

LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN LANJUT



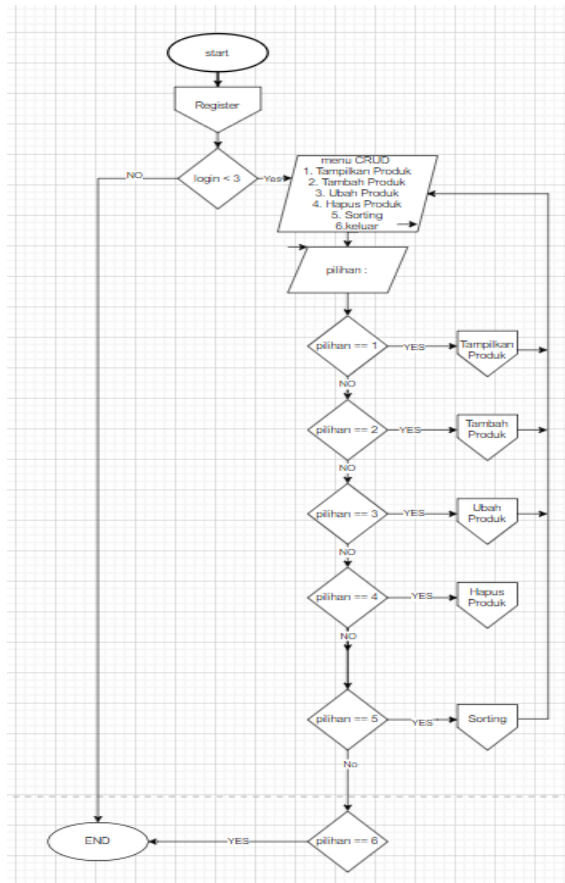
Disusun oleh:

Ayu Azzahrah Alwi (2409106022)

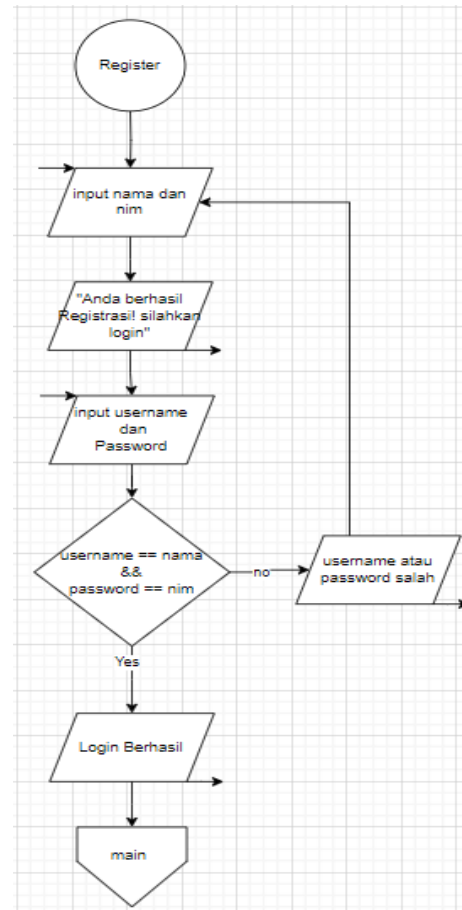
Kelas A1'24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

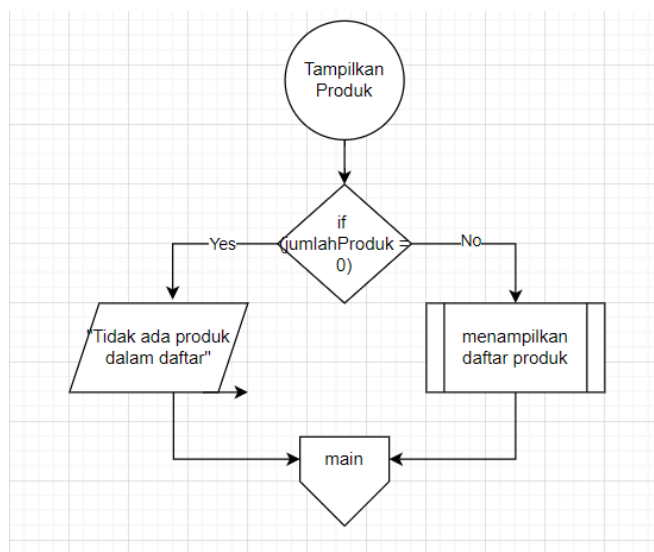
1. Flowchart



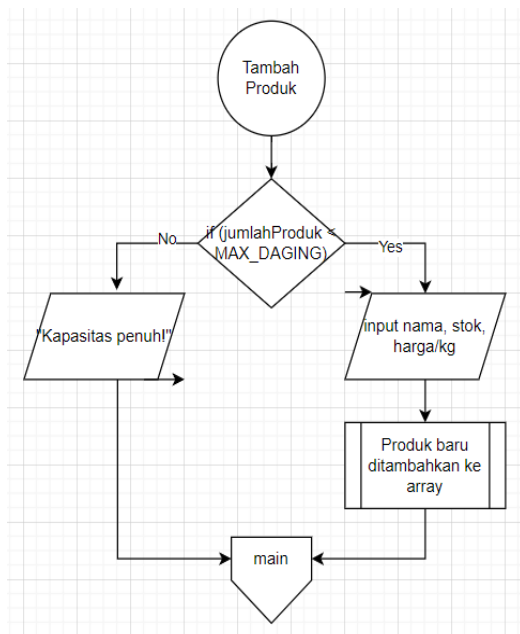
Gambar 1. 1 Main Flow



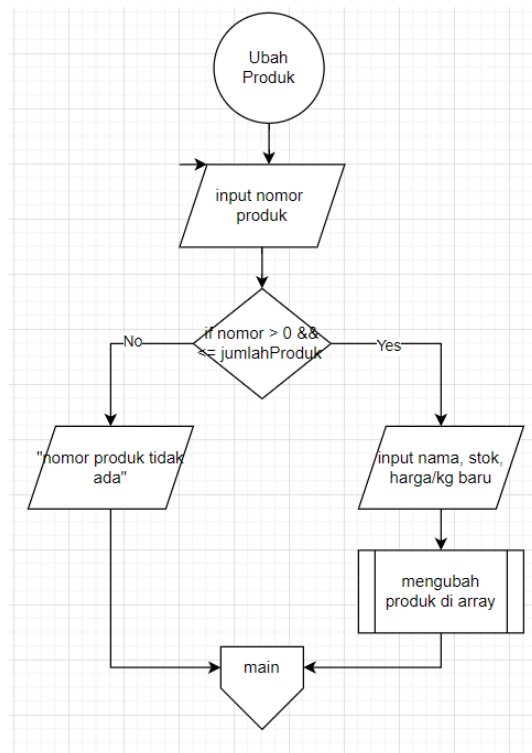
Gambar 1. 2 Register dan Login



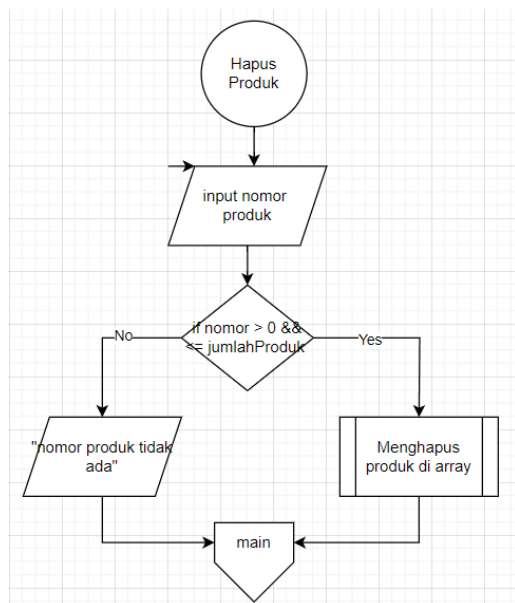
Gambar 1. 3 Tampilkan Produk



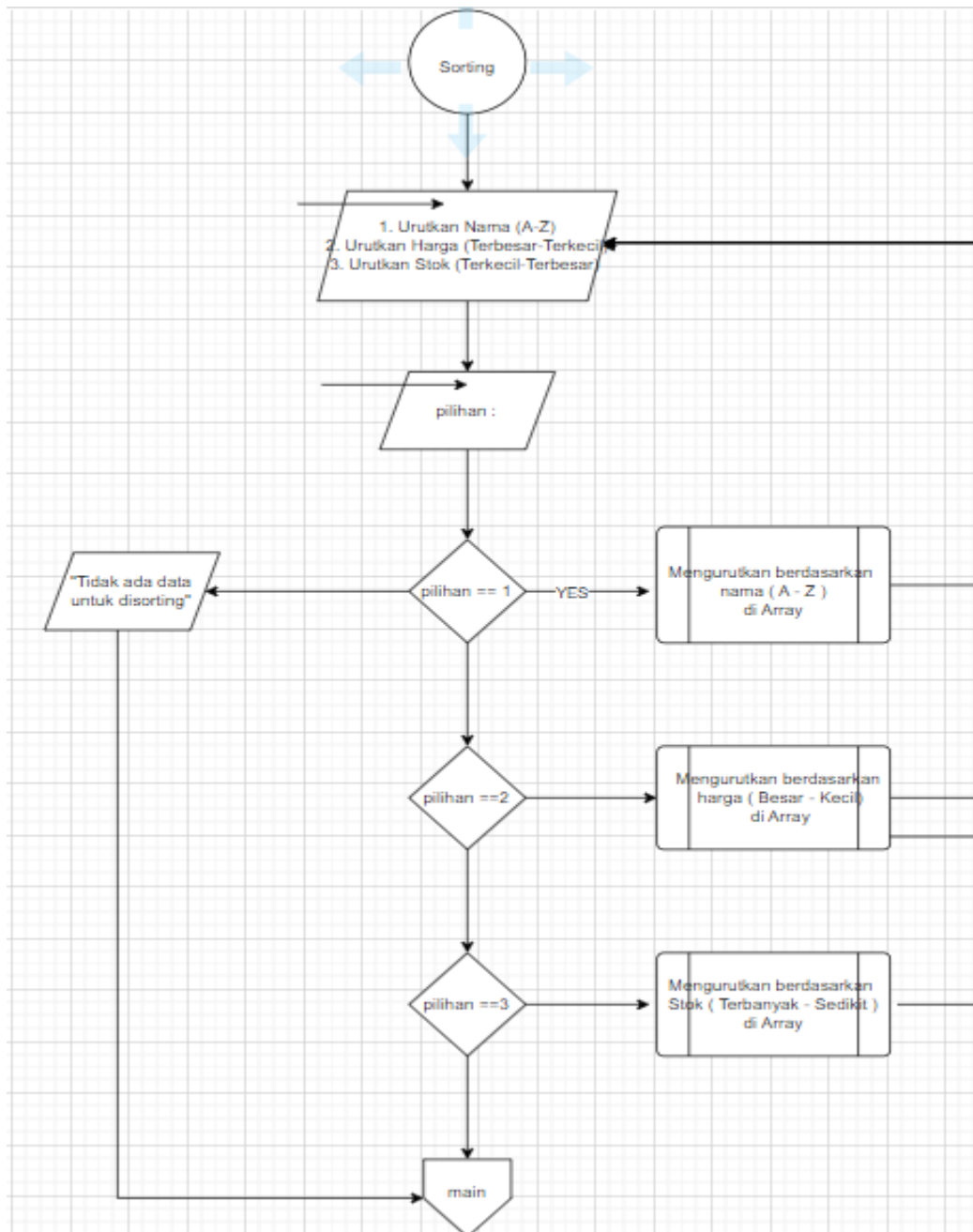
Gambar 1. 4 Tambah Produk



Gambar 1. 5 Ubah Produk



Gambar 1. 6 Hapus Produk



Gambar 1. 7 Menu Sorting

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) {
    cout << "\n==== REGISTRASI USER ==== \n";
    if (*jumlahUser < MAX_USER) {
        cout << " Masukkan Nama          : ";
        cin.ignore();
        getline(cin, users[*jumlahUser].nama);
        cout << " Masukkan NIM (sebagai PIN) : ";
        getline(cin, users[*jumlahUser].nim);
        (*jumlahUser)++;
        cout << "\nRegistrasi berhasil! Silakan login untuk melanjutkan.\n";
    } else {
        cout << "\nKuota user sudah penuh! Tidak bisa registrasi lagi.\n";
    }
}

bool login(User* users, int jumlahUser, int kesempatan = 3) {
    if (kesempatan == 0) return false;
```

```

string nama, nim;
cout << "\n===== LOGIN USER =====\n";
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) {
        cout << "\nLogin berhasil! Selamat datang, " << nama << ".\n";
        return true;
    }
}

cout << "\nLogin gagal! Nama atau NIM salah.\n";
cout << "Sisa kesempatan: " << kesempatan - 1 << "\n";
return login(users, jumlahUser, kesempatan - 1);
}

```

C. Menu

Terdapat beberapa menu untuk memilih fitur pada aplikasi

Source Code :

```

do {
    cout << "\n===== MENU UTAMA =====\n";
    cout << "1. Tampilkan Produk\n";
    cout << "2. Tambah Produk\n";
    cout << "3. Ubah Produk\n";
    cout << "4. Hapus Produk\n";
    cout << "5. Sorting Produk\n";
    cout << "6. 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:
            menuSorting(produk, jumlahProduk);
            break;
        case 6:
            cout << "Terima kasih telah menggunakan program!\n";
            break;
        default:
            cout << "Pilihan tidak valid!\n";
    }

    } while (pilihan != 6);

    return 0;
}

```

D. 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";
}

```

```
}
```

E. Menu Sorting

Menu sorting ini berfungsi untuk memberikan beberapa opsi kepada pengguna untuk mengurutkan data produk berdasarkan kriteria tertentu. Data produk tersebut bisa diurutkan berdasarkan:

1. Nama produk (A-Z)
2. Harga produk (dari yang terbesar ke terkecil)
3. Stok produk (dari yang terkecil ke terbesar) menggunakan algoritma Bubble Sort.

Source Code :

```
void sortNamaAscending(Produk* produk, int jumlahProduk) {
    for (int i = 0; i < jumlahProduk - 1; i++) {
        for (int j = 0; j < jumlahProduk - i - 1; j++) {
            if (produk[j].nama > produk[j + 1].nama) {
                swap(produk[j], produk[j + 1]);
            }
        }
    }
    cout << "Produk berhasil diurutkan berdasarkan nama (A-Z).\n";
}

void sortHargaDescending(Produk* produk, int jumlahProduk) {
    for (int i = 0; i < jumlahProduk - 1; i++) {
        for (int j = 0; j < jumlahProduk - i - 1; j++) {
            if (produk[j].harga < produk[j + 1].harga) {
                swap(produk[j], produk[j + 1]);
            }
        }
    }
    cout << "Produk berhasil diurutkan berdasarkan harga (terbesar ke terkecil).\n";
}

void bubbleSortStokAscending(Produk* produk, int jumlahProduk) {
    for (int i = 0; i < jumlahProduk - 1; i++) {
        for (int j = 0; j < jumlahProduk - i - 1; j++) {
            if (produk[j].stok > produk[j + 1].stok) {
                swap(produk[j], produk[j + 1]);
            }
        }
    }
}
```

```

    }
}
    cout << "Produk berhasil diurutkan berdasarkan stok (terkecil ke
terbesar) menggunakan Bubble Sort.\n";
}

void menuSorting(Produk* produk, int jumlahProduk) {
    int pilihan;
    do {
        cout << "\n===== MENU SORTING =====\n";
        cout << "1. Urutkan berdasarkan Nama (A-Z)\n";
        cout << "2. Urutkan berdasarkan Harga (Terbesar-Terkecil)\n";
        cout << "3. Urutkan berdasarkan Stok (Terkecil-Terbesar) - Bubble
Sort\n";
        cout << "4. Kembali ke Menu Utama\n";
        cout << "Pilihan: ";
        cin >> pilihan;
    }
}

```

4. Uji Coba dan Hasil Output

```
===== REGISTRASI USER =====
Masukkan Nama          : Ayu13
Masukkan NIM (sebagai PIN) : 22

Registrasi berhasil! Silakan login untuk melanjutkan.

===== LOGIN USER =====
Masukkan Nama : Ayu13
Masukkan NIM : 22

Login berhasil! Selamat datang, yu13.

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

Gambar 4. 1 Berhasil Login

```
===== MENU UTAMA =====
1. Tampilkan Produk
2. Tambah Produk
3. Ubah Produk
4. Hapus Produk
5. Sorting Produk
6. Keluar
Pilihan: 2
Masukkan Nama Produk: Jerapah
Masukkan Stok Produk: 42
Masukkan Harga per Kg: 83400
Produk berhasil ditambahkan.
```

Gambar 4. 2 OutPut Tambah Produk

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

===== DAFTAR PRODUK =====
+---+-----+-----+-----+
| No | Nama Produk | Stok | Harga/Kg |
+---+-----+-----+-----+
| 1  | Ayam       | 321  | 21300    |
| 2  | Kelinci    | 22   | 12433    |
| 3  | Zebra      | 342  | 23600    |
| 4  | Jerapah    | 42   | 83400    |
+---+-----+-----+-----+
```

Gambar 4. 3 OutPut Tampilkan Produk

```

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

===== DAFTAR PRODUK =====
+---+-----+-----+-----+
| No | Nama Produk | Stok | Harga/Kg |
+---+-----+-----+-----+
| 1  | Ayam       | 321  | 21300    |
| 2  | Kelinci    | 22   | 12433    |
| 3  | Zebra      | 342  | 23600    |
| 4  | Jerapah    | 42   | 83400    |
+---+-----+-----+-----+
Pilih produk yang akan diubah: 2
Masukkan Nama Baru: Anak Kelinci
Masukkan Stok Baru: 42
Masukkan Harga Baru: 10993
Produk berhasil diubah!

```

Gambar 4. 4 OutPut Ubah Produk

```

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

===== DAFTAR PRODUK =====
+---+-----+-----+-----+
| No | Nama Produk | Stok | Harga/Kg |
+---+-----+-----+-----+
| 1  | Ayam       | 321  | 21300    |
| 2  | Anak Kelinci | 42   | 10993    |
| 3  | Zebra      | 342  | 23600    |
| 4  | Jerapah    | 42   | 83400    |
+---+-----+-----+-----+
Pilih produk yang akan dihapus: 3
Produk berhasil dihapus!

```

Gambar 4. 5 OutPut Hapus Produk

```

Pilihan: 1

===== DAFTAR PRODUK =====
+---+-----+-----+-----+
| No | Nama Produk | Stok | Harga/Kg |
+---+-----+-----+-----+
| 1  | Ayam       | 321  | 21300    |
| 2  | Anak Kelinci | 42   | 10993    |
| 3  | Jerapah    | 42   | 83400    |
+---+-----+-----+-----+

```

Gambar 4. 6 Daftar Produk Setelah di Ubah dan Hapus

```

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

===== MENU SORTING =====
1. Urutkan berdasarkan Nama (A-Z)
2. Urutkan berdasarkan Harga (Terbesar-Terkecil)
3. Urutkan berdasarkan Stok (Terkecil-Terbesar) - Bubble Sort
4. Kembali ke Menu Utama
Pilihan: 

```

Gambar 4. 7 Menu Sorting

```

Pilihan: 1
Produk berhasil diurutkan berdasarkan nama (A-Z).

===== DAFTAR PRODUK =====
+---+-----+---+-----+
| No | Nama Produk | Stok | Harga/Kg |
+---+-----+---+-----+
| 1 | Anak Kelinci | 42 | 10993 |
| 2 | Ayam | 321 | 21300 |
| 3 | Jerapah | 42 | 83400 |
+---+-----+---+-----+

```

Gambar 4. 8 Sorting Berdasarkan Nama

```

Pilihan: 2
Produk berhasil diurutkan berdasarkan harga (terbesar ke terkecil).

===== DAFTAR PRODUK =====
+---+-----+---+-----+
| No | Nama Produk | Stok | Harga/Kg |
+---+-----+---+-----+
| 1 | Jerapah | 42 | 83400 |
| 2 | Ayam | 321 | 21300 |
| 3 | Anak Kelinci | 42 | 10993 |
+---+-----+---+-----+

```

Gambar 4. 9 Sorting Berdasarkan Harga

```

Pilihan: 3
Produk berhasil diurutkan berdasarkan stok (terkecil ke terbesar) menggunakan Bubble Sort.

===== DAFTAR PRODUK =====
+---+-----+---+-----+
| No | Nama Produk | Stok | Harga/Kg |
+---+-----+---+-----+
| 1 | Jerapah | 42 | 83400 |
| 2 | Anak Kelinci | 42 | 10993 |
| 3 | Ayam | 321 | 21300 |
+---+-----+---+-----+

```

Gambar 4. 10 Sorting Berdasarkan Stok

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 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 "Finis Post Test 6"
[main 69461d2] Finis Post Test 6
3 files changed, 246 insertions(+), 2 deletions(-)
create mode 100644 post-test/post-tset-6/2409106022-AyuAzzhahrahAlwi-PT-6.cpp
create mode 100644 post-test/post-tset-6/2409106022-AyuAzzhahrahAlwi-PT-6.exe
PS C:\Users\ASUS GK\OneDrive\Dokumen\praktikum-apl>
```

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