

LAPORAN PRAKTIKUM
POSTTEST 8
ALGORITMA PEMROGRAMAN DASAR



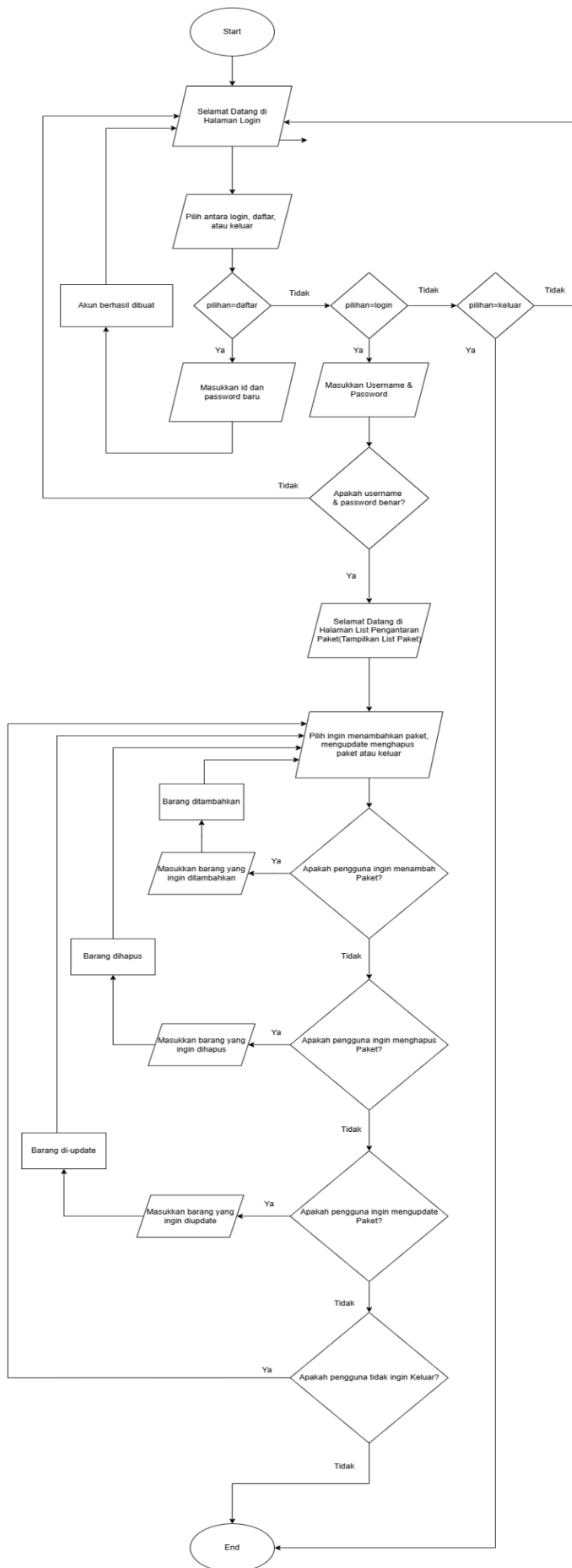
Disusun oleh:

Muhammad Noval Arifinnur **2509106079**

Kelas B2'25

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Gambar 1.1 Flowchart

Penjelasan singkat mengenai Flowchart:

Pada flowchart diatas kita dapat melihat tentang halaman login dan juga dapat membuat, menambahkan, melihat, dan menghapus berbagai data di dalamnya.

2. Deskripsi Singkat Program

Program ini bertujuan untuk menampilkan halaman yang pengantaran paket untuk kurir yang dapat dilihat, di tambah, di kurangi. Dan program ini juga berguna untuk memudahkan para kurir agar dapat melihat daftar paket paket mana saja yang perlu diantar.

3. Source Code

A. Variabel Global

Menampilkan 3 variabel global yang digunakan pada program

Source Code:



```
1 pengguna = {'Noval': '079'}
2 paket = ['Paket A', 'Paket B', 'Paket C', 'Paket D', 'Paket E', 'Paket F', 'Paket G', 'Paket H']
3 program_berjalan=True
4
```

Gambar 3.1

B. Fungsi untuk membersihkan terminal

Pada fungsi ini ditampilkan command untuk membersihkan terminal agar terlihat lebih rapi dengan command `os.system('cls')`

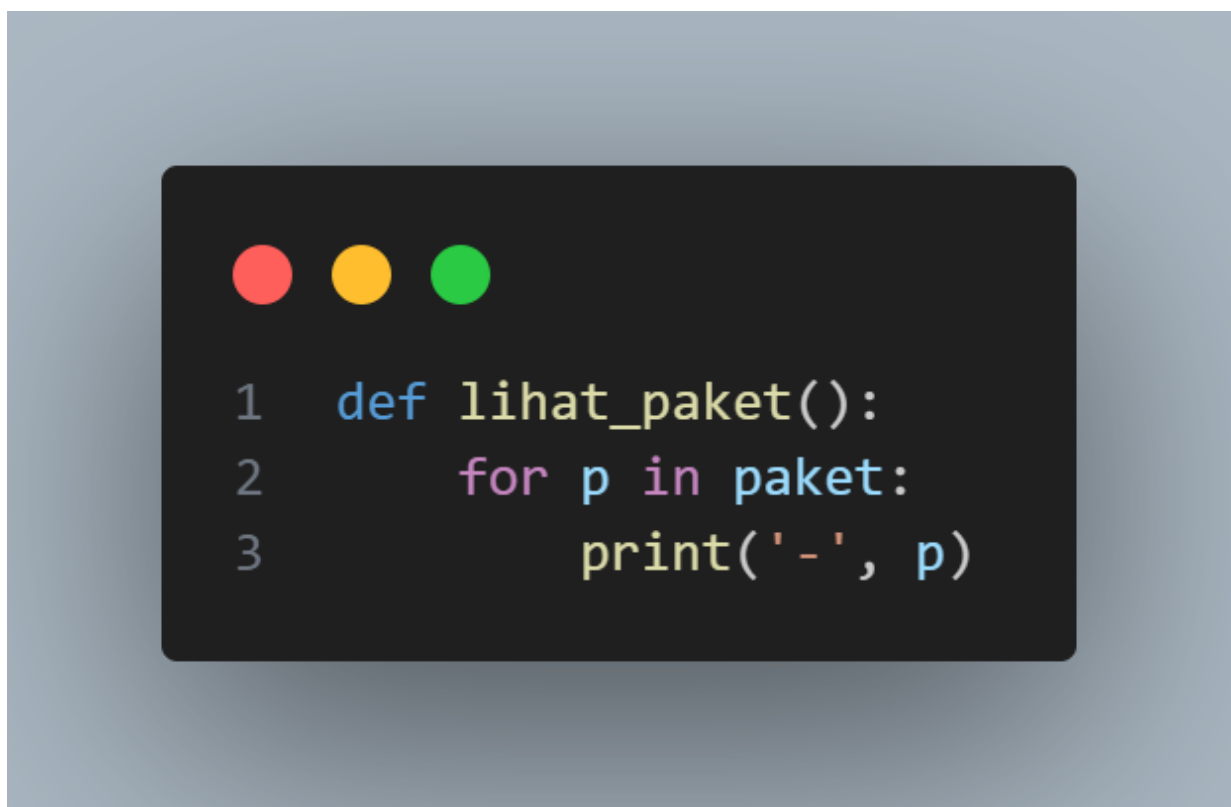
Source Code:



Gambar 3.2

C. Fungsi untuk menampilkan paket


Source Code:



Gambar 3.3

D. Fungsi untuk menambahkan pengguna baru

Source Code:




```
1 def daftar_pengguna(u, p):  
2     if not u or not p:  
3         return False  
4     if u in pengguna:  
5         return False  
6     pengguna[u] = p  
7     return True
```

Gambar 3.4

E. Fungsi untuk mengecek status login

Source Code:

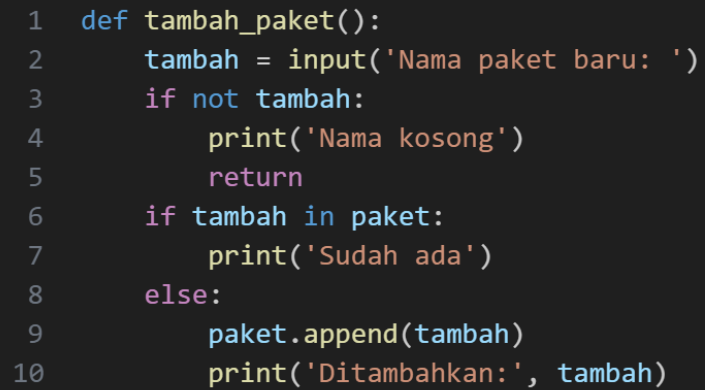


```
1  
2 def cek_login(u, p):  
3     return pengguna.get(u) == p
```

Gambar 3.5

F. Fungsi untuk menambah paket

Source Code:

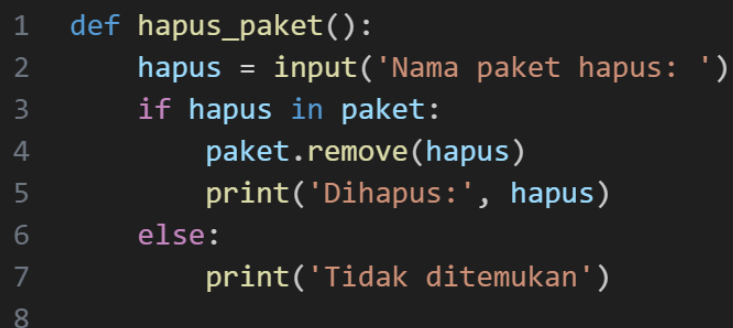
A screenshot of a code editor with a dark background and three colored window control buttons (red, yellow, green) at the top left. The code is written in Python and defines a function named 'tambah_paket()'. It takes an input 'tambah' and checks if it's empty. If empty, it prints 'Nama kosong' and returns. If not empty, it checks if 'tambah' is already in the 'paket' list. If it is, it prints 'Sudah ada'. If not, it appends 'tambah' to the 'paket' list and prints 'Ditambahkan:', followed by 'tambah'.

```
1 def tambah_paket():
2     tambah = input('Nama paket baru: ')
3     if not tambah:
4         print('Nama kosong')
5         return
6     if tambah in paket:
7         print('Sudah ada')
8     else:
9         paket.append(tambah)
10        print('Ditambahkan:', tambah)
```

Gambar 3.6

G. Fungsi untuk menghapus paket

Source Code:

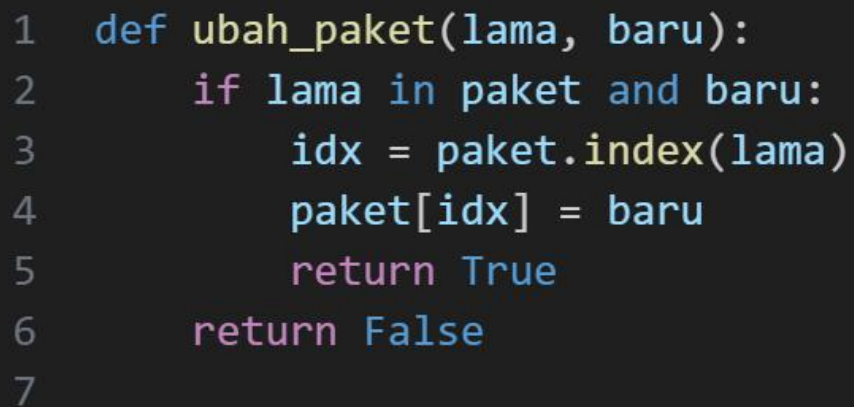
A screenshot of a code editor with a dark background and three colored window control buttons (red, yellow, green) at the top left. The code is written in Python and defines a function named 'hapus_paket()'. It takes an input 'hapus' and checks if it's in the 'paket' list. If it is, it removes 'hapus' from the list and prints 'Dihapus:', followed by 'hapus'. If not, it prints 'Tidak ditemukan'.

```
1 def hapus_paket():
2     hapus = input('Nama paket hapus: ')
3     if hapus in paket:
4         paket.remove(hapus)
5         print('Dihapus:', hapus)
6     else:
7         print('Tidak ditemukan')
8
```

Gambar 3.7

H. Fungsi untuk mengubah paket

Source Code:

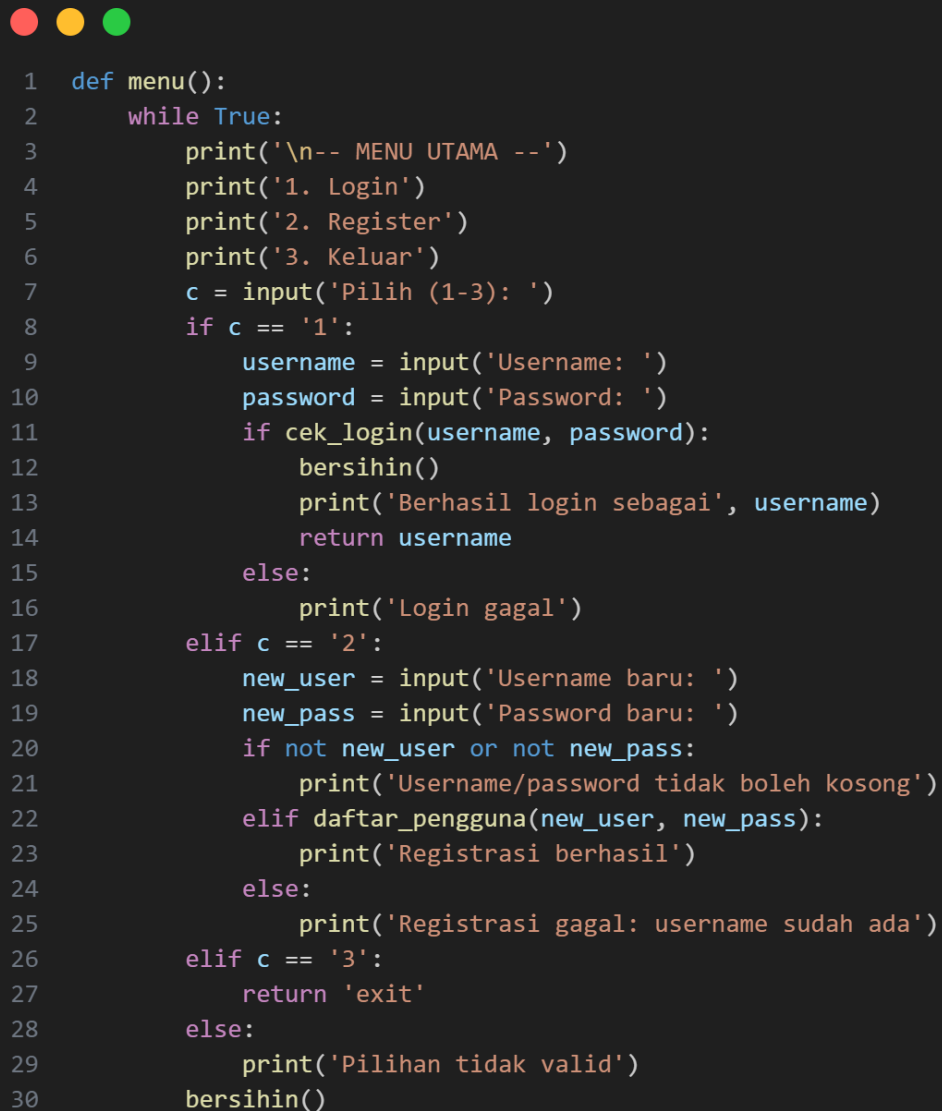


```
1 def ubah_paket(lama, baru):
2     if lama in paket and baru:
3         idx = paket.index(lama)
4         paket[idx] = baru
5         return True
6     return False
7
```

Gambar 3.8

I. Fungsi untuk menu utama

Source Code:



```

1  def menu():
2      while True:
3          print('\n-- MENU UTAMA --')
4          print('1. Login')
5          print('2. Register')
6          print('3. Keluar')
7          c = input('Pilih (1-3): ')
8          if c == '1':
9              username = input('Username: ')
10             password = input('Password: ')
11             if cek_login(username, password):
12                 bersihin()
13                 print('Berhasil login sebagai', username)
14                 return username
15             else:
16                 print('Login gagal')
17         elif c == '2':
18             new_user = input('Username baru: ')
19             new_pass = input('Password baru: ')
20             if not new_user or not new_pass:
21                 print('Username/password tidak boleh kosong')
22             elif daftar_pengguna(new_user, new_pass):
23                 print('Registrasi berhasil')
24             else:
25                 print('Registrasi gagal: username sudah ada')
26         elif c == '3':
27             return 'exit'
28         else:
29             print('Pilihan tidak valid')
30         bersihin()

```

Gambar 3.9

J. Fungsi untuk menu paket

Source Code:

```

1  def menu_paket():
2      while True:
3          print('\n-- PAKET MENU --')
4          print('1. Lihat Paket')
5          print('2. Tambah Paket')
6          print('3. Update Paket')
7          print('4. Hapus Paket')
8          print('5. Kembali')
9          c = input('Pilih (1-5): ')
10         if c == '1':
11             bersihin()
12             print('Daftar paket:')
13             table = PrettyTable()
14             table.field_names = ["No", "Nama Paket"]
15             for i, p in enumerate(paket, 1):
16                 table.add_row([i, p])
17             print(table)
18         elif c == '2':
19             print('Daftar paket:')
20             table = PrettyTable()
21             table.field_names = ["No", "Nama Paket"]
22             for i, p in enumerate(paket, 1):
23                 table.add_row([i, p])
24             print(table)
25             tambah_paket()
26         elif c == '3':
27             print('Daftar paket:')
28             table = PrettyTable()
29             table.field_names = ["No", "Nama Paket"]
30             for i, p in enumerate(paket, 1):
31                 table.add_row([i, p])
32             print(table)
33             lama = input('Nama paket yang diupdate: ')
34             baru = input('Nama paket baru: ')
35             if ubah_paket(lama, baru):
36                 print('Paket berhasil diupdate')
37             else:
38                 print('Gagal update paket')
39         elif c == '4':
40             print('Daftar paket:')
41             table = PrettyTable()
42             table.field_names = ["No", "Nama Paket"]
43             for i, p in enumerate(paket, 1):
44                 table.add_row([i, p])
45             print(table)
46             hapus_paket()
47         elif c == '5':
48             return
49         else:
50             print('Pilihan tidak valid')

```

Gambar 3.10

K. Program Utama

Source Code:

```
1  pengguna = {'Noval': '079'}
2  paket = ['Paket A', 'Paket B', 'Paket C', 'Paket D', 'Paket E', 'Paket F', 'Paket G', 'Paket H']
3  program_berjalan=True
4  import os
5  def bersihin():
6      os.system('cls')
7
8
9  from login import menu
10 from crud import menu_paket
11
12 try:
13     while program_berjalan:
14         hasil = menu()
15         if hasil == 'exit':
16             print('Keluar...')
17             program_berjalan = False
18             break
19         elif hasil:
20             menu_paket()
21 except KeyboardInterrupt:
22     print('\nProgram dihentikan')
```

Gambar 3.11

4. Hasil Output

```
-- MENU UTAMA --
1. Login
2. Register
3. Keluar
Pilih (1-3): █
```

Gambar 4.1 Tampilan Menu

```
-- MENU UTAMA --  
1. Login  
2. Register  
3. Keluar  
Pilih (1-3): 2  
Username baru: Mamat  
Password baru: 123
```

Gambar 4.2 Tampilan Register

```
Berhasil login sebagai Mamat  
  
-- PAKET MENU --  
1. Lihat Paket  
2. Tambah Paket  
3. Update Paket  
4. Hapus Paket  
5. Kembali  
Pilih (1-5):
```

Gambar 4.3 Tampilan Login Setelah Register

Daftar paket:

+-----+	
No	Nama Paket
+-----+	
1	Paket A
2	Paket B
3	Paket C
4	Paket D
5	Paket E
6	Paket F
7	Paket G
8	Paket H
+-----+	

-- PAKET MENU --

1. Lihat Paket
 2. Tambah Paket
 3. Update Paket
 4. Hapus Paket
 5. Kembali
- Pilih (1-5): █

Gambar 4.4 Tampilan Untuk Pilihan 1

```
Pilih (1-5): 2
Daftar paket:
+-----+-----+
| No | Nama Paket |
+-----+-----+
| 1 | Paket A |
| 2 | Paket B |
| 3 | Paket C |
| 4 | Paket D |
| 5 | Paket E |
| 6 | Paket F |
| 7 | Paket G |
| 8 | Paket H |
+-----+-----+
Nama paket baru: Paket Baru
```

Gambar 4.5 Tampilan Untuk Pilihan Tambah Paket

```
Ditambahkan: Paket Baru

-- PAKET MENU --
1. Lihat Paket
2. Tambah Paket
3. Update Paket
4. Hapus Paket
5. Kembali
Pilih (1-5):
```

Gambar 4.6 Tampilan Setelah Ditambahkan

```
-- PAKET MENU --
1. Lihat Paket
2. Tambah Paket
3. Update Paket
4. Hapus Paket
5. Kembali
Pilih (1-5): 3
Daftar paket:
+----+-----+
| No | Nama Paket |
+----+-----+
| 1  | Paket A    |
| 2  | Paket B    |
| 3  | Paket C    |
| 4  | Paket D    |
| 5  | Paket E    |
| 6  | Paket F    |
| 7  | Paket G    |
| 8  | Paket H    |
| 9  | Paket Baru |
+----+-----+
Nama paket yang diupdate: █
```

Gambar 4.7 Tampilan Untuk Pilihan Update Paket

Paket berhasil diupdate

-- PAKET MENU --

1. Lihat Paket
2. Tambah Paket
3. Update Paket
4. Hapus Paket
5. Kembali

Pilih (1-5):

Gambar 4.8 Tampilan Setelah Di-Update

-- PAKET MENU --

1. Lihat Paket
2. Tambah Paket
3. Update Paket
4. Hapus Paket
5. Kembali

Pilih (1-5): 4

Daftar paket:

+---+-----+		
No Nama Paket		
+---+-----+		
1	Paket A	
2	Paket B	
3	Paket C	
4	Paket D	
5	Paket E	
6	Paket F	
7	Paket G	
8	Paket H	
9	Paket Update	

+---+-----+

Nama paket hapus:

Gambar 4.9 Tampilan Pilihan Hapus Paket

```
Nama paket hapus: Paket Update
Dihapus: Paket Update

-- PAKET MENU --
1. Lihat Paket
2. Tambah Paket
3. Update Paket
4. Hapus Paket
5. Kembali
Pilih (1-5):
```

Gambar 4.10 Tampilan Setelah Dihapus

```
-- MENU UTAMA --
1. Login
2. Register
3. Keluar
Pilih (1-3): 3
Keluar...
PS C:\APD-B2-25>
```

5. Langkah-langkah GIT

5.1 GIT Add

Command `git add` . ini berfungsi untuk menyimpan semua perubahan yang telah dilakukan oleh user baik perubahan penambahan atau pengurangan akan disimpan di staging area.

```
PS C:\APD-B2-25> git add .
warning: in the working copy of 'pertemuan-7/Post-Test-7/APD-2509106079-PT-7.xml', LF will be replaced by CRLF the next time Git touches it
```

Gambar 5.1 GIT Add

5.2 GIT Commit

Command `git commit` digunakan untuk menyimpan perubahan dari staging area ke repo lokal.

```
PS C:\APD-B2-25> git commit -m "Upload Lagi"
[main f1c97da] Upload Lagi
 2 files changed, 277 insertions(+)
 create mode 100644 pertemuan-7/Post-Test-7/APD-2509106079-PT-7.xml
 delete mode 100644 pertemuan-7/Post-Test-7/__pycache__/_APD-2509106079-PT-7.cpython-313.pyc
PS C:\APD-B2-25>
```

Gambar 5.2 GIT Commit

5.3 GIT Push

Command `git push` digunakan untuk mengirim commit ke github dengan alamatnya berdasarkan git remote.

```
PS C:\APD-B2-25> git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 16 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 2.70 KiB | 2.70 MiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Opang6411/APD-B2-25.git
 f027b74..f1c97da main -> main
PS C:\APD-B2-25>
```

Gambar 5.3 GIT Push