

LAPORAN TUGAS AKHIR

“Sistem manajemen nilai dan presensi mahasiswa berbasis CLI menggunakan bahasa pemrograman Python”

Dosen pengampu : Ni Luh Putu Ika Candraweni S.Stat.,M.Stat



Anggota:

Maria Stefania Misa

(42539937)

Adelia Surya Putri

(42530038)

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS PENDIDIKAN NASIONAL
2025

1. Penjelasan Struktur Program

Program manajemen nilai dan presensi mahasiswa ini dirancang dengan membagi proses kerja ke dalam beberapa fungsi. Pembagian ini dilakukan agar alur program lebih teratur, mudah dipahami, dan setiap fungsi dapat menangani tugas tertentu tanpa saling bercampur, sehingga program lebih mudah dikelola dan dikembangkan.

a. Alasan Pembagian Fungsi-Fungsi Program

Alasan pembagian fungsi tersebut adalah sebagai berikut:

1. Kode lebih terstruktur dan rapi

Setiap fungsi dibuat untuk satu pekerjaan tertentu, seperti menambah data mahasiswa, menghapus data, memasukkan nilai, mencatat kehadiran, dan menyimpan data ke file CSV. Dengan cara ini, program jadi lebih rapi dan lebih mudah dibaca.

2. Mudah dipahami dan dipelihara

Jika terjadi kesalahan atau ingin melakukan pengembangan fitur, perbaikan dapat dilakukan pada fungsi yang bermasalah tanpa harus mengubah keseluruhan program.

3. Menghindari pengulangan kode (reusability)

Beberapa fungsi dapat digunakan kembali di berbagai menu, misalnya fungsi `find_student_by_nim()` yang digunakan untuk mencari data mahasiswa berdasarkan NIM tanpa perlu menulis ulang kode pencarian.

4. Sesuai konsep modular programming

Program dibagi menjadi bagian input, proses, dan output sehingga alur kerja program menjadi lebih jelas dan sistematis.

Contoh pembagian fungsi dalam program ini antara lain:

- `tambah_mahasiswa()` → untuk menginput data mahasiswa baru
- `input_nilai_akademik()` → untuk menginput nilai tugas, UTS, dan UAS
- `input_presensi()` → untuk mencatat kehadiran mahasiswa setiap pertemuan
- `tampilkan_laporan()` → untuk menampilkan nilai akhir, grade, dan persentase kehadiran
- `load_data()` dan `save_data()` → untuk membaca dan menyimpan data ke file CSV

b. Penjelasan Struktur Class Student

Class Student digunakan untuk menggambarkan satu mahasiswa dalam program. Di dalam class ini disimpan data mahasiswa, seperti identitas dan nilai, serta fungsi-fungsi yang digunakan untuk mengolah data tersebut.

Atribut dalam class Student:

- nim → Nomor Induk Mahasiswa
- nama → Nama mahasiswa
- tugas → Nilai tugas
- uts → Nilai UTS
- uas → Nilai UAS
- presensi → List kehadiran mahasiswa (H = Hadir, A = Alpha, I = Izin)

Method dalam class Student:

- hitung_nilai_akhir()

Digunakan untuk menghitung nilai akhir mahasiswa berdasarkan bobot:

- Tugas: 30%
- UTS: 35%
- UAS: 35%

- to_csv_row()

Digunakan untuk mengubah data mahasiswa menjadi bentuk list agar dapat disimpan ke dalam file CSV.

- from_csv_row()

Digunakan untuk membaca data dari file CSV dan mengubahnya kembali menjadi objek Student.

Dengan menggunakan class Student, program ini menerapkan *konsep Object Oriented Programming (OOP)*, dimana data mahasiswa dan proses pengelolaannya digabungkan dalam satu struktur yang saling berkaitan.

2. Penjelasan Logika Khusus

a. Implementasi Advance Function (Konversi Grade)

Program menggunakan fungsi khusus untuk mengubah nilai angka menjadi nilai huruf A sampai E, yaitu fungsi konversi_grade(). Fungsi ini dibuat agar proses penentuan grade tidak bercampur dengan bagian lain dari program.

Dengan adanya fungsi ini:

1. Proses penilaian menjadi lebih sederhana
2. Aturan konversi nilai lebih mudah dipahami
3. Jika aturan grade berubah, cukup mengubah satu fungsi saja

Sehingga kode program menjadi lebih rapi dan tidak berulang.

b. Logika Presensi

Presensi mahasiswa disimpan dalam bentuk list, contohnya:

`['H', 'H', 'A', 'H']`

Logika yang digunakan adalah:

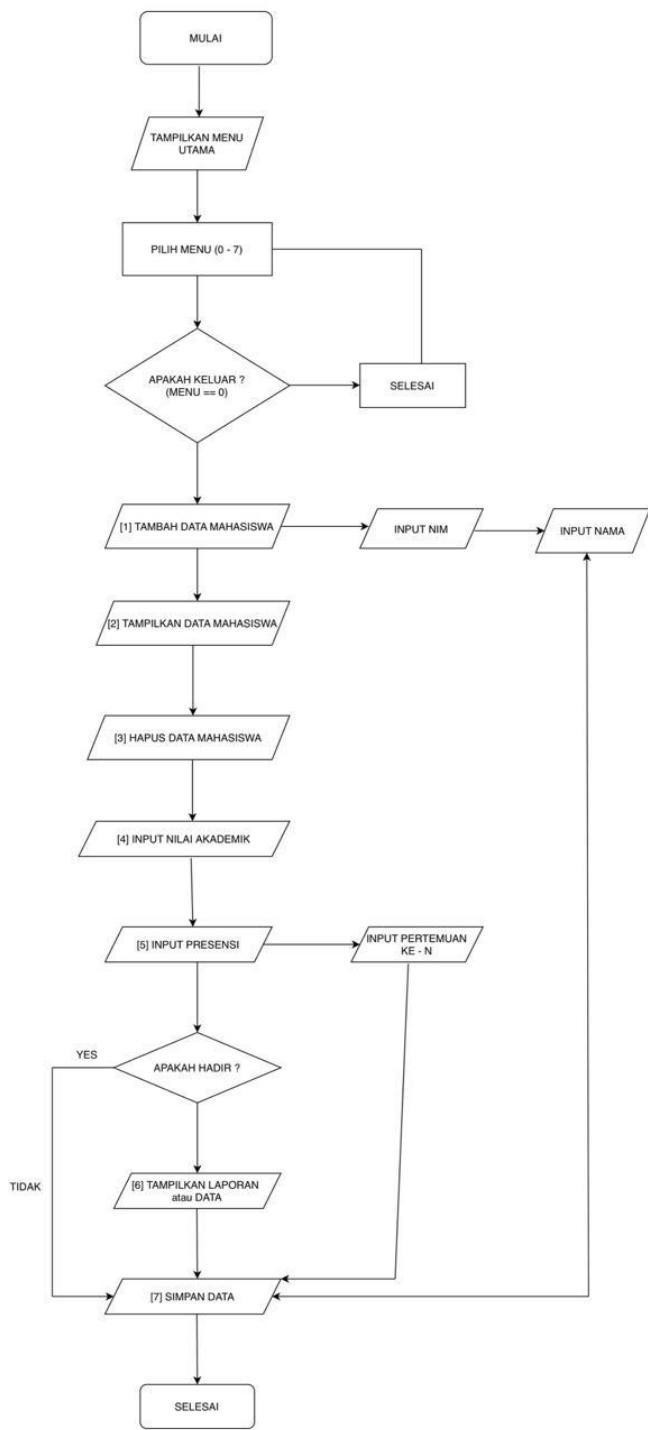
- Jumlah kehadiran dihitung dengan `s.presensi.count("H")`
- Total pertemuan dihitung dengan `len(s.presensi)`

Hasil presensi ditampilkan dalam bentuk:

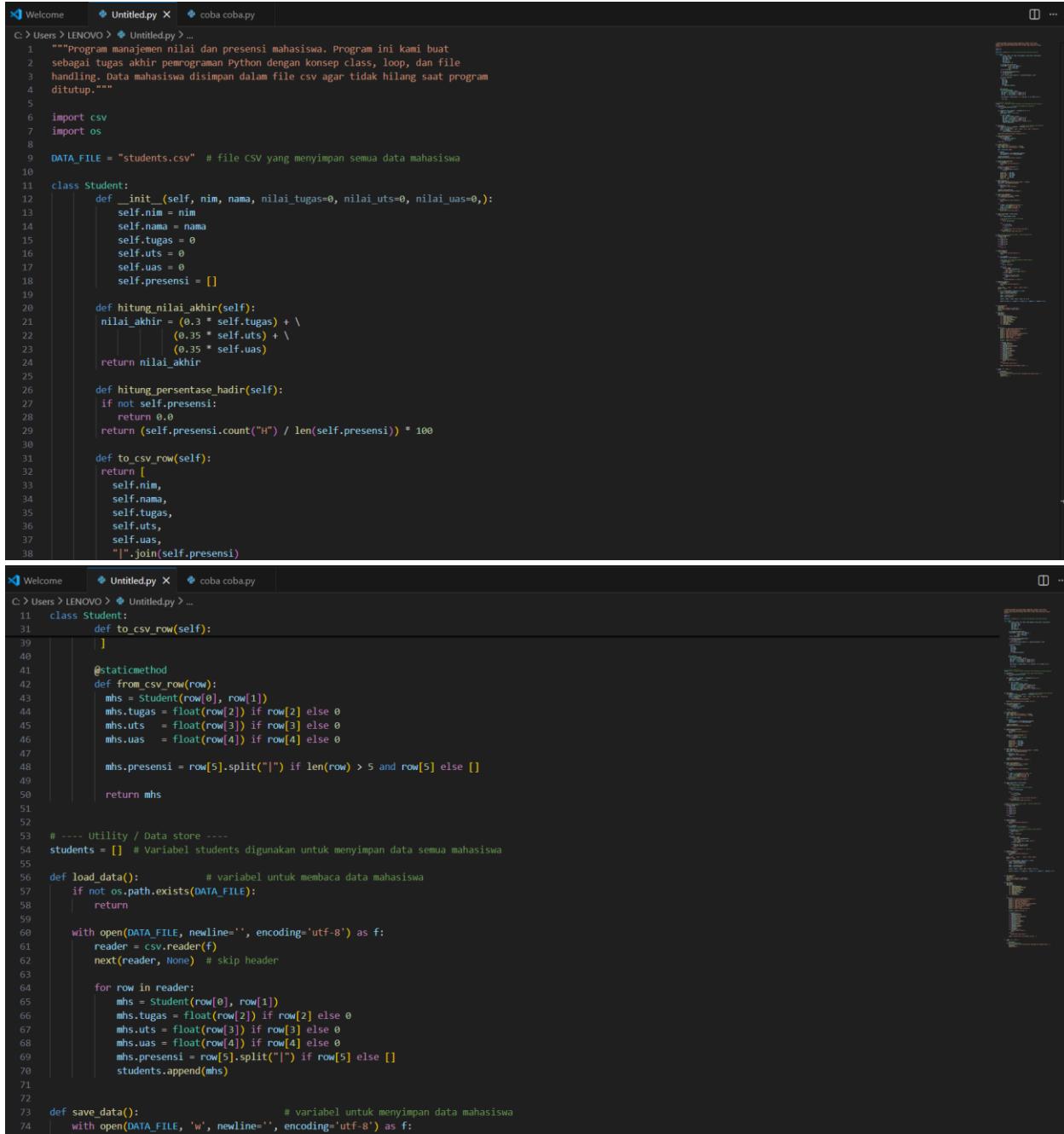
`jumlah` hadir / total pertemuan

Logika ini membuat sistem presensi menjadi lebih fleksibel karena jumlah pertemuan dapat bertambah tanpa harus ditentukan sejak awal. Selain itu, data presensi tetap tersimpan dengan baik dan tidak hilang saat program ditutup dan dijalankan kembali.

3. Flowchart (Diagram Alur)



4. Screenshot Output



The screenshot shows a code editor with two tabs open: 'Untitled.py' and 'coba_coba.py'. Both tabs are displayed in dark mode.

Untitled.py:

```
c:\> Users > LENOVO > Untitled.py > ...
1  """Program manajemen nilai dan presensi mahasiswa. Program ini kami buat
2  sebagai tugas akhir pemrograman Python dengan konsep class, loop, dan file
3  handling. Data mahasiswa disimpan dalam file csv agar tidak hilang saat program
4  ditutup."""
5
6  import csv
7  import os
8
9  DATA_FILE = "students.csv" # file CSV yang menyimpan semua data mahasiswa
10
11 class Student:
12     def __init__(self, nim, nama, nilai_tugas=0, nilai_uts=0, nilai_uas=0):
13         self.nim = nim
14         self.nama = nama
15         self.tugas = 0
16         self.uts = 0
17         self.uas = 0
18         self.presensi = []
19
20     def hitung_nilai_akhir(self):
21         nilai_akhir = (0.3 * self.tugas) + \
22                         (0.35 * self.uts) + \
23                         (0.35 * self.uas)
24         return nilai_akhir
25
26     def hitung_persentase_hadir(self):
27         if not self.presensi:
28             return 0.0
29         return (self.presensi.count("H") / len(self.presensi)) * 100
30
31     def to_csv_row(self):
32         return [
33             self.nim,
34             self.nama,
35             self.tugas,
36             self.uts,
37             self.uas,
38             "|".join(self.presensi)
39
40
41     @staticmethod
42     def from_csv_row(row):
43         mhs = Student(row[0], row[1])
44         mhs.tugas = float(row[2]) if row[2] else 0
45         mhs.uts = float(row[3]) if row[3] else 0
46         mhs.uas = float(row[4]) if row[4] else 0
47
48         mhs.presensi = row[5].split("|") if len(row) > 5 and row[5] else []
49
50         return mhs
51
52
53     # ---- Utility / Data store ----
54     students = [] # Variabel students digunakan untuk menyimpan data semua mahasiswa
55
56     def load_data(): # variabel untuk membaca data mahasiswa
57         if not os.path.exists(DATA_FILE):
58             return
59
60         with open(DATA_FILE, newline='', encoding='utf-8') as f:
61             reader = csv.reader(f)
62             next(reader, None) # skip header
63
64             for row in reader:
65                 mhs = Student(row[0], row[1])
66                 mhs.tugas = float(row[2]) if row[2] else 0
67                 mhs.uts = float(row[3]) if row[3] else 0
68                 mhs.uas = float(row[4]) if row[4] else 0
69                 mhs.presensi = row[5].split("|") if row[5] else []
70                 students.append(mhs)
71
72
73     def save_data(): # variabel untuk menyimpan data mahasiswa
74         with open(DATA_FILE, 'w', newline='', encoding='utf-8') as f:
```

coba_coba.py:

```
c:\> Users > LENOVO > Untitled.py > ...
11  class Student:
12      def to_csv_row(self):
13          ]
14
15      @staticmethod
16      def from_csv_row(row):
17          mhs = Student(row[0], row[1])
18          mhs.tugas = float(row[2]) if row[2] else 0
19          mhs.uts = float(row[3]) if row[3] else 0
20          mhs.uas = float(row[4]) if row[4] else 0
21
22          mhs.presensi = row[5].split("|") if len(row) > 5 and row[5] else []
23
24          return mhs
25
26
27     # ---- Utility / Data store ----
28     students = [] # Variabel students digunakan untuk menyimpan data semua mahasiswa
29
30     def load_data(): # variabel untuk membaca data mahasiswa
31         if not os.path.exists(DATA_FILE):
32             return
33
34         with open(DATA_FILE, newline='', encoding='utf-8') as f:
35             reader = csv.reader(f)
36             next(reader, None) # skip header
37
38             for row in reader:
39                 mhs = Student(row[0], row[1])
40                 mhs.tugas = float(row[2]) if row[2] else 0
41                 mhs.uts = float(row[3]) if row[3] else 0
42                 mhs.uas = float(row[4]) if row[4] else 0
43                 mhs.presensi = row[5].split("|") if row[5] else []
44                 students.append(mhs)
45
46
47     def save_data(): # variabel untuk menyimpan data mahasiswa
48         with open(DATA_FILE, 'w', newline='', encoding='utf-8') as f:
```

```

Welcome Untitled.py coba coba.py
C:\Users> lenovo > Untitled.py ...
73     def save_data():
74         # variabel untuk menyimpan data mahasiswa
75         writer = csv.writer(f)
76         writer.writerow(["NIM", "Nama", "Tugas", "UTS", "UAS", "Presensi"])
77         for s in students:
78             writer.writerow(s.to_csv_row())
79
80         print("Data mahasiswa berhasil disimpan ke file.")
81
82     def find_student_by_nim(nim):
83         for mhs in students:
84             if isinstance(mhs, Student):
85                 if mhs.nim == nim:
86                     return mhs
87
88         return None
89
90     # ---- Menu features ----
91     def tambah_mahasiswa():
92         nim = input("Input NIM mahasiswa: ").strip()
93         nama = input("Input nama mahasiswa: ").strip()
94
95         mhs = Student(nim, nama)
96
97         if students:
98             total_pertemuan = len(students[0].presensi)
99             mhs.presensi = ["A"] * total_pertemuan
100
101         students.append(mhs)
102
103     def tampilkan_data_mahasiswa():
104         if not students:
105             print("Belum ada data mahasiswa.")
106             return
107
108     print("\n==== DAFTAR MAHASISWA ===")
109     for mhs in students:
110         if not isinstance(mhs, Student):
111
112             continue
113
114             print("NIM : ", mhs.nim)
115             print("Nama : ", mhs.nama)
116             print("Tugas: ", mhs.tugas)
117             print("UTS : ", mhs.uts)
118             print("UAS : ", mhs.uas)
119             print("-" * 30)
120
121     def hapus_mahasiswa():
122         nim = input("Masukkan NIM yang akan dihapus: ").strip()
123         mahasiswa = find_student_by_nim(nim)
124
125         if mahasiswa is None:
126             print("Data tidak ditemukan")
127             return
128
129         students.remove(mahasiswa)
130         print("Data mahasiswa berhasil dihapus.")
131
132     def input_nilai_akademik():
133         nim = input("Masukkan NIM: ").strip()
134         s = find_student_by_nim(nim)
135
136         if s is None:
137             print("Mahasiswa tidak ditemukan.")
138             return
139
140         try:
141             s.tugas = float(input("Nilai Tugas: "))
142             s.uts = float(input("Nilai UTS: "))
143             s.uas = float(input("Nilai UAS: "))
144             print("Nilai berhasil disimpan.")
145         except ValueError:
146             print("Input nilai harus angka.")
147
148

```

```
❶ Welcome Untitled.py coba coba.py
C:\Users\LENOVO> Untitled.py > ...
149
150     def input_score(prompt, current_value):
151         while True:
152             val = input(prompt).strip()
153
154             # jika ENTER ditekan, nilai lama dipakai
155             if val == "":
156                 return current_value
157
158             try:
159                 f = float(val)
160                 if 0 <= f <= 100:
161                     return f
162                 else:
163                     print("Nilai harus di antara 0 dan 100.")
164             except ValueError:
165                 print("Masukkan angka yang valid.")
166
167
168     # Lambda function untuk konversi grade = konversi_grade(nilai)
169     def konversi_grade(nilai):
170         if nilai >= 85:
171             return "A"
172         elif nilai >= 70:
173             return "B"
174         elif nilai >= 55:
175             return "C"
176         elif nilai >= 40:
177             return "D"
178         else:
179             return "E"
180
181
182     def input_presensi():
183         if not students:
184             print("Belum ada data mahasiswa.")
185         return
186
```

```
❶ Welcome Untitled.py coba coba.py
C:\Users\LENOVO> Untitled.py > ...
182     def input_presensi():
183         for s in students:
184             pertemuan_ke = len(s.presensi) + 1
185
186             # Menentukan sesi berdasarkan pertemuan (logika modulus)
187             if pertemuan_ke % 2 == 1:
188                 sesi = "teori"
189             else:
190                 sesi = "Praktikum"
191
192             while True:
193                 status = input(
194                     f"Pertemuan {pertemuan_ke} | "
195                     f"Sesi {sesi} | "
196                     f"NIM {s.nim} Nama {s.nama} (H/A/I): "
197                 ).strip().upper()
198
199                 if status in ("H", "A", "I"):
200                     s.presensi.append(status)
201                     break
202                 else:
203                     print("Masukkan H, A, atau I.")
204
205         def tampilan_laporan():
206             if not students:
207                 print("Belum ada data mahasiswa.")
208             return
209
210             print("\nNIM | Nama | Akhir | Grade | Hadir")
211             print("-" * 50)
212
213             for s in sorted(students, key=lambda x: x.nim):
214                 nilai = s.hitung_nilai_akhir()
215                 grade = konversi_grade(nilai)
216
217                 hadir = s.presensi.count('H')
218                 total = len(s.presensi)
```

```
>Welcome Untitled.py x coba.coba.py
C:\Users\LENOVO> Untitled.py ...
209 def tampilkan_laporan():
210
211     persen = (hadir / total * 100) if total > 0 else 0
212
213     print(f"{{s.nim:<6}} | {{s.nama:<8}} | {{nilai:<5.1f}} | {{grade:<5}} | {{persen:<5.1f}}")
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230 def save_and_exit():
231     save_data()
232     print("Data disimpan ke", DATA_FILE)
233     print("Keluar program. Sampai jumpa!")
234     exit(0)
235
236 def main_menu():
237     load_data()
238     menu_actions = {
239         "1": tambah_mahasiswa,
240         "2": tampilkan_data_mahasiswa,
241         "3": hapus_mahasiswa,
242         "4": input_nilai_akademik,
243         "5": input_presensi,
244         "6": tampilkan_laporan,
245         "7": save_data,
246         "8": save_and_exit
247     }
248
249     while True:
250         print("\n==== MENU UTAMA SISTEM MAHASISWA ====")
251         print("1. Tambah data Mahasiswa")
252         print("2. Tampilkan Data Mahasiswa")
253         print("3. Hapus data Mahasiswa")
254         print("4. Input Nilai Akademik (Tugas/UTS/UAS)")
255         print("5. Input Presensi (pertemuan)")
256         print("6. Tampilkan Laporan / Data")
257         print("7. Simpan Data")
258         print("8. Keluar (Simpan otomatis)")
259
260
261     pilihan = input("pilih menu: ")
262
263     if pilihan == "1":
264         tambah_mahasiswa()
265     elif pilihan == "2":
266         tampilkan_data_mahasiswa()
267     elif pilihan == "3":
268         hapus_mahasiswa()
269     elif pilihan == "4":
270         input_nilai_akademik()
271     elif pilihan == "5":
272         input_presensi()
273     elif pilihan == "6":
274         tampilkan_laporan()
275     elif pilihan == "7":
276         save_data()
277     elif pilihan == "8":
278         print("program selesai.")
279         break
280     else:
281         print("Menu tidak valid.")
282
283     input ("\nTekan Enter untuk kembali ke menu...")
284
285 if __name__ == "__main__":
286     try:
287         main_menu()
288     except KeyboardInterrupt:
289         print("\nKeyboard interrupt diterima. Menyimpan data sebelum keluar...")
290         save_data()
291         print("selesai.")
```