Structures dan array adalah penggunaan dari dua fitur bahasa pemrograman C secara bersamaan. Structures adalah kumpulan variabel dengan tipe data yang berbeda dan dijadikan sebagai anggota atau elemen untuk deklarasi struct tersebut. dan array adalah variabel larik yang memungkinkan kita untuk menyimpan data lebih dari satu dalam satu variabel dan satu tipe data. Ada dua cara untuk menggunakan array pada structures. Yang pertama adalah penggunaan array untuk anggota structur dan kedua adalah penggunaan array untuk object.

Array pada Anggota / Elemen Structur

Dengan cara ini kita dimungkinkan untuk memiliki struct dengan salah satu elemenya dapat menyimpanan data lebih dari satu dan terstruktur.

Bentuk penulisan

//Deklarasi struct

struct identifier{

tipeData identifier[index];

…

} ;

//Deklarasi Object

struct identifierStruct indentifierObject

Contoh penulisan

struct a{

int bilanganArr[4];

int bilangan;

};

struct a a\_object;

Cara Inisialisasi

Untuk memberi nilai awal pada anggota berjenis array pada struct. Sama seperti kita memberi nilai awal pada penjelasan “Nesting Struct”. Array akan memiliki satu tempat elemen pada inisialisasi object dipisahkan dengan tanda koma (,) dengan elemen struct lainya, dan akan ditulis di dalam sepasang tanda “{“ dan “}”.

struct a a\_object = {{3,4,2,7},99};

Cara Mengakses

Dari contoh diatas kita telah mendirikan sebuah struct dengan nama ‘a’, yang didalamnya terdapat anggota atau elemen berjenis variabel larik/array dengan nama “bilangan” dan mendirikan object bernama “a\_object”. untuk mengakses elemen dari struct dibutuhkan “Member Access Operator (.)” dan juga untuk mengakses setiap elemen pada array kita membutuhkan nomer elemen yang diapit oleh tanda “[“ dan “]”.

a\_object.bilanganArr[3] = 6;

Contoh program

#include <iostream>

using namespace std;

struct a{

int bilanganArr[4];

int bilangan;

};

int main() {

//Inisialisasi

struct a a\_object={{9,8,7},1};

//Pemberian nilai

for(int i=0;i<4;i++){

a\_object.bilanganArr[i] = i;

}

a\_object.bilanganArr[2]=222;

a\_object.bilangan=9;

//Output

for(int i=0;i<4;i++){

cout<<"bilanganArr ["<<i<<"] = "<<a\_object.bilanganArr[i]<<endl;

}

cout<<endl<<"bilangan = "<<a\_object.bilangan<<endl;

return 0;

}

Array pada Object

Penggunaan Array pada Object memungkinkan kita untuk menyimpan data dengan beberapa tipe data yang berbeda pada struct dan object itu sendiri memiliki banyak tumpukan struct terstruktur yang disebut sebagai array.

struct b{

int bilanngan;

char karakter;

}

struct b b\_object[2];

Cara inisialisasi’

Caranya masih sama seperti inisialisasi “array pada anggota struct”. Dalam inisialisasi array pada object, setiap elemen array akan dipisahkan dengan tanda koma (,) dan di tulis di dalam sepasang tanda “{“ dan “}”.

struct b b\_object[2]={ {9,'a'},{8,'b'} };

Cara Mengakses

Untuk mengakses array dibutuhkan sepasang tanda “[“ dan “]” untuk memnentukan jumlah elemen array. karena disini objectlah yang menggunakan array maka dibutuhkan tanda “[” dan “]” sesudah penulisan object dan selanjutnya dilanjutkan dengan penulisan penentu elemen struct.

b\_object[1].bilangan=2;

b\_object[0].karakter='z';

Contoh program

#include <iostream>

using namespace std;

struct b{

int bilangan;

char karakter;

};

int main() {

//Inisialisasi

struct b b\_object[2]={{9,'a'},{8,'b'}};

//Pemberian nilai

b\_object[0].bilangan=1;

b\_object[1].bilangan=2;

b\_object[0].karakter='z';

b\_object[1].karakter='y';

//Output

for(int i=0;i<2;i++){

cout<<"b\_object["<<i<<"].bilangan = "<<b\_object[i].bilangan<<endl;

cout<<"b\_object["<<i<<"].karakter = "<<b\_object[i].karakter<<endl;

}

return 0;

}