

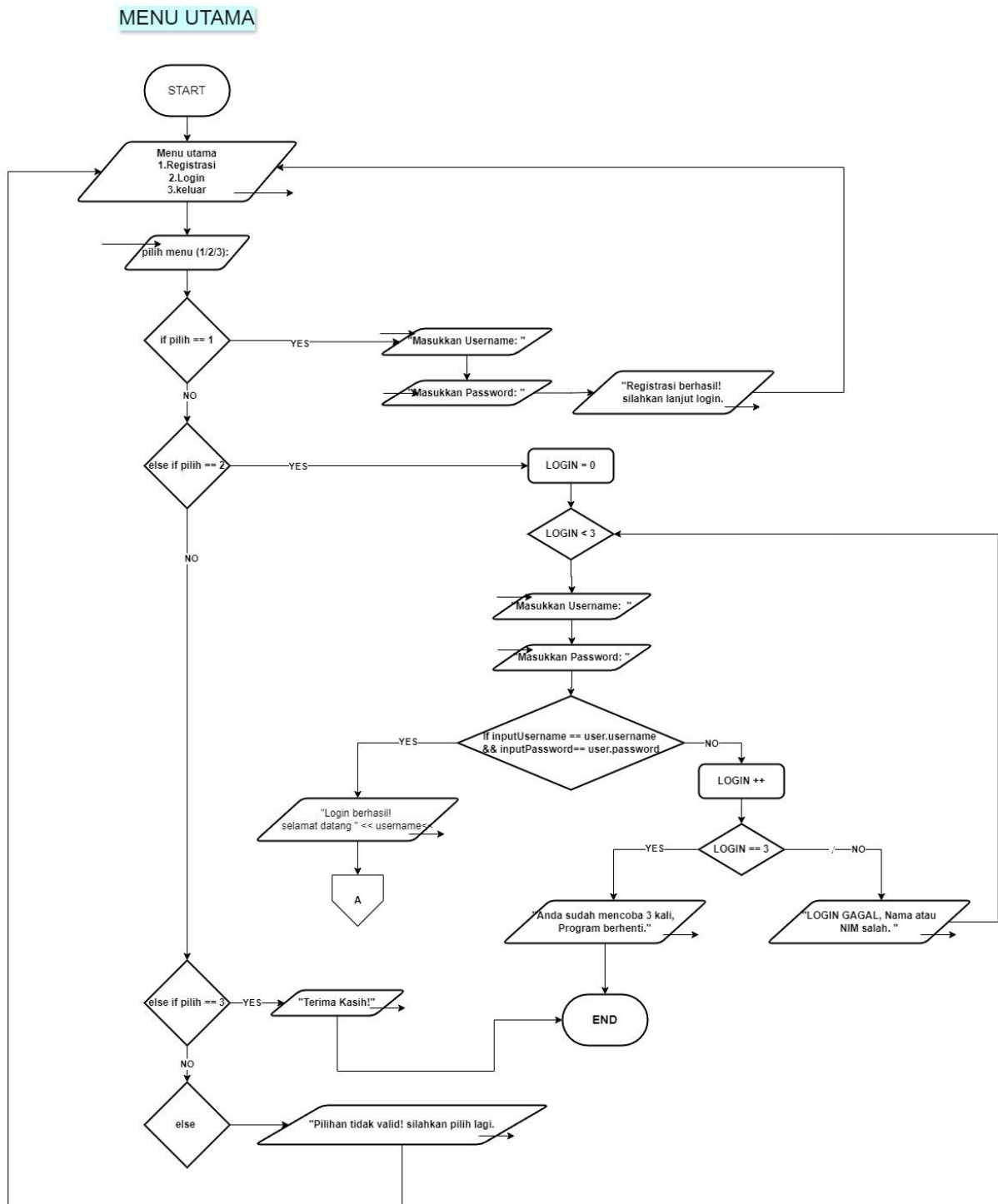
**LAPORAN PRAKTIKUM**  
**POSTTEST 4**  
**ALGORITMA PEMROGRAMAN LANJUT**



**Disusun oleh:**  
**Nadia Rahmah (2409106018)**  
**Kelas (A1 '24)**

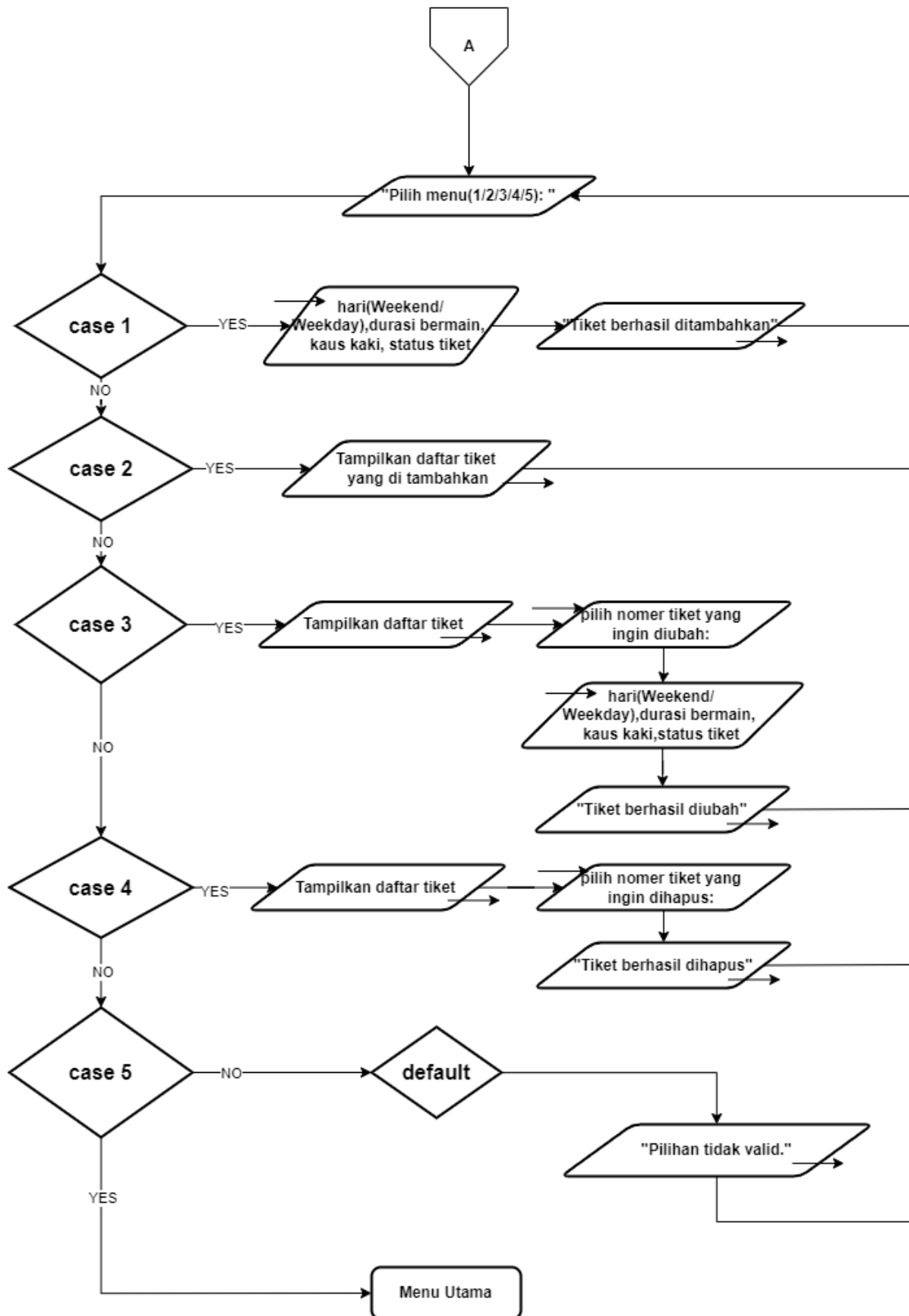
**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

## 1. Flowchart



Gambar 1.1 Flowchart Menu Utama

## MENU TIKET



Gambar 1.2 Flowchart Menu Tiket

## 2. Analisis Program

### 2.1 Deskripsi Singkat Program

Program ini dibuat menggunakan Bahasa pemrograman C++ dengan judul “Manajemen pembelian tiket *Mini Playground*” untuk memudahkan pengelolaan data tiket. Struktur data pada program ini menggunakan struct Tiket, struct User, array tiket dan array user. Program juga sudah dilengkapi array of struct (Tiket daftarTiket[MaxTiket]). Masing-masing fitur yang ada pada menu program dipisahkan dari program utama menjadi subprogram menggunakan fungsi dan prosedur. Fitur tersebut diantaranya Registrasi, Login, Tambah tiket, Tampilkan tiket, Ubah tiket, dan Hapus tiket. Pada program ini juga dilengkapi proses keamanan login untuk memastikan hanya pengguna yang sudah ditetapkan yang bisa mengakses program dan juga membatasi jumlah percobaan login sebanyak 3 kali.

## 3. Source Code

### A. MENU UTAMA

Pada menu utama terdapat tiga pilihan menu yaitu menu Registrasi, menu login, dan menu keluar. Terdapat fungsi Registrasi dan fungsi Login, Program akan memanggil fungsi yang sesuai dengan pilihan user. Menu keluar untuk mengakhiri program karena program ini menggunakan loop yang membuat program akan terus berulang.

#### Source Code:

```
void menuUtama(){
    User user;
    while (true){
        cout << "=====" << endl;
        cout << "|      MENU UTAMA      |" << endl;
        cout << "=====" << endl;
        cout << "1. Registrasi\n";
        cout << "2. Login\n";
        cout << "3. Keluar\n";
        cout << "Pilih Menu (1/2/3): ";
        int pilih;
        cin >> pilih;
```

```

        if (pilih == 1){
            registrasi(user);
        }else if(pilih == 2){
            if (login(user)){
                menuTiket();
            }
        } else if (pilih == 3) {
            cout << "Terima Kasih!" << endl;
            break;
        } else {
            cout << "Pilihan tidak valid! Silakan pilih lagi." << endl;
        }
    }
}

```

## B. Fitur Registrasi

Fitur Registrasi digunakan untuk mendaftarkan akun baru dengan membuat username dan password. Data akun user baru akan disimpan dalam array of struct sehingga program ini dapat multiuser.

### Source Code:

```

void registrasi(User &user) {
    cout << "=== Registrasi ===" << endl;
    cout << "Masukkan Username: ";
    cin.ignore();
    getline(cin, daftarUser[jumlahUser].username);
    cout << "Masukkan Password: ";
    getline(cin, daftarUser[jumlahUser].password);
    cout << "Registrasi berhasil! Silakan lanjut login." << endl;
    jumlahUser++;
}

```

## C. Fitur Login

Fitur Login digunakan untuk mengamankan akses program, memastikan hanya pengguna yang sudah memiliki akun yang dapat login. Fitur ini juga membatasi percobaan login untuk memberi kesempatan untuk memasukkan *input*-an yang benar.

### Source Code:

```

bool login(User &user) {

```

```

string inputUsername, inputPassword;
int login = 0;
while (login < 3) {
    cout << "Masukkan Username: ";
    cin >> ws;
    getline(cin, inputUsername);
    cout << "Masukkan Password: ";
    cin >> ws;
    getline(cin, inputPassword);

    for (int i = 0; i < jumlahUser; i++) {
        if (inputUsername == daftarUser[i].username && inputPassword ==
daftarUser[i].password) {
            user = daftarUser[i]; // menyimpan user aktif
            cout << "Login berhasil! Selamat datang " << user.username <<
"!" << endl;
            return true;
        }
    }
    login++;
    if (login < 3) {
        cout << "Username atau password salah! Coba lagi" << endl;
    }
}
cout << "Anda sudah mencoba 3 Kali. Program berhenti." << endl;
exit(0);
}

```

#### D. Menu CRUD Tiket

Fitur ini memiliki beberapa menu memudahkan dalam pengelolaan tiket. Menu dapat diakses dengan melakukan pemanggilan fungsi.

##### Source Code:

```

void menuTiket() {
    while (true) {
        cout << "=====" << endl;
        cout << "|          MENU TIKET          |" << endl;
        cout << "=====" << endl;
        cout << "1. Tambah Tiket" << endl;
        cout << "2. Tampilkan Tiket" << endl;
        cout << "3. Ubah data Tiket" << endl;
        cout << "4. Hapus Tiket" << endl;
        cout << "5. Keluar" << endl;
        cout << "Pilih menu (1/2/3/4/5): ";
        int pilihan;
    }
}

```

```

        cin >> pilihan;
        switch (pilihan) {
            case 1:
                tambahTiket();
                break;
            case 2:
                tampilkanTiket();
                break;
            case 3:
                ubahTiket();
                break;
            case 4:
                hapusTiket();
                break;
            case 5:
                return;
            default:
                cout << "Pilihan tidak valid" << endl;
        }
    }
}

```

### C. Fitur Tambah Tiket

Fitur ini digunakan untuk menambahkan data baru ke dalam Array. Data ditambahkan dengan input hari(Weekend/Weekday), durasi bermain, apakah membawa Kaus kaki(Y/T) dan menghitung total harga tiket berdasarkan data tersebut. Data status tiket juga ditambahkan dengan bersifat default “Aktif”.

#### Source Code:

```

void tambahTiket() {
    if (jumlahTiket < MaxTiket) {
        Tiket tiketBaru;
        int inputHari, inputDurasi;
        char inputKausKaki;

        cout << "1. Weekend (+ 5.000)" << endl;
        cout << "2. Weekday" << endl;
        cout << "Masukkan Hari (1/2): ";
        cin >> inputHari;
        tiketBaru.hari = (inputHari == 1) ? "Weekend" : "Weekday";

        cout << "1. 1 Jam 20.000" << endl;
        cout << "2. 2 Jam 35.000" << endl;
        cout << "3. Sepuasnya 45.000" << endl;
    }
}

```

```

        cout << "Masukkan Durasi (1/2/3): ";
        cin >> inputDurasi;
        if (inputDurasi == 1) {
            tiketBaru.durasi = "1 Jam";
            tiketBaru.harga = 20000;
        } else if (inputDurasi == 2) {
            tiketBaru.durasi = "2 Jam";
            tiketBaru.harga = 35000;
        } else {
            tiketBaru.durasi = "Sepuasnya";
            tiketBaru.harga = 45000;
        }

        cout << "***Jika tidak membawa kaus kaki, maka diwajibkan membeli
seharga 10.000 ***" << endl;
        cout << "Apakah Anda membawa kaus kaki? (Y untuk Ya, T untuk Tidak):
";
        cin >> inputKausKaki;
        tiketBaru.kausKaki = (inputKausKaki == 'Y' || inputKausKaki == 'y');
        tiketBaru.totalHarga = tiketBaru.harga + ((inputHari == 1) ? 5000 : 0)
+ (tiketBaru.kausKaki ? 0 : 10000);
        tiketBaru.status = "Aktif";
        daftarTiket[jumlahTiket++] = tiketBaru;
        cout << "Tiket berhasil ditambahkan!" << endl;
    } else {
        cout << "Jumlah tiket sudah mencapai batas maksimum!" << endl;
    }
}

```

#### D. Fitur Tampilkan Tiket

Fitur ini digunakan untuk menampilkan semua daftar tiket yang sudah ditambahkan.

##### Source Code:

```

void tampilkanTiket() {
    if (jumlahTiket == 0) {
        cout << "Tidak ada tiket tersedia" << endl;
    } else {
        cout << setw(10) << "No" << setw(15) << "Hari" << setw(15) << "Durasi"
<< setw(10) << "Harga" << setw(15) << "Kaus Kaki" << setw(15) << "Total Harga"
<< setw(15) << "Status" << endl;
        for (int i = 0; i < jumlahTiket; i++) {
            cout << setw(10) << (i + 1)
                << setw(15) << daftarTiket[i].hari
                << setw(15) << daftarTiket[i].durasi
                << setw(10) << daftarTiket[i].harga

```



```

                << setw(15) << (daftarTiket[i].kausKaki ? "Ya" : "Tidak")
                << setw(15) << daftarTiket[i].totalHarga
                << setw(15) << daftarTiket[i].status << endl;
            }
        }
    }
}

```

## E. Fitur Ubah data Tiket

Fitur ini digunakan untuk mengubah data tiket yang sudah ada, dengan menampilkan daftar tiket terlebih dahulu agar memudahkan user memilih nomor tiket yang ingin diubah. Setelah itu mengubah pilihan hari, durasi, kaus kaki, status tiket dan total harga.

### Source Code:

```

void ubahTiket() {
    tampilkanTiket();
    int no;
    cout << "Masukkan nomor tiket yang ingin diubah: ";
    cin >> no;

    if (no < 1 || no > jumlahTiket) {
        cout << "Nomor tiket tidak valid!" << endl;
        return;
    }

    Tiket &tiket = daftarTiket[no - 1];
    int inputHari, inputDurasi;
    char inputKausKaki;

    cout << "1. Weekend" << endl;
    cout << "2. Weekday" << endl;
    cout << "Masukkan Hari baru (1/2): ";
    cin >> inputHari;
    tiket.hari = (inputHari == 1) ? "Weekend" : "Weekday";

    cout << "1. 1 Jam 20.000" << endl;
    cout << "2. 2 Jam 35.000" << endl;
    cout << "3. Sepuasnya 45.000" << endl;
    cout << "Masukkan Durasi baru (1/2/3): ";
    cin >> inputDurasi;
    if (inputDurasi == 1) {
        tiket.durasi = "1 Jam";
        tiket.harga = 20000;
    } else if (inputDurasi == 2) {
        tiket.durasi = "2 Jam";
        tiket.harga = 35000;
    }
}

```

```

    } else {
        tiket.durasi = "Sepuasnya";
        tiket.harga = 45000;
    }

    cout << "Apakah Anda membawa kaus kaki? (Y untuk Ya, T untuk Tidak): ";
    cin >> inputKausKaki;
    tiket.kausKaki = (inputKausKaki == 'Y' || inputKausKaki == 'y');
    tiket.totalHarga = tiket.harga + ((inputHari == 1) ? 5000 : 0) +
(tiket.kausKaki ? 0 : 10000);

    cout << "Masukkan status baru tiket: ";
    cin.ignore();
    getline(cin, tiket.status);

    cout << "Tiket berhasil diubah!" << endl;
}

```

## F. Fitur Hapus Tiket

Fitur ini berfungsi untuk menghapus Tiket dengan menampilkan daftar tiket terlebih dahulu untuk memudahkan user memilih menghapus tiket berdasarkan input nomer tiket. Setelah penghapusan, Tiket yang tersisa akan digeser.

### Source Code:

```

void hapusTiket() {
    tampilkanTiket();
    int no;
    cout << "Masukkan nomor tiket yang ingin dihapus: ";
    cin >> no;

    if (no < 1 || no > jumlahTiket) {
        cout << "Nomor tiket tidak valid!" << endl;
        return;
    }

    for (int i = no - 1; i < jumlahTiket - 1; i++) {
        daftarTiket[i] = daftarTiket[i + 1]; // Menggeser tiket ke kiri
    }
    jumlahTiket--; // Mengurangi jumlah tiket
    cout << "Tiket berhasil dihapus!" << endl;
}

```

## E. Keluar dari Program

Penggunaan Loop yang digunakan pada menu tiket memungkinkan untuk terus meminta input sehingga fitur ini berfungsi agar program keluar dari menu tiket dahulu lalu pilih menu 3 pada menu utama untuk keluar program. fitur ini berguna agar dapat berhenti sesuai keinginan dan kebutuhan pengguna.

#### Source Code:

```
int main() {  
    cout << "=====  
<< endl;  
    cout << "| SELAMAT DATANG DI APLIKASI PEMBELIAN TIKET MINI PLAYGROUND |"  
<< endl;  
    cout << "=====  
<< endl;  
    cout << " Silakan registrasi atau login untuk mulai menggunakan\n";  
    menuUtama();  
    return 0;  
}
```

## 4. Uji Coba dan Hasil Output

### 4.1 Uji Coba dan Hasil Output

```
=====
| SELAMAT DATANG DI APLIKASI PEMBELIAN TIKET MINI PLAYGROUND |
=====
silakan registrasi atau login untuk mulai menggunakan
=====
|      MENU UTAMA      |
=====
1. Registrasi
2. Login
3. Keluar
Pilih Menu (1/2/3): 1
=== Registrasi ===
Masukkan Username: NADIA
Masukkan Password: NAD1A
Registrasi berhasil! Silakan lanjut login.
```

Gambar 4.1 Registrasi

```
=====
|      MENU UTAMA      |
=====
1. Registrasi
2. Login
3. Keluar
Pilih Menu (1/2/3): 2
Masukkan Username: GAGAL
Masukkan Password: GAGAL
Username atau password salah! Coba lagi
Masukkan Username: GAGAL
Masukkan Password: GAGAL
Username atau password salah! Coba lagi
Masukkan Username: GAGAL
Masukkan Password: GAGAL
Anda sudah mencoba 3 Kali. Program berhenti.
PS C:\pratikum-apl\post-test\post-test-4> █
```

Gambar 4.2 Login gagal

```

=====
|      MENU UTAMA      |
=====
1. Registrasi
2. Login
2. Login
3. Keluar
Pilih Menu (1/2/3): 2
Masukkan Username: Nadia Rahmah
Masukkan Password: 2409106018
Login berhasil! Selamat datang Nadia Rahmah!

```

Gambar 4.3 Login Berhasil

```

=====
|      MENU TIKET      |
=====
1. Tambah Tiket
2. Tampilkan Tiket
3. Ubah data Tiket
4. Hapus Tiket
5. Keluar
Pilih menu (1/2/3/4/5): 1
1. Weekend (+ 5.000)
2. Weekday
Masukkan Hari (1/2): 1
1. 1 Jam 20.000
2. 2 Jam 35.000
3. Sepuasnya 45.000
Masukkan Durasi (1/2/3): 1
***Jika tidak membawa kaus kaki, maka diwajibkan membeli seharga 10.000 ***
Apakah Anda membawa kaus kaki? (Y untuk Ya, T untuk Tidak): Y
Tiket berhasil ditambahkan!

```

Gambar 4.4 pilih Menu tiket 1

```

=====
|      MENU TIKET      |
=====
1. Tambah Tiket
2. Tampilkan Tiket
3. Ubah data Tiket
4. Hapus Tiket
5. Keluar
Pilih menu (1/2/3/4/5): 2

```

No	Hari	Durasi	Harga	Kaus Kaki	Total Harga	Status
1	Weekend	1 Jam	20000	Ya	25000	Aktif

```

=====

```

Gambar 4.5 pilih Menu tiket 2

```

=====
|      MENU TIKET      |
=====
1. Tambah Tiket
2. Tampilkan Tiket
3. Ubah data Tiket
4. Hapus Tiket
5. Keluar
Pilih menu (1/2/3/4/5): 3
  No      Hari      Durasi      Harga      Kaus Kaki      Total Harga      Status
    1      Weekend      1 Jam      20000      Ya      25000      Aktif
Masukkan nomor tiket yang ingin diubah: 1
1. Weekend
2. Weekday
Masukkan Hari baru (1/2): 1
1. 1 Jam 20.000
2. 2 Jam 35.000
3. Sepuasnya 45.000
Masukkan Durasi baru (1/2/3): 2
Apakah Anda membawa kaus kaki? (Y untuk Ya, T untuk Tidak): T
Masukkan status baru tiket: Non-Aktif
Tiket berhasil diubah!
=====
|      MENU TIKET      |
=====
1. Tambah Tiket
2. Tampilkan Tiket
3. Ubah data Tiket
4. Hapus Tiket
5. Keluar
Pilih menu (1/2/3/4/5): 2
  No      Hari      Durasi      Harga      Kaus Kaki      Total Harga      Status
    1      Weekend      2 Jam      35000      Tidak      50000      Non-Aktif
=====

```

Gambar 4.6 pilih menu tiket 3 dan 2

```

=====
|      MENU TIKET      |
=====
1. Tambah Tiket
2. Tampilkan Tiket
3. Ubah data Tiket
4. Hapus Tiket
5. Keluar
Pilih menu (1/2/3/4/5): 4
      No      Hari      Durasi      Harga      Kaus Kaki      Total Harga      Status
      1      Weekend      2 Jam      35000      Tidak      50000      Non-Aktif
Masukkan nomor tiket yang ingin dihapus: 1
Tiket berhasil dihapus!
=====
|      MENU TIKET      |
=====
1. Tambah Tiket
2. Tampilkan Tiket
3. Ubah data Tiket
4. Hapus Tiket
5. Keluar
Pilih menu (1/2/3/4/5): 2
Tidak ada tiket tersedia

```

Gambar 4.7 pilih menu tiket 4 dan 2

```

=====
|      MENU TIKET      |
=====
1. Tambah Tiket
2. Tampilkan Tiket
3. Ubah data Tiket
4. Hapus Tiket
5. Keluar
Pilih menu (1/2/3/4/5): 0
Pilihan tidak valid
=====
|      MENU TIKET      |
=====
1. Tambah Tiket
2. Tampilkan Tiket
3. Ubah data Tiket
4. Hapus Tiket
5. Keluar
Pilih menu (1/2/3/4/5): 9
Pilihan tidak valid
=====
|      MENU TIKET      |
=====
1. Tambah Tiket
2. Tampilkan Tiket
3. Ubah data Tiket
4. Hapus Tiket
5. Keluar
Pilih menu (1/2/3/4/5): -12
Pilihan tidak valid

```

Gambar 4.8 pilih menu tiket selain 1-5

```
=====
|      MENU TIKET      |
=====
1. Tambah Tiket
2. Tampilkan Tiket
3. Ubah data Tiket
4. Hapus Tiket
5. Keluar
Pilih menu (1/2/3/4/5): 5
=====
|      MENU UTAMA      |
=====
1. Registrasi
2. Login
3. Keluar
```

Gambar 4.9 pilih menu tiket 5

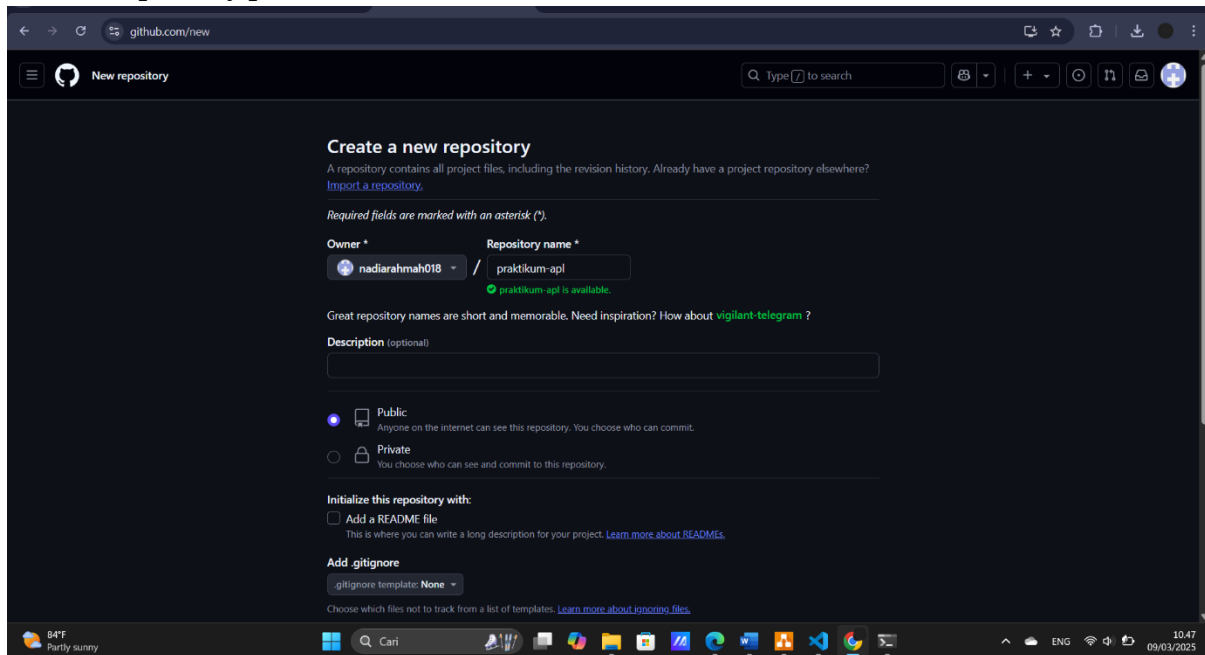
```
=====
|      MENU UTAMA      |
=====
1. Registrasi
2. Login
3. Keluar
Pilih Menu (1/2/3): 3
Terima Kasih!
PS C:\pratikum-apl\post-test\post-test-4>
```

Gambar 4.10 pilih menu utama 3



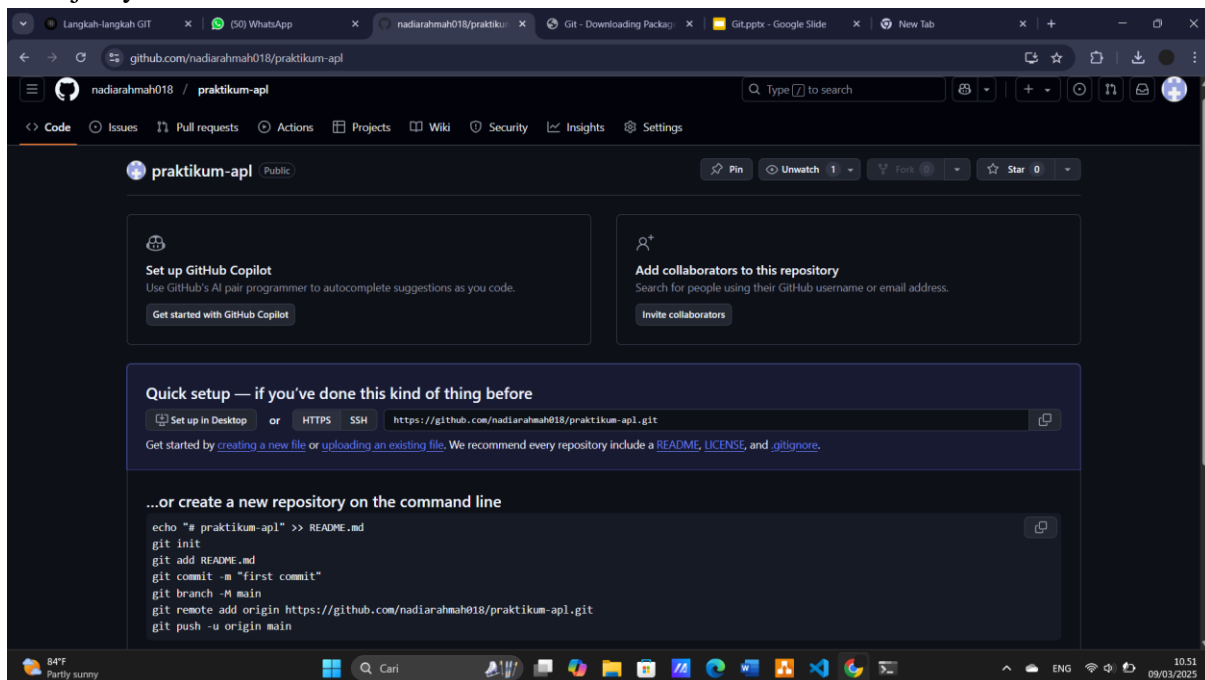
## 5. Langkah-Langkah GIT

### 1. Buat Repository pada Github



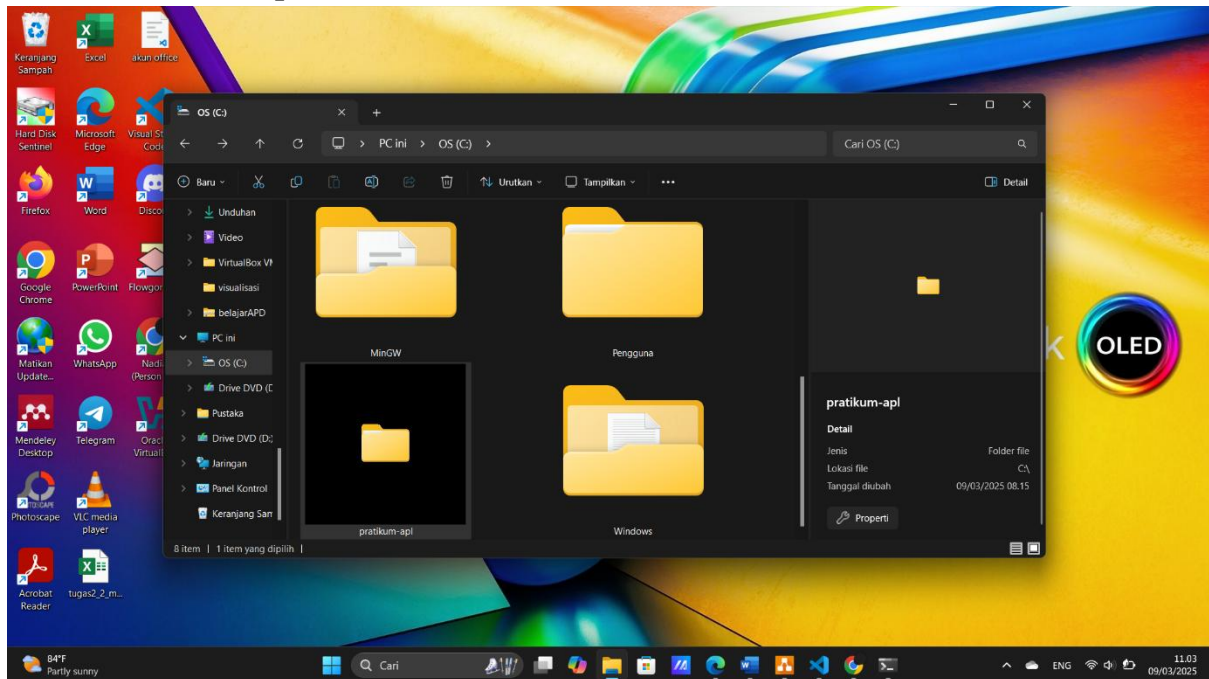
Gambar 5.1 Membuat Repository

Isi Nama repository yaitu praktikum-apl. Tetapkan Nama repository bersifat public. Kemudian penting memperhatikan Langkah-langkah command yang akan dilakukan di terminal. Tab Github di bawah ini jangan ditutup karena akan digunakan diproses selanjutnya.



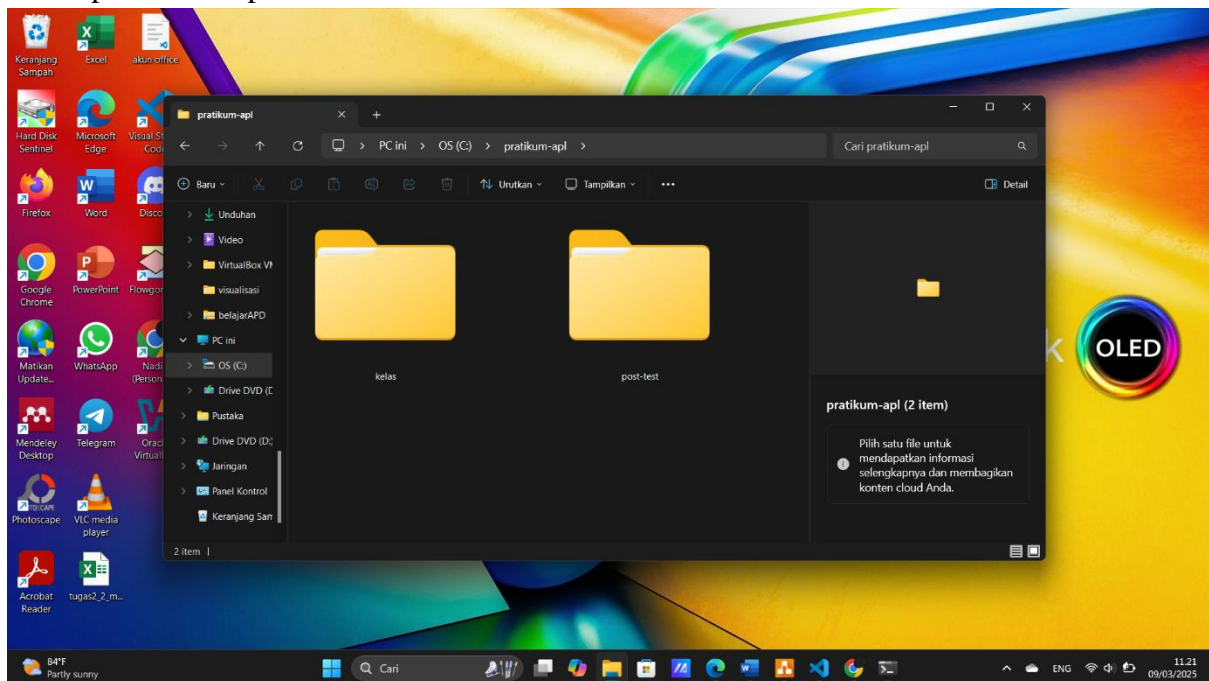
Gambar 5.2 Repository APL

## 2. Buat folder di Explorer



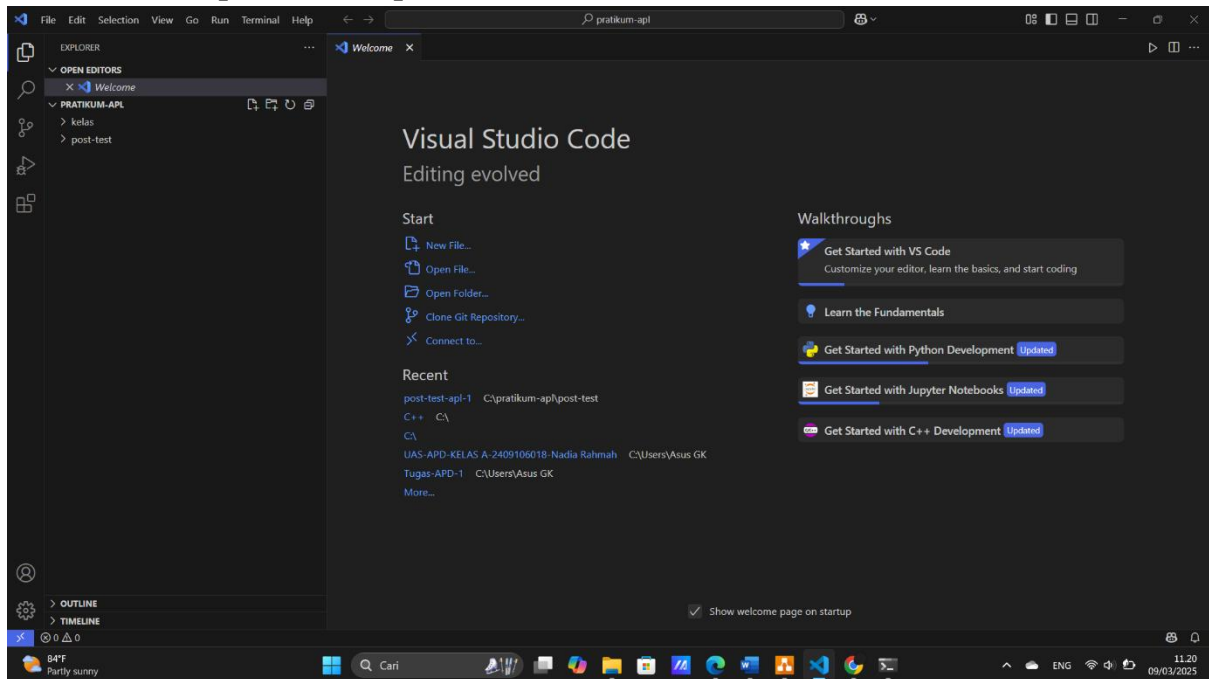
Gambar 5.3 Folder praktikum-apl

Buat folder pada explorer masing-masing dengan nama praktikum-apl sesuai nama repository Github. Selanjutnya tambahkan 2 folder lagi yang diberi nama kelas dan post-test di dalam folder praktikum-apl.



Gambar 5.4 Folder kelas dan post-test

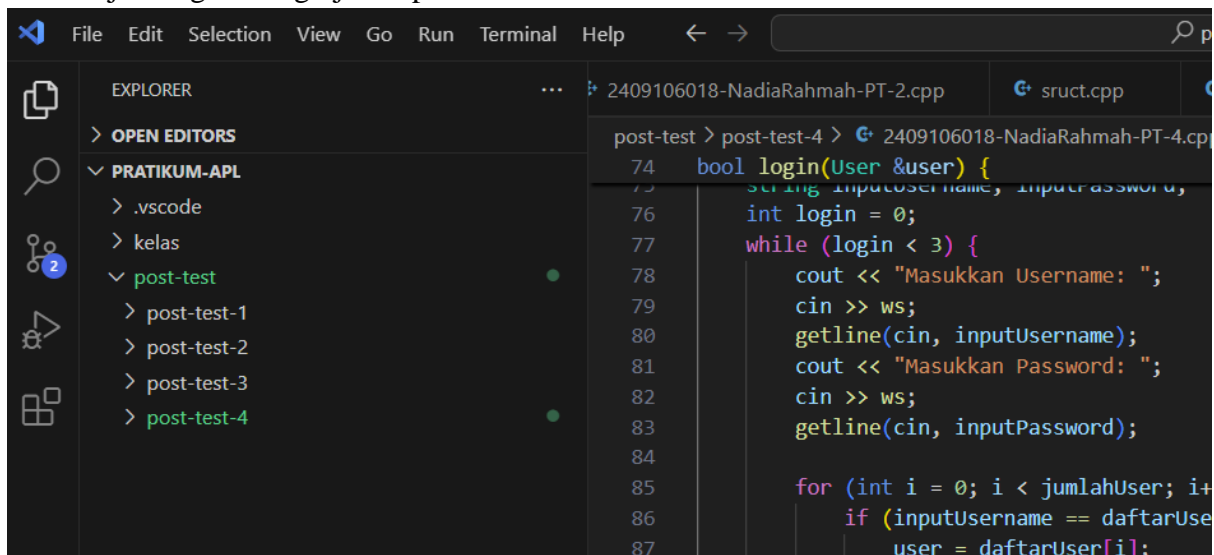
### 3. Buka Folder praktikum-apl di VSCode



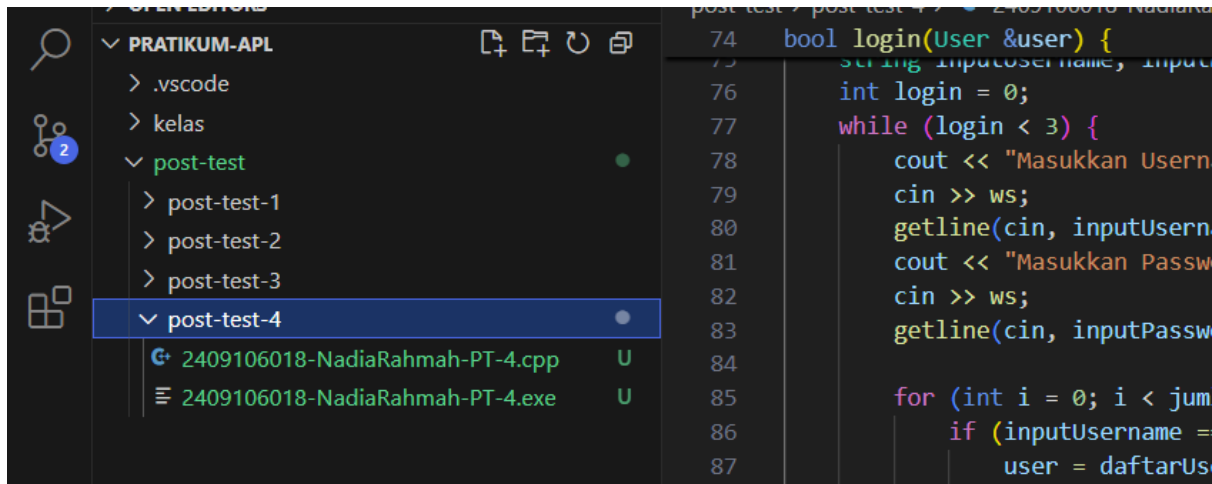
Gambar 5.5 VSCode

### 4. Buat Folder post-test-x di Folder post-test

Contoh jika ingin mengerjakan post-test-4.



Gambar 5.6 Folder post-test-4

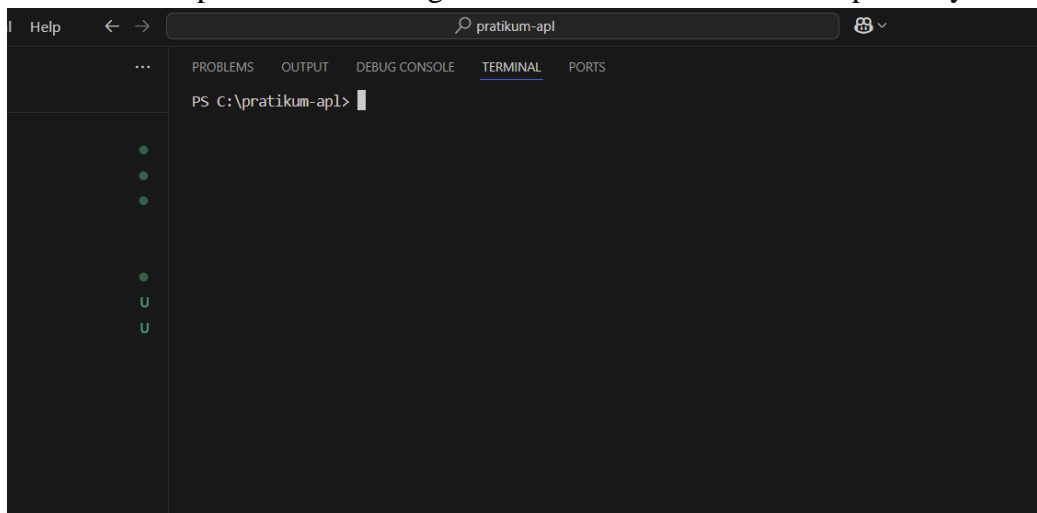


Gambar 5.7 File post-test-4

Kemudian pada folder post-test-4, buat file yang diperlukan dengan nama yang ditentukan.

## 5. Buka Terminal

Buka terminal pada VSCode dengan cara menekan tombol ctrl+~ pada keyboard

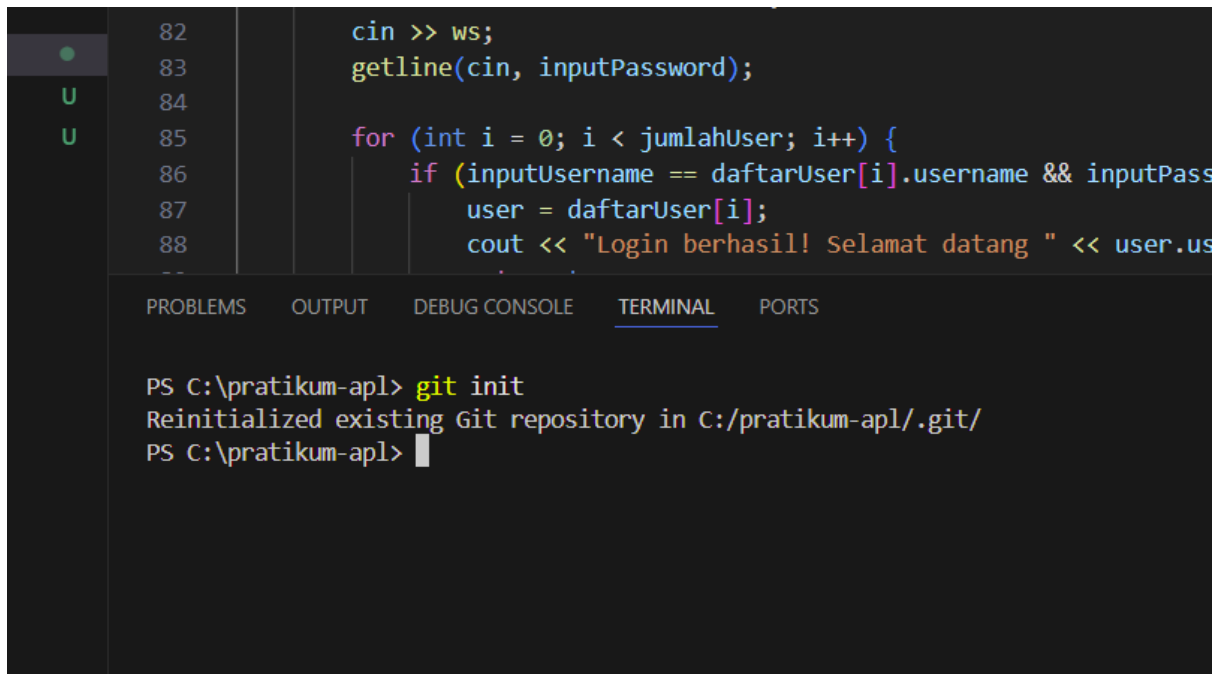


Gambar 5.8 Terminal VSCode

Pastikan path sesuai, jika path belum sesuai, naik folder paling atas dengan cara ketik (cd ../../ atau cd ..).

## 6. Git Init

Inisiasi repository dengan ketik perintah 'git init' pada terminal.



```
82      cin >> ws;
83      getline(cin, inputPassword);
84
85      for (int i = 0; i < jumlahUser; i++) {
86          if (inputUsername == daftarUser[i].username && inputPass
87              user = daftarUser[i];
88          cout << "Login berhasil! Selamat datang " << user.us
```

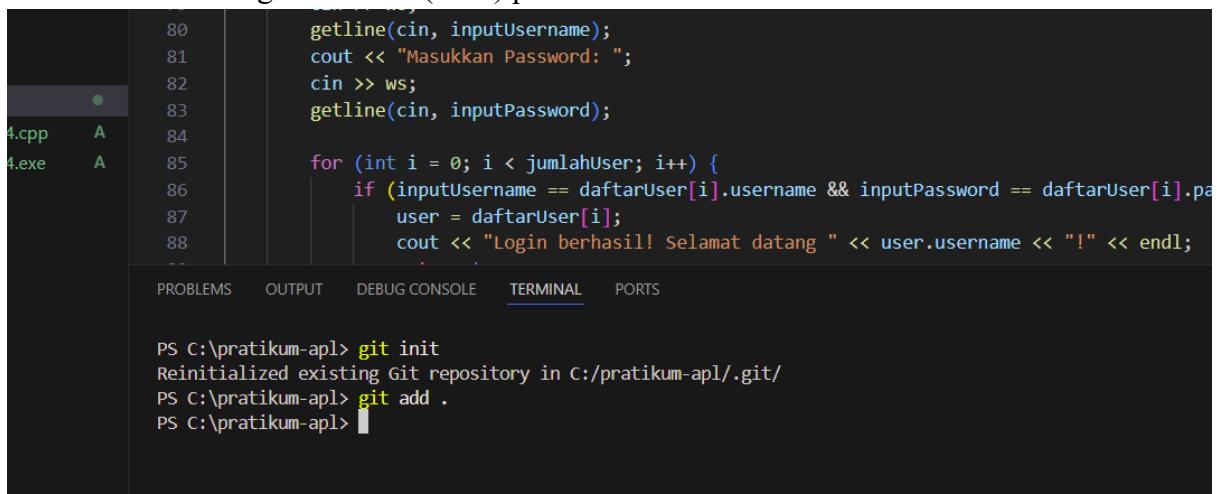
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\pratikum-apl> git init
Reinitialized existing Git repository in C:/pratikum-apl/.git/
PS C:\pratikum-apl>
```

Gambar 5.9 Git Init

## 7. Git Add

Lakukan git add untuk menambahkan file apa saja yang ingin kita commit. Kemudian. Ketikkan Perintah ‘git add atau .(tittik) pada terminal.



```
80      getline(cin, inputUsername);
81      cout << "Masukkan Password: ";
82      cin >> ws;
83      getline(cin, inputPassword);
84
85      for (int i = 0; i < jumlahUser; i++) {
86          if (inputUsername == daftarUser[i].username && inputPassword == daftarUser[i].pa
87              user = daftarUser[i];
88          cout << "Login berhasil! Selamat datang " << user.username << "!" << endl;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\pratikum-apl> git init
Reinitialized existing Git repository in C:/pratikum-apl/.git/
PS C:\pratikum-apl> git add .
PS C:\pratikum-apl>
```

Gambar 5.10 Git Add

## 8. Git Commit

Melakukan git commit untuk membuat cekpoint. Ketik perintah git commit -m “Pesan commit” pada terminal.

```
88      cout << "Login berhasil! Selamat datang " << user.user  
--  
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  
  
PS C:\pratikum-apl> git init  
Reinitialized existing Git repository in C:/pratikum-apl/.git/  
PS C:\pratikum-apl> git add .  
PS C:\pratikum-apl> git commit -m "Menambahkan file cpp dan exe"  
[main 819fbad] Menambahkan file cpp dan exe  
2 files changed, 266 insertions(+)  
create mode 100644 post-test/post-test-4/2409106018-NadiaRahmah-PT-4.cpp  
create mode 100644 post-test/post-test-4/2409106018-NadiaRahmah-PT-4.exe  
PS C:\pratikum-apl>
```

Gambar 5.11 Git Commit

## 9. Git Remote

Melakukan git remote untuk menghubungkan repository yang ada di local komputer dengan repository cloud pada Github. Copy git remote yang ada pada github, kemudian paste pada terminal.

```
PS C:\pratikum-apl> git commit -m "Menambahkan file exe"  
On branch main  
nothing to commit, working tree clean  
PS C:\pratikum-apl> git remote add origin https://github.com/nadiarahmah018/praktikum-apl.git  
PS C:\pratikum-apl>
```

Gambar 5.11 Git Remote

## 10. Git Push

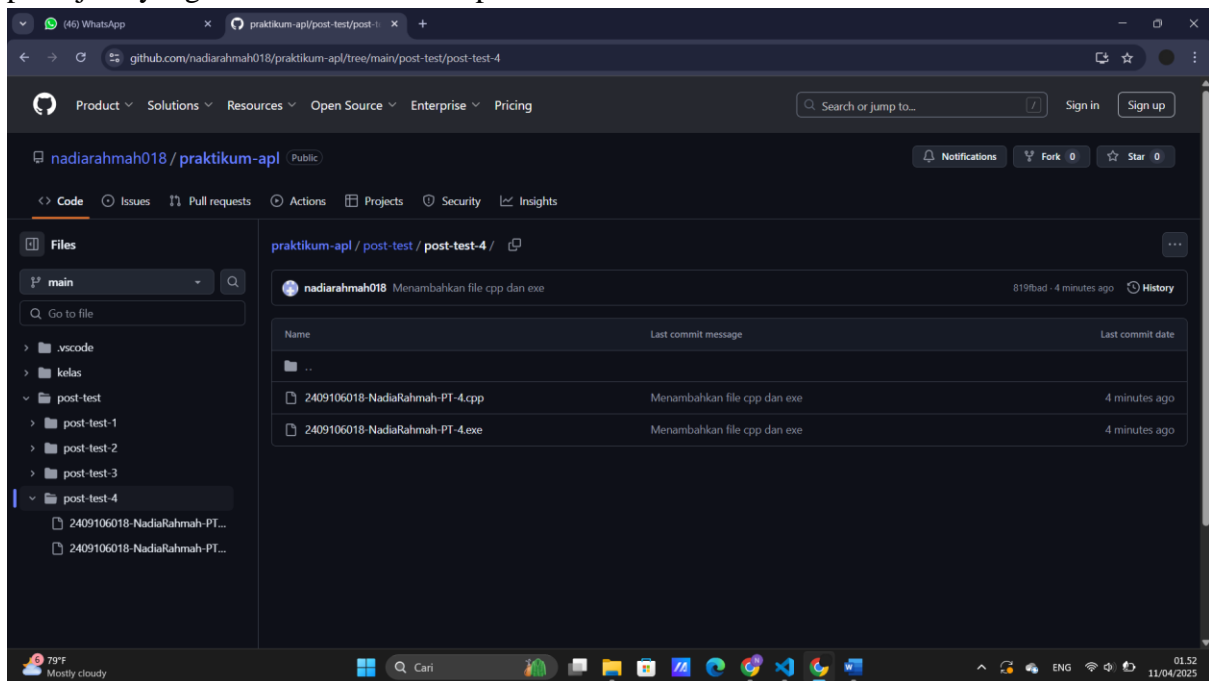
Git push adalah perintah yang digunakan untuk mengirimkan (meng-upload) perubahan yang telah di commit di repository lokal ke repository jarak jauh(remote repository). Ketik git push -u origin main pada terminal.

```
PS C:\pratikum-apl> git init
Reinitialized existing Git repository in C:/pratikum-apl/.git/
PS C:\pratikum-apl> git add .
PS C:\pratikum-apl> git commit -m "Menambahkan file cpp dan exe"
[main 819fbad] Menambahkan file cpp dan exe
2 files changed, 266 insertions(+)
create mode 100644 post-test/post-test-4/2409106018-NadiaRahmah-PT-4.cpp
create mode 100644 post-test/post-test-4/2409106018-NadiaRahmah-PT-4.exe
PS C:\pratikum-apl> git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 679.05 KiB | 4.32 MiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/nadiarahmah018/praktikum-apl.git
bc74e8a..819fbad main -> main
branch 'main' set up to track 'origin/main'.
PS C:\pratikum-apl>
```

Gambar 5.12 Git Push

## 11. Reload Tab Github di Browser

Setelah semua langkah-langkah selesai, reload tab Github di browser. Maka tampilan semua pekerjaan yang dilakukan akan tampil di Github.



Gambar 5.13 File Github