Di dalam structures kita dibolehkan untuk mengunakan tipe data struct untuk anggota struct hal itu disebut Nested Structures. Nested Structures adalah penggunaan tipe data itu sendiri atau strut yang telah dideklarasikan sebelumnya sebagai anggota dari bagian struct. Jadi struct tersebut akan memiliki anggota variabel yang bertipe data struct itu sendiri atau struct yang sudah didirikan, dan anggota variabel yang menggunakan tipe data struct akan memiliki semua anggota struct yang dimaksud, di dalamnya.

Contoh penulisan

struct anak{

string nama[50];

int umur;

};

struct siswa{

struct anak anak;

int noInduk;

float niaiUjian;

};

Cara Inisialisasi Nested Structures

inisialisasi nested structures mirip seperti cara kita inisialisasi array dimensi banyak (lebih dari satu). Untuk inisialisasi anggota dari struct tersebut yang bertipe data struct kita bisa melakukanya dengan menggunakan tanda { dan }. Dan pengisianya mirip seperti inisialisasi struct biasanya.

struct siswa budi{{"Budi ini Budi", 19}, 123, 70.5};

Cara Akses Nested Structures

untuk mengakses anggota struct kita membutuhkan tanda ”Member Access Operator ( . )” di antara nama object, Elemen pada object tersebut dan nama elemen dari tipe data yang digunakan elemen object variabel struct.

strcpy(tono.anak.nama, "Tono Ono Loro");

tono.anak.umur=18;

Contoh program

#include <iostream>

#include <string.h>

using namespace std;

struct anak{

char nama[50];

int umur;

};

struct siswa{

struct anak anak;

int noInduk;

float nilaiUjian;

};

void printIt(struct siswa a){

cout<<"Nama \t\t = "<<a.anak.nama<<endl;

cout<<"Umur \t\t = "<<a.anak.umur<<endl;

cout<<"Nomer Induk \t = "<<a.noInduk<<endl;

cout<<"Nilai Ujian \t = "<<a.nilaiUjian<<endl<<endl;

}

int main() {

//Inisialisasi

struct siswa budi{{"Budi ini Budi", 19}, 123, 70.5};

struct siswa tono;

//Assignment Operator

strcpy(tono.anak.nama, "Tono Ono Loro");

tono.anak.umur=18;

tono.noInduk=124;

tono.nilaiUjian=7.8;

//Output

printIt(budi);

printIt(tono);

return 0;

}

Mendeklarasikan nested structures, dengan anggota variabel struct itu sendiri, tidak akan bekerja jika kita mendeklarasikan hanya sebagai variabel biasa. Untuk melakukan itu dibutuhkan deklarasi pointer. dan nested structures dengan struct itu sendiri memungkinkan kita untuk menyimpan data secara dinamis. Untuk penjelasan tentang nested structures dengan anggota struct itu sendiri penulis akan sampaikan pada artikel selanjutnya, pada penjelasan linked list.