**MAKALAH PENGENALAN**

**BAHASA C**



Oleh :

**Diki Candra**

**NIM 2022903430010**

**D4 TEKNOLOGY REKAYASA KOMPUTER JARINGAN**

**POLITEKNIK NEGERI LHOKSEUMAWE**

**2022/2023**

**Daftar Isi**

Bab 1 ...…….……………………………………………………….… 2

Pendahuluan ……………………………………………………….…. 2

1. 1 Latar Belankang …………………………………………….. 2
2. 2 Rumusan Masalah …………………………………………… 3
3. 3 Manfaat ..…………………………………………………….. 4
4. 4 Tujuan ……………………………………………………….. 5

Bab 2 …………………………………………………………………. 4

Landasan Teori ……………………………………………………..... 4

1. 5 Memahami Bahasa Pemrograman C ……………………….. 4
2. 6 Sejarah Bahasa C ………………………....………………… 4
3. 7 Perkembangan Bahasa C …………………………………… 6
4. 8 Jenis Variabel ………………………………………………. 6
5. 9 Struktur Program Bahasa C ………………………………… 6
6. 0 Dokumentasi Program ……………………………………… 7

2. 1 Pengarah Prapenggolahan …………………………………... 7

2. 2 Deklarasi Global ……………………………………………. 8

2. 3 Fungsi Main () ……………………………………………… 8

Kelebihan Dan Kekurangan …………………………………………. 10

Bab 3 ………………………………………………………………… 11

Jenis Data ……………………………………………………………. 11

2. 4 Tipe Data ……………………………………………………. 11

2. 5 Konstanta …………………………………………………… 13

2. 6 Variabel …………………………………………………….. 15

2. 7 Operator ……………………………………………………. 20

2. 8 Pengkondisian Bahasa Pemrograman C …………………… 23

2. 9 Ekspresi ……………………………………………………. 27

3. 0 Operasi Input Dan Output …………………………………. 28

3. 1 Pengulangan ……………………………………………….. 31

Bab 4 ……………………………………………………………….. 34

Contoh Program ……………………………………………………. 34

3. 2 Program Pertama …………………………………………... 34

3. 3 Program Kedua ..…………………………………………... 35

3. 4 Program Ketiga ……………………………………………. 36

3. 5 Program Keempat …………………………………………. 37

3. 6 Program Kelima …………………………………………… 38

3. 7 Program Keenam ………………………………………….. 39

3. 8 Program Ketujuh ………………………………………….. 40

Bab 5 ………………………………………………………………. 35

Penutup ……………………………………………………………. 35

SELESAI

**BAB I**

**PENDAHULUAN**

**1.1 Latar Belakang**

Bahasa C oleh Dennis M. Ritchie pada tahun 1972 di laboratorium Bell ditemukan.Pengembangan BPCL (Basic Combined Program Language) yang dibuat oleh Dr. Martin Richard yang selanjutnya dikembangkan oleh Ken Thompson dan diseleksi dengan Language B.

Dari ketertarikan Dennis pada penerjemah bahasa B, kemudian dikembangkan menjadi sebuah kompiler yang disebut C. Bahasa C peka huruf besar-kecil, yang sangat memperhatikan dan membedakan penggunaan huruf besar dan kecil.[1]

Meskipun C diciptakan untuk memprogram sistem dan jaringan komputer, bahasa ini juga sering digunakan dalam mengembangkan aplikasi perangkat lunak. C juga banyak digunakan oleh berbagai jenis platform sistem operasi dan arsitektur komputer, bahkan tersedia beberapa compiler yang sangat populer. C sangat mempengaruhi bahasa populer lainnya, terutama C++ yang merupakan perpanjangan dari C.

Sejak tahun 2000, C secara konsisten menempati peringkat di antara dua bahasa teratas dalam indeks TIOBE, ukuran popularitas bahasa pemrograman.

Dalam perkembangannya, muncul versi C lain yang akhirnya menimbulkan kebingungan di kalangan programmer. Oleh karena itu, pada tahun 1983, American National Standards Institute (ANSI) membentuk sebuah komite untuk membuat versi standar bahasa C. Setelah melalui proses yang panjang dan mengerikan, pada tahun 1989 berhasil diratifikasi yang menghasilkan ANSI X3.159-1989, versi ini sering ANSI C, atau terkadang C89.

Pada tahun 1990, versi ANSI C diadopsi oleh Organization for Standardization (ISO) dengan sedikit perubahan dengan nama ISO/IEC 9899:1990. Versi ini sering dilakukan ISO C atau C90. Karena versi ANSI C dan ISO C hanya memiliki sedikit perbedaan, panggilan C90 dan C89 merujuk ke bahasa yang sama] Versi C99 dibuat oleh ISO C pada tahun 1999. Versi ini terutama ditujukan untuk meningkatkan dukungan terfokus, terutama setelah C++, yang didasarkan pada tempat khusus di antara pemrogram yang banyak diminati.

**1.2 Rumusan masalah**

Berdasarkan latar belakang tersebut, penulis merumuskan masalah yaitu Bagaimana mengidentifikasi bahasa pemrograman bahasa C?

**1.3** **Manfaat**

Manfaat yang diperoleh dari makalah tersebut adalah sebagai berikut:

1. Pembaca
2. Pembaca
3. Menjadikan pembaca makalah semata-mata agar pembaca memahami dasar-dasar pemrograman.
4. Untuk memotivasi pembaca untuk membuat makalah yang lebih baik dari Ini.
5. Pengaramg
6. Untuk memberikan semangat pada mata kuliah yang sedang diambil saat ini
7. Sebagai titik awal untuk melakukan penelitian agar penelitian selanjutnya dapat dilakukan dengan lebih baik lagi.

**1.4 Tujuan**

1. Mengetahui Bahasa C
2. Mengetahui tipe data variabel
3. Dapat menerapkan Bahasa

**BAB II**

**LANDASAN TEORI**

* 1. **Memahami Bahasa Pemrograman C**

Bahasa pemrograman C merupakan bahasa Intermediate yang artinya bahasa tersebut dapat dikatakan sebagai bahasa tingkat tinggi atau high level language, dimana programmer diberikan serangkaian aturan penulisan sintaks yang dapat dipahami oleh manusia. Namun bahasa C juga dapat digolongkan sebagai bahasa tingkat rendah karena bahasa C juga menyediakan sintaks berupa bahasa rakitan dimana bahasa rakitan merupakan salah satu bahasa tingkat rendah.

* 1. **Sejarah Bahasa C**

C adalah bahasa pemrograman. Bahasa C dirancang oleh Dennis M. Ritchie pada tahun 1972 di AT&T Bell Labs. Bahasa C dikembangkan dari bahasa BPCL (Basic Combined Programming Language) dan bahasa B. Bahasa BPCL dikembangkan oleh Martin Richard pada tahun 1967 sebagai bahasa sistem operasi dan compiler. Ken Thompson pada tahun 1970 telah merancang bahasa B dengan memasukkan fitur BPCL. Bahasa B dirancang untuk membuat sistem operasi UNIX/LINUX untuk komputer DEC PDP-7 di Bell Laboratories.

Pada tahun 1978 Dennis Ritchie dan Brian W. Kerninghan menerbitkan bahasa C melalui buku "The C Programming Language". Buku ini diterbitkan oleh Prentice Hall dan saat ini sedang diterjemahkan ke berbagai bahasa di seluruh dunia. Dan buku ini juga yang membuat legenda dan sejarah bahasa C dikenal sebagai K&R C (K=Kerninghan | R=Ritchie).

Seiring dengan pesatnya perkembangan bahasa C, banyak vendor dan pengembang perangkat lunak yang mengembangkan bahasa C sesuai dengan versinya masing-masing. Hal ini mendorong ANSI (American National Standards Institute) pada tahun 1983 untuk membentuk komite teknis yang disebut X3J11 untuk mengerjakan standar bahasa C yang bertujuan untuk membuat definisi standar bahasa C yang lebih modern dan komprehensif, dengan meningkatkan sintaks dan tata bahasa dari standar bahasa C. bahasa C. Pada akhir tahun 1989 panitia telah menyetujui standar ANSI untuk bahasa C yang kemudian dikenal sebagai ANSI C

Bahasa C adalah bahasa pemrograman tingkat tinggi. Bahkan bahasa C adalah salah satu bahasa tujuan umum yang paling populer. Dikatakan sebagai bahasa pemrograman tingkat tinggi karena kedekatannya dengan bahasa manusia. Semakin dekat dengan bahasa manusia, semakin tinggi bahasanya. Namun, ini memiliki efek meningkatkan kemampuan untuk mengakses instruksi bahasa mesin dasar secara langsung. Namun tidak sedikit yang mengatakan bahasa C merupakan bahasa pemrograman tingkat menengah karena bahasa C juga dapat berinteraksi langsung dengan mesin. Lebih tepatnya, bahasa C berada di antara High-level dan Low-level. Bahasa tingkat rendah adalah bahasa mesin (misalnya assembler), bahasa yang dapat berinteraksi langsung dengan mesin.

Bahasa C memiliki 32 kata kunci yang telah didefinisikan oleh ANSI. 32 kata kunci tersebut adalah: auto, break, case, char, const, continue, default, do, double, else, enum, extern, float, for, goto, if, int, long, register, return, short, signed , sizeof , static, struct, switch, typedef, union, unsigned, void, volatile, while.

Keuntungan pertama menggunakan bahasa C adalah dari segi portabilitas (portable). Program C yang kami tulis untuk satu platform dapat digunakan di platform lain, dengan atau tanpa perubahan. Hal ini terwujud karena adanya standarisasi ANSI untuk C.

Keuntungan kedua menggunakan bahasa C adalah bahasa pemrograman yang fleksibel. Dengan menguasai bahasa C kita dapat menulis dan mengembangkan berbagai jenis program mulai dari sistem operasi, pengolah kata, pengolah grafis, atau compiler hingga bahasa pemrograman yang satu ke bahasa pemrograman yang lain.

Keuntungan ketiga adalah reuse: bahasa pemrograman yang dapat digunakan berkali-kali di program atau aplikasi lain. Karena banyak bahasa yang dibuat menggunakan bahasa C, maka bahasa C cocok untuk bahasa pemula untuk belajar pemrograman. Seperti bahasa PERL yang sangat populer di kalangan World Wide Web (WWW), juga banyak mengambil fitur dari bahasa C. Contoh lain adalah C++ yang merupakan perluasan dari bahasa C. Belajar JAVA akan sangat mudah jika Anda mengetahui bahasa C.

* 1. **Perkembangan Bahasa C**

Sebelum tahun 1970-an (atau sebelumnya), menggunakan komputer seperti sekarang ini dari mengetik hingga bermain game komputer adalah hal yang mustahil. Revolusi yang terjadi di dunia komputer membutuhkan dua tahap untuk sampai pada zaman sekarang ini, yaitu ketika komputer dibangun untuk membantu para ilmuwan menyelesaikan tugas-tugas ilmiahnya, dan harga komputer turun secara signifikan dan dapat digunakan oleh pengguna. yang bukan programmer. , yang berlangsung beberapa dekade setelah fase pertama selesai.

Bahasa C merupakan pengembangan dari bahasa B yang ditulis oleh Ken Thompson pada tahun 1970. Bahasa C pertama kali ditulis oleh Brian W. Kernighan dan Denies M. Ricthie pada tahun 1972. Bahasa C, pada awalnya dioperasikan pada sistem operasi UNIX.

Bahasa C merupakan bahasa pemrograman tingkat menengah yang berada di antara bahasa tingkat rendah dan tingkat tinggi yang biasa disebut dengan Bahasa Tingkat Tinggi dengan Command Assambly. Bahasa C memiliki banyak kemampuan yang sering digunakan diantaranya kemampuan membuat software, seperti dBASE, Word Star dan lain-lain. Pada tahun 1980 seorang ahli bernama Bjarne Stroustrup mengembangkan beberapa hal dari bahasa C yang disebut “C with Classes” yang pada tahun 1983 diganti namanya menjadi C++. Tambahan untuk C++ adalah Pemrograman Berorientasi Objek (OOP), yang tujuan utamanya adalah untuk membantu membuat dan mengelola program besar dan kompleks.

* 1. **Jenis Variabel**

Beberapa variabel yang akan digunakan dalam pemrograman C adalah sebagai berikut, walaupun tidak tertutup kemungkinan ada penambahan lain:

**1.9 Struktur Program Bahasa C**

Program bahasa C adalah program yang terdiri dari satu atau lebih fungsi. Fungsi utama dan harus ada pada program C yang kita buat adalah fungsi main(). Fungsi main() adalah fungsi pertama yang akan diproses saat program dikompilasi dan dieksekusi, sehingga dapat disebut sebagai fungsi yang mengontrol fungsi lainnya. Karena struktur program C terdiri dari fungsi-fungsi lain sebagai subrutin, maka Bahasa C biasa disebut sebagai bahasa pemrograman terstruktur. Cara menulis fungsi dalam program bahasa C adalah dengan memberi nama fungsi tersebut dan kemudian membukanya dengan membuka ({) yang lebih kecil dan menutupnya.

Fungsi lain selain fungsi utama dapat ditulis setelah atau sebelum fungsi utama dengan deskripsi prototipe fungsi di awal program. Itu juga dapat ditulis ke file lain yang jika kita ingin menggunakan atau memanggil fungsi di file lain, kita harus menulis file header, dengan direktif preprocessor #include. File ini disebut file perpustakaan. Program C mencakup dokumentasi program, direktur prapemrosesan, deklarasi global, fungsi main(), fungsi pembuatan pemrograman, pembatas, pernyataan akhir, dan gaya program.

1. **Dokumentasi Program**

Menyediakan dokumentasi pada program sangat berguna untuk membantu memperjelas alur logika pemrograman. Karena tujuannya hanya sebagai dokumentasi, komentar yang ditulis pada program tidak diproses oleh compiler. Komentar dapat dimulai dengan simbol dua karakter yang terdiri dari garis miring dan tanda bintang (/\*) dan diakhiri dengan tanda bintang dan garis miring.

Karakter komentar /\* dapat ditempatkan di mana saja dalam program dan dapat menyertakan lebih dari satu komentar, dengan syarat setiap /\* harus ditutup dengan \*/. Di awal program, komentar yang diberikan biasanya ditujukan untuk menjelaskan apa yang dilakukan program, sedangkan di bagian lain program tujuannya adalah untuk memperjelas logika program.

2.1  **Pengarah Prapengoloahan**

Dalam mengolah kode-kode program, compiler C melakukan beberapa tahapan, yaitu preprocessing untuk melakukan persiapan-persiapan yang diperlukan untuk sebuah file program kompilasi. Dalam program perujuk prapemrosesan, karakter # ditulis pada baris pertama program.

Preprocessing memperlakukan file program sebagai serangkaian baris teks: membaca, memproses, dan menulis ulang hasil pemrosesan ke dalam file asli. Praprosesor menghapus semua baris perintah yang telah diproses sebelumnya dari file sumber dan membuat perubahan pada file sumber sesuai dengan prompt perintah yang diberikan. Secara garis besar, layanan ini dapat dibagi menjadi tiga kelompok:

1. Penyisihan Berkas (#include)
2. Pendefinisian Makro (#define)
3. Pengarah Kendali Kompiler (#ifdef, #ifindef)
   1. **Deklarasi Global**

Pada bagian deklarasi global terdapat deklarasi variabel dan prototipe fungsi. Semua variabel yang dideklarasikan pada bagian ini akan dikenali oleh semua bagian program yang berada di bawahnya. Sedangkan prototype fungsi merupakan deklarasi tentang penggunaan suatu fungsi dalam program. Ada tiga elemen yang perlu disebutkan: tipe data yang dikembalikan oleh fungsi, nama fungsi dan daftar argumen yang diberikan ke fungsi yang diberikan jumlah argumen (arity) dan tipe data dari setiap argumen.

* 1. **Fungsi Main ()**

Fungsi main() memegang peranan penting dalam sebuah program. Fungsi ini merupakan fungsi utama dalam setiap program C dimana eksekusi seluruh program dimulai. Tidak peduli berapa banyak fungsi yang ada dalam program C, main ( ) adalah fungsi utama yang akan dilakukan oleh kompiler.

1. **Fungsi Buatan Pemrograman**

Selain main ( ) yang mempunyai kedudukan khusus dalam sebuah program, terdapat pula fungsi-fungsi pemrogram buatan. Pada fungsi ini dapat diberikan deklarasi prototipe fungsi lain secara lokal, sehingga hanya dikenal oleh fungsi tersebut. Kode program yang dapat dieksekusi juga berupa fungsi input/output standar, konstruksi runtunan, konstruksi keputusan, dan konstruksi.

1. **Pembatas**

Setelah mendefinisikan fungsi, ada kurung kurawal buka "{" dan kurung kurawal tutup "}" yang menunjuk ke ujung blok fungsi, kurung kurawal ini disebut pembatas. Di badan program, kurung kurawal juga dapat digunakan untuk membatasi pernyataan majemuk yang termasuk dalam blok kode program. Selain kurung kurawal, ada juga pembatas lainnya, antara lain: [ ], < >, ( ), “ “, dan ' '.

1. **Akhir Pernyataan**

Setiap pernyataan dalam C diakhiri dengan titik koma ( ; ) yang memberitahu compiler akhir dari pernyataan tersebut. Pengembalian carriage yang diperoleh saat menekan tombol bukan merupakan indikasi akhir dari pernyataan, karena C mengabaikan semua yang disebut karakter spasi, yaitu spasi, tabulator, dan carriage return (baris baru).

1. **StyleProgram**

Keterbacaan adalah elemen yang sangat penting dalam menyusun sebuah program. Program yang dikompilasi harus mudah dibaca dan dapat menggambarkan kerangka kerja dan algoritma yang digunakan. Kemudahan menulis program sangat dipengaruhi oleh sintaks (aturan penulisan) yang dapat digunakan. Sintaks program juga dapat mempermudah pengujian dan pemahaman jika sewaktu-waktu program perlu dimodifikasi.

**Kelebihan dan Kekurangan**

**Kelebihan Bahasa C**

1. Bahasa C tersedia di hampir semua jenis komputer, baik komputer mikro, mini, maupun besar (mainframe).
2. Kode bahasa C bersifat portabel. Aplikasi yang ditulis dalam bahasa C untuk komputer tertentu dapat digunakan di komputer lain dengan hanya sedikit modifikasi.
3. Berbagai struktur data dan pengendalian proses yang disediakan dalam C sehingga memungkinkan untuk membuat program yang terstruktur. Struktur bahasa yang baik, selain itu mudah dipelajari juga memudahkan dalam pembuatan program, menghapus kesalahan program dan akan menghasilkan dokumentasi program yang baik.
4. Dibandingkan dengan bahasa mesin atau assembly, C jauh lebih mudah dipahami dan programmer tidak perlu mengetahui mesin komputer secara detail.
5. C memungkinkan memanipulasi data dalam bentuk bit dan byte. Selain itu, juga dimungkinkan untuk memanipulasi alamat data atau pointer.

**Kekurangan Bahasa C**

* 1. Jumlah operator dan fleksibilitas penulisan program terkadang membingungkan pengguna. Jika Anda tidak menguasainya, itu pasti akan menimbulkan masalah.
  2. Pemrogram C tingkat pemula umumnya tidak pernah mengenal pointer dan tidak terbiasa menggunakannya. Meskipun kekuatan C terletak pada pointer.

**BAB 3**

**JENIS DATA**

**2.4**  **Tipe Data**

Tipe data merupakan bagian terpenting dari program karena tipe data mempengaruhi setiap instruksi yang akan dieksekusi oleh komputer. Misalnya, 5 dibagi 2 dapat menghasilkan hasil yang berbeda tergantung pada tipe datanya. Jika 5 dan 2 bertipe integer maka akan mengembalikan nilai 2, tetapi jika keduanya bertipe float maka akan mengembalikan nilai 2.5000000. Memilih tipe data yang tepat akan membuat proses operasi data lebih efisien dan efektifTipe data merupakan bagian program yang paling penting karena tipe data mempengaruhi setiap instruksi yang akan dilaksanakan oleh komputer. Misalnya saja 5 dibagi 2 saja bisa menghasilkan hasil yang berbeda tergantung tipe datanya. Jika 5 dan 2 bertipe integer maka akan menghasilkan nilai 2, namun jika keduanya bertipe float maka akan menghasilkan nilai 2.5000000. Pemilihan tipe data yang tepat akan membuat proses operasi data menjadi lebih efisien dan efektif.

Data adalah suatu nilai yang dapat dinyatakan dalam bentuk konstanta atau variabel. Konstanta mewakili nilai tetap, sedangkan variabel mewakili nilai yang dapat berubah selama eksekusi.

Data menurut jenisnya dapat dibagi menjadi lima kelompok, yang disebut sebagai tipe dasar.

Lima tipe dasar adalah:

1.      Bilangan bulat (*integer*).

2.      Bilangan pecahan (*float*)

3.      Bilangan pecahan (*double*).

4.      Deretan karakter (*string*)

5.      Tak bertipe (*void*)

Dalam bahasa C terdapat beberapa tipe data, diantaranya adalah sebagai berikut:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Tipe Data** | **Ukuran** | **Range (Jangkauan)** | **Format** | **Keterangan** |
| 1 | Char | 1 byte | -128 s/d 127 | %c | karakter |
| 2 | Int | 2 byte | -32768 s/d 32767 | %d, %i | integer/bilangan bulat |
| 3 | Float | 4 byte | -3.4e-38 s/d 3.4e+38 | %f | float/pecahan |
| 4 | Double | 8 byte | -1.7e-308 s/d 1.7e+308 | %lf | pecahan presisi ganda |
| 5 | Void | 0 byte | - | - | tidak bertipe |

1. **Tipe Data Char**

Tipe karakter data yang digunakan untuk mewakili karakter. Setiap karakter memori adalah 1 byte. Satu byte terdiri dari 8 bit, sehingga dapat menampung 28 = 256 nilai yang berbeda. Nomor ini sesuai dengan jumlah karakter ASCII yang digunakan pada komputer mikro, dan diberi nomor 0 hingga 255.

Dari 256 karakter tersebut, beberapa di antaranya adalah karakter yang tidak dapat dicetak. Untuk dapat merepresentasikan 256 karakter, variabel bertipe char harus dideklarasikan sebagai karakter yang tidak ditandatangani. Jika tidak, variabel akan terlihat sebagai formulir char yang ditandatangani sebagai default untuk tipe data char, sehingga hanya dapat menampung setengah dari total karakter ASCII.

1. **Tipe Data Int**

Tipe data yang digunakan untuk merepresentasikan nilai integer. Tipe data integer disimpan dalam memori sebesar 4 byte (32 bit). Pada komputer 32-bit, tipe data integer menempati satu kata memori.

Grup bilangan bulat terdiri dari tiga jenis: int, pendek, dan panjang, yang masing-masing dibagi menjadi dua grup: ditandatangani dan tidak ditandatangani. Jika tidak dinyatakan sebaliknya, kompilator akan memperlakukan ketiga jenis bilangan bulat sebagai int bertanda yang dapat berisi angka positif dan negatif. Jika hanya bilangan positif , harus dinyatakan secara eksplisit dengan memberikan kata unsigned.

1. **Tipe Data Float**

Variabel bertipe float termasuk dalam kelompok floating point yang menampung bilangan real. Bilangan real adalah bilangan yang memiliki bagian desimal, titik, bagian pecahan, dan bagian eksponensial. Deklarasi variabel bertipe float memiliki format yang sama dengan tipe integer: mendeklarasikan tipe data float, diikuti dengan nama variabel.

1. **Tipe Data Double**

Variabel bertipe ganda juga termasuk dalam kelompok floating point tetapi memiliki bilangan presisi ganda. Tingkat akurasi hingga 15 digit di belakang koma. Selain itu, ada juga variable tipe ganda panjang yang memiliki tingkat akurasi hingga 18 digit di belakang koma. Kedua jenis double ini masing-masing menempati 8 dan 12 byte lokasi memori.

* 1. **Konstanta**

Konstanta adalah nilai yang tidak dapat diubah selama proses program. Nilai konstan selalu tetap. Konstanta harus didefinisikan di awal program. Konstanta dapat berupa bilangan bulat, pecahan, karakter, dan string. Contoh konstanta: 50, 13, 3.14, 4.50005, 'A', 'Bahasa C'.

Selain itu, bahasa C juga menyediakan beberapa karakter khusus yang disebut karakter pelarian, di antaranya:

1.      \a : untuk bunyi bell (*alert*)

2.      \b : mundur satu spasi (*backspace*)

3.      \f : ganti halaman (*form feed*)

4.      \n : ganti baris baru (*new line*)

5.      \r : ke kolom pertama, baris yang sama (*carriage return*)

6.      \v : tabulasi vertikal

7.      \0 : nilai kosong (*null*)

8.      \’ : karakter petik tunggal

9.      \” : karakter petik ganda

10.   \\ : karakter garis miring

Ada beberapa jenis konstanta dalam bahasa C, diantaranya sebagai berikut:

1.  Konstanta karakter ditunjukkan dengan tanda petik tunggal (‘) di awal dan akhir karakter. Contoh 'A' dan '@'. Konstanta karakter adalah karakter yang diapit oleh tanda kutip tunggal. Karakter yang diapit tanda kutip tunggal di atas adalah karakter cetak. Selain itu, ada juga karakter non-printable yang disebut karakter escape, seperti carriage return, bells, newlines, dan null character dan harus didahului dengan karakter backslash.

2.  Konstanta bilangan bulat ditulis dengan tanda yang mengandung pemisah ribuan dan tidak ada pemisah pecahan, misalnya -1 dan 32767. Konstanta bilangan bulat terdiri dari deretan bilangan bulat. Berdasarkan dasarnya, konstanta bilangan bulat yang biasa digunakan dalam C dapat dibagi menjadi tiga jenis: desimal, oktal, dan heksadesimal.

1. Konstanta nyata (mengambang dan ganda) dapat berisi pecahan (dengan titik). Misalnya 27.5f (untuk float) dan 27.5( untuk double).
2. Konstanta string adalah string karakter yang diawali dan diakhiri dengan tanda kutip ganda (“), misalnya “Pemrograman C Dasar”. Karakter tersebut dapat berupa huruf, angka, karakter khusus, dan spasi. Karakter yang membentuk string disimpan dalam memori dengan nilai numeriknya yang mewakili alamat lokasi memori. Di akhir string, kompiler akan menambahkan karakter nol ('\0') untuk menandai akhir string*.*
   1. **Variabel**

Sebelum kita dapat menggunakan variabel tertentu dalam program C, kita harus memberi tahu kompiler tentang jenis variabel itu sendiri. Dilihat dari contoh di atas, kita melihat bahwa semua variabel mengandung angka. Namun, komputer tidak hanya memproses angka, tetapi selain angka variabel, juga mampu menyimpan berbagai jenis data. Bahkan, untuk jenis angka, C mendukung tiga jenis:

* 1. Saat kita menulis sebuah program, yang mengharuskan adanya bilangan bulat (bilangan bulat), seperti populasi siswa di sekolah, atau lainnya, C menawarkan kepada kita apa yang disebut "integer”.
  2. Ketika kita menulis program yang berhubungan dengan masalah keuangan (yang sering digunakan dalam industri), kita harus menggunakan angka dengan angka yang panjang, mungkin dalam triliunan. Bahasa C menawarkan kepada kita apa yang disebut "decimal."‎
  3. Saat kita menulis program yang digunakan untuk menghitung hal-hal ilmiah, seperti rumus relativitas dan fisika kuantum, kita mungkin memerlukan koma dan eksponen bilangan yang direpresentasikan dalam notasi ilmiah (mantissa dan eksponen). Di C, angka seperti itu juga disebut angka floating point, yang diwakili dengan menggunakan kata kunci ‎‎"float".‎

Ketiga jenis bilangan di atas dapat kita gabungkan dalam sebuah program. Untuk saat ini, kita akan belajar tentang bilangan bulat (integer), yang dapat berupa bilangan bulat positif atau bilangan bulat negatif.

Variabel adalah pengidentifikasi yang digunakan untuk mewakili nilai tertentu dalam proses program. Tidak seperti konstanta yang nilainya selalu tetap, nilai suatu variabel dapat diubah sesuai kebutuhan. Nama sebuah variabel dapat ditentukan oleh programmer dengan aturan sebagai berikut:

* 1. Terdiri dari kombinasi huruf dan angka dengan karakter pertama harus huruf.
  2. b. Bahasa C bersifat case sensitive, artinya huruf besar dan huruf kecil dianggap berbeda. Jadi antara Metal, dengan metal itu berbeda.
  3. Tidak boleh mengandung spasi.
  4. Tidak boleh mengandung simbol khusus, kecuali garis bawah, seperti: $, ?, %, #, !, &, \*, (, ), -, +, dll..
  5. Panjangnya bebas, tetapi hanya 32 karakter pertama yang digunakan.

1. **Aturan Pendefisian Variabel dalam Bahasa C**

Berikut aturan penamaan variabel di dalam bahasa pemrograman C:

1. Variabel bisa terdiri dari huruf, angka dan karakter underscore / garis bawah ( \_ ).
2. Karakter pertama dari variabel hanya boleh berupa huruf dan underscore ( \_ ), tidak bisa berupa angka. Meskipun dibolehkan, sebaiknya tidak menggunakan karakter underscore sebagai awal dari variabel karena bisa bentrok dengan beberapa variabel settingan program.
3. Variabel harus selain dari keyword. Sebagai contoh, kita tidak bisa memakai kata int sebagai nama variabel, karena int merupakan keyword untuk menandakan tipe data integer.
4. Beberapa compiler bahasa C ada yang membatasi panjang variabel maksimal 31 karakter. Agar lebih aman, sebaiknya tidak menulis nama variabel yang lebih dari 31 karakter.
5. **Mendeklarasikan Variabel**

Variabel digunakan dalam program untuk menyimpan suatu nilai, dan nilai yang ada padanya dapat diubah-ubah selama eksekusi program berlangsung. Variabel yang akan digunakan dalam program haruslah dideklarasikan terlebih dahulu.

Pengertian deklarasi di sini berarti memesan memori dan menentukan jenis data yang bisa disimpan di dalamnya.

**Bentuk umum deklarasi variabel**:

*Tipe Data-Variabel;*

Pada pendeklarasian varibel, daftar-variabel dapat berupa sebuah variabel atau beberapa variabel yang dipisahkan dengan koma.

Contoh:

int var\_bulat1;

float var\_pecahan1, var\_pecahan2;

1. **Memberikan Nilai Pada Variabel**

Untuk memberikan nilai ke variabel yang telah dideklarasikan, maka bentuk umum

pernyataan yang digunakan adalah :

*nama\_variabel = nilai;*

Contoh:

int var\_bulat = 10;

double var\_pecahan = 10.5;

1. **Inisialisasi Variabel**

Adakalanya dalam penulisan program, setelah dideklarasikan, variabel langsung

diberi nilai awal. Sebagai contoh yaitu variabel nilai :

*Int nilai;*

*nilai = 10*;

Dua pernyataan di atas sebenarnya dapat disingkat melalui pendeklarasian yang disertaipenugasan nilai, sebagai berikut :

*int nilai= 10;*

Cara seperti ini banyak dipakai dalam program C, di samping menghemat penulisan pernyataan, juga lebih memberikan kejelasan, khususnya untuk variabel yang perlu diberi nilai awal (diinisialisasi

1. **Aturan Pendefisian Variabel**

Ketika akan menulis variabel, dalam hampir semua bahasa pemrograman terdapat 2 proses: deklarasi dan inisialisasi.

Deklarasi adalah proses untuk memberitahukan compiler bahasa C bahwa kita akan membuat sebuah variabel. Bahasa C termasuk bahasa pemrograman yang menggunakan konsep strongly typed programming language, yang artinya untuk setiap variabel harus ditulis akan berisi tipe data apa. Apakah itu angka bulat (integer), angka pecahan (float), huruf (char), atau yang lain.

Ketika kita ingin membuat variabel, harus ditentukan dulu apa tipe data dari variabel tersebut. Pembahasan mengenai tipe data di dalam bahasa C akan kita bahas dalam tutorial tersendiri, namun berikut tipe data yang sering dipakai:

1. *Tipe data integer*, yakni tipe data angka bulat seperti 1, 5 atau 1000. Tipe data integer ditulis dengan keyword int.
2. *Tipe data float*, yakni tipe data angka pecahan seperti 1.33,  5.90 atau 1000.99. Tipe data float ditulis dengan keyword float.
3. *Tipe data character*, yakni tipe data huruf seperti ‘A’, ‘a’, atau ‘Z’. Tipe data character ditulis dengan keyword char**.**
   1. **Operator**

Operator adalah instruksi khusus yang ditugaskan ke variabel. Operator yang sering digunakan dalam pemrograman adalah Operator Aritmatika, Operator Perbandingan, Operator Logika, dan lain-lain, namun yang akan kita bahas kali ini adalah Operator Aritmatika, Operator Perbandingan, dan Operator Logika.

1. **Operator Aritmatika**

Operator aritmatika adalah operator yang digunakan untuk fungsi/operasi matematika, operator aritmatika dasar untuk C++ dan tanpa proses include.:

* 1. \* : untuk perkalian
  2. /  : untuk pembagian
  3. % : untuk sisa hasil bagi (*modulus*)
  4. + : untuk penjumlahan
  5. - : untuk pengurangan

Namun yang akan dibahas kali ini adalah operator aritmatika dasar, namun untuk pengetahuan saja, operator aritmatika yang menggunakan file header math.h sebagai berikut:

* 1. pow(a,b) : untuk pemangkatan
  2. max() : untuk mencari nilai maximal
  3. min() : untuk mencari nilai minimum
  4. sqrt() : untuk menghitung akar suatu bilangan

1. **Operator Pembanding**

Operator perbandingan digunakan untuk membandingkan antara dua variabel. Operator perbandingan sering digunakan ketika memeriksa atau pernyataan if, loop atau pernyataan untuk, sementara, lakukan sementara. Tapi sekarang kita akan membahas penggunaan dasarnya dulu.

1. **Operator Logika**

Operator logika adalah operator yang membandingkan hubungan antara dua hasil operasi perbandingan. Operator logika membandingkan hasil logika dari dua jika salah maka akan melakukan kebalikannya. Ada tiga operator yang digunakan dalam operasi logika, yaitu:

|  |  |
| --- | --- |
| **Logika** | **Keterangan** |
| && | Logika dan (AND) |
| || | Logika or (ATAU) |
| ! | Logika not (INGKARAN) |

1. **Operator Penegasan**

Komputer pada awalnya diciptakan untuk membantu manusia melakukan “perhitungan numerik”, sehingga tidak mengherankan jika bahasa pemrograman juga mengadopsi “elemen aljabar tertentu, termasuk penggunaan huruf atau kata sebagai “pengganti angka.” Berikut beberapa contoh aljabar . sederhana:‎

A = 3

B = 5

C = A + B

Bagi saya, persamaan di atas memang merupakan persamaan yang tidak rumit sama sekali, tetapi jika kita dapat membujuk komputer untuk menyelesaikan persamaan di atas, maka kita juga dapat membujuknya untuk menyelesaikan persamaan atau masalah lain yang jauh lebih kompleks. Sebagai langkah pertama, kita akan mencoba mengubah aljabar di atas menjadi sintaks dalam bahasa C, dan bentuknya seperti di bawah ini:‎

A = 3;

B = 5;

C = A + B;

Tiga baris kode di atas, tidak membentuk program C yang lengkap, tetapi contoh kode di atas dapat ditemukan dalam program C, mungkin program yang dapat kita buat sendiri. Kita dapat menulis kode di atas menggunakan beberapa editor teks (seperti notepad atau WordPad) untuk menggunakan alat khusus yang dirancang untuk menyederhanakan pemrograman menggunakan bahasa C, seperti [www.icsharpcode.net/OpenSource/SD #develop (SharpDevelop)] dan Microsoft Edisi Visual C Express. Setelah menulis kode di atas, langkah selanjutnya yang harus kita lakukan adalah mengubah kode di atas menjadi file executable (\*.EXE) yang dapat kita jalankan.‎

Perbedaan yang signifikan antara bentuk pertama (aljabar) dan bentuk kedua (kode dalam bentuk bahasa C) adalah bahwa setiap baris dalam bahasa C selalu diakhiri dengan karakter titik koma (titik koma). Setiap baris yang diakhiri dengan karakter titik koma disebut pernyataan program C. Sebuah program komputer yang dibuat dengan menggunakan C umumnya terdiri dari berbagai pernyataan. Huruf A, B, dan C yang disebutkan di atas adalah yang disebut variabel, dan masing-masing variabel mewakili lokasi di memori tempat nilai variabel akan disimpan.

1. **Operator Relasional**

Operator relasional adalah operator biner yang digunakan untuk membandingkan nilai. Dalam C, ada enam jenis operator relasional: lebih besar, lebih besar dari atau sama dengan, lebih kecil dari, kurang dari atau sama dengan, sama dengan, dan tidak sama dengan.

    Perbandingan dilakukan dengan mengikuti logika Boolean, yang menghasilkan nilai numerik 1 jika relasinya benar atau numerik 0 jika relasinya salah. Operator relasional umumnya digunakan dalam struktur keputusan sebagai dasar untuk menentukan pernyataan yang akan dieksekusi. Selain itu, juga digunakan sebagai kondisi pengujian pada struktur pengulangan.

* 1. **Pengkondisian Bahasa Pemrograman C**
  2. **Struktur Kondisi “if…”**

Struktur if dibentuk dari pernyataan if dan sering digunakan untuk menyeleksi suatu kondisi tunggal. Bila proses yang diseleksi terpenuhi atau bernilai benar, maka pernyataan yang ada di dalam blok if akan diproses dan dikerjakan. Bentuk umum struktur kondisi if **:**

if(kondisi){     Bila kondisi bernilai benar atau true maka bagian ini dijalankan     Bila kondisi benilai salah atau false maka bagian ini tidak dijalankan }

* 1. **Struktur kondisi “if..else..”**

Dalam struktur kondisi if…..else minimal terdapat dua pernyataan. Jika kondisi yang diperiksa bernilai benar atau terpenuhi maka pernyataan pertama yang dilaksanakan dan jika kondisi yang diperiksa bernilai salah maka pernyataan yang kedua yang dilaksanakan. Bentuk umumnya adalah sebagai berikut :

if(kondisi){

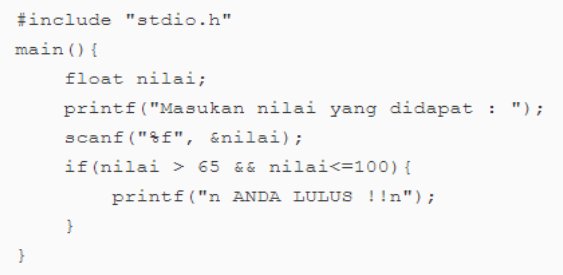
    Bila kondisi bernilai benar atau true maka bagian ini dijalankan

}else{

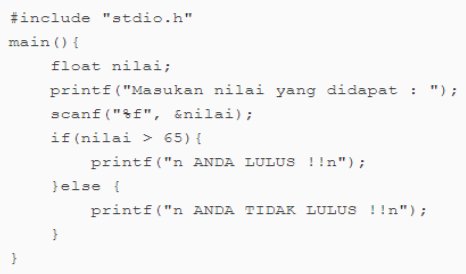
    Bila kondisi benilai salah atau false maka bagian ini yang dijalankan

}

**Contoh** “if..”



**Contoh “if else..”**

****

* 1. **Struktur Kondisi “**if..else..if..else**”**

Dalam struktur kondisi if…..elseif…else minimal terdapat 3 pernyataan. Bentuk umumnya adalah sebagai berikut :

*if(kondisi 1){*

*Bila kondisi1 bernilai benar atau true maka*

*HANYA bagian ini dijalankan.*

*}else if(kondisi 2){*

*Bila kondisi2 bernilai benar atau true maka*

*HANYA bagian ini dijalankan.*

*}else if(kondisi 3){*

*Bila kondisi3 bernilai benar atau true maka*

*HANYA bagian ini dijalankan.*

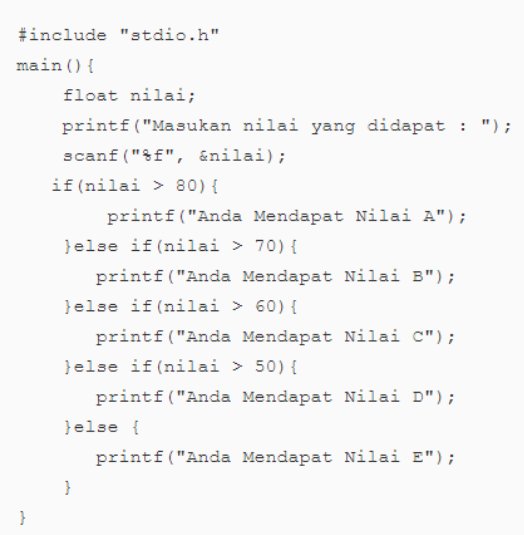
*}else{*

*Bila kondisi1, kondisi2 dan kondisi3 bernilai salah atau false maka*

*bagian ini yang dijalankan*

*}*

**Contoh** “if..else..if..else..”

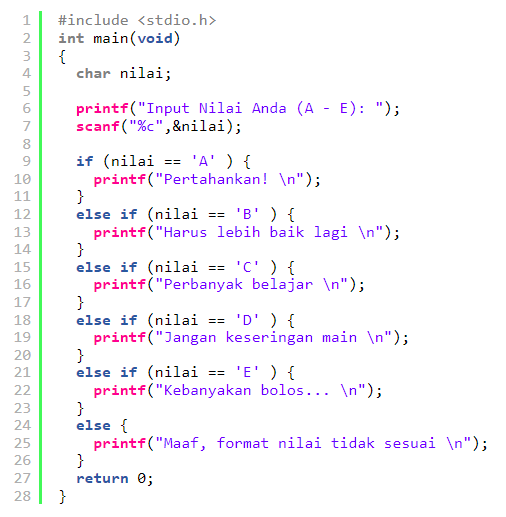
****

* 1. **Switch Case**

Kondisi SWITCH CASE adalah percabangan kode program dimana kita membandingkan isi sebuah variabel dengan beberapa nilai. Kondisi SWITCH CASE terdiri dari 2 bagian, yakni perintah SWITCH dimana terdapat nama variabel yang akan diperiksa, serta 1 atau lebih perintah CASE,

*switch (nama\_variabel) {  case 'nilai\_1':    // Kode program yang dijalankan jika nama\_variabel == nilai\_1    break;  case 'nilai\_2':    // Kode program yang dijalankan jika nama\_variabel == nilai\_2    break;  case 'nilai\_3':    // Kode program yang dijalankan jika nama\_variabel == nilai\_3    break;  ...  ...  default:     // Kode program yang dijalankan jika tidak ada kondisi yang terpenuh*

**contoh** switch case

****

**2. 9 Ekspresi**

Ekspresi adalah kombinasi dari operator dan operan. Dari menggabungkan operator dan operan, ekspresi yang lebih kompleks dapat dibangun. Namun, dalam bentuknya yang paling sederhana, sebuah ekspresi mungkin hanya terdiri dari satu operan. Sebagai contoh, perhatikan ekspresi berikut:

-15

8+7

((x + y) / z) \* 10) / 2 /\* (x + ((y / z) \* 10) / 2)) \* /

a = b + 5

x = ++y  %  3

(x >= 0) || ((x % 2) == 0)

6 + (c = 3 + 8 ) – (d = 1 + 3)

    Seperti yang terlihat pada contoh di atas, ekspresi dapat berupa konstanta, variabel, atau kombinasi keduanya. Beberapa ekspresi juga dapat merupakan kombinasi dari ekspresi yang lebih kecil (subekspresi), misalnya x + y adalah subekspresi dari ((x + y) / z ) \* 10)/

Demikian pula (x>=0), (x % 2), ( c = 3+ 8 ), dan (d = 1 + 3) dalam dua ekspresi terakhir Properti penting untuk dicatat dalam C adalah bahwa setiap ekspresi memiliki nilai. Untuk mendapatkan nilai ekspresi, lakukan operasi sesuai dengan prioritas operator yang terdapat dalam ekspresi.

**3. 0 Operasi Input dan Output**

* 1. Printf

Fungsi printf digunakan untuk menampilkan output di layar. Perhatikan penggunaan printf berikut:  
  
#include  
main()  
{  
int nilai=3;  
printf(“Bahasa C menyenangkan \n”);  
printf(“nilai = %d \n”,nilai);  
}

    Fungsi printf pertama disertai dengan escape sequence yang berfungsi untuk memindahkan garis. Jadi dengan fungsi printf pertama kursor akan turun satu baris. Sedangkan fungsi printf kedua yang terdapat format data %d berfungsi untuk menampilkan data dengan tipe integer.

Jangan lupa bahwa variabel yang menyimpan nilai harus dimasukkan setelah tanda kutip terakhir.

* 1. **Puts**

Cara lain untuk menampilkan output ke layar adalah dengan menggunakan fungsi puts. Namun fungsi puts hanya digunakan untuk menampilkan nilai string dan sudah berisi line feed atau urutan escape line-break. Perhatikan contoh berikut:

#include  
main()  
{  
puts(“Bahasa C menyenangkan ”);  
puts(“Belajar bahasa C “);  
}

* 1. **Putschar**

Fungsi ini digunakan untuk menampilkan hanya satu karakter dan tidak berisi urutan escape “\n”. perhatikan contoh berikut:  
  
#include  
main()

{  
Char a;  
printf(“a = ”);scanf(“%c”,&a);  
printf(“Data yang anda masukkan ”);  
putchar(‘\n’);  
putchar(a);  
}

     Pada putchar pertama kita ingin membuat program untuk memutus baris menggunakan "\n". kami menggunakan tanda kutip tunggal (‘) karena karakter dalam program akan dikenali jika mereka adalah tanda kutip tunggal. Fungsi putchar ini hanya dapat menampilkan satu karakter, jadi berapa pun nilai yang kita masukkan, hanya karakter pertama yang ditampilkan.

* 1. **Scanf**

Jika kita ingin memasukkan data dari keyboard, kita bisa menggunakan fungsi scanf ini. Data selanjutnya akan didefinisikan sebagai data variabel. Jika fungsi scanf digunakan untuk membaca data dengan tipe array, karakter yang selanjutnya akan kita sebut sebagai string, maka fungsi ini hanya akan membaca data sampai ditemukan titik kosong. Dengan demikian nilai setelah dikosongkan dianggap bukan lagi nilai variabel yang akan mengisi variabel tersebut. Fungsi scanf lebih cocok untuk data numerik.

   Fungsi scanf, jika digunakan, harus disertai dengan operator penanda alamat & di depan nama variabel yang digunakan untuk menyimpan data. Jika Anda tidak menggunakan operator alamat & ini, saat data dimasukkan, akan muncul pesan kesalahan Segmentasi kesalahan. Sedangkan untuk data string, tidak perlu menggunakan operator ini. Tapi kalau dipakai sih nggak masalah. Fungsi scanf biasanya digunakan bersama dengan fungsi printf. Perhatikan contoh berikut:

#include  
main()  
{  
char a[25];  
int b;  
printf(“a = ”);scanf(“%s”,a);  
printf(“b = ”);scanf(“%d”,&b);  
printf(“Data yang anda masukkan \n”);  
printf(“a = %s \n”,a);  
printf(“b = %d \n”,b);  
}

* 1. **Gets**

Jika kita menggunakan fungsi scanf untuk membaca data bertipe string, maka data hanya akan dibaca sampai ditemukan spasi. Misalnya nama “naufal rasendriya” hanya kata “naufal” yang akan terbaca oleh program, sedangkan kata “resendriya” tidak terbaca karena sebelum kata tersebut program telah menemukan spasi yang menyebabkan data ke variabel yang menyimpan nama tersebut untuk dianggap lengkap.

Untuk keperluan memasukkan data string panjang yang dipisahkan oleh spasi, bahasa C menyediakan fungsi get. Perhatikan contoh berikut:

#include  
main()  
{  
char nama[25];  
char alamat[50];  
printf(“Nama : ”);gets(nama);  
printf(“Alamat : ”);gets(alamat);  
printf(“\n”);  
printf(“Data yang anda masukkan \n”);  
printf(“Nama : %s \n”,nama);  
printf(“Alamat : %s \n”,alamat);  
}

* 1. **Getchar**

Jika Anda ingin memasukkan nilai karakter ke dalam variabel bertipe karakter, Anda dapat menggunakan perintah getchar. Perhatikan contoh program berikut:  
  
#include  
main() {

Char nilai;  
printf(“Nilai anda : ”);  
nilai=getchar();  
printf(“Nilai yang anda masukkan = %c \n”,nilai );

}

**3. 1   Pengulangan**

Struktur pengulangan adalah struktur program yang digunakan untuk melakukan suatu pernyataan secara berulang-ulang. C mempunyai tiga struktur pengulangan yaitu For, While, dan Do-while.

**1.  Pengulangan For**

For loop adalah pengulangan dimana terdapat tiga ekspresi yang digunakan untuk mengontrol proses pengulangan.

Bentuk umum struktur pengulangan

**For** (*ekspr 1; ekpr 2; eksr 3*)

*Contoh:*

#include<stdio.h>

int main() {

    int i;

    for (i=5; i>0; i--)

    {

printf("saya suka belajar dasar-dasar pemrograman\n",i);

     }

     return 0;

}

**2.   Pengulangan While**

Perulangan while membentuk struktur perulangan yang akan mengulangi eksekusi pernyataan di badan perulangan selama ekspresi yang diperiksa bernilai benar.

Bentuk umum struktur pengulangan

**While**(*ekspresi*)

Contoh :

#include<stdio.h>

int main()

{

     int n = 5;

     while(n<7)

{

          printf("n = %d\n", n);

          n++;

          printf("Sekarang n = %d\n", n);

     }

     return 0;

}

**3.   Pengulangan do-while**

Perulangan do-while adalah pengujian yang dilakukan pada akhir setelah melalui badan perulangan, sehingga badan perulangan harus telah dieksekusi minimal satu kali.

Bentuk umum struktur pengulangan

**Do**

    Contoh :

**While** (*ekspresi*);

#include <stdio.h>

int main()

{

     int v = 1;

     do

     {

          printf("dasar-dasar pemrograman\n");

          v++;

     }

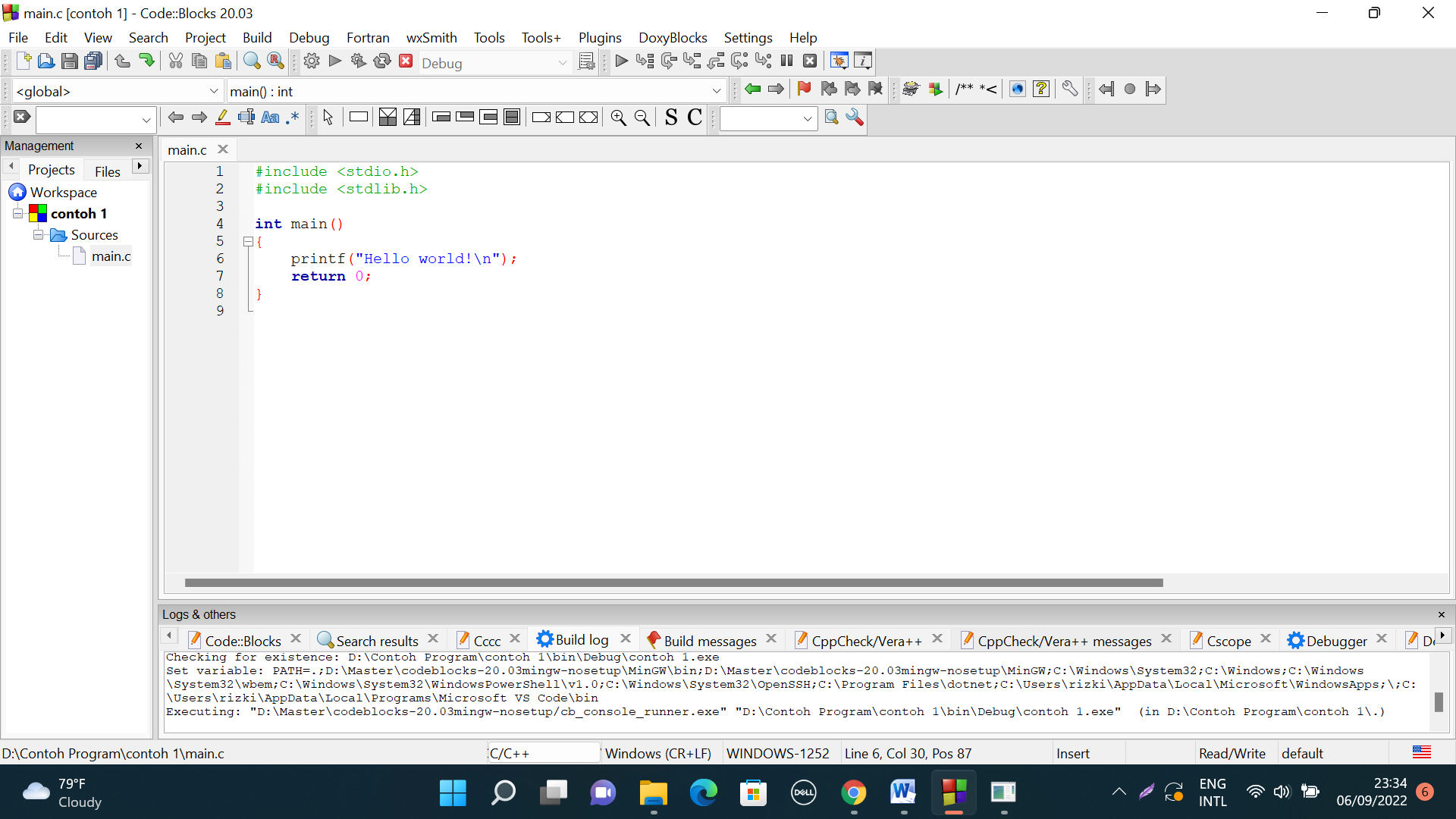
     while(v<=6);

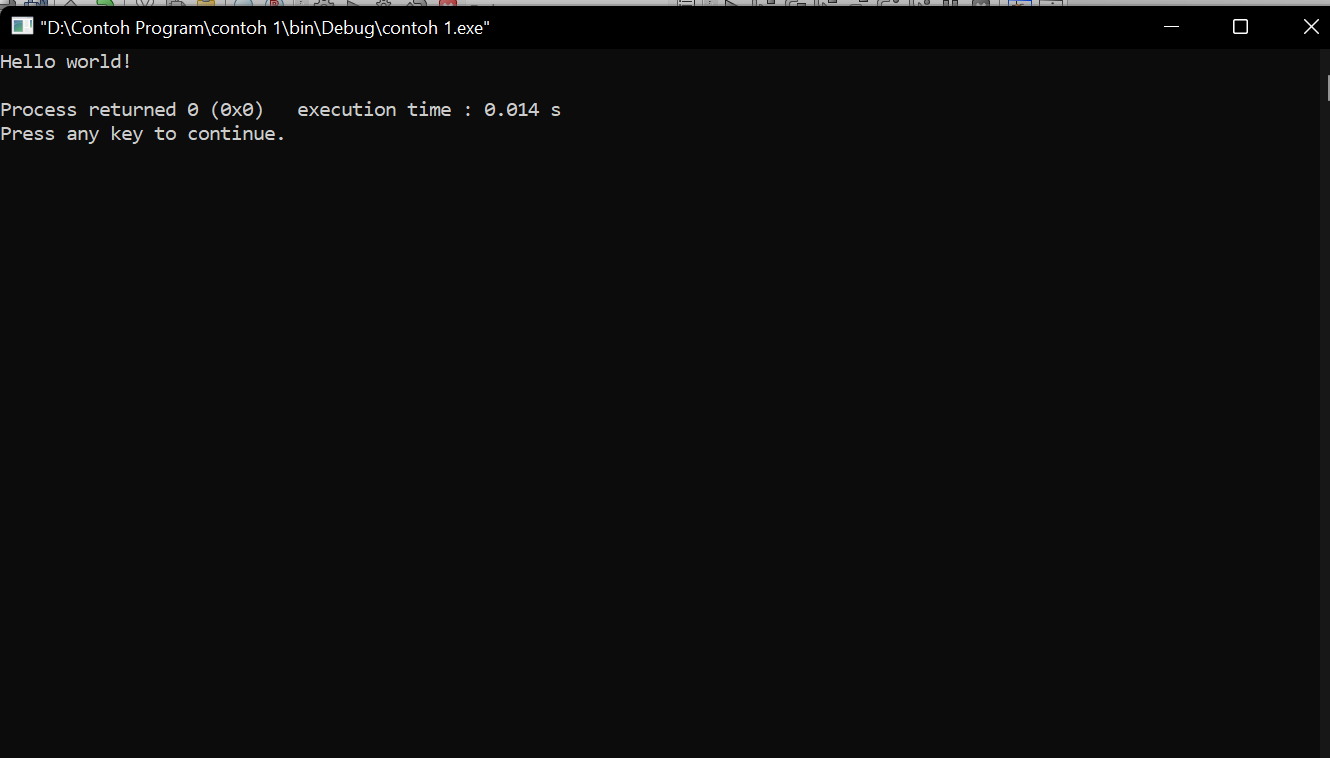
     return 0;

**BAB IV**

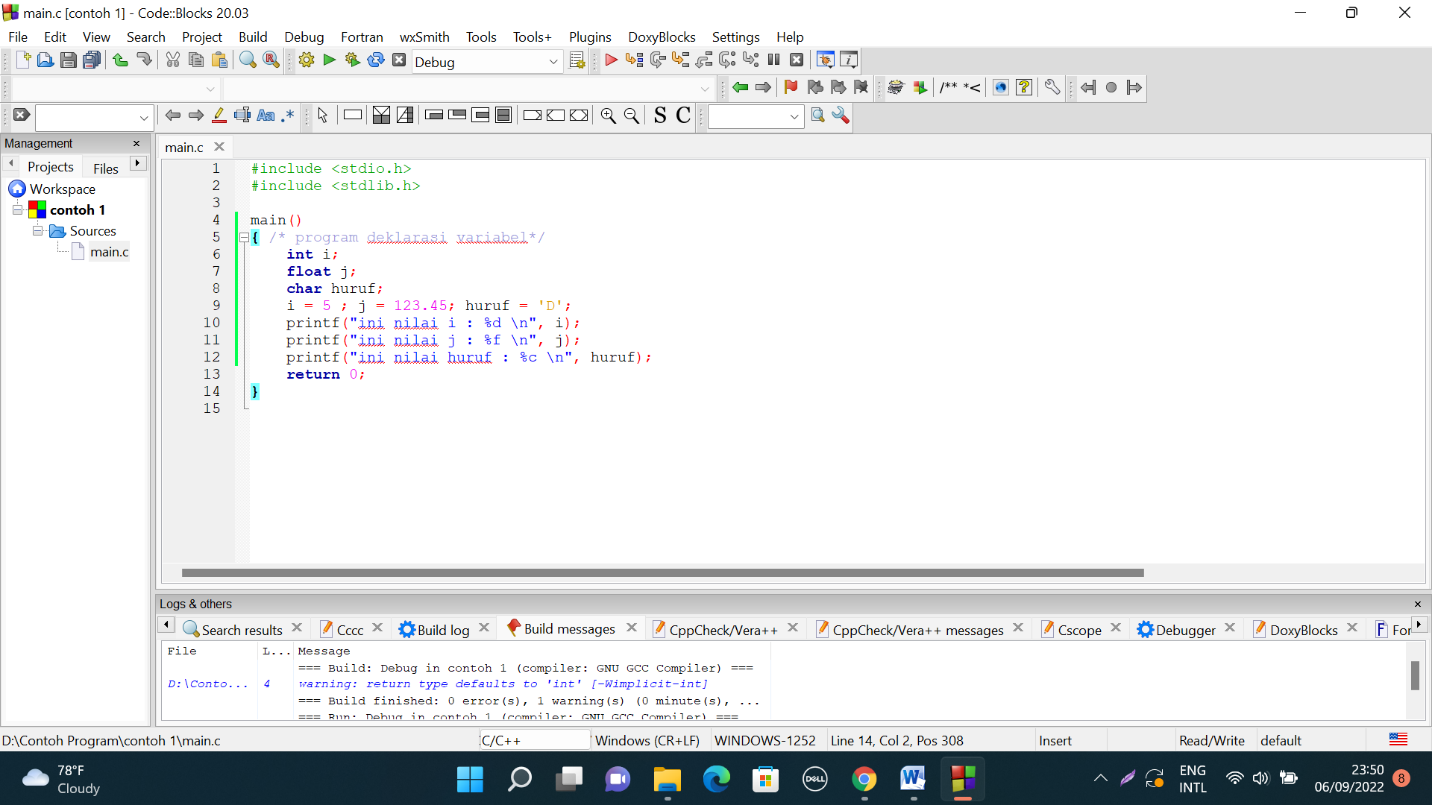
**CONTOH PROGRAM**

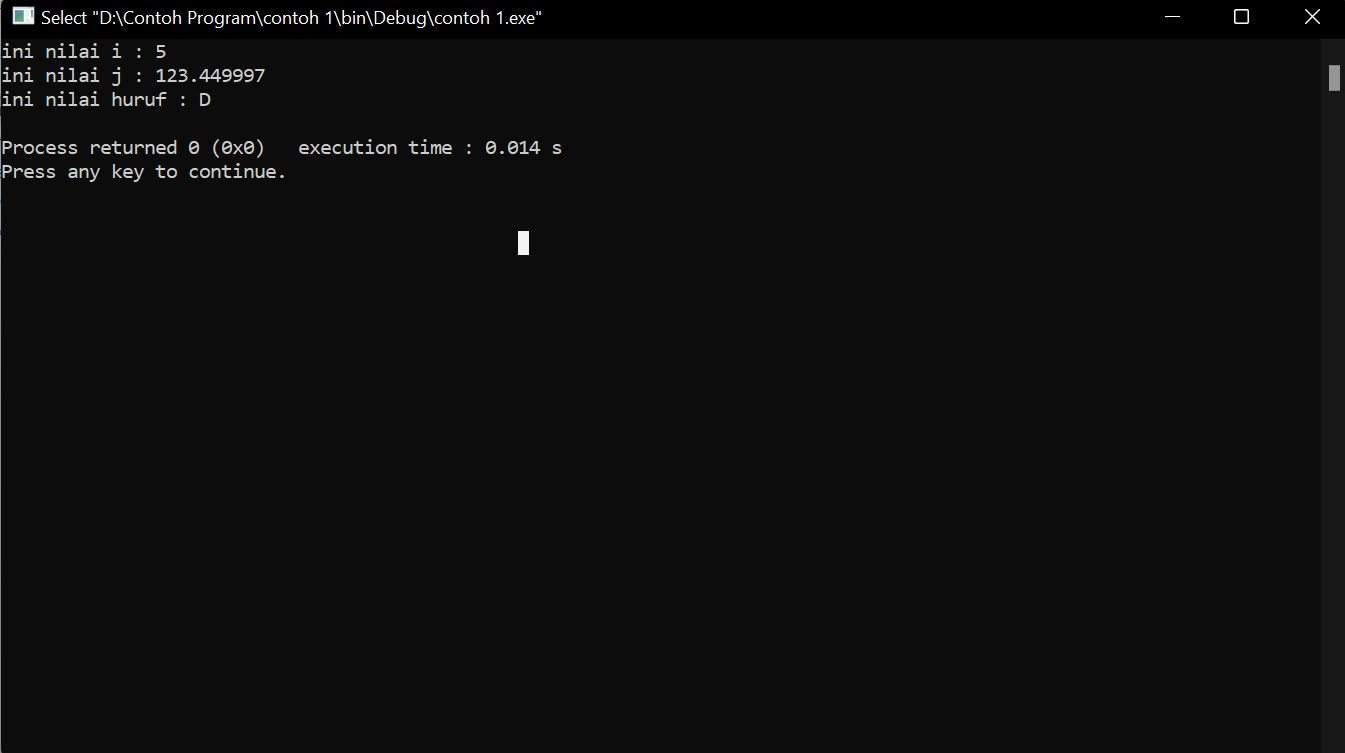
**3.2 Program Pertama**

****

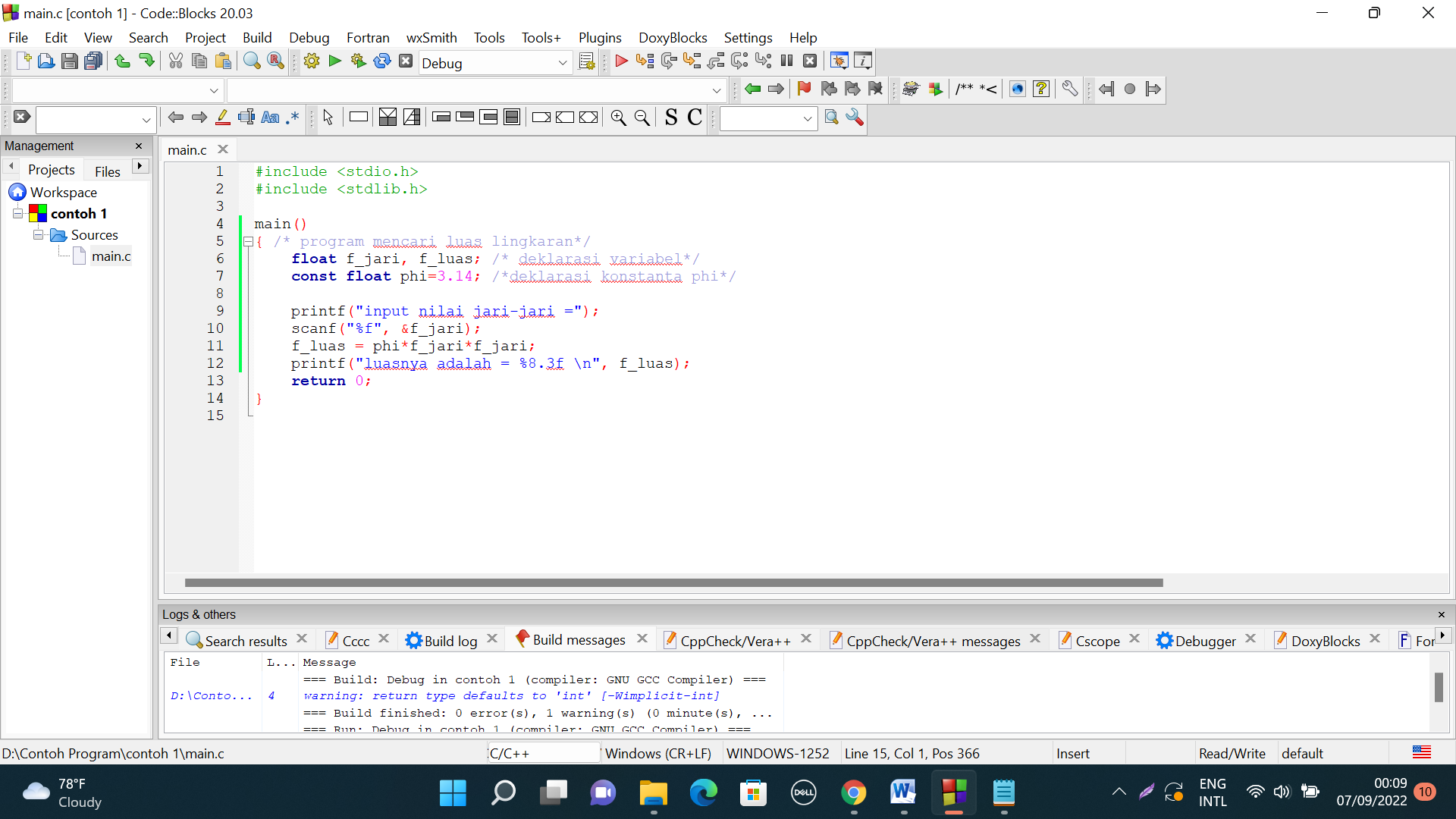
****

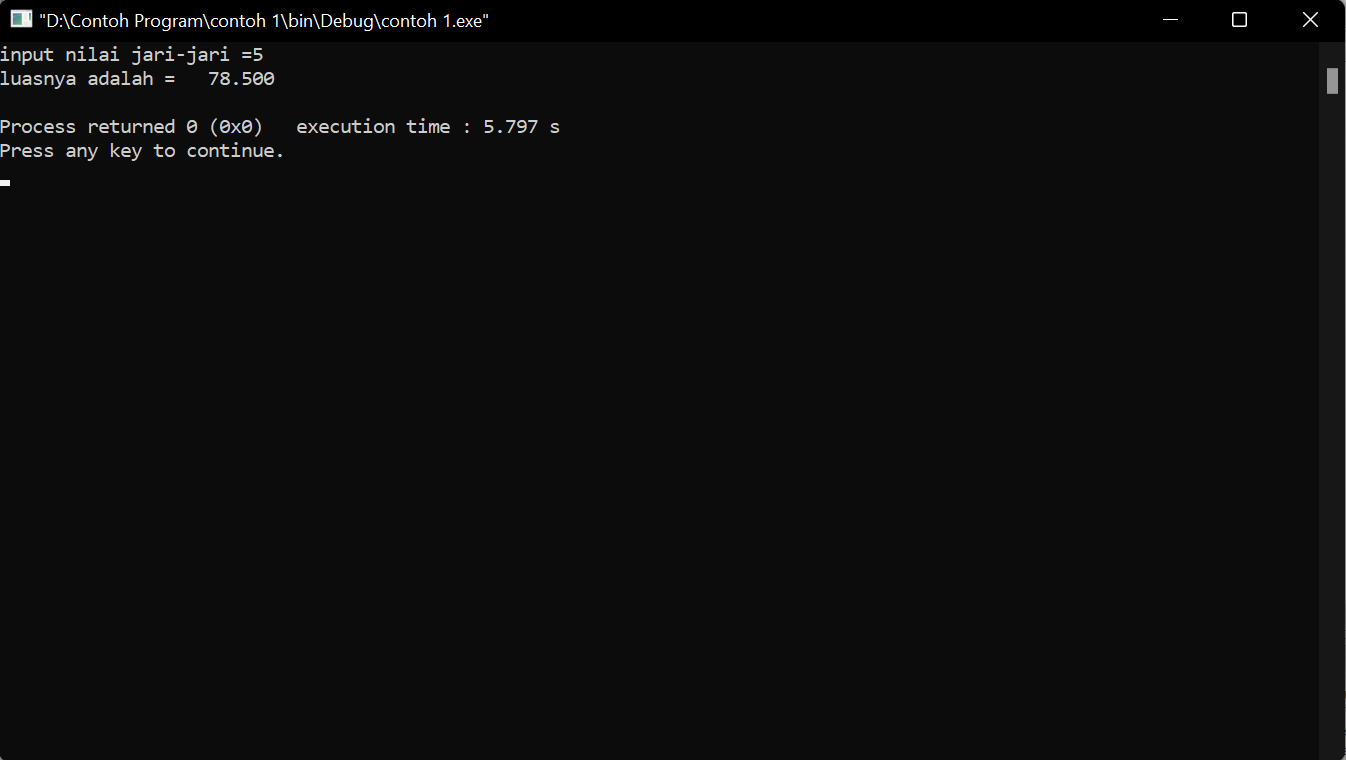
**3.3 Program Kedua**

****

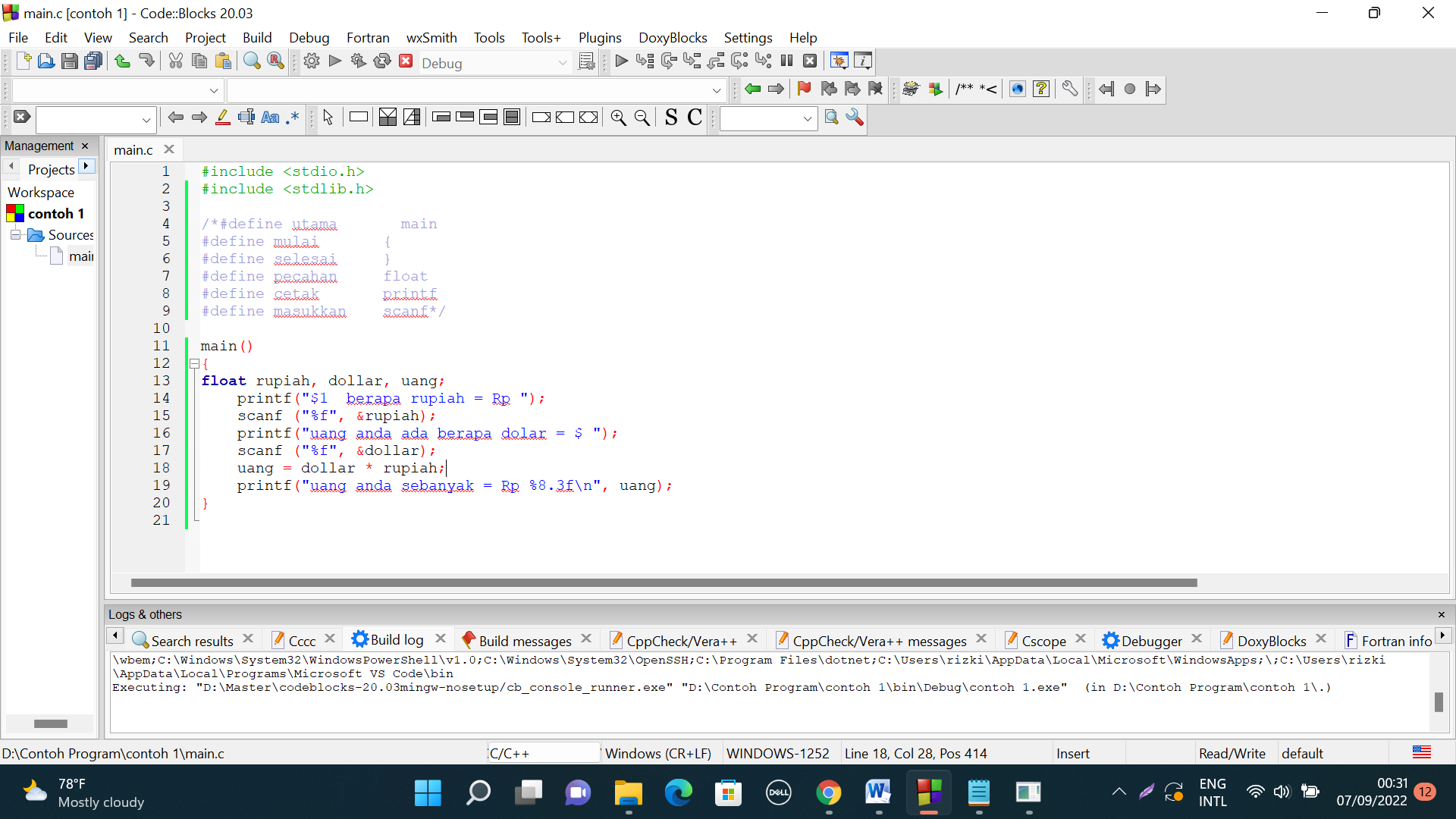
****

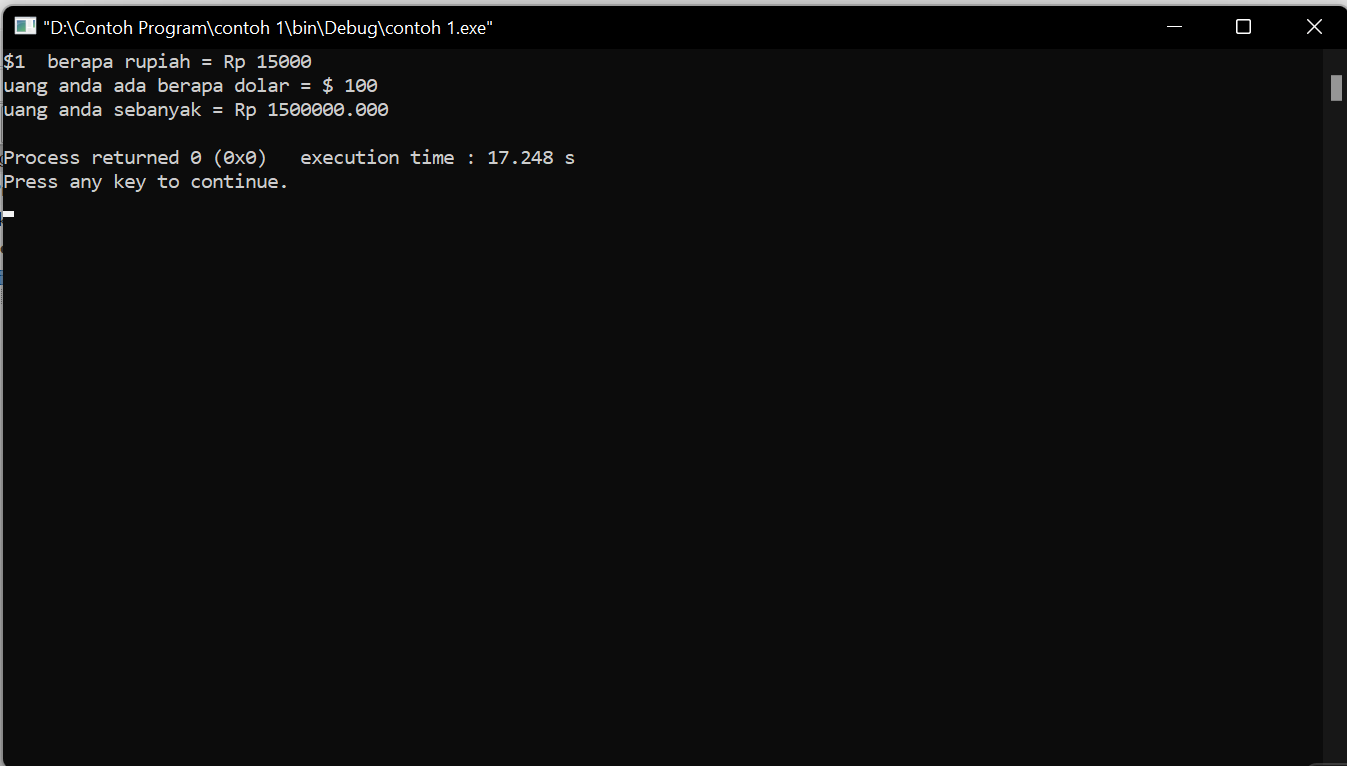
**3.4 Program Ketiga**

****

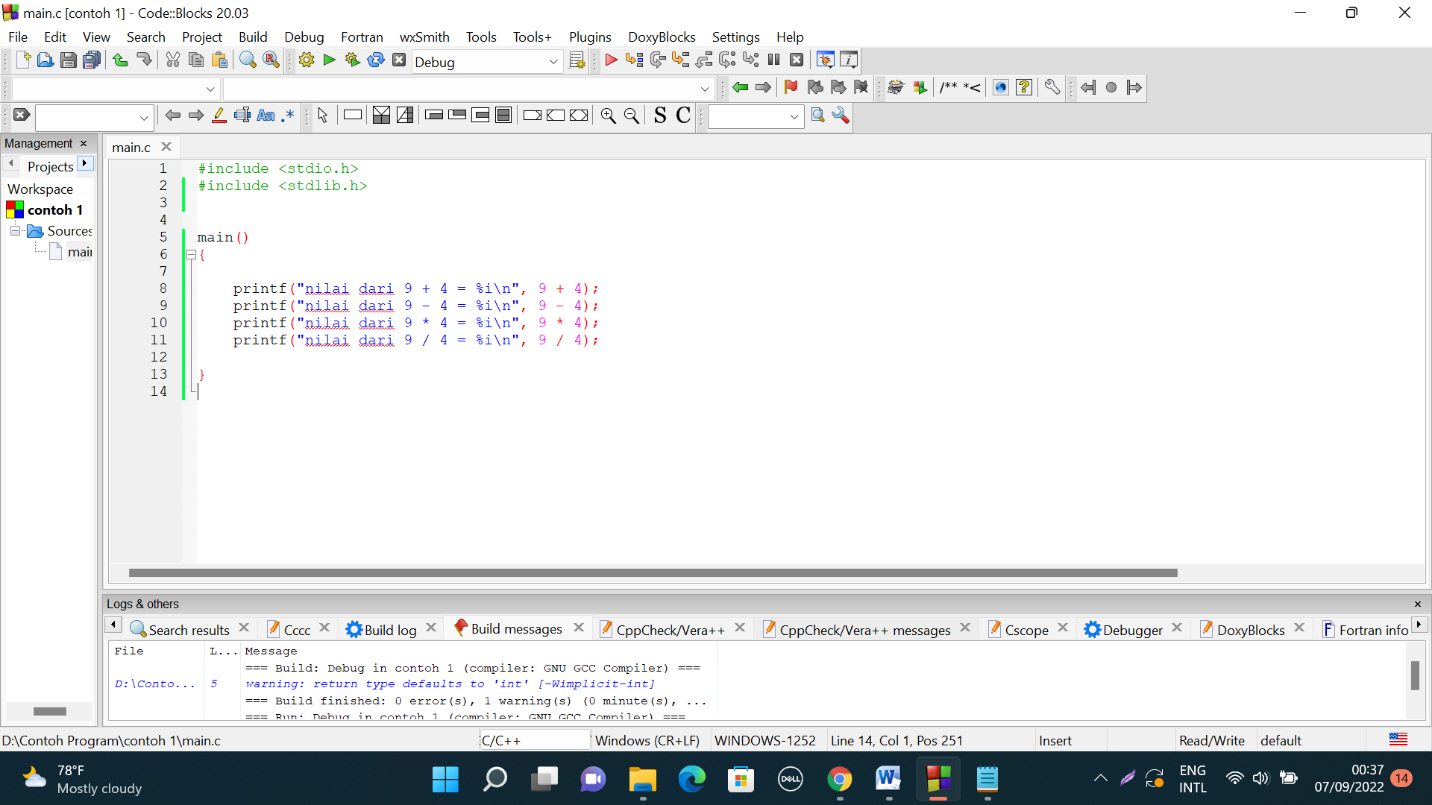
****

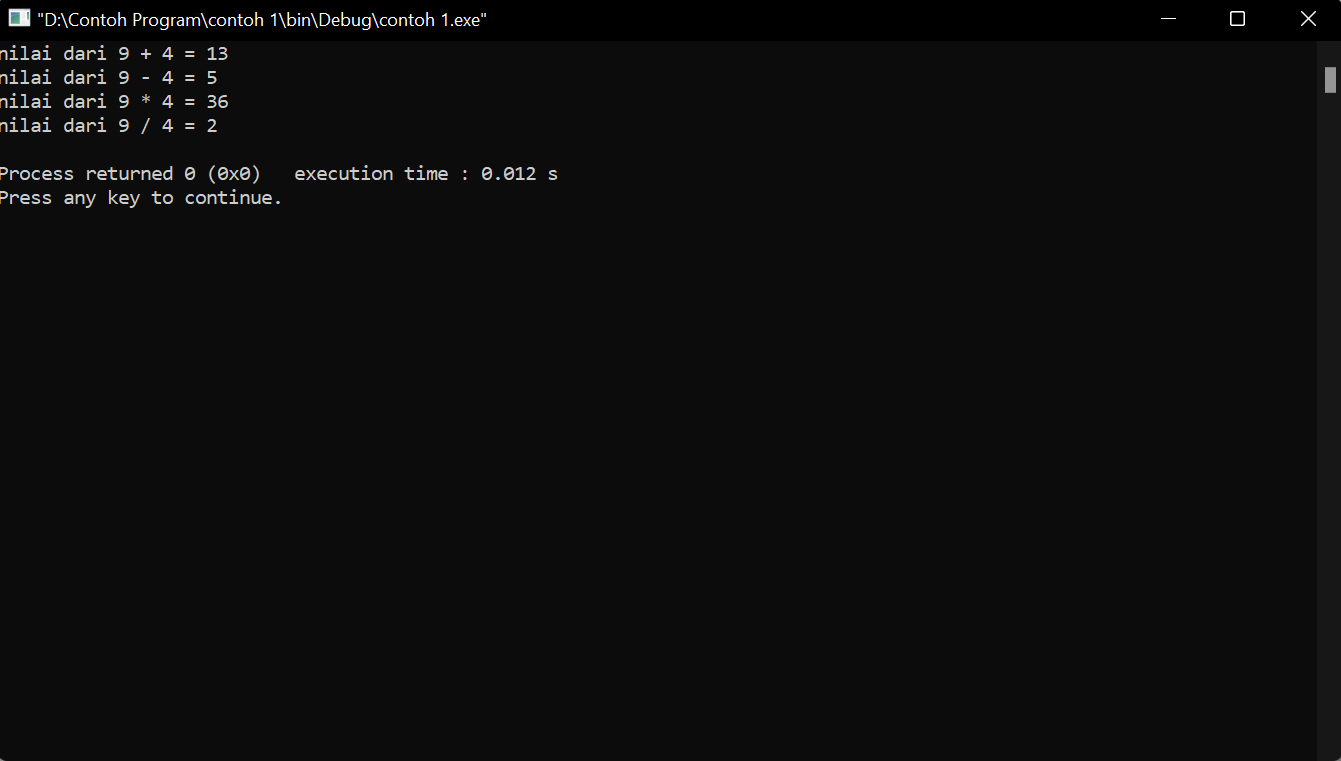
**3.5 Program Keempat**

****

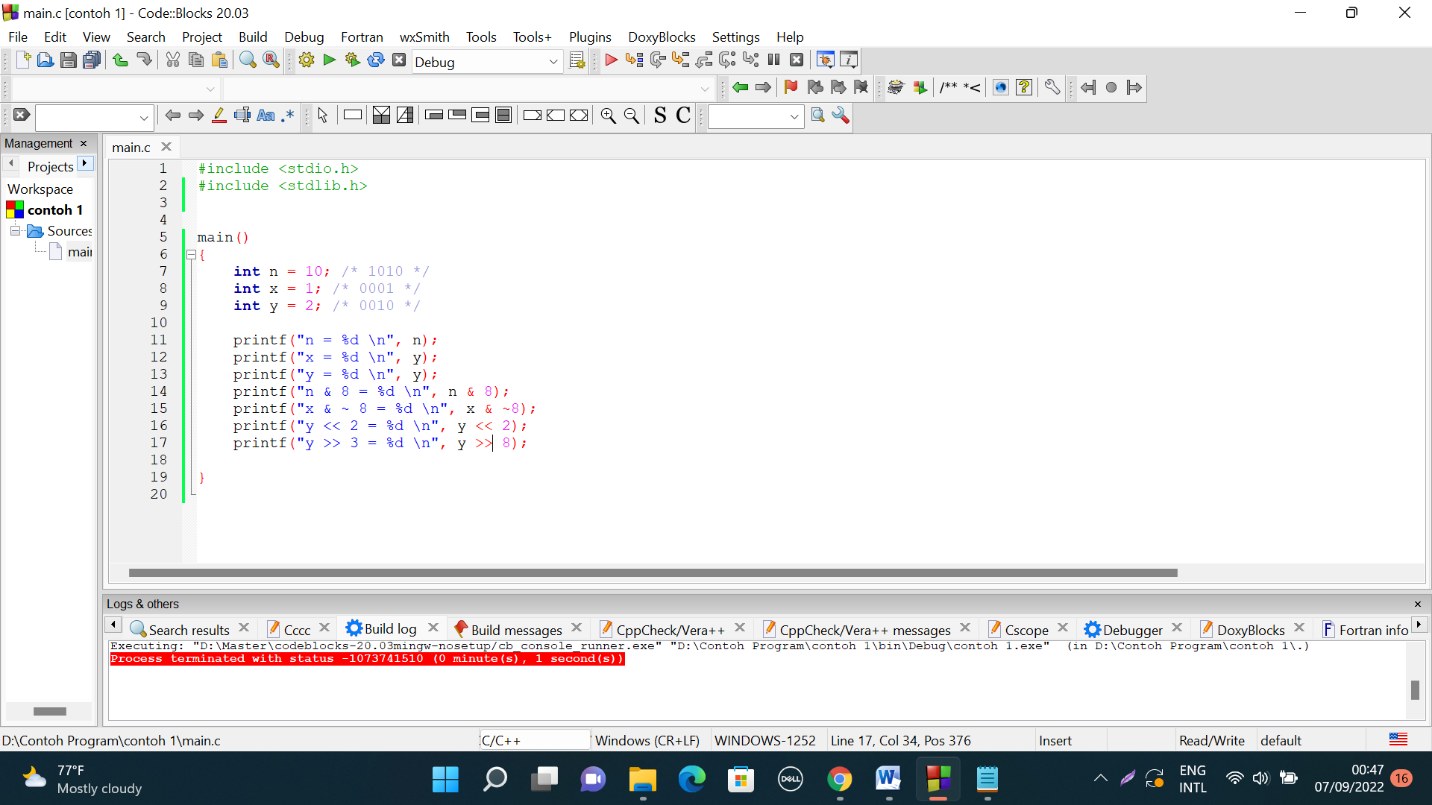
****

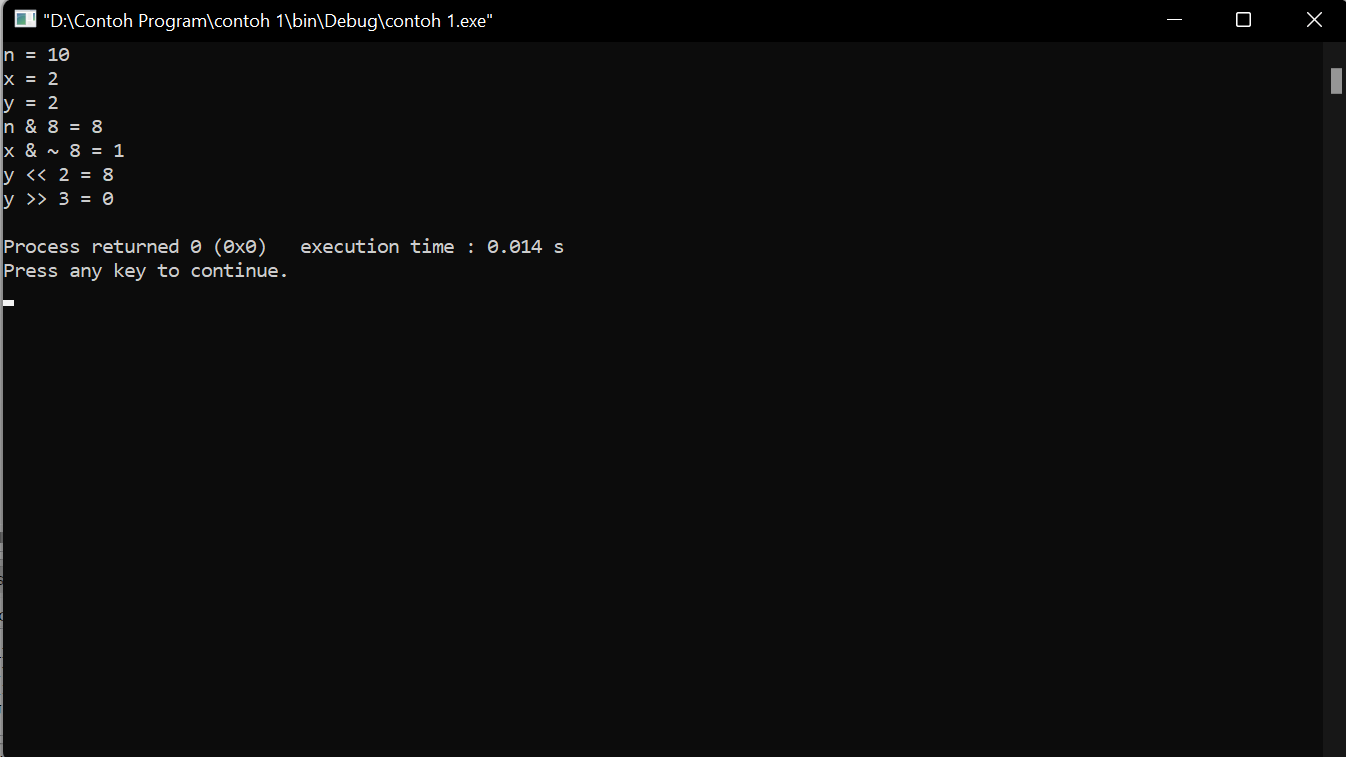
**3.6 Program Kelima**



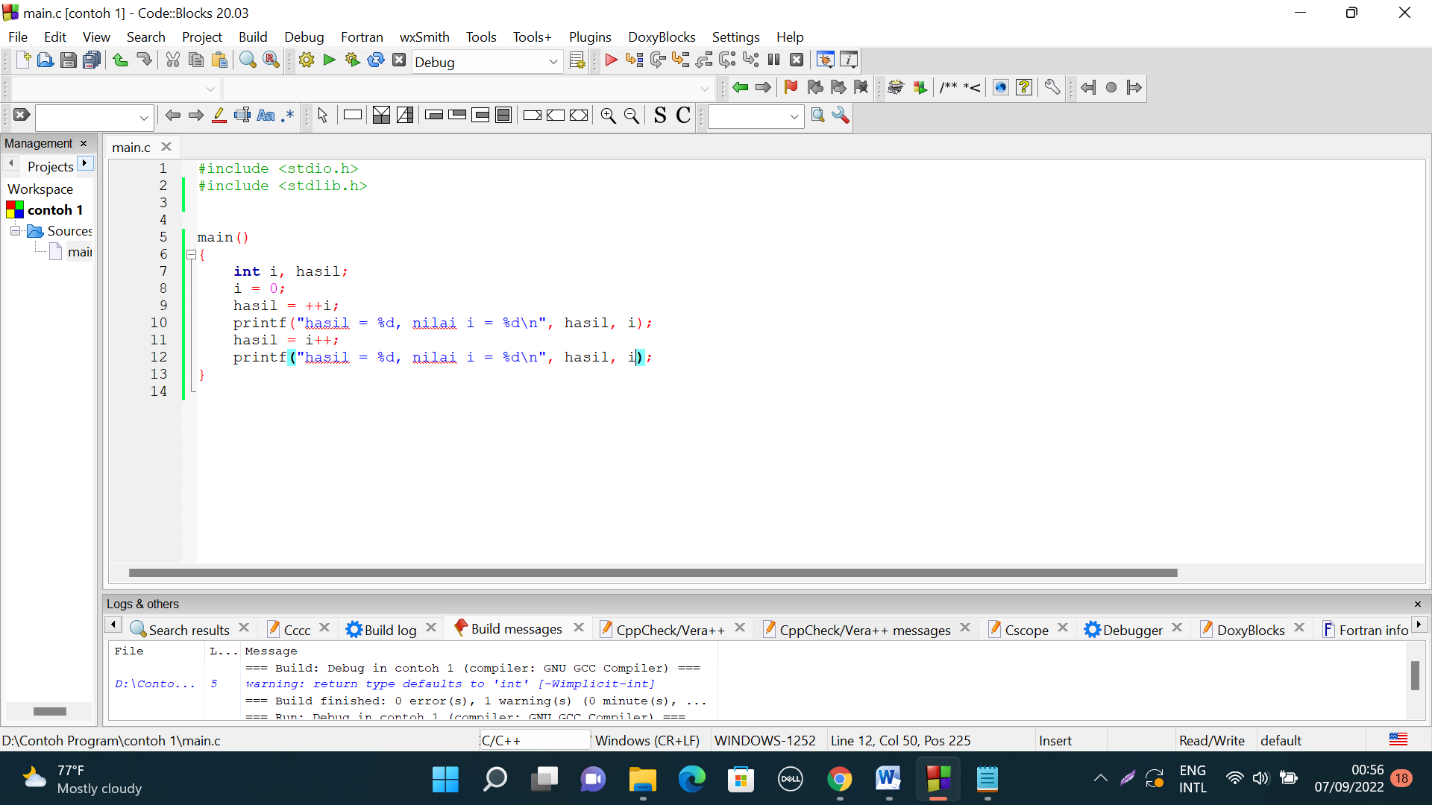


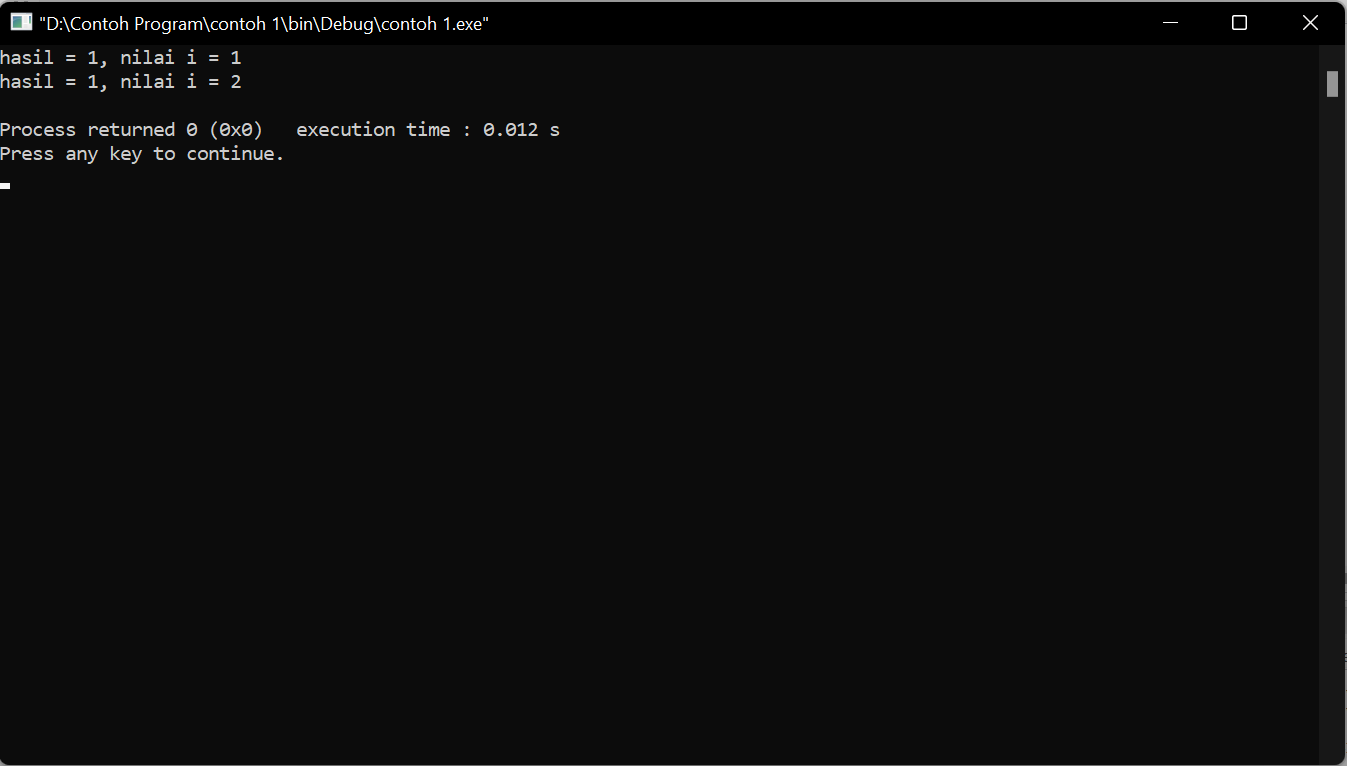
**3.7 Program Keenam**





**3.8 Program Ketujuh**





**BAB V**

**PENUTUP**

Demikianlah Makalah ini saya buat dengan sebaik-baiknya. Terima kasih buat Orang tua yang selalu mendukung saya baik materil maupun moril,dan terima kasih untuk abang dan kakak assisten laboratorium paket applikasi, juga terima kasih buat kawan-kawan yang telah membantu dalam mengerjakan makalah ini.semoga makalah ini dapat berguna buat kita semua dan buat angkatan ke depannya kelak.

lebih dan kurangnya saya mohon maaf, kepada abang dan kakak saya ucapkan terima kasih

Wassalammu’alaikum Wr.Wb.

DAFTAR PUSTAKA

[1]<http://belajarhardwaredipesat.blogspot.com/2019/08/hai-hai-guys-welcome-to-my-blog-2-i.html>

[2]<https://id.wikipedia.org/wiki/C_(bahasa_pemrograman)#:~:text=Bahasa%20pemrograman%20C%20adalah%20bahasa,mencerminkan%20kemampuan%20CPU%20yang%20ditargetkan>

[3]<https://www.novriadi.com/apa-itu-pengertian-bahasa-c/>