertama setelah kesepakatan ini dibuat pihak pertama tidak konsisten untuk menyanggupi maka pihak kedua berhak membatalkan kesepakatan ini dan mempersilahkan pihak pertama untuk ganti dosen pembimbing.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

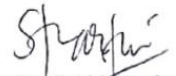
Tegal, 9 Maret 2023

Pihak Pertama



Saksono Bayu Ajie Sumantri  
NIM. 19098001

Pihak Kedua



Sharfina Febbi Handayani, M.Kom.  
NIPY. 08.020.451

Mengetahui,  
Ketua Program Studi Sarjana Terapan Teknik Informatika



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

## SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Saksono Bayu Ajie Sumantri  
NIM : 19098001  
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Dega Suroño Wibowo  
Status : Dosen  
NIDN : 0607108202  
Jabatan Fungsional : Lektor  
Pangkat/Golongan : III/c

Pada hari ini Kamis tanggal 9 Maret 2023 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan, dengan syarat melakukan bimbingan 1 kali dalam seminggu / setidaknya 3 kali bimbingan dalam 1 bulannya (bimbingan dengan progress) apabila saya tidak memenuhi syarat tersebut pada poin diatas maka saya tidak berhak mendapatkan surat rekomendasi mengikuti ujian sidang skripsi dan saya berjanji akan memenuhi persyaratan tersebut dan menyelesaikan tepat waktu. Adapun waktu dan tempat pelaksanaan disepakati antar pihak. Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

Tegal, 9 Maret 2023

Pihak Pertama



Saksono Bayu Ajie Sumantri

Pihak Kedua



Dega Suroño Wibowo, M.Kom.  
NIPY. 06.014.183

Mengetahui  
Ketua Program Studi Sarjana Terapan Teknik Informatika



Slaimer Wiyono, S.Pd., M.Eng.  
NIPY: 08.015.222

## B. SURAT PERNYATAAN PENGAJUAN HKI

### SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta :

1. Nama : Saksono Bayu Ajie Sumantri  
Kewarganegaraan : Indonesia  
Alamat : Dsn. Mranak RT 04 RW 09 Kelurahan Wonorejo, Kecamatan Pringapus, Kabupaten Semarang, Provinsi Jawa Tengah, Kode Pos 50212.
2. Nama : Dega Surono Wibowo, S.T., M.Kom.  
Kewarganegaraan : Indonesia  
Alamat : Perumahan Sapphire Regency Block H No. 1 RT 04 RW 01 Kelurahan Pulosari, Kecamatan Brebes, Provinsi Jawa Tengah, Kode Pos 52213
3. Nama : Sharfina Febbi Handayani, M.Kom.  
Kewarganegaraan : Indonesia  
Alamat : Jalan Brayon 1 Desa Grobog Kulon RT 02 RW 06 Kecamatan Pangkah, Kabupaten Tegal, Provinsi Jawa Tengah, Kode Pos 52471

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:  
Berupa : Program Komputer  
Berjudul : Aplikasi Tracking Activity untuk Fasting, Drink, dan Step berbasis Android
  - 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 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, 29 Februari 2024



Saksono Bayu Ajie Sumantri  
Pemegang Hak Cipta\*

Dega Surono Wibowo, S.T., M.Kom.  
Pemegang Hak Cipta\*

Sharfina Febbi Handayani, M.Kom.  
Pemegang Hak Cipta\*

Pemegang Hak Cipta\*

\* Semua pemegang hak cipta agar menandatangani di atas materai.

## C. SURAT PENGALIHAN HKI

### SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

1. Nama : Saksono Bayu Ajie Sumantri  
Kewarganegaraan : Indonesia  
Alamat : Dsn. Mranak RT 04 RW 09 Kelurahan Wonorejo, Kecamatan Pringapus, Kabupaten Semarang, Provinsi Jawa Tengah, Kode Pos 50212.
2. Nama : Dega Surono Wibowo, S.T., M.Kom.  
Kewarganegaraan : Indonesia  
Alamat : Perumahan Sapphire Regency Block H No. 1 RT 04 RW 01 Kelurahan Pulosari, Kecamatan Brebes, Provinsi Jawa Tengah, Kode Pos 52213
3. Nama : Sharfina Febbi Handayani, M.Kom.  
Kewarganegaraan : Indonesia  
Alamat : Jalan Brayan 1 Desa Grobog Kulon RT 02 RW 06 Kecamatan Pangkah, Kabupaten Tegal, Provinsi Jawa Tengah, Kode Pos 52471

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

Nama : 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 “**Aplikasi Tracking Activity untuk Fasting, Drink, dan Step berbasis Android**” 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, 29 Februari 2024  
Pencipta

Pemegang Hak Cipta  
Ketua P3M



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



(Saksono Bayu Ajie Sumantri)

(Dega Surono Wibowo, S.T., M.Kom.)

(Sharfina Febbi Handayani, M.Kom.)

## **D. SYARAT PENGAJUAN HKI**

# **BUKU PETUNJUK PENGGUNAAN APLIKASI (USER MANUAL)**

## **Aplikasi Tracking Activity untuk Fasting, Drink, dan Step Berbasis Android**

Oleh :

Saksono Bayu Ajie Sumantri

Dega Surono Wibowo, S.T., M.Kom.

Sharfina Febbi Handayani, M.Kom.

## **1. PENDAHULUAN**

### **1.1. Tujuan Pembuatan Dokumen**

Dokumen *user manual* Aplikasi *Tracking Activity* untuk *Fasting, Drink, dan Step* Berbasis *Android* ini dibuat untuk tujuan menggambarkan dan menjelaskan penggunaan aplikasi untuk pengguna.

### **1.2. Deskripsi Umum Sistem**

Aplikasi *Tracking Activity* untuk *Fasting, Drink, dan Step* berbasis *Android* merupakan aplikasi berbasis *android* yang digunakan untuk membantu pengguna memonitoring kegiatan sehari – hari seperti *Fasting* untuk mengatur dan mengetahui waktu puasa atau diet, *Drink* untuk mengatur dan mengetahui berapa banyak air yang harus diminum tiap jam dalam sehari, dan *Step* untuk mengetahui seberapa banyak langkah yang dilakukan tiap aktifitas dengan hanya perlu membawa *smartphone*.

### **1.3. Deskripsi Dokumen**

Dokumen ini dibuat untuk memberikan panduan penggunaan “Aplikasi *Tracking Activity* untuk *Fasting, Drink, dan Step* Berbasis *Android*”. Dokumen ini berisikan informasi sebagai berikut :

#### **1. BAB I.**

Berisi informasi umum yang merupakan bagian pendahuluan, yang meliputi tujuan pembuatan dokumen, deskripsi umum sistem serta deskripsi dokumen.

#### **2. BAB II.**

Berisi perangkat yang dibutuhkan untuk penggunaan aplikasi meliputi perangkat lunak dan perangkat keras.

#### **3. BAB III.**

Berisi panduan penggunaan *Android*.

## **2. PERANGKAT YANG DIBUTUHKAN**

### **2.1. Perangkat Lunak**

Perangkat lunak yang digunakan adalah *Web Browser*.

### **2.2. Perangkat Keras**

Perangkat keras yang digunakan adalah *Android Smartphone*

### 3. MENU DAN CARA PENGGUNAAN

#### 3.1. Struktur Menu

Adapun struktur menu pada Aplikasi *Tracking Activity* untuk *Fasting*, *Drink*, dan *Step* Berbasis *Android* adalah sebagai berikut :

Menu Login	Menu Register	Menu Home	Menu Stats	Menu Daily	Menu Setting
1. Username	1. Username	1. Button	1. Stats	1. Fasting	1. Switch
2. Password	2. Password	Logout	Fasting	2. Drink	On/Off
3. Button	3. Email		2. Stats	3. Step	Drink
Login	4. Button		Drink	4. Nav	2. Nav
4. Button	Login		3. Stats	Stats	Stats
Register	5. Button		Step	5. Nav	3. Nav
5. Lupa	Register		4. Nav	Daily	Daily
Password			Stats	6. Nav	4. Nav
			5. Nav	Setting	Setting
			Daily		
			6. Nav		
			Setting		

#### 3.2. Penggunaan

Pada bagian ini akan dijelaskan mengenai tata cara Menggunakan Aplikasi *Tracking Activity* untuk *Fasting*, *Drink*, dan *Step* Berbasis *Android*.

##### 3.2.1. Penggunaan Aplikasi

Aplikasi *Tracking Activity* untuk *Fasting*, *Drink*, dan *Step* Berbasis *Android* memiliki 6 menu fitur untuk memonitoring aktifitas *Fasting*, *Drink*, dan *Step*, fitur – fiturnya sebagai berikut :



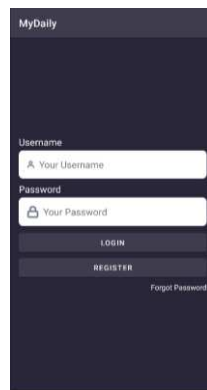
- Menu Login : Fitur ini digunakan untuk melakukan login dengan username dan password untuk pengguna yang sudah melakukan register sebelumnya. Kemudian lupa password untuk pengguna yang merasa lupa dengan password akunya dapat menggunakan fitur tersebut yang akan diarahkan ke menu reset dengan mengisi username dan email, beberapa saat akan mendapat email yang memuat link reset untuk mengganti password, setelah diakses link tersebut maka diarahkan ke website yang berisi new password dan confirm password disini pengguna dapat membuat password baru.
- Menu Register : Fitur ini digunakan untuk pengguna yang melakukan register akun yang memuat username, password, dan email.
- Menu Home : Fitur ini muncul ketika pengguna pertama kali mengakses aplikasi, yang memuat deskripsi singkat aplikasi dan tatacara penggunaan tiap fiturnya. Kemudian terdapat button logout untuk logout dari aplikasi.
- Menu Stats : Fitur ini untuk memonitoring history weekly dari Fasting yang berisi tabel day dan hours, Drink yang berisi tabel day dan intake, Step yang berisi tabel day dan steps.
- Menu Daily : Fitur ini untuk mengatur dan memonitoring fitur harian, Fasting dapat mengatur berapa jam untuk puasa atau diet dengan countdown timer, Drink untuk menampilkan konsumsi air tiap jam, Step untuk menampilkan langkah kaki dengan target 10000 langkah setiap hari.
- Menu Setting : Fitur ini untuk mengatur switch on off dari fitur Drink yang akan muncul notifikasi tiap jam untuk mengingatkan konsumsi air sebanyak 250mL dengan hanya perlu click intak water di notifikasi tanpa perlu akses ke dalam aplikasi.

Dalam Manual Book ini, pengguna akan diberikan petunjuk langkah demi langkah tentang penggunaan masing – masing fitur dengan jelas dan terperinci. Dengan memahami dan mengikuti alur fitur yang ada pada

aplikasi pengguna akan dapat memperoleh pengalaman dan pemahaman yang mendalam dan mudah dipahami.

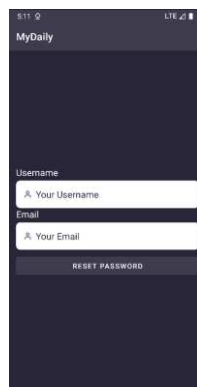
### 3.2.1.1. Menu Login

- 1) Buka Aplikasi *Daily Activity* di *Android Smartphone*. Seperti pada gambar 3.1 dibawah ini :



Gambar 3.1 Tampilan Awal Aplikasi.

- 2) Pengguna dapat melakukan *login* pada aplikasi dengan *username* dan *password*. Terdapat *forgot password* untuk pengguna yang lupa dengan *password* akunnya, yang apabila dipilih maka akan diarahkan ke halaman seperti pada gambar 3.2 berikut :



Gambar 3.2 Tampilan Reset Password.

- 3) Pengguna dapat mengisi *username* dan *password* untuk melakukan *reset password*. Pengguna mendapat email untuk *reset password* dan

setelah mengakses *link* tersebut akan diarahkan ke sebuah *form website* untuk membuat *password* baru halaman seperti pada gambar 3.3 dan gambar 3.4 berikut :



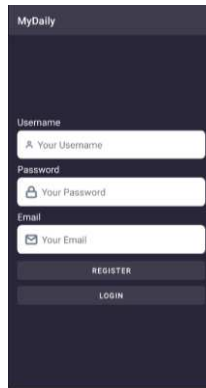
Gambar 3.3 Tampilan Email Reset Password.



Gambar 3.4 Tampilan Website Reset Password.

### 3.2.1.2. Menu Register

- 1) Pengguna dapat melakukan *register* untuk membuat akun dengan mengisi *username*, *password*, dan *email* seperti pada gambar 3.5.



Gambar 3.5 Tampilan Register.

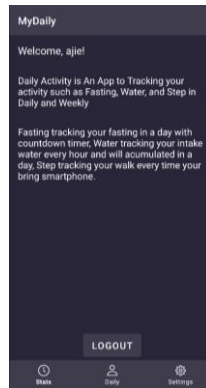
- 2) Jika pengguna berhasil membuat akun maka akan mendapat email berhasil mendaftar seperti pada gambar 3.6 :



Gambar 3.6 Tampilan Email berhasil mendaftar

### 3.2.1.3. Menu Home

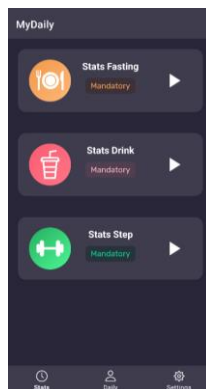
- 1) Berikut tampilan setelah *login* ke aplikasi, disini terdapat *username* pengguna yang digunakan, penjelasan singkat dari aplikasi beserta fitur – fitur *Fasting*, *Drink*, dan *Step*. Terdapat *logout* untuk keluar dari aplikasi. Kemudian ada 3 navigasi yang dapat mengarahkan ke fitur aplikasi, seperti pada gambar 3.7 berikut :



Gambar 3.7 Tampilan Home.

#### 3.2.1.4. Menu Stats

- 1) Tampilan Menu Stats memuat 3 fitur *Stats Fasting*, *Stats Drink*, dan *Stats Step*. Di menu stats terdapat 3 navigasi untuk mengarahkan ke fitur aplikasi seperti pada gambar 3.8 berikut :



Gambar 3.8 Tampilan Stats

- 2) Tampilan *Stats Fasting* memuat Day sebagai hari dan tanggal kemudian *hours* sebagai seberapa banyak jam puasa atau diet yang dilakukan dalam sehari seperti pada gambar 3.9 berikut :

Day	Hours
Sunday, February 25	8
Saturday, February 24	8
Friday, February 23	8
Thursday, February 22	1
Wednesday, February 21	1
Tuesday, February 20	1
Monday, February 19	1

Gambar 3.9 Tampilan Stats Fasting

- 3) Tampilan *Stats Drink* memuat Day sebagai hari dan tanggal kemudian *intake(mL)* sebagai seberapa banyak air yang dikonsumsi tiap hari, seperti pada gambar 4.0 berikut :

Day	Intake (mL)
Monday, February 26	250
Wednesday, November 8	1500

Gambar 4.0 Tampilan Stats Drink

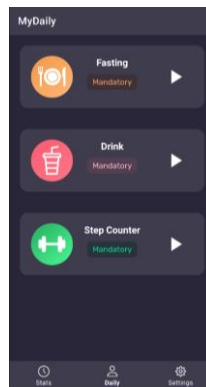
- 4) Tampilan *Stats Step* memuat Day sebagai hari dan tanggal kemudian *Steps* sebagai seberapa banyak langkah yang dilakukan selama aktifitas tiap hari, seperti pada gambar 4.1 berikut :

Day	Steps
Monday, February 26	9684
Sunday, February 25	7303
Saturday, February 17	3012
Friday, February 16	289
Thursday, February 15	511
Wednesday, February 14	1000
Tuesday, February 13	19021
Monday, February 12	18965
Sunday, February 11	10420

Gambar 4.1 Tampilan Stats Step

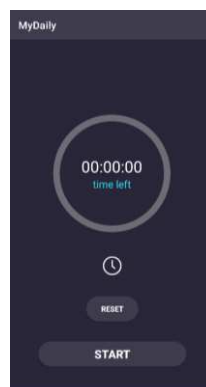
### 3.2.1.5. Menu Daily

- 1) Tampilan *Menu Daily* memuat 3 fitur *Fasting*, *Drink*, dan *Step*. Di menu stats terdapat 3 navigasi untuk mengarahkan ke fitur aplikasi seperti pada gambar 4.2 berikut :

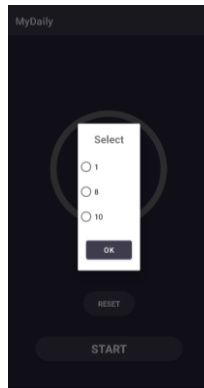


Gambar 4.2 Tampilan Daily

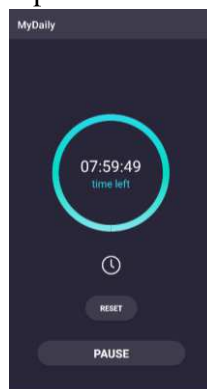
- 2) Tampilan *Fasting* memuat *timer* untuk mengatur waktu puasa atau diet yang disimbolkan *icon* jam, kemudian *start* setelah memilih waktu yang diinginkan maka *timer* akan berjalan hingga waktu habis dan memberikan notifikasi suara untuk mengingatkan, adapun *reset* untuk *reset* ulang waktu puasa atau diet. Di *Fasting* ini hanya dapat mengatur satu kali puasa atau diet dalam sehari, seperti pada gambar 4.3, 4.4, 4.5 berikut :



Gambar 4.3 Tampilan Fasting

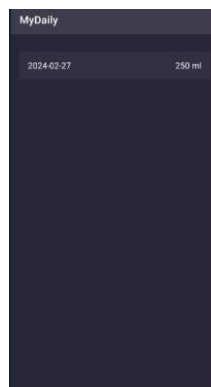


Gambar 4.4 Tampilan memilih waktu puasa atau diet



Gambar 4.5 Tampilan Timer countdown

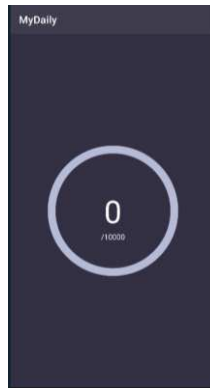
- 3) Tampilan *Drink* untuk menampilkan apakah sudah memilih *intake water* yang muncul di aplikasi tiap satu jam, seperti pada gambar 4.6 berikut :



Gambar 4.6 Tampilan Daily

- 4) Tampilan *Step* untuk menampilkan seberapa banyak langkah yang dilakukan secara *real time* dalam sehari, seperti pada gambar 4.7 berikut :

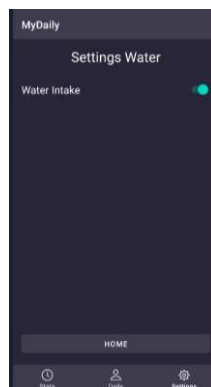




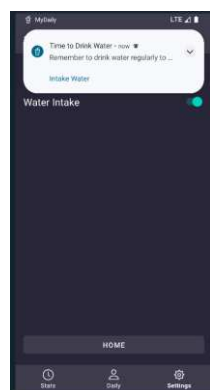
Gambar 4.7 Tampilan Step

### 3.2.1.6. Menu Setting

- 1) Tampilan untuk mengatur *switch* on/off dari fitur *Drink* dimana akan muncul notifikasi setiap 1 jam sekali untuk mengingatkan minum dengan memilih *intake water* di notifikasi untuk menyimpan data tanpa perlu masuk ke dalam aplikasi seperti pada gambar 4.8, 4.9 berikut :



Gambar 4.1 Tampilan Settings



Gambar 4.2 Tampilan Notifikasi Drink

# DOKUMEN TEKNIKAL

## Aplikasi Tracking Activity untuk Fasting, Drink, dan Step Berbasis Android

Oleh:

~~Saksono Bayu Aji Sumantri~~

~~Dega Surono Wibowo, S.T., M.Kom.~~

~~Sharfina Febhi Handayani, M.Kom.~~

## **Profile**

Aplikasi Tracking Activity untuk Fasting, Drink, dan Step berbasis Android adalah aplikasi berbasis *android* yang digunakan untuk membantu pengguna memonitoring aktifitas sehari-hari, Aplikasi ini juga menawarkan fitur-fitur yang terdapat di beberapa aplikasi yang memudahkan pengguna menerapkan karena terangkum dalam satu aplikasi, dan juga dapat menampilkan hasil mingguan aktifitas yang dilakukan.

## **Latar Belakang**

Saat ini di mana kesehatan dan kebugaran telah menjadi prioritas utama bagi banyak orang, banyak aplikasi kesehatan yang beredar untuk memenuhi kebutuhan masyarakat. Akan tetapi aplikasi yang beredar hanya memberikan satu fitur kesehatan sehingga orang perlu untuk menginstall beberapa aplikasi untuk memonitor kesehatan fisiknya. Oleh sebab itu perlu aplikasi yang memuat beberapa fitur yang akan membantu pengguna memonitor kesehatan fisik mereka. Aplikasi yang dikembangkan menciptakan solusi lengkap yang melacak *fasting*, *drink*, dan *step* yang diambil sepanjang hari. Aplikasi yang dinamakan "Aplikasi *Tracking Activity* untuk *Fasting*, *Drink*, dan *Step* berbasis *Android* adalah aplikasi berbasis *android*" ini dirancang agar mudah digunakan dan dapat disesuaikan, sehingga pengguna dapat menetapkan tujuan dan memonitor progress mereka dari waktu ke waktu.

## **Manfaat**

Adapun Manfaat yang dapat dihasilkan, Sebagai berikut :

- a. Membantu pengguna menyederhanakan penggunaan aplikasi ini karena memuat fitur dari beberapa aplikasi yaitu *fasting*, *drink*, dan *step*
- b. Membantu pengguna untuk tetap bertanggung jawab dan termotivasi dalam perjalanan kesehatan dan kebugaran mereka. Dengan menetapkan tujuan yang spesifik dan terukur serta memantau kemajuan mereka dari waktu ke waktu, pengguna akan lebih mungkin untuk tetap berpegang pada kebiasaan mereka dan melihat hasil yang nyata.

- c. Mendapatkan pemahaman yang lebih baik tentang tubuh mereka dan kebutuhan kesehatan mereka yang lebih spesifik. Dengan memonitor periode *fasting*, *drink*, dan *step*, pengguna dapat mengidentifikasi pola dan melakukan penyesuaian sesuai kebutuhan.

## Spesifikasi Teknis

Spesifikasi Teknik meliputi:

- a. *Source Code*

Berikut uraian spesifikasi untuk pembangunan aplikasi *android*, antara lain :

- a. Android Studio
- b. Visual Studio Code

Berikut uraian spesifikasi modul :

### Source Code

Pada source code kali ini akan menjelaskan sebagian besar kode yang digunakan untuk membuat aplikasi ini, khususnya client kotlin dan server python yang digunakan pada aplikasi. Berikut ini adalah penjelasan mengenai kode – kode tersebut

#### 1. Proses Pembuatan Client Kotlin

```
1 package com.assistant.mydaily.user.login
2
3 import androidx.appcompat.app.AppCompatActivity
4
5 class LoginActivity : AppCompatActivity() {
6
7     private lateinit var jetToken: String
8
9     override fun onCreate(savedInstanceState: Bundle?) {
10         super.onCreate(savedInstanceState)
11
12         val isLoggedIn = checkIfUserLoggedIn()
13
14         if (isLoggedIn) {
15             navigateToMainActivity()
16             return
17         }
18
19         setContentView(R.layout.activity_login)
20
21         val usernameEditText = findViewById<EditText>(R.id.username)
22         val passwordEditText = findViewById<EditText>(R.id.password)
23         val loginButton = findViewById<Button>(R.id.loginBtnLog)
24         val registerButton = findViewById<Button>(R.id.registerBtnLog)
25         val forgotPasswordTextView = findViewById<TextView>(R.id.forgotPwdT)
26
27         loginButton.setOnClickListener {
28             val username = usernameEditText.text.toString()
29             val password = passwordEditText.text.toString()
30             if (username.isEmpty() || password.isEmpty()) {
31
32             }
33         }
34     }
35 }
```

Gambar 1 Menu Login

Dari gambar di atas menjabarkan bahwa kode memuat menu untuk mengisi *username*, *password*, tombol *login*, tombol *register*, dan *forgot password*. Tombol *login* saat dipilih akan diarahkan masuk ke dalam aplikasi sedangkan tombol *register* saat dipilih akan diarahkan ke menu

*register*. Saat pertama kali mengakses Aplikasi *Tracking Activity* untuk *Fasting, Drink, dan Step* berbasis *Android* pengguna akan diarahkan ke menu login dan pengguna dapat menggunakan fitur *forgot password* untuk memulihkan atau membuat password baru.

```
1 package com.assistant.ydaily.user.register
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import androidx.appcompat.widget.Toolbar
6 import androidx.core.app.ActivityCompat
7 import androidx.core.app.NotificationCompat
8 import androidx.core.app.NotificationManagerCompat
9 import androidx.core.content.ContextCompat
10 import androidx.core.widget.TextViewCompat
11 import androidx.core.widget.TextViewCompat
12 import androidx.core.widget.TextViewCompat
13 import androidx.core.widget.TextViewCompat
14 import androidx.core.widget.TextViewCompat
15 import androidx.core.widget.TextViewCompat
16 import androidx.core.widget.TextViewCompat
17 import androidx.core.widget.TextViewCompat
18 import androidx.core.widget.TextViewCompat
19 import androidx.core.widget.TextViewCompat
20 import androidx.core.widget.TextViewCompat
21
22 class RegisterActivity : AppCompatActivity() {
23
24     private val authService: AuthService by lazy {
25         RetrofitClient.apiService
26     }
27
28     override fun onCreate(savedInstanceState: Bundle?) {
29         super.onCreate(savedInstanceState)
30         setContentView(R.layout.activity_register)
31
32         val usernameReg = findViewById<EditText>(R.id.userReg)
33         val passwordReg = findViewById<EditText>(R.id.pwdReg)
34         val emailReg = findViewById<EditText>(R.id.emailReg)
35         val registerButton = findViewById<Button>(R.id.registerBtnReg)
36         val loginButton = findViewById<Button>(R.id.loginBtnReg)
37
38         registerButton.setOnClickListener { @SuppressLint("SetText")
39             Log.d("RegisterActivity", "Register button clicked")
40             val username = usernameReg.text.toString()
41             val password = passwordReg.text.toString()
42             val email = emailReg.text.toString()
43
44             if (username.isNotEmpty() && password.isNotEmpty()) {
45                 Log.d("RegisterActivity", "Calling registerAccount function")
46                 registerAccount(username, password, email)
47             }
48         }
49     }
50 }
```

Gambar 2 Menu Register

Dari gambar diatas menjabarkan bahwa kode memuat menu untuk mengisi *username, password, email*, tombol *login*, dan tombol *register*. Tombol *login* saat dipilih akan ke menu login sedangkan tombol *register* saat dipilih akan muncul notifikasi akun berhasil dibuat dan mendapat notifikasi dari *email* terima kasih sudah mendaftar. Pengguna yang sudah berhasil mendaftar dapat melakukan *login* ke dalam aplikasi.

```
1 package com.assistant.ydaily.user.reset
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import androidx.appcompat.widget.Toolbar
6 import androidx.core.app.ActivityCompat
7 import androidx.core.app.NotificationCompat
8 import androidx.core.app.NotificationManagerCompat
9 import androidx.core.content.ContextCompat
10 import androidx.core.widget.TextViewCompat
11 import androidx.core.widget.TextViewCompat
12 import androidx.core.widget.TextViewCompat
13 import androidx.core.widget.TextViewCompat
14 import androidx.core.widget.TextViewCompat
15 import androidx.core.widget.TextViewCompat
16 import androidx.core.widget.TextViewCompat
17 import androidx.core.widget.TextViewCompat
18 import androidx.core.widget.TextViewCompat
19 import androidx.core.widget.TextViewCompat
20 import androidx.core.widget.TextViewCompat
21
22 class ResetActivity : AppCompatActivity() {
23
24     override fun onCreate(savedInstanceState: Bundle?) {
25         super.onCreate(savedInstanceState)
26         setContentView(R.layout.activity_reset)
27
28         val userEditText = findViewById<EditText>(R.id.userReset)
29         val emailEditText = findViewById<EditText>(R.id.emailReset)
30         val resetButton = findViewById<Button>(R.id.resetBtn)
31
32         resetButton.setOnClickListener { @SuppressLint("SetText")
33             val username = userEditText.text.toString()
34             val email = emailEditText.text.toString()
35
36             sendResetPasswordRequest(username, email)
37         }
38     }
39
40     private fun sendResetPasswordRequest(username: String, email: String) {
41         val resetRequest = ResetRequest(username, email)
42
43         CoroutineScope(Dispatchers.ID).launch { @SuppressLint("SetText")
44             try {
45                 val response = RetrofitClient.apiService.generateResetToken(resetRequest)
46                 if (response.isSuccessful) {
47                     // handle successful response
48                     val resetResponse = response.body()
49                     runOnUiThread {
50
51                     }
52                 }
53             } catch (e: Exception) {
54                 // handle error response
55             }
56         }
57     }
58 }
```

Gambar 3 Menu Reset

Dari gambar diatas menjabarkan bahwa kode memuat menu untuk mengisi *username* dan *email* untuk melakukan reset password. Setelah *reset password* dipilih maka pengguna akan mendapatkan notifikasi dari *email*

berupa *link* untuk membuat *password* baru. Setelah link dipilih pengguna akan diarahkan ke sebuah *website* yang memuat *form new password* dan *confirm password*, di sini pengguna dapat membuat *password* baru dan setelah itu submit dipilih maka secara otomatis *password* baru akan tersimpan ke dalam *database*. Pengguna dapat mengakses aplikasi menggunakan *password* baru.

```

1 package com.assistant.mydaily.activity.fasting
2
3 import androidx.appcompat.app.AppCompatActivity
4
5 class FastingTimer : AppCompatActivity() {
6     private lateinit var viewModel: FastingTimerViewModel
7     private var timeSelected: Int = 0
8     private var timeCountDown: CountdownTimer? = null
9     private var timeProgress = 0
10    private var isStart = true
11    private var mediaPlayer: MediaPlayer? = null
12    private lateinit var sharedPreferences: SharedPreferences
13
14    // Save the instance state
15    override fun onSaveInstanceState(outState: Bundle?) {
16        super.onSaveInstanceState(outState)
17        outState.putInt("timeSelected", timeSelected)
18        outState.putInt("timeProgress", timeProgress)
19        outState.putBoolean("isStart", isStart)
20    }
21
22    // Restore the instance state
23    override fun onRestoreInstanceState(savedInstanceState: Bundle?) {
24        super.onRestoreInstanceState(savedInstanceState)
25        timeSelected = savedInstanceState.getInt("timeSelected", 0)
26        timeProgress = savedInstanceState.getInt("timeProgress", 0)
27        isStart = savedInstanceState.getBoolean("isStart", true)
28    }
29
30    // Logic of timer
31
32 }

```

Gambar 4 Fasting Timer

Dari gambar diatas menjabarkan bahwa kode mendefinisikan *timeSelected* sebagai waktu yang dipilih yaitu 1 jam, 6 jam, 8 jam. *Timeprogress* sebagai countdown timer yaitu waktu mundur yang berjalan hingga 0 setelah memilih start. *Isstart* sebagai tombol start yang dapat dipilih pengguna setelah memilih waktu fasting yang diinginkan. Di Fasting Timer ini juga sudah mendefinisikan untuk melakukan start fasting time hanya satu kali dalam sehari, sehingga pengguna tidak bisa asal memilih dan start waktu fasting.

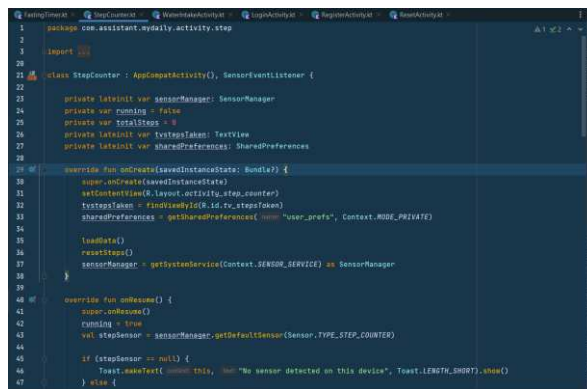
```

1 package com.assistant.mydaily.activity.water
2
3 import androidx.appcompat.app.AppCompatActivity
4
5 class WaterIntakeActivity : AppCompatActivity() {
6     private lateinit var sharedPreferences: SharedPreferences
7     private lateinit var recyclerView: RecyclerView
8     private lateinit var adapter: WaterIntakeAdapter
9     private var waterIntakeList: MutableList<WaterIntake> = mutableListOf()
10    private lateinit var apiService: ApiService
11
12    override fun onCreate(savedInstanceState: Bundle?) {
13        super.onCreate(savedInstanceState)
14        setContentView(R.layout.activity_water_intake)
15
16        sharedPreferences = getSharedPreferences("my_prefs", MODE_PRIVATE)
17        recyclerView = findViewById(R.id.rv_water_intake)
18        adapter = WaterIntakeAdapter(waterIntakeList)
19        recyclerView.layoutManager = LinearLayoutManager(this)
20        recyclerView.adapter = adapter
21
22        // Initialize ApiService here
23        apiService = RetrofitClient.apiService
24
25        val userId = sharedPreferences.getInt("userId", 0).toString()
26        Log.d("WaterIntakeActivity", "User ID: $userId")
27
28        if (userId.isNotEmpty()) {
29            addIntakeToServer(userId)
30        } else {
31
32 }

```

Gambar 5 Water Intake

Dari gambar diatas menjabarkan bahwa kode mendefinisikan *userId* untuk menyimpan data *intake water*. Kemudian didefinisikan juga untuk melakukan *intake water* cukup dari notifikasi yang muncul 1 jam sekali di *smartphone android*. Disini juga didefinisikan seberapa banyak air yang harus dikonsumsi pengguna yaitu 250mL setiap jam. Apabila pengguna tidak memilih *Intake Water* saat muncul notifikasi maka data tidak akan tersimpan ke dalam aplikasi, sehingga ini dapat membangun kebiasaan agar pengguna dapat memonitor konsumsi air setiap hari.



```
1 package com.assistant.mydaily.activity.step
2
3 import java.util.*
4
5 class StepCounter : AppCompatActivity(), SensorEventListener {
6
7     private lateinit var sensorManager: SensorManager
8     private var running = false
9     private var totalSteps = 0
10    private lateinit var tvStepsLabel: TextView
11    private lateinit var sharedPreferences: SharedPreferences
12
13    override fun onCreate(savedInstanceState: Bundle?) {
14        super.onCreate(savedInstanceState)
15        setContentView(R.layout.activity_step_counter)
16        tvStepsLabel = findViewById(R.id.tv_steps_label)
17        sharedPreferences = getSharedPreferences("user_prefs", Context.MODE_PRIVATE)
18
19        loadData()
20        resetSteps()
21        sensorManager = getSystemService(Context.SENSOR_SERVICE) as SensorManager
22    }
23
24    override fun onResume() {
25        super.onResume()
26        running = true
27        val stepSensor = sensorManager.getDefaultSensor(Sensor.TYPE_STEP_COUNTER)
28
29        if (stepSensor == null) {
30            Toast.makeText(this, "No sensor detected on this device", Toast.LENGTH_SHORT).show()
31        } else {
32
33        }
```

Gambar 6 Step Counter

Dari gambar diatas menjabarkan bahwa kode mendefinisikan sensor *gyro* dari *smartphone android* yaitu *sensormanager* yang berfungsi untuk memanggil fungsi dari step counter. Disini didefinisikan untuk menyimpan data dari *sharedpreference*, sehingga *history* fitur *step counter* akan terus berjalan walau *smartphone* tidak digunakan tetapi perlu untuk dibawa pengguna.

## 2. Proses Pembuatan Server Python

```
# Login route
Codiumate: Options | Test this function
@app.route('/login', methods=['POST'])
def login():
    data = request.json
    username = data.get('username')
    password = data.get('password')

    if validate_login(username, password):
        user_id = get_user_id_by_username(username)
        new_access_token = create_access_token(identity=user_id)
        access_token_set.add(new_access_token) # Add the token to the global set
        print(f"Current access_token_set (login): {access_token_set}")
        return jsonify({'access_token_set': new_access_token, 'user_id': user_id, 'username': username, 'message': 'Login successful',
        else:
            return jsonify({'message': 'Invalid username or password', 'isLogged': False}), 401
```

Gambar 7 Route Login

Dari gambar diatas menjabarkan bahwa kode *route /login* untuk menyimpan data. Data yang disimpan adalah *username* dan *password*. Disini melakukan *validasi login* *user\_id* dan *username* untuk membuat atau *create jwt token* sehingga dapat dilakukan autentikasi. Apabila pengguna berhasil *login* maka akan terdapat *log* atau *history* di *server* '*Login successful*' dengan *user\_id* dan *username* terkait. Apabila pengguna salah mengisi *username* dan *password* maka akan muncul log '*Invalid username or password*'.

```
# register route
Codiumate: Options | Test this function
@app.route('/register', methods=['POST'])
def register():
    data = request.json
    username = data.get('username')
    password = data.get('password')
    email = data.get('email')

    # Check if email is already registered
    if is_email_registered(email):
        return jsonify({'message': 'Email is already registered'}), 400

    # Check if username is already registered
    if is_username_registered(username):
        return jsonify({'message': 'Username is already registered'}), 400

    # Store the user
    store_user(username, password, email)

    # Send the registration email
    send_registration_email(email)

    response_data = {'message': 'Registration successful', 'isRegistered': True}
    return jsonify(response_data)
```

Gambar 8 Route Register

Dari gambar diatas menjabarkan bahwa kode *route /register* untuk menyimpan data. Data yang disimpan adalah *username*, *password* dan *email*. Disini melakukan validasi apakah *email* dan usernamenya sudah ada sebelumnya atau belum, jika belum maka data akan tersimpan ke *database*. Setelah pengguna berhasil membuat akun maka pengguna akan mendapat



notifikasi email selamat sudah mendaftar. Apabila pengguna berhasil *login* maka akan terdapat *log* atau *history* di *server* '*Registration successful*'.

```
#send reset mail
Codiumate: Options | Test this function
def send_reset_email(username, recipient, reset_token):
    try:
        subject = "Password Reset"
        sender_email = "19098001.sumantri@student.poltektegal.ac.id"
        reset_link = f"http://192.168.43.165:5000/reset-password?token={reset_token}"

        body = f"Hello {username},\n\nClick the following link to reset your password: {reset_link}"

        msg = Message(subject, sender=sender_email, recipients=[recipient])
        msg.body = body
        mail.send(msg)

        app.logger.info(f"Reset email sent successfully to {recipient}")
    except Exception as e:
        app.logger.error(f"Error sending reset email to {recipient}: {str(e)}")
```

Gambar 9 Route Reset 1

Dari gambar diatas menjabarkan bahwa kode untuk mengirim email *reset* berisi *link* yang diarahkan ke *website* yang berisi *form new password* dan *confirm password*. Apabila pengguna berhasil memilih *reset password* di aplikasi maka akan muncul *log* '*Reset email sent successfully*'. Apabila pengguna salah mengisi *username* dan *email* maka akan muncul *log* '*Error sending reset email*'.

```
Codiumate: Options | Test this function
app.route('/reset-password', methods=['GET', 'POST'])
def reset_password():
    if request.method == 'POST':
        # Handle the POST request to reset the password
        data = request.json # use request.json to get JSON data
        new_password = data.get('new_password')
        reset_token = data.get('reset_token')

    try:
        # Check if the form data is missing
        if new_password is None or reset_token is None:
            raise ValueError("Missing form data. Please provide new_password and reset_token.")

        # Log the received form data
        app.logger.info(f"Received form data - new_password: {new_password}, reset_token: {reset_token}")

        # Decode the reset token
        decoded_token = decode(reset_token.encode('utf-8'), options={"verify_signature": False})

        # Log the decoded token for debugging
        app.logger.info(f"Decoded token: {decoded_token}")

        # Extract information from the token
        username = decoded_token.get('sub')

        # Log the generated username for debugging
        app.logger.info(f"Username: {username}")

        if username is None:
            raise ValueError("Token does not contain 'identity' field.")

        # Update the user's password in the database
        update_password(username, new_password)
```

Gambar 10 Route Reset 2

```

# Optionally, create a new access token after password reset
new_access_token = create_access_token(identity=username)

return jsonify({'message': 'Password reset successful', 'new_access_token': new_access_token}), 200
except Exception as e:
    # Log the exception
    app.logger.error("Error resetting password:", exc_info=True)
    return jsonify({'message': f'Invalid reset token or missing form data: {str(e)}'}), 401
elif request.method == 'GET':
    # Generate a new reset token with the correct identity (username)
    # You should replace "username_from_reset_request" with the actual username from your reset request logic
    new_reset_token = create_access_token(identity=get_username_for_reset(request.json), expires_delta=False)

    # Render the template with the new reset token
    return render_template('reset_password_form.html', reset_token=new_reset_token)
else:
    # Handle other methods
    return jsonify({'message': 'Method Not Allowed'}), 405

```

Gambar 11 Route Reset 3

Dari gambar 10 dan 11 diatas menjabarkan bahwa kode untuk mengisi *password* baru. Dengan mengisi *new password* dan *confirm password* dan memilih *submit* maka password baru akan tersimpan ke *database*. Apabila pengguna berhasil mengisi *password* baru di *form website* maka akan muncul log '*Password reset successfully*'. Apabila sistem gagal untuk menyimpan *password* baru maka akan muncul log '*Error Resetting Password*'.

```

#save_fasting
CodeMate Online! Test this function
@app.route('/save_fasting', methods=['POST'])
def save_fasting():
    try:
        data = request.get_json()
        user_id = data.get('user_id')
        hours = data.get('hours')
        date = data.get('date') # Get the date from the request

        print(f"Received fast: (hours), user_id: (user_id), date: (date)")

        # Connect to the database and save data
        with pymysql.connect(**db_config) as conn:
            with conn.cursor() as cursor:
                cursor.execute("INSERT INTO fast_data (user_id, hours, date) VALUES (%s, %s, %s)", (user_id, hours, date))
            conn.commit()

        return jsonify({'message': 'Fast Data saved successfully'}), 200

    except Exception as e:
        print(f"Error: {e}")
        return jsonify({'error': 'Internal Server Error'}), 500

```

Gambar 12 Route Save Fasting

Dari gambar diatas menjabarkan bahwa kode untuk menyimpan data *fasting* berupa *user\_id*, *hours*, dan *date*. *User\_id* adalah id pengguna, *hours* adalah waktu *fasting* pengguna, *date* adalah tanggal ketika pengguna melakukan *fasting*. Disini didefinisikan data disimpan ke dalam tabel *fast\_data* di kolom *user\_id*, *hours*, dan *date*. Apabila sistem berhasil

menyimpan data maka akan muncul log *'Fast Data Saved successfully'*. Apabila sistem gagal menyimpan data maka akan muncul log *'Internal Server Error'*.

```
from intake water
columns: (rows) (url) (function)
app.route('/add_intake', methods=['POST'])
def add_intake():
    try:
        data = request.get_json()

        # Ensure 'user_id' key is present in the JSON data
        if 'user_id' not in data:
            return jsonify({'error': 'Missing user_id in input data'}), 400

        # Validate input data
        user_id = data['user_id']
        date = data['date'] # Changed from 'dateTime' to 'date'
        amount = data['amount']

        if not all([user_id, date, amount]):
            return jsonify({'error': 'Invalid input data'}), 400

        # Parse date string into a Python date object
        date_obj = datetime.strptime(date, "%Y-%m-%d").date() # Assuming date format is 'YYYY-MM-DD'

        # Use a context manager to connect to the MySQL database
        with pymysql.connect(**db_config) as conn:
            try:
                # Begin a transaction
                conn.begin()

                with conn.cursor() as cursor:
                    # Insert the data into the MySQL database
                    insert_query = "INSERT INTO water_intake (user_id, date, amount) VALUES (%s, %s, %s)" # Changed from 'dateTime' to
                    data = (user_id, date_obj, amount) # Changed 'datetime_obj' to 'date_obj'
                    cursor.execute(insert_query, data)

            # Commit the transaction
            conn.commit()
```

Gambar 13 Route Add Intake 1

```
# Prepare the response data
response_data = {
    'user_id': user_id,
    'date': date, # Changed from 'dateTime' to 'date'
    'amount': amount,
    'message': 'Water intake record added successfully'
}

# Return a success message
return jsonify({'message': 'Water intake record added successfully'}), 200
except pymysql.Error as e:
    # Rollback the transaction in case of an error
    conn.rollback()
    error_message = f"MySQL Error: {str(e)}"
    app.logger.error(error_message)
    return jsonify({'error': error_message}), 500
except Exception as e:
    # Handle other exceptions
    error_message = f"An error occurred: {str(e)}"
    app.logger.error(error_message)
    return jsonify({'error': error_message}), 500
```

Gambar 14 Route Add Intake 2

Dari gambar 13 dan gambar 14 diatas menjabarkan bahwa kode untuk menyimpan data *intake water*. Data yang disimpan adalah *user\_id*, *date*, dan *amount*. *User\_id* adalah id pengguna, *date* adalah *waktu intake water* pengguna, *amount* adalah jumlah konsumsi air ketika pengguna melakukan *intake water*. Disini didefinisikan data disimpan ke dalam tabel *water\_intake* di kolom *user\_id*, *amount*, dan *date*. Apabila sistem berhasil menyimpan data maka akan muncul log *'Water Intake Record Add successfully'*. Apabila sistem gagal menyimpan data maka akan muncul log *'An error occured'*.

```

#save steps data
Codiumate: Options | Test this function
@app.route('/save_steps', methods=['POST'])
def save_steps():
    try:
        data = request.get_json()
        steps = data['steps']
        user_id = data['user_id'] # Make sure to include 'user_id' in the JSON data

        print(f"Received steps: {steps}, user_id: {user_id}")
        # Get the current date and time
        current_date_time = datetime.utcnow().strftime('%Y-%m-%d')

        with pymysql.connect(**db_config) as conn:
            with conn.cursor() as cursor:
                # Assuming your step_data table has 'user_id', 'steps', and 'timestamp' columns
                cursor.execute("INSERT INTO steps_data (user_id, steps, date) VALUES (%s, %s, %s)", (user_id, steps, current_date_time))
            conn.commit()

    return jsonify({'message': 'Steps saved successfully'}), 200

```

Gambar 15 Route Save Step Counter

Dari gambar diatas menjabarkan bahwa kode untuk menyimpan *step counter*. Data yang disimpan adalah *user\_id*, *steps*, dan *date*. *User\_id* adalah id pengguna, *step* adalah jumlah langkah yang dilakukan pengguna, *date* adalah tanggal pengguna melakukan aktifitas yang berupa langkah. Disini didefinisikan data disimpan ke dalam tabel *steps\_data* di kolom *user\_id*, *steps*, dan *date*. Apabila sistem berhasil menyimpan data maka akan muncul log '*Step saved successfully*'. Apabila sistem gagal menyimpan data maka akan muncul log '*An error occured*'.

## E. SERTIFIKAT HKI YANG TERBIT

  
REPUBLIC 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 : EC00202421634, 7 Maret 2024

**Pencipta**  
Nama : **Saksono Bayu Ajie Sumantri, Dega Surono Wibowo, S.T., M.Kom. dkk**

Alamat : Dsn. Mranak RT 04 RW 09 Kelurahan Wonorejo, Kecamatan Pringapus, Kabupaten Semarang, Propinsi Jawa Tengah 50212, Pringapus, Semarang, Jawa Tengah, 50212

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 : **Aplikasi Tracking Activity Untuk Fasting, Drink, Dan Step Berbasis Android**

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 7 Maret 2024, di Tegal

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor pencatatan : 000596991

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  
u.b  
Direktur Hak Cipta dan Desain Industri

  
Anggoro Dasananto  
NIP. 196412081991031002



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



**LAMPIRAN PENCIPTA**

No	Nama	Alamat
1	Saksono Bayu Ajie Sumantri	Dsn. Mranak RT 04 RW 09 Kelurahan Wonorejo, Kecamatan Pringapus, Kabupaten Semarang, Propinsi Jawa Tengah 50212, Pringapus, Semarang
2	Dega Surono Wibowo, S.T., M.Kom.	Perumahan SAPPHIRE REGENCY Block H NO. 1 RT 04 RW 01 Kelurahan Pulosasi, Kecamatan Brebes, Kabupaten Brebes, Propinsi Jawa Tengah 52213, Brebes, Brebes
3	Sharfina Febbi Handayani, M.Kom.	Jl. Brayon 1 No 46, Desa Grobog Kulon, Kecamatan Pangkah, Kabupaten Tegal, Jawa Tengah, 52471, Pangkah, Tegal



## F. LEMBAR BIMBINGAN SKRIPSI



SARJANA TERAPAN TEKNIK INFORMATIKA  
POLITEKNIK HARAPAN BERSAMA

### LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Saksono Bayu Ajie Sumantri  
 NIM : 19098001  
 No. Ponsel : 087764453893  
 Judul TA : ASISTEN VIRTUAL UNTUK SMART  
 DAILY ACTIVITIES BERBASIS ANDROID  
 Dosen Pembimbing I: Dega Surono Wihowo, M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	6/3/23	* Buat use case - Fungsi-fungsi user		<i>[Signature]</i>
2	27/3/23	* use case dibuat ulang dengan request yang benar		<i>[Signature]</i>
3	5/4/23	* Use case sudah fix * UI/UX warna dan tulisan - to do (+) * memvalidasi user (email) * validasi sleep, awake, ... * atur bar sistem dan layout mana		<i>[Signature]</i>
4.	28/4/23	* silahkan build apk sambil mencari (Get-Set-Hold) * berikan manual book & beladiri		<i>[Signature]</i>
5	24/8/23	silahkan & lampirkan screenshot apk		<i>[Signature]</i>

6.	20/2 24	Gilabuan masalah Laporan	✗
7	7/3 24	Laporan 1. Tinjauan pustaka tambah 2 referensi 2. Stop - diperbaiki lagi	✗
8	21/3 24	Gilabuan - Dypar Sifatnya persiapan untuk Sifatnya	✗



--	--	--	--	--

Dca  
Si Jang

Tegal, 21-3-2024  
Dosen Pembimbing I



Dega Strono Wibowo, M.Kom.  
NIPY. 06.014.183



SARJANA TERAPAN TEKNIK INFORMATIKA  
POLITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

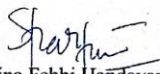
Nama : Salsono Bayu Ajie Sumantri  
NIM : 19098001  
No. Ponsel : 087764453893  
Judul TA : ASISTEN VIRTUAL UNTUK SMART  
DAILY ACTIVITIES BERBASIS ANDROID  
Dosen Pembimbing II : Sharfina Febbi Handayani, M.Kom

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1	20 Maret 2023	Latarbelakang  alur penelitian	ditambahkan Statistik/ fenomena terkait pola minum dan pola tidur  detailkan kembali dataset yang akan digunakan	Spt  Sp
2	27 Maret 2023	Dataset	Can' kembali Kanal/ source yang di gunakan untuk Lifestyle Eat, drink, sleep workout	Sp
3	31 Maret 2023	Diet	ditambahkan detail diet di aplikasi berupa list dan reminder	Sp
4	3 Mei 2023	Progres Aplikasi	Ditambahkan Statistik w/ histori	Sp
5	4 Juli 2023	Progres Aplikasi	diperbaiki aplikasi mobilens	Sharfina

6.	2 Agustus 2023	Progres Aplikasi	<ul style="list-style-type: none"> <li>- Pertimbangkan menggunakan flask sbg web service nya</li> <li>- Database Stoba &amp; ganti ke mysql saja</li> </ul>	Shafiq
7.	23 Agustus 2023	Progres Aplikasi	<ul style="list-style-type: none"> <li>- Lanjutkan w/ menyelesaikan activity sleep, fasting</li> </ul>	Shafiq
8.	12 Oktober 2023	Progress Aplikasi	<ul style="list-style-type: none"> <li>- Lanjutkan w/ Progres step dan fasting</li> <li>- diskusi dengan teman terkait Kotlin</li> </ul>	Shafiq
9.	7 Nov 2023	Progress Aplikasi	<ul style="list-style-type: none"> <li>- Lanjutkan w/ menyelesaikan Beta Simpan Session mysql flask</li> <li>- Persiapkan pembuatan manual book.</li> </ul>	Shafiq
10.	16 Nov 2023	Progress Aplikasi	<ul style="list-style-type: none"> <li>- Lanjutkan w/ menyelesaikan fitur drink</li> </ul>	Shafiq
11.	2 Feb 2024	Progress Aplikasi	<ul style="list-style-type: none"> <li>- Lanjutkan w/ menyelesaikan fitur fasting</li> </ul>	Shafiq
12.	26 Feb 2024	Progress Aplikasi	<ul style="list-style-type: none"> <li>- Fitur Fasting, Step dan Drink sudah dapat berjalan dg baik</li> </ul>	Shafiq

13	19/03/2024	Laporan Bab 1 dan Bab 2	Pertemuan Latar Gelombang tambahkan Data/ Fenomena Awareness terkait Kesehatan	Sharfina
14.	22/03/2024	ACC		Sharfina

Tegal, ..... 22 Maret 2024  
Dosen Pembimbing II

  
Sharfina Febbi Handayani, M.Kom  
NIPY. 08.020.451