

## **Ketentuan Penulisan Laporan adalah sebagai berikut**

- a. Laporan Praktikum bersifat perorangan
  - b. Waktu pengumpulan laporan adalah satu minggu atau H-1 Praktikum maksimal pukul pukul 23.59.
  - c. Margin Halaman untuk Laporan Praktikum adalah Top: 4cm, Left: 4cm, Bottom: 3cm, Right: 3cm.
  - d. Laporan Praktikum mempunyai sistematika sebagai berikut:
    1. Halaman Judul/Cover
      - A. Dasar Teori
      - B. Guided (berisi screenshot source code & output program disertai penjelasannya file program .cpp dikumpulkan ke github sesuai jumlah guided tiap modul
      - C. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)  
file program .cpp dikumpulkan ke github sesuai jumlah unguided tiap
      - D. Kesimpulan
      - E. Referensi (jurnal 5 tahun terakhir)
- \*Screenshot wajib pakai watermark yang menerangkan NAMA dan NIM
- e. Laporan Praktikum yang tidak sesuai ketentuan dianggap tidak mengumpulkan.
  - f. Plagiasi = 0.

## **Ketentuan Penilaian**

- Telat 1 menit – 1jam = -5
- Telat 1 -7 hari = -20
- Telat 8- 14 hari = -40
- Telat 15-21 hari = -60
- Telat 22- 30 hari = -80
- Telat lebih dari 1 bulan = 0

g. Contoh Halaman Judul Laporan Praktikum

**LAPORAN PRAKTIKUM  
STRUKTUR DATA**

**MODUL I  
PENGENALAN CODE BLOCKS**



**Disusun Oleh :**

Muhammad Rasyid Ridho  
103122400018

**Dosen**

Diah Septiani, S.Kom., M.Cs.

**PROGRAM STUDI REKAYASA PERANGKAT LUNAK  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

## A. Dasar Teori

Struktur data merupakan cara untuk menyimpan, mengatur, dan mengelola data agar dapat diakses serta dimodifikasi secara efisien. Menurut [Rani et al., 2021], struktur data menjadi dasar dalam pengembangan perangkat lunak karena berpengaruh pada kompleksitas waktu dan ruang dalam algoritma.

Salah satu konsep dasar dalam struktur data adalah **pointer**, yaitu variabel yang menyimpan alamat memori dari variabel lain. Pointer memungkinkan pengaksesan dan manipulasi data secara langsung pada level memori, sehingga dapat meningkatkan efisiensi program (Kaur & Singh, 2022).

Selain itu, **array** adalah struktur data statis yang menyimpan elemen dengan tipe data sama dalam memori yang berurutan. Array banyak digunakan dalam pemrograman karena kemudahannya dalam melakukan pengaksesan data menggunakan indeks (Patel & Sharma, 2020).

Kemudian, terdapat konsep **Abstract Data Type (ADT)**, yaitu representasi matematis dari suatu struktur data beserta operasi-operasi yang dapat dilakukan terhadapnya. ADT berfungsi sebagai penghubung antara logika pemrograman dengan implementasi struktur data pada bahasa pemrograman tertentu (Wang et al., 2021).

## B. Guided (berisi screenshot source code & output program disertai penjelasannya)

### Guided 1

```
#include <iostream>
#include <conio.h>

using namespace std;

int main()
{
    int x, y;
    int *px;
    x = 87;
    px = &x;
    y = *px;

    cout << "Alamat x = " << &x << endl;
    cout << "Isi px = " << &px << endl;
    cout << "Isi x = " << x << endl;
    cout << "Nilai px = " << *px << endl;
    cout << "Nilai y = " << y << endl;

    getch();
    return 0;
}
```

## Screenshots Output

```
rasyi@LAPTOP-353DN77J MINGW64 /c/Project/CPP (main)
$ /usr/bin/env c:\Users\rasyi\.vscode\extensions\m
ft-MIEngine-In-4usgeswn.eg3 --stdout=Microsoft-MIEngine-
uqjps.1tw --dbgExe=C:\msys64\ucrt64\bin\gdb.exe --in
Alamat x = 0x5ffec0
Isi px = 0x5ffec0
Isi x = 87
Nilai *px = 87
Nilai y = 87

103122400018 - Muhammad Rasyid Ridho
```

## Deskripsi:

kode itu intinya nunjukin gimana sebuah pointer bernama px bisa nyimpen alamat memori dari variabel x. Setelah px tau alamatnya si x, variabel y kemudian ngambil nilainya bukan dari x langsung, tapi dari alamat yang disimpan sama px itu. Proses ini pake tanda \* (bintang), makanya hasil akhirnya nunjukin kalo nilai si x, si y, dan nilai yang diliat lewat \*px itu semuanya jadi sama, yaitu 87.

## Guided 2

```
#include <iostream>
#include <conio.h>
#define MAX 5

using namespace std;

int main()
{
    int i, j;
    float nilai[MAX];
    static int nilai_tahun[MAX][MAX] = {
        {0, 2, 2, 0, 0},
        {0, 1, 1, 1, 0},
        {0, 3, 3, 3, 0},
        {4, 4, 0, 0, 4},
        {5, 0, 0, 0, 5},
    };

    for (i = 0; i < MAX; i++)
    {
        cout << "Masukkan nilai ke-" << i + 1 << " : ";
        cin >> nilai[i];
    }

    cout << "\nData nilai siswa:\n";
    for (i = 0; i < MAX; i++)
```

```

    {
        cout << "Nilai ke-" << i + 1 << " = " << nilai[i] << endl;
    }

    cout << "\nNilai tahunan:\n";
    for (i = 0; i < MAX; i++)
    {
        for (j = 0; j < MAX; j++)
        {
            cout << nilai_tahun[i][j] << " ";
        }
        cout << endl;
    }

    getch(); // Menahan Layar
    return 0;
}

```

## Screenshots Output

```

rasyi@LAPTOP-353DN77J MINGW64 /c/Project/CPP (main)
$ /usr/bin/env c:\Users\rasyi\.vscode\extensions\
ft-MIEngine-In-o50vuiyh.i2p --stdout=Microsoft-MIEngi
dcofk.soxx --dbgExe=C:\msys64\ucrt64\bin\gdb.exe -
Masukkan nilai ke-1 : 90

Masukkan nilai ke-2 : 80

Masukkan nilai ke-3 : 70

Masukkan nilai ke-4 : 100

Masukkan nilai ke-5 : 50

Data nilai siswa :
Nilai ke-1 = 90
Nilai ke-2 = 80
Nilai ke-3 = 70
Nilai ke-4 = 100
Nilai ke-5 = 50

Nilai tahunan :
0 2 2 0 0
0 1 1 1 0
0 3 3 3 0
4 4 0 0 4
5 0 0 0 5

103122400018 - Muhammad Rasyid Ridho

```

## Deskripsi:

kode itu intinya nunjukin gimana sebuah pointer bernama px bisa nyimpen alamat memori dari variabel x. Setelah px tau alamatnya si x, variabel y kemudian ngambil

nilainya bukan dari x langsung, tapi dari alamat yang disimpan sama px itu. Proses ini pake tanda \* (bintang), makanya hasil akhirnya nunjukin kalo nilai si x, si y, dan nilai yang dilihat lewat \*px itu semuanya jadi sama, yaitu 87.

E. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

No. 1

```
struct nilaiSTD {  
    double CLO1;  
    double CLO2;  
    double CLO3;  
    double CLO4;  
    double finalScore;  
    string indexScore;  
};
```

No. 2

```
nilaiSTD data1;
```

No. 3

```
cout << "Input Score CLO1: ";  
cin >> clo1;  
cout << "Input Score CLO2: ";  
cin >> clo2;  
cout << "Input Score CLO3: ";  
cin >> clo3;
```

```
cout << "Input Score CLO4: ";  
cin >> clo4;
```

No. 4

```
nilaiSTD editDataCLO(double clo1, double clo2, double clo3, double clo4) {  
    }  
}
```

No. 5

```
nilaiSTD data;  
data.CLO1 = clo1;  
data.CLO2 = clo2;  
data.CLO3 = clo3;  
data.CLO4 = clo4;  
return data;
```

No. 6

```
double countFinalScore(double clo1, double clo2, double clo3, double clo4) {  
    double finalScore = (0.3 * clo1) + (0.3 * clo2) + (0.2 * clo3) + (0.2 * clo4);  
    return finalScore;  
}
```

No. 7

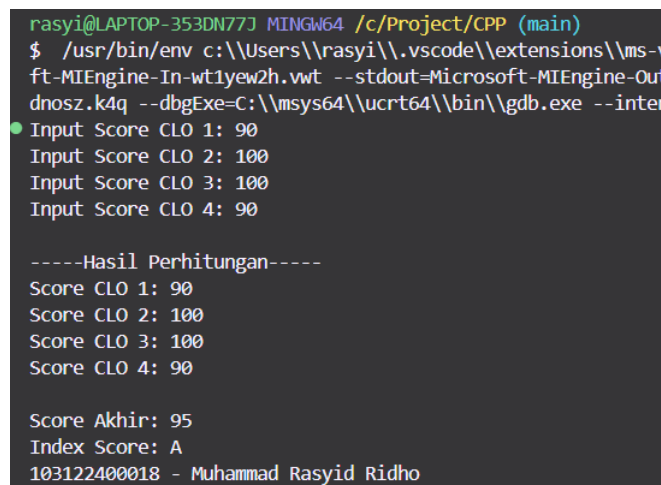
```
string idxScore(double finalScore) {  
    if (finalScore > 80) {  
        return "A";  
    } else if (finalScore > 70 && finalScore <= 80) {  
        return "AB";  
    } else if (finalScore > 65 && finalScore <= 70) {
```

```

        return "B";
    } else if (finalScore > 60 && finalScore <= 65) {
        return "BC";
    } else if (finalScore > 50 && finalScore <= 60) {
        return "C";
    } else if (finalScore > 40 && finalScore <= 50) {
        return "D";
    } else {
        return "E";
    }
}
}

```

### Screenshots Output



```

rasyi@LAPTOP-353DN77J MINGW64 /c/Project/CPP (main)
$ /usr/bin/env c:\Users\rasyi\.vscode\extensions\ms-vscode.cpptools\bin\code-cli.exe --stdout=Microsoft-MIEngine-Output.txt --dbgExe=C:\msys64\ucrt64\bin\gdb.exe --interpreter=python3
Input Score CLO 1: 90
Input Score CLO 2: 100
Input Score CLO 3: 100
Input Score CLO 4: 90

-----Hasil Perhitungan-----
Score CLO 1: 90
Score CLO 2: 100
Score CLO 3: 100
Score CLO 4: 90

Score Akhir: 95
Index Score: A
103122400018 - Muhammad Rasyid Ridho

```

### Deskripsi:

Kode program tersebut dimulai dari mendefinisikan tipe data variable untuk menyimpan data nilai, membuat inputan data nilai, kemudian membuat function untuk mengolah data menjadi nilai akhir dan indeks nilainya.



## F. Kesimpulan

Kesimpulannya adalah menunjukkan konsep fundamental dalam Algoritma Pemrograman, Tipe Data Abstrak dan sedikit pengenalan pointer pada bagian guided.

## G. Referensi

Kaur, P., & Singh, R. (2022). Role of pointers in memory management and data structures. *International Journal of Computer Applications*, 184(17), 12–18.

Patel, V., & Sharma, K. (2020). Comparative study of array and linked list implementation in data structures. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 6(2), 45–49.

Rani, S., Gupta, A., & Mehra, P. (2021). Data structures and algorithms: A review on efficiency and performance. *Journal of Information and Computational Science*, 11(9), 89–96.

Wang, Y., Li, H., & Zhang, J. (2021). Teaching abstract data types in computer science education. *Education and Information Technologies*, 26, 6059–6074