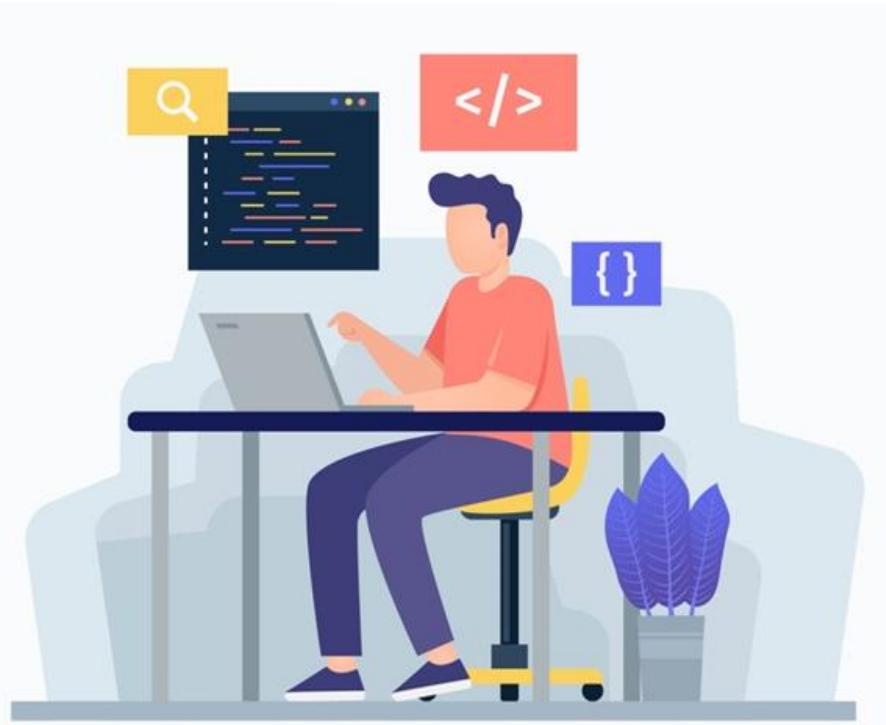




**PROGRAM STUDI**  
**TEKNIK INFORMATIKA**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS DIAN NUSWANTORO**

Mata Kuliah  
**Dasar Pemrograman**



# **Abstract Data Type (ADT)**

TIM DASAR PEMROGRAMAN  
TEKNIK INFORMATIKA S1  
UNIVERSITAS DIAN NUSWANTORO

# Capaian Pembelajaran

1. Menjelaskan konsep abstract data type
2. Merancang struct untuk merepresentasikan data organisasi dalam program prosedural
3. Merancang abstract data type dengan array dalam program
4. Merancang abstract data type dengan pointer dalam program

# Abstract Data Type

- ADT merupakan jenis spesial dari tipe data
- Didefinisikan oleh set dari nilai-nilai dan set operasi dari operasi-operasi
- Kata “Abstract” dapat diartikan sebagai “kita dapat menggunakan tipe data ini dan bisa menggunakan operasi yang berbeda-beda tanpa harus mengetahui bagaimana operasi tersebut bekerja”
- Jadi “Abstract” disini dari sudut pandang user atau pengguna atau programmer yang menggunakan ADT yang kita bangun.
- ADT dibentuk oleh tipe data primitif yang sudah ada, tetapi operasi logika yang ada didalamnya disembunyikan.

# Objective

STRUCTURE

STRUCT

TYPEDEF

ARRAY OF STRUCTURE

POINTER OF STRUCTURE

STRUCTURE & FUNCTION

# Structure

- **Structure** penting dalam computer science karena digunakan dalam pemrograman dan aplikasi seperti relasi database, pemrosesan file, dan konsep OOP
- Berdasarkan tipe data seperti integer dan character, structure sering disebut sebagai ***data structure***
- Structure adalah kumpulan variabel yang berkaitan sering digunakan untuk mendefinisikan object – orang, tempat, atau record dalam database atau file

# STRUCT

- Proses awal dalam membuat structure adalah mendefinisikannya menggunakan **struct**
- Keyword struct digunakan untuk mendefinisikan koleksi variabel dari bermacam tipe yang disebut structure, dimana dapat digunakan sebagai *single unit*
- Dalam penulisan program, keyword struct diikuti oleh brace **{ }** dengan variabel individual didalamnya (member)
- Contoh Penulisan Struct:

```
struct hitung{  
    int x;  
    int y;  
};
```



hitung diketahui sebagai tag structure  
Dimana digunakan untuk membuat  
Deklarasi variable

# STRUCT - DEKLARASI

- Ketika mendefinisikan structure menggunakan struct, memory belum dialokasikan untuk structure, sampai deklarasi variabel dibuat
- Contoh:

---

```
struct hitung{  
    int x;  
    int y;  
}  
struct hitung jumlah;
```

---

- Contoh diatas dapat diartikan bahwa : variable **jumlah** dengan tipe data **hitung**

# STRUCT - INISIALISASI

- **Inisialisasi nilai** pada structure, format penulisannya sama seperti inisialisasi nilai pada array
- Contoh:

---

```
struct hitung{  
    int x;  
    int y;  
}  
struct hitung jumlah;  
Jumlah = {10,20}
```

---

## STRUCT – AKSES STRUKTUR MEMBER

- Akses pada member memudahkan dalam inisialisasi nilai
- Prosedurnya adalah nama variable diikuti oleh operator dot ( . ) dan nama member
- Contoh:

---

```
struct hitung{  
    int x;  
    int y;  
}  
struct hitung jumlah;  
jumlah.x = 10;  
jumlah.y = 20;
```

# CONTOH

```
struct hitung {  
    int x,y,jml;  
};  
  
int main(){  
    struct hitung jumlah;  
    jumlah.x=10;  
    jumlah.y=20;  
    jumlah.jml=jumlah.x + jumlah.y;  
    cout <<“hasil ” << jumlah.x << “ + ” << jumlah.y << “adalah ” << jumlah.jml <<  
    endl;  
}
```

# LATIHAN 1

- Buat structure sepeda dengan member:
  - Merk = “Polygon”
  - Type = “Sepeda Gunung”
  - Tahun = 2013
  - Harga = “5.000.000”
- Tampilkan hasilnya

## LATIHAN 2

- Buatlah program untuk menghitung rumus berikut menggunakan structure dengan member
  - Luas persegi panjang = panjang \* lebar
  - Luas lingkaran =  $22/7 * r * r$
  - Volume kerucut =  $1/3 * \pi * r^2 * t$
  - Volume bola =  $4/3 * \pi * r^3$

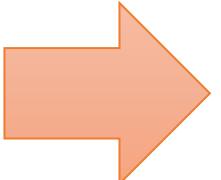
# TYPEDEF

- **Typedef** adalah perintah atau keyword yang digunakan untuk memberikan nama lain atau alias dari tipe data
- Dalam deklarasi variabel dari sebuah structure, harus menyertakan keyword struct
- Typedef merupakan alternative dalam deklarasi variabel tanpa harus menyertakan struct

---

```
struct hitung{  
    int x;  
    int y;  
}  
struct hitung jumlah;  
struct hitung kali;
```

---



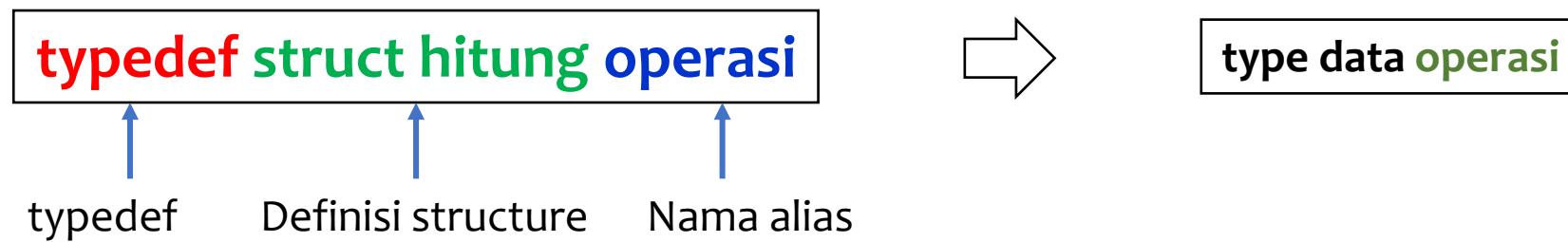
---

```
struct hitung{  
    int x;  
    int y;  
}  
typedef struct hitung  
operasi;  
operasi jumlah;  
operasi kali;
```

---

# PROSEDUR PENGGUNAAN TYPEDEF

- Letakkan typedef pada awal definisi structure
- Pada akhir definisi structure, letakkan nama alias untuk melengkapi pendefinisian menggunakan typedef



- Deklarasi variabel menggunakan typedef, memungkinkan deklarasi sama seperti deklarasi menggunakan type data int, char, double

## LATIHAN 3

- Buatlah program untuk menghitung rumus berikut menggunakan structure dengan member
  - Luas persegi panjang = panjang \* lebar
  - Luas lingkaran =  $22/7 * r * r$
  - Volume kerucut =  $1/3 * \pi * r^2 * t$
  - Volume bola =  $4/3 * \pi * r^3$
- Gunakan **typedef** dalam deklarasi variabel

# ARRAY OF STRUCTURE

- Proses pembuatan dan kerja **array of structure** prinsipnya sama dengan kerja array yang didalamnya terdapat tipe data seperti integer, character, float
- Proses pembuatan array of structure menentukan jumlah elemen array yang diletakkan setelah definisi structure.
- Contoh:

---

```
typedef struct pekerja{  
    int id;  
    char nama[50];  
}emp;  
emp data[2];
```

---

# CONTOH

```
typedef struct pekerja {
    int id;
    char nama[50];
}emp;

int main(){
    emp data[2];
    strcpy(data[0].nama, "Maxidena");
    data[0].id = 0001;
    strcpy(data[1].nama, "Rhyne");
    data[1].id = 0001;

    int x;
    for(x=0;x<3;x++){
        cout << data[x].id, data[x].nama << endl;
    }
}
```

# LATIHAN 4

- Cetak data berikut menggunakan array of structure

NIM	Nama	Jurusan	Tahun Lulus
A11.2020.01234	Andi	Broadcasting	2023
A11.2010.01234	Budi	Sistem Informasi	2013
A11.2000.01234	Ali	DKV	2003
A11.1990.01234	Siti	Kesehatan	1993

- Lakukan terhadap array statis dan array dinamis
- Analisa jawabanmu

# ADT Mahasiswa

```
#include <iostream>

using namespace std;

struct Mahasiswa{
    string nim;
    string nama;
    float ipk;
};

typedef struct Mahasiswa mhs;
```

```
int main()
{
    mhs Dilan;
    Dilan.nim = "A11.123"; Dilan.nama = "Dilan Ramadhan"; Dilan.ipk = 3.5;

    mhs dasPro[40];
    dasPro[0].nim = "A11.124"; dasPro[0].nama = "Milea"; dasPro[0].ipk = 2;

    int i, n;
    cout << "Berapa mahasiswa yang ingin diinput? "; cin >> n;
    for(i=0; i<n; i++){ //INPUT
        cout << "Mahasiswa ke-" << i+1 << endl;
        cout << "Masukkan NIM = "; cin >> dasPro[i].nim;
        cout << "Masukkan Nama = "; cin >> dasPro[i].nama;
        cout << "Masukkan IPK = "; cin >> dasPro[i].ipk;
    }
    cout << endl;
    for(i=0; i<n; i++){ //OUTPUT
        cout << "Mahasiswa ke-" << i+1 << endl;
        cout << "NIM = " << dasPro[i].nim << endl;
        cout << "Nama = " << dasPro[i].nama << endl;
        cout << "IPK = " << dasPro[i].ipk << endl;
    }
    return 0;
}
```

## POINTERS OF STRUCTURES - OBJECTIVE

Pointer

- Pass by Value
- Pass by Reference

Pointer of Structure

## Pass By Value

- Membutuhkan cadangan ruang memory ekstra untuk membuat copy dari variabel
- Copy variabel akan dikirimkan ke fungsi sebagai argument
- **Passing by value** tidak mampu digunakan untuk melakukan modifikasi konten original dari parameter yang masuk

# Contoh

```
int PassByValue(int);

int main(){
    int x=0;
    cout << "Input number: "; cin >> x;
    PassByValue(x);
    cout << "Result: " << x << endl;
    return 0;
}

int PassByValue(int x){
    x = x + 5;
    cout << "the value is : " << x << endl;
}
```

# Contoh Program

```
1 #include <iostream>
2
3 using namespace std;
4 int PassByValue(int);
5
6 int main()
7 {
8     int x;
9     cout << "masukan nilai x: "; cin >> x;
10    PassByValue(x);
11    cout << "nilai x di main(): " << x << endl;
12
13    return 0;
14 }
15
16 int PassByValue(int x) {
17     x = x + 5;
18     cout << "nilai x di fungsi Pass By Value(): " << x << endl;
19 }
```

```
masukan nilai x: 10
nilai x di fungsi Pass By Value(): 15
nilai x di main(): 10

Process returned 0 (0x0) execution time : 10.073 s
Press any key to continue.
```

# Pass By Reference

- Melewati alamat memory dari variabel dengan menempatkan **operator unary (&)**
- Untuk memodifikasi konten original sebuah argument, harus menggunakan operator indirection (\*) dimana akses konten dari sebuah lokasi memory terkandung variabel pointer

# Contoh

```
int PassByReference(int *);  
  
int main(){  
    int x=0;  
    cout << "Input number: "; cin >> x;  
    PassByReference(&x);  
    cout << "Result: " << x << endl;  
    return 0;  
}  
  
int PassByReference(int *nilai){  
    *nilai = *nilai + 5;  
    cout << "the value is : " << *nilai << endl;  
}
```

# Contoh Program

```
1 #include <iostream>
2
3 using namespace std;
4 int PassByValue(int);
5 int PassByReference(int *);
6
7 int main()
8 {
9     int x;
10    cout << "masukan nilai x: "; cin >> x;
11    PassByValue(x);
12    cout << "nilai x di main(): " << x << endl;
13    cout << "\n";
14    PassByReference(&x);
15    cout << "nilai x di main(): " << x << endl;
16
17    return 0;
18 }
19
20 int PassByValue(int x) {
21     x = x + 5;
22     cout << "nilai x di fungsi Pass By Value(): " << x << endl;
23 }
24
25 int PassByReference(int *x) {
26     *x = *x + 5;
27     cout << "nilai x di fungsi Pass By Reference(): " << *x << endl;
28 }
```

```
masukan nilai x: 10
nilai x di fungsi Pass By Value(): 15
nilai x di main(): 10

nilai x di fungsi Pass By Reference(): 15
nilai x di main(): 15

Process returned 0 (0x0)  execution time : 2.583 s
Press any key to continue.
```

# Pointers of Structures

- Sama seperti materi pointer, penggunaan pointer pada structure digunakan untuk mengakses alamat yang ada pada structure

---

```
hitung jumlah =  
{10,20,30};  
pdata = &jumlah;
```

---

- Pointer **pdata** menyimpan alamat dari **jumlah**

# Contoh – Pointers of Structures

```
typedef struct pekerja {
    int id;
    char nama[50];
}emp;

int main(){
    emp data[2];
    emp data = {0,0};
    emp *pdata;
    pdata = &data;

    strcpy(pdata->nama, "Maxidena");
    pdata->id = 0001;

    cout << pdata->id, pdata->nama << endl;
}
```

# Referensi

## Utama:

1. Bjarne Stroustrup, 2014, Programming: Principles and Practice Using C++ (Second Edition), Addison-Wesley Professional

## Pendukung:

1. Introduction to Computer Science and Programming in Python, MIT  
<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016>
2. Introduction to Computer Science and Programming, MIT  
<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-00sc-introduction-to-computer-science-and-programming-spring-2011/index.htm>



# TERIMA KASIH

ANY QUESTIONS?