

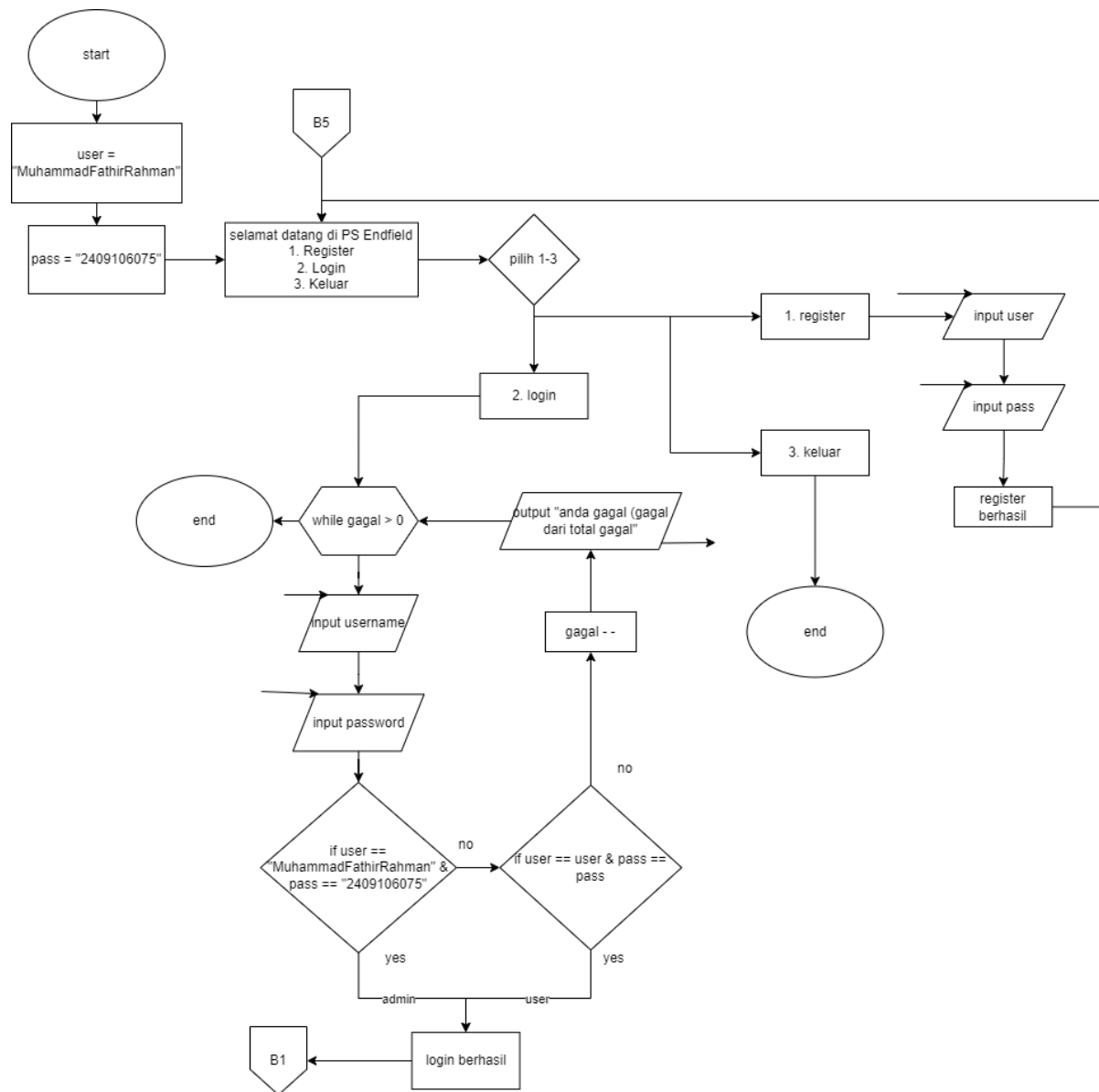
**LAPORAN PRAKTIKUM**  
**POSTTEST (3)**  
**ALGORITMA PEMROGRAMAN LANJUT**



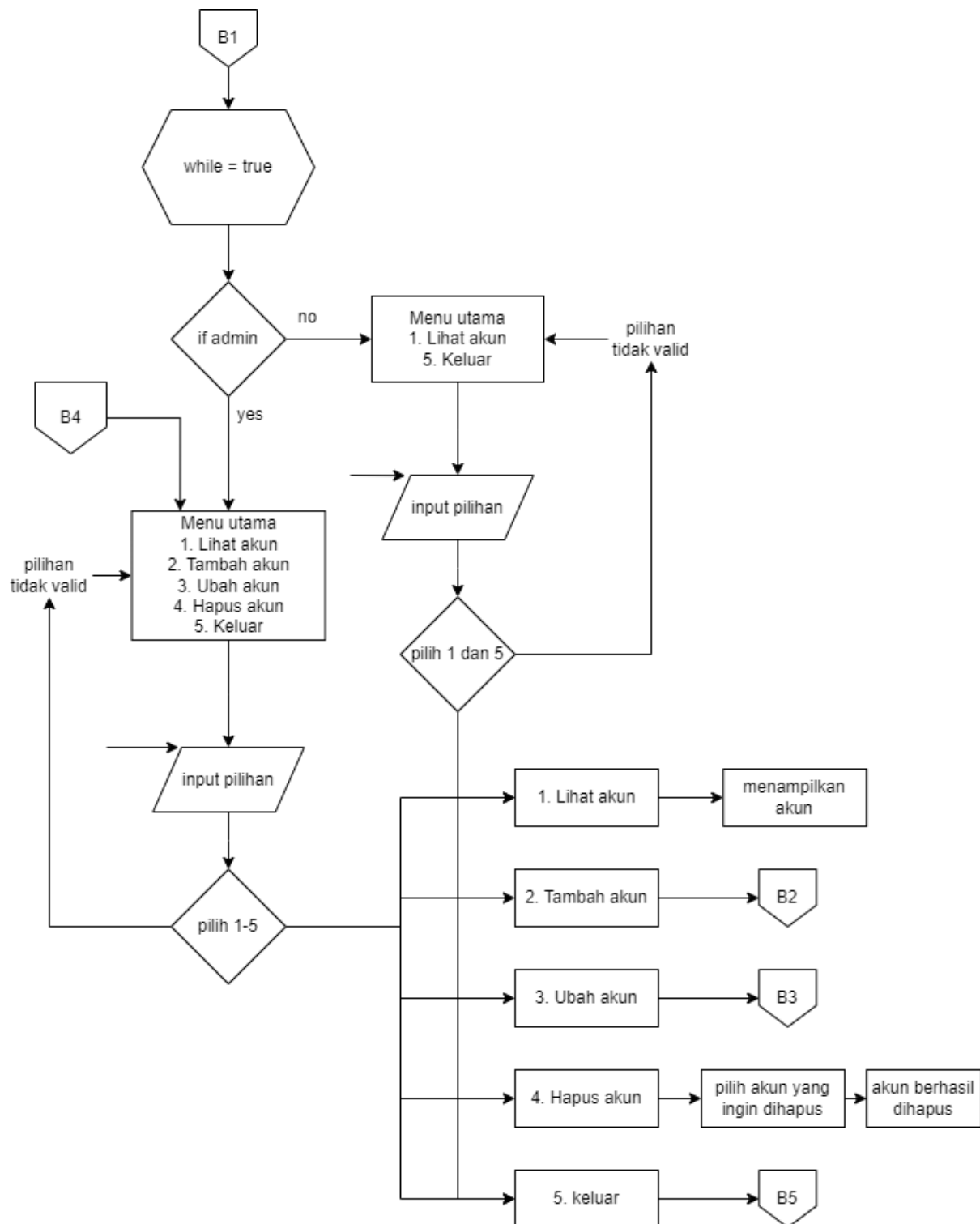
**Disusun oleh:**  
**Muhammad Fathir Rahman (2409106075)**  
**Kelas (B2'24)**

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

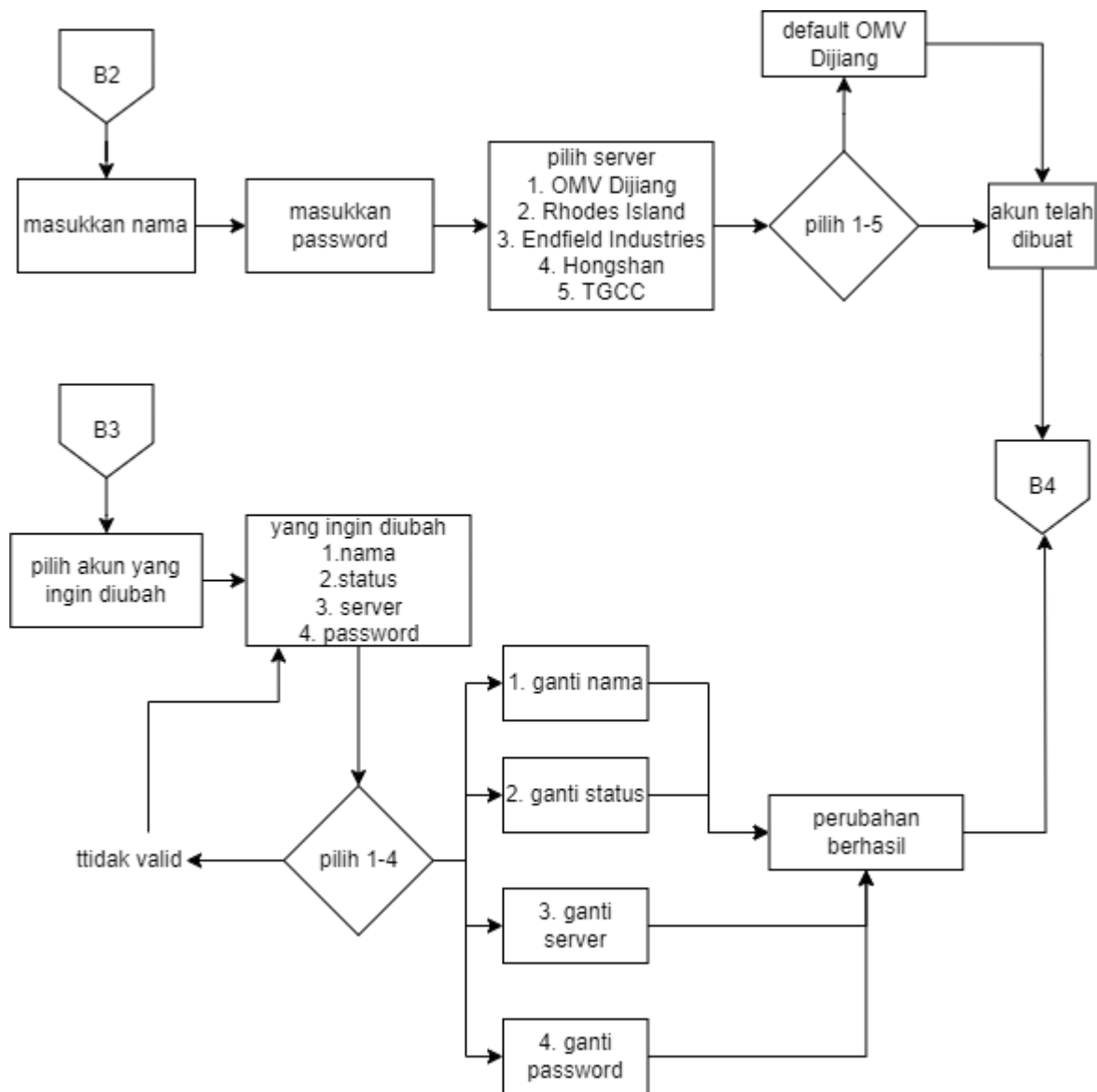
## 1. Flowchart



Gambar 1.1 Register dan Login



Gambar 1.2 Menu utama



Gambar 1.3 Tambah akun dan Ubah akun

## **2. Analisis Program**

Pada posttest kali ini mirip seperti posttest sebelumnya yang dimana kita hanya mengganti beberapa 3 variabel array yang lalu menjadi 1 variabel array of struct. Program ini sendiri merupakan program CRUD yang berisi Create, Read, Update dan Delete,

### 3. Source Code

#### A. Fitur Register

Fitur ini digunakan untuk mendaftarkan user

Sourcecode:

```
// Fungsi untuk register
void registerUser() {
    if (userCount < MAX_USER) {
        cout << "Masukkan username: ";
        cin >> users[userCount].username;
        cout << "Masukkan password: ";
        cin >> users[userCount].password;
        users[userCount].role = "Commoners"; // Default role adalah "User"
        userCount++;
        cout << "Registrasi berhasil!" << endl;
    } else {
        cout << "Kapasitas pengguna penuh! Tidak bisa menambah pengguna lagi." << endl;
    }
}
```

Gambar 3.1 Register

#### B. Fitur Login

Fitur ini sendiri digunakan untuk melakukan login

Sourcecode:

```
/ Fungsi untuk login
User* login() {
    string username, password;
    int percobaan = 0;
    const int maksPercobaan = 3;

    cout << "Selamat datang di PS Endfield" << endl;
    cout << "Silakan login terlebih dahulu." << endl;

    while (percobaan < maksPercobaan) {
        cout << "Username: ";
        cin >> username;
        cout << "Password: ";
        cin >> password;

        // Cek username dan password
    }
```

```

        for (int i = 0; i < userCount; i++) {
            if (users[i].username == username && users[i].password ==
password) {
                cout << "Login berhasil! Selamat datang, " <<
users[i].role << "!" << endl;
                this_thread::sleep_for(chrono::seconds(2)); // Jeda 2
detik
                return &users[i]; // Mengembalikan pointer ke pengguna
yang login
            }
        }

        percobaan++;
        cout << "Username atau password salah! Percobaan ke-" <<
percobaan << " dari " << maksPercobaan << "." << endl;
    }

    cout << "Anda telah melebihi batas percobaan login. Program
berhenti." << endl;
    exit(0); // Menghentikan program jika gagal login 3 kali
}

```

Gambar 3.2 Login

### C. Menu utama

Dimenu utama sendiri ada 5 pilihan sebagai admin dan 2 pilihan sebagai user.

Sourcecode:

```

// Fungsi untuk menampilkan menu
int tampilkanMenu(User* user) {
    int pilihan;
    cout << "Menu Program" << endl;
    cout << "1. Tampilkan Akun" << endl;
    if (user->role == "Endministrator") {
        cout << "2. Tambah Akun" << endl;
        cout << "3. Ubah Akun" << endl;
        cout << "4. Hapus Akun" << endl;
    }
    cout << "5. Keluar" << endl;
    cout << "Pilihan: ";
    cin >> pilihan;
    return pilihan;
}

```

Gambar 3.3 Menu utama

#### D. Tampilkan akun

Bagian ini berfungsi untuk menampilkan akun

Sourcecode:

```
// Fungsi untuk menampilkan semua akun dalam bentuk tabel dengan nomor
urut
int tampilkanAkun() {
    if (akunCount == 0) {
        cout << "Belum ada akun" << endl;
        return 0; // Mengembalikan 0 jika tidak ada akun
    } else {
        cout << "+-----+-----+-----+-----+-----+-----+"
        -----+-----+-----+-----+-----+-----+
        cout << "| No.      | Nama Akun              | Status          |" << endl;
        Server Tujuan      | Password                  |" << endl;
        cout << "+-----+-----+-----+-----+-----+-----+"
        -----+-----+-----+-----+-----+-----+
        for (int i = 0; i < akunCount; i++) {
            cout << "| " << setw(5) << left << i + 1
                << " | " << setw(20) << left << akun[i].nama
                << "| " << setw(20) << left << akun[i].status
                << "| " << setw(20) << left << akun[i].serverTujuan
                << "| " << setw(20) << left << akun[i].password << " |"
<< endl;
        }

        cout << "+-----+-----+-----+-----+-----+-----+"
        -----+-----+-----+-----+-----+-----+
        return 1; // Mengembalikan 1 jika berhasil menampilkan akun
    }
}
```

Gambar 3.4 Tampilkan akun

### E. Tambah akun

Bagian ini berfungsi untuk menambahkan akun

Sourcecode:

```
// Fungsi untuk menambah akun
int tambahAkun(User* user) {
    if (akunCount < MAX_AKUN) {
        cout << "Masukkan nama akun: ";
```



```

    cin.ignore();
    getline(cin, akun[akunCount].nama);

    // Input password (minimal 8 karakter)
    string passwordInput;
    do {
        cout << "Masukkan password akun (minimal 8 karakter): ";
        getline(cin, passwordInput);
        if (passwordInput.length() < 8) {
            cout << "Password harus minimal 8 karakter!" << endl;
        }
    } while (passwordInput.length() < 8);
    akun[akunCount].password = passwordInput;

    // Input server tujuan (hanya 5 pilihan)
    akun[akunCount].serverTujuan = pilihServerTujuan();

    // Status akun default "Aktif" saat menambah akun
    akun[akunCount].status = "Aktif";

    // Menyimpan informasi pengguna yang membuat akun
    akun[akunCount].user = *user;

    akunCount++;
    cout << "Akun berhasil ditambahkan" << endl;
    return 1; // Mengembalikan 1 jika berhasil menambah akun
} else {
    cout << "Kapasitas penuh! Tidak bisa menambah akun lagi." <<
endl;
    return 0; // Mengembalikan 0 jika gagal menambah akun
}
}

```

Gambar 3.5 Tambah akun

#### F. Ubah akun

Bagian ini berfungsi untuk mengubah akun yang telah dibuat

Sourcecode:

```

// Fungsi untuk mengubah akun
int ubahAkun(User* user) {
    int index;
    if (akunCount == 0) {
        cout << "Belum ada akun untuk diubah." << endl;
        return 0;
    } else {
        tampilkanAkun();
    }
}

```

```

    cout << "Masukkan nomor akun yang akan diubah: ";
    cin >> index;

    if (index > 0 && index <= akunCount) {
        // Hanya admin yang bisa mengubah
        if (user->role == "Endministrator" || akun[index -
1].user.username == user->username) {
            int pilihanUbah;
            cout << "Pilih yang ingin diubah:" << endl;
            cout << "1. Nama Akun" << endl;
            cout << "2. Status Akun" << endl;
            cout << "3. Server Tujuan" << endl;
            cout << "4. Password" << endl;
            cout << "Pilihan: ";
            cin >> pilihanUbah;
            cin.ignore(); // Membersihkan buffer

            switch (pilihanUbah) {
                case 1: // Ubah nama akun
                    cout << "Masukkan nama akun baru: ";
                    getline(cin, akun[index - 1].nama);
                    break;
                case 2: // Ubah status akun
                    {
                        string statusInput;
                        do {
                            cout << "Masukkan status akun
(Aktif/Tidak Aktif): ";
                            getline(cin, statusInput);
                            if (statusInput != "Aktif" &&
statusInput != "Tidak Aktif") {
                                cout << "Status tidak valid! Harap
masukkan 'Aktif' atau 'Tidak Aktif'." << endl;
                            }
                        } while (statusInput != "Aktif" &&
statusInput != "Tidak Aktif");
                        akun[index - 1].status = statusInput;
                    }
                    break;
                case 3: // Ubah server tujuan
                    akun[index - 1].serverTujuan =
pilihServerTujuan();
                    break;
                case 4: // Ubah password
                    {
                        string passwordInput;
                        do {
                            cout << "Masukkan password akun baru
(minimal 8 karakter): ";
                            getline(cin, passwordInput);
                            if (passwordInput.length() < 8) {
                                cout << "Password harus minimal 8
karakter!" << endl;

```

```

        }
        } while (passwordInput.length() < 8);
        akun[index - 1].password = passwordInput;
    }
    break;
default:
    cout << "Pilihan tidak valid!" << endl;
    return 0;
}

    cout << "Akun berhasil diubah" << endl;
    return 1; // Mengembalikan 1 jika berhasil mengubah akun
} else {
    cout << "Anda tidak memiliki izin untuk mengubah akun
ini." << endl;
    return 0;
}
} else {
    cout << "Nomor akun tidak valid" << endl;
    return 0; // Mengembalikan 0 jika nomor akun tidak valid
}
}
}
}

```

Gambar 3.6 Ubah akun

## G. Hapus akun

Bagian ini berfungsi untuk menghapus akun yang sudah ada

Sourcecode:

```

// Fungsi untuk menghapus akun
int hapusAkun(User* user) {
    int index;
    if (akunCount == 0) {
        cout << "Belum ada akun untuk dihapus." << endl;
        return 0; // Mengembalikan 0 jika tidak ada akun
    } else {
        tampilkanAkun();
        cout << "Masukkan nomor akun yang akan dihapus: ";
        cin >> index;

        if (index > 0 && index <= akunCount) {
            // Hanya admin yang bisa menghapus
            if (user->role == "Endministrato" || akun[index -
1].user.username == user->username) {
                for (int i = index - 1; i < akunCount - 1; i++) {
                    akun[i] = akun[i + 1];
                }
            }
        }
    }
}

```

```

        }
        akunCount--;
        cout << "Akun berhasil dihapus" << endl;
        return 1; // Mengembalikan 1 jika berhasil menghapus
    }
    } else {
        cout << "Anda tidak memiliki izin untuk menghapus akun
    ini." << endl;
        return 0;
    }
    } else {
        cout << "Nomor akun tidak valid" << endl;
        return 0; // Mengembalikan 0 jika nomor akun tidak valid
    }
}
}

```

Gambar 3.7 Hapus akun

#### H. Keluar

Ini sendiri merupakan fungsi untuk keluar dari program CRUD ini sendiri

Sourcecode:

```

        case 5:
            cout << "Keluar dari menu." << endl;
            break;
        default:
            cout << "Pilihan tidak valid" << endl;
            break;
    }
    if (pilihanMenu != 5) {
        system("pause");
    }
    } while (pilihanMenu != 5);
    }
    break;
case 3:
    cout << "Program selesai." << endl;
    return 0;
default:
    cout << "Pilihan tidak valid!" << endl;
    break;
}
if (pilihan != 3) {
    system("pause");
}

```

```
}  
    return 0;  
}
```

Gambar 3.8 Keluar

#### 4. Uji Coba dan Hasil Output

```
Selamat datang di PS Endfield
1. Register
2. Login
3. Keluar
Pilihan: █
```

Gambar 4.1 Metode masuk

```
Selamat datang di PS Endfield
1. Register
2. Login
3. Keluar
Pilihan: 1
Masukkan username: Ajimu
Masukkan password: Ajimu
Registrasi berhasil!
Press any key to continue . . . █
```

Gambar 4.2 Register

```
Pilihan: 2
Selamat datang di PS Endfield
Silakan login terlebih dahulu.
Username: MuhammadFathirRahman
Password: 2409106075
Login berhasil! Selamat datang, Endministrator!
█
```

Gambar 4.3 Login sebagai admin

```
Pilihan: 2
Selamat datang di PS Endfield
Silakan login terlebih dahulu.
Username: Ajimu
Password: Ajimu
Login berhasil! Selamat datang, Commoners!
█
```

Gambar 4.4 Login sebagai user

```
Selamat datang di PS Endfield
Silakan login terlebih dahulu.
Username: a
Password: a
Username atau password salah! Percobaan ke-1 dari 3.
Username: a
```

Gambar 4.5 gagal kurang dari 3

```
Username atau password salah! Percobaan ke-2 dari 3.
Username: a
Password: a
Username atau password salah! Percobaan ke-3 dari 3.
Anda telah melebihi batas percobaan login. Program berhenti.
PS E:\rusak ngini rek\rusak ngini rek\.vscode\output> █
```

Gambar 4.6 gagal 3 kali

```
Selamat datang di PS Endfield
Daftar Server dan Jumlah Akun:
+-----+-----+
| Server Tujuan      | Jumlah Akun      |
+-----+-----+
+-----+-----+
Menu Program
1. Tampilkan Akun
2. Tambah Akun
3. Ubah Akun
4. Hapus Akun
5. Keluar
Pilihan: █
```

Gambar 4.7 menu utama admin

```

Selamat datang di PS Endfield
Daftar Server dan Jumlah Akun:
+-----+-----+
| Server Tujuan      | Jumlah Akun      |
+-----+-----+
+-----+-----+
Menu Program
1. Tampilkan Akun
5. Keluar
Pilihan: █

```

Gambar 4.8 menu utama user

```

Pilihan: 1
+-----+-----+-----+-----+-----+
| No.  | Nama Akun      | Status      | Server Tujuan | Password      |
+-----+-----+-----+-----+-----+
| 1    | Ajimu          | Aktif       | Rhodes Island | AngelaAjimu   |
| 2    | Leavatain      | Aktif       | Rhodes Island | Leavatcream   |
+-----+-----+-----+-----+-----+
Press any key to continue . . . █

```

4.9 Lihat akun

```

Pilihan: 2
Masukkan nama akun: Avywenna
Masukkan password akun (minimal 8 karakter): TGCCchamber
Pilih server tujuan:
1. OMV Dijiang
2. Rhodes Island
3. Endfield Industries
4. Hongshan
5. TGCC
Pilihan: 5
Akun berhasil ditambahkan
Press any key to continue . . . █

```

Gambar 4.10 Tambah akun



```

Pilihan: 3
+-----+-----+-----+-----+-----+
| No.   | Nama Akun      | Status      | Server Tujuan | Password      |
+-----+-----+-----+-----+-----+
| 1     | Ajimu          | Aktif       | Rhodes Island | AngelaAjimu   |
| 2     | Leavatain      | Aktif       | Rhodes Island | Leavatcream   |
| 3     | Atywenna       | Aktif       | TGCC          | TGCCChamber   |
+-----+-----+-----+-----+-----+
Masukkan nomor akun yang akan diubah: 1
Pilih yang ingin diubah:
1. Nama Akun
2. Status Akun
3. Server Tujuan
4. Password
Pilihan: 1
Masukkan nama akun baru: Angelina
Akun berhasil diubah
Press any key to continue . . .

```

Gambar 4.11 Ubah akun

```

Pilihan: 4
+-----+-----+-----+-----+-----+
| No.   | Nama Akun      | Status      | Server Tujuan | Password      |
+-----+-----+-----+-----+-----+
| 1     | Angelina       | Aktif       | Rhodes Island | AngelaAjimu   |
| 2     | Leavatain      | Aktif       | Rhodes Island | Leavatcream   |
| 3     | Atywenna       | Aktif       | TGCC          | TGCCChamber   |
+-----+-----+-----+-----+-----+
Masukkan nomor akun yang akan dihapus: 3
Akun berhasil dihapus
Press any key to continue . . .

```

Gambar 4.12 Hapus akun

```

Selamat datang di PS Endfield
1. Register
2. Login
3. Keluar
Pilihan: 3
Program selesai.
PS E:\rusak ngini rek\rusak ngini rek\.vscode\output>

```

Gambar 4.13 keluar program

## 5. Langkah-Langkah Git pada VSCode

### A. Git init dan Git add

Perintah Git init berfungsi untuk menginisialisasikan repo git pada folder dan Git add untuk menambah file apa saja sebelum melakukan commit

```
PS E:\Praktikum-APL> git init
Reinitialized existing Git repository in E:/Praktikum-APL/.git/
PS E:\Praktikum-APL> git add .
PS E:\Praktikum-APL> |
```

Gambar 5.1 Git init dan Git add

### B. Git commit

Perintah Git commit digunakan untuk menyimpan perubahan yang telah ditambahkan ke dalam repo git

```
PS E:\Praktikum-APL> git commit -m 'selesai'
[main ec05ce9] selesai
2 files changed, 402 insertions(+)
create mode 100644 Post-Test/post-test-apl-3/2409106075-MuhammadFathirRahman-PT-3.cpp
create mode 100644 Post-Test/post-test-apl-3/2409106075-MuhammadFathirRahman-PT-3.exe
PS E:\Praktikum-APL>
```

Gambar 5.2 Git commit

### C. Git push

Karena sebelumnya sudah pernah mengremote jadi kita langsung ke Git push yang berguna untuk ngepush file ke repo github

```
PS E:\Praktikum-APL> git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 743.48 KiB | 3.38 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/SeiSendai/Praktikum-APL.git
   8643d4f..ec05ce9  main -> main
branch 'main' set up to track 'origin/main'.
PS E:\Praktikum-APL> |
```