



**ULTRA
XPLOERER**

PROJECT TEAM PROGRAMA KOMPUTER 2023

KELOMPOK 1

Achmad Sidiq Herman	I0322001
Ayusika Yumna Agil N.	I0322020
Brian Winson Jo	I0322025
Chanisa Ulkarimah	I0322026

BAB I

DESKRIPSI MASALAH

Bab ini membahas mengenai deskripsi masalah *Project Team* Praktikum Program Komputer.

Di masa perkuliahan, nilai merupakan salah satu hal yang cukup krusial bagi mahasiswa. Namun, sebagian besar mahasiswa mengeluh karena kesulitan untuk memantau dan melacak perkembangan nilai mereka. Sistem pemantauan nilai yang biasa digunakan seringkali melibatkan proses manual yang memakan waktu dan rentan terhadap kesalahan manusia. Oleh karena itu, kami menciptakan sebuah sistem yang dapat membantu mahasiswa dalam melakukan pemantauan dan pelacakan perkembangan nilai tanpa harus melakukan perhitungan manual.

Sistem pemantauan ini akan memberikan banyak manfaat. Sistem ini memungkinkan mahasiswa untuk melihat dan memahami perkembangan nilai mereka sepanjang semester. Dengan informasi ini, mahasiswa dapat mengetahui apakah mereka telah mencapai target akademik yang ditetapkan atau perlu meningkatkan upaya belajar mereka. Mahasiswa juga dapat melakukan refleksi terhadap metode belajar mereka. Mereka dapat melihat mata kuliah mana yang nilainya rendah dan mengidentifikasi hal-hal yang mempengaruhi performa akademik mereka. Dengan mengetahui hal ini, mereka dapat mengubah strategi belajar mereka atau mencari alternatif lain yang mungkin diperlukan.

Selain itu, sistem ini juga dapat memberikan motivasi atau dorongan bagi mahasiswa. Dengan melihat perkembangan nilai mereka secara terus-menerus, mahasiswa dapat merasa termotivasi untuk tetap berusaha dan meningkatkan prestasi akademik mereka. Mereka dapat melihat bahwa usaha mereka berdampak langsung pada peningkatan nilai, yang kemudian dapat meningkatkan kepercayaan diri dan semangat belajar.

Dalam keseluruhan, sistem pemantauan nilai mahasiswa sangatlah bermanfaat terutama untuk mahasiswa. Sistem ini membantu mahasiswa dalam melakukan refleksi strategi pembelajaran dan memberikan motivasi untuk mencapai prestasi akademik yang lebih baik. Dengan adanya program ini, mahasiswa dapat memiliki kendali yang lebih besar terhadap perkembangan nilai mereka dan memaksimalkan potensi belajar mereka.

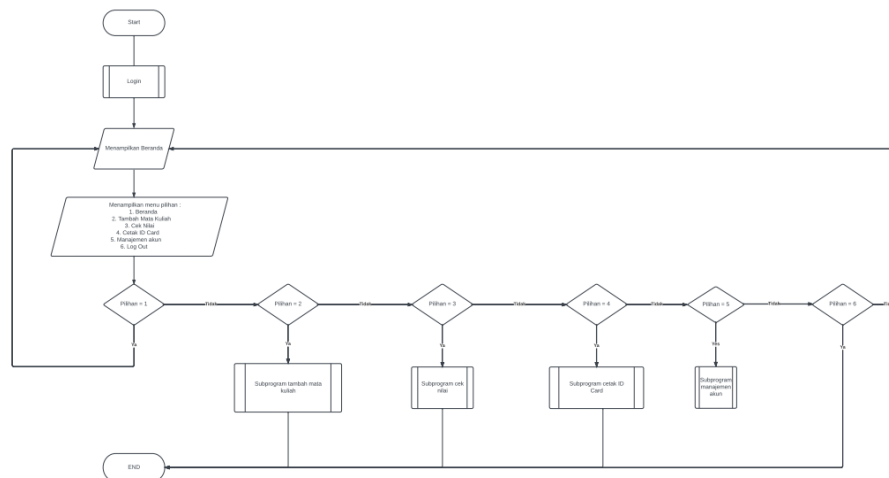
BAB II

FLOWCHART PEMROGRAMAN BESERTA PENJELASANNYA

Bab ini membahas mengenai *flowchart* pemrograman beserta penjelasannya *Project Team* Praktikum Program Komputer.

Program UltraXplorer dapat dibagi menjadi tujuh bagian yang terdiri dari *main program* dan beberapa sub program. Berikut merupakan diagram alir atau *flowchart* dari program tersebut:

1. Main Program

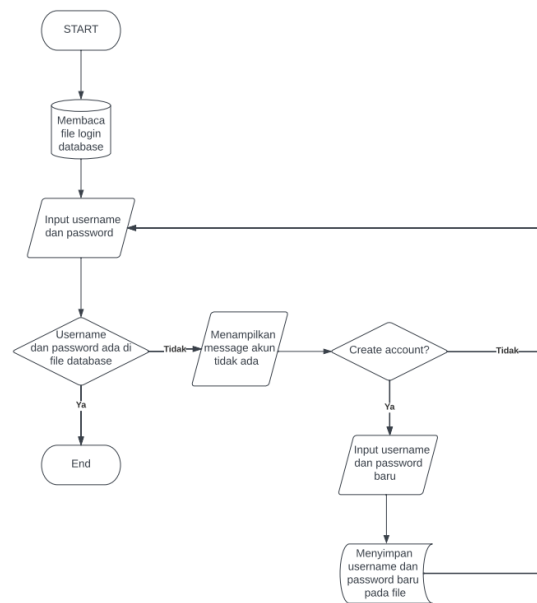


Gambaran *flowchart* main program:

- Program akan menjalankan sub-program login.
- Selanjutnya, program akan menampilkan beranda.
- Setelah beranda ditampilkan, maka program juga akan menampilkan beberapa menu pilihan.
- Dalam menu pilihan terdapat menu beranda, menu tambah mata kuliah, menu cek nilai, menu cetak ID Card, menu manajemen akun, dan menu log out.
- Ketika user memilih salah satu menu pilihan, maka program akan menjalankan sub-program tersebut.
- Ketika user memilih menu pilihan logout, maka program akan kembali menampilkan sub-program login.

2. Sub-program Login

Subprogram Login

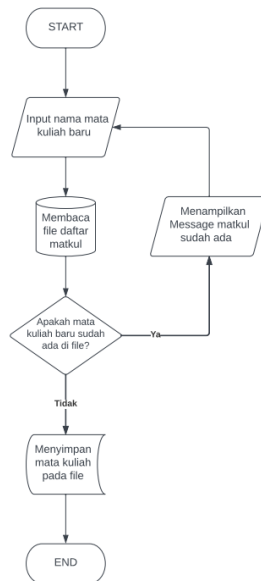


Gambaran *flowchart* sub-program login:

- Program akan membaca file login pada database.
- User akan diminta untuk menginputkan *username* dan juga *password*.
- Apabila *username* dan *password* tidak ada dalam file database, maka program akan menampilkan *message* bahwa akun tidak ada.
- Kemudian user akan diarahkan untuk *create account*.
- Jika user tidak ingin membuat akun maka akan kembali ke input *username* dan *password*.
- Jika user memilih *create account* maka user diminta untuk menginputkan *username* dan *password* baru.
- Username dan password yang telah dibuat akan disimpan pada file.
- Apabila *username* dan *password* ada dalam file database, maka user akan langsung masuk dalam menu beranda.

3. Sub-program Tambah Mata Kuliah

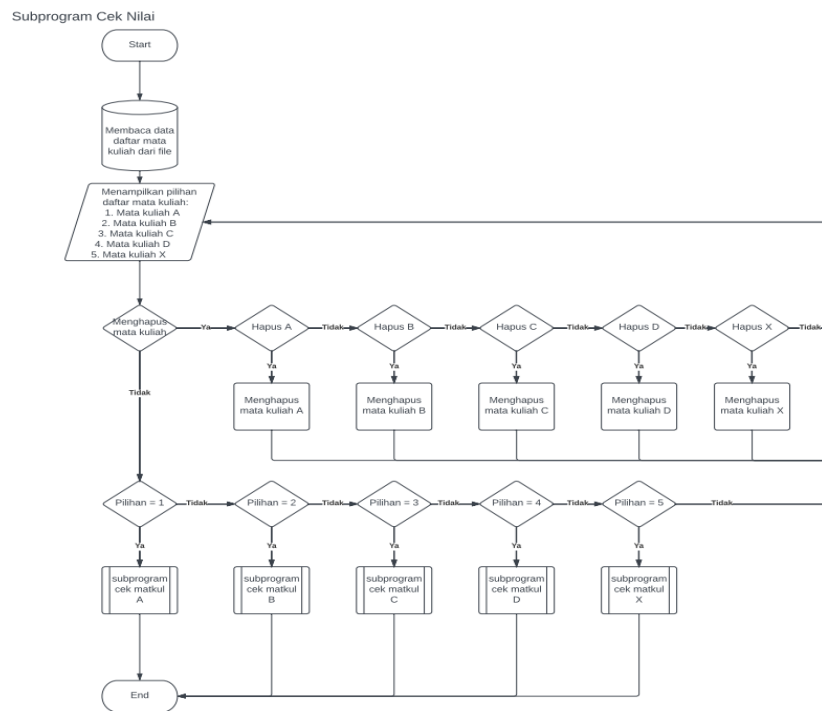
Subprogram Tambah Mata Kuliah



Gambaran *flowchart* sub-program tambah mata kuliah:

- Program akan menampilkan input nama mata kuliah baru.
- Setelah user menginputkan nama mata kuliah maka program akan membaca file daftar matkul.
- Apabila nama mata kuliah baru sudah ada di file maka program akan menampilkan *message* bahwa matkul sudah ada dan akan kembali ke input mata kuliah baru.
- Apabila nama mata kuliah baru belum ada pada file maka nama mata kuliah akan disimpan dalam file.

4. Sub-program Cek Nilai

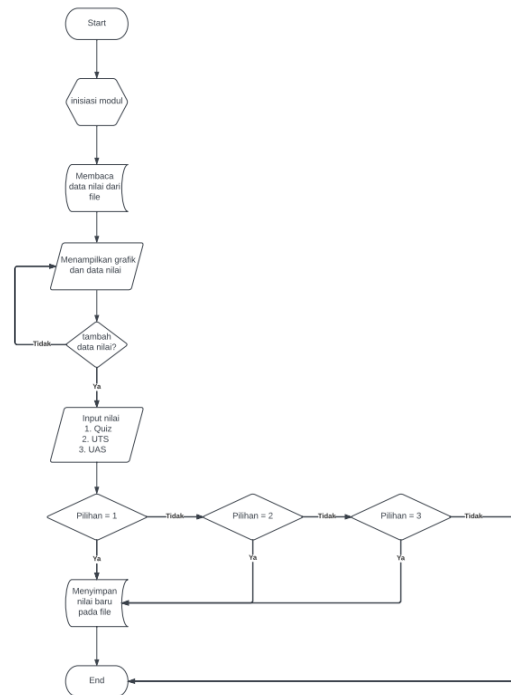


Gambaran *flowchart* sub-program cek nilai:

- Program akan membaca data daftar mata kuliah dari file.
- Kemudian program akan menampilkan beberapa pilihan daftar mata kuliah.
- Dalam menu pilihan terdapat menu mata kuliah A, menu mata kuliah B, menu mata kuliah C, menu mata kuliah X.
- User akan dihadapkan dengan menu hapus mata kuliah.
- Ketika user memilih untuk menghapus mata kuliah, maka mata kuliah akan dihapus lalu user akan kembali ke tampilan pilihan daftar mata kuliah.
- Ketika user memilih tidak ingin menghapus mata kuliah, maka user dapat memilih pilihan matkul dan program akan menjalankan sub-program tersebut.

5. Sub-program Cek Mata Kuliah X

Subprogram Cek Mata Kuliah X

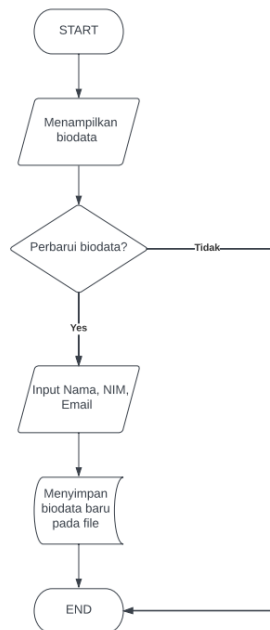


Gambaran *flowchart* sub-program cek mata kuliah x:

- Program akan menginisiasi modul.
- Program akan membaca data nilai dari file.
- Kemudian program akan menampilkan grafik dan data nilai.
- Terdapat menu tambah nilai.
- Apabila user tidak ingin menambahkan maka akan kembali ke menampilkan grafik dan data nilai.
- Apabila user ingin menambahkan nilai, maka user diminta menginput nilai pada beberapa pilihan sub-nilai yang dipilih.
- Jika nilai sudah diinput maka program akan menyimpan nilai baru pada file.

6. Sub-program Manajemen Akun

Subprogram Manajemen Akun



Gambaran *flowchart* sub-program manajemen akun:

- Program akan menampilkan biodata.
- Kemudian muncul menu perbarui biodata.
- Apabila user ingin memperbarui biodata maka user harus menginputkan nama, NIM, dan email.
- Kemudian program akan menyimpan biodata baru kedalam file.
- Apabila user tidak ingin memperbarui biodata maka user dapat menuju ke sub-program lainnya.

7. Sub-program Cetak ID Card

Subprogram Cetak ID Card



Gambaran *flowchart* sub-program cetak ID Card:

- Program akan menginisiasi template ID Card.
- Kemudian program akan membaca biodata user
- Lalu program akan memasukkan data ke dalam template.
- Program akan menampilkan ID Card.
- Apabila user ingin mencetak ID Card maka ID Card akan dikirimkan melalui email.
- Apabila user tidak ingin mencetak ID Card maka user dapat berpindah ke program lainnya.

BAB III

KODE PROGRAM DI PYTHON

Bab ini membahas mengenai kode program di Python *Project Team* Praktikum Program Komputer.

- Menampilkan menu Signin

```
from tkinter import *
import tkinter as tk
from tkinter import messagebox
import Program as md
import os

# Menampilkan menu login dan create account
def mainprogram():
    global username_value
    font_login = "Roboto Condensed"

    window = tk.Tk()
    window.title("Login")
    window.geometry("1200x675")
    window.configure(bg="White")
    window.resizable(True, True)

    img = PhotoImage(file="gambar/Modern Initial E Logo.png")
    Label(window, image=img, bg="White").place(x=175, y=125)

    frame = Frame(window, width=350, height=350, bg="White")
    frame.place(x=600, y=200)
    heading = Label(frame, text="Sign in",
fg="#57a1f8", bg="White", font=("Montserrat", 23, "bold"))
    heading.place(x=100, y=5)
    Button(frame, width=29, height=1, text="Login",
fg="white", bg="#57a1f8", border=0).place(x=60, y=155)

    login = Button(frame, width=29, text="Login",
fg="White", bg="#57a1f8", cursor="hand2", border=0, command = login)
    login.place(x=60, y=155)

    Button(frame, width=29, height=1, text="Create Account",
fg="white", bg="#57a1f8", border=0).place(x=60, y=185)

    signup = Button(frame, width=29, text="Create Account",
fg="White", bg="#57a1f8", cursor="hand2", border=0, command = signup)
    signup.place(x=60, y=185)
```

Penjelasan kode program untuk menampilkan menu Sign in:

1. Mengimpor modul yang diperlukan seperti modul tkinter dan modul Program
2. Membuat fungsi `mainprogram()` untuk tampilan utama program. Fungsi ini berfungsi untuk membuat *main window*, mengatur tampilan, dan menampilkan elemen-elemen GUI yang dibutuhkan seperti label, tombol, dan gambar.
3. Di dalam fungsi `mainprogram()`, digunakan kode `PhotoImage` untuk membaca gambar dari file dan menampilkannya di dalam label menggunakan `Label(window,image=img,bg="White")`. Gambar ini digunakan sebagai logo yang akan ditampilkan di window.
4. Selanjutnya, fungsi `mainprogram()` membuat sebuah frame di dalam window menggunakan kode `Frame(window, width=350, height=350, bg="White")`. Frame ini akan memuat elemen-elemen GUI yang berkaitan dengan login dan create account.
5. Terakhir, tombol-tombol "Login" dan "Create Account" memiliki atribut `command` yang dikaitkan dengan fungsi `login()` dan `signup()` masing-masing.

- Menampilkan menu Sign Up

```
def signup():
    global username2
    global password2
    global screen
    window.destroy()
    screen=tk.Tk()
    screen.title(f"Create Account")
    screen.geometry("925x500+300+200")
    frame = Frame(screen,width=350,height=350,bg="White")
    frame.place(x=400,y=100)
    heading = Label(frame,text="Sign Up",
fg="#57a1f8",bg="White",font=("Montserrat",23,"bold"))
    heading.place(x=100,y=5)
    back_button =
Button(screen,width=3,text="Back",font=(font_login,11,"underline"),
fg="Black",cursor="hand2",border=0,command=back_login)
    back_button.place(x=10,y=10)

    def on_enter(e):
        username2.delete(0, "end")

    def on_leave(e):
        if username2 == username2.get():
            username2.delete(0, "end")

    username2= Entry(frame,width=25, fg = "black",
border="2",bg="White",font=(font_login,11))
    username2.place(x=60,y=75)
    username2.insert(0,"username")
    username2.bind("<FocusIn>", on_enter)
```

```

username2.bind("<FocusOut>", on_leave)

=====
#
def on_enter(e):
    password2.delete(0, "end")

def on_leave(e):
    if password2 == password2.get():
        password2.delete(0, "end")

password2 = Entry(frame,width=25, fg = "black",
border="2",bg="White",font=(font_login,11))
password2.place(x=60,y=115)
password2.insert(0,"password")
password2.bind("<FocusIn>", on_enter)
password2.bind("<FocusOut>", on_leave)

=====
#

Button(frame,width=29,height=1,text="Create Account",
fg="white",bg="#57a1f8",border=0).place(x=60,y=185)

signup_button = Button(frame,width=29,text="Create Account",
fg="White",bg="#57a1f8",cursor="hand2",border=0,command = sign_up)
signup_button.place(x=60,y=185)

```

Penjelasan kode program untuk menampilkan menu Sign up:

1. Fungsi `signup()` digunakan untuk menampilkan window create account. Di awal fungsi ini, variabel-variabel global seperti `username2`, `password2`, dan `screen` dideklarasikan.
2. Jendela utama (`window`) yang telah dibuat sebelumnya dihancurkan menggunakan command `window.destroy()` agar jendela ini tidak muncul saat jendela baru dibuat.
3. Membuat jendela baru (`screen`) menggunakan `tk.Tk()` dengan judul "Create Account".
4. Di dalam jendela `screen`, dibuat frame baru dengan menggunakan kode `Frame(screen, width=350, height=350, bg="White")`. Frame ini akan memuat elemen-elemen GUI terkait create account.
5. Menambahkan elemen-elemen GUI yang dibutuhkan ke dalam frame seperti judul ("Sign Up"), tombol "Back" untuk kembali ke menu login, dan elemen input seperti field username dan password.
6. Field username (`username2`) dan field password (`password2`) menggunakan objek Entry untuk menerima input dari pengguna. Saat user memfokuskan field, teks default

akan dihapus menggunakan metode `bind()` yang terkait dengan peristiwa "FocusIn" dan "FocusOut".

7. Tombol "Create Account" ditambahkan ke dalam frame untuk mengeksekusi fungsi `sign_up()`.

- Tampilan Awal Program

```
def program():
    global username_user
    global a
    global n
    a = 0
    n = 0
    username_user = modul.Login.username_value
    global window2
    window2 = tk.Tk()
    window2.title(f"UltraXplorer {modul.Login.username_value}")
    window2.geometry("1200x675")

    img = PhotoImage(file="gambar/bg1.png")
    Label(window2, image=img, bg="White").place(x=248, y=0)
    window2.resizable(False, False)

    frame = Frame(window2, width=250, height=675, bg="#403F64", border = "1")
    frame.place(x=0, y=0)
    # Menampilkan button pada menu akun
    akun =
    Button(frame, width=28, pady=4, text=username_user, font=(font_main, 11), fg="White", bg="
    black", cursor="hand2", border=0, command=manajemen_akun0)
    akun.place(x=0, y=15)
```

Penjelasan kode program untuk tampilan awal program:

1. Fungsi `program()` digunakan untuk menampilkan tampilan utama program setelah pengguna berhasil login. Di awal fungsi ini akan mendeklarasikan variabel-variabel global seperti `username_user`, `a`, `n`, dan `window2`.
2. Variabel `username_user` diinisiasi dari `modul.Login.username_value` merupakan variabel global yang menyimpan username user yang sudah berhasil login.
3. Jendela baru (`window2`) dibuat menggunakan `tk.Tk()` dengan judul yang mencakup username user. Jendela ini memiliki ukuran tetap ("1200x675") dan tidak dapat diubah ukurannya menggunakan `window2.resizable(False, False)`.
4. Di dalam jendela `window2`, dibuat frame baru (`frame`) dengan ukuran dan posisi tertentu. Frame ini berfungsi sebagai menu akun user dan berada di sebelah kiri jendela. Frame ini memiliki tepian (border) dengan ketebalan 1.
5. Tombol `akun` ditambahkan ke dalam frame untuk menampilkan username pengguna saat ini. Tombol ini terhubung dengan fungsi `manajemen_akun0()` yang diimpor dari modul Program

- Menampilkan menu akun

```
def manajemen_akun():
    global ganti_biodata
    global frame2
    global a
    frame2 = Frame(window2,width=800,height=400,bg="#F8B410")
    frame2.place(x=350,y=100)

    def simpan_biodata():
        file_biodata = open(f"databaseuser/{username_user}/biodata.csv", "w")
        text_biodata = f"""Nama,NIM,Email
{nama.get()}, {nim.get()}, {tgllahir.get()}"""
        file_biodata.write(text_biodata)
        file_biodata.close()
        messagebox.showinfo("Berhasil","Data Biodata Telah Disimpan")
        manajemen_akun0()

    def change_biodata():
        global nama
        global nim
        global tgllahir
        ganti_biodata = Button(frame2,width=28,pady=3,text="Simpan
Biodata",font=(font_main,11),fg="White",bg="black",cursor="hand2",border=0,command=
simpan_biodata)
        ganti_biodata.place(x=100,y=260)

        nama = Entry(frame2,width=40, fg = "black",
border="1",bg="White",font=(font_main,15))
        nama.place(x=208,y=120)

        nim = Entry(frame2,width=40, fg = "black",
border="1",bg="White",font=(font_main,15))
        nim.place(x=208,y=155)

        tgllahir = Entry(frame2,width=40, fg = "black",
border="1",bg="White",font=(font_main,15))
        tgllahir.place(x=208,y=190)

        text_biodata = f"""
Nama\t\t\t :
NIM \t\t\t :
Email\t\t\t :
        """
```

```

        biodata =
Label(frame2,text=text_biodata,justify="left",fg="black",bg="#F8B410",font=(font_ma
in,12))

        biodata.place(x=50,y=100)

```

Penjelasan kode program untuk menampilkan menu akun:

1. Fungsi `manajemen_akun()` digunakan untuk menampilkan tampilan untuk manajemen akun user. Di awal fungsi ini akan mendeklarasikan variabel-variabel global seperti `ganti_biodata`, `frame2`, dan `a`.
2. Membuat frame baru (`frame2`) di dalam jendela utama (`window2`). Frame ini akan digunakan untuk menampilkan elemen-elemen yang berkaitan dengan manajemen akun.
3. Di dalam frame `frame2`, terdapat beberapa fungsi yang digunakan dalam perubahan biodata. Fungsi `simpan_biodata()` digunakan untuk menyimpan data biodata yang diubah oleh user ke dalam file csv. Data biodata diambil dari elemen-elemen Entry `nama`, `nim`, dan `email`. Setelah data disimpan, pesan informasi akan ditampilkan menggunakan command `messagebox.showinfo()`. Setelah itu, fungsi `manajemen_akun0()` akan dipanggil untuk kembali ke tampilan manajemen akun awal.
4. Fungsi `change_biodata()` digunakan untuk mengatur tampilan dan elemen-elemen GUI yang berkaitan dengan perubahan biodata. Tombol `ganti_biodata` ditambahkan ke frame untuk mengeksekusi fungsi `simpan_biodata()`. Entry fields (`nama`, `nim`, dan `email`) juga ditambahkan untuk mengambil input dari user. Selain itu, label `biodata` ditampilkan untuk menampilkan teks yang menjelaskan bidang-bidang yang harus diisi.

- Menampilkan menu tambah mata kuliah

```

def tambahmatkul():
    global frame2
    global a
    def simpan_matkul():
        matkul_value = input_matkul.get()
        v = 0
        w = 0
        if matkul_value == "":
            messagebox.showerror("error","nama matkul tidak boleh kosong")
            v = 1
            tambahmatkul0()
        else:
            with open(f"databaseuser/{username_user}/nilai quiz.csv") as
matkul:

                matkul = csv.reader(matkul,delimiter=",")
                matkul = list(matkul)
                for i in matkul:
                    w +=1

```

```

        if i[0].lower() == matkul_value.lower():
            messagebox.showinfo("error", "Nama mata kuliah sudah
ada")

            tambahmatkul0()
            break
        else:
            v += 1

    if v == w:
        file_nilai = open(f"databaseuser/{username_user}/nilai
quiz.csv", "a")
        file_nilai.write(f"{matkul_value}\n")
        file_nilai.close()

        file_nilai = open(f"databaseuser/{username_user}/nilai
tugas.csv", "a")
        file_nilai.write(f"{matkul_value}\n")
        file_nilai.close()

        file_nilai = open(f"databaseuser/{username_user}/nilai
UTS-UAS.csv", "a")
        file_nilai.write(f"{matkul_value}\n")
        file_nilai.close()

        messagebox.showinfo("Berhasil", "Mata kuliah telah ditambahkan")
        tambahmatkul0()

a = 1

frame2 = Frame(window2, width=630, height=250, bg="#F8B410")
frame2.place(x=400, y=200)

label_matkul = Label(frame2, text="Masukkan Mata Kuliah Baru:",
fg="black", bg="#F8B410", font=(font_main, 15, "bold"))
label_matkul.place(x=40, y=80)

input_matkul = Entry(frame2, width=40, fg="black",
border="1", bg="White", font=(font_main, 15))
input_matkul.place(x=40, y=120)

inputmatkulbutton = Button(frame2, width=40, pady=3, text="Tambah Mata
Kuliah", font=(font_main, 11), fg="White", bg="#403F64", cursor="hand2", border=0, command
=simpan_matkul)
inputmatkulbutton.place(x=40, y=160)

```


Penjelasan kode program untuk menampilkan menu tambah mata kuliah:

1. Fungsi `tambahmatkul()` digunakan untuk menampilkan tampilan dan fungsi terkait penambahan mata kuliah. Di awal fungsi ini akan mendeklarasikan variabel-variabel global seperti `frame2` dan `a`.
2. Fungsi `simpan_matkul()` digunakan untuk menyimpan mata kuliah baru yang diinput oleh user. Validasi dilakukan untuk memastikan bahwa nama mata kuliah tidak kosong dan tidak ada duplikat nama mata kuliah yang sudah ada sebelumnya. Jika terdapat kesalahan validasi, pesan error akan ditampilkan menggunakan `messagebox.showerror()` atau `messagebox.showinfo()`.
3. Jika mata kuliah sudah divalidasi, data mata kuliah akan ditambahkan ke tiga file csv yang berbeda sesuai input user (`nilai quiz.csv`, `nilai tugas.csv`, dan `nilai UTS-UAS.csv`).
4. Membuat frame baru (`frame2`) dalam jendela utama (`window2`). Frame ini akan digunakan untuk menampilkan elemen-elemen yang berkaitan dengan penambahan mata kuliah.
5. Elemen-elemen yang dibutuhkan ditambahkan ke dalam frame. Label `label_matkul` digunakan untuk menampilkan teks instruksi kepada user. Entry field (`input_matkul`) digunakan untuk mengambil input mata kuliah yang akan ditambahkan dari user. Button `inputmatkulbutton` digunakan untuk mengeksekusi fungsi `simpan_matkul()` saat diklik.

- Menampilkan menu cek nilai

```
def ceknilai():
    global frame2
    global a

    frame2 = Frame(window2,width=950,height=575,bg="#F8B511")
    frame2.place(x=250,y=50)
    with open(f"databaseuser/{username_user}/nilai quiz.csv") as nilai:
        nilai = csv.reader(nilai,delimiter=",")
        nilai = list(nilai)
        letak = 50
        n = 0
        for i in nilai[1:]:
            n += 1
            matkul =
            Button(frame2,text=i[0],width=50,justify="left",fg="white",bg="#403F64",font=(font_
            main,12),border=0,command=lambda n=nilai[n:n+1]: tabell(n))
            matkul.place(x=50,y=letak)
            hapus =
            Button(frame2,text="X",width=3,justify="left",fg="white",bg="red",font=(font_main,1
            2),border=0,command=lambda n=nilai[n:n+1]: hapus_matkul(n))
            hapus.place(x=520,y=letak)
            letak += 50
```

```
if letak == 50:
    Label(frame2,width=50,justify="left",text="silahkan tambah mata kuliah
terlebih
dahulu",fg="white",bg="#403F64",font=(font_main,12),border=0).place(x=50,y=50)

a = 1
```

Penjelasan kode program untuk menampilkan menu cek nilai:

1. Fungsi `ceknilai()` digunakan untuk menampilkan tampilan dan fungsi terkait pengecekan nilai mata kuliah. Di awal fungsi ini akan mendeklarasikan variabel-variabel global seperti `frame2` dan `a`.
2. Membuat frame baru (`frame2`) dalam jendela utama (`window2`). Frame ini akan digunakan untuk menampilkan daftar mata kuliah dan nilai yang terkait dengan akun user.
3. Data nilai mata kuliah pengguna diambil dari file csv yang sudah ada sebelumnya. Data ini kemudian ditampilkan dalam bentuk tombol dalam frame.
4. Tombol "X" (hapus) juga ditambahkan untuk setiap mata kuliah. Tombol ini memiliki fungsi untuk menghapus mata kuliah dari daftar nilai pengguna saat diklik.
5. Jika tidak ada mata kuliah yang tersedia, label "silahkan tambah mata kuliah terlebih dahulu" akan ditampilkan di frame.

- Modul untuk membuat tabel

```
from tabulate import tabulate

def tabel(data_nilai):
    global table
    angka = 0
    rata_rata = 0
    for i in data_nilai[0][1:]:
        angka = float(i) + angka
        rata_rata = angka/len(data_nilai[0][1:])
    data_nilai[0].append(str(rata_rata))
    header = ["mata kuliah", "Nilai1", "Nilai2", "Nilai3", "Nilai4", "Nilai5"]
    header = header[0:len(data_nilai[0][1:])]
    header.append("Rata-rata")
    table = tabulate(data_nilai, headers=header, tablefmt="grid")
```

Penjelasan kode program modul untuk membuat tabel:

1. Modul `tabulate` diimpor untuk melakukan akses langsung ke fungsi `tabulate` dari modul tersebut.
2. Fungsi `tabel(data_nilai)` didefinisikan dan menerima parameter `data_nilai`, yang berisi data nilai mata kuliah yang akan ditampilkan dalam tabel.
3. Dalam fungsi `tabel`, dilakukan perhitungan rata-rata nilai dari data yang disimpan. Nilai rata-rata kemudian ditambahkan ke baris data pertama (`data_nilai[0]`) dengan menggunakan metode `append`.

4. Variabel `header` digunakan untuk menentukan judul kolom yang mencakup nama mata kuliah dan nilai-nilai yang ada di dalam tabel
5. Fungsi `tabulate` dipanggil dengan parameter `data_nilai`, `headers`, dan `tablefmt`. `data_nilai` berisi data nilai mata kuliah yang telah diperbarui dengan rata-rata. `headers` berisi judul kolom tabel. `tablefmt` menentukan format tabel yang diinginkan yaitu grid.
6. Tabel kemudian disimpan dalam variabel `table`. Tabel ini akan digunakan untuk menampilkan nilai kepada user.

- Modul untuk membuat grafik

```
import csv
import matplotlib.pyplot as plt
from matplotlib.figure import Figure
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

nilai = ["23", "45", "23",]

# Data contoh
def grafik(nilai):
    global fig
    nilai = nilai[0][1:]
    nilai_int = []
    for i in nilai:
        nilai_int.append(float(i))
    header = []
    for j in range(len(nilai)):
        header2 = "nilai" + str(j+1)
        header.append(header2)

    # Plotting nilai

    # Konfigurasi plot
    plt.title('Monitoring Nilai Quiz')
    plt.ylabel('Nilai')

    # Tampilkan plot

    fig = Figure(figsize=(7, 3), dpi=100)
    ax = fig.add_subplot(111)
    ax.plot(header, nilai_int, marker='o', linestyle='-', color='b')

if __name__ == "__main__":
    grafik(nilai)
```

Penjelasan kode program modul untuk membuat grafik:

1. Mengimpor pustaka yang dibutuhkan, yaitu pustaka `csv` untuk membaca data dari file CSV, pustaka `matplotlib.pyplot`, kelas `Figure` dan `FigureCanvasTkAgg` dari `matplotlib.backends.backend_tkagg` untuk membuat dan menampilkan plot grafik.
2. Data nilai yang akan digunakan untuk membuat grafik disimpan dalam variabel `nilai`.
3. Fungsi `grafik(nilai)` didefinisikan untuk mengambil data nilai sebagai parameter.
4. Di dalam fungsi `grafik`, nilai-nilai dalam `nilai` diubah menjadi tipe data `float` dan disimpan dalam `nilai_int`.
5. Membuat objek `Figure` dan menambahkan objek `ax` (sumbu) kedalamnya
6. Membuat plot grafik menggunakan `ax.plot` dengan parameter `header` sebagai sumbu x dan `nilai_int` sebagai sumbu y. Plot ini menggunakan marker "o" untuk menandai titik data dan garis putus-putus untuk menghubungkan titik-titik tersebut.

- Modul untuk mengirimkan file melalui Email

```
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.image import MIMEImage
from email.mime.text import MIMEText
import os

def send_png_email(receiver_email, png_file_path):

    sender_email = os.environ.get('DB_USER')
    sender_password = os.environ.get('DB_PASS')
    smtp_server = 'smtp.gmail.com'
    smtp_port = 587

    # Inisialisasi objek MIMEMultipart
    message = MIMEMultipart()
    message['From'] = sender_email
    message['To'] = receiver_email
    message['Subject'] = "ID Card UltraXplorer"

    # Baca file PNG
    with open(png_file_path, 'rb') as file:
        image_data = file.read()

    # Buat objek MIMEImage
    image = MIMEImage(image_data, name='image.png')

    # Tambahkan objek MIMEImage ke objek MIMEMultipart

    # Menambahkan isi pesan
    body = """
Hai! Terima kasih telah menggunakan program UltraXplorer. Berikut merupakan hasil cetak ID Card!!

____
UltraXplorer"""
```

```
message.attach(MIMEText(body, 'plain'))
message.attach(image)

# Mengirim email menggunakan SMTP
with smtplib.SMTP(smtp_server, smtp_port) as server:
    server.starttls()
    server.login(sender_email, sender_password)
    server.sendmail(sender_email, receiver_email, message.as_string())
```

Penjelasan kode program modul untuk mengirimkan file melalui email:

1. Mengimpor pustaka yang diperlukan seperti pustaka `smtplib` untuk mengirim email melalui server SMTP, kelas-kelas `MIMEMultipart`, `MIMEImage`, dan `MIMEText` dari pustaka `email.mime` untuk membangun objek pesan email dengan lampiran gambar.
2. Modul `os` diimpor untuk mengambil nilai dari variabel yang berisi email dan kata sandi pengirim.
3. Fungsi `send_png_email(receiver_email, png_file_path)` didefinisikan untuk mengambil alamat email penerima (`receiver_email`) dan path file PNG (`png_file_path`) sebagai parameter.
4. Variabel `sender_email`, `sender_password`, `smtp_server`, dan `smtp_port` digunakan untuk mengatur pengirim email dan pengaturan server SMTP.
5. Objek `MIMEMultipart` (`message`) dibuat untuk menyimpan semua bagian email.
6. File PNG ID Card dibaca sebagai data biner menggunakan `open` dan disimpan dalam variabel `image_data`.
7. Objek `MIMEImage` (`image`) dibuat menggunakan `image_data` dan nama file `image.png`. Objek `MIMEText` digunakan untuk menambahkan isi pesan email, dan objek `MIMEMultipart` digunakan untuk menambahkan lampiran gambar.
8. Membuat koneksi SMTP dengan server menggunakan `smtplib.SMTP`, kemudian dilakukan login dengan akun pengirim menggunakan `server.login`.
9. Email akan dikirim menggunakan `server.sendmail` dengan pengirim, penerima, dan isi pesan yang dikonversi menjadi string.

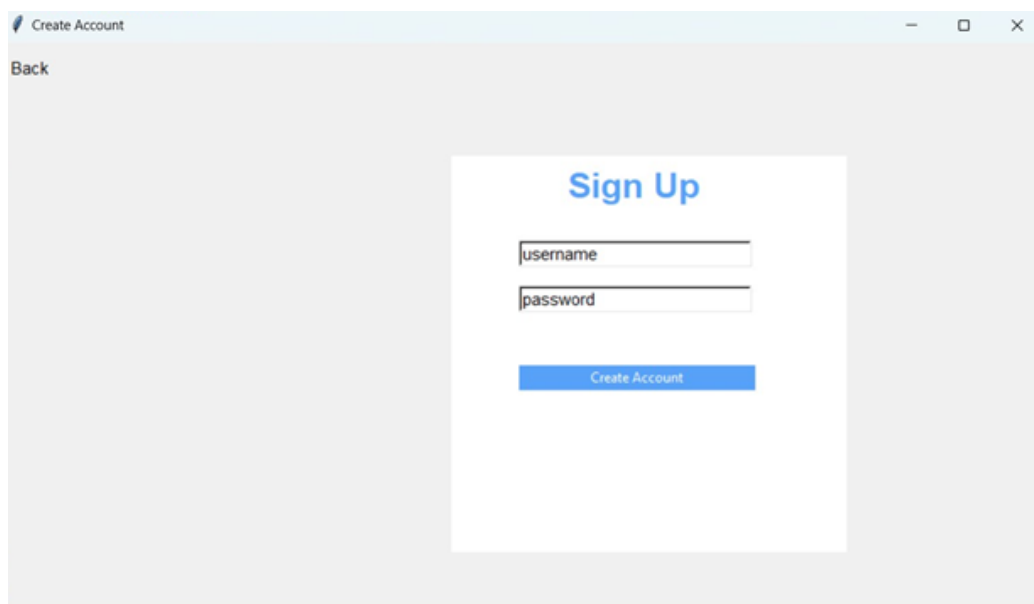
BAB IV

HASIL RUNNING PYTHON

Bab ini membahas mengenai hasil *running* python *Project Team* Praktikum Program Komputer.



Gambar 4.1 Tampilan Menu *Login/Sign In* UltraXplorer



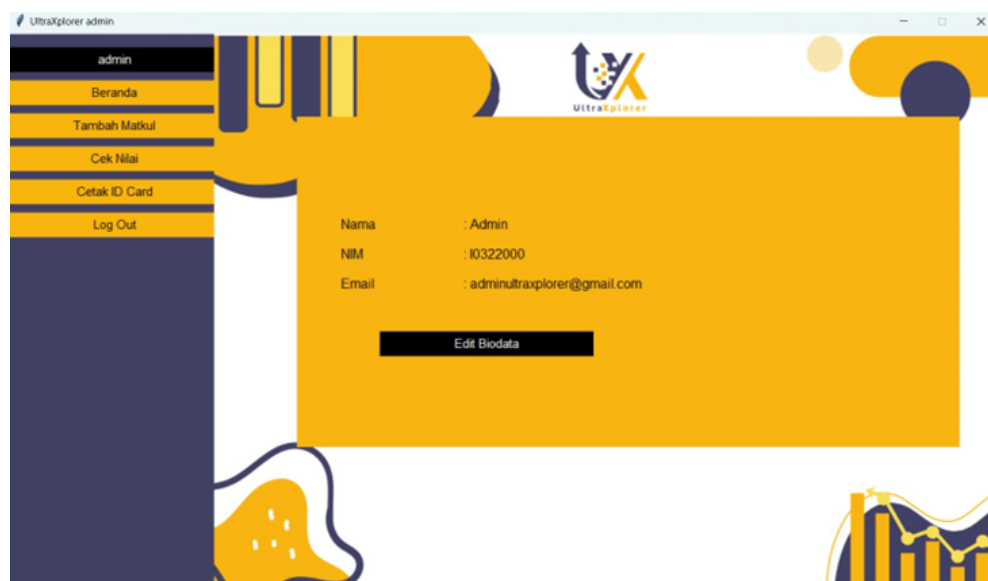
Gambar 4.2 Tampilan Menu *Create Account/Sign Up* UltraXplorer

Berikut merupakan hasil dari tampilan menu *login*. Pada menu tersebut *user* dapat meng-*input*-kan *username* dan *password* yang telah dibuat sebelumnya. Jika *user* belum memiliki akun, maka *user* dapat memilih tombol “*Create Account*”. Kemudian, *user* akan diarahkan menuju halaman buat akun dan *user* dapat membuat *username* serta *password* agar bisa *login* ke dalam program.

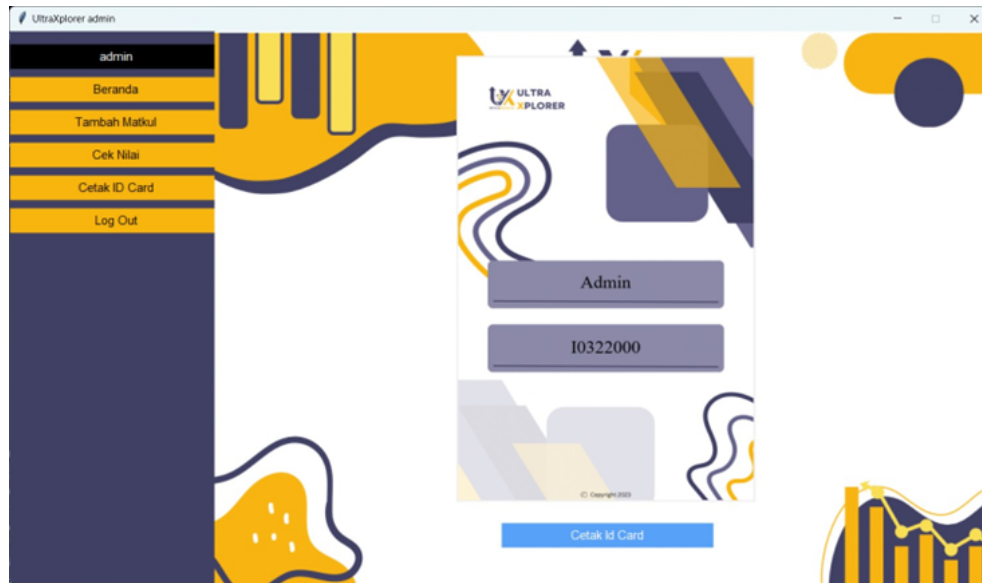


Gambar 4.3 Tampilan Menu Beranda UltraXplorer

Berikut merupakan hasil dari tampilan beranda. Pada beranda, terdapat beberapa kalimat yang menjelaskan identitas dari program yang dibuat. Kemudian *user* dapat memilih beberapa menu, yaitu *Manage Account*, Tambah Mata Kuliah, Cek Nilai dan Tambah Nilai, dan Cetak *ID Card*.

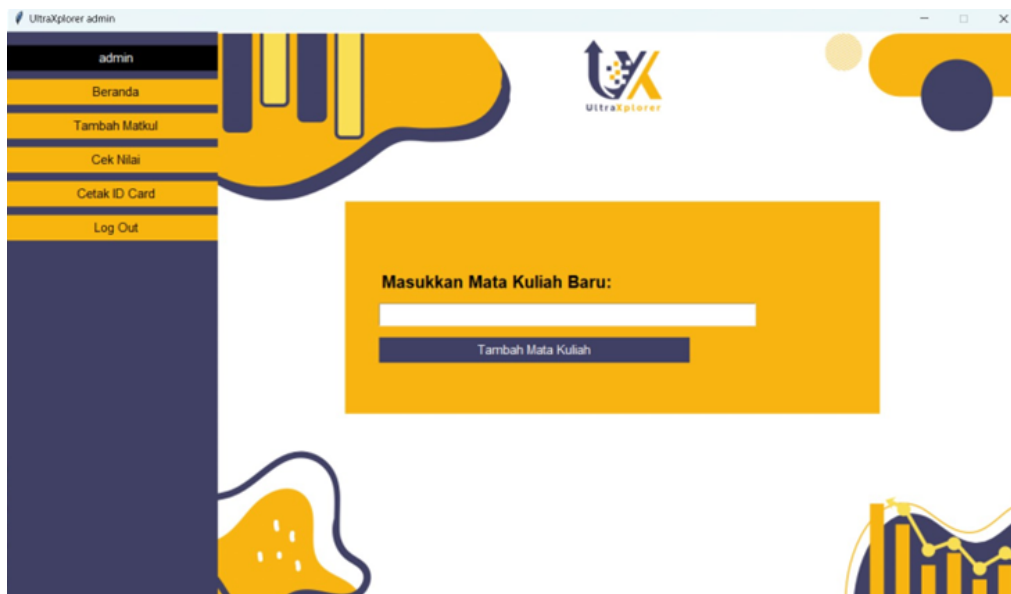


Gambar 4.4 Tampilan Menu *Manage Account* UltraXplorer

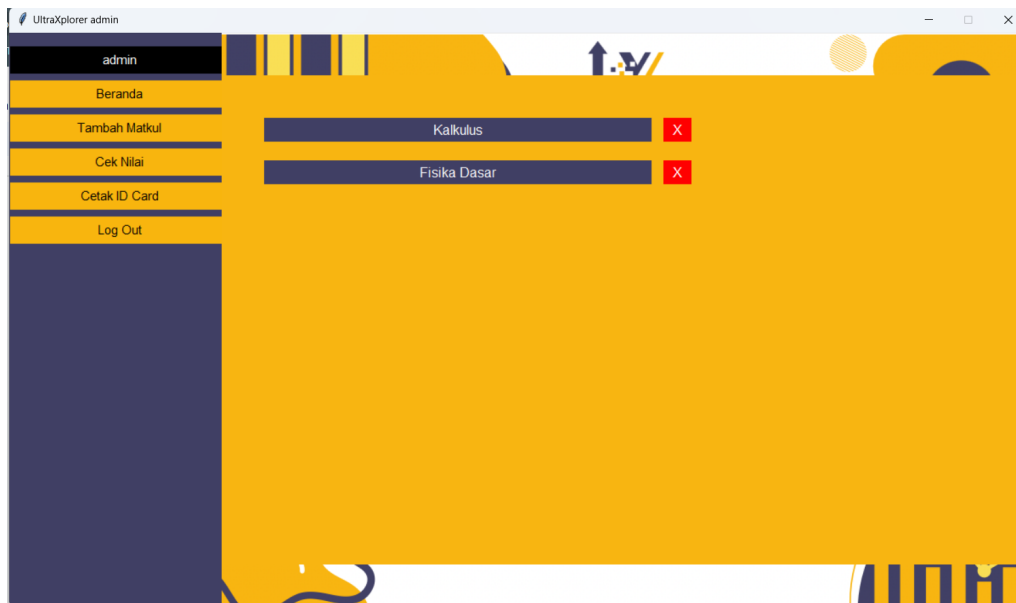


Gambar 4.5 Tampilan Menu Cetak *ID Card* UltraXplorer

Pada menu *Manage Account*, *user* dapat menambahkan data diri berupa nama, NIM, dan *email*. Nama dan NIM kemudian akan ditampilkan pada *ID Card* dan *email* disini berguna untuk mengirim hasil cetak *ID Card*. *User* juga dapat mengubah biodata tersebut sewaktu-waktu.



Gambar 4.6 Tampilan Menu Tambah Mata Kuliah UltraXplorer



Gambar 4.7 Tampilan Menu Pilihan Mata Kuliah UltraXplorer



Gambar 4.8 Tampilan Menu Cek Nilai dan Grafik UltraXplorer

Pada menu tambah mata kuliah *user* dapat menambahkan nama mata kuliah yang diinginkan, program akan mengirimkan pesan jika mata kuliah sudah ada sebelumnya. Kemudian nama mata kuliah akan muncul pada menu cek nilai. Pada menu cek nilai, *user* dapat memilih mata kuliah yang akan ditambahkan nilainya, kemudian *user* dapat menambahkan nilainya sehingga akan muncul grafik pada program.