

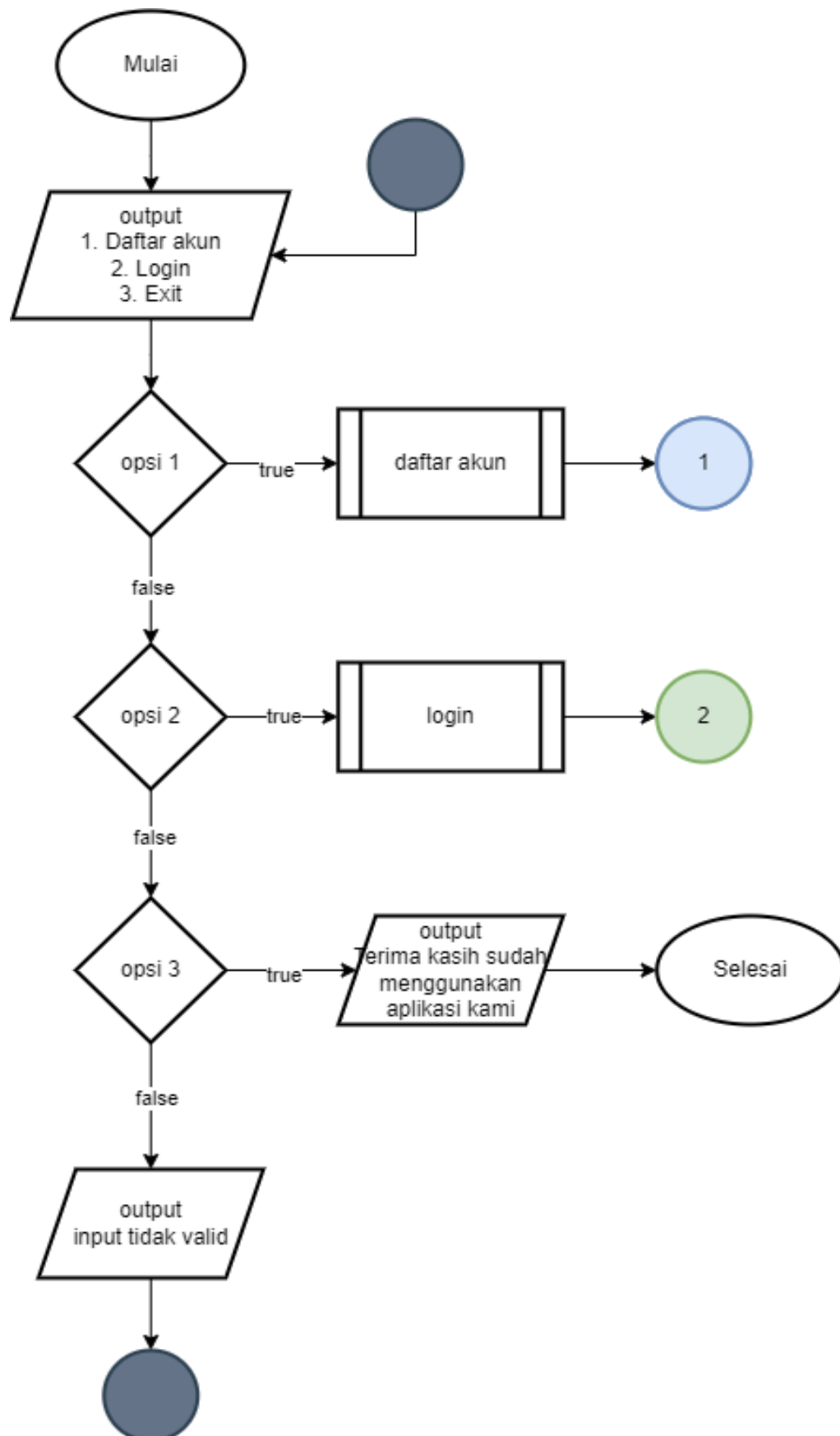
LAPORAN PRAKTIKUM
POSTTEST 5
ALGORITMA PEMROGRAMAN LANJUT

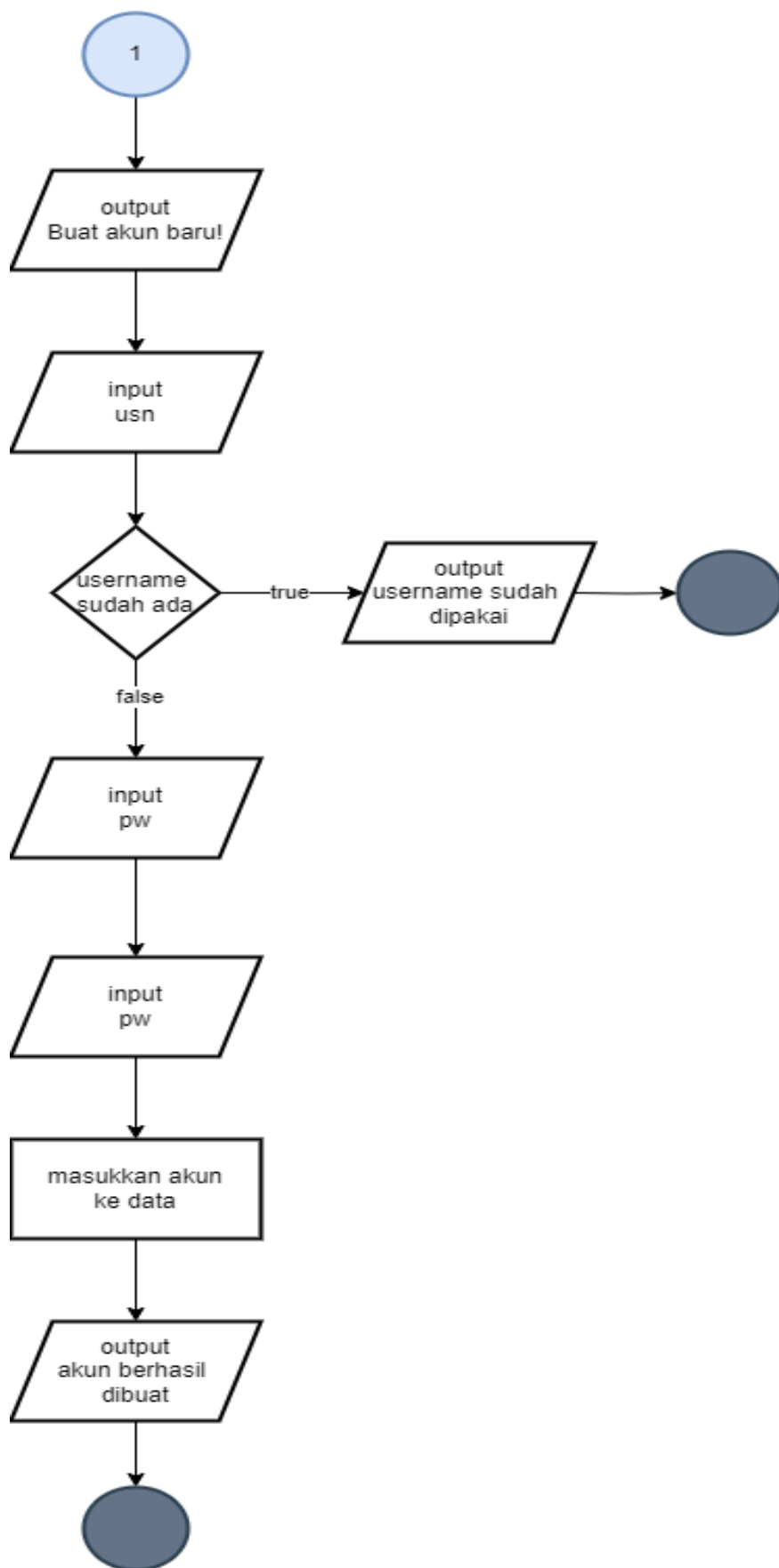


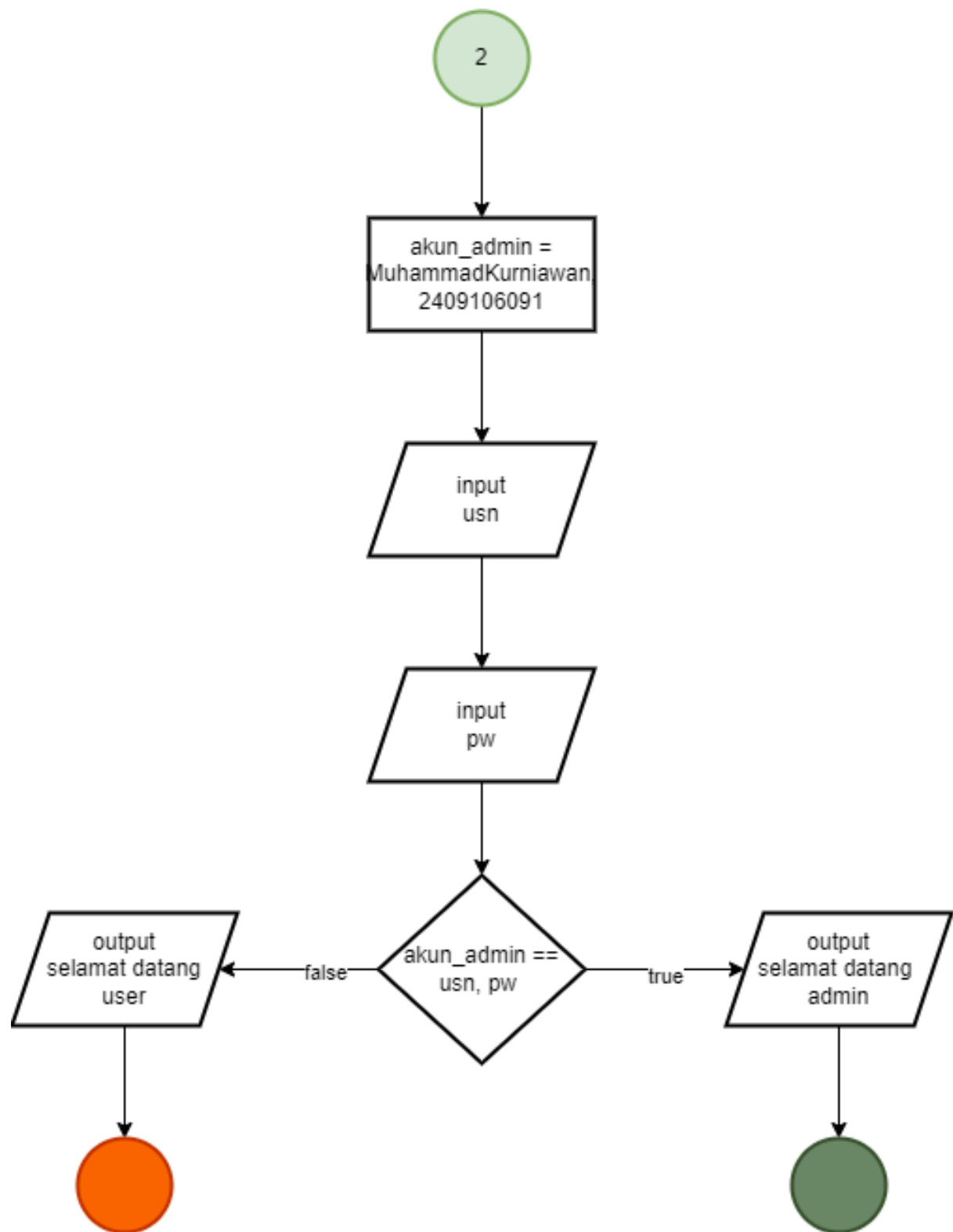
Disusun oleh:
Muhammad Kurniawan
(2409106091)
Kelas (C'24)

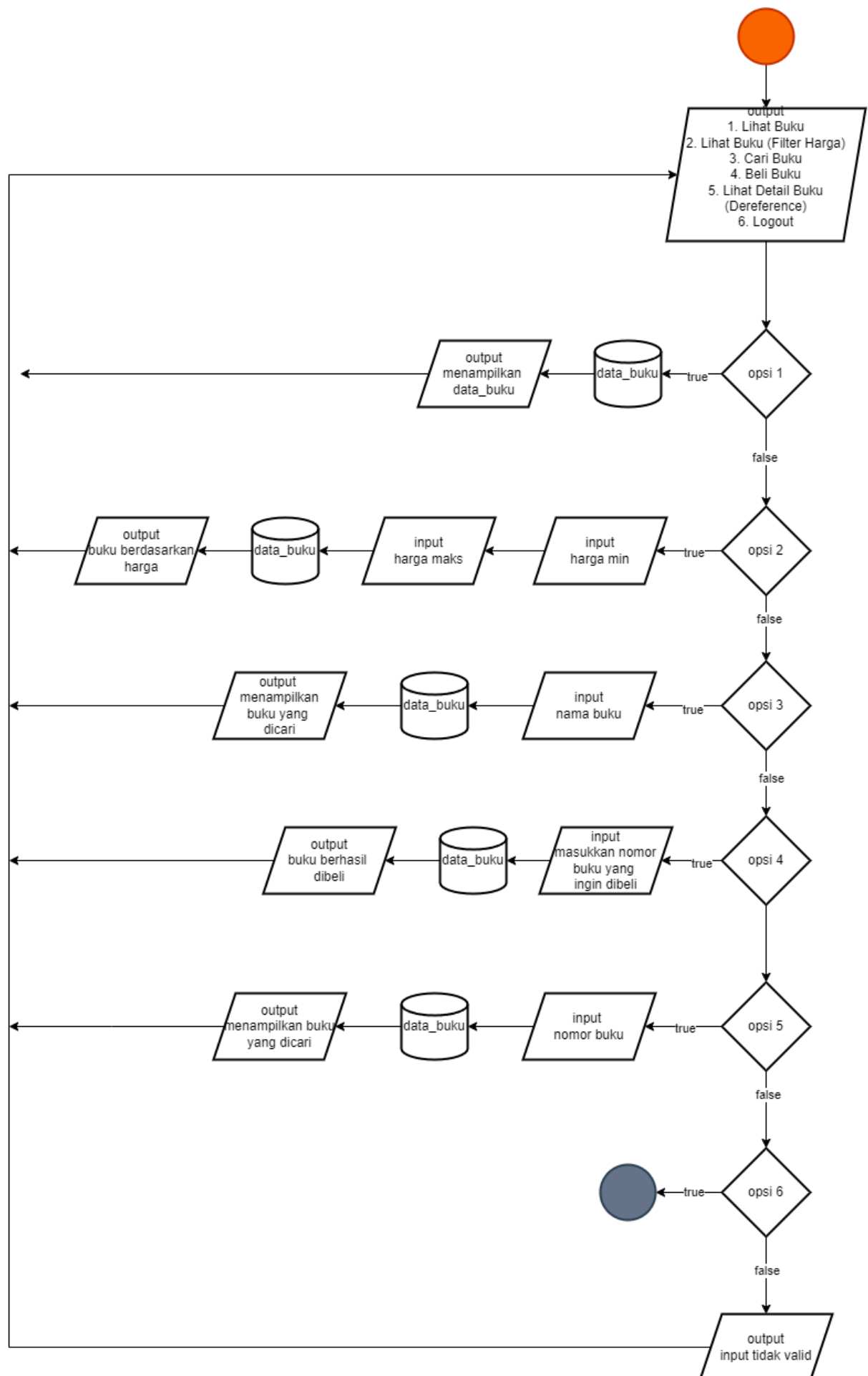
PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

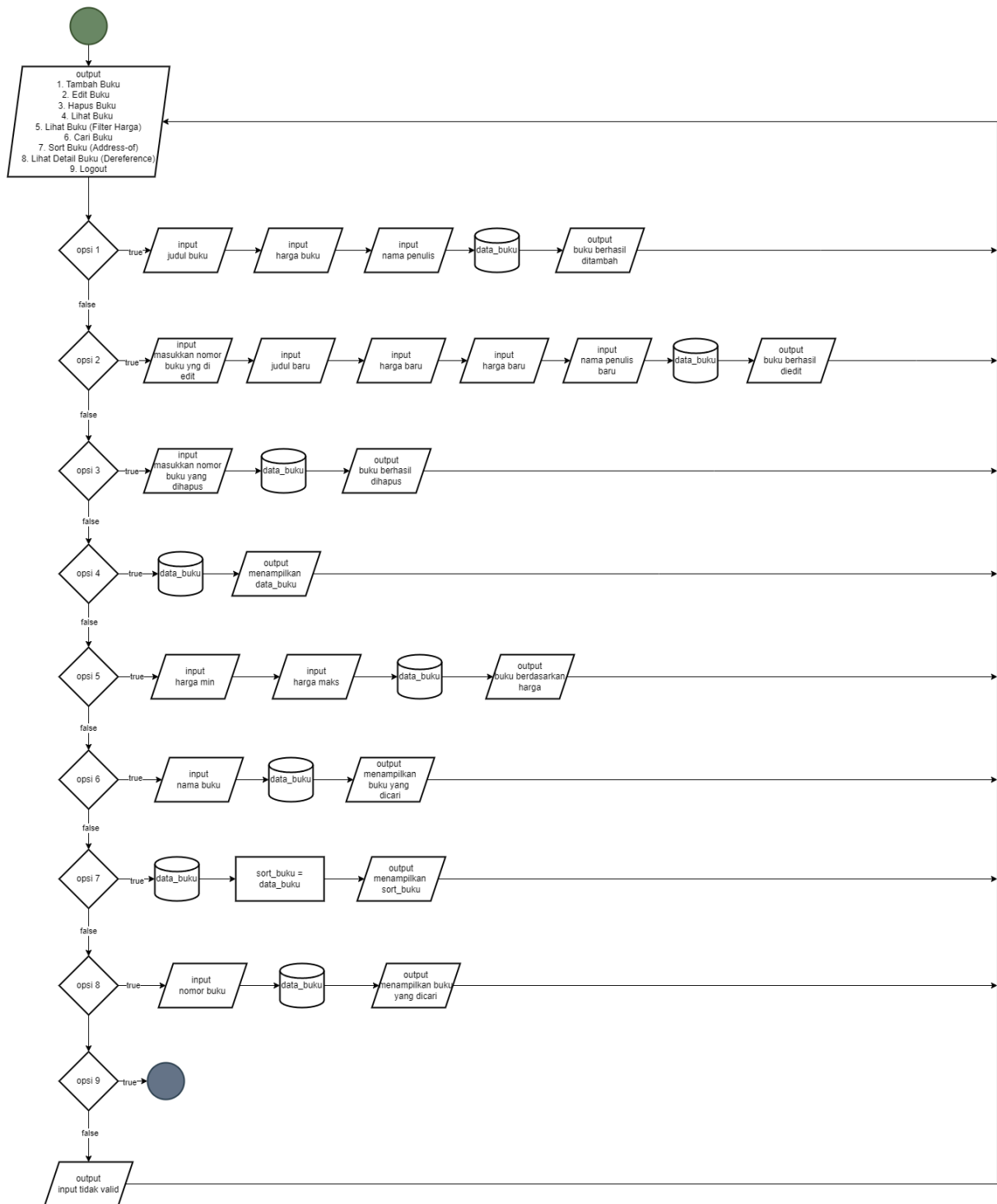
1. Flowchart











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 telah ditambahkan dengan penerapan address-of dan dereference, pada address-of terdapat sort buku dari harga jadi disini juga menggunakan algoritma bubble sort setelah itu tampilan buku akan terurut dari terendah ke terbesar tetapi nilai asli yang ada di tampilan buku juga ikut berubah karena terdapat address-of dan untuk dereference terdapat lihat buku detail saat kita mau mencari buku yang kita mau inputan tersebut mendapatkan alamat dari nilai yang kita cari setelah itu akan dieksekusi mengambil nilai aslinya dari alamat tersebut

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. **Lihat Buku(Filter Harga)** → Melihat Buku dengan harga tertinggi ke terendah
 6. **Cari Buku** → Mencari Buku dengan input nama buku
 7. **Sort Buku(Addres of)** → Mengurutkan harga ke terendah ke terbesar
 8. **Lihat Detail Buku (Dereference)** → Melihat buku yang ingin dicari secara detail
 9. **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. **Lihat Buku Berdasarkan harga** → Menampilkan daftar buku sesuai harga yang diinginkan user
3. **Cari Buku** → Menampilkan buku yang dicari oleh user
4. **Beli Buku** → Pembeli dapat memasukkan nomor buku yang ingin dibeli. Setelah berhasil, sistem akan menampilkan pesan konfirmasi.

5. **Lihat Detail Buku (Dereference)** → Melihat buku yang ingin dicari secara detail
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

A. Deklarasi Fungsi Address-of & Dereference

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

Source Code:

```
void sortBukuByHarga(Buku daftar_buku[], int &jumlah_buku);  
void lihatDetailBuku(Buku *buku_ptr);
```

B. Address-of

Penerapan address of pada sort buku berdasarkan harga

Source code

```
void sortBukuByHarga(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 (stoi(daftar_buku[j].harga) > stoi(daftar_buku[j + 1].harga))  
            {  
                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  
harga (terendah-tertinggi)!" << endl;  
    cout << "Jumlah buku setelah pengurutan: " << jumlah_buku << endl;  
    tampilkanBuku(daftar_buku, jumlah_buku);  
}
```

C. Dereference

Penerapan dereference pada lihat detail buku

Source code

```
void lihatDetailBuku(Buku *buku_ptr) {
    cout << "\n==== Detail Buku =====> endl;
    cout << "Judul    : " << buku_ptr->judul << endl;
    cout << "Harga    : Rp " << buku_ptr->harga << endl;
    cout << "Penulis   : " << buku_ptr->penulis << endl;
    cout << "\nDetail dengan operator dereference (*)" << endl;
    cout << "Judul buku: " << (*buku_ptr).judul << endl;
    cout << "=====> endl;
}
```

D. Function Fitur CRUD ADMIN

kode ini potongan dari menu Function CRUD ADMIN

Source code

```
void menuAdmin(Buku daftar_buku[], int &jumlah_buku) {
    while (true) {
        cout << "\nSelamat Datang Admin, silahkan pilih menu yang anda inginkan!";
        cout << "\n1. Tambah Buku\n2. Edit Buku\n3. Hapus Buku\n4. Lihat Buku\n";
        cout << "5. Lihat Buku (Filter Harga)\n6. Cari Buku\n7. Sort Buku (Address-of)\n";
        cout << "8. Lihat Detail Buku (Dereference)\n9. Logout\n\nPilih opsi: ";

        int pilihan = getNumericInput();

        if (pilihan == -1) {
            cout << "Pilihan tidak valid! Harap masukkan angka." << endl;
            continue;
        }

        switch (pilihan) {
            case 1: tambahBuku(daftar_buku, jumlah_buku); break;
            case 2: editBuku(daftar_buku, jumlah_buku); break;
            case 3: hapusBuku(daftar_buku, jumlah_buku); break;
            case 4: tampilkanBuku(daftar_buku, jumlah_buku); break;
            case 5: {
                cout << "Masukkan harga minimum: ";
                int min = getNumericInput();
                if (min == -1) {
                    cout << "Input tidak valid! Harap masukkan angka." << endl;
                    continue;
                }

                cout << "Masukkan harga maksimum: ";
                int max = getNumericInput();
                if (max == -1) {
                    cout << "Input tidak valid! Harap masukkan angka." << endl;
                    continue;
                }

                tampilkanBuku(daftar_buku, jumlah_buku, min, max);
                break;
            }
            case 6: {
                string judul;
```

```

        cout << "Masukkan judul buku yang dicari: ";
        cin.ignore(0);
        getline(cin, judul);

        if (isEmptyOrWhitespace(judul)) {
            cout << "Judul tidak boleh kosong!" << endl;
            continue;
        }

        int hasil = cariBuku(daftar_buku, jumlah_buku, judul);
        if (hasil != -1) {
            cout << "Buku ditemukan pada indeks " << hasil+1 <<
endl;

            cout << "Judul: " << daftar_buku[hasil].judul << endl;
            cout << "Harga: " << daftar_buku[hasil].harga << endl;
            cout << "Penulis: " << daftar_buku[hasil].penulis <<
endl;

        } else {
            cout << "Buku dengan judul '" << judul << "' tidak
ditemukan." << endl;
        }
        break;
    }
    case 7: sortBukuByHarga(daftar_buku, jumlah_buku); break;
    case 8: {
        cout << "Masukkan nomor buku yang ingin dilihat detailnya:
";

        int index = getNumericInput();

        if (index == -1) {
            cout << "Input tidak valid! Harap masukkan angka." <<
endl;

            continue;
        }

        if (index > 0 && index <= jumlah_buku) {
            lihatDetailBuku(&daftar_buku[index-1]);
        } else {
            cout << "Nomor buku tidak valid!" << endl;
        }
        break;
    }
    case 9: return;
    default: cout << "Pilihan tidak valid! Harap masukkan angka
1-9." << endl;
        }
    }
}

```

E. Function Fitur User Dapat Memilih Menu

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

Source code

```
void menuPembeli(Buku daftar_buku[], int jumlah_buku) {
    while (true) {
        cout << "\nSelamat datang di aplikasi kami!";
        cout << "\n1. Lihat Buku\n2. Lihat Buku (Filter Harga)\n3. Cari Buku\n4. Beli Buku\n";
        cout << "5. Lihat Detail Buku (Dereference)\n6. Logout\n\nPilih opsi: ";

        int pilihan = getNumericInput();

        if (pilihan == -1) {
            cout << "Pilihan tidak valid! Harap masukkan angka." << endl;
            continue;
        }

        switch (pilihan) {
            case 1: tampilkanBuku(daftar_buku, jumlah_buku); break;
            case 2: {
                cout << "Masukkan harga minimum: ";
                int min = getNumericInput();
                if (min == -1) {
                    cout << "Input tidak valid! Harap masukkan angka." << endl;
                    continue;
                }

                cout << "Masukkan harga maksimum: ";
                int max = getNumericInput();
                if (max == -1) {
                    cout << "Input tidak valid! Harap masukkan angka." << endl;
                    continue;
                }

                tampilkanBuku(daftar_buku, jumlah_buku, min, max);
                break;
            }
            case 3: {
                string judul;
                cout << "Masukkan judul buku yang dicari: ";
                cin.ignore(0);
                getline(cin, judul);

                if (isEmptyOrWhitespace(judul)) {
                    cout << "Judul tidak boleh kosong!" << endl;
                    continue;
                }
            }
        }
    }
}
```

```

        int hasil = cariBuku(daftar_buku, jumlah_buku, judul);
        if (hasil != -1) {
            cout << "Buku ditemukan pada indeks " << hasil+1 <<
endl;

            cout << "Judul: " << daftar_buku[hasil].judul << endl;
            cout << "Harga: " << daftar_buku[hasil].harga << endl;
            cout << "Penulis: " << daftar_buku[hasil].penulis <<
endl;

            } else {
                cout << "Buku dengan judul '" << judul << "' tidak
ditemukan." << endl;
            }
            break;
        }
        case 4: beliBuku(daftar_buku, jumlah_buku); break;
        case 5: {
            cout << "Masukkan nomor buku yang ingin dilihat detailnya:
";

            int index = getNumericInput();

            if (index == -1) {
                cout << "Input tidak valid! Harap masukkan angka." <<
endl;

                continue;
            }

            if (index > 0 && index <= jumlah_buku) {
                lihatDetailBuku(&daftar_buku[index-1]);
            } else {
                cout << "Nomor buku tidak valid!" << endl;
            }
            break;
        }
        case 6: return;
        default: cout << "Pilihan tidak valid! Harap masukkan angka
1-6." << endl;
    }
}
}

```


4. Uji Coba dan Hasil Output

Dalam pengujian hanya menguji pada penerapan yang baru seperti Address-of dan Dereference

4.1 Uji Coba

1. Pengujian Sort Buku Berdasarkan Harga (Address-of)

input : Sort buku berdasarkan harga rendah ke tinggi

output yang diharapkan : Berhasil Menampilkan Buku dengan harga terendah ke terbesar

2. Pengujian Lihat Detail Buku (Dereference)

input : Mencari buku yang diinginkan

output yang diharapkan : Berhasil Menampilkan buku yang dicari

4.2 Hasil Output

```
Selamat Datang Admin, silahkan pilih menu yang anda inginkan!
1. Tambah Buku
2. Edit Buku
3. Hapus Buku
4. Lihat Buku
5. Lihat Buku (Filter Harga)
6. Cari Buku
7. Sort Buku (Address-of)
8. Lihat Detail Buku (Dereference)
9. Logout

Pilih opsi: 7
Total 3 buku telah diurutkan berdasarkan harga (terendah-tertinggi)!
Jumlah buku setelah pengurutan: 3

Daftar Buku:
1. Naruto - 45000 - Masashi Kishimoto
2. One Piece - 50000 - Eiichiro Oda
3. Dragon Ball - 55000 - Akira Toriyama
```

Gambar 4.1.1

```
Selamat Datang Admin, silahkan pilih menu yang anda inginkan!
1. Tambah Buku
2. Edit Buku
3. Hapus Buku
4. Lihat Buku
5. Lihat Buku (Filter Harga)
6. Cari Buku
7. Sort Buku (Address-of)
8. Lihat Detail Buku (Dereference)
9. Logout

Pilih opsi: 8
Masukkan nomor buku yang ingin dilihat detailnya: 2

===== Detail Buku =====
Judul   : One Piece
Harga   : Rp 50000
Penulis : Eiichiro Oda

Detail dengan operator dereference (*)
Judul buku: One Piece
=====
```

Gambar 4.1.2

