

**LAPORAN PROYEK AKHIR PRAKTIKUM
MATA KULIAH ALGORITMA DAN PEMROGRAMAN DASAR**



MANAJEMEN KONTAK INFORMASI

Oleh:

Kelompok 05

Ahnaf Aliyyu	2409106035
Dwi Prasetyawan	2409106028
Nabila Putri Karni	2409106041

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

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Allah SWT, karena atas rahmat dan karunia-Nya kami dapat menyelesaikan laporan Program Manajemen Kontak Informasi ini. Laporan ini disusun untuk memenuhi tugas mata kuliah Pemrograman dan bertujuan untuk menjelaskan program yang telah kami buat.

Program ini dibuat untuk membantu pengguna dalam mengelola data kontak secara sederhana. Fitur utama dari program ini adalah pendaftaran akun, login, pengelolaan kontak (tambah, lihat, ubah, hapus), dan sistem feedback. Program ini dirancang menggunakan bahasa Python dan berjalan di antarmuka berbasis teks.

Selama pembuatan program dan laporan ini, kami menghadapi beberapa kendala, seperti kesalahan saat debugging dan pengorganisasian data. Namun, berkat bantuan dari asisten, dan teman-teman, kami dapat menyelesaikan tugas ini dengan baik. Kami mengucapkan terima kasih atas semua dukungan dan bantuan yang telah diberikan.

Kami berharap laporan ini dapat bermanfaat bagi pembaca. Kami juga menyadari masih ada kekurangan, sehingga kami terbuka untuk kritik dan saran agar dapat menjadi lebih baik di masa depan.

Samarinda, 21 November 2024

Kelompok 05

TAKARIR

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

Database: Basis
Eksternal: Luar
Else: Lainnya
Email: Pesan
Error: Kesalahan
Feedback: Umpan
File: Berkas
Handling: Penanganan
If: Jika
Indeks: Posisi
Input: Masukan
Integrasi: Penggabungan
Log out: Keluar
Mail: Surat
Manajemen: Mengatur
Password: Sandi
Protocol: Aturan
Username: Identitas
Validasi: Verifikasi

DAFTAR ISI

KATA PENGANTAR.....	ii
TAKARIR	iii
DAFTAR ISI.....	iv
DAFTAR GAMBAR.....	v vi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Kebutuhan Fungsional	1
1.3 Rumusan Masalah.....	3
1.4 Batasan Masalah	4
1.5 Tujuan	4
BAB II PERANCANGAN.....	5
2.1 Analisis Program.....	5
2.2 Flowchart	7
2.3 Konsep/Materi Praktikum yang dipakai.....	10 11
BAB III HASIL DAN PEMBAHASAN	12 13
3.1 Tampilan Program.....	12 13
1. Tampilan Menu	12 13
2. Menu Admin	13 14
3. Menu User.....	14 15
3.2 Source Code	15 17
BAB IV PENUTUP	30 31
4.1 Kesimpulan.....	30 31
4.2 Saran	30 32
DAFTAR PUSTAKA	31 32
LAMPIRAN	32 33

DAFTAR GAMBAR

Gambar 2.2.1 Tampilan Flowchart Menu Utama 7

Gambar 2.2.2 Tampilan Menu Admin..... 8

Gambar 2.2.3 Tampilan Menu User..... 9

Gambar 3.1.1 Tampilan Awal Program..... 12

Gambar 3.1.2 Tampilan Menu Admin..... 13

Gambar 3.1.2 Tampilan Menu User..... ~~13~~14

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era digital saat ini, informasi menjadi salah satu aset paling berharga, termasuk data kontak. Baik individu maupun organisasi membutuhkan sistem yang efisien untuk menyimpan, mengelola, dan mengakses data kontak secara cepat dan aman. Permasalahan yang sering muncul adalah sulitnya mengorganisasi data kontak yang terus bertambah, potensi kehilangan data akibat penyimpanan manual, serta kurangnya integrasi untuk berbagi informasi penting seperti layanan darurat.

Program manajemen data kontak ini dibuat untuk mengatasi permasalahan tersebut dengan menyediakan solusi digital yang sederhana namun fungsional, manajemen data kontak juga dapat digunakan untuk berbagai bidang pekerjaan, bisa di gunakan di kantor sebagai penyimpanan data karyawan, tetapi pada saat ini program yang kami buat adalah berfokus kepada buku kontak yang berbentuk digital, hal ini mempermudah user untuk memperbarui dan menyimpan data seseorang. Apabila dikembangkan dalam satu platform, maka program ini dapat menjadi solusi untuk mencegah kehilangan isi kontak atau hubungan terhadap orang orang yang ingin di simpan.

1.2 Kebutuhan Fungsional

Kebutuhan fungsional adalah semua kebutuhan atau semua bentuk dari fitur-fitur yang harus bisa dijalankan untuk mencapai kesesuaian program dengan tujuan awalnya. Adapun fitur-fitur yang ada pada program kami adalah:

1. Pengenalan pengguna

Sistem kami menggunakan sistem login yang mampu memisahkan antara admin dan user biasa, dimana hal ini adalah bentuk validasi program menggunakan username dan password.

2. Pendaftaran pengguna

Sistem dapat mengenali apabila terdapat hal baru atau seseorang yang mencoba masuk tanpa menggunakan password atau username yang tersedia, maka akan di arahkan untuk masuk ke menu pendaftaran untuk

menciptakan validasi dari pada username dan password baru yang telah di buat.

3. Pengelolaan data kontak oleh pengguna

Tambah Kontak:

- Pengguna dapat menambahkan kontak baru dengan informasi seperti nama, nomor telepon, email, domisili, dan pekerjaan

Lihat Kontak:

- Pengguna dapat melihat daftar kontak yang telah mereka tambahkan.
- Sistem juga harus menampilkan daftar kontak umum yang disediakan oleh admin.

Ubah Kontak:

- Pengguna dapat memperbarui informasi pada kontak tertentu yang mereka miliki.

Hapus Kontak:

- Pengguna dapat menghapus kontak yang tidak lagi relevan dari daftar mereka.

4. Pegelolaan kontak khusus dan publik oleh admin

Tambah Kontak Publik:

- Admin dapat menambahkan kontak publik (misalnya layanan darurat) yang dapat diakses oleh semua pengguna.

Lihat Kontak Publik:

- Admin dapat melihat daftar semua kontak publik.

Ubah Kontak Publik:

- Admin dapat memperbarui informasi kontak publik yang telah disimpan.

Hapus Kontak Publik:

- Admin dapat menghapus kontak publik yang tidak lagi relevan.

5. Feedback

fitur feedback dijalankan dengan menggunakan library SMTP (Simple Mail Transfer Protocol) yang berfungsi sebagai tempat untuk mengirim e-mail dan mengarahkan e-mail ke alamat tujuan

SMTP sendiri memerlukan server, akun, serta email tujuan agar bisa beroperasi, maka dari itu kami menggunakan server gmail sebagai server tempat mengirim email, menggunakan akun google dengan pengaturan khusus agar bisa berinteraksi dengan program, serta memakai email salah satu dari kami sebagai tujuan email terkirim

Cara kerja nya cukup pilih fitur Feedback pada menu user (terdapat pada fitur ke-5) lalu program akan memproses untuk menuju ke bagian feedback, dan disitu pengguna dapat mengirimkan umpan balik yang dapat menjadi kritik dan saran mengenai program yang telah kami buat

Setelah feedback terkirim, admin dapat melihat apa saja feedback yang telah diberikan dari user pada menu admin, dan akan tertera seperti nama user serta isi feedback nya.

6. Penyimpanan data kontak

Penggunaan file eksternal seperti Json adalah salah satu bentuk kemudahan dalam mencadangkan file, mengupdate data dan meneteokan sesi sebelumnya agar lebih mudah lagi di akses.

1.3 Rumusan Masalah

Sering kali dalam penyimpanan kontak seseorang secara manual kita sering melupakan dimana menaruhnya atau bahkan saat kita salah mengisi data dari kontak tersebut maka akan sulit untuk menghapusnya, hal ini menjadi masalah clasic dalam penggunaan buku kontak, dan dalam penyimpanan kontak pada telepon kita sering kali hanya sebatas nama dan nomor, namun dalam program kali ini kami telah merangkum semua massalah itu dengan menciptakan suatu program ynag mampu dengan mudah menulis, membaca, mengupdate dan

bahkan untuk menghapus kontak tersebut, kami telah menciptakan buku kontak secara digital yang mudah untuk dipergunakan.

1.4 Batasan Masalah

Batasan dari program kami hanyalah sebagai penyedia data kosong yang akan ditulis sebagai bentuk penyimpanan kontak, keterbatasan program dalam mengelola user dan admin yang hanya memedakan user biasa dan admin dengan validasi input berbeda untuk setiap rolenya.

1.5 Tujuan

Tujuan awal dalam penciptaan atau pemetaan program dengan judul "sistem manajemen data kontak" pada awalnya adalah hanya sebatas membantu dalam penyimpanan dan memudahkan dalam mengelola data yang di anggap penting hingga perlu untuk di simpan. Namun, kami melihat bahwa program kami memiliki potensi dalam fungsi dasar untuk pengawasan terutama dalam ranah pekerjaan misalnya seperti: "database pekerja harian lepas pada perusahaan X".

BAB II

PERANCANGAN

2.1 Analisis Program

Program manajemen kontak informasi adalah aplikasi berbasis teks yang dirancang untuk membantu pengguna menyimpan, mengelola, dan mengakses data kontak pribadi maupun publik dengan mudah. Program ini memiliki fitur autentikasi, pengelolaan kontak (tambah, lihat, ubah, hapus), serta sistem feedback untuk menerima masukan dari pengguna.

1. Menu Utama

Pengguna diberikan tiga pilihan:

- Login untuk masuk ke akun yang sudah ada.
- Buat Akun untuk mendaftar sebagai pengguna baru.
- Keluar Program untuk menutup aplikasi.

2. Login

Jika berhasil login:

- Admin diarahkan ke menu khusus admin.
- Pengguna Umum diarahkan ke menu kontak pribadi.

Jika gagal login, pengguna dapat mencoba lagi atau kembali ke menu utama.

3. Menu Pengguna Umum

Setelah login sebagai pengguna, opsi yang tersedia adalah:

- Tambah Kontak: Menambahkan kontak baru ke daftar pribadi.
- Lihat Kontak: Melihat kontak pribadi dan kontak publik.
- Ubah Kontak: Memperbarui data kontak pribadi.
- Hapus Kontak: Menghapus kontak pribadi.
- Log Out: Kembali ke menu utama.

4. Menu Admin

Setelah login sebagai admin, opsi yang tersedia adalah:

- Tambahkan Kontak Khusus: Menambahkan kontak yang dapat diakses semua pengguna.
- Lihat Semua Daftar Kontak: Melihat semua kontak khusus dan kontak dari pengguna.
- Ubah Kontak Khusus: Memperbarui kontak khusus.
- Log Out: Kembali ke menu utama.

5. Sistem Feedback

Pengguna dapat memberikan masukan tentang program.

Masukan disimpan dalam file `feedback.csv`, dan admin dapat melihat seluruh feedback.

6. Penyimpanan Data

Data kontak pengguna dan kontak publik disimpan dalam file `dataContacts.csv`.

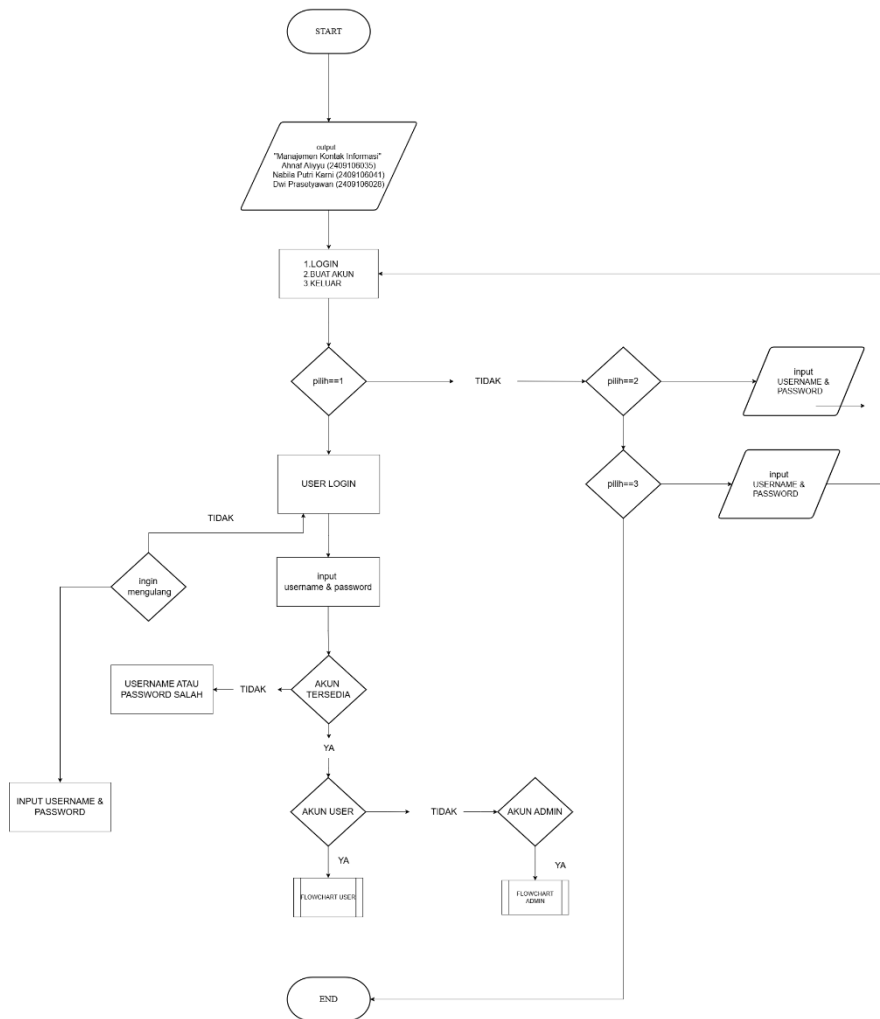
Setiap perubahan data secara otomatis diperbarui dalam file tersebut.

7. Keluar Program

Program akan menutup aplikasi setelah pengguna memilih opsi keluar di menu utama

2.2 Flowchart

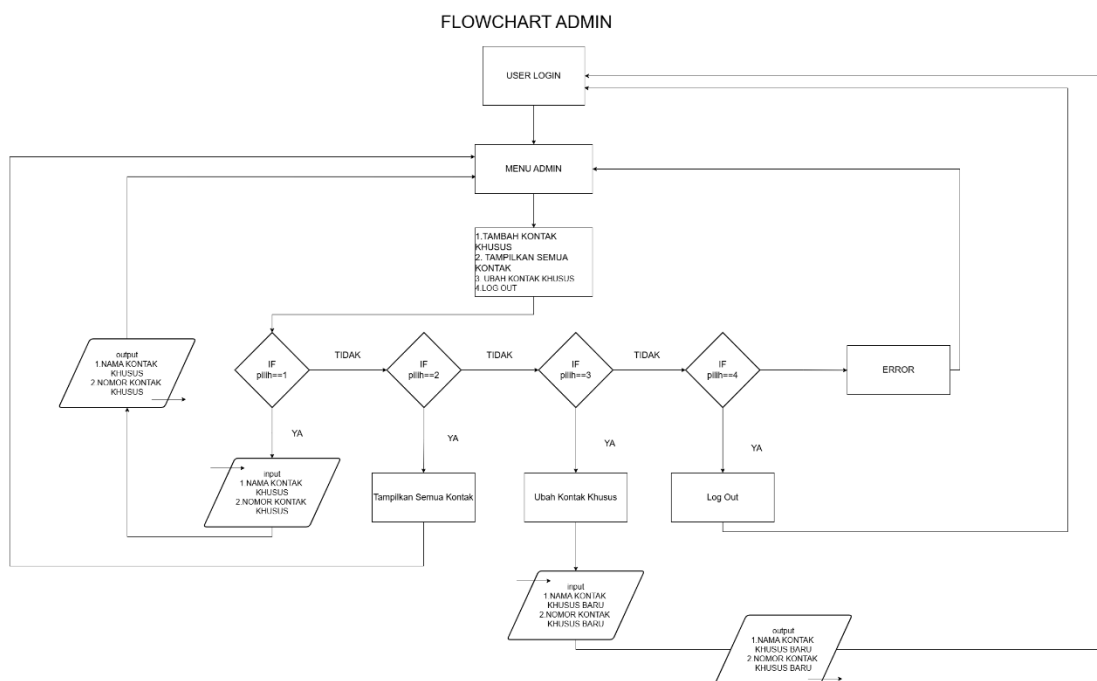
Flowchart Login



Gambar 2.2.1 Tampilan Flowchart Menu Utama

Pertama, akan ditampilkan “Manajemen Kontak Informasi” dan nama anggota kelompok setelah itu user akan dihadapkan dengan menu utama yang terdiri atas 3 opsi, yakni “Login”, “Buat Akun”, dan “Keluar”. Jika user memasukkan pilihan 1, user diminta untuk memasukkan username dan password. Jika username = “admin” dan password = “admin”, maka sistem akan memasukkan username dan password

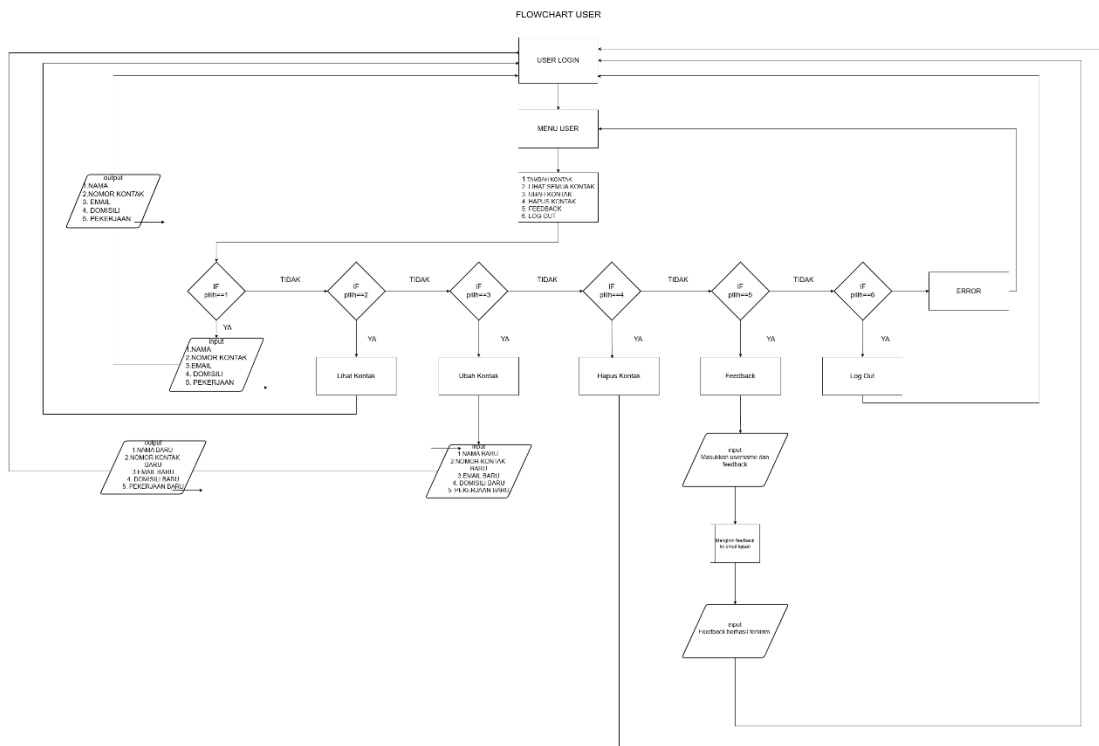
milik admin yang berada di dalam file eksternal data user json. maka user akan masuk sebagai admin dan memiliki akses ke menu admin. Ketika memilih pilihan 2, maka user akan diminta untuk memasukkan username dan password. Namun, apabila username telah terdaftar, maka user akan diminta untuk mengisi ulang username dan password. Namun, apabila username dan password bukan “admin”, maka user akan masuk sebagai pengguna. Selain dari itu, ketika username atau password yang dimasukkan tidak terdaftar, maka user diminta untuk mengisi ulang username dan password. Jika user memilih opsi 3, maka ia akan dikeluarkan dari program.



Gambar 2.2.2 Tampilan Menu Admin.

Jika user memiliki akses sebagai admin, maka ia akan diarahkan menuju menu admin. Setelah itu, user diminta untuk memasukkan pilihan diantara angka 1 hingga

4. Jika admin memasukkan angka 1, maka program akan masuk kepada prosedur untuk menambahkan kontak khusus yang terdiri dari nama dan nomor kontak, setelah yakin kontak tersebut valid maka sistem akan memasukkan ke daftar kontak khusus milik admin yang berada di dalam file eksternal data user json. Jika admin memasukkan angka 2, maka program akan menampilkan semua kontak yang telah dimasukkan oleh user. Jika admin memasukkan angka 3, maka program akan meminta untuk mengubah Kontak khusus lalu meminta admin untuk memasukkan data kontak yang akan diubah. Jika user memasukkan angka 4, maka program akan mengarahkan user kembali ke menu utama. Selain dari angka 1 sampai 4, program akan menampilkan "Perintah tidak diketahui. (pilih antara 1 sampai 4).” dan mengantarkan user kembali ke menu admin.



Gambar 2.2.3 Tampilan Menu User

Ketika user masuk sebagai pengguna, program akan menampilkan menu pengguna dan meminta pengguna untuk memasukkan pilihan di antara angka 1 sampai 6. Jika pengguna memasukkan angka 1, maka program akan masuk kepada prosedur untuk menambahkan kontak yang terdiri dari nama, nomor email, domisili, dan pekerjaan, setelah yakin kontak tersebut valid maka sistem akan memasukkan ke daftar kontak milik user yang berada di dalam file eksternal data user json. Jika pengguna memasukkan angka 2, maka program akan menampilkan seluruh kontak yang telah ditambahkan oleh user dan menampilkan kontak khusus. Jika pengguna memasukkan angka 3, maka program akan meminta untuk mengubah Kontak lalu meminta user untuk memasukkan data kontak yang akan diubah. Jika pengguna memasukkan angka 4, maka program akan menampilkan data kontak user dan meminta pengguna untuk memasukkan indeks data kontak yang ingin dihapus dari favorite. Jika pengguna memasukkan angka 5, program akan menampilkan opsi untuk memberikan feedback atau tanggapan yang akan dikirimkan melalui email kepada salah satu kontributor kami. Jika memasukkan angka 6, maka program akan mengembalikan pengguna ke menu utama. Namun, apabila pengguna memasukkan angka selain dari angka 1, 2, 3, 4, 5 dan 6 maka program akan menampilkan “Perintah tidak diketahui. (pilih antara 1 sampai 6).” dan mengembalikan pengguna ke menu pengguna.

2.3 Konsep/Materi Praktikum yang dipakai

Konsep yang kami gunakan dalam pembuatan laporan ini sesuai dengan persyaratan yang terdapat pada modul diantaranya:

1. Menggunakan percabangan

Menggunakan if dan else sebagai opsi pilihan dalam validasi input jika terjadi input benar serta selain input yang dimaksud

2. Menggunakan perulangan

Menggunakan modul perulangan sebagai bentuk pemecahan masalah atau memutar suatu program agar mendapat hasil yang di capai

3. Menggunakan List/Tuple

Tuple yang digunakan pada program berada pada bagian copyright yang berfungsi untuk menampilkan judul serta kontributor dalam program, pemilihan dimaksudkan agar data tidak berubah.

4. Menggunakan Dictionary

Dictionary tidak serta merta hanya berada di dalam file internal python tetapi juga bisa terdapat pada file eksternal, pada kali ini kami menggunakan json sebagai file eksternal untuk menampilkan dan menampung data admin&user.

5. Menggunakan Fungsi & Prosedur

Menggunakan Fungsi dan prosedur yang dimaksudkan untuk mempermudah dalam menulis code dan bahkan dapat mempersingkat code program.

6. Menggunakan Error Handling

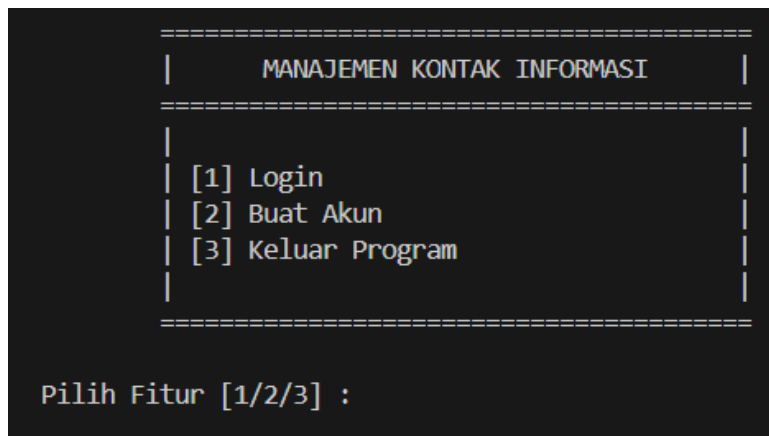
Digunakan sebagai solusi jika ada kesalahan atau berbagai jenis error yang muncul dalam program, yang berfungsi untuk tidak menampilkan error tersebut

BAB III

HASIL DAN PEMBAHASAN

3.1 Tampilan Program

1. Tampilan Menu



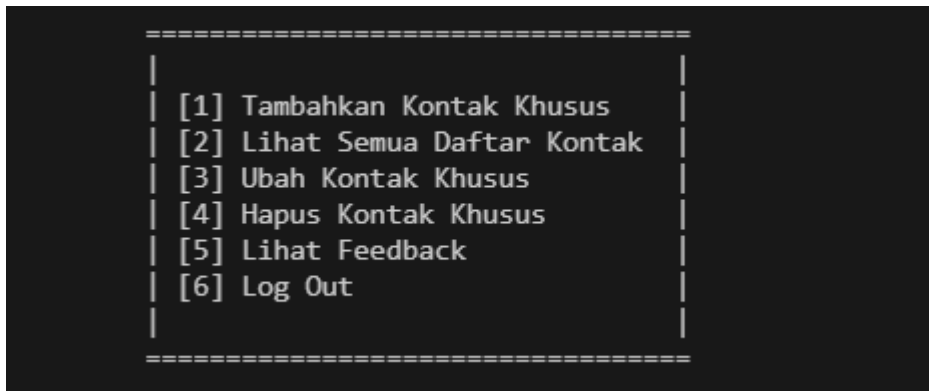
Gambar 3.1.1 Tampilan awal program.

- Login: Mengarahkan pengguna ke proses autentikasi.
- Buat Akun: Memberikan opsi kepada pengguna untuk mendaftar akun baru.
- Keluar Program: Mengakhiri program.

Tampilan ini dirancang agar sederhana dan intuitif, sehingga mudah dipahami oleh pengguna pemula. Pilihan hanya memerlukan input berupa angka (1, 2, atau 3).

Tampilan ini akan diikuti dengan tampilan menu lain sesuai dengan jenis pengguna yang login (admin atau pengguna umum).

2. Menu Admin



Gambar 3.1.2 Tampilan *menu admin*.

Menu ini adalah menu yang muncul setelah admin berhasil masuk ke dalam program. Menu ini berisi fitur-fitur khusus yang dapat dilihat hanya oleh admin untuk mengelola data kontak khusus, melihat daftar kontak pengguna, serta melihat masukan dari pengguna melalui sistem feedback.

Daftar Opsi dalam Menu Admin:

1. Tambahkan Kontak Khusus

Admin dapat menambahkan kontak baru yang bersifat publik. Kontak ini dapat dilihat oleh semua pengguna program.

2. Lihat Semua Daftar Kontak

Menampilkan daftar lengkap semua kontak, baik kontak publik yang dikelola admin maupun kontak pribadi dari setiap pengguna yang terdaftar.

3. Ubah Kontak Khusus

Admin dapat memperbarui data kontak publik yang sudah ada. Fitur ini memungkinkan perubahan nama kontak atau nomor telepon.

4. Hapus Kontak Khusus

Admin dapat menghapus kontak publik tertentu dari daftar, sehingga tidak lagi terlihat oleh pengguna.

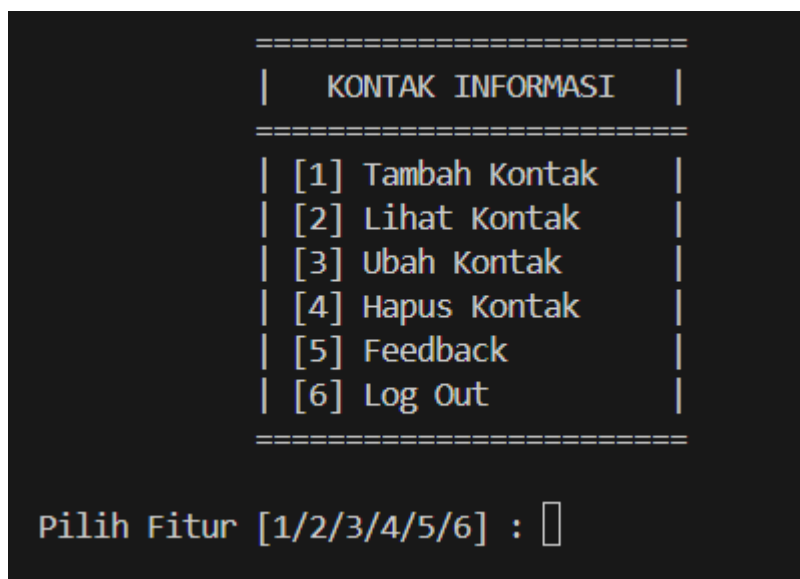
5. Lihat Feedback

Menampilkan feedback yang telah diberikan oleh pengguna melalui sistem feedback. Fitur ini memungkinkan admin memahami pendapat atau saran dari pengguna untuk program.

6. Log Out

Admin keluar dari mode login dan kembali ke menu utama

3. Menu User



Gambar 3.1.1 Tampilan Menu User

Menu ini adalah tampilan yang muncul setelah pengguna berhasil login ke dalam program. Menu ini menyediakan berbagai fitur utama yang dapat digunakan untuk mengelola kontak pribadi serta memberikan feedback terkait pengalaman menggunakan program.

Daftar Fitur dalam Menu User:

1. Tambah Kontak

Fitur ini digunakan untuk menambahkan kontak baru ke dalam daftar kontak pribadi pengguna. Pengguna akan diminta mengisi informasi seperti nama, nomor telepon, email (opsional), dan akun sosial media (opsional).

2. Lihat Kontak

Fitur ini menampilkan semua kontak yang telah ditambahkan oleh pengguna. Selain itu, juga akan menampilkan daftar kontak publik yang dikelola oleh admin.

3. Ubah Kontak

Dengan fitur ini, pengguna dapat memperbarui informasi kontak yang sudah ada, seperti nama, nomor telepon, email, atau akun sosial media.

4. Hapus Kontak

Fitur ini memberikan opsi kepada pengguna untuk menghapus kontak pribadi yang tidak lagi diperlukan.

5. Feedback

Fitur ini memungkinkan pengguna memberikan masukan atau saran kepada pengembang terkait pengalaman menggunakan program.

6. Log Out

Fitur ini digunakan untuk keluar dari akun pengguna dan kembali ke menu utama.

3.2 Source Code

```
import json
import smtplib
import os
```

```

start = True # Untuk Pengulangan saat login berhasil

# Deklarasi Tempat json
json_s = str(os.path.dirname(os.path.abspath(__file__))) + "\\Contacts.json"
json_feedback = str(os.path.dirname(os.path.abspath(__file__))) + "\\feedback.json"

# Membaca Contacts.json
def load_contacts():
    with open(json_s, 'r') as Contacts:
        return json.load(Contacts)
data = load_contacts()

# Simpan Perubahan kedalam json
def save_contacts(data):
    with open(json_s, 'w') as dataContacts:
        json.dump(data, dataContacts, indent=4)

def refresh():
    if os.name == 'nt':
        _ = os.system('cls')
    else:
        _ = os.system('clear')

# Fitur Umpan Balik yang bisa langsung masuk kedalam email
def feedback(feedback_message,sender):
    smtp_server = "smtp.gmail.com"
    port = 587 # For starttls
    sender_email = "ebot3412@gmail.com"
    password = "fdlvihxcutptrrd1"
    receiver_email = "chrogam@gmail.com"
    try:
        server = smtplib.SMTP(smtp_server,port)
        server.starttls()
        server.login(sender_email, password)
        message = f"""
Umpan balik baru dari {sender}

Pesan : {feedback_message}
"""

```

```

        server.sendmail(sender_email, receiver_email, message)
        server.quit()
    except Exception as e:
        print(e)

# Membaca feedback.json
def load_feedback():
    try:
        with open(json_feedback, 'r') as feedback_file:
            return json.load(feedback_file)
    except FileNotFoundError:
        return {"Feedback": []}
fb = load_feedback()

# Simpan Perubahan ke dalam feedback.json
def save_feedback (feedback_data):
    with open(json_feedback, 'w') as feedback_file:
        json.dump(feedback_data, feedback_file, indent=4)

# Fitur Umpan Balik yang menyimpan ke JSON
def feed_back():
    print("===== FEEDBACK =====")
    feedback_message = input("Berikan Feedback Anda : ")
    if feedback_message != None:
        for kontak in data['User']:
            if pengguna_aktif == kontak['username']:
                sender = kontak['name']
                # Tambahkan umpan balik ke data JSON
                feedback(feedback_message, sender)
                fb["Feedback"].append({"username": pengguna_aktif, "name": sender, "pesan":
feedback_message})
                save_feedback(fb)
                print("Feedback Anda tersimpan. Terima kasih!")
    else:
        print("Pesan kosong, feedback tidak tersimpan.")

# Fungsi untuk Admin melihat feedback yang terkirim pada program
def check_feedback():
    print("-----|| Daftar Feedback ||-----")

```

```

    for num, feed in enumerate(fb['Feedback']):
        print(f"\nFeedback ke-{num + 1}\nUsername : {feed['username']}\nNama : {feed['name']}\nPesan : {feed['pesan']}")

# Tuple yang muncul saat berada di menu awal
def copyright():
    copyright = ("Manajemen Kontak Informasi", "Ahnaf Aliyyu (2409106035)", "Nabila Putri Karni (2409106041)", "Dwi Prasetyawan (2409106028)")
    print("="*20)
    print(f"{copyright[0]}\n{copyright[1]}\n{copyright[2]}\n{copyright[3]}")
    print("="*20)

# Menambahkan Kontak Pengguna
def add_user_contact():
    while True:
        nama = input("Masukkan nama kontak : ")
        if nama.strip() == "":
            print("Nama tidak boleh kosong")
        else:
            break
    while True:
        nomor = input("Masukkan nomor kontak : ")
        if not nomor.isnumeric():
            print("nomor tidak terdiri dari angka, silahkan ulangi lagi")
        else:
            break
    while True:
        e_mail = input("Masukkan E-Mail (Tekan Enter jika tidak ada) : ").strip()
        if e_mail == "":
            e_mail = "N/A"
            break
        elif "@" not in e_mail:
            print("E-mail tidak lengkap")
        else:
            break
    domisili = input("Masukkan Domisili (Tekan Enter jika tidak ada) : ")
    if domisili.strip() == "":
        domisili = "N/A"
    pekerjaan = input("Masukkan Pekerjaan (Tekan Enter jika tidak ada) : ")
    if pekerjaan.strip() == "":

```

```

        pekerjaan = "N/A"
    for kontak in data['User']:
        if kontak['username'] == pengguna_aktif:
            kontak['kontak'].append({"nama": nama, "nomor": nomor, "email": e_mail,
"domisili": domisili, "pekerjaan": pekerjaan})
            save_contacts(data)
    print("Kontak Berhasil ditambahkan")

# Melihat Daftar Kontak Pengguna
def check_user_contact():
    for kontak in data['User']:
        if pengguna_aktif == kontak['username']:
            for num, contact in enumerate(kontak['kontak']):
                print(f"Kontak ke-{num + 1}\nNama : {contact['nama']}\nNomor :
{contact['nomor']}\nE-mail : {contact['email']}\nDomisili :
{contact['domisili']}\nPekerjaan : {contact['pekerjaan']}\n")
            print("Kontak Lainnya")
    for kontak in data['Admin']:
        for num, contact in enumerate(kontak['kontak_K']):
            print(f>Nama : {contact['nama']}\nNomor : {contact['nomor']}\n")

# Proses Mengubah Data Kontak
def updating_user_contact(pengguna_aktif, pilihan_kontak):
    while True:
        nama_baru = input("Masukkan nama kontak : ")
        if nama_baru.strip() == "":
            print("Nama tidak boleh kosong")
        else:
            break
    while True:
        nomor_baru = input("Nomor : ").strip()
        if not nomor_baru.isnumeric():
            print("nomor baru tidak terdiri dari angka, silahkan ulangi lagi")
        else:
            break
    while True:
        email_baru = input("E-mail : ")
        if email_baru.strip() == "":
            email_baru = "N/A"
            break

```



```

        elif "@" not in email_baru:
            print("E-mail tidak lengkap")
        else:
            break
    dom_baru = input("Domisili : ")
    if dom_baru.strip() == "":
        dom_baru = "N/A"
    job_baru = input("Pekerjaan : ")
    if job_baru.strip() == "":
        job_baru = "N/A"
    for kontak in data['User']:
        if kontak['username'] == pengguna_aktif:
            up = kontak['kontak'][pilihan_kontak]
            up['nama'] = nama_baru
            up['nomor'] = nomor_baru
            up['email'] = email_baru
            up['domisili'] = dom_baru
            up['pekerjaan'] = job_baru

# Fungsi Mengubah Data Kontak Pengguna Tertentu
def update_user_contact():
    show_user_contact()
    try:
        pilihan_kontak = int(input("Pilih kontak yang ingin diubah (hanya angka) : "))
- 1
        for kontak in data['User']:
            if pengguna_aktif == kontak["username"]:
                if 0 <= pilihan_kontak < len(kontak['kontak']):
                    updating_user_contact(pengguna_aktif, pilihan_kontak)
                    save_contacts(data)
                    print("Kontak berhasil diubah")
                else:
                    print("Kontak tidak ditemukan")
                    break
    except ValueError:
        print("Input tidak valid, silakan masukkan angka.")

# Menampilkan Kontak Pengguna
def show_user_contact():
    for kontak in data['User']:

```

```

        if pengguna_aktif == kontak['username']:
            for num, contact in enumerate(kontak['kontak']):
                print(f"Kontak ke-{num + 1}\nNama : {contact['nama']}\nNomor : {contact['nomor']}\nE-mail : {contact['email']}\nDomisili : {contact['domisili']}\nPekerjaan : {contact['pekerjaan']}\n")

# Menghapus Kontak Pengguna
def delete_user_contact():
    show_user_contact()
    try:
        indeks_hapus = int(input("Pilih kontak yang ingin dihapus (hanya angka) : "))
        - 1
        for kontak in data['User']:
            if pengguna_aktif == kontak['username']:
                if 0 <= indeks_hapus < len(kontak['kontak']):
                    kontak_lama = kontak['kontak'].pop(indeks_hapus)
                    save_contacts(data)
                    print(f"Kontak [{kontak_lama['nama']}] Berhasil dihapus")
                else:
                    print("Kontak tidak ditemukan")
                    break
            except ValueError:
                print("Input tidak valid, silakan masukkan angka.")

# Menambah kontak baru sbg admin
def add_admin_contact():
    tried = 0
    while True:
        nama_adm = input("Masukkan nama kontak : ")
        if nama_adm.strip() == "":
            print("Nama tidak boleh kosong")
        else:
            break
    while tried < 3:
        nomor_adm = input("Masukkan nomor kontak : ").strip()
        if not nomor_adm.isnumeric():
            print("Nomor tidak valid")
            tried += 1
        else:
            break

```

```

    if tried == 3:
        return
    for atmin in data['Admin']:
        if any(kontakk['nama'] == nama_adm for kontakk in atmin['kontak_K']):
            print("Nama kontak telah terdaftar, silahkan masukkan kontak lain")
        else:
            if any(kontakk['nomor'] == nomor_adm for kontakk in atmin['kontak_K']):
                print("Nomor kontak telah terdaftar, silahkan masukkan kontak lain")
            else:
                for kontak in data['Admin']:
                    kontak['kontak_K'].append({"nama": nama_adm, "nomor": nomor_adm})
                save_contacts(data)
                print("Kontak admin berhasil ditambahkan.")

# Melihat Semua Daftar Kontak sbg admin
def check_admin_contact():
    for user in data['User']:
        print(f"\nPengguna: {user['name']} ({user['username']})")
        for num, contact in enumerate(user['kontak']):
            print(f"Kontak ke-{num+1}\nNama : {contact['nama']}\nNomor : {contact['nomor']}\nE-mail : {contact['email']}\nDomisili : {contact['domisili']}\nPekerjaan : {contact['pekerjaan']}\n")

    print("\nDaftar Kontak Khusus:")
    for admin in data['Admin']:
        for num, contact in enumerate(admin['kontak_K']):
            print(f>Nama : {contact['nama']}\nNomor : {contact['nomor']}\n")

# Menampilkan Daftar Kontak Admin
def show_admin_contact():
    for kontakk in data['Admin']:
        for num, contact in enumerate(kontakk['kontak_K']):
            print(f"kontak ke-{num+1}\nNama : {contact['nama']}\nNomor : {contact['nomor']}\n")

# Proses Mengubah Kontak Admin
def updating_admin_contact(pilihan_kontak_adm):
    tried = 0
    while True:

```

```

        nama_adm_baru = input("Nama : ")
        if nama_adm_baru.strip() == "":
            print("Nama tidak boleh kosong")
        else:
            break
    while tried < 3:
        nomor_adm_baru = input("Nomor : ").strip()
        if not nomor_adm_baru.isnumeric():
            print("Nomor tidak valid")
            tried += 1
        else:
            break
    if tried == 3:
        return
    for atmin in data['Admin']:
        if any(kontakk['nama'] == nama_adm_baru for kontakk in atmin['kontak_K']):
            print("Nama kontak telah terdaftar, silahkan masukkan kontak lain")
        else:
            if any(kontakk['nomor'] == nomor_adm_baru for kontakk in
atmin['kontak_K']):
                print("Nomor kontak telah terdaftar, silahkan masukkan kontak lain")
            for kontak in data['Admin']:
                up = kontak['kontak_K'][pilihan_kontak_admin]
                up['nama'] = nama_adm_baru
                up['nomor'] = nomor_adm_baru
            save_contacts(data)
            print("Kontak berhasil diubah")

# Mengubah Kontak Admin
def update_admin_contact():
    show_admin_contact()
    try:
        pilihan_kontak_admin = int(input("Pilih kontak yang ingin diubah (hanya angka) :
")) - 1
        for kontakk in data['Admin']:
            if pilihan_kontak_admin >= 0 and pilihan_kontak_admin <=
len(kontakk['kontak_K']):
                updating_admin_contact(pilihan_kontak_admin)
            else:
                print("Kontak tidak ada")

```

```

        except ValueError:
            print("Kontak tidak ditemukan")

def delete_admin_contact():
    show_admin_contact()
    try:
        atmin_hapus = int(input("Pilih kontak yang ingin dihapus (hanya angka) : ")) -
1
        for atmin in data['Admin']:
            if 0 <= atmin_hapus < len(atmin["kontak_K"]):
                kontak_lama = atmin["kontak_K"].pop(atmin_hapus)
                save_contacts(data)
                print(f"Kontak [{kontak_lama['nama']}] Berhasil dihapus")
            else:
                print("Kontak tidak ditemukan")
        except ValueError:
            print("Kontak tidak ditemukan")

# Membuat Akun Pengguna Baru
def create_account(data):
    print("Silahkan isi pertanyaan berikut :")
    while True:
        username = input("Masukkan Username : ").strip()
        if not username.isalnum() or username == ".":
            print("Username hanya boleh menggunakan huruf, angka, serta titik(.)")
        else:
            break
        if any(akun['username'] == username for akun in data['User']):
            print("Username telah terdaftar, silahkan masukkan username lain atau masuk
melalui Login")
            return data
        password = input("Masukkan Password : ").strip()
        Nama = input("Masukkan Nama Kamu :")

    data['User'].append({"username":username,"name":Nama,"password":password,"kontak":[]})
    save_contacts(data)
    print("Akun anda telah dibuat, silahkan pilih opsi login untuk masuk")

# Program yang dijalankan dalam pengulangan sampai pengguna memilih keluar

```

```

while True:
    try:
        # menu utama
        refresh()
        copyright()
        print("""

=====
|      MANAJEMEN KONTAK INFORMASI      |
=====
|                                     |
| [1] Login                           |
| [2] Buat Akun                       |
| [3] Keluar Program                  |
|                                     |
=====

""")
        pilihan = int(input("Pilih Fitur [1/2/3] : "))
        if pilihan == 1:
            refresh()
            i = 0
            # Login
            while i < 1:
                print("""

=====
|      LOGIN      |
=====

""")
                pengguna_aktif = input("Masukkan Username : ")
                kata_sandi = input("Masukkan Password : ")
                akses = 0
                for akun in data['User']:
                    if pengguna_aktif == akun["username"] and akun["password"] ==
kata_sandi:
                        akses = 1
                for admin in data['Admin']:
                    if pengguna_aktif == admin['username'] and admin['password'] ==
kata_sandi:
                        akses = 2
                else:

```

```

        continue
    if akses == 1:
        aktif = None
        for akun in data['User']:
            global nama_akun
            nama_akun = akun['name']
            if akun['username'] == pengguna_aktif:
                aktif = akun
                break
        refresh()
        print("Login Berhasil!")
        start = True
        print(f"Selamat Datang, {nama_akun}!")
        i = 1
        while start == True:
            # Menu Pengguna
            print("""
=====
|   KONTAK INFORMASI   |
=====
| [1] Tambah Kontak    |
| [2] Lihat Kontak     |
| [3] Ubah Kontak      |
| [4] Hapus Kontak     |
| [5] Feedback         |
| [6] Log Out          |
=====
""")

            ans = input("Pilih Fitur [1/2/3/4/5/6] : ")
            match ans:
                case "1":
                    add_user_contact()

                case "2":
                    check_user_contact()

                case "3":
                    update_user_contact()

```

```

        case "4":
            delete_user_contact()

        case "5":
            feed_back()
        case "6":
            i = 1
            break

        case _:
            print("Perintah tidak diketahui (Pilih opsi antara 1
sampai 6)")

    elif akses == 2:
        refresh()
        start = True
        print("Login Berhasil!")
        while start == True:
            # Menu Admin
            print("""
=====
|          KONTAK ADMIN          |
=====
|                                 |
| [1] Tambahkan Kontak Khusus   |
| [2] Lihat Semua Daftar Kontak |
| [3] Ubah Kontak Khusus        |
| [4] Hapus Kontak Khusus       |
| [5] Lihat Feedback            |
| [6] Log Out                   |
|                                 |
=====
""")

            pilihan_admin = input("Pilih Fitur [1/2/3/4/5/6] : ")
            match pilihan_admin:
                case "1":
                    add_admin_contact()

                case "2":

```



```

        check_admin_contact()

    case "3":
        update_admin_contact()

    case "4":
        delete_admin_contact()

    case "5":
        check_feedback()

    case "6":
        # Log out
        print("Keluar dari mode Admin. . .")
        start = False
        pengguna_aktif = None
        i = 1

    case _:
        print("Perintah Tidak Diketahui (Pilih Antara 1 sampai
6)")

    else:
        print("Login Gagal")
        while True:
            ul1 = input("Coba lagi? [Y/N] :").lower()
            if ul1 == "y":
                break
            elif ul1 == "n":
                i = 1
                break
            else:
                print("Perintah tidak ditemukan (Hanya Y/N) ")
        elif pilihan == 2:
            refresh()
            create_account(data)

        elif pilihan == 3:
            # Keluar Program
            break

```

```
        else:  
            print("Fitur tidak ada (pilih antara 1 dan 3)")  
    except ValueError:  
        print("Fitur tidak ada (pilih antara 1 dan 3)")
```

BAB IV

PENUTUP

4.1 Kesimpulan

Program manajemen kontak informasi yang kami rancang bertujuan untuk mengatasi permasalahan dalam penyimpanan dan pengelolaan data kontak. Program ini dirancang untuk mempermudah pengguna dalam menambahkan, mencari, memperbarui, atau menghapus data kontak secara efisien dan terstruktur.

Selain itu, program ini juga membantu dalam meminimalkan risiko kehilangan data dan mempermudah akses informasi penting seperti kontak darurat. Dengan fitur-fitur yang disediakan, program ini mampu meningkatkan efisiensi dan memudahkan pengguna dalam mengelola data kontak sesuai kebutuhan.

4.2 Saran

Meskipun program masih sederhana namun program ini sudah menerapkan beberapa materi yang berkaitan dengan praktikum python, saran nantinya bermanfaat bagi pengembangan system selanjutnya yaitu :

1. Dengan adanya program manajemen kontak informasi ini, diharapkan dapat membantu pengguna dalam mengelola data kontak secara lebih terstruktur dan efisien, baik untuk keperluan pribadi maupun profesional.
2. Diharapkan dengan sistem ini, pengguna dapat meminimalkan waktu yang dibutuhkan untuk memperbarui atau menambah informasi kontak.
3. Diharapkan ke depannya, program ini dapat terus berkembang dengan menambahkan fitur-fitur baru seperti integrasi dengan aplikasi lain, peningkatan keamanan, dan kemampuan untuk menyimpan lebih banyak informasi terkait kontak.

DAFTAR PUSTAKA

Penjelasan dasar tentang JSON, struktur, dan penggunaannya. (n.d.). Retrieved from Introducing JSON: <https://www.json.org/>

Python Documentation - Panduan resmi untuk menggunakan modul JSON di Python:. (n.d.). Retrieved from Python Software Foundation. (n.d.). JSON — JavaScript Object Notation: <https://docs.python.org/3/library/json.html>

Real Python - Tutorial praktis untuk memahami penggunaan JSON dalam aplikasi Python:. (n.d.). Retrieved from Real Python. (n.d.). Working with JSON Data in Python: <https://realpython.com/python-json/>

Sending Emails With Python. (2023, November 17). Retrieved from Real Python: <https://realpython.com/python-send-email/>

W3Schools - Panduan untuk memahami kelebihan dan cara kerja JSON dibandingkan dengan format lainnya:. (n.d.). Retrieved from W3Schools. (n.d.). JSON Tutorial: https://www.w3schools.com/js/js_json_intro.asp

LAMPIRAN

Aturan Konsultasi :

- A. Kartu Konsul wajib dibawa saat dilakukan konsultasi
- B. Ketua Kelompok dan Anggota diwajibkan untuk hadir tiap konsultasi
- C. Konsul dilaksanakan minimal 2 kali dengan ketentuan sebagai berikut :
 - Konsul 1 : Konsep Program
 - Konsul 2 : Penyelesaian Program

N.B : Batas waktu konsultasi ialah H+7 untuk konsul 1 dan H+14 untuk konsul 2 (Dimulai sejak pengumuman dan pembentuk kelompok PA)

Manajemen Kontak Informasi Algoritma dan Pemrograman Dasar

KELAS IF A2 24:

KELOMPOK 5:

Ahnaf Aliyyu / 2409106035



Dwi Prasetyawan / 2409106028



Nabila Putri Karni / 2409106041



LABORATORIUM FAKULTAS
TEKNIK UNIVERSITAS
MULAWARMAN

2024

Tanggal Konsultasi Sabtu, 2 November 2024	
<p>Uraian / Pembahasan</p> <ul style="list-style-type: none"> • masuki sistem feedback • bikin sistem pengumuman • gedein kotak flowchart • Tuple : fitur copyright 	
<p>Asisten Lab</p>  <p>Nama: Ahmed Nur Fauzan</p>	<p>Ketua Kelompok</p>  <p>Nama: Ahnaf Aliyyu</p>

Tanggal Konsultasi Kamis, 14, November, 2024	
<p>Uraian / Pembahasan</p> <ul style="list-style-type: none"> • Flowchart lebih dijelaskan • Cara menjelaskan program lebih dimaksimalkan • Feedback: masukkan nama user • menggunakan study kasus sebagai patokan program • Tambahkan atribut User • Berikan nama saat daftar akun, jgn hanya username • Menambahkan hapus kontak khusus • Error Handling • Fitur admin dapat melihat feedback. 	
<p>Asisten Lab</p>  <p>Nama: Ahmed Nur Fauzan</p>	<p>Ketua Kelompok</p>  <p>Nama: Ahnaf Aliyyu</p>