

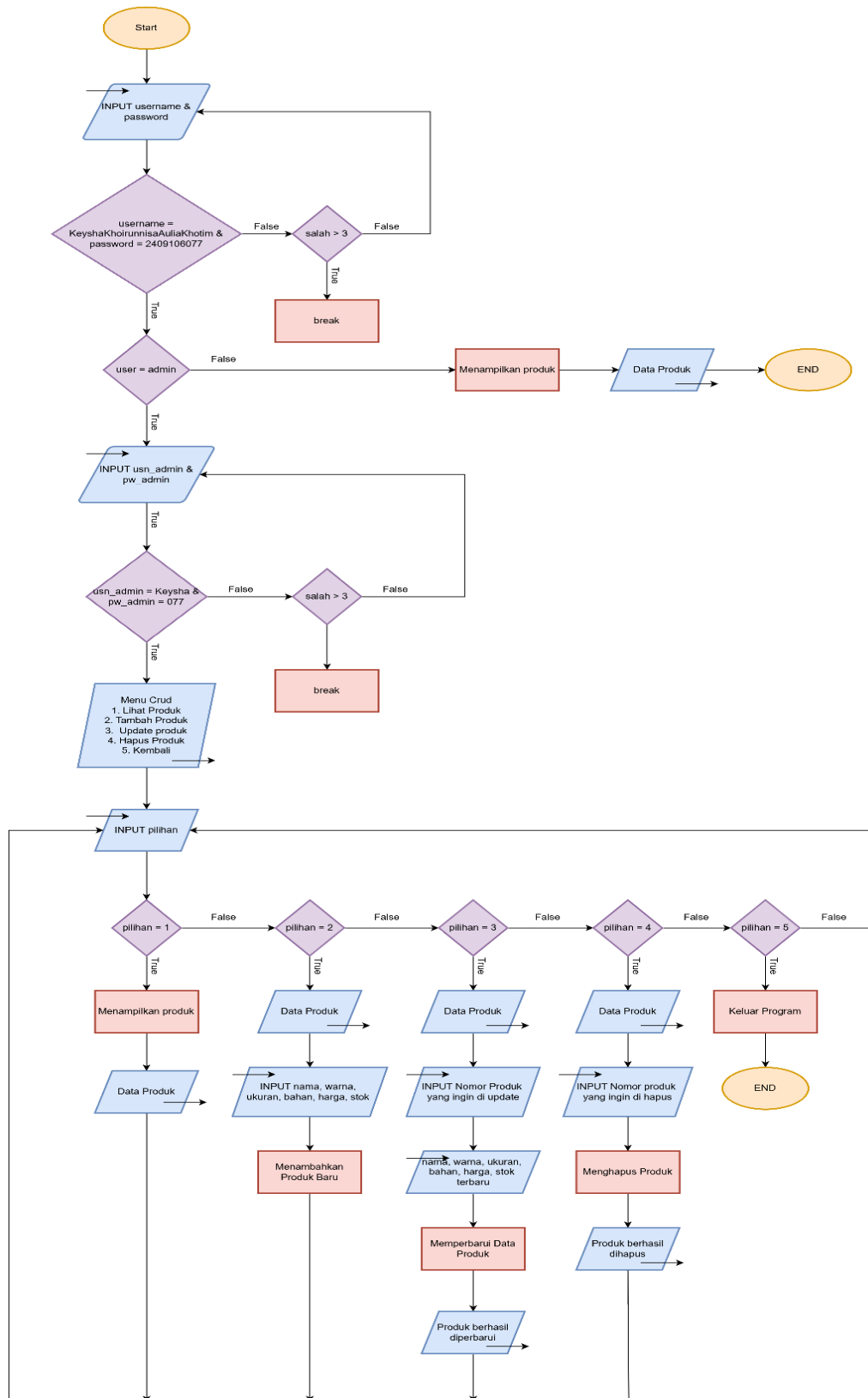
LAPORAN PRAKTIKUM
POSTTEST 4
ALGORITMA PEMROGRAMAN LANJUT



Disusun oleh:
Keysha Khoirunnisa Aulia Khotim (2409106077)
Kelas (B2 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



gambar 1.1 flowchart

2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini dibuat untuk mensimulasikan program manajemen produk berupa baju muslimah yang melibatkan sistem login menggunakan username (Nama), password (NIM), dan maksimal 3 kali percobaan. Jika berhasil login maka akan muncul 5 menu admin yaitu Create, Read, Update, Delete, dan kembali. Dan menu user Read.

3. Source Code

A. Struct dan Nested Struct

Struct *produk* digunakan untuk menyimpan detail produk, seperti nama, ukuran, warna, bahan, harga, dan stok. Struct ini juga memiliki anggota berupa *ukuran* sebagai nested struct untuk menyimpan informasi tambahan. Dalam penggunaan nested struct, data diakses melalui notasi titik (dot notation). Misalnya, `produk[0].ukuran.ukuranbaju` mengakses ukuran baju produk pertama. Hal ini membuat data terorganisir secara hierarkis.

```
struct Login {
    string nama;
    string password;
};

struct ukuran {
    string jenisukuran;
};

struct produk {
    ukuran ukuranbaju;
    string namabaju;
    string warnabaju;
    string bahanbaju;
    int hargabaju;
    int stokbaju;
};

struct Login user = {"KeyshaKhoirunnisa", "2409106077"};
struct Login admin = {"Keysha", "077"};
```

B. Verifikasi Username dan Password

Fungsi ini digunakan untuk memverifikasi input username dan password yang diinput oleh pengguna sesuai dengan yang sudah diinisialisasikan, jika pengguna salah hingga percobaan ke 3 maka program akan berhenti.

```

void loginuser(Login user, int salah) {
    string username_input , password_input;
    for(salah = 0; salah < 3; salah++) {
        cout << "username : ";
        cin >> username_input;
        cout << "password : ";
        cin >> password_input;

        if (username_input == user.nama && password_input == user.password)
        {
            cout << "Login Berhasil!" << endl;
            break;
        } cout << "Login Gagal! Coba Lagi" << endl;
    }
    if (salah == 3) {
        cout << "Kesempatan Habis! Program Berhenti.";
        return;
    }
}

```

C. Verifikasi User

Fungsi ini digunakan untuk memverifikasi user sebagai admin dengan menginput username dan password admin sesuai dengan yang sudah diinisialisasikan, jika pengguna salah hingga percobaan ke 3 maka program akan berhenti.

```

void loginadmin(Login admin, int salah) {
    string usn_admin_input, pw_admin_input;
    for (salah = 0; salah < 3; salah++) {
        cout << "username admin : ";
        cin >> usn_admin_input;
        cout << "password admin : ";
        cin >> pw_admin_input;

        system ("cls");
        if (admin.nama == usn_admin_input && admin.password ==
pw_admin_input) {
            cout << "Anda sebagai Admin!" << endl;
            break;
        } cout << "Anda bukan admin!" << endl;
    }
    if (salah == 3) {
        cout << "Kesempatan Habis! Program Berhenti";
        return;
    }
}

```

D. Menu Admin

Fitur ini terdiri atas 5 pilihan, yaitu Lihat Produk, Tambah Produk, Update Produk, Delete Produk, dan Log Out.

```
int pilihan ;
do {
    cout << "===Manajemen Produk Baju Muslimah===" << endl;
    cout << "1. Lihat Produk" << endl;
    cout << "2. Tambah Produk" << endl;
    cout << "3. Update Produk" << endl;
    cout << "4. Hapus Produk" << endl;
    cout << "5. Kembali" << endl;
    cout << "Masukkan pilihan : ";
    cin >> pilihan;
```

E. Lihat Produk

Fungsi ini digunakan untuk melihat daftar produk yang sudah ada.

```
void lihatproduk(produk produk[], int jumlahProduk) {
    if (jumlahProduk == 0) {
        cout << "Tidak ada produk yang tersedia" << endl;
        return;
    }
    cout << "=====Daftar Produk===== " << endl;
    for (int i = 0; i < jumlahProduk; i++) {
        cout << "Produk ke-" << i + 1 << endl;
        cout << "Produk          : " << produk[i].namabaju << endl;
        cout << "Warna             : " << produk[i].warnabaju << endl;
        cout << "Ukuran            : " << produk[i].ukuranbaju.jenisukuran <<
endl;
        cout << "Harga              : Rp " << fixed << setprecision(0) <<
produk[i].hargabaju << endl;
        cout << "Bahan              : " << produk[i].bahanbaju << endl;
        cout << "Stok               : " << produk[i].stokbaju << " pcs" << endl;
    }
}
```

F. Tambah Produk

Fungsi ini digunakan untuk menambah produk baru pada program, yang di batasi oleh jumlah maksimal produk.

```
void tambahproduk(produk produk[], int &jumlahProduk, const int max_produk)
{
    if (jumlahProduk == 0) {
        cout << "Tidak ada produk yang tersedia" << endl;
        return;
    }
    lihatproduk(produk, jumlahProduk);
    cout << "Tekan enter untuk melanjutkan.." << endl;
    cin.ignore();
    cin.get();
    if (jumlahProduk >= max_produk) {
        cout << "Penuh! Tidak bisa menambah produk baru!";
        return;
    }
    cout << "=====Tambah Produk===== " << endl;
    cin.ignore();
    cout << "Masukkan nama produk   : ";
    getline(cin, produk[jumlahProduk].namabaju);
    cout << "Masukkan Warna             : ";
    getline(cin, produk[jumlahProduk].warnabaju);
    cout << "Masukkan Ukuran              : ";
    getline(cin, produk[jumlahProduk].ukuranbaju.jenisukuran);
    cout << "Masukkan Harga               : ";
    cin >> produk[jumlahProduk].hargabaju;
    cout << "Masukkan Bahan               : ";
    cin.ignore();
    getline(cin, produk[jumlahProduk].bahanbaju);
    cout << "Masukkan Stok                : ";
    cin >> produk[jumlahProduk].stokbaju;

    jumlahProduk++;
    cout << "Produk berhasil ditambahkan!" << endl;
}
```

G. Update Produk

Fungsi ini digunakan untuk mengupdate produk yang sudah ada.

```
void updateproduk(produk produk[], int jumlahProduk) {
```

```

if (jumlahProduk == 0) {
    cout << "Tidak ada produk untuk di update";
    return;
}
int index;
cout << "=====Update Produk===== " << endl;
cout << "Masukkan nomor produk yang ingin di update : ";
cin >> index;

if (index < 1 || index > jumlahProduk) {
    cout << "Produk tidak Ditemukan"<< endl;
    return;
}
cout << "Masukkan data produk yang ingin di update : " << endl;
cout << "Produk          : " << produk[index-1].namabaju << endl;
cout << "Warna           : " << produk[index-1].warnabaju << endl;
cout << "Ukuran          : " << produk[index-1].ukuranbaju.jenisukuran <<
endl;
cout << "Harga           : " << produk[index-1].hargabaju << endl;
cout << "Bahan           : " << produk[index-1].bahanbaju << endl;
cout << "Stok            : " << produk[index-1].stokbaju << endl;

cout << "Masukkan data baru:" << endl;
cout << "Nama produk   : ";
cin.ignore();
getline(cin, produk[index-1].namabaju);
cout << "Warna        : ";
getline(cin, produk[index-1].warnabaju);
cout << "Ukuran       : ";
getline(cin, produk[index-1].ukuranbaju.jenisukuran);
cout << "Harga        : ";
cin >> produk[index-1].hargabaju;
cout << "Bahan        : ";
cin.ignore();
getline(cin, produk[index-1].bahanbaju);
cout << "Stok         : ";
cin >> produk[index-1].stokbaju;
cout << "Produk berhasil diupdate!" << endl;
}

```

H. Hapus Produk

Fungsir ini digunakan untuk menghapus produk yang sudah ada pada program.

```

void hapusproduk(produk produk[], int &jumlahProduk) {
    if (jumlahProduk == 0) {
        cout << "Data masih kosong!" << endl;
        return;
    }
    int index;
    cout << "=====Hapus Produk===== " << endl;
    cout << "Masukkan nomor produk yang ingin dihapus : ";
    cin >> index;

    if (index < 1 || index > jumlahProduk) {
        cout << "Produk tidak Ditemukan"<< endl;
        return;
    }
    cout << "Data produk yang akan dihapus:" << endl;
    cout << "Produk      : " << produk[index - 1].namabaju << endl;
    cout << "Warna       : " << produk[index - 1].warnabaju << endl;
    cout << "Ukuran     : " << produk[index - 1].ukuranbaju.jenisukuran
    << endl;
    cout << "Harga      : Rp " << fixed << setprecision(0) <<
    produk[index - 1].hargabaju << endl;
    cout << "Bahan      : " << produk[index - 1].bahanbaju << endl;
    cout << "Stok       : " << produk[index - 1].stokbaju << " pcs" <<
    endl;

    string konfirmasi;
    cout << "Apakah anda yakin ingin menghapus data ini? : ";
    cin >> konfirmasi;

    if (konfirmasi == "y" || konfirmasi == "Y"){
        for (int i = index - 1; i < jumlahProduk - 1; i++){
            produk[i] = produk[i+1];
        }
    }
    jumlahProduk--;
    cout << "Produk berhasil dihapus!" << endl;
}

```

I. Log Out

Fitur ini digunakan untuk keluar dari program, fitur ini sudah diwakili oleh perulangan do-while, yaitu while (pilihan != 5)

```

}
    } while (pilihan != 5);

```


J. Main Menu

```
int main () {
    const int max_produk = 100;
    produk produk[max_produk];
    produk[0].namabaju = "Gamis";
    produk[0].warnabaju = "Hitam";
    produk[0].ukuranbaju.jenisukuran = "Small";
    produk[0].bahanbaju = "Katun";
    produk[0].hargabaju = 145000;
    produk[0].stokbaju = 45;

    system("cls");
    string validasi_admin;
    cout << "Apakah anda admin? (y/n) : ";
    cin >> validasi_admin;

    system("cls");
    if (validasi_admin == "y" || validasi_admin == "Y") {
        loginadmin(admin, salah);
        int pilihan ;
        do {
            cout << "===Manajemen Produk Baju Muslimah===" << endl;
            cout << "1. Lihat Produk" << endl;
            cout << "2. Tambah Produk" << endl;
            cout << "3. Update Produk" << endl;
            cout << "4. Hapus Produk" << endl;
            cout << "5. Kembali" << endl;
            cout << "Masukkan pilihan : ";
            cin >> pilihan;
            system("cls");
            if (pilihan == 1) {
                lihatproduk(produk, jumlahProduk);
                cout << "Tekan enter untuk melanjutkan.." << endl;
                cin.ignore();
                cin.get();
                system("cls");
            }
            else if (pilihan == 2) {
                tambahproduk(produk, jumlahProduk, max_produk);
                cout << "Tekan enter untuk melanjutkan.." << endl;
                cin.ignore();
                cin.get();
                system("cls");
            }
            else if (pilihan == 3) {
                lihatproduk(produk, jumlahProduk);
                updateproduk(produk, jumlahProduk);
                cout << "Tekan enter untuk melanjutkan.." << endl;
            }
        } while (pilihan != 5);
    }
}
```

```

        cin.ignore();
        cin.get();
        system("cls");
    }
    else if (pilihan == 4) {
        lihatproduk(produk, jumlahProduk);
        hapusproduk(produk, jumlahProduk);
        cout << "Tekan enter untuk melanjutkan.." << endl;
        cin.ignore();
        cin.get();
        system("cls");
    } else if (pilihan == 5) {
        cout << "Terimakasih sudah menggunakan sistem ini!";
    }
    } while (pilihan != 5);
} else {
    string pilihan;
    do {
        system("cls");
        loginuser(user, salah);
        lihatproduk(produk, jumlahProduk);
        cout << "Apakah anda ingin keluar? (y/n) : " ;
        cin >> pilihan;
        break;
    } while (pilihan != "y" || pilihan != "Y");
}
}

```

4. Uji Coba dan Hasil Output

```

=====Daftar Produk=====
Produk ke-1
Produk      : Gamis
Warna       : Hitam
Ukuran     : Small
Harga      : Rp 145000
Bahan      : Katun
Stok       : 45 pcs
Tekan enter untuk melanjutkan..

```

gambar 4.1 output pilihan 1

```
=====Daftar Produk=====
Produk ke-1
Produk      : Gamis
Warna       : Hitam
Ukuran     : Small
Harga       : Rp 145000
Bahan       : Katun
Stok        : 45 pcs
Tekan enter untuk melanjutkan..

=====Tambah Produk=====

Masukkan nama produk   : Gamis
Masukkan Warna         : Hitam
Masukkan Ukuran        : Large
Masukkan Harga         : 145000
Masukkan Bahan         : Katun
Masukkan Stok          : 50
Produk berhasil ditambahkan!
Tekan enter untuk melanjutkan..
```

gambar 4.2 output pilihan 2

```
=====Daftar Produk=====
Produk ke-1
Produk      : Gamis
Warna       : Hitam
Ukuran     : Small
Harga       : Rp 145000
Bahan       : Katun
Stok        : 45 pcs
Produk ke-2
Produk      : Gamis
Warna       : Hitam
Ukuran     : Large
Harga       : Rp 145000
Bahan       : Katun
Stok        : 50 pcs
Tekan enter untuk melanjutkan..

=====Update Produk=====
Masukkan nomor produk yang ingin di update : 1
Masukkan data produk yang ingin di update :
Produk      : Gamis
Warna       : Hitam
Ukuran     : Small
Harga       : 145000
Bahan       : Katun
Stok        : 45
Masukkan data baru:
Nama produk : Gamis
Warna       : Biru
Ukuran     : Large
Harga       : 15000
Bahan       : Katun
Stok        : 30
Produk berhasil diupdate!
Tekan enter untuk melanjutkan..
```

gambar 4.3 output pilihan 3

```

=====Daftar Produk=====
Produk ke-1
Produk      : Gamis
Warna       : Biru
Ukuran     : Large
Harga      : Rp 15000
Bahan      : Katun
Stok       : 30 pcs
Produk ke-2
Produk      : Gamis
Warna       : Hitam
Ukuran     : Large
Harga      : Rp 145000
Bahan      : Katun
Stok       : 50 pcs
Tekan enter untuk melanjutkan..

=====Hapus Produk=====
Masukkan nomor produk yang ingin dihapus : 1
Data produk yang akan dihapus:
Produk      : Gamis
Warna       : Biru
Ukuran     : Large
Harga      : Rp 15000
Bahan      : Katun
Stok       : 30 pcs
Apakah anda yakin ingin menghapus data ini? : y
Produk berhasil dihapus!
Tekan enter untuk melanjutkan..

```

gambar 4.4 output pilihan 4

```

Terimakasih sudah menggunakan sistem ini!
PS C:\Users\User\OneDrive\Desktop\praktikum-apl\post-test\post-test-apl-3>

```

gambar 4.5 output pilihan 5

5. Git

5.1 Git Add

```
User@DESKTOP-SAT40VG MINGW64 ~/OneDrive/Desktop/praktikum-apl/post-test/post-test-apl-4 (main)
$ git add .
```

5.2 Git Commit

```
User@DESKTOP-SAT40VG MINGW64 ~/OneDrive/Desktop/praktikum-apl/post-test/post-test-apl-4 (main)
$ git commit -m "code done"
[main c796fa3] code done
3 files changed, 273 insertions(+)
create mode 100644 post-test/post-test-apl-4/.vscode/settings.json
create mode 100644 post-test/post-test-apl-4/2409106077-KeyshaKhoirunnisaAuliaKhotim-PT-4.cpp
create mode 100644 post-test/post-test-apl-4/2409106077-KeyshaKhoirunnisaAuliaKhotim-PT-4.exe
```

5.3 Git Push Origin Main

```
User@DESKTOP-SAT40VG MINGW64 ~/OneDrive/Desktop/praktikum-apl/post-test/post-test-apl-4 (main)
$ git push -u origin main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 681.04 KiB | 5.97 MiB/s, done.
Total 8 (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/kyshaulia/praktikum-apl.git
 6f46721..c796fa3  main -> main
branch 'main' set up to track 'origin/main'.
```