

**LAPORAN PRAKTIKUM**  
**POSTTEST (5)**  
**ALGORITMA PEMROGRAMAN DASAR**

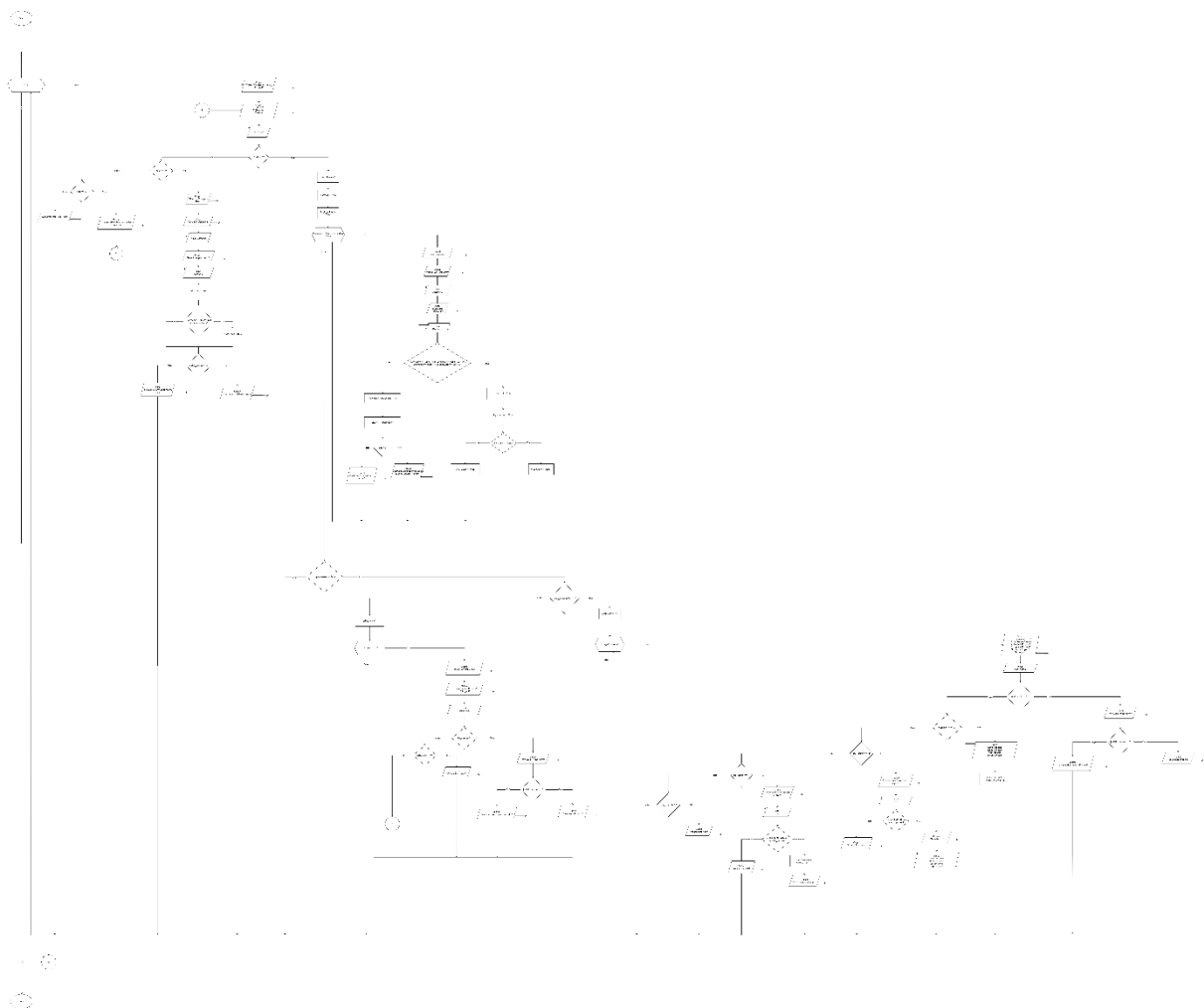


**Disusun oleh:**  
**Rendy (2509106069)**  
**(B2'25)**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

## 1. Flowchart

Manajemen Listing Properti adalah aplikasi berbasis teks yang memiliki fasilitas untuk mengelola data properti secara interaktif. Program ini dijalankan dengan menampilkan menu utama yang berisi pilihan login, register, dan keluar. Apabila pengguna memilih login, sistem memverifikasi login dengan batas hingga tiga kali percobaan. Jika login berhasil, pengguna dibawa ke menu sesuai dengan tingkat aksesnya. Admin dapat mengakses semua fungsi data properti, seperti melihat, menambah, mengubah, dan menghapus data. Sedangkan, user hanya dapat melihat daftar properti yang telah disediakan. Fasilitas register di program ini dapat digunakan untuk membuat akun baru dengan level user. Program ini akan terus berulang menampilkan menu utama hingga pengguna memilih keluar sebagai tanda bahwa program telah berakhir.



## **2. Deskripsi Singkat Program**

Manajemen Listing Properti adalah aplikasi konsol yang mengelola dan membantu data untuk properti seperti rumah, restoran, dan bangunan lainnya. Program ini termasuk sistem login dan registrasi untuk dua peran, admin dan pengguna. Admin melakukan operasi CRUD (Buat, Baca, Perbarui, Hapus) pada data properti, dan pengguna hanya bisa mengakses tampilan daftar properti. Data ditampilkan dengan rapi menggunakan pustaka PrettyTable. Keamanan login ditingkatkan dengan menggunakan pwinput yang melindungi kata sandi dan membuat sistem lebih aman. Program ini dimaksudkan untuk memperkenalkan dasar-dasar manajemen data, dan sistem autentikasi yang sederhana.

## **3. Source Code**

### **A. Fitur Menampilkan menu utama**

Menampilkan tiga opsi utama:

1. Login
2. Register
3. Keluar dari program

```

1  while True:
2      os.system("cls || clear")
3      print("🏠 SISTEM MANAJEMEN LISTING PROPERTI 🏠.")
4      tabel_menu = PrettyTable()
5      tabel_menu.field_names = ["No", "🗝️ Menu Utama"]
6      tabel_menu.align = "l"
7      tabel_menu.hrules = 1
8      tabel_menu.add_row(["1", "🔑 Login"])
9      tabel_menu.add_row(["2", "📄 Register"])
10     tabel_menu.add_row(["3", "🚪 Keluar"])
11     print(tabel_menu)
12
13     opsi = input("Pilih Menu : ")

```

## B. Fitur Login

1. Pengguna harus memasukkan username dan password.
2. Sistem mencocokkan data dengan list data\_pengguna.
3. Terdapat batas percobaan login maksimal 3 kali.
4. Jika gagal 3 kali berturut turut, pengguna harus menunggu 5 menit sebelum mencoba lagi.
5. Menggunakan library pwininput untuk menyembunyikan input password (dengan tanda \*).

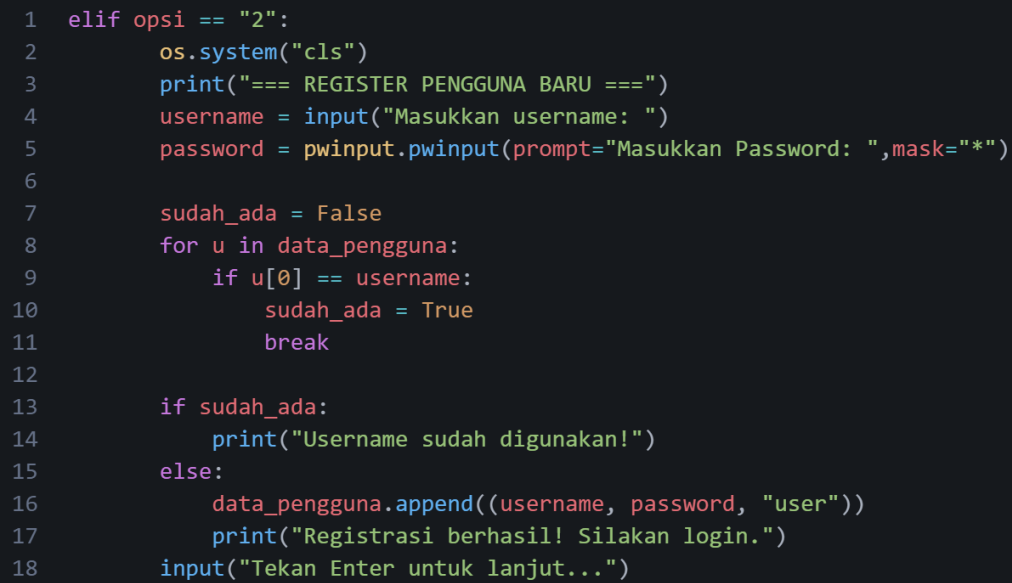
```

1  if opsi == "1":
2      os.system("cls || clear")
3      print("=== LOGIN ===")
4      percobaan = 0
5      berhasil = False
6
7      while percobaan < 3 and not berhasil:
8          username = input("Masukkan Username: ").lower()
9          password = pwinput.pwinput(prompt="Masukkan Password: ",mask="*")
10
11         level_akses = None
12         for u in data_pengguna:
13             if u[0] == username and u[1] == password:
14                 level_akses = u[2]
15                 break
16
17         if level_akses: # login berhasil
18             berhasil = True
19         else:
20             percobaan += 1
21             sisa = 3 - percobaan
22             if sisa > 0:
23                 print(f"Username atau password salah! Sisa percobaan: {sisa}")
24             else:
25                 print("Username atau password salah 3 kali. Silakan tunggu 5 menit sebelum mencoba lagi.")
26                 time.sleep(300)

```

### C. Fitur Register

1. Pengguna baru dapat membuat akun dengan username dan password.
2. Sistem akan memberikan apakah username sudah digunakan.
3. Jika belum, akun baru akan otomatis disimpan sebagai user(bukan admin).
4. Password disembunyikan saat input menggunakan pwinput.



```
1 elif opsi == "2":
2     os.system("cls")
3     print("=== REGISTER PENGGUNA BARU ===")
4     username = input("Masukkan username: ")
5     password = pwinput.pwinput(prompt="Masukkan Password: ",mask="*")
6
7     sudah_ada = False
8     for u in data_pengguna:
9         if u[0] == username:
10             sudah_ada = True
11             break
12
13     if sudah_ada:
14         print("Username sudah digunakan!")
15     else:
16         data_pengguna.append((username, password, "user"))
17         print("Registrasi berhasil! Silakan login.")
18     input("Tekan Enter untuk lanjut...")
```

#### D. Fitur hak akses berdasarkan Level

1. Admin : dapat mengelola semua data properti (CRUD)
2. User: hanya dapat melihat daftar properti

```
1  if berhasil:
2      #ADMIN
3      if level_akses == "admin":
4          while True:
5              os.system("cls || clear")
6
7              print("=" * 60)
8              print("  MENU ADMIN ".center(60))
9              print("=" * 60)
10
11             tabel_admin = PrettyTable()
12             tabel_admin.field_names = ["No", "Pilihan Menu"]
13             tabel_admin.align = "l"
14             tabel_admin.hrules = 1
15             tabel_admin.add_row(["1", "Lihat Semua Properti"])
16             tabel_admin.add_row(["2", "Tambah Properti"])
17             tabel_admin.add_row(["3", "Update Properti"])
18             tabel_admin.add_row(["4", "Hapus Properti"])
19             tabel_admin.add_row(["5", "Logout"])
20
21             print(tabel_admin)
22
23             opsi_admin = input("Pilih menu: ")
```

```
1  # USER
2      elif level_akses == "user":
3          while True:
4              os.system("cls")
5              print("=== MENU USER ===")
6
7              tabel_user = PrettyTable()
8              tabel_user.field_names = ["No", "Pilihan Menu"]
9              tabel_user.align = "l"
10             tabel_user.hrules = 1
11
12             tabel_user.add_row(["1", "Lihat Semua Properti"])
13             tabel_user.add_row(["2", "Logout"])
14             print(tabel_user)
15
16             pilih_user = input("Pilih menu: ")
```

## E. Fitur Menu Admin

Menampilkan 5 opsi utama:

1. Lihat Properti (Read)
2. Tambah Properti (Create)
3. Update Properti (Update)
4. Hapus Properti (Delete)
5. Logout



```

1  if berhasil:
2      #ADMIN
3      if level_akses == "admin":
4          while True:
5              os.system("cls || clear")
6
7              print("-" * 60)
8              print("  MENU ADMIN ".center(60))
9              print("-" * 60)
10
11             tabel_admin = PrettyTable()
12             tabel_admin.field_names = ["No", "Pilihan Menu"]
13             tabel_admin.align = "l"
14             tabel_admin.hrules = 1
15             tabel_admin.add_row(["1", "Lihat Semua Properti"])
16             tabel_admin.add_row(["2", "Tambah Properti"])
17             tabel_admin.add_row(["3", "Update Properti"])
18             tabel_admin.add_row(["4", "Hapus Properti"])
19             tabel_admin.add_row(["5", "Logout"])
20
21             print(tabel_admin)
22
23             opsi_admin = input("Pilih menu: ")
24
25             # Read
26             if opsi_admin == "1":
27                 os.system("cls")
28                 print("=== Daftar Properti ===")
29                 if len(daftar_properti) == 0:
30                     print("Belum ada Properti")
31                 else:
32                     tabel = PrettyTable()
33                     tabel.field_names = ["No", "ID", "Nama", "Lokasi", "Tipe Listing", "Harga (Rp)"]
34                     tabel.hrules = 1
35                     for i in range(len(daftar_properti)):
36                         id, nama, lokasi, tipe_listing, harga = daftar_properti[i]
37                         tabel.add_row([i+1, id, nama, lokasi, tipe_listing, f"{harga:,}"])
38                     print(tabel)
39                     input("\nTekan Enter untuk kembali...")
40
41             # Create
42             elif opsi_admin == "2":
43                 os.system("cls")
44                 print("=== TAMBAH PROPERTI ===")
45                 id = f"R{len(daftar_properti) + 1:03d}" # otomatis buat id
46                 nama = input("Nama Properti: ")
47                 lokasi = input("Lokasi: ")
48                 tipe_listing = input("Tipe Listing: ")
49                 harga = input("Harga (angka): ")
50
51                 if harga.isdigit():
52                     daftar_properti.append([id, nama, lokasi, tipe_listing, int(harga)])
53                     print("Properti berhasil ditambahkan!")
54                 else:
55                     print("Harga harus berupa angka!")
56
57                 input("Tekan Enter untuk kembali...")
58
59             # Update
60             elif opsi_admin == "3":
61                 os.system("cls || clear")
62                 print("=== UPDATE DATA PROPERTI ===")
63                 tabel_update = PrettyTable()
64                 tabel_update.field_names = ["No", "ID", "Nama Properti", "Lokasi", "Tipe Listing", "Harga (Rp)"]
65                 tabel_update.align = "l"
66                 tabel_update.hrules = 1
67
68                 for i in range(len(daftar_properti)):
69                     id, nama, lokasi, tipe_listing, harga = daftar_properti[i]
70                     tabel_update.add_row([i+1, id, nama, lokasi, tipe_listing, f"{harga:,}"])
71                 print(tabel_update)
72
73                 pilih = input("Nomor properti yang ingin diupdate: ")
74
75                 if pilih.isdigit() and 1 <= int(pilih) <= len(daftar_properti):
76                     id_baru = int(pilih) - 1
77                     id, nama_lama, lokasi_lama, tipe_listing_lama, harga_lama = daftar_properti[id_baru]
78                     print("Kosongkan jika tidak ingin mengubah.")
79                     nama_baru = input(f"Nama baru ({nama_lama}): ")
80                     lokasi_baru = input(f"Lokasi baru ({lokasi_lama}): ")
81                     tipe_listing_baru = input(f"Tipe listing baru ({tipe_listing_lama}): ")
82                     harga_baru = input(f"Harga baru ({harga_lama}): ")
83
84                     if nama_baru != "":
85                         daftar_properti[id_baru][1] = nama_baru
86                     if lokasi_baru != "":
87                         daftar_properti[id_baru][2] = lokasi_baru
88                     if tipe_listing_baru != "":
89                         daftar_properti[id_baru][3] = tipe_listing_baru
90                     if harga_baru != "":
91                         if harga_baru.isdigit():
92                             daftar_properti[id_baru][4] = int(harga_baru)
93                         else:
94                             print("Harga harus angka, tidak diubah.")
95                     print("Data berhasil diperbarui.")
96                 else:
97                     print("Nomor tidak valid!")
98                     input("Tekan Enter untuk kembali...")
99
100            # Delete
101            elif opsi_admin == "4":
102                os.system("cls")
103                print("=== HAPUS PROPERTI ===")
104                tabel_hapus = PrettyTable()
105                tabel_hapus.field_names = ["No", "ID", "Nama Properti", "Lokasi", "Tipe Listing", "Harga (Rp)"]
106                tabel_hapus.align = "l"
107                tabel_hapus.hrules = 1
108
109                for i in range(len(daftar_properti)):
110                    id, nama, lokasi, tipe_listing, harga = daftar_properti[i]
111                    tabel_hapus.add_row([i+1, id, nama, lokasi, tipe_listing, f"{harga:,}"])
112                print(tabel_hapus)
113
114                pilih = input("Nomor properti yang ingin dihapus: ")
115
116                if pilih.isdigit() and 1 <= int(pilih) <= len(daftar_properti):
117                    del daftar_properti[pilih-1]
118                    print("Data berhasil dihapus.")
119                else:
120                    print("Nomor tidak valid!")
121                    input("Tekan Enter untuk kembali...")
122
123            # Logout
124            elif opsi_admin == "5":
125                break
126            else:
127                print("Pilihan tidak valid!")
128                input("Tekan Enter untuk lanjut...")

```

## F. Fitur Menu User

Menampilkan 2 opsi utama:

1. Lihat semua Properti (Menampilkan daftar properti dalam format tabel **Read only**).
2. Logout (kembali ke menu utama).

```
1 elif level_akses == "user":
2     while True:
3         os.system("cls")
4         print("=== MENU USER ===")
5
6         tabel_user = PrettyTable()
7         tabel_user.field_names = ["No", "Pilihan Menu"]
8         tabel_user.align = "l"
9         tabel_user.hrules = 1
10
11         tabel_user.add_row(["1", "Lihat Semua Properti"])
12         tabel_user.add_row(["2", "Logout"])
13         print(tabel_user)
14
15         pilih_user = input("Pilih menu: ")
16
17         if pilih_user == "1":
18             os.system("cls")
19             print("=== DAFTAR PROPERTI ===")
20             if len(daftar_properti) == 0:
21                 print("Belum ada properti.")
22             else:
23                 tabel_user_properti = PrettyTable()
24                 tabel_user_properti.field_names = ["No", "ID", "Nama Properti", "Lokasi", "Tipe Listing", "Harga (Rp)"]
25                 tabel_user_properti.align = "l"
26                 tabel_user_properti.hrules = 1
27
28                 for i in range(len(daftar_properti)):
29                     id, nama, lokasi, tipe_listing, harga = daftar_properti[i]
30                     tabel_user_properti.add_row([i+1, id, nama, lokasi, tipe_listing, f"{harga:,}"])
31                 print(tabel_user_properti)
32                 input("\nTekan Enter untuk kembali...")
33
34         elif pilih_user == "2":
35             break
```

#### 4. Hasil Output

```
🏠 SISTEM MANAJEMEN LISTING PROPERTI 🏠.
+-----+-----+
| No | 🕒 Menu Utama |
+-----+-----+
| 1 | 🔑 Login      |
+-----+-----+
| 2 | 📄 Register   |
+-----+-----+
| 3 | 🚪 Keluar     |
+-----+-----+
Pilih Menu : █
```

Gambar 1. Menu utama

```
=====
                        MENU ADMIN
=====
+-----+-----+
| No | Pilihan Menu      |
+-----+-----+
| 1 | Lihat Semua Properti |
+-----+-----+
| 2 | Tambah Properti     |
+-----+-----+
| 3 | Update Properti     |
+-----+-----+
| 4 | Hapus Properti      |
+-----+-----+
| 5 | Logout              |
+-----+-----+
Pilih menu: █
```

Gambar 2. Menu Admin

```
=== Daftar Properti ===
+-----+-----+-----+-----+-----+-----+
| No | ID | Nama | Lokasi | Tipe Listing | Harga (Rp) |
+-----+-----+-----+-----+-----+-----+
| 1 | R001 | Krusty Krab | Bikini Bottom | Restoran | 5,000,000 |
+-----+-----+-----+-----+-----+-----+
| 2 | R002 | Rumah Upin & Ipin | Kampung Durian Runtuh | Rumah | 350,000 |
+-----+-----+-----+-----+-----+-----+
| 3 | R003 | Superlab Gustavo Fring | Albuquerque, New Mexico | Laboratorium | 12,000,000 |
+-----+-----+-----+-----+-----+-----+
Tekan Enter untuk kembali...
```

Gambar 3. Read

```
=== TAMBAH PROPERTI ===
Nama Properti: Tok dalang homestay
Lokasi: durian runtuh
Tipe listing: disewakan
Harga (angka): 5000000
Properti berhasil ditambahkan!
Tekan Enter untuk kembali...
```

Gambar 4. Create

```
=== UPDATE DATA PROPERTI ===
+-----+-----+-----+-----+-----+-----+
| No | ID | Nama Properti | Lokasi | Tipe Listing | Harga (Rp) |
+-----+-----+-----+-----+-----+-----+
| 1 | R001 | Krusty Krab | Bikini Bottom | Restoran | 5,000,000 |
+-----+-----+-----+-----+-----+-----+
| 2 | R002 | Rumah Upin & Ipin | Kampung Durian Runtuh | Rumah | 350,000 |
+-----+-----+-----+-----+-----+-----+
| 3 | R003 | Superlab Gustavo Fring | Albuquerque, New Mexico | Laboratorium | 12,000,000 |
+-----+-----+-----+-----+-----+-----+
| 4 | R004 | 2 | 2 | 2 | 2 |
+-----+-----+-----+-----+-----+-----+
Nomor properti yang ingin diupdate: 
```

Gambar 5. Update

```
=== HAPUS PROPERTI ===
+-----+-----+-----+-----+-----+-----+
| No | ID  | Nama Properti      | Lokasi              | Tipe Listing | Harga (Rp) |
+-----+-----+-----+-----+-----+-----+
| 1  | R001 | Krusty Krab        | Bikini Bottom       | Restoran    | 5,000,000  |
+-----+-----+-----+-----+-----+-----+
| 2  | R002 | Rumah Upin & Ipin  | Kampung Durian Runtuh | Rumah       | 350,000    |
+-----+-----+-----+-----+-----+-----+
| 3  | R003 | Superlab Gustavo Fring | Albuquerque, New Mexico | Laboratorium | 12,000,000 |
+-----+-----+-----+-----+-----+-----+
| 4  | R004 | 2                  | 2                   | 2           | 2          |
+-----+-----+-----+-----+-----+-----+
Nomor properti yang ingin dihapus: 
```

Gambar 6. Delete

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
=== REGISTER PENGGUNA BARU ===
Masukkan username: randy
Masukkan Password: *****
Registrasi berhasil! Silakan login.
Tekan Enter untuk lanjut... 
```

Gambar 7. Registrasi

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
🏠 SISTEM MANAJEMEN LISTING PROPERTI 🏠
+---+-----+
| No | 🕒 Menu Utama |
+---+-----+
| 1  | 🔑 Login      |
+---+-----+
| 2  | 📄 Register   |
+---+-----+
| 3  | 🚪 Keluar     |
+---+-----+
Pilih Menu : 3
Terima kasih telah menggunakan Program ini!
```

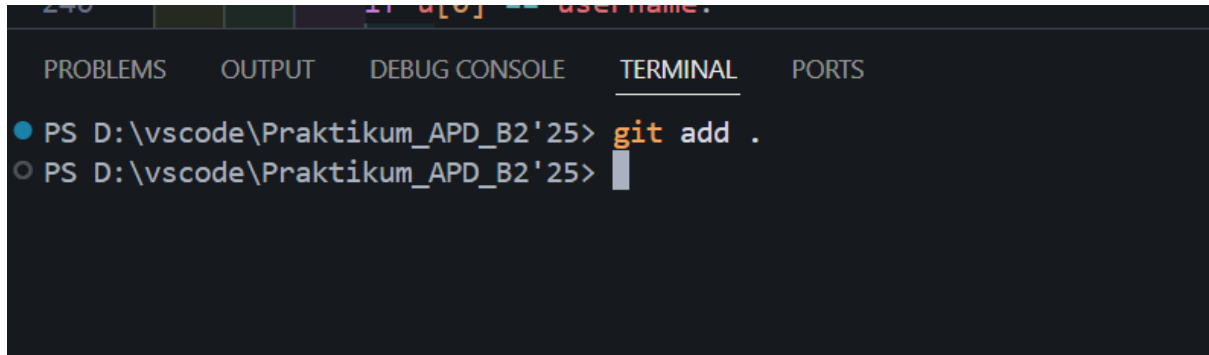
Gambar 8. Keluar dari program

```
=== LOGIN ===
Masukkan Username: renfsefe
Masukkan Password: **
Username atau password salah! Sisa percobaan: 2
Masukkan Username: efefsfeefg
Masukkan Password: *****
Username atau password salah! Sisa percobaan: 1
Masukkan Username: egegge
Masukkan Password: *****
Username atau password salah 3 kali. Silakan tunggu 5 menit sebelum mencoba lagi.
```

Gambar 9. Percobaan 3x gagal

## 5. Langkah-langkah GIT

### 5.1 GIT Add

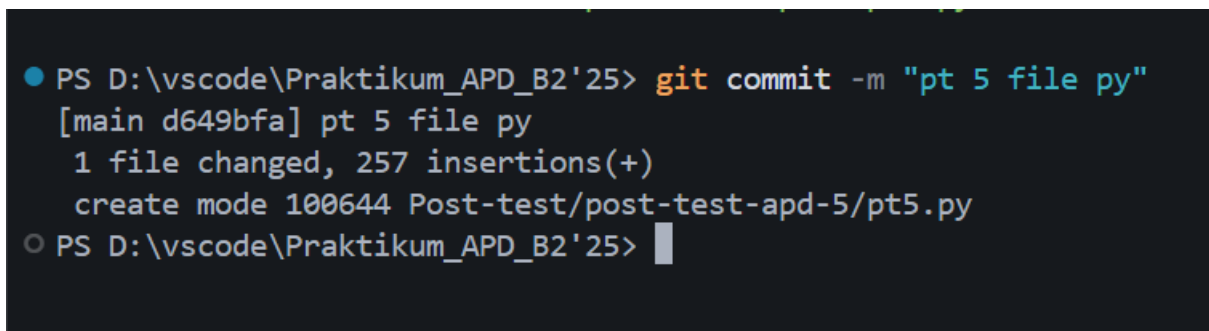


```
240 17 u[0] -- username.  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
● PS D:\vscode\Praktikum_APD_B2'25> git add .  
○ PS D:\vscode\Praktikum_APD_B2'25> █
```

Menambahkan semua perubahan file di folder kerja (working directory) ke staging area agar siap dikomit.

Titik . artinya semua file yang berubah akan ditambahkan.

### 5.2 GIT Commit



```
● PS D:\vscode\Praktikum_APD_B2'25> git commit -m "pt 5 file py"  
[main d649bfa] pt 5 file py  
1 file changed, 257 insertions(+)  
create mode 100644 Post-test/post-test-apd-5/pt5.py  
○ PS D:\vscode\Praktikum_APD_B2'25> █
```

Menyimpan perubahan yang sudah di-*add* ke dalam riwayat Git (repository lokal).

## 5.3 GIT Push

```
● PS D:\vscode\Praktikum_APD_B2'25> git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 20 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 2.60 KiB | 2.60 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/r3nddy/Praktikum_APD_B2-25.git
    313b4d6..d649bfa  main -> main
○ PS D:\vscode\Praktikum_APD_B2'25> █
```

Mengirim (mendorong) commit dari repository lokal ke repository remote GitHub.  
Dengan kata lain, perubahanmu diunggah ke server.