

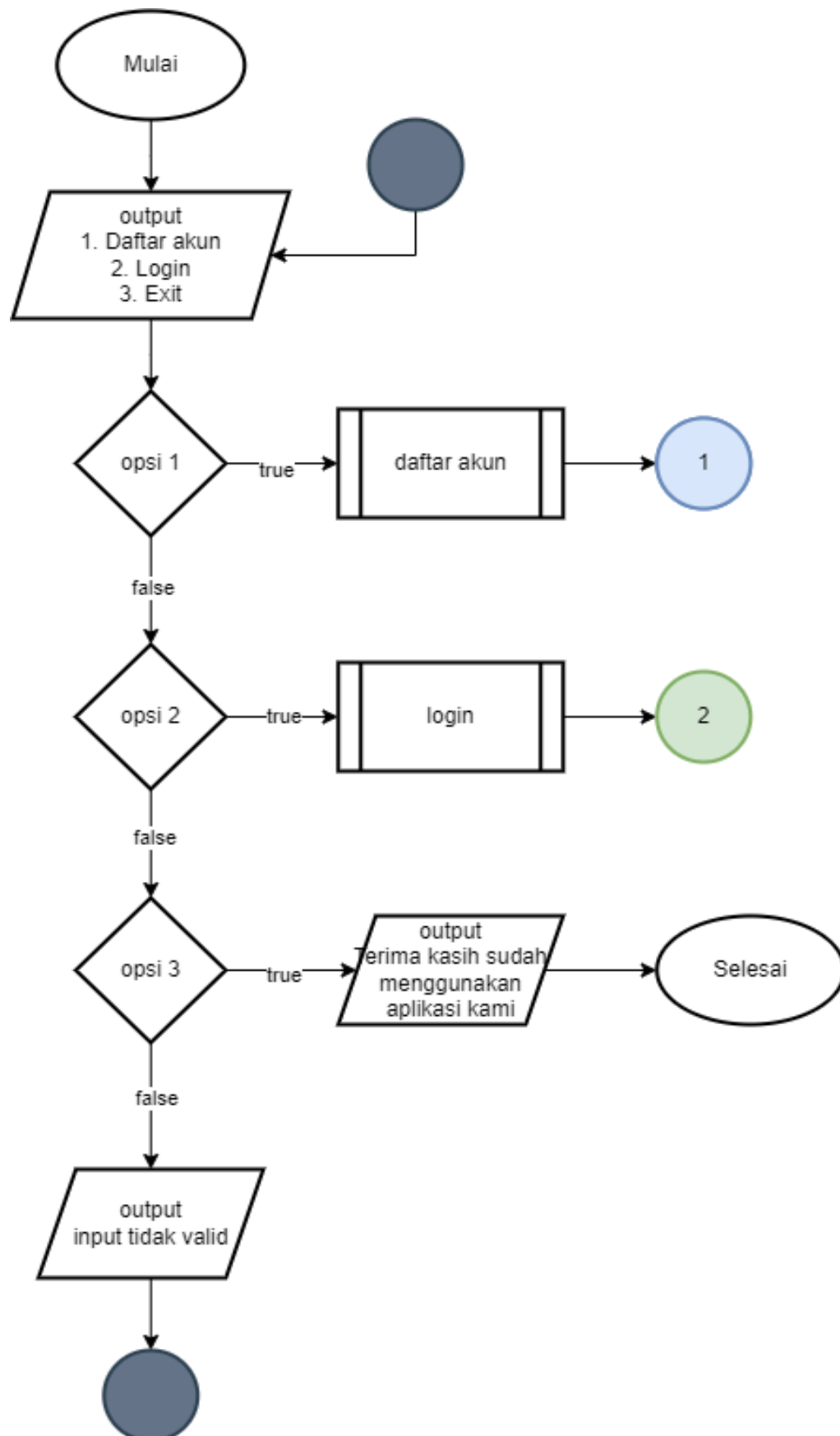
LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN LANJUT

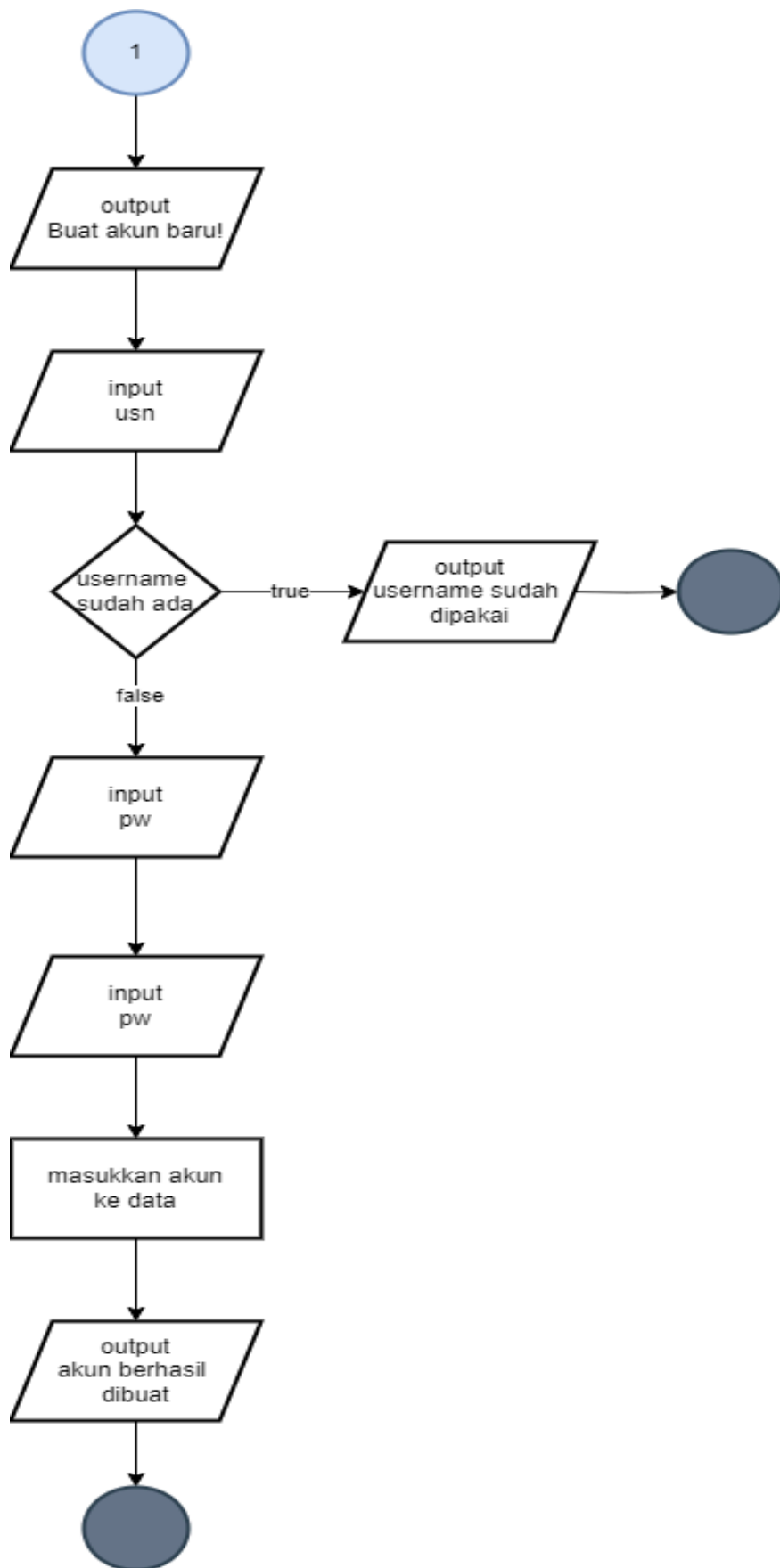


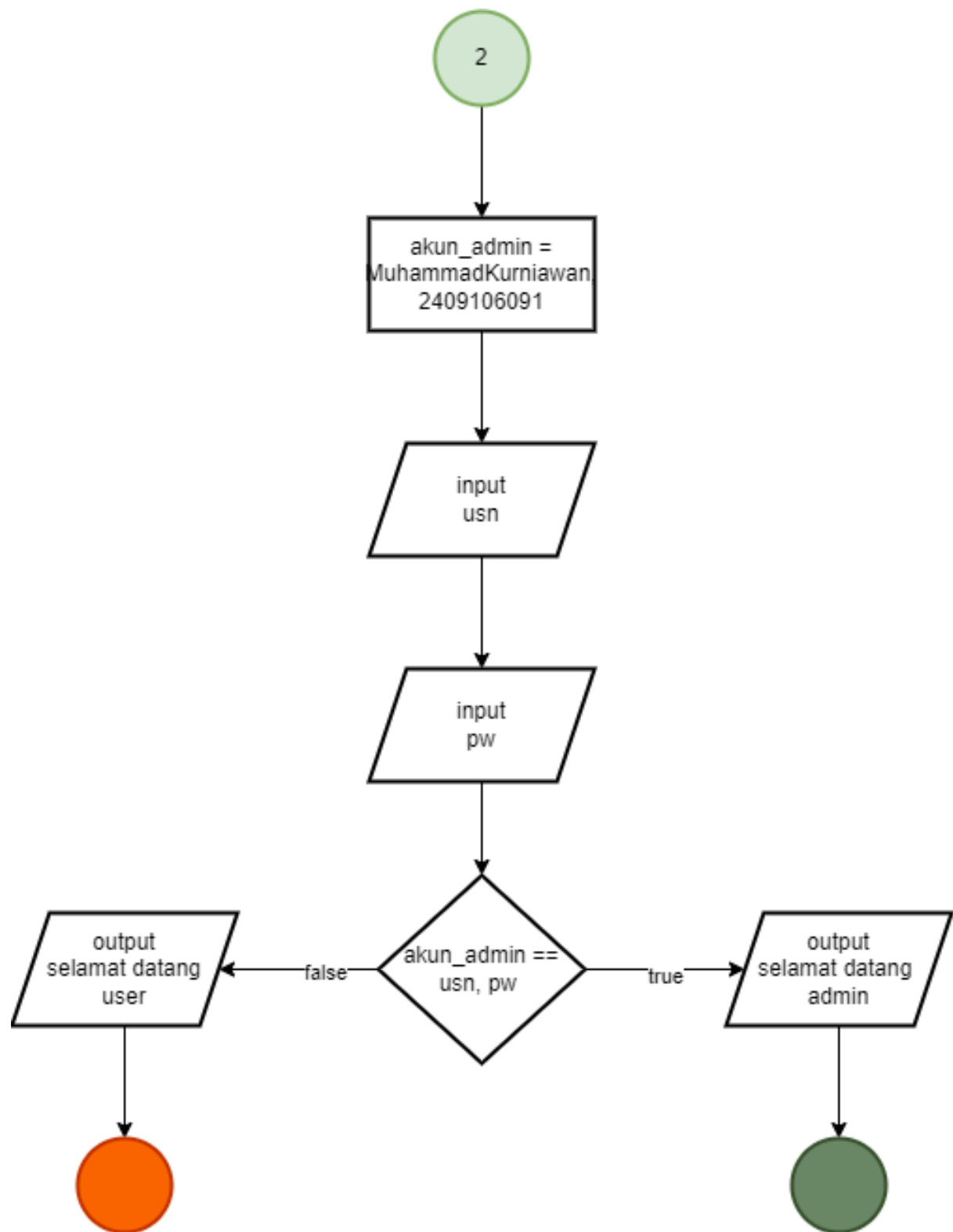
Disusun oleh:
Muhammad Kurniawan
(2409106091)
Kelas (C'24)

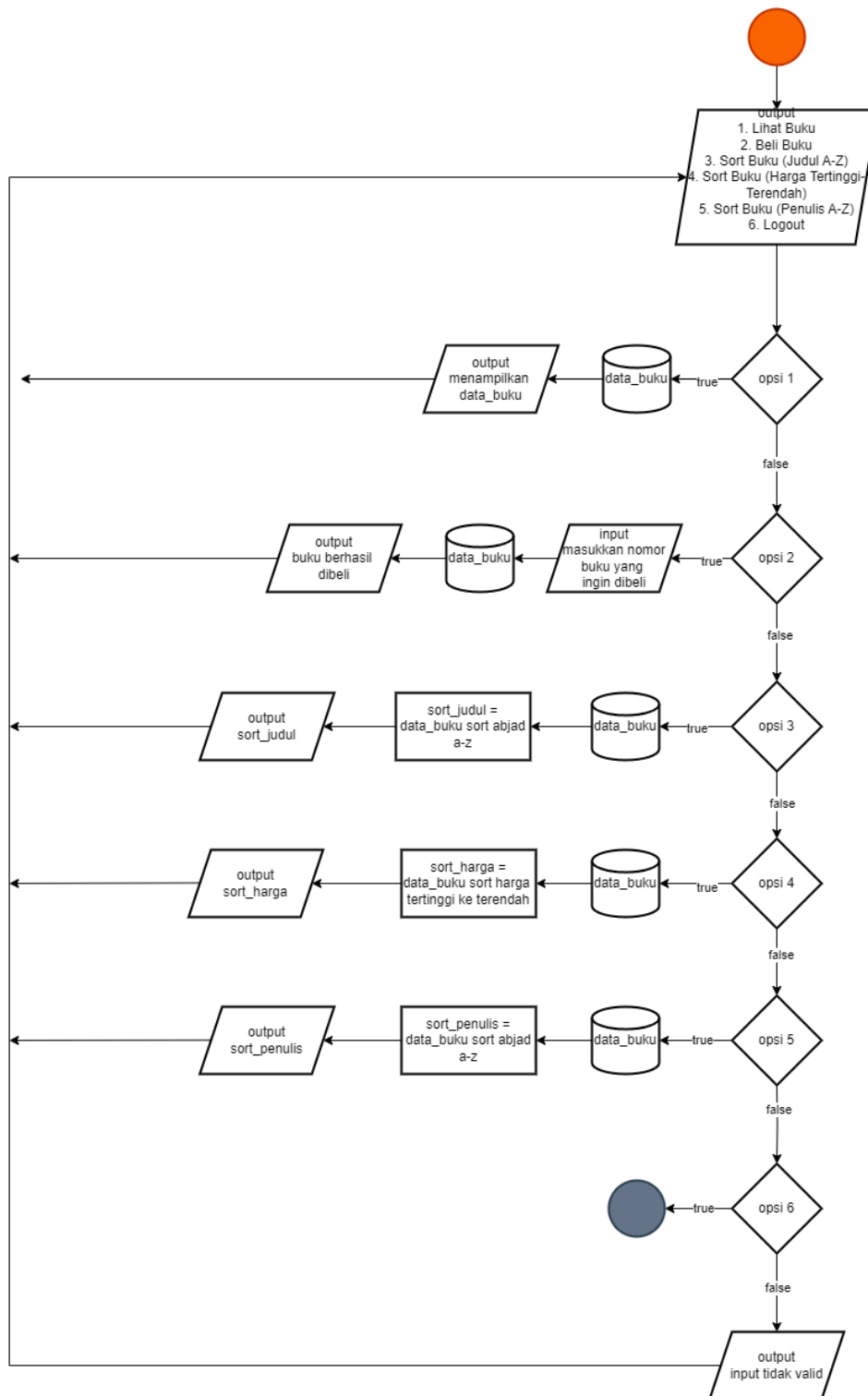
PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

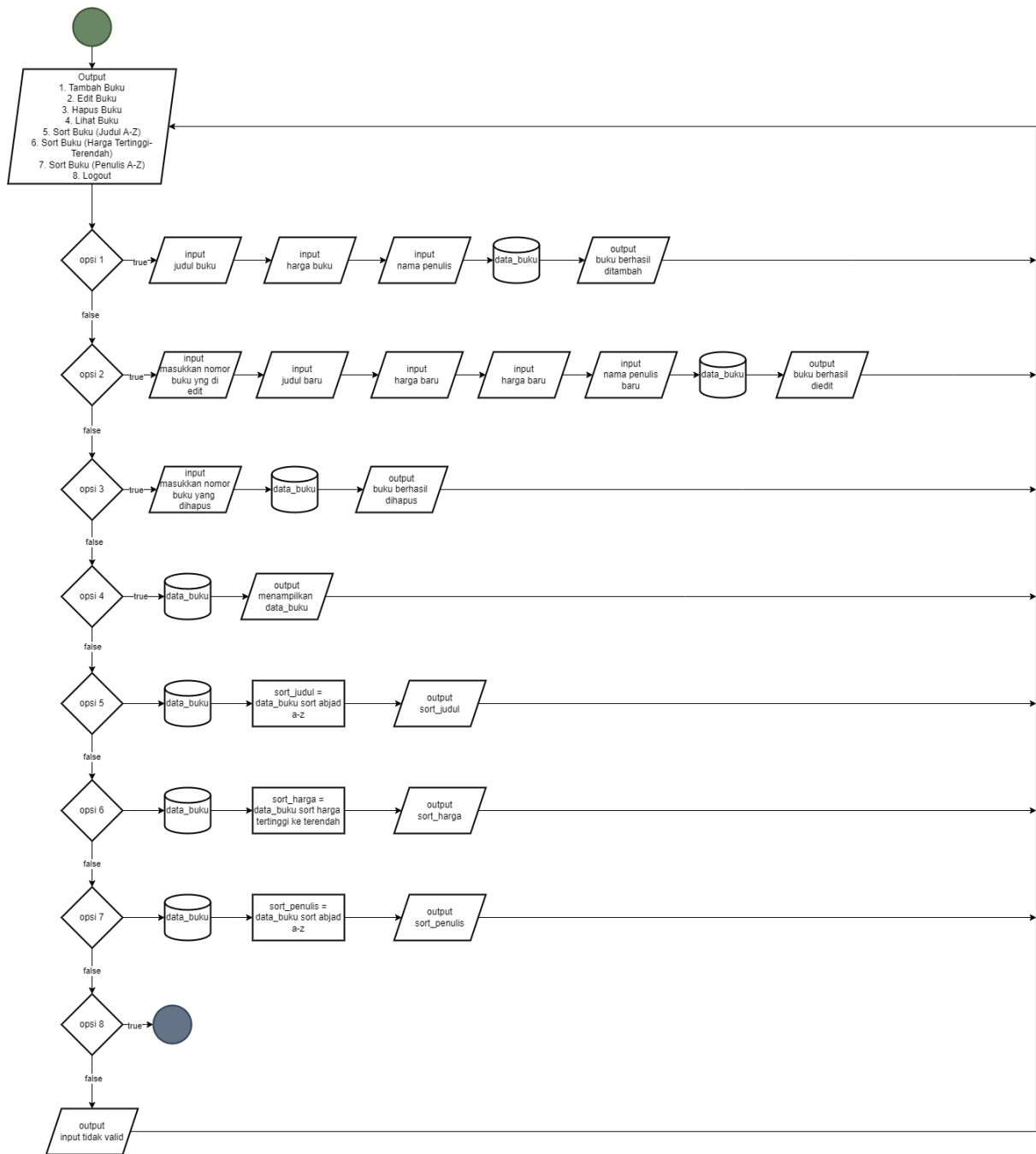
1. Flowchart











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.

Program ini sama saja seperti sebelumnya hanya saja menambahkan fitur baru, yaitu yang pertama sort buku judul (ascending) dimana dimulai dari abjad a sampai z menggunakan algoritma Bubble sort, lalu yang kedua sort buku berdasarkan harga tertinggi ke terendah (descending) menggunakan algoritma selection sort karna kita cari nilai tertinggi maka selection sort selalu mencari nilai maks dan yang terakhir sort buku berdasarkan penulis (ascending) dimana dimulai dari abjad a sampai z menggunakan Insertiom sort. Itu saja fitur tambahan dari program ini karena nim saya ganjil maka ascending nya huruf dan descending nya angka.

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.
- 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. **Sort Buku (Judul A-Z)** → Sorting Buku dengan judul sesuai abjad dari A - Z
 6. **Sort Buku (Harga Tertinggi-Terendah)** → Sorting buku dengan harga tertinggi ke terendah
 7. **Sort Buku (Penulis A-Z)** → Sorting buku dengan nama penulis sesuai abjad dari A-Z
 8. **Logout** → Kembali ke menu utama.

5 Mode Pembeli (Melihat dan Membeli Buku)

- Jika pengguna login sebagai pembeli, maka mereka dapat memilih dari **5 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. **Sort Buku (Judul A-Z)** → Sorting Buku dengan judul sesuai abjad dari A - Z
4. **Sort Buku (Harga Tertinggi-Terendah)** → Sorting buku dengan harga tertinggi ke terendah
5. **Sort Buku (Penulis A-Z)** → Sorting buku dengan nama penulis sesuai abjad dari A-Z
6. **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

Pada source code ini saya tidak memasukkan semua source code, tapi source code yang sedang dikerjakan pada Posttest ini

A. Deklarasi Function Fitur Baru Menggunakan Algoritma

Pada potongan kode ini untuk membuat fungsi dan prosedur harus di deklarasikan dahulu

Source Code:

```
void sortBukuJudulAsc(Buku daftar_buku[], int &jumlah_buku);  
void sortBukuHargaDesc(Buku daftar_buku[], int &jumlah_buku);  
void sortBukuPenulisAsc(Buku daftar_buku[], int &jumlah_buku);
```

B. Function Sorting Judul Ascending

Penerapan algoritma bubble sort pada kode ini

Source code

```
void sortBukuJudulAsc(Buku daftar_buku[], int &jumlah_buku) {  
    for (int i = 0; i < jumlah_buku - 1; i++) {  
        for (int j = 0; j < jumlah_buku - i - 1; j++) {  
            if (daftar_buku[j].judul > daftar_buku[j + 1].judul) {  
                Buku temp = daftar_buku[j];  
                daftar_buku[j] = daftar_buku[j + 1];  
                daftar_buku[j + 1] = temp;  
            }  
        }  
    }  
  
    cout << "Total " << jumlah_buku << " buku telah diurutkan berdasarkan  
judul (A-Z)!" << endl;  
    cout << "Jumlah buku setelah pengurutan: " << jumlah_buku << endl;  
    tampilkanBuku(daftar_buku, jumlah_buku);  
}
```

C. Function Sorting Harga Descending

Data buku akan disorting melalui harga tertinggi ke terendah

Source code

```
void sortBukuHargaDesc(Buku daftar_buku[], int &jumlah_buku) {
    for (int i = 0; i < jumlah_buku - 1; i++) {
        int maxIndex = i;
        for (int j = i + 1; j < jumlah_buku; j++) {
            if (stoi(daftar_buku[j].harga) >
stoi(daftar_buku[maxIndex].harga)) {
                maxIndex = j;
            }
        }
        if (maxIndex != i) {
            Buku temp = daftar_buku[i];
            daftar_buku[i] = daftar_buku[maxIndex];
            daftar_buku[maxIndex] = temp;
        }
    }

    cout << "Total " << jumlah_buku << " buku telah diurutkan berdasarkan
harga (tertinggi-terendah) menggunakan Selection Sort!" << endl;
    tampilkanBuku(daftar_buku, jumlah_buku);
}
```

D. Function Sorting Penulis Ascending

Data buku akan disorting melalui nama penulis sesuai abjad dari A - Z

Source code

```
void sortBukuPenulisAsc(Buku daftar_buku[], int &jumlah_buku) {
    for (int i = 1; i < jumlah_buku; i++) {
        Buku key = daftar_buku[i];
        int j = i - 1;

        while (j >= 0 && daftar_buku[j].penulis > key.penulis) {
            daftar_buku[j + 1] = daftar_buku[j];
            j--;
        }
        daftar_buku[j + 1] = key;
    }

    cout << "Total " << jumlah_buku << " buku telah diurutkan berdasarkan
nama penulis (A-Z) menggunakan Insertion Sort!" << endl;
    tampilkanBuku(daftar_buku, jumlah_buku);
}
```

4. Uji Coba dan Hasil Output

Dalam pengujian hanya menguji pada penerapan algoritma yang baru saja ditambahkan saja

4.1 Uji Coba

1. Pengujian Mengurutkan Berdasarkan Judul

input : Menginput pilihan 3 untuk mengurutkan berdasarkan judul

output yang diharapkan : Berhasil menampilkan data buku yang diurutkan sesuai judul

2. Pengujian Mengurutkan Berdasarkan Harga

input : Menginput pilihan 4 untuk mengurutkan berdasarkan Harga

output yang diharapkan : Berhasil menampilkan data buku yang diurutkan sesuai harga

3. Pengujian Mengurutkan Berdasarkan Penulis

input : Menginput pilihan 5 untuk mengurutkan berdasarkan nama penulis

output yang diharapkan : Berhasil menampilkan data buku yang diurutkan sesuai nama penulis

4.2 Hasil Output

```
Selamat datang di aplikasi kami!  
1. Lihat Buku  
2. Beli Buku  
3. Sort Buku (Judul A-Z)  
4. Sort Buku (Harga Tertinggi-Terendah)  
5. Sort Buku (Penulis A-Z)  
6. Logout  
  
Pilih opsi: 3  
Total 3 buku telah diurutkan berdasarkan judul (A-Z)!  
Jumlah buku setelah pengurutan: 3  
  
Daftar Buku:  
1. Dragon Ball - 55000 - Akira Toriyama  
2. Naruto - 45000 - Masashi Kishimoto  
3. One Piece - 50000 - Eiichiro Oda
```

Gambar 4.1.1

```
Selamat datang di aplikasi kami!  
1. Lihat Buku  
2. Beli Buku  
3. Sort Buku (Judul A-Z)  
4. Sort Buku (Harga Tertinggi-Terendah)  
5. Sort Buku (Penulis A-Z)  
6. Logout  
  
Pilih opsi: 4  
Total 3 buku telah diurutkan berdasarkan harga (tertinggi-terendah) menggunakan Selection Sort!  
  
Daftar Buku:  
1. Dragon Ball - 55000 - Akira Toriyama  
2. One Piece - 50000 - Eiichiro Oda  
3. Naruto - 45000 - Masashi Kishimoto
```

Gambar 4.1.2

```
Selamat datang di aplikasi kami!  
1. Lihat Buku  
2. Beli Buku  
3. Sort Buku (Judul A-Z)  
4. Sort Buku (Harga Tertinggi-Terendah)  
5. Sort Buku (Penulis A-Z)  
6. Logout  
  
Pilih opsi: 5  
Total 3 buku telah diurutkan berdasarkan nama penulis (A-Z) menggunakan Insertion Sort!  
  
Daftar Buku:  
1. Dragon Ball - 55000 - Akira Toriyama  
2. One Piece - 50000 - Eiichiro Oda  
3. Naruto - 45000 - Masashi Kishimoto
```

Gambar 4.1.3

