

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL I
PENGENALAN CODE BLOCKS**



Disusun Oleh :

NAMA : Taufik Hafit Zakaria

NIM : 103112430093

Dosen

WAHYU ANDI SAPUTRA

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

A. Dasar Teori

Code Blocks adalah sebuah Integrated Development Environment (IDE) yang bersifat gratis, sumber terbuka (open-source), dan lintas platform (cross-platform). IDE ini dirancang untuk mempermudah proses pengembangan program, khususnya yang menggunakan bahasa C++. Code Blocks menyediakan lingkungan terpadu yang efisien bagi pengguna untuk menulis kode, melakukan kompilasi (membangun program), menjalankan, serta melakukan proses debugging untuk mencari dan memperbaiki kesalahan. Dengan antarmuka grafis yang intuitif, Code Blocks memberikan kemudahan dalam pengelolaan proyek dan proses pengembangan perangkat lunak secara keseluruhan.

Bahasa pemrograman C++ merupakan bahasa pemrograman tingkat tinggi yang diciptakan oleh Bjarne Stroustrup sebagai pengembangan dari bahasa C. Keunggulan utamanya adalah penambahan konsep pemrograman berorientasi objek (Object-Oriented Programming), yang diwujudkan melalui penggunaan class. Dalam praktikum, proses pengembangan program menggunakan Code Blocks meliputi beberapa tahapan penulisan kode sumber (source code) pada editor, kompilasi kode untuk mengubahnya menjadi berkas yang dapat dieksekusi (executable file), dan eksekusi program untuk melihat hasilnya.


Proses kompilasi dan eksekusi di dalam Code Blocks dapat diakses melalui menu Build. Terdapat beberapa perintah utama yang sering digunakan:

- Build : Berfungsi untuk membangun atau mengompilasi seluruh kode sumber dalam proyek menjadi sebuah program.
- Run : Berfungsi untuk menjalankan program yang sebelumnya telah berhasil di-build.
- Build and Run : Merupakan kombinasi perintah yang melakukan proses Build dan Run secara berurutan. Perintah ini sangat efisien digunakan untuk menguji program setelah melakukan perubahan pada kode.

Untuk dapat mengembangkan program secara efektif, pemahaman terhadap elemen-elemen dasar C++ sangatlah penting. Setiap program C++ memiliki fungsi utama `int main()` yang menjadi titik awal eksekusi program. Perintah untuk menampilkan output ke layar menggunakan `cout`, sedangkan untuk menerima input dari pengguna menggunakan `cin`. Keduanya memerlukan penyertaan header file `#include <iostream>`. Selain itu, penggunaan variabel sebagai tempat penyimpanan data yang memiliki tipe data spesifik (seperti `int`, `float`, dan `char`) serta pemahaman struktur kontrol program seperti percabangan (`if-else`) dan perulangan (`for`, `while`) menjadi landasan dalam membangun logika program.

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

Guided 1

```
[Struktur Data] > LAPRAK1 >  guided1.cpp
1  #include <iostream>
2  using namespace std;
3
Windsurf: Refactor | Explain | Generate Function Comment | X
4  int main() { //Taufik Hafit Zakaria || 103112430093 || 12-IF-06
5      int hari;
6      cout << "Hari 1-7: ";
7      cin >> hari;
8      if (hari == 7)
9      {
10         cout << "Hari minggu\n";
11     }
12     else
13     {
14         cout << "Hari kerja\n";
15     }
16
17     switch (hari)
18     {
19     case 7:
20         cout << "Hari minggu\n";
21         break;
22
23     default:
24         cout << "Hari kerja\n";
25         break;
26     }
27 }
```

Screenshots Output


```
Hari 1-7: 6
Hari kerja
Hari kerja
PS C:\programing\Struktur Data\LAPRAK1>
```

```
Hari 1-7: 7
Hari minggu
Hari minggu
PS C:\programing\Struktur Data\LAPRAK1>
```

Deskripsi:

Program ini merupakan program penentu hari yang dibuat dengan bahasa C++. Program ini akan meminta pengguna untuk menginputkan sebuah angka dari 1 sampai 7 yang mewakili hari dalam seminggu. Khusus untuk proses penentuannya, program ini sengaja menggunakan dua metode kondisional yang berbeda, yaitu if-else dan switch, dengan alasan untuk mendemonstrasikan bagaimana kedua struktur tersebut dapat digunakan untuk menyelesaikan masalah yang sama. Pengecekan ini penting untuk memisahkan satu kondisi spesifik, yaitu jika angka yang diinput adalah 7 yang dianggap sebagai "Hari minggu", dari semua kemungkinan angka lainnya yang akan dianggap sebagai "Hari kerja", sehingga program dapat memberikan output yang akurat sesuai dengan input yang diberikan.

Guided 2

```
[Struktur Data] > LAPRAK1 >  guided2.cpp
1  #include <iostream>
2  using namespace std;
3
Windsurf: Refactor | Explain | Generate Function Comment | X
4  int main() { //Taufik Hafit Zakaria || 103112430093 || 12-IF-06
5      for (int i = 0; i < 10; i++)
6      {
7          cout << "Taufik" << endl;
8      }
9      cout << endl;
10
11     int i = 0;
12     while (i < 10)
13     {
14         cout << "103112430093" << endl;
15         i++;
16     }
17     cout << endl;
18
19     int j = 0;
20     do
21     {
22         cout << j;
23     } while (j != 0);
24
25 }
```

Screenshots Output

```
Taufik
Taufik
Taufik
Taufik
Taufik
Taufik
Taufik
Taufik
Taufik
Taufik
Taufik

103112430093
103112430093
103112430093
103112430093
103112430093
103112430093
103112430093
103112430093
103112430093
103112430093
103112430093

0
PS C:\programing\Struktur Data\LAPRAK1>
```

Deskripsi:

Program ini merupakan program demonstrasi perulangan yang dibuat dengan bahasa C++. Program ini akan mencetak teks dan angka secara berulang untuk menunjukkan cara kerja dari tiga jenis perulangan yang berbeda. Khusus untuk proses perulangannya, program ini sengaja menggunakan tiga struktur berbeda yaitu for, while, dan do-while dengan alasan untuk mendemonstrasikan bagaimana setiap jenis perulangan memiliki sintaks dan perilaku yang khas. Pengecekan ini penting untuk membedakan perulangan for dan while yang mengecek kondisi di awal, dengan perulangan do-while yang menjalankan aksinya minimal satu kali sebelum mengecek kondisi di akhir, sehingga program dapat secara efektif menampilkan perbedaan fundamental antara ketiga struktur perulangan tersebut.

Guided 3

```
[Struktur Data] > LAPRAK1 > guided3.cpp
1  #include <iostream>
2  using namespace std;
3
4  struct Mahasiswa
5  {
6      string nama;
7      string NIM;
8  };
9
10 int main() { //Taufik Hafit Zakaria || 103112430093 || 12-IF-06
11     Mahasiswa mhs;
12     mhs.nama = "Taufik";
13     mhs.NIM = "1103112430093";
14
15     cout << "Nama : " << mhs.nama << endl
16         << "NIM : " << mhs.NIM;
17 }
```


Screenshots Output

```
Nama : Taufik
NIM : 1103112430093
PS C:\programing\Struktur Data\LAPRAK1>
```

Deskripsi:

Program ini merupakan program data mahasiswa sederhana yang dibuat dengan bahasa C++. Program ini akan menyimpan dan menampilkan data seorang mahasiswa yang terdiri dari nama dan NIM. Khusus untuk proses pengelompokan datanya, program dilengkapi dengan struktur struct dengan alasan untuk menggabungkan beberapa variabel yang saling terkait (nama dan NIM) menjadi satu kesatuan tipe data baru bernama Mahasiswa. Struktur ini penting untuk membuat kode lebih terorganisir, karena data-data yang saling berhubungan dapat dikelola sebagai satu objek tunggal (mhs), sehingga program dapat menjamin data ditampilkan dan dikelola sebagai satu kesatuan yang logis dan rapi.

Guided 4

```
[Struktur Data] > LAPRAK1 >  guided4.cpp
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  Windsurf: Refactor | Explain
6  struct Mahasiswa
7  {
8      string nama;
9      string NIM;
10 };
11
12 Windsurf: Refactor | Explain | Generate Function Comment | X
13 int main() { //Taufik Hafit Zakaria || 103112430093 || 12-IF-06
14     Mahasiswa mhs;
15
16     cout << "Nama : ";
17     getline(cin, mhs.nama);
18     cout << "NIM : ";
19     cin >> mhs.NIM;
20
21     cout << "Nama : " << mhs.nama << endl
22         << "NIM : " << mhs.NIM;
23 }
```

Screenshots Output

```
Nama : Taufik Hafit Zakaria
NIM : 103112430093
=====
Nama : Taufik Hafit Zakaria
NIM : 103112430093
PS C:\programing\Struktur Data\LAPRAK1> |
```

Deskripsi:

Program ini merupakan program input data mahasiswa yang interaktif, dibuat dengan bahasa C++. Program ini akan meminta pengguna untuk menginputkan nama dan NIM, kemudian menampilkannya kembali ke layar. Khusus untuk proses input nama, program dilengkapi dengan fungsi `getline` dengan alasan untuk memastikan seluruh nama lengkap, termasuk yang mengandung spasi, dapat dibaca dengan benar. Penggunaan fungsi ini penting karena `cin` biasa akan berhenti membaca setelah menemukan spasi, sehingga tidak akan menangkap nama yang terdiri lebih dari satu kata. Dengan demikian, program dapat menjamin bahwa keseluruhan nama mahasiswa dapat diinput dan ditampilkan secara akurat.

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

Unguided 1

```
[Struktur Data] > LAPRAK1 > G+ operasiDasar.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  int main() { //Taufik Hafit Zakaria || 103112430093 || 12-IF-06
5      float angka1, angka2, tambah, kurang, kali, bagi;
6      cout << "\nMasukkan angka pertama: ";
7      cin >> angka1;
8      cout << "Masukkan angka kedua: ";
9      cin >> angka2;
10     tambah = angka1 + angka2;
11     kurang = angka1 - angka2;
12     kali = angka1 * angka2;
13     cout << "=====\n";
14     cout << "Hasil tambah: " << tambah << endl;
15     cout << "Hasil kurang: " << kurang << endl;
16     cout << "Hasil kali: " << kali << endl;
17     if (angka2 != 0) {
18         bagi = angka1 / angka2;
19         cout << "Hasil bagi: " << bagi << endl;
20     } else {
21         cout << "Pembagi tidak boleh 0" << endl;
22     }
23     return 0;
24 }
```

Screenshots Output

```
Masukkan angka pertama: 4
Masukkan angka kedua: 3
=====
Hasil tambah: 7
Hasil kurang: 1
Hasil kali: 12
Hasil bagi: 1.33333
PS C:\programing\Struktur Data\LAPRAK1>
```



```

Masukkan angka pertama: 6
Masukkan angka kedua: 0
=====
Hasil tambah: 6
Hasil kurang: 6
Hasil kali: 0
Pembagi tidak boleh 0
PS C:\programing\Struktur Data\LAPRAK1>

```

Deskripsi:

Program ini merupakan kalkulator sederhana yang dibuat dengan bahasa C++. Program ini akan meminta pengguna untuk memasukkan angka pertama dan angka kedua, berupa bilangan float agar dapat memproses dan menampilkan hasil dalam bentuk bilangan desimal (pecahan). Khusus untuk operasi pembagian, program dilengkapi dengan struktur kondisional if dengan alasan untuk melakukan validasi input dan mencegah error. Pengecekan ini penting untuk memastikan angka kedua yang bertindak sebagai pembagi tidak bernilai nol, karena pembagian dengan nol adalah operasi yang tidak terdefinisi dan akan menghasilkan output yang tidak valid, sehingga program dapat menjamin keakuratan hasil yang ditampilkan.

Unguided 2

```

[Struktur Data] > LAPRAK1 > G bilangan.cpp
1  #include <iostream>
2  using namespace std;
3
4  Windsurf: Refactor | Explain | Generate Function Comment | X
5  int main() { //Taufik Hafit Zakaria || 103112430093 || 12-IF-06
6      string satuan={"nol","satu","dua","tiga","empat","lima","enam","tujuh","delapan","sembilan"};
7      int angka;
8      cout<<"masukan angka (0-100): ";
9      cin>>angka;
10
11     cout<<angka<<" dalam kata adalah: ";
12     if (angka==100) cout<<" seratus";
13     else if (angka>=10 && angka<20) {
14         if (angka==10) cout<<"sepuluh";
15         else if (angka==11) cout<<"sebelas";
16         else cout<<satuan[angka%10]<<" belas";
17     } else if (angka>=20 && angka<100) {
18         cout<<satuan[angka/10]<<" puluh";
19         if (angka%10!=0) cout<<" "<<satuan[angka%10];
20     } else if (angka>=0 && angka<10) {
21         cout<<satuan[angka];
22     } else {
23         cout<<"angka di luar jangkauan";
24     }

```

Screenshots Output

```
masukan angka (0-100): 69  
69 dalam kata adalah: enam puluh sembilan  
PS C:\programing\Struktur Data\LAPRAK1> cd
```

```
masukan angka (0-100): 100  
100 dalam kata adalah: seratus  
PS C:\programing\Struktur Data\LAPRAK1> 
```

Deskripsi:

Program ini merupakan program konversi angka ke tulisan yang dibuat dengan bahasa C++. Program ini akan meminta pengguna untuk menginputkan sebuah bilangan bulat dari 0 hingga 100, lalu menampilkannya dalam bentuk tulisan. Khusus untuk proses konversinya, program dilengkapi dengan array string dan struktur kondisional if-else if dengan alasan untuk menangani aturan penulisan yang berbeda untuk setiap rentang angka. Penggunaan array untuk menyimpan nama satuan dan if-else if untuk memisahkan logika penanganan angka satuan, belasan, dan puluhan ini penting karena menyederhanakan proses yang kompleks, sehingga program dapat menjamin keakuratan hasil tulisan yang ditampilkan.

Unguided 3

```
[Struktur Data] > LAPRAK1 > pola.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main() { //Taufik Hafit Zakaria || 103112430093 || 12-IF-06
5      int angka;
6      cout << "masukan angka : ";
7      cin >> angka;
8
9      for (int i = angka; i >= 1; i--)
10     {
11         for (int s = 0; s < angka - i; s++)
12             cout << " ";
13
14         for (int j = i; j >= 1; j--)
15             cout << j << " ";
16
17         cout << "* ";
18
19         for (int j = 1; j <= i; j++)
20             cout << j << " ";
21
22         cout << endl;
23     }
24
25     for (int s = 0; s < angka; s++)
26         cout << " ";
27     cout << "*" << endl;
28
29     return 0;
30 }
```

Screenshots Output

```
masukan angka : 3
3 2 1 * 1 2 3
 2 1 * 1 2
  1 * 1
   *
PS C:\programming\Struktur Data\LAPRAK1>
```

Deskripsi:

Program ini merupakan program pencetak pola piramida yang dibuat dengan bahasa C++. Program ini akan meminta pengguna untuk memasukkan sebuah bilangan, lalu mencetak pola piramida terbalik yang simetris lengkap dengan angka dan bintang. Khusus untuk proses pembentukan polanya, program dilengkapi dengan struktur perulangan bersarang (nested loop) dengan alasan untuk mengontrol setiap elemen pada tiap baris secara terpisah, termasuk spasi untuk perataan tengah. Struktur ini penting karena memungkinkan program untuk secara dinamis mengatur jumlah spasi di awal, urutan angka menurun, dan urutan angka menaik pada setiap baris, sehingga program dapat menjamin pola piramida yang simetris dan rapi dapat ditampilkan sesuai dengan input yang diberikan.

D. Kesimpulan

Praktikum ini berhasil memberikan pemahaman mendalam mengenai penerapan konsep dasar pemrograman C++ menggunakan IDE Code::Blocks. Melalui implementasi serangkaian program yang berbeda, baik dari bagian guided maupun unguided, mahasiswa dapat menguasai struktur dasar pemrograman meliputi input/output, percabangan, perulangan, dan operator aritmatika. Program kalkulator melatih logika operasi matematika dengan validasi kondisi; program konversi angka ke teks mengajarkan manipulasi data menggunakan array dan struktur kontrol yang kompleks; sedangkan program pencetak pola piramida melatih kreativitas dalam penerapan perulangan bersarang. Secara keseluruhan, praktikum ini tidak hanya memperkuat pemahaman teoretis algoritma tetapi juga mengembangkan kemampuan logika pemrograman yang menjadi fondasi esensial untuk mempelajari konsep pemrograman yang lebih kompleks di masa depan.

E. Referensi

1. Ma'arif, A. (2022). Modul Praktikum Dasar Pemrograman dengan Code Blocks. Universitas Ahmad Dahlan. Diakses dari <http://eprints.uad.ac.id/id/eprint/32511>
2. Schildt, H. (2017). C++: The Complete Reference (5th ed.). McGraw-Hill Education.
3. Stroustrup, B. (2013). The C++ Programming Language (4th ed.). Addison-Wesley Professional.
4. Saputra, W. A. (2025). Modul 1: Pengenalan Code Blocks. Purwokerto: Program Studi Struktur Data, Fakultas Informatika, Telkom University Purwokerto.