

LAPORAN PROYEK AKHIR PRAKTIKUM
MATA KULIAH ALGORITMA DAN PEMROGRAMAN DASAR



Manajemen Pendaftaran Lomba MIT-WEEK 2024

Oleh:

Kelompok 4

Zulfikar Aryawinata	2409106020
Renaya Putri Alika	2409106002
Ken Bilqis Nuraini	2409106015

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MULAWARMAN
SAMARINDA 2024**

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan laporan ini dengan baik. Penyusunan laporan ini bertujuan untuk memberikan dokumentasi dan evaluasi terhadap program yang telah kami rancang dan implementasikan, dengan harapan dapat menjadi bahan pembelajaran dan pengembangan di masa mendatang.

Program yang kami kembangkan merupakan sistem pendaftaran lomba MIT-WEEK berbasis Python yang dirancang untuk mendukung kemudahan administrasi berbagai jenis kompetisi. Program ini melibatkan fitur pembuatan akun, pendaftaran peserta, pengelolaan data oleh admin, hingga pencatatan rinci tentang peserta yang terdaftar dalam setiap lomba.

Selama proses penyusunan dan pengembangan program ini, kami menghadapi beberapa kendala, seperti error handling yang kompleks dan pengelolaan data berbasis dictionary yang memerlukan perhatian ekstra. Namun, berkat bimbingan dari asisten praktikum dan kerja sama yang solid dari rekan-rekan, kami berhasil menyelesaikan laporan ini. Ucapan terima kasih yang mendalam kami sampaikan kepada semua pihak yang telah memberikan kontribusi, baik secara langsung maupun tidak langsung.

Akhir kata, kami menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kami sangat terbuka terhadap saran dan kritik yang membangun. Semoga laporan ini dapat memberikan manfaat bagi semua pembaca dan pihak yang berkepentingan.

Samarinda, 19 November 2024

TAKARIR

Daftar padanan kata bahasa asing dalam bahasa Indonesia yang digunakan adalah sebagai berikut :

<i>Admin</i>	Pengelola
<i>User</i>	Pengguna
<i>Register</i>	Pendaftaran
<i>Login</i>	Masuk
<i>Logout</i>	Keluar
<i>Account</i>	Akun
<i>Menu</i>	Pilihan
<i>Error Handling</i>	Penanganan Kesalahan
<i>Input</i>	Masukan
<i>Output</i>	Keluaran
<i>Data</i>	Informasi
<i>Username</i>	Nama Pengguna
<i>Password</i>	Kata Sandi
<i>Dictionary</i>	Kamus
<i>Try-Except</i>	Coba-Kecuali
<i>Syntax Error</i>	Kesalahan Sintaksis
<i>Key Error</i>	Kesalahan Kunci
<i>Value Error</i>	Kesalahan Nilai
<i>Attribute Error</i>	Kesalahan Atribut
<i>Memory Error</i>	Kesalahan Memori

<i>Type Error</i>	Kesalahan Jenis
<i>File Not Found Error</i>	File Tidak Ditemukan
<i>Indentation Error</i>	Kesalahan Indentasi
<i>Validation</i>	Validasi

DAFTAR ISI

KATA PENGANTAR	ii
TAKARIR.....	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	vii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Kebutuhan Fungsional	1
1.3 Rumusan Masalah	2
1.4 Batasan Masalah.....	2
1.5 Tujuan	3
BAB II PERANCANGAN.....	4
2.1 Analisis Program.....	4
2.2 Flowchart.....	7
2.3 Konsep/Materi Praktikum yang dipakai	16
BAB III HASIL DAN PEMBAHASAN	18
3.1 Tampilan Program.....	18
1. Tampilan Menu.....	18
2. Buat Akun User	19
3. Menu Masuk.....	20
4. Menampilkan Lomba.....	21
5. Menu Pendaftaran.	21

6. Menu Login Admin.....	22
7. Menampilkan Data Pendaftar	23
8. Ubah Data Pendaftar	24
9. Menampilkan Akun User.....	24
10. Menghapus Data Pendaftar.....	25
11. Menu Kembali & Keluar	26
3.2 Source Code.....	27
BAB IV PENUTUP.....	43
4.1 Kesimpulan.....	43
4.2 Saran.....	44
DAFTAR PUSTAKA	45
LAMPIRAN.....	46

DAFTAR GAMBAR

Gambar 2.1 Flowchart Bagian Menu Utama	7
Gambar 2.2 Flowchart Bagian Login Admin	9
Gambar 2.3 Flowchart Bagian Menu Admin	11
Gambar 2.4 Flowchart Bagian Login User	13
Gambar 2.5 Flowchart Bagian Menu User	15
Gambar 3.1 Tampilan Menu	18
Gambar 3.2 Buat Akun User	19
Gambar 3.3 Menu Masuk	20
Gambar 3.4 Menampilkan Lomba	21
Gambar 3.5 Menu Pendaftaran	21
Gambar 3.6 Menu Login Admin	22
Gambar 3.7 Menampilkan Data Pendaftar	23
Gambar 3.8 Mengubah Data Pendaftar	24
Gambar 3.9 Menampilkan Akun user	24
Gambar 3.10 Menghapus Data Pendaftar	25
Gambar 3.11 Menu Kembali & Keluar	26

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam berbagai acara perlombaan, pengelolaan data peserta sering kali menjadi tantangan besar, terutama ketika jumlah peserta dan jenis lomba yang diadakan cukup banyak. Proses manual cenderung memakan waktu dan berisiko terhadap kesalahan pencatatan data. Oleh karena itu, diperlukan sistem yang mampu mendigitalisasi proses pendaftaran, pengelolaan, dan penyimpanan data peserta secara efisien. Sistem berbasis Python ini hadir untuk menyediakan solusi sederhana namun fungsional untuk mempermudah penyelenggara acara dalam mengelola data peserta lomba.

1.2 Kebutuhan Fungsional

Program ini dirancang untuk memenuhi kebutuhan berikut:

- 1. Pendaftaran Pengguna:** Memungkinkan pengguna untuk membuat akun dengan validasi nama pengguna dan kata sandi.
- 2. Login:** Memberikan akses pengguna untuk melihat dan memilih lomba yang akan diikuti.
- 3. Pendaftaran Lomba:** Memungkinkan peserta mendaftar untuk lomba tertentu dengan pengisian data diri.
- 4. Manajemen Data oleh Admin:** Memfasilitasi admin untuk melihat, mengubah, atau menghapus data peserta sesuai kebutuhan.
- 5. Pengelolaan Data Berdasarkan Jenis Lomba:** Data peserta dikelompokkan sesuai jenis lomba, baik individu maupun tim.
- 6. Penanganan Kesalahan:** Program memiliki mekanisme untuk menangani berbagai jenis kesalahan, seperti kesalahan input dan validasi data.

1.3 Rumusan Masalah

1. Bagaimana menciptakan sistem pendaftaran lomba yang sederhana namun efektif untuk berbagai jenis lomba?
2. Bagaimana memastikan data peserta tersimpan dengan aman dan mudah diakses oleh admin?
3. Bagaimana meminimalkan kesalahan input dan memastikan integritas data selama proses pendaftaran?
4. Bagaimana memberikan pengalaman pengguna yang intuitif baik untuk peserta maupun admin?

1.4 Batasan Masalah

1. Program ini hanya berbasis teks (*text-based*) dan dioperasikan melalui terminal/command prompt.
2. Data peserta tidak disimpan secara permanen; data hanya tersedia selama program berjalan.
3. Sistem tidak mendukung pengiriman notifikasi atau integrasi dengan platform lain.
4. Jumlah peserta dan tim yang didaftarkan tidak dibatasi oleh program, tetapi tergantung pada kemampuan pengelolaan pengguna.
5. Tidak ada antarmuka grafis (GUI) dalam versi ini; pengembangan dilakukan menggunakan Python dengan fitur bawaan (*built-in*).

1.5 Tujuan

1. Membangun sistem digital yang mempermudah pendaftaran lomba untuk peserta dan penyelenggara.
2. Menyediakan fitur pengelolaan data yang terstruktur untuk memastikan akurasi dan efisiensi administrasi.
3. Meningkatkan efisiensi pengelolaan lomba dengan mengelompokkan data berdasarkan kategori lomba.
4. Memberikan pengalaman pengguna yang sederhana namun fungsional.
5. Menjadi dasar untuk pengembangan sistem pendaftaran lomba yang lebih canggih di masa depan.

BAB II

PERANCANGAN

2.1 Analisis Program

1. Struktur Program

- **Struktur Modular:** Program diorganisasikan menjadi beberapa fungsi utama seperti `registrasi()`, `login_user()`, dan `login_admin()`, yang memungkinkan pengelolaan alur kerja dengan baik.
- **Pengelompokan Data:** Data peserta disimpan dalam *dictionary* yang terpisah berdasarkan jenis lomba, memungkinkan akses data yang terorganisasi dan efisien.
- **Penggunaan Menu Utama:** Program memulai alur interaksi dengan pengguna melalui menu utama yang memberikan beberapa pilihan, termasuk pembuatan akun, login pengguna, dan akses admin.

2. Fitur Utama

- **Registrasi Akun:**
 - Program memungkinkan pengguna untuk membuat akun dengan validasi nama pengguna dan kata sandi.
 - Validasi sederhana memastikan nama pengguna hanya berisi huruf dan memiliki panjang minimum 4 karakter.
- **Login Pengguna dan Admin:**
 - Login untuk pengguna memungkinkan akses ke daftar lomba dan pendaftaran.
 - Login admin memberikan kontrol penuh atas data peserta, termasuk melihat, mengubah, dan menghapus data.

- **Pendaftaran Lomba:**
 - Peserta dapat mendaftar pada lomba perorangan atau tim.
 - Untuk lomba tim, pengguna harus memasukkan nama tim dan data anggota tim.
- **Manajemen Data oleh Admin:**
 - Admin dapat melihat data peserta yang dikelompokkan berdasarkan jenis lomba.
 - Admin juga dapat mengubah atau menghapus data peserta jika diperlukan.
- **Error Handling:**
 - Program menggunakan *try-except* untuk menangani berbagai kesalahan input atau teknis yang mungkin terjadi selama program dijalankan.

3. Keunggulan Program

1. **Pengelolaan Data Terstruktur:** Data peserta disimpan secara terorganisasi dalam *dictionary*, mempermudah akses dan pengelolaan.
2. **Fleksibilitas Pendaftaran:** Program mendukung pendaftaran individu maupun tim, dengan validasi yang membantu menjaga integritas data.
3. **Error Handling yang Komprehensif:** Program mengantisipasi berbagai kesalahan, termasuk kesalahan input, tipe data, atau pengelolaan memori.
4. **Sederhana namun Efektif:** Antarmuka berbasis teks mudah dipahami dan digunakan, bahkan oleh pengguna dengan pengetahuan teknis yang minim.
5. **Fitur Admin yang Kuat:** Admin memiliki kontrol penuh atas data peserta, memberikan fleksibilitas dalam pengelolaan lomba.

4. Kelemahan Program

1. Tidak Ada Penyimpanan Permanen:

- Data peserta hanya tersedia selama program berjalan, karena tidak ada integrasi dengan database atau file eksternal.

2. Tidak Ada Antarmuka Grafis (GUI):

- Program berbasis teks mungkin kurang menarik bagi pengguna non-teknis.

3. Validasi Data Terbatas:

- Validasi input masih dapat diperbaiki, seperti memastikan format tertentu untuk data NIM atau institusi.

4. Tidak Mendukung Multi-User:

- Program hanya dapat digunakan oleh satu pengguna (atau admin) dalam satu waktu.

5. Kurangnya Notifikasi atau Feedback Lanjutan:

- Tidak ada fitur untuk memberikan konfirmasi lanjutan (misalnya, pengiriman email atau laporan terperinci).

5. Peluang Pengembangan

1. Penyimpanan Data Permanen:

- Integrasikan program dengan database atau file eksternal untuk menyimpan data secara permanen.

2. Validasi Data yang Lebih Ketat:

- Perbaiki validasi input, seperti format email, nomor telepon, atau NIM.

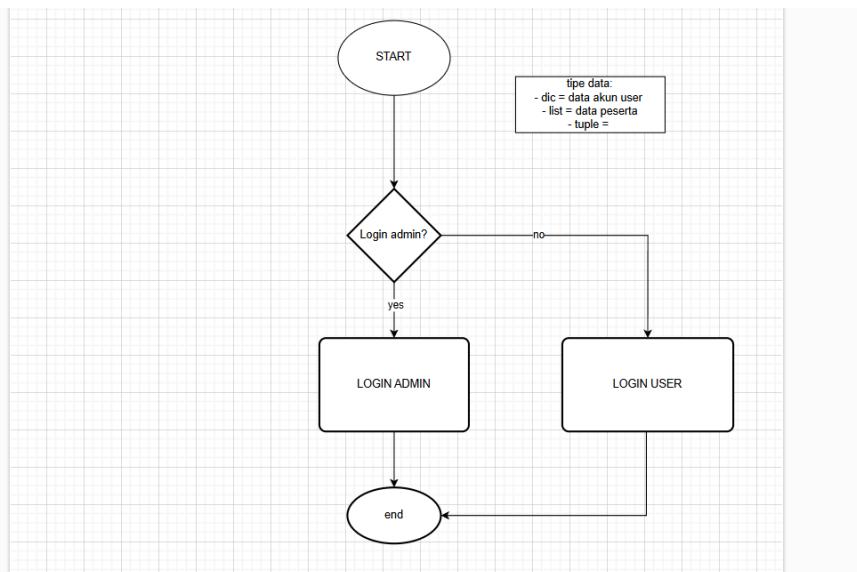
3. Dukungan Multi-User:

- Tambahkan fitur untuk memungkinkan beberapa pengguna atau admin mengakses program secara bersamaan.

4. Pelaporan Otomatis:

- Tambahkan fitur untuk menghasilkan laporan peserta yang dapat diunduh dalam format CSV atau PDF.

2.2 Flowchart



Gambar 2.1 Flowchart Bagian Menu Utama

Berdasarkan diagram yang diberikan, berikut adalah penjelasan alur program pada gambar 2.1 diatas:

1. Start

Program dimulai dengan inisialisasi dan menampilkan menu awal untuk pengguna.

2. Login Admin atau User

Pengguna diminta untuk menentukan apakah ingin login sebagai Admin atau User.

- Jika memilih Admin, alur akan masuk ke fungsi Login Admin.
- Jika memilih User, alur akan masuk ke fungsi Login User.

3. Login Admin

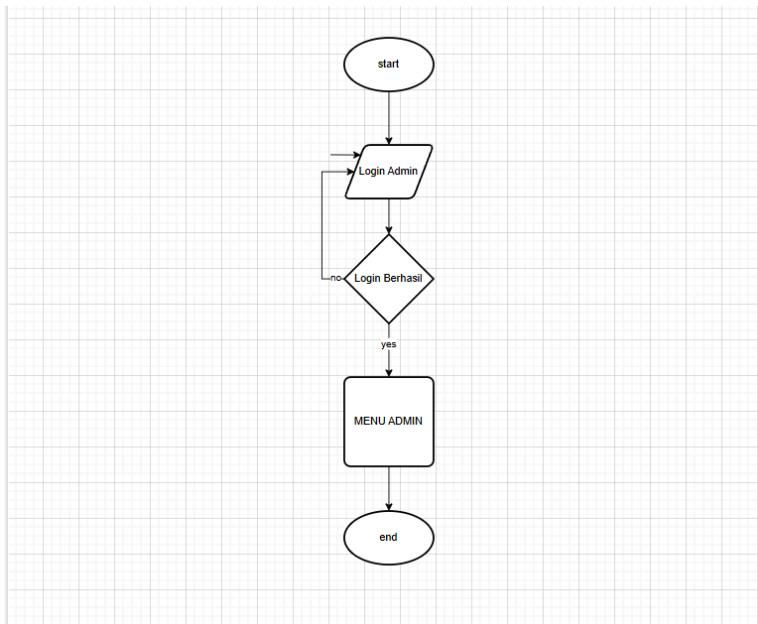
- Admin akan diminta memasukkan kredensial (*username* dan *password*).
- Setelah berhasil login, Admin diberikan akses ke berbagai fungsi manajemen, seperti:
 - Melihat data peserta.
 - Mengubah atau menghapus data peserta.
 - Mengelola data lomba.
- Setelah selesai, alur program kembali ke end.

4. Login User

- User akan diminta memasukkan kredensial untuk login.
- Setelah berhasil login, User dapat:
 - Melihat daftar lomba yang tersedia.
 - Mendaftar ke lomba tertentu (individu atau tim).
 - Memperbarui informasi akun.
 - Setelah selesai, alur program kembali ke end.

5. End

Program selesai atau kembali ke menu awal jika pengguna ingin melanjutkan proses lain.



Gambar 2.2 Flowchart Bagian Login Admin

Berikut adalah penjelasan alur program pada gambar 2.2 diatas:

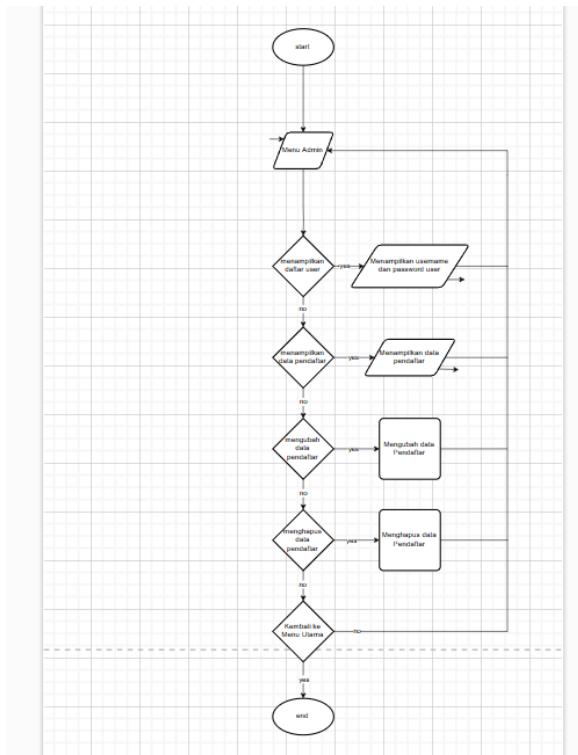
1. **Start**
 - Program dimulai dan mengarahkan alur ke proses login untuk **Admin**.
2. **Proses Login Admin**
 - Admin diminta untuk memasukkan kredensial login berupa username dan password.
 - Program memverifikasi data yang dimasukkan oleh Admin.
3. **Login Berhasil**
 - **Jika Login Berhasil (Yes):** Admin diarahkan ke **Menu Admin**, di mana mereka dapat mengakses fitur-fitur manajemen data seperti melihat, mengedit, atau menghapus data peserta lomba.
 - **Jika Login Gagal (No):** Program akan mengarahkan kembali ke proses **Login Admin**, dan Admin diminta mengulangi input hingga login berhasil.

4. Menu Admin

- Setelah berhasil login, Admin dapat mengakses fitur manajemen seperti:
 - Mengelola data peserta.
 - Melihat informasi pendaftaran lomba.
 - Melakukan penghapusan atau pembaruan data peserta.

5. End

- Setelah selesai, program mengakhiri proses, kembali ke titik awal, atau keluar dari sistem.
6. **Pengulangan (Looping):** Jika login gagal, program akan terus meminta Admin untuk memasukkan kredensial yang benar. Hal ini menghindari akses yang tidak sah ke sistem.
 7. **Pengelompokan Fitur:** Setelah login berhasil, semua fitur untuk admin dikelompokkan dalam Menu Admin, sehingga alur program menjadi rapi dan terorganisasi.
 8. **Fokus Peran:** Diagram ini secara spesifik mengatur proses untuk admin saja, sehingga memisahkan peran antara admin dan user dengan jelas.



Gambar 2.3 Flowchart Bagian Menu Admin

Berikut adalah penjelasan alur program pada gambar 2.3 diatas:

1. Start

- Program dimulai dan diarahkan ke **Menu Admin** setelah Admin berhasil login.

2. Menu Admin

- Admin diberikan beberapa opsi fitur yang dapat dipilih:
 - Menampilkan daftar user.
 - Menampilkan data pendaftar.
 - Mengubah data pendaftar.
 - Menghapus data pendaftar.
 - Kembali ke menu utama.

3. Menampilkan Daftar User

- Jika Admin memilih opsi ini:
 - Sistem menampilkan seluruh username dan password user yang terdaftar.
- Setelah selesai, Admin diarahkan kembali ke menu utama.

4. Menampilkan Data Pendaftar

- Jika Admin memilih opsi ini:
 - Sistem menampilkan data seluruh pendaftar berdasarkan informasi yang tersimpan (misalnya nama, lomba yang diikuti, dll.).
- Setelah selesai, Admin diarahkan kembali ke menu utama.

5. Mengubah Data Pendaftar

- Jika Admin memilih opsi ini:
 - Sistem meminta Admin untuk memasukkan informasi yang akan diubah.
 - Data pendaftar diperbarui sesuai input Admin.
- Setelah selesai, Admin diarahkan kembali ke menu utama.

6. Menghapus Data Pendaftar

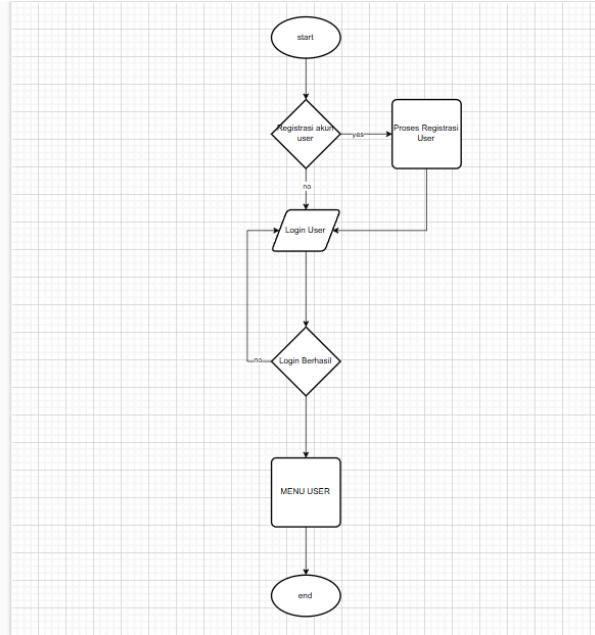
- Jika Admin memilih opsi ini:
 - Sistem meminta Admin untuk memilih data pendaftar yang akan dihapus.
 - Data yang dipilih akan dihapus dari sistem.
- Setelah selesai, Admin diarahkan kembali ke menu utama.

7. Kembali ke Menu Utama

- Jika Admin memilih opsi ini:
 - Sistem akan mengembalikan Admin ke menu awal atau proses lain di luar **Menu Admin**.

8. End

- Program mengakhiri proses setelah Admin keluar dari menu atau menyelesaikan semua tugas yang dipilih.



Gambar 2.4 Flowchart Bagian Login User

Berikut adalah penjelasan alur program pada gambar 2.4 diatas:

1. Start

- Program dimulai dan memberikan opsi kepada pengguna untuk melakukan registrasi akun baru atau langsung login ke sistem.

2. Registrasi Akun User

- **Jika pengguna memilih untuk melakukan registrasi (Yes):**

- Sistem akan menjalankan **Proses Registrasi User**, di mana pengguna diminta memasukkan data seperti username dan password untuk membuat akun baru.
- Setelah selesai, program akan kembali ke proses awal, memberikan opsi login atau registrasi ulang.

- **Jika pengguna memilih tidak (No):**

- Pengguna diarahkan langsung ke proses login.

3. Login User

- Pengguna diminta memasukkan kredensial login berupa username dan password.
- Program akan memverifikasi kredensial tersebut.

4. Login Berhasil

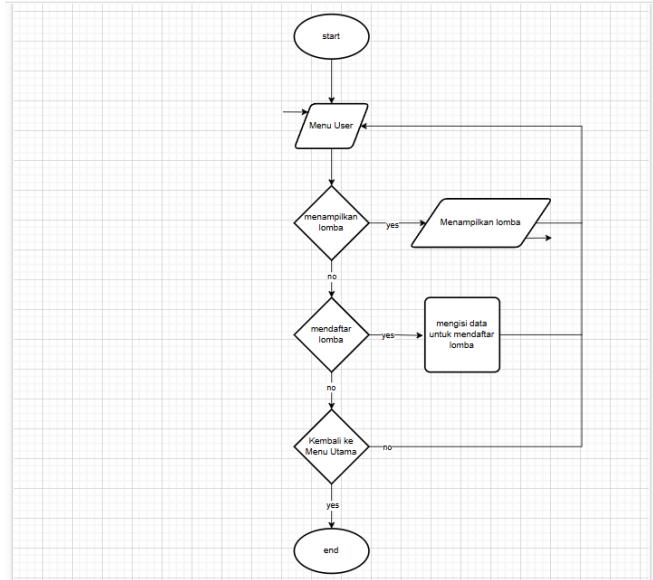
- **Jika Login Berhasil (Yes):**
 - Pengguna diarahkan ke **Menu User**, di mana mereka dapat mengakses fitur-fitur khusus untuk user.
- **Jika Login Gagal (No):**
 - Sistem akan mengulangi proses login, meminta pengguna memasukkan ulang kredensial yang benar.

5. Menu User

- Setelah berhasil login, pengguna akan diarahkan ke menu utama untuk user. Di sini, mereka dapat mengakses fitur yang sesuai dengan kebutuhan (misalnya melihat atau mendaftar lomba).

6. End

- Program berakhir setelah pengguna menyelesaikan aktivitas di sistem atau keluar dari aplikasi.



Gambar 2.5 Flowchart Bagian Menu User

Berikut adalah penjelasan alur program pada gambar 2.5 diatas:

1. **Start:** Proses dimulai.
2. **Menu User:** Pengguna masuk ke *Menu User* yang menjadi pusat pilihan tindakan.
3. **Menampilkan Lomba:**
 - Pengguna diberikan opsi untuk menampilkan daftar lomba.
 - Jika memilih "Yes", sistem akan menampilkan daftar lomba yang tersedia.
 - Jika memilih "No", alur dilanjutkan ke langkah berikutnya.
4. **Mendaftar Lomba:**
 - Pengguna diberi opsi untuk mendaftar lomba.
 - Jika memilih "Yes", sistem akan mengarahkan pengguna untuk mengisi data pendaftaran lomba.
 - Jika memilih "No", alur dilanjutkan ke langkah berikutnya.
5. **Kembali ke Menu Utama:**
 - Pengguna diberi pilihan untuk kembali ke menu utama.
 - Jika memilih "Yes", pengguna akan diarahkan kembali ke *Menu User*.
 - Jika memilih "No", sistem akan mengakhiri proses.
6. **End:** Proses selesai.

2.3 Konsep/Materi Praktikum yang dipakai

Konsep yang kami gunakan dalam pembuatan laporan ini sesuai dengan fungsi dan logika pemrograman Python yang tercantum pada program, di antaranya:

1. Fungsi Dasar:

- **Fungsi print:** Program menggunakan fungsi print untuk menampilkan keluaran di layar. Fungsi ini sangat penting dalam memberikan umpan balik langsung kepada pengguna selama eksekusi program.

2. Input dan Output:

- **Fungsi input:** Program melibatkan masukan data dari pengguna melalui fungsi input. Fungsi ini digunakan untuk membaca informasi yang diberikan oleh pengguna dalam bentuk teks.

3. Struktur Data Dasar:

- **Variabel:** Variabel digunakan untuk menyimpan data sementara yang diperlukan dalam proses perhitungan.
- **String:** Terdapat manipulasi string sederhana untuk pengolahan data yang dimasukkan oleh pengguna.

4. Operasi Logika dan Kontrol Alur Program:

- **Percabangan if-elif-else:** Program memanfaatkan percabangan untuk memproses data berdasarkan kondisi tertentu. Hal ini memungkinkan pengambilan keputusan yang dinamis berdasarkan masukan pengguna.

5. Perulangan (Looping):

- **for atau while:** Jika terdapat loop dalam program (silakan konfirmasi jika Anda ingin penjelasan lebih spesifik), loop tersebut digunakan untuk menjalankan blok kode berulang kali berdasarkan kondisi tertentu.

6. Pemrograman Modular:

- **Fungsi:** Program dapat memanfaatkan fungsi untuk membagi kode menjadi bagian-bagian yang lebih kecil dan modular. Fungsi-fungsi ini membantu dalam mengorganisasi logika program serta mengurangi pengulangan kode.

7. Konsep Error Handling:

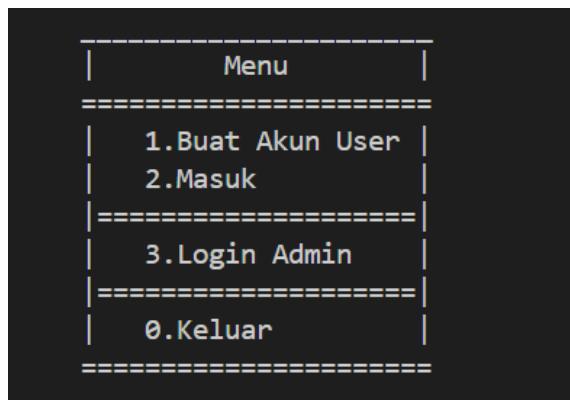
- Jika terdapat penanganan error dalam program, konsep ini digunakan untuk memastikan program tetap berjalan meskipun terjadi kesalahan pada saat runtime.

BAB III

HASIL DAN PEMBAHASAN

3.1 Tampilan Program

1. Tampilan Menu



Gambar 3.1 Tampilan Menu

Program pada gambar 3.1 ini adalah sistem tampilan menu yang memiliki fungsi sebagai berikut:

1. Buat Akun User

Pengguna dapat membuat akun dengan memasukkan username dan password. Data ini kemudian disimpan oleh program.

2. Masuk

Pengguna bisa login menggunakan username dan password yang telah dibuat sebelumnya. Sistem akan mencocokkan input dengan data yang tersimpan.

3. Login Admin

Digunakan untuk login sebagai admin, biasanya menggunakan username dan password default (contoh: admin/admin).

4. Keluar

Mengakhiri program

2. Buat Akun User

```
|-----|  
|      Menu      |  
|-----|  
| 1.Buat Akun User |  
| 2.Masuk          |  
|-----|  
| 3.Login Admin   |  
|-----|  
| 0.Keluar        |  
|-----|  
  
Masukkan Pilihan Anda (1/2/3/0): 1  
  
Masukkan username yang ingin dipakai : nayul  
Password : nayul  
username dan password telah disimpan
```

Gambar 3.2 Buat Akun User

Berikut adalah penjelasan dari program 3.2:

1. Menu Utama

Program menampilkan menu utama dengan beberapa opsi:

- 1. Buat Akun User: Pilihan untuk membuat akun pengguna baru.
- 2. Masuk: Pilihan untuk login sebagai pengguna biasa (user).
- 3. Login Admin: Pilihan untuk login sebagai admin.
- 0. Keluar: Pilihan untuk keluar dari program.

2. Input Pilihan

Pengguna diminta untuk memasukkan pilihan (dalam contoh ini, pengguna memasukkan angka 1, yaitu "Buat Akun User"). Setelah itu, program meminta

username dan password. Pada gambar, pengguna memasukkan:

- Username : nayul
- Password : nayul

Program menerima input ini dan menampilkan pesan “username dan password telah disimpan.

3. Menu Masuk

```
-----  
|       Menu      |  
=====| 1.Buat Akun User |  
| 2.Masuk          |  
=====| 3.Login Admin   |  
=====| 0.Keluar        |  
=====  
  
Masukkan Pilihan Anda (1/2/3/0): 2  
  
username : nayul  
Password : nayul  
  
SELAMAT DATANG :^)
```

Gambar 3.3 Menu Masuk

Pengguna diminta untuk memasukkan pilihan (dalam contoh ini, pengguna memasukkan angka **2**, yaitu "Masuk"). Setelah itu, program meminta **username** dan **password**. Program menerima input ini dan menampilkan pesan: "**SELAMAT DATANG :^)**", menandakan login user berhasil.

4. Menampilkan Lomba

```
SELAMAT DATANG :^)

1> Tampilkan Lomba
2> Daftar
3> Kembali

Masukkan Pilihan (1/2/3): 1

=====
|                               LOMBA
=====
| Perorangan      | Tim
|-----+-----|
| 1> Network      | 3> Robotik
| 2> Ranking 1    | 4> UI/UX
|                 | 5> Mobile Legends
|                 | 6> Valorant
=====
```

Gambar 3.4 Menampilkan Lomba

Pada gambar 3.4, Pengguna memilih opsi **1. Tampilkan Lomba** dari menu utama untuk melihat daftar kompetisi yang tersedia. Fitur ini hanya menampilkan daftar kompetisi kepada pengguna tanpa memerlukan input lebih lanjut. Pengguna bisa mempelajari kategori lomba sebelum memilih menu lain, seperti **Daftar** atau **Kembali**.

5. Menu Pendaftaran

```
1> Tampilkan Lomba
2> Daftar
3> Kembali

Masukkan Pilihan (1/2/3): 2

=====
|                               LOMBA
=====
| Perorangan      | Tim
|-----+-----|
| 1> Network      | 3> Robotik
| 2> Ranking 1    | 4> UI/UX
|                 | 5> Mobile Legends
|                 | 6> Valorant
=====

0> Keluar

Lomba (1/2/3/4/5/6/0) : 2
Nama : nayul
Masukkan NIM (isi dengan 0 jika tidak memiliki NIM) : 1234567890
Asal Prodi (isi dengan (-) jika tidak di prodi manapun): infor
Asal Instansi: ummul

Data Telah Disimpan
```

Gambar 3.5 Menu Pendaftaran

Gambar ini menunjukkan proses pendaftaran lomba dengan pilihan Perorangan atau Tim.

1. Pilih Lomba: Peserta memilih jenis lomba (perorangan atau tim) dari daftar yang tersedia.
2. Isi Data: Peserta diminta mengisi nama, NIM (atau 0 jika tidak ada), asal prodi (atau "-" jika tidak ada), dan asal instansi.
3. Konfirmasi: Setelah semua data diisi dengan benar, program menyimpan informasi dan menampilkan pesan "Data Telah Disimpan."

Seperti itu juga jika memilih Tim. Program ini juga memvalidasi input Seperti panjang NIM agar sesuai aturan.

6. Menu Login Admin

```
-----  
|       Menu      |  
=====| 1.Buat Akun User |  
| 2.Masuk         |  
=====| 3.Login Admin   |  
| 0.Keluar        |  
=====  
  
Masukkan Pilihan Anda (1/2/3/0): 3  
Masukkan username : admin  
Password : admin  
  
SELAMAT DATANG DI MENU ADMIN  
  
=====| 1> Tampilkan Data Pendaftar |  
| 2> Ubah Data Pendaftar  |  
| 3> Tampilkan Akun Users |  
| 4> Hapus Data Pendaftar |  
| 5> Kembali             |  
=====
```

Gambar 3.6 Menu Login Admin

Jika ingin login sebagai admin, maka pengguna memilih opsi **3. Login Admin** Setelah itu pengguna akan diminta memasukkan Username dan

Password yaitu ‘admin’ dan jika berhasil login, maka program akan menampilkan pesan “SELAMAT DATANG DI MENU ADMIN” dan program juga akan menampilkan isi menu admin.

7. Menampilkan Data Pendaftar

```
=====
|1> Tampilkan Data Pendaftar |
|2> Ubah Data Pendaftar |
|3> Tampilkan Akun Users |
|4> Hapus Data Pendaftar |
|5> Kembali |
=====

Masukkan Pilihan Anda (1/2/3/4/5) : 1

=====
|Pilih Lomba yang ingin dilihat :
| 1> Network      4> UI/UX
| 2> Ranking 1     5> Mobile Legends
| 3> Robotik       6> Valorant
=====

Lomba (1/2/3/4/5/6) : 2

=====
PENDAFTAR LOMBA RANKING #1
=====
Akun : nayul
      Nama   : nayul
      NIM    : 1234567890
      Prodi  : infor
      Instansi : unmul
      Mendaftar pada Hari Jumat, tanggal 22-11-2024.
```

Gambar 3.7 Menampilkan Data Pendaftar

Pada tampilan menu admin, Pengguna memilih opsi **1: Tampilkan Data Pendaftar** dari menu admin. Setelah memilih opsi **1**, pengguna diarahkan ke menu untuk memilih jenis lomba. Pada contoh gambar diatas, Pengguna memilih opsi **2: Ranking 1**. Program menampilkan data pendaftar lomba *Ranking 1*, yang berisi informasi seperti pada gambar. Program ini memungkinkan admin untuk melihat data pendaftar berdasarkan kategori lomba tertentu. Dalam contoh ini, admin melihat detail peserta lomba *Ranking 1* yang terdaftar.

8. Ubah Data Pendaftar

```
Masukkan NamaPendaftar : nayul  
PROSES UBAH DATA PENDAFTAR LOMBA RANKING 1  
Nama : nayaw  
Masukkan NIM (isi dengan 0 jika tidak memiliki NIM) : 0  
Asal Prodi (isi dengan (-) jika tidak di prodi manapun): -  
Masukkan Asal Instansi : unmul
```

Gambar 3.8 Mengubah Data Pendaftar

Pengguna diminta memasukkan **NamaPendaftar** untuk memilih peserta yang datanya akan diubah (contoh: "nayul"). Setelah nama ditemukan, sistem menampilkan data yang ada dan meminta pengguna untuk mengubah atau mengisi ulang informasi. Menu ini digunakan untuk memperbarui informasi peserta yang mungkin salah atau belum lengkap. Menu ini dirancang agar admin atau pengguna dapat memodifikasi data pendaftaran.

9. Menampilkan Akun User

```
|-----|  
|1> Tampilkan Data Pendaftar|  
|2> Ubah Data Pendaftar|  
|3> Tampilkan Akun Users|  
|4> Hapus Data Pendaftar|  
|5> Kembali|  
|-----|  
  
Masukkan Pilihan Anda (1/2/3/4/5) : 3  
  
Daftar Akun User :  
nayul
```

Gambar 3.9 Menampilkan Akun User

Menu "Tampilkan Akun Users" pada program tersebut bertujuan untuk menampilkan daftar akun pengguna yang terdaftar dalam sistem. Menu ini digunakan untuk melihat semua akun yang telah dibuat atau terdaftar di

sistem. Prosesnya : Pengguna memilih opsi nomor **3. Tampilkan Akun Users** dari menu utama. Sistem akan membaca data dari penyimpanan dan menampilkan semua akun pengguna. Pada contoh di gambar, sistem menampilkan satu akun pengguna dengan username "nayul". Menu ini biasanya berguna untuk administrator atau pengguna yang ingin mengetahui daftar akun aktif dalam sistem.

10. Menghapus Data Pendaftar

```
Masukkan Pilihan Anda (1/2/3/4/5) : 4

-----
|Pilih Lomba:
 | 1> Network      4> UI/UX
 | 2> Ranking 1     5> Mobile Legends
 | 3> Robotik       6> Valorant
|-----|


Pilih Lomba : 2

=====
PENDAFTAR LOMBA RANKING #1
=====
Akun : nayul
      Nama    : nayaw
      NIM     : 0
      Prodi   : -
      Instansi: unmul
      Mendaftar pada Hari Jumat, tanggal 22-11-2024.

Masukkan username peserta yang ingin dihapus: nayul
Data pendaftar 'nayul' di lomba 'Ranking 1' telah dihapus.
```

Gambar 3.10 Menghapus Data Pendaftar

Pengguna diminta untuk memilih Menu Admin (4. Hapus Data Pendaftar) kemudian memilih salah satu dari beberapa kategori lomba (contoh: Network, Ranking 1, Robotik, UI/UX, Mobile Legends, Valorant). Pengguna memilih opsi "2" (lomba Ranking 1). Informasi peserta yang telah mendaftar ditampilkan, seperti username (nayul), nama lengkap (nayaw), NIM, prodi, dan instansi. Pengguna diminta memasukkan username untuk menghapus data. Program menampilkan konfirmasi bahwa data peserta dengan username "nayul" telah dihapus dari lomba "Ranking 1".

11. Menu Kembali & Keluar

```
|=====|
|1> Tampilkan Data Pendaftar |
|2> Ubah Data Pendaftar |
|3> Tampilkan Akun Users |
|4> Hapus Data Pendaftar |
|5> Kembali |
|=====|
Masukkan Pilihan Anda (1/2/3/4/5) : 5

-----|      Menu      |-----
| 1.Buat Akun User |
| 2.Masuk |
|-----|
| 3.Login Admin |
|-----|
| 0.Keluar |
|-----|

Masukkan Pilihan Anda (1/2/3/0): 0
Made by :
    > Ken Bilqis Nuraini
        NIM: 2409106015
    > Renaya Putri Alika
        NIM: 2409106002
    > Zulfikar Aryawinata
        NIM: 2409106020
```

Gambar 3.11 Menu Kembali & Keluar

Pada gambar 3.10, Setelah memilih opsi **3. Kembali** dari menu sebelumnya, pengguna diarahkan kembali ke Menu Utama dengan pilihan Buat Akun User, Masuk, Login Admin, dan Keluar. Pengguna memilih opsi **0. Keluar** dari menu utama. Program kemudian menampilkan "Made by", yang menampilkan Nama dan NIM pembuat program.

3.2 Source Code

```
1 import re
2 import os
3 import locale
4 from datetime import datetime
5 locale.setlocale(locale.LC_TIME, 'id_ID.UTF-8')
6 waktu = datetime.now()
7 hari = waktu.strftime("%A")
8 tanggal = waktu.strftime("%d-%m-%Y")
9
10 admin = {'admin': 'admin'}
11 lomba_solo = ['Network', 'Ranking 1']
12 lomba_tim = ['Robotik', 'UI/UX', 'Mobile Legends', 'Valorant']
13 users = {}
14 pendaftar_network = {}
15 pendaftar_ranking1 = {}
16 pendaftar_robotik = {}
17 pendaftar_uiux = {}
18 pendaftar_mobilelegends = {}
19 pendaftar_valorant = {}
20 perlombaan = {
21     "Network": pendaftar_network,
22     "Ranking 1": pendaftar_ranking1,
23     "Robotik": pendaftar_robotik,
24     "UI/UX": pendaftar_uiux,
25     "Mobile Legends": pendaftar_mobilelegends,
26     "Valorant": pendaftar_valorant,
27 }
28 credit = ("Ken Bilqis Nuraini", 2409106015, "Renaya Putri Arika", 2409106002, "Zulfikar Aryawinata", 2409106020)

30 # tampilan data tersimpan
31 def data_tersimpan():
32     print('''
33 =====
34 Data Telah Disimpan
35 =====''')
36
37 # tampilan mengisi data
38 def silahkan_isi_data():
39     print('''
40 =====
41 Silahkan Isi Data yang diminta
42 =====''')
43
44 # Fungsi untuk menampilkan lomba
45 def menampilkan_lomba_admin():
46     print(""""
47         |-----
48         |Lomba:
49         |     1> Network      4> UI/UX
50         |     2> Ranking 1    5> Mobile Legends
51         |     3> Robotik       6> Valorant
52         |_
53     """)
54
55
56 # tampilan menu admin
57 def menu_admin():
58     print(""""

59 |=====
60 |1> Tampilkan Data Pendaftar |
61 |2> Ubah Data Pendaftar |
62 |3> Tampilkan Akun Users |
63 |4> Hapus Data Pendaftar |
64 |5> Kembali |
65 |_
66 """)

67
68 # tampilan peserta lomba
69 def menampilkan_lomba_peserta():
70     print("""
71 =====
72 |          LOMBA           |
73 =====
74 | Perorangan | Tim |
75 |-----|-----|
76 | 1> {lomba_solo[0]} | 3> {lomba_tim[0]} |
77 | 2> {lomba_solo[1]} | 4> {lomba_tim[1]} |
78 |-----|-----|
79 | 5> {lomba_tim[2]} | 6> {lomba_tim[3]} |
80 =====""")

81
82 # fungsi untuk menampilkan data lomba individu
83 def menampilkan_data_pendaftar(nama_lomba):
84     if not nama_lomba:
85         print(''
```

```

86 -----
87 | Belum ada peserta yang mendaftar
88 -----
89 |     else:
90 |         for akun, data in nama_lomba.items():
91 |             print(f"\n{data['Nama']}\n{data['NIM']}\n{data['Prodi']}\n{data['Instansi']}\nMendaftar pada Hari {data['Hari']}, tanggal {data['Tanggal']}.\n")
92 |
93 |
94 |
95 |
96 |
97 |
98 |
99 |
100 # menampilkan data lomba bertim
101 def menampilkan_data_pendaftar_tim(nama_lomba):
102     if not nama_lomba:
103         print('')
104 -----
105 | Belum ada peserta yang mendaftar
106 -----
107 |     else:
108 |         for akun, data in nama_lomba.items():
109 |             print(f"\n{data['Nama Tim']}\n{data['Anggota']}\n{data['Prodi']}\n{data['Instansi']}\nMendaftar pada Hari {data['Hari']}, tanggal {data['Tanggal']}.\n")
110 |
111 |
112 |
113 |
114 |
115 |
116 |
117 |
118 |
119 # fungsi menginput nama peserta individu
120 def input_nama_peserta():
121     while True:
122         nama = input("\nNama : ")
123         if re.fullmatch(r"[A-Za-z ]+", nama):
124             return nama
125         else:
126             print("\nNama Harus Berupa Huruf.")
127
128 # mengecek nim
129 def cek_nim(nim, data_pendaftar):
130     for username, data in data_pendaftar.items():
131         if data["NIM"] == nim:
132             return True
133     return False
134
135 # fungsi memasukkan data ke dalam dictionary
136 def masukkan_data(data, username, nama, nim, prodi, instansi):
137     data[username]={
138         "Nama":nama,
139         "NIM":nim,
140         "Prodi":prodi,
141         "Instansi":instansi,
142         "Hari":hari,"Tanggal":tanggal
143     }
144
145 # Fungsi Mendaftar Akun User
146 def registrasi():
147     while True:
148         username = input("\nMasukkan username yang ingin dipakai : ")
149         password = input("Password : ")
150         # Kondisi ketika username yang dimasukkan sudah ada dalam data akun user
151         if username in users:
152             print("Username sudah dipakai")
153         # Kondisi Ketika berhasil mendaftar akun user
154         elif len(username) >= 4 and username.isalpha() and len(password) >= 4:
155             users[username]=password
156             print("Username dan password telah disimpan")
157             break
158         # Kondisi Ketika gagal mendaftar akun user
159         else:
160             print("Username/password tidak valid")
161             (username dan password Minimal 4 Huruf dan username Terdiri Dari Huruf)"""
162
163 # Fungsi Login Sebagai User
164 def login_user():
165     try:
166         hitung = 0
167         while (hitung < 3):
168             hitung += 1
169             username = input("\nusername : ")
170
171             password = input("Password : ")
172             # berhasil login sebagai user
173             if username in users and users[username] == password:
174                 print("\n\nSELAMAT DATANG ^_^")
175                 while True:
176                     print(f"""
177 1> Tampilan Lomba
178 2> Daftar
179 3> Kembali
180 4> Keluar""")
181
182             pilhan = int(input("\nMasukkan Pilihan (1/2/3) : "))
183             # Menampilkan daftar lomba
184             if pilhan == 1:
185                 menampilkan_lomba_peserta()
186             # User mendaftar pada lomba yang ada
187             elif pilhan == 2:
188                 while True:
189                     menampilkan_lomba_peserta()
190                     print("""
191                     lomba = int(input("\nLomba (1/2/3/4/5/6/0) : "))
192                     # Mendaftar lomba Network
193                     if lomba == 1:
194                         def input_nim():
195                             while True:
196                                 nim = input("\nMasukkan NIM (isi dengan 0 jika tidak memiliki NIM) : ")
197                                 if nim == "0":
198                                     return nim
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1088
1089
1090
1091
1092
1093
1094
1095
1095
1096
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1188
1189
1190
1191
1192
1193
1194
1195
1195
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1490
1491
1492
1493
1494
1495
1496
1496
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
21
```

```

198         elif nim.isdigit() and len(nim) >= 10:
199             return nim
200         elif nim.isalpha():
201             print("NIM Harus Berupa Angka")
202             continue
203         else:
204             print("NIM Harus 10 Angka atau lebih")
205             continue
206     #Input Data
207     hitung = 0
208     silahkan_isi_data()
209     while True:
210         nama = input_nama_peserta()
211         if len(nama) < 4:
212             print("Nama Harus 4 Huruf Atau Lebih")
213             continue
214         else:
215             nim = input_nim()
216             if cek_nim(nim, pendaftar_network):
217                 print("Maaf NIM sudah terdaftar! Pendaftaran gagal")
218                 break
219             else:
220                 while True:
221                     prodi = input("\nAsal Prodi (isi dengan (-) jika tidak di prodi manapun): ")
222                     if len(prodi) >= 5 or prodi == "-":
223                         if re.fullmatch("[A-Za-z -]+", prodi):
224                             while True:
225                                 instansi = input("\nAsal Instansi: ")

226                     if re.fullmatch("[A-Za-z -]", instansi) and len(instansi) >= 3:
227                         masukkan_data(pendaftar_network, username, nama, nim, prodi, instansi)
228                         data_tersimpan()
229                         break
230                     else:
231                         print("Instansi Harus Berupa Huruf dan Tidak kurang dari 3")
232                         continue
233                     else:
234                         print("Prodi Harus Berupa Huruf")
235                         continue
236                     break
237                     elif not re.fullmatch("[A-Za-z ]+", prodi) and len(prodi) >= 5:
238                         print("Prodi Harus Berupa Huruf")
239                         continue
240                     elif prodi.isdigit():
241                         print("Prodi Harus Berupa Huruf")
242                         continue
243                     else:
244                         print("Nama Prodi Harus Lengkap")
245                         continue
246                     break
247                 # Mendaftar lomba Ranking 1
248                 elif lomba == 2:
249                     def input_nim():
250                         while True:
251                             nim = input("\nMasukkan NIM (isi dengan 0 jika tidak memiliki NIM) : ")
252                             if nim == "0":
253                                 return nim
254
255                     elif nim.isdigit() and len(nim) >= 10:
256                         return nim
257                     elif nim.isalpha():
258                         print("NIM Harus Berupa Angka")
259                         continue
260                     else:
261                         print("NIM Harus 10 Angka atau lebih")
262                         continue
263                     #Input Data
264                     hitung = 0
265                     silahkan_isi_data()
266                     while True:
267                         nama = input_nama_peserta()
268                         if len(nama) < 4:
269                             print("Nama Harus 4 Huruf Atau Lebih")
270                             continue
271                         else:
272                             nim = input_nim()
273                             if cek_nim(nim, pendaftar_ranking1):
274                                 print("Maaf NIM sudah terdaftar! Pendaftaran gagal")
275                                 break
276                             else:
277                                 while True:
278                                     prodi = input("\nAsal Prodi (isi dengan (-) jika tidak di prodi manapun): ")
279                                     if len(prodi) >= 5 or prodi == "-":
280                                         if re.fullmatch("[A-Za-z -]+", prodi):
281                                             while True:
282                                                 instansi = input("\nAsal Instansi: ")

283                     if re.fullmatch("[A-Za-z -]", instansi) and len(instansi) >= 3:
284                         masukkan_data(pendaftar_ranking1, username, nama, nim, prodi, instansi)
285                         data_tersimpan()
286                         break
287                     else:
288                         print("Instansi Harus Berupa Huruf dan Tidak kurang dari 3")
289                         continue
290                     else:
291                         print("Prodi Harus Berupa Huruf")
292                         continue
293                     break
294                     elif not re.fullmatch("[A-Za-z ]+", prodi) and len(prodi) >= 5:
295                         print("Prodi Harus Berupa Huruf")
296                         continue
297                     elif prodi.isdigit():
298                         print("Prodi Harus Berupa Huruf")
299                         continue
300                     else:
301                         print("Nama Prodi Harus Lengkap")
302                         continue
303                     break
304                 # Mendaftar lomba Robotik
305                 elif lomba == 3:
306                     def input_nama_anggota(anggota_ke):
307                         while True:
308                             nama = input(f"\nNama Anggota {anggota_ke}: ")
309                             if re.fullmatch("[A-Za-z ]+", nama):
310                                 if len(nama) >= 4:

```

```

310             return nama
311         else:
312             print("Nama Harus 4 Huruf Atau Lebih")
313             continue
314     else:
315         print("Nama hanya boleh terdiri dari huruf")
316         continue
317     def input_nim(anggota_ke):
318         while True:
319             nim = input(f"\nMasukkan NIM Anggota {anggota_ke} (isi dengan 0 jika tidak memiliki NIM): ")
320             if nim == "0":
321                 return nim
322             elif nim.isdigit() and len(nim) >= 10:
323                 return nim
324             elif nim.isalpha():
325                 print("NIM Harus Berupa Angka")
326                 continue
327             else:
328                 print("NIM Harus 10 Angka atau lebih")
329                 continue
330     # Input data
331     silahkan_isi_data()
332     while True:
333         nama_tim = input("\nApakah Nama Tim Anda: ")
334         if len(nama_tim) < 4:
335             print("Nama Tim Minimal 4 Huruf")
336             continue
337         else:
338             anggota1 = input_nama_anggota(1)
339             nim1 = input_nim(1)
340             if cek_nim(nim1, pendaftar_robotik):
341                 print("Maaf NIM sudah terdaftar! Pendaftaran gagal")
342                 break
343             else:
344                 anggota2 = input_nama_anggota(2)
345                 while True:
346                     nim2 = input_nim(2)
347                     if cek_nim(nim2, pendaftar_robotik):
348                         print("Maaf NIM sudah terdaftar! Pendaftaran gagal")
349                         break
350                     elif nim1 == nim2:
351                         print("Maaf NIM tidak bisa sama")
352                         continue
353                     else:
354                         anggota3 = input_nama_anggota(3)
355                         while True:
356                             nim3 = input_nim(3)
357                             if cek_nim(nim3, pendaftar_robotik):
358                                 print("Maaf NIM sudah terdaftar! Pendaftaran gagal")
359                                 break
360                             elif nim1 == nim3 or nim2 == nim3:
361                                 print("Maaf NIM tidak bisa sama")
362                                 continue
363                             else:
364                                 while True:
365                                     prodi = input("\nMasukkan Prodi (isi dengan (-) jika tidak diambil): ")
366                                     if len(prodi) >= 5 or prodi == "-":
367                                         if re.fullmatch("[A-Za-z -]+", prodi):
368                                             while True:
369                                                 instansi = input("\nMasukkan Instansi: ")
370                                                 if re.fullmatch("[A-Za-z -]+", instansi) and len(instansi) >= 3:
371                                                     pendaftar_robotik[username] = {
372                                                         "Nama_Tim": nama_tim,
373                                                         "Anggota": [
374                                                             anggota1:nim1,
375                                                             anggota2:nim2,
376                                                             anggota3:nim3],
377                                                         "Prodi": prodi,
378                                                         "Instansi": instansi,
379                                                         "Hari":hari,"Tanggal":tanggal
380                                                       }
381                                                     data_tersimpan()
382                                                     break
383                                                 else:
384                                                     print("Instansi Harus Berupa Huruf dan Tidak kurang dari 3")
385                                                     continue
386                                             else:
387                                                 print("Prodi Harus Berupa Huruf")
388                                                 continue
389                                             elif not re.fullmatch("[A-Za-z ]+", prodi) and len(prodi) >= 5:
390                                                 print("Prodi Harus Berupa Huruf")
391                                                 continue
392                                             elif prodi.isdigit():
393                                                 print("Prodi Harus Berupa Huruf")
394
395             continue
396         else:
397             print("Nama Prodi Harus Lengkap")
398             continue
399         break
400     break
401
402     # Mendaftar Lomba UI UX
403     elif lomba == 4:
404         def input_nama_anggota(anggota_ke):
405             while True:
406                 nama = input(f"\nMasukkan Nama Anggota {anggota_ke}: ")
407                 if re.fullmatch("[A-Za-z ]+", nama):
408                     if len(nama) >= 4:
409                         return nama
410                     else:
411                         print("Nama Harus 4 Huruf Atau Lebih")
412                         continue
413                     else:
414                         print("Nama hanya boleh terdiri dari huruf")
415                         continue
416         def input_nim(anggota_ke):
417             while True:
418                 nim = input(f"\nMasukkan NIM Anggota {anggota_ke} (isi dengan 0 jika tidak memiliki NIM): ")
419                 if nim == "0":
420                     return nim
421                 elif nim.isdigit() and len(nim) >= 10:

```

```

422             return nim
423         elif nim.isalpha():
424             print("NIM Harus Berupa Angka")
425             continue
426         else:
427             print("NIM Harus 10 Angka atau lebih")
428             continue
429
430     # Input data
431     silahkan_isi_data()
432     while True:
433         nama_tim = input("\nApa Nama Tim Anda: ")
434         if len(nama_tim) < 4:
435             print("Nama Tim Minimal 4 Huruf")
436             continue
437         else:
438             anggota1 = input_nama_anggota(1)
439             nim1 = input_nim(1)
440             if cek_nim(nim1, pendaftar_uiux):
441                 print("Maaf NIM sudah terdaftar! Pendaftaran gagal")
442                 break
443             else:
444                 anggota2 = input_nama_anggota(2)
445                 while True:
446                     nim2 = input_nim(2)
447                     if cek_nim(nim2, pendaftar_uiux):
448                         print("Maaf NIM sudah terdaftar! Pendaftaran gagal")
449                         break
450                     elif nim1 == nim2:
451                         print("Maaf NIM tidak bisa sama")
452                         continue
453                     else:
454                         anggota3 = input_nama_anggota(3)
455                         while True:
456                             nim3 = input_nim(3)
457                             if cek_nim(nim3, pendaftar_uiux):
458                                 print("Maaf NIM sudah terdaftar! Pendaftaran gagal")
459                                 break
460                             elif nim1 == nim3 or nim2 == nim3:
461                                 print("Maaf NIM tidak bisa sama")
462                                 continue
463                             else:
464                                 while True:
465                                     prodi = input("\nAsal Prodi (isi dengan (-) jika tidak di prodi manapun): ")
466                                     if len(prodi) >= 5 or prodi == "-":
467                                         if re.fullmatch("[A-Za-z -]*", prodi):
468                                             while True:
469                                                 instansi = input("\nAsal Instansi: ")
470                                                 if re.fullmatch("[A-Za-z -]*", instansi) and len(instansi) >= 3:
471                                                     pendaftar_uiux[username] = {
472                                                         "Nama Tim": nama_tim,
473                                                         "Anggota": [
474                                                             anggota1:nim1,
475                                                             anggota2:nim2,
476                                                             anggota3:nim3],
477                                                         "Prodi": prodi,
478                                                         "Instansi": instansi,
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
# Mendaftar Lomba ML
elif lomba == 5:
    def input_nama_anggota(anggota_ke):
        while True:
            nama = input("\nNama Anggota (anggota_ke): ")

            if re.fullmatch("[A-Za-z ]+", nama):
                if len(nama) >= 4:
                    return nama
                else:
                    print("Nama Harus 4 Huruf Atau Lebih")
                    continue
            else:
                print("Nama hanya boleh terdiri dari huruf")
                continue

def input_nim(anggota_ke):
    while True:
        nim = input("\nMasukkan NIM Anggota (anggota_ke) (isi dengan 0 jika tidak memiliki NIM): ")
        if nim == "0":
            return nim
        elif nim.isdigit() and len(nim) >= 10:
            return nim
        elif nim.isalpha():
            print("NIM Harus Berupa Angka")
            continue
        else:
            print("NIM Harus 10 Angka atau lebih")
            continue

# Input data
silahkan_isi_data()
while True:
    nama_tim = input("Apa Nama Tim Anda: ")
    if len(nama_tim) < 4:
        print("Nama Tim Minimal 4 Huruf")

```

```

534         continue
535     else:
536         anggota1 = input_nama_anggota(1)
537         nim1 = input_nim(1)
538         if cek_nim(nim1, pendaftar_mobilelegends):
539             print("Maaf NIM sudah terdaftar! Pendaftaran gagal")
540             break
541         else:
542             anggota2 = input_nama_anggota(2)
543             while True:
544                 nim2 = input_nim(2)
545                 if cek_nim(nim2, pendaftar_mobilelegends):
546                     print("Maaf NIM sudah terdaftar! Pendaftaran gagal")
547                     break
548                 elif nim1 == nim2:
549                     print("Maaf NIM tidak bisa sama")
550                     continue
551                 else:
552                     anggota3 = input_nama_anggota(3)
553                     while True:
554                         nim3 = input_nim(3)
555                         if cek_nim(nim3, pendaftar_mobilelegends):
556                             print("Maaf NIM sudah terdaftar! Pendaftaran gagal")
557                             break
558                         elif nim1 == nim3 or nim2 == nim3:
559                             print("Maaf NIM tidak bisa sama")
560                             continue
561                         else:
562
563             anggota4 = input_nama_anggota(4)
564             while True:
565                 nim4 = input_nim(4)
566                 if cek_nim(nim4, pendaftar_mobilelegends):
567                     print("Maaf NIM sudah terdaftar! Pendaftaran gagal")
568                     break
569                 elif nim4 == nim1 or nim4 == nim2 or nim4 == nim3:
570                     print("Maaf NIM tidak bisa sama")
571                     continue
572                 else:
573                     anggota5 = input_nama_anggota(5)
574                     while True:
575                         nim5 = input_nim(5)
576                         if cek_nim(nim5, pendaftar_mobilelegends):
577                             print("Maaf NIM sudah terdaftar! Pendaftaran gagal")
578                             break
579                         elif nim5 == nim1 or nim5 == nim2 or nim5 == nim3 or nim5 == nim4:
580                             print("Maaf NIM tidak bisa sama")
581                             continue
582                         else:
583                             while True:
584                                 prodi = input("\nAsal Prodi (isi dengan (-) jika tidak di prodi manapun): ")
585                                 if len(prodi) >= 5 or prodi == "-":
586                                     if re.fullmatch("[A-Za-z -]+", prodi):
587                                         while True:
588                                             instansi = input("\nAsal Instansi: ")
589                                             if re.fullmatch("[A-Za-z -]+", instansi) and len(instansi) >= 3:
590                                                 pendaftar_mobilelegends[username] = {
591
592                                         "Nama_Tim": nama_tim,
593                                         "Anggota": [
594                                             anggota1:nim1,
595                                             anggota2:nim2,
596                                             anggota3:nim3,
597                                             anggota4:nim4,
598                                             anggota5:nim5],
599                                         "Prodi": prodi,
600                                         "Instansi": instansi,
601                                         "Hari":hari,"Tanggal":tanggal
602                                         }
603                                         data_tersimpan()
604                                         break
605                                         else:
606                                             print("Instansi Harus Berupa Huruf dan Tidak kurang dari 3")
607                                         continue
608                                         else:
609                                             print("Prodi Harus Berupa Huruf")
610                                         continue
611                                         elif re.fullmatch("[A-Za-z ]+", prodi) and len(prodi) >= 5:
612                                             print("Prodi Harus Berupa Huruf")
613                                             continue
614                                         elif prodi.isdigit():
615                                             print("Prodi Harus Berupa Huruf")
616                                             continue
617                                         else:
618                                             print("Nama Prodi Harus Lengkap")
619                                             continue
620
621                                         break
622                                         break
623                                         break
624                                         # Mendaftar lomba Valorant
625                                         elif lomba == 6:
626                                             def input_nama_anggota(anggota_ke):
627                                                 while True:
628                                                     nama = input(f"\nNama Anggota {anggota_ke}: ")
629                                                     if re.fullmatch("[A-Za-z ]+", nama):
630                                                         if len(nama) >= 4:
631                                                             return nama
632                                                         else:
633                                                             print("Nama Harus 4 Huruf Atau Lebih")
634                                                             continue
635                                                         else:
636                                                             print("Nama hanya boleh terdiri dari huruf")
637                                                             continue
638                                             def input_nim(anggota_ke):
639                                                 while True:
640                                                     nim = input(f"\nMasukkan NIM Anggota {anggota_ke} (isi dengan 0 jika tidak memiliki NIM): ")
641                                                     if nim == "0":
642                                                         return nim
643                                                     elif nim.isdigit() and len(nim) >= 10:
644                                                         return nim
645                                                     elif not nim.isalpha():
646

```

```

646     print("NIM Harus Berupa Angka")
647     continue
648   else:
649     print("NIM Harus 10 Angka atau lebih")
650     continue
651   # Input data
652   silahkan_isi_data
653   while True:
654     nama_tim = input("Apa Nama Tim Anda: ")
655     if len(nama_tim) < 4:
656       print("Nama Tim Minimal 4 Huruf")
657       continue
658     else:
659       anggota1 = input_nama_anggota(1)
660       nim1 = input_nim(1)
661       if cek_nim(nim1, pendaftar_valorant):
662         print("Maaf NIM sudah terdaftar! Pendaftaran gagal")
663         break
664     else:
665       anggota2 = input_nama_anggota(2)
666       while True:
667         nim2 = input_nim(2)
668         if cek_nim(nim2, pendaftar_valorant):
669           print("Maaf NIM sudah terdaftar! Pendaftaran gagal")
670           break
671         elif nim1 == nim2:
672           print("Maaf NIM tidak bisa sama")
673
674       continue
675     else:
676       anggota3 = input_nama_anggota(3)
677       while True:
678         nim3 = input_nim(3)
679         if cek_nim(nim3, pendaftar_valorant):
680           print("Maaf NIM sudah terdaftar! Pendaftaran gagal")
681           break
682         elif nim1 == nim3 or nim2 == nim3:
683           print("Maaf NIM tidak bisa sama")
684           continue
685     else:
686       anggota4 = input_nama_anggota(4)
687       while True:
688         nim4 = input_nim(4)
689         if cek_nim(nim4, pendaftar_valorant):
690           print("Maaf NIM sudah terdaftar! Pendaftaran gagal")
691           break
692         elif nim1 == nim4 or nim2 == nim4 or nim3 == nim4:
693           print("Maaf NIM tidak bisa sama")
694           continue
695     else:
696       anggota5 = input_nama_anggota(5)
697       while True:
698         nim5 = input_nim(5)
699         if cek_nim(nim5, pendaftar_valorant):
700           print("Maaf NIM sudah terdaftar! Pendaftaran gagal")
701           break
702
703       elif nim5 == nim1 or nim5 == nim2 or nim5 == nim3 or nim5 == nim4:
704         print("Maaf NIM tidak bisa sama")
705         continue
706     else:
707       while True:
708         prodi = input("\nMasukkan Prodi. (isi dengan (-) jika tidak di prodi manapun): ")
709         if len(prodi) >= 5 or prodi == "-":
710           if re.fullmatch("[A-Za-z-]", prodi):
711             while True:
712               instansi = input("\nMasukkan Instansi: ")
713               if re.fullmatch("[A-Za-z-]", instansi) and len(instansi) >= 3:
714                 pendaftar_valorant[username] = {
715                   "Nama Tim": nama_tim,
716                   "Anggota": [
717                     anggota1:nim1,
718                     anggota2:nim2,
719                     anggota3:nim3,
720                     anggota4:nim4,
721                     anggota5:nim5],
722                   "Prodi": prodi,
723                   "Instansi": instansi,
724                   "Hari":hari,"Tanggal":tanggal
725                 }
726                 data_tersimpan()
727                 break
728               else:
729                 print("Instansi Harus Berupa Huruf dan Tidak kurang dari 3")
730                 continue
731
732             break
733
734           else:
735             print("Prodi Harus Berupa Huruf")
736             continue
737           elif not re.fullmatch("[A-Za-z-]", prodi) and len(prodi) >= 5:
738             print("Prodi Harus Berupa Huruf")
739             continue
740           elif prodi.isdigit():
741             print("Prodi Harus Berupa Huruf")
742             continue
743           else:
744             print("Nama Prodi Harus Lengkap")
745             continue
746
747           break
748
749         elif lomba == 0:
750           break
751         else:
752           print("\nlomba tidak valid")
753
754       # Kembali ke menu sebelumnya
755       elif pilihan == 3:
756         return False
757       # Error handling ketika memasukkan pilihan yang tidak sesuai
758     else:

```

```

757         print("\nPILIHAN TIDAK VALID")
758     # Gagal login sebagai user
759     else:
760         print("\n\nusername/password salah")
761         if hitung == 3:
762             print("\nsilahkan coba lagi nanti")
763             break
764     # Error handling try except
765     except ValueError:
766         print("\n!!!Maaf anda salah memasukan tipe data!!!")
767     except SyntaxError:
768         print("\n!!!Maaf ada kesalahan penulisan syntax sehingga terjadi error!!!")
769     except IndexError:
770         print("\n!!!Maaf ada kesalahan penulisan index sehingga terjadi error!!!")
771     except AssertionError:
772         print("\n!!!Maaf syarat tidak terpenuhi sehingga terjadi error!!!")
773     except AttributeError:
774         print("\n!!!Maaf ada kesalahan penulisan attribute sehingga terjadi error!!!")
775     except ImportError:
776         print("\n!!!Maaf gagal mengimport sehingga terjadi error!!!")
777     except KeyError:
778         print("\n!!!Maaf key tidak ditemukan sehingga terjadi error!!!")
779     except NameError:
780         print("\n!!!Maaf variabel belum didefinisikan sehingga terjadi error!!!")
781     except MemoryError:
782         print("\n!!!Maaf memory tidak cukup sehingga terjadi error!!!")
783     except TypeError:
784         print("\n!!!Maaf ada kesalahan penggunaan sehingga terjadi error!!!")

785     except IndentationError:
786         print("\n!!!Maaf ada kesalahan indentasi sehingga terjadi error!!!")
787     except FileNotFoundError:
788         print("\n!!!Maaf file tidak ditemukan sehingga terjadi error!!!")
789
790 # Fungsi login sebagai admin
791 def login_admin():
792     # Error Handling try except
793     try:
794         hitung = 0
795         while (hitung < 3):
796             hitung += 1
797             username = input("\nMasukkan username : ")
798             password = input("Password : ")
799             # Berhasil login sebagai admin
800             if username in admin and password == admin[username]:
801                 print("\n SELAMAT DATANG DI MENU ADMIN")
802                 while True:
803                     menu_admin()
804                     pilihan = int(input("Masukkan Pilihan Anda (1/2/3/4/5) : "))
805                     # Menampilkan data pendaftar lomba-lomba yang ada
806                     if pilihan == 1:
807                         menampilkan_lomba_admin()
808                         nama_lomba = int(input("Pilih lomba yang ingin dilihat [1/2/3/4/5/6] : "))
809                         # Menampilkan data pendaftar lomba Network
810                         if nama_lomba == 1:
811                             print(f"\n{'*'*55}\nPENDAFTAR LOMBA NETWORK")
812                             menampilkan_data_pendaftar(pendaftar_network)
813
814                         input("Tekan enter untuk melanjutkan...")
815                         # Menampilkan data pendaftar lomba Ranking 1
816                         elif nama_lomba == 2:
817                             print(f"\n{'*'*55}\nPENDAFTAR LOMBA RANKING #1")
818                             menampilkan_data_pendaftar(pendaftar_ranking1)
819                             input("Tekan enter untuk melanjutkan...")
820                         # Menampilkan data pendaftar lomba Robotik
821                         elif nama_lomba == 3:
822                             print(f"\n{'*'*55}\nPENDAFTAR LOMBA ROBOTIK")
823                             menampilkan_data_pendaftar_tim(pendaftar_robotik)
824                             input("Tekan enter untuk melanjutkan...")
825                         # Menampilkan data pendaftar lomba UI UX
826                         elif nama_lomba == 4:
827                             print(f"\n{'*'*55}\nPENDAFTAR LOMBA UI/UX")
828                             menampilkan_data_pendaftar_tim(pendaftar_uiux)
829                             input("Tekan enter untuk melanjutkan...")
830                         # Menampilkan data pendaftar lomba ML
831                         elif nama_lomba == 5:
832                             print(f"\n{'*'*55}\nPENDAFTAR LOMBA MOBILE LEGENDS")
833                             menampilkan_data_pendaftar_tim(pendaftar_mobilelegends)
834                             input("Tekan enter untuk melanjutkan...")
835                         # Menampilkan data pendaftar lomba Valorant
836                         elif nama_lomba == 6:
837                             print(f"\n{'*'*55}\nPENDAFTAR LOMBA VALORANT")
838                             menampilkan_data_pendaftar_tim(pendaftar_valorant)
839                             input("Tekan enter untuk melanjutkan...")
840                         # Error handling ketika inputan tidak benar
841                         else:
842                             print("\nInputan tidak valid")
843
844                         # Mengubah data pendaftar
845                         elif pilihan == 2:
846                             menampilkan_lomba_admin()
847                             lomba = int(input("\nPilih lomba yang ingin diubah [1/2/3/4/5/6] : "))
848                             # Mengucapkan data pendaftar lomba Network
849                             if lomba == 1:
850                                 print(f"\n{'*'*55}\nPENDAFTAR LOMBA NETWORK")
851                                 if not pendaftar_network:
852                                     print("Belum ada peserta yang mendaftar")
853                                     print("-----")
854                                 else:
855                                     for akun, data in pendaftar_network.items():
856                                         print(f"\n{'*'*55}")
857                                         Akun : {akun}
858                                         Nama : {data["Nama"]}
859                                         NIM : {data["NIM"]}
860                                         Prodi : {data["Prodi"]}
861                                         Instansi : {data["Instansi"]}
862                                         Mendaftar pada Hari {data["Hari"]}, tanggal {data["Tanggal"]}.
863                                         ")
864                                         nama_ubah = input("Masukkan Nama Pendaftaran : ")
865                                         for user, data in pendaftar_network.items():
866                                             if nama_ubah == data["Nama"]:
867                                                 print("\nPROSES UBAH DATA PENDAFTAR LOMBA NETWORK")
868                                                 def input_nim():

```

```

869         while True:
870             nim = input("\nMasukkan NIM (isi dengan 0 jika tidak memiliki NIM) : ")
871             if nim == "0":
872                 return nim
873             elif nim.isdigit() and len(nim) >= 10:
874                 return nim
875             elif nim.isalpha():
876                 print("NIM Harus Berupa Angka")
877                 continue
878             else:
879                 print("NIM Harus 10 Angka atau lebih")
880                 continue
881         while True:
882             nama_update = input_nama_peserta()
883             if len(nama_update) < 4:
884                 print("Nama Harus 4 Huruf atau Lebih")
885                 continue
886             else:
887                 NIM_update = input_nim()
888                 if cek_nim(NIM_update, pendaftar_network):
889                     print("Maaf NIM sudah terdaftar! Pendaftaran gagal")
890                     break
891                 else:
892                     while True:
893                         prodi_update = input("\nAsal Prodi (isi dengan (-) jika tidak di prodi manapun): ")
894                         if len(prodi_update) >= 5 or prodi_update == "-":
895                             if re.fullmatch("[A-Za-z -]*", prodi_update):
896                                 while True:
897                                     instansi_update = input("\nInstansi: ")
898                                     if re.fullmatch("[A-Za-z -]*", instansi_update) and len(instansi_update) >= 3:
899                                         masukkan_data(pendaftar_network, user, nama_update, NIM_update, prodi_update, instansi_update)
900                                         data_tersimpan()
901                                         break
902                                     else:
903                                         print("Instansi Harus Berupa Huruf dan Tidak kurang dari 3")
904                                         continue
905                                     else:
906                                         print("Prodi Harus Berupa Huruf")
907                                         continue
908                                     break
909                                 elif not re.fullmatch("[A-Za-z ]*", prodi_update) and len(prodi_update) >= 5:
910                                     print("Prodi Harus Berupa Huruf")
911                                     continue
912                                 elif prodi_update.isdigit():
913                                     print("Prodi Harus Berupa Huruf")
914                                     continue
915                                 else:
916                                     print("Nama Prodi Harus Lengkap")
917                                     continue
918                                 break
919                             else:
920                                 print("\nNama Tidak Ditemukan")
921                             # Mengubah data pendaftar lomba Rangking 1
922                             elif lomba == 2:
923                                 print("\n(" + "*" * 55 + ")PENDAFTAR LOMBA RANKING #1")
924                                 if not pendaftar_ranking1:
925                                     print('')
926                                     -----
927                                     Belum ada peserta yang mendaftar
928                                     -----
929                                     else:
930                                         for akun, data in pendaftar_ranking1.items():
931                                             print(f"\n{data['=" + "*" * 55 + "]")
932                                         Akun : {akun}
933                                         Nama : {data["Nama"]}
934                                         NIM : {data["NIM"]}
935                                         Prodi : {data["Prodi"]}
936                                         Instansi : {data["Instansi"]}
937                                         Mendaftar pada Hari [{data["Hari"]}], tanggal {data["Tanggal"]}],
```

....

```

938                                         ....)
939                                         nama_ubah = input("Masukkan Nama Pendaftar : ")
940                                         for user, data in pendaftar_ranking1.items():
941                                             if nama_ubah == data["Nama"]:
942                                                 print("\nPROSES UBAH DATA PENDAFTAR LOMBA RANGKING 1")
943                                                 def input_nim():
944                                                     while True:
945                                                         nim = input("\nMasukkan NIM (isi dengan 0 jika tidak memiliki NIM) : ")
946
947                                                         return nim
948                                                         elif nim.isdigit() and len(nim) >= 10:
949                                                             return nim
950                                                         elif nim.isalpha():
951                                                             print("NIM Harus Berupa Angka")
952                                                             continue
953                                                         else:
954                                                             print("NIM Harus 10 Angka atau lebih")
955                                                             continue
956         while True:
957             nama_update = input_nama_peserta()
958             if len(nama_update) < 4:
959                 print("Nama Harus 4 Huruf atau Lebih")
960                 continue
961             else:
962                 NIM_update = input_nim()
963                 if cek_nim(NIM_update, pendaftar_ranking1):
964                     print("Maaf NIM sudah terdaftar! Pendaftaran gagal")
965                     break
966                 else:
967                     while True:
968                         prodi_update = input("\nAsal Prodi (isi dengan (-) jika tidak di prodi manapun): ")
969                         if len(prodi_update) >= 5 or prodi_update == "-":
970                             if re.fullmatch("[A-Za-z -]*", prodi_update):
971                                 while True:
972                                     instansi_update = input("\nInstansi: ")
973                                     if re.fullmatch("[A-Za-z -]*", instansi_update) and len(instansi_update) >= 3:
974                                         masukkan_data(pendaftar_ranking1, user, nama_update, NIM_update, prodi_update, instansi_update)
```

....

```

975                                         data_tersimpan()
976                                         break
977                                         else:
978                                         print("Instansi Harus Berupa Huruf dan Tidak kurang dari 3")
979                                         continue
980                                         else:
981                                         print("Prodi Harus Berupa Huruf")
982                                         continue
983                                         break
984                                         elif not re.fullmatch("[A-Za-z ]*", prodi_update) and len(prodi_update) >= 5:
985                                         print("Prodi Harus Berupa Huruf")
986                                         continue
987                                         elif prodi_update.isdigit():
988                                         print("Prodi Harus Berupa Huruf")
989                                         continue
990                                         else:
991                                         print("Nama Prodi Harus Lengkap")
992                                         continue
993                                         break
994                                         else:
995                                         print("\nNama Tidak Ditemukan")
996                                         # Mengubah data pendaftar lomba Robotik
997                                         elif lomba == 3:
998                                             print("\n(" + "*" * 55 + ")PENDAFTAR LOMBA ROBOTIK")
999                                             for akun, data in pendaftar_robotik.items():
1000                                                 print(f"\n{data['=" + "*" * 55 + "]")
1001                                         Akun : {akun}
1002                                         Nama Tim : {data["Nama Tim"]}
```

```

1003     Anggota : {data["Anggota"]}
1004     Prodi : {data["Prodi"]}
1005     Instansi : {data["Instansi"]}
1006     Mendaftar pada Hari {data["Hari"]}, tanggal {data["Tanggal"]}.
1007 """
1008     nama_ubah = input("Masukkan Nama Tim Pendaftar : ")
1009     for user, data in pendaftar_robottik.items():
1010         if nama_ubah == data["Nama Tim"]:
1011             print("\nPROSES UBAH DATA PENDAFTAR LOMBA ROBOTIK")
1012             def input_nama_anggota(anggota_ke):
1013                 while True:
1014                     nama = input(f"Nama Anggota {anggota_ke}: ")
1015                     if nama.isalpha():
1016                         if len(nama) >= 4:
1017                             return nama
1018                         else:
1019                             print("Nama Minimal 4 Huruf")
1020                     else:
1021                         print("Error: Nama hanya boleh terdiri dari huruf.")
1022
1023             def input_nim(anggota_ke):
1024                 while True:
1025                     try:
1026                         NIM = input(f"Masukkan NIM Anggota {anggota_ke} (isi dengan 0 jika tidak memiliki NIM): ")
1027                         if NIM == "0":
1028                             return NIM
1029                         elif NIM.isdigit() and len(NIM) >= 10:
1030                             return NIM
1031
1032                         elif NIM.isalpha():
1033                             print("NIM Harus Berupa Angka")
1034                         else:
1035                             print("NIM Harus 10 Angka atau lebih")
1036                         except ValueError:
1037                             print("Error: NIM harus berupa angka.")
1038
1039             hitung = 0
1040             while True:
1041                 nama_tim_update = input("Apa Nama Tim Anda: ")
1042                 if len(nama_tim_update) < 4:
1043                     print("Nama Tim Minimal 4 Huruf")
1044                     hitung += 1
1045                 else:
1046                     anggota1_update = input_nama_anggota(1)
1047                     anggota2_update = input_nama_anggota(2)
1048                     anggota3_update = input_nama_anggota(3)
1049                     NIM3_update = input_nim(3)
1050
1051                     prodi_update = input("Asal Prodi (isi dengan (-) jika tidak di prodi manapun): ")
1052                     if len(prodi_update) >= 5 or prodi_update == "-":
1053                         instansi_update = input("Masukkan Asal Instansi : ")
1054                         pendaftar_robottik[user] = {
1055                             "Nama Tim":nama_tim_update,
1056                             "Anggota":{
1057                                 "anggota1_update":NIM1_update,
1058                                 "anggota2_update":NIM2_update,
1059                                 "anggota3_update":NIM3_update},
1060                             "Prodi":prodi_update,
1061                             "Instansi":instansi_update,
1062                             "Hari":hari,"Tanggal":tanggal
1063                         }
1064                     else:
1065                         print("Nama Prodi Tidak Valid")
1066
1067                 else:
1068                     print("\nNama Tim Tidak Ditemukan")
1069
1070             # Mengubah data pendaftar lomba UI UX
1071             elif lomba == 4:
1072                 print(f"\n{'*'*55}\nPENDAFTAR LOMBA UI/UX")
1073                 for akun, data in pendaftar_uix.items():
1074                     print(f"\n{'*'*55}")
1075
1076             Akun : {akun}
1077             Nama Tim : {data["Nama Tim"]}
1078             Anggota : {data["Anggota"]}
1079             Prodi : {data["Prodi"]}
1080             Instansi : {data["Instansi"]}
1081             Mendaftar pada Hari {data["Hari"]}, tanggal {data["Tanggal"]}.
1082 """
1083     nama_ubah = input("Masukkan Nama Tim Pendaftar : ")
1084     for user, data in pendaftar_uix.items():
1085         if nama_ubah == data["Nama Tim"]:
1086             print("\nPROSES UBAH DATA PENDAFTAR LOMBA UI/UX")
1087             def input_nama_anggota(anggota_ke):
1088                 while True:
1089                     nama = input(f"Nama Anggota {anggota_ke}: ")
1090                     if nama.isalpha():
1091                         if len(nama) >= 4:
1092                             return nama
1093                         else:
1094                             print("Nama Minimal 4 Huruf")
1095                     else:
1096                         print("Error: Nama hanya boleh terdiri dari huruf.")
1097
1098             def input_nim(anggota_ke):
1099                 while True:
1100                     try:
1101                         NIM = input(f"Masukkan NIM Anggota {anggota_ke} (isi dengan 0 jika tidak memiliki NIM): ")
1102                         if NIM == "0":
1103                             return NIM
1104                         elif NIM.isdigit() and len(NIM) >= 10:
1105                             return NIM
1106                         elif NIM.isalpha():
1107                             print("NIM Harus Berupa Angka")
1108                         else:
1109                             print("NIM Harus 10 Angka atau lebih")
1110                         except ValueError:
1111                             print("Error: NIM harus berupa angka.")
1112
1113             hitung = 0
1114             while True:
1115                 nama_tim_update = input("Apa Nama Tim Anda: ")
1116                 if len(nama_tim_update) <= 4:
1117                     print("Nama Tim Minimal 4 Huruf")

```

```

1115             hitung += 1
1116         else:
1117             anggota1_update = input_nama_anggota(1)
1118             NIM1_update = input_nim(1)
1119             anggota2_update = input_nama_anggota(2)
1120             NIM2_update = input_nim(2)
1121             anggota3_update = input_nama_anggota(3)
1122             NIM3_update = input_nim(3)
1123
1124             while True:
1125                 prodi_update = input("Asal Prodi (isi dengan (-) jika tidak di prodi manapun): ")
1126                 if len(prodi_update) >= 5 or prodi_update == "-":
1127                     instansi_update = input("Masukkan Asal Instansi : ")
1128                     pendaftra_uix(user) = {
1129                         "Nama Tim":nama_tim_update,
1130                         "Anggota": [
1131                             anggota1_update:NIM1_update,
1132                             anggota2_update:NIM2_update,
1133                             anggota3_update:NIM3_update],
1134                         "Prodi":prodi_update,
1135                         "Instansi":instansi_update,
1136                         "Hari":hari,"Tanggal":tanggal
1137                     }
1138                     break
1139                 else:
1140                     print("Nama Prodi Tidak Valid")
1141             break
1142         print("\nNama Tim Tidak Ditemukan")
1143
1144     # Mengubah data pendaftar lomba ML
1145     elif lomba == 5:
1146         print(f"\n{'*'*65}\nPENDAFTAR LOMBA MOBILE LEGENDS")
1147         for akun, data in pendaftar_mobilelegends.items():
1148             print(f"\n{akun} : {data}")
1149             Nama_Tim : {data["Nama_Tim"]}
1150             Anggota : {data["Anggota"]}
1151             Prodi : {data["Prodi"]}
1152             Instansi : {data["Instansi"]}
1153             Mendaftar pada Hari {data["Hari"]}, tanggal {data["Tanggal"]}.
1154
1155             nama_ubah = input("Masukkan Nama Tim Pendaftar : ")
1156             for user, data in pendaftar_mobilelegends.items():
1157                 if nama_ubah == data["Nama_Tim"]:
1158                     print("\nPROSES UBAH DATA PENDAFTAR LOMBA MOBILE LEGENDS")
1159                     def input_name_anggota(anggota_ke):
1160                         while True:
1161                             nama = input(f"Nama Anggota {anggota_ke}: ")
1162                             if nama.isalpha():
1163                                 if len(nama) >= 4:
1164                                     return nama
1165                                 else:
1166                                     print("Nama Minimal 4 Huruf")
1167                                 else:
1168                                     print("Error: Nama hanya boleh terdiri dari huruf.")
1169
1170             def input_nim(anggota_ke):
1171
1172             while True:
1173                 try:
1174                     NIM = input(f"Masukkan NIM Anggota {anggota_ke} (isi dengan 0 jika tidak memiliki NIM): ")
1175                     if NIM == "0":
1176                         return NIM
1177                     elif NIM.isdigit() and len(NIM) >= 10:
1178                         return NIM
1179                     elif NIM.isalpha():
1180                         print("NIM Harus Berupa Angka")
1181                     else:
1182                         print("NIM Harus 10 Angka atau lebih")
1183                 except ValueError:
1184                     print("Error: NIM harus berupa angka.")
1185             hitung = 0
1186             while True:
1187                 nama_tim = input("Apa Nama Tim Anda: ")
1188                 if len(nama_tim) < 4:
1189                     print("Nama Tim Minimal 4 Huruf")
1190                     hitung += 1
1191                 else:
1192                     anggota1 = input_nama_anggota(1)
1193                     NIM1 = input_nim(1)
1194                     anggota2 = input_nama_anggota(2)
1195                     NIM2 = input_nim(2)
1196                     anggota3 = input_nama_anggota(3)
1197                     NIM3 = input_nim(3)
1198                     anggota4 = input_nama_anggota(4)
1199                     NIM4 = input_nim(4)
1200
1201                     anggota5 = input_nama_anggota(5)
1202                     NIM5 = input_nim(5)
1203
1204                     while True:
1205                         anggota6 = input("Nama Anggota 6 (opsional, isi dengan (-) jika tidak ada) :")
1206                         if len(anggota6) >= 4 and anggota6.isalpha() or anggota6 == "-":
1207                             NIM6 = input("Masukkan NIM Anggota 6 (isi dengan 0 jika tidak ada) :")
1208                             if NIM6.isdigit() and len(NIM6) >= 10 or NIM6 == "0":
1209                                 break
1210                             elif NIM6.isalpha():
1211                                 print("NIM Harus Berupa Angka")
1212                             else:
1213                                 print("NIM Harus Lengkap")
1214                             else:
1215                                 print("Nama Minimal 4 Huruf")
1216
1217                     anggota7 = input("Nama Anggota 7 (opsional, isi dengan (-) jika tidak ada) :")
1218                     if len(anggota7) >= 4 and anggota7.isalpha() or anggota7 == "-":
1219                         NIM7 = input("Masukkan NIM Anggota 7 (isi dengan 0 jika tidak ada) :")
1220                         if NIM7.isdigit() and len(NIM7) >= 10 or NIM7 == "0":
1221                             break
1222                         elif NIM7.isalpha():
1223                             print("NIM Harus Berupa Angka")
1224                         else:
1225                             print("NIM Harus Lengkap")
1226                         else:
1227                             print("Nama Minimal 4 Huruf")
1228
1229                     while True:
1230                         prodi = input("Asal Prodi (isi dengan (-) jika tidak di prodi manapun): ")

```

```

1227             if len(prodi) >= 5 or prodi == "-":
1228                 instansi = input("Asal Instansi: ")
1229                 pendaftar_mobilelegends[user] = {
1230                     "Nama Tim": nama_tim,
1231                     "Anggota": {
1232                         "anggota1:NIM1",
1233                         "anggota2:NIM2",
1234                         "anggota3:NIM3",
1235                         "anggota4:NIM4",
1236                         "anggota5:NIM5",
1237                         "anggota6:NIM6",
1238                         "anggota7:NIM7",
1239                     },
1240                     "Prodi": prodi,
1241                     "Instansi": instansi,
1242                     "Hari":hari,"Tanggal":tanggal
1243                 }
1244                 print("\nData Telah Disimpan")
1245                 break
1246             else:
1247                 print("\nNama Prodi Harus Lengkap")
1248                 break
1249             if hitung == 4:
1250                 print("\nGagal Input Data")
1251                 break
1252             else:
1253                 print("\nNama Tim Tidak Ditemukan")
1254             # Mengubah data pendaftar lomba Valorant
1255         elif lomba == 6:

```

```

1255             print(f"\n{=""*55}\nPENGAFTAR LOMBA VALORANT")
1256             for akun, data in pendaftar_valorant.items():
1257                 print(f"""\n{=""*55}
1258                 Akun : {akun}
1259                     Nama Tim : {data["Nama Tim"]}
1260                     Anggota : {data["Anggota"]}
1261                     Prodi : {data["Prodi"]}
1262                     Instansi : {data["Instansi"]}
1263                     Mendaftar pada Hari {data["Hari"]}, tanggal {data["Tanggal"]}.
1264                 """)
1265             nama_ubah = input("Masukkan Nama Tim Pendaftar : ")
1266             for user, data in pendaftar_valorant.items():
1267                 if nama_ubah == data["Nama Tim"]:
1268                     print("\nPROSES UBAH DATA PENDAFTAR LOMBA VALORANT")
1269                     def input_nama_anggota(anggota_ke):
1270                         while True:
1271                             nama = input(f"Nama Anggota {anggota_ke}: ")
1272                             if nama.isalpha():
1273                                 if len(nama) >= 4:
1274                                     return nama
1275                                 else:
1276                                     print("Nama Minimal 4 Huruf")
1277                                 else:
1278                                     print("Error: Nama hanya boleh terdiri dari huruf.")
1279
1280                     def input_nim(anggota_ke):
1281                         while True:
1282                             try:

```

```

1283     NIM = input(f"Masukkan NIM Anggota {anggota_ke} (isi dengan 0 jika tidak memiliki NIM): ")
1284     if NIM == "0":
1285         return NIM
1286     elif NIM.isdigit() and len(NIM) >= 10:
1287         return NIM
1288     elif NIM.isalpha():
1289         print("NIM Harus Berupa Angka")
1290     else:
1291         print("NIM Harus 10 Angka atau lebih")
1292     except ValueError:
1293         print("Error: NIM harus berupa angka.")
1294     hitung = 0
1295     while True:
1296         nama_tim = input("Apa Nama Tim Anda: ")
1297         if len(nama_tim) < 4:
1298             print("Nama Tim Minimal 4 Huruf")
1299             hitung += 1
1300         else:
1301             anggota1 = input_nama_anggota(1)
1302             NIM1 = input_nim(1)
1303             anggota2 = input_nama_anggota(2)
1304             NIM2 = input_nim(2)
1305             anggota3 = input_nama_anggota(3)
1306             NIM3 = input_nim(3)
1307             anggota4 = input_nama_anggota(4)
1308             NIM4 = input_nim(4)
1309             anggota5 = input_nama_anggota(5)

1310             NIM5 = input_nim(5)
1311             while True:
1312                 anggota6 = input("Nama Anggota 6 (opsional, isi dengan (-) jika tidak ada) :")
1313                 if len(anggota6) >= 4 and anggota6.isalpha() or anggota6 == "-":
1314                     NIM6 = input("Masukkan NIM Anggota 6 (isi dengan 0 jika tidak ada) :")
1315                     if NIM6.isdigit() and len(NIM6) >= 10 or NIM6 == "0":
1316                         break
1317                     elif NIM6.isalpha():
1318                         print("NIM Harus Berupa Angka")
1319                     else:
1320                         print("NIM Harus Lengkap")
1321                 else:
1322                     print("Nama Minimal 4 Huruf")
1323             while True:
1324                 anggota7 = input("Nama Anggota 7 (opsional, isi dengan (-) jika tidak ada) :")
1325                 if len(anggota7) >= 4 and anggota7.isalpha() or anggota7 == "-":
1326                     NIM7 = input("Masukkan NIM Anggota 7 (isi dengan 0 jika tidak ada) :")
1327                     if NIM7.isdigit() and len(NIM7) >= 10 or NIM7 == "0":
1328                         break
1329                     elif NIM7.isalpha():
1330                         print("NIM Harus Berupa Angka")
1331                     else:
1332                         print("NIM Harus Lengkap")
1333                 else:
1334                     print("Nama Minimal 4 Huruf")
1335             while True:
1336                 prodi = input("Asal Prodi (isi dengan (-) jika tidak di prodi manapun): ")
1337                 if len(prodi) >= 5 or prodi == "-":

```

```

1338         instansi = input("Asal Instansi: ")
1339         pendaftar_valorant[user] = {
1340             "Nama Tim": nama_tim,
1341             "Anggota": [
1342                 anggota1:NIM1,
1343                 anggota2:NIM2,
1344                 anggota3:NIM3,
1345                 anggota4:NIM4,
1346                 anggota5:NIM5,
1347                 anggota6:NIM6,
1348                 anggota7:NIM7],
1349             "Prodi": prodi,
1350             "Instansi": instansi,
1351             "Hari":hari,"Tanggal":tanggal
1352         }
1353         print("\nData Telah Disimpan")
1354         break
1355     else:
1356         print("\nNama Prodi Harus Lengkap")
1357         break
1358     if hitung == 4:
1359         print("\nGagal Input Data")
1360         break
1361     break
1362 else:
1363     print("\nNama Tim Tidak Ditemukan")
1364 # Error handling ketika lomba yang diinput salah/ tidak ada
1365 else:

```

```

1366         print("Nama Lomba Tidak Valid")
1367     # Menampilkan data akun user
1368     elif pilihan == 3:
1369         print("\nDaftar Akun User :")
1370         for akun, pw in users.items():
1371             print(akun)
1372         # Menghapus data akun user
1373     elif pilihan == 4:
1374         print(""""
1375         -----
1376         |Pilih Lomba:
1377         |  1> Network          4> UI/UX
1378         |  2> Ranking 1          5> Mobile Legends
1379         |  3> Robotik            6> Valorant
1380         |_"""
1381     """
1382     pilih_lomba = int(input("\nPilih Lomba : "))
1383     if pilih_lomba == 1:
1384         print("\n{"+"*55}\nPENDAFTAR LOMBA NETWORK")
1385         menampilkan_data_pendaftar(pendaftar_network)
1386         lomba = "Network"
1387     elif pilih_lomba == 2:
1388         print("\n{"+"*55}\nPENDAFTAR LOMBA RANKING #1")
1389         menampilkan_data_pendaftar(pendaftar_ranking1)
1390         lomba = "Ranking 1"
1391     elif pilih_lomba == 3:
1392         print("\n{"+"*55}\nPENDAFTAR LOMBA ROBOTIK")

```

```
1394 |             menampilkan_data_pendaftar_tim(pendaftar_robotik)
1395 |             lomba = "Robotik"
1396 |         elif pilih_lomba == 4:
1397 |             print(f"\n{"*55}\nPENDAFTAR LOMBA UI/UX")
1398 |             menampilkan_data_pendaftar_tim(pendaftar_uiux)
1399 |             lomba = "UI/UX"
1400 |         elif pilih_lomba == 5:
1401 |             print(f"\n{"*55}\nPENDAFTAR LOMBA MOBILE LEGENDS")
1402 |             menampilkan_data_pendaftar_tim(pendaftar_mobilelegends)
1403 |             lomba = "Mobile Legends"
1404 |         elif pilih_lomba == 6:
1405 |             print(f"\n{"*55}\nPENDAFTAR LOMBA VALORANT")
1406 |             menampilkan_data_pendaftar_tim(pendaftar_valorant)
1407 |             lomba = "Valorant"
1408 |         else:
1409 |             print("Pilihan Lomba Tidak Valid")
1410 |             username = input("Masukkan username peserta yang ingin dihapus: ")
1411 |             if lomba in perlombaan:
1412 |                 if username in perlombaan[lomba]:
1413 |                     del perlombaan[lomba][username]
1414 |                     print(f"Data pendaftar '{username}' di lomba '{lomba}' telah dihapus.")
1415 |                 else:
1416 |                     print(f"Tidak ditemukan pendaftar dengan username '{username}' di lomba '{lomba}'")
1417 |                 else:
1418 |                     print(f"\nPenulisan '{lomba}' tidak sesuai.")
1419 | # Kembali ke menu sebelumnya
1420 |         elif pilihan == 5:
1421 |             hitung += 4
```

```
1422 |         break
1423 |     # Eror handling ketika memasukkan pilihan yang tidak sesuai
1424 |     else:
1425 |         print("\nPILIHAN TIDAK VALID")
1426 |     else:
1427 |         print("\nusername/password salah")
1428 |     # Eror handling try except
1429 |     except ValueError:
1430 |         print("\n!!!Maaf anda salah memasukan tipe data!!!")
1431 |     except SyntaxError:
1432 |         print("\n!!!Maaf ada kesalahan sehingga terjadi error!!!")
1433 |     except IndexError:
1434 |         print("\n!!!Maaf ada kesalahan sehingga terjadi error!!!")
1435 |     except AssertionError:
1436 |         print("\n!!!Maaf ada kesalahan sehingga terjadi error!!!")
1437 |     except AttributeError:
1438 |         print("\n!!!Maaf ada kesalahan sehingga terjadi error!!!")
1439 |     except ImportError:
1440 |         print("\n!!!Maaf ada kesalahan sehingga terjadi error!!!")
1441 |     except KeyError:
1442 |         print("\n!!!Maaf ada kesalahan sehingga terjadi error!!!")
1443 |     except NameError:
1444 |         print("\n!!!Maaf ada kesalahan sehingga terjadi error!!!")
1445 |     except MemoryError:
1446 |         print("\n!!!Maaf ada kesalahan sehingga terjadi error!!!")
1447 |     except TypeError:
1448 |         print("\n!!!Maaf ada kesalahan sehingga terjadi error!!!")
1449 |     except IndentationError:
```

```

1450     |     print("\n!!!Maaf ada kesalahan sehingga terjadi error!!!")
1451     | except FileNotFoundError:
1452     |     print("\n!!!Maaf ada kesalahan sehingga terjadi error!!!")
1453
1454 # Fungsi utama/ program utama
1455 def menu_utama():
1456     try:
1457         # Menampilkan menu utama
1458         print("""
1459             """
1460             |     Menu
1461             =====
1462             |     1.Buat Akun User |
1463             |     2.Masuk
1464             =====
1465             |     3.Login Admin |
1466             =====
1467             |     0.Keluar
1468             =====
1469             """
1470         pilihan = int(input("\nMasukkan Pilihan Anda (1/2/3/0): "))
1471         # Pilihan membuat akun user
1472         if pilihan == 1:
1473             |     registrasi()
1474
1475         # Login sebagai user
1476         elif pilihan == 2:
1477             |     login_user()
1478
1479         # Login sebagai admin
1480         elif pilihan == 3:
1481             |     login_admin()
1482
1483         # Keluar dari program/ memberhentikan program
1484         elif pilihan == 0:
1485             |     print(f"""
1486             > {credit[0]}
1487             |     NIM: {credit[1]}
1488             > {credit[2]}
1489             |     NIM: {credit[3]}
1490             > {credit[4]}
1491             |     NIM: {credit[5]}""")
1492             |     exit()
1493
1494         # Eror handling ketika memasukkan pilihan yang tidak sesuai
1495         else:
1496             |     print("\n\nINPUTAN TIDAK VALID")
1497
1498         # Eror handling try except
1499         except ValueError:
1500             |     print("\n!!!Maaf anda salah memasukan tipe data!!!")
1501         except SyntaxError:
1502             |     print("\n!!!Maaf ada kesalahan sehingga terjadi error!!!")
1503         except IndexError:
1504             |     print("\n!!!Maaf ada kesalahan sehingga terjadi error!!!")
1505         except AssertionErro:
1506             |     print("\n!!!Maaf ada kesalahan sehingga terjadi error!!!")
1507
1508         # Eror handling try except
1509         except AttributeError:
1510             |     print("\n!!!Maaf ada kesalahan sehingga terjadi error!!!")
1511         except ImportError:
1512             |     print("\n!!!Maaf ada kesalahan sehingga terjadi error!!!")
1513         except KeyError:
1514             |     print("\n!!!Maaf ada kesalahan sehingga terjadi error!!!")
1515         except NameError:
1516             |     print("\n!!!Maaf ada kesalahan sehingga terjadi error!!!")
1517         except MemoryError:
1518             |     print("\n!!!Maaf ada kesalahan sehingga terjadi error!!!")
1519         except TypeError:
1520             |     print("\n!!!Maaf ada kesalahan sehingga terjadi error!!!")
1521         except IndentationError:
1522             |     print("\n!!!Maaf ada kesalahan sehingga terjadi error!!!")
1523         except FileNotFoundError:
1524             |     print("\n!!!Maaf ada kesalahan sehingga terjadi error!!!")
1525
1526         # melakukan perulangan menggunakan fungsi rekursif
1527         menu_utama()
1528
1529     # Memanggil/ memunculkan fungsi menu utama
1530     menu_utama()

```

BAB IV

PENUTUP

4.1 Kesimpulan

Kesimpulan dari program Manajemen Pendaftaran Lomba MIT-WEEK 2024 yang kami buat memungkinkan pengguna untuk :

- 1. Membuat akun user:** Pengguna dapat membuat akun dengan syarat tertentu (username dan password minimal 4 karakter).
- 2. Login sebagai user atau admin:**
 - **User** dapat mendaftar ke lomba berdasarkan kategori yang tersedia.
 - **Admin** dapat mengelola data lomba, termasuk melihat, mengubah, dan menghapus data pendaftar.
- 3. Pengelolaan lomba:** Program mendukung berbagai kategori lomba untuk individu dan tim, seperti Network, Ranking 1, Robotik, UI/UX, Mobile Legends, dan Valorant.
- 4. Fleksibilitas input data:** Program menangani validasi input data seperti panjang nama, format NIM, serta informasi prodi dan instansi.
- 5. Fitur lengkap:** Meliputi pendaftaran user, pengelolaan lomba, hingga akses admin untuk memodifikasi data.
- 6. Error handling yang komprehensif:** Program menangani berbagai potensi error, sehingga lebih stabil.

4.2 Saran

Saran untuk pengembangan selanjutnya adalah :

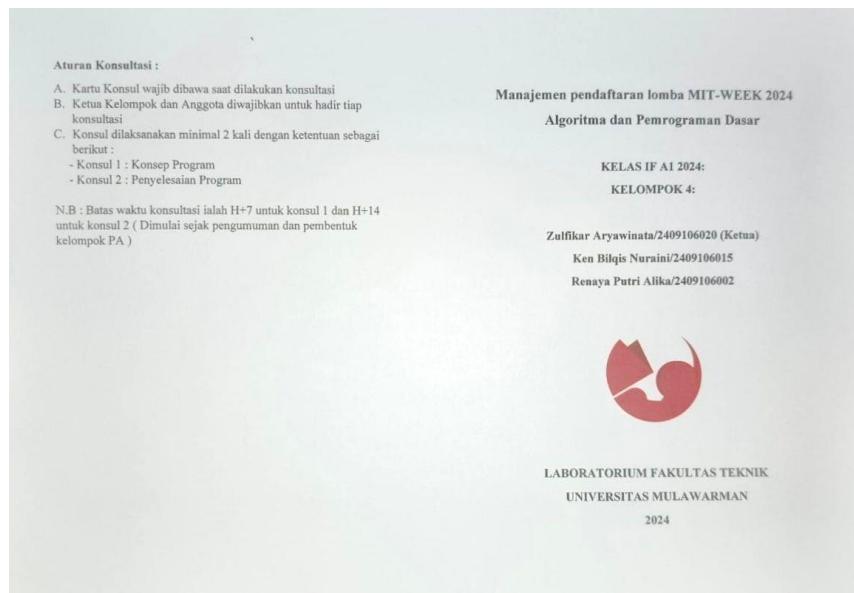
1. **Tambahkan penyimpanan permanen:** Gunakan database seperti SQLite atau file JSON untuk menyimpan data pendaftar dan akun secara permanen
2. **Tambahkan fitur laporan:** Sediakan opsi bagi admin untuk menghasilkan laporan pendaftar dalam bentuk file (contoh: PDF atau Excel).
3. **Berikan batasan waktu pendaftaran:** Implementasikan fitur untuk mengatur batas waktu pendaftaran berdasarkan kategori lomba.
4. **Optimalkan untuk skala besar:** Jika data pendaftar meningkat, pertimbangkan untuk mengintegrasikan sistem dengan database berbasis server.

Dengan saran perbaikan tersebut, program dapat menjadi lebih fungsional, fleksibel, dan siap digunakan untuk berbagai kebutuhan skala besar.

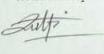
DAFTAR PUSTAKA

- Python Software Foundation. (2024). *Built-in Exceptions*. Penanganan error menjadi elemen penting dalam program ini untuk memastikan program tetap berjalan meskipun terjadi input yang salah atau kesalahan lainnya. Dokumentasi ini memberikan penjelasan tentang berbagai jenis exception seperti ValueError, KeyError, dan TypeError.
- Python Software Foundation. (2024). *Mapping Types — dict*. Struktur data dictionary digunakan untuk menyimpan data peserta lomba, seperti username dan detail pendaftaran. Dokumentasi ini menjelaskan operasi seperti items(), del, dan validasi data di dalam dictionary.
- GeeksforGeeks. (2024). *Error and Exception Handling in Python*. Artikel ini membahas strategi penanganan error di Python, seperti penggunaan blok try-except yang banyak diterapkan dalam program ini untuk menangani kesalahan input atau system.

LAMPIRAN



Cover Kartu Konsul

Tanggal Konsultasi : 31 Oktober 2024	
Uraian / Pembahasan :	
<p>Pada Konsultasi 1 ini, terdapat kesimpulan:</p> <ul style="list-style-type: none">• Flawchart global (opsional).• Tuple: Digunakan untuk mencapai kredit akhir.• Menampilkan pendaftaran berdasarkan perlobatan.• Apakah pendaftaran user dibatasi?• Kuota peserta lomba	
Asisten Lab  Nama: Ahmad Nur Fauzan	Ketua Kelompok  Nama: Zulfikar Aryawinata
Tanggal Konsultasi : 14 November 2024	
Uraian / Pembahasan :	
<p>Pada konsultasi 2, terdapat pembahasan:</p> <ul style="list-style-type: none">• Alur user tidak boleh angka/ terdiri dari huruf saja.• Pada menu ubah dan hapus data muncul tampilan juga daftar data pendaftar.• Error handling alur membuat user kembali mengisi inputan yang salah, bukan ke menu utama.• Membuat error handling agar nimna pendaftar bukan angka.• Pilih menu menggunakan angka.	
Asisten Lab  Nama: Ahmad Nur Fauzan	Ketua Kelompok  Nama: Zulfikar Aryawinata

Isi Kartu Konsul K4