

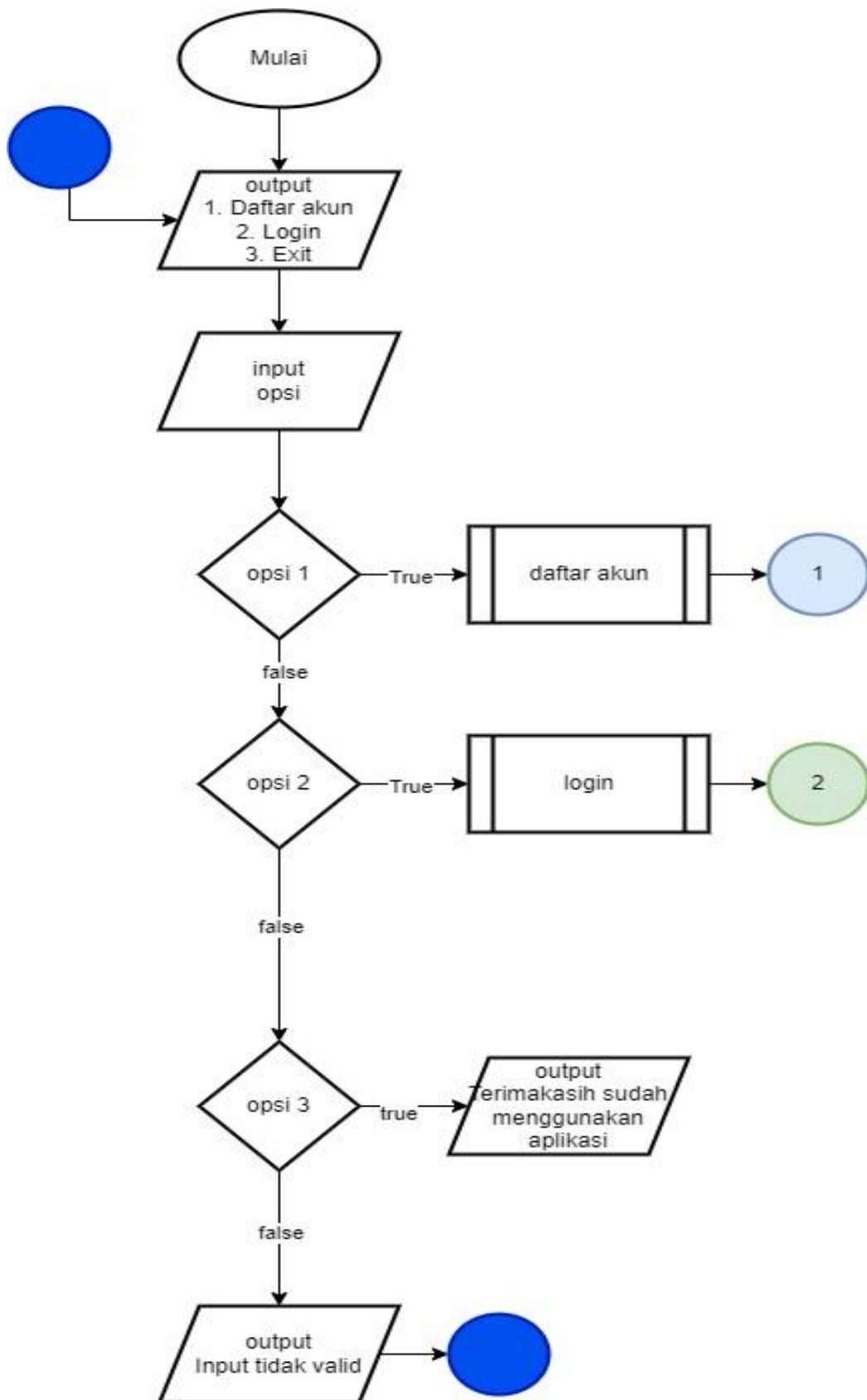
**LAPORAN PRAKTIKUM**  
**POSTTEST 2**  
**ALGORITMA PEMROGRAMAN LANJUT**

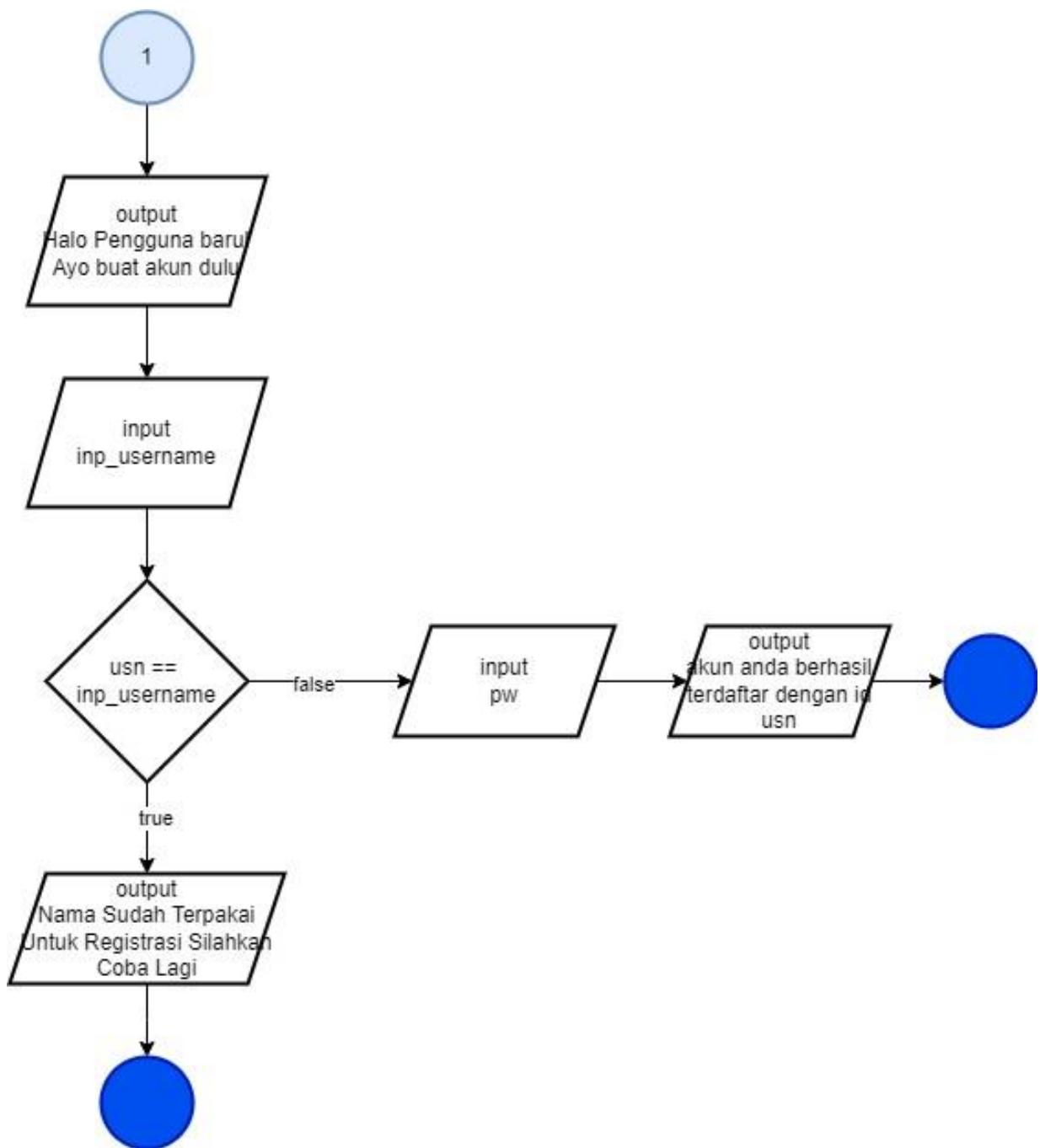


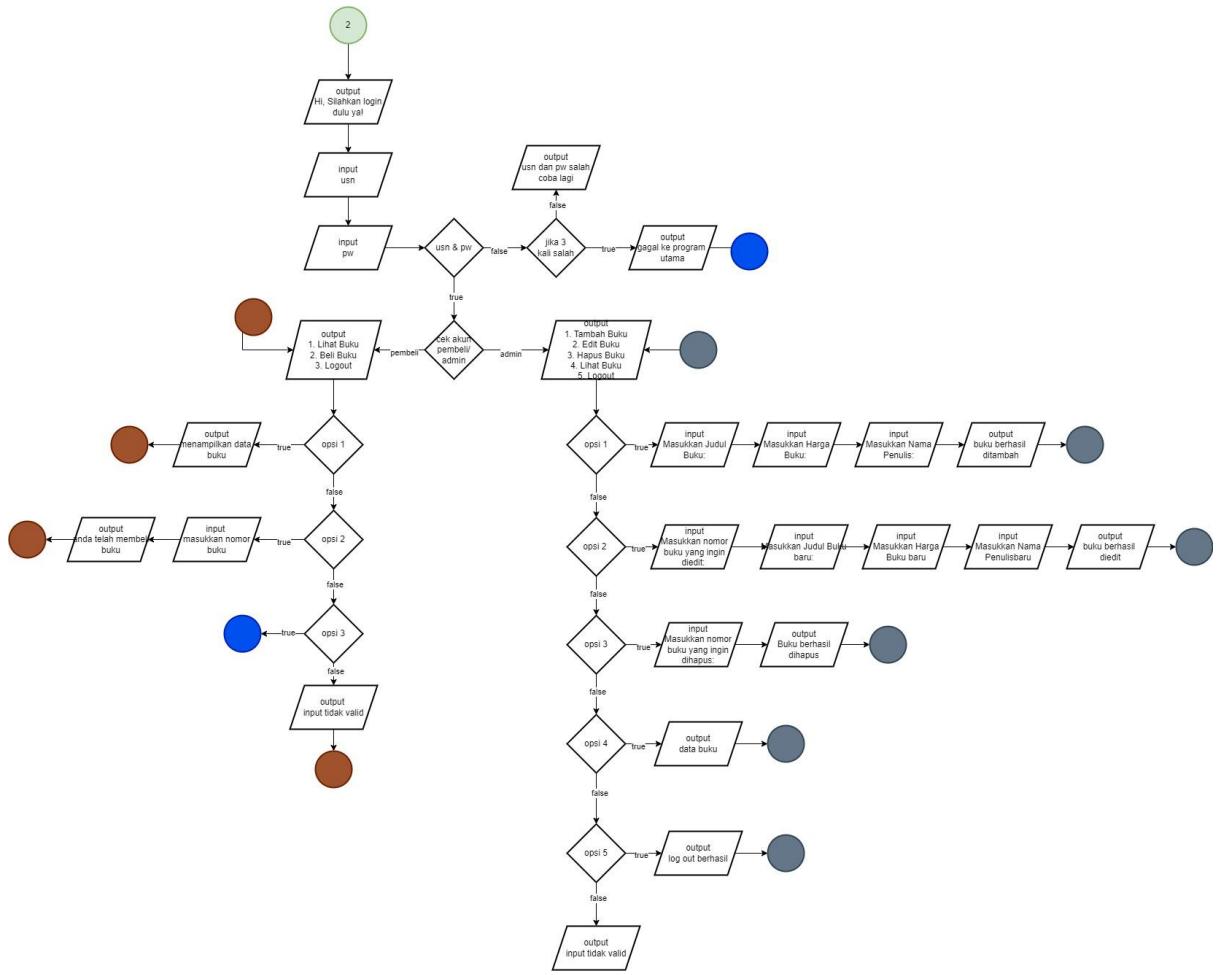
**Disusun oleh:**  
**Muhammad Kurniawan (2409106091)**  
**Kelas (C'24)**

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

## 1. Flowchart







## **2. Analisis Program**

### **2.1 Deskripsi Singkat Program**

Program ini dibuat untuk memvalidasi pembelian buku komik dan mengelola data buku setelah pengguna berhasil masuk ke sistem. Validasi dalam program ini mencakup beberapa hal, seperti membatasi percobaan login agar tidak lebih dari tiga kali, serta menampilkan daftar buku dalam format tabel yang lebih terstruktur agar mudah dibaca.

Selain itu, program ini memiliki sistem login yang ketat, sehingga hanya pengguna dengan username dan password yang benar yang bisa mengakses fitur di dalamnya. Setelah login berhasil, pengguna akan diarahkan ke menu sesuai dengan peran mereka. Admin memiliki akses untuk menambah, mengedit, menghapus, dan melihat daftar buku, sedangkan pembeli hanya dapat melihat dan membeli buku yang tersedia. Jika pengguna gagal login tiga kali berturut-turut, mereka akan dikembalikan ke menu utama untuk mencegah akses yang tidak sah.

Tujuan:

Tujuan dari program ini adalah mempermudah proses pendaftaran, login, dan pembelian buku secara terstruktur. Dengan adanya sistem ini, pembeli dapat dengan mudah melihat daftar buku yang tersedia, sedangkan admin dapat mengelola data buku dengan lebih efisien.

Fungsi:

Program ini berfungsi sebagai sistem manajemen pembelian buku komik yang memungkinkan pengguna untuk mendaftar akun, login, melihat daftar buku, dan membeli buku. Selain itu, program juga memiliki fitur khusus untuk admin agar dapat mengelola daftar buku, seperti menambah, mengedit, dan menghapus buku yang tersedia.

## 2.2 Penjelasan Alur & Algoritma

### 1 Menampilkan Menu Utama

- Program pertama-tama akan menampilkan menu utama yang terdiri dari:
  1. **Daftar akun** → Untuk pengguna baru yang ingin mendaftar sebagai pembeli.
  2. **Login** → Untuk pengguna yang sudah memiliki akun.
  3. **Exit** → Untuk keluar dari program.

### 2 Pendaftaran Akun (Hanya untuk Pembeli)

- Pengguna yang memilih "Daftar akun" akan diminta memasukkan **username dan password**.
- Program akan **mengecek apakah username sudah terdaftar sebelumnya**. Jika sudah ada, pengguna diminta untuk memilih username lain.
- Jika username unik, akun akan disimpan dalam array **akun\_pengguna[MAX\_AKUN][3]**.
- Semua akun yang terdaftar melalui menu ini **secara otomatis diberikan peran sebagai "pembeli"**.

### 3 Login dengan Batasan 3 Kali Salah

- Pengguna yang sudah memiliki akun dapat memilih opsi "Login".
- Program akan meminta pengguna memasukkan **username dan password**.
- Jika pengguna salah memasukkan username atau password **sebanyak 3 kali berturut-turut**, program akan **mengembalikan pengguna ke menu utama**.
- Jika login berhasil, pengguna akan masuk ke salah satu mode:
  - **Admin**: Jika username dan password cocok dengan akun admin.
  - **Pembeli**: Jika username cocok dengan akun yang sudah terdaftar.

### 4 Mode Admin (Mengelola Buku di Sistem)

- Jika pengguna login sebagai admin, maka mereka memiliki akses ke **5 fitur utama**:
  1. **Tambah Buku** → Admin bisa memasukkan buku baru ke dalam sistem.
  2. **Edit Buku** → Admin dapat mengganti judul, harga, atau penulis dari buku yang ada.
  3. **Hapus Buku** → Admin bisa menghapus buku dari daftar.
  4. **Lihat Buku** → Admin bisa melihat daftar semua buku dalam tabel rapi.
  5. **Logout** → Kembali ke menu utama.

### 5 Mode Pembeli (Melihat dan Membeli Buku)

- Jika pengguna login sebagai pembeli, maka mereka dapat memilih dari **3 opsi**:
  1. **Lihat Buku** → Menampilkan daftar buku dalam format tabel, termasuk informasi harga dan penulis.

2. **Beli Buku** → Pembeli dapat memasukkan nomor buku yang ingin dibeli.  
Setelah berhasil, sistem akan menampilkan pesan konfirmasi.
3. **Logout** → Kembali ke menu utama.

## 6 Program Berjalan Hingga Pengguna Memilih Exit

- Program akan terus berjalan dalam **looping utama**, sehingga setelah pengguna logout, mereka kembali ke menu utama dan bisa melakukan login lagi, mendaftar akun baru, atau memilih exit untuk mengakhiri program.

### 3. Source Code

#### A. Fitur Menu Utama

Jadi ini adalah fitur utama kami yang dimana sudah dijelaskan pada 2.2 terdapat daftar akun, login dan exit. jika memasukkan semua kode maka jadi panjang, jadi saya hanya mengambil potongan kode yang muncul menjadi fiturnya **Source Code**:

```
cout << "Halo! Selamat Datang di List Pembelian Buku Komik " << endl;
cout << "Silakan pilih 'Daftar akun' jika belum punya akun, dan jika sudah
memiliki akun silahkan 'Login' 😊" << endl;
    cout << "-----" <<
endl;        cout << "1. Daftar akun" << endl;           cout << "2.
Login" << endl;          cout << "3. Exit" << endl;
    cout << "-----" << endl;
int opsi;
cout << "Pilih opsi: ";
cin >> opsi;
```

#### B. Fitur Pembatasan Percobaan Login (Maksimal 3 Kali)

Jika pengguna gagal login 3 kali, program akan kembali ke menu utama.

#### Source code

```
int attempts = 0; bool
Login_berhasil = false; while
(attempts < 3) {    cout <<
"Masukkan Username: ";    cin >>
username;    cout << "Masukkan
Password: ";    cin >> password;

    if (username == admin_username && password == admin_password) {
Login_berhasil = true;            break;
    }
```

```

        for (int i = 0; i < jumlah_akun; i++) {
            if (akun_pengguna[i][0] == username && akun_pengguna[i][1] == password)
            {
                Login_berhasil = true;
                break;
            }
        }

        if (Login_berhasil) break;

        attempts++;
        cout << "Username atau password salah! Percobaan ke-"
        " << attempts << "/3" << endl;
    }

    if (!Login_berhasil) {
        cout << "Anda telah gagal login 3 kali.
        Program kembali ke menu utama.\n";
        continue;
    }
}

```

**C. Variabel Array** ini adalah 3 variabel array satu dimensi dan variabel array multidimensi

#### Source code

```

string akun_pengguna[MAX_AKUN][3]; // [username, password, role]

string judul_buku[MAX_BUKU] = {"One Piece", "Naruto", "Dragon Ball"};
string harga_buku[MAX_BUKU] = {"50000", "45000", "55000"};
string penulis_buku[MAX_BUKU] = {"Eiichiro Oda", "Masashi Kishimoto",
"Akira Toriyama"};

```

#### D. Fitur CRUD ADMIN

kode ini potongan dari menu CRUD karna jika dimasukkan semua jadi panjang, jadi ini adalah menu CRUD dari program yang saya buat

## Source code

```
while (true) {                                cout << "\nSelamat datang !"
<< endl;                                     cout << "-----Silakan pilih langkah
yang kamu mau!-----" << endl;                cout << "1.
Tambah Buku" << endl;                         cout << "2. Edit Buku"
<< endl;                                     cout << "3. Hapus Buku" << endl;
cout << "4. Lihat Buku" << endl;                 cout << "5.
Logout" << endl;

int pilihan;
cout << "Pilih opsi: ";
cin >> pilihan;
```

#### **E. Fitur User Dapat Memilih Menu**

Tidak hanya admin saja yang bisa memilih menu tetapi sebagai user juga bisa

## Source code

## **F. Data Buku Menggunakan Tabel**

Fungsi ini berguna untuk data - data buku tersebut bisa terlihat lebih rapi dengan menggunakan tabel

## Source code

```
if (pilihan == 1) { cout << "nDaftar Buku:n";
```

```
        cout << left << setw(5) << "No" << setw(20)
<< "Judul" << setw(10) << "Harga" << setw(20) << "Penulis" << endl;
cout <<
"-----\n";
for (int i = 0; i < jumlah_buku; i++) {                                cout <<
left << setw(5) << i+1 << setw(20) << judul_buku[i] << setw(10) <<
harga_buku[i] << setw(20) << penulis_buku[i] << endl;
}
```

## **4. Uji Coba dan Hasil Output**

### **4.1 Uji Coba**

**1. Pengujian Daftar Akun** input : Mendaftar akun dengan

username yang sudah digunakan

output yang diharapkan : Menampilkan pesan "Username sudah digunakan!"

**2. Pengujian Login** input : Login dengan username atau password

yang salah 3 kali output yang diharapkan : Program kembali ke

menu utama

### **3. Pengujian Pada Menu Admin Tambah**

input : memasukkan data buku yang ingin dijual output yang

diharapkan : berhasil memasukkan barang ke data barang

**4. Pengujian Pada Menu Admin Edit** input : mengedit data buku

yang sudah ada output yang diharapkan : berhasil mengedit /

mengubah data buku

### **5. Pengujian Pada Menu Admin Hapus**

input : memilih barang yang akan dihapus di data buku output

yang diharapkan : berhasil menghapus data buku

**6. Pengujian Pada User Melihat Buku** input :

memilih fitur melihat barang output yang

diharapkan : data berhasil keluar

## **7. Pengujian Pada User Membeli Buku**

input : user dapat memilih buku mana yang ingin dibeli setelah menggunakan fitur melihat barang output yang diharapkan : user berhasil membeli barang

## 4.2 Hasil Output

```
1. Daftar akun
2. Login
3. Exit
Pilih opsi: 1
Halo Pengguna baru! Ayo buat akun dulu
Username: wawan
Password: 123
Akun berhasil terdaftar wawan!
```

```
1. Daftar akun
2. Login
3. Exit
Pilih opsi: 1
Halo Pengguna baru! Ayo buat akun dulu
Username: wawan
Username sudah dipakai! Silakan pilih username lain.
```

Gambar 4.1.1

```
1. Daftar akun
2. Login
3. Exit
Pilih opsi: 2
Hi, Silahkan login dulu ya!
Masukkan Username: wawan
Masukkan Password: 12
Username atau password salah! Percobaan ke-1/3
Masukkan Username: wawan
Masukkan Password: 23
Username atau password salah! Percobaan ke-2/3
Masukkan Username: wawa
Masukkan Password: 123
Username atau password salah! Percobaan ke-3/3
Anda telah gagal login 3 kali. Program kembali ke menu utama.
```

Gambar 4.1.2

Daftar Buku:			
No	Judul	Harga	Penulis
1	One Piece	50000	Eiichiro Oda
2	Naruto	45000	Masashi Kishimoto
3	Dragon Ball	55000	Akira Toriyama

Daftar Buku:			
No	Judul	Harga	Penulis
1	One Piece	50000	Eiichiro Oda
2	Naruto	45000	Masashi Kishimoto
3	Dragon Ball	55000	Akira Toriyama
4	Boruto	45000	Eiichiro

Gambar 4.1.3

Daftar Buku:			
No	Judul	Harga	Penulis
1	One Piece	50000	Eiichiro Oda
2	Naruto	45000	Masashi Kishimoto
3	Dragon Ball	55000	Akira Toriyama
4	Boruto	45000	Eiichiro

Daftar Buku:			
No	Judul	Harga	Penulis
1	One Piece	50000	Eiichiro Oda
2	Naruto	45000	Masashi Kishimoto
3	Dragon Ball	55000	Akira Toriyama
4	Bleach	50000	Oda

Gambar 4.1.4

Daftar Buku:			
No	Judul	Harga	Penulis
1	One Piece	50000	Eiichiro Oda
2	Naruto	45000	Masashi Kishimoto
3	Dragon Ball	55000	Akira Toriyama
4	Bleach	50000	Oda

Daftar Buku:			
No	Judul	Harga	Penulis
1	One Piece	50000	Eiichiro Oda
2	Naruto	45000	Masashi Kishimoto
3	Dragon Ball	55000	Akira Toriyama

Gambar 4.1.5

1. Lihat Buku			
2. Beli Buku			
3. Logout			
Pilih opsi: 1			
Daftar Buku:			
No	Judul	Harga	Penulis
1	One Piece	50000	Eiichiro Oda
2	Naruto	45000	Masashi Kishimoto
3	Dragon Ball	55000	Akira Toriyama

Gambar 4.1.6

1. Lihat Buku
2. Beli Buku
3. Logout
Pilih opsi: 2
Masukkan nomor buku yang ingin dibeli: 1
Anda telah membeli buku One Piece!

Gambar 4.1.7

## 5.Git

```
ASUS@LAPTOP-SSIOUP20 MINGW64 ~/Documents/Praktikum-APL/Post-test/post-test-2 (master)
$ git add .

ASUS@LAPTOP-SSIOUP20 MINGW64 ~/Documents/Praktikum-APL/Post-test/post-test-2 (master)
$ git ls-files
2409106091-Muhammad Kurniawan-PT-2.pdf
2409106091-MuhammadKurniawan-PT-2.cpp
2409106091-MuhammadKurniawan-PT-2.exe

ASUS@LAPTOP-SSIOUP20 MINGW64 ~/Documents/Praktikum-APL/Post-test/post-test-2 (master)
$ git remote add origin https://github.com/MuhammadKurniawan18/Praktikum-Apl.git
error: remote origin already exists.

ASUS@LAPTOP-SSIOUP20 MINGW64 ~/Documents/Praktikum-APL/Post-test/post-test-2 (master)
$ git remote -v
origin  https://github.com/MuhammadKurniawan18/Praktikum-Apl.git (fetch)
origin  https://github.com/MuhammadKurniawan18/Praktikum-Apl.git (push)

ASUS@LAPTOP-SSIOUP20 MINGW64 ~/Documents/Praktikum-APL/Post-test/post-test-2 (master)
$ git push -u origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 20 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.26 MiB | 815.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:     https://github.com/MuhammadKurniawan18/Praktikum-Apl/pull/new/master
remote:
To https://github.com/MuhammadKurniawan18/Praktikum-Apl.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

1. Git add .
  - Perintah ini digunakan untuk menambahkan semua file dalam direktori kerja ke staging area, sehingga siap untuk dikomit.
2. git ls-files
  - Perintah ini menampilkan daftar file yang sedang dilacak oleh Git dalam repositori.
  - Hasilnya menunjukkan tiga file:
  - 2409106091-Muhammad Kurniawan-PT-2.pdf
  - 2409106091-Muhammad Kurniawan-PT-2.cpp
  - 2409106091-Muhammad Kurniawan-PT-2.exe
3. git remote add origin <https://github.com/Muhammadkurniawan18/Praktikum-Apl.git>
  - Perintah ini digunakan untuk menambahkan repositori jarak jauh dengan nama origin
  - Namun, terjadi error karena remote dengan nama originya sudah ada sebelumnya.
4. git remote -v

- Perintah ini digunakan untuk memverifikasi daftar repositori jarak jauh yang telah ditambahkan.
- Hasilnya menunjukkan bahwa originsudah dikonfigurasi untuk fetch(mengambil perubahan) dan push(mengunggah perubahan) ke repositori GitHub.

#### 5. git push -u origin master

- Perintah ini digunakan untuk mengunggah (push) commit dari cabang masterke repositori GitHub yang terhubung ke remote origin.
- Proses yang terjadi:
  - Git menghitung jumlah objek yang akan dikirim.
  - Melakukan kompresi data untuk efisiensi pengiriman.
  - Mengunggah perubahan ke GitHub.
- Setelah selesai, muncul pesan bahwa Branch mastertelah diunggah ke GitHub dan sekarang terlacak dengan origin/master

#### 6. Informasi tambahan dari keluaran Git:

- Pesan dari GitHub memberikan tautan untuk membuat pull request jika diinginkan
- branch 'master' set up to track 'origin/master'berarti cabang mastersekarang terhubung dengan cabang masterdi repositori GitHub.

Kesimpulan:

Pengguna telah berhasil menambahkan file, memverifikasi repositori jarak jauh, dan mengunggah perubahan ke cabang masterdi GitHub.