

UJIAN AKHIR SEMESTER (UAS)

SEMETER II

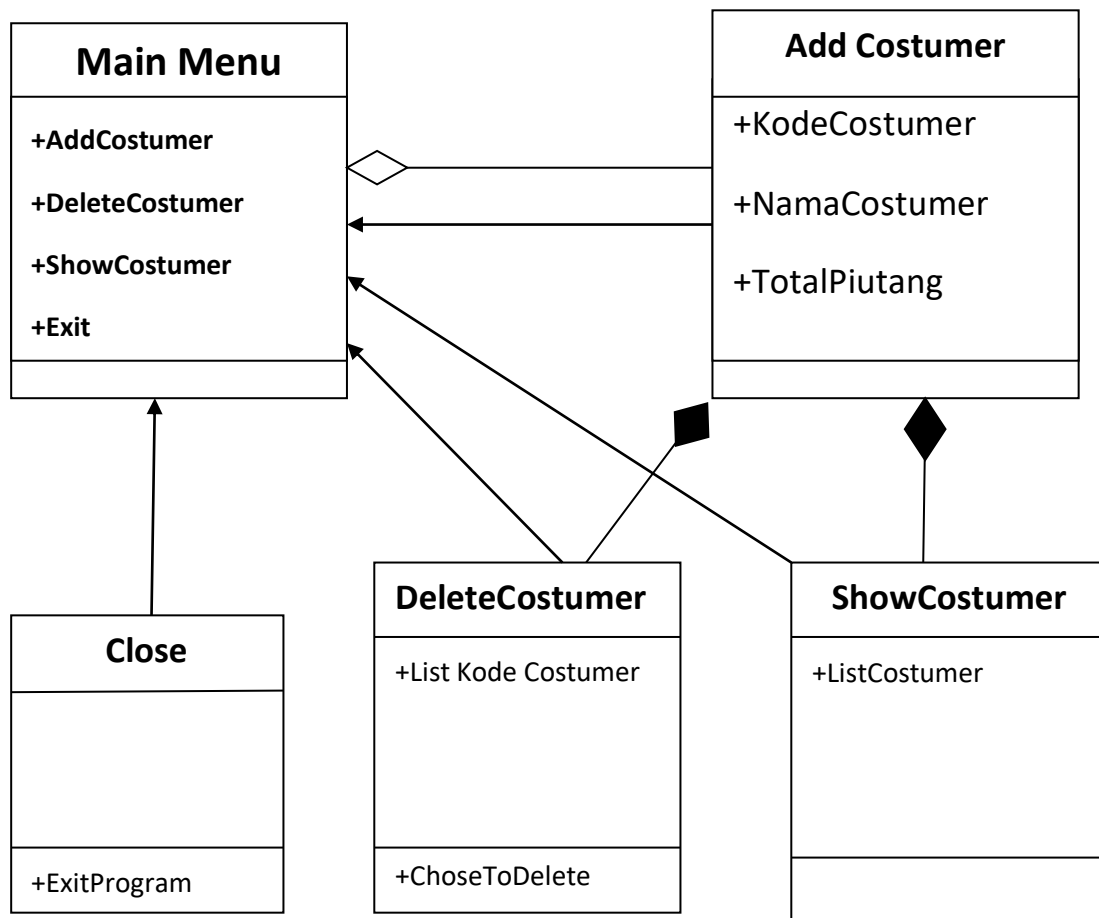
Dosen : Kamarudin M,Kom.
Mata kuliah : Pemrograman



NAMA : UNTUNG RIAN TO
KELAS : INFORMATIKA 07
NIM : 18.11.2302

PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
2019

Class Diagram



Keterangan :

Pada ilustrasi diagram, setiap kotak memiliki tiga bagian atau dibagi menjadi 3 baris. Yang mana memiliki keterangan ; baris pertama adalah nama kelas, baris kedua adalah properti kelas, dan baris ketiga adalah method. Ini lumrah digunakan dalam membuat diagram kelas.

Dalam ilustrasi diagram kelas diatas , Menu utama(Main menu) menjadi yang dipusatkan , artinya ia menjadi awal sekaligus inti dari program itu sendiri. Ia mempunyai property : AddCostumer, DeleteCostumer, ShowCostumer, dan Exit.

Untuk lebih jelasnya berikut keterangan symbol relationship:

- : menunjukkan bahwa kelas ini digunakan oleh kelas lain yang ditunjuk
- ◇ : hubungan antar kelas dengan makna semua bagian
- ◆ : hubungan antar kelas yang berarti kelas ini tak dapat berdiri sendiri, ia memerlukan kelas lain dan ia bagian dari kelas yang ditunjuk.

Penjelasan Program

```
using System;                                //Library yang dibutuhkan program
using System.Collections.Generic;            //Library yang dibutuhkan program
using System.Linq;                          //Library yang dibutuhkan program
using System.Text;                          //Library yang dibutuhkan program
using System.Threading.Tasks;               //Library yang dibutuhkan program

namespace ProjectCustomer
{
    class Program
    {
        // deklarasi objek collection untuk menampung objek customer
        static List<Customer> daftarCustomer = new List<Customer>();

        static void Main(string[] args)
        {
            Console.Title = "Responsi UAS Matakuliah Pemrograman";

            while (true) // Menggunakan struktur perulangan
            {
                TampilMenu();

                Console.Write("\nNomor Menu [1..4]: ");
                int nomorMenu = Convert.ToInt32(Console.ReadLine());

                switch (nomorMenu) //menggunakan struktur percabangan Switch-Case
                {
                    case 1:
                        TambahCustomer();
                        break;

                    case 2:
                        HapusCustomer();
                        break;

                    case 3:
                        TampilCustomer();
                        break;

                    case 4: Keluar           // keluar dari program
                        return;

                    default:
                        break;
                }
            }
        }
    }
}
```

```

static void TampilMenu()
{
    Console.Clear();

    Console.WriteLine("Pilih Menu Aplikasi\n1. Tambah Customer\n2. Hapus
Costumer\n3. Tampilkan Customer\n4. Keluar");
}

static void TambahCustomer()
{
    Console.Clear();

    Console.WriteLine("Tambah Data Customer\n\n");
    Customer newCustomer = new Customer();
    Console.Write("Kode Customer: ");
    newCustomer.KodeCust = Console.ReadLine();
    Console.Write("Nama Customer: ");
    newCustomer>NamaCust = Console.ReadLine();
    Console.Write("Total Piutang: ");
    newCustomer.TotalPiutang = Convert.ToInt32(Console.ReadLine());

    daftarCustomer.Add(newCustomer);

    Console.WriteLine("\nTekan ENTER untuk kembali ke menu");
    Console.ReadKey();
}

static void HapusCustomer() /*metode penghapusan berdasar kode costumer, yang
disimpan dalam index */
{
    Console.Clear();

    Console.Write("Masukkan kode customer: ");
    string inputKode = Console.ReadLine();

    List<int> totalFound = new List<int>();
    int index = 0;
    foreach(Customer findCust in daftarCustomer)
    {
        if (findCust.KodeCust == inputKode)
        {
            totalFound.Add(index);
        }
        index++;
    }
    if(totalFound.Count == 0)
    {
        Console.WriteLine("\nKode Customer tidak ditemukan");
    }
}

```

```

else
{
    for (int i = totalFound.Count; i > 0; i--)
    {
        daftarCustomer.RemoveAt(totalFound[i - 1]);
    }

    Console.WriteLine("\nData Customer berhasil dihapus");
}

Console.WriteLine("\nTekan ENTER untuk kembali ke menu");
Console.ReadKey();
}

static void TampilCustomer()
{
    Console.Clear();

    Console.WriteLine("Data Customer\n\n");
    int index = 0;
    foreach (Customer printCust in daftarCustomer)
    {
        Console.WriteLine("{0}. {1}, {2}, {3}", index + 1, printCust.KodeCust,
printCust>NamaCust, printCust.TotalPiutang);
        index++;
    }

    Console.WriteLine("\nTekan enter untuk kembali ke menu");
    Console.ReadKey();
}

```

No Plagiasi();