



Pengenalan Pemrograman Fungsional



Oleh : Agus Priyanto, M.Kom



INSTITUT TEKNOLOGI TELKOM PURWOKERTO
Smart, Trustworthy, And Teamwork



Outline Materi

- Paradigma Pemrograman
- Pemrograman Fungsional
- Python Basic

Paradigma Pemrograman

a. Pemrograman Prosedural

- Paradigma ini didasari oleh Konsep mesin Von Newman (stored program concept)
- Algoritma + Struktur Data = Program

b. Pemrograman Berorientasi Objek

- Paradigma ini didasari oleh Kelas dan Objek
- Objek adalah instansiasi dari Kelas



c. Pemrograman Fungsional

- Didasari konsep pemetaan dari fungsi matematika
- Fungsi dapat berbentuk sebagai fungsi “*primitif*” atau komposisi dari fungsi lainnya



```
#include <iostream.h>
struct data //Deklarasi structure
{
    char name[20];
    char address[20];
    int salary;
};
data employee1; //Inisialisasi variabel structure
int main()
{
    cout << "Employee's name : ";
    cin >> employee1.name;
    cout << "Employee's address : ";
    cin >> employee1.address;
    cout << "Employee's salary : ";
    cin >> employee1.salary;
    cout << endl << "Employee's data : ";
    cout << employee1.name << ", ";
    cout << employee1.address << ", ";
    cout << employee1.salary;
    return 0;
}
```

Pemrograman Prosedural

```

import java.io.*;
class DemoObjectOriented
{
    public void helloWorld(int jmlh_looping)
    {
        System.out.print ("\n-Display Hello World-\n");
        for(int i=0;i< jmlh_looping;i++)
        {
            System.out.print (+(i+1)+". Hello
World....\n");
        }
    }
}

```

Pemrograman Berorientasi Objek

```

public static void main(String args[])
{
    {
        int jumlah_looping=5;
        int bil1=5;
        int bil2=2;
        DemoObjectOriented Demo = new DemoObjectOriented ();
        Demo.helloWorld(jumlah_looping);
        System.out.print ("\n-Pemanggilan Fungsi Tambah-\n");
        System.out.print (" Hasil Operasi Tambah :
"+Demo.tambah(bil1,bil2)+"\n");
    }
}

```



#lambda expression

```
>>> mult_3_5(3)
```

```
True
```

```
>>> mult_3_5(4)
```

```
False
```

```
>>> mult_3_5(5)
```

```
True
```

Pemrograman Fungsional

#Menggunakan fungsi until()

```
>>> until(10, lambda x: x%3==0 or x%5==0, 0)
```

```
[0, 3, 5, 6, 9]
```

#functional hybrid

```
>>> sum( n for n in range(1, 10) if n%3==0 or n%5==0 )
```

```
23
```

