

LAPORAN PRAKTIKUM
POSTTEST 3
ALGORITMA PEMROGRAMAN LANJUT



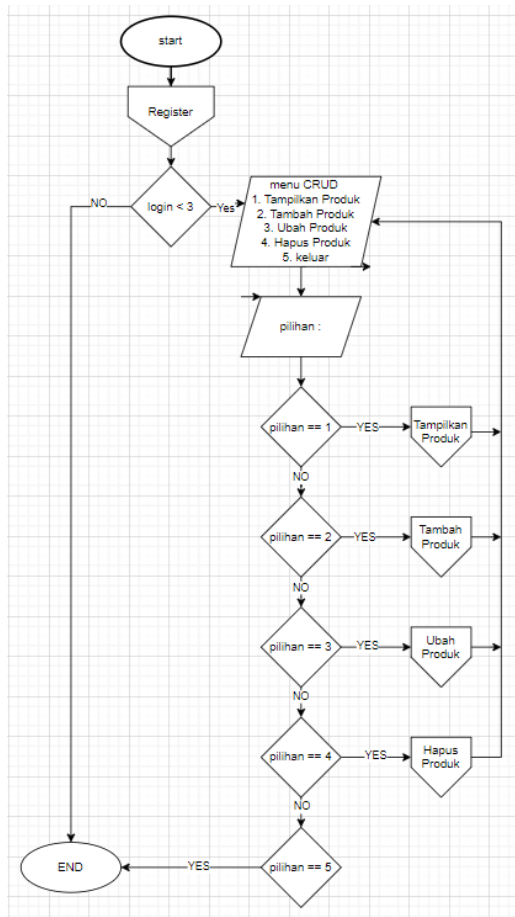
Disusun oleh:

Ayu Azzahrah Alwi (2409106022)

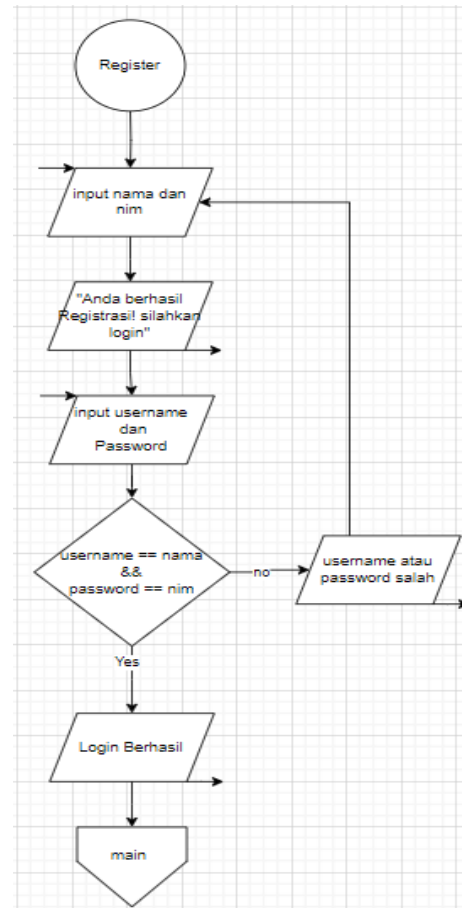
Kelas A1'24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

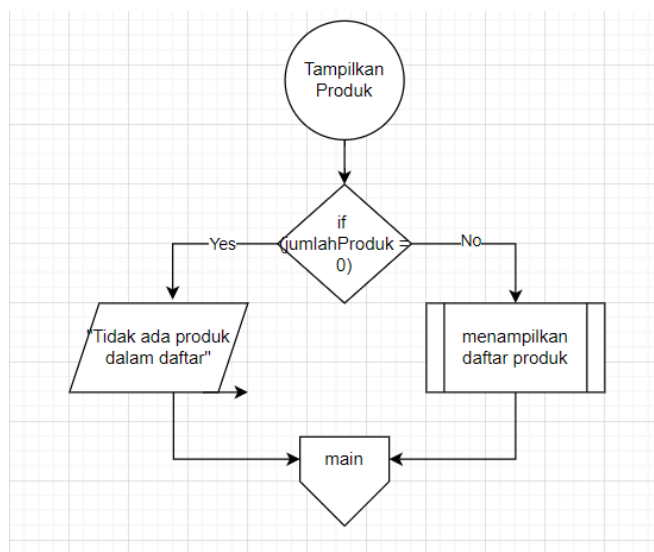
1. Flowchart



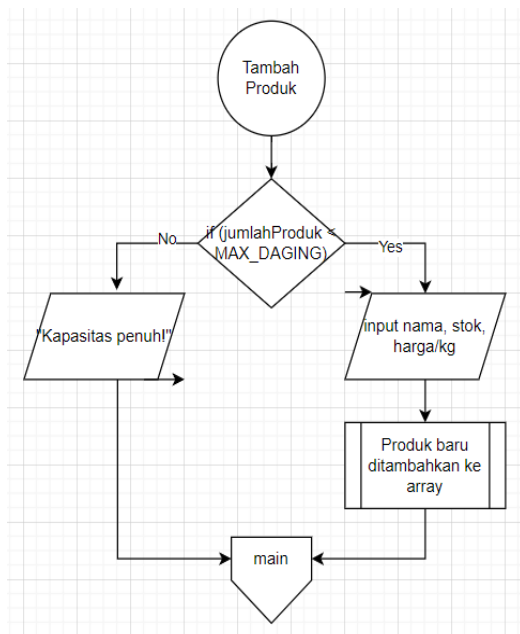
Gambar 1. 1 Main Flow



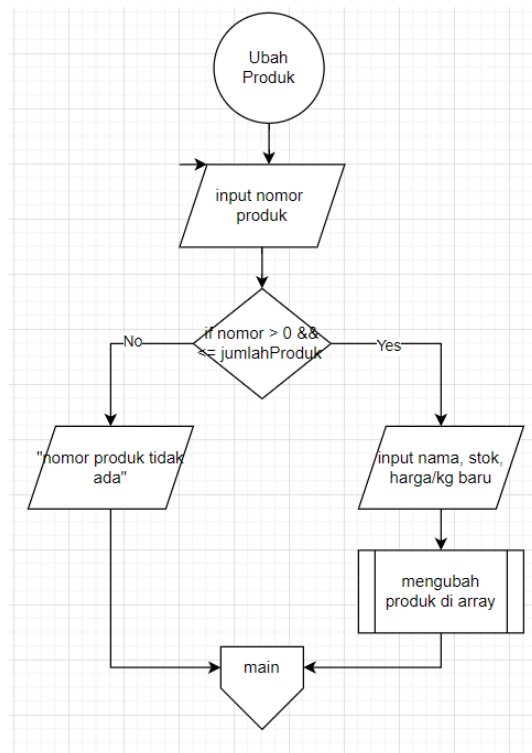
Gambar 1. 2 Register dan Login



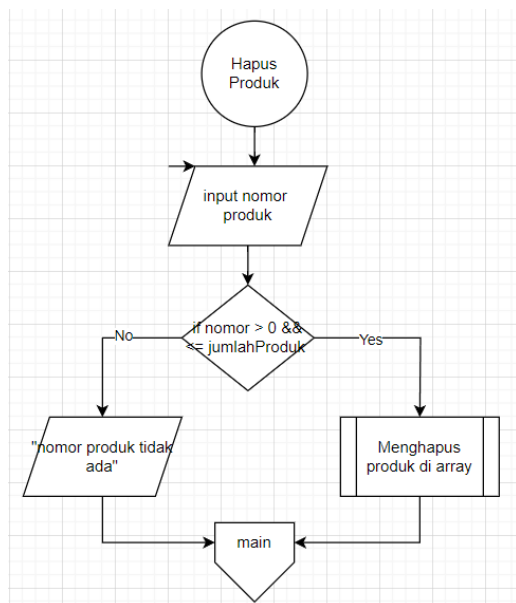
Gambar 1. 3 Tampilkan Produk



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 :

```
// Struct data pengguna
struct User {
    string nama;
    string nim;
};

// Struct data produk daging
struct Produk {
    string nama;
    int stok;
    int harga;
};

// Nested user dan produk
struct Database {
    User users[MAX_USER];
    int jumlahUser = 0;
    Produk produk[MAX_PRODUK];
    int jumlahProduk = 0;
};
```

B. Register dan Login

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

source code :

```
if (db.jumlahUser < MAX_USER) {
    cout << "Silakan lakukan registrasi.\n";
    cout << "Masukkan Nama: ";
    getline(cin, db.users[db.jumlahUser].nama);
    cout << "Masukkan Password: ";
    cin >> db.users[db.jumlahUser].nim;
    db.jumlahUser++;
    cout << "Registrasi berhasil! silahkan login.\n";
}

bool loginBerhasil = false;
```

```

while (kesempatan > 0 && !loginBerhasil) {
    cout << "Masukkan Nama: ";
    cin.ignore();
    getline(cin, nama);
    cout << "Masukkan NIM: ";
    cin >> nim;

    for (int i = 0; i < db.jumlahUser; i++) {
        if (db.users[i].nama == nama && db.users[i].nim == nim) {
            loginBerhasil = true;
            cout << "Login berhasil!\n";
            break;
        }
    }

    if (!loginBerhasil) {
        kesempatan--;
        cout << "Login gagal! " << kesempatan << endl;
    }
}

if (!loginBerhasil) {
    cout << "Anda gagal login 3 kali. Program berhenti.\n";
    return 0;
}

```

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;
}

```

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:

```
if (pilihan == 1) {
    if (db.jumlahProduk == 0) {
        cout << "Tidak ada produk dalam daftar.\n";
    } else {
        cout << "\n===== DAFTAR PRODUK =====\n";
        cout << "+-----+-----+-----+-----+\n";
        cout << "| No | Nama Produk | Stok | Harga/Kg |\n";
        cout << "+-----+-----+-----+-----+\n";
        for (int i = 0; i < db.jumlahProduk; i++) {
            cout << "| " << setw(2) << i + 1 << " | " << setw(13) <<
db.produk[i].nama
                        << " | " << setw(4) << db.produk[i].stok
                        << " | " << setw(8) << db.produk[i].harga << " |\n";
        }
        cout << "+-----+-----+-----+-----+\n";
    }
} else if (pilihan == 2) {
    if (db.jumlahProduk < MAX_PRODUK) {
        cout << "Masukkan Nama Produk: ";
        cin.ignore();
        getline(cin, db.produk[db.jumlahProduk].nama);
        cout << "Masukkan Stok Produk: ";
        cin >> db.produk[db.jumlahProduk].stok;
        cout << "Masukkan Harga per Kg: ";
        cin >> db.produk[db.jumlahProduk].harga;

        db.jumlahProduk++;
        cout << "Produk berhasil ditambahkan!\n";
    } else {
        cout << "Kapasitas penuh!\n";
    }
} else if (pilihan == 3) {
    int index;
    cout << "Masukkan nomor produk yang ingin diubah: ";
    cin >> index;

    if (index < 1 || index > db.jumlahProduk) {
        cout << "Nomor produk tidak valid!\n";
    } else {
        cout << "Masukkan Nama Baru: ";
        cin.ignore();
    }
}
```

```

        getline(cin, db.produk[index - 1].nama);
        cout << "Masukkan Stok Baru: ";
        cin >> db.produk[index - 1].stok;
        cout << "Masukkan Harga Baru: ";
        cin >> db.produk[index - 1].harga;
        cout << "Produk berhasil diubah!\n";
    }
} else if (pilihan == 4) {
    int index;
    cout << "Masukkan nomor produk yang ingin dihapus: ";
    cin >> index;

    if (index < 1 || index > db.jumlahProduk) {
        cout << "Nomor produk tidak valid!\n";
    } else {
        for (int i = index - 1; i < db.jumlahProduk - 1; i++) {
            db.produk[i] = db.produk[i + 1];
        }
        db.jumlahProduk--;
        cout << "Produk berhasil dihapus!\n";
    }
} else if (pilihan == 5) {
    cout << "Terima kasih telah menggunakan program ini!\n";
} else {
    cout << "Pilihan tidak valid!\n";
}
} while (pilihan != 5);

return 0;
}

```


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 Gagal Login

```

===== 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. 4 Tambah 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. 3 Tampilkan Produk

```

===== MENU =====
1. Tampilkan Produk
2. Tambah Produk
3. Ubah Produk
4. Hapus Produk
5. Keluar
Pilihan: 3
Masukkan nomor produk yang ingin diubah: 1
Masukkan Nama Baru: sapi
Masukkan Stok Baru: 22
Masukkan Harga Baru: 57000
Produk berhasil diubah!

===== DAFTAR PRODUK =====
+---+-----+---+-----+
| No | Nama Produk | Stok | Harga/Kg |
+---+-----+---+-----+
| 1  | sapi       | 22   | 57000    |
+---+-----+---+-----+

```

Gambar 4. 6 Ubah Produk

```

===== MENU =====
1. Tampilkan Produk
2. Tambah Produk
3. Ubah Produk
4. Hapus Produk
5. Keluar
Pilihan: 4
Masukkan nomor produk yang ingin dihapus: 1
Produk berhasil dihapus!

```

Gambar 4. 5 Hapus Produk

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> █
```

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> █
```

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 3"
```

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
```

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