

**LAPORAN PRAKTIKUM  
STRUKTUR DATA**

**MODUL I  
REVIEW ALGORITMA TOPIK  
TIPE BENTUKAN, FUNCTION,  
PERCABANGAN**



**Disusun Oleh :**

NAMA : Muhammad Restu Aditya

NIM : 103122400022

**Dosen**

DIAH SEPTIANI

**PROGRAM STUDI S1 REKAYASA PERANGKAT  
LUNAK  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

## A. Dasar Teori

Bahasa pemrograman C++ merupakan pengembangan dari bahasa C yang diperkenalkan oleh Bjarne Stroustrup pada awal tahun 1980-an. C++ mendukung pemrograman prosedural sekaligus berorientasi objek, sehingga banyak digunakan untuk membangun aplikasi yang membutuhkan efisiensi tinggi namun tetap terstruktur. Konsep dasar yang menjadi fondasi dalam C++ meliputi tipe data, variabel, operator, struktur kontrol, fungsi, hingga penggunaan pointer untuk mengatur memori secara langsung. Oleh karena itu, hingga kini C++ tetap menjadi salah satu bahasa pemrograman penting dalam bidang akademik maupun industri, terutama di sistem yang menuntut performa tinggi[1].

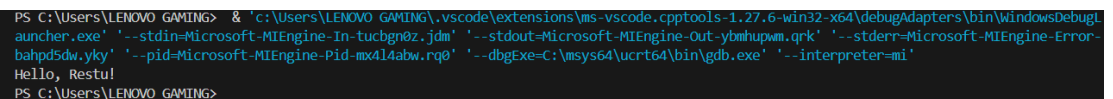
## B. Guided (berisi screenshot source code & output program disertai penjelasannya)

### Guided 1

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

int main() {
    cout << "Hello, Restu!" << endl;
    return 0;
}
```

### Screenshots Output



```
PS C:\Users\LENOVO GAMING> g++ -std=Microsoft-MIEngine-In-tucbgn0z.jdm -std=Microsoft-MIEngine-Out-ybmhupwm.qrk -std=Microsoft-MIEngine-Error-
bahpd5dw.yky -pid=Microsoft-MIEngine-Pid-mx414abw.rq0 -dbgExe=C:\msys64\ucrt64\bin\gdb.exe --interpreter=mi
Hello, Restu!
PS C:\Users\LENOVO GAMING>
```

### Deskripsi:

Kode di atas adalah program dasar C++ untuk menampilkan teks ke layar. `#include <iostream>` memanggil pustaka input-output, `cout` digunakan untuk mencetak teks, dan `main()` adalah fungsi utama tempat eksekusi program dimulai, sehingga ketika code di run muncul teks “Hello, Restu”[2].

## Guided 2

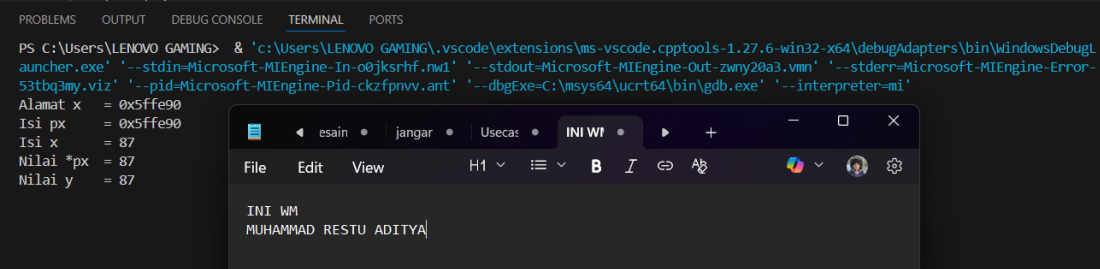
```
#include <iostream>
#include <conio.h>
using namespace std;

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



```
PS C:\Users\LENOVO GAMING> & 'c:\Users\LENOVO GAMING\.vscode\extensions\ms-vscode.cpptools-1.27.6-win32-x64\debugAdapters\bin\WindowsDebugL
auncher.exe' '--stdin=Microsoft-MIEngine-In-00jksrhf.nw1' '--stdout=Microsoft-MIEngine-Out-zwny20a3.vmn' '--stderr=Microsoft-MIEngine-Error-
53tbq3my.viz' '--pid=Microsoft-MIEngine-Pid-ckzfprnv.ant' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Alamat x  = 0x5ffe90
Isi px    = 0x5ffe90
Isi x     = 87
Nilai *px = 87
Nilai y   = 87
```

## Deskripsi:

Program di atas menunjukkan penggunaan **pointer** dalam C++. Variabel `x` menyimpan nilai 87, kemudian alamat memori `x` disimpan pada pointer `px` dengan `px = &x;`. Selanjutnya nilai yang ditunjuk pointer (`*px`) disalin ke

variabel y. Output menampilkan alamat variabel x, isi dari pointer, nilai x, nilai yang ditunjuk pointer, serta nilai y. Hal ini membuktikan bahwa pointer berfungsi untuk mengakses dan memanipulasi data melalui alamat memori[3].

### Guided 3

```
#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},
        {0,2,2,0,0},
        {0,2,2,0,0},
        {0,2,2,0,0},
        {0,2,2,0,0}
    };

    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();
    return 0;
}

```

## Screenshots Output

The screenshot shows a Windows command prompt window with the following output:

```

PS C:\Users\LENOVO GAMING> & 'c:\Users\LENOVO GAMING\.vscode\extensions\ms-vscode.cpptools-1.27.6-win32-x64\debugAdapters\bin\WindowsDebugL
auncher.exe' '--stdin=Microsoft-MIEngine-In-tkyitg02.ybz' '--stdout=Microsoft-MIEngine-Out-pmrusfaf.s1a' '--stderr=Microsoft-MIEngine-Error-
dvtm4pt0.10u' '--pid=Microsoft-MIEngine-Pid-2hlbubvu.wde' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Masukkan nilai ke-1: 1
Masukkan nilai ke-2: 2
Masukkan nilai ke-3: 3
Masukkan nilai ke-4: 4
Masukkan nilai ke-5: 5

Data nilai siswa:
Nilai ke-1 = 1
Nilai ke-2 = 2
Nilai ke-3 = 3
Nilai ke-4 = 4
Nilai ke-5 = 5

Nilai tahunan:
0 2 2 0 0
0 2 2 0 0
0 2 2 0 0
0 2 2 0 0
0 2 2 0 0

```

Overlaid on the command prompt is a Notepad++ window titled 'INI WM' containing the following source code:

```

INI WM
MUHAMMAD RESTU ADITYA

```

The Notepad++ status bar at the bottom indicates: Ln 2, Col 22 | 28 characters | Plain text | 100% | Windows (CRLF) | UTF-8.

## Deskripsi:

Program ini menggunakan **array satu dimensi** untuk menyimpan data nilai yang dimasukkan oleh pengguna, serta **array dua dimensi** nilai\_tahun yang sudah diinisialisasi sebelumnya. Pada bagian awal, pengguna diminta menginput lima nilai yang kemudian disimpan dalam array. Setelah itu, program menampilkan kembali nilai yang sudah dimasukkan, lalu menampilkan isi array dua dimensi dalam bentuk matriks. Program ini menunjukkan bagaimana array dapat dipakai untuk menyimpan dan mengolah data dalam jumlah banyak secara terstruktur[4].

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

## Unguided 1

```

TYPE nilaiSTD = RECORD
    nilaiCLO1 : real
    nilaiCLO2 : real

```

```
nilaiCLO3 : real  
nilaiCLO4 : real  
nilaiAkhir : real  
indeks : string  
END RECORD
```

#### Deskripsi:

Struktur data bentukan (record/struct) digunakan untuk menyimpan kumpulan data dengan tipe berbeda dalam satu entitas. Menurut Nugroho (2019), penggunaan record dapat menyederhanakan pengelolaan data kompleks dalam pemrograman[1].

Tipe bentukan *nilaiSTD* menyimpan data nilai mahasiswa untuk mata kuliah Struktur Data. Ada empat nilai CLO, nilai akhir, dan indeks mutu berupa huruf. Struct/record memudahkan pengelompokan data yang berkaitan[5].

#### Unguided 2

```
VAR mahasiswa1, mahasiswa2 : nilaiSTD
```

#### Deskripsi:

Dua variabel *mahasiswa1* dan *mahasiswa2* di atas dibuat dengan tipe *nilaiSTD*. Setiap variabel menyimpan satu set data lengkap (CLO1–CLO4, nilai akhir, dan indeks), sehingga bisa digunakan untuk merepresentasikan nilai dari dua mahasiswa yang berbeda[6].

#### Unguided 3

```
Kamus:  
CLO1, CLO2, CLO3, CLO4 : real  
  
Algoritma:  
Print("Masukkan nilai CLO-1 : ")  
Input(CLO1)
```

```
Print("Masukkan nilai CLO-2 : ")
Input(CLO2)

Print("Masukkan nilai CLO-3 : ")
Input(CLO3)

Print("Masukkan nilai CLO-4 : ")
Input(CLO4)
```

Deskripsi:

Program ini mendeklarasikan empat variabel bertipe real untuk menampung nilai CLO1–CLO4. User diminta memasukkan nilai melalui perintah input [7].

Unguided 4

```
Function ubahData(CLO1, CLO2, CLO3, CLO4 : real) -> nilaiSTD
```

Deskripsi:

Function yang akan mengubah data nilai CLO1 s/d CLO4 menjadi data yang tersimpan dalam variable berupa bentukan nilaiSTD[8].

Unguided 5

```
Function ubahData(CLO1, CLO2, CLO3, CLO4 : real) → nilaiSTD
  VAR hasilakhir : nilaiSTD
  hasilakhir.CLO1 ← CLO1
  hasilakhir.CLO2 ← CLO2
  hasilakhir.CLO3 ← CLO3
  hasilakhir.CLO4 ← CLO4
  return hasilakhir
EndFunction
```

Deskripsi:

Fungsi *ubahData* digunakan untuk membuat sebuah record *nilaiSTD* dari data

input nilai CLO1–CLO4. Pertama, dibuat variabel sementara *hasilakhir* bertipe nilaiSTD. Lalu setiap field dalam record diisi dengan nilai sesuai parameter yang diberikan. Setelah semua field terisi, fungsi mengembalikan *hasilakhir* sebagai hasilnya.

Dengan adanya fungsi ini, data mahasiswa bisa langsung dikemas dalam satu variabel terstruktur. Hal ini mempermudah pengolahan karena kita tidak perlu lagi mengelola nilai CLO secara terpisah, cukup lewat satu variabel record[9].

#### Unguided 6

*Kamus:*

*CLO1, CLO2, CLO3, CLO4 : real*  
*nilaiAkhir : real*

*Algoritma:*

*Function hitungNilaiAkhir(CLO1, CLO2, CLO3, CLO4 : real) -> real*  
*nilaiAkhir*  $\leftarrow (0.3 * CLO1) + (0.3 * CLO2) + (0.2 * CLO3) + (0.2 * CLO4)$   
*return nilaiAkhir*  
*EndFunction*

Deskripsi:

Fungsi *hitungNilaiAkhir* menerima nilai CLO1–CLO4 sebagai input. Masing-masing nilai dikalikan dengan bobotnya: CLO1 dan CLO2 berbobot 30%, sedangkan CLO3 dan CLO4 berbobot 20%. Hasil perhitungan dijumlahkan menjadi satu nilai akhir. Dengan algoritma ini, perhitungan nilai menjadi otomatis, konsisten, dan sesuai aturan pembobotan yang berlaku[10].

#### Unguided 7

*Kamus:*

*nilaiAkhir : real*  
*indeks : string*

*Algoritma:*

*Function tentukanIndeks(nilaiAkhir : real) -> string*  
*If (nilaiAkhir >= 85) Then*



```
    indeks ← "A"  
Else If (nilaiAkhir ≥ 80) Then  
    indeks ← "AB"  
Else If (nilaiAkhir ≥ 70) Then  
    indeks ← "B"  
Else If (nilaiAkhir ≥ 65) Then  
    indeks ← "BC"  
Else If (nilaiAkhir ≥ 55) Then  
    indeks ← "C"  
Else If (nilaiAkhir ≥ 45) Then  
    indeks ← "D"  
Else  
    indeks ← "E"  
EndIf  
return indeks  
EndFunction
```

Deskripsi:

Fungsi *tentukanIndeks* mengonversi nilai akhir mahasiswa ke indeks huruf sesuai standar penilaian. Percabangan IF – ELSE IF digunakan untuk mengecek kisaran nilai. Misalnya, nilai  $\geq 85$  diberi indeks “A”, nilai antara 70–79 diberi “B”, dan seterusnya. Dengan algoritma ini, hasil penilaian lebih mudah dibaca karena langsung dalam bentuk indeks mutu, bukan hanya angka.[11]

#### E. Kesimpulan

Guided :

Dari ketiga percobaan program C++ yang telah dibuat, dapat disimpulkan bahwa percobaan ini memberikan pemahaman dasar mengenai struktur program C++, pemanfaatan pointer, dan pengelolaan data dengan array yang menjadi fondasi penting dalam mempelajari pemrograman lebih lanjut.

Unguided :

Dari praktikum ini dapat disimpulkan bahwa penggunaan tipe bentukan, variabel, input-output, function, perhitungan berbobot, serta percabangan sangat membantu dalam menyusun program yang terstruktur, efisien, dan mudah dipahami untuk mengelola data nilai mahasiswa.

## F. Referensi

- [1]. Setiyawan, R. D., Hermawan, D., Abdillah, A. F., Mujayanah, A., & Vindua, R. (2024). *Penggunaan Struktur Data Stack dalam Pemrograman C++ dengan Pendekatan Array dan Linked List*. JUTECH: Journal Education and Technology.
- [2]. Hidayat, R. (2020). "Penerapan Bahasa Pemrograman C++ dalam Pembelajaran Algoritma." *Jurnal Teknologi dan Sistem Informasi*, 6(2), 45–50.
- [3]. Putra, A. R. (2021). "Pemanfaatan Pointer dalam Bahasa Pemrograman C++ untuk Pengolahan Data." *Jurnal Teknologi Informasi dan Ilmu Komputer*, 8(3), 233–240.
- [4]. Wibowo, A. S. (2020). "Implementasi Array dalam Bahasa Pemrograman C++ untuk Penyimpanan Data." *Jurnal Ilmiah Teknologi Informasi Asia*, 14(1), 12–18.
- [5]. Kurniawan, D., & Santoso, H. (2020). "Penerapan Konsep Struktur Data dalam Pemrograman." *Jurnal Sistem Informasi dan Teknologi*, 8(2), 101–108.
- [6]. Putra, R. (2021). *Pemanfaatan Tipe Bentuk untuk Representasi Data Mahasiswa*. Jurnal Ilmu Komputer, 10(2), 33–39
- [7]. Santosa, A. (2020). *Konsep Input dan Output dalam Algoritma Pemrograman*. Jurnal Teknologi Informasi, 8(3), 22–28.
- [8]. Saputra, R. A., & Kurniawan, T. (2022). "Perancangan Fungsi Modular dalam Pengembangan Algoritma Pemrograman." *Jurnal Teknologi dan Sistem Informasi*, 10(1), 21–28.
- [9]. Fadilah, M., & Pratama, Y. (2021). "Implementasi Fungsi dalam Pengelolaan Data Mahasiswa." *Jurnal Teknologi Informasi dan Komputer*, 9(2), 55–62.
- [10]. Setiawan, B. (2021). *Sistem Penilaian Berbasis CLO dalam Pendidikan Tinggi*. Jurnal Pendidikan, 12(3), 55–61.
- [11]. Yusliani, T., & Subroto, E. (2020). *Pemodelan Sistem Penilaian Akademik Berbasis Algoritma Percabangan*. Jurnal Teknologi Pendidikan, 9(2), 66–72.