

**MODUL PRAKTIKUM**  
**BAHASA PEMROGRAMAN DASAR**  
**(PG168)**



**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS BUDI LUHUR**  
**2015**

## DAFTAR ISI

DAFTAR ISI .....	ii
PRAKTIKUM 1 PENGENALAN BAHASA C.....	1
1.1 TUJUAN PRAKTIKUM.....	1
1.2 TEORI SINGKAT .....	1
1.3 PELAKSANAAN PRAKTIKUM.....	2
1.4 LATIHAN .....	4
1.5 TUGAS MANDIRI .....	5
PRAKTIKUM 2 STRUKTUR DASAR BAHASA C .....	6
2.1 TUJUAN PRAKTIKUM.....	6
2.2 TEORI SINGKAT .....	6
2.2.1 Tipe Data .....	6
2.2.2 Variabel .....	7
2.2.3 Karakter Khusus ( <i>Special Character</i> ) .....	7
2.2.4 Deklarasi.....	8
2.2.5 Operator .....	9
2.2.6 Kata Tercadang (Reserved Word) .....	11
2.2.7 Komentar Program .....	11
2.3 PELAKSANAAN PRAKTIKUM.....	11
2.4 LATIHAN .....	16
2.5 TUGAS MANDIRI .....	18

# **PRAKTIKUM 1**

## **PENGENALAN BAHASA C**

### **1.1 TUJUAN PRAKTIKUM**

#### **Tujuan Umum**

Mahasiswa memahami konsep dasar bahasa pemrograman serta mampu membuat dan menjalankan suatu program sederhana di komputer.

#### **Tujuan Khusus**

Mahasiswa dapat :

1. Menguraikan konsep dasar bahasa pemrograman dan kaitannya dengan algoritma.
2. Menyebutkan bermacam-macam jenis Bahasa Pemrograman
3. Menjelaskan cara kerja sebuah program dapat berjalan (kode sumber, kompilasi, executable)
4. Membuka dan memahami fitur dasar perangkat lunak editor pembuatan program (IDE).
5. Menulis, meng-kompilasi dan menjalankan (run) program sederhana

### **1.2 TEORI SINGKAT**

Bahasa C dirancang oleh Dennis M. Ritchie, seorang pegawai Bell Telephone Laboratories, Inc. di Murray Hill, New Jersey, Amerika Serikat (sekarang dikenal dengan AT&T Bell Laboratories) pada tahun 1972. Ketika itu ia sedang bertugas membuat sebuah sistem operasi yang terbuka dan interaktif untuk Bell Laboratories. Sistem operasi tersebut dikemudian hari dikenal dengan nama sistem operasi UNIX.

Pada mulanya bahasa pemrograman C digunakan dan dikembangkan hanya terbatas pada lingkungan Bell Laboratories saja, bersama dengan sistem operasi UNIX. Setelah Dennis Ritchie dan Brian Kernighan menerbitkan buku yang berjudul "The C Programming Language" (Bahasa Pemrograman C) pada tahun 1978, barulah bahasa pemrograman C dikenal dan berkembang luas.

Dalam perkembangannya, muncul banyak varian dari bahasa pemrograman C. Untuk menjaga kompatibilitas dan fleksibilitasnya, lembaga standarisasi ANSI (American National Standards Institute) menetapkan standar unsur-unsur bahasa pemrograman C yang harus terdapat pada suatu varian dari bahasa pemrograman C. Versi standar ini dikenal dengan sebutan ANSI C. Beberapa varian bahasa pemrograman C yang dikenal antara lain Microsoft C, Microsoft Quick C, Borland Turbo C, Borland C, Symantec C, Run/C dan Lattice C.

Bahasa pemrograman C merupakan bahasa pemrograman tingkat tinggi tetapi berorientasi pada sistem operasi komputer yang menggunakan operasi tingkat rendah (bahasa C dikembangkan bersama dengan sistem operasi UNIX, bahkan sistem operasi UNIX kemudian dibuat dengan menggunakan bahasa pemrograman C) sehingga banyak yang mengelompokkan bahasa pemrograman C sebagai bahasa pemrograman tingkat menengah (bukan tingkat tinggi, bukan pula tingkat rendah).

Salah satu ciri khas dari bahasa pemrograman C adalah programnya terdiri fungsi-fungsi serta seringnya digunakan tipe data pointer.

Berikut ini beberapa keunggulan Bahasa C:

- Bahasa C dapat dijalankan hampir di semua jenis komputer.
- Kode bahasa C sifatnya adalah portable dan fleksibel untuk semua jenis komputer.
- Bahasa hanya menyediakan sedikit kata-kata kunci, hanya terdapat 32 kata kunci.
- Proses executable program bahasa C lebih cepat
- Dukungan pustaka yang banyak.
- Bahasa C adalah bahasa yang terstruktur
- Bahasa C termasuk bahasa tingkat menengah

### 1.3 PELAKSANAAN PRAKTIKUM

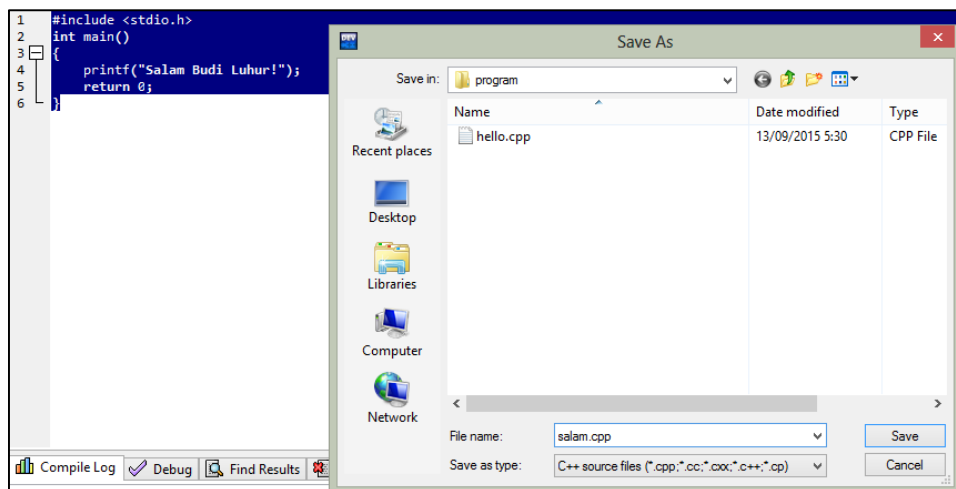
Langkah-langkah Praktikum

1. Buka Editor Bahasa C **Dev-C++ 5.11**.
2. Buatlah file baru dengan membuka menu **File > New > Source File** atau dengan shortcut **Ctrl + N**.
3. Tulislah Program 1.1 berikut ini.

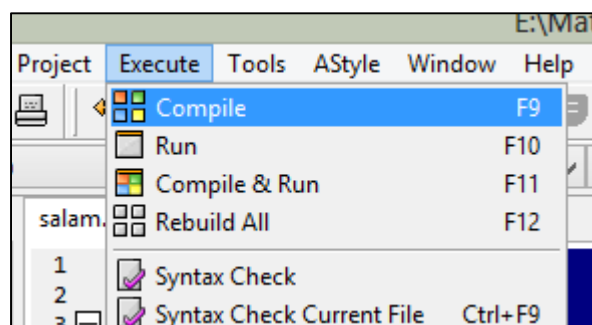
**Program 1.1 salam.cpp**

```
1 #include <stdio.h>
2 int main()
3 {
4     printf("Salam Budi Luhur!");
5     return 0;
6 }
```

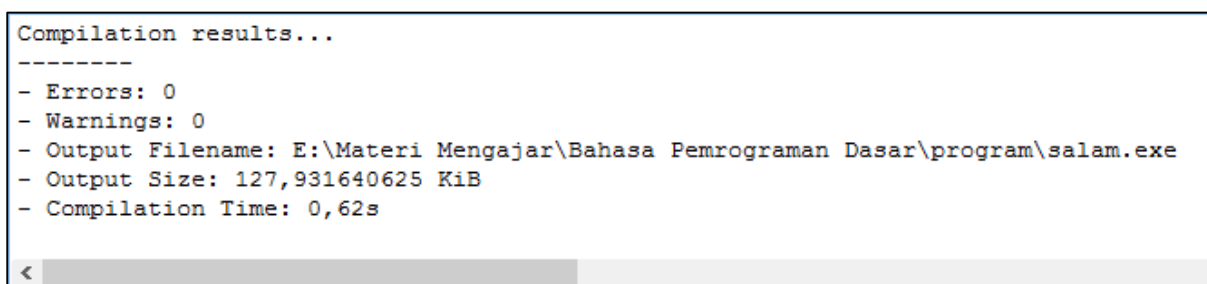
4. Simpan program yang telah dituliskan dengan membuka menu **File > Save as...**. Pilih lokasi penyimpanan dan beri nama file dengan "**salam.cpp**".



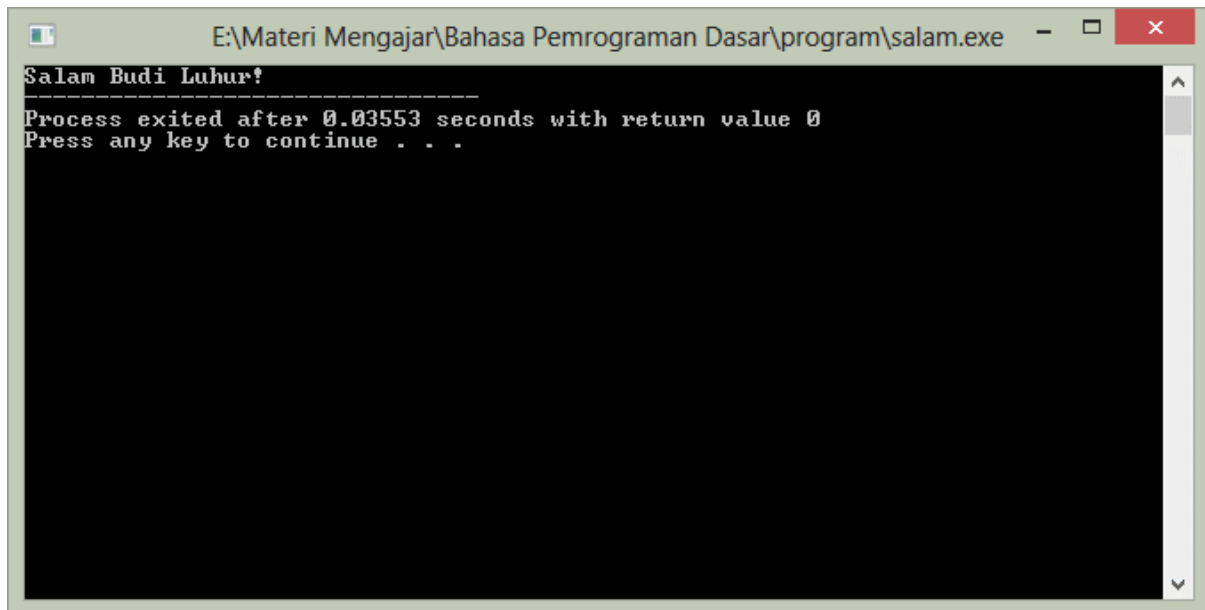
5. Lakukan kompilasi program melalui menu **Execute > Compile** atau dengan menekan shortcut **F9**.



6. Perhatikan hasil kompilasi program pada bagian "Compile Log". Jika kompilasi sukses, maka akan ditampilkan pesan kurang lebih sebagai berikut:



7. Jalankan program (running) melalui menu **Execute > Run** atau dengan menekan shortcut **F10**. Hasil program akan ditampilkan pada window seperti pada gambar berikut ini.



8. Proses kompilasi dan running dapat dilaksanakan sekaligus melalui menu **Execute > Compile & Run** atau dengan shortcut **F11**.
9. Selesai.

## 1.4 LATIHAN

Tuliskan dan jalankan beberapa program berikut ini dan tuliskan hasilnya di tempat yang sudah disediakan.

### Program 1.2: total1.cpp

```
1 #include "stdio.h"
2 int main()
3 {
4     printf("%i", (10 + 20));
5     return 0;
6 }
```

Hasil Program 1.2

**Program 1.3: total2.cpp**

```
1 #include "stdio.h"
2 int main()
3 {
4     int A, B;
5     A = 10;
6     B = 20;
7     printf("%i", (A+B));
8     return 0;
9 }
```

Hasil Program 1.3

**Program 1.4: total3.cpp**

```
1 #include "stdio.h"
2 int main()
3 {
4     int A, B, T;
5     A = 10;
6     B = 20;
7     T = A + B;
8     printf("%i", T);
9     return 0;
10 }
```

Hasil Program 1.4

## 1.5 TUGAS MANDIRI

Kerjakan soal-soal berikut ini:

1. Buatlah sebuah program Bahasa C yang menampilkan NIM, NAMA dan JURUSAN Anda di layar!
2. Buatlah sebuah program Bahasa C untuk menghitung luas persegi panjang dengan ukuran Panjang 10 cm dan lebar 7 cm!

# PRAKTIKUM 2

## STRUKTUR DASAR BAHASA C

### 2.1 TUJUAN PRAKTIKUM

#### Tujuan Umum

Mahasiswa dapat memahami:

1. Struktur penulisan bahasa pemrograman
2. Sintaks assignment statement dan output statement,
3. Keperluan sebuah variable,
4. Tipe data standar bahasa pemrograman.
5. Mengenal berbagai operator dalam Bahasa C

#### Tujuan Khusus

Mahasiswa dapat :

1. Menuliskan sintaks instruksi : assignment statement, dan output Statement
2. Mendeklarasikan dan menggunakan variabel dalam berbagai tipe data dalam sebuah program
3. Memilih tipe data sesuai dengan kegunaan data tersebut.
4. Menulis program untuk menampilkan isi dari suatu variabel
5. Menulis program untuk menampilkan string yang mengandung karakter khusus
6. Membuat program sederhana yang melibatkan berbagai operator
7. Memberi komentar program

### 2.2 TEORI SINGKAT

#### 2.2.1 Tipe Data

Tipe data merupakan bagian program yang paling penting karena tipe data mempengaruhi setiap instruksi yang akan dilaksanakan oleh komputer. Misalnya saja 5 dibagi 2 bisa saja menghasilkan hasil yang berbeda tergantung tipe datanya. Jika 5 dan 2 bertipe integer maka akan menghasilkan nilai 2, namun jika keduanya bertipe float maka akan menghasilkan nilai 2.5000000. Pemilihan tipe data yang tepat akan membuat proses operasi data menjadi lebih efisien dan efektif.

Dalam bahasa C terdapat lima tipe data dasar, yaitu :

No	Tipe Data	Ukuran	Range (Jangkauan)	Format	Keterangan
1	char	1 byte	-128 s/d 127	%c	Karakter/string
2	int	2 byte	-32768 s/d 32767	%i , %d	Integer/bilangan bulat



No	Tipe Data	Ukuran	Range (Jangkauan)	Format	Keterangan
3	float	4 byte	-3.4E-38 s/d 3.4E+38	%f	Float/bilangan pecahan
4	double	8 byte	-1.7E-308 s/d 1.7+308	%lf	Pecahan presisi ganda
5	void	0 byte	-	-	Tidak bertipe

### 2.2.2 Variabel

Variabel adalah suatu pengenal (identifier) yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program. Berbeda dengan konstanta yang nilainya selalu tetap, nilai dari suatu variable bisa diubah-ubah sesuai kebutuhan. Nama dari suatu variabel dapat ditentukan sendiri oleh pemrogram dengan aturan sebagai berikut :

1. Terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf. Bahasa C bersifat case-sensitive artinya huruf besar dan kecil dianggap berbeda. Jadi antara **nim**, **NIM** dan **Nim** dianggap berbeda.
2. Tidak boleh mengandung spasi.
3. Tidak boleh mengandung simbol-simbol khusus, kecuali garis bawah (underscore). Yang termasuk symbol khusus yang tidak diperbolehkan antara lain : \$, ?, %, #, !, &, \*, (, ), -, +, = dsb
4. Panjangnya bebas, tetapi hanya 32 karakter pertama yang terpakai.

Contoh penamaan variabel yang benar :

NIM, a, x, nama\_mhs, f3098, f4, nilai, budi.

Contoh penamaan variable yang salah :

%nilai\_mahasiswa, 80mahasiswa, rata-rata, ada spasi, penting!

### 2.2.3 Karakter Khusus (*Special Character*)

Pada Bahasa C, pada umumnya karakter atau string dapat ditampilkan dengan menuliskan karakter / string tersebut secara langsung. Namun demikian, terdapat beberapa karakter khusus yang penulisannya sedikit berbeda. Berikut ini karakter khusus yang dikenal di bahasa C beserta penjelasannya.

Karakter Khusus	Penjelasan
\a	Untuk bunyi bell (alert)
\b	Mundur satu spasi (backspace)
\f	Ganti halaman (form feed)
\n	Ganti baris baru (new line)
\r	Menuju ke kolom pertama, baris yang sama (carriage return)
\v	Tabulasi vertikal

Karakter Khusus	Penjelasan
\t	Tabulasi horizontal
\0	Nilai kosong (null)
\'	Kutip tunggal
\"	Kutip ganda
\\	Karakter garis miring (backslash)

### 2.2.4 Deklarasi

Deklarasi diperlukan bila kita akan menggunakan pengenalan (identifikasi) dalam program. Identifikasi dapat berupa variabel, konstanta dan fungsi.

#### Deklarasi Variabel

Bentuk umum pendeklarasian suatu variabel adalah :

```
Nama_tipe nama_variabel;
```

Contoh :

```
int x;
char y, huruf, nim[10];
char float nilai;
double beta;
int array[5][4];
char *p;
```

#### Deklarasi Konstanta

Dalam bahasa C konstanta dideklarasikan menggunakan preprocessor #define. Contohnya :

```
#define PHI 3.14
#define nim "0111500382"
#define nama "Achmad Solichin"
```

#### Deklarasi Fungsi

Fungsi merupakan bagian yang terpisah dari program dan dapat diaktifkan atau dipanggil di manapun di dalam program. Fungsi dalam bahasa C ada yang sudah disediakan sebagai fungsi pustaka seperti printf(), scanf(), getch() dan untuk menggunakannya tidak perlu dideklarasikan. Fungsi yang perlu dideklarasikan terlebih dahulu adalah fungsi yang dibuat oleh programmer. Bentuk umum deklarasi sebuah fungsi adalah : Tipe\_fungsi nama\_fungsi(parameter\_fungsi); Contohnya : float luas\_lingkaran(int jari); void tampil(); int tambah(int x, int y);

## 2.2.5 Operator

### Operator Penugasan

Operator Penugasan (*Assignment operator*) dalam bahasa C berupa tanda sama dengan ("="). Contoh :

```
nilai = 80;
A = x * y;
```

Artinya : variable "nilai" diisi dengan 80 dan variable "A" diisi dengan hasil perkalian antara x dan y.

### Operator Aritmatika

Bahasa C menyediakan lima operator aritmatika, yaitu :

- \* : untuk perkalian
- / : untuk pembagian
- % : untuk sisa pembagian (modulus)
- + : untuk penambahan
- - : untuk pengurangan

Catatan : operator % digunakan untuk mencari sisa pembagian antara dua bilangan. Misalnya :

```
9 % 2 = 1
9 % 3 = 0
9 % 5 = 4
9 % 6 = 3
```

### Operator Hubungan (Perbandingan)

Operator Hubungan digunakan untuk membandingkan hubungan antara dua buah operand (sebuah nilai atau variable. Operator hubungan dalam bahasa C :

Operator	Arti	Contoh	
<	Kurang dari	$x < y$	Apakah x kurang dari y
<=	Kurang dari sama dengan	$x <= y$	Apakah x kurang dari sama dengan y
>	Lebih dari	$x > y$	Apakah x lebih dari y
>=	Lebih dari sama dengan	$x >= y$	Apakah x lebih dari sama dengan y
==	Sama dengan	$x == y$	Apakah x sama dengan y
!=	Tidak sama dengan	$x != y$	Apakah x tidak sama dengan y

### Operator Logika

Jika operator hubungan membandingkan hubungan antara dua buah operand, maka operator logika digunakan untuk membandingkan logika hasil dari operator-operator hubungan. Operator logika ada tiga macam, yaitu :

- && : Logika AND (DAN)
- || : Logika OR (ATAU)
- ! : Logika NOT (INGKARAN)

### Operator Bitwise

Operator bitwise digunakan untuk memanipulasi bit-bit dari nilai data yang ada di memori. Operator bitwise dalam bahasa C :

- << : Pergeseran bit ke kiri
- >> : Pergeseran bit ke kanan
- & : Bitwise AND
- ^ : Bitwise XOR (exclusive OR)
- | : Bitwise OR
- ~ : Bitwise NOT

### Operator Unary

Operator Unary merupakan operator yang hanya membutuhkan satu operand saja. Dalam bahasa C terdapat beberapa operator unary, yaitu :

<b>Operator</b>	<b>Arti/Maksud</b>	<b>Letak</b>	<b>Contoh</b>	<b>Equivalen</b>
-	Unary minus	Sebelum operator	A + -B * C	A + (-B) * C
++	Peningkatan dengan penambahan nilai 1	Sebelum dan sesudah	A++	A = A + 1
--	Penurunan dengan pengurangan nilai 1	Sebelum dan sesudah	A--	A = A - 1
sizeof	Ukuran dari operand dalam byte	Sebelum	sizeof(I)	-
!	Unary NOT	Sebelum	!A	-
~	Bitwise NOT	Sebelum	~A	-
&	Menghasilkan alamat memori operand	Sebelum	&A	-
*	Menghasilkan nilai dari pointer	Sebelum	*A	-

### **Catatan Penting ! :**

Operator peningkatan ++ dan penurunan -- jika diletakkan sebelum atau sesudah operand terdapat perbedaan.

### **2.2.6 Kata Tercadang (Reserved Word)**

Bahasa C standar ANSI memiliki 32 kata tercadang (reserved word) dan Turbo C menambahkannya dengan 7 kata tercadang. Semua *reserved word* tidak boleh digunakan dalam penamaan identifier (variable, nama fungsi dll). Kata tercadang yang tersedia dalam bahasa C adalah sbb (tanda \* menunjukkan kata tercadang pada Turbo C):

```
*asm, default, for, *pascal, switch, auto, do, goto, register,
typedef, break, double, *huge, return, union, case, else, if,
short, unsigned, *cdecl, enum, int, signed, void, char, extern,
*interrupt, sizeof volatile const *far long static while
continue float *near struct
```

### **2.2.7 Komentar Program**

Komentar program hanya diperlukan untuk memudahkan pembacaan dan pemahaman suatu program (untuk keperluan dokumentasi program). Dengan kata lain, komentar program hanya merupakan keterangan atau penjelasan program. Untuk memberikan komentar atau penjelasan dalam bahasa C digunakan pembatas /\* dan \*/ atau menggunakan tanda // untuk komentar yang hanya terdiri dari satu baris. Komentar program tidak akan ikut diproses dalam kompilasi program (akan diabaikan).

## **2.3 PELAKSANAAN PRAKTIKUM**

1. Tuliskan Program 2.1 berikut ini pada editor Dev-C++.

**Program 2.1 variabel1.cpp**

```
1 #include <stdio.h>
2 int main()
3 {
4     char nim[10];
5     char nama[30];
6     int nilai;
7
8     printf("NIM : %s", nim);
9     printf("NAMA : %s", nama);
10    printf("NILAI : %i", nilai);
11
12    return 0;
13 }
```

2. Jalankan program 2.1 di atas dan tuliskan apa yang tercetak di layar.

3. Pada Program 2.1, variabel yang dideklarasikan belum diisi dengan nilai tertentu (belum diinisialisasi). Lakukan perubahan Program 2.1 dengan menambahkan perintah untuk mengisi variabel nim, nama dan nilai seperti pada Program 2.2 berikut ini.

**Program 2.2 variabel2.cpp**

```
1 #include <stdio.h>
2 int main()
3 {
4     char nim[10];
5     char nama[30];
6     int nilai;
7
8     nim = "141150123";
9     nama = "Achmad Solichin";
10    nilai = 85;
11
12    printf("NIM : %s", nim);
13    printf("NAMA : %s", nama);
14    printf("NILAI : %i", nilai);
15
16    return 0;
17 }
```

4. Lakukan kompilasi Program 2.2 di atas dan perhatikan hasilnya. Apakah terjadi error? Apa error yang akan ditampilkan? Tuliskan error yang ditampilkan dan baris berapa terjadi error!

Error yang ditampilkan:

Error terjadi pada baris ke:

5. Kesalahan pada Program 2.2 terjadi karena cara pengisian variabel nim dan nama yang kurang tepat. Variabel nim dan nama dideklarasikan sebagai sebuah variabel bertipe char dan berupa array (ditunjukkan dengan adanya tanda kurung siku []). Pembahasan mengenai array akan dilakukan secara khusus pada Pertemuan ke-9.
6. Sekarang kita perbaiki kesalahan pada Program 2.2 dengan mengisi nim dan nama menggunakan fungsi strcpy(). Penggunaan fungsi strcpy() harus menyertakan header fungsi <string.h>. Ubahlah Program 2.2 menjadi Program 2.3 berikut ini.

**Program 2.3 variabel3.cpp**

```
1 #include <stdio.h>
2 #include <string.h>
3 int main()
4 {
5     char nim[10];
6     char nama[30];
7     int nilai;
8
9     strcpy(nim, "141150123");
10    strcpy(nama, "Achmad Solichin");
11    nilai = 85;
12
13    printf("NIM : %s", nim);
14    printf("NAMA : %s", nama);
15    printf("NILAI : %i", nilai);
16
17    return 0;
18 }
```

7. Jalankan Program 2.3 dan tuliskan apa yang tercetak di layar!

8. Tampilan Program 2.3 sedikit berantakan bukan? Mari kita buat lebih rapi dengan menambahkan karakter khusus \n (pindah baris) dan \t (tabulasi horizontal). Perhatikan Program 2.4 berikut ini dan jalankan!

**Program 2.4 variabel4.cpp**

```
1 #include <stdio.h>
2 #include <string.h>
3 int main()
4 {
5     char nim[10];
6     char nama[30];
7     int nilai;
8
9     strcpy(nim, "141150123");
10    strcpy(nama, "Achmad Solichin");
11    nilai = 85;
12
13    printf("NIM \t: %s", nim);
14    printf("\nNAMA \t: %s", nama);
15    printf("\nNILAI \t: %i", nilai);
16
17    return 0;
18 }
```

9. Jalankan Program 2.4 dan tuliskan apa yang tercetak di layar!

10. Selanjutnya kita akan ubah Program 2.4 menjadi Program 2.5 dengan menambahkan nilai uts, uas, tugas dan kehadiran serta menghitung nilai akhir yang diperoleh mahasiswa. Nilai akhir diperoleh dengan menggunakan rumus sebagai berikut:

$$\text{NILAI AKHIR} = 10\% \text{ kehadiran} + 20\% \text{ tugas} + 30\% \text{ uts} + 40\% \text{ uas}$$

Kita akan mendeklarasikan variabel kehadiran, tugas, uts dan uas bertipe integer dan variabel nilai\_akhir bertipe float (pecahan).



**Program 2.5 variabel5.cpp**

```
1 #include <stdio.h>
2 #include <string.h>
3 int main()
4 {
5     char nim[10];
6     char nama[30];
7     int kehadiran, tugas, uts, uas;
8     float nilai_akhir;
9
10    strcpy(nim, "141150123");
11    strcpy(nama, "Achmad Solichin");
12    kehadiran = 100;
13    tugas      = 90;
14    uts        = 83;
15    uas        = 86;
16
17    nilai_akhir = (0.1 * kehadiran) + (0.2 * tugas) + (0.3
18 * uts) + (0.4 * uas);
19
20    printf("NIM \t: %s", nim);
21    printf("\nNAMA \t: %s", nama);
22    printf("\nKEHADIRAN \t: %i", kehadiran);
23    printf("\nTUGAS \t: %i", tugas);
24    printf("\nUTS \t: %i", uts);
25    printf("\nUAS \t: %i", uas);
26    printf("\nNILAI AKHIR \t: %.2f", nilai_akhir);
27
28    return 0;
29 }
```

11. Jalankan Program 2.5 dan tuliskan apa yang tercetak!

12. Untuk lebih mempermudah pembacaan program (dan untuk keperluan belajar), kita dapat menambahkan komentar-komentar program. Ubah Program 2.5 menjadi Program 2.6 berikut ini.

**Program 2.6 variabel6.cpp**

```

1  #include <stdio.h>
2  #include <string.h>
3  int main()
4  {
5      char nim[10];
6      char nama[30];
7      int kehadiran, tugas, uts, uas;
8      float nilai_akhir;
9
10     strcpy(nim, "141150123");
11     strcpy(nama, "Achmad Solichin");
12     kehadiran = 100;    //nilai kehadiran
13     tugas     = 90;    //nilai tugas
14     uts       = 83;    //nilai uts
15     uas      = 86;    //nilai uas
16
17     /* perhitungan nilai akhir
18        sesuai peraturan di UBL
19     */
20     nilai_akhir = (0.1 * kehadiran) + (0.2 * tugas) + (0.3
* uts) + (0.4 * uas);
21
22     //tampilkan data
23     printf("NIM \t: %s", nim);
24     printf("\nNAMA \t: %s", nama);
25     printf("\nKEHADIRAN \t: %i", kehadiran);
26     printf("\nTUGAS \t: %i", tugas);
27     printf("\nUTS \t: %i", uts);
28     printf("\nUAS \t: %i", uas);
29     printf("\nNILAI AKHIR \t: %.2f", nilai_akhir);
30
31     return 0;
32 }

```

13. Jalankan Program 2.6 dan lihat hasilnya. Bandingkan dengan hasil Program 2.5. Sama bukan? Hal tersebut menunjukkan bahwa komentar program tidak mempengaruhi hasil program. Komentar program diperlukan oleh programmer sendiri agar dalam mempermudah pembacaan program.

## 2.4 LATIHAN

Tuliskan dan jalankan beberapa program berikut ini dan tuliskan hasilnya di tempat yang sudah disediakan.

**Program 2.7 unary.cpp**

```
1 #include <stdio.h>
2 int main()
3 {
4     int A, B;
5     A = 5;
6     printf("A = %i", A);
7     printf("\nA = %i", A++);
8     printf("\nA = %i", A);
9
10    B = 10;
11    printf("\n\nB = %i", B);
12    printf("\nB = %i", ++B);
13    printf("\nB = %i", B);
14
15    return 0;
16 }
```

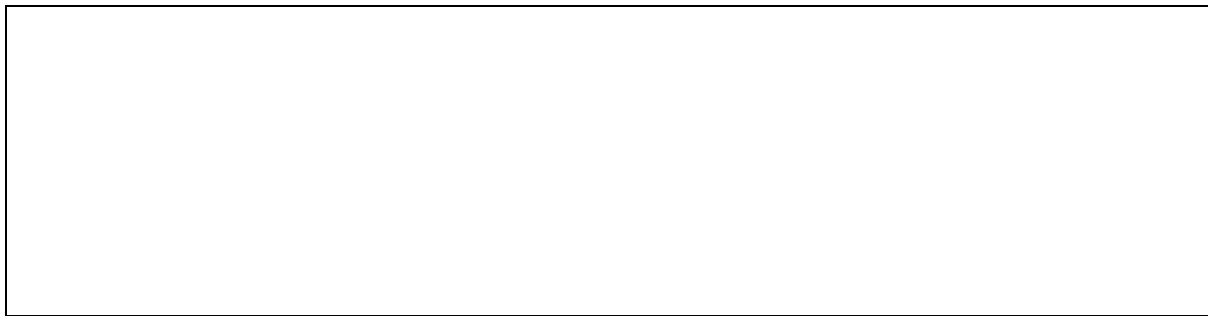
**Hasil Program 2.7**



**Program 2.8 lingkaran.cpp**

```
1 #include <stdio.h>
2 int main()
3 {
4     int jari;
5     float luas, keliling;
6     jari = 7;
7
8     luas = 3.14 * jari * jari;
9     keliling = 2 * 3.14 * jari;
10
11    printf("LUAS dan KELILING LINGKARAN");
12    printf("\nJari-jari = %i", jari);
13    printf("\nLUAS = %.3f", luas);
14    printf("\nKELILING = %.2f", keliling);
15
16    return 0;
17 }
```

### Hasil Program 2.8



## 2.5 TUGAS MANDIRI

Kerjakan soal-soal berikut ini:

1. Diketahui sebuah segitiga memiliki panjang alas = 8 cm dan tinggi = 5 cm. Buatlah sebuah program dalam Bahasa C untuk menghitung dan menampilkan luas dari segitiga tersebut!
2. Diketahui sebuah bangun ruang berbentuk bola dan memiliki diameter 15 cm. Buatlah sebuah program dalam Bahasa C untuk menghitung dan menampilkan isi (volume) dari bangun ruang bola tersebut!
3. Diketahui sebuah segitiga siku-siku dengan panjang sisi alas 4 cm dan sisi tinggi 5 cm. Dengan menggunakan rumus Phitagoras, buatlah sebuah program Bahasa C untuk menghitung sisi miring segitiga tersebut dan menampilkannya di layar!