

LAPORAN PRAKTIKUM
STRUKTUR DATA
"Pengenal C++: Subprogram & Array"



Oleh:

NAMA: Ammar Dzaki Nandana

NIM: 2311104071

KELAS: SE 07 B

DOSEN: Wahyu Andi Saputra

PRODI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2023/2024

I. TUJUAN

- Tujuan dari penginstalan **Code::Blocks** adalah untuk menyediakan lingkungan pengembangan terpadu bagi pemrograman menggunakan bahasa C dan C++. Dengan Code::Blocks, pengguna dapat menulis, mengedit, mengompilasi, serta menjalankan program dengan lebih efisien. Selain itu, penginstalan ini bertujuan untuk mempermudah proses debugging, di mana pengguna dapat menemukan dan memperbaiki kesalahan (bug) dalam program secara interaktif, sehingga dapat meningkatkan produktivitas pengembangan perangkat lunak. Melalui instalasi ini, kompiler **MinGW** juga terpasang sebagai bagian dari paket instalasi Code::Blocks pada sistem Windows, yang memungkinkan pengguna untuk langsung mengompilasi kode C dan C++ tanpa perlu menginstal kompiler secara terpisah. Code::Blocks juga menawarkan antarmuka yang ramah pengguna, sehingga baik pemula maupun profesional dapat menggunakan IDE ini dengan mudah dalam proses belajar maupun pengembangan aplikasi. Akhirnya, tujuan penginstalan ini juga mencakup konfigurasi dan menjalankan program pertama sebagai langkah verifikasi untuk memastikan bahwa IDE dan kompiler berfungsi dengan baik setelah instalasi selesai.

II. LANDASAN TEORI

1. Integrated Development Environment (IDE)

Integrated Development Environment (IDE) adalah perangkat lunak yang menyediakan fasilitas lengkap untuk pemrograman, seperti editor kode sumber, alat kompilasi, dan debugger. IDE bertujuan untuk mempercepat dan mempermudah pengembangan perangkat lunak dengan mengintegrasikan berbagai alat dalam satu aplikasi. Code::Blocks adalah salah satu IDE yang dirancang khusus untuk bahasa pemrograman C dan C++, meskipun dapat digunakan untuk bahasa lain dengan menggunakan plugin. Dalam IDE ini, pengguna dapat menulis kode, mengelola proyek, melakukan kompilasi, serta menjalankan dan melakukan debugging program secara interaktif.

2. Bahasa Pemrograman C dan C++

Bahasa pemrograman C dan C++ adalah bahasa pemrograman tingkat menengah yang banyak digunakan untuk pengembangan perangkat lunak sistem, aplikasi, game, hingga embedded systems. C merupakan bahasa yang sangat efisien dan telah menjadi dasar untuk banyak bahasa modern, sementara C++ adalah pengembangan dari C yang mendukung pemrograman berorientasi objek. Karena kepopulerannya, C dan C++ memerlukan kompiler yang mampu menerjemahkan kode sumber ke dalam kode mesin yang dapat dieksekusi. IDE seperti

Code::Blocks menyediakan integrasi dengan berbagai kompiler, seperti GCC dan MinGW, untuk memfasilitasi proses ini.

3. Kompiler MinGW

MinGW (Minimalist GNU for Windows) adalah sebuah port dari GCC (GNU Compiler Collection) yang ditujukan untuk Windows. MinGW menyediakan kompiler yang mampu mengompilasi program C dan C++ agar bisa berjalan di lingkungan Windows. Pada instalasi Code::Blocks, MinGW biasanya disertakan secara default sehingga pengguna dapat langsung menggunakannya untuk kompilasi tanpa harus melakukan konfigurasi tambahan. Kompiler berfungsi untuk menerjemahkan kode sumber yang ditulis dalam bahasa tingkat tinggi (seperti C atau C++) ke dalam bahasa mesin agar bisa dieksekusi oleh komputer.

4. Proses Debugging

Debugging adalah proses identifikasi, analisis, dan penghapusan kesalahan atau bug dalam kode program. Debugging merupakan bagian penting dalam siklus pengembangan perangkat lunak karena bug dapat menyebabkan program tidak berfungsi sebagaimana mestinya. Code::Blocks memiliki fitur debugger bawaan yang memanfaatkan GDB (GNU Debugger). Dengan debugger ini, pengembang dapat melakukan breakpoints, melihat nilai variabel, melacak alur eksekusi program, dan mendeteksi kesalahan secara lebih mudah. Ini membantu pengembang memahami bagaimana program berjalan dan di mana kesalahan terjadi.

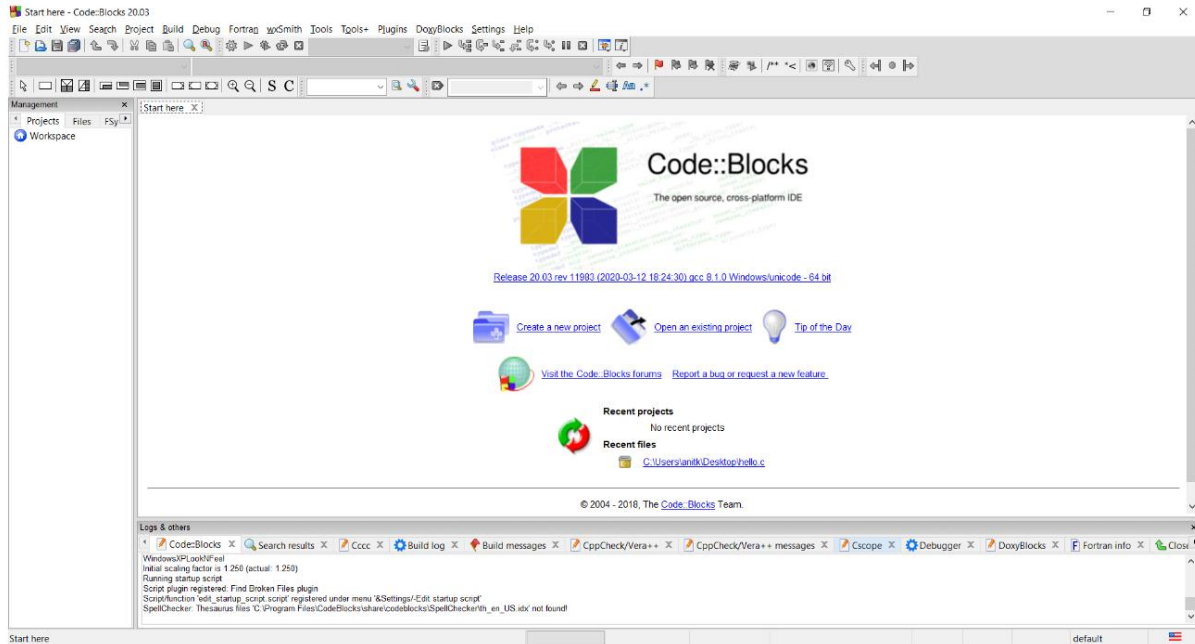
5. Kompilasi dan Eksekusi Program

Setelah menulis program, langkah penting berikutnya adalah kompilasi. Kompilasi adalah proses di mana kode sumber diterjemahkan ke dalam bahasa mesin yang dapat dijalankan oleh komputer. Dalam Code::Blocks, ketika pengguna menekan tombol “Build and Run”, IDE secara otomatis memanggil kompiler (misalnya, GCC atau MinGW) untuk mengubah kode sumber menjadi file executable. Jika ada kesalahan sintaks dalam program, kompiler akan mengeluarkan pesan error dan proses kompilasi gagal. Setelah program berhasil dikompilasi, hasil akhirnya adalah program executable yang dapat dijalankan langsung dari IDE.

III. GUIDED

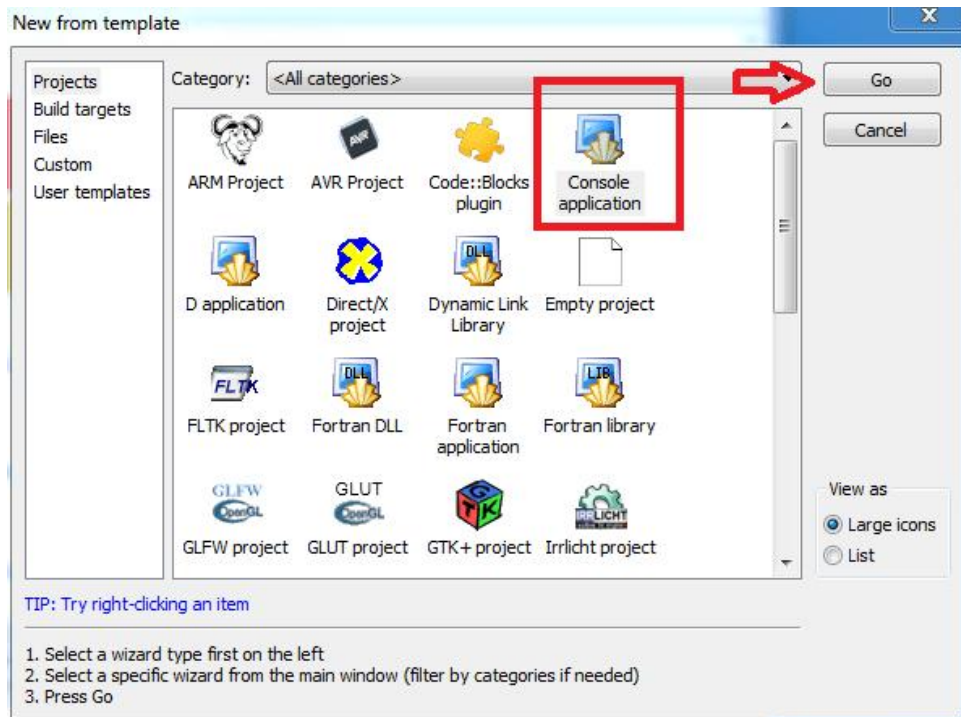
Instalasi Code Blocks Adapun cara menginstall Code Blocks adalah sebagai berikut.

1. Download terlebih dahulu file exe pada <http://www.codeblocks.org/downloads>. Pilih Download the binary release kemudian pilih file yang menggunakan mingw-setup (e.g. codeblocks-20.03mingw-setup.exe).
2. Setelah itu install file tersebut, akan muncul tampilan seperti pada Error! Reference source not found.



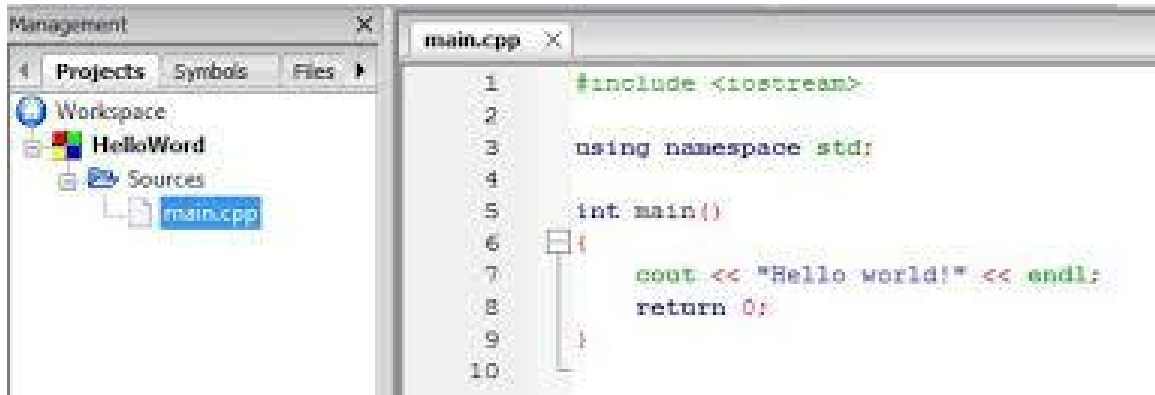
Cara Menggunakan Code Blocks Adapun cara menggunakan Code Blocks adalah sebagai berikut.

1. Membuat Project Baru dengan cara memilih File > New > Projects. Kemudian pada panel kiri pilih Project, pada panel kanan pilih Console application kemudian klik Go seperti pada Error! Reference source not found.

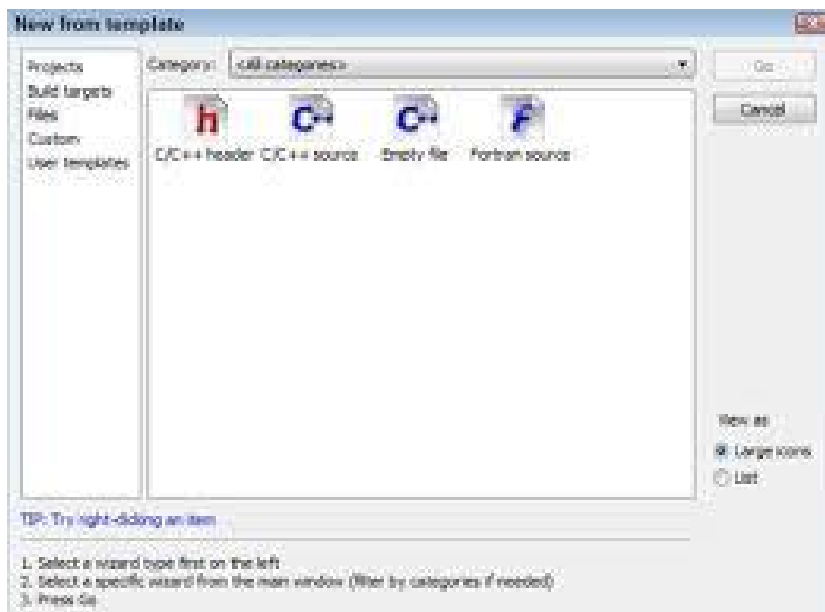


Setelah itu klik Next > Pilih Bahasa Pemrograman yang akan digunakan > Isi Project title dan Folder to create project in (tempat menyimpan project) > Klik Finish.

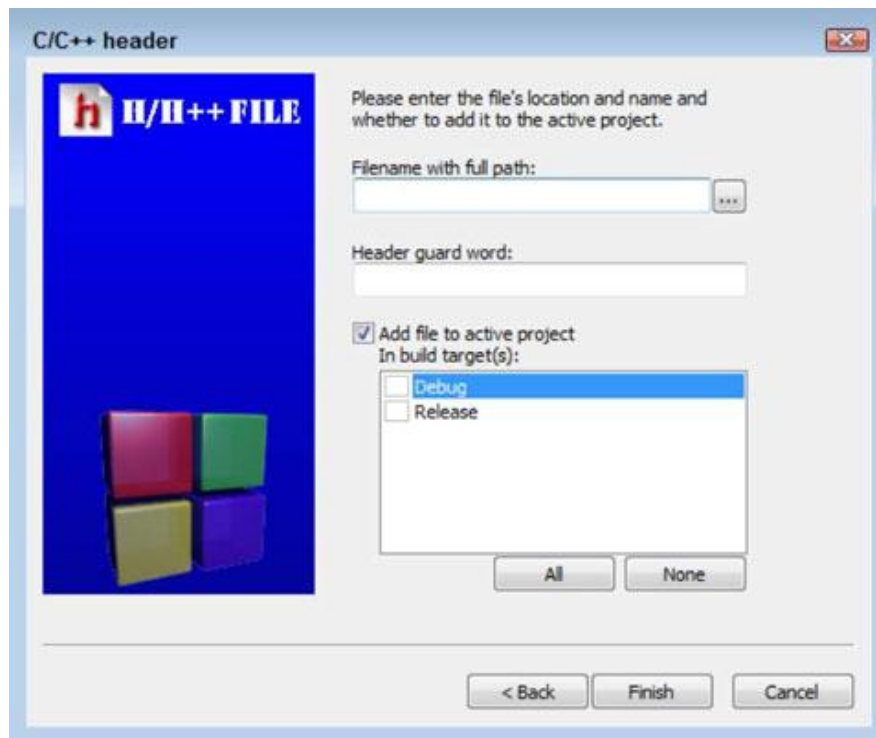
2. Menulis Sintak pada editor seperti pada Error! Reference source not found.



3. Membuat Class Baru dengan cara klik File > New > File. Pada panel kiri pilih Files, dan pada panel kanan pilih C/C++ source Kemudian Klik Go seperti pada Error! Reference source not found.



Kemudian klik Next > Pilih bahasa pemrograman > Isi Filename with full path > Centang all in build target > Finish, seperti pada Error! Reference source not found.



Jika anda lupa mencentang build target, dapat dilakukan setting manual dengan cara klik kanan pada project > properties > Build targets > Debug > Centang semua target files seperti pada Error! Reference source not found.

```

1 #include <iostream>
2 #include <conio.h>
3
4 using namespace std;
5
6 int main()
7 {
8     /*int angka = 10;
9     float desimal = 10.5;
10    string kalimat = "ammar";
11    double tingpi = 10.4;
12    char jenis kelamin = 'L';
13    bool isSunny = true;
14
15    cout << "Angka: " << angka << endl;
16
17    */
18
19    /*int angka;
20
21    cout << "Masukan angka: ";
22    cin >> angka;
23
24    cout << "Angka: " << angka << endl;
25
26    getch();
27    */
28
29    //operator aritmatika
30    /*int angka1 = 10;
31    int angka2 = 2;
32    int hasil = angka1 + angka2;
33
34    cout << "Hasilnya adalah " << hasil << endl;
35    */
36
37    //operator perbandingan
38
39    /*int angka1 = 5;
40    int angka2 = 5;
41
42    bool hasil = (angka1 == angka2);
43
44    cout << "hasilnya adalah " << boolalpha << hasil << endl;
45    */
46
47    //operator logika
48
49    /*bool kondisi1 = true;
50    bool kondisi2 = true;
51
52    bool hasil = (kondisi1 || kondisi2);
53    cout << "Hasilnya adalah: " << boolalpha << hasil << endl;
54    */
55
56    /*bool kondisi1 = false;
57    bool hasil = !kondisi1;
58    cout << "Hasilnya adalah: " << boolalpha << hasil << endl;
59    */
60
61    //percabangan
62
63    /*string Kata;
64    cout << "Masukan kata= HALO" << endl;
65    cin >> Kata;
66
67    if(Kata == "HALO") {
68        cout << "Kata sesuai" << endl;
69
70    } else {
71        cout << "Kata tidak sesuai" << endl;
72    }
73    */
74
75    /*int tv;
76    cout << "Daftar Channel tv" << endl;
77    cout << "1. RCTI" << endl;
78    cout << "2. INDOSIAR" << endl;
79
80    cout << "Masukan channel pilihan: ";
81    cin >> tv;
82
83    switch(tv) {
84    case 1 :
85        cout << "Channel yang anda pilih RCTI" << endl;
86        break;
87    case 2 :
88        cout << "Channel yang anda pilih INDOSIAR" << endl;
89        break;
90    default:
91        cout << "Channel tidak tersedia" << endl;
92    }
93    */
94
95    //perulangan
96
97    int i;
98
99
100
101 }

```

IV. UNGUIDED

NO. 1

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      float bil1, bil2;
7
8      // Input dua buah bilangan dari user
9      cout << "Masukkan bilangan pertama: ";
10     cin >> bil1;
11     cout << "Masukkan bilangan kedua: ";
12     cin >> bil2;
13
14     // Tampilkan hasil penjumlahan
15     cout << "Hasil penjumlahan: " << bil1 + bil2 << endl;
16
17     // Tampilkan hasil pengurangan
18     cout << "Hasil pengurangan: " << bil1 - bil2 << endl;
19
20     // Tampilkan hasil perkalian
21     cout << "Hasil perkalian: " << bil1 * bil2 << endl;
22
23     // Tampilkan hasil pembagian
24     // Pengecekan untuk menghindari pembagian dengan nol
25     if (bil2 != 0) {
26         cout << "Hasil pembagian: " << bil1 / bil2 << endl;
27     } else {
28         cout << "Pembagian tidak bisa dilakukan karena bilangan kedua adalah nol." << endl;
29     }
30
31     return 0;
32 }
33
34
35 //2311104071_AMMAR DZAKI NANDANA
36
```


NO. 2

```
1 #include <iostream>
2 #include <string>
3
4 using namespace std;
5
6 // Fungsi untuk mengubah angka menjadi teks
7 string angkaKeTeks(int angka)
8 {
9     string satuan[] = {"", "satu", "dua", "tiga", "empat", "lima", "enam", "tujuh", "delapan", "sembilan"};
10    string belasan[] = {"sepuluh", "sebelas", "dua belas", "tiga belas", "empat belas", "lima belas", "enam belas", "tujuh belas", "delapan belas", "sembilan belas"};
11    string puluhan[] = {"", "", "dua puluh", "tiga puluh", "empat puluh", "lima puluh", "enam puluh", "tujuh puluh", "delapan puluh", "sembilan puluh"};
12
13    // Jika angka adalah 0
14    if (angka == 0)
15    {
16        return "nol";
17    }
18
19    // Jika angka adalah 100
20    if (angka == 100)
21    {
22        return "seratus";
23    }
24
25    // Jika angka di bawah 10
26    if (angka < 10)
27    {
28        return satuan[angka];
29    }
30
31    // Jika angka di antara 10 dan 19 (belasan)
32    if (angka >= 10 && angka < 20)
33    {
34        return belasan[angka - 10];
35    }
36
37    // Jika angka di antara 20 dan 99 (puluhan)
38    if (angka >= 20 && angka < 100)
39    {
40        return puluhan[angka / 10] + (angka % 10 != 0 ? " " + satuan[angka % 10] : "");
41    }
42
43    return "";
44 }
45
46 int main()
47 {
48     int angka;
49
50     // Input dari pengguna
51     cout << "Masukkan angka antara 0 sampai 100: ";
52     cin >> angka;
53
54     // Memastikan input valid
55     if (angka < 0 || angka > 100)
56     {
57         cout << "Masukkan angka yang valid (0 sampai 100)." << endl;
58     }
59     else
60     {
61         // Output hasil dalam bentuk teks
62         cout << angka << ": " << angkaKeTeks(angka) << endl;
63     }
64
65     return 0;
66 }
67 //2311104071_AMMAR DZAKI NANDANA
```

NO. 3

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7
8     // Input dari pengguna
9     cout << "Masukkan angka: ";
10    cin >> n;
11
12    // Loop untuk mencetak pola
13    for (int i = n; i >= 1; i--) {
14        // Bagian kiri: menurun dari i hingga 1
15        for (int j = i; j >= 1; j--) {
16            cout << j << " ";
17        }
18
19        // Cetak tanda bintang *
20        cout << " ";
21
22        // Bagian kanan: menaik dari 1 hingga i
23        for (int j = 1; j <= i; j++) {
24            cout << " " << j;
25        }
26
27        // Baris baru untuk pola berikutnya
28        cout << endl;
29    }
30
31    // Output terakhir hanya tanda *
32    cout << "*" << endl;
33
34    return 0;
35 }
36 #include <iostream>
37 #include <string>
38
39 using namespace std;
40
41 // Fungsi untuk mengubah angka menjadi teks
42 string angkaKeTeks(int angka)
43 {
44     string satuan[] = {"", "satu", "dua", "tiga", "empat", "lima", "enam", "tujuh", "delapan", "sembilan"};
45     string belasan[] = {"sepuluh", "sebelas", "dua belas", "tiga belas", "empat belas", "lima belas", "enam belas", "tujuh belas", "delapan belas", "sembilan belas"};
46     string puluhan[] = {"", "", "dua puluh", "tiga puluh", "empat puluh", "lima puluh", "enam puluh", "tujuh puluh", "delapan puluh", "sembilan puluh"};
47
48     // Jika angka adalah 0
49     if (angka == 0)
50     {
51         return "nol";
52     }
53
54     // Jika angka adalah 100
55     if (angka == 100)
56     {
57         return "seratus";
58     }
59
60     // Jika angka di bawah 10
61     if (angka < 10)
62     {
63         return satuan[angka];
64     }
65
66     // Jika angka di antara 10 dan 19 (belasan)
67     if (angka >= 10 && angka < 20)
68     {
69         return belasan[angka - 10];
70     }
71
72     // Jika angka di antara 20 dan 99 (puluhan)
73     if (angka >= 20 && angka < 100)
74     {
75         return puluhan[angka / 10] + (angka % 10 != 0 ? " " + satuan[angka % 10] : "");
76     }
77
78     return "";
79 }
80
81 int main()
82 {
83     int angka;
84
85     // Input dari pengguna
86     cout << "Masukkan angka antara 0 sampai 100: ";
87     cin >> angka;
88
89     // Memastikan input valid
90     if (angka < 0 || angka > 100)
91     {
92         cout << "Masukkan angka yang valid (0 sampai 100)." << endl;
93     }
94     else
95     {
96         // Output hasil dalam bentuk teks
97         cout << angka << " : " << angkaKeTeks(angka) << endl;
98     }
99
100    return 0;
101 }
102 //2311104071_AMMAR DZAKI NANDANA
```

V. KESIMPULAN

Penginstalan Code::Blocks beserta kompiler MinGW (untuk pengguna Windows) telah berhasil dilakukan. IDE ini siap digunakan untuk pengembangan program dalam bahasa C atau C++. Dengan menggunakan Code::Blocks, proses pengembangan menjadi lebih mudah karena tersedianya fitur debugging dan kompilasi otomatis.