MAKALAH ARRAY DALAM PEMROGRAMAN

BAHASA C



Oleh :

DIKI CANDRA

NIM 2022903430010

D4 TEKNOLOGI REKAYASA KOMPUTER JARINGAN

POLITEKNIK NEGERI LHOKSEMAWE

2022/2023

DAFTAR ISI

[BAB I 3](#_Toc115375427)

[PENDAHULUAN 3](#_Toc115375428)

[A. Latar Belakang 3](#_Toc115375429)

[B. Rumusan Masalah 3](#_Toc115375430)

[C. Tujuan 3](#_Toc115375431)

[D. Manfaat 4](#_Toc115375432)

[1.Pembaca 4](#_Toc115375433)

[2.Pengarang 4](#_Toc115375434)

[BAB II 5](#_Toc115375435)

[PEMBAHASAN 5](#_Toc115375436)

[A. Pengenalan 5](#_Toc115375437)

[B. MENGAKSES ELEMEN ARRAY 6](#_Toc115375438)

[C. MENGAPA HARUS MENGGUNAKAN ARRAY 8](#_Toc115375439)

[1. ARRAY SATU DIMENSI 21](#_Toc115375440)

[2. ARRAY DUA DEMENSI 26](#_Toc115375441)

[3. ARRAY MULTIDIMENSI 31](#_Toc115375442)

[BAB III 37](#_Toc115375443)

[PENUTUP 37](#_Toc115375444)

[DAFTAR PUSTAKA 38](#_Toc115375445)

# 

# BAB I

# PENDAHULUAN

A. Latar Be

Variabel bisa kita gunakan yang saat ini hanya bisa menampung satu data hanya pada satu variabel saja. Dalam banyak kasus kita akan repot menggunakan banyak variabel, sebagai contoh perhitungan nilai, jika terdapat banyak siswa, maka variabel nilai yang dibutuhkan disesuaikan dengan banyaknya siswa. Maka dilihat dari kasus diatas, maka dengan array, kita bisa mendeklarasikan nilai.

Array adalah tipe data terstruktur yang terdiri dari sejumlah komponen-komponen yang mempunyai tipe sama. Komponen-komponen tersebut disebut sebagai komponen type, array mempunyai jumlah komponen yang jumlahnya tetap. Banyaknya komponen dalam array ditunjukkan oleh suatu index, dimana tiap komponen di array dapat diakses dengan menunjukkan nilai indexnya atau subscript. Array dapat bertipe data sederhana seperti byte, word, integer, real, bolean, char, string dan tipe data scalar atau subrange. Tipe array mengartikan isi dari aray atau komponen- komponenya mempunyai nilai dengan tipe data tersebut.

## B. Rumusan Masalah

Pada makalah ini kami merumuskan beberapa hal :

1)      Pengertian Struktur

2)      Pengertian Array

a.  Karakteristik array

b.  Deklarasi array

3)      Penggunaan Array

4)      Pengurutan Array

5)      Operasi Dasar Pada Array

6)      Bentuk-bentuk Array

7)      Keunggulan Array

8)      Kelemahan Array

## C. Tujuan

* Untuk memenuhi tugas dari dosen pada mata kuliah struktur data.
* Agar pembaca mengerti dan apa yang dimaksud dengan struktur array, serta paham dalam penggunaan, penerapan dan pengaplikasiannya.

## D. Manfaat

Manfaat yang diperoleh dari makalah tersebut adalah sebagai berikut:

### 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.Pengarang

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

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

# 

# BAB II

# PEMBAHASAN

## A. Pengenalan

PENGERTIAN ARRAY

Menurut definisinya, array (larik) adalah sebuah variable yang dapat menyimpan lebih dari satu nilai sejenis (memiliki tipe data yang sama). Hal ini tentu berbeda dengan variable biasa yang hanya mampu menampung satu buah nilai. Setiap nilai yang disimpan di dalam array di sebut dengan elemen array, sedangkan nilai urut yang digunakan untuk mengakses elemennya disebut dengan indeks array.

DEKLARASI ARRAY

Sama seperti variable lain, array juga dideklarasikan di dalam bagian deklarasi variable. Bila akan didefinisikan sebagai tipe bentukan, maka array juga akan dideklarasikan di bagian definisi tipe (dibawah kata kunci tipe). Dalam bahasa pascal, pendeklaraasian array dilakukan dengan menggunakan kata kunci array dan tipe data yang akan disimpan di dalamnya, selain itu juga harus disertai dengan batas-batas indeksnya yang diapit oleh tanda bracket([]). Berikut ini bentuk umum pendeklarasiannya.

NamaArray : array [IndeksArray..IndeksAkhir] of tipe\_data;

Sebagai contoh, apabila kita ingin mendeklarasikan array dengan nama A yang berisi 10 buah elemen bertipe integer, maka kita harus mendeklarasikannya dengan cara berikut.

Var

A : array [1..10] of integer;

Pada kode tersebut, indeks awal dimulai dari satu. Perlu diperhatikan bahwa bahasa pascal berbeda dengan bahasa C yang indeks array-nya selalu dimulai dari nol. Pada bahasa pascal, indeks array dapat dimulai dari bilangan berapapun. Selain itu, indeks array juga dapat bertipe karakter maupun tipe enumerasi. Berikut ini contoh-contoh kode yang dapat digunakan untuk mendeklarasikan 10 buah elemen array bertipe integer sebagai pengganti kode diatas.

Var

A1: array[0..9] of integer;

A2: array[5..15] of integer;

A3: array[‘a’..’j’] of integer;

A4: array[‘A’..’J’] of integer;

Dalam bahasa pascal, tersedia dua buah fungsi yang dapat digunakan untuk mengambil indek terendah dan tertinggi dari sebuah array, yaitu fungsi Low dan High. Adapun parameter dari kedua fungsi tersebut adalah nama array yang akan dicari indeknya. Perhatikan contoh kode berikut.

Var

A: array[1..100] of integer;

Terendah, tertinggi: integer;

Begin

Terendah:= low(A); {akan menghasilkan nilai 1}

Tertinggi:= high(A); {akan menghasilkan nilai 100}

…

End

.

## B. MENGAKSES ELEMEN ARRAY

Cara untuk memanipulasi array, langkah pertama yang harus dilakukan adalah mengisikan nilai ke dalam elemen-elemen array bersangkutan. Adapun bentuk umum untuk pengisian eleman array adalah sebagai berikut.

NamaArray[indeks]:= nilai;

Untuk lebih memahaminya, coba anda perhatikan terlebih dahulu contoh kode di bawah ini.

Var

A: array[1..100] of integer;

Begin

A[1]:= 1; {mengisi elemen pertama dengan nilai 1}

A[2]:= 2; {mengisi elemen kedua dengan nilai 2}

A[3]:= 3; {mengisi elemen ketiga dengan nilai 3}

……..

A[100]:= 100; {mengisi elemen keseratus dengan nilai 100}

End.

Kode tersebut akan melakukan pengisian 100 elemen array dengan nilai 1 sampai 100 sehingga kode tersebut akan lebih sederhana apabila dituliskan dengan menggunakan struktur pengulangan seperti yang terlihat pada kode berikut.

Var

A: array[1..100] of integer;

I: integer;

Begin

For i:=1 to 100 do begin

A[i]:= I ;

End;

End.

## C. MENGAPA HARUS MENGGUNAKAN ARRAY

Mungkin anda yang merupakan pemula akan bertanya mengapa kita perlu untuk mendeklarasikan array? untuk menjawab pertanyaan tersebut, coba Anda perhatikan dulu contoh kasus berikut. Apabila kita akan membuat program untuk menyimpan sekumpulan data, misalnya data-data hasil penelitian yang berupa bilangan, dimana jumlah dari data tersebut puluhan, ratusan, atau bahkan ribuan, apakah anda akan menggunakan variable sebanyak data yang ada? Jawabanya tentu tidak, karena hal tersebut merupakan hal yang sangat tidak efisien. Sebagai contoh, asumsikan bahwa banyak data tersebut adalah sepuluh, maka apabila kita menggunakan variabel biasa tentu kita akan menuliskannya melalui contoh kode berikut.

Var

N1, n2, n3, n4, n5, n6, n7, n8, n9, n10: real;

Begin

Writeln(‘masukkan data ke-1 : ‘); readln(n1);

Writeln(‘masukkan data ke-2 : ‘); readln(n2);

Writeln(‘masukkan data ke-3 : ‘); readln(n3);

Writeln(‘masukkan data ke-4 : ‘); readln(n4);

Writeln(‘masukkan data ke-5 : ‘); readln(n5);

Writeln(‘masukkan data ke-6 : ‘); readln(n6);

Writeln(‘masukkan data ke-7 : ‘); readln(n7);

Writeln(‘masukkan data ke-8 : ‘); readln(n8);

Writeln(‘masukkan data ke-9 : ‘); readln(n9);

Writeln(‘masukkan data ke-10 : ‘); readln(n10);

End.

Hal ini tentu akan merepotkan diri kita. Apabila dilihat, program di atas memang masih pendek karena datanya hanya sepuluh, bagaimana bila ratusan atau bahkan ribuan?

Untuk mengatasi masalah ini, seharusnya kita menggunakan array untuk menyimpan data-data tersebut sehingga program akan jauh lebih sederhana dan mudah dalam pengerjaannya. Berikut iini perbaikan program diataas apabila kita menampung data-datanya kedalam sebuah array..

Const max= 10;

N: array[1..max] of real;

I: integer;

Begin

For i:=1 to max do

Writeln(‘masukkan data ke-‘, I,’ : ‘); readln(n[i]);

End.

Apabila ternyata data berjumlah 100 atau 1000, maka anda hanya perlu mengganti nilai dari konstanta max di atas dengan nilai yang sesuai. Alasan seperti inilah yang menyebabkan kita perlu menggunakan array.

Contoh program:

{author: Noor Fitriana, S1 Ilmu Komputer 2008}

Program rata2;

uses wincrt;

const max=100;

var

nilai : array[1..max] of integer;

jml, rata: real;

n,i:integer;

begin

writeln(’\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*’);

writeln(’\*\*\*program untuk mencari rata-rata \*\*\*’);

writeln(’\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*’);

writeln;

writeln(’TEKAN ENTER UNTUK MENAMPILKAN DATA!’);

readln;

write(’Masukkan jumlah nilai yang di masukkan: ‘); readln(n);

writeln;

for i:=1 to n do begin

write(’nilai ke-’,i:2,’: ‘); readln(nilai[i]);

end;

jml:=0;

for i:=1 to n do

jml:=jml+nilai[i];

rata:=jml/n;

writeln;

writeln(’—————————’);

writeln(’Jumlah = ‘,jml:9:2);

writeln(’Rata-rata = ‘,rata:9:2);

writeln(’—————————’);

readln;

end.

Berikut Beberapa Jenis Array :

1. **Array Satu Dimensi** adalah suatu array yang hanya terdiri satu subcsript (elemen indeks) array.
2. **Array Dua Dimensi** adalah suatu array yang terdiri dari baris dan kolom, dua disini juga berarti dua indeks dimana indeks pertama berupa baris dan indeks kedua berupa kolom.
3. **Array Multi Dimensi** yaitu suatu array yang memiliki lebih dari dua dimensi, misalnya 3 dimensi yang banyak digunakan untuk suatu bangun ruang (tiga dimensi).

D. CARA MEMBUAT ARRAY PADA C

Pada pemrograman C, array dapat kita buat dengan cara seperti ini.

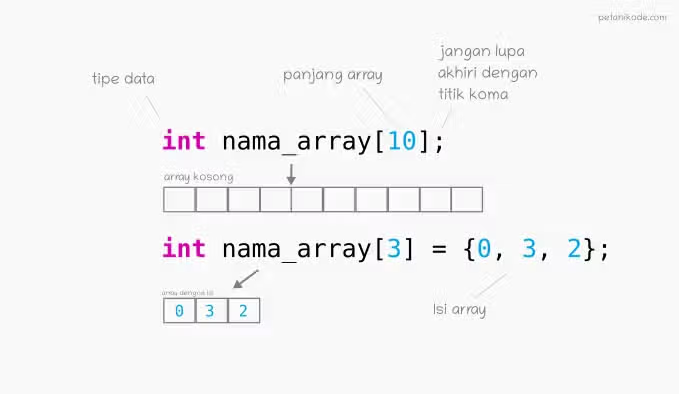
// membuat array kosong dengan tipe data integer dan panjang 10

int nama\_array[10];

// membuat array dengan langsung diisi

int nama\_arr[3] = {0, 3, 2}

Cara membuat array hampir sama seperti cara membuat variabel biasa.

Bedanya… pada array kita harus menentukan panjangnya

**Cara Mengambil Data dari Array**

Seperti yang sudah kita kethaui…

Array akan menyimpan sekumpulan data dan memberinya nomer indeks agar mudah diakses.

Indeks array selalu dimauli dari nol ‘0’.

Misalkan kita punya array seperti ini:

Bagaimana cara mengambil huruf ‘c’?

Jawabannya:

Mengapa bukan ‘huruf[3]’?

Ingat: indeks array selalu dimulai dari nol.

Biar lebih jelas, mari kita coba dalam program:

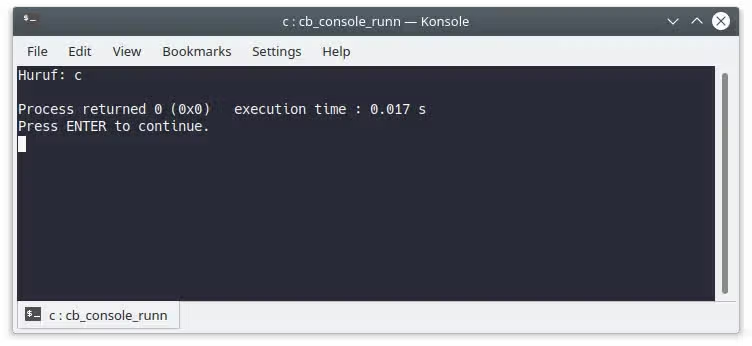
#include <stdio.h>

void main(){

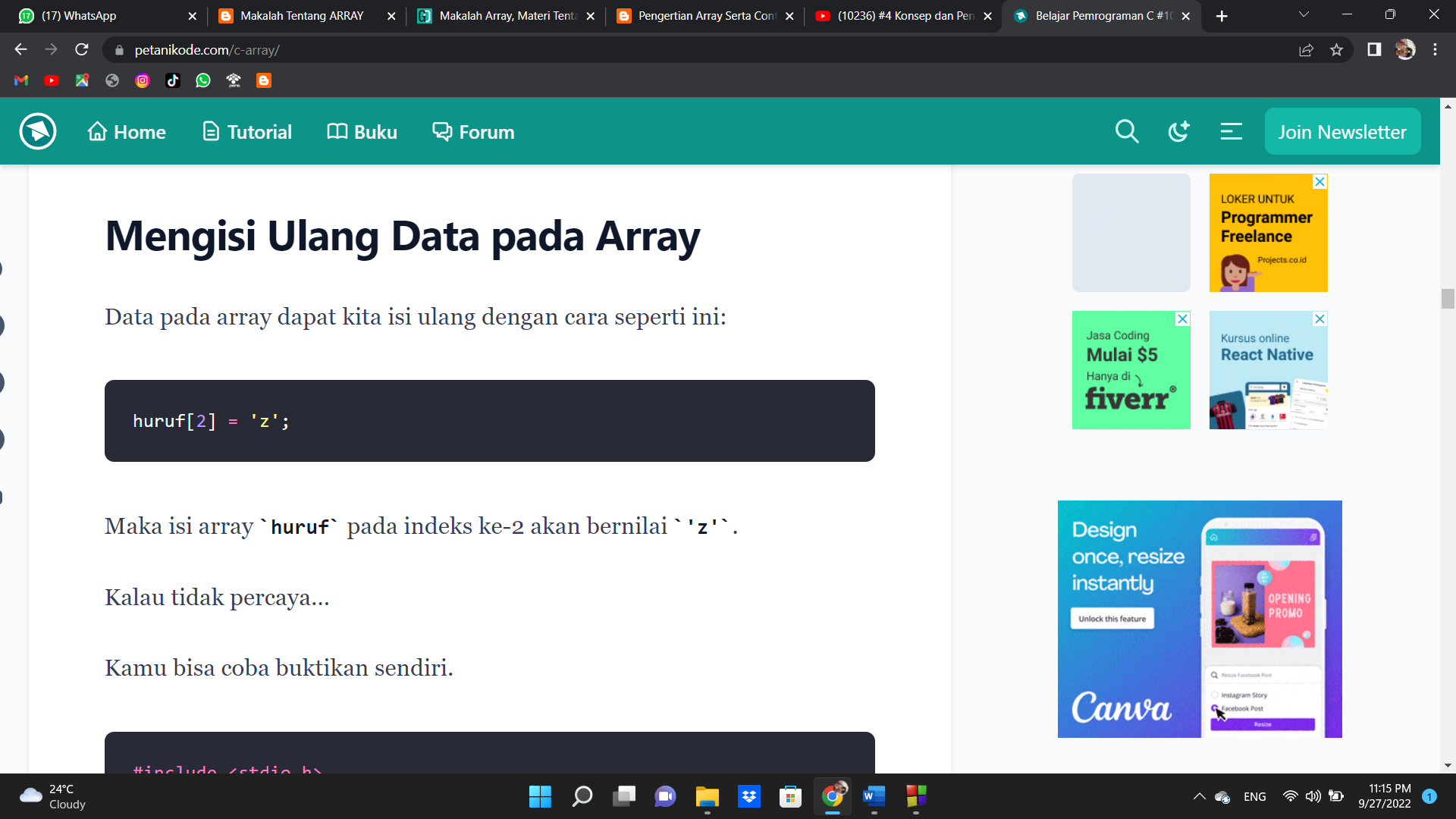
char huruf[5] = {'a', 'b', 'c', 'd', 'e'};

printf("Huruf: %c\n", huruf[2]);

}

Hasilnya:

**Mengisi Ulang Data pada Array**

Data pada array dapat kita isi ulang dengan cara seperti ini:

Maka isi array huruf pada indeks ke-2 akan bernilai 'z'.

Kalau tidak percaya…

Kamu bisa coba buktikan sendiri.

Contoh :

#include <stdio.h>

void main(){

// isi awal array

char huruf[5] = {'a', 'b', 'c', 'd', 'e'};

// mengubah isi array

huruf[2] = 'z';

// mencetak isi array

printf("Huruf: %c\n", huruf[2]);

}Contoh lain:

#include <stdio.h>

void main(){

// isi awal array

char huruf[5] = {'a', 'b', 'c', 'd', 'e'};

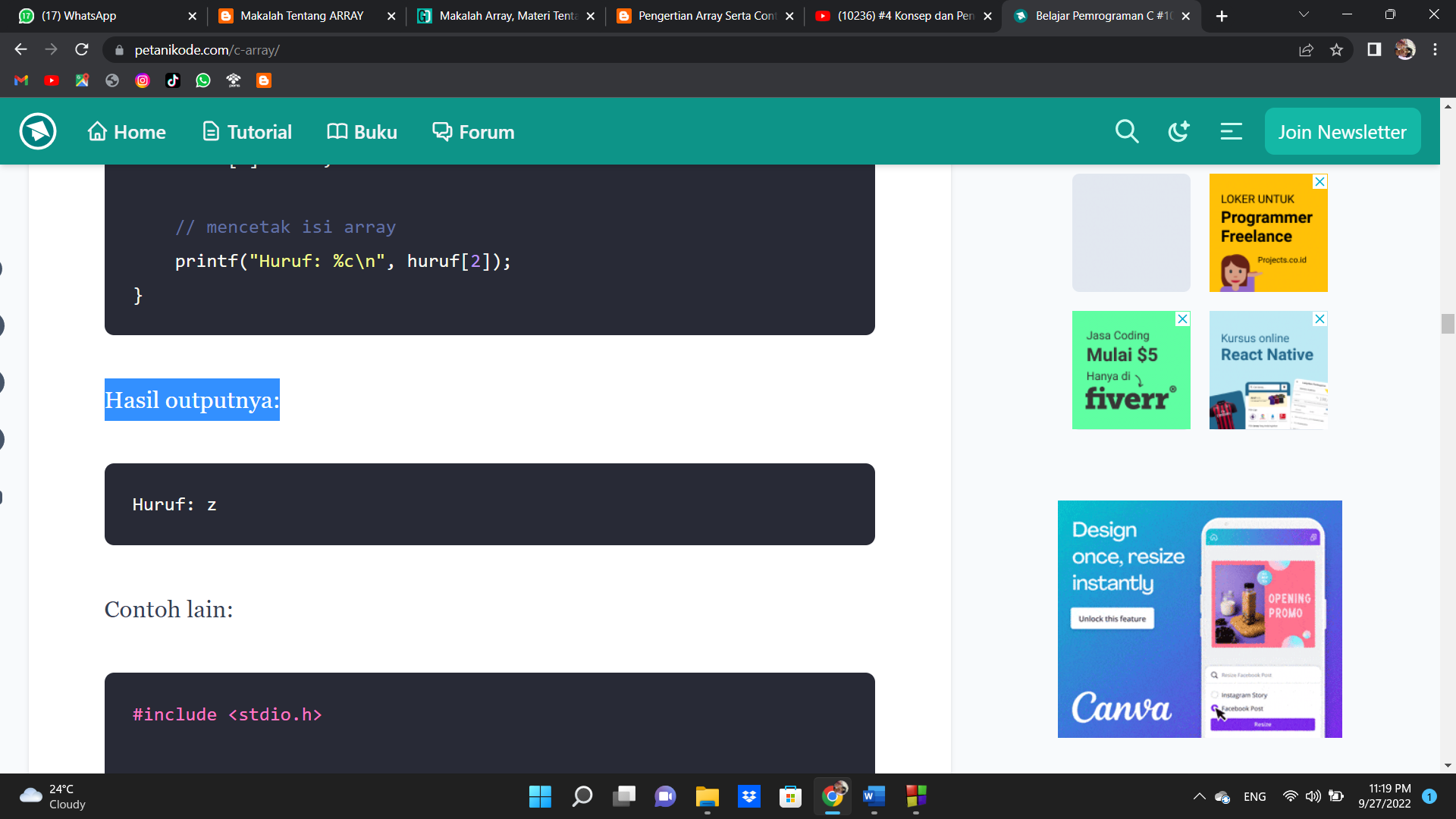
// mengubah isi array

huruf[2] = 'z';

// mencetak isi array

printf("Huruf: %c\n", huruf[2]);

}

Hasilnya:

Contoh Lainnya:

#include <stdio.h>

void main(){

// membuat array kosong

int nilai[5];

// mengisi array

nilai[0] = 32;

nilai[1] = 42;

nilai[2] = 76;

nilai[3] = 31;

nilai[4] = 57;

// mencetak isi array

printf("Nilai ke-1: %d\n", nilai[0]);

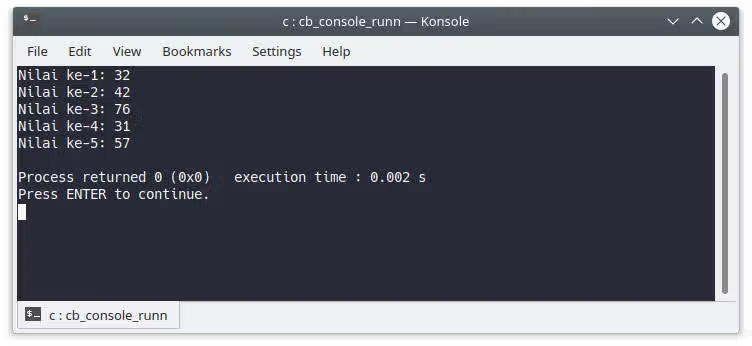
printf("Nilai ke-2: %d\n", nilai[1]);

printf("Nilai ke-3: %d\n", nilai[2]);

printf("Nilai ke-4: %d\n", nilai[3]);

printf("Nilai ke-5: %d\n", nilai[4]);

}

Output :

**Menggunakan Perulangan dan Array**

Pada contoh di atas, kita menggunakan perintah ‘printf()’ secara berulang untuk mencetak semua isi array.

printf("Nilai ke-1: %d\n", nilai[0]);

printf("Nilai ke-2: %d\n", nilai[1]);

printf("Nilai ke-3: %d\n", nilai[2]);

printf("Nilai ke-4: %d\n", nilai[3]);

printf("Nilai ke-5: %d\n", nilai[4]);

Kalau isi array-nya ada ribuan, apa kamu akan sanggup menulis ini berulang-ulang?

Tentu saja tidak!

Karena itu, kita bisa memanfaatkan perulangan untuk mencetaknya.

Contohnya

#include <stdio.h>

void main(){

// membuat array kosong

int nilai[5];

// mengisi array

nilai[0] = 32;

nilai[1] = 42;

nilai[2] = 76;

nilai[3] = 31;

nilai[4] = 57;

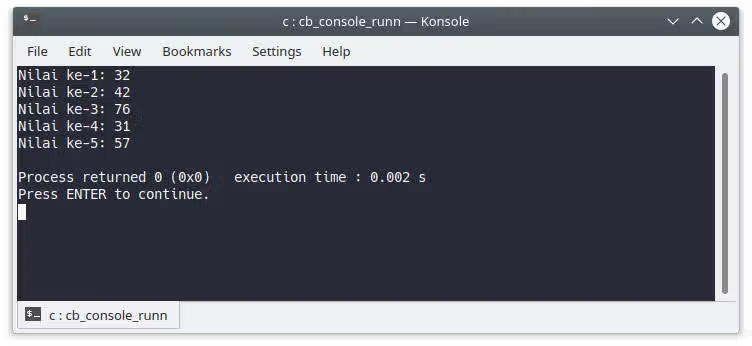
// mencetak isi array dengan perulangan

for(int i = 0; i < 5; i++){

printf("Nilai ke-1: %d\n", nilai[i]);

}

}

Output :

Jauh lebih sederhana ‘kan…

Tapi ada yang masih kurang.

Pada perulangan di atas kita memberikan batas maksimal secara manual, yaitu: ‘i < 5’.

Nilai ‘5’ seharusnya diganti dengan ukuran atau jumlah isi array.

Sehingga akan menjadi seperti ini:

int length = sizeof(nilai) / sizeof(\*nilai);

for (int i = 0; i < length; i++){

printf("Nilai ke-1: %d\n", nilai[i]);

}

**Mengambil Panjang Array**

Pada contoh di atas, kita menggunakan fungsi ‘sizeof()’ untuk mengambil panajang atau ukuran array.

Fungsi ‘sizeof()’ sebenarnya akan mengambil ukuran memori dari array.

Misalkan saya punya array seperti ini:

Isi array-nya berupa integer, tipe data integer memiliki ukuran 4 byte dalam memori. Maka ukuran array tersebut adalah 8 byte.

Lalu bagaimana cara mendapatkan banyaknya isi array?

Bahasa pemrograman C belum memiliki fungsi khusus untuk mengambil banyaknya isi array.

Tapi kita bisa mengambil panjang array dengan membagi besar array dan ukuran pointernya.

Contoh:

int array[] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };

/\* ukuran array dalam byte \*/

int size = sizeof(array);

/\* banyaknya isi array `array` \*/

int length = sizeof(array) / sizeof(\*array);

Mari kita coba…

#include <stdio.h>

void main(){

// membuat array

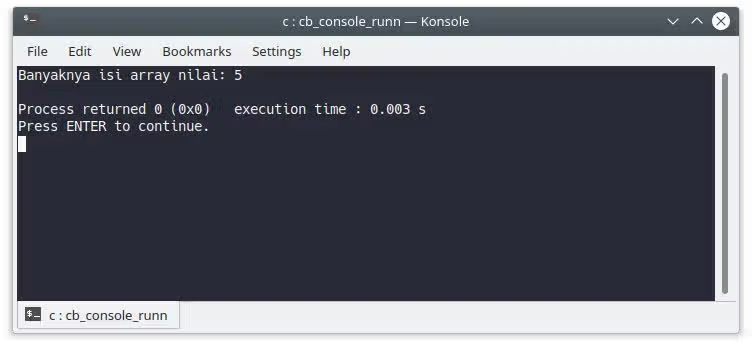
int nilai[5] = {33, 22, 11, 44, 21};

// mengambil banyaknya isi array

int length = sizeof(nilai) / sizeof(\*nilai);

printf("Banyaknya isi array nilai: %d\n", length);

}

Hasilnya:

**Latihan: Program Hitung Rata-Rata**

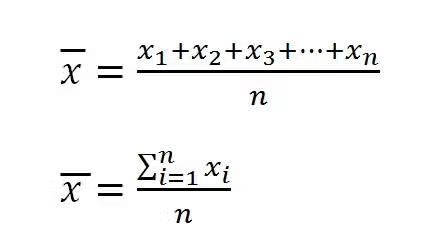
Anggap saja kita diminta untuk membuat program untuk menghitung tinggi rata-rata dari sepuluh orang.

Pertama kita pasti akan membutuhkan sebuah array dengan panjang ‘10’ yang berisi kumpulan tinggi badan



Berikutnya kita harus menghitung nilai rata-rata dari sekumpulan nilai tersebut.

Rumus nilai rata-rata:



Mari kita buat programnya…

#include <stdio.h>

void main(){

// membuat array dari tinggi badan

int tinggi\_badan[10] = {175, 165, 166, 157, 184, 156, 163, 176, 171, 169};

// length itu sama dengan n

int length = sizeof(tinggi\_badan) / sizeof(\*tinggi\_badan);

int sum = 0;

for(int i = 0; i < length; i++){

sum += tinggi\_badan[i];

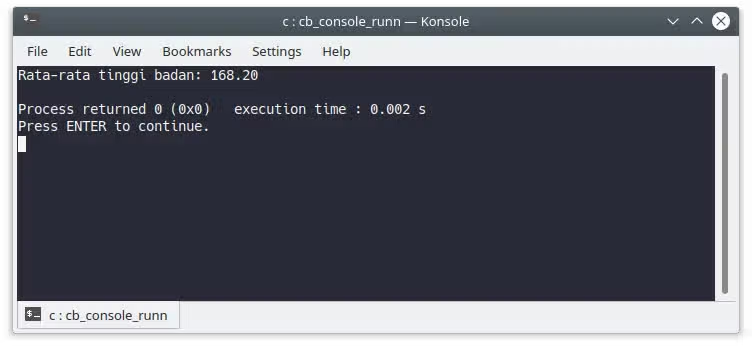
}

// rata-rata tinggi badan

float rata\_rata = (float)sum / (float)length;

printf("Rata-rata tinggi badan: %.2f\n", rata\_rata);

}

Hasilnya:

### 1. ARRAY SATU DIMENSI

Array satu dimensi adalah suatu array yang hanya terdiri satu subcsript (elemen indeks) array. Cara mendeklarasikannya yaitu :

Type namaArray [x];

Berikut beberapa contoh program Array berdimensi satu :

Contoh 1:

Input:

#include <stdio.h>

int main () {

int n[ 10 ]; // n adalah bilangan array bertipe integer, yang bernilai 10

int i,j;

//inialisasi elemen array n - 0

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

n[ i ] = i + 0; //membuat elemen pada lokasi i dengan i + 0

}

//output dari setiap nilai elemen array

for (j = 0; j < 10; j++ ) {

printf("Elemen[%d] = %d\n", j, n[j] );

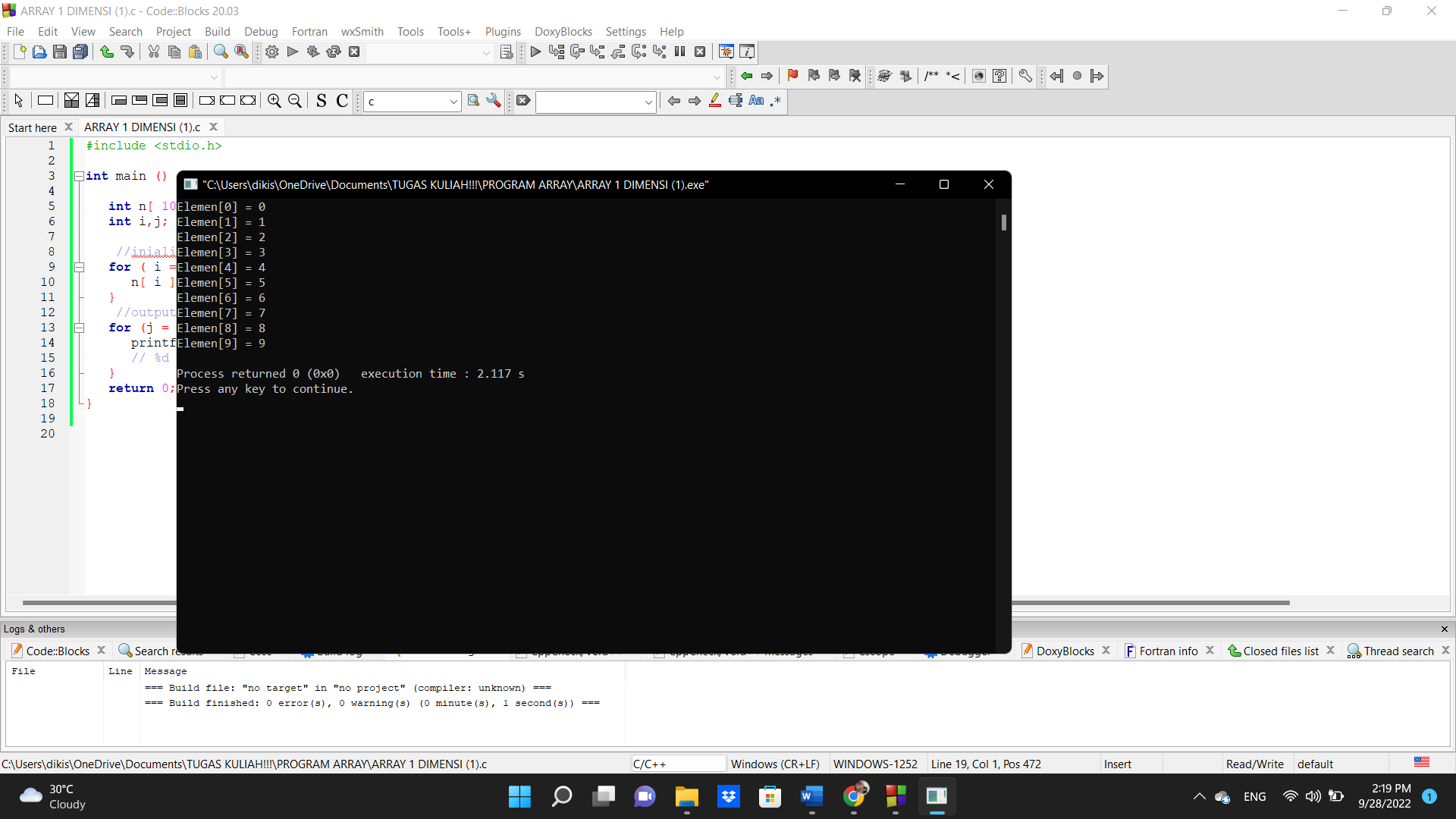
// %d berarti bilangan integer , not double

}

return 0;

}

Output:



Contoh 2:

#include <stdio.h>

main(){

int data1[5];

int i;

for(i=0;i<5;i++){

printf("Masukkan nilai ke-%d : ",(i+1));

scanf("%d",&data1[i]);

}

prinf("\n");

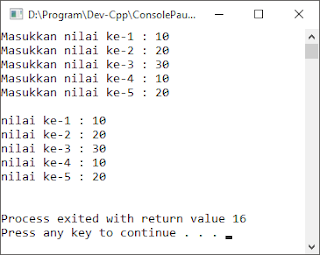
for(i=0;i<5;i++){

printf("nilai ke-%d : %d",(i+1),data1[i]);

}

}

Output:



Contoh 3:

//menampilkan jumlah hari dalam tiap bulan

#include <stdio.h>

main(){

int jml\_hari[12] = {31,28,31,30,31,30,31,31,30,31,30,31};

int i;

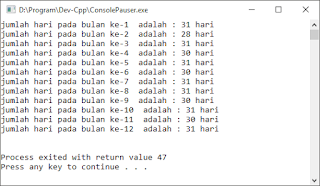
for(i=0;i<12;i++){

printf("jumlah hari pada bulan ke-%d adalah : %d hari\n",(i+1),jml\_hari[i]);

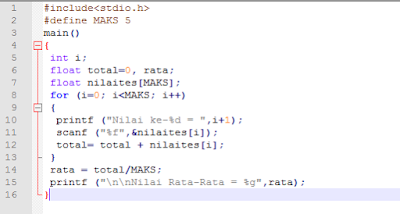
}

}

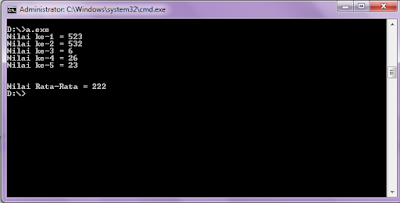
Output:



Contoh 4:

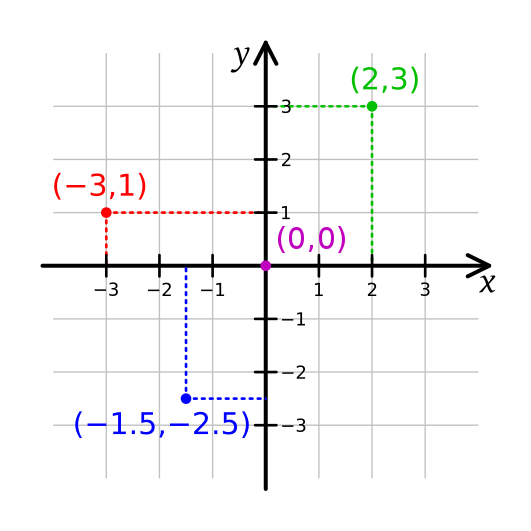


Output:



### 2. ARRAY DUA DEMENSI

Array dua demensi adalah **sebutan untuk array penomoran index-nya menggunakan 2 buah angka**. Analoginya yang sering dipakai seperti titik koordinat dalam diagram katesius.

**Diagram kartesius** merupakan diagram yang biasa kita pakai untuk membuat grafik. Disini terdapat sumbu X dan sumbu Y. Sebuah titik dalam diagram kartesius ini harus disebut secara berpasangan, seperti (2,3) atau (-3, 1).

Analogi lain adalah **matriks**. Dalam matematika, matrik terdiri dari kolom dan baris. Kembali, untuk menentukan nilai dari sebuah matriks, kita harus sebut secara berpasangan seperti baris 1 kolom 2, atau baris 3 kolom 1. Konsep seperti inilah yang menjadi dasar dari array 2 dimensi.

Untuk membuat array 2 dimensi di dalam bahasa C, kita membuat 2 kali tanda kurung siku setelah nama variabel, seperti contoh berikut:

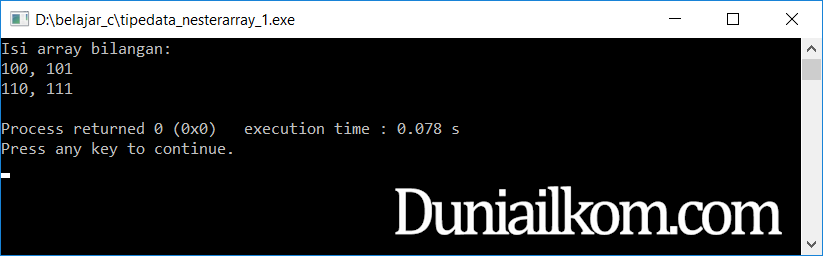
|  |
| --- |
| **int** bilangan[2][2]; |

Baris diatas akan membuat array 2 dimensi dengan nama variabel: **bilangan**. Variabel **bilangan** ini akan berisi 4 element (2 x 2). Atau jika diibaratkan sebagai matriks, disini kita membuat matriks 2 x 2.

Untuk mengakses setiap element array, penulisan index juga harus ditulis 2 kali, seperti contoh berikut:

|  |
| --- |
| bilangan[0][0] = 100;  bilangan[0][1] = 101;  bilangan[1][0] = 110;  bilangan[1][1] = 111;  Contoh Program Array 2 Dimensi Bahasa C  Contoh 1:  #include <stdio.h>    int main(void)  {  int bilangan[2][2];    bilangan[0][0] = 100;  bilangan[0][1] = 101;  bilangan[1][0] = 110;  bilangan[1][1] = 111;    printf("Isi array bilangan: \n");  printf("%d, %d \n",bilangan[0][0],bilangan[0][1]);  printf("%d, %d \n",bilangan[1][0],bilangan[1][1]);    return 0;  } |

Output:



Contoh 2:

#include <stdio.h>

int main(void)

{

int matrix[2][3] = {{1,2,3},{7,8,9}};

printf("Isi array matrix: \n");

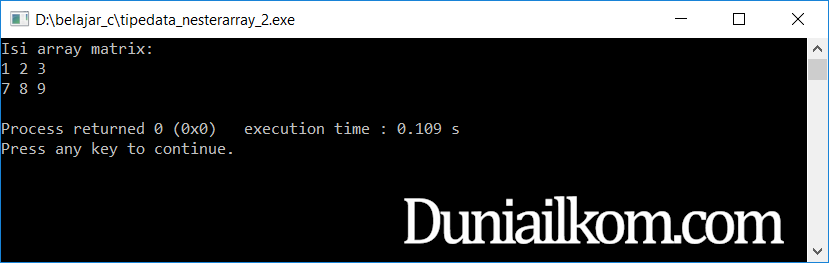
printf("%d %d %d \n", matrix[0][0], matrix[0][1], matrix[0][2]);

printf("%d %d %d \n", matrix[1][0], matrix[1][1], matrix[1][2]);

return 0;

}

Output:



Conroh 3:

int main (){

int i,j,baris,kolom;

printf ("masukan baris : "); //user diminta memasukan banyak baris

scanf ("%d",&baris);

printf ("masukan kolom : "); //user diminta memasukan banyak kolom

scanf ("%d",&kolom);

int m[baris][kolom];

printf ("masukkan angka ke dalam matriks : ");

printf ("\n");

/\*

pemasukan angka ke dalam matriks oleh user menggunakan perulangan

\*/

for (i=0;i<baris;i++){

for(j=0;j<kolom;j++){

printf ("baris %d kolom %d: ",i+1,j+1);

scanf ("%d",&m[i][j]);

}

}

printf ("\n");

/\* menampilkan hasil masukan user berbentuk matriks \*/

for (i=0;i<baris;i++){

for(j=0;j<kolom;j++){

printf ("%d ",m[i][j]);

}

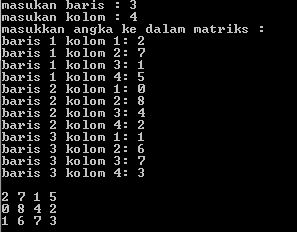
printf ("\n");

}

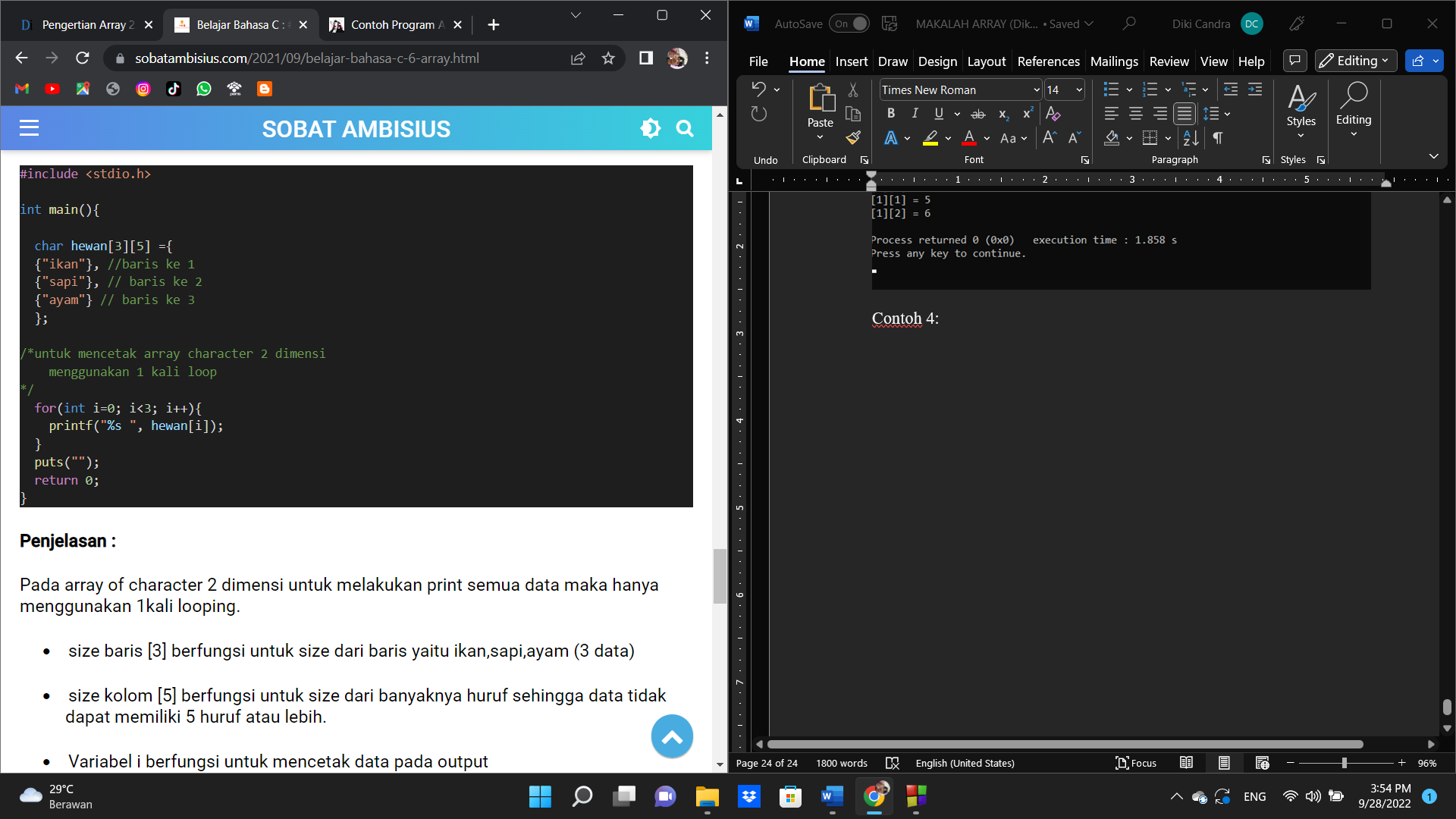
return 0;

}

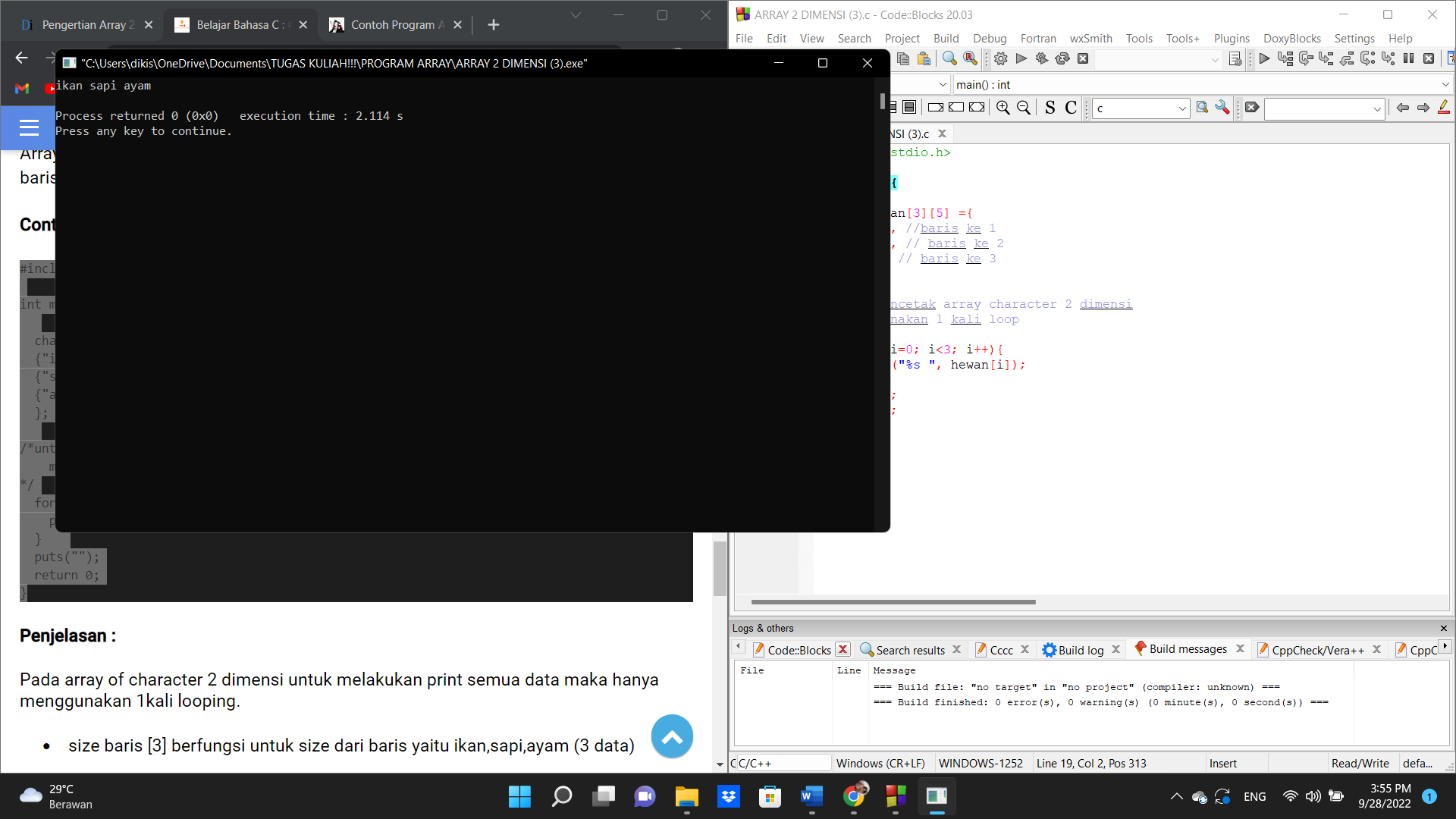
Output:



Contoh 4:



Output:



### 3. ARRAY MULTIDIMENSI

Array yang memiliki lebih dari satu dimensi disebut dengan array multidimensi pada MATLAB. Sebagian besar operasi yang dapat di lakukan pada matrix antara lain adalah array dua dimensi, array ini juga dapat dilakukan pada array multi dimensi. Array multi dimensi di dalam MATLAB merupakan perpanjangan dari matrix dua dimensi yang normal.

Matrix mempunyai dua buah dimensi yaitu, dimensi baris serta dimensi kolom. Anda dapat melakukan akses pada matrix dua dimensi dengan menggunakan dua subskrip: yang pertama yaitu mewakili indeks baris, dan yang kedua yaitu mewakili indeks kolom.

Array multi-dimensi yaitu jenis array yang memiliki banyak dimensi atau lebih dari satu dimensi dan sebuah array yang juga memiliki sebuah level, adapun jenis array multi-dimensi sederhana yaitu array dua Dimensi, atau array dua dimensi. Secara teknisnya array ini di sebut sebagai array – array, untuk bisa mendeklarasikan array multi dimensi serupa dengan mendeklarasikan array satu dimensi.

Untuk array 2 dimensi, kita sangat perlu memberi tahu pada program C bahwa kita mempunyai array 2 dimensi. Pada array multi dimensi array menggunakan subskrip tambahan untuk pengindeks-an. Array tiga dimensi, seperti menggunakan tiga subskrip seperti berikut:

* Referensi pertama yaitu dimensi array 1, baris
* Referensi kedua yaitu dimensi 2, kolom
* Dimensi referensi ketiga3. Ilustrasi yang memakai sebuah konsep halaman untuk dapat mewakili sebuah dimensi 3 dan ynag lebih tinggi.

Pada saat anda ingin menambahkan dimensi ke array, anda juga menambahkan subskrip. Pada array 4 dimensi, misalnya mempunyai empat buah subskrip. Dua referensi pertama merupakan pasangan baris dan kolom, dua yang ke dua mengakses dimensi data ketiga serta data keempat.

Deklarasi Array 3 Dimensi

Untuk dapat mendeklarasikan array 3 dimensi yaitu dengan menambahkan kurung siku [] atau subskrip menjadi 3, [][][]. Secara umum deklarasinya adalah di bawah ini

data\_type[][][] array\_name = new data\_type[x][y][z];

x = index array

y = baris

z = kolom

Contoh: int[][][] arr = new int[10][20][30];

Inisialisasi Array 3 Dimensi

Untuk memudahkan inisialisasi array 3D, bisa menggunakan representasi array 2D. Atau secara umum dapat digunakan dapat disajikan seperti di bawah ini

array\_name[array\_index][baris][kolom] = value;

contoh: arr[0][0][0] = 1;

data\_type[][][] array\_name = {

{

{valueA1B1C1, valueA1B1C2, ....},

{valueA1B2C1, valueA1B2C2, ....}

},

{

{valueA2B1C1, valueA2B1C2, ....},

{valueA2B2C1, valueA2B2C2, ....}

}

};

Keterangan

A adalah indek array

B adalah baris

C adalah kolom

Ketika diimplementasikan menggunakan Java kurang lebih seperti di bawah ini

int[][][] a = {

{

{1,2},

{3,4}

},

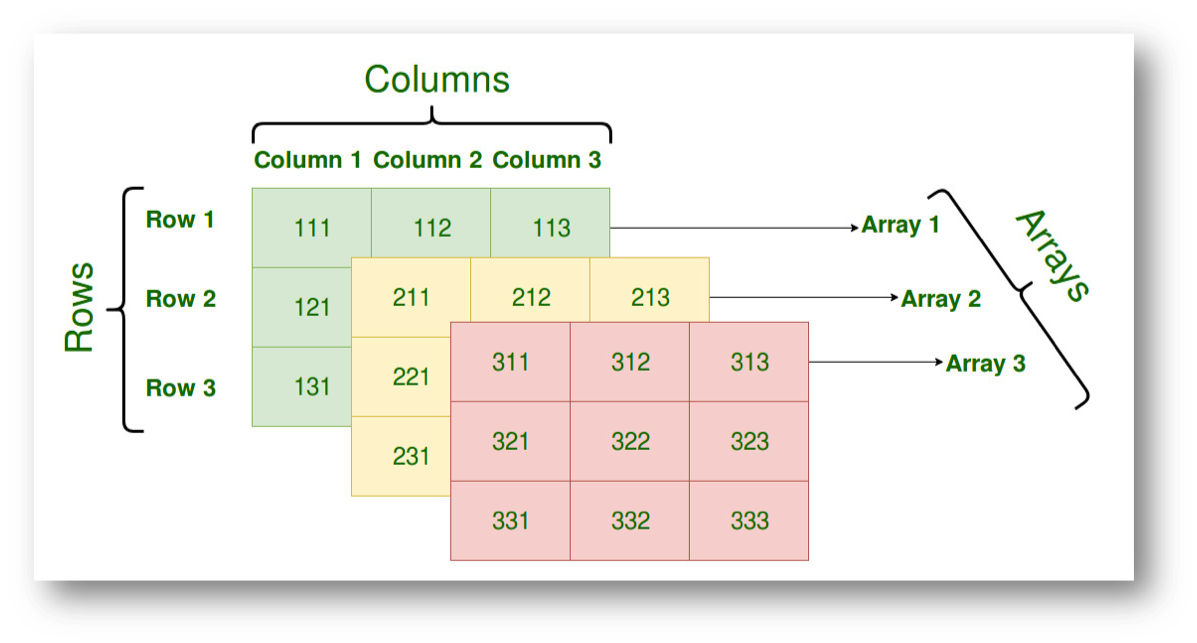
{

{5,6},

{7,8}

}

};

Atau misalkan dilakukan visualisasi adalah seperti di bawah ini

Contoh Array Multidimensi :

Contoh 1:

Input :

#include<stdio.h>

Main()

{

Char data [2][3][6] = {“1”, “soto”, “5000”, “2”, “bakso”, “3500”};

Int i,j;

For(i=0; i<2; i++)

{

For(j=0; j<3; j++)

{

Printf(“\t%s”, data [i][j]);

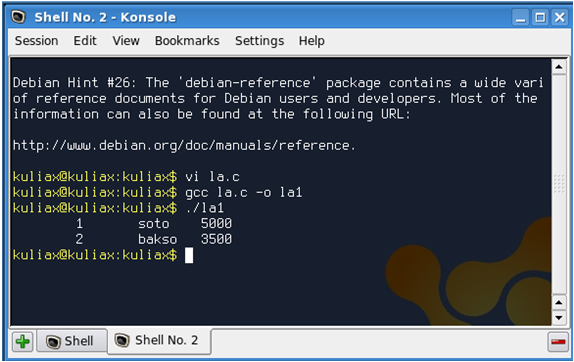
}

Printf(“\n”);

}

Return 0;

}

Output :

Contoh 2:

#include <stdio.h>

int main(void)

{

int matrix[2][3][4] =

{

{ {7, 4, 12, 3}, {-9, 29, 3, 11}, {6, 34, 23, 20} },

{ {6, 15, 1, 5}, {17, 8, -3, 15}, {99, -1, 44, 9} }

};

printf("Isi matrix[0][0][0]: %d \n",matrix[0][0][0]);

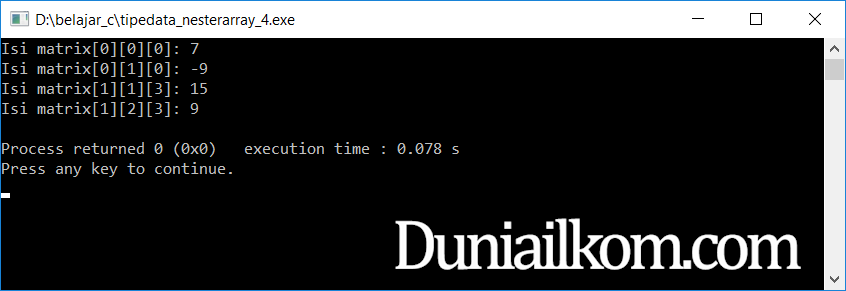
printf("Isi matrix[0][1][0]: %d \n",matrix[0][1][0]);

printf("Isi matrix[1][1][3]: %d \n",matrix[1][1][3]);

printf("Isi matrix[1][2][3]: %d \n",matrix[1][2][3]);

return 0;

}

Output :

# 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

* <http://sastrawanmha.blogspot.com/2015/07/programming-in-c-part-13-array_19.html>
* <http://abangkoding.blogspot.com/2018/08/array-multidimensi-bahasa-c.html>
* <https://www.petanikode.com/c-array/>
* <https://renisacadel.wordpress.com/tag/array-2-dimensi-bahasa-c/>