

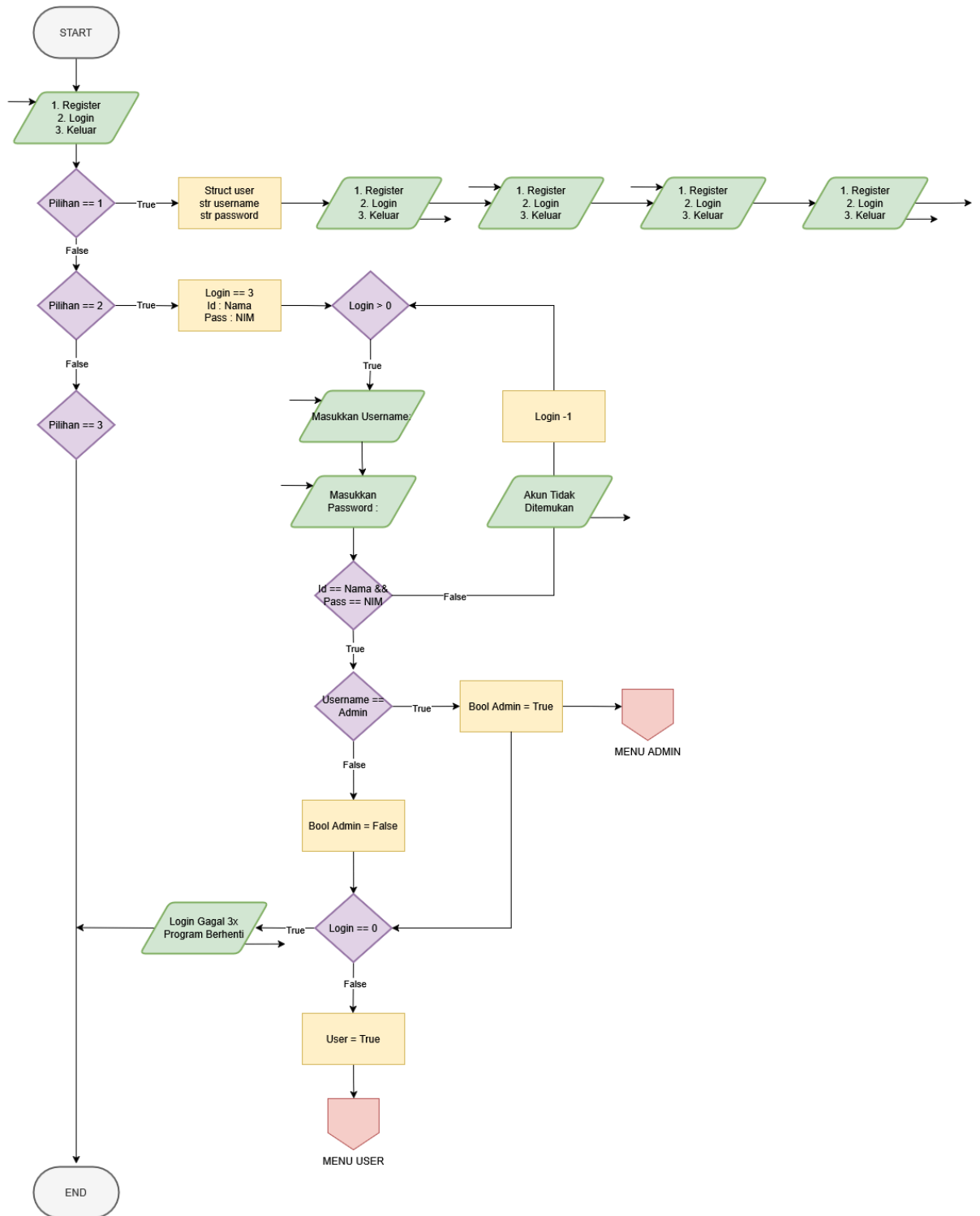
**LAPORAN PRAKTIKUM**  
**POSTTEST (6)**  
**ALGORITMA PEMROGRAMAN LANJUT**

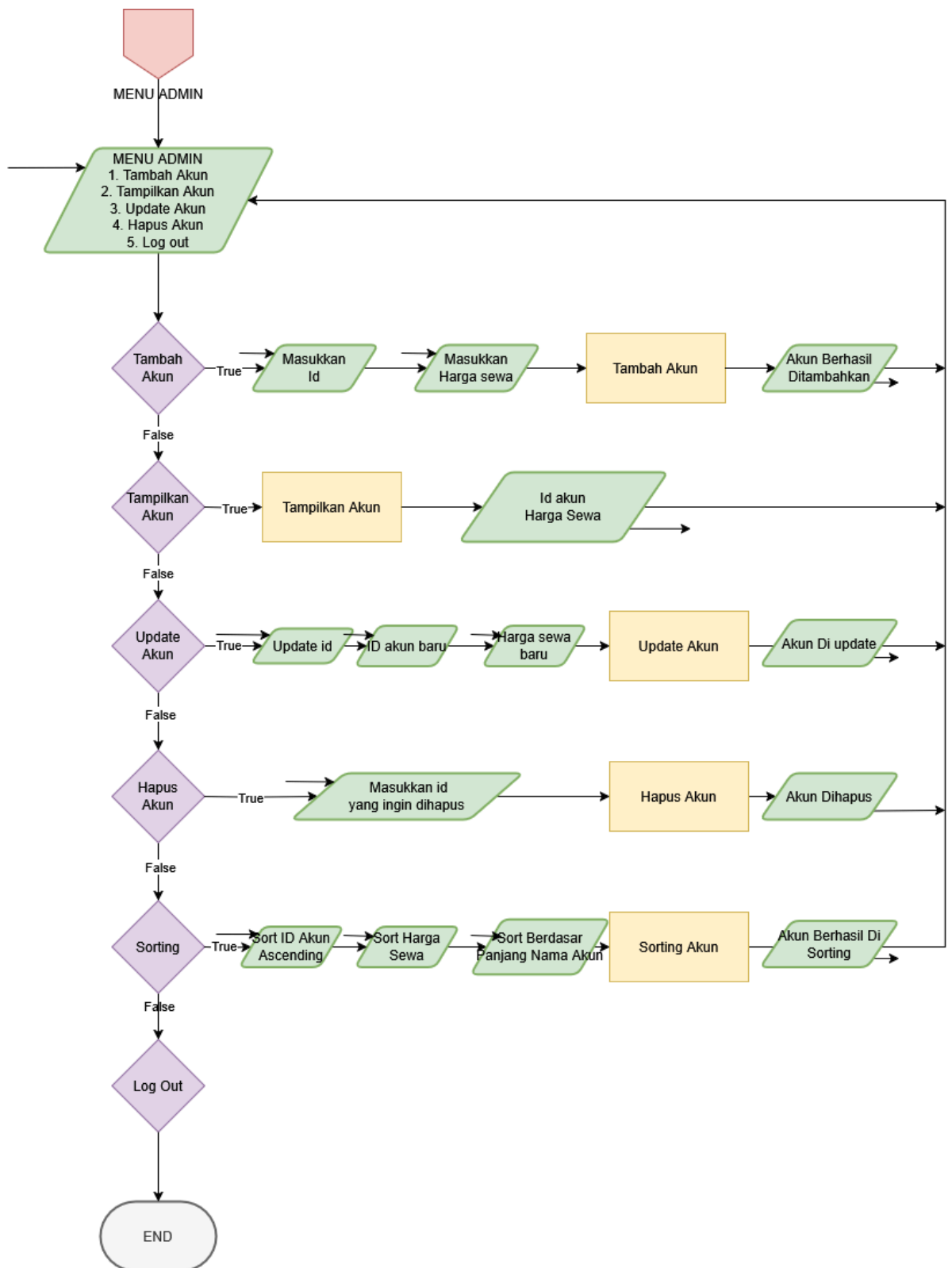


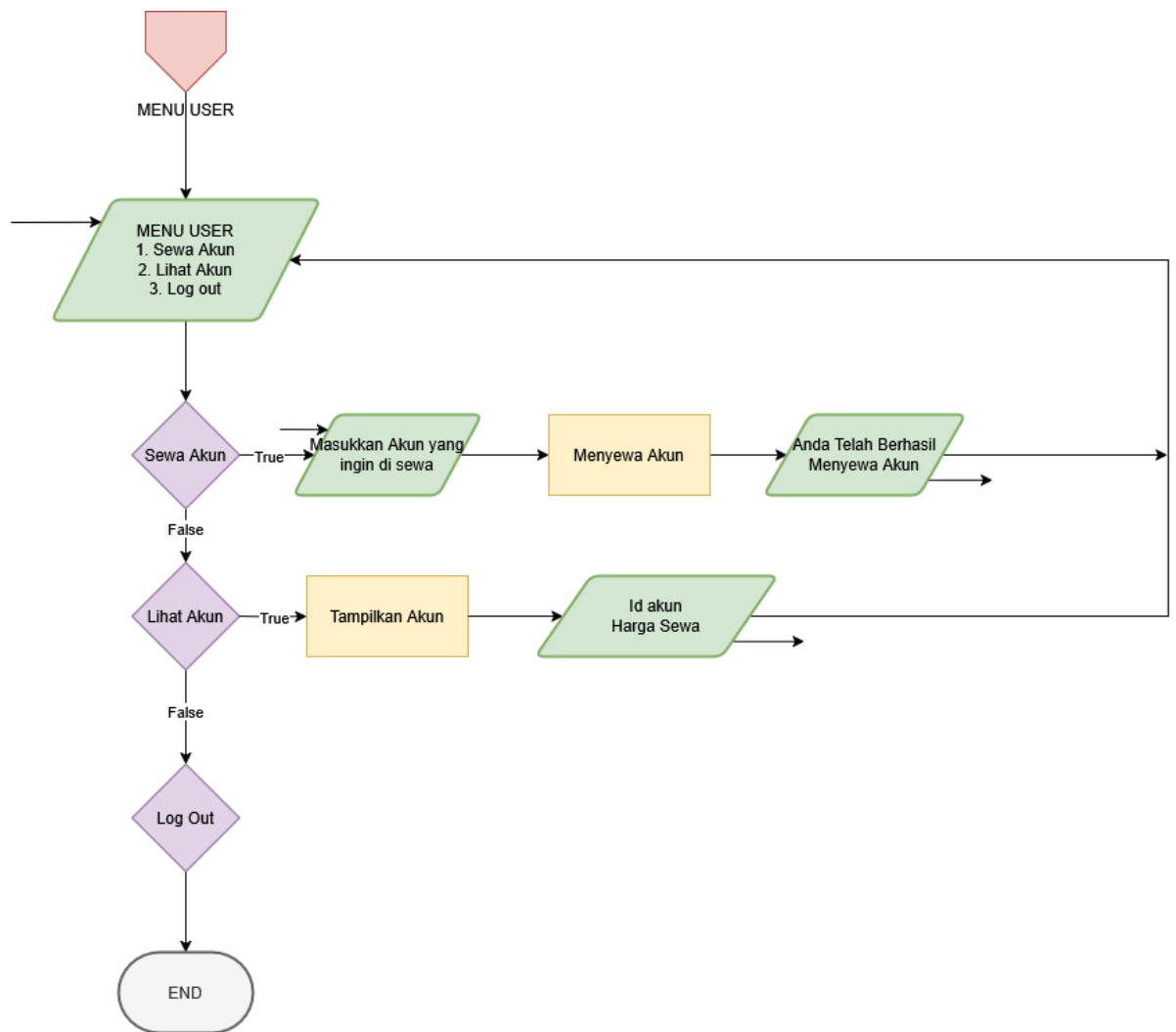
**Disusun oleh:**  
**Benyamin Haqie (2409106081)**  
**Kelas B2 '24**

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

# 1. Flowchart







## 2. Analisis Program

### 2.1 Deskripsi Singkat Program

Program ini merupakan aplikasi manajemen penyewaan akun PUBG Mobile yang memungkinkan admin dan pengguna untuk berinteraksi dengan sistem. Admin memiliki akses untuk mengelola akun yang tersedia, sedangkan pengguna dapat mendaftar, login, melihat daftar akun, serta menyewa akun yang tersedia. Sistem dilengkapi dengan fitur login dengan batasan tiga kali percobaan gagal sebelum program berhenti.

### 2.2 Penjelasan Alur & Algoritma

Berikut adalah penjelasan alur dan algoritma dari program penyewaan akun PUBG Mobile.

#### LOGIN DAN REGISTRASI:

1. Pengguna baru dapat melakukan registrasi dengan memasukkan username dan password.
2. Setelah registrasi, pengguna dapat login dengan memasukkan username dan password yang valid.
3. Jika login gagal sebanyak tiga kali berturut-turut, program akan berhenti.
4. Admin dapat login menggunakan Akses khusus dan memiliki akses lebih luas dibandingkan pengguna biasa.

#### **MENU UTAMA:**

- **Admin Menu:**

1. **Tambah Akun:** Menambahkan ID akun beserta harga sewa ke dalam daftar.
2. **Lihat Akun:** Menampilkan semua akun yang tersedia.
3. **Update Akun:** Mengubah ID atau harga akun yang sudah ada.
4. **Hapus Akun:** Menghapus akun dari daftar.
5. **Logout:** Keluar dari sistem.

- **User Menu:**

1. **Sewa Akun:** Pengguna dapat memilih akun dari daftar dan menyewanya.
2. **Lihat Akun:** Pengguna dapat melihat daftar akun yang tersedia untuk disewa.
3. **Logout:** Keluar dari sistem.

#### **Algoritma Singkat:**

1. **Login/Register:**
  - Pengguna baru dapat mendaftar.
  - Login dicek berdasarkan username dan password.
  - Jika gagal tiga kali, program berhenti.
2. **Menu Admin:**
  - Admin dapat menambah, mengubah, menampilkan, dan menghapus akun.
3. **Menu User:**
  - User hanya bisa melihat daftar akun dan menyewa akun tertentu.
4. **Logout:**
  - Admin atau user keluar dari sistem.

### **2.3 Penggunaan pointer pada sistem**

Pointer adalah sebuah variabel khusus dalam bahasa pemrograman C++ yang digunakan untuk menyimpan alamat memori dari suatu data atau variabel lain. Alamat memori ini merujuk ke lokasi di mana data tersebut disimpan di dalam memori komputer. Dalam program ini, pointer digunakan untuk mengakses dan memodifikasi data akun dan user secara langsung melalui alamat memorinya. Contohnya, saat memperbarui akun, digunakan pointer seperti “**Akun\* akunToUpdate = &daftarAkun[index - 1];**” agar data dapat diubah langsung tanpa menyalin seluruh objek. Dengan penggunaan pointer, program menjadi lebih efisien dan terstruktur.

### 2.3 Penerapan Sorting pada sistem

Program ini telah menerapkan sistem sorting, yang dimana sistem ini mempunyai 3 metode yang sesuai ketentuan yang telah di berikan yaitu :

1. Sorting Ascending (Alfabetik): yang dimana dipergunakan untuk Menyusun id akun dari A ke Z menggunakan perbandingan string secara langsung.
2. Sorting Descending (Harga) : Kemudian Pada sorting yang kedua ini menggunakan sistem sorting descendingnya yang dimana sistemnya dipergunakan untuk menyusun harga sewa akun mulai dari yang tertinggi ke ter rendah.
3. Sorting Ascending (Panjang ID Akun) : Kemudian Pada sorting yang terakhir menggunakan ascending yang dimana sistemnya pada kali ini dapat melakukan pengurutan berdasar panjang nama id akun, yang dimana diurutkan mulai dari yang terpendek ke panjang.

## 3. Source Code

### A. Fitur Autentikasi

#### Source Code:

```
bool login(string username, string password, bool& isAdmin) {
    if (username == "Haci" && password == "081") {
        isAdmin = true;
        return true;
    }
    for (int i = 0; i < jumlahUser; i++) {
        User* user = &daftarUser[i];
        if (user->username == username && user->password == password) {
            isAdmin = false;
            return true;
        }
    }
    return false;
}
```

```

int main() {
    int pilihan;
    string username, password;
    bool isAdmin;
    int percobaanLogin = 0;

    do {
        cout << "\n===== MENU =====\n";
        cout << "1. Register User\n";
        cout << "2. Login\n";
        cout << "3. Keluar\n";
        cout << "Pilihan: ";
        cin >> pilihan;
        if (pilihan == 1) {
            if (jumlahUser < MAX_USER) {
                User* newUser = &daftarUser[jumlahUser];
                cout << "Masukkan Username: ";
                cin >> newUser->username;
                cout << "Masukkan Password: ";
                cin >> newUser->password;
                jumlahUser++;
                cout << "Registrasi berhasil!\n";
            } else {
                cout << "Pendaftaran user penuh!\n";
            }
        } else if (pilihan == 2) {
            while (percobaanLogin < 3) {
                cout << "\n===== LOGIN =====\n";
                cout << "Username: ";
                cin >> username;
                cout << "Password: ";
                cin >> password;
                if (login(username, password, isAdmin)) {
                    cout << "\nSelamat datang, " << username << endl;
                    if (isAdmin) cout << "(Aturlah bang, abangkan admin)\n";
                }
            }
        } else if (pilihan == 3) {
            cout << "Keluar\n";
        }
    } while (pilihan != 3);
}

```

## B. Fitur Menu Utama

### Source Code:

```

int main() {
    int pilihan;
    string username, password;
    bool isAdmin;

```

```

int percobaanLogin = 0;

do {
    cout << "\n===== MENU =====\n";
    cout << "1. Register User\n";
    cout << "2. Login\n";
    cout << "3. Keluar\n";
    cout << "Pilihan: ";
    cin >> pilihan;
    if (pilihan == 1) {
        if (jumlahUser < MAX_USER) {
            User* newUser = &daftarUser[jumlahUser];
            cout << "Masukkan Username: ";
            cin >> newUser->username;
            cout << "Masukkan Password: ";
            cin >> newUser->password;
            jumlahUser++;
            cout << "Registrasi berhasil!\n";
        } else {
            cout << "Pendaftaran user penuh!\n";
        }
    } else if (pilihan == 2) {
        while (percobaanLogin < 3) {
            cout << "\n===== LOGIN =====\n";
            cout << "Username: ";
            cin >> username;
            cout << "Password: ";
            cin >> password;
            if (login(username, password, isAdmin)) {
                cout << "\nSelamat datang, " << username << endl;
                if (isAdmin) cout << "(Aturlah bang, abangkan admin)\n";
                int menuPilihan;
                do {
                    cout << "\n===== MENU " << (isAdmin ? "ADMIN" :
"USER") << " =====\n";
                    if (isAdmin) {
                        cout << "1. Tambah Akun\n";
                        cout << "2. Lihat Akun\n";
                        cout << "3. Update Akun\n";
                        cout << "4. Hapus Akun\n";
                        cout << "5. Sorting Akun\n";
                        cout << "6. Logout\n";
                    } else {
                        cout << "1. Sewa Akun\n";
                        cout << "2. Lihat Akun\n";
                        cout << "3. Logout\n";
                    }
                }
                cout << "Pilihan: ";
            }
        }
    }
} while (percobaanLogin < 3);

```



```

        cin >> menuPilihan;
        if (isAdmin) {
            switch (menuPilihan) {
                case 1: tambahAkun(); break;
                case 2: tampilkanSemuaAkun(); break;
                case 3: updateAkun(); break;
                case 4: hapusAkun(); break;
                case 5: {
                    int sortPilihan;
                    cout << "\n=== Menu Sorting ===\n";
                    cout << "1. Sort ID Akun Ascending
(A-Z)\n";

                    cout << "2. Sort Harga Sewa
Descending\n";

                    cout << "3. Sort ID berdasarkan Panjang
Karakter\n";

                    cout << "Pilihan: ";
                    cin >> sortPilihan;
                    switch (sortPilihan) {
                        case 1: sortingIDAscending(); break;
                        case 2: sortingHargaDescending();

                        case 3: sortingIDPanjang(); break;
                        default: cout << "Pilihan sorting
break;

tidak valid!\n";

                    }
                    break;
                }
                case 6: break;
                default: cout << "Pilihan tidak valid!\n";
            }
            if (menuPilihan == 6) break;
        } else {
            switch (menuPilihan) {
                case 1: sewaAkun(); break;
                case 2: tampilkanSemuaAkun(); break;
                case 3: break;
                default: cout << "Pilihan tidak valid!\n";
            }
            if (menuPilihan == 3) break;
        }
    } while (true);
    break;
} else {
    cout << "Login gagal. Silakan coba lagi.\n";
    percobaanLogin++;
}
}

```

```

        if (percobaanLogin == 3) {
            cout << "Gagal login 3 kali. Program dihentikan.\n";
            return 0;
        }
    } else if (pilihan == 3) {
        cout << "Terima kasih. Program selesai.\n";
        break;
    } else {
        cout << "Pilihan tidak valid!\n";
    }
} while (true);

return 0;
}

```

### C. Menu Admin

#### Source Code:

```

void cetakAkun(Akun* akun) {
    cout << "ID Akun: " << akun->id << ", Harga Sewa: " << akun->harga << endl;
}

void tampilkanSemuaAkun(int index = 0) {
    if (index >= jumlahAkun) return;
    cout << index + 1 << ". ";
    cetakAkun(&daftarAkun[index]);
    tampilkanSemuaAkun(index + 1);
}

void sortingIDAscending() {
    for (int i = 0; i < jumlahAkun - 1; i++) {
        for (int j = i + 1; j < jumlahAkun; j++) {
            if (daftarAkun[i].id > daftarAkun[j].id) {
                swap(daftarAkun[i], daftarAkun[j]);
            }
        }
    }
    cout << "ID Akun berhasil disorting secara ASCENDING (A-Z)!\n";
}

void sortingHargaDescending() {
    for (int i = 0; i < jumlahAkun - 1; i++) {
        for (int j = i + 1; j < jumlahAkun; j++) {
            if (daftarAkun[i].harga < daftarAkun[j].harga) {

```

```

        swap(daftarAkun[i], daftarAkun[j]);
    }
}

}

cout << "Harga Sewa berhasil disorting secara DESCENDING!\n";
}

void sortingIDPanjang() {
    for (int i = 0; i < jumlahAkun - 1; i++) {
        for (int j = i + 1; j < jumlahAkun; j++) {
            if (daftarAkun[i].id.length() > daftarAkun[j].id.length()) {
                swap(daftarAkun[i], daftarAkun[j]);
            }
        }
    }
    cout << "ID Akun berhasil disorting berdasarkan PANJANG karakter!\n";
}

void tambahAkun() {
    if (jumlahAkun < MAX_AKUN) {
        Akun* newAkun = &daftarAkun[jumlahAkun];
        cout << "Masukkan ID Akun: ";
        cin.ignore();
        getline(cin, newAkun->id);
        cout << "Masukkan Harga Sewa: ";
        cin >> newAkun->harga;
        jumlahAkun++;
        cout << "Akun berhasil ditambahkan!\n";
    } else {
        cout << "Kapasitas akun penuh!\n";
    }
}

void updateAkun() {
    int index;
    tampilkanSemuaAkun();
    cout << "Masukkan nomor akun yang ingin diupdate: ";
    cin >> index;
    if (index > 0 && index <= jumlahAkun) {
        Akun* akunToUpdate = &daftarAkun[index - 1];
        cout << "Masukkan ID baru: ";
        cin.ignore();
        getline(cin, akunToUpdate->id);
        cout << "Masukkan Harga Sewa baru: ";
        cin >> akunToUpdate->harga;
        cout << "Akun berhasil diperbarui!\n";
    } else {
        cout << "Nomor akun tidak valid!\n";
    }
}

```

```

    }
}

void hapusAkun() {
    int index;
    tampilkanSemuaAkun();
    cout << "Masukkan nomor akun yang ingin dihapus: ";
    cin >> index;
    if (index > 0 && index <= jumlahAkun) {
        for (int i = index - 1; i < jumlahAkun - 1; i++) {
            daftarAkun[i] = daftarAkun[i + 1];
        }
        jumlahAkun--;
        cout << "Akun berhasil dihapus!\n";
    } else {
        cout << "Nomor akun tidak valid!\n";
    }
}

void sewaAkun() {
    int index;
    tampilkanSemuaAkun();
    cout << "Masukkan nomor akun yang ingin disewa: ";
    cin >> index;
    if (index > 0 && index <= jumlahAkun) {
        Akun* akunToSewa = &daftarAkun[index - 1];
        cout << "Anda telah menyewa akun " << akunToSewa->id << " dengan
harga " << akunToSewa->harga << "\n";
    } else {
        cout << "Nomor akun tidak valid!\n";
    }
}
}

```

#### D. Menu User

##### Source Code:

```

void sewaAkun() {
    int index;
    tampilkanSemuaAkun();
    cout << "Masukkan nomor akun yang ingin disewa: ";
    cin >> index;
    if (index > 0 && index <= jumlahAkun) {
        Akun* akunToSewa = &daftarAkun[index - 1];
        cout << "Anda telah menyewa akun " << akunToSewa->id << " dengan
harga " << akunToSewa->harga << "\n";
    } else {

```

```

        cout << "Nomor akun tidak valid!\n";
    }
}

bool login(string username, string password, bool& isAdmin) {
    if (username == "Haci" && password == "081") {
        isAdmin = true;
        return true;
    }
    for (int i = 0; i < jumlahUser; i++) {
        User* user = &daftarUser[i];
        if (user->username == username && user->password == password) {
            isAdmin = false;
            return true;
        }
    }
    return false;
}

```

## 4. Uji Coba dan Hasil Output

### 4.1 Uji Coba

Berikut adalah beberapa skenario uji coba:

1. Registrasi pengguna baru
2. Login dengan kredensial yang benar
3. Login gagal sebanyak 3 kali
4. Admin memilih menu CRUD
5. User menyewa akun
6. Sorting
7. Logout:

### 4.2 Hasil Output

**Output Register Dan Login :**

```
===== MENU =====
1. Register User
2. Login
3. Keluar
Pilihan: 1
Masukkan Username: Fathir
Masukkan Password: Cyntrexdt
Registrasi berhasil! Silakan login.
===== MENU =====
1. Register User
2. Login
3. Keluar
Pilihan: 2
===== LOGIN =====
Masukkan Username: Fathir
Masukkan Password: Cyntrexdt
Halo, Selamat datang di lapak peminjaman akun
===== MENU USER =====
1. Sewa Akun
2. Lihat Akun
5. Logout
Pilihan: █
```

**Output Login Admin :**

```
===== MENU =====  
1. Register User  
2. Login  
3. Keluar  
Pilihan: 2  
===== LOGIN =====  
Masukkan Username: Hacı  
Masukkan Password: 081  
Halo, Selamat datang di lapak peminjaman akun  
Awat Ada ATMINN!  
===== MENU ADMIN =====  
1. Tambah Akun  
2. Lihat Akun  
3. Update Akun  
4. Hapus Akun  
5. Logout  
Pilihan: 
```

**Output Admin Tambah Akun:**

```
===== MENU ADMIN =====  
1. Tambah Akun  
2. Lihat Akun  
3. Update Akun  
4. Hapus Akun  
5. Logout  
Pilihan: 1  
Masukkan ID Akun: AyamSayur  
Masukkan Harga Sewa: 115  
Akun berhasil ditambahkan!
```

**Output Admin Menampilkan Akun :**

```
===== MENU ADMIN =====
1. Tambah Akun
2. Lihat Akun
3. Update Akun
4. Hapus Akun
5. Logout
Pilihan: 2
Daftar Akun:
1. ID: Reki, Harga Sewa: 150
2. ID: Lyev, Harga Sewa: 200
3. ID: Alex Killer Pro, Harga Sewa: 250
4. ID: AyamSayur, Harga Sewa: 115
```

**Output Admin Update Akun:**

```
===== MENU ADMIN =====
1. Tambah Akun
2. Lihat Akun
3. Update Akun
4. Hapus Akun
5. Logout
Pilihan: 3
Masukkan nomor akun yang ingin diupdate: 3
Masukkan ID baru: SahabatInformatika
Masukkan Harga Sewa baru: 160
Akun berhasil diperbarui!
===== MENU ADMIN =====
1. Tambah Akun
2. Lihat Akun
3. Update Akun
4. Hapus Akun
5. Logout
Pilihan: 2
Daftar Akun:
1. ID: Reki, Harga Sewa: 150
2. ID: Lyev, Harga Sewa: 200
3. ID: SahabatInformatika, Harga Sewa: 160
4. ID: AyamSayur, Harga Sewa: 115
```

**Output Admin Hapus Akun:**



```
===== MENU ADMIN =====
1. Tambah Akun
2. Lihat Akun
3. Update Akun
4. Hapus Akun
5. Logout
Pilihan: 4
Masukkan nomor akun yang ingin dihapus: 4
Akun berhasil dihapus!
===== MENU ADMIN =====
1. Tambah Akun
2. Lihat Akun
3. Update Akun
4. Hapus Akun
5. Logout
Pilihan: 2
Daftar Akun:
1. ID: Reki, Harga Sewa: 150
2. ID: Lyev, Harga Sewa: 200
3. ID: SahabatInformatika, Harga Sewa: 160
```

**Output User Sewa Akun:**

```
Halo, Selamat datang di lapak peminjaman akun
===== MENU USER =====
1. Sewa Akun
2. Lihat Akun
5. Logout
Pilihan: 1
Masukkan nomor akun yang ingin disewa: 3
Anda telah menyewa akun SahabatInformatika dengan harga 160
```

**Output Sorting :**

```

1. ID Akun: Reki, Harga Sewa: 200
2. ID Akun: Lyev, Harga Sewa: 100
3. ID Akun: Alex Killer Pro, Harga Sewa: 500

===== MENU ADMIN =====
1. Tambah Akun
2. Lihat Akun
3. Update Akun
4. Hapus Akun
5. Sorting Akun
6. Logout
Pilihan: 5

=== Menu Sorting ===
1. Sort ID Akun Ascending (A-Z)
2. Sort Harga Sewa Descending
3. Sort ID berdasarkan Panjang Karakter
Pilihan: 1
ID Akun berhasil disorting secara ASCENDING (A-Z)

===== MENU ADMIN =====
1. Tambah Akun
2. Lihat Akun
3. Update Akun
4. Hapus Akun
5. Sorting Akun
6. Logout
Pilihan: 2
1. ID Akun: Alex Killer Pro, Harga Sewa: 500
2. ID Akun: Lyev, Harga Sewa: 100
3. ID Akun: Reki, Harga Sewa: 200

```

Pada Output diatas, Program mengurutkan akun berdasarkan huruf Alphabet.

```
1. ID Akun: Alex Killer Pro, Harga Sewa: 500
2. ID Akun: Reki, Harga Sewa: 200
3. ID Akun: Lyev, Harga Sewa: 100
4. ID Akun: ayam sayur , Harga Sewa: 1000

===== MENU ADMIN =====
1. Tambah Akun
2. Lihat Akun
3. Update Akun
4. Hapus Akun
5. Sorting Akun
6. Logout
Pilihan: 5

=== Menu Sorting ===
1. Sort ID Akun Ascending (A-Z)
2. Sort Harga Sewa Descending
3. Sort ID berdasarkan Panjang Karakter
Pilihan: 2
Harga Sewa berhasil disorting secara DESCENDING!

===== MENU ADMIN =====
1. Tambah Akun
2. Lihat Akun
3. Update Akun
4. Hapus Akun
5. Sorting Akun
6. Logout
Pilihan: 2
1. ID Akun: ayam sayur , Harga Sewa: 1000
2. ID Akun: Alex Killer Pro, Harga Sewa: 500
3. ID Akun: Reki, Harga Sewa: 200
4. ID Akun: Lyev, Harga Sewa: 100
```

Output diatas merupakan sorting yang mengurutkan, harga termahal sampai ke termurah.

```

1. ID Akun: ayam sayur , Harga Sewa: 1000
2. ID Akun: Alex Killer Pro, Harga Sewa: 500
3. ID Akun: Reki, Harga Sewa: 200
4. ID Akun: Lyev, Harga Sewa: 100

===== MENU ADMIN =====
1. Tambah Akun
2. Lihat Akun
3. Update Akun
4. Hapus Akun
5. Sorting Akun
6. Logout
Pilihan: 5

=== Menu Sorting ===
1. Sort ID Akun Ascending (A-Z)
2. Sort Harga Sewa Descending
3. Sort ID berdasarkan Panjang Karakter
Pilihan: 3
ID Akun berhasil disorting berdasarkan PANJANG karakter!

===== MENU ADMIN =====
1. Tambah Akun
2. Lihat Akun
3. Update Akun
4. Hapus Akun
5. Sorting Akun
6. Logout
Pilihan: 2
1. ID Akun: Reki, Harga Sewa: 200
2. ID Akun: Lyev, Harga Sewa: 100
3. ID Akun: ayam sayur , Harga Sewa: 1000
4. ID Akun: Alex Killer Pro, Harga Sewa: 500

```

Output diatas merupakan sorting yang mengurutkan nama id mulai dari terpendek sampai terpanjang.

## 5. GIT

```
Lenovo@lyev MINGW64 /c/APL/project-git/praktikum-apl (main)
$ git add .

Lenovo@lyev MINGW64 /c/APL/project-git/praktikum-apl (main)
$ git commit -m "hwhw"
[main 1a0ece3] hwhw
3 files changed, 255 insertions(+)
create mode 100644 praktikum-apl/post-test/post-test-6/2409106081_BenyaminHaqie_Posttest6.cpp
create mode 100644 praktikum-apl/post-test/post-test-6/2409106081_BenyaminHaqie_Posttest6.exe

Lenovo@lyev MINGW64 /c/APL/project-git/praktikum-apl (main)
$ git push origin main
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 20 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 680.09 KiB | 6.80 MiB/s, done.
Total 9 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/haqi2/praktikum-apl.git
96d79d6..1a0ece3  main -> main
```

Git Add : Git add digunakan untuk merubah status file, file yang belum di add maka akan memiliki status Untracked. dan akan berubah menjadi staged jika sudah di add ke repository tetapi belum di-commit.

Git Commit : Git commit berfungsi untuk menyimpan perubahan file di repository

Git Push : git push -u origin main digunakan untuk mengupload perubahan yang ada pada repository lokal ke repository github pada branch main. 10