

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL I
PENGENALAN CODE BLOCKS**



Disusun Oleh :

NAMA : Muhammad Fachri Auravyano Saka

NIM : 103112430180

Dosen

FAHRUDIN MUKTI WIBOWO

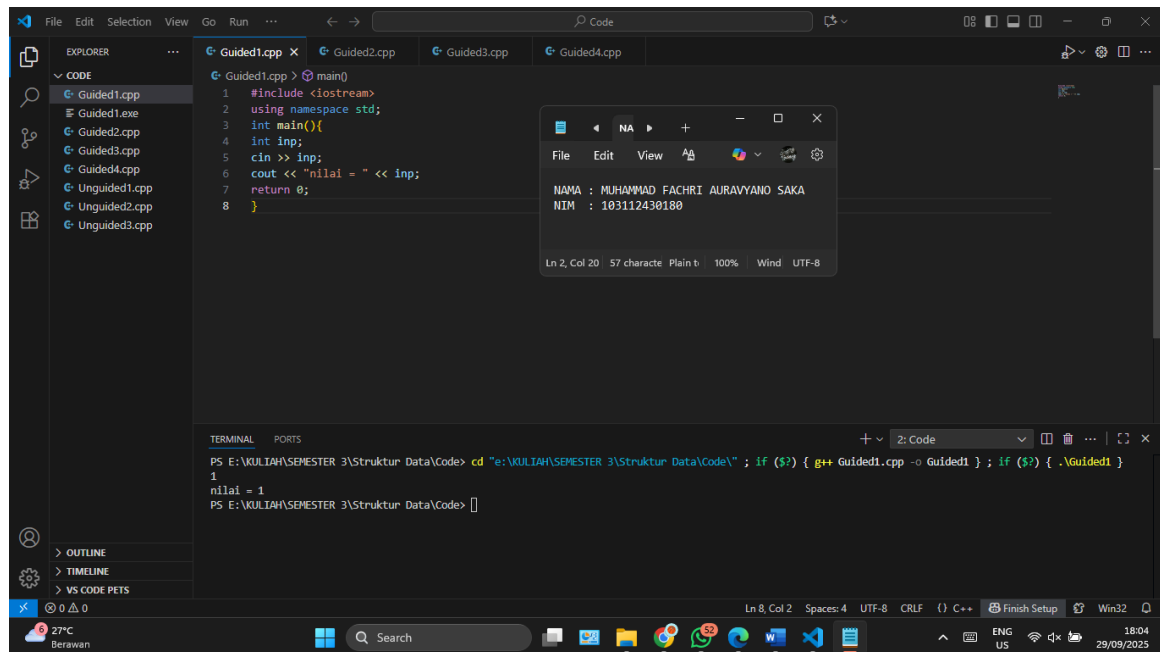
**PROGRAM STUDI STRUKTUR DATA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

A. Dasar Teori

Dalam pemrograman C++, terdapat beberapa komponen dasar yang penting dipahami. Pertama, struktur program yang biasanya terdiri dari fungsi utama `main()` sebagai titik awal eksekusi program. Kedua, penggunaan library seperti `<iostream>` yang berguna untuk melakukan proses input dan output data. Ketiga, konsep variabel dan tipe data yang berfungsi sebagai tempat penyimpanan nilai dengan jenis tertentu, misalnya `int` untuk bilangan bulat dan `float` untuk bilangan pecahan. Selain itu, ada juga operator yang digunakan untuk melakukan operasi aritmetika maupun logika, serta statement atau perintah yang menentukan alur eksekusi program.

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

Guided 1



```
#include <iostream>
using namespace std;
int main(){
    int inp;
    cin >> inp;
    cout << "nilai = " << inp;
    return 0;
}
```

Screenshots Output

The screenshot shows the Visual Studio Code interface with the file `Guided1.cpp` open. The code is as follows:

```
1 #include <iostream>
2 using namespace std;
3 int main(){
4     int inp;
5     cin >> inp;
6     cout << "nilai = " << inp;
7     return 0;
8 }
```

The terminal window shows the command to compile and run the program:

```
PS E:\KULIAH\SEMESTER 3\Struktur Data\Code> cd "e:\KULIAH\SEMESTER 3\Struktur Data\Code\" ; if ($?) { g++ Guided1.cpp -o Guided1 } ; if ($?) { .\Guided1 }
nilai = 1
PS E:\KULIAH\SEMESTER 3\Struktur Data\Code>
```

A small window titled "NA" is overlaid on the code, displaying the output:

```
NAMA : MUHAMMAD FACHRI AURAVYANO SAKA
NIM : 103112430180
```

Deskripsi:

Program di atas adalah contoh sederhana penggunaan bahasa C++ untuk menerima input dari pengguna dan menampilkannya kembali. Pertama, program menyertakan library `<iostream>` yang digunakan untuk proses input dan output. Kemudian, fungsi `main()` menjadi titik awal eksekusi program. Variabel `inp` bertipe `int` dideklarasikan untuk menyimpan nilai yang dimasukkan pengguna. Perintah `cin >> inp;` berfungsi untuk mengambil input berupa angka dari pengguna, sedangkan `cout << "nilai = " << inp;` akan menampilkan kembali angka yang sudah dimasukkan ke layar dengan format “nilai = ...”. Dengan demikian, program ini dapat digunakan untuk membaca sebuah bilangan dari pengguna dan langsung menampilkan hasilnya.

Guided 2

The screenshot shows the Visual Studio Code interface with the file `Guided2.cpp` open. The code is as follows:

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     char ch;
5     cout << "Masukkan sebuah karakter: ";
6     ch = getchar(); // Menggunakan getchar() untuk membaca satu karakter
7     cout << "Karakter yang Anda masukkan adalah: " << ch << endl;
8     return 0;
9 }
```

The terminal window shows the command to compile and run the program:

```
PS E:\KULIAH\SEMESTER 3\Struktur Data\Code> cd "e:\KULIAH\SEMESTER 3\Struktur Data\Code\" ; if ($?) { g++ Guided2.cpp -o Guided2 } ; if ($?) { .\Guided2 }
Masukkan sebuah karakter: P
Karakter yang Anda masukkan adalah: P
PS E:\KULIAH\SEMESTER 3\Struktur Data\Code> cd "e:\KULIAH\SEMESTER 3\Struktur Data\Code\" ; if ($?) { g++ Guided2.cpp -o Guided2 } ; if ($?) { .\Guided2 }
Masukkan sebuah karakter: n
Karakter yang Anda masukkan adalah: n
PS E:\KULIAH\SEMESTER 3\Struktur Data\Code>
```

A small window titled "NA" is overlaid on the code, displaying the output:

```
NAMA : MUHAMMAD FACHRI AURAVYANO SAKA
NIM : 103112430180
```

```

#include <iostream>

using namespace std;

int main() {

char ch;

cout << "Masukkan sebuah karakter: ";

ch = getchar(); // Menggunakan getchar() untuk membaca satu karakter

cout << "Karakter yang Anda masukkan adalah: " << ch << endl;

return 0;

}

```

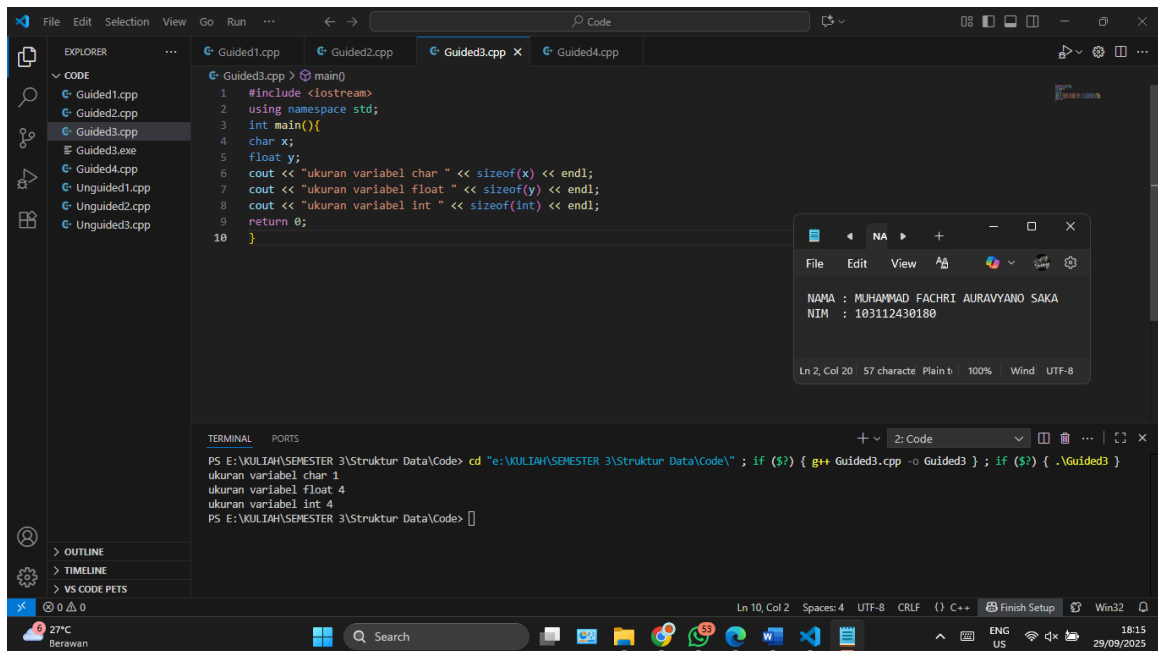
Screenshot Ouput

The screenshot shows the Visual Studio Code interface. The Explorer pane on the left lists files: Guided1.cpp, Guided2.cpp (selected), Guided2.exe, Guided3.cpp, Guided4.cpp, Unguided1.cpp, Unguided2.cpp, and Unguided3.cpp. The Code pane displays the C++ code from the previous block. The Output pane on the right shows the program's execution: it prompts for a character, reads 'P', and displays 'Karakter yang Anda masukkan adalah: P'. The Terminal pane at the bottom shows the command prompt session where the program is compiled and run, confirming the input and output.

Deskripsi

Program di atas merupakan contoh sederhana penggunaan bahasa C++ untuk membaca dan menampilkan sebuah karakter. Pertama, library `<iostream>` digunakan agar program dapat melakukan proses input dan output. Di dalam fungsi `main()`, dideklarasikan variabel `ch` dengan tipe data `char` untuk menampung satu karakter yang dimasukkan oleh pengguna. Program menampilkan pesan “Masukkan sebuah karakter: ” menggunakan `cout`, lalu menunggu input pengguna melalui fungsi `getchar()`. Setelah pengguna mengetikkan sebuah karakter, nilai tersebut disimpan ke variabel `ch` dan ditampilkan kembali ke layar dengan pernyataan `cout << "Karakter yang Anda masukkan adalah: " << ch << endl;`. Dengan demikian, program ini memungkinkan pengguna untuk memasukkan satu karakter saja, kemudian program akan menampilkannya kembali sebagai output.

Guided 3



```
#include <iostream>

using namespace std;

int main(){

char x;

float y;

cout << "ukuran variabel char " << sizeof(x) << endl;
cout << "ukuran variabel float " << sizeof(y) << endl;
cout << "ukuran variabel int " << sizeof(int) << endl;

return 0;

}
```

Screenshot Output

The screenshot shows the Visual Studio Code interface with the file `Guided3.cpp` open. The code defines a `main` function that declares a `char` variable `x` and a `float` variable `y`. It then uses `cout` to print the memory size of these variables using the `sizeof` operator. The output window shows the program's output, and the terminal shows the command to compile and run the program.

```
1 #include <iostream>
2 using namespace std;
3 int main(){
4     char x;
5     float y;
6     cout << "ukuran variabel char " << sizeof(x) << endl;
7     cout << "ukuran variabel float " << sizeof(y) << endl;
8     cout << "ukuran variabel int " << sizeof(int) << endl;
9     return 0;
10 }
```

Output:

```
NAMA : MUHAMMAD FACHRI AURAVYANO SAKA
NIM : 103112430180
```

Terminal:

```
PS E:\KULIAH\SEMESTER 3\Struktur Data\Code> cd "e:\KULIAH\SEMESTER 3\Struktur Data\Code\"; if ($?) { g++ Guided3.cpp -o Guided3 }; if ($?) { .\Guided3 }
ukuran variabel char 1
ukuran variabel float 4
ukuran variabel int 4
PS E:\KULIAH\SEMESTER 3\Struktur Data\Code>
```

Deskripsi

Program di atas adalah contoh penggunaan operator `sizeof` dalam bahasa C++ untuk mengetahui ukuran memori dari suatu tipe data. Di dalam fungsi `main()`, dideklarasikan dua variabel yaitu `x` dengan tipe `char` dan `y` dengan tipe `float`. Melalui perintah `cout`, program menampilkan ukuran variabel `char`, `float`, dan juga tipe data `int`. Operator `sizeof` akan mengembalikan nilai berupa jumlah byte yang digunakan oleh masing-masing tipe data di dalam memori komputer. Hasil output dapat berbeda tergantung pada sistem dan compiler yang digunakan, namun umumnya `char` berukuran 1 byte, `float` berukuran 4 byte, dan `int` bisa 4 byte atau lebih tergantung arsitektur sistem. Program ini bermanfaat untuk memahami bagaimana data disimpan dalam memori serta membantu dalam pengelolaan efisiensi program.

Guided 4

The screenshot shows the Visual Studio Code interface with the file `Guided4.cpp` open. The code defines a `main` function that declares two integer variables `r` and `s`, initializes `r` to 10, and then increments `s` by the value of `r`. It then uses `cout` to print the values of `r` and `s`. The output window shows the program's output, and the terminal shows the command to compile and run the program.

```
1 #include <iostream>
2 using namespace std;
3 int main(){
4     int r = 10;
5     int s;
6     s = 10 + ++r;
7     cout << "Nilai r = " << r << endl;
8     cout << "Nilai s = " << s << endl;
9     return 0;
10 }
```

Output:

```
NAMA : MUHAMMAD FACHRI AURAVYANO SAKA
NIM : 103112430180
```

Terminal:

```
PS E:\KULIAH\SEMESTER 3\Struktur Data\Code> cd "e:\KULIAH\SEMESTER 3\Struktur Data\Code\"; if ($?) { g++ Guided4.cpp -o Guided4 }; if ($?) { .\Guided4 }
Nilai r = 11
Nilai s = 21
PS E:\KULIAH\SEMESTER 3\Struktur Data\Code>
```

```

#include <iostream>

using namespace std;

int main(){

int r = 10;

int s;

s=10 + ++r;

cout<< "Nilai r= "<<r<<endl;

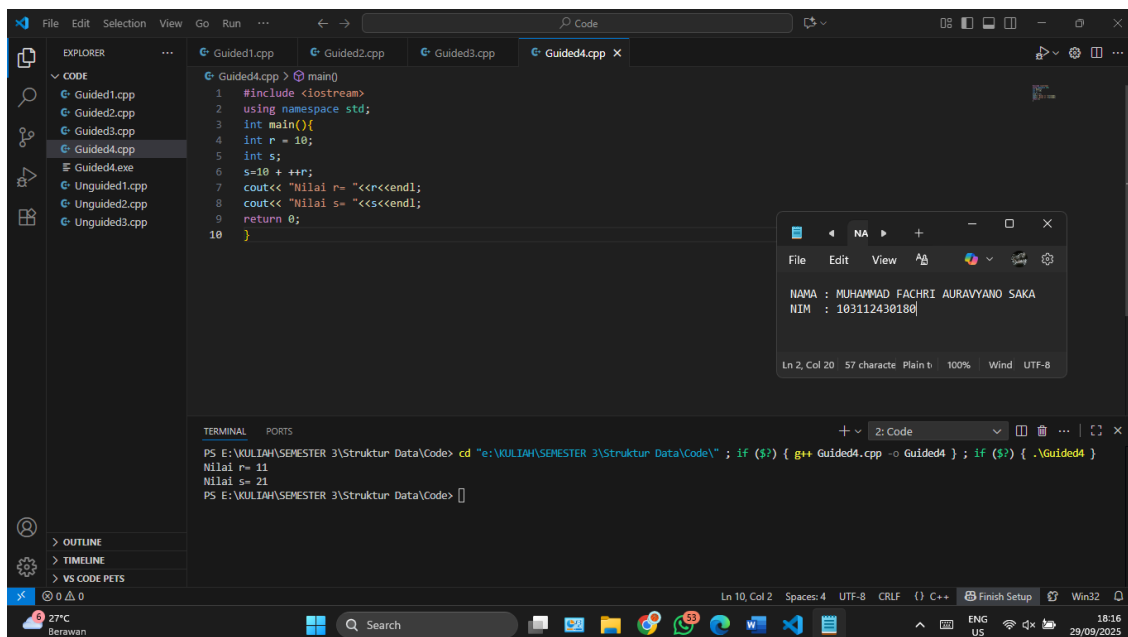
cout<< "Nilai s= "<<s<<endl;

return 0;

}

```

Screensshot Output

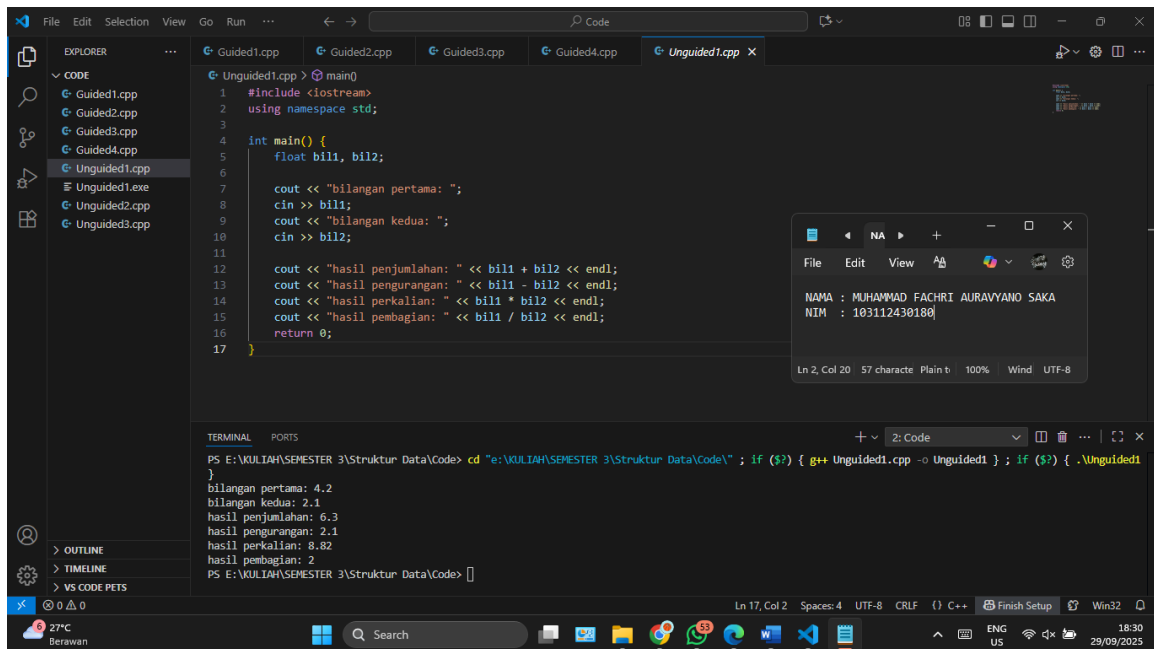


Deskripsi

Program di atas menunjukkan penggunaan operator increment (++) dalam C++. Pada awalnya, variabel r diberi nilai 10. Kemudian, variabel s diberi nilai hasil dari $10 + ++r$. Tanda $++r$ disebut pre-increment, artinya nilai r akan ditambahkan terlebih dahulu (dari 10 menjadi 11) sebelum digunakan dalam operasi penjumlahan. Maka, ekspresi $10 + ++r$ menghasilkan $10 + 11 = 21$. Setelah itu, program menampilkan nilai r dan s menggunakan cout. Output yang dihasilkan adalah Nilai r = 11 dan Nilai s = 21. Program ini bermanfaat untuk memahami perbedaan antara pre-increment (++) dan post-increment (r++) dalam operasi perhitungan di C++.

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

Unguided 1



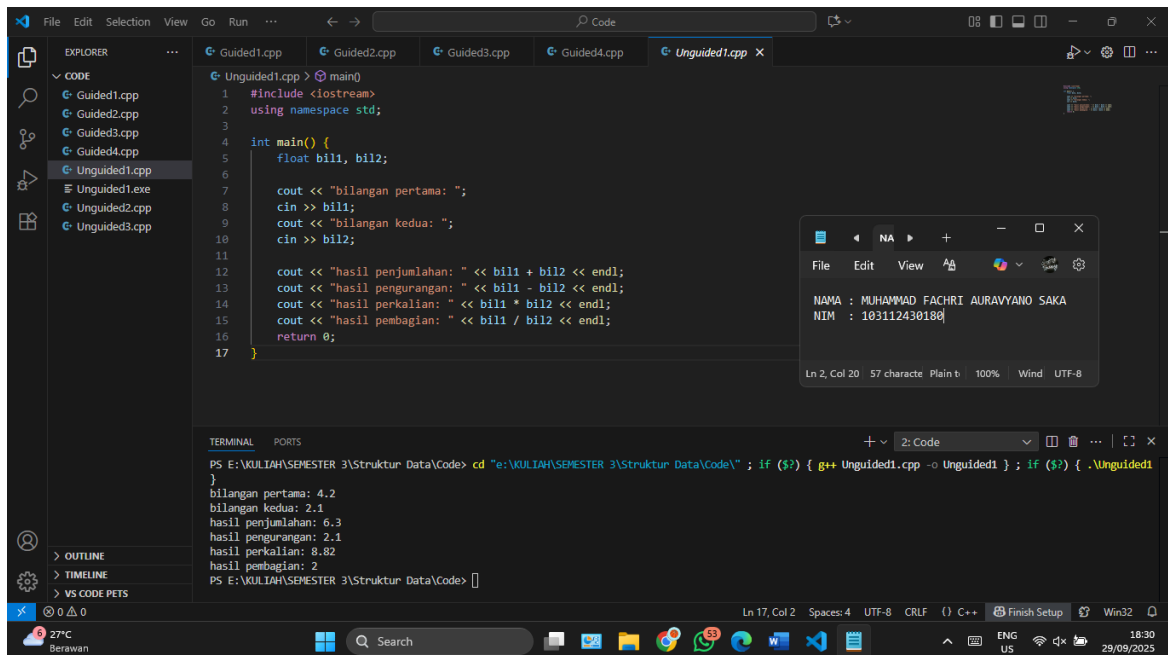
```
#include <iostream>
using namespace std;

int main() {
    float bil1, bil2;

    cout << "bilangan pertama: ";
    cin >> bil1;
    cout << "bilangan kedua: ";
    cin >> bil2;

    cout << "hasil penjumlahan: " << bil1 + bil2 << endl;
    cout << "hasil pengurangan: " << bil1 - bil2 << endl;
    cout << "hasil perkalian: " << bil1 * bil2 << endl;
    cout << "hasil pembagian: " << bil1 / bil2 << endl;
    return 0;
}
```

Screenshots Output



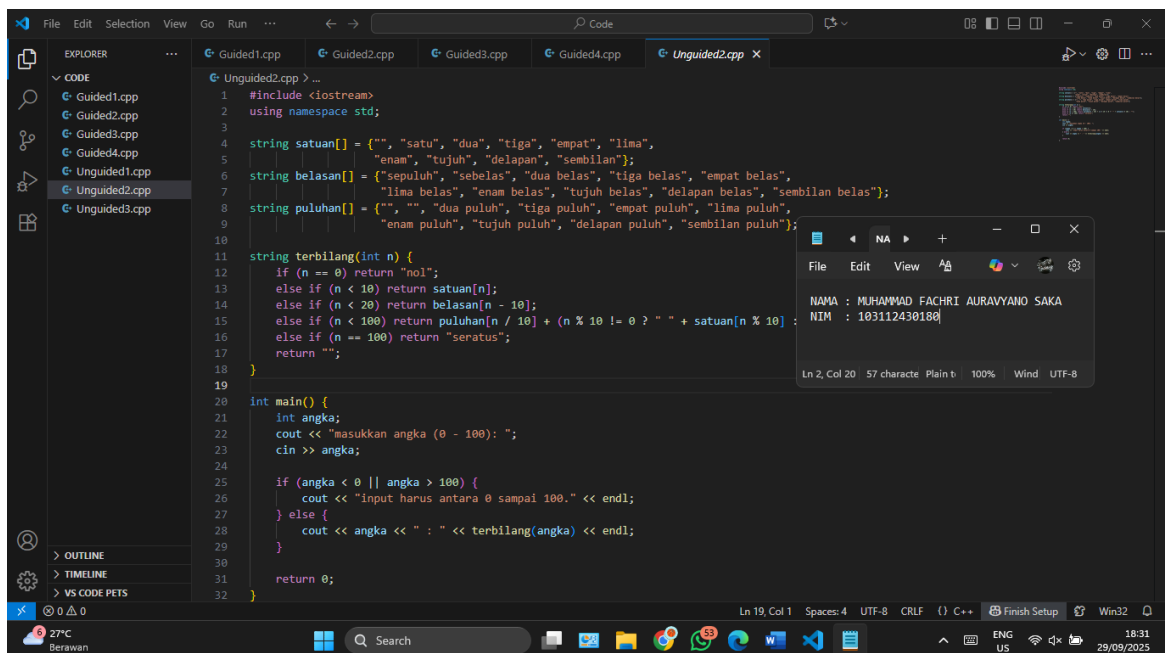
```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     float bil1, bil2;
6
7     cout << "bilangan pertama: ";
8     cin >> bil1;
9     cout << "bilangan kedua: ";
10    cin >> bil2;
11
12    cout << "hasil penjumlahan: " << bil1 + bil2 << endl;
13    cout << "hasil pengurangan: " << bil1 - bil2 << endl;
14    cout << "hasil perkalian: " << bil1 * bil2 << endl;
15    cout << "hasil pembagian: " << bil1 / bil2 << endl;
16    return 0;
17 }
```

```
PS E:\KULIAH\SEMESTER 3\Struktur Data\Code> cd "e:\KULIAH\SEMESTER 3\Struktur Data\Code\"; if ($?) { g++ Unguided1.cpp -o Unguided1 }; if ($?) { .\Unguided1 }
bilangan pertama: 4.2
bilangan kedua: 2.1
hasil penjumlahan: 6.3
hasil pengurangan: 2.1
hasil perkalian: 8.82
hasil pembagian: 2
PS E:\KULIAH\SEMESTER 3\Struktur Data\Code>
```

Deskripsi:

Program di atas adalah contoh sederhana operasi aritmetika dasar menggunakan bahasa C++. Pertama, dua variabel bertipe float bernama bil1 dan bil2 dideklarasikan untuk menyimpan input berupa bilangan desimal dari pengguna. Program kemudian meminta pengguna memasukkan nilai untuk kedua bilangan tersebut dengan menggunakan cin. Setelah nilai dimasukkan, program melakukan empat operasi aritmetika: penjumlahan ($bil1 + bil2$), pengurangan ($bil1 - bil2$), perkalian ($bil1 * bil2$), dan pembagian ($bil1 / bil2$). Hasil dari setiap operasi ditampilkan ke layar menggunakan cout. Dengan demikian, program ini berfungsi sebagai kalkulator sederhana yang mampu mengolah dua bilangan dengan berbagai operasi matematika dasar.

Unguided 2



```
1 #include <iostream>
2 using namespace std;
3
4 string satuan[] = {"", "satu", "dua", "tiga", "empat", "lima",
5                  "enam", "tujuh", "delapan", "sembilan"};
6 string belasan[] = {"sepuluh", "sebelas", "dua belas", "tiga belas", "empat belas",
7                  "lima belas", "enam belas", "tujuh belas", "delapan belas", "sembilan belas"};
8 string puluhan[] = {"", "", "dua puluh", "tiga puluh", "empat puluh", "lima puluh",
9                  "enam puluh", "tujuh puluh", "delapan puluh", "sembilan puluh"};
10
11 string terbilang(int n) {
12     if (n == 0) return "nol";
13     else if (n < 10) return satuan[n];
14     else if (n < 20) return belasan[n - 10];
15     else if (n < 100) return puluhan[n / 10] + (n % 10 != 0 ? " " + satuan[n % 10] : "");
16     else if (n == 100) return "seratus";
17     return "";
18 }
19
20 int main() {
21     int angka;
22     cout << "masukkan angka (0 - 100): ";
23     cin >> angka;
24
25     if (angka < 0 || angka > 100) {
26         cout << "Input harus antara 0 sampai 100." << endl;
27     } else {
28         cout << angka << " : " << terbilang(angka) << endl;
29     }
30
31     return 0;
32 }
```

```

#include <iostream>
using namespace std;

string satuan[] = {"", "satu", "dua", "tiga", "empat", "lima",
                  "enam", "tujuh", "delapan", "sembilan"};
string belasan[] = {"sepuluh", "sebelas", "dua belas", "tiga
belas", "empat belas",
                  "lima belas", "enam belas", "tujuh belas",
"delapan belas", "sembilan belas"};
string puluhan[] = {"", "", "dua puluh", "tiga puluh", "empat
puluh", "lima puluh",
                  "enam puluh", "tujuh puluh", "delapan
puluh", "sembilan puluh"};

string terbilang(int n) {
    if (n == 0) return "nol";
    else if (n < 10) return satuan[n];
    else if (n < 20) return belasan[n - 10];
    else if (n < 100) return puluhan[n / 10] + (n % 10 != 0 ? "
" + satuan[n % 10] : "");
    else if (n == 100) return "seratus";
    return "";
}

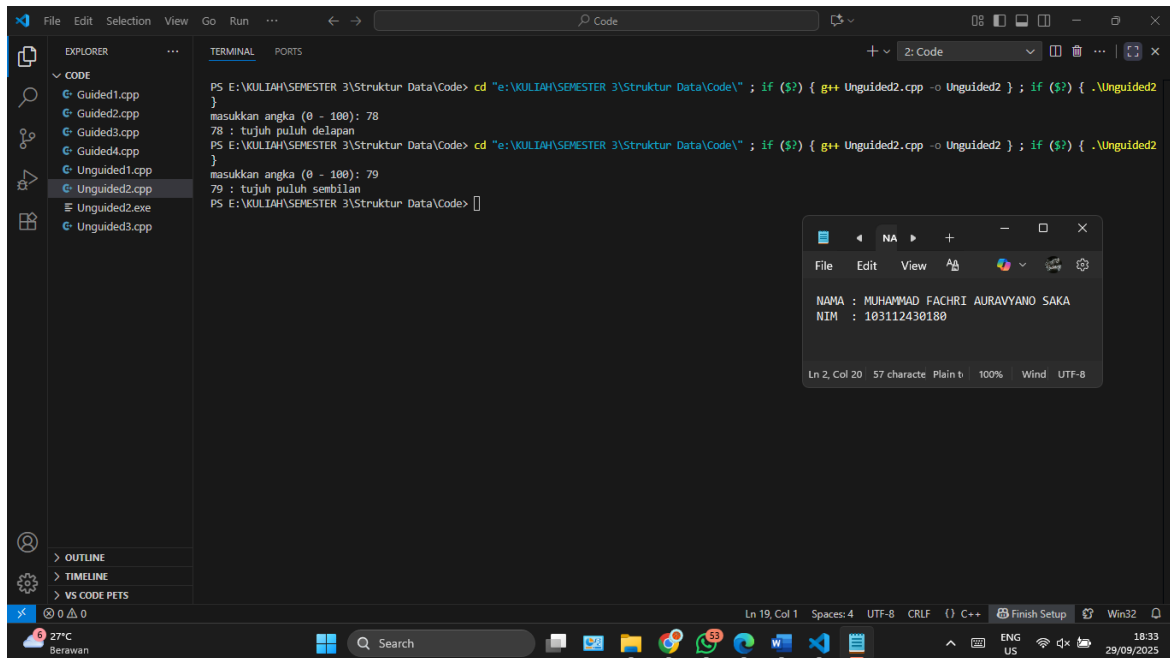
int main() {
    int angka;
    cout << "masukkan angka (0 - 100): ";
    cin >> angka;

    if (angka < 0 || angka > 100) {
        cout << "input harus antara 0 sampai 100." << endl;
    } else {
        cout << angka << " : " << terbilang(angka) << endl;
    }
}

```

```
return 0;
}
```

Screenshot Output



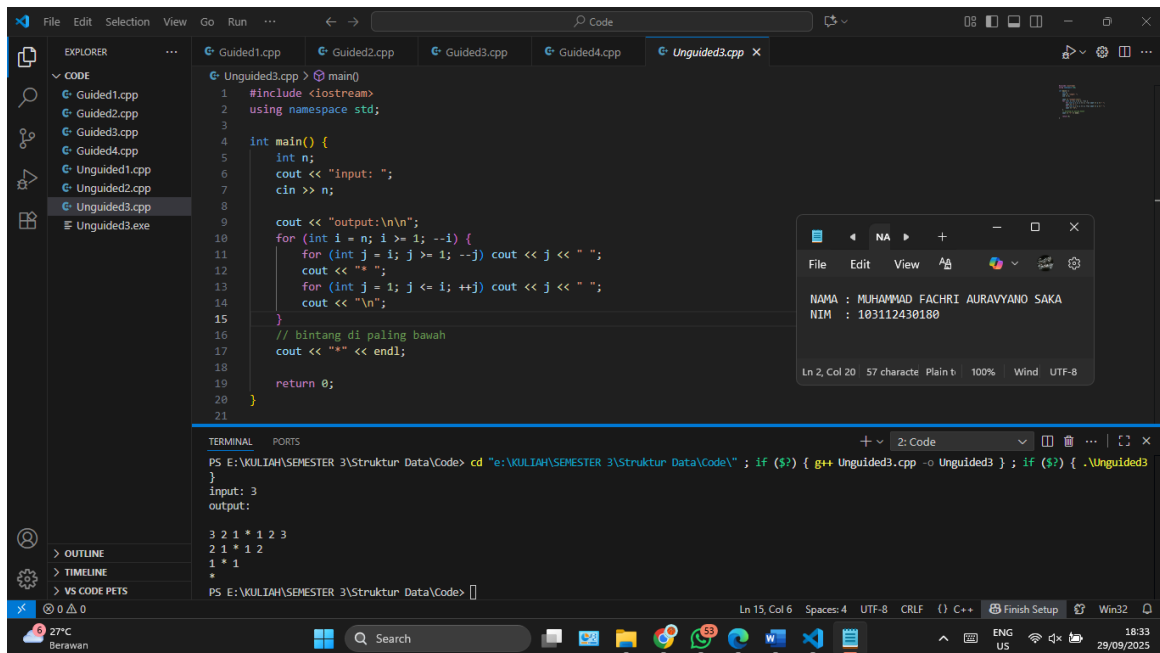
The screenshot shows the Visual Studio Code interface. The Explorer pane on the left lists files: Guided1.cpp, Guided2.cpp, Guided3.cpp, Guided4.cpp, Unguided1.cpp, Unguided2.cpp, Unguided2.exe, and Unguided3.cpp. The main editor displays the code for Unguided2.cpp, which includes a main function that prompts the user to enter a number between 0 and 100. The terminal shows the execution of the program, with the user inputting '78' and '79'. The output shows the program correctly identifying the tens and units digits and displaying them in Indonesian. A small window titled 'NA' is also visible, showing the user's name 'MUHAMMAD FACHRI AURAVYANO SAKA' and NIM '103112430180'.

```
PS E:\KULIAH\SEMESTER 3\Struktur Data\Code> cd "e:\KULIAH\SEMESTER 3\Struktur Data\Code\" ; if ($?) { g++ Unguided2.cpp -o Unguided2 } ; if ($?) { .\Unguided2
}
masukkan angka (0 - 100): 78
78 : tujuh puluh delapan
PS E:\KULIAH\SEMESTER 3\Struktur Data\Code> cd "e:\KULIAH\SEMESTER 3\Struktur Data\Code\" ; if ($?) { g++ Unguided2.cpp -o Unguided2 } ; if ($?) { .\Unguided2
}
masukkan angka (0 - 100): 79
79 : tujuh puluh sembilan
PS E:\KULIAH\SEMESTER 3\Struktur Data\Code>
```

Deksripsi

Program di atas digunakan untuk mengubah angka menjadi bentuk tulisan atau terbilang dalam bahasa Indonesia. Terdapat tiga array string, yaitu satuan, belasan, dan puluhan, yang menyimpan kata-kata dasar untuk angka-angka tertentu. Fungsi terbilang(int n) berperan untuk memproses input angka dengan beberapa kondisi: jika nilai nol maka hasilnya “nol”, angka kurang dari sepuluh diambil dari array satuan, angka 10 hingga 19 diambil dari array belasan, angka puluhan dibentuk dari puluhan ditambah satuan jika ada sisa, dan angka seratus langsung ditampilkan sebagai “seratus”. Pada bagian main(), pengguna diminta memasukkan angka antara 0 hingga 100. Jika input berada di luar rentang, program akan menampilkan pesan error, sedangkan jika valid maka angka ditampilkan bersama hasil terbilangannya. Program ini menunjukkan penerapan kondisi if-else, array string, serta fungsi dalam pemrograman C++.

Unguided 3



```
#include <iostream>

using namespace std;

int main() {
    int n;

    cout << "input: ";
    cin >> n;

    cout << "output:\n\n";
    for (int i = n; i >= 1; --i) {
        for (int j = i; j >= 1; --j) cout << j << " ";
        cout << "* ";
        for (int j = 1; j <= i; ++j) cout << j << " ";
        cout << "\n";
    }

    // bintang di paling bawah
    cout << "*" << endl;

    return 0;
}
```

Screenshot Output

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n;
6     cout << "input: ";
7     cin >> n;
8
9     cout << "output:\n\n";
10    for (int i = n; i >= 1; --i) {
11        for (int j = 1; j >= i; ++j) cout << j << " ";
12        cout << "\n";
13        for (int j = i; j >= 1; --j) cout << j << " ";
14        cout << "\n";
15    }
16    // bintang di paling bawah
17    cout << "*" << endl;
18
19    return 0;
20 }
```

File Edit View A+ 100% Win32

NAMA : MUHAMMAD FACHRI AURAVYAND SAKA
NIM : 103112430180

Ln 2, Col 20 57 character Plain text 100% Win UTF-8

TERMINAL

PS E:\VULIAH\SEMESTER 3\Struktur Data\Code> cd "e:\VULIAH\SEMESTER 3\Struktur Data\Code\" ; if (\$?) { g++ Unguided3.cpp -o Unguided3 } ; if (\$?) { .\Unguided3 }
input: 3
output:
3 2 1 * 1 2 3
2 1 * 1 2
1 * 1
*
PS E:\VULIAH\SEMESTER 3\Struktur Data\Code>

Dekripsi

Program di atas merupakan contoh penggunaan perulangan bersarang (nested loop) dalam bahasa C++. Pertama, program meminta input berupa sebuah bilangan bulat n dari pengguna. Setelah itu, program menampilkan pola angka menurun dari n ke 1 diikuti simbol bintang (*), lalu angka menaik dari 1 kembali ke i . Proses ini dilakukan berulang kali hingga nilai i mencapai 1. Bagian akhir program menambahkan satu baris berisi hanya tanda bintang di paling bawah sebagai penutup pola.

D. Kesimpulan

Dari praktikum ini dapat disimpulkan bahwa penggunaan Code::Blocks sebagai IDE sangat membantu dalam proses menulis, mengompilasi, dan menjalankan program C++ dengan lebih mudah. Melalui percobaan program yang telah dilakukan, dipahami konsep dasar bahasa C++ seperti struktur program dengan fungsi `main()`, penggunaan library `<iostream>`, variabel dan tipe data, operator aritmetika, perulangan, hingga penerapan fungsi. Selain itu, praktikum juga memberikan pemahaman mengenai cara memproses input dan output, penggunaan kondisi if-else, serta implementasi array string dalam menyelesaikan masalah. Dengan dasar ini, mahasiswa memiliki pondasi yang kuat untuk mempelajari konsep pemrograman C++ yang lebih kompleks pada tahap selanjutnya.

E. Referensi

Al Fatta, H. (2006). *Dasar pemrograman C++ disertai dengan pengenalan pemrograman berorientasi objek*. Yogyakarta: Penerbit Andi. ISBN 979-763-582-1

Naugler, D. (2007). C# 2.0 for C++ and Java programmer: Conference workshop. *Journal of Computing Sciences in Colleges*, 22(5).