**LAPORAN JOBSHEET 7**

**ARRAY**



Disusun Oleh:

Dinda Cahya Nur Santoso

3.34.18.1.09

IK-1B

**PROGRAM STUDI D3 TEKNIK INFORMATIKA**

**JURUSAN TEKNIK ELEKTRO**

**POLITEKNIK NEGERI SEMARANG**

**2018**

1. **Dasar Teori**

Array merupakan sekumpulan data yang semua tipe datanya harus sama dan menggunakan nama variable yang sama. Berdasarkan indeksnya, array dibedakan menjadi by default dimulai dari 0 s/d (n-1) , dengan n adalah jumlah elemen dalam array. Array bisa berupa array berdimensi satu, dua, tiga atau lebih.

* Array berdimensi satu (one-dimensional array) mewakili bentuk suatu vektor.
* Array berdimensi dua (two-dimensional array) mewakili bentuk dari suatu matriks atau table.
* Array berdimensi tiga (three-dimensional array) mewakili bentuk suatu ruang.

Mendeklarasikan Array

* Suatu array berdimensi satu dideklarasikan dalam bentuk umum berupa :

tipe\_data nama\_var[ukuran];

* dengan :
* tipe\_data : untuk menyatakan tipe dari elemen array, misalnya int, char, float.
* nama\_var : nama variabel array
* ukuran : untuk menyatakan jumlah maksimal elemen array.
* Contoh pendeklarasian array :

float nilai[5];

menyatakan bahwa variabel nilai bertipe array of float dan memiliki 5 elemen bertipe float.

**Mengakses Elemen Array**

* Pada C, data array akan disimpan dalam memori yang berurutan.
* Elemen pertama mempunyai indeks bernilai 0.
* Jika nilai dideklarasikan sebagai array of float dengan 5 elemen, maka elemen pertama memiliki indeks sama dengan 0, dan elemen terakhir memiliki indeks 4.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

Tipe float

nilai[0]

nilai[1]

nilai[2]

nilai[3]

nilai[4]

Total 5 elemen

Bentuk umum pengaksesan array adalah sbb:

Sehingga, untuk array nilai, maka :

nilai [0] elemen pertama dari nilai

nilai [4] elemen ke-5 dari nilai

contoh :

nilai [0] = 70;

scanf (“%f”, &nilai[2]);

nama\_var[indeks]

Float nilai [5]

Contoh dalam bahasa C

#include <stdio.h>

#define MAKS 5

main()

{

int i;

float total=0, rata, nilai[MAKS];

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

{ printf("Nilai ke-%d : ", i+1);

scanf("%f", &nilai[i]);

total = total + nilai[i]; //hitung jml total nilai

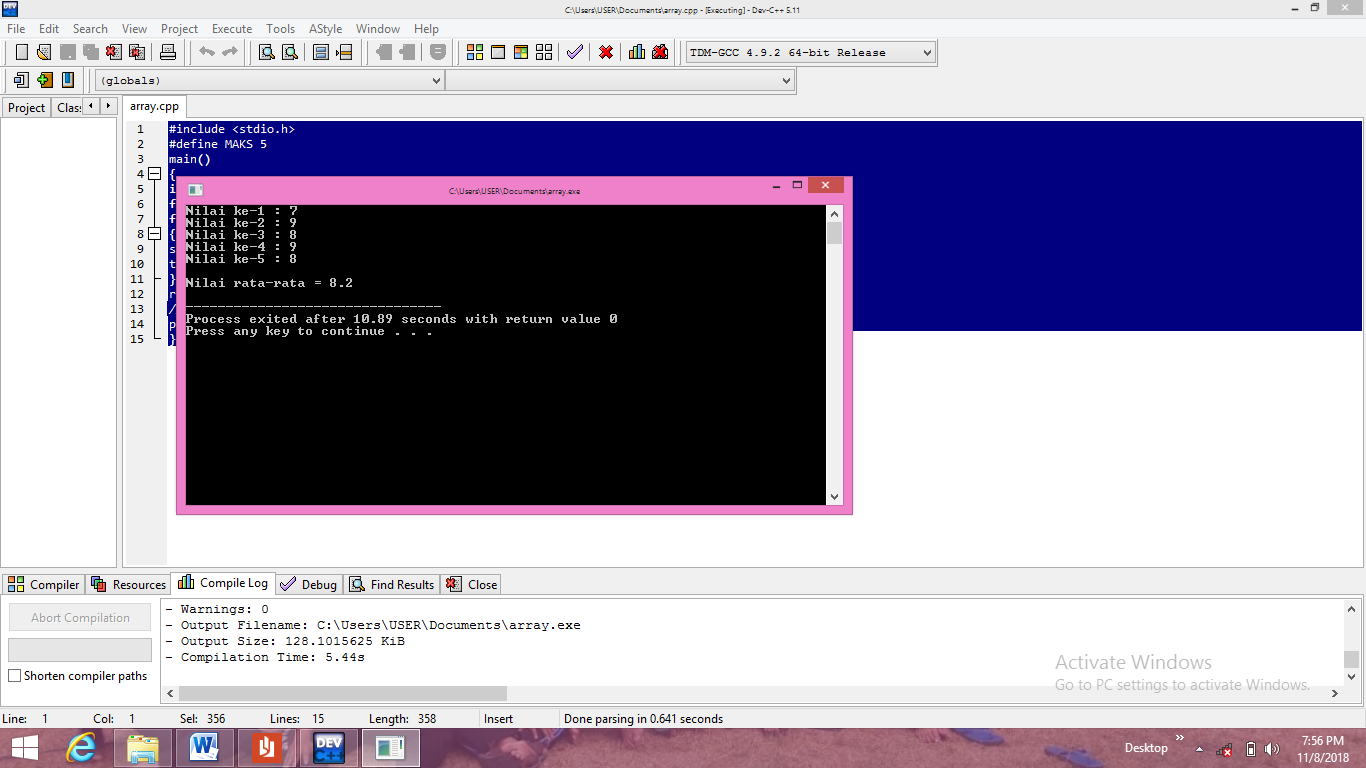
}

rata = total / MAKS; //hitung nilai rata2

//cetak nilai rata-rata

printf("\nNilai rata-rata = %g\n", rata);

}



**Inisialiasasi Array**

* Sebuah array dapat diinisialisasi sekaligus pada saat dideklarasikan.
* Untuk mendeklarasikan array, nilai-nilai yang diinisialisasikan dituliskan di antara kurung kurawal ({}) yang dipisahkan dengan koma.

Contoh dalam bahasa C

#include<stdio.h>

#include<conio.h>

main()

{

int bln, thn, jhari; int jum\_hari[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

puts("MEMPEROLEH JUMLAH HARI");

puts("PADA SUATU BULAN DAN SUATU TAHUN");

puts("--------------------------------");

printf("Masukkan bulan (1..12) : "); scanf("%d", &bln);

printf("Masukkan tahunnya : ");

scanf("%d", &thn);

if(bln == 2)

if(thn % 4 == 0) //thn kabisat

jhari = 29;

else

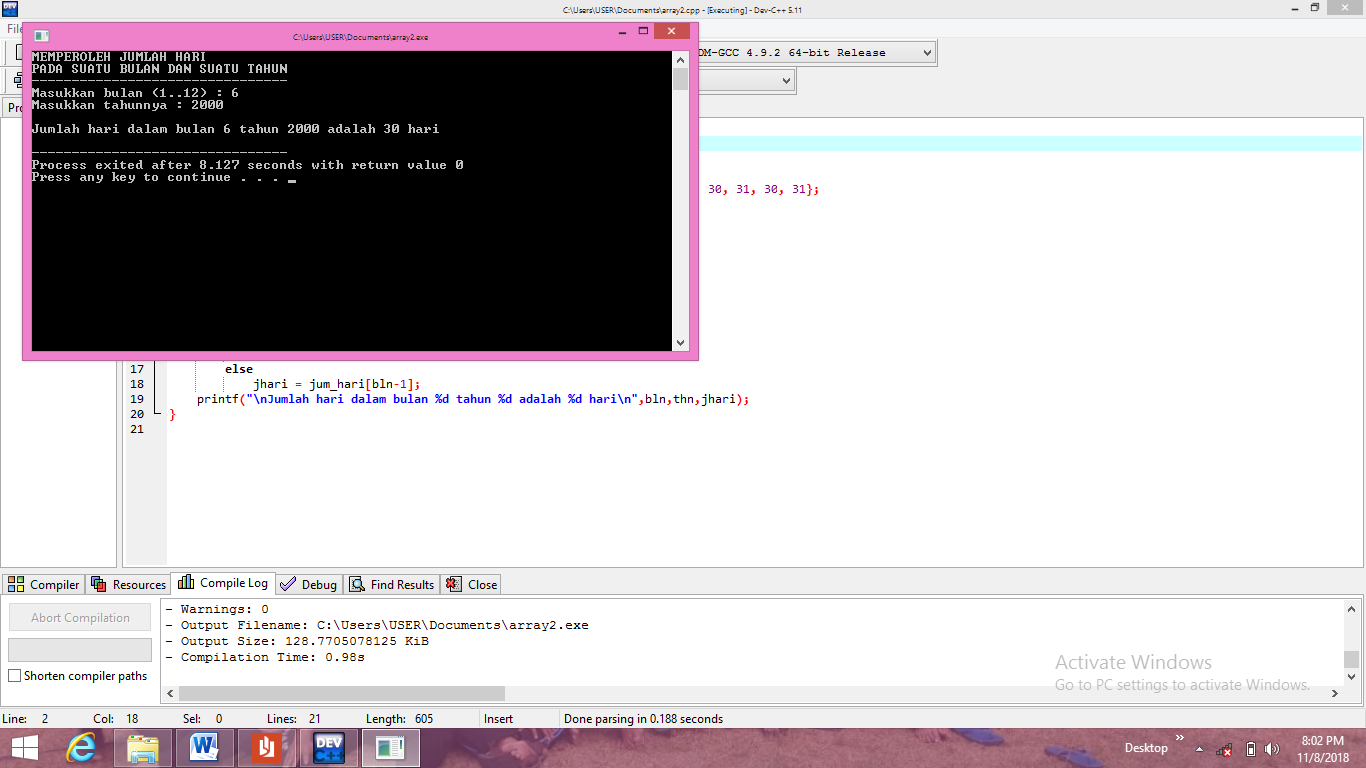
jhari = 28;

else

jhari = jum\_hari[bln-1];

printf("\nJumlah hari dalam bulan %d tahun %d adalah %d hari\n",bln,thn,jhari);

}



**Variasi dalam Mendeklarasikan Array**

* Ada beberapa variasi cara mendeklarasikan sebuah array (dalam hal ini yang berdimensi satu), di antaranya adalah sebagai berikut :
* int numbers[10];
* int numbers[10] = {34, 27, 16 };
* int numbers[] = {2, -3, 45, 79, -14, 5, 0, 28, -1, 0 };
* char text[] = "Welcome to New Zealand.";
* float radix[12] = {134.362, 1913.248};
* double radians[1000];
* Pendeklarasian nama variabel array diperbolehkan tidak disertai ukuran yang mengindikasikan besarnya array asalkan langsung diinisialisasi.
* Dalam kondisi seperti ini, C akan menginisialisasi ukuran array tersebut sejumlah elemen yang diberikan di dalam kurung kurawal pada saat proses inisialisasi.

**Array Berdimensi Dua**

|  |  |  |
| --- | --- | --- |
| NRP | Konsep Pemrograman | Matematika 1 |
| 1 | 90 | 87 |
| 2 | 75 | 93 |
| 3 | 88 | 92 |
| 4 | 67 | 80 |
| 5 | 84 | 80 |

* Data seperti yang disajikan pada tabel di atas, dapat disimpan pada sebuah array berdimensi dua:
* Dimensi pertama dari array digunakan untuk menyatakan NRP
* Dimensi kedua untuk menyatakan Matakuliah

**Mendeklarasikan Array (dimensi 2)**

Indeks kedua (matakuliah)

0 1

0

1

2

3

4

|  |  |
| --- | --- |
| 90 | 87 |
| 75 | 93 |
| 88 | 92 |
| 67 | 80 |
| 84 | 80 |

Indeks pertama (nrp)

Int nilai [S] [2] ;

Pendeklarasian yang diperlukan untuk menyimpan nilai mahasiswa pada tabel tsb adalah:

int nilai[5][2];

5 menyatakan banyaknya mahasiswa dan 2 menyatakan banyaknya matakuliah

**Mengakses Elemen Array (dimensi 2)**

* Array nilai dapat diakses dalam bentuk

nilai[indeks pertama][indeks kedua]

0 1

|  |  |
| --- | --- |
|  | 87 |
|  |  |
|  |  |
|  |  |

0

1

2

3

Nilai [0] [1] = 87;

* Contoh :
* nilai[0][1] = 87;

merupakan instruksi untuk memberikan nilai 87 ke array nilai untuk indeks pertama = 0 dan indeks kedua bernilai 1.

* printf(“%d”,nilai[2][0]);

merupakan perintah untuk menampilkan elemen yang memiliki indeks pertama = 2 dan indeks kedua = 0.

**Array Berdimensi Banyak**

* C memungkinkan untuk membuat array yang dimensinya lebih dari dua. Bentuk umum pendeklarasian array berdimensi banyak :

tipe nama\_var[ukuran1][ukuran2}…[ukuranN];

* sebagai contoh :

int data\_ruang[2][8][8];

* merupakan pendeklarasian array data\_ruang sebagai array berdimensi tiga.

**Inisialiasasi Array tak berukuran**

* Inisialisasi array yang tak berukuran dapat dilakukan untuk array berdimensi satu atau lebih.
* Untuk array berdimensi lebih dari satu, dimensi terkirilah yang boleh tak berukuran.
* Dengan cara ini tabel dalam array dapat diperluas atau dikurangi tanpa mengubah ukuran array.

Tak berukuran

* Sebagai cont

intskala ( ) =

( 1, 2, 4, 6, 8 );

* merupakan pendeklarasian array berdimensi satu yang tak berukuran. Secara otomatis

skala[ 0 ] bernilai 1

skala[ 1 ] bernilai 2

skala[ 2 ] bernilai 4

skala[ 3 ] bernilai 6

skala[ 4 ] bernilai 8

**Array sebagai Parameter Fungsi**

Ketika diinginkan hasil proses di dalam fungsi terbaca di tempat fungsi tsb dipanggil , maka gunakan salah satu mekanisme sbb :

1. return value maks nilai yang dilaporkan = SATU
2. pass by reference nilai yg dilaporkan lebih dari 1, bisa berbeda tipe datanya
3. menjadikan array sbg parameter jika hasil prosesnya banyak dan semua tipenya sama.

* jadikan array sebagai parameter aktual (tanpa kurung siku) & array sebagai parameter formalnya (tanpa size)
* antara parameter aktual dengan parameter formal sebenarnya merupakan variabel yang berada pada lokasi/address yang SAMA, namun berbeda namanya (ALIAS)
* SO, perubahan apapun pada parameter formal PASTI akan berpengaruh pada parameter aktual!!!

Contoh dalam bahasa C

**Latihan**

1. Deklarasikan sebuah variabel array of int, selanjutnya isi array tsb kemudian tampilkan isi variabel tersebut menggunakan statement for(). Ingat apabila jumlah deklarasi index adalah n maka nilai index-nya adalah dimulai dari 0 sampai dengan n-1.

#include<stdio.h>

#include<conio.h>

#define maks 100

void input(int);

main()

{

int n;

printf("\t====================================\n");

printf("\tMENAMPILKAN ISI VARIABLE ARRAY of INT\n");

printf("\t====================================\n");

printf("\nMasukkan jumlah data : ");

scanf("%d", &n);

input(n);

getch();

}

void input(int n)

{

int a[maks],i;

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

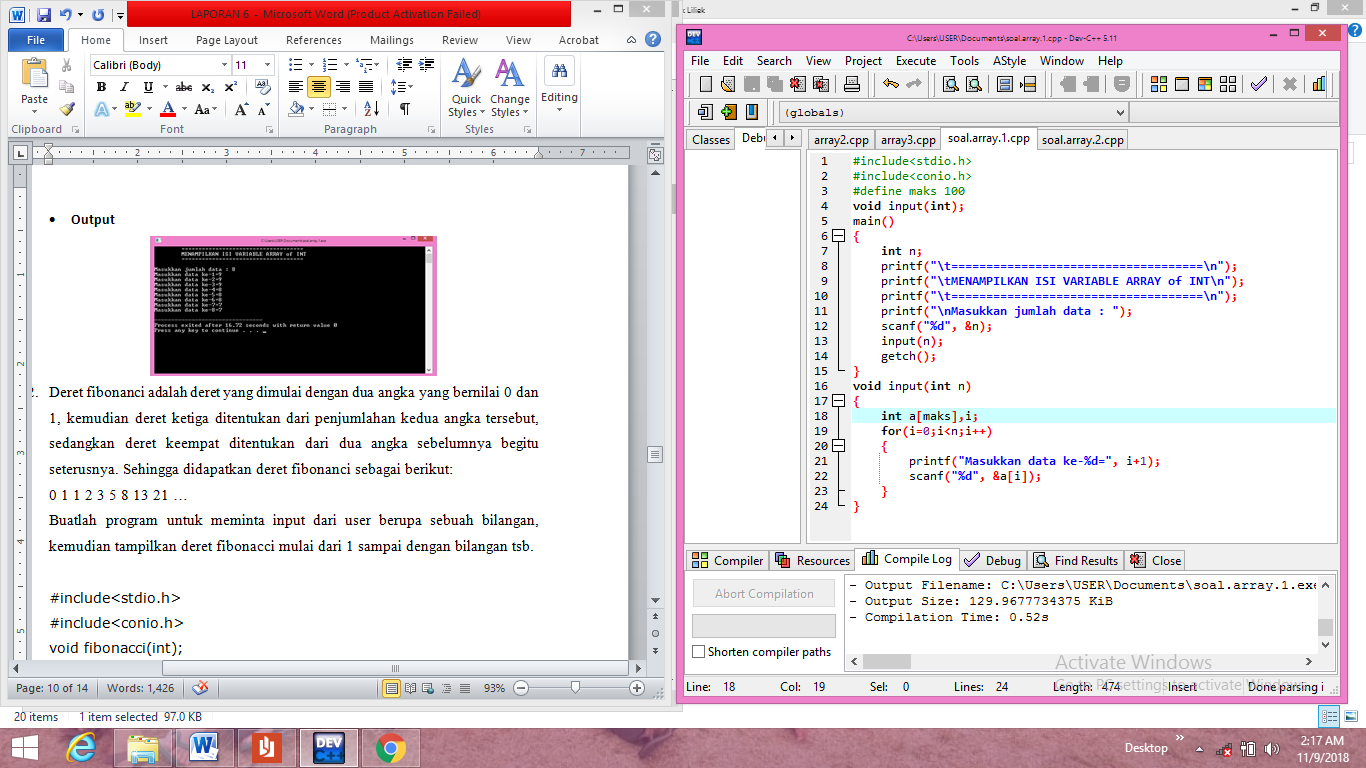
{

printf("Masukkan data ke-%d=", i+1);

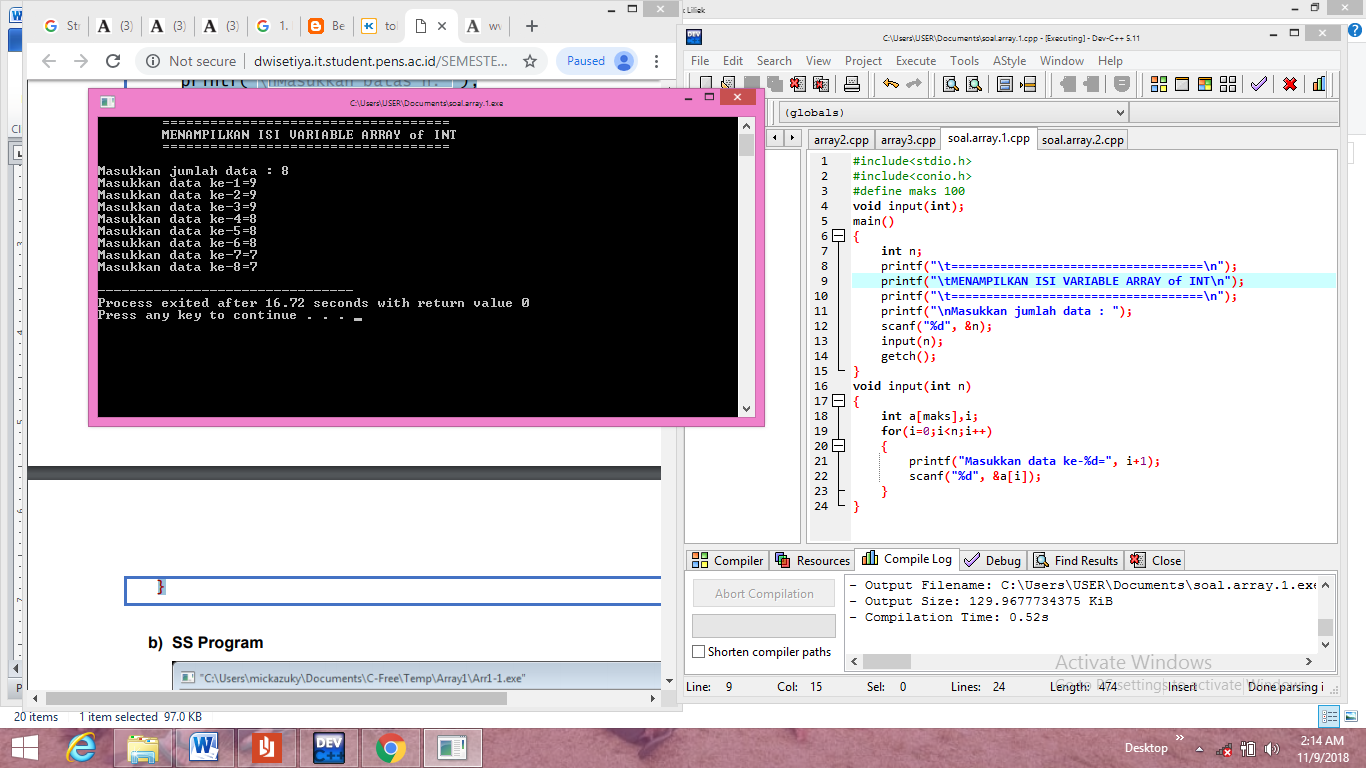
scanf("%d", &a[i]);

}

}



* **Output**



1. Deret fibonanci adalah deret yang dimulai dengan dua angka yang bernilai 0 dan 1, kemudian deret ketiga ditentukan dari penjumlahan kedua angka tersebut, sedangkan deret keempat ditentukan dari dua angka sebelumnya begitu seterusnya. Sehingga didapatkan deret fibonanci sebagai berikut:

0 1 1 2 3 5 8 13 21 …

Buatlah program untuk meminta input dari user berupa sebuah bilangan, kemudian tampilkan deret fibonacci mulai dari 1 sampai dengan bilangan tsb.

#include<stdio.h>

#include<conio.h>

void fibonacci(int);

int main(){

int n;

printf("\t====================================\n");

printf("\t\tFIBONACCI of ARRAY\n");

printf("\t====================================\n");

printf("\nMasukkan jumlah data : ");

scanf("%d", &n);

fibonacci (n);

printf("\n");

getch();

}

void fibonacci (int n)

{

int i,a[100];

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

{

if(i==0)

{

a[i]=0;

printf("0");

}

else

if(i==1)

{

a[i]=1;

printf("1");

}

else

{

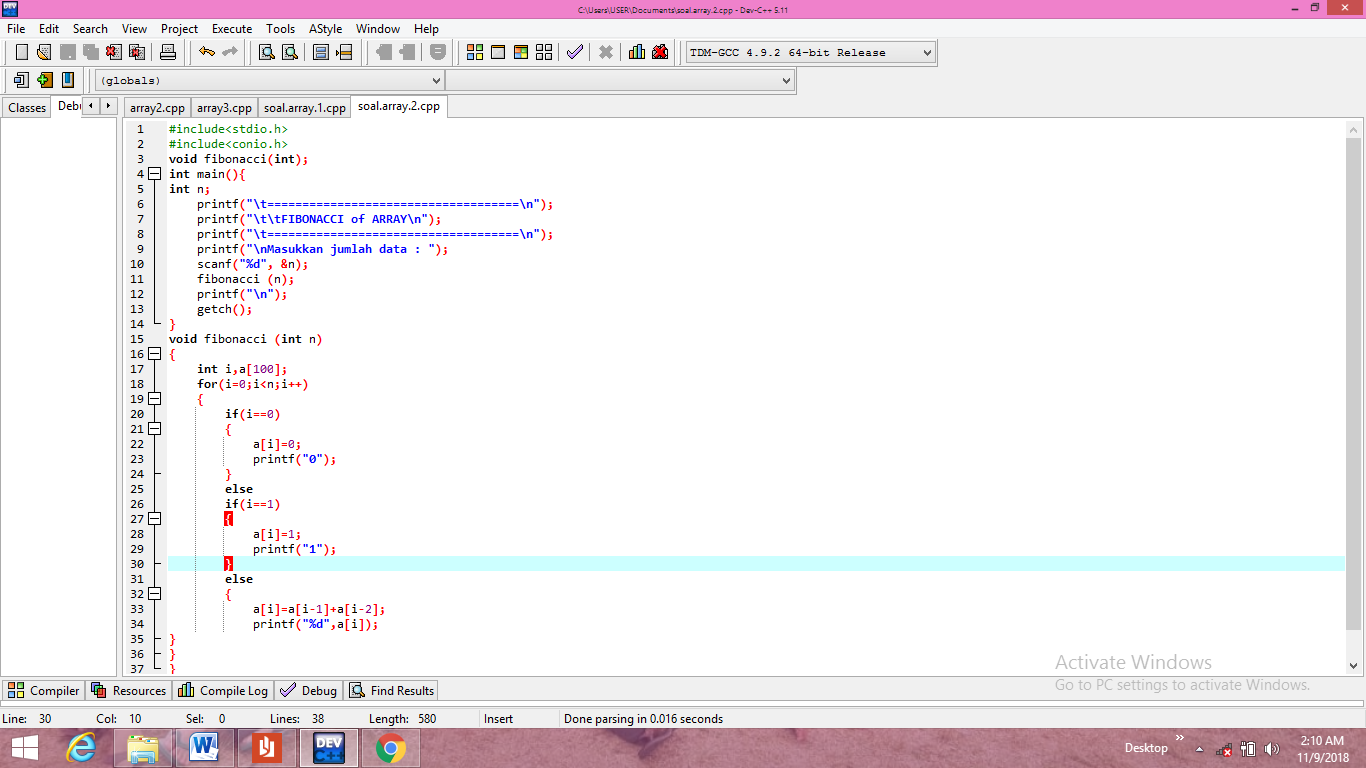
a[i]=a[i-1]+a[i-2];

printf("%d",a[i]);

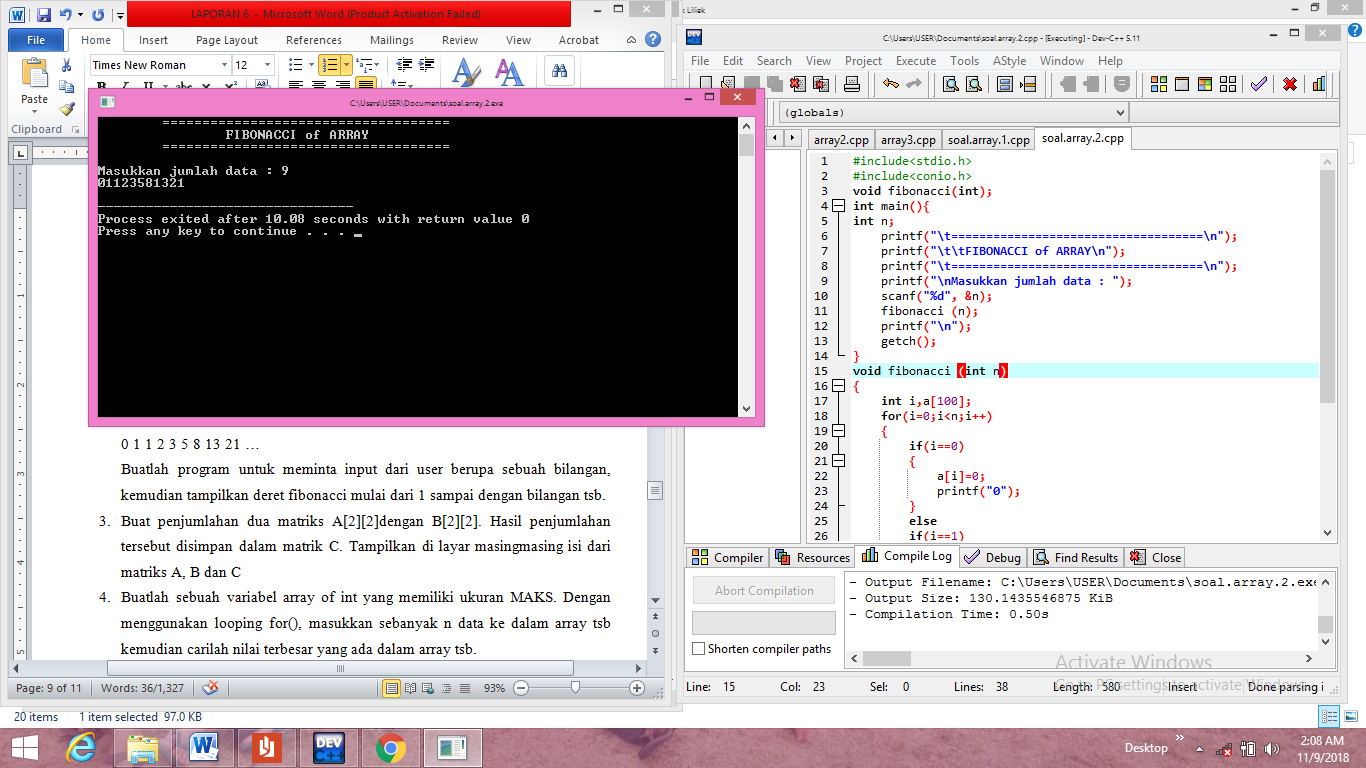
}

}

}



* **Output**



1. Buat penjumlahan dua matriks A[2][2]dengan B[2][2]. Hasil penjumlahan tersebut disimpan dalam matrik C. Tampilkan di layar masingmasing isi dari matriks A, B dan C

#include<stdio.h>

#include<conio.h>

main()

{

int a,b,A[2][2],B[2][2], hasjum[2][2], baris, kolom;

printf("Matriks A\n");

for(a=0;a<2;a++)

{

for(b=0;b<2;b++)

{

printf("Masukkan Baris %d dan Kolom %d : ",a+1,b+1);

scanf("%d",&A[a][b]);

}

}

printf("\nMatriks B\n");

for(a=0;a<2;a++)

{

for(b=0;b<2;b++)

{

printf("Masukkan Baris %d dan Kolom %d : ",a+1,b+1);

scanf("%d",&B[a][b]);

}

}

printf("\n");

for(baris=0;baris<2;baris++)

{

for(kolom=0;kolom<2;kolom++)

{

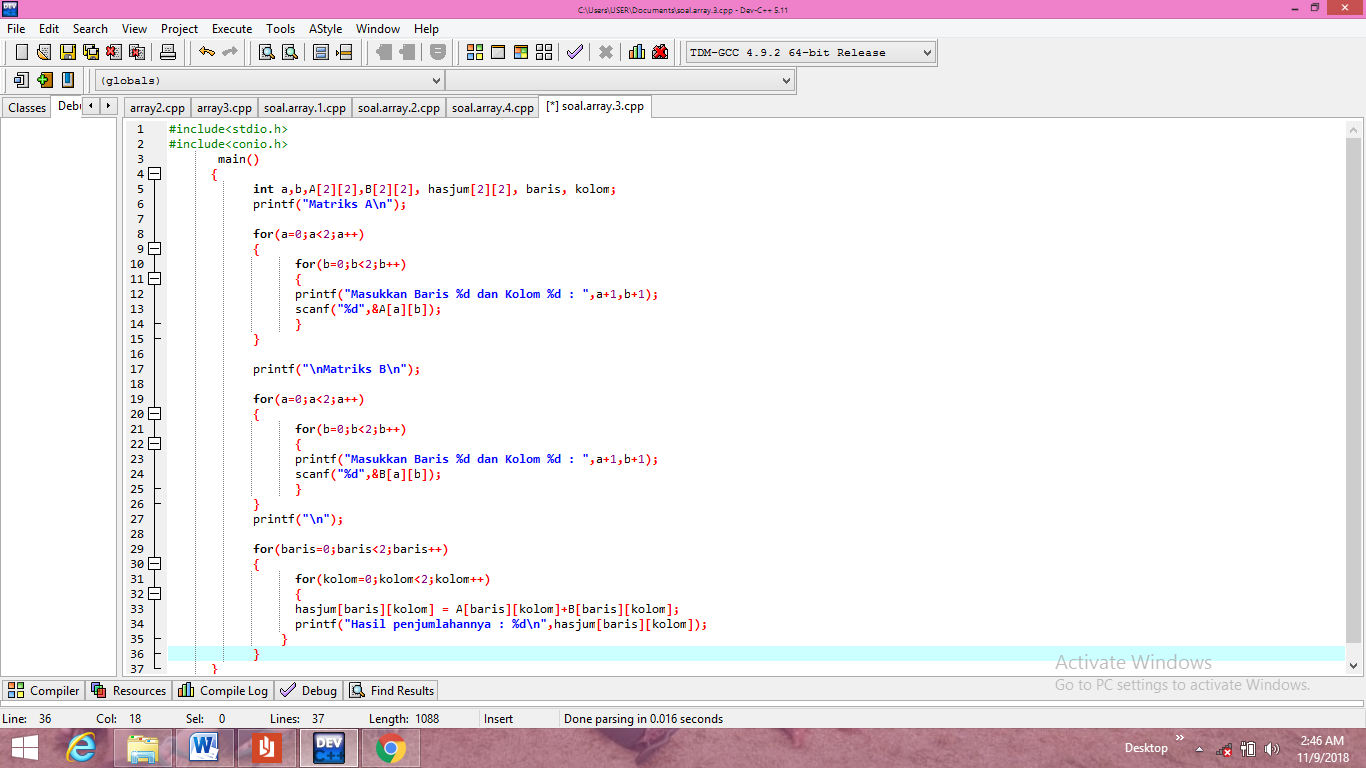
hasjum[baris][kolom] =A[baris][kolom]+B[baris][kolom];

printf("Hasilpenjumlahannya:%d\n",hasjum[baris][kolom]);

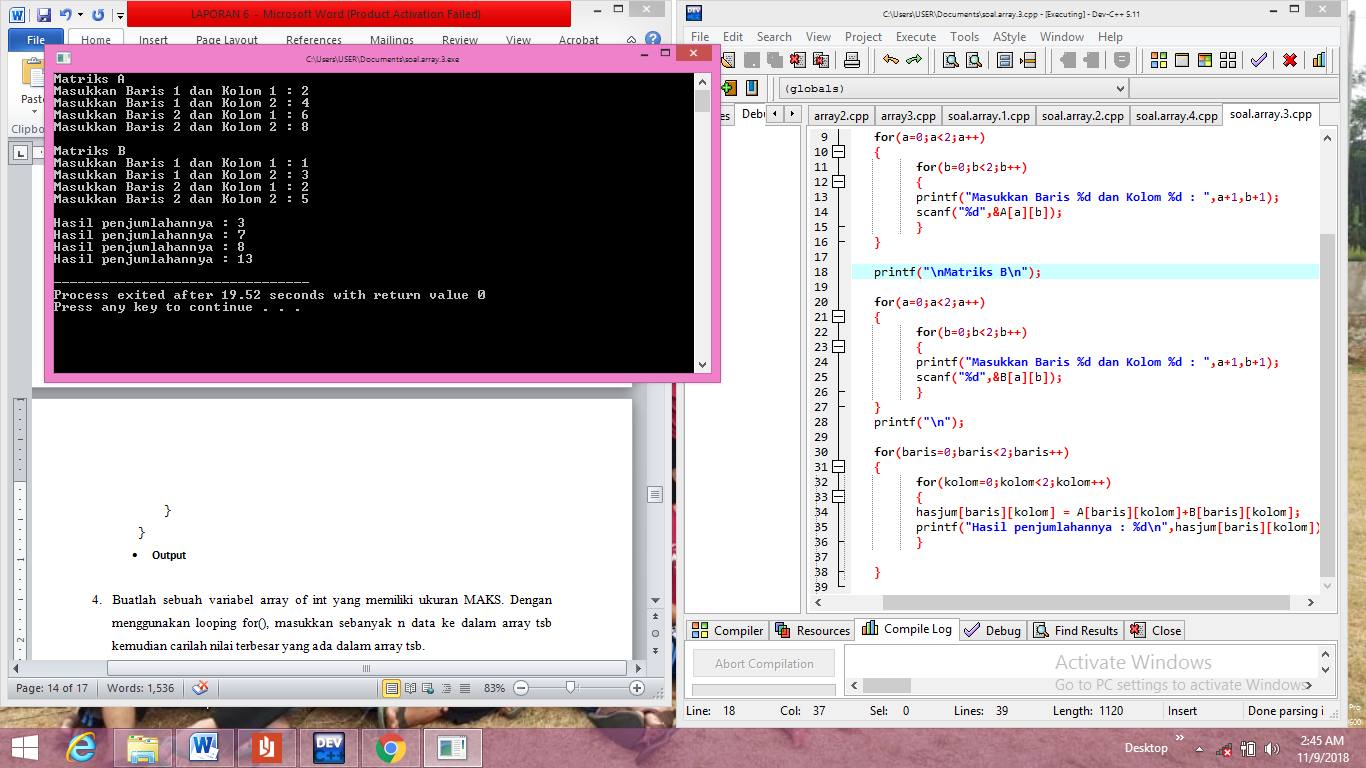
}

}

}



* **Output**



1. Buatlah sebuah variabel array of int yang memiliki ukuran MAKS. Dengan menggunakan looping for(), masukkan sebanyak n data ke dalam array tsb kemudian carilah nilai terbesar yang ada dalam array tsb.

#include <stdio.h>

#include <conio.h>

#define Maks[5]

main ()

{

int data [5];

int a,b;

int maks,min;

printf("jumlah bilangan yang dimasukkan (maks 5): ");

scanf ("%d",&a);

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

{

printf("\nbilangan ke-%d: ",b);

scanf("%d",&data [b]);

}

maks=data[1];

min=data[1];

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

if (data[b]>=maks)

{

maks=data[b];

}

else

if (data[b]<=min)

{

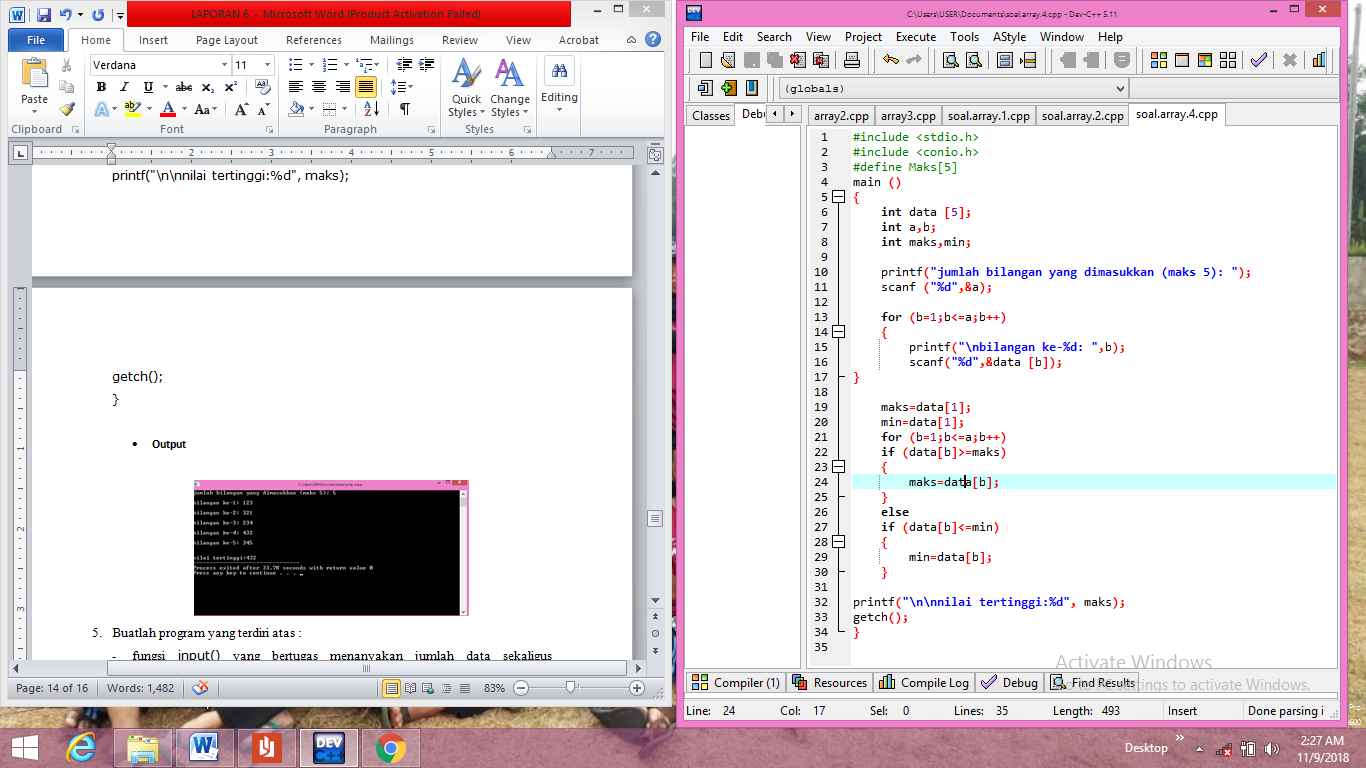
min=data[b];

}

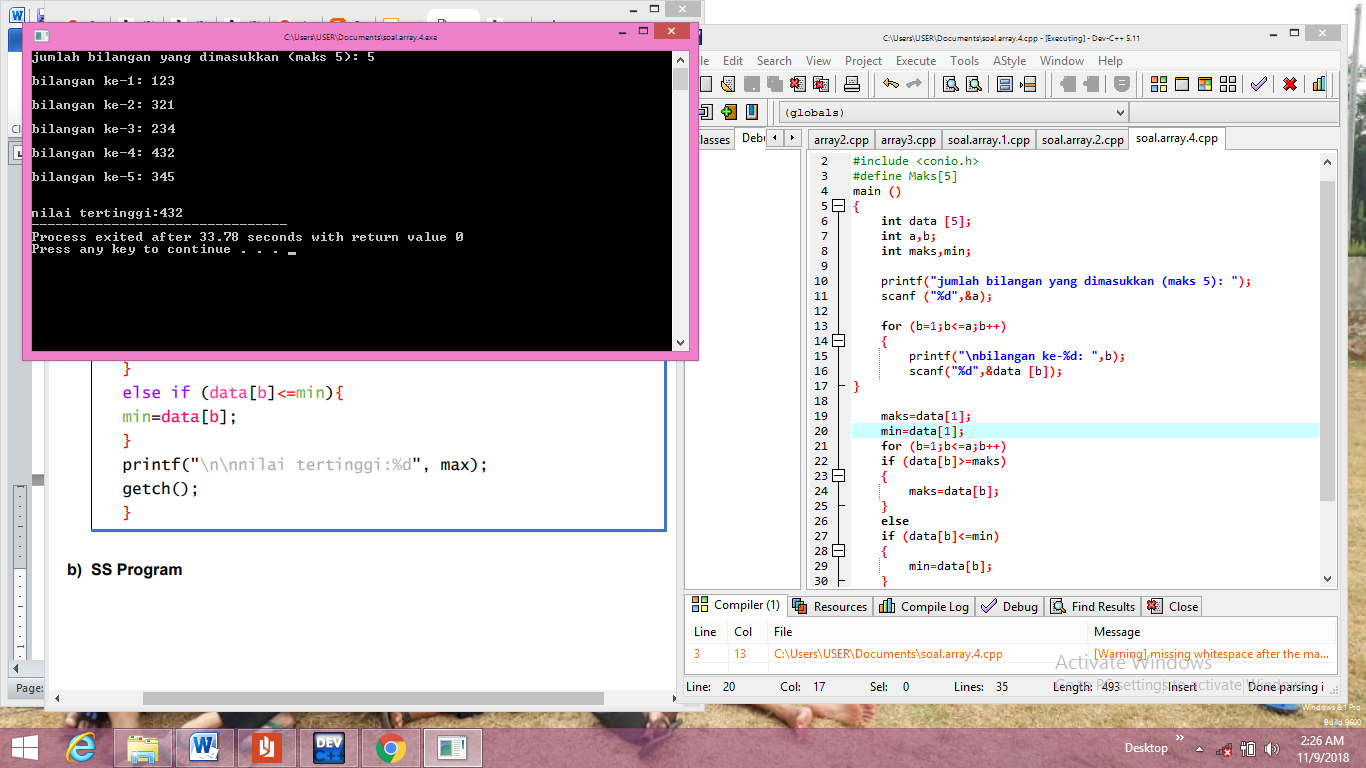
printf("\n\nnilai tertinggi:%d", maks);

getch();

}



* **Output**



1. Buatlah program yang terdiri atas :

* fungsi input() yang bertugas menanyakan jumlah data sekaligus memasukkan data bilangan sebanyak yang diminta ke dalam array nilai[]
* fungsi findmax()yang bertugas mencari nilai terbesar yang ada pada array nilai[] tsb
* fungsi main() yang memanggil fungsi input() dan fungsi findmax() , selanjutnya menampilkan nilai terbesar yang ada pada array nilai[]

**Petunjuk** : jadikan array sebagai parameter fungsi

1. Diketahui daftar nilai siswa sebagai berikut:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Nama Mhs | BAHASA | MATEMATIKA | DIGITAL |
| 1 | ahmad | 81 | 90 | 62 |
| 2 | Adang | 50 | 83 | 87 |
| 3 | Dani | 89 | 55 | 65 |
| 4 | Edi | 77 | 70 | 92 |

Buatlah program untuk menampilkan laporan sebagai berikut:

No. Mhs Rata-rata

=================

1 77,67

2 73,33

3 69,67

4 79,67

==================

Petunjuk: Gunakan variable array dua dimensi untuk menyimpan data.

|  |  |  |  |
| --- | --- | --- | --- |
| x | Sin(x) | Cos(x) | Tan(x) |
| 0 |  |  |  |
| 30 |  |  |  |
| 60 |  |  |  |
| 90 |  |  |  |
| 120 |  |  |  |
| 150 |  |  |  |
| 180 |  |  |  |
| 210 |  |  |  |
| 240 |  |  |  |
| 270 |  |  |  |
| 300 |  |  |  |
| 330 |  |  |  |
| 360 |  |  |  |

1. Buat program untuk menampilkan tabel berikut ini dengan menggunakan looping for dan 2 buah array :

* X sbg array berdimensi satu yang diinisialisasi dgn sudut 0 s/d 360
* data sbg array berdimensi dua. data adalah array yg berisi nilai sin, cos dan tan dari masing-masing nilai sudut yang ada dalam array x.

Petunjuk :

* gunakan fungsi radian() untuk mengkonversi nilai derajat menjadi radian. rad = drjt / 180.0 \* PI.

PI = 3.14159

**Kesimpulan**

Array adalah kumpulan dari nilai-nilai data bertipe sama dalam urutan tertentu yang menggunakan sebuah nama yang sama. Elemen dari array array dapat kita akses dengan memanggil indeksnya. Array dapat memiliki dimensi sebanyak satu, dua, tiga atau lebih. Pada bahasa C, ukuran dari array tidak dapat diubah setelah dideklarasikan.