**Tugas Kelompok Dasar Pemrograman**

**(materi : OOP)**

**Tema : OOP CRUD Sederhana system inventory barang Toko Komputer**

**Disusun Oleh:**

1. **Stevanus Andika Galih Setiawan(202303110008)**
2. **Rheza Nanda()**

****

**JURUSAN REKAYASA PERANGKAT LUNAK**

**UNIVERSITAS IPWIJA**

**TAHUN 2023]**

**A.** Analisis Kebutuhan dan Landasan Teori:

1. Tujuan Program:

Program ini dirancang untuk mengelola inventaris barang dengan beberapa operasi dasar seperti menambah, menghapus, mengedit, dan menampilkan barang.

2. Kebutuhan Fungsional:

Tambah Barang (Menu 1):

Meminta input nama, jumlah, dan kategori barang dari pengguna.

Menambahkan barang baru ke dalam daftar barang.

Hapus Barang (Menu 2):

Meminta input ID barang yang akan dihapus.

Menggunakan LINQ untuk mencari dan menghapus barang berdasarkan ID.

Edit Barang (Menu 3):

Meminta input ID barang yang akan diubah.

Menggunakan LINQ untuk mencari barang berdasarkan ID.

Meminta input baru untuk nama, jumlah, dan kategori barang.

Mengubah data barang sesuai dengan input pengguna.

Tampilkan Semua Barang (Menu 4):

Menampilkan semua barang dalam format tabel jika ada barang dalam daftar.

Menampilkan pesan jika belum ada barang dalam sistem.

Keluar dari Program (Menu 5):

Menyajikan opsi untuk keluar dari program.

3. Kebutuhan Non-Fungsional:

Interaktif:

Program berinteraksi dengan pengguna melalui konsol.

Persistensi Data:

Data barang disimpan dalam memori selama program berjalan. Data akan hilang setelah program ditutup.

Validasi Input:

Input pengguna perlu divalidasi untuk memastikan bahwa mereka sesuai dengan tipe data yang diharapkan.

User-Friendly:

Menyediakan pesan yang jelas dan informatif kepada pengguna.

4. Landasan Teori:

List<T>:

Digunakan untuk menyimpan daftar barang (List<Barang>).

Mempermudah penambahan, penghapusan, dan pencarian item.

LINQ (Language Integrated Query):

Digunakan untuk mencari dan mengelola objek dalam daftar berdasarkan kondisi tertentu.

Rekursi:

Digunakan dalam fungsi TampilkanMenu() untuk menampilkan menu secara berulang setelah setiap operasi.

Properties:

Menerapkan properti pada kelas Barang untuk mengakses dan memodifikasi data barang dengan lebih terstruktur.

Exception Handling:

Perlu diterapkan untuk menangani kesalahan input pengguna yang mungkin terjadi saat mengonversi string menjadi tipe data numerik.

Kode Program :

using System;

using System.Collections.Generic;

class Barang

{

public int ID { get; set; }

public string Nama { get; set; }

public int Jumlah { get; set; }

public string Kategori { get; set; }

}

class Program

{

// Inisialisasi daftar barang dan ID awal

static List<Barang> daftarBarang = new List<Barang>();

static int currentID = 1;

static void Main()

{

// Menampilkan informasi kelompok

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("| Tugas Kelompok Dasar Pemrograman - OOP |");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

Console.WriteLine("Nama : 1.Stevanus Andika Galih Setiawan");

Console.WriteLine(" : 2.Rheza Nanda \n");

Console.WriteLine("NIM : 1.202303110008 ");

Console.WriteLine("NIM : 2. \n");

Console.WriteLine("Prodi : Rekayasa Perangkat Lunak \n");

Console.WriteLine("Kelas : RK231 \n");

// Memanggil fungsi untuk menampilkan menu

TampilkanMenu();

}

static void TampilkanMenu()

{

// Menampilkan menu utama

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

Console.WriteLine("Program Inventory Toko Komputer");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

Console.WriteLine("1. Tambah Barang");

Console.WriteLine("2. Hapus Barang");

Console.WriteLine("3. Edit Barang");

Console.WriteLine("4. Tampilkan Semua Barang");

Console.WriteLine("5. Keluar");

Console.Write("Pilih opsi (1-5):");

// Membaca pilihan dari pengguna

int pilihan = int.Parse(Console.ReadLine());

// Memilih tindakan berdasarkan pilihan pengguna

switch (pilihan)

{

case 1:

TambahBarang();

break;

case 2:

HapusBarang();

break;

case 3:

EditBarang();

break;

case 4:

TampilkanSemuaBarang();

break;

case 5:

// Keluar dari program

Console.WriteLine("Anda telah keluar dari program...");

Console.WriteLine("Terimakasih telah menggunakan program inventory barang");

Environment.Exit(0);

break;

default:

Console.WriteLine("Pilihan tidak valid. Silakan pilih lagi.");

break;

}

// Kembali menampilkan menu setelah aksi selesai

TampilkanMenu();

}

static void TambahBarang()

{

// Meminta input informasi barang baru dari pengguna

Console.Write("Nama Barang: ");

string nama = Console.ReadLine();

Console.Write("Jumlah Barang: ");

int jumlah = int.Parse(Console.ReadLine());

Console.WriteLine("Pilih Kategori Barang (1. Elektronik, 2. Aksesoris, 3. PC Desktop,4.Lainnya): ");

int kategoriChoice = int.Parse(Console.ReadLine());

// Menentukan kategori berdasarkan pilihan pengguna

string kategori;

switch (kategoriChoice)

{

case 1:

kategori = "Elektronik";

break;

case 2:

kategori = "Aksesoris";

break;

case 3:

kategori = "PC Dekstop";

break;

default:

kategori = "Lainnya";

break;

}

// Membuat objek Barang baru dan menambahkannya ke daftar barang

Barang barangBaru = new Barang

{

ID = currentID++,

Nama = nama,

Jumlah = jumlah,

Kategori = kategori

};

daftarBarang.Add(barangBaru);

Console.WriteLine("Barang berhasil ditambahkan!");

}

static void HapusBarang()

{

// Meminta input ID barang yang akan dihapus

Console.Write("Masukkan ID Barang yang akan dihapus: ");

int idHapus = int.Parse(Console.ReadLine());

// Mencari barang dalam daftar berdasarkan ID

Barang barangHapus = daftarBarang.Find(barang => barang.ID == idHapus);

// Menghapus barang jika ditemukan, jika tidak memberikan pesan

if (barangHapus != null)

{

daftarBarang.Remove(barangHapus);

Console.WriteLine("Barang berhasil dihapus!");

}

else

{

Console.WriteLine("Barang tidak ditemukan.");

}

}

static void EditBarang()

{

// Meminta input ID barang yang akan diubah

Console.Write("Masukkan ID Barang yang akan diubah: ");

int idEdit = int.Parse(Console.ReadLine());

// Mencari barang dalam daftar berdasarkan ID

Barang barangEdit = daftarBarang.Find(barang => barang.ID == idEdit);

// Mengubah informasi barang jika ditemukan, jika tidak memberikan pesan

if (barangEdit != null)

{

Console.Write("Nama Barang baru: ");

barangEdit.Nama = Console.ReadLine();

Console.Write("Jumlah Barang baru: ");

barangEdit.Jumlah = int.Parse(Console.ReadLine());

Console.WriteLine("Pilih Kategori Barang (1. Elektronik, 2. Aksesoris, 3. PC,4.Lainnya): ");

int kategoriChoice = int.Parse(Console.ReadLine());

// Menentukan kategori berdasarkan pilihan pengguna

switch (kategoriChoice)

{

case 1:

barangEdit.Kategori = "Elektronik";

break;

case 2:

barangEdit.Kategori = "Aksesoris";

break;

case 3:

barangEdit.Kategori = "Lainnya";

break;

default:

barangEdit.Kategori = "Lainnya";

break;

}

Console.WriteLine("Data Barang berhasil diubah!");

}

else

{

Console.WriteLine("Barang tidak ditemukan.");

}

}

static void TampilkanSemuaBarang()

{

// Menampilkan semua barang dalam daftar jika ada, jika tidak memberikan pesan

if (daftarBarang.Count > 0)

{

Console.WriteLine("ID\tNama\tJumlah\tKategori");

foreach (Barang barang in daftarBarang)

{

Console.WriteLine($"{barang.ID}\t{barang.Nama}\t{barang.Jumlah}\t{barang.Kategori}");

}

}

else

{

Console.WriteLine("Belum ada barang dalam sistem.");

}

}}

Output Program :

Logika program :

1.Definisikan kelas barang yang berisi property :

- ID

- Nama

- Jumlah

- Kategori

2. Kelas program :

Mendeklarasikan list “daftarbarang” untuk menyimpan barang – barang.

Mendeklarasikan ID barang untuk menampilkan ID barang dan sekaligus ID Barang akan dijadikan sebagai nomor yang akan ditampilkan(dalam hal ini,ID Barang dimulai dari 1).

Dan main method adalah method/entry program.

3. Menu Utama (method tampilkan menu) :

Menampilkan menu operasi utama dengan operasi CRUD

Meminta dan membaca inputan untuk memilih inputan menu.

Menjalankan method/function untuk menjalankan setiap fungsi yang sesuai dengan inputan yang dipilih oleh user.

Jika inputan tidak sesuai dengan nomor,maka system akan meminta inputan Kembali kepada user.

Menggunakan rekursif/loop untuk terus menampilkan menu setelah setiap operasi selesai dieksekusi.

4.method tambah barang :

Mengambil input dari user berupa nama,jumlah dan kategori

Membuat objek “barangBaru” akan ditambahkan kedalam list “daftarBarang”dengan parameter :

Dimana ID akan bertambah setiap barang baru ditambahkan,

Var Nama,Jumlah, dan Kategori diambil dari inputan user.dan pilih kategori menggunakan switch case

5.Method hapus barang :

Meminta input ID/ Nomor barang yang akan dihapus oleh pengguna.

Menggunakan Linq untuk mencari barang berdasarkan ID.

Jika barang dan ID Ditemukan,maka hapus dari list “daftarBarang”,

Jika tidak,munculkan pesan error “barang tidak ditemukan”,dan proses hapus barang tidak bisa dijalankan

6. Method Edit barang :

Meminta input ID/nomor barang yang akan diubah oleh user

Menggunakan Linq untuk mencari barang berdasarkan ID barang.

Jika barang ditemukan,tampilkan input baru untuk user,dengan isi property sebagai berikut :

Nama

Jumlah

Edit Kategori barang

Lalu perbarui propertinya

7. method tampilkansemuabarang()

Lakukan pengecekan,jika jumlah lebih dari 0,tampilkan semua informasi yang terdiri dari ID,Nama barang,jumlah dan kategori barang.

Dan tampilkan barang dengan menggunakan perulangan foreach.

Jika tidak ada barang,tampilkan pesan “belum ada barang dalam system”