# BAB IX

# ARRAY

## Array

Selama ini kita menggunakan satu variabel untuk menyimpan 1 buah nilai dengan tipe data tertentu. Misalnya :

|  |
| --- |
| int a1, a2, a3, a4, a5; |

Deklarasi variabel diatas digunakan untuk menyimpan 5 data integer dimana masing masing variabel diberi nama a1, a2, a3, a4, dan a5. Jika kita memiliki 10 data, 100 data integer bahkan mungkin data yang ingin kita proses tidak kita ketahui atau bersifat dinamis? Kita tidak mungkin menggunakan variabel seperti diatas.

Solusi?? **→ Array**

Array adalah sebuah variabel yang menyimpan sekumpulan data yang memiliki tipe sama. setiap data tersebut menempati lokasi atau alamat memory yang berbeda-beda dan selanjutnya disebut dengan elemen array. Elemen array tersebut kemudian dapat kita akses melalui indeks yang terdapat di dalamnya namun penting sekali untuk diperhatikan bahwa dalam C++, Indeks array selalu di mualai dari 0, bukan 1.

Untuk mendeklarasian sebuah array kita harus menggunakan tanda **[..]**(bracket). Bentuk umum dari variabel array dapat ditulis seperti dibawah ini :

|  |
| --- |
| Tipe\_data nama\_variabel[indeks]; //ketentuan  int data[4]; //contoh |

Pada contoh diatas, data yang dapat ditampung yaitu 4 data. Indeks array selalu dimulai dari angka 0. Sehingga pada kasus diatas, data yang tersimpan dimulai dari indeks ke 0 hingga 3.

Namun anda juga dapat mendeklarasikan array sekaligus memberikan inisialisasi tanpa harus memberikan tambahan deklarasi ukuran array, sebagai berikut:

|  |
| --- |
| int daftarNilai [] = {10,9,8,10,9}; |

### Array 1 Dimensi

Contoh sebelumnya yang kita praktekkan yaitu merupakan array 1 dimensi. Cara diatas merupakan cara untuk melakukan inisialisasi. Untuk mengisikan data ke dalam element-element array kita dapat melakukannya langsung untuk setiap element , contohnya :

|  |
| --- |
| DATA[0]=1  DATA[1]=2  DATA[2]=3  DATA[3]=4  ......  Dst |

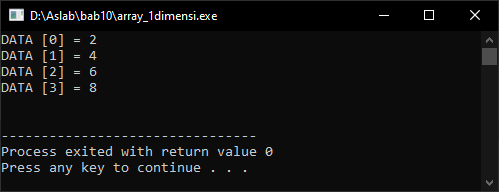
Namun cara di atas tidak efisien karena jika memerlukan data yang banyak harus menuliskan script yang banyak juga. Cara yang lebih umum dan banyak di gunakan untuk mengisikan nilai ke dalam element array adalah dengan menggunakan perulangan (looping). Contoh jika ingin mengisikan 4 elemen array, maka kita dapat menulis sintaks seperti berikut :

|  |
| --- |
| for(int index=0; index<4; index++){  cout<<"DATA["<<index<<"] = ";cin>>data[index];  } |

Contoh :

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {  int data[4];  int index; // mingisikan nilai ke dalam element array  for(index=0;index<4;index++)  {  cout<<"DATA ["<<index<<"] = ";cin>>data[index];  }  cout<<endl;  return 0;  } |

Tampilan:



Analisa :

Pada program diatas terdapat deklarasi array yang bertype data integer dengan jumlah array sebanyak 4. Untuk melakukan input data disertakan perulangan yang dimulai dari 0 sampai <4 yang berarti dari 0 sampai 3.

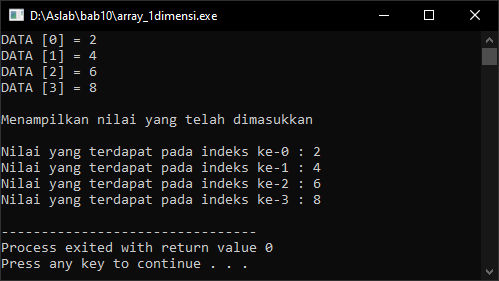
Setelah memahami cara mengisikan nilai ke dalam elemen array, sekarang kita memahami cara menampilkan nilai-nilai yang telah di inputkan ke dalam elemn-elemen tsb. Konsepnya sama, kita akan menggunakan perulangan untuk menampilkannya. Tambahkan syntaks berikut di bawah cout<<endl pada program diatas :

|  |
| --- |
| cout<<"Menampilkan nilai yang telah dimasukkan"<<endl<<endl;  for (index=0; index<4; index++)  {  cout<<"Nilai yang terdapat pada indeks ke-";  cout<<index<<" : "<<data[index]<<endl;  } |

Sehingga keseluruhan kode menjadi seperti di bawah:

|  |
| --- |
| #include <iostream>  using namespace std;  int main(){  int data[4];  int index; // mingisikan nilai ke dalam element array  for(index=0;index<4;index++){  cout<<"DATA ["<<index<<"] = ";cin>>data[index];  }  cout<<endl;  cout<<"Menampilkan nilai yang telah dimasukkan"<<endl<<endl;  for (int index=0; index<4; index++){  cout<<"Nilai yang terdapat pada indeks ke-";  cout<<index<<" : "<<data[index]<<endl;  }  return 0;  } |

Tampilan :



Analisa :

Setelah ditambahkan syntaks tersebut dapat menghasilkan output perulangan yang sudah diinputkan dan disimpan pada index tersebut.

### Array 2 Dimensi

Hampir sama dengan *array* 1 dimensi, yang membuat *array* 2 dimensi berbeda adalah daya tampung datanya. Jika di *array* 1 dimensi, kita perlu inisialisasi 1 angka untuk menunjukkan berapa data yang bisa dimasukkan. Di *array* 2 dimensi, kita bisa menginisialisasikan 2 angka. *Array* seperti ini sering di gunakan untuk pemerosesan matrik, Bentuk umum dari array 2 dimensi adalah sebagai berikut :

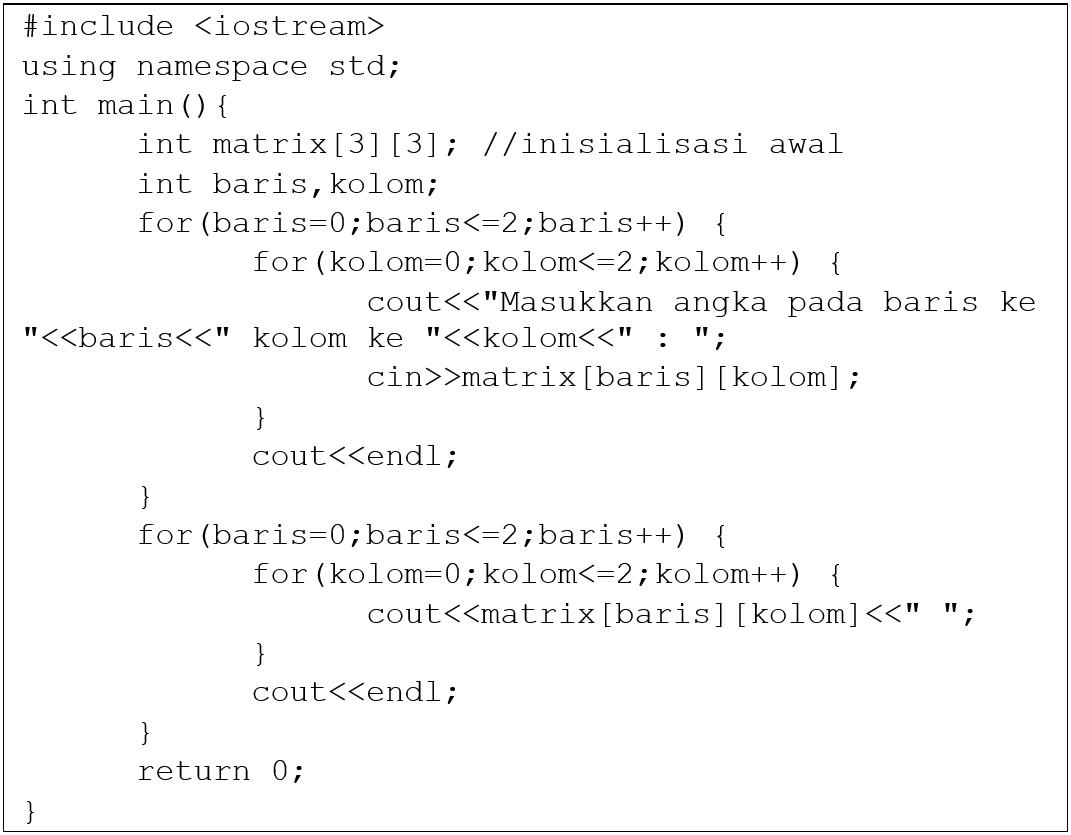
|  |
| --- |
| tipedata nama\_array[jmlh baris][jmlah kolom];  float bil[2][2]; |

Ilustrasi tabel array sebagai berikut :

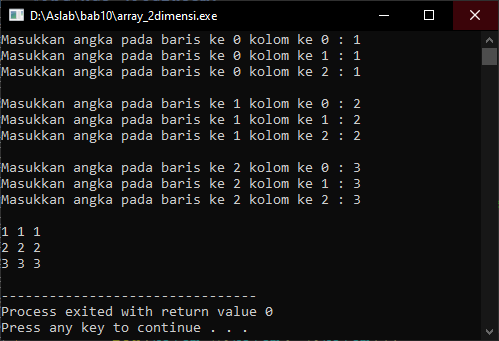
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [0][0] | [0][1] | [0][2] | [0][3] | [0][4] |
| [1][0] | [1][1] | [1][2] | [1][3] | [1][4] |

Contoh diatas menunjukkan bahwa kita dapat mengisi data dari array 00, 01, 02, 03, 04, 10, 11 dan seterusnya. Sehingga, data yang diisi totalnya menjadi 2 x 5 = 10 data.

Contoh :



Tampilan :



Analisa :

Dengan menggunakan array 2 dimensi dapat mengeluarkan output 111 pada baris ke 0 dan kolom 0,1,2 dan seterusnya yang sudah diiputkan sesuai dengan nilai pada matrix ordo 3 x 3.

|  |  |
| --- | --- |
| **#taugaksih**  **String**  String merupakan bentuk data yang biasa dipakai dalam bahasa pemrograman  untuk keperluan menampung dan memanipulasi data teks, misalnya untuk menampung (menyimpan) suatu kalimat. Pada bahasa C, string bukanlah merupakan tipe data tersendiri, melainkan hanyalah kumpulan dari nilai-nilai karakter yang berurutan dalam bentuk array berdimensi satu.  Contoh Program :   |  | | --- | | #include <iostream>  using namespace std;  int main ()  {  string a;  cout<<"Masukan string = ";cin>>a;  cout<<endl;  for(int b=0;b<a.length();b++){  cout<<"string index ke-"<<b<<" = "<<a[b]<<endl;  }  } |   Tampilan Program :    Analisa :  Pada program di atas adalah pembuktian bahwa string sebenarnya merupakan array berdimensi 1 yang ditampilkan di dalam sebuah looping berdasarkan indexnya, a.length() di gunakan untuk mengetahui banyak karakter string dan hasilnya berupa integer digunakan untuk batas perulangan. |

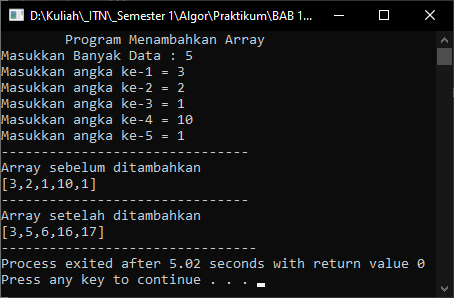
**Latihan!**

1. Buatlah Program Penjumlahan Array, dengan nilai akhir merupakan penjumlahan dari semua array sebelumnya.

Jika input data=[3,2,1,10,1]

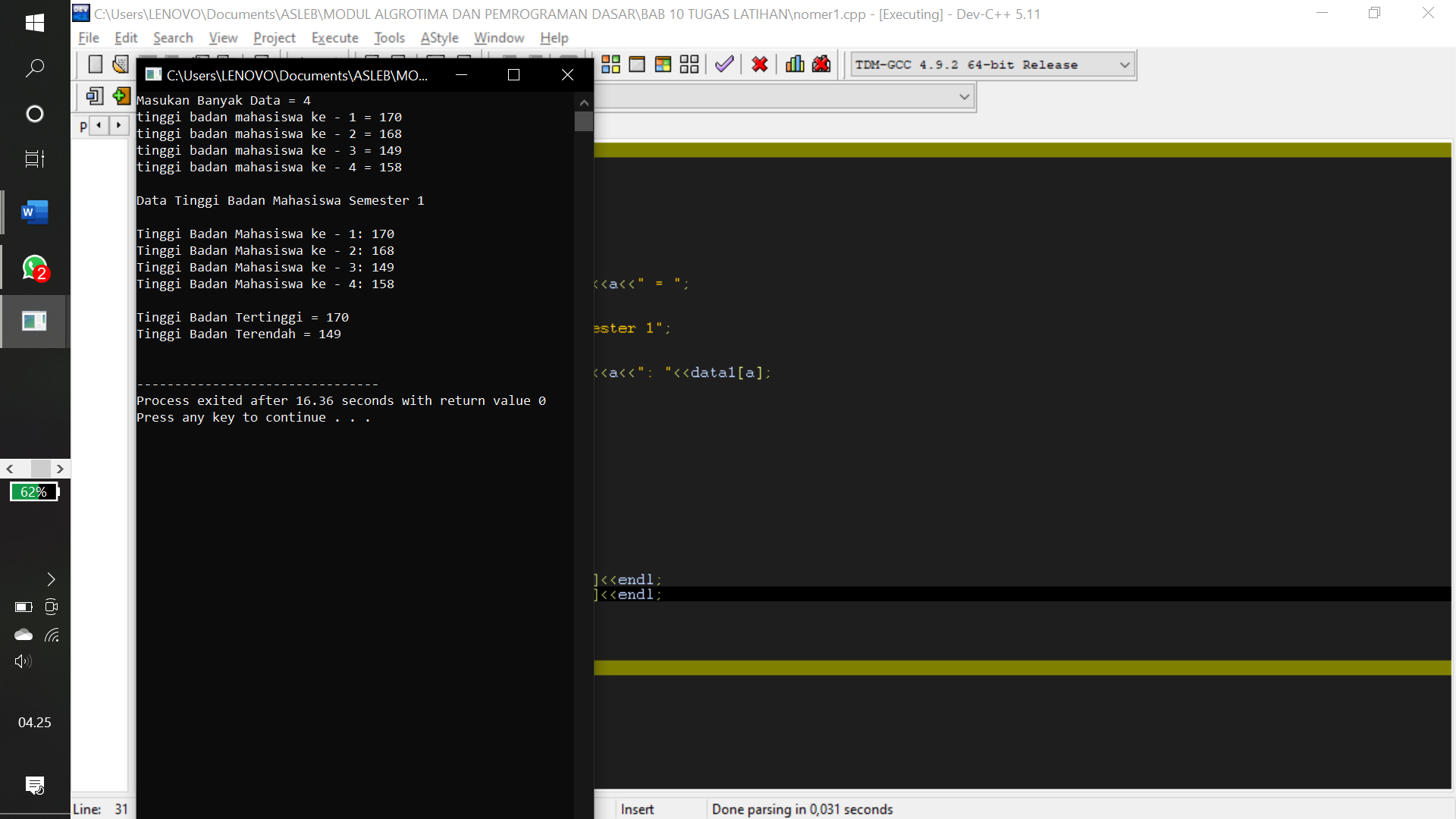
Maka output data=[3,5,6,16,17]

Dengan penjumlahan seperti [3, 3+2, 3+2+1, 3+2+1+10, 3+2+1+10+1]



1. Buatlah program inputan tinggi badan mahasiswa dalam 1 kelas. Kemudian tentukan tinggi badan tertinggi dan terendah, seperti gambar dibawah!

\*opsional(jika bisa menampilkan Rata-Rata Tinggi Badan Nilai++++++)



1. Buatlah program menentukan grafik poin dengan (\*) perdasarkan inputan user

