AGUSTUS 2024

MODUL

PRAKTEK DASAR C++



| Dibuat Oleh | : | Joni Iskandar, S.Kom,.M.Kom |
| --- | --- | --- |

SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER

STMIK – INSAN PEMBANGUNAN

JL. RAYA SERANiG KM. 10 BITUNG – TANGERANG

Pengertian Algoritma & Pemrograman

Algoritma berasal dari algiris dan ritmis , pertama kali diungkapkan oleh Abu Ja’far Mohammad Ibn Musa Al Khowarizmi (825M) , yang berarti suatu metode khusus yang tepat dan terdiri dari langkah-langkah secara terstruktur

Jadi Algoritma itu adalah urutan logis pengambilan keputusan untuk pemecahan masalah dengan langkah-langkah secara terstruktur.

Ciri-ciri Algoritma yang baik :

* Algoritma memiliki logika perhituangn atau metode yang tepat untuk menyelesaikan masalah
* Menghasilkan output yang tepat dan benar dalam waktu singkat
* Algoritma ditulis dengan bahasa standar secara sistematis
* Algoritma ditulis dengan format yang mudah dipahami dan mudah diimplementasikan
* Semua operasi yang dibutuhkan terdefinisi dengan jelas
* Semua proses dalam algoritma harus berakhir setelah sejumlah langkah dilakukan

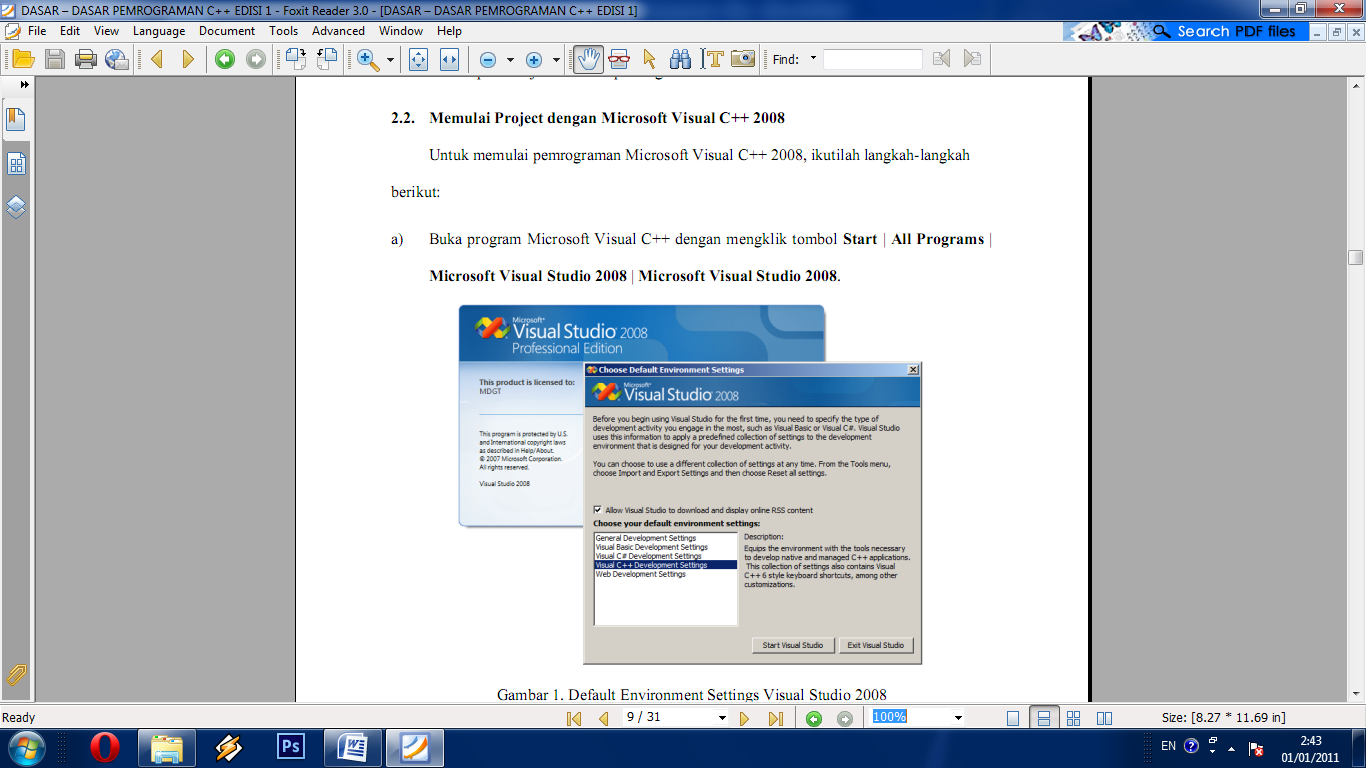
Materi 1

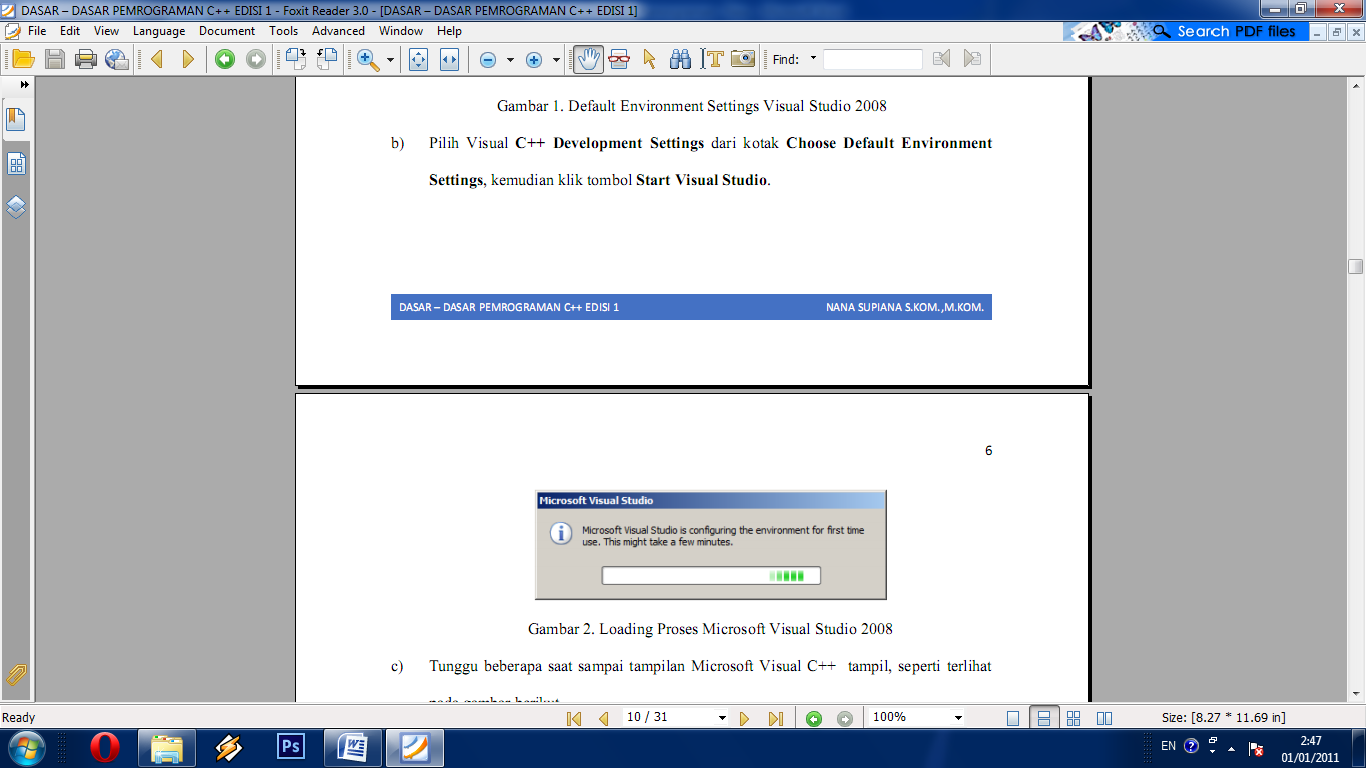
1. PERKEMBANGAN C++

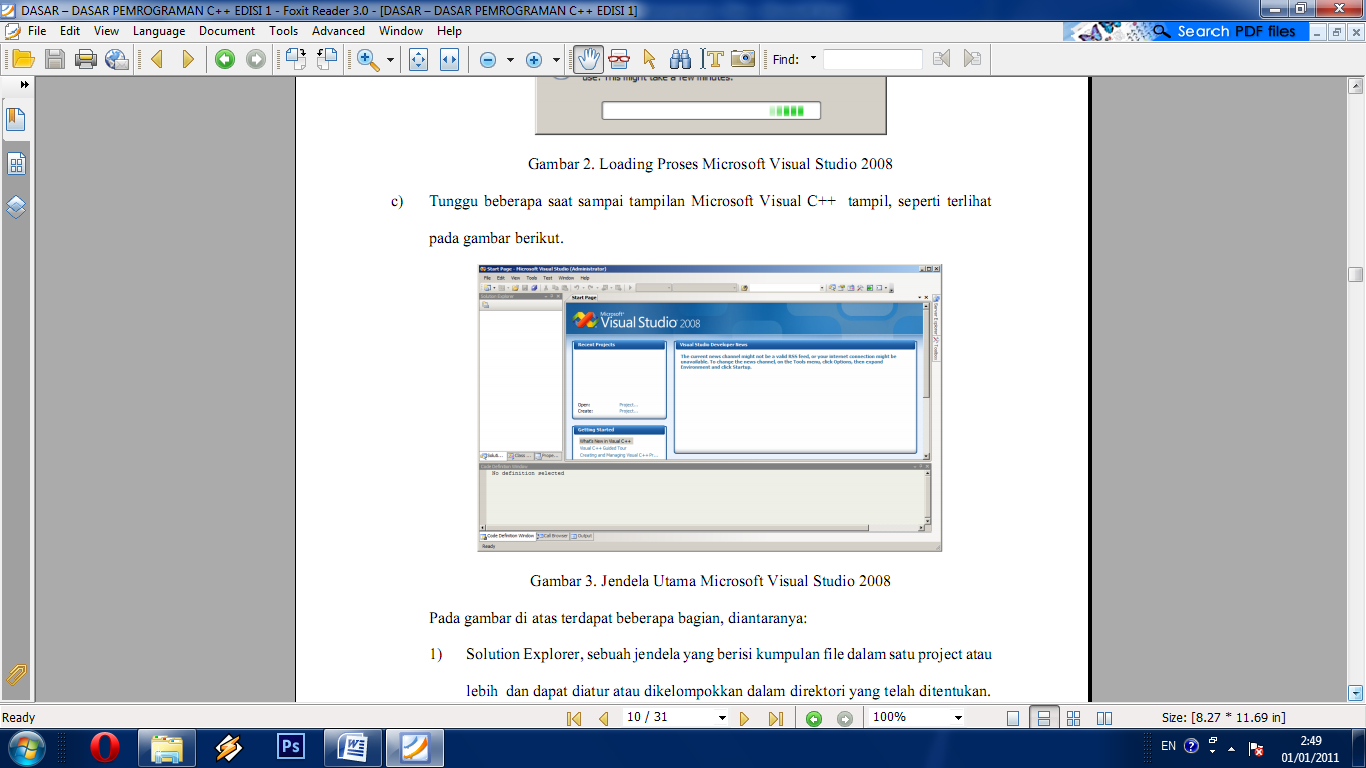
Bahasa C dan C++ merupakan bahasa pemrograman terstruktur yang membagi program dalam bentuk sejumlah blok.

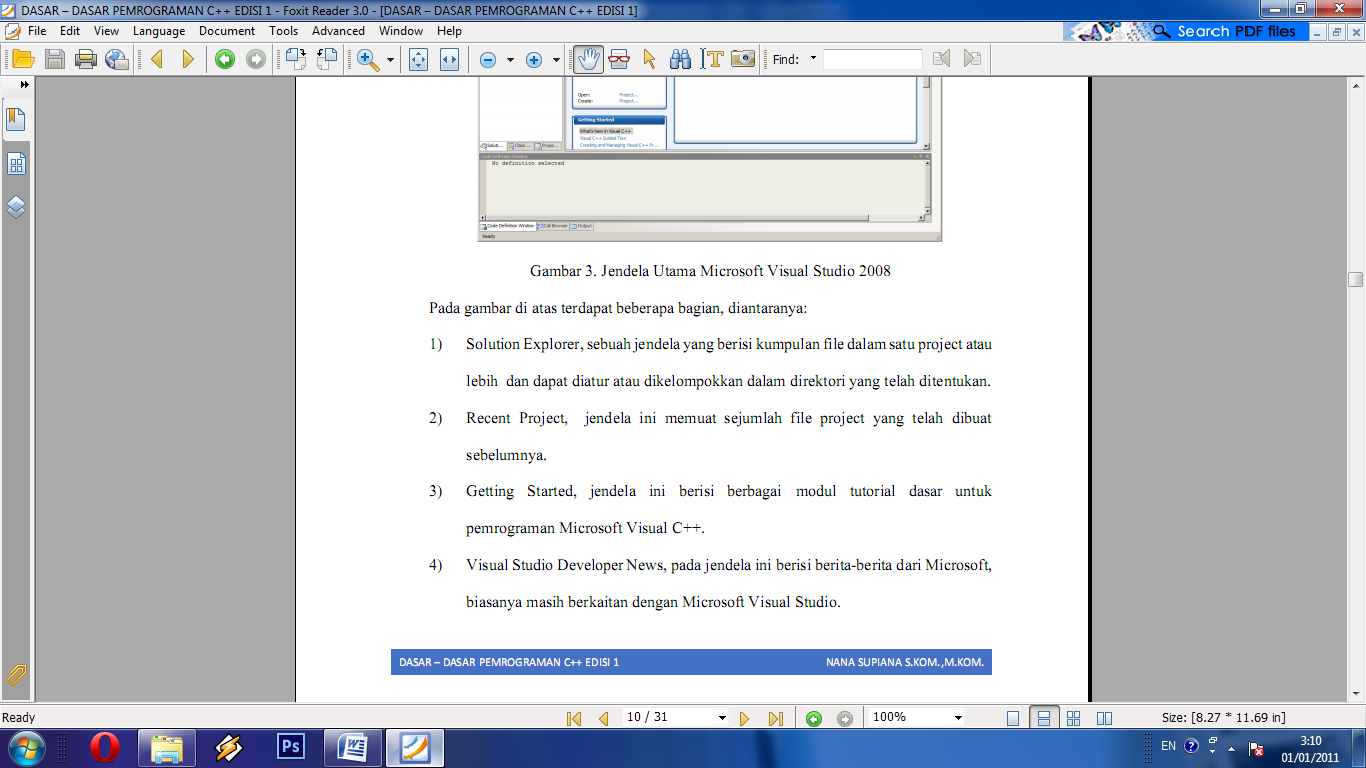
* Tujuannya adalah untuk memudahkan dalam pembuatan dan pengembangan program.
* Keistimewaan yang sangat berarti dari bahasa C++ adalah karena bahasa mendukung pemrograman yang berorientasi objek (PBO)/Oriented Object Programming (OOP).
* Program C++ biasa ditulis dengan nama ekstensi.CPP (dari kata C plus plus).
* Agar program ini dapat dijalankan (Eksekusi), program harus dikompilasi terlebih dahulu dengan menggunakan compiler C++.
* Pada saat pengkompilasi, program sumber (.cpp) bersama file-file header (bereksistensi .h atau .hpp).

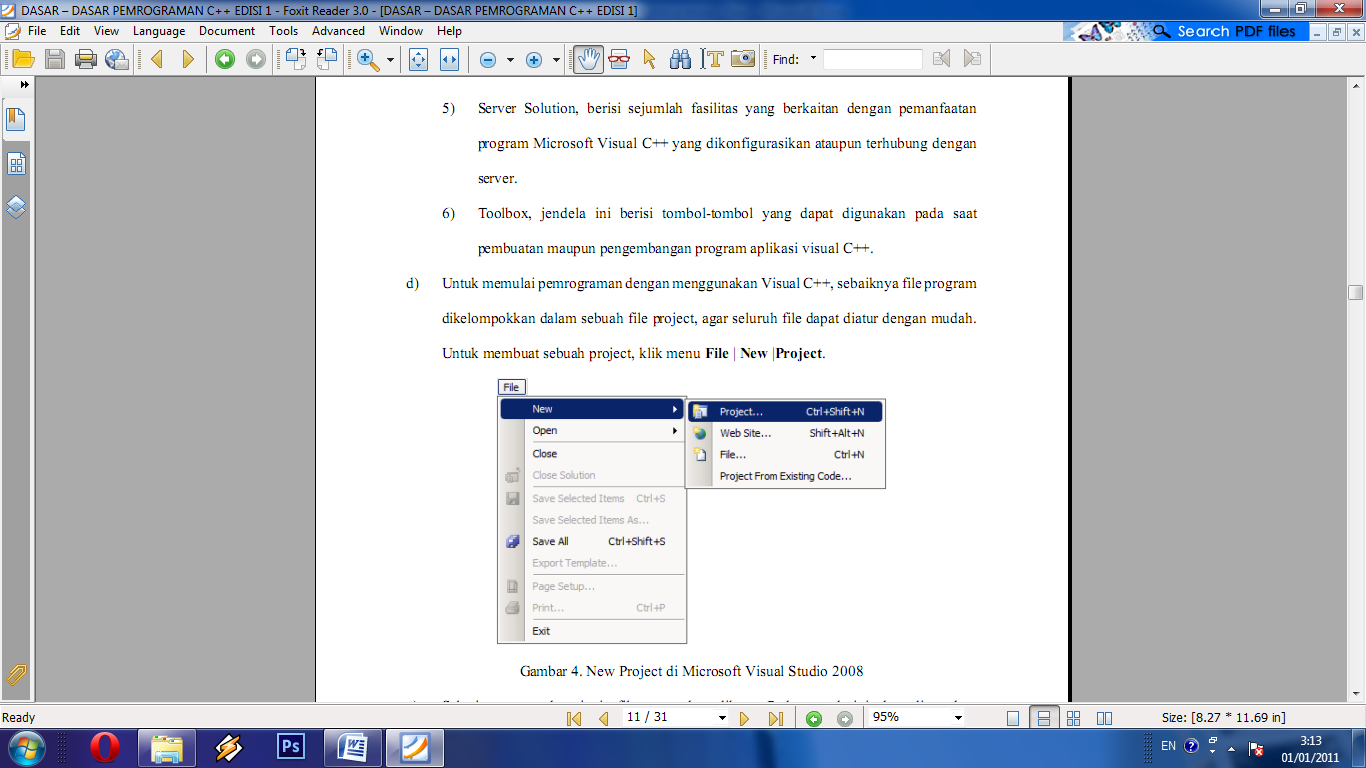
Menjalankan Program Visual C++

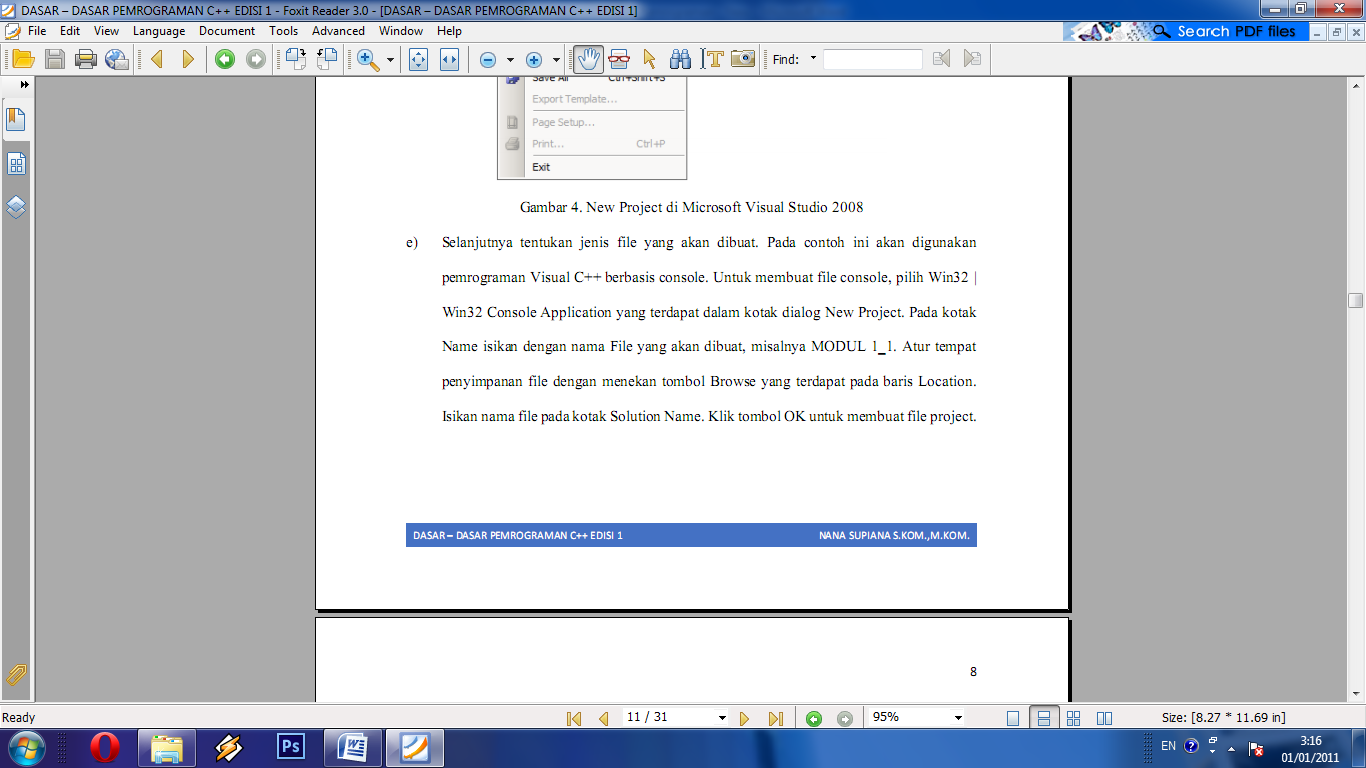


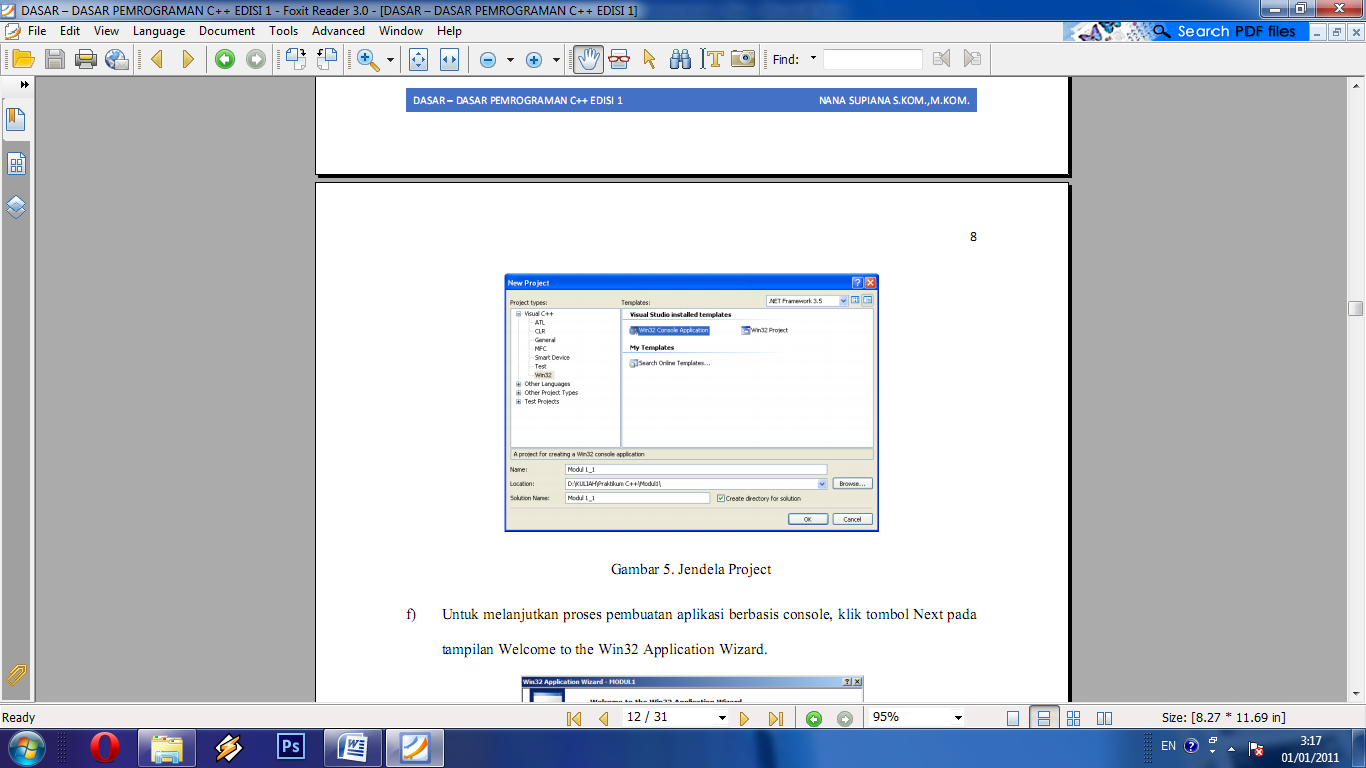


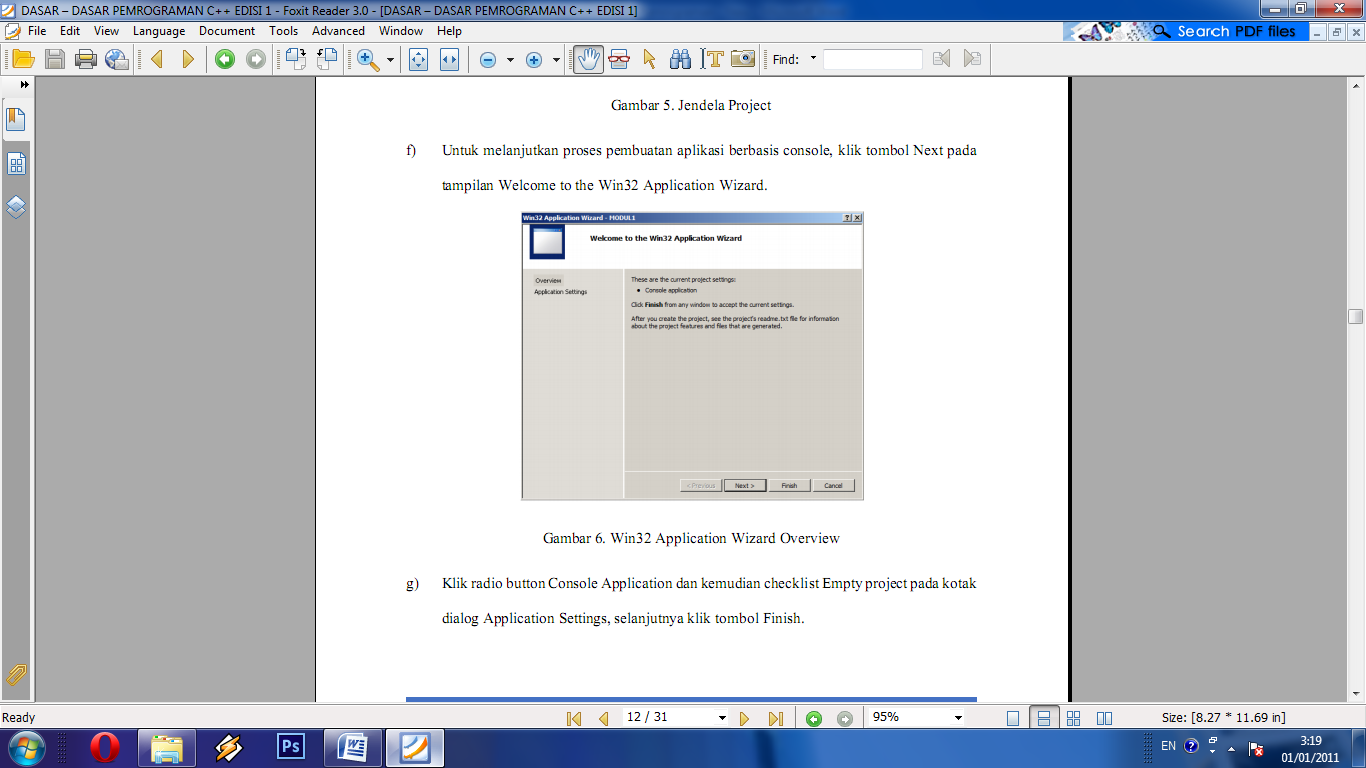


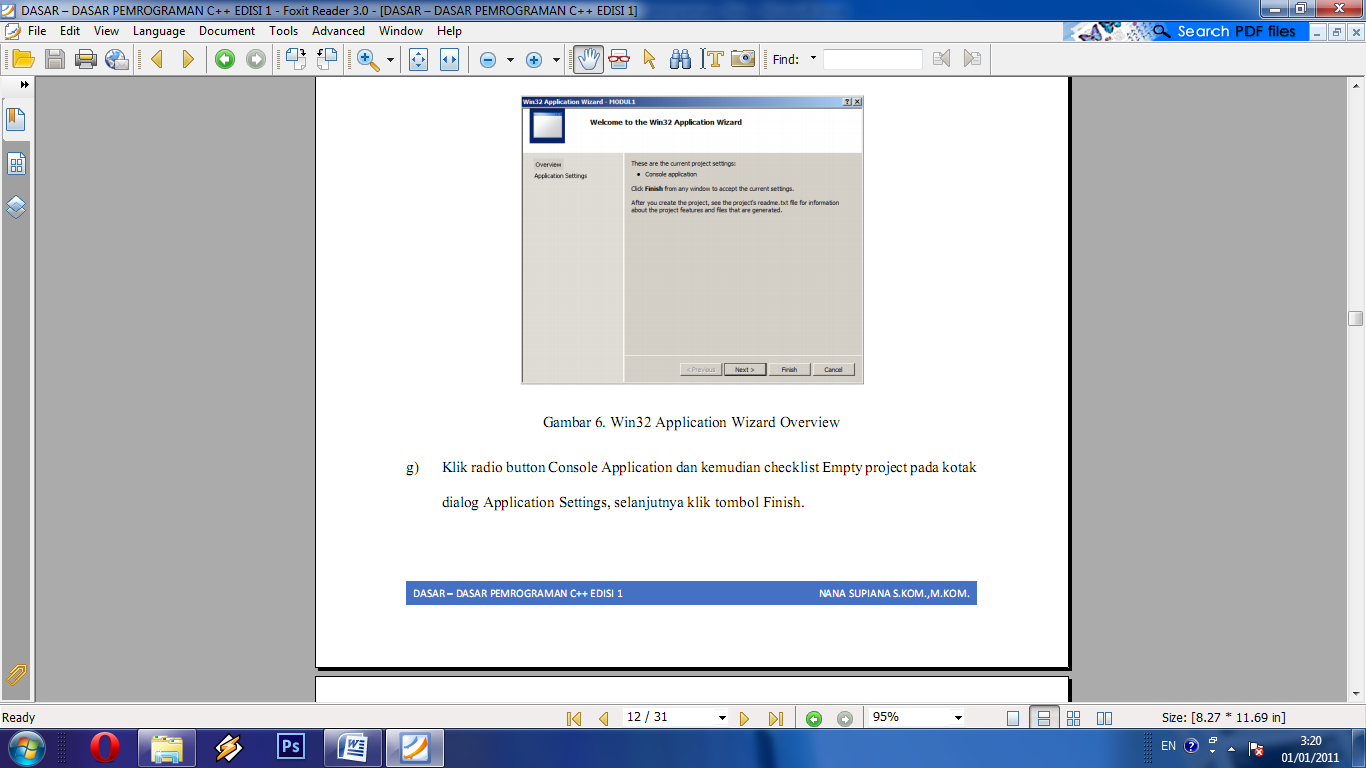


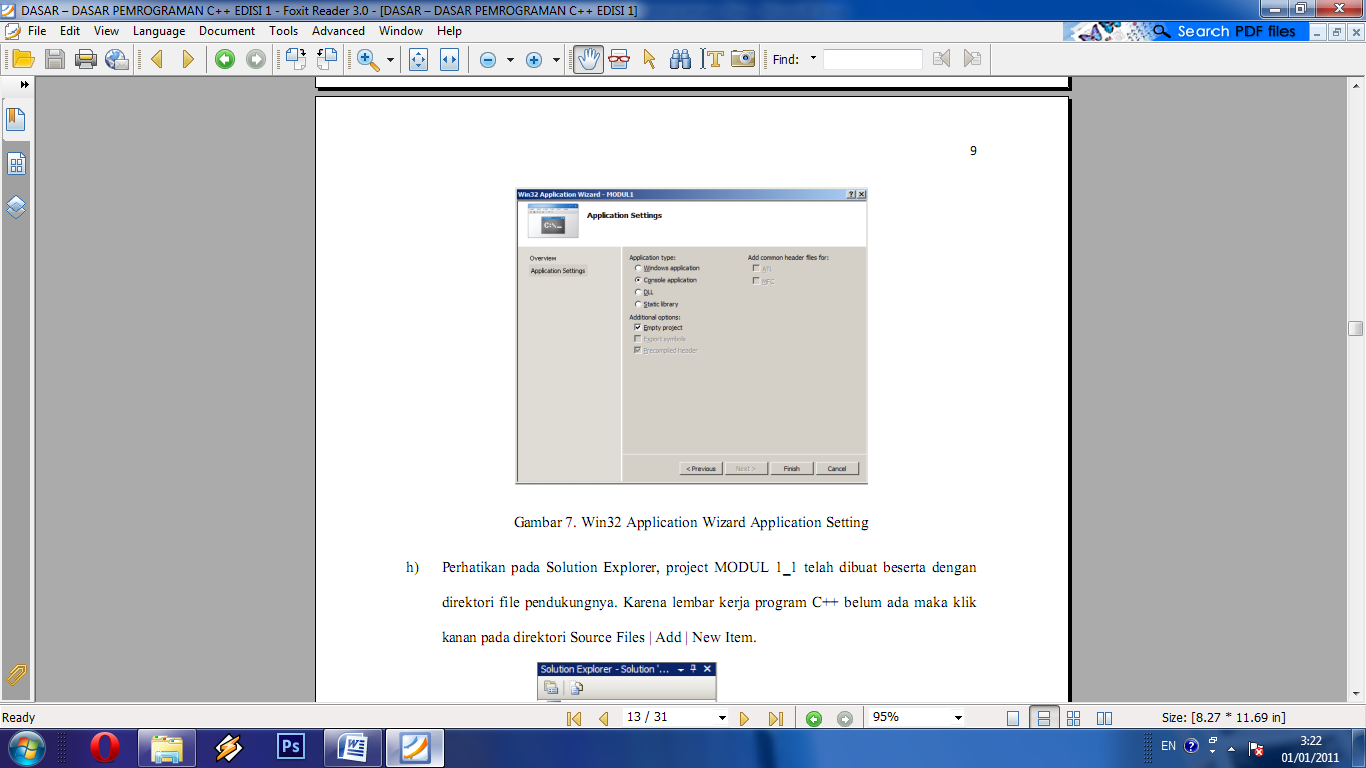


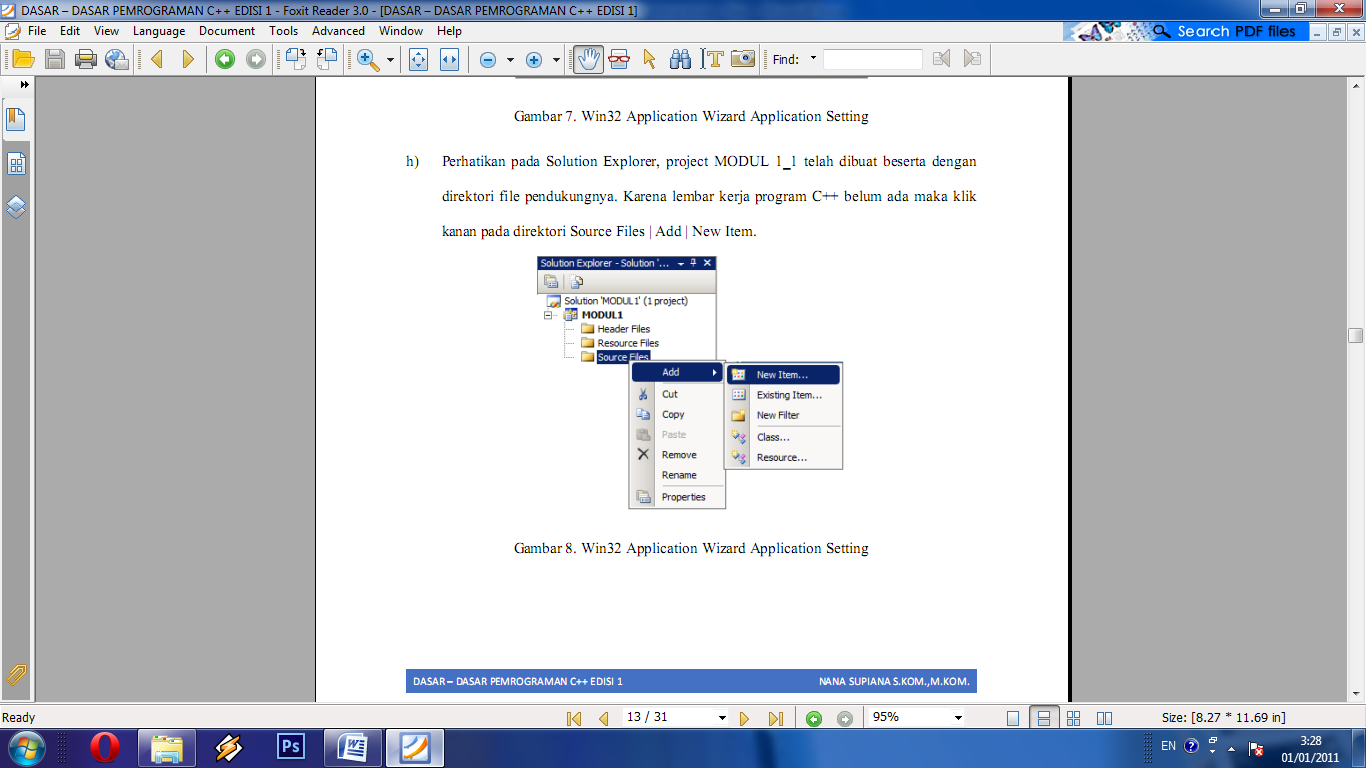


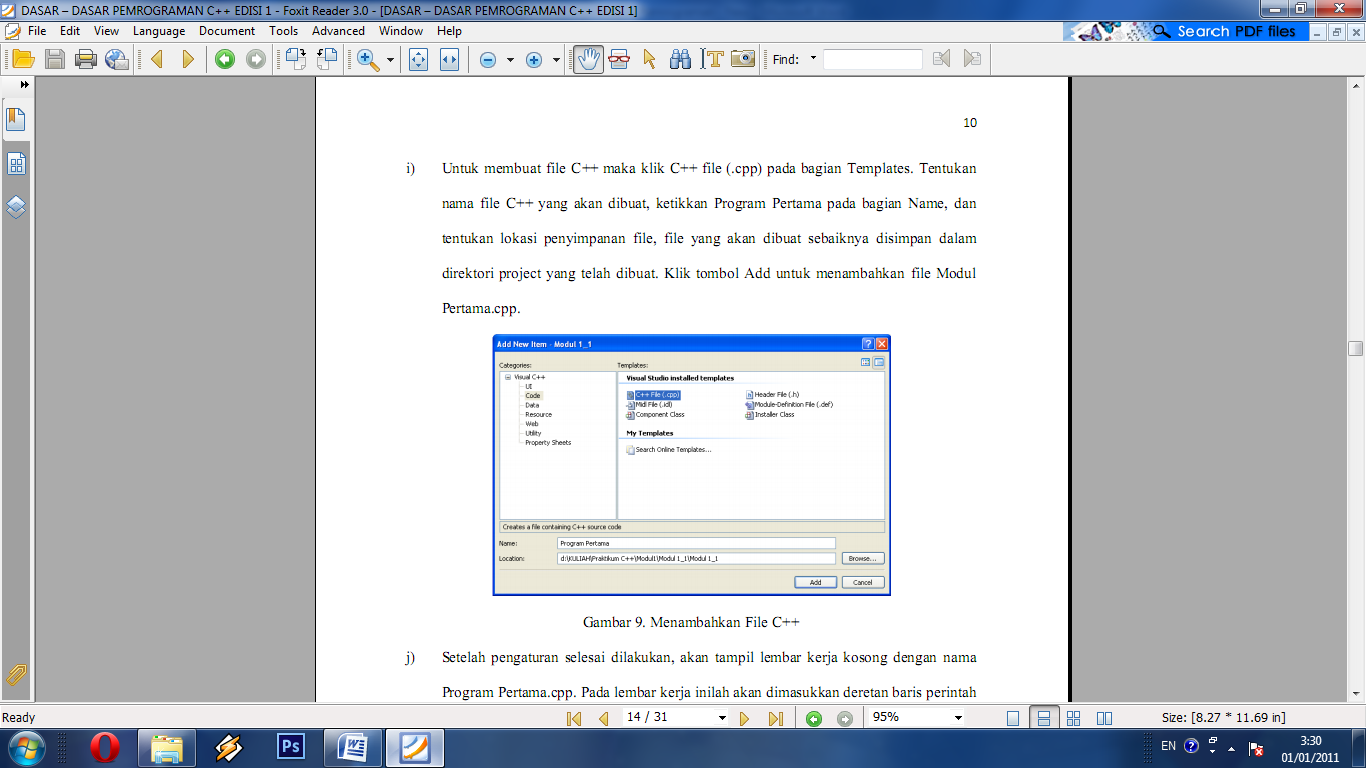


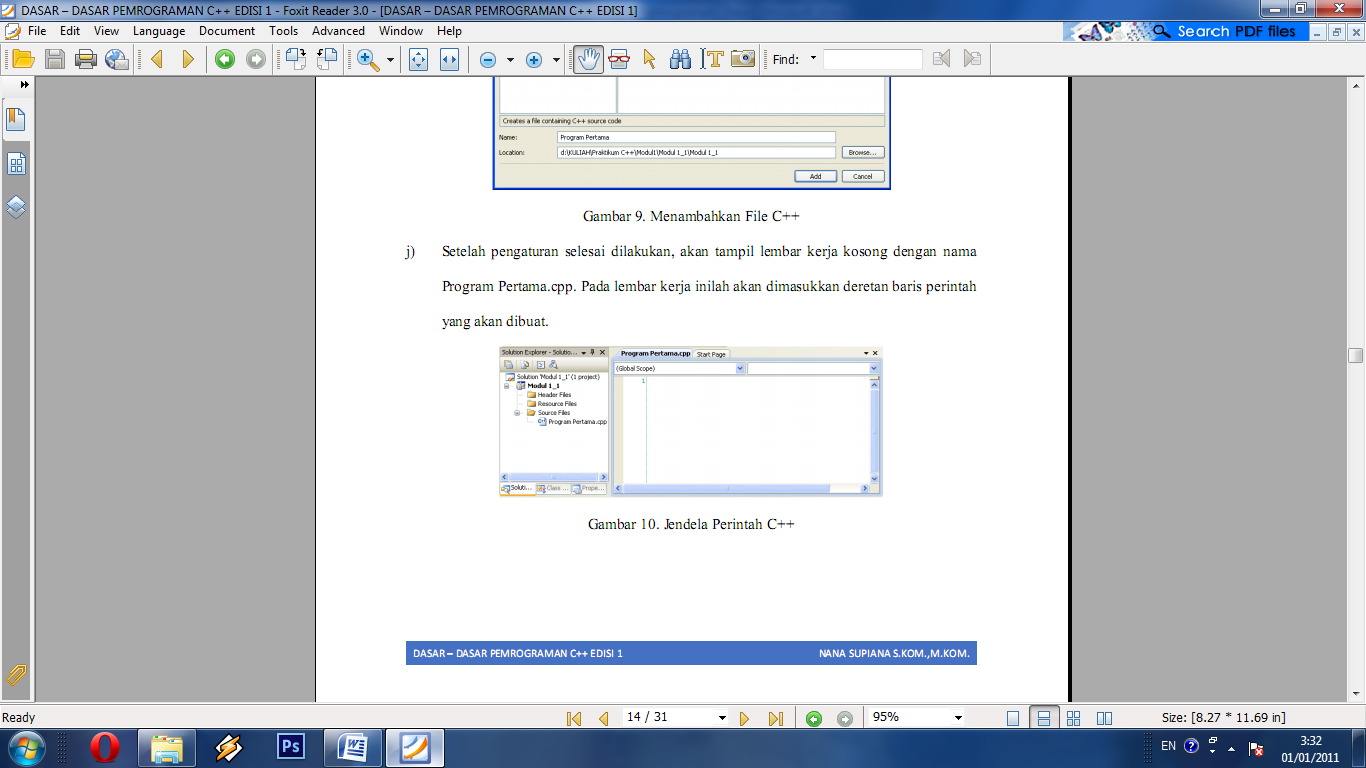


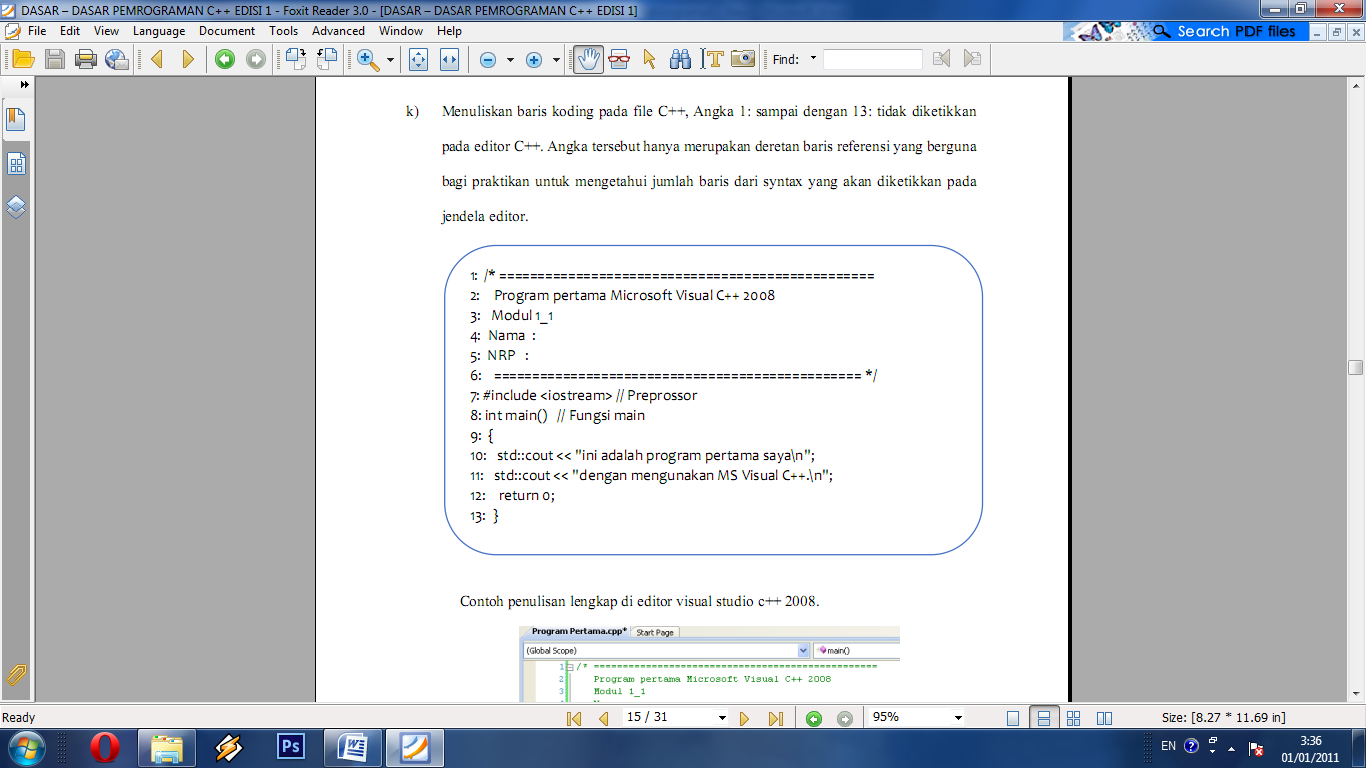


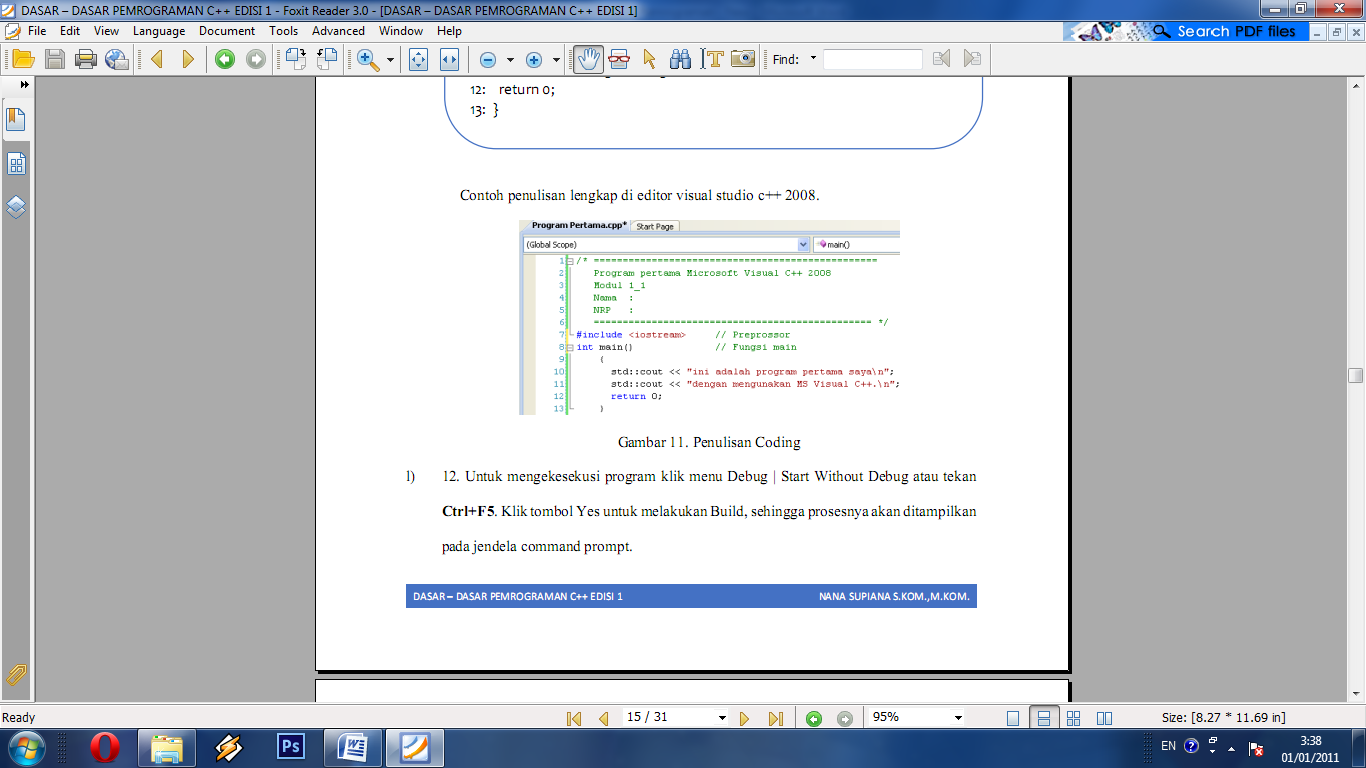


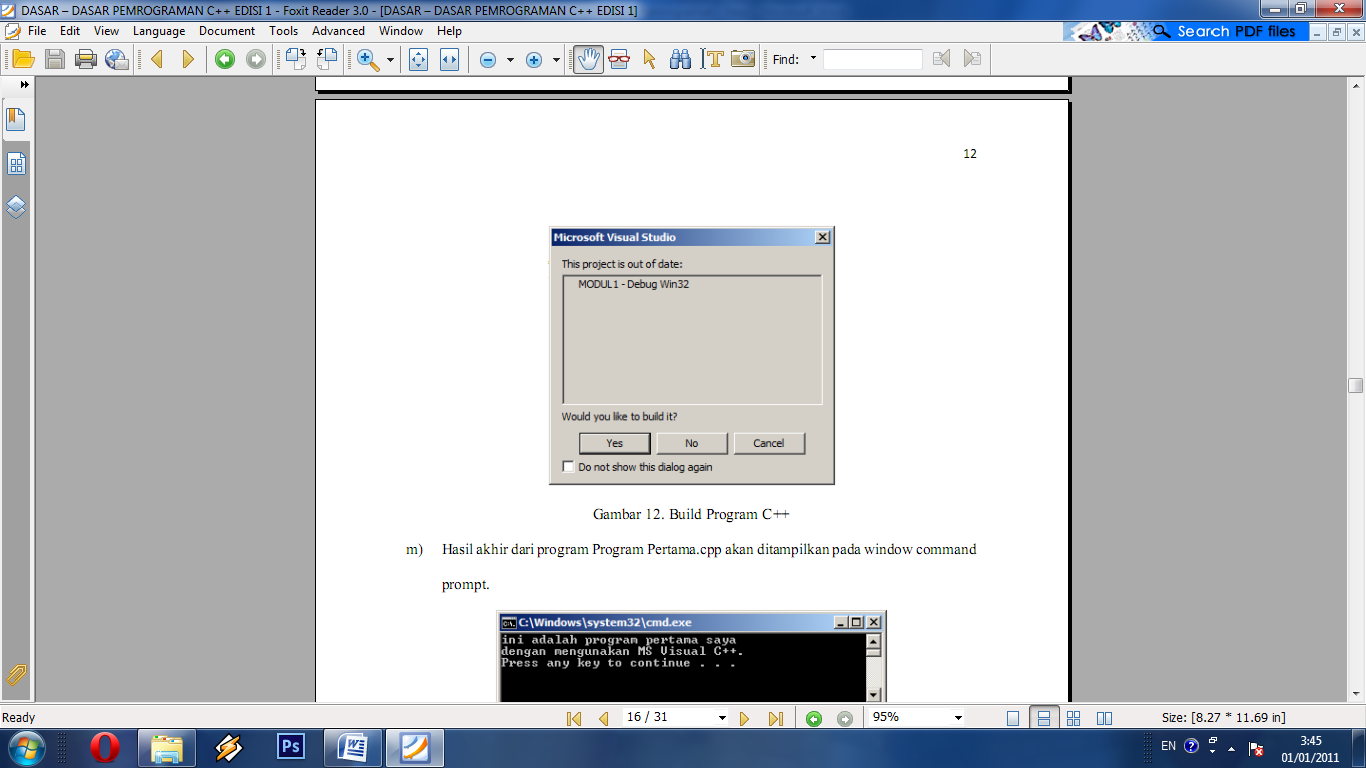


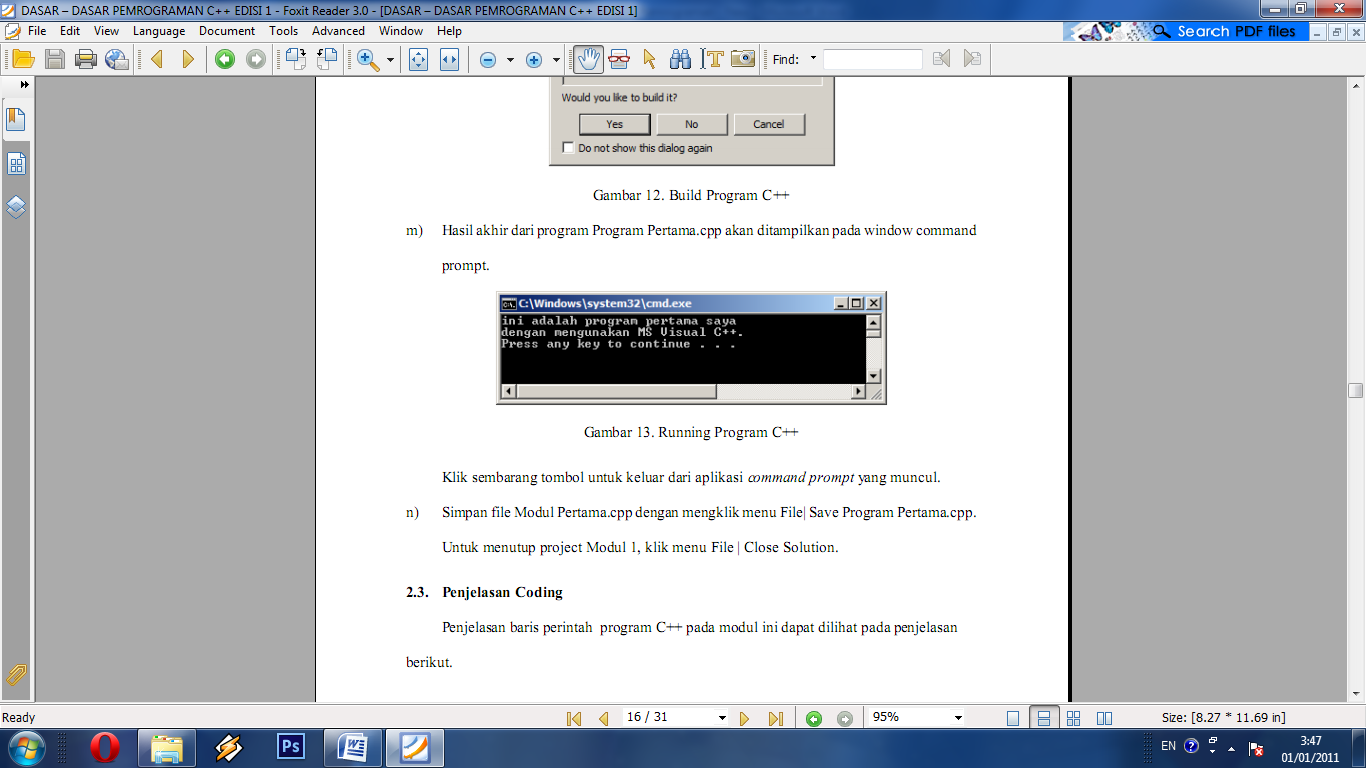












1. **Penulisan Program**

**Contoh program1: lat1**

//Program Tampil Pesan

#include <iostream>

using namespace std;

int main()

{

cout<<”Selamat Datang di Program C++”<<endl;

cout<<”--------------------------“<<endl;

cout<<” Nama Siswa : Andi. S”<<endl;

cout<<” Semester : 3”<<endl;

cout<<” Jurusan : Sistem Informasi”<<endl;

return 0;

}

**Contoh Program 2 : Lat2**

//Program tampil

#include <iostream>

using namespace std;

int main()

{

cout<<”Bahasa Pemrograman C++”<<endl;

cout<<”STMIK Insan Pembangunan”<<endl;

return 0;

}

1. TAMPILAN PROGRAM

Didalam tampilan pemrograman C++ menyediakan fasilitas IDE (Integratec Development Environment), dimana semua proses pembuatan program dapat dilakukan melalui fasilitas ini. Program ini memiliki modus teks (CUI/Character User Interface) dan modus grafis (GUI/Graphical User Interface).

Dalam penulisan pemrograman C++ bersifat case sensitive, Maksudnya adalah penulisan huruf besar dan kecil dianggap berbeda, contoh : Nama\_Siswa tidak sama dengan nama\_siswa.

1. STRUKTUR PENULISAN PROGRAM

Penulisan program C++ dapat dilihat sbb:

Void main()

{

Statement-statement

}

Fungsi-fungsi lain

{

Statement-statement

}

1. BAGIAN-BAGIAN PADA STRUKTUR PENULISAN

a). File Header

Merupakan file-file yang berisi berbagai deklarasi, seperti fungsi, variable, dsb.

File header diikuti dengan kata #include didepannya yang akan mengintrusikan compiler untuk menyisipkan file library missal:

<iostream.h> deklarasi yg diperlukan oleh Cout,

<iomanip.h> fungsi untuk memformat output,

<conio.h> untuk menampilkan perintah getch dsb.

b). Fungsi Main()

Program C++ memang tidak pernah lepas dari fungsi, sebab fungsi adalah suatu dasar dalam program C++.

Program C++ minimal mengandung sebuah fungsi yaitu fungsi main().

Fungsi main() merupakan fungsi yang menjadi awal dan akhir eksekusi program C++

Contoh program :

//Program tampil

#include <iostream.h>

Using namespace std;

int main()

{

cout<<”Bahasa Pemrograman C++”<<endl;

cout<<”STMIK Insan Pembangunan”<<endl;

return 0;

}

Penjelasan program :

* **//**🡪 merupakan tanda komentar dalam program yang tidak akan diproses/eksekusi
* **#include (file header)**🡪 merupakan instruksi kepada compiler untuk menyisipkan file pustaka/library (<iostream.h>, <conio.h>, <iomanip.h>) dengan nama ekstension filenya adalah .h
* **<iostream.h>**🡪 file library/pustaka yang perlu disertakan pada program yang melibatkan obyek **cout** karena berisi deklarasi yang diperlukan cout yang berhubungan dengan masukan dan keluaran.
* **Using namespace std 🡪** mendeklarasikan/memberitahukan compailer bahwa kita akan menggunakan semua fungsi/class/file yang terdapat dalam namespace std, seperti ; cin dan cout.
* **int main()** 🡪 merupakan fungsi main() yang merupakan awal dan akhir dari pembuatan program, kata void dipakai untuk menyatakan bahwa fungsi ini tidak memiliki nilai balik (return value)
* Semua yang terletak didalam tanda { } disebut blok, tanda { untuk menyatakan awal eksekusi program dan tanda } merupakan akhir dari eksekusi program.
* **cout**🡪 obyek yang disediakan C++ untuk mengarahkan data ke standard output (normalnya layar)/menampilkan pernyataan/statement yang terletak pada sepasang tanda petik **(“ “)**
* **Tanda <<**🡪 merupakan operator “penyisipan/peletakan pernyataan/statement”
* Setiap pernyataan harus diakhiri dengan tanda titik koma (;)
* **Endl (end line)**🡪 digunakan untuk mengakhiri baris pernyataan dan akan berpindah baris baru dan bukan tersambung.
* **Return 0** 🡪 menutup main() dan mengakhiri program yang sudah ditulis.

Materi 2

1. PENJELASAN PEMROGRAMAN C++

Dalam program C++ tidak lepas dengan identifier (pengenal) dalam menyusun program. Identifier adalah suatu nama yang biasa dipakai dalam pemrograman C++ untuk menyatakan :

Variabel, Konstanta, Tipe Data, Fungsi, Label, Objek , dsb.

1. Variabel

Merupakan sebuah penamaan dalam program yang nilainya dapat diubah saat pelaksanaan program . Pendeklarasiaan variable harus diikuti dengan jenis tipe data yang didefinisikan.

Contoh : tipe\_data nama\_variabel; (contoh : int siswa; )

Penamaan variable mempunyai syarat :

1. Tidak boleh ada spasi, contoh : data pegawai (salah), data\_pegawai (benar)
2. Tidak boleh ada operator aritmatika, contoh : data-pegawai
3. Tidak boleh ada angka didepan nama variable, contoh : 6siswa (salah), siswa6(benar)
4. Tidak boleh ada karakter khusus seperti #, $, &, %, ^, @, dan sebagainya.

1. Tipe Data

Tipe data dibagi dua yaitu :

1. Tipe data dasar

| Tipe Data | Ukuran memori | Rentang Nilai |
| --- | --- | --- |
| Char | 1 byte | -128 hingga +127 |
| Int | 2 byte | -32768 hingga +32767 |
| Long | 4 byte | -2.147.433.648 hingga +2.147.438.647 |
| Float | 4 byte | 3.4X10-38 hingga 3.4X1038 |
| Double | 8 byte | 1.7X10-808 hingga 1.7X10808 |
| Long Double | 10 byte | 3.4X10-4932 hingga 3.4X104932 |

Keterangan :

**Char :** merupakan tipe data dasar yang dipakai untuk mendeklarasikan peubah bilangan bulat, karakter, atau string.

**Int dan Long :** merupakan tipe data dasar yang dipakai untuk mendeklarasikan peubah bilangan bulat.

**Float, double, dan Long Double :** Merupakan tipe data dasar yang dipakai untuk mendeklarasikan peubah bilangan pecahan.

1. Tipe Bentukan

Merupakan bentukan dari tipe data dasar dengan nama lain, missal didefinisikan sebuah peubah X bertipe Int.

Dalam C++ untuk membentuk sebuah tipe bentukan dibutuhkan kata kunci : Typedef

Contoh: *typedef int x ;*

1. Konstanta

Merupakan sebuah penamaan yang nilainya tidak dapat diubah selama pelaksanaan program

Sebuah konstanta dideklarasikan sebagai berikut :

Const tipe\_data nama\_variable = nilai ;

Contoh :

*Const int nilai = 70 ;*

1. Keyword (kata kunci)

Merupakan pengenal khusus yang digunakan bagi proses kompilasi, dan kata kunci ini tidak dapat digunakan sebagai variable atau identifier.

| Asm | Else | operator | template |
| --- | --- | --- | --- |
| Auto | Enum | private | this |
| Break | Extern | public | typedef |
| Case | Float | protected | union |
| Char | For | register | unsigned |
| Class | Friend | return | virtual |
| Const | Goto | short | void |
| Continue | If | signed | volatile |
| Default | Lline | sizeof | while |
| Delete | Int | struct |  |
| Do | Long | switch |  |
| Double | New | static |  |

1. Perintah Keluaran (Output)

Perintah standar output pada C++, diantaranya :cout()

contoh : ***cout<<”..........pernyataan.........”<<endl;***

untuk menggunakan fungsi perintah keluaran harus disertakan file header = ***iostream.h***

1. Perintah Masukan (Input)

cin()

contoh : ***cout<<” Masukan Nilai : “;cin>>a;***

untuk menggunakan fungsi perintah masukan harus disertakan file header = ***iostream.h***

**Program Latihan 3 :lat3**

*//program jumlah*

#include <iostream>

using namespace std;

int main()

{

intx, y, jumlah;

x=100;

y=50;

jumlah=x+y;

cout<<x<<” + ”<<y<<” = “<<jumlah<<endl;

return 0;

}

**Program Latihan 4 : lat4**

*//jumlah barang*

#include <iostream>

using namespace std;

int main()

{

int a,b,c,d,e;

a=5000;

b=10;

c=3;

d=b-c;

e=a\*c;

cout<<” DATA BARANG “<<endl;

cout<<”---------------------------“<<endl;

cout<<”Harga Barang adalah :”<<a<<endl;

cout<<”Jumlah Barang adalah : “<<b<<endl;

cout<<”Barang terjual adalah : “<<c<<endl;

cout<<”----------------------------“<<endl;

cout<<”Sisa Barang adalah : “<<d<<endl;

cout<<”Jumlah bayar adalah : “<<e<<endl;

return 0;

}

**program Latihan 5 : lat5**

//Program Operator Majemuk

#include <iostream>

Using namespace std;

int main()

{

int x;

x=2;

cout<<“Nilai awal x = “<<x<<endl;

x+=8;

cout<<“Nilai x menjadi = “<<x<<endl;

x/=2;

cout<<“Nilai x menjadi = “ <<x<<endl;

return 0;

}

**program Latihan 6 : lat6**

*//program input data*

#include <iostream>

using namespace std;

int main()

{

float a,b,c,d,e;

cout<<"Masukan Nilai A : ";cin>>a;

cout<<"Masukan Nilai B : ";cin>>b;

c=a + b;

cout<<" Nilai C adalah : "<<c<<endl;

d=c/a;

cout<<”Nilai D adalah : “<<d<<endl;

e=d\*c;

cout<<”Nilai E adalah : “<<e<<endl;

return 0;

}

**program Latihan 7 : lat7**

//Program hasil bagi

#include <iostream>

using namespace std;

int main()

{

int a,b,c,d,e;

a=5;

b=10;

c=15;

d=a+b\*c-b;

e=c%b;

cout<<“Hasil D adalah : “<<d<<endl;

cout<<“sisa bagi c%b : “<<e<<endl;

return 0;

}

Materi 3

**Operator Relasi dan Logik**

Operator yang digunakan untuk membandingkan dua buah nilai

| **Operator** | **Keterangan** |
| --- | --- |
| == | Sama dengan (bukan penugasan |
| != | Tidak sama dengan |
| > | Lebih dari |
| < | Kurang dari |
| >= | Lebih dari atau sama dengan |
| <= | Kurang dari atau sama dengan |
| && | Dan (AND) kedua-duanya harus true hasil true |
| || | Atau (OR) salah satu true hasil true |
| ! | Tidak (NOT) keblikan, jika input true hasil false |

**Latihan8**

//Melihat relasi dan logik

#include <iostream>

using namespace std;

int main()

{

int a,b,c,d,e,f,g,h,i;

cout<<“ Masukan Nilai a = “;cin>>a;

cout<<“ Masukan Nilai b = “;cin>>b;

cout<<“ Masukan Nilai c = “;cin>>c;

d=a == b;

e=a != b;

f=c >= a;

g=c > b;

h=a < c;

i=c >= b;

cout<<“Hasil dari : “<<a<<“ == “<<b<<“ = “<<d<<endl;

cout<<“Hasil dari : “<<a<<“ != “<<b<<“ = “<<e<<endl;

cout<<“Hasil dari : “<<c<<“ >= “<<a<<“ = “<<f<<endl;

cout<<“Hasil dari : “<<c<<“ > “<<b<<“ = “<<g<<endl;

cout<<“Hasil dari : “<<a<<“ < “<<c<<“ = “<<h<<endl;

cout<<“Hasil dari : “<<c<<“ >= “<<b<<“ = “<<i<<endl;

return 0;

}

**Latihan9**

//Penggunaan Operasi Logika OR

#include <iostream>

using namespace std;

int main()

{

int a,b,c,d,e,f,g;

cout<<“ Masukan Nilai A = “;cin>>a;

cout<<“ Masukan Nilai B = “;cin>>b;

cout<<“ Masukan Nilai C = “;cin>>c;

d=a + 4 < 10;

e=b > a + 5;

f=c – 3 > 4;

g=d || e || f;

cout<<endl<<endl;

cout<<“Program Ekspresi OR”<<endl<<endl;

cout<<“Hasil dari d=a+4<10 adalah : “<<d<<endl;

cout<<“Hasil dari e=b>a+5 adalah : “<<e<<endl;

cout<<“Hasil dari f=c-3>4 adalah : “<<f<<endl;

cout<<“Hasil dari g=d||e||f adalah : “<<g<<endl;

cout<<endl;

return 0;

}

***Materi 4***

**OPERASI KONDISI**

**1.STATEMENT (Pernyataan)**

Statement Kendali IF

adalah merupakan statement yang berfungsi mengendalikan proses program , statement kendali dapat berupa Penyeleksian kondisi (pemilihan), Pengulangan (Looping), atau lompatan (jumping).

Statement Selection (Pemilihan)

contoh :

If

Kondisi

else

Kondisi

**Program Latihan 10 : lat10**

//Mencari bilangan terbesar dari 3 bilangan bulat

#include <iostream>

using namespace std;

int main ()

{

int x,y,z;

cout<<”Nilai X ?”;cin>>x;

cout<<”Nilai Y ?”;cin>>y;

cout<<”Nilai Z ?”;cin>>z;

if ((x>y) && (x>z))

cout<<”Bilangan terbesar adalah “<<x<<endl;

else

if ((y>x) && (y>z))

cout<<”Bilangan terbesar adalah”<<y<<endl;

else

cout<<”Bilangan terbesar adalah “<<z<<endl;

return 0;

}

**Program Latihan11 : lat11**

//program IF , ELSE

#include <iostream>

using namespace std;

int main()

{

double tot\_beli, potongan=0, jum\_bayar=0;

cout<<”Total Pembelian Rp. “;cin>>tot\_beli;

if (tot\_beli >= 50000)

potongan = 0.2 \* tot\_beli;

else

potongan = 0.05 \* tot\_beli;

cout<<”besarnya potongan Rp. “<<potongan<<endl;

jum\_bayar = tot\_beli – potongan;

cout<<”jumlah yang harus dibayarkan Rp. “<<jum\_bayar;

return 0;

}

**ProgramLatihan12 : lat 12**

//mencari indeks dari nilai ujian mahasiswa

#include <iostream.h>

using namespace std;

int main()

{

float nilai;

char indeks;

cout<<”Nilai Ujian ? “;cin>>nilai;

if ((nilai>= 80)&&(nilai <=100))

indeks = ‘A’;

else

if ((nilai>69)&&(nilai < 80))

indeks = ‘B’;

else

if ((nilai > 59)&&(nilai < 70))

indeks = ‘C’;

else

if ((nilai > 49)&&(nilai<60))

indeks = ‘D’;

else

indeks = ‘E’;

cout<<”indeks = “<<indeks<<endl;

return 0;

}

***@ IF tersarang.***

If bersarang (nested if) merupakan bentuk dari suatu statement if yang berada dalam lingkungan statement if lainnya.

**Program Latihan13 : lat13**

// Mencari nilai terbesar dari terbesar dari tiga buah bilangan bulat menggunakan if tersarang

#include <iostream>

using namespace std;

int main()

{

int A,B,C;

cout<<”Masukan nilai A : “;cin>>A;

cout<<”Masukan nilai B : “;cin>>B;

cout<<”Masukan nilai C : “;cin>>C;

cout<<”-------------------------------------“<<endl;

if (A>B)

if(A>C)

cout<<”Bilangan terbesar adalah : “<<A<<endl;

else //(C>=A)

cout<<”Bialngan terbesar adalah : “<<C<<endl;

else //(B>=A)

if (B>C)

cout<<”Bilangan terbesar adalah : “<<B<<endl;

else

cout<<”Bilangan terbesar adalah : “<<C<<endl;

return 0;

}

**Program Latihan14 : lat14**

#include <iostream>

using namespace std;

int main()

{

float pendptan, jasa=0, komisi=0, total=0;

cout<<”Pendapatan Hari ini Rp. “;cin>>pendptan;

if (pendptan >= 0 && pendptan <= 200000)

{

jasa=10000;

komisi=0.1\*pendptan;

}

else

{

if (pendptan >200000 && pendptan<= 500000)

{

jasa=20000;

komisi=0.15\*pendptan;

}

else

{

jasa=30000;

komisi=0.2\*pendptan;

}

}

total = komisi + jasa;

cout<<”Uang Jasa Rp. “<<jasa<<endl;

cout<<”Uang Komisi Rp. “<<komisi<<endl;

cout<<”=========================”<<endl;

cout<<”Hasil Total Rp. “<<total<<endl;

return 0;

}

***Materi 5***

***@ Statement pemilihan switch***

Pernyataan switch adalah pernyataan yang digunakan untuk mejalankan salah satu pernyataan dari beberapa kemungkinan pernyataan berdasarkan nilai dari sebuah ungkapan dan nilai penyeleksi. secara singkat pernyataan ini merupakan bentuk pernyataan pemilihan berganda.

notasi algoritma pernyataan switch adalah :

switch

{

case ungkapan 1;

pernyataan \_1;

break;

case ungkapan 2;

pernyataan\_2;

break;

…………

default /\*optional\*/

pernyataan\_N; /\*optional\*/

}

Pengulangan (Repetition/Looping)

Statement kendali pengulangan merupakan sebuah statement dimana sejumlah instruksi dapat dilakukan secara berulang kali sesuai dengan keinginan. Didalam algoritma pengulangan atau yang sering disebut repetition (Loop) dapat dilakukan sejumlah kali atau sampai kondisi pengulangan tercapai.

Notasi Algoritma untuk struktur pengulangan :

<inisialisasi>

Awal pengulangan

badan (body) pengulangan

Akhir pengulangan

<terminasi>

didalam C++ ada tiga jenis struktur pengulangan yaitu : for, while, dan do while.

Pengulangan FOR

adalah struktur pengulangan tanpa kondisi artinya instruksi-instruksi didalam badan pengulangan diulang sejumlah kali yang dispesifikasikan. dalam hal ini jumlah pengulangan sudah diketahui sebelum eksekusi program.

FOR tersarang

for tersarang merupakan pernyataan for didalam pernyataan for lainnya,

perulangan for yang berada didalam akan diproses terlebih dahulu sampai waktu habis, kemudian perulangan for yang lebih luar baru akan bertambah, mengerjakan perulangan for yang lebih dalam lagi.

**Program Latihan 15 : lat15**

//program menu hitung segi empat

#include <iostream>

#include <math>

#include <iomanip>

using namespace std;

int main()

{

int pil;

float panjang, lebar, luas, kel,diag;

cout<<”=========================================”<<endl;

cout<<setw(5)<<” “<<”MENU UTAMA”<<endl;

cout<<”=========================================”<<endl;

cout<<”1. Hitung Luas “<<endl;

cout<<”2. Hitung Keliling “<<endl;

cout<<”3. Hitung Diagonal “<<endl;

cout<<”4. Keluar Program “<<endl;

cout<<”=========================================”<<endl;

cout<<” Pilih Nomor menu (1/2/3/4) : “;cin>>pil;

cout<<”=========================================”<<endl;

switch(pil)

{

case 1:

{

cout<<”Panjang ? “;cin>>panjang;

cout<<”Lebar ? “;cin>>lebar;

luas = panjang\*lebar;

cout<<”Luas segiempat = “<<luas<<endl;

break;

}

case 2:

{

cout<<”Panjang ? “;cin>>panjang;

cout<<”Lebar ? “;cin>>lebar;

kel = 2\*(panjang+lebar);

cout<<”Keliling segiempat = “<<kel<<endl;

break;

}

case 3:

{

cout<<”Panjang ? “;cin>>panjang;

cout<<”Lebar ? “;cin>>lebar;

diag = sqrt((panjang\*panjang) + (lebar\*lebar));

cout<<”Diagonal segiempat = “<<diag<<endl;

break;

}

case 4:

cout<<” Keluar Program……bye2 “<<endl;

break;

default:

cout<<”Anda salah memilih nomor menu…….!!!”<<endl;

break;

return 0;

}

}

**Program Latihan16 : lat16**

#include<iostream>

using namespace std;

int main()

{

char kode;

cout<<” Masukan Kode Barang [1..4] : “;cin>>kode;

switch(kode)

{

case ‘1’ :

cout<<”Alat Olah Raga”;

break;

case ‘2’ :

cout<<”Alat Elektronik”;

break;

case ‘3’ :

cout<<”Alat Masak”;

break;

default:

cout<<”Anda Salah Masukan Kode.......”;

break;

} return 0;

}

**Program Latihan17 : lat17**

//hitung rata-rata nilai

#include <iostream>

using namespace std;

int main()

{

int k;

int n;

float nilai, jumlah, rata2;

jumlah=0;

cout<<”Jumlah data : “;cin>>n;

for(k=1 ; k<=n; k++)

{

cout<<”Nilai “<<k<<” : “;cin>>nilai;

jumlah= jumlah + nilai;

}

rata2= jumlah/n;

cout<<”Rata-rata nilai = “<<rata2<<endl;

return 0;

}

**Program Latihan18 : lat18**

//melihat efek for tersarang

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

int a,b;

for (a=1; a<5; a++)

{

for (b=1; b<3; b++)

cout<<setw(5)<<a<setw(3)<<b;

cout<<endl;

} return 0;

}

***Materi 6***

***Pengulangan While***

Pengulangan while termasuk ke dalam pengulangan dengan kondisi *(conditional looping)*,

merupakan salah satu bentuk pengulangan yang berguna untuk memproses suatu pernyataan atau beberapa pernyataan. Notasi Algoritmanya sebagai berikut :



Aksi atau runtunan akan dilaksanakan selama kondisi bernilai ***true,*** jika kondisi bernilai ***false*** maka pengulangan tidak akan dilaksanakan, yang berarti pengulangan selesai.

***Pengulangan Do-While***

Pernyataan do-while juga berguna untuk mengulang proses, bentuk dari pernyataan do-while ini adalah :



***Pernyataan Continue***

Pernyataan continue dipakai untuk mengarahkan eksekusi ke putaran atau literasi berikutnya pada pernyataan oleh dua kondisi

**Program latihan19 : lat19**

//Menghitung Rata-rata nilai

#include <iostream>

using namespace std;

int main()

{

int k ;

float N, jumlah, x, rata2;

cout<<”Jumlah data : “;cin>>N;

jumlah=0;

k=1;

while(k < N)

{

cout<<”Nilai “<<k<<” : “;cin>>x;

k++;

jumlah= jumlah + x;

}

rata2= jumlah/N;

cout<<”Rata-rata Nilai = “<<rata2<<endl;

return 0;

}

**Program latihan20 : lat20**

//Program tekan tombol ‘Y’ untuk mengulang dan ‘t’ untuk keluar

#include <iostream.h>

using namespace std;

int main()

{

float panjang, lebar, luas;

char pil;

pil = ‘y’;

while(pil == ‘y’)

{

cout<<”LUAS SEGI EMPAT “<<endl;

cout<<”===============”<<endl;

cout<<”Panjang ? “;cin>>panjang;

cout<<”Lebar ? “;cin>>lebar;

luas=panjang \* lebar;

cout<<”Luas segi empat = “<<luas<<endl;

cout<<”Hitung lagi ? (y/t) ? “;cin>>pil;

} return 0;

}

**Program latihan21 : lat21**

//konversi Celcius ke Fahreinhet

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

float celcius, fahreinhat;

cout<<”================================”<<endl;

cout<<”Celcius”<<setw(12)<<”Fahreinhat”<<endl;

cout<<”================================”<<endl;

celcius=0;

do{

fahreinhat=1.8\*celcius + 32;

cout<<celcius<<setw(12)<<fahreinhat<<endl;

celcius +=10;

}

while(celcius < 100);

cout<<”==========================”<<endl;

return 0;

}

**Program latihan 22 : lat22**

//program hitung jumlah nilai jika ganjil

#include <iostream.h>

using namespace std;

int main()

{

int k, N, x, jumlah;

cout<<”Jumlah Data : “;cin>>N;

jumlah=0;

for(k=0; k<N; k++)

{

cout<<”Nilai “<<(k + 1)<<” : “;cin>>x;

if(x % 2 == 0)

continue;

jumlah=jumlah + x;

}

cout<<”Jumlah = “<<jumlah<<endl;

return 0;

}

***Materi 7***

***Pernyataan exit()***

Suatu eksekusi program dapat dihentikan melalui pemanggilan fungsi exit(). hal ini biasa dilakukan jika di dalam suatu eksekusi terdapat suatu kondisi yang tidak dikehendaki

file library yang digunakan objek exit() adalah ***stdlib.h***

dalam proses pernyataan exit selalu diiringi objek ***toupper*** untuk melaksanakan eksekusi lanjutan setelah eksekusi pemilihan.

**PROSEDUR**

Prosedur merupakan suatu teknik pemrograman modular, yang berarti sebuah program dipecah menjadi sub-sub program (modul)yang lebih kecil tujuannya adalah :

* Memudahkan dalam mengembangkan program
* Menghemat ukuran program, bila ada sederetan instruksi yang sama digunakan

***a, pemanggilan prosedur***

prosedur diakses dengan cara memanggil nama (prototype)-nya dari program pemanggil, diikuti dengan daftar parameter aktual (bila ada), karena prosedur menghasilkan nilai, maka nilai tersebut dapat ditampung dalam sebuah peubah yang bertipe sama dengan tipe prosedur





***b, Nama Lokal, Global***

Nama-nama variabel (konstanta, peubah, tipe dll) yang dideklarasikan didalam deklarasi prosedur dikatakan bersifat “lokal”, nama-nama variabel lokal tersebut hanya dapat digunakan didalam prosedur yang melingkupinya saja.

sedangkan nama-nama variabel yang dideklarasikan diluar prosedur maupun program utama dikatakan bersifat “Global”, yang dapat digunakan di bagian manapun di dalam program baik diprogram utama maupun prosedur.

***Latihan 23, //melihat efek exit***

#include <iostream>

#include <ctype>

#include <stdlib>

using namespace std;

int main()

{

char jawab;

long int a, b, c;

hitung:

cout<<” Nilai A = “;cin>>a;

cout<<” Nilai B = “;cin>>b;

c= a\*b;

cout<<” Nilai A\*B = “<<c<<endl;

cout<<” Coba Lagi [Y]=ya [T]=tidak ? “<<endl;

for( ; ; )

{

jawab= toupper(getch());

if((jawab==’Y’) || (jawab==’T’))

{

cout<<jawab<<endl;

if(jawab==’Y’)

goto hitung;

else

exit(0);

} return 0;

}

}

***Latihan 24***

//melihat efek continue

#include <iostream>

#include <iomanip>

#include<math>

using namespace std;

int main()

{

int pil, panjang,lebar,luas,kel,diag;

pil=1;

while(pil != 5)

{

cout<<setw(10)<<”======================================”<<endl;

cout<<setw(10)<<” ”<<” MENU UTAMA “<<endl;

cout<<setw(10)<<”======================================”<<endl;

cout<<setw(10)<<”1. Entry Data “<<endl;

cout<<setw(10)<<”2. Hitung Luas “<<endl;

cout<<setw(10)<<”3. Hitung Keliling “<<endl;

cout<<setw(10)<<”4. Hitung Diagonal “<<endl;

cout<<setw(10)<<”5. Keluar Program “<<endl;

cout<<setw(10)<<”=========================================”<<endl;

cout<<setw(10)<<” Pilih Nomor menu (1/2/3/4/5) : “;cin>>pil;

if(pil == 5)

break;

switch(pil)

{

case 1:

{

cout<<”Anda memilih input data “<<endl;

cout<<”Tekan Enter untuk kembali ke Menu”<<endl;

getch();

continue;

}

case 2:

{

cout<<”Panjang ? “;cin>>panjang;

cout<<”Lebar ? “;cin>>lebar;

luas = panjang\*lebar;

cout<<”Luas segiempat = “<<luas<<endl;

cout<<”Tekan Enter untuk kembali ke Menu”<<endl;

getch();

continue;

}

case 3:

{

cout<<”Panjang ? “;cin>>panjang;

cout<<”Lebar ? “;cin>>lebar;

kel = 2\*(panjang+lebar);

cout<<”Keliling segiempat = “<<kel<<endl;

cout<<”Tekan Enter untuk kembali ke Menu”<<endl;

getch();

continue;

}

case 4:

{

cout<<”Panjang ? “;cin>>panjang;

cout<<”Lebar ? “;cin>>lebar;

diag = sqrt((panjang\*panjang) + (lebar\*lebar));

cout<<”Diagonal segiempat = “<<diag<<endl;

cout<<”Tekan Enter untuk kembali ke Menu”<<endl;

getch();

continue;

}

case 5:

cout<<” Keluar Program……bye2 “<<endl;

goto selesai;

}

}

selesai:

cout<<” Anda sudah keluar program........terima kasih”<<endl;

}

***Latihan 25***

//melihat efek prosedur

#include <iostream.h>

#include <conio.h>

void Ratarata();

void main()

{

Ratarata();

}

void Ratarata()

{

int k, N;

float x, jumlah, rata2;

cout<<” Jumlah data : “;cin>>N;

jumlah=0;

for (k=1; k<=N; k++)

{

cout<<” X “<<k<<” = “;cin>>x;

jumlah= jumlah + x;

}

rata2= jumlah / N;

cout<<” Nilai Rata-rata = “<<rata2<<endl;

}

***Latihan 26***

//cari nilai maksimum dan minimum dengan nama global

#include <iostream.h>

#include <conio.h>

const Nmaks=100;

int k, maks, min, N;

int nilai[Nmaks];

void Jumlahdata();

void CariMaks();

void CariMin();

void main()

{

Jumlahdata();

for(k=1; k<=N; k++)

cout<<nilai[k]<<endl;

CariMaks();

CariMin();

}

void Jumlahdata()

{

cout<<”Jumlah data : “;cin>>N;

for(k=1; k<=N; k++)

{

cout<<” Nilai “<<k<<” = “;cin>>nilai[k];

}

}

void CariMaks()

{

maks= nilai[1];

for(k=2; k<=N; k++)

{

if(nilai[k] > maks)

maks= nilai[k];

}

cout<<”Nilai Maksimum “<<maks<<endl;

}

void CariMin()

{

min= nilai[1];

for(k=2; k<=N; k++)

{

if(nilai[k] < min)

min= nilai[k];

}

cout<<” Nilai minimum “<<min<<endl;

}

***Materi 8***

***LARIK (ARRAY)***

Secara umum ***Larik (Array)*** didefinisikan sebagai struktur data yang menyimpan elemen yang bertipe sama. Namun pada pengembangannya larik banyak digunakan untuk mengolah data yang mempunyai nama dan tipe yang berbeda contoh hasil ujian mahasiswa dengan field-field missal NIM, Nama, Nilai, Indeks dsb.

sehingga mempunyai tipe yang kompleks.

Tipe larik yang tersusun atas field-field dng nama dan tipe yang berbeda disebut sebagai ***“ Larik Terstruktur “***

Gambaran logika sebuah larik dapat dijelaskan sebagai berikut :

A

1 

2 

3

4

5

Tiap kotak pada larik (array) diberi indeks integer 1,2,3,4,5 tiap elemen larik ditulis dengan notasi A[1], A[2], A[3], A[4], A[5].

***Mendefinisikan Larik di dalam deklarasi***

***larik*** adalah struktur data yang ***static***, artinya jumlah elemen larik harus sudah diketahui sebelum program dieksekusi.

Tipe elemen larik dapat berupa tipe dasar ***(integer, real, Boolean, string***), tipe terstruktur (tipe bentukan sprt ***record***), atau bahkan bertipe larik lain.

cara mendefinisikan larik dalam C++ adalah :

contoh : int ***L***[5], char ***Nama***[5] [25] dsb

ket :

**L**  adalah nama peubah yang mempunyai 5 buah elemen yang bertipe integer, indeks larik bertipe integer

**Nama** adalah peubah larik yang mempunyai 5 buah elemen bertipe string dengan panjang string 25

***Latihan 27***

//program larik berisi nilai ujian mahasiswa

#include<iostream.h>

#include<conio.h>

#include<iomanip.h>

void main()

{

int nilai[6];

int k;

for (k=1; k<6; k++)

{

cout<<"Nilai "<<k<<" = ";cin>>nilai[k];

}

cout<<"==================================="<<endl;

cout<<"Data NilaiUjianAkhir Semester"<<endl;

cout<<"==================================="<<endl;

for(k=1 ; k<6; k++)

{

cout<<setiosflags(ios::right)<<setw(6)<<nilai[k]<<endl;

cout<<"==========="<<endl;

}

}

***Latihan 28***

//program larik data namamahasiswa

#include<iostream.h>

#include<conio.h>

void main()

{

char Nama[15][25];

int k;

for(k=1; k<15; k++)

{

cout<<"Nama " <<k<<" = ";cin>>Nama[k];

}

cout<<"======================="<<endl;

cout<<"Data NamaMahasiswa"<<endl;

cout<<"======================="<<endl;

for(k=1; k<15; k++)

cout<<Nama[k]<<endl;

cout<<"======================="<<endl;

}

***Latihan 29***

//program hitung rata-rata dari kumpulan nilai

#include<iostream.h>

#include<conio.h>

void main()

{

float Nilai[10];

int k;

float jumlah, rata2;

jumlah=0;

cout<<"Data NilaiUjian "<<endl;

for (k=1;k<10; k++)

{

cout<<"Nilai " <<k<<" = ";cin>>Nilai[k];

jumlah=jumlah + Nilai[k];

}

rata2=jumlah/10;

cout<<"Rata-rata = "<<rata2<<endl;

}

***Materi 9***

***Inisialisasi Larik (Array)***

Menginisialisasikan larik (Array) adalah memberikan nilai awal untuk seluruh elemen larik. Seperti halnya variable biasa, larik juga dapat diberi nilai awal (diinisialisasi) pada saat didefinisikan.

contoh :

int jum\_hari[12] =

{

31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31

};

merupakan contoh pendefinisisan larik bernama jum\_hari yang terdiri dari 12 elemen bertipe integer dan secara berurutan (dimulai dari indeks 0) diberi nilai 31, 28 dst.

***Mencari Nilai Maksimum Larik (Array)***

Pencarian nilai maksimum pada larik (Array) harus data bertipe integer atau real.

Bila elemen larik (Array) sudah terurut menaik, nilai maksimum langsung dapat diperoleh pada elemennya, tetapi bila elemen larik tersusun acak, kita harus ”melihat” seluruh elemen larik (array) satu persatu untuk mendapatkan nilai maksimum.

***Mencari Nilai Minimum Larik (Array)***

Teknik mencari nilai minimum pada larik dikembangkan dengan cara yang sama dengan metode mencari nilai maksimum larik (array)

***Mencari Nilai tertentu didalam Larik (array)***

Pencariaan data tertentu di dalam larik (array) merupakan proses melihat elemen larik satu persatu mulai dari elemen pertama dengan membandingkan apakah elemen tersebut sama dengan nilai yang dicari Jika sama maka pencariaan selesai dan jika belum sama maka pencarian dilanjutkan ke elemen berikutnya. Proses pencarian diteruskan sampai nilai yang dicari ditemukan atau seluruh elemen larik sudah diperiksa.

***Mengakses Array berdimensi satu***

Suatu array dapat diakses dengan menggunakan indeksnya. bentuk umum dalam pengaksesan adalah :

Nama array [index]

***Mengakses Array berdimensi dua dan tiga dimensi***

Array dimensi dua tersusun dalam bentuk baris dan kolom, dimana indeks pertama menunjukan baris dan indeks kedua menunjukan kolom, sedangkan array dimensi tiga ditambahkan jumlah isi dari baris

bentuk umum pendeklarasian array :

dimensi dua = tipe\_data Nama\_variabel [index 1] [index 2]

dimensi tiga = tipe\_data nama\_variabel [index 1] [index 2] [index 3]

***Latihan 30***

//program cari nilai maksimum

#include <iostream.h>

#include <conio.h>

void main()

{

intnilai[5];

intk,maks;

for(k=1;k<6;k++)

{

cout<<"Nilai "<<k<<" = ";cin>>nilai[k];

}

maks=nilai[1];

for(k=1;k<6;k++)

{

if (nilai[k] >maks)

maks=nilai[k];

}

cout<<"NilaiUjiantertinggi = "<<maks<<endl;

}

***Latihan 31***

//program Array dimensi satu

#include <iostream.h>

#include <conio.h>

#include <stdio.h>

#include <iomanip.h>

void main()

{

int i;

char nama[5] [20];

float nilai1[5];

float nilai2[5];

float hasil[5];

for(i=1; i<=4; i++)

{

cout<<”Data ke – “<<i<<endl;

cout<<”Nama Siswa : “;cin>>nama[i];

cout<<”Nilai MidTest : “;cin>>nilai1[i];

cout<<”Nilai Final : “;cin>>nilai2[i];

hasil[i] = (nilai1[i] \* 0.40) + (nilai2[i] \* 0.60);

cout<<endl;

}

cout<<”-----------------------------------------------------------“<<endl;

cout<<”No. Nama Siswa Nilai Nilai Hasil “<<endl;

cout<<” Midtest Final Ujian “<<endl;

cout<<”-----------------------------------------------------------“<<endl;

for(i=1; i<=4; i++)

{

cout<<setiosflags(ios::left)<<setw(4)<<i;

cout<<setiosflags(ios::left)<<setw(10)<<nama[i];

cout<<setprecision(4)<<” “<<nilai1[i];

cout<<setprecision(4)<<” “<<nilai2[i];

cout<<setprecision(4)<<” “<<hasil[i]<<endl;

}

cout<<”------------------------------------------------------------------------------------------“<<endl;

getch();

}

***Latihan 32***

//program Array dimensi dua

#include <iostream.h>

#include <conio.h>

#include <stdio.h>

#include <iomanip.h>

void main()

{

int i, j ;

int data\_jual[4][4];

for(i=1; i<=3; i++)

{

for(j=1; j<=3; j++)

{

cout<<”Data Ke – “<<i<<” tahun ke : “<<j<<endl;

cout<<”Jumlah Penjualan : “;cin>>data\_jual[i][j];

}

}

cout<<”Data Penjualan Pertahun “<<endl;

cout<<”--------------------------- “<<endl;

cout<<”No 2001 2002 2003 “<<endl;

cout<<”---------------------------“<<endl;

for(i=1; i<=3; i++)

{

cout<<setiosflags(ios::left)<<setw(5)<<i;

for(j=1; j<=3; j++)

{

cout<<setiosflags(ios::right)<<setw(4);

cout<<data\_jual[i][j];

cout<<” “;

}

cout<<endl;

}

cout<<”---------------------------------------------“<<endl;

getch();

}

***Latihan 33***

//program Array dimensi tiga

#include <conio.h>

#include <stdio.h>

#include <iostream.h>

#include <iomanip.h>

void main()

{

int i,j,k;

float data\_jual[2][3][2];

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

{

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

{

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

{

cout<<"Data Tahun Ke - "<<i+1<<endl;

cout<<"Data ke - "<<j+1<<" "<<k+1<<endl;

cout<<"Jumlah Penjualan : ";

cin>>data\_jual[i][j][k];

}

cout<<endl;

}

cout<<endl;

}

cout<<endl;

cout<<" Data Penjualan Pertahun "<<endl;

cout<<"|--------------------------------------|"<<endl;

cout<<"|Tahun Hasil Tahun Penjualan Ke -|"<<endl;

cout<<"| Ke Ke -------------------|"<<endl;

cout<<"| 1 2 |"<<endl;

cout<<"----------------------------------------"<<endl;

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

{

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

{

cout<<setiosflags(ios::left)<<setw(11)<<i+1;

cout<<setiosflags(ios::left)<<setw(9)<<j+1;

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

{

cout<<setiosflags(ios::right)<<setw(5)<<data\_jual[i][j][k]<<" ";

}

cout<<endl;

}

cout<<endl;

}

cout<<"-------------------------------------";

cout<<endl;

getch();

}

***Latihan 34***

//program cari data padalarik

#include<iostream.h>

#include<conio.h>

void main()

{

const Nmaks=100;

int nilai[Nmaks];

int k, N, x, ix;

cout<<"Jumlah Data : ";cin>>N;

for(k=1; k<=N; k++)

{

cout<<"Nilai "<<k<<" = ";cin>>nilai[k];

}

cout<<"Masukan data yang ingindicari : ";cin>>x;

k=1;

while((k<N) && (nilai[k] != x))

{

k++;

}

if(nilai[k] == x)

{

ix=k;

cout<<"Data ditemukanpadaindekske "<<ix<<endl;

}

else

cout<<"Maaf data yang andacaritidakada"<<endl;

}

***Materi 10***

***Mengurutkan data***

Pengurutan (sorting) adalah proses mengatur sekumpulan obyek menurut urutan atau susunan tertentu. Urutan obyek tersebut dapat menaik (Ascending) atau menurun (Descending).

Metode pengurutan data yang banyak digunakan adalah :

1. Algoritma Pengurutan Gelembung (Bubble Sort)

metode pengurutan data dengan cara larik terurut menaik atau menurun melalui proses pertukaran dari yang terkecil sampai dengan yang terbesar.

1. Algoritma Pengurutan Pilih (Selection Sort)

Pengurutan yang digunakan dengan dasar memilih elemen maksimum/minimumdari larik lalu menempatkan elemen tersebut itu pada awal atau akhir larik (elemen terujung)

***Latihan 35***

//program urut ascending

#include <iostream.h>

#include <conio.h>

void main()

{

int nilai[5];

int temp;

int i;

int k;

int N;

cout<<"JumlahNilai : ";cin>>N;

for(k=1; k<N; k++)

{

cout<<"Nilai "<<k<<" = ";

cin>>nilai[k];

}

cout<<"Nilai sebelum terurut menaik = ";

for(k=1; k<N; k++)

cout<<nilai[k]<<” “;

cout<<endl;

for(i=0; i<(N-1); i++)

{

for(k=N; k>(i+1); k--)

{

if(nilai[k]<nilai[k-1])

{

temp=nilai[k];

nilai[k]=nilai[k-1];

nilai[k-1]=temp;

}

}

}

cout<<"Nilai setelah terurut=";

for(k=1; k<N; k++)

cout<<nilai[k]<<” “;

}

**Latihan 36**

//Program urut data dengan metode pilih maksimum

#include <iostream.h>

#include <conio.h>

void main()

{

int nilai[5];

int imaks, temp, k, j;

for(k=1; k<=5; k++)

{

cout<<”Nilai “<<k<<” = “;cin>>nilai[k];

}

cout<<”Nilai Sebelum Terurut menaik = “;

for(k=1; k<=5; k++)

cout<<nilai[k]<<’ ‘;

for(k=5; k>=2; k--)

{

imaks=1;

for(j=2; j<=k; j++)

{

if(nilai[j] > nilai[imaks])

{

imaks=j;

}

temp= nilai[imaks];

nilai[imaks]= nilai[k];

nilai[k]= temp;

}

}

cout<<endl;

cout<<”Nilai sesudah terurut menaik = “;

for(k=1; k<=5; k++)

cout<<nilai[k]<<’ ‘;

}

***Latihan 37***

//program larikstruktur

#include <iostream.h>

#include <conio.h>

#include <iomanip.h>

void main()

{

constNmaks=100;

structdata\_pegawai

{

char nip[11];

charnama[26];

longgaji;

};

data\_pegawai data[Nmaks];

int k, N;

cout<<"JumlahKaryawan : ";cin>>N;

for(k=0; k<N; k++)

{

cout<<"Data Karyawan "<<(k+1)<<endl;

cout<<"Nama :";cin>>data[k].nama;

cout<<"NIP :";cin>>data[k].nip;

cout<<"Gaji :";cin>>data[k].gaji;

}

cout<<" DAFTAR GAJI "<<endl;

cout<<" KARYAWAN PT.ELANG PERSADA "<<endl;

cout<<endl;

cout<<"=================================================="<<endl;

cout<<setw(3)<<"No.";

cout<<setw(5)<<"NIP";

cout<<setw(13)<<"NAMA";

cout<<setw(20)<<"GAJI"<<endl;

for(k=0; k<N; k++)

{

cout<<setiosflags(ios::left)<<setw(5)<<k;

cout<<setiosflags(ios::left)<<setw(12)<<data[k].nip;

cout<<setiosflags(ios::left)<<setw(20)<<data[k].nama;

cout<<setw(4)<<"Rp.";

cout<<setiosflags(ios::right)<<setw(7)<<data[k].gaji<<endl;

}

cout<<"================================================="<<endl;

}

**Latihan 38**

// Larik struktur bukan bertipe larik

#include <iostream.h>

#include <conio.h>

#include <iomanip.h>

void main()

{

const Nmaks=100;

struct data\_tanggal

{

int tanggal;

int bulan;

int tahun;

};

struct data\_pegawai

{

char nip[11];

char nama[25];

long gaji;

tanggal\_lahir tgl;

};

data\_pegawai data[Nmaks];

int k, N;

cout<<” Jumlah Karyawan : “;cin>>N;

for(k=0; k<N; k++)

{

cout<<” Data Karyawan “<<(k+1)<<endl;

cout<<”Nama : “;cin>>data[k].nama;

cout<<”NIP : “;cin>>data[k].nip;

cout<<”Gaji : “;cin>>data[k].gaji;

cout<<endl;

cout<<” Data Kelahiran “<<endl;

cout<<”Tanggal : “;cin>>data[k]. tgl.tanggal;

cout<<”Bulan : “;cin>>data[k].tgl.bulan;

cout<<”Tahun : “;cin>>data[k].tgl.tahun;

}

cout<<” DAFTAR GAJI “<<endl;

cout<<” KARYAWAN PT. ELANG PERKASA “<<endl;

cout<<endl;

cout<<”======================================================================”<<endl;

cout<<” NO. “

<<setw(5)<<” NIP”

<<setw(13)<<” NAMA “

<<setw(25)<<” TGL LAHIR “

<<setw(9)<<” GAJI “<<endl;

cout<<”======================================================================”<<endl;

for(k=0; k<N; k++)

{

cout<<setiosflags(ios::left)<<setw(5)<<k

<<setiosflags(ios::left)<<setw(12)<<data[k].nip

<<setosflags(ios::left)<<setw(20)<<data[k].nama

<<data[k].tgl.tanggal<<” – “<<data[k].tgl.bulan<<” – “

<<setw(8)<<data[k].tgl.tahun

<<setw(4)<<” Rp. “

<<setiosflags(ios::right)<<setw(7)<<data[k].gaji<<endl;

}

cout<<”=======================================================================”<<endl;

}

**LARIK STRUKTUR BERTIPE LAIN**

adalah larik terstruktur yang mempunyai elemen larik dengan tipe larik yang berbeda seperti struktur tipe bentukan yang cukup kompleks

**LARIK STRUKTUR BERTIPE SAMA**

adalah pemberian nilai terhadap suatu struktur dilakukan dengan bentuk variable struktur yang bertipe sama

analisa logika sbb:

**var1 = var2**

contoh :

**tgl2 = tgl1 ;**

diperkenankan dalam hal ini seluruh anggota pada variabel tgl2 diisi dengan anggota terkait yang ada pada variable tgl1. pernyataan diatas merupakan penyederhanaan dari tiga pernyataan berikut :

**tgl2.bulan = tgl1.bulan;**

**tgl2.tahun = tgl1.tahun;**

**tgl2.tanggal = tgl1.tanggal;**

**OBYEK DAN KELAS**

Obyek adalah kesatuan unit yang dapat mengkombinasikan data-data dan fungsi untuk mengakses data

kelas adalah kumpulan dari beberapa obyek yang sama dan saling berinteraksi satu dengan yang lain.

**UNION**

Union merupakan struktur yang biasa dipakai untuk menyatakan suatu memori dengan namalebih dari Satu

contoh :

**variable = lingkaran**

dapat dideklarasikan sbb :

**union {**

**float kel;**

**double luas;**

**} lingkaran;**

berarti elemen keliling dan luas akan menempati lokasi memori yang sama secara bergantian, besarnya lokasi memori akan digunakan adalah sebesar 8 byte yang merupakan ukuran elemen terbesar

**KATA KUNCI typedef**

adalah obyek yang digunakan untuk mendefinisikan kembali suatu nama tipe data yang baru berdasarkan tipe data yang lama.

Latihan 39 :

//mencari luas dan keliling lingkaran

#include <iostream.h>

#include <conio.h>

#include <math.h>

void main()

{

union {

float kel;

double luas;

}lingkaran;

float jari2;

cout<<”Masukan Jari-jari : “;cin>>jari2;

lingkaran.kel=2\*3.14\*jari2;

cout<<”Keliling Lingkaran = “<<lingkaran.kel<<endl;

lingkaran.luas=3.14\*jari2\*jari2;

cout<<” Luas Lingkaran = “<<lingkaran.luas<<endl;

}

Latihan 40

//program hitung titik tengah

#include <iostream.h>

#include <conio.h>

#include <iomanip.h>

void main()

{

typedef struct {

float x, y;

}titik;

titik p1, p2, p3;

cout<<”Titik P1 : “<<endl;

cout<<” x1 : “;cin>>p1.x;

cout<<” y1 : “;cin>>p1.y;

cout<<” Titik p2 : “<<endl;

cout<<” x2 : “;cin>>p2.x;

cout<<” y2 : “;cin>>p2.y;

p3.x=(p1.x + p2.x) / 2;

p3.y=(p1.y + p2.y) / 2;

cout<<” Titik tengah adalah : “<<” p3 (“<<p3.x<<” , “<<p3.y<<” ) “<<endl;

}

Latihan 41

// Larik struktur bukan bertipe larik

#include <iostream.h>

#include <conio.h>

#include <iomanip.h>

void main()

{

const Nmaks=100;

struct data\_tanggal

{

int tanggal;

int bulan;

int tahun;

};

struct data\_pegawai

{

char nip[11];

char nama[25];

long gaji;

tanggal\_lahir tgl;

};

data\_pegawai data[Nmaks];

int k, N;

cout<<” Jumlah Karyawan : “;cin>>N;

for(k=0; k<N; k++)

{

cout<<” Data Karyawan “<<(k+1)<<endl;

cout<<”Nama : “;cin>>data[k].nama;

cout<<”NIP : “;cin>>data[k].nip;

cout<<”Gaji : “;cin>>data[k].gaji;

cout<<endl;

cout<<” Data Kelahiran “<<endl;

cout<<”Tanggal : “;cin>>data[k].tgl.tanggal;

cout<<”Bulan : “;cin>>data[k].tgl.bulan;

cout<<”Tahun : “;cin>>data[k].tgl.tahun;

cout<<endl;

}

cout<<” DAFTAR GAJI “<<endl;

cout<<” KARYAWAN PT. ELANG PERKASA “<<endl;

cout<<endl;

cout<<”======================================================================”<<endl;

cout<<” NO. “

<<setw(5)<<” NIP”

<<setw(13)<<” NAMA “

<<setw(25)<<” TGL LAHIR “

<<setw(9)<<” GAJI “<<endl;

cout<<”======================================================================”<<endl;

for(k=0; k<N; k++)

{

cout<<setiosflags(ios::left)<<setw(5)<<k

<<setiosflags(ios::left)<<setw(12)<<data[k].nip

<<setosflags(ios::left)<<setw(20)<<data[k].nama

<<data[k].tgl.tanggal<<” – “<<data[k].tgl.bulan<<” – “

<<setw(8)<<data[k].tgl.tahun

<<setw(4)<<” Rp. “

<<setiosflags(ios::right)<<setw(7)<<data[k].gaji<<endl;

}

cout<<”=======================================================================”<<endl;

}

Latihan 42

//Larik struktur dengan tipe berbeda

#include <iostream.h>

#include <conio.h>

#include <iomanip.h>

void main()

{

const Nmaks=100;

struct mata\_kuliah

{

char kodeMK[7];

char namaMK[15];

int nilai;

};

struct data\_mhs

{

int nim;

char nama[26];

mata\_kuliah MK[4];

};

data\_mhs data[Nmaks];

int k,i,N;

cout<<"Jumlah Mahasiswa : ";cin>>N;

for(k=0; k<N; k++)

{

cout<<"Data Mahasiswa "<<(k+1)<<endl;

cout<<"Nama : ";cin>>data[k].nama;

cout<<"NIM : ";cin>>data[k].nim;

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

{

cout<<"Kode MK "<<(i+1)<<" = ";cin>>data[k].MK[i].kodeMK;

cout<<"Nama MK "<<(i+1)<<" = ";cin>>data[k].MK[i].namaMK;

cout<<"Nilai "<<(i+1)<<" = ";cin>>data[k].MK[i].nilai;

clrscr();

}

}

cout<<endl;

cout<<setw(10)<<" "<<"DAFTAR NILAI MAHASISWA "<<endl;

cout<<setw(13)<<" "<<"SEMESTER GENAP "<<endl;

cout<<endl;

cout<<"========================================="<<endl;

cout<<"NIM";

cout<<setw(13)<<"NAMA";

cout<<setw(25)<<"Kode MK";

cout<<setw(9)<<"Nama MK";

cout<<setw(13)<<"Nilai"<<endl;

cout<<"=========================================="<<endl;

for(k=0; k<N; k++)

{

cout<<setiosflags(ios::left)<<setw(12)<<data[k].nim

<<setiosflags(ios::left)<<setw(22)<<data[k].nama;

for(i-0; i<4; i++)

{

if(i == 0){

cout<<setiosflags(ios::left)<<setw(9)<<data[k].MK[i].kodeMK

<<setiosflags(ios::left)<<setw(15)<<data[k].MK[i].namaMK

<<setiosflags(ios::right)<<setw(5)<<data[k].MK[i].nilai<<endl;

}

else

{

cout<<setw(34)<<" "

<<setiosflags(ios::left)<<setw(9)<<data[k].MK[i].kodeMK

<<setiosflags(ios::left)<<setw(15)<<data[k].MK[i].namaMK

<<setiosflags(ios::right)<<setw(5)<<data[k].MK[i].nilai<<endl;

}

}

}

cout<<"==========================================="<<endl;

}

Latihan 43:

//program struktur class

#include <iostream.h>

#include <conio.h>

#include <string.h>

class buku

{

public:

char judul[35];

char pengarang[25];

int jumlah;

};

void main()

{

buku novel;

strcpy(novel.judul,” Gun Of The Navarone “);

strcpy(novel.pengarang,” Alistair Maclean”);

novel.jumlah=12;

cout<<novel.judul<<endl;

cout<<novel.pengarang<<endl;

cout<<novel.jumlah<<endl;

}