



LAPORAN PROJECT TEAM

**BY :
KELOMPOK 16**

Luthfi Daffa H(I0322070)
Mahardika W.N (I0322073)
M. Natan ydh (I0322086)
Nurrafi Narendraji (i0322096)
Raisa Azzahra (I0322103)

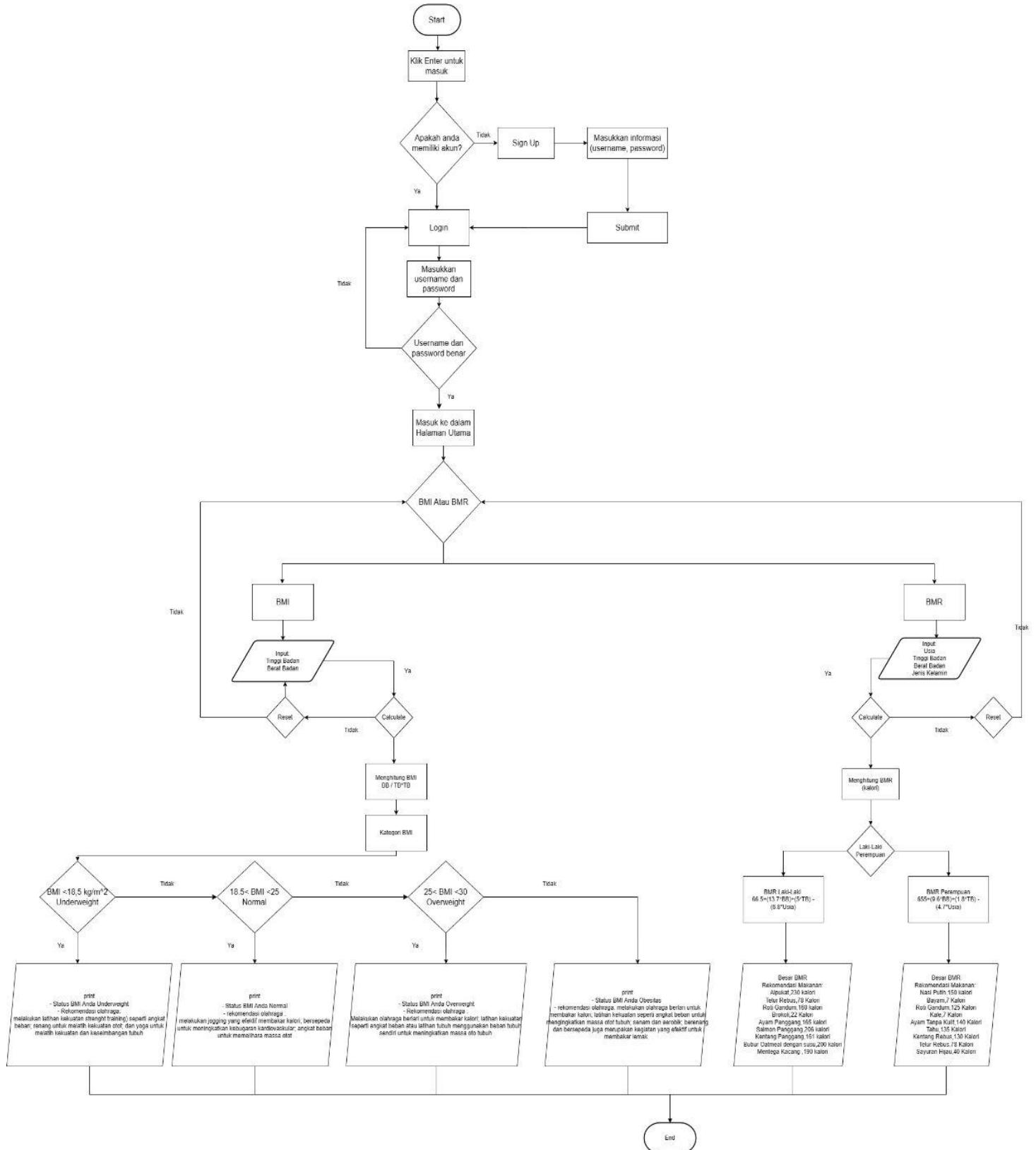


BAB I

DESKRIPSI MASALAH

Asupan gizi yang seimbang dan berkualitas sangat penting bagi pertumbuhan dan perkembangan anak dan remaja. Namun, sayangnya, masalah asupan gizi pada anak dan remaja menjadi isu yang signifikan dalam masyarakat kita saat ini. Beberapa masalah yang muncul antara lain pola makan yang tidak seimbang. Banyak anak dan remaja cenderung mengonsumsi makanan cepat saji, camilan tidak sehat, dan minuman manis yang kaya akan gula dan lemak jenuh. Mereka sering kali mengabaikan konsumsi buah, sayur, protein, dan serat yang penting untuk pertumbuhan dan perkembangan mereka. Hal ini melatar belakangi kami untuk mengembangkan program ini. Tujuan kelompok kita adalah membuat program menggunakan bahasa pemrograman Python untuk menghitung Body Mass Index (BMI) dan Basal Metabolic Rate (BMR). BMI dan BMR adalah dua pengukuran yang umum digunakan dalam dunia kesehatan dan kebugaran. Dengan adanya program ini, pengguna dapat dengan mudah menghitung BMI dan BMR mereka untuk memantau kesehatan dan membantu mereka merencanakan pola makan yang tepat.

FLOWCHART PROGRAM



A. Penjelasan Flowchart

Dalam menjalankan program ini, pengguna akan diberikan pilihan apakah pengguna memiliki akun. Jika pengguna memiliki akun, pengguna dapat melakukan *log in* sesuai dengan *username* dan *password* yang telah di buat di fitur *sign in* sebelumnya. Jika pengguna tidak memiliki akun sebelumnya, pengguna dapat memilih fitur *sign in* dan meng-inputkan *username* baru dan *password*. Setelah melakukan *submit* pada menu *sign in*, pengguna dapat melakukan *log in* dengan masukkan *username* dan *password* yang telah didaftarkan sebelumnya pada fitur *sign in*. Ketika input *username* dan *password* benar pengguna akan diarahkan ke halaman utama BMI atau BMR. Namun, ketika input yang dimasukkan salah, program akan kembali ke menu *log in*.

Ketika pengguna masuk ke halaman utama, pengguna dapat memilih opsi untuk menggunakan fitur BMI atau BMR

1. BMI

Ketika pengguna memilih fitur BMI, pengguna dapat memasukan usia, jenis kelamin, tinggi badan, dan berat badan. Setelah semua poin di-inputkan, pengguna dapat memilih *button calculate* sehingga diperoleh hasil BMI yang memberikan kategori apakah pengguna masuk ke kategori *Underweight*, *Normal*, *Overweight*, atau *Obesity*. Ketika pengguna telah mendapatkan hasil kategori BMI, program juga akan memberikan rekomendasi olahraga yang sesuai dengan kebutuhan tubuh pengguna sesuai dengan kondisi BMI yang dihasilkan. Selain *option* untuk melakukan *calculate* BMI, disediakan *option reset* agar pengguna dapat memasukkan kembali data yang benar. Selain itu terdapat *option back* untuk mengembalikan program pada halaman utama BMI atau BMR.

2. BMR

Ketika pengguna memilih fitur BMR, pengguna dapat memasukan usia, tinggi badan, berat badan, dan jenis kelamin (laki-laki/perempuan). Ketika pengguna tidak ingin menggunakan program BMI, pengguna dapat memilih *option back* dan program akan mereset kembali lagi ke halaman utama BMI atau BMR. Ketika pengguna telah memasukan semua *input* yang diperlukan Pengguna dapat memilih *calculate* dan program akan menghitung hasil BMR. Ketika pengguna adalah pria maka akan terlihat hasil dari BMR dan program akan merekomendasikan makanan yang cocok bagi pengguna

pria dan besar kalorinya. Namun, ketika pengguna adalah wanita maka program akan menampilkan output hasil dari BMR dan program akan merekomendasikan makanan yang cocok bagi pengguna dan besar kalorinya. Kemudian, pengguna dapat memilih menu *exit* untuk keluar dari program.

B. Fitur Program

Pada program ini terdapat beberapa fitur yang diunggulkan. Melalui program ini, pengguna dapat melakukan :

1. *Sign in /Log in* Pengguna

Pada program yang dibuat oleh perancang, pengguna dapat melakukan *sign in* terlebih dahulu sebelum berlanjut ke menu berikutnya. Pada menu *sign in* ini, pengguna dapat melakukan pendaftaran akun terlebih dahulu dengan memasukkan data seperti nama, *username*, serta *password*. Kemudian pengguna dapat melakukan *log in* menggunakan akun yang telah didaftarkan pengguna pada menu *sign in* sebelumnya. Setelah melakukan *log in*, pengguna diarahkan ke halaman utama BMI atau BMR .

2. Menghitung BMI

Pada fitur ini pengguna dapat menghitung BMI dengan memasukan usia, tinggi badan, berat badan, dan pengguna harus memilih jenis kelamin pria atau wanita. Melalui fitur ini pengguna dapat mengetahui hasil perhitungan BMI sesuai klasifikasi yang telah ditentukan dari program. Program BMI ini akan memberikan rekomendasi olahraga beserta kalori yang dibakar serta durasi olahraga yang di rekomendasikan.

3. Menghitung BMR

Pada fitur ini pengguna dapat menghitung hasil BMR dengan memasukan jenis kelamin, usia, berat badan, dan tinggi badan. Melalui fitur ini pengguna dapat mengetahui hasil perhitungan BMR sesuai klasifikasi serta akan memberikan rekomendasi makanan dan besar kalori yang dapat dijadikan referensi jumlah asupan makanan tiap harinya.

BAB III

KODE PROGRAM

Pada Bab ini menjelaskan mengenai kode program yang digunakan dalam program BMI dan BMR Calculator.

A. Kode Aplikasi Utama

```
main.py > ...
1  import tkinter as tk
2  from PIL import Image, ImageTk
3  import fixbisa as fb
4
5  window = tk.Tk()
6  window.title("BMI Calculator")
7  window.state('zoomed')
8
9  bg = tk.PhotoImage(file='firstpage.png')
10 bg1=Image.open('firstpage.png')
11 resize = bg1.resize((1300,650),Image.LANCZOS)
12 firstpage= ImageTk.PhotoImage(resize)
13
14 bg = tk.PhotoImage(file='bglogin.png')
15 bg2=Image.open('bglogin.png')
16 resize = bg2.resize((1300,650),Image.LANCZOS)
17 bgloginpage= ImageTk.PhotoImage(resize)
18
19 bg = tk.PhotoImage(file='bgsignup.png')
20 bg3=Image.open('bgsignup.png')
21 resize = bg3.resize((1300,650),Image.LANCZOS)
22 bgsignuppage= ImageTk.PhotoImage(resize)
23
24 bg = tk.PhotoImage(file='awalkalku.png')
25 bg4=Image.open('awalkalku.png')
26 resize = bg4.resize((1300,650),Image.LANCZOS)
27 awalkalkupage= ImageTk.PhotoImage(resize)
28
29 bg = tk.PhotoImage(file='kalkulator.png')
```

Gambar 3.1

```
main.py > ...
28
29  bg = tk.PhotoImage(file='kalkulator.png')
30  bg5=Image.open('kalkulator.png')
31  resize = bg5.resize((1300,650),Image.LANCZOS)
32  bmibmr= ImageTk.PhotoImage(resize)
33
34  fb.awal(window,
35          firstpage,
36          bgloginpage,
37          bgsignuppage,
38          awalkalkupage,
39          bmibmr)
40  window.mainloop()
```

Gambar 3.2

Kode di atas merupakan kode dari aplikasi utama. Aplikasi BMI dan BMR Calculator menggunakan import modul tkinter untuk menampilkan *Graphic User Interface*. Kemudian terdapat modul PIL untuk memasukkan gambar ke dalam program. Selanjutnya aplikasi tersebut juga meng-*import* modul yang dibuat sendiri berisikan fungsi-fungsi untuk menjalankan program BMI dan BMR Kalkulator.

B. Kode Modul Utama

```
fixbisa.py > signin
1  import tkinter as tk
2  import pandas as pd
3  from tkinter import *
4  from tkinter import messagebox
5  from PIL import Image, ImageTk
6  import csv
```

Gambar 3.3

Gambar 3.3 menampilkan kode dari modul 'fixbisa.py' yang menampilkan fungsi-fungsi dalam menjalankan program BMI dan BMR *Calculator*. Pada *line* 1 menampilkan import modul *tkinter* dari *python* untuk menampilkan *Graphic User Interface* (GUI). Selanjutnya 2 *import* modul *pandas* yang digunakan untuk membaca *csv*. *Line* 2, 3, dan 4 menampilkan *import* pada modul *tkinter* untuk *messagebox*. Selanjutnya *line* 5 untuk meng-*import* gambar dan *line* 6 untuk meng-*import* database *CSV*.

```
def awal(window, firstpage, bgloginpage, bgsignuppage, awalkalkupage, bmibmr):
    frame1=tk.Frame(window,bg='#8FBC8F')
    frame1.pack(ipadx=1300,ipady=650)
    a= tk.Label(frame1,image=firstpage,border=0)
    a.pack(fill=tk.BOTH,expand=tk.YES)
    button1 = tk.Button(a, text="Sign Up", bg='#fabb17', fg='white', border=0, font=('Arial',18),
                        command=lambda: halaman_Signup(frame1,window,bgsignuppage,bgloginpage,awalkalkupage,bmibmr))
    button1.place(x=495,y=450,relwidth=0.1)
    button2 = tk.Button(a, text="Log In", bg='#fabb17', fg='white', border=0, font=('Arial',18),
                        command=lambda: halaman_SignIn(frame1,window,bgloginpage,awalkalkupage,bmibmr))
    button2.place(x=650,y=450,relwidth=0.1)
```

Gambar 3.4

```
def signin(user_name, user_password, frame, window, awalkalupage, bmibmr):
    username=user_name.get()
    password=user_password.get()

    with open('database.csv','r') as file:
        csv_reader=csv.reader(file)
        for row in csv_reader:
            if row[1]==username and row[2]==password:
                messagebox.showinfo('sign in', 'Log In Berhasil')
                halaman_awal(frame, window, awalkalupage, bmibmr)
                return
            messagebox.showerror('invalid', 'Username atau Password Salah')
```

Gambar 3.5

Gambar 3.5 menampilkan kode untuk *sign in/log in* dengan input *username* dan *password*. Kode ini juga mengambil data dari *database* dalam bentuk *csv* dengan format nama 'database.csv'. Jika input yang dimasukkan sesuai, maka akan muncul *messagebox* yang menginformasikan apakah *sign in* berhasil atau tidak.

```
def halaman_SignIn(frame, window, bgloginpage, awalkalupage, bmibmr):
    frame.pack_forget()
    frame2=tk.LabelFrame(window)
    frame2.pack(ipadx=1300, ipady=650)
    bglogin= tk.Label(frame2, image=bgloginpage, border=0)
    bglogin.pack(fill=tk.BOTH, expand=tk.YES)
    username_label = Label(bglogin, text="Username", fg="#94221e", bg="#f3e1d1", font=('The Seasons Light', 18, 'bold'))
    username_label.place(x=885, y=220)
    username_entry = tk.Entry(bglogin)
    username_entry.place(x=850, y=250, relwidth=0.15)
    password_label = Label(bglogin, text="Password", fg="#94221e", bg="#f3e1d1", font=('The Seasons Light', 18, 'bold'))
    password_label.place(x=885, y=290)
    password_entry = tk.Entry(bglogin)
    password_entry.place(x=850, y=320, relwidth=0.15)
    login_button = tk.Button(bglogin, text="Login", command=lambda:signin(username_entry, password_entry, frame2, window, awalkalupage, bmibmr),
                             bg='#fabb17', fg='white', border=0, font=('Arial', 14))
    login_button.place(x=885, y=370, relwidth=0.1)
```

Gambar 3.6

Gambar 3.6 menampilkan kode untuk halaman *sign in* yang berada pada frame 2. Pada halaman *sign in* terdapat label dan *entry* untuk *username* dan *password*. Pada halaman *sign in* juga terdapat tombol *log in* untuk selanjutnya ke halaman awal kalkulator.

```
def signup(user_name_entry, user_name, user_password, frame, window, converted1, converted2, converted3):
    name=user_name_entry.get()
    username=user_name.get()
    password=user_password.get()
    if user_name_entry.get()==' ' or user_name.get()==' ' or user_password.get()==' ':
        messagebox.showerror('invalid', 'Mohon Isi Semual')
    else:
        with open('database.csv', 'a', newline='') as file:
            csv_writer=csv.writer(file)
            csv_writer.writerow([name, username, password])
            messagebox.showinfo('Sign Up', 'Sign Up Berhasil')
            halaman_SignIn(frame, window, converted1, converted2, converted3)
```


Gambar 3.7

Gambar 3.7 menampilkan fungsi untuk *sign up* dengan input nama, *username*, dan *password*. Kode ini juga berfungsi untuk memasukkan input ke dalam database *csv* yang telah dibuat dengan format nama 'database.csv'. Jika input yang dimasukkan lengkap, maka akan muncul *messagebox* yang menginformasikan apakah *sign in* berhasil atau tidak.

```
def halaman_Signup(frame,window,converted1,converted2,converted3,converted4):
    frame.pack_forget()
    frame3=tk.LabelFrame(window)
    frame3.pack(ipadx=1300,ipady=650)
    bgsignup= tk.Label(frame3,image=converted1,border=0)
    bgsignup.pack(fill=tk.BOTH,expand=tk.YES)
    email_label = Label(bgsignup,text="Nama", fg="#94221e", bg="#f3e1d1",font=('Arial',18,'bold'))
    email_label.place(x=910,y=220)
    name_entry = tk.Entry(bgsignup)
    name_entry.place(x=850,y=250,relwidth=0.15)
    user1_label = Label(bgsignup,text="Username", fg="#94221e", bg="#f3e1d1",font=('Arial',18,'bold'))
    user1_label.place(x=885,y=280)
    user1_entry = tk.Entry(bgsignup)
    user1_entry.place(x=850,y=310,relwidth=0.15)
    password1_label = Label(bgsignup,text="Password", fg="#94221e", bg="#f3e1d1",font=('Arial',18,'bold'))
    password1_label.place(x=885,y=340)
    password1_entry = tk.Entry(bgsignup)
    password1_entry.place(x=850,y=370,relwidth=0.15)
    signup_button = tk.Button(bgsignup,text="SignUp",
                              command=lambda:signup(name_entry,user1_entry,password1_entry,frame3>window,converted2,converted3,converted4),
                              bg='#fabb17',fg='white',border=0,font=('Arial',14))
    signup_button.place(x=883,y=440,relwidth=0.1)
```

Gambar 3.8

Gambar 3.8 menampilkan kode fungsi untuk halaman *sign in* yang terdapat pada *frame* 3. Pada halaman *sign in* ini terdapat label dan entry untuk nama, *username*, dan *password*. Selanjutnya terdapat tombol untuk *sign in* untuk selanjutnya masuk ke halaman *sign in* atau *log in*.

```
def halaman_awal(frame, window, awalkalkupage, bmibmr):
    frame.pack_forget()
    frame4=tk.LabelFrame(window)
    frame4.pack(ipadx=1300,ipady=650)
    frame4.configure(bg='#8F8C8F')
    bghal_awal= tk.Label(frame4,image=awalkalkupage,border=0)
    bghal_awal.pack(fill=tk.BOTH,expand=tk.YES)
    button1 = tk.Button(bghal_awal,text='BMI Calculator',command=lambda:open_bmi(frame4>window,bmibmr,awalkalkupage),
                        bg='#fabb17',fg='white',border=0,font=('Helvetica',14))
    button1.place(x=450,y=350)
    button2 = tk.Button(bghal_awal,text='BMR Calculator',command=lambda:open_bmr(frame4>window,bmibmr,awalkalkupage),
                        bg='#fabb17',fg='white',border=0,font=('Arial',14))
    button2.place(x=680,y=350)
```

Gambar 3.9

Gambar 3.9 menampilkan kode fungsi untuk halaman awal yaitu halaman yang menampilkan pilihan untuk *BMI Calculator* atau *BMR Calculator*. Halaman awal terletak di *frame* 4 dan menampilkan tombol *BMI Calculator* dan tombol *BMR Calculator*.

```

def bmi_index(bmi):
    if bmi < 18.5:
        with open("olahraga_bmi.csv",'r') as file:
            csv_reader = csv.reader(file)
            data=list(csv_reader)

            selected_data = [ ]
            for row in data:
                selected_data.append((row[0],row[1],row[2]))
            message = ''
            for row in selected_data:
                message += ':'.join(row) + "\n"
            messagebox.showinfo(f'BMI = {bmi} is Underweight',message)

    elif (bmi>18.5) and (bmi<24.9):
        with open("olahraga_bmi.csv",'r') as file:
            csv_reader = csv.reader(file)
            data=list(csv_reader)

            selected_data = [ ]
            for row in data:
                selected_data.append((row[3],row[4],row[5]))
            message = ''
            for row in selected_data:
                message += ':'.join(row) + "\n"
            messagebox.showinfo(f'BMI = {bmi} is Normal',message)

```

Gambar 3.10

```

elif (bmi>18.5) and (bmi<24.9):
    with open("olahraga_bmi.csv",'r') as file:
        csv_reader = csv.reader(file)
        data=list(csv_reader)

        selected_data = [ ]
        for row in data:
            selected_data.append((row[3],row[4],row[5]))
        message = ''
        for row in selected_data:
            message += ':'.join(row) + "\n"
        messagebox.showinfo(f'BMI = {bmi} is Normal',message)

elif (bmi>24.9) and (bmi <29.9):
    with open("olahraga_bmi.csv",'r') as file:
        csv_reader = csv.reader(file)
        data=list(csv_reader)

        selected_data = [ ]
        for row in data:
            selected_data.append((row[6],row[7],row[8]))
        message = ''
        for row in selected_data:
            message += ':'.join(row) + "\n"
        messagebox.showinfo(f'BMI = {bmi} is Overweight',message)

```

Gambar 3.11

```

else:
    with open("olahraga_bmi.csv",'r') as file:
        csv_reader = csv.reader(file)
        data=list(csv_reader)

        selected_data = [ ]
        for row in data:
            selected_data.append((row[9],row[10],row[11]))
        message = ''
        for row in selected_data:
            message += ':'.join(row) + "\n"
        messagebox.showinfo(f'BMI = {bmi} is Obesity',message)

```

Gambar 3.12

Pada gambar 3.10, 3.11, dan 3.12 menampilkan fungsi untuk BMI *Index* yang menjelaskan mengenai batas dari nilai BMI untuk menentukan kategori dari BMI *Index* tersebut yaitu *underweight*, *normal*, *overweight*, atau *obesity*. Di setiap kategori menampilkan output berupa *list* olahraga beserta jumlah kalori yang dibakar yang diambil dari *database* dalam bentuk *csv* yang bernama *olahraga_bmi.csv*. *Output* tersebut ditampilkan dalam bentuk *messagebox*.

```

def reset_entry1(age,height,weight):
    age.delete(0,'end')
    height.delete(0,'end')
    weight.delete(0,'end')

def calculate_bmi(weight,height):
    kg = int(weight.get())
    m = int(height.get())/100
    bmi = kg/(m*m)
    bmi = round(bmi, 1)
    bmi_index(bmi)

```

Gambar 3.13

Gambar 3.13 menampilkan fungsi kode *reset* dan *calculate*. Fungsi *reset* adalah fungsi yang digunakan dalam BMI *Calculator* untuk me-reset hasil *input* yang dimasukkan pada program. Untuk fungsi *calculate* adalah fungsi yang berisi rumus untuk menghitung nilai dari BMI dengan *input* yang telah dimasukkan *user*.

```

def open_bmi(frame,window,bmibmr,awalkalkupage):
    frame.pack_forget()
    var=IntVar()
    frame5=tk.Label(window)
    frame5.pack(expand=True,ipadx=1920,ipady=1080)
    bgkalku= tk.Label(frame5,image=bmibmr,border=0)
    bgkalku.pack(fill=tk.BOTH,expand=tk.YES)
    age_lb=Label(bgkalku,text='Umur',font=('Arial',16),fg="#94221e",bg="#f3e1d1")
    age_lb.place(x=495,y=120)
    age_ent=Entry(bgkalku)
    age_ent.place(x=695,y=125)
    gen_lb=Label(bgkalku,text='Pilih Jenis Kelamin',fg="#94221e",bg="#f3e1d1",font=('Arial',14))
    gen_lb.place(x=495,y=180)
    male_rb=Radiobutton(bgkalku,text='Pria',variable=var,value=1,fg="#94221e",bg="#f3e1d1",font=('Arial',14))
    male_rb.place(x=665,y=182)
    female_rb=Radiobutton(bgkalku,text='Wanita',variable=var,value=2,fg="#94221e",bg="#f3e1d1",font=('Arial',14))
    female_rb.place(x=753,y=182)
    height_lb=Label(bgkalku,text='Tinggi Badan (cm)',bg="#f3e1d1",fg="#94221e",font=('Arial',16))
    height_lb.place(x=495,y=237)
    weight_lb=Label(bgkalku,text='Berat Badan (kg)',bg="#f3e1d1",fg="#94221e",font=('Arial',16))
    weight_lb.place(x=495,y=292)
    height=Entry(bgkalku)
    height.place(x=695,y=245)
    weight=Entry(bgkalku)
    weight.place(x=695,y=300)

cal_btn=Button(bgkalku,text='Calculate',command=lambda:calculate_bmi(weight,height),bg="#fabb17",border=1,fg='white',font=('Arial',14))
cal_btn.place(x=499,y=380,relwidth=0.07)
reset_btn=Button(bgkalku,text='Reset',command=lambda:reset_entry1(age_ent,height,weight),bg="#fabb17",border=1,fg='white',font=('Arial',14))
reset_btn.place(x=600,y=380,relwidth=0.07)
back_btn=Button(bgkalku,text='Back',bg="#fabb17",border=1,fg='white',font=('Arial',14),
                command=lambda:halaman_awal(frame5,window,awalkalkupage,bmibmr))
back_btn.place(x=699,y=380,relwidth=0.07)

```

Gambar 3.14

Gambar 3.14 menampilkan fungsi kode untuk halaman atau membuka aplikasi BMI Calculator. BMI Calculator berada di *frame* 5. Pada halaman BMI Calculator terdapat label dan *entry* untuk umur, pilihan jenis kelamin, tinggi badan dalam cm, berat badan dalam kg, dan juga tombol-tombol untuk menjalankan program ini. Tombol tersebut yaitu tombol *calculate* untuk menghitung BMI sekaligus menampilkan output berupa *messagebox*, kemudian tombol *reset* yang berguna untuk me-*reset* data yang telah di-*input* user sehingga dapat dimasukkan data lain, dan yang terakhir adalah tombol *back* untuk kembali ke halaman awal *calculator* untuk memiliki jenis kalkulator lainnya.

```

def hitung_bmr(weight,height,age,gender):
    berat_badan = float(weight.get())
    tinggi_badan = float(height.get())
    umur = int(age.get())
    gend = gender.get()

    if gend == "Pria":
        bmr = 66 + (13.75 * berat_badan) + (5 * tinggi_badan) - (6.75 * umur)
        with open("makanan_bmr.csv",'r') as file:
            csv_reader = csv.reader(file)
            data=list(csv_reader)

            selected_data = []
            for row in data:
                selected_data.append((row[0],row[1]))
            message = ''
            for row in selected_data:
                message += ':'.join(row) + "\n"
            messagebox.showinfo(f'BMR kamu adalah {bmr:.2f}',message)
    else:
        bmr = 655 + (9.56 * berat_badan) + (1.85 * tinggi_badan) - (4.68 * umur)
        with open("makanan_bmr.csv",'r') as file:
            csv_reader = csv.reader(file)
            data=list(csv_reader)

            selected_data = []
            for row in data:
                selected_data.append((row[2],row[3]))
            message = ''
            for row in selected_data:
                message += ':'.join(row) + "\n"
            messagebox.showinfo(f'BMR kamu adalah {bmr:.2f}',message)

```

Gambar 3.15

Gambar 3.15 menampilkan fungsi kode untuk menghitung nilai BMR dengan hasil input dari berat badan, tinggi badan, umur, dan jenis kelamin. Selanjutnya kode tersebut juga menampilkan rumus BMR jika *input* pria atau pun wanita dan menampilkan output dengan mengambil data dari *database* berupa *csv* yang bernama 'makanan_bmr.csv' yang kemudian ditampilkan dalam bentuk *messagebox*.

```
def reset_entry(weight,height,age,gender):
    weight.delete(0,'end')
    height.delete(0,'end')
    age.delete(0,'end')
    gender.delete(0,'end')
```

Gambar 3.16

Gambar 3.16 menampilkan fungsi kode *reset* hasil input dari *BMR Calculator* sehingga *user* dapat meng-input data ulang.

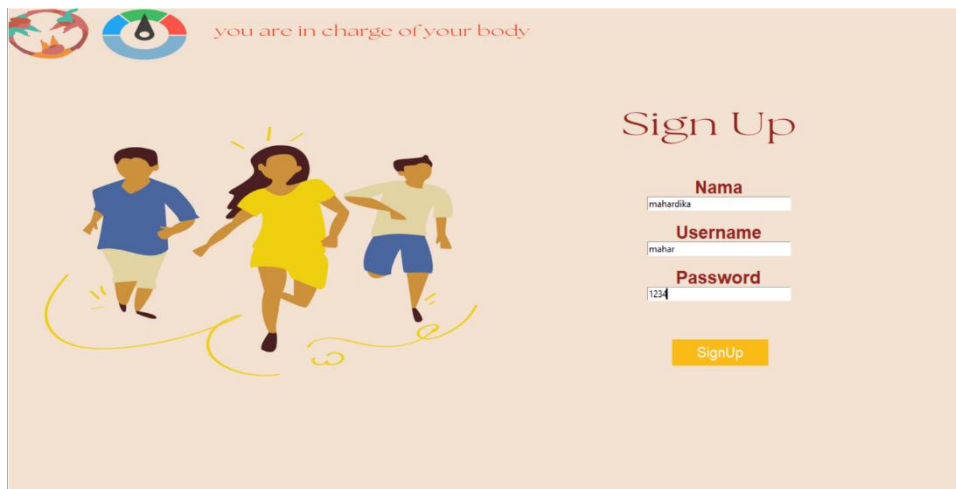
```
disapy / open_bmr
def open_bmr(frame,window,bmibmr,awalkalkupage):
    frame.pack_forget()
    frame6=tk.Label(window)
    frame6.pack(expand=True,ipadx=600,ipady=400)
    bgkalku2= tk.Label(frame6,image=bmibmr,border=0)
    bgkalku2.pack(fill=tk.BOTH,expand=tk.YES)
    label_berat=Label(bgkalku2,text='Berat Badan (kg)',bg='#f3e1d1',fg='#94221e',font=('Arial',18))
    label_berat.place(x=495,y=120)
    input_berat=Entry(bgkalku2)
    input_berat.place(x=710,y=125)
    label_tinggi = Label(bgkalku2, text="Tinggi Badan (cm)",bg='#f3e1d1',fg='#94221e',font=('Arial',18))
    label_tinggi.place(x=495,y=180)
    input_tinggi = Entry(bgkalku2)
    input_tinggi.place(x=710,y=184)
    label_umur = Label(bgkalku2, text="Umur",bg='#f3e1d1',fg='#94221e',font=('Arial',18))
    label_umur.place(x=495,y=237)
    input_umur = Entry(bgkalku2)
    input_umur.place(x=710,y=245)
    label_gender = Label(bgkalku2, text="Jenis Kelamin",bg='#f3e1d1',fg='#94221e',font=('Arial',18))
    label_gender.place(x=495,y=292)
    input_gender = Entry(bgkalku2)
    input_gender.place(x=710,y=300)
    tombol_hitung = Button(bgkalku2, text="Hitung BMR", command=lambda:hitung_bmr(input_berat,input_tinggi,input_umur,input_gender),
        fg='white',bg='#fabb17',font=('Arial',14))
    tombol_hitung.place(x=690,y=380)
    reset_btn=Button(bgkalku2,text='Reset',command=lambda:reset_entry(input_berat,input_tinggi,input_umur,input_gender),
        bg='#fabb17',border=1,fg='white',font=('Arial',14))
    reset_btn.place(x=495,y=380)
    back_btn=Button(bgkalku2,text='Back',bg='#fabb17',border=1,fg='white',font=('Arial',14),
        command=lambda:halaman_awal(frame6.window.awalkalkupage,bmibmr))
```

Gambar 3.17

Gambar 3.17 menampilkan fungsi kode untuk membuka halaman *BMR Calculator* yang terdapat pada *frame 6*. Pada *BMR Calculator* terdapat *label* dan *entry* untuk *input* berat badan dalam kg, tinggi badan dalam cm, umur user, dan jenis kelamin user. Selanjutnya terdapat tombol-tombol untuk menjalankan program *BMR Calculator*. Tombol pertama adalah tombol untuk menghitung BMR, selanjutnya tombol *reset* untuk me-*reset* data sehingga *user* dapat memasukkan data ulang, dan tombol *back* untuk kembali ke halaman awal untuk memilih jenis kalkulator lainnya.

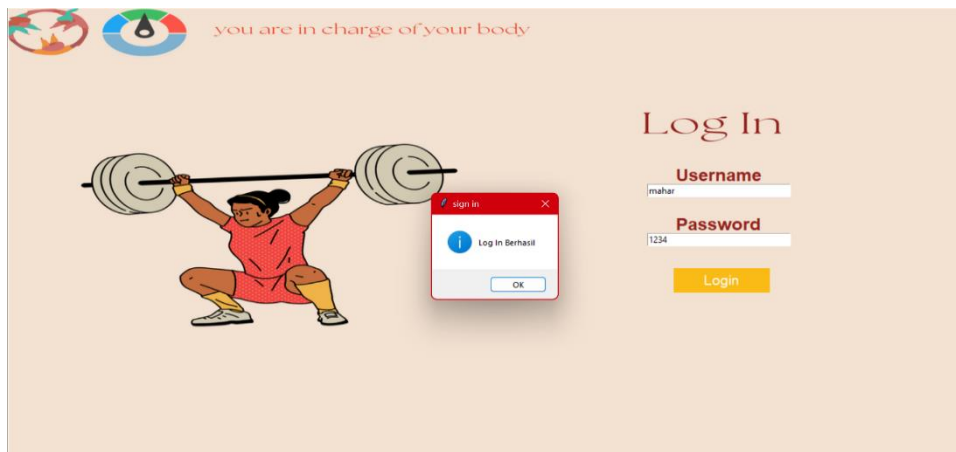
BAB IV

HASIL RUNNING



Gambar 4.1 Menu Sign up

Gambar 4.1 merupakan hasil *output* dari menu *sign up*. Ketika tombol *sign up* ditekan akan menampilkan *messagebox* yang menginformasikan apakah *sign up* berhasil atau tidak. Jika berhasil akan langsung menuju halaman selanjutnya yaitu menu *log in*.



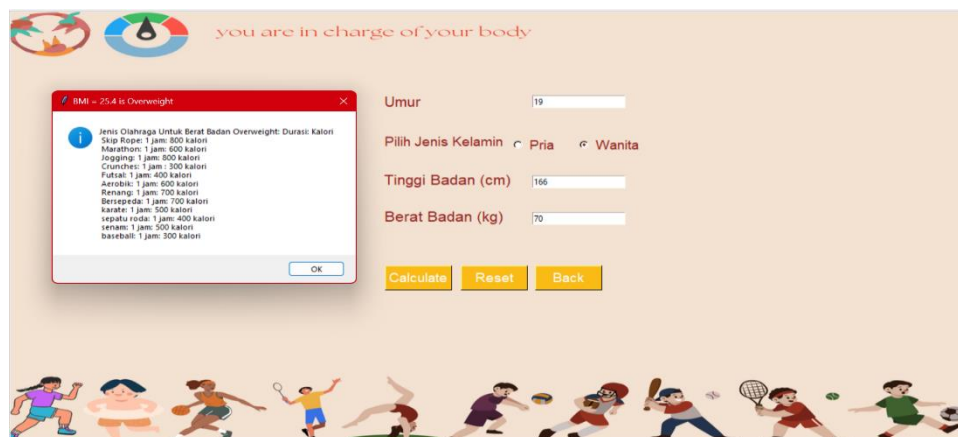
Gambar 4.2 Menu Log in

Gambar 4.2 merupakan hasil *output* dari menu *Log in* pengguna setelah dari menu *Sign up*. Jika *username* dan *password* telah sesuai dengan data yang *user* masukkan saat *sign up*, pada menu *log in* akan ditampilkan *messagebox* yang menginformasikan *log in* berhasil atau tidak. Jika *log in* berhasil akan dilanjutkan ke menu awal kalkulator.



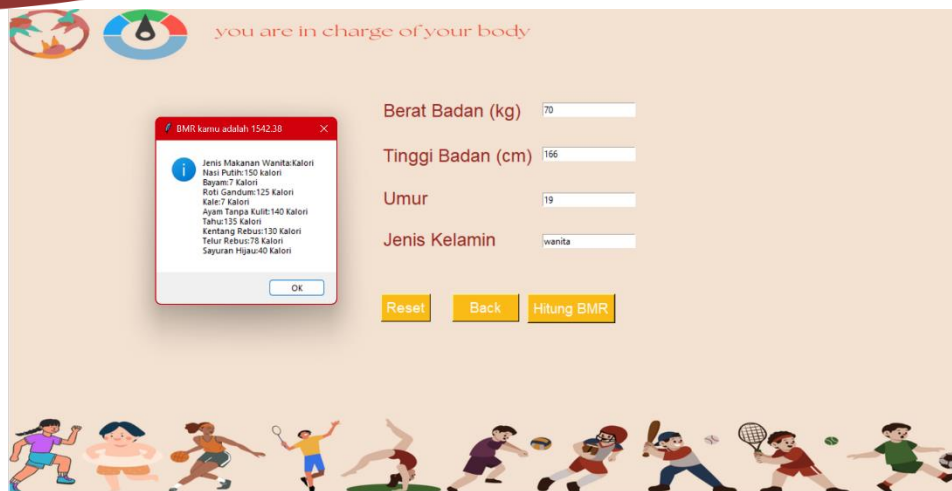
Gambar 4.3 Menu Halaman Awal

Gambar 4.3 merupakan tampilan awal untuk pemilihan *BMI Calculator* atau *BMR Calculator* sehingga *user* bebas memilih jenis kalkulator yang diinginkan.



Gambar 4.4 Menu BMI

Gambar 4.3 merupakan hasil *output* BMI setelah pengguna mengisi umur, jenis kelamin, tinggi badan, dan berat badan. *Output BMI Calculator* berupa *messagebox* yang menampilkan jenis olahraga sesuai kategori BMI beserta durasi olahraga dan jumlah kalori yang dibakar ketika melakukan rekomendasi olahraga tersebut.



Gambar 4.5 Menu BMR

Gambar 4.5 merupakan hasil *output* BMR setelah pengguna menginput berat badan, tinggi badan, umur dan jenis kelamin. *Output BMR Calculator* berupa *messagebox* yang menampilkan nilai BMR *user* beserta rekomendasi jenis makanan sesuai *gender user* dan informasi kalorinya

