

PEMROGRAMAN WEB LANJUTAN

ALOYSIUS ARI WICAKSONO

Topik

- ▶ Looping
- ▶ OOP
- ▶ List

Pre-Test

- ▶ Apa perbedaan List dan array di C#?

For Loop

- Digunakan untuk looping dengan jumlah iterasi yang pasti

```
for (int i = 0; i < 5; ++i)
{
    // block of code to be executed if condition1 is True
}
```

While Loop

- Digunakan untuk looping dengan jumlah iterasi yang tidak pasti

```
while (!Finish)
{
    // block of code to be executed if condition1 is True
    if(a == 5)
        Finish = true;
}
```

Object Oriented Programming

- ▶ Encapsulation
- ▶ Inheritance
- ▶ Polymorphism
- ▶ Access Modifier
- ▶ Static function

Object Oriented Programming

- ▶ C# menggunakan full OOP
- ▶ Setiap halaman akan dibuat 1 class
- ▶ Setiap task akan dibungkus menjadi function (encapsulation)
- ▶ Contoh : OnGet

Inheritance

- ▶ Sudah ada beberapa base class bawaan dari .Net Core
- ▶ Contoh : ketika Membuat page baru, model akan inherit dari class PageModel

Polymorphism

- ▶ Kita bisa memberikan nilai default untuk function parameter
- ▶ Jika tanpa parameter n2, otomatis akan terisi 5

```
public int perkalian (int n1, int n2 = 5)
{
    return n1 * n2;
}

public void OnGet()
{
    perkalian(5, 4); // hasil : 20
    perkalian(5);    // hasil : 25
}
```

Access Modifier

- ▶ Public : bisa diakses dari luar kelas tersebut
- ▶ Protected : hanya bisa diakses dirinya dan kelas anaknya
- ▶ Private : hanya bisa diakses dari dalam kelas tersebut

Access Modifier

- ▶ Jika terdapat function / variable dari Model yang ingin agar bisa diakses dari View, gunakan access “public”
- ▶ Menggunakan akses private / protected tidak akan bisa diakses dari View

Static

- ▶ C# menggunakan full OOP
- ▶ Tidak bisa terdapat function atau variable yang berdiri sendiri
- ▶ Terkadang kita butuh Membuat function helper yang bisa digunakan di banyak tempat tanpa perlu membuat objek dari kelas tersebut
- ▶ Gunakan static
- ▶ Pada static, function / variable bisa dipanggil tanpa perlu membuat object dari kelas tersebut.

Static

- ▶ Pada contoh di sebelah, untuk memanggil perkalian, kita cukup menggunakan nama kelas.nama function
- ▶ Helper.Perkalian

```
public static class Helper
{
    public static int Perkalian(int n1, int n2)
    {
        return n1 * n2;
    }
}
```

Untuk memanggil dari kelas lain, gunakan :

```
Helper.Perkalian(5, 4);
```

List

- ▶ Array dinamis
- ▶ Bisa menambah atau mengurangi isi dari list saat runtime
- ▶ Bisa menambah di akhir, menambah di awal, di tengah, dst

```
List<Bank> lstBank = new List<Bank>();  
lstBank.Add(new Bank { BankID = 1, BankName = "BCA" });  
lstBank.Insert(0, new Bank { BankID = 2, BankName = "BCA" });
```

Foreach

- Untuk looping dari array / object

```
foreach(Bank bank in lstBank)
{
}
```

```
for (int i = 0; i < lstBank.Count(); ++i) {
    Bank bank = lstBank[i];
}
```

Pemrograman Web Lanjutan

SEKIAN DAN TERIMA KASIH