

**LAPORAN PRAKTIKUM
ALGORITMA & STRUKTUR DATA
MODUL 1**



STRUCT & POINTER

Oleh:

Indra Suryadilaga NIM. 2410817310014

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
APRIL 2025**

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM ALGORITMA & STRUKTUR DATA
MODUL 1

Laporan Praktikum Algoritma & Struktur Data Modul 1: Struct & Pointer ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Web I. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Indra Suryadilaga
NIM : 2410817310014

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Zulfa Auliya Akbar
NIM. 2210817210010

Andreyan Rizky Baskara, S.Kom.,
M.Kom
NIP. 199307032019031011

DAFTAR ISI

LEMBAR PENGESAHAN	2
DAFTAR ISI.....	3
DAFTAR GAMBAR	4
DAFTAR TABEL	5
SOAL 1	6
Pembahasan.....	7
Algoritma	9
SOAL 2	11
Source code	11
Pembahasan.....	12
Algoritma	12
SOAL 3	14
Source code	14
Pembahasan.....	14
Algoritma	15

DAFTAR GAMBAR

No table of contents entries found.

DAFTAR TABEL

Table 1 Pembahasan Source Code 1.1	7
Table 2 Pembahasan Source Code 1.2	7
Table 3 Pembahasan Source Code 1.3	7
Table 4 Pembahasan Source Code 1.4	8
Table 5 Pembahasan Source Code 1.5	8
Table 6 Pembahasan Source Code 1.6	9
Table 7 Source Code Soal 2	11
Table 8 Pembahasan Source Code 2.1	12
Table 9 Pembahasan Source Code 2.2	12
Table 10 Pembahasan Source Code 2.3	12
Table 11 Source Code Soal 3	14
Table 12 Pembahasan Source Code 3.1	15
Table 13 Pembahasan Source Code 3.2	15
Table 14 Pembahasan Source Code 3.3	15

SOAL 1

Cobalah program berikut, running dan analisis hasilnya. Buatlah algoritma untuk program tersebut.

```
#include <iostream>

using namespace std;

struct mhs
{
    char nama[20], nim[10], jurusan[2];
    int sks, program;
};

struct mhs bayar[2];

main() {
    int bts, var, tetap;
    for(int i=0; i<2; i++)
    {
        //Input data
        cout<<"\n\n-----\n";
        cout<<"\nNama mhs      = "; cin>>bayar[i].nama;
        cout<<"NIM          = "; cin>>bayar[i].nim;
        cout<<"Jurusan [TI, PTK] = "; cin>>bayar[i].jurusan;
        input:
        cout<<"Program [1=D3, 2=S1] = ";
        cin>>bayar[i].program;

        if(bayar[i].program<0 || bayar[i].program>2)
        {
            cout<<"Program tidak sesuai\n";
            goto input;
        } cout<<"Jumlah sks      = "; cin>>bayar[i].sks;

        if(bayar[i].program==1)
        {
            tetap=500000;
            var=bayar[i].sks*25000;

            }else if(bayar[i].program==2)
            {
                tetap=750000;
                var=bayar[i].sks*50000;
            }cout<<endl;

        //Output data
        cout<<"\n\n-----\n";
        cout<<" Output ";
        cout<<"\n-----\n";
        cout<<"\nNama mhs      = "<<bayar[i].nama;
        cout<<"\nNIM          = "<<bayar[i].nim;
        cout<<"\nJurusan      = "<<bayar[i].jurusan;
        cout<<"\nProgram      = "<<bayar[i].program;
        cout<<"\nJumlah sks    = "<<bayar[i].sks;
        cout<<"\nSPP tetap     = "<<tetap;
        cout<<"\nSPP variabel  = "<<var;
        cout<<endl<<endl;
    }
}
```

Pembahasan

Program pada gambar tersebut digunakan untuk menghitung biaya SPP berdasarkan program studi D3 atau S1 dan jumlah SKS yang diambil.

```
struct mhs {  
    char nama[20], nim[10], jurusan[2];  
    int sks, program;  
};  
  
struct mhs bayar[2];
```

Table 1 Pembahasan Source Code 1.1

Struct pertama digunakan untuk menyimpan data-data dari mahasiswa seperti nama, nim, jurusan, sks, serta program. Sedangkan struct yang kedua untuk menyimpan array berisi 2 objek mhs. Artinya terdapat 2 mahasiswa dengan struktur data yang sama.

- nama[20]: Menyimpan nama mahasiswa (maksimal 20 karakter).
- nim[10]: Menyimpan NIM mahasiswa (maksimal 10 karakter).
- jurusan[2]: Menyimpan jurusan (maksimal 2 karakter).
- sks: Menyimpan jumlah SKS yang diambil.
- program: Menyimpan jenis program (1 untuk D3, 2 untuk S1).

```
for(int i=0; i<2; i++)  
{  
    cout<<"\nNama mhs = "; cin>>bayar[i].nama;  
    cout<<"\nNIM = "; cin>>bayar[i].nim;  
    cout<<"\nJurusan [TI, PTK] = ";  
    cin>>bayar[i].jurusan;  
    input:  
    cout<<"\nProgram [1=D3, 2=S1] = ";  
    cin>>bayar[i].program;
```

Table 2 Pembahasan Source Code 1.2

Loop ini berjalan 2 iterasi yang digunakan untuk menginput data 2 mahasiswa.

```
if (bayar[i].program != 1 && bayar[i].program != 2)  
{  
    cout << "Program tidak sesuai\n";  
    goto input; // Lompat kembali ke label 'input'  
}  
}
```

Table 3 Pembahasan Source Code 1.3

Pengkondisian yang digunakan untuk mengecek jika input dari Program bukan 1 atau 2, maka program tidak akan melanjutkan ke langkah berikutnya dan justru mengulang input Program.

```
cout<<"\nJumlah sks = "; cin>>bayar[i].sks;

if (bayar[i].program==1)
{
    tetap=500000;
    var=bayar[i].sks*25000;
}
else if (bayar[i].program==2)
{
    tetap=750000;
    var=bayar[i].sks*50000;
}
```

Table 4 Pembahasan Source Code 1.4

Pengkondisian untuk menghitung jumlah SPP dan biaya tetap sesuai dengan Program dan jumlah sks.

- Jika program D3 (1): Biaya tetap 500.000, biaya per SKS 25.000.
- Jika program S1 (2): Biaya tetap 750.000, biaya per SKS 50.000.
- Total SPP = Biaya tetap + (Jumlah SKS * biaya per SKS).

```
cout<<"\nOutput ";
cout<<"\nNama mhs = "<<bayar[i].nama;
cout<<"\nNIM = "<<bayar[i].nim;
cout<<"\nJurusan = "<<bayar[i].jurusan;
cout<<"\nProgram = "<<bayar[i].program;
cout<<"\nJumlah sks = "<<bayar[i].sks;
cout<<"\nSPP tetap = "<<tetap;
cout<<"\nSPP variable = "<<var;
cout<<endl;
```

Table 5 Pembahasan Source Code 1.5

Output untuk menampilkan nama, nim, program, jumlah sks, biaya tetep, dan biaya variable.

Adapun kesalahan yang diperbaiki adalah:

- char jurusan[2]: jika pengguna memasukan “TI” atau “PTK”, harusnya alokasi karakter lebih dari 2. Solusinya mengubah batasnya menjadi 3 karakter.

- goto input: membuat source code sulit dibaca dan dimaintain. Solusinya bisa menggunakan while loop.

```
do {
    cout<<"\nProgram [1=D3, 2=S1] = ";
    cin>>bayar[i].program;
    if (bayar[i].program != 1 && bayar[i].program !=
2)
        cout << "Program tidak sesuai, silakan
ulangi.\n";
} while (bayar[i].program != 1 && bayar[i].program !=
2);
```

Table 6 Pembahasan Source Code 1.6

Algoritma

1. Mulai Program

2. Deklarasi Variabel

- Buat struktur data mhs untuk menyimpan informasi mahasiswa:
 - nama (string)
 - nim (string)
 - jurusan (string)
 - sks (integer)
 - program (integer)
- Buat array bayar[2] untuk menyimpan data 2 mahasiswa.
- Deklarasikan variabel bts, var, tetap untuk perhitungan SPP.

3. Input Data Mahasiswa (Loop untuk 2 mahasiswa)

Untuk setiap mahasiswa i (0 hingga 1):

- Minta input nama mahasiswa.
- Minta input NIM mahasiswa.
- Minta input jurusan mahasiswa (TI atau PTK).

4. Validasi Program (D3 atau S1)

- Minta input program studi (1 = D3, 2 = S1).
- Jika input bukan 1 atau 2, tampilkan pesan "Program tidak sesuai".
- Kembalikan ke langkah input program.

5. Input Jumlah SKS

- Minta input jumlah SKS yang diambil mahasiswa.

6. Hitung SPP Mahasiswa

- Jika program = 1 (D3)
 - Biaya tetap = 500.000
 - Biaya SKS = 25.000 per SKS.
- Jika program = 2 (S1)
 - Biaya tetap = 750.000
 - Biaya SKS = 50.000 per SKS.

- Hitung SPP total = biaya tetap + (jumlah SKS * biaya per SKS).
7. Output Data Mahasiswa
- Tampilkan Nama, NIM, Jurusan, Program, Jumlah SKS, Biaya Tetap, dan Total SPP.
8. Selesai

SOAL 2

Buatlah program dengan menggunakan struct dengan hasil eksekusi program sebagai berikut:

- a. Plat Nomor Kendaraan : DA1234MK
- b. Jenis Kendaraan : RUSH
- c. Nama Pemilik : Andika Hartanto
- d. Alamat : Jl. Kayu Tangi 1
- e. Kota : Banjarmasin.

Source code

```
#include <iostream>
using namespace std;

// Deklarasi struct Kendaraan
struct Kendaraan {
    string platNomor;
    string jenisKendaraan;
    string namaPemilik;
    string alamat;
    string kota;
};

int main() {
    // Inisialisasi data kendaraan menggunakan struct
    Kendaraan mobil = {"DA1234MK", "RUSH", "Andika
Hartanto", "Jl. Kayu Tangi 1", "Banjarmasin"};

    // Menampilkan hasil eksekusi
    cout << "Plat Nomor Kendaraan : " <<
mobil.platNomor << endl;
    cout << "Jenis Kendaraan : " <<
mobil.jenisKendaraan << endl;
    cout << "Nama Pemili : " << mobil.namaPemilik <<
endl;
    cout << "Alamat : " << mobil.alamat << endl;
    cout << "Kota : " << mobil.kota << endl;

    return 0;
}
```

Table 7 Source Code Soal 2

Pembahasan

Program ini menggunakan struct untuk mengelompokkan beberapa atribut yang berkaitan dalam satu entitas yaitu kendaraan.

```
struct Kendaraan {  
    string platNomor;  
    string jenisKendaraan;  
    string namaPemilik;  
    string alamat;  
    string kota;  
};
```

Table 8 Pembahasan Source Code 2.1

Struct ini digunakan untuk menyimpan informasi kendaraan seperti plat nomor, jenis kendaraan, nama pemilik, alamat, dan kota. Yang mana semua tipe data variabelnya adalah string.

```
Kendaraan mobil = {"DA1234MK", "RUSH", "Andika  
Hartanto", "Jl. Kayu Tangi 1", "Banjarmasin"};
```

Table 9 Pembahasan Source Code 2.2

Variabel mobil dideklarasikan dengan tipe data kendaraan dengan menginisialisasikan dengan data kendaraan yang ditampilkan.

```
cout << "Plat Nomor Kendaraan : " << mobil.platNomor <<  
endl;  
cout << "Jenis Kendaraan : " << mobil.jenisKendaraan <<  
endl;  
cout << "Nama Pemilik : " << mobil.namaPemilik << endl;  
cout << "Alamat : " << mobil.alamat << endl;  
cout << "Kota : " << mobil.kota << endl;
```

Table 10 Pembahasan Source Code 2.3

Output untuk menampilkan data dari mobil yang sudah diinisialisasikan.

Algoritma

Algoritma Program Struct Kendaraan:

1. Mulai.
2. Deklarasi struct Kendaraan dengan atribut:
 - a. platNomor (string).
 - b. jenisKendaraan (string).
 - c. namaPemilik (string).
 - d. alamat (string).
 - e. kota (string).
3. Inisialisasi variabel mobil dari struct Kendaraan dengan data:
 - a. Plat Nomor Kendaraan = "DA1234MK".

- b. Jenis Kendaraan = "RUSH".
 - c. Nama Pemilik = "Andika Hartanto".
 - d. Alamat = "Jl. Kayu Tangi 1".
 - e. Kota = "Banjarmasin".
4. Cetak hasil ke layar:
- a. "Plat Nomor Kendaraan :" diikuti dengan mobil.platNomor.
 - b. "Jenis Kendaraan :" diikuti dengan mobil.jenisKendaraan.
 - c. "Nama Pemilik :" diikuti dengan mobil.namaPemilik.
 - d. "Alamat :" diikuti dengan mobil.alamat.
 - e. "Kota :" diikuti dengan mobil.kota.
5. Selesai.

SOAL 3

Buatlah program dengan tampilan sebagai berikut:

- Masukkan sebuah huruf =
- Masukan sebuah kata =
- Masukkan Angka =
- Huruf yang Anda masukkan adalah
- Kata yang Anda masukkan adalah
- Angka yang Anda masukkan adalah

Source code

```
#include <iostream>
using namespace std;

int main() {
    char huruf;
    string kata;
    int angka;

    // Input data
    cout << "Masukkan sebuah huruf = ";
    cin >> huruf;
    cin.ignore(); // Mengatasi masalah input buffer

    cout << "Masukkan sebuah kata = ";
    getline(cin, kata);

    cout << "Masukkan sebuah angka = ";
    cin >> angka;

    // Output hasil
    cout << "\nHuruf yang Anda masukkan adalah " << huruf
    << endl;
    cout << "Kata yang Anda masukkan adalah " << kata <<
    endl;
    cout << "Angka yang Anda masukkan adalah " << angka
    << endl;

    return 0;
}
```

Table 11 Source Code Soal 3

Pembahasan

```
char huruf;
string kata;
```

```
int angka;
```

Table 12 Pembahasan Source Code 3.1

- huruf : Variabel bertipe char, digunakan untuk menyimpan satu karakter.
- Kata: Variabel bertipe string, digunakan untuk menyimpan satu kata atau kalimat.
- Angka: Variabel bertipe int, digunakan untuk menyimpan bilangan bulat yang dimasukkan pengguna.

```
cout << "Masukkan sebuah huruf = ";  
cin >> huruf;  
cin.ignore(); // Mengatasi masalah input buffer  
  
cout << "Masukkan sebuah kata = ";  
getline(cin, kata);  
  
cout << "Masukkan sebuah angka = ";  
cin >> angka;
```

Table 13 Pembahasan Source Code 3.2

Input untuk pengguna memasukkan sebuah huruf, sebuah kata, dan sebuah angka. `cin.ignore()` digunakan untuk membersihkan buffer input agar `getline(cin, kata)` bisa berfungsi dengan baik setelah `cin >> huruf`. Sedangkan `getline(cin, kata)` digunakan untuk input bisa berupa teks yang mengandung spasi.

```
cout << "\nHuruf yang Anda masukkan adalah " << huruf <<  
endl;  
cout << "Kata yang Anda masukkan adalah " << kata <<  
endl;  
cout << "Angka yang Anda masukkan adalah " << angka <<  
endl;
```

Table 14 Pembahasan Source Code 3.3

Output untuk menampilkan kembali huruf, kata, dan angka yang telah dimasukan.

Algoritma

1. Mulai.
2. Deklarasi struct `DataInput` dengan atribut:
 - a. huruf (char).
 - b. kata (string).
 - c. angka (int).
3. Inisialisasi variabel data dari struct `DataInput`.
4. Membaca input dari pengguna:
 - a. Tampilkan "Masukkan sebuah huruf = ".
 - b. Simpan input ke dalam `data.huruf`.
 - c. Kosongkan buffer dengan `cin.ignore()`.

- d. Tampilkan "Masukkan sebuah kata = ".
 - e. Simpan input ke dalam data.kata menggunakan `getline()`.
 - f. Tampilkan "Masukkan sebuah angka = ".
 - g. Simpan input ke dalam data.angka.
5. Cetak hasil ke layar:
- a. Tampilkan "Huruf yang Anda masukkan adalah " diikuti dengan `data.huruf`.
 - b. Tampilkan "Kata yang Anda masukkan adalah " diikuti dengan `data.kata`.
 - c. Tampilkan "Angka yang Anda masukkan adalah " diikuti dengan `data.angka`.
6. Selesai.