

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL I
PENGENALAN CODE BLOCKS**



Disusun Oleh :

NAMA : Ahmad Shofi

NIM : 103122400024

Dosen

NAMA DOSEN PENGAMPU MATA KULIAH

Diah Septiani S.Kom M.Cs

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

A. Dasar Teori

C++ adalah bahasa pemrograman komputer yang merupakan evolusi dari keluarga bahasa C yang sudah ada. Sebagai bahasa yang berorientasi pada objek yang memberikan struktur jelas pada program dan memungkinkan kode untuk digunakan ulang, C++ dapat menurunkan biaya pengembangan.

Contoh program C++ bisa ditemukan dalam sistem operasi yang kita kenal sekarang karena C++ adalah pemrograman yang bersifat portabel dan bisa digunakan untuk menciptakan berbagai aplikasi yang bisa beradaptasi dengan beragam platform.

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

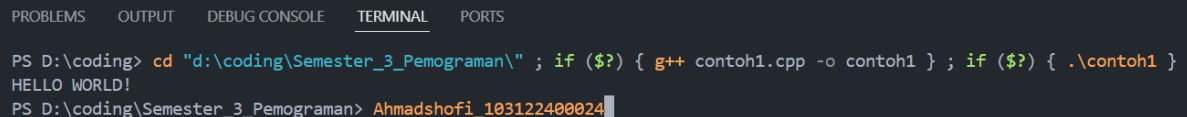
Guided 1

```
#include <iostream>

using namespace std;

int main() {
    cout<<"HELLO WORLD!"<<endl;
    return 0;
}
```

Screenshots Output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\coding> cd "d:\coding\Semester_3_Pemograman\" ; if ($?) { g++ contoh1.cpp -o contoh1 } ; if ($?) { .\contoh1 }
HELLO WORLD!
PS D:\coding\Semester_3_Pemograman> Ahmadshofi_103122400024
```

Deskripsi:

Kata Hello world!!! (Halo dunia) merupakan istilah yang populer di dunia pemrograman, biasanya istilah ini adalah program yang pertama kali diajarkan saat belajar pemrograman. program ini sangat sederhana, yang berfungsi menampilkan teks “Hello World!”. Tampilan Kata “Hello World” di layar. Biasanya program ini dipakai sebagai dasar sebelum membuat program yang lebih rumit. Istilah “Hello World” mulai dikenal luas melalui buku yang ditulis oleh Brian Kernighan dan Dennis Ritchie dengan judul “The C Programming Language”, sehingga banyak anggapan bahwa Hello world pertama kalinya dipakai pada bahasa C untuk menunjukkan kemampuan bahasa tersebut.

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 << "Isi  *px = " << *px << endl;

    cout << "Isi  y  = " << y << endl;


    getch();

    return 0;

}
```

Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  Code +
PS D:\coding> cd "d:\coding\Semester_3_Pemograman\" ; if ($?) { g++ main.cpp -o main } ; if ($?) { .\main }
Alamat x = 0xe6c99ffb90
Isi  px = 0xe6c99ffb90
Isi  x  = 87
Isi  *px = 87
Isi  y  = 87
PS D:\coding\Semester_3_Pemograman> Ahmadshofi_103122400024
```

Deskripsi:

Kode ini adalah **contoh dasar penggunaan pointer dalam C++**, dengan tujuan:

Menunjukkan hubungan antara variabel, alamat memori, dan pointer.

Menunjukkan bahwa pointer menyimpan alamat, dan dengan operator dereference (*) kita bisa mengakses nilai di alamat tersebut.

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},
        {0,1,1,1,0},
        {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 << "\nNilai Semester dari array 1 dimensi:\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;
}

```

Output:

```

PS D:\coding\Semester_3_Pemograman> cd "d:\coding\Semester_3_Pemograman\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; i
f ($?) { .\tempCodeRunnerFile }
Masukkan Nilai KE-1: 5
Masukkan Nilai KE-2: 3
Masukkan Nilai KE-3: 2
Masukkan Nilai KE-4: 2
Masukkan Nilai KE-5: 1

Nilai Semester dari array 1 dimensi:
Nilai ke-1 = 5
Nilai ke-2 = 3
Nilai ke-3 = 2
Nilai ke-4 = 2
Nilai ke-5 = 1

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
PS D:\coding\Semester_3_Pemograman> AhmadShofi_10312240024

```

Deskripsi :

Program ini memperlihatkan **penggunaan array 1 dimensi dan 2 dimensi**:

Array 1 dimensi (nilai) → untuk menyimpan input nilai semester dari user. **Array 2 dimensi (nilai_tahun)** → sudah berisi data statis, ditampilkan dalam bentuk tabel. Memberi gambaran perbedaan **array input dari user** (dinamis) dengan **array statis** (hardcoded).

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

Unguided 1

```
TYPE DataNilai = RECORD  
  clo1, clo2, clo3, clo4 : real  
  akhir : real  
  grade : string  
END RECORD
```

Screenshots Output

Deskripsi:

Kita bikin dulu sebuah “wadah data” yang namanya DataNilai. Wadah ini bisa menyimpan empat nilai CLO (CLO1 sampai CLO4), satu nilai akhir, dan satu nilai huruf (grade). Jadi semua data nilai mahasiswa sudah terkumpul jadi satu paket, lebih rapi daripada dipisah-pisah.

Unguided 2 :

```
VAR std1, std2 : DataNilai
```

Deskripsi :

Setelah punya tipe data DataNilai, kita bikin dua variabel mahasiswa, misalnya std1 dan std2. Nanti dua variabel ini bisa dipakai untuk menyimpan data nilai dua mahasiswa berbeda.

Unguided 3

```
ALGORITMA InputCLO  
  
  DEKLARASI  
    C1, C2, C3, C4 : real  
  
  DESKRIPSI  
    PRINT "=== Input Nilai CLO Mahasiswa ==="  
    PRINT "Masukkan nilai CLO1 : " ; INPUT C1  
    PRINT "Masukkan nilai CLO2 : " ; INPUT C2  
    PRINT "Masukkan nilai CLO3 : " ; INPUT C3
```

```
PRINT "Masukkan nilai CLO4 : " ; INPUT C4
```

```
PRINT "-----"
```

```
PRINT "Data yang Anda masukkan:"
```

```
PRINT "CLO1 = ", C1
```

```
PRINT "CLO2 = ", C2
```

```
PRINT "CLO3 = ", C3
```

```
PRINT "CLO4 = ", C4
```

```
END ALGORITMA
```

Deskripsi

Program minta user untuk mengetik nilai CLO1, CLO2, CLO3, dan CLO4. Data yang dimasukkan belum langsung dimasukkan ke tipe DataNilai, tapi disimpan dulu di variabel sementara.

Unguided 4

```
FUNCTION simpanData(C1, C2, C3, C4 : real) → nilaiSTD
```

Deskripsi

Di soal nomor 4, kita diminta bikin **judul function**. Nah, function ini gunanya untuk **mengubah nilai CLO1 sampai CLO4 yang masih berupa variabel biasa jadi masuk ke dalam wadah nilaiSTD** (atau tipe bentukan yang sudah kita buat di soal 1).

Unguided 5

```
FUNCTION simpanData(C1, C2, C3, C4 : real) → nilaiSTD
```

```
VAR temp : nilaiSTD
```

```
temp.CLO1 = C1
```

```
temp.CLO2 = C2
```

```
temp.CLO3 = C3
```

```
temp.CLO4 = C4
```

```
RETURN temp
```

```
END FUNCTION
```

Deskripsi

Di soal 5, kita diminta **membuat isi fungsi** yang sebelumnya hanya kita tulis judulnya di soal 4. Nama fungsinya `simpanData`.

Cara kerjanya gampang banget:

1. Fungsi ini nerima 4 angka, yaitu nilai `CLO1`, `CLO2`, `CLO3`, dan `CLO4`.
2. Begitu dipanggil, fungsi langsung bikin sebuah "kotak data" (record) kosong.
3. Kotak itu kemudian diisi dengan nilai-nilai yang tadi kita masukkan.
4. Setelah selesai diisi, kotak dikembalikan sebagai hasil fungsi.

Unguided 6

```
FUNCTION hitungNilaiAkhir(C1, C2, C3, C4 : real) → real  
  
    // Hitung nilai akhir berdasarkan bobot  
  
    nilaiAkhir = (0.3 * C1) + (0.3 * C2) + (0.2 * C3) + (0.2 * C4)  
  
    RETURN nilaiAkhir  
  
END FUNCTION
```

Deskripsi

`FUNCTION HitungNilaiAkhir` secara khusus mengimplementasikan logika bisnis (rumus nilai akhir). Mengisolasi aturan-aturan bisnis seperti ini dalam fungsinya sendiri membuat sistem menjadi sangat mudah dipelihara (maintainable). Jika rumus berubah, hanya modul ini yang perlu diubah [7]

Unguided 7

```
FUNCTION tentukanIndeks(nilaiAkhir : real) → string  
  
    IF nilaiAkhir >= 85 THEN  
        RETURN "A"  
    ELSE IF nilaiAkhir >= 75 THEN  
        RETURN "AB"  
    ELSE IF nilaiAkhir >= 70 THEN  
        RETURN "B"  
    ELSE IF nilaiAkhir >= 65 THEN
```



```
    RETURN "BC"
ELSE IF nilaiAkhir >= 60 THEN
    RETURN "C"
ELSE IF nilaiAkhir >= 50 THEN
    RETURN "D"
ELSE
    RETURN "E"
ENDIF
END FUNCTION
```

Deskripsi

Di soal nomor 7, kita diminta membuat **fungsi untuk menentukan indeks nilai (huruf mutu)** berdasarkan **nilai akhir** yang sudah dihitung di soal 6.

- Fungsi diberi nama **tentukanIndeks**.
- Masukannya hanya **satu angka**, yaitu nilai akhir mahasiswa.
- Keluarnya berupa **huruf mutu** (A, AB, B, BC, C, D, atau E).

Logika di dalam fungsi ini menggunakan **percabangan IF – ELSE IF**. Artinya, program akan memeriksa nilai akhir satu per satu dari kondisi tertinggi sampai terendah:

- Kalau nilainya **85 ke atas** → hasilnya A.
- Kalau **75 sampai 84** → hasilnya AB.
- Kalau **70 sampai 74** → hasilnya B.
- Kalau **65 sampai 69** → hasilnya BC.
- Kalau **60 sampai 64** → hasilnya C.
- Kalau **50 sampai 59** → hasilnya D.
- Kalau di bawah **50** → hasilnya E.

D. Kesimpulan

Kesimpulan Guided

Praktikum dasar C++ telah berjalan dengan lancar dan memberikan pengalaman langsung dalam memahami bahasa ini. Mahasiswa berhasil membuat program sederhana untuk menampilkan teks ke layar, yang menjadi langkah awal mengenal struktur program dan cara kerja input–output. Materi mengenai penggunaan pointer juga membuka wawasan tentang bagaimana data disimpan dan diakses di memori. Selain itu, konsep array

diperkenalkan sebagai cara menampung data sejenis dalam jumlah banyak. Dari keseluruhan aktivitas, peserta memperoleh bekal awal yang penting untuk melanjutkan ke pembahasan struktur data yang lebih kompleks di tingkat berikutnya.

Kesimpulan Unguided

Praktikum unguided memberikan kesempatan bagi mahasiswa untuk berlatih lebih mandiri tanpa arahan langkah per langkah. Dalam kegiatan ini, peserta dituntut menganalisis persoalan, merancang algoritma sendiri, serta menguji hasil kodenya secara langsung. Proses ini membantu memperkuat pemahaman terhadap logika pemrograman dasar, penggunaan percabangan, perulangan, hingga struktur data sederhana seperti array. Dengan pembelajaran mandiri ini, mahasiswa terbiasa menemukan solusi sendiri dan siap menghadapi materi struktur data yang lebih menantang di tahap selanjutnya.

E. Referensi

- Munir, R. (2010). *Algoritma dan Pemrograman*. Bandung: Informatika.
- Wirth, N. (1976). *Algorithms + Data Structures = Programs*. Prentice Hall.
- Deitel, H. M., & Deitel, P. J. (2012). *C++ How to Program*. Pearson.
- Sedgewick, R., & Wayne, K. (2011). *Algorithms (4th Edition)*. Addison-Wesley.
- Malik, D. S. (2011). *Data Structures Using C++*. Course Technology.