

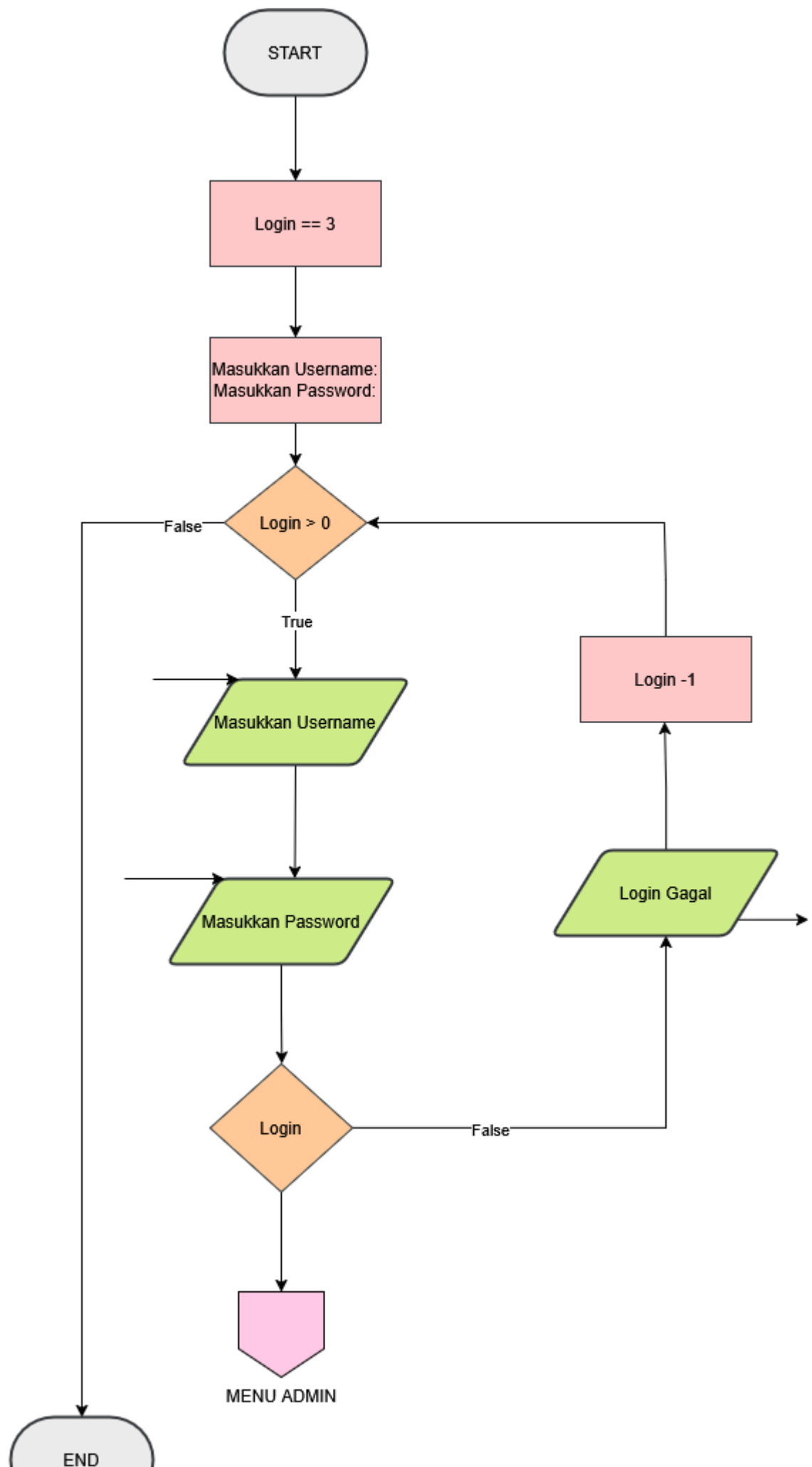
LAPORAN PRAKTIKUM
POSTTEST (2)
ALGORITMA PEMROGRAMAN LANJUT

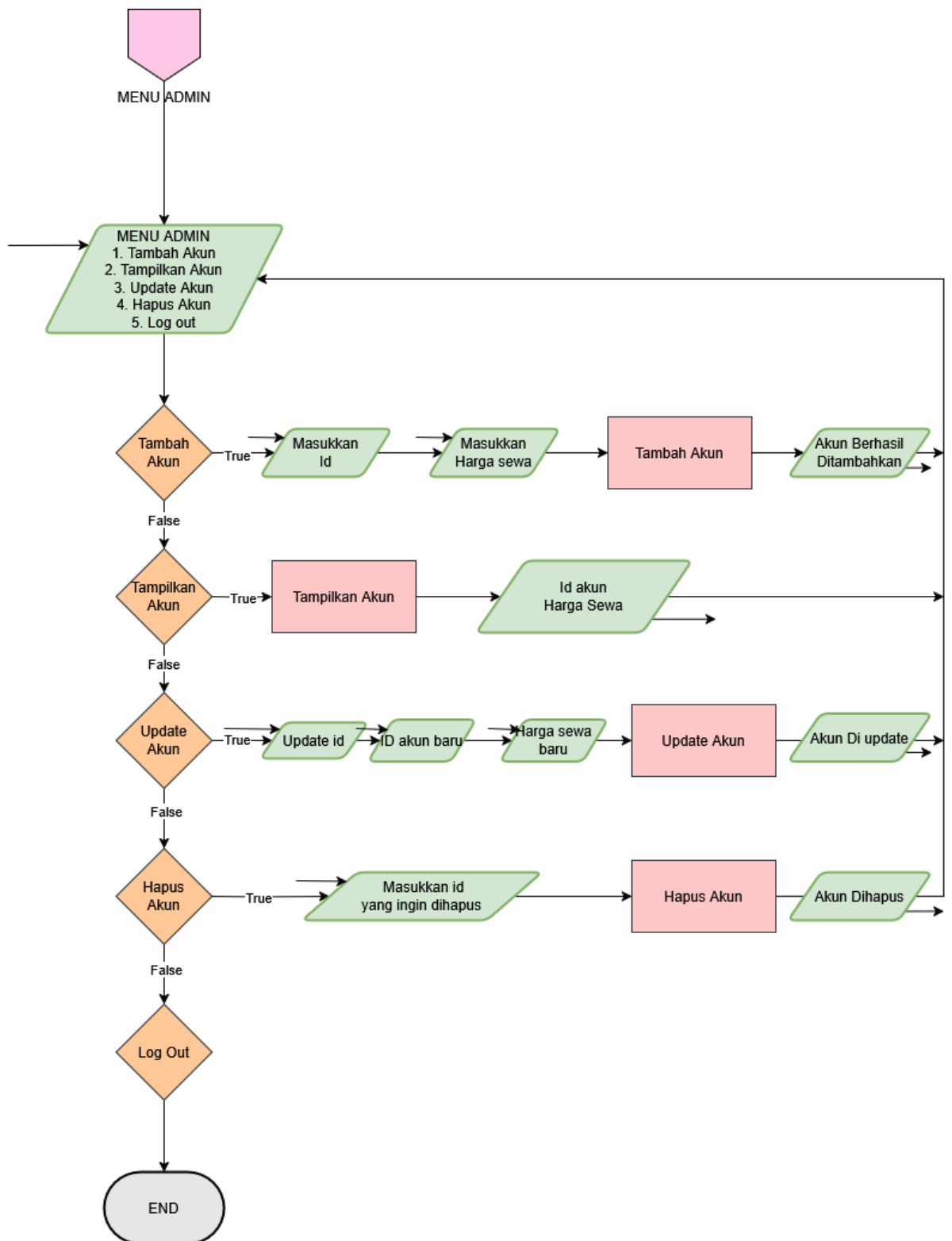


Disusun oleh:
Benyamin Haqie 24091060081
Kelas B2 '24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart





2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini merupakan aplikasi sederhana untuk mengelola penyewaan akun PUBG Mobile. Admin dapat menambahkan, menampilkan, memperbarui, dan menghapus akun yang

tersedia untuk disewa. Sistem juga dilengkapi fitur login untuk memastikan hanya admin yang bisa mengakses menu utama. pada program ini hanya menampilkan proses yang dilakukan oleh admin.

2.2 Penjelasan Alur & Algoritma

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

1. **LOGIN:** Yang pertama admin diarahkan untuk memasukkan username dan password yang benar pada menu login. Jika salah 3 kali, program akan berhenti.
2. **MENU UTAMA :** Kemudian pada menu utama ini admin dapat memilih opsi Tambah Akun, Lihat Akun, Update Akun, Hapus Akun, atau Logout.
3. **CRUD (Crate, Read, Update, Delete):**
 - **Tambah Akun :** Memasukkan ID dan harga akun ke dalam daftar.
 - **Lihat Akun :** Menampilkan semua akun yang tersedia.
 - **Update Akun :** Mengubah ID atau harga akun yang sudah ada.
 - **Hapus Akun :** Menghapus akun dari daftar.
4. **Logout :** Keluar dari sistem dan program selesai.

Algoritma:

Dibawah ini merupakan penjelasan algoritma singkat:

1. Login : Masukkan username dan password kemudian dilakukan cek username dan password. Jika benar, lanjut ke menu jika salah 3x keluar program.
2. Menu Admin : Tampilkan pilihan dan jalankan fitur sesuai input admin.
3. CRUD : Memproses data akun sesuai perintah.
4. Logout : Admin keluar dan program berhenti.

3. Source Code

A. Fitur Tambah Data

Source Code:

```
int TambahAkun(string id, double harga) {
```

```

    if (jumlahakun < MAX_AKUN) {
        idakun[jumlahakun] = id;
        hargasewa[jumlahakun] = harga;
        system("cls");
        jumlahakun++;
        cout << "Akun dengan ID " << id << " berhasil ditambahkan dengan harga
sewa " << harga << "!" << endl;
        return 1;
    } else {
        cout << "Penyimpanan akun penuh!" << endl;
        return 0;
    }
}

```

B. Fitur Tampilkan Data

Source Code:

```

int TampilkanAkun() {
    if (jumlahakun == 0) {
        cout << "Tidak ada akun yang tersedia." << endl;
        return 0;
    } else {
        cout << "\n===== " << endl;
        cout << "        Daftar Akun Tersedia        " << endl;
        cout << "===== " << endl;
        for (int i = 0; i < jumlahakun; i++) {
            cout << i + 1 << ". ID Akun: " << idakun[i] << ", Harga Sewa: " <<
hargasewa[i] << endl;
        }
        cout << "===== " << endl;
        return 1;
    }
}

```

C. Fitur Update Data

Source Code:

```

int UpdateAkun(string id, string newId, double harga) {
    bool found = false;
    for (int i = 0; i < jumlahakun; i++) {
        if (idakun[i] == id) {
            idakun[i] = newId;
            hargasewa[i] = harga;
            found = true;
            cout << "Akun dengan ID " << id << " berhasil diupdate menjadi ID "
<< newId << " dengan harga sewa baru " << harga << "!" << endl;
            break;
        }
    }
    if (!found) {
        cout << "ID akun tidak ditemukan." << endl;
        return 0;
    }
    return 1;
}

```

D. Fitur Hapus Data

Source Code:

```

int HapusAkun(string id) {
    bool found = false;
    for (int i = 0; i < jumlahakun; i++) {
        if (idakun[i] == id) {
            for (int j = i; j < jumlahakun - 1; j++) {
                idakun[j] = idakun[j + 1];
                hargasewa[j] = hargasewa[j + 1];
            }
            jumlahakun--;
            found = true;
            cout << "Akun dengan ID " << id << " berhasil dihapus!" << endl;
            return 1;
        }
    }
    if (!found) {
        cout << "ID akun tidak ditemukan." << endl;
        return 0;
    }
    return 0;
}

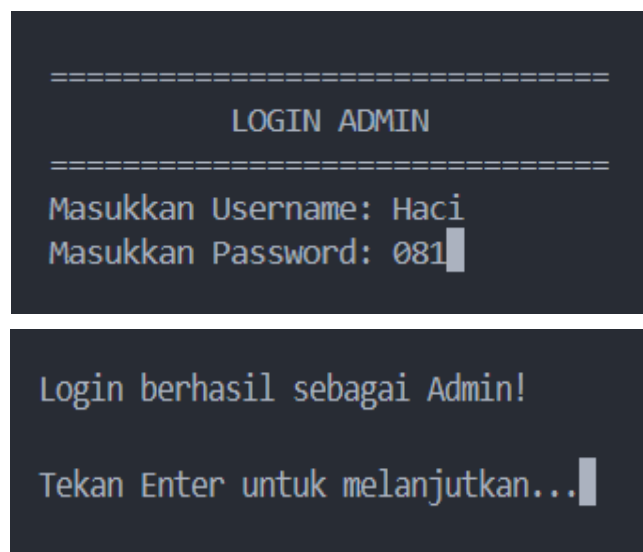
```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

1. Login memasukkan username dan password dengan benar
2. Login salah memasukkan username atau password sebanyak 3 kali
3. Admin memilih menu
5. Admin memproses pendataan akun
6. Log out

4.2 Hasil Output



```
=====
                LOGIN ADMIN
=====
Masukkan Username: Haci
Masukkan Password: 081

Login berhasil sebagai Admin!

Tekan Enter untuk melanjutkan...
```

Gambar 4.2.1 Program login


```

=====
                LOGIN ADMIN
=====
Masukkan Username: melchi
Masukkan Password: 1
Login gagal! Coba lagi.

Tekan Enter untuk melanjutkan...
=====
                LOGIN ADMIN
=====
Masukkan Username: Dzaki
Masukkan Password: 2
Login gagal! Coba lagi.

Tekan Enter untuk melanjutkan...
=====
                LOGIN ADMIN
=====
Masukkan Username: Melchi java script
Masukkan Password: 1
Login gagal! Coba lagi.

Tekan Enter untuk melanjutkan...
Anda telah gagal login 3 kali. Program berakhir.

```

Gambar 4.2.2 Program login jika gagal

```

=====
                MENU ADMIN
=====
1. Masukkan Id Akun
2. Tampilkan Daftar Akun
3. Update Akun
4. Hapus Akun
5. LOGOUT
=====
Pilih menu (1-5): 1
Masukkan ID Akun: ayam sayur
Masukkan Harga Sewa: 500k
Akun dengan ID ayam sayur berhasil ditambahkan dengan harga sewa 500!

```

Gambar 4.2.3 Program jika menambah data

```
=====
                        MENU ADMIN
=====
1. Masukkan Id Akun
2. Tampilkan Daftar Akun
3. Update Akun
4. Hapus Akun
5. LOGOUT
=====
Pilih menu (1-5): 2

=====
                        Daftar Akun Tersedia
=====
1. ID Akun: Reki, Harga Sewa: 150
2. ID Akun: Lyev, Harga Sewa: 200
3. ID Akun: Alex Killer Pro, Harga Sewa: 250
4. ID Akun: ayam sayur, Harga Sewa: 500
=====
```

Gambar 4.2.4 Program menampilkan data

```
=====
                        MENU ADMIN
=====
1. Masukkan Id Akun
2. Tampilkan Daftar Akun
3. Update Akun
4. Hapus Akun
5. LOGOUT
=====
Pilih menu (1-5): 3
Masukkan ID Akun yang ingin diupdate: Alex Killer Pro
Masukkan ID Akun Baru: Alex Biasa Aja
Masukkan Harga Sewa Baru: 100
Akun dengan ID Alex Killer Pro berhasil diupdate menjadi ID Alex Biasa Aja dengan harga sewa baru 100!
```

Gambar 4.2.5 Program Update Akun

```
4. Hapus Akun
5. LOGOUT
=====
Pilih menu (1-5): 4
Masukkan ID akun yang ingin dihapus: Lyev
Akun dengan ID Lyev berhasil dihapus!

Tekan Enter untuk melanjutkan...

=====
                        MENU ADMIN
=====
```

Gambar 4.2.6 Program Hapus Akun

```
=====
                        MENU ADMIN
=====
1. Masukkan Id Akun
2. Tampilkan Daftar Akun
3. Update Akun
4. Hapus Akun
5. LOGOUT
=====
Pilih menu (1-5): 5
Terima kasih, logout berhasil.

PS C:\APL>
```

Gambar 4.2.7 Keluar Program

5. GIT

5.1 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.

```
Lenovo@lyev MINGW64 /c/APL/project-git/praktikum-apl/post-test/post-test-2 (main)
$ git add .
```

5.2 GIT COMMIT

Git commit berfungsi untuk menyimpan perubahan file di repository.

```
Lenovo@lyev MINGW64 /c/APL/project-git/praktikum-apl/post-test/post-test-2 (main)
$ git commit -m "nguawur"
[main aeb2d0f] nguawur
2 files changed, 189 insertions(+)
create mode 100644 praktikum-apl/post-test/post-test-2/2409106081_BenyaminHaqie_PT_2.cpp
create mode 100644 praktikum-apl/post-test/post-test-2/2409106081_BenyaminHaqie_PT_2.exe
```

5.3 GIT PUSH

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

```
Lenovo@lyev MINGW64 /c/APL/project-git/praktikum-apl/post-test/post-test-2 (main)
$ git push origin main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 20 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 678.42 KiB | 3.85 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/haqi2/praktikum-apl.git
   551eb04..aeb2d0f  main -> main

Lenovo@lyev MINGW64 /c/APL/project-git/praktikum-apl/post-test/post-test-2 (main)
$
```