

LAPORAN PRAKTIKUM
POSTTEST 3
ALGORITMA PEMROGRAMAN LANJUT



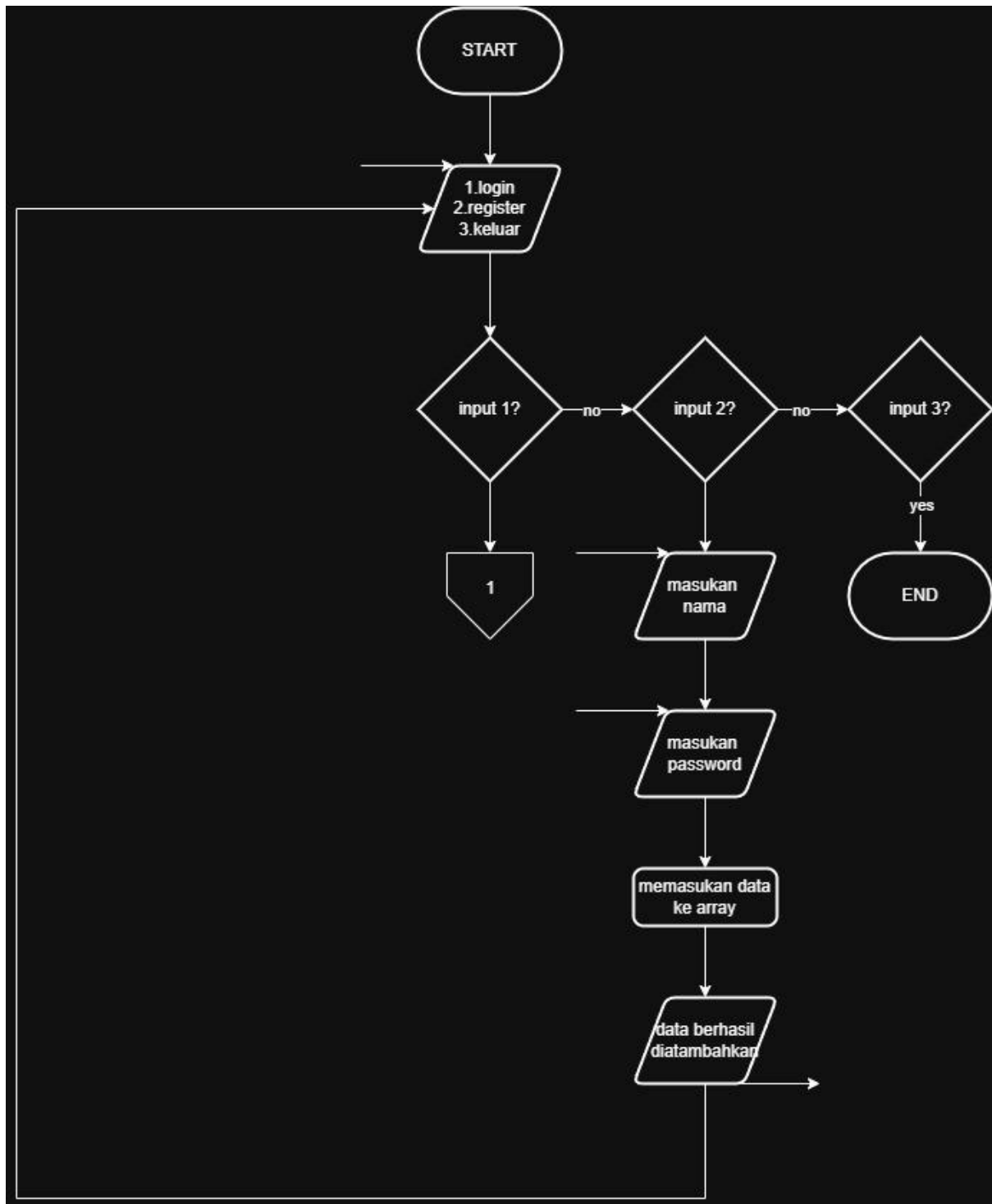
Disusun oleh:

Nama (2409106056)

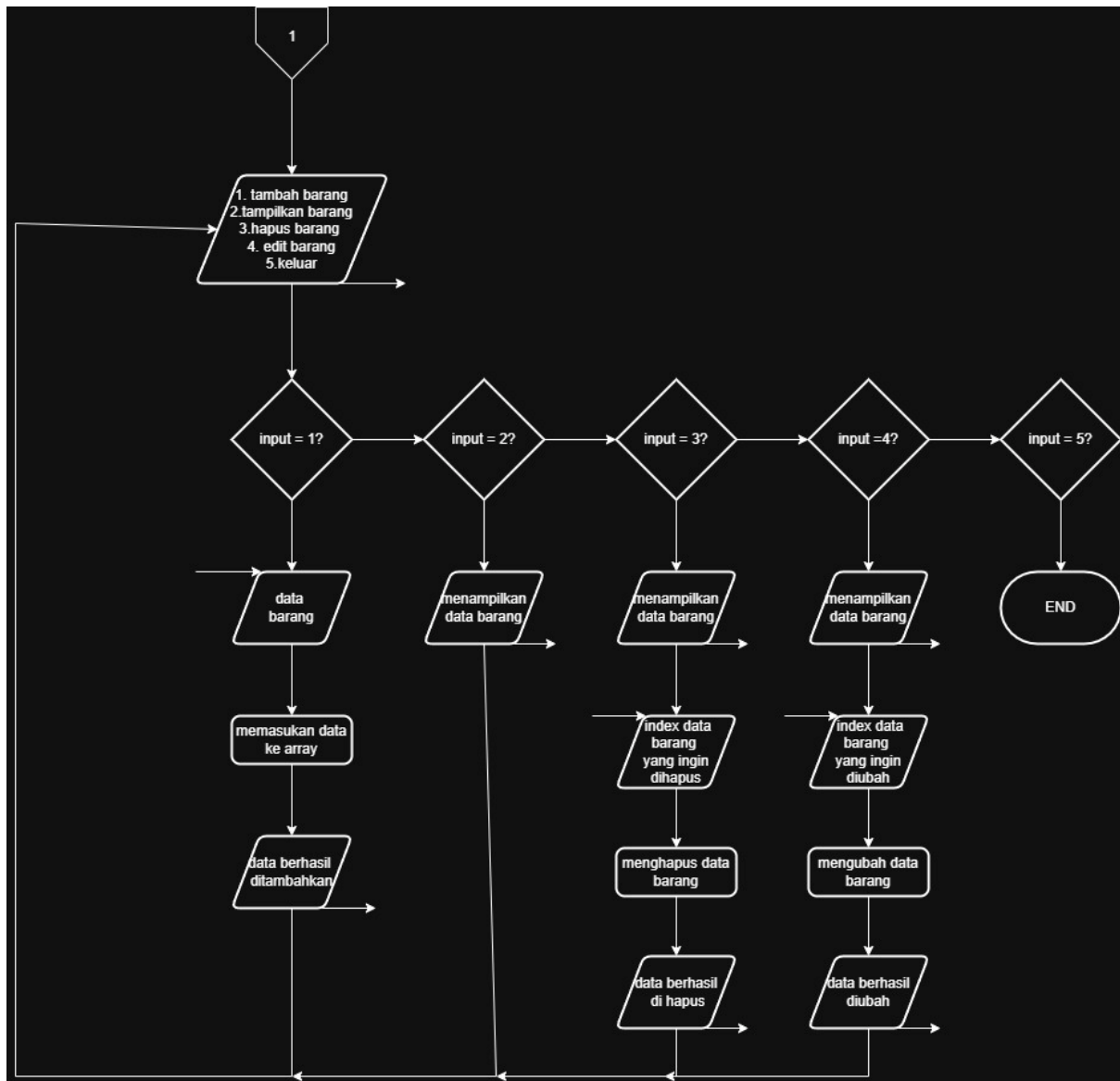
Kelas (B1 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Gambar 1.1 Menu login dan register



Gambar 1.2 CRUD

2. Analisis Program

Program ini adalah sistem manajemen barang aksesoris komputer berbasis dengan fitur login dan register

1. Autentikasi Pengguna

- Pengguna harus login terlebih dahulu dengan username dan password.
- Jika salah memasukkan password lebih dari 3 kali, login akan gagal.
- Terdapat fitur pendaftaran pengguna baru (register).

2. Manajemen Barang (CRUD - Create, Read, Update, Delete)

- **Tambah Barang:** Pengguna bisa menambahkan barang dengan kode, nama, dan harga.
- **Tampilkan Barang:** Menampilkan daftar barang yang telah tersimpan.
- **Hapus Barang:** Menghapus barang dari daftar berdasarkan nomor urutnya.
- **Edit Barang:** Mengubah nama dan harga barang berdasarkan nomor urutnya.

3. Struktur Data

- Menggunakan struct untuk menyimpan data barang dan pengguna.
- Data barang disimpan dalam array dataBarang[MAX_DATA].
- Data pengguna disimpan dalam array users[MAX_USER].

4. Looping Menu

- Program berjalan dalam loop hingga pengguna memilih opsi keluar.
- Setelah login berhasil, pengguna dapat mengakses menu utama untuk mengelola barang.

3. Source Code

A. Autentikasi pengguna

```
do {  
    system("cls");  
    cout << "\n=====\\n";  
    cout << "|          SELAMAT DATANG          |\\n";  
    cout << "=====\\n";  
    cout << "| 1. Login                          |\\n";  
    cout << "| 2. Register                       |\\n";  
    cout << "| 3. Keluar                         |\\n";  
    cout << "=====\\n";  
    cout << "Pilih menu: ";  
    cin >> state.pilihMenu;  
    cin.ignore();  
  
    if (state.pilihMenu == 1) {  
        state.attempts = 0;  
        state.loggedIn = false;  
  
        while (state.attempts < 3) {  
            system("cls");  
            cout << "\n=====\\n";  
            cout << "|          LOGIN SISTEM          |\\n";  
            cout << "=====\\n";  
            cout << "Nama: ";  
            getline(cin, state.inputNama);  
            cout << "Password: ";  
            getline(cin, state.inputPassword);  
  
            for (int i = 0; i < jumlahUser; i++) {  
                if (state.inputNama == users[i].login.nama &&  
state.inputPassword == users[i].login.password) {  
                    state.loggedIn = true;
```

```

        string newNama, newPassword;
        system("cls");
        cout << "\n=====\\n";
        cout << "|          REGISTER AKUN          |\\n";
        cout << "=====\\n";
        cout << "Masukkan Nama: ";
        getline(cin, newNama);
        cout << "Masukkan Password: ";
        getline(cin, newPassword);

        users[jumlahUser].login.nama = newNama;
        users[jumlahUser].login.password = newPassword;
        jumlahUser++;
        cout << "\\nRegistrasi berhasil! Silakan login.\\n";
        cin.ignore();
        cin.get();
    } while (state.pilihMenu != 3);
    return 0;

```

B.CRUD

```

do {
    system("cls");
    cout << "\n=====\\n";
    cout << "|          MENU UTAMA          |\\n";
    cout << "=====\\n";
    cout << "| 1. Tambah Barang          |\\n";
    cout << "| 2. Tampilkan Barang      |\\n";
    cout << "| 3. Hapus Barang         |\\n";
    cout << "| 4. Edit Barang          |\\n";
    cout << "| 5. Keluar                |\\n";
    cout << "=====\\n";
    cout << "Pilih menu: ";
    cin >> pilihan;
    cin.ignore();

```

```

        if (pilihan == 1) {
            system("cls");
            if (jumlahData >= MAX_DATA) {
                cout << "\nData penuh!\n";
                cin.get();
                continue;
            }
            cout << "\nMasukkan Kode Barang: ";
            getline(cin, barang.kode);
            cout << "Masukkan Nama Barang: ";
            getline(cin, barang.info.nama);
            string harga;
            bool valid;
            do {
                valid = true;
                cout << "Masukkan Harga Barang: ";
                cin >> harga;
                for (char c : harga) {
                    if (c < '0' || c > '9') {
                        valid = false;
                        break;
                    }
                }
            } while (!valid);
            if (!valid) {
                cout << "Harga harus berupa angka! Coba
lagi.\n";
            }
            barang.info.harga = stoi(harga);
            dataBarang[jumlahData++] = barang;
            cout << "Data berhasil ditambahkan!\n";
            cin.get();
        } else if (pilihan == 2) {

```

```

        system("cls");
        if (jumlahData == 0) {
            cout << "\nData masih kosong!\n";
        } else {
            cout << "\n-----\n";
            -----\n";

            cout << " | No |      Kode      |      Nama Barang      |
            Harga      |";

            cout << "\n-----\n";
            -----\n";

            for (int i = 0; i < jumlahData; i++) {
                cout << " | " << setw(2) << i + 1 << " | "
<< setw(8) << dataBarang[i].kode << " | "
                << setw(15) <<
dataBarang[i].info.nama << " | " << setw(6) << dataBarang[i].info.harga << " | \n";
            }
            cout << "-----\n";
            -----\n";

        }
        cin.get();
    } else if (pilihan == 3) {
        system("cls");
        if (jumlahData == 0) {
            cout << "\nData masih kosong!\n";
        } else {
            cout << "\n-----\n";
            -----\n";

            cout << " | No |      Kode      |      Nama Barang      |
            Harga      |";

            cout << "\n-----\n";
            -----\n";

            for (int i = 0; i < jumlahData; i++) {
                cout << " | " << setw(2) << i + 1 << " | "
<< setw(8) << dataBarang[i].kode << " | "

```



```

                                << setw(15) <<
dataBarang[i].info.nama << " | " << setw(6) << dataBarang[i].info.harga << " |\n";
    }
    cout << "-----\n";

    cout << "Masukkan nomor barang yang akan
dihapus: ";

    int hapusIndex;
    cin >> hapusIndex;
    if (hapusIndex > 0 && hapusIndex <=
jumlahData) {
        for (int i = hapusIndex - 1; i <
jumlahData - 1; i++) {
            dataBarang[i] = dataBarang[i + 1];
        }
        jumlahData--;
        cout << "Data berhasil dihapus!\n";
    } else {
        cout << "Nomor tidak valid!\n";
    }
}
cin.ignore();
cin.get();
} else if (pilihan == 4) {
    system("cls");
    if (jumlahData == 0) {
        cout << "\nData masih kosong!\n";
    } else {
        cout << "\n-----\n";
        cout << "
-----\n";

        cout << " | No |      Kode      |      Nama Barang      |
Harga      |";

```

```

        cout << "\n-----\n";

        for (int i = 0; i < jumlahData; i++) {
            cout << " | " << setw(2) << i + 1 << " | "
<< setw(8) << dataBarang[i].kode << " | "
            << setw(15) <<
dataBarang[i].info.nama << " | " << setw(6) << dataBarang[i].info.harga << " |\n";
        }
        cout << "-----\n";

        diedit: ";

        int index;
        cin >> index;
        if (index < 1 || index > jumlahData) {
            cout << "Nomor tidak valid!\n";
            cin.ignore();
            cin.get();
            continue;
        }
        cin.ignore();
        cout << "Masukkan Nama Baru: ";
        getline(cin, dataBarang[index - 1].info.nama);
        cout << "Masukkan Harga Baru: ";
        cin >> dataBarang[index - 1].info.harga;
        cin.ignore();
        cout << "Data berhasil diubah!\n";
    }
    cin.get();
}
} while (pilihan != 5);

```

C.Struktur Data

```
const int MAX_DATA = 100;
const int MAX_USER = 15;

struct InfoBarang {
    string nama;
    int harga;
};

struct Barang {
    string kode;
    InfoBarang info;
};

struct loginState {
    int pilihMenu;
    int attempts = 0;
    string inputNama;
    string inputPassword;
    bool loggedIn = false;
};

struct loginData {
    string nama;
    string password;
};

struct User {
    loginData login;
};

Barang dataBarang[MAX_DATA];
User users[MAX_USER];
int jumlahData = 0;
```

```
int jumlahUser = 1;
```

4. Keseluruhan kode

```
#include <iostream>
#include <iostream>
#include <iomanip>
#include <string>
#include <cstdlib>

using namespace std;

const int MAX_DATA = 100;
const int MAX_USER = 15;

struct InfoBarang {
    string nama;
    int harga;
};

struct Barang {
    string kode;
    InfoBarang info;
};

struct loginState {
    int pilihMenu;
    int attempts = 0;
    string inputNama;
    string inputPassword;
    bool loggedIn = false;
};

struct loginData {
    string nama;
    string password;
```

```

};

struct User {
    loginData login;
};

Barang dataBarang[MAX_DATA];
User users[MAX_USER];
int jumlahData = 0;
int jumlahUser = 1;

int main() {
    users[0].login.nama = "dzaki";
    users[0].login.password = "056";

    loginState state;

    do {
        system("cls");
        cout << "\n=====\\n";
        cout << "|          SELAMAT DATANG          |\\n";
        cout << "=====\\n";
        cout << "| 1. Login                          |\\n";
        cout << "| 2. Register                       |\\n";
        cout << "| 3. Keluar                         |\\n";
        cout << "=====\\n";
        cout << "Pilih menu: ";
        cin >> state.pilihMenu;
        cin.ignore();

        if (state.pilihMenu == 1) {
            state.attempts = 0;
            state.loggedIn = false;

```

```

while (state.attempts < 3) {
    system("cls");

    cout << "\n=====\\n";
    cout << "|          LOGIN SISTEM          |\\n";
    cout << "=====\\n";
    cout << "Nama: ";
    getline(cin, state.inputNama);
    cout << "Password: ";
    getline(cin, state.inputPassword);

    for (int i = 0; i < jumlahUser; i++) {
        if (state.inputNama == users[i].login.nama &&
state.inputPassword == users[i].login.password) {
            state.loggedIn = true;
            int pilihan;
            Barang barang;

            do {
                system("cls");
                cout << "\\n=====\\n";
                cout << "|          MENU UTAMA          |\\n";
                cout << "=====\\n";
                cout << "| 1. Tambah Barang          |\\n";
                cout << "| 2. Tampilkan Barang      |\\n";
                cout << "| 3. Hapus Barang          |\\n";
                cout << "| 4. Edit Barang           |\\n";
                cout << "| 5. Keluar                 |\\n";
                cout << "=====\\n";
                cout << "Pilih menu: ";
                cin >> pilihan;
                cin.ignore();

                if (pilihan == 1) {
                    system("cls");

```

```

        if (jumlahData >= MAX_DATA) {
            cout << "\nData penuh!\n";
            cin.get();
            continue;
        }
        cout << "\nMasukkan Kode Barang: ";
        getline(cin, barang.kode);
        cout << "Masukkan Nama Barang: ";
        getline(cin, barang.info.nama);
        string harga;
        bool valid;
        do {
            valid = true;
            cout << "Masukkan Harga Barang: ";
            cin >> harga;
            for (char c : harga) {
                if (c < '0' || c > '9') {
                    valid = false;
                    break;
                }
            }
            if (!valid) {
                cout << "Harga harus berupa angka! Coba
lagi.\n";
            }
        } while (!valid);
        barang.info.harga = stoi(harga);
        dataBarang[jumlahData++] = barang;
        cout << "Data berhasil ditambahkan!\n";
        cin.get();
    } else if (pilihan == 2) {
        system("cls");
        if (jumlahData == 0) {
            cout << "\nData masih kosong!\n";

```



```

        } else {
            cout << "\n-----\n";

            cout << "| No | Kode | Nama Barang |  

            Harga |";

            cout << "\n-----\n";

            for (int i = 0; i < jumlahData; i++) {
                cout << "| " << setw(2) << i + 1 << " | "
<< setw(8) << dataBarang[i].kode << " | "
                << setw(15) <<
dataBarang[i].info.nama << " | " << setw(6) << dataBarang[i].info.harga << " |\n";
            }
            cout << "\n-----\n";

        }
        cin.get();
    } else if (pilihan == 3) {
        system("cls");
        if (jumlahData == 0) {
            cout << "\nData masih kosong!\n";
        } else {
            cout << "\n-----\n";

            cout << "| No | Kode | Nama Barang |  

            Harga |";

            cout << "\n-----\n";

            for (int i = 0; i < jumlahData; i++) {
                cout << "| " << setw(2) << i + 1 << " | "
<< setw(8) << dataBarang[i].kode << " | "
                << setw(15) <<
dataBarang[i].info.nama << " | " << setw(6) << dataBarang[i].info.harga << " |\n";
            }

```

```

cout << "-----\n";

cout << "Masukkan nomor barang yang akan
dihapus: ";

int hapusIndex;
cin >> hapusIndex;
if (hapusIndex > 0 && hapusIndex <=
jumlahData) {
    for (int i = hapusIndex - 1; i <
jumlahData - 1; i++) {
        dataBarang[i] = dataBarang[i + 1];
    }
    jumlahData--;
    cout << "Data berhasil dihapus!\n";
} else {
    cout << "Nomor tidak valid!\n";
}
}
cin.ignore();
cin.get();
} else if (pilihan == 4) {
    system("cls");
    if (jumlahData == 0) {
        cout << "\nData masih kosong!\n";
    } else {
        cout << "\n-----
-----\n";

        cout << "| No | Kode | Nama Barang |
Harga |";

        cout << "\n-----
-----\n";

        for (int i = 0; i < jumlahData; i++) {

```

```

        cout << " | " << setw(2) << i + 1 << " | "
<< setw(8) << dataBarang[i].kode << " | "
        << setw(15) <<
dataBarang[i].info.nama << " | " << setw(6) << dataBarang[i].info.harga << " |\n";
    }
    cout << "-----\n";

    cout << "Masukkan nomor barang yang akan
diedit: ";

    int index;
    cin >> index;
    if (index < 1 || index > jumlahData) {
        cout << "Nomor tidak valid!\n";
        cin.ignore();
        cin.get();
        continue;
    }
    cin.ignore();
    cout << "Masukkan Nama Baru: ";
    getline(cin, dataBarang[index - 1].info.nama);
    cout << "Masukkan Harga Baru: ";
    cin >> dataBarang[index - 1].info.harga;
    cin.ignore();
    cout << "Data berhasil diubah!\n";
}
cin.get();
}
} while (pilihan != 5);
}
}
}

string newNama, newPassword;
system("cls");

```

```

        cout << "\n=====\\n";
        cout << "|          REGISTER AKUN          |\\n";
        cout << "=====\\n";
        cout << "Masukkan Nama: ";
        getline(cin, newNama);
        cout << "Masukkan Password: ";
        getline(cin, newPassword);

        users[jumlahUser].login.nama = newNama;
        users[jumlahUser].login.password = newPassword;
        jumlahUser++;
        cout << "\\nRegistrasi berhasil! Silakan login.\\n";
        cin.ignore();
        cin.get();
    } while (state.pilihMenu != 3);
    return 0;
}

```

5. Langkah-Langkah Git pada VSCode

1. Git add .

Git add . berfungsi untuk menambahkan file-file yang ada di dalam suatu folder atau repositori karena . sendiri berarti semua

```
PS C:\Users\ASUS\Praktikum-APL\Post-test\Post-test-3> git add .
```

2. Git commit -m "done"

```
PS C:\Users\ASUS\Praktikum-APL\Post-test\Post-test-3> git commit -m "done"
[main 682e70c] done
1 file changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 Post-test/Post-test-3/crud.cpp
PS C:\Users\ASUS\Praktikum-APL\Post-test\Post-test-3>
```

Git commit -m "done" berfungsi untuk mencommit pekerjaan kita sebelum di push ke github

-m sendiri berarti message karena saat mencommit kita harus meninggalkan pesan dan pesan yang ditinggalkan adalah "done"

3. Git push origin main

```
PS C:\Users\ASUS\Praktikum-APL\Post-test\Post-test-3> git push origin main
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 20 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (11/11), 679.32 KiB | 8.60 MiB/s, done.
Total 11 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
```

Git push origin main berfungsi untuk mengupload file yang sudah kita add dan commit ke github