MAKALAH PENGKONDISIAN DALAM BAHASA

PEMROGRAMAN C

****

Oleh :

Diki Candra

NIM 2022903430010

D4 TEKNOLOGI REKAYASA KOMPUTER JARINGAN

POLITEKNIK NEGERI LHOKSEMAWE

2022/2023

DAFTAR ISI

**BAB 1**

Pendahuluan

A. Later Belakang

B. Rumusan Masalah

C. Tujuan

D. Manfaat

**BAB II**

Pembahasan

A. Pengenalan

B. Perulangan for

C. Perulangan while

D. Perulangan Do While

E. Nested Loop

**BAB III**

Penutup

BAB I

PENDAHULUAN

# A. Latar Belakang

Looping atau perulangan merupakan sebuah struktur perintah yang seringdigunakan pada sebuah aplikasi atau program. Perulangan juga sangat bergunapada program karena dapat kita pakai dalam struktur-struktur rumit yang besaryang memiliki kumpulan algoritma yang kompleks.

Untuk mencacah berapa kali perulangan dilakukan, diperlukan suatu variabelpencacah yang bertipe integer. Variabel tersebut akan bertambah nilainya setiapkali perulangan dilakukan. Konstruksi while digunakan untuk melakukanperulangan terhadap baris kode tertentu selama suatu kondisi terpenuhi. Jikakondisi sudah tidak terpenuhi, maka program akan keluar dari perulangan.Konstruksi repeat digunakan untuk melakukan perulangan sampai suatu kondisiterpenuhi.

Hampir setiap program mutlak memerlukan suatu perulangan danpercabangan. Tujuan perulangan disini adalah untuk mengulang statementberulang kali sesuai jumlah yang ditentukan pemakai. Program perulangan terbagiatas beberapa bagian yang digunakan sesuai dengan kebutuhan. Setiap perulanganakan memberikan output sesuai dengan algoritma yang dikehendaki.

# B. Rumusan Masalah

Berdasarkan judul dari makalah ini, Saya merumuskan masalah yaitu perulangan dalam bahasa pemrograman bahasa C!.

# C. Tujuan

Mengetahui perulangan pada Bahasa C*.*

# D. Manfaat

1. Pembaca

A. Menjadikan pembaca makalah semata-mata agar pembaca memahami dasar-dasar pemrograman.

B. Untuk memotivasi pembaca untuk membuat makalah yang lebih baik dari Ini.

2.Pengaramg

C. Untuk memberikan semangat pada mata kuliah yang sedang diambil saat ini.

D. Sebagai titik awal untuk melakukan penelitian agar penelitian selanjutnya dapat dilakukan dengan lebih baik lagi

BAB II

PEMBAHASAN

A. Pengenalan

Dalam bahasa C (dan juga bahasa turunan C seperti C++, PHP dan Java), terdapat 3 buah struktur perulangan atau looping, yakni perulangan for, perulangan while dan perulangan do while. Struktur perulangan (atau dalam bahasa inggris disebut dengan loop) adalah instruksi kode program yang bertujuan untuk mengulang beberapa baris perintah.

Dalam merancang perulangan, kita setidaknya harus mengetahui 3 komponen:

* Kondisi awal perulangan.
* Kondisi pada saat perulangan.
* Kondisi yang harus dipenuhi agar perulangan berhenti.

Berikut format dasar struktur perulangan for dalam bahasa C:

for (start; condition; increment)

{

// kode program

// kode program

}

Statement yang termasuk Control statement adalah :

* for digunakan untuk membuat perulangan yang dikontrol initial (nilai awal), condition (kondisi), increment decrement (penambahan/penurunan).
* while digunakan juga untuk perulangan seperti statement for(), tetapi statement while() memeriksa kondisi dulu, bila kondisi memenuhi (benar), maka perulangan dilaksanakan.
* do-while cara kerja dari statement do-while adalah langsung mengerjakan badan loop, kemudian kondisi diperiksa bila kondisi memenuhi perulangan dilanjutkan bila tidak peulangan selesai.

Start adalah kondisi pada saat awal perulangan. Biasanya kondisi awal ini berisi perintah untuk memberikan nilai kepada variabel counter. Variabel counter sendiri adalah sebuah variabel yang akan menentukan berapa banyak perulangan dilakukan. Kebanyakan programmer menggunakan variabel i sebagai variabel counter (ini tidak harus).

Condition adalah kondisi yang harus dipenuhi agar perulangan berjalan. Selama kondisi ini terpenuhi, maka compiler bahasa C akan terus melakukan perulangan. Misalnya condition ini berisi perintah i < 5, maka selama variabel counter i berisi angka yang kurang dari 5, maka lakukan perulangan.

Increment adalah bagian yang dipakai untuk memproses variabel counter agar bisa memenuhi kondisi akhir perulangan. Bagian ini akan selalu di eksekusi di setiap perulangan.

Disebut increment karena biasanya berisi operasi increment seperti i++, yang sama dengan i = i + 1. Maksudnya, dalam setiap perulangan naikkan variabel i sebanyak 1 angka. Namun kita juga bisa memberikan nilai lain, misalnya i = i + 2, sehingga variabel counter akan naik 2 angka setiap perulangan.

Sebagai tambahan, terdapat istilah iterasi (iteration), yang berarti 1 kali perulangan. Istilah ini cukup sering dipakai ketika membahas tentang struktur perulangan.

B. Perulangan For

Apa yang akan kita lakukan jika ingin membuat program yang berulang? Apakah akan menulis manual menggunakan kode printf? Atau kita akan menggunakan looping? Kode printf memang diperuntukan untuk membuat program yang berulang. Namun, apakah kita akan menggunakan kode printf jika kita akan mengulanginya hingga 1000 kata? Seperti ini.

#include <studio.h>

Void main(){

Printf (“Perulangan dalam bahasa C”);

Printf (“Perulangan dalam bahasa C”);

Printf (“Perulangan dalam bahasa C”);

Printf (“Perulangan dalam bahasa C”);

Printf (“Perulangan dalam bahasa C”);

Printf (“Perulangan dalam bahasa C”);

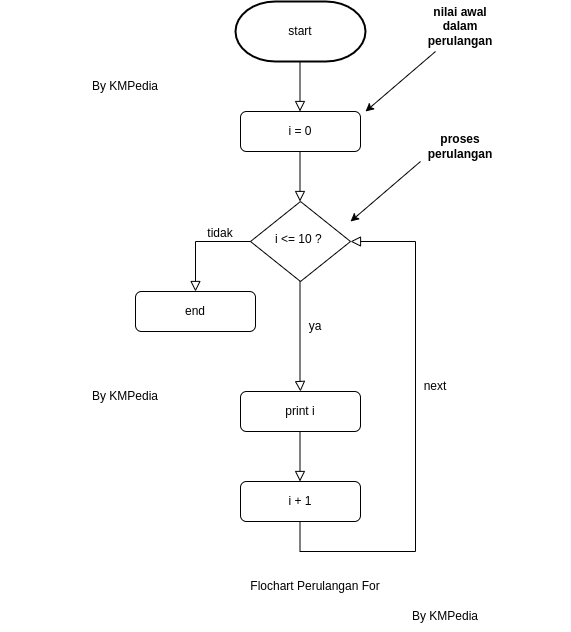
}

Jika kita menulis manual seperti contoh diatas maka, akan membutuhkan banyak waktu. Jadi, untuk mengefisiensikan waktu, kita akan menggunakan looping atau perulangan. Lalu, apakah yang dimaksud dengan looping?

**Pengertian Perulangan atau Looping**

Struktur perulangan atau loop merupakan intruksi program untuk mengulang perintah. Sedangkan, Looping atau sering dikenal dengan perulangan merupakan proses menjalankan statement/ perintah yang dilakukan berulang-ulang dalam batas yang telah kita tentukan didalam program. Perulangan tersebut diperuntukan untuk memudahkan seorang programmer dalam membuat sintaks yang diinginkan.

Perulangan for digunakan untuk membuat perulangan yang memiliki sintaks dasar seperti nilai awal, kondisi, dan penambahan ataupun pengurangan.



Sintaks dasar dari for dalam bahasa C :

For (eks1; eks2; eks3)

{

// kode program

//kode program

}

Dimana :

* Eks1 : ekspresi untuk nilai awal perulangan. Biasanya digunakan untuk memberikan nilai kepada variable counter. Variable counter merupakan variable yang menentukan berapa banyak perulangan yang akan dijalankan.
* Eks2 : ekspresi untuk kondisi. Kondisi harus dipenuhi agar perulangan bisa berjalan. Jika kondisi ini terpenuhi makan compiler bahasa C akan melakukan perulangan.
* Eks3 : ekspresi untuk increment (penambahan) atau decrement (pengurangan). Bagian ini digunakan untuk memproses variable counter agar bisa menghasilkan akhir dari proses perulangan. Bagian ini yang akan selalu di jalankan pada setiap perulangan.

**Contoh program perulangan for : 1**

#include <studio.h>

int main ()

{

int i ;

/\*for\*/

for ( i = 0; <=10; i = i + 1){

printf (“perulangan bahasa C %i \n”, i);

}

Return 0;

}

Hasil kode program :

perulangan bahasa C 0

perulangan bahasa C 1

perulangan bahasa C 2

perulangan bahasa C 3

perulangan bahasa C 4

perulangan bahasa C 5

perulangan bahasa C 6

perulangan bahasa C 7

perulangan bahasa C 8

perulangan bahasa C 9

perulangan bahasa C 10

**Contoh program perulangan for : 2**

for dalam for untuk mengulang 5 angka yang berbeda sebanyak 5 kali.

#include<stdio.h>

int main()

{

int i,j;

for (i=1;i<=5;i++)

{

for (j=1;j<=5;j++)

{

printf("%d",i);

}

printf("\n");

}

}

hasilnya seperti ini :

11111

22222

33333

44444

55555

**Contoh program perulangan for : 3**

for dalam for untuk mengulang karakter yang sama.

#include<stdio.h>

int main()

{

int i,j;

for (i=1;i<=5;i++)

{

for (j=1;j<=5;j++)

{

printf("\*");

}

printf("\n");

}

}

hasilnya seperti ini :

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

**Contoh program perulangan for : 3**

for dalam for mengulang pasangan angka

#include<stdio.h>

int main()

{

int i,j;

for (i=1;i<=5;i++)

{

for (j=1;j<=5;j++)

{

printf("(%d,%d)",i,j);

}

printf("\n");

}

}

hasilnya seperti ini :

(1,1)(1,2)(1,3)(1,4)(1,5)

(2,1)(2,2)(2,3)(2,4)(2,5)

(3,1)(3,2)(3,3)(3,4)(3,5)

(4,1)(4,2)(4,3)(4,4)(4,5)

(5,1)(5,2)(5,3)(5,4)(5,5)

**Contoh program perulangan for : 4**

for dalam for untuk membuat setengah piramid terbalik dengan spasi

#include<stdio.h>

int main()

{

int i,j,k;

for (i=5 ; i>=1 ; i--)

{

for (k=i ; k<=4 ; k++)

{

printf(" ");

}

for (j=i ; j>=1 ;j--)

{

printf("%d",j);

}

printf("\n");

}

}

hasilnya seperti ini :

hasilnya seperti ini :

54321

4321

321

21

1

**Contoh program perulangan for : 5**

for dalam for untuk membuat setengah piramid dengan banyaknya sesuai dengan angka inputan.

#include<stdio.h>

int main()

{

int i,j,k;

int n;

printf("input n : ");

scanf("%d",&n);

for (i=n ; i>=1 ; i--)

{

for (k=n ; k>=i ; k--)

{

printf(" ");

}

for (j=i ; j>=1 ;j--)

{

printf("%d",j);

}

printf("\n");

}

}

hasilnya seperti ini :

input n : 5

54321

4321

321

21

1

keterangan : Jadi program ini dituliskan pada **codeblock** bisa ya teman-teman, kita akan jelaskan satu persatu maksud dari contoh kode program diatas.

* [#include](https://www.kmtech.id/kmpedia/hashtags/include) <studio.h>

[#include](https://www.kmtech.id/kmpedia/hashtags/include) <studio.h> merupakan salah satu *library* yang terdapat dalam bahasa C, kode tersebut wajib di ikutsertakan karena digunakan untuk *input* (masukan) dan *output* (pengeluaran). Jika tidak menuliskan kode program tersebut maka tidak akan menampilkan atau meng*input*kan sesuatu.

* int main ()

int merupakan tipe data yang akan di*eksekusi*, sedangkan main () wajib berpasangan atau disertakan return 0 yang berfungsi untuk mengembalikan nilai 0 setelah di*eksekusi.*

* int i

int adalah salah satu tipe data dalam bahasa C, int berarti *interger* yang menunjukan bilangan bulat dan menggunakan %i ataupun %d. sedangkan i merupakan *variable* tambahan, itu bisa diganti dengan huruf abjad lainnya.

* for ( i = 0; <=10; i = i + 1){

Pada kondisi di atas yang perlu diperhatikan didalam kurung setelah kata for ialah:

- Hitungan akan dimulai dari 0, karena menggunakan( i = 0). Kondisi ini akan memulai perulangan dan hasil program dari 0. Contoh : perulangan bahasa C 0

- Hitungannya akan sampai berapa? Nah, disini program yang akan mengeksekusi akan sampai berapa program diulang. Jadi sintaks nya adalah (<=10). Maksud dari kurang dari samadengan sepuluh (<=10) yaitu program yang dijalankan akan diulang atau akan looping hingga samadengan 10.

- Lalu disetiap looping atau perulangan i akan bertambah +1 (i + 1). Karena kita mencoba menggunakan increment (penambahan) jadi, program akan melakukan perulangan yang menambah 1 angka.

* printf (“perulangan bahasa C %i \n”, i);

printf merupakan hasil output yang akan tertamil pada cmd (command prompt) ataupun serial monitor. Kemudian, pada kondisi di atas yang perlu diperhatikan didalam kurung setelah kata printf ialah:

- “perulangan bahasa C”

“perulangan bahasa C”, apa yang ingin kita tampilan itulah di*input*kan dalam printf.

- %i, merupakan istilah dari *interger.*

- \n, digunakan untuk membuat baris baru. Maksudnya tulisan *output* akan tertampil memanjang kebawah, jika tidak menggunakan \n tidak masalah, akan tetapi *output* yang dihasilkan akan lurus kesamping.

Itulah pembahasan perulangan for dalam bahasa C, memudahkan dalam membuat suatu program jika kita menggunakan looping.

C. Perulangan While

Pada artikel sebelumnya, kita telah mempelajari perulangan *for* dalam bahasa C. Dimana perulangan *for*, membantu dan mengefisiensikan waktu kita dalam mengerjakan sebuah *sketch*. Kemudian dalam *statement* perulangan atau biasa disebut *looping*, ada beberapa perulangan yaitu *while*. Sebenarnya apa yang dimaksud dengan perulangan *while.*

Pengertian Perulangan atau Looping

Seperti yang telah dijelaskan pada artikel sebelumnya, struktur perulangan atau loop merupakan instruksi program untuk mengulang perintah. Sedangkan, looping atau sering dikenal dengan perulangan merupakan sebuah proses menjalankan statement / perintah yang dilakukan berulang-ulang dalam batas yang telah ditentukan dalam program. Perulangan tersebut diperuntukan untuk memudahkan seseorang programmer dalam membuat syntaks yang diinginkan.

Selain dalam bahasa C, perulangan tersebut juga bisa digunakan untuk bahasa turunan C seperti C++, PHP, dan Java. Ada empat macam perulangan dalam bahasa C, kemudian dikelompokan menjadi dua yaitu :

1. Counted Loop, merupakan perulangan yang sudah pasti berulang dan sudah jelas berapa kali berulangnya.

2. Uncounted Loop, merupakan perulangan yang tidak pasti statement tersebut berulang berapa kali.

Instruksi while digunakan untuk membuat statement berulang selama kondisi syarat diberikan bernilai benar. Jika kondisi bernilai syarat maka statement tidak akan diulang. Perulangan while termasuk uncounted loop.

Syntaks dari Perulangan While :

while(condition) {

statement(s);

}

Dalam perulangan for, ketiga syarat statement ditulis menjadi satu kesatuan, berbeda dengan perulanggan while.

Dimana perulangan for syntaks nya adalah:

for (i=0; i<=10; i ++)

Sedangkan syntaks dari perulangan while ketiga syarat tersebut saling berpisah. Seperti :

while(condition) {

statement(s);

}

Instruksi while bisa memiliki statement lebih dari satu. Perulangan tersebut akan terus berjalan selama perintah atau statement dalam kondisi yang benar. Umumnya, perulangan while menggunakan operator perbandinngan.

**Contoh Kode Program Perulangan While dalam Bahasa C**

Contoh 1:

#include <stdio.h>

int main (void)

{

int i = 1;

while ( i>=7 ){

printf(“perulangan while dalam bahasa C \n”);

i++;

}

return 0;

}

Hasil kode program jika dijalankan :

perulangan while dalam bahasa

perulangan while dalam bahasa

perulangan while dalam bahasa

perulangan while dalam bahasa

perulangan while dalam bahasa

perulangan while dalam bahasa

perulangan while dalam Bahasa

Contoh 2:

Sebagai praktek pertama, berikut kode program perulangan WHILE untuk menampilkan teks “Hello World” sebanyak 5 kali:

#include <stdio.h>

int main(void)

{

int i = 1;

while (i <= 5){

printf("Hello World \n");

i++;

}

return 0;

}

Hasil kode program:

Hello World

Hello World

Hello World

Hello World

Hello World

Di baris 5 terdapat perintah untuk menginput angka 1 ke dalam variabel i. Nantinya, variabel i ini akan menjadi variabel counter yang dipakai untuk menentukan jumlah perulangan.

Proses perulangan di mulai di baris 6. Perintah while (i <= 5) artinya, selama nilai variabel i kurang atau sama dengan 5, maka jalankan perulangan.

Di dalam blok perulangan terdapat perintah printf(“Hello World \n”) di baris 7. Ini dipakai untuk menampilkan teks “Hello World“. Kemudian di baris 8 terdapat perintah increment, yakni i++. Perintah ini akan menaikkan nilai variabel i sebanyak 1 angka dalam setiap iterasi.

Perulangan while di atas akan di ulang sebanyak 5 kali, mulai dari i = 1, i = 2, i = 3, i = 4, hingga i = 5. Ketika nilai variabel counter i sudah mencapai 6, maka kondisi while (i <= 5) tidak terpenuhi lagi (false), sehingga perulangan berhenti.

Contoh 3:

Salah satu hal yang harus selalu diingat ketika membuat perulangan while adalah, jangan lupa membuat perintah increment. Jika tidak, kondisi akhir tidak akan pernah terpenuhi dan perulangan akan berjalan terus menerus. Ini dikenal dengan istilah infinity loop. Berikut contohnya:

#include <stdio.h>

int main(void)

{

int i = 1;

while (i <= 5){

printf("Hello World \n");

}

return 0;

}

Jika anda menjalankan kode program di atas, teks “Hello World” akan ditampilkan terus menerus, tidak pernah selesai. Penyebabnya karena kondisi while (i <= 5) akan selalu bernilai true. Di dalam blok perulangan tidak ada perintah yang bisa mengubah nilai variabel i agar kondisi while (i <= 5) bernilai false.

Untuk menghentikan infinity loop, tutup paksa jendela hasil dengan men-klik tanda (x) di sudut kanan atas, atau tekan kombinasi CTRL + C.

Hasilnya :



Contoh 4:

Sama seperti perulangan FOR, di dalam block perulangan WHILE kita juga bisa mengakses nilai dari variabel counter i:

#include <stdio.h>

int main(void)

{

int i = 1;

while (i <= 5){

printf("Hello World %i \n",i);

i++;

}

return 0;

}

Hasil kode program:

Hello World 1

Hello World 2

Hello World 3

Hello World 4

Hello World 5

Contoh 5:

Bagaimana dengan perulangan menurun? tidak masalah. Kita tinggal mengatur kondisi awal, kondisi akhir, serta proses decrement:

#include <stdio.h>

int main(void)

{

int i = 10;

while (i > 5){

printf("Hello World %i \n",i);

i--;

}

return 0;

}

Hasil kode program:

Hello World 10

Hello World 9

Hello World 8

Hello World 7

Hello World 6

Di sini saya mengisi nilai awal variabel counter i dengan angka 10. Kondisi perulangan adalah while (i > 5), artinya selama nilai variabel i di atas 5, jalankan perulangan. Dan karena kita ingin membuat perulangan menurun, maka dipakai perintah decrement i-- yang akan mengurangi nilai variabel i sebanyak 1 angka dalam setiap iterasi.

D. Perulangan Do While

Perulangan DO WHILE merupakan modifikasi dari perulangan WHILE, yakni dengan memindahkan posisi pemeriksaan kondisi ke akhir perulangan. Artinya, kita akan lakukan dulu sebuah perulangan, baru di akhir diperiksa apakah kondisi variabel counter sudah terpenuhi atau belum.

Berikut format dasar struktur perulangan DO WHILE dalam bahasa C:

start;

do

{

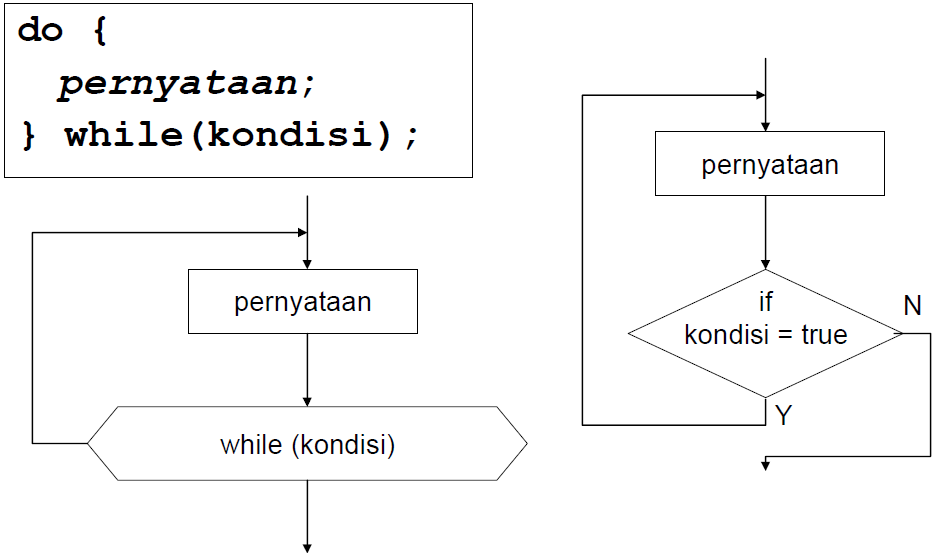
// kode program

// kode program

increment;

}

while (condition)



Jadi perbedaanya:

Perulangan do/while akan mengecek kondisi di belakang (sesudah mengulang), sedangkan while akan mencek kondisi di depan atau awal (sbelum mengulang).

Sama seperti perulangan WHILE, di bagian start biasanya terdapat perintah inisialisasi variabel counter, misalnya i = 0. Kemudian di dalam block do ditulis kode program yang akan di ulang, tidak lupa sebuah perintah untuk menaikkan nilai variabel counter, misalnya dengan perintah i++.

Di bagian paling bawah, terdapat perintah while (condition). Di sinilah kondisi perulangan akan diperiksa. Selama kondisi ini menghasilkan nilai true, maka perulangan akan lanjut ke iterasi berikutnya.

**Contoh Kode Program Perulangan Do While Bahasa C:**

Agar bisa dibandingkan dengan bahasan di perulangan FOR dan WHILE, berikut contoh kode program bahasa C untuk menampilkan 5 kali teks ‘Hello World’ menggunakan perulangan DO WHILE:

Contoh 1:

#include <stdio.h>

int main(void)

{

int i = 1;

do {

printf("Hello World \n");

i++;

}

while (i <= 5);

return 0;

}

Hasil kode program:

Hello World

Hello World

Hello World

Hello World

Hello World

Di baris 5 saya mendefinisikan variabel counter i dan memberikan nilai awal 1.

Kemudian terdapat block kode program do di baris 6 – 9. Inilah kode program yang akan di ulang, yakni berisi sebuah perintah printf untuk menampilkan teks ‘Hello World’, dan sebuah perintah increment i++ untuk menaikkan nilai variabel i sebanyak 1 angka.

Terakhir di baris 10 terdapat perintah while (i <= 5). Artinya, selama variabel counter i bernilai kurang dari atau sama dengan 5, jalankan kembali perulangan.

Contoh 2:

Berikut adalah contoh program penjumlahan menggunakan perulangan Do-While :

#include <stdio.h>

int main()

{

int i= 0, N, jumlah = 0;

do

{

printf("Masukkan Bilangan : ");

scanf("%d", &N);

jumlah += N;

printf("Jumlahnya adalah %d\n", jumlah);

i++;

} while (i<5);

return 0;

}

Hasilnya:

Masukkan Bilangan : 34

Jumlahnya adalah 34

Masukkan Bilangan : 98

Jumlahnya adalah 132

Masukkan Bilangan : 76

Jumlahnya adalah 208

Masukkan Bilangan : 87

Jumlahnya adalah 295

Masukkan Bilangan : 56

Jumlahnya adalah 351

Contoh 3:

Mencetak matriks identitas.

#include <stdio.h>

main()

{

int baris,kolom;

for(baris=1;baris<=5;baris++)

{printf(“\n”);

for(kolom=1;kolom<=5;kolom++)

{if (baris==kolom)

printf(“1 “);

else

printf(“0 “);

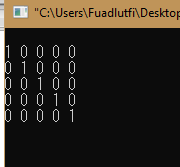
}

}

getch();

}

Hasilnya :



Contoh 4:

membuat tabel perkalian.

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

main()

{

int brs,klm,hasil, a;

printf(“X “);

for(a=1;a<=10;a++)

printf(“%5d “,a);

for(brs=1;brs<=10;brs++)

{printf(“\n \n%d “,brs);

for(klm=1;klm<=10;klm++)

{

hasil=brs\*klm;

printf(“%5d “,hasil);

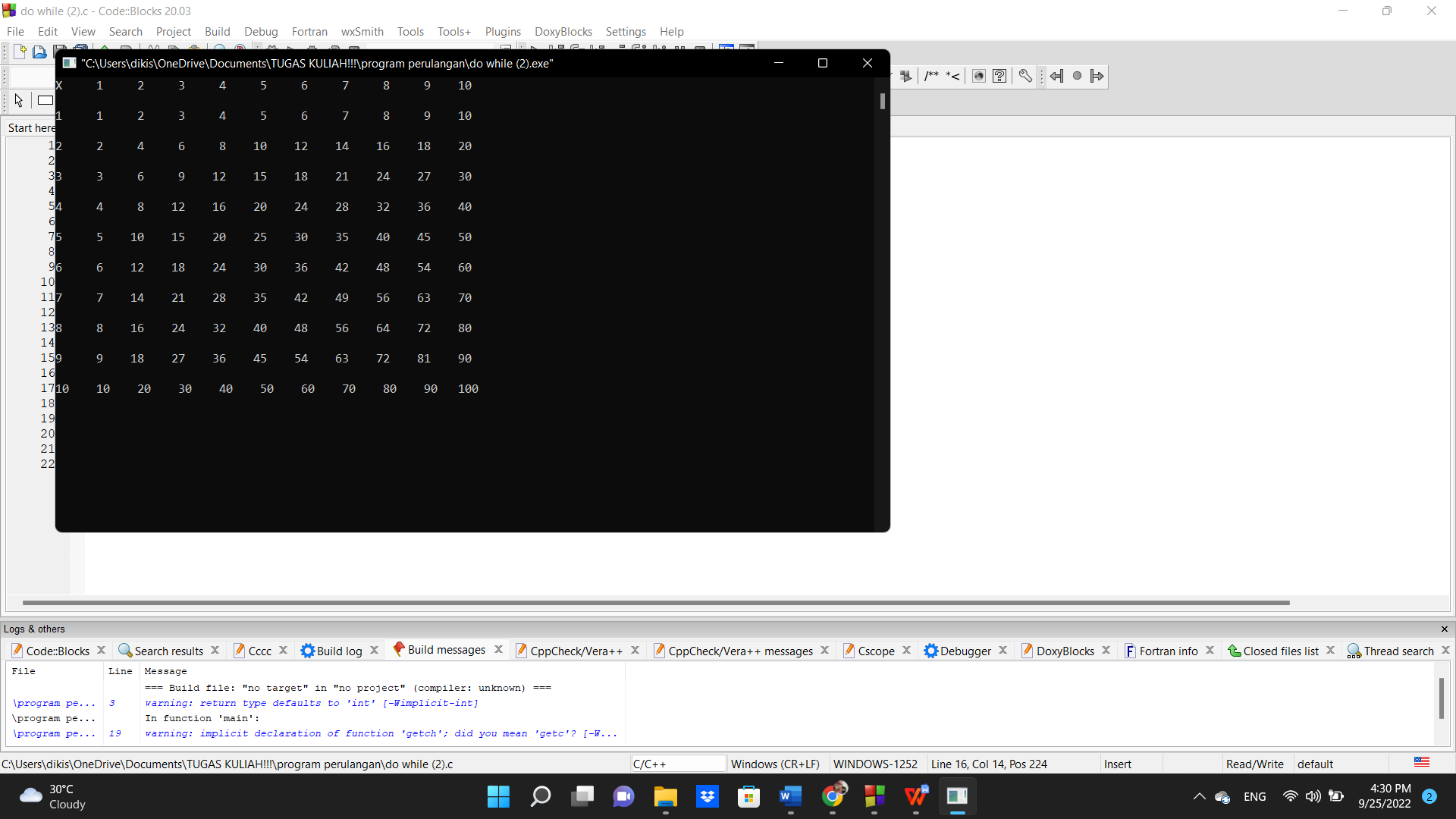
}

}

getch();

}

Hasilnya :



Contoh 5:

membuat tabel konversi temperature.

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

main()

{

int f,r,k, no,c;

printf(” No.”);

printf(” celsius”);

printf(” farenheit”);

printf(” reamur”);

printf(” kelvin”);

for(no=1,c=0;c<=100,no<=10;no++,c+=10)

{

printf(“\n\n%5d %5d”,no,c);

f=9\*c/5+32;

r=4\*c/5;

k=c+273;

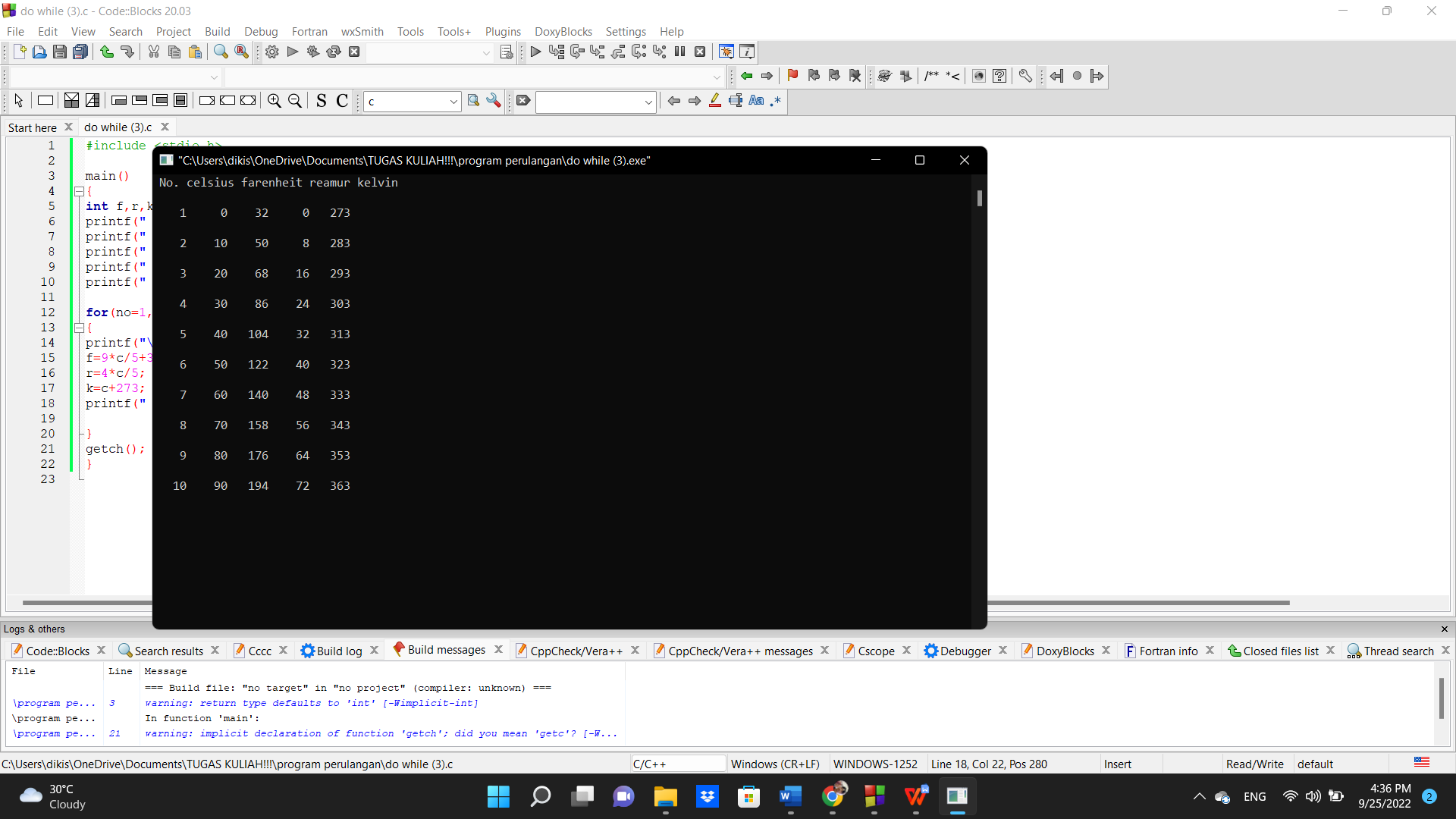
printf(” %5d %5d %5d”,f,r,k);

}

getch();

}

Hasilnya :



Contoh 6:

membalik urutan angka.

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

main()

{

int bil1;

printf(“Masukkan Angka = “);

scanf(“%d”,&bil1);

printf(“Hasil pembalikan angka = “);

while(bil1>0)

{

printf(“%d”,bil1%10);

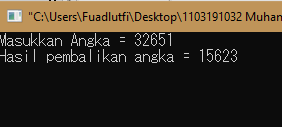
bil1=(bil1-bil1%10)/10;

}

getch();

}

Hasilnya :



Contoh 7:

membuat program menghitung nilai yang dimasukkan.

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

main()

{

int a,b,c,d=0;

printf(“Masukkan nilai data nilai = “);

scanf(“%d”,&a);

for(b=1;b<=a;b++)

{

printf(“masukkan data nilai ke %d =”,b);

scanf(“%d”,&c);

d=c+d;

}

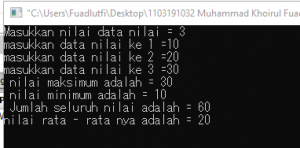
printf(” Jumlah seluruh nilai adalah = %d\n”,d);

printf(“nilai rata – rata nya adalah = %d”,d/a);

getch();

}

Hasilnya :



E. Perulangan Bersarang (Nested Loop)

Perulangan bersarang adalah sebutan untuk perulangan di dalam perulangan. Konsep seperti ini sering dipakai untuk memecahkan masalah programming yang cukup kompleks. Semua jenis perulangan bisa dibuat dalam bentuk perulangan bersarang, termasuk perulangan FOR, WHILE dan DO WHILE. Dalam bahasa inggris, perulangan bersarang ini dikenal dengan sebutan nested loop.

Berikut contoh format dasar perulangan bersarang dalam bahasa C:

for (start1; condition1; increment1)

{

// kode program

for (start2; condition2; increment2)

{

// kode program

}

}

Di dalam perulangan bersarang terdapat istilah outer loop dan inner loop. Sesuai dengan namanya, outer loop adalah sebutan untuk perulangan luar, sedangkan inner loop sebutan untuk perulangan dalam.

Pada contoh di atas, outer loop adalah perulangan di baris 1, sedangkan inner loop adalah perulangan di baris 4. Kode program di dalam outer loop akan dijalankan sejumlah kondisi perulangan di outer saja. Sedangkan kode program yang ada di dalam perulangan inner loop akan dijalankan sebanyak perulangan outer \* inner.

Tidak ada batasan seberapa banyak “kedalaman” dari sebuah perulangan bersarang. Kita bisa saja membuat perulangan di dalam perulangan di dalam perulangan, dst. Tentu saja semakin banyak perulangan yang “bersarang”, kode programnya juga akan makin kompleks.

Dalam membuat perulangan berasarang kita juga harus sangat teliti dalam penggunaan tanda kurung kurawal “{ }” untuk menandakan awal dan akhir sebuah blok kode program. Tidak jarang hasilnya jadi berantakan karena salah menulis posisi penutup kurung kurawal.

**Contoh Kode Program Perulangan Bersarang (Nested Loop)**

Mari kita masuk ke contoh kode program dari perulangan bersarang dalam bahasa C. Namun sebelum itu, kita berangkat dari perulangan “normal” terlebih dahulu.

Latihan pertama, bisakah anda merancang kode program untuk membuat daftar perkalian 3 dari 1 sampai 10? Hasil akhir yang saya inginkan adalah seperti ini:

3 \* 1 = 3

3 \* 2 = 6

3 \* 3 = 9

3 \* 4 = 12

3 \* 5 = 15

3 \* 6 = 18

3 \* 7 = 21

3 \* 8 = 24

3 \* 9 = 27

3 \* 10 = 30

Tentu saja baris di atas bisa dihasilkan dengan membuat perintah printf sebanyak 10 kali, namun hal itu tidak efisien. Hasil di atas harus dibuat menggunakan perulangan FOR.

Baik, berikut kode program yang saya gunakan:

#include <stdio.h>

int main(void)

{

for (int i = 1; i <= 10; i++) {

printf("3 \* %i = %i\n",i,3\*i);

}

return 0;

}

Disini saya menggunakan sebuah perulangan FOR. Perulangan mulai dari variabel counter i = 1 sampai i <= 10. Sepanjang perulangan, tampilkan hasil dari printf(“3 \* %i = %i\n”,i,3\*i).

Jika anda sudah memahami konsep dasar perulangan FOR, maka kode program ini seharusnya bisa dipahami dengan mudah.

Berikutnya, bagaimana dengan membuat kode program untuk menghasilkan daftar perkalian yang sama seperti di atas, tapi kali ini bukan hanya perulangan 3 saja, tapi dari 1 – 10. Maksudnya, saya ingin membuat kode program yang menghasilkan 100 baris daftar perkalian, mulai dari:

1 \* 1 = 1

1 \* 2 = 2

1 \* 3 = 3

....

....

Sampai dengan:

....

....

10 \* 8 = 80

10 \* 9 = 90

10 \* 10 = 100

Ada beberapa cara untuk membuat daftar perkalian seperti ini, misalnya bisa dengan membuat 10 kali perulangan FOR (1 perulangan untuk setiap angka). Namun cara paling efisien adalah menggunakan perulangan bersarang.

Berikut contoh kode program yang bisa dipakai:

#include <stdio.h>

int main(void)

{

for (int i = 1; i <= 10; i++) {

for (int j = 1; j <= 10; j++) {

printf("%i \* %i = %i\n",i,j,j\*i);

}

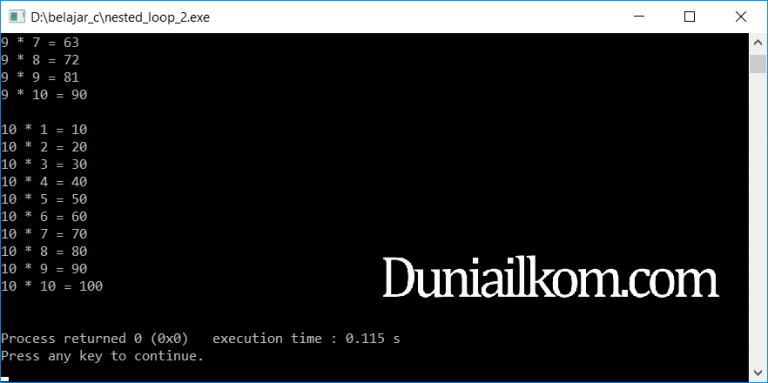
printf("\n");

}

return 0;

}

Hasilnya :



Untuk membuat perulangan bersarang, kita butuh 2 buah variabel counter, yakni 1 untuk outer loop, dan 1 untuk inner loop. Dalam contoh ini saya menggunakan variabel i dan j.

Perulangan i (outer loop) akan diulang sebanyak 10 kali, yakni dari i = 1 sampai i <= 10. Di dalam setiap perulangan i ini, akan di proses perulangan j (inner loop). Perulangan j jukan akan di proses sebanyak 10 kali. Dengan demikian, perintah printf di baris 7 akan dijalankan sebanyak 100 kali, yakni total dari i \* j.

Inilah salah satu contoh penerapan perulangan bersarang dalam bahasa C.

Sebagai tantangan terakhir, bisakah anda meng-konversi nested loop diatas menjadi perulangan WHILE? Silahkan dicoba sebentar.

Baik, berikut kode program untuk hasil yang sama namun kali ini dengan perulangan WHILE:

#include <stdio.h>

int main(void)

{

int i = 1;

int j;

while (i <= 10) {

j = 1;

while (j <= 10) {

printf("%i \* %i = %i\n",i,j,j\*i);

j++;

}

printf("\n");

i++;

}

}

Membuat nested loop dengan perulangan WHILE memang butuh tantangan tersendiri, karena kita harus memikirkan kapan sebuah variabel harus di deklarasikan. Sebagai contoh, perintah j = 1 harus berada di dalam perulangan i, namun sebelum perulangan j.

**BAB III**

**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

* <https://www.duniailkom.com/tutorial-belajar-c-cara-membuat-perulangan-bersarang-nested-loop/>
* <https://www.petanikode.com/c-perulangan/>
* <https://kumparan.com/how-to-tekno/perulangan-for-while-dan-do-while-dalam-bahasa-pemrograman-c-1xpOMbgh2Pr>
* <https://www.kmtech.id/post/perulangan-while-dan-do-while-dalam-bahasa-c>
* <https://www.duniailkom.com/tutorial-belajar-c-perulangan-for-bahasa-c/>