

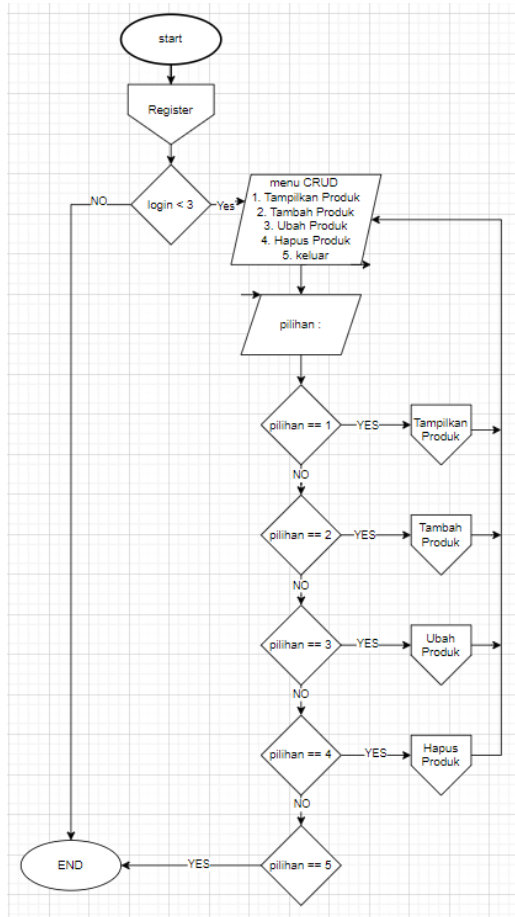
LAPORAN PRAKTIKUM
POSTTEST 4
ALGORITMA PEMROGRAMAN LANJUT



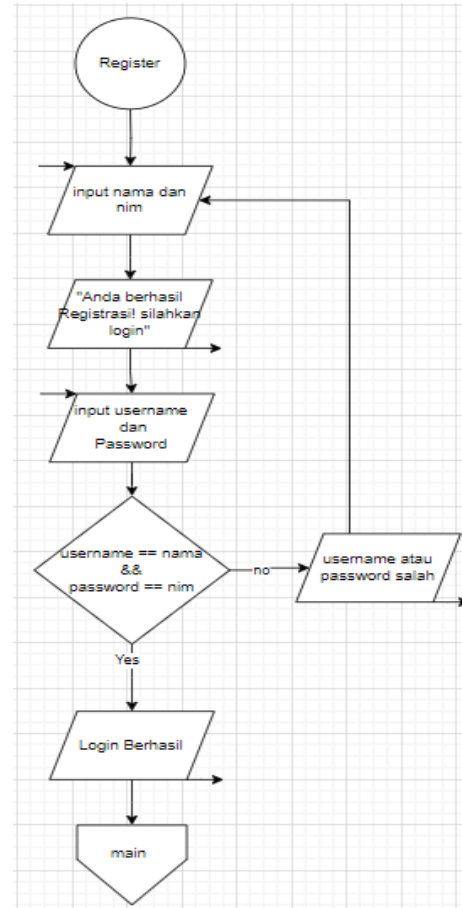
Disusun oleh:
Ayu Azzahrah Alwi (2409106022)
Kelas A1'24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

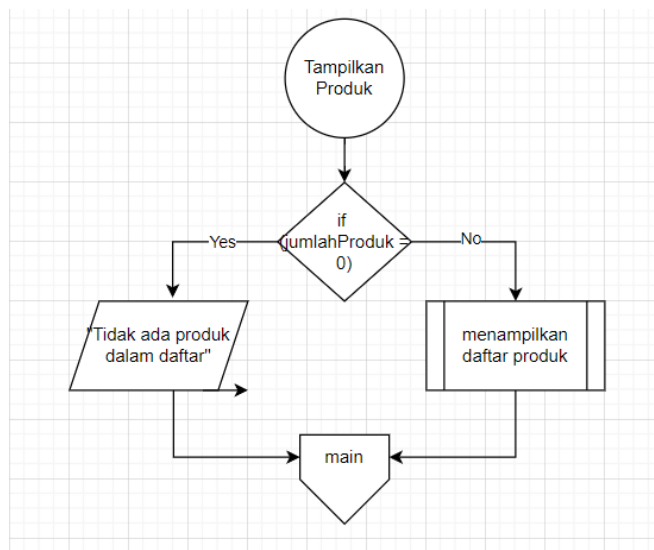
1. Flowchart



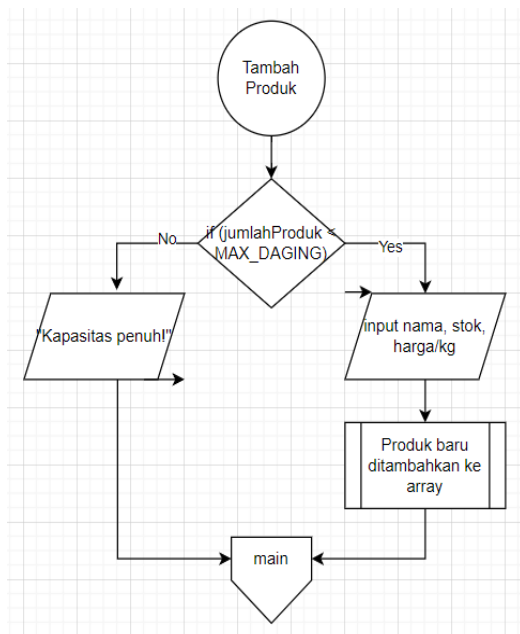
Gambar 1. 2 main flow



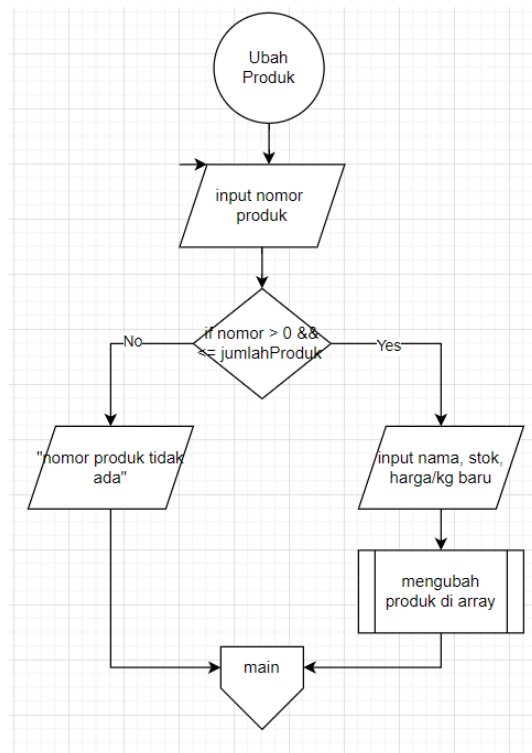
Gambar 1. 1 Register



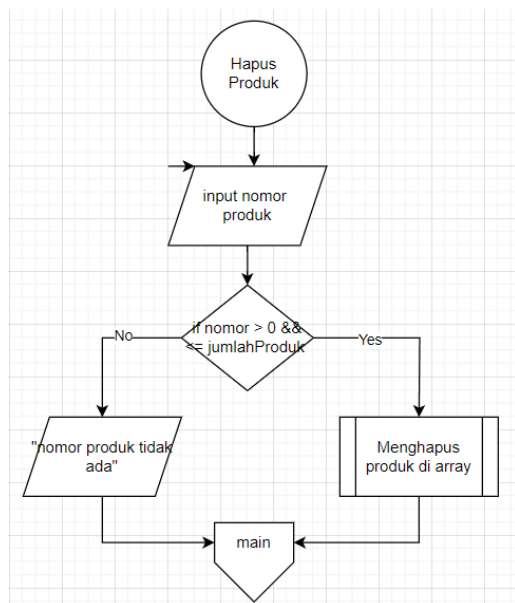
Gambar 1. 3 tampilkan produk



Gambar 1. 4 tambah produk



Gambar 1. 5 ubah produk



Gambar 1. 6 Hapus Produk

2. Analisis Program

2.1 Deskripsi Singkat Program

Program "Manajemen Toko Daging" dirancang untuk mengelola data produk daging dalam sebuah toko. Program ini memungkinkan pengguna untuk menambahkan, menampilkan, mengubah, dan menghapus data produk. Selain itu, program ini juga dilengkapi dengan fitur multiuser yang memungkinkan pengguna untuk melakukan registrasi dan login sebelum mengakses sistem. Manfaat utama dari program ini adalah membantu toko dalam mencatat stok dan harga produk secara sistematis serta memudahkan pengelolaan data produk menggunakan array dua dimensi.

3. Source Code

A. Struct

Fitur ini digunakan untuk Mengelompokkan data yang terkait dalam satu entitas

Source Code :

```
#define MAX_USER 10
#define MAX_PRODUK 100

struct User {
    string nama;
    string nim;
};

struct Produk {
    string nama;
    int stok;
    int harga;
};
```

B. Register dan Login

Fitur ini digunakan untuk Digunakan untuk mendaftarkan pengguna baru sebelum mereka bisa mengakses sistem.

source code :

```
void registrasi(User users[], int &jumlahUser) {
    if (jumlahUser < MAX_USER) {
        cout << "Masukkan Nama: ";
        cin.ignore();
        getline(cin, users[jumlahUser].nama);
        cout << "Masukkan NIM (sebagai password): ";
        getline(cin, users[jumlahUser].nim);
        jumlahUser++;
        cout << "Registrasi berhasil!\n";
    } else {
        cout << "Kuota user penuh!\n";
    }
}

bool login(User users[], int jumlahUser, int kesempatan = 3) {
    if (kesempatan == 0) return false;
    string nama, nim;
```

```

    cout << "Masukkan Nama: ";
    cin.ignore();
    getline(cin, nama);
    cout << "Masukkan NIM: ";
    getline(cin, nim);
    for (int i = 0; i < jumlahUser; i++) {
        if (users[i].nama == nama && users[i].nim == nim) {
            return true;
        }
    }
    cout << "Login gagal! Sisa kesempatan: " << kesempatan - 1 << endl;
    return login(users, jumlahUser, kesempatan - 1);
}

```

C. Menu

Terdapat beberapa menu untuk memilih fitur pada aplikasi

Source Code :

```

do {
    cout << "\n===== MENU =====\n";
    cout << "1. Tampilkan Produk\n";
    cout << "2. Tambah Produk\n";
    cout << "3. Ubah Produk\n";
    cout << "4. Hapus Produk\n";
    cout << "5. Keluar\n";
    cout << "Pilihan: ";
    cin >> pilihan;

    switch (pilihan) {
        case 1: tampilkan(produk, jumlahProduk); break;
        case 2: tambahProduk(produk, jumlahProduk); break;
        case 3: ubahProduk(produk, jumlahProduk); break;
        case 4: hapusProduk(produk, jumlahProduk); break;
        case 5: cout << "Terima kasih telah menggunakan program!\n"; break;
        default: cout << "Pilihan tidak valid!\n";
    }

} while (pilihan != 5);

```

C. CRUD Manajemen Toko Daging

Fitur **CRUD** dalam pemrograman ini mengacu pada empat operasi utama yaitu Create (membuat/menambahkan data baru), Read (Mengambil dan menampilkan data yang

tersimpan), Update (Mengubah atau memperbarui data yang sudah ada.), Delet (Menghapus data dari Array.) yang digunakan untuk mengelola data produk. Data disimpan dalam array of struct agar lebih terstruktur.

Source Code:

```
void tampilkan(Produk produk[], int jumlahProduk) {
    if (jumlahProduk == 0) {
        cout << "Belum ada produk.\n";
    } else {
        cout << "\n===== DAFTAR PRODUK =====\n";
        cout << "+-----+-----+-----+-----+\n";
        cout << "| No | Nama Produk | Stok | Harga/Kg |\n";
        cout << "+-----+-----+-----+-----+\n";
        for (int i = 0; i < jumlahProduk; i++) {
            cout << "| " << setw(2) << i + 1 << " | " << setw(13) <<
produk[i].nama
                << " | " << setw(4) << produk[i].stok
                << " | " << setw(8) << produk[i].harga << " |\n";
        }
        cout << "+-----+-----+-----+-----+\n";
    }
}

void tambahProduk(Produk produk[], int &jumlahProduk) {
    if (jumlahProduk < MAX_PRODUK) {
        cout << "Masukkan Nama Produk: ";
        cin.ignore();
        getline(cin, produk[jumlahProduk].nama);
        cout << "Masukkan Stok Produk: ";
        cin >> produk[jumlahProduk].stok;
        cout << "Masukkan Harga per Kg: ";
        cin >> produk[jumlahProduk].harga;
        jumlahProduk++;
        cout << "Produk berhasil ditambahkan.\n";
    } else {
        cout << "Kapasitas produk penuh!\n";
    }
}

void ubahProduk(Produk produk[], int jumlahProduk) {
    tampilkan(produk, jumlahProduk);
    int index;
    cout << "Pilih produk yang akan diubah: ";
    cin >> index;
    if (index < 1 || index > jumlahProduk) {
        cout << "Nomor tidak valid!\n";
    }
}
```

```

        return;
    }
    cout << "Masukkan Nama Baru: ";
    cin.ignore();
    getline(cin, produk[index - 1].nama);
    cout << "Masukkan Stok Baru: ";
    cin >> produk[index - 1].stok;
    cout << "Masukkan Harga Baru: ";
    cin >> produk[index - 1].harga;
    cout << "Produk berhasil diubah!\n";
}

void hapusProduk(Produk produk[], int &jumlahProduk) {
    tampilkan(produk, jumlahProduk);
    int index;
    cout << "Pilih produk yang akan dihapus: ";
    cin >> index;
    if (index < 1 || index > jumlahProduk) {
        cout << "Nomor tidak valid!\n";
        return;
    }
    for (int i = index - 1; i < jumlahProduk - 1; i++) {
        produk[i] = produk[i + 1];
    }
    jumlahProduk--;
    cout << "Produk berhasil dihapus!\n";
}

```


4. Uji Coba dan Hasil Output

```
Silakan lakukan registrasi.  
Masukkan Nama: Ayu13  
Masukkan Password: 222  
Registrasi berhasil! silahkan login.  
Masukkan Nama: Ayu13  
Masukkan NIM: 222  
Login berhasil!
```

Gambar 4. 1 login berhasil

```
===== PROGRAM MANAJEMEN PENJUALAN DAGING =====  
Silakan lakukan registrasi.  
Masukkan Nama: Ayu13  
Masukkan Password: 222  
Registrasi berhasil! silahkan login.  
Masukkan Nama: aaa  
Masukkan NIM: 321  
Login gagal! 2  
Masukkan Nama: Ayu1  
Masukkan NIM: 234  
Login gagal! 1  
Masukkan Nama: Ayu13  
Masukkan NIM: 221  
Login gagal! 0  
Anda gagal login 3 kali. Program berhenti.
```

Gambar 4. 2 login gagal

```
===== MENU =====
1. Tampilkan Produk
2. Tambah Produk
3. Ubah Produk
4. Hapus Produk
5. Keluar
Pilihan: 2
Masukkan Nama Produk: kambing
Masukkan Stok Produk: 20
Masukkan Harga per Kg: 65000
Produk berhasil ditambahkan!
```

Gambar 4. 3 menambah produk

```
===== MENU =====
1. Tampilkan Produk
2. Tambah Produk
3. Ubah Produk
4. Hapus Produk
5. Keluar
Pilihan: 1

===== DAFTAR PRODUK =====
+---+-----+---+-----+
| No | Nama Produk | Stok | Harga/Kg |
+---+-----+---+-----+
| 1  | kambing    | 20   | 65000    |
+---+-----+---+-----+
```

Gambar 4. 4 melihat produk

```

===== MENU =====
1. Tampilkan Produk
2. Tambah Produk
3. Ubah Produk
4. Hapus Produk
5. Keluar
Pilihan: 3

===== DAFTAR PRODUK =====
+---+-----+-----+---+-----+
| No | Nama Produk | Stok | Harga/Kg |
+---+-----+-----+---+-----+
| 1 | kambing | 20 | 65000 |
+---+-----+-----+---+-----+
Pilih produk yang akan diubah: 1
Masukkan Nama Baru: sapi
Masukkan Stok Baru: 22
Masukkan Harga Baru: 57000
Produk berhasil diubah!

```

Gambar 4. 5 mengubah produk

```

===== MENU =====
1. Tampilkan Produk
2. Tambah Produk
3. Ubah Produk
4. Hapus Produk
5. Keluar
Pilihan: 4

===== DAFTAR PRODUK =====
+---+-----+-----+---+-----+
| No | Nama Produk | Stok | Harga/Kg |
+---+-----+-----+---+-----+
| 1 | sapi | 22 | 57000 |
+---+-----+-----+---+-----+
Pilih produk yang akan dihapus: 1
Produk berhasil dihapus!

```

Gambar 4. 7 menghapus produk

```

Pilihan: 5
Terima kasih telah menggunakan program!

```

Gambar 4. 6 keluar dari program

5. Langkah-Langkah GIT

1. Git Init

Git init adalah perintah Git yang digunakan untuk menginisialisasi repository Git baru dalam sebuah folder. Cukup ketik “git init” pada terminal vscode

```
PS C:\Users\ASUS GK\OneDrive\Dokumen\praktikum-apl> git init
Reinitialized existing Git repository in C:/Users/ASUS GK/OneDrive/Dokumen/praktikum-apl/.git/
PS C:\Users\ASUS GK\OneDrive\Dokumen\praktikum-apl> █
```

Gambar 5. 1 Git init

2. Git add

Git add adalah perintah Git yang digunakan untuk menambahkan perubahan pada file ke staging area sebelum dilakukan commit. Perintah ini memungkinkan Git mengetahui file mana saja yang akan dimasukkan dalam commit berikutnya

```
PS C:\Users\ASUS GK\OneDrive\Dokumen\praktikum-apl> git add .
PS C:\Users\ASUS GK\OneDrive\Dokumen\praktikum-apl> █
```

Gambar 5. 2 Git add

2. Git commit

Git commit adalah perintah Git yang digunakan untuk menyimpan perubahan yang telah ditambahkan ke staging area ke dalam repository. Setiap commit akan memiliki hash unik, pesan commit, dan menyimpan snapshot dari perubahan yang dilakukan.

```
PS C:\Users\ASUS GK\OneDrive\Dokumen\praktikum-apl> git commit -m "Finish Post Test 4"
[main fc9c313] Finish Post Test 4
```

Gambar 5. 3 Git commit

3. Git remote

Git remote adalah perintah Git yang digunakan untuk menghubungkan repository lokal dengan repository remote (misalnya di GitHub, GitLab, atau Bitbucket). Dengan perintah ini, kita bisa mengelola koneksi ke repository jarak jauh, memungkinkan push, pull, dan fetch dari repository tersebut.

```
PS C:\Users\ASUS GK\OneDrive\Dokumen\praktikum-apl> git remote set-url https://github.com/ayuaa13/praktikum-apl.git
```

Gambar 5. 4 Git remote

4. Git Push

Git push adalah perintah Git yang digunakan untuk mengirim (mengunggah) perubahan dari repository lokal ke repository remote (seperti GitHub, GitLab, atau Bitbucket). Perintah ini memastikan bahwa perubahan yang sudah dikomit di lokal tersedia di repository jarak jauh sehingga bisa diakses oleh orang lain atau untuk cadangan.

```
PS C:\Users\ASUS GK\OneDrive\Dokumen\praktikum-apl> git push -u origin main
```

Gambar 5. 5 Git_Push