

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL I
PENGENALAN CODE BLOCKS**



Disusun Oleh :

NAMA : AGUNG RAMADHAN

NIM : 103112430060

Dosen

FAHRUDIN MUKTI WIBOWO

**PROGRAM STUDI STRUKTUR DATA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

A. Dasar Teori

C++ merupakan bahasa pemrograman turunan dari C yang dirancang untuk menggabungkan efisiensi tingkat rendah dengan abstraksi tingkat tinggi. Bahasa ini mendukung berbagai paradigma, mulai dari prosedural, berorientasi objek, hingga pemrograman generik, sehingga fleksibel untuk digunakan di banyak bidang.

Seiring waktu, C++ distandarisasi oleh ISO dan terus berkembang melalui berbagai versi (C++98, C++11, C++17, C++20, hingga C++23). Dengan perpustakaan standar yang kaya, C++ memungkinkan pengembang membuat aplikasi kompleks dengan lebih cepat, tanpa kehilangan kontrol penuh terhadap performa dan manajemen sumber daya.

.

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

Guided 1

```
// Agung Ramadhan
// 103112430060

#include <iostream>
using namespace std;

int main() {
    int hari;
    cout << "Hari 1-7 : ";
    cin >> hari;

    if (hari == 7) { // if else
        cout << "Hari Minggu\n";
    } else {
        cout << "Hari Kerja\n";
    }

    switch (hari) { // Versi switch case
        case 7:
            cout << "Hari Minggu\n";
            break;
        default:
            cout << "Hari Kerja\n";
            break;
    }

    return 0;
}
```

Screenshots Output

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Hari 1-7 : 7
Hari Minggu
Hari Minggu
PS D:\Cicak di dinding\Guided> cd "d:\Cicak di dinding\Guided\" ; if ($?) { g++ hari.cpp -o hari } ; if ($?) { .\hari }
Hari 1-7 : 5
Hari Kerja
Hari Kerja
PS D:\Cicak di dinding\Guided> █
```

Deskripsi:

Program ini dibuat untuk menentukan apakah suatu hari adalah Hari Minggu atau Hari Kerja. Pengguna memasukkan angka 1–7, lalu program memeriksa dengan if-else dan juga switch-case. Jika input bernilai 7, hasilnya “Hari Minggu”, sedangkan untuk angka lain hasilnya “Hari Kerja”. Output ditampilkan dua kali karena kedua metode dipakai.

Guided 2

```
// Agung Ramadhan
// 103112430060

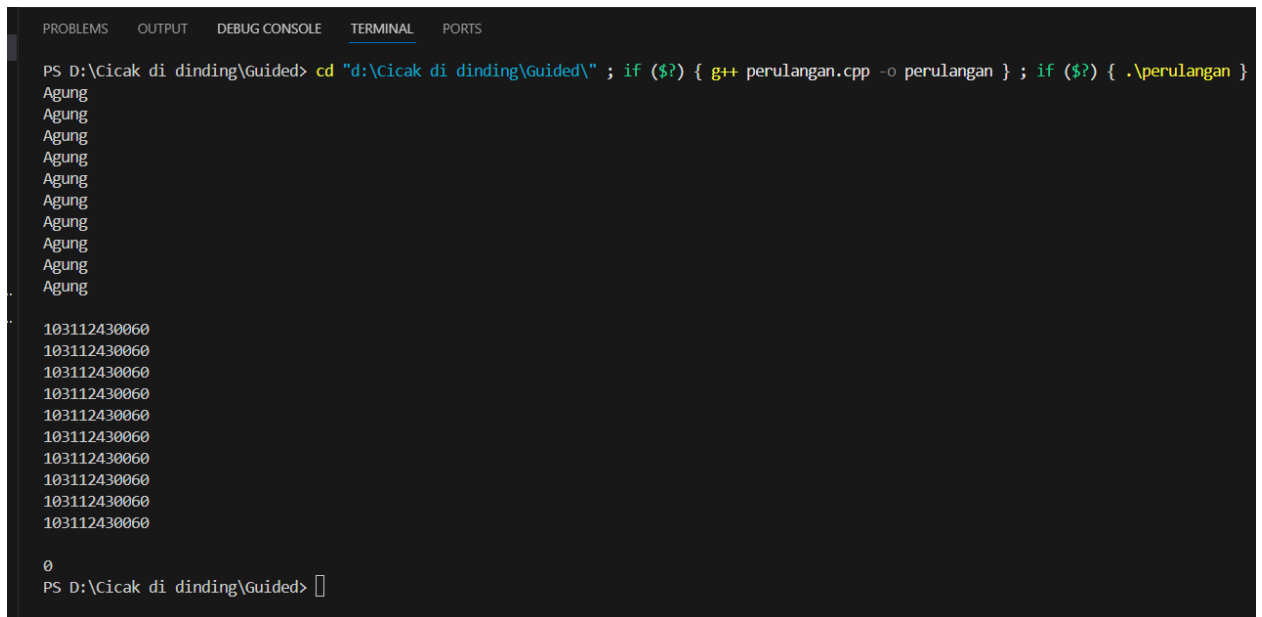
#include<iostream>
using namespace std;

int main(){
    for (int i = 0; i < 10; i++){
        cout << "Agung" << endl;
    }
    cout << endl;

    int i = 0;
    while (i < 10){
        cout << "103112430060" << endl;
        i++;
    }
    cout << endl;

    int j = 0;
    do {
        cout << j;
    } while (j != 0);
}
```

Screenshots Output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Cicak di dinding\Guided> cd "d:\Cicak di dinding\Guided\" ; if ($?) { g++ perulangan.cpp -o perulangan } ; if ($?) { .\perulangan }
Agung
Agung
Agung
Agung
Agung
Agung
Agung
Agung
Agung
Agung
103112430060
103112430060
103112430060
103112430060
103112430060
103112430060
103112430060
103112430060
103112430060
103112430060
0
PS D:\Cicak di dinding\Guided> 
```

Deskripsi:

Program ini menampilkan contoh penggunaan tiga jenis perulangan dalam C++. Pertama, perulangan for digunakan untuk mencetak kata "Agung" sebanyak 10 kali. Kedua, perulangan while mencetak NIM "103112430060" juga sebanyak 10 kali, dengan variabel penghitung yang bertambah setiap iterasi. Terakhir, perulangan do-while digunakan untuk mencetak angka 0 satu kali, karena kondisi $j \neq 0$ langsung salah setelah iterasi pertama. Dengan begitu, program ini menunjukkan perbedaan cara kerja perulangan for, while, dan do-while.

Guided 3

```
// Agung Ramadhan
// 103112430060

#include <iostream>
using namespace std;

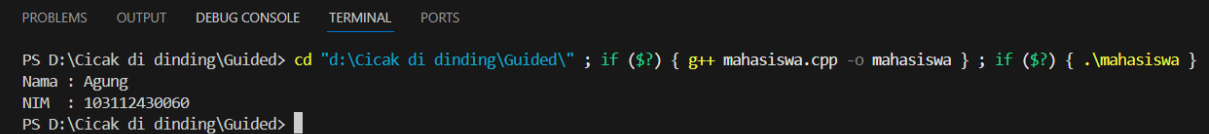
struct Mahasiswa {
    string nama;
    string NIM;
};

int main() {
    Mahasiswa mhs;
    mhs.nama = "Agung";
    mhs.NIM = "103112430060";

    cout << "Nama : " << mhs.nama << endl
         << "NIM  : " << mhs.NIM << endl;

    return 0;
}
```

Screenshots Output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Cicak di dinding\Guided> cd "d:\Cicak di dinding\Guided\" ; if ($?) { g++ mahasiswa.cpp -o mahasiswa } ; if ($?) { .\mahasiswa }
Nama : Agung
NIM : 103112430060
PS D:\Cicak di dinding\Guided> |
```

Deskripsi:

Program ini menggunakan struct untuk menyimpan data mahasiswa berupa nama dan NIM. Di dalam main(), dibuat variabel mhs lalu diisi dengan nilai "Agung" untuk nama dan "103112430060" untuk NIM. Program kemudian menampilkan kedua data tersebut ke layar sehingga hasilnya berupa teks berisi nama dan NIM mahasiswa.

Guided 4

```
// Agung Ramadhan
// 103112430060

#include <iostream>
using namespace std;

struct Mahasiswa
{
    string nama;
    string NIM;
};

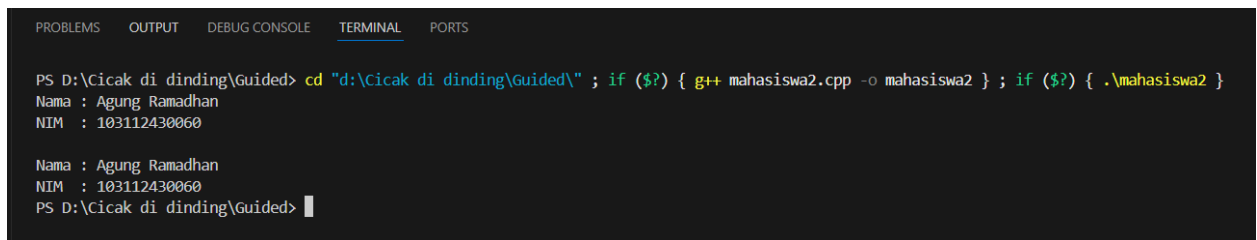
int main() {
    Mahasiswa mhs;

    cout << "Nama : ";
    getline(cin, mhs.nama);
    cout << "NIM : ";
    cin >> mhs.NIM;

    cout << endl;

    cout << "Nama : " << mhs.nama << endl
         << "NIM : " << mhs.NIM << endl;
}
```

Screenshots Output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Cicak di dinding\Guided> cd "d:\Cicak di dinding\Guided\" ; if ($?) { g++ mahasiswa2.cpp -o mahasiswa2 } ; if ($?) { .\mahasiswa2 }
Nama : Agung Ramadhan
NIM : 103112430060

Nama : Agung Ramadhan
NIM : 103112430060
PS D:\Cicak di dinding\Guided> |
```

Deskripsi:

Program ini berfungsi untuk menerima input data mahasiswa menggunakan struct di C++. Struktur “Mahasiswa” memiliki dua atribut yaitu “nama” dan ”NIM”. Di dalam fungsi “main()”, program meminta pengguna mengetikkan nama dengan “getline(cin, mhs.nama)” agar bisa membaca satu baris penuh termasuk spasi, lalu meminta NIM dengan “cin >> mhs.NIM”. Setelah itu, program menampilkan kembali data yang sudah dimasukkan, yaitu nama dan NIM mahasiswa. Dengan begitu, program ini menunjukkan cara menyimpan dan menampilkan input pengguna menggunakan struct.

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

Unguided 1

```
// Agung Ramadhan
// 103112430060

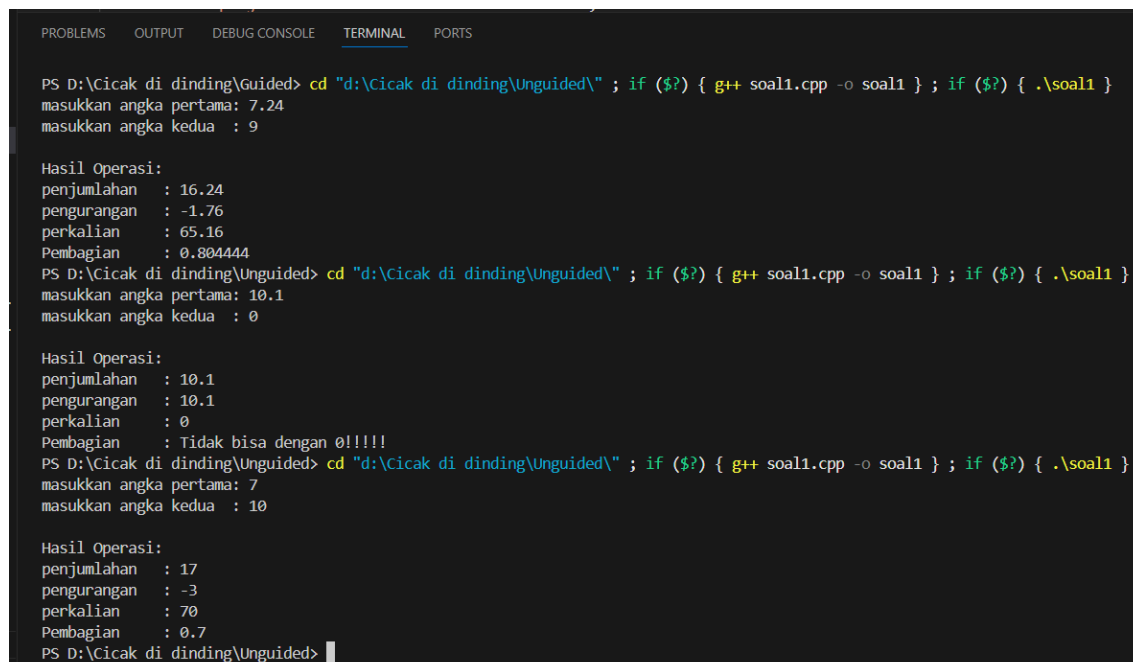
#include <iostream>
using namespace std;

int main() {
    float num1, num2;
    cout << "masukkan angka pertama: ";
    cin >> num1;
    cout << "masukkan angka kedua : ";
    cin >> num2;

    cout << "\nHasil Operasi: " << endl;
    cout << "penjumlahan : " << num1 + num2 << endl;
    cout << "pengurangan : " << num1 - num2 << endl;
    cout << "perkalian : " << num1 * num2 << endl;

    if (num2 !=0) {
        cout << "Pembagian : " << num1 / num2 << endl;
    } else {
        cout << "Pembagian : Tidak bisa dengan 0!!!!!" << endl;
    }
    return 0;
}
```

Screenshots Output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Cicak di dinding\Guided> cd "d:\Cicak di dinding\Unguided\" ; if ($?) { g++ soal1.cpp -o soal1 } ; if ($?) { .\soal1 }
masukkan angka pertama: 7.24
masukkan angka kedua : 9

Hasil Operasi:
penjumlahan : 16.24
pengurangan : -1.76
perkalian : 65.16
Pembagian : 0.804444
PS D:\Cicak di dinding\Unguided> cd "d:\Cicak di dinding\Unguided\" ; if ($?) { g++ soal1.cpp -o soal1 } ; if ($?) { .\soal1 }
masukkan angka pertama: 10.1
masukkan angka kedua : 0

Hasil Operasi:
penjumlahan : 10.1
pengurangan : 10.1
perkalian : 0
Pembagian : Tidak bisa dengan 0!!!!
PS D:\Cicak di dinding\Unguided> cd "d:\Cicak di dinding\Unguided\" ; if ($?) { g++ soal1.cpp -o soal1 } ; if ($?) { .\soal1 }
masukkan angka pertama: 7
masukkan angka kedua : 10

Hasil Operasi:
penjumlahan : 17
pengurangan : -3
perkalian : 70
Pembagian : 0.7
PS D:\Cicak di dinding\Unguided>
```

Deskripsi:

Program ini merupakan kalkulator sederhana dengan bahasa C++. Program meminta pengguna memasukkan dua angka, yaitu num1 dan num2. Setelah input diterima, program menampilkan hasil dari operasi dasar aritmatika: penjumlahan, pengurangan, dan perkalian. Untuk operasi pembagian, program menggunakan percabangan if agar lebih aman. Jika num2 tidak sama dengan nol, maka hasil pembagian ditampilkan, tetapi jika num2 bernilai nol, program akan menampilkan pesan bahwa pembagian dengan nol tidak bisa dilakukan. Dengan demikian, program ini memperlihatkan penggunaan input-output, operasi aritmatika, dan kontrol alur dengan if dalam C++.

Unguided 2

```
// Agung Ramadhan
// 103112430060

#include <iostream>
using namespace std;

int main() {
    int num;
    cout << "Masukkan angka (0-100): ";
    cin >> num;

    string satuan[] = {"", "satu", "dua", "tiga", "empat", "lima",
                       "enam", "tujuh", "delapan", "sembilan"};

    if (num < 0 || num > 100) {
        cout << "Angka harus 0 s.d 100\n";
        return 0;
    }

    cout << num << " : ";

    switch (num) {
        case 0: cout << "nol"; break;
        case 10: cout << "sepuluh"; break;
        case 11: cout << "sebelas"; break;
        case 100: cout << "seratus"; break;
        default:
            if (num < 10) cout << satuan[num];
            else if (num < 20) cout << satuan[num - 10] << " belas";
            else {
                cout << satuan[num / 10] << " puluh";
                if (num % 10 != 0) cout << " " << satuan[num % 10];
            }
    }
    cout << endl;
    return 0;
}
```

Screenshots Output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Cicak di dinding\Unguided> cd "d:\Cicak di dinding\Unguided\" ; if ($?) { g++ soal2.cpp -o soal2 } ; if ($?) { .\soal2 }
Masukkan angka (0-100): 97
97 : sembilan puluh tujuh
PS D:\Cicak di dinding\Unguided> cd "d:\Cicak di dinding\Unguided\" ; if ($?) { g++ soal2.cpp -o soal2 } ; if ($?) { .\soal2 }
Masukkan angka (0-100): 100
100 : seratus
PS D:\Cicak di dinding\Unguided> cd "d:\Cicak di dinding\Unguided\" ; if ($?) { g++ soal2.cpp -o soal2 } ; if ($?) { .\soal2 }
Masukkan angka (0-100): 17
17 : tujuh belas
PS D:\Cicak di dinding\Unguided> cd "d:\Cicak di dinding\Unguided\" ; if ($?) { g++ soal2.cpp -o soal2 } ; if ($?) { .\soal2 }
Masukkan angka (0-100): 7
7 : tujuh
PS D:\Cicak di dinding\Unguided> 
```

Deskripsi:

Program ini digunakan untuk mengubah angka menjadi bentuk tulisan dalam bahasa Indonesia untuk rentang 0 sampai 100. Pertama, pengguna diminta memasukkan sebuah angka, lalu program memeriksa apakah nilainya valid (0–100). Jika tidak sesuai, program menampilkan pesan kesalahan. Data kata-kata dasar angka dari satu sampai sembilan disimpan dalam array satuan. Kemudian program menggunakan switch-case untuk menangani beberapa kasus khusus, yaitu angka 0, 10, 11, dan 100. Untuk angka lain, logika if-else digunakan: jika angka kurang dari 10 ditulis sesuai array satuan, jika antara 12–19 ditulis dengan format “belas”, sedangkan jika lebih besar dituliskan dengan pola “puluh” dan ditambah satuan jika tidak genap kelipatan 10. Dengan begitu, program ini mampu menampilkan angka yang dimasukkan dalam bentuk kata, misalnya input 25 akan menghasilkan output 25 : dua puluh lima.

Unguided 3

```
// Agung Ramadhan
// 103112430060

#include <iostream>
using namespace std;

int main() {
    int num;
    cout << "Input: ";
    cin >> num;

    for (int i = num; i >= 1; i--) {
        for (int s = 0; s < (num - i); s++) {
            cout << " ";
        }
        for (int j = i; j >= 1; j--) {
            cout << j << " ";
        }
        cout << "* ";
        for (int j = 1; j <= i; j++) {
            cout << j << " ";
        }
        cout << endl;
    }
    for (int s = 0; s < num; s++) {
        cout << " ";
    }
    cout << "*" << endl;

    return 0;
}
```

Screenshots Output

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Cicak di dinding\Unguided> cd "d:\Cicak di dinding\Unguided\" ; if ($?) { g++ soal3.cpp -o soal3 } ; if ($?) { .\soal3 }
Input: 10
10 9 8 7 6 5 4 3 2 1 * 1 2 3 4 5 6 7 8 9 10
 9 8 7 6 5 4 3 2 1 * 1 2 3 4 5 6 7 8 9
 8 7 6 5 4 3 2 1 * 1 2 3 4 5 6 7 8
 7 6 5 4 3 2 1 * 1 2 3 4 5 6 7
 6 5 4 3 2 1 * 1 2 3 4 5 6
 5 4 3 2 1 * 1 2 3 4 5
 4 3 2 1 * 1 2 3 4
 3 2 1 * 1 2 3
 2 1 * 1 2
 1 * 1
 *

PS D:\Cicak di dinding\Unguided> cd "d:\Cicak di dinding\Unguided\" ; if ($?) { g++ soal3.cpp -o soal3 } ; if ($?) { .\soal3 }
Input: 7
7 6 5 4 3 2 1 * 1 2 3 4 5 6 7
6 5 4 3 2 1 * 1 2 3 4 5 6
5 4 3 2 1 * 1 2 3 4 5
4 3 2 1 * 1 2 3 4
3 2 1 * 1 2 3
2 1 * 1 2
1 * 1
 *

PS D:\Cicak di dinding\Unguided> 
```

Deskripsi:

Program ini menampilkan pola simetris angka dengan tanda * di tengahnya berdasarkan input pengguna. Angka akan dicetak menurun dari nilai input hingga 1 di sisi kiri, lalu dicetak kembali menaik dari 1 hingga nilai yang sama di sisi kanan, dengan simbol * sebagai pemisah di tengah. Setiap baris diberi spasi agar pola terlihat rapih, dan pada akhir program ditambahkan satu baris berisi sebuah * yang sejajar di bawah pola.

D. Kesimpulan

Kesimpulan dari seluruh program yang telah dipaparkan adalah bahwa semuanya berfungsi sebagai latihan dasar untuk memahami konsep fundamental dalam bahasa C++. Program-program tersebut memperlihatkan bagaimana cara menerima input dari pengguna, mengolahnya, dan menampilkan output ke layar. Konsep percabangan seperti if-else dan switch-case digunakan untuk mengambil keputusan berdasarkan kondisi tertentu, sementara konsep perulangan seperti for, while, dan do-while digunakan untuk mengulang proses sesuai kebutuhan. Penggunaan struct menunjukkan cara menyimpan data yang lebih terorganisir, misalnya menyimpan nama dan NIM mahasiswa. Ada pula program yang mengimplementasikan operasi aritmatika sederhana (kalkulator), mengubah angka menjadi bentuk tulisan, serta membentuk pola angka dengan simbol *. Dari semua contoh tersebut, dapat disimpulkan bahwa program-program ini melatih logika, pemahaman alur kerja program, serta keterampilan dasar dalam menulis kode C++ sebagai bekal untuk membuat program yang lebih kompleks.

E. Referensi

<https://en.wikipedia.org/wiki/C%2B%2B>

<https://www.geeksforgeeks.org/cpp/structures-in-cpp/>