

Laporan Praktikum
Algoritma dan Pemrograman
(Studi Kasus Sistem Pengelolaan Data Mahasiswa)



Disusun Oleh:

Muhammad Zakki Fitra Ramadhan (250535618409)

Aldiaz Hafiz Afiansyah (250535626275)

PROGRAM STUDI S1 TEKNIK INFORMATIKA
DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI MALANG
2025

KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Allah SWT atas berkatnya kami dapat menyelesaikan tugas yang diberikan oleh Bapak Muhammad Jauharul Fuady, S.T., M.T. sebagai dosen pelajaran Algoritma dan Pemrograman merupakan salah satu faktor pengukur bagi mahasiswa dalam memenuhi mata pelajaran Algoritma dan Pemrograman.

Kami menyadari bahwa Laporan Praktikum ini masih jauh dari sempurna, baik dari teknik penulisan maupun materi mengingat kemampuan yang kami miliki. oleh karena itu kritik dan saran dari semua pihak yang bersifat membangun selalu saya harapkan demi kesempurnaan makalah ini.

Sebelumnya, kami mohon maaf jika terdapat kata-kata yang kurang berkenan. Semoga dengan penyusunan Laporan Praktikum ini memberikan manfaat bagi pembaca sehingga dapat menambah pengetahuan dan pemahaman diri. Akhir kata, kami sampaikan terima kasih kepada semua pihak yang telah berperan serta dalam penyusunan makalah ini. Semoga Allah SWT senantiasa meridhai segala usaha kita. Aamiin.

Malang, 11 Oktober 2025

Penulis

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI	ii
BAB I PENDAHULUAN	1
A. Latar Belakang	1
B. Tujuan.....	1
C. Manfaat.....	2
BAB II PEMBAHASAN	3
A. Deskripsi Tugas.....	3
B. Cuplikan Kode Program	3
C. Penjelasan Program	7
D. Contoh Output Program.....	8
BAB III PENUTUP	12
A. Kesimpulan	12
B. Saran.....	13
DAFTAR PUSTAKA	14

BAB I PENDAHULUAN

A. Latar Belakang

Di sebuah institusi pendidikan, terdapat bagian administrasi akademik yang berfungsi untuk mengelola data-data pokok mahasiswa. Proses pencatatan secara manual dapat berisiko terjadinya *human error*, memakan banyak waktu, mempersulit saat ada pencarian atau pembaruan data. Oleh karena itu, bagian administrasi akademik membutuhkan sebuah sistem yang bisa mengelola data-data mahasiswa dengan efisien dan terstruktur.

Sistem ini dikembangkan menggunakan bahasa pemrograman python dengan menerapkan fungsi-fungsi fundamental, seperti fungsi, file handling, looping, dan percabangan if-else.

Tujuannya adalah untuk menyediakan antarmuka berbasis teks yang intuitif bagi staf administrasi untuk melakukan tiga operasi utama: menambahkan data mahasiswa baru (NIM, Nama, Jurusan), menampilkan seluruh data yang tersimpan dalam format tabel yang rapi, dan menghapus data mahasiswa berdasarkan NIM sebagai pengenalan unik. Program ini dirancang untuk menjadi solusi digital yang ringkas guna menyederhanakan dan meningkatkan akurasi dalam pengelolaan data dasar mahasiswa.

B. Tujuan

1. Memenuhi tugas Algoritma dan Pemrograman
2. Merancang sistem pengolahan data mahasiswa
3. untuk menyediakan antarmuka berbasis teks yang intuitif bagi staf administrasi untuk melakukan tiga operasi utama: menambahkan data mahasiswa baru (NIM, Nama, Jurusan), Melihat data mahasiswa, Menghapus data mahasiswa

C. Manfaat

1. Meningkatkan pemahaman mahasiswa dalam praktik
2. Meningkatkan penyelesaian masalah mahasiswa
3. Meningkatkan berpikir kritis mahasiswa

BAB II PEMBAHASAN

A. Deskripsi Tugas

Program ini dapat melakukan hal-hal berikut:

1. Memungkinkan pengguna untuk **menambahkan data mahasiswa baru** yang meliputi NIM (Nomor Induk Mahasiswa), nama, dan jurusan melalui input interaktif.
2. Menerapkan sistem **validasi pada saat penambahan data**, di mana program akan memeriksa apakah NIM yang dimasukkan terdiri dari 12 angka, hanya berisi digit, dan belum terdaftar sebelumnya untuk memastikan integritas data.
3. **Menampilkan keseluruhan data mahasiswa** yang tersimpan dalam format tabel yang rapi dan terstruktur, mencakup nomor urut, NIM, nama, dan jurusan untuk kemudahan pembacaan.
4. Menyediakan fungsionalitas untuk **menghapus data mahasiswa** berdasarkan NIM yang diinput oleh pengguna. Program akan menampilkan data yang akan dihapus dan meminta konfirmasi terlebih dahulu untuk mencegah kesalahan.
5. Mengelola penyimpanan data secara persisten dalam sebuah **file teks** (**mahasiswa.txt**), sehingga data mahasiswa tidak akan hilang meskipun program telah ditutup dan dijalankan kembali.
6. Menyajikan **antarmuka menu utama berbasis teks** yang interaktif, memungkinkan pengguna untuk dengan mudah memilih operasi yang ingin dilakukan (tambah, tampilkan, hapus, atau keluar dari program).

B. Cuplikan Kode Program

```
#Program Data Mahasiswa Sederhana
#Fungsi untuk menambah data mahasiswa
def tambahMahasiswa():
    print("\n=== Tambah Data Mahasiswa ===")

    #Buka file untuk membaca data yang sudah ada
    file = open('mahasiswa.txt', 'r')
    isiFile = file.read()
    file.close()

    #Input data dari user
```

```

while True:
    nim = input("Masukkan NIM (12 angka): ")
    if not nim.isdigit():
        print("✗ NIM hanya boleh berisi angka!")
    elif len(nim) != 12:
        print("✗ NIM harus 12 angka!")
    elif nim in isiFile:
        print("✗ NIM sudah terdaftar!")
        return
    else:
        break

#Input nama dan jurusan
nama = input("Masukkan Nama: ")
jurusan = input("Masukkan Jurusan: ")

#Tulis data ke file
file = open('mahasiswa.txt', 'a') #Mode 'a' agar tidak menimpa
data lama
file.write(nim + "," + nama + "," + jurusan + "\n")
file.close()

print("Data berhasil ditambahkan!")

#Fungsi untuk menampilkan semua data mahasiswa
def tampilkanMahasiswa():
    print("\n=== Data Mahasiswa ===")

    #Buka file untuk dibaca
    file = open('mahasiswa.txt', 'r')
    data = file.readlines()
    file.close()

    #Cek apakah ada data
    if len(data) == 0:
        print("Tidak ada data!")
        return

    #Tampilkan data dengan format yang rapi
    print(f"{'No':<5} {'NIM':<15} {'Nama':<20} {'Jurusan':<20}")
    print("-" * 70)

    nomor = 1

```

```

for baris in data:
    if baris.strip() != "": # Abaikan baris kosong
        pisah = baris.split(",")
        nim = pisah[0]
        nama = pisah[1]
        jurusan = pisah[2].strip()

        # Format output agar rapi dan sejajar
        print(f"{nomor:<5} {nim:<15} {nama:<25} {jurusan:<20}")
        nomor += 1

print("-" * 70)

#Fungsi untuk menghapus data mahasiswa
def hapusMahasiswa():
    print("\n=== Hapus Data Mahasiswa ===")

    #Baca semua data
    file = open('mahasiswa.txt', 'r')
    data = file.readlines()
    file.close()

    nim = input("Masukkan NIM yang akan dihapus: ")

    #Cari dan hapus data
    dataBaru = [] #list untuk menyimpan data yang tidak dihapus
    ketemu = False

    for baris in data:
        pisah = baris.split(",")
        if pisah[0] != nim:
            dataBaru.append(baris)
        else:
            ketemu = True
            print("Data ditemukan:")
            print("NIM:", pisah[0])
            print("Nama:", pisah[1])
            print("Jurusan:", pisah[2].strip())

    if ketemu:
        yakin = input("Yakin ingin menghapus? (y/n): ")
        if yakin == 'y':
            #Tulis ulang data tanpa yang dihapus

```



```

        file = open('mahasiswa.txt', 'w')
        for baris in dataBaru:
            file.write(baris)
        file.close()
        print("Data berhasil dihapus!")
    else:
        print("Penghapusan dibatalkan")
else:
    print("NIM tidak ditemukan!")

#Program utama
print("=" * 50)
print("PROGRAM DATA MAHASISWA")
print("=" * 50)

#Cek apakah file ada, kalau tidak buat file baru
try:
    file = open('mahasiswa.txt', 'r')
    file.close()
except:
    file = open('mahasiswa.txt', 'w')
    file.close()
    print("File baru dibuat")

#Loop menu utama
while True:
    print("\n=== MENU UTAMA ===")
    print("1. Tambah Data")
    print("2. Tampilkan Data")
    print("3. Hapus Data")
    print("4. Keluar")

    pilihan = input("Pilih menu (1-4): ")

    if pilihan == '1':
        tambahMahasiswa()
    elif pilihan == '2':
        tampilkanMahasiswa()
    elif pilihan == '3':
        hapusMahasiswa()
    elif pilihan == '4':
        print("Terima kasih!")
        break

```

```
else:  
    print("Pilihan tidak valid!")
```

C. Penjelasan Program

1. Inisialisasi & Persiapan File: Program diawali dengan sebuah blok try-except yang berfungsi sebagai persiapan sistem penyimpanan. Blok ini memeriksa keberadaan file mahasiswa.txt. Jika file tersebut belum ada, program akan secara otomatis membuatnya. Langkah ini memastikan bahwa program selalu memiliki tempat untuk membaca dan menulis data, bahkan saat pertama kali dijalankan.
2. Perulangan Menu Utama: Logika inti program dibungkus dalam sebuah perulangan while True yang menampilkan menu utama secara terus-menerus. Perulangan ini menjaga agar program tetap berjalan dan siap menerima perintah dari pengguna sampai pengguna secara eksplisit memilih opsi untuk keluar.
3. Navigasi Menu dan Pemanggilan Fungsi: Di dalam setiap iterasi, program menampilkan daftar pilihan menu (Tambah, Tampilkan, Hapus, Keluar). Program kemudian menangkap input pengguna dan menggunakan struktur percabangan if-elif-else untuk mengeksekusi perintah yang sesuai. Setiap pilihan (kecuali keluar) akan memanggil fungsi spesifik yang telah dirancang untuk tugas tersebut (tambahMahasiswa(), tampilkanMahasiswa(), hapusMahasiswa()).
4. Logika Operasi Data pada Setiap Fungsi:
 - Penambahan Data (tambahMahasiswa): Sebelum menyimpan, fungsi ini melakukan validasi berlapis pada input NIM. Pertama, ia memeriksa apakah NIM hanya berisi angka dan panjangnya tepat 12 karakter. Kedua, ia membaca seluruh isi file mahasiswa.txt untuk

memastikan NIM yang dimasukkan belum pernah terdaftar sebelumnya, sehingga menjamin keunikan data.

- **Penampilan Data (tampilkanMahasiswa):** Fungsi ini bertanggung jawab untuk membaca seluruh baris data dari file. Kemudian, ia menggunakan perulangan for untuk memproses setiap baris, memisahkan NIM, nama, dan jurusan berdasarkan delimiter koma, dan menampilkannya dalam format tabel yang rapi dan mudah dibaca menggunakan f-string.
 - **Penghapusan Data (hapusMahasiswa):** Untuk menghapus data, program terlebih dahulu membaca semua catatan ke dalam sebuah list di memori. Program mencari baris yang cocok dengan NIM yang diinput. Jika ditemukan, data tersebut ditampilkan kepada pengguna untuk konfirmasi. Setelah konfirmasi, program akan menulis ulang seluruh isi file mahasiswa.txt dari awal, namun dengan mengecualikan data yang ingin dihapus.
5. **Manajemen File sebagai Penyimpanan Persisten:** Program ini memanfaatkan file mahasiswa.txt sebagai database sederhana. Ia secara dinamis menggunakan mode file yang berbeda sesuai kebutuhan: mode 'a' (append) untuk menambahkan data baru di akhir file tanpa menghapus yang lama, mode 'r' (read) untuk membaca data saat menampilkan atau melakukan validasi, dan mode 'w' (write) untuk menimpa file sepenuhnya saat operasi penghapusan data.
 6. **Sesi Akhir:** Perulangan utama akan berhenti ketika pengguna memilih opsi '4'. Pernyataan break akan dieksekusi, menghentikan loop, dan program akan menampilkan pesan penutup "Terima kasih!" sebelum benar-benar berakhir.

D. Contoh Output Program

1. Menjalankan program untuk pertama kali

```
=====
PROGRAM DATA MAHASISWA
=====
```

File baru dibuat

```
=== MENU UTAMA ===
```

1. Tambah Data
2. Tampilkan Data
3. Hapus Data
4. Keluar

Pilih menu (1-4):

2. Menampilkan data saat masih kosong

Pilih menu (1-4): 2

```
=== Data Mahasiswa ===
```

Tidak ada data!

```
=== MENU UTAMA ===
```

1. Tambah Data
2. Tampilkan Data
3. Hapus Data
4. Keluar

3. Menambahkan data mahasiswa baru (berhasil)

Pilih menu (1-4): **1**

```
=== Tambah Data Mahasiswa ===
```

Masukkan NIM (12 angka): **202311001122**

Masukkan Nama: **Budi Santoso**

Masukkan Jurusan: **Teknik Informatika**

Data berhasil ditambahkan!

==== MENU UTAMA ====

1. Tambah Data
2. Tampilkan Data
3. Hapus Data
4. Keluar

Pilih menu (1-4): **1**

==== Tambah Data Mahasiswa ====

Masukkan NIM (12 angka): **202312003344**

Masukkan Nama: **Citra Lestari**

Masukkan Jurusan: **Sistem Informasi**

Data berhasil ditambahkan!

4. Menampilkan data yang telah ditambahkan

Pilih menu (1-4): **2**

==== Data Mahasiswa ====

No	NIM	Nama	Jurusan

1	202311001122	Budi Santoso	Teknik Informatika
2	202312003344	Citra Lestari	Sistem Informasi

5. Menambahkan data dengan NIM yang salah

- NIM mengandung huruf

Pilih menu (1-4): **1**

==== Tambah Data Mahasiswa ====

Masukkan NIM (12 angka): **20231100abcd**

✗ NIM hanya boleh berisi angka!

Masukkan NIM (12 angka):

- Panjang NIM tidak sesuai
Masukkan NIM (12 angka): **12345**
✗ NIM harus 12 angka!
Masukkan NIM (12 angka):
- NIM telah terdaftar
Masukkan NIM (12 angka): **202311001122**
✗ NIM sudah terdaftar!

==== MENU UTAMA ====

...

6. Menghapus data mahasiswa

Pilih menu (1-4): **3**

==== Hapus Data Mahasiswa ====

Masukkan NIM yang akan dihapus: **202311001122**

Data ditemukan:

NIM: 202311001122

Nama: Budi Santoso

Jurusan: Teknik Informatika

Yakin ingin menghapus? (y/n): **y**

Data berhasil dihapus!

7. Menampilkan data keseluruhan setelah penghapusan

Pilih menu (1-4): **2**

==== Data Mahasiswa ====

No	NIM	Nama	Jurusan
----	-----	------	---------

1	202312003344	Citra Lestari	Sistem Informasi
---	--------------	---------------	------------------

8. Memilih opsi yang tidak valid

=== MENU UTAMA ===

1. Tambah Data
2. Tampilkan Data
3. Hapus Data
4. Keluar

Pilih menu (1-4): **5**

Pilihan tidak valid!

9. Keluar dari program

=== MENU UTAMA ===

1. Tambah Data
2. Tampilkan Data
3. Hapus Data
4. Keluar

Pilih menu (1-4): **4**

Terima kasih!

BAB III PENUTUP

A. Kesimpulan

Program manajemen data mahasiswa sederhana ini berhasil dibuat dengan Python. Program ini mampu menjalankan fungsi-fungsi esensial pengelolaan data, yaitu menambahkan, menampilkan, dan menghapus catatan mahasiswa, di mana semua data disimpan secara persisten dalam sebuah file teks.

Fitur unggulan dari program ini adalah penerapan aturan validasi yang ketat pada input NIM untuk menjaga integritas data, seperti pengecekan keunikan, format, dan panjang karakter. Selain itu, program ini juga dilengkapi dengan antarmuka menu yang interaktif, mekanisme konfirmasi sebelum

menghapus data, dan umpan balik status yang jelas, sehingga alur operasionalnya mudah untuk diikuti dan dipahami.

B. Saran

Laporan Praktikum ini jauh dari kata sempurna, oleh karena itu kami mengharapkan saran dan kritik yang bisa membuat Laporan Praktikum ini lebih baik lagi. Sekian Laporan Praktikum ini dibuat, semoga sistem manajemen sederhana yang kami buat berjalan dengan lancar dan bisa memberikan hasil yang maksimal dalam penilaian. Penulis mengucapkan Terima Kasih.

DAFTAR PUSTAKA

Taroreh, Dika.Aurelya.A., dan Fuady, Muhammad,Jauharul. (2024).
ALGORITMA DAN PEMROGRAMAN PYTHON. [Naskah yang belum
diterbitkan]