

Buku Panduan  
Laboratorium Pemrograman C

I Wayan Sudiarta, Ph.D

10 Februari 2012



# Daftar Isi

<b>1</b>	<b>Lab 0: Cara Menggunakan Cygwin dan Compile Code C++ dengan g++</b>	<b>1</b>
1.1	Cygwin . . . . .	1
<b>2</b>	<b>Lab 1: Pengenalan Bahasa C</b>	<b>3</b>
2.1	Penulisan Kode C . . . . .	3
2.2	Modifikasi . . . . .	4
2.3	Lab 2: Pengenalan Variabel dan Operasi Matematis . . . . .	5
2.3.1	Deklarasi Variabel . . . . .	5
2.3.2	Kesalahan Penulisan Variabel . . . . .	5
2.3.3	Tipe Data Variabel . . . . .	6
2.3.4	Scanf . . . . .	6
<b>3</b>	<b>Lab 3: Pengenalan define dan math.h</b>	<b>9</b>
3.1	Define dan const . . . . .	9
3.2	math.h . . . . .	10
3.3	Latihan-latihan . . . . .	10
<b>4</b>	<b>Lab 4: Pengenalan for, while dan do while</b>	<b>11</b>
4.1	for . . . . .	11
4.2	while . . . . .	12
4.3	do-while . . . . .	13
<b>5</b>	<b>Lab 5: Pengenalan if, else dan switch</b>	<b>15</b>
5.1	if . . . . .	15
5.2	if else . . . . .	16
5.3	if else berantai . . . . .	16
5.4	()?:. . . . .	17
5.5	switch . . . . .	18

---

<b>6</b>	<b>Lab 6: Pengenalan Fungsi dan Array</b>	<b>19</b>
6.1	Fungsi . . . . .	19
6.2	Pengenalan Array . . . . .	20
<b>7</b>	<b>Lab 7: Pengenalan Pointer</b>	<b>21</b>
7.1	Pointer . . . . .	21
7.2	Pengenalan Operasi Pada Pointer . . . . .	22
7.3	Latihan . . . . .	22
<b>8</b>	<b>Lab 1: Pengenalan Bahasa C++</b>	<b>25</b>
8.1	Penulisan Kode C++ . . . . .	25
8.2	Modifikasi . . . . .	26
<b>9</b>	<b>Lab 2: Pengenalan Variabel dan Operasi Matematis</b>	<b>27</b>
9.1	Deklarasi Variabel . . . . .	27
9.2	Kesalahan Penulisan Variabel . . . . .	28
9.3	Tipe Data Variable . . . . .	28
<b>10</b>	<b>Lab 3: Pengenalan Define dan cmath</b>	<b>29</b>
10.1	Define dan const . . . . .	29
10.2	cmath . . . . .	30
10.3	Latihan-latihan . . . . .	30
<b>11</b>	<b>Lab 4: Pengenalan Fungsi dan Array</b>	<b>31</b>
11.1	Fungsi . . . . .	31
11.2	Pengenalan Array . . . . .	32
<b>12</b>	<b>Lab 5: Pengenalan Pointer</b>	<b>33</b>
12.1	Pointer . . . . .	33
12.2	Pengenalan Operasi Pada Pointer . . . . .	34
12.3	Latihan . . . . .	34
<b>13</b>	<b>Lab 6: Pengenalan Character dan String</b>	<b>37</b>
13.1	Character . . . . .	37
13.2	String . . . . .	38
13.3	getline() . . . . .	38
<b>14</b>	<b>Lab 7: Pengenalan Object Oriented Programming (OOP) dengan Class</b>	<b>41</b>
14.1	Struct . . . . .	41
14.2	Class . . . . .	42
14.3	Inheritance . . . . .	43

<b>15 Lab 8: Menggunakan Make</b>	<b>47</b>
<b>16 Lab 9: Pengenalan Library GSL</b>	<b>53</b>
16.1 Contoh GSL . . . . .	54
<b>17 Lab 10: Pengenalan Memori</b>	<b>55</b>
17.1 New atau New [] . . . . .	55



# Bab 1

## Lab 0: Cara Menggunakan Cygwin dan Compile Code C++ dengan g++

### 1.1 Cygwin

Setelah cygwin diinstall, sekarang kita bisa menjalankan cygwin terminal/shell. Shell ini sama fungsinya dengan shell yang ada di Unix atau Linux. Cygwin shell bisa di jalankan dengan mengklik shortcut yang ada di start menu atau di Desktop, seperti pada gambar berikut ini.

Setelah itu akan ditampilkan Cygwin shell seperti contoh di bawah ini. Shell ini hampir sama dengan shell yang ada di linux. Hampir semua perintah-perintah yang ada di linux bisa dilakukan di cygwin. Jadi kita bisa belajar perintah-perintah yang ada di Linux dengan menggunakan Cygwin. Seperti contoh perintah-perintah yang ada di dalam gambar di bawah ini.

Dengan cygwin shell ini, kita tidak bisa menjalankan program-program yang memerlukan X-Windows. Untuk mendapatkan shell dan fasilitas X-Windows, kita bisa menjalankan X-Windows dengan mengetik startxwin.sh di cygwin shell, seperti pada gambar dibawah ini. Kemudian muncul shell dan X-Windows telah dijalankan.

Setelah itu kita bisa menjalankan program-program yang memerlukan X-Windows. Contohnya sebuah editor yang bagus bernama nedit. Kita bisa jalankan nedit dengan ketik nedit di shell seperti gambar diatas. Kemudian muncul program nedit. Kita bisa membuat program C++ dengan menggunakan nedit.

Setelah source code C++ (sebagai contoh disini hello.cpp) diketik dengan menggunakan nedit atau texmaker atau wordpad atau notepad, kita bisa

## 2 Lab 0: Cara Menggunakan Cygwin dan Compile Code C++ dengan g++

kompilasi dengan g++ di cygwin shell seperti pada gambar. Tentunya kita harus ke direktori dimana source code itu berada.

Keterangan untuk gambar diatas: 1. Kita kompilasi source code C++ dari program hello (hello.cpp) dengan menggunakan **g++** dan outputnya (ini dilakukan dengan -o) adalah hello.exe. 2. Kemudian hello.exe dijalankan. Hasil program hello.exe adalah Hello World!. 3. Kita bisa mengarahkan output dari hello.exe ke sebuah file bernama results.txt dengan menggunakan >. 4. Kita bisa membaca isi dari file results.txt dengan menggunakan perintah cat.



# Bab 2

## Lab 1: Pengenalan Bahasa C

Pada Lab 1 ini kita akan mempelajari tentang bahasa C yang sederhana terutama tentang bagaimana mengkompilasi dan menjalankan program yang kita buat. Untuk lebih mengerti proses yang akan dilakukan dalam bahasa C, mari kita lakukan hal-hal dibawah ini.

### 2.1 Penulisan Kode C

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c berikut ini.

```
1  /* tulis komentar tentang kode anda disini */
2
3
4  #include <stdio.h>
5
6  int main()
7  {
8      printf(" Hello C++ \n");
9      printf(" Saya bisa membuat program dengan C");
10
11     return 0;
12 }
```

Setelah selesai mengetik kode di atas, kemudian simpan kode tersebut ke file dengan nama dan ekstensi **.c**. Sebagai contoh kita beri nama **helo.c**.

Kemudian jalankan program cygwin dan ketik **cd nama/folder/lokasi/file/cpp** untuk pindah ke folder tempat file yang telah kita telah buat berada.

Kemudian kita kompilasi kode tersebut dengan salah satu perintah berikut ini:

```
$ gcc namafile.c  
$ gcc namafile.c -o namafile.exe
```

Ingat tanda **\$** tidak diketik melainkan hanya menunjukkan baris perintah yang ada pada cygwin atau **command prompt**.

## 2.2 Modifikasi

Langkah selanjutnya adalah mengubah program sederhana di atas, dengan menambahkan beberapa baris perintah. Cobalah tambahkan baris ini satu per satu dan lihat apa yang dihasilkan dilayar setelah program kompilasi dan dieksekusi.

```
printf(" ini tab a\t b \t c");  
printf(" ini baris baru  a\n b \n c \n");  
printf(" ini garis miring \\ \" ");
```

Cobalah modifikasi apa saja yang anda ingin tampilkan pada layar.

## 2.3 Lab 2: Pengenalan Variabel dan Operasi Matematis

### 2.3.1 Deklarasi Variabel

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <stdio.h>
5
6  int main()
7  {
8      int a, b = 3, c = 2;
9
10     printf("  nilai a = %d \n", a);
11     printf("  nilai b = %d \n", b);
12     printf("  nilai c = %d \n", c);
13
14     a = b + c;
15
16     printf("  nilai a = %d \n", a);
17
18     return 0;
19 }
```

Berapakah nilai  $a$ ,  $b$  dan  $c$ ?

### 2.3.2 Kesalahan Penulisan Variabel

Tambahkan kode berikut ini pada kode di atas dan catat apa kesalahan yang muncul pada waktu kita kompilasi kode.

```
1
2  int 2apel;
3  long banyak!;
4  double desi-mal;
5  double desi_mal;
6  double DESI_mal;
```

### 2.3.3 Tipe Data Variabel

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <stdio.h>
5
6  int main()
7  {
8      int a = 2, b = 3, c, d;
9      float p = 2, q = 3, r;
10
11     c = a/b;
12     r = p/q;
13     a = i + j;
14     b = i%4;
15
16     printf("Baris 1: nilai c = %d \n", c);
17     printf("Baris 2: nilai r = %d \n", r);
18     printf("Baris 3: nilai r = %f \n", r);
19     printf("Baris 4: nilai b = %d", b);
20
21     return 0;
22 }
```

Berapakah nilai  $c$  (Baris 1) dan  $r$  (Baris 2 dan 3)? Apa yang ditampilkan pada layar ketika kita kompilasi? dan ketika program dieksekusi? Mengapa nilai  $c$  tidak sama dengan  $r$ ? Mengapa keluaran Baris 2 tidak sama dengan Baris 3?

### 2.3.4 Scanf

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini dan perhatikan baris **scanf**.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <stdio.h>
5
6  int main()
7  {
8      float x, y;
9
```

```
10 printf("Masukkan nilai x:\n");
11 scanf("%f",&x);
12
13 printf("Nilai yang Anda masukkan adalah %f \n", x);
14
15 printf("Masukkan dua nilai x dan y:\n");
16 scanf("%f %f",&x, &y);
17
18 printf("Nilai yang Anda masukkan adalah %f dan %f\n", x, y);
19 printf("Perkalian antara x dan y menghasilkan %f \n", x*y);
20
21 return 0;
22 }
```

Perhatikan baris yang mengandung fungsi `scanf`. Mengapa kita menggunakan simbol `&`? Apa yang terjadi jika tanda format `%f` diganti dengan `%d` pada fungsi `scanf`? Mengapa hasil yang dilihat pada layar berbeda dengan apa yang dimasukkan?



# Bab 3

## Lab 3: Pengenalan define dan math.h

### 3.1 Define dan const

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <stdio.h>
5
6  #define pi1 3.14159
7
8  int main()
9  {
10     const float pi2 = 3.14159;
11
12     printf(" nilai pi1 = %f \n", pi1);
13     printf(" nilai pi2 = %f \n", pi2);
14
15     return 0;
16 }
```

Apa perbedaan pi1 dan pi2?

## 3.2 math.h

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```

1  /* tulis komentar tentang kode anda disini */
2
3
4  #include <stdio.h>
5  #include <math.h>
6
7  #define pi 3.14159
8
9  using namespace std;
10
11 int main()
12 {
13     float sudut = 38; // dalam derajat
14
15     printf(" sin(%f) = %f \n", sin(pi*sudut/180.0));
16     printf(" cos(%f) = %f \n", cos(pi*sudut/180.0));
17
18     return 0;
19 }
```

Berapa nilai  $\sin(38)$  dan  $\cos(38)$ ?  
 Tambahkan fungsi-fungsi matematis lainnya.

## 3.3 Latihan-latihan

1. Buatlah program untuk mengkonversi sebuah nilai temperature dalam derajat Celcius (TC) ke derajat Fahrenheit (TF ). Gunakan rumus

$$T_F = \frac{9}{5}T_C + 32 \quad (3.1)$$

. Disini input program adalah TC dan output program adalah TF .

2. Buatlah program yang diberikan input tiga bilangan (x, y dan z) dan hitung nilai rata-rata bilangan-bilangan ini dengan dua rumus:  $s = (x + y + z) / 3$  dan  $p = \sqrt[3]{xyz}$ . Disini input program adalah x, y dan z, dan output program adalah s dan p.



# Bab 4

## Lab 4: Pengenalan for, while dan do while

### 4.1 for

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <stdio.h>
5
6  int main()
7  {
8      int i;
9      int s;
10
11     for (i=1; i <= 20; i++){
12         printf("  nilai i = %d \n", i);
13     }
14
15     s = 0;
16     for (i=1; i <= 20; i++){
17         s = s + 1;
18     }
19
20     printf("  nilai s = %d \n", s);
21
22     return 0;
23 }
```

Apa yang ditampilkan pada layar? berapakah nilai  $s$ ? Mengapa nilai  $s$  demikian? Jika menggunakan rumus deret aritmatika atau dengan kalkulator, hitunglah nilai  $s$ . Bandingkan hasilnya.

## 4.2 while

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <stdio.h>
5  #include <math.h>
6
7  int main()
8  {
9      int i, n;
10     float c;
11     float dx;
12     float x;
13
14     dx = 0.1;
15     n=10;
16
17     i=1;
18     printf(" NO  X  SIN(X) \n");
19     while(i<n){
20         x = i*dx;
21         c = cos(x);
22         printf("%d  %f  %f \n",i,x,c );
23
24         i++;
25     }
26
27     return 0;
28 }
```

Apakah yang dilakukan oleh program C di atas? Apa kegunaan baris `i=1;` dan `i++;`?

## 4.3 do-while

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <stdio.h>
5  #include <math.h>
6
7  int main()
8  {
9      int i, n;
10     float c;
11     float dx;
12     float x;
13
14     dx = 0.1;
15     n= 10;
16
17     i=1;
18     printf(" NO  X  SIN(X)  \n");
19     do{
20         x = i*dx;
21         c = cos(x);
22         printf("%d  %f  %f  \n",i,x,c );
23
24         i++;
25     }while(i<n);
26
27     return 0;
28 }
```

Apakah yang ditampilkan dilayar? apakah sama dengan hasil yang diperoleh dengan menggunakan while saja? Perhatikan perbedaan penggunaan **while** dan **do...while** di atas? Jika kita ganti baris **n=10;** menjadi **n=0;** pada kedua contoh kode menggunakan **while** dan **do...while**, apa yang ditampilkan pada layar? Apa perbedaannya antara kedua program tersebut?



# Bab 5

## Lab 5: Pengenalan if, else dan switch

### 5.1 if

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <stdio.h>
5
6  int main()
7  {
8      float x, y;
9
10     printf("Masukkan nilai x:\n");
11     scanf("%f",&x);
12
13     if(x<5.0) printf("Angka lebih kecil dari 5.0 \n");
14
15     if(x>5.0){
16         printf("Angka lebih besar dari 5.0 \n");
17     }
18
19     return 0;
20 }
```

Apa yang ditampilkan pada layar? Apa perbedaan dua fungsi if di atas?

## 5.2 if else

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <stdio.h>
5
6  int main()
7  {
8      float x, y;
9
10     printf("Masukkan nilai x:\n");
11     scanf("%f",&x);
12
13     if(x<5.0) printf("Angka lebih kecil dari 5.0 \n");
14     else printf("Angka lebih dari/sama dengan 5.0 \n");
15
16     return 0;
17 }
```

Apa yang ditampilkan pada layar? Perhatikan penggunaan else. Apa perbedaan penggunaan if pada program sebelumnya?

## 5.3 if else berantai

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <stdio.h>
5
6  int main()
7  {
8      float x;
9
10     printf("Masukkan nilai x:\n");
11     scanf("%f",&x);
12
13     if(x<5.0) printf("Angka lebih kecil dari 5.0 \n");
14     else if(x<10.0) printf("Angka lebih besar dari 5.0 dan lebih
        kecil dari 10.0 \n");
```

```

15     else if(x<100.0) printf("Angka lebih besar dari 10.0 dan
16         lebih kecil dari 100.0 \n");
17     else if(x<1000.0) printf("Angka lebih besar dari 100.0 dan
18         lebih kecil dari 1000.0 \n");
19     else printf("Angka lebih besar dari 1000.0\n");
20     return 0;
    }

```

Apa yang ditampilkan pada layar? Perhatikan penggunaan else. Apa perbedaan penggunaan if pada program sebelumnya?

## 5.4 ()?::

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```

1
2  /* tulis komentar tentang kode anda disini */
3
4  #include<stdio.h>
5
6  int main()
7  {
8      float x, y, z;
9
10     printf("Masukkan nilai x:\n");
11     scanf("%f",&x);
12
13     printf("Masukkan nilai y:\n");
14     scanf("%f",&y);
15
16     z = (x<y)? x: y;
17
18     printf("Nilai z = %f\n",z);
19     return 0;
20 }

```

Apa yang ditampilkan pada layar? Perhatikan penggunaan operator conditional ((?:)). Nilai apa yang ditampilkan oleh program? lebih besar atau lebih kecil?

## 5.5 switch

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2 /* tulis komentar tentang kode anda disini */
3
4 #include<stdio.h>
5
6 int main()
7 {
8     int a;
9
10    printf("Masukkan bilangan bulat antara 1 sampai 10:\n");
11    scanf("%d",&a);
12
13    switch(a){
14        case 1:
15        printf("Pilihan Satu\n");
16        break;
17        case 2:
18        printf("Pilihan Dua\n");
19        break;
20        case 3:
21        printf("Pilihan Tiga\n");
22        break;
23        case 4:
24        printf("Pilihan Empat\n");
25        break;
26        default:
27        printf("Pilihan Lain-lain\n");
28    }
29
30    return 0;
31 }
```

Apa yang ditampilkan pada layar? Perhatikan penggunaan **switch**.



# Bab 6

## Lab 6: Pengenalan Fungsi dan Array

### 6.1 Fungsi

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <stdio.h>
5
6  int jumlah(int a, int b);
7
8  int bagidua(int a);
9
10 int main()
11 {
12     int a = 3, b = 5, c;
13
14     c = jumlah(a,b);
15
16     printf(" nilai a = %d \n", a);
17     printf(" nilai b = %d \n", b);
18     printf(" nilai jumlah(a,b) = %d", c);
19
20     return 0;
21 }
22
23 int jumlah(int a, int b)
24 {
25     return (a+b);
26 }
```

```
27
28 int bagidua(int a)
29 {
30     return (a/2);
31 }
```

Jelaskan apa saja yang diperlukan untuk mendefinisikan sebuah fungsi? dan apa kegunaan fungsi jumlah dan bagidua?

## 6.2 Pengenalan Array

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2 /* tulis komentar tentang kode anda disini */
3
4 #include <stdio.h>
5
6 int main()
7 {
8     float a[10];
9     double b[3][4];
10
11     a[0] = 7;
12     a[1] = 10;
13     a[11] = 1;
14     b[2][3] = 1002;
15
16     printf(" a[0] = %f \n", a[0]);
17     printf(" a[1] = %f \n", a[1]);
18     printf(" a[4] = %f \n", a[4]);
19     printf(" b[2][3] = %f \n", b[2][3]);
20
21     return 0;
22 }
```

Apa kesalahan kode di atas? apa yang tampil di layar ketika kita kompilasi kode ini? dan perbaiki kode ini sehingga dapat dikompilasi. Apa yang ditampilkan pada layar setelah kita dapat mengkompilasi.

# Bab 7

## Lab 7: Pengenalan Pointer

### 7.1 Pointer

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <stdio.h>
5
6  int main()
7  {
8      float a; // variabel tipe float
9      float *p; // sebuah pointer tipe float
10     a = 2.3;
11     p = &a; // variabel p berisi alamat variabel a
12
13     printf("%f\n", a);
14
15     *p = 4.2; // kita ubah nilai variabel a
16
17     printf("%f\n", a);
18
19     return 0;
20 }
```

Dengan menggunakan kode ini jelaskan apa saja yang diperlukan untuk mendefinisikan sebuah pointer? dan bagaimana menggunakan pointer?

## 7.2 Pengenalan Operasi Pada Pointer

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <stdio.h>
5
6  int main()
7  {
8      int *p, b[5];
9      b[1] = 1;
10     b[2] = 2;
11     b[3] = 3;
12     b[4] = 4;
13
14     p = b; // p menunjuk ke alamat b[0]
15
16     printf(" nilai yang ditunjuk oleh p = %d\n", *p);
17
18     p = p + 1; // p menunjuk ke alamat b[1]
19
20     printf(" nilai yang ditunjuk oleh p = %d\n", *p);
21
22     p--; // kembali ke alamat b[0]
23
24     printf(" nilai yang ditunjuk oleh p = %d\n", *p);
25
26     p = p + 3; // p menunjuk ke alamat b[3]
27
28     printf(" nilai yang ditunjuk oleh p = %d\n", *p);
29
30     return 0;
31 }
```

Apa yang ditampilkan pada layar setelah kita dapat mengkompilasi? Pelajari operasi-operasi yang dapat dilakukan untuk pointer.

## 7.3 Latihan

1. Baca program C++ dibawah ini.

```
1  #include <stdio.h>
2
```

```
3 void swap(float *, float *);
4
5 int main()
6 {
7     float a = 2.5, b = 6.0;
8
9     printf("%f %f\n", a, b);
10
11    swap(&a, &b);
12
13    printf("%f %f\n", a, b);
14
15    return 0;
16 }
17
18 void swap(float *x, float *y)
19 {
20     float t;
21
22     t = *x;
23     *x = *y;
24     *y = t;
25 }
```

Apa yang dilakukan oleh fungsi swap?

2. Buatlah program yang diberikan input tiga bilangan ( $x$ ,  $y$  dan  $z$ ) dan hitung nilai rata-rata bilangan-bilangan ini dengan dua rumus:  $s = (x + y + z)/3$  dan  $p = \sqrt[3]{xyz}$ . Disini input program adalah  $x$ ,  $y$  dan  $z$ , dan output program adalah  $s$  dan  $p$ .



# Bab 8

## Lab 1: Pengenalan Bahasa C++

Pada Lab 1 ini kita akan mempelajari tentang bahasa C++ yang sederhana dan bagaimana mengkompilasi dan menjalankan program yang kita buat. Untuk lebih mengerti proses yang akan dilakukan dalam bahasa C++, mari kita lakukan hal-hal dibawah ini.

### 8.1 Penulisan Kode C++

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1  /* tulis komentar tentang kode anda disini */
2
3
4  #include <iostream>
5
6  using namespace std;
7
8  int main()
9  {
10     cout << " Hello C++" << endl;
11     cout << " Saya bisa membuat program dengan C++" << endl;
12
13     return 0;
14 }
```

Setelah selesai mengetik kode di atas, kemudian simpan kode tersebut ke file dengan nama dan ekstensi **.cpp**.

Kemudian jalankan program cygwin dan ketik `cd nama/folder/lokasi/file/cpp` untuk pindah ke folder tempat file yang telah kita ketik berada.

Kemudian kita kompilasi kode tersebut dengan salah satu perintah berikut ini:

```
1  
2 $ g++ nama.cpp  
3 $ g++ nama.cpp -o nama.exe
```

Ingat tanda `$` tidak diketik melainkan hanya menunjukkan baris perintah pada cygwin atau **command prompt**.

## 8.2 Modifikasi

Langkah selanjutnya adalah mengubah program sederhana di atas, dengan menambahkan beberapa baris perintah. Cobalah tambahkan baris ini satu per satu dan lihat apa yang dihasilkan dilayar setelah program dieksekusi.

```
1  
2 cout <<" ini tab a\t b \t c" << endl;  
3 cout <<" ini baris baru a\n b \n c \n" << endl;  
4 cout <<" ini garis miring \\ \" " << endl;
```

Cobalah modifikasi apa saja yang anda ingin tampilkan pada layar.



# Bab 9

## Lab 2: Pengenalan Variabel dan Operasi Matematis

### 9.1 Deklarasi Variabel

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <iostream>
5
6  using namespace std;
7
8  int main()
9  {
10     int a, b = 3, c(2);
11
12     cout << " nilai a = " << a << endl;
13     cout << " nilai b = " << b << endl;
14     cout << " nilai c = " << c << endl;
15
16     a = b + c;
17
18     cout << " nilai a = " << a << endl;
19
20     return 0;
21 }
```

Berapakah nilai  $a$ ,  $b$  dan  $c$ ?

## 9.2 Kesalahan Penulisan Variabel

Tambahkan kode berikut ini dan catat apa kesalahan yang muncul pada waktu kita kompilasi kode.

```
1
2  int 2apel;
3  long banyak!;
4  double desi-mal;
5  double desi_mal;
6  double DESI_mal;
```

## 9.3 Tipe Data Variable

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <iostream>
5
6  using namespace std;
7
8  int main()
9  {
10     int a = 2, b = 3, c;
11     float i = 2, j = 3, k;
12
13     c = 2/3;
14     k = i/j;
15     cout << " nilai c = " << c << endl;
16     cout << " nilai b = " << b << endl;
17
18     a = i + j;
19
20     return 0;
21 }
```

Berapakah nilai  $c$  dan  $k$ ? Apa yang ditampilkan pada layar ketika kita kompilasi? dan ketika program dieksekusi? Mengapa nilai  $c$  tidak sama dengan  $k$ ?

# Bab 10

## Lab 3: Pengenalan Define dan cmath

### 10.1 Define dan const

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <iostream>
5
6  #define pi1 3.14159
7
8  using namespace std;
9
10 int main()
11 {
12     const float pi2 = 3.14159;
13
14     cout << " nilai pi1 = " << pi1 << endl;
15     cout << " nilai pi2 = " << pi2 << endl;
16
17     return 0;
18 }
```

Apa perbedaan pi1 dan pi2?

## 10.2 cmath

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```

1  /* tulis komentar tentang kode anda disini */
2
3
4  #include<iostream>
5  #include<cmath>
6
7  #define pi 3.14159
8
9  using namespace std;
10
11 int main()
12 {
13     float sudut = 38; // dalam derajat
14
15     cout << " sin(" << sudut <<") = " << sin(pi*sudut/180.0) <<
16         endl;
17     cout << " cos(" << sudut <<") = " << cos(pi*sudut/180.0) <<
18         endl;
19
20     return 0;
21 }
```

Berapa nilai  $\sin(38)$  dan  $\cos(38)$ ?  
 Tambahkan fungsi-fungsi matematis lainnya.

## 10.3 Latihan-latihan

1. Buatlah program untuk mengkonversi sebuah nilai temperature dalam derajat Celcius (TC) ke derajat Fahrenheit (TF ). Gunakan rumus

$$T_F = \frac{9}{5}T_C + 32 \quad (10.1)$$

. Disini input program adalah TC dan output program adalah TF .

2. Buatlah program yang diberikan input tiga bilangan (x, y dan z) dan hitung nilai rata-rata bilangan-bilangan ini dengan dua rumus:  $s = (x + y + z) / 3$  dan  $p = \sqrt[3]{xyz}$ . Disini input program adalah x, y dan z, dan output program adalah s dan p.

# Bab 11

## Lab 4: Pengenalan Fungsi dan Array

### 11.1 Fungsi

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <iostream>
5
6  using namespace std;
7
8  int jumlah(int a, int b);
9
10 int main()
11 {
12     int a = 3, b = 5, c;
13
14     c = jumlah(a,b);
15
16     cout << " nilai a = " << pi1 << endl;
17     cout << " nilai b = " << pi2 << endl;
18     cout << " nilai jumlah(a,b) = " << c << endl;
19
20     return 0;
21 }
22
23 int jumlah(int a, int b)
24 {
25     return (a+b);
26 }
```

Jelaskan apa saja yang diperlukan untuk mendefinisikan sebuah fungsi? dan apa kegunaan fungsi jumlah?

## 11.2 Pengenalan Array

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include<iostream>
5  using namespace std;
6
7  int main()
8  {
9      float a[10];
10     double b[3][4];
11
12     a[0] = 7;
13     a[1] = 10;
14     a[11] = 1;
15     b[2][3] = 1002;
16
17     cout << " a[0] = " << a[0] << endl;
18     cout << " a[1] = " << a[1] << endl;
19     cout << " a[4] = " << a[4] << endl;
20     cout << " b[2][3] = " << b[2][3] << endl;
21
22     return 0;
23 }
```

Apa kesalahan kode di atas? apa yang tampil di layar ketika kita kompilasi kode ini? dan perbaiki kode ini sehingga dapat dikompilasi. Apa yang ditampilkan pada layar setelah kita dapat mengkompilasi.

# Bab 12

## Lab 5: Pengenalan Pointer

### 12.1 Pointer

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <iostream>
5
6  using namespace std;
7
8  int main()
9  {
10     float a; // variabel tipe float
11     float *p; // sebuah pointer tipe float
12     a = 2.3;
13     p = &a; // variabel p berisi alamat variabel a
14
15     cout << a << endl;
16
17     *p = 4.2; // kita ubah nilai variabel a
18
19     cout << a << endl;
20 }
21 return 0;
22 }
```

Dengan menggunakan kode ini jelaskan apa saja yang diperlukan untuk mendefinisikan sebuah pointer? dan bagaimana menggunakan pointer?

## 12.2 Pengenalan Operasi Pada Pointer

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <iostream>
5
6  using namespace std;
7
8  int main()
9  {
10     int *p, b[5];
11     b[1] = 1;
12     b[2] = 2;
13     b[3] = 3;
14     b[4] = 4;
15
16     p = b; // p menunjuk ke alamat b[0]
17
18     cout << " nilai yang ditunjuk oleh p = " << *p << endl;
19
20     p = p + 1; // p menunjuk ke alamat b[1]
21
22     cout << " nilai yang ditunjuk oleh p = " << *p << endl;
23
24     p--; // kembali ke alamat b[0]
25
26     cout << " nilai yang ditunjuk oleh p = " << *p << endl;
27
28     p = p + 3; // p menunjuk ke alamat b[3]
29
30     cout << " nilai yang ditunjuk oleh p = " << *p << endl;
31
32     return 0;
33 }
```

Apa yang ditampilkan pada layar setelah kita dapat mengkompilasi? Pelajari operasi-operasi yang dapat dilakukan untuk pointer.

## 12.3 Latihan

1. Baca program C++ dibawah ini.



```
1 #include<iostream>
2
3 using namespace std;
4
5 void swap(float *, float *);
6
7 int main()
8 {
9     float a = 2.5, b = 6.0;
10
11     cout << a << " " << b << endl;
12
13     swap(&a, &b);
14
15     cout << a << " " << b << endl;
16
17     return 0;
18 }
19
20 void swap(float *x, float *y)
21 {
22     float t;
23
24     t = *x;
25     *x = *y;
26     *y = t;
27 }
```

Apa

2. Buatlah program yang diberikan input tiga bilangan ( $x$ ,  $y$  dan  $z$ ) dan hitung nilai rata-rata bilangan-bilangan ini dengan dua rumus:  $s = (x + y + z)/3$  dan  $p = \sqrt[3]{xyz}$ . Disini input program adalah  $x$ ,  $y$  dan  $z$ , dan output program adalah  $s$  dan  $p$ .



# Bab 13

## Lab 6: Pengenalan Character dan String

### 13.1 Character

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <iostream>
5
6  using namespace std;
7
8  int main()
9  {
10     char a;
11     char nama[30];
12     char ilmuwan[30]={ 'N', 'E', 'W', 'T', 'O', 'N' };
13     char satuan[30]= newton ;
14
15     a = 'H';
16
17     cout << " nilai a = " << a << endl;
18     cout << " nilai nama = " << nama << endl;
19     cout << " nilai ilmuwan = " << ilmuwan << endl;
20     cout << " nilai satuan = " << satuan << endl;
21
22     return 0;
23 }
```

Bagaimana mendefinisikan sebuah tipe data char? Apa yang ditampilkan pada layar?

## 13.2 String

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <iostream>
5  #include <string>
6
7  using namespace std;
8
9  int main()
10 {
11     string kalimatUtama;
12     string kataSambung;
13     string kalimatPenting;
14
15     kalimatUtama = Saya pasti bisa C++ ;
16     kataSambung = karena ;
17     kalimatPenting = C++ mudah ;
18     kalimatUtama += kataSambung + kalimatPenting;
19
20     cout << kalimatUtama << endl;
21
22     return 0;
23 }
```

Bagaimana mendefinisikan sebuah string? dan bagaimana menggunakannya?

## 13.3 getline()

Kita menggunakan fungsi `getline()` untuk mengambil input dalam satu baris.

```
1
2  #include <iostream>
3  #include <string>
4
5  using namespace std;
6
```

```
7 int main()
8 {
9     string kalimat;
10    cout << "masukkan sebuah kalimat : <<endl;
11
12    getline(cin, kalimat);
13
14    cout << "kalimat anda adalah: \n << kalimat;
15
16    return 0;
17 }
```

Jadi yang perlu diingat adalah cara pemakaian `getline(cin, kalimat)`. Modifikasi kode ini untuk keperluan anda.



# Bab 14

## Lab 7: Pengenalan Object Oriented Programming (OOP) dengan Class

### 14.1 Struct

Sebelum kita mempelajari tentang OOP, kita perlu mengenal terlebih dahulu tentang struktur data dengan menggunakan **struct**.

Struktur data (data structure) adalah sebuah grup/gabungan elemen data menjadi satu dalam satu nama. Dengan kata lain kita mendefinisikan tipe data yang baru. Elemen data dinamakan anggota struktur. Anggota bisa bertipe apa saja dan berbeda dan ukuran berbeda.

Untuk membuat struktur data kita menggunakan kata kunci atau keyword **struct** dengan format sebagai berikut:

```
1 struct namastruktur{  
2     tipe1 nama_anggota1;  
3     tipe2 nama_anggota2;  
4     ...  
5 } nama_variabel;
```

Sebagai contoh data mahasiswa berikut ini.

```
1  
2 struct mahasiswa {  
3     string nama;  
4     string nim;  
5     long int umur;  
6     float tinggi;  
7     float berat;
```

```

8 } andi;    // definisi mahasiswa dan deklarasi andi
9
10 // deklarasi variabel bella
11 mahasiswa bella;

```

Contoh cara akses anggota struct dengan tanda titik dan nama anggota:  
Untuk lebih jelas, kita ketik kode berikut ini.

```

1
2 /* tulis komentar tentang kode anda disini */
3
4 #include <iostream>
5 #include <string>
6
7 using namespace std;
8
9 // definisi struktur mahasiswa
10 struct mahasiswa {
11     string nama;
12     string nim;
13     long int umur;
14     float tinggi;
15     float berat;
16 };
17
18 int main()
19 {
20     mahasiswa m;
21     m.nama = Albert Einstein ;
22     m.umur = 25;
23     m.tinggi = 170.0;
24
25     cout << " nama = " << m.nama << endl;
26     cout << " umur = " << m.umur;
27
28     return 0;
29 }

```

Hal-hal apa saja yang diperlukan untuk membentuk struktur data?

## 14.2 Class

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.



```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <iostream>
5
6  using namespace std;
7
8  class Posisi{
9      protected:
10         double x, y;
11     public:
12         void Posisi(double ,double );
13         double getX();
14         double getY();
15 };
16
17 int main()
18 {
19     Posisi a, b(2,3);
20
21     a = Posisi(4,2);
22     b = a;
23     cout << " nilai komponen x dari a = " << a.getX() << endl;
24     cout << " nilai komponen y dari a = " << a.getY() << endl;
25
26     return 0;
27 }
28
29 double Posisi::getX(){
30     return x;
31 }
32 double Posisi::getY(){
33     return y;
34 }
35 double Posisi::Posisi(double a, double b){
36     x = a; y = b;
37 }
```

Bagaimana mendefinisikan sebuah tipe data baru dengan class? Bagaimana penggunaannya?

## 14.3 Inheritance

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <iostream>
5
6  using namespace std;
7
8  class Posisi{
9      protected:
10         double x, y;
11     public:
12         void Posisi(double, double);
13         double getx();
14         double gety();
15 };
16
17 class Lingkaran: public Posisi{
18     double r;
19
20     Public:
21         void Lingkaran(double, double, double);
22         double getr();
23 };
24 int main()
25 {
26     Lingkaran g(2,3,10);
27
28     cout << " nilai komponen x dari g = " << g.getx() << endl;
29     cout << " nilai komponen y dari g = " << g.gety() << endl;
30     cout << " nilai jari-jari r dari g = " << g.getr() << endl;
31
32     return 0;
33 }
34
35 double Posisi::getx(){
36     return x;
37 }
38 double Posisi::gety(){
39     return y;
40 }
41 double Posisi::Posisi(double a, double b){
42     x = a; y = b;
43 }
44
45 double Lingkaran::getr(){
46     return r;
47 }
48
```

```
49 double Lingkaran::Lingkaran(double a, double  
50 b, double c){  
51     x = a; y = b; r = c;  
52 }
```

Bagaimana mendefinisikan sebuah tipe data baru yang merupakan penurunan dari tipe data yang lain? dan bagaimana menggunakannya? Mengapa konsep inheritance berguna? keuntungannya apa?



# Bab 15

## Lab 8: Menggunakan Make

Dalam membuat program, terutama untuk program yang besar, kita perlu membagi kode program tersebut menjadi banyak fungsi-fungsi atau class-class yang selanjutnya disimpan dalam file/berkas yang berbeda. Ini dilakukan agar kode program yang kita buat bisa lebih manageable (lebih mudah diedit, dikoreksi dan dikompile). Sebagai contoh jika kita menulis sebuah buku, apakah kita menulis buku ini dalam sebuah file? Tentunya ini bisa dilakukan! Tetapi, alangkah lebih baik jika kita bagi file tersebut menjadi beberapa file sesuai dengan bab atau chapter. Jadi setiap kita bekerja kita membuka satu file yang ukurannya jauh lebih kecil dibandingkan sebuah file untuk satu buku. Begitu pula sebuah program, kita bisa lebih mudah mengerjakannya jika membagi kode program menjadi beberapa file.

Untuk lebih jelas, kita bisa lihat contoh kode program dibawah ini.

### Contoh 15.1. *(Kode Program dalam satu berkas/file)*

```
1 // nama file : factorial.cpp
2 #include <iostream>
3 #include <cmath>
4
5 using namespace std;
6
7 // Deklarasi atau prototype fungsi-fungsi
8 void input(long &);
9 long hitung_factorial(long);
10 void output(long);
11
12 int main()
13 {
14     long bilangan, fact;
15
16     input(bilangan);
17     fact = hitung_factorial(bilangan);
```

```

18     output(bilangan);
19
20     return 0;
21 }
22
23 //Definisi fungsi fungsi
24
25 void input(long &a)
26 {
27     cout << "Masukkan bilangan bulat : " << endl;
28     cin >> a;
29 }
30
31 void output(long a)
32 {
33     cout << "Faktorial bilangan ini adalah " << a << endl;
34 }
35
36
37 long hitung_factorial(long a)
38 {
39     long f=1;
40
41     for(long i = 1; i<=a; i++){
42         f *= i;
43     }
44
45     return f;
46 }

```

Kode program di atas bisa secara langsung dikompile dengan

```
1 g++ -O factorial.cpp -o fact.exe
```

Disini cara kompilasi ditambahkan `-O` (o besar, bukan angka nol) untuk membuat program yang dihasilkan lebih cepat.

Sebenarnya cara kompilasi diatas melakukan dua proses sekaligus. Kita bisa membuat *object* file terlebih dahulu dan kemudian membuat program file. Seperti contoh di bawah ini.

```

1 g++ -c -O factorial.cpp
2
3 g++ factorial.o -o fact.exe

```

Perintah pertama kita menggunakan tambahan `-c` pada `++_` untuk mengompilasi ke object atau tanpa membuat program (proses compile). File yang dihasilkan berakhiran `.o` (atau object file). Karena nama filenya adalah `factorial.cpp` maka dihasilkan `factorial.o`. Setelah dikompilasi kita bisa membuat program dengan perintah kedua (disebut proses linking). Nanti akan lebih jelas mengapa ini diperlukan.

Kode program di atas kita akan bagi menjadi tiga file yaitu `main.cpp`, `factorial.cpp` dan `factorial.h`.

**Contoh 15.2.** *(Kode Program dibagi menjadi tiga berkas/file)*

```

1  ----- file: factorial.h -----
2  // Deklarasi atau prototype fungsi-fungsi
3  void input(long &);
4  long hitung_factorial(long);
5  void output(long);
6
7
8  ----- file: main.cpp -----
9  #include <iostream>
10 #include <cmath>
11 #include "factorial.h"
12
13 using namespace std;
14
15
16 int main()
17 {
18     long bilangan, fact;
19
20     input(bilangan);
21     fact = hitung_factorial(bilangan);
22     output(bilangan);
23
24     return 0;
25 }
26
27 ----- file: factorial.cpp -----
28
29 //Definisi fungsi fungsi
30 #include "factorial.h"
31 #include <iostream>
32
33 using namespace std;
34
35 void input(long &a)
36 {
37     cout << "Masukkan bilangan bulat : " << endl;

```

```

38     cin >> a;
39 }
40
41 void output(long a)
42 {
43     cout << "Faktorial bilangan ini adalah " << a << endl;
44 }
45
46
47 long hitung_factorial(long a)
48 {
49     long f=1;
50
51     for(long i = 1; i<=a; i++){
52         f *= i;
53     }
54
55     return f;
56 }

```

Jadi di atas, file **factorial.h** yang juga disebut header file mengandung deklarasi/prototipe dari fungsi-fungsi atau class-class. Sedangkan **factorial.cpp** mengandung definisi fungsi atau class.

Cara kompilasi dan linking secara manual sebagai berikut:

```

1 g++ -c -O main.cpp
2 g++ -c -O factorial.cpp
3 g++ main.o factorial.o -o fact.exe

```

Perintah pertama dan kedua di atas dilakukan untuk mengkompilasi dua file bagian kode program. Kemudian dua object file digabungkan (atau linking) menjadi sebuah program.

Keuntungan dengan memecah file menjadi beberapa file adalah lebih mudah mengompilasi program karena kita hanya perlu mengompilasi file yang diedit saja. Object file yang lain tidak berubah jadi tidak perlu dikompilasi lagi. Sebagai contoh, umpamanya jika kita hanya mengedit file **main.cpp** maka untuk mengompilasi program cukup kita melakukan dua perintah diatas, mengompilasi kode **cpp** dan melakukan linking.

```

1 g++ -c -O main.cpp
2 g++ main.o factorial.o -o fact.exe

```



Bisa dibayangkan jika kita mempunyai jumlah file yang banyak, proses kompilasi akan jauh lebih cepat dengan mengkompilasi satu atau beberapa file saja daripada semua file lagi.

Jika kita mempunyai sejumlah file dengan akhiran `.cpp` maka kita harus menggunakan perintah kompilasi sebanyak jumlah file tersebut. Akan menjadi sangat tidak efisien jika ini kita lakukan secara manual. Untuk hal ini program `make` bisa menolong kita untuk mengkompilasi secara otomatis.

Untuk menggunakan program `make` kita harus membuat file dependency yang diberi nama *Makefile* atau *makefile*.

File **Makefile** berisi cara mengkompilasi dan file apa saja yang diperlukan. Sebagai contoh untuk kompilasi program sebelumnya file **Makefile** yang sederhana sebagai berikut.

```

1 program : main.o factorial.o
2   g++ main.o factorial.o -o fact.exe
3
4 main.o   : main.cpp
5   g++ -c main.cpp
6
7 factorial.o : factorial.cpp
8   g++ -c factorial.cpp
9
10 clean :
11   rm *.o

```

Baris pertama di file **Makefile** di atas terdiri dari nama/keyword untuk sebuah project yang akan dibuat/dikerjakan. Setelah nama/keyword ini diikuti tanda `:`, dan diikuti dengan nama-nama file yang diperlukan dalam membuat project tersebut. Setelah baris ini diikuti dengan baris perintah atau cara mengkompilasi/membuat project. Setelah itu kita membuat `main.o`. Dan seterusnya. Dua baris terakhir digunakan untuk menghapus semua object file jika sudah tidak diperlukan lagi. Penting diingat bahwa untuk baris perintah/cara membuat dicorongkan ke dalam menggunakan tombol tab.

Setelah file **Makefile** dibuat, kita bisa menjalankan *make* untuk memproses **Makefile** dengan perintah:

```

1 make program

```

Perintah ini menunjukkan kita melakukan/memproses nama/keyword project `program`.

Jika kita hanya memproses project pada baris pertama kita bisa langsung menggunakan `make` saja tanpa nama project/keyword.

Jika kita ingin mendelete semua object file, kita menggunakan perintah `make clean`.

# Bab 16

## Lab 9: Pengenalan Library GSL

GSL (GNU Scientific Library) merupakan library ini sangat penting untuk pemrograman pada khususnya untuk komputasi numerik. Oleh karena itu ini sangat berguna untuk fisikawan yang memerlukan perhitungan yang menggunakan operasi-operasi dan fungsi-fungsi yang tidak ada pada library cmath. Library GSL mempunyai banyak hal yang bisa digunakan untuk memecahkan banyak permasalahan di Fisika.

Sebagai contoh, hal-hal yang terkandung pada GSL library seperti berikut ini:

1. Complex Numbers
2. Roots of Polynomials
3. Special Functions
4. Vectors and Matrices
5. Permutations ,Sorting
6. Linear Algebra, Eigensystems
7. Fast Fourier Transforms

Dan Banyak lagi fasilitas yang tersedia pada GSL.

GSL dapat diperoleh gratis dengan mengunduh di <http://www.gnu.org/software/gsl/>.

Library ini sudah tersedia di Cygwin. Terutama jika kita menginstal semua software yang ada pada Cygwin. Kita bisa instal GSL dengan memilih semua pada pilihan lib dan math pada saat instalasi Cygwin.

Untuk mengenal bagaimana cara penggunaan GSL, perhatikan contoh berikut ini:

## 16.1 Contoh GSL

Dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode c++ berikut ini.

```
1
2  /* tulis komentar tentang kode anda disini */
3
4  #include <iostream>
5  #include <gsl/gsl_sf_bessel.h>
6
7  using namespace std;
8
9  int main (void)
10 {
11     double x = 5.0;
12     double y = gsl_sf_bessel_J0 (x);
13     cout << y;
14
15     return 0;
16 }
```

Apa saja yang diperlukan untuk menggunakan library GSL? Bagaimana menggunakannya?

# Bab 17

## Lab 10: Pengenalan Memori

Ada dua jenis memori yang bisa dialokasikan sesuai dengan deklarasinya: memori statis dan dinamis.

Sebagai contoh memori statis adalah dalam membuat suatu array, kita mendeklarasi dengan `float s[100];`. Dengan deklarasi ini, komputer mengalokasikan memori untuk menampung 100 variabel yang bertipe data float. Besarnya array `s` tidak bisa diubah atau statis. Jadi jika kita menggunakan metode ini kita harus menyiapkan jumlah array yang cukup. Jika kurang akan terjadi error atau kesalahan karena lokasi memori tidak ada.

Sedangkan memori dinamis, besarnya memori yang dipakai dapat diubah sesuai kebutuhan. Jika kita memiliki array yang besarnya/ukurannya bisa berubah kita perlu menggunakan array dinamik. Caranya dengan menggunakan operator `new` dan `new []`.

### 17.1 New atau New []

Untuk menggunakan operator `new`, kita perlu menambahkan `#include<new>`. Dan untuk alokasi memori, kita memerlukan sebuah pointer dan dengan cara `pointer = new tipe[jumlahnya];`. Setelah selesai menggunakan memori atau jika kita tidak memerlukan lagi alokasi memori, kita perlu menghapus array itu agar memori bisa digunakan untuk hal yang lain. Maka pada akhir program biasanya ada operator `delete`. Cara penggunaan `delete` adalah

```
1 delete pointer; // untuk satu elemen
2 delete [] pointer; // untuk array
```

Untuk mempelajari penggunaannya, dengan menggunakan **notepad** atau **notepad++** atau **nedit** ketik beberapa baris kode `c++` berikut ini.

```
1 #include <iostream>
2 #include <string>
3
4 using namespace std;
5
6 int main()
7 {
8     int n;
9     float *a; // pointer untuk alokasi memory
10
11     cout << "berapa jumlah array? ";
12     cin >> n;
13
14     a = new float[n];
15     a[9] = 3;
16
17     delete [] a;
18     return 0;
19 }
```

Jelaskan apa saja yang diperlukan untuk membuat array dinamis?