

Pengantar Pemrograman

Ir. Fariani Hermin, MT.
Ratna Widyati, S.Si., M.Kom.



PENDAHULUAN

Di dalam modul ini akan dikaji pokok bahasan pengantar pemrograman. Butir-butir pembicaraan dipusatkan pada dua hal, yaitu:

1. Pengantar Algoritma Pemrograman
2. Pengantar Bahasa Pemrograman dalam Bahasa Pascal dan C++.

Penjelasan tentang Pengantar Algoritma Pemrograman merupakan pengetahuan dan keterampilan yang perlu dimiliki oleh *programmer* yaitu terdiri dari algoritma dan bagan alir (*flowchart*). Sedangkan Pengantar Bahasa Pemrograman merupakan pengetahuan tentang *software* atau perangkat lunak yang dirancang untuk membuat program. Dalam hal ini yang akan dibahas adalah bahasa Pascal dan C++.

Modul ini dapat dipelajari dengan beberapa cara seperti mandiri, kerja kelompok, tutorial dan sebagainya. Selain itu dalam mempelajari algoritma pemrograman dan pengantar bahasa pemrograman perlu dilengkapi dengan seperangkat komputer dan perangkat lunak (*software*) Pascal dan C++.

Setelah mempelajari modul ini, Anda diharapkan dapat menjelaskan pengertian algoritma dan pemrograman dalam bahasa Pascal dan C++. Tujuan yang masih bersifat umum ini dapat dirinci dengan beberapa tujuan yang bersifat khusus sebagai berikut.

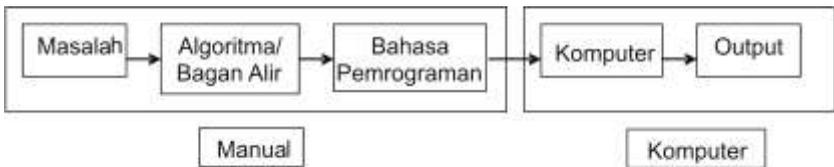
- a. mendefinisikan algoritma pemrograman;
- b. mendefinisikan bahasa pemrograman Pascal dan C++;
- c. membedakan algoritma pemrograman dan bagan alir;
- d. menjelaskan arti simbol bagan alir;
- e. membuat algoritma pemrograman dan bagan alir sesuai dengan masalah;
- f. membuat program komputer yang sesuai dengan algoritma pemrograman dalam bahasa Pascal dan C++ ; dan

- g. menjelaskan pengertian dari algoritma pemrograman dan bahasa pemrograman Pascal dan C++.

KEGIATAN BELAJAR 1

Pengantar Algoritma Pemrograman

Secara garis besar tahapan pembuatan program komputer dapat digambarkan sebagai berikut.



Gambar 1.1
Tahapan pembuatan program komputer

Masalah diuraikan secara jelas apa inti permasalahannya, input (masukan) yang diperlukan dan output (keluarannya) yang diharapkan. Setelah dipelajari proses penyelesaian masalah, kemudian dibuat algoritma/bagan alir (*flowchart*) yang berupa urutan langkah-langkah dalam penyelesaian masalah. Kemudian algoritma/bagan alir diterjemahkan dalam bahasa pemrograman, misalnya bahasa Pascal dan C++. Selanjutnya, komputer akan menyelesaikan masalah dan memperlihatkan hasil yang diperoleh dari program yang telah dibuat. Berikut ini akan dijelaskan tentang algoritma pemrograman dan bagan alir (*flowchart*).

A. ALGORITMA PEMROGRAMAN

Program komputer diciptakan baik untuk mengotomasikan suatu proses maupun untuk menyelesaikan masalah tertentu. Urutan langkah yang harus diikuti untuk memecahkan masalah tertentu disebut algoritma.

Sebuah ilustrasi akan digunakan untuk memperjelas konsep algoritma tersebut. Masalah yang akan dipecahkan adalah “mengirim surat melalui kantor pos dengan kilat khusus”. Maka langkah awal yang harus dilakukan adalah menyatakan masalahnya, menentukan input dan output.

Masalah : Mengirim surat melalui kantor pos

Input : Surat, uang

Output : Slip pengiriman surat

Algoritma:

1. Membawa surat yang telah dibuat.
2. Pergi ke kantor pos.
3. Membayar biaya pengiriman surat kilat khusus.
4. Menerima slip pengiriman surat.

Urutan dalam pembuatan langkah penyelesaian mungkin berbeda-beda. Dalam pemrograman komputer hal itu bisa saja terjadi untuk mendapatkan output yang terbaik. Karena algoritma yang diberikan menggunakan cara prosedural maka bagian data dan instruksi dipisahkan tempatnya. Pada dasarnya, penulisan algoritma selalu disusun oleh tiga bagian, yaitu bagian kepala (*header*) algoritma, bagian deklarasi dan bagian deskripsi algoritma. Setiap bagian disertai dengan komentar untuk memperjelas maksud penulisan. Komentar ditulis dalam tanda kurung kurawal ('{}'). Berikut ini merupakan penjelasan masing-masing bagian.

1. Kepala Algoritma

Kepala algoritma adalah bagian yang terdiri atas nama algoritma dan penjelasan (spesifikasi) tentang algoritma tersebut. Nama algoritma sebaiknya singkat namun cukup menggambarkan apa yang dilakukan oleh algoritma tersebut. Di bawah nama algoritma disertai dengan penjelasan singkat tentang apa yang dilakukan oleh algoritma. Penjelasan di bawah judul algoritma sering dinamakan juga spesifikasi algoritma. Algoritma harus ditulis sesuai dengan spesifikasi yang didefinisikan. Sebagai contoh untuk kepala algoritma sebagai berikut.

```
Algoritma VOLUME_KUBUS
{Menghitung volume kubus dengan ukuran sisi tertentu.
Algoritma menerima masukan sisi-sisi kubus, menghitung
volumenya dan mencetak volume kubus ke piranti
keluaran}
```

```
Algoritma Luas_Lingkaran
{Menghitung luas lingkaran dengan ukuran jari-jari
tertentu. Algoritma menerima masukan jari-jari
lingkaran, menghitung luasnya dan mencetak luas
lingkaran ke piranti keluaran}
```

Penulisan judul dapat ditulis dalam huruf besar semua, huruf kecil semua maupun campuran antara huruf besar dan kecil. Di dalam nama algoritma sebaiknya tidak ada spasi dalam memisahkan kata. Hal ini akan bermanfaat dalam penulisan judul program terutama untuk bahasa Pascal.

2. Deklarasi

Deklarasi nama di dalam algoritma adalah bagian kecil untuk mendefinisikan semua nama yang dipakai di dalam algoritma. Nama tersebut dapat berupa nama tetapan, nama peubah, nama tipe, nama prosedur dan nama fungsi. Sebagai contoh untuk deklarasi nama algoritma adalah sebagai berikut.

```
DEKLARASI
{nama tetapan}
const
    baris = 10 {jumlah baris}
const
    phi = 3.14 {nilai  $\pi$ }

{nama type}
type
    matrix=array[1..baris,1..kolom]ofreal{ukuran matriks}

{nama peubah}
a : char           { karakter yang dibaca }
o : titik          {titik dalam koordinat kartesius}
pilihan : Boolean  {keadaan hasil pilihan}

function Tambah(input a,b:integer)
{menjumlahkan dua bilangan dengan variabel a, b adalah
integer}

procedure Kurang(input a, b : integer)
{mengurangkan dua buah bilangan dengan variabel a, b
integer}
```

3. Deskripsi

Deskripsi merupakan bagian inti dari algoritma. Bagian ini berisi uraian langkah-langkah penyelesaian masalah dengan notasi misalnya *write* untuk mencetak data, *read* untuk membaca data dan sebagainya. Setiap langkah

algoritma dibaca dari atas ke bawah dan urutan penulisan menentukan urutan pelaksanaan perintah. Sebagai contoh deskripsi yaitu:

```
DESKRIPSI  
Read(r, phi)  
Luas ← phi * r * r  
Write(Luas)
```

Catatan: Lambang “*” adalah operasi kali pada bilangan, lihat Tabel 1.3 Modul 1 Kegiatan Belajar 2.

Berikut ini adalah beberapa contoh algoritma pemrograman secara lengkap.

Contoh 1.1:

Membuat algoritma untuk menghitung volume kubus.

Penyelesaian:

Masalah : Volume kubus

Input : Sisi kubus

Output : Volume kubus

Langkah penyelesaian masalah:

- a. Masukkan sisi-sisi kubus.
- b. Hitung volume kubus dengan menggunakan rumus $\text{volume} = \text{sisi} \times \text{sisi} \times \text{sisi}$.
- c. Cetak volumenya.

Algoritma:

```
Algoritma Volume_Kubus  
{menghitung volume kubus dengan sisi tertentu}  
  
DEKLARASI  
a : integer  
volume : integer  
  
DESKRIPSI  
Read(a)  
Volume ← a*a*a  
Write(Volume)
```

Contoh 1.2:

Membuat algoritma untuk menghitung luas lingkaran

Penyelesaian:

- a. Masalah : Luas Lingkaran
- b. Input : Jari-jari (r), π (ϕ)
- c. Output : Luas lingkaran

Langkah penyelesaian:

- a. Masukkan jari-jari lingkaran
- b. Hitung luas lingkaran dengan menggunakan rumus luas = $\pi \times (r)^2$
- c. Cetak luas lingkaran

Algoritma:

```

Algoritma Luas_Lingkaran
{menghitung luas lingkaran dengan jari-jari tertentu}

DEKLARASI
phi : 3.14 {nilai dari  $\pi$ }
r : integer {jari-jari lingkaran}
luas : integer

DESKRIPSI
Read(r,phi)
Luas  $\leftarrow$  phi*r*r
Write(Luas)

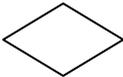
```

B. BAGAN ALIR (*FLOWCHART*)

Pada prinsipnya bagan alir (*flowchart*) tidak berbeda dengan algoritma, yaitu urutan langkah penyelesaian masalah pemrograman komputer. Perbedaan utama yaitu bagan alir dinyatakan dalam bentuk gambar atau bagan.

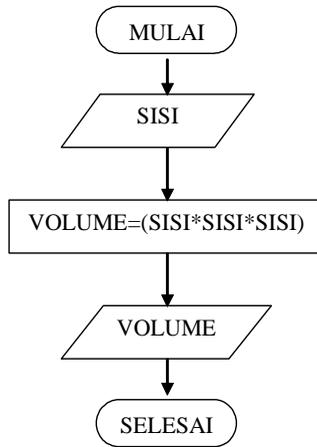
Beberapa symbol bagan alir yang biasa digunakan:

Tabel 1.1
Arti Simbol Bagan Alir

Lambang	Arti	Keterangan
	Start/End/Stop	Awal atau akhir program
	Input/Print	Menuliskan masukan/keluaran
	Proses/Pernyataan	Perintah, penugasan, rumus
	Kondisional	Memeriksa kondisi yang dipenuhi atau tidak
	Awal looping	Pernyataan awal looping
	Akhir Looping Penyambung bagan alir	Pernyataan akhir looping pada halaman yang sama
	Alur program	Jalannya aliran program
	Subrutin	Sub program
	Penyambung bagan alir	Pada halaman yang berbeda

Contoh 1.3.

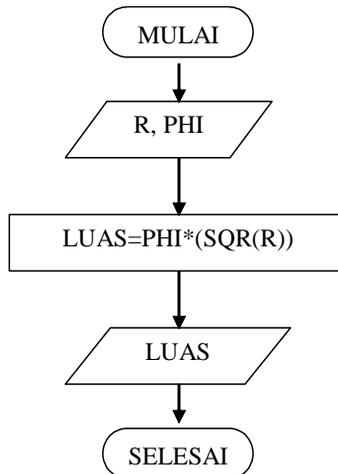
Dengan menggunakan algoritma dari contoh 1.1 maka bagan alirnya dapat digambarkan sebagai berikut.



Bagan Alir 1.1
Volume Kubus

Contoh 1.4:

Dengan menggunakan algoritma dari contoh 1.2, maka bagan alirnya adalah sebagai berikut.



Bagan Alir 1.2
Luas Lingkaran

Catatan: SQR adalah lambang kuadrat bilangan. Lihat Tabel 1.5 Modul 1 Kegiatan Belajar 2.



LATIHAN

Untuk memperdalam pemahaman Anda mengenai materi di atas, kerjakanlah latihan berikut!

- 1) Buatlah algoritma pemrograman dan bagan alir (*flowchart*) untuk menghitung keliling lingkaran dengan ketentuan sebagai berikut.
Input : Jari-jari (r), π (Phi)
Output : K (Keliling Lingkaran)
- 2) Buatlah algoritma pemrograman dan bagan alir (*flowchart*) untuk menghitung luas bujur sangkar dengan ketentuan sebagai berikut.
Input : sisi (s)
Output : L (Luas bujur sangkar)
- 3) Buatlah algoritma pemrograman dan bagan alir untuk menghitung rata-rata nilai Kalkulus I mahasiswa angkatan 2000, jika diketahui :
Input : jumlah dan nilai mahasiswa angkatan 2000 yang mengikuti mata kuliah Kalkulus I
Output : Nilai rata-rata

Petunjuk Jawaban Latihan

Sekarang periksalah baik-baik hasil kerja Anda. Kunci Jawaban berikut ini dapat Anda gunakan sebagai bahan perbandingan. Jawaban yang benar dari soal-soal di atas adalah sebagai berikut:

- 1) Menghitung Keliling Lingkaran
 - a. Algoritma Pemrograman

```

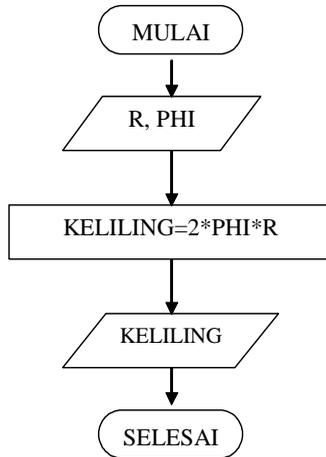
Algoritma Keliling_Lingkaran
{menghitung keliling lingkaran dengan jari-jari
tertentu}

DEKLARASI
phi : 3.14 {nilai dari  $\pi$ }
r : integer {jari-jari lingkaran}
keliling : integer

DESKRIPSI
Read(r,phi)
keliling  $\leftarrow$  2*phi*r
Write(Luas)

```

b. Bagan Alir



2) Algoritma dan bagan alir untuk menghitung luas bujur sangkar dapat mengikuti cara pada nomor 1 di atas dengan menggunakan rumus luas bujur sangkar sisi \times sisi.

a. Algoritma Pemrograman

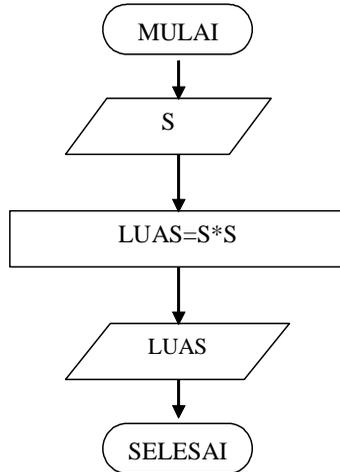
```

Algoritma Luas_Bujursangkar
{menghitung luas bujur sangkar dengan sisi-sisi tertentu}

DEKLARASI
s : integer {sisi bujur sangkar}
luas : integer

DESKRIPSI
Read(s)
luas ← s*s
Write(Luas)
  
```

b. Bagan Alir



- 3) Algoritma dan bagan alir untuk menghitung rata-rata nilai Kalkulus I dari mahasiswa angkatan 2000 adalah dengan menggunakan rumus mencari rata-rata jumlah seluruh nilai Kalkulus I yang diperoleh mahasiswa angkatan 2000 dibagi banyaknya mahasiswa yang mengambil mata kuliah Kalkulus I.

a. Algoritma Pemrograman

```

Algoritma RataRata_Nilai_Kalkulus
{menghitung rata-rata nilai kalkulus mahasiswa angkatan
2000}
  
```

DEKLARASI

```

n : integer (jumlah total mahasiswa yang mengikuti
kalkulus)
x : integer {nilai kalkulus masing-masing mahasiswa}
i : integer {jumlah pengulangan}
Jumlah : integer {jumlah seluruh nilai kalkulus
mahasiswa}
Rata : real {nilai rata-rata kalkulus yang diperoleh}
  
```

DESKRIPSI

```

Read(n)      {baca banyaknya data nilai}
i ← 1        {mulai dari data nilai yang pertama}
Jumlah ← 0
While i <= n do
  Read(x)
  
```

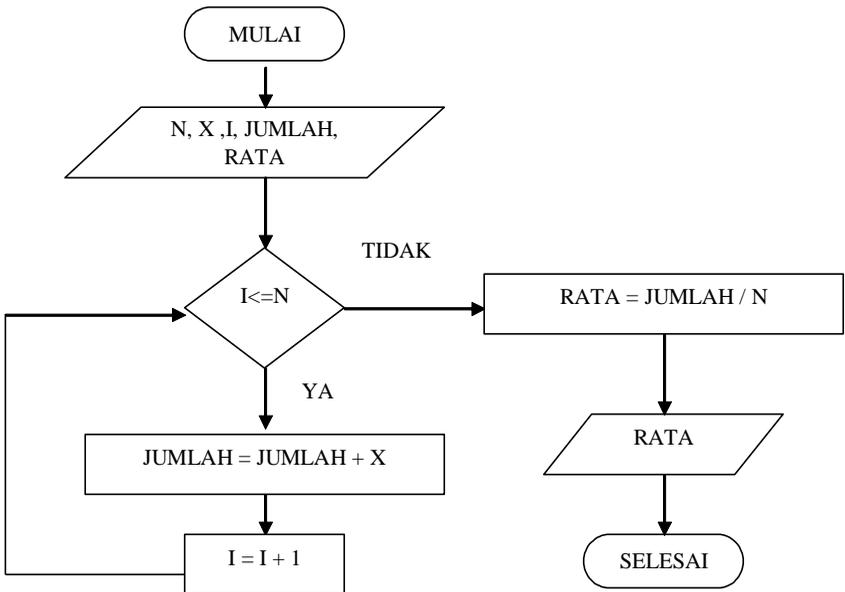
```

    Jumlah ← Jumlah+x
    i ← i+1    {pengulangan selanjutnya}
endwhile
{i > n}

Rata ← jumlah/n    {rata-rata nilai kalkulus mahasiswa}
Write(Rata)

```

b. Bagan Alir



Catatan:

1. \leftarrow adalah lambang sama dengan atau lebih kecil. (Lihat Tabel 2.1 Modul 2 Kegiatan Belajar 2).
2. $/$ adalah lambang operasi bagi pada bilangan. (Lihat Tabel 1.3 Modul 1 Kegiatan Belajar 2).



RANGKUMAN

Rangkuman berikut adalah intisari dari pembahasan Kegiatan Belajar 1.

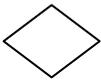
1. Algoritma adalah urutan langkah yang harus diikuti untuk memecahkan masalah tertentu.
2. Algoritma yang diberikan menggunakan cara prosedural, oleh karena itu bagian data dan instruksi dipisahkan tempatnya.
3. Pada dasarnya, penulisan algoritma selalu disusun oleh tiga bagian, yaitu bagian kepala (*header*) algoritma, bagian deklarasi dan bagian deskripsi algoritma. Setiap bagian disertai dengan komentar untuk memperjelas maksud penulisan. Komentar ditulis dalam tanda kurung kurawal ('{}').
4. Pada prinsipnya bagan alir (*flowchart*) tidak berbeda dengan algoritma, yaitu urutan langkah penyelesaian masalah pemrograman komputer. Perbedaan utama yaitu bagan alir dinyatakan dalam bentuk gambar atau bagan.



TES FORMATIF 1

Pilihlah satu jawaban yang paling tepat!

- 1) Yang dimaksud dengan algoritma adalah
 - A. kata-kata yang digunakan untuk menyelesaikan masalah
 - B. urutan langkah yang harus diikuti dalam menyelesaikan masalah
 - C. urutan data-data
 - D. cara memecahkan suatu masalah
- 2) Perbedaan utama bagan alir dengan algoritma adalah
 - A. dalam penulisan kata-kata
 - B. urutan langkahnya
 - C. dalam bentuk gambar/bagan
 - D. tidak ada perbedaan
- 3) Yang termasuk tahapan pembuatan program komputer dapat digambarkan sebagai berikut, *kecuali*
 - A. masalah
 - B. algoritma/bagan alir
 - C. bahasa pemrograman
 - D. hipotesis

- 4) Berikut ini merupakan bagian penyusunan dari algoritma, *kecuali*
- teks penulisan
 - deklarasi
 - deskripsi
 - kepala algoritma
- 5) Apa yang dimaksud dengan Deklarasi?
- Bagian yang terdiri atas nama algoritma dan penjelasan tentang algoritma tersebut.
 - Hasil pemikiran konseptual.
 - Bagian inti dari suatu algoritma.
 - Bagian untuk mendefinisikan semua nama yang dipakai dalam algoritma.
- 6) Apa yang dimaksud dengan Deskripsi?
- Bagian yang terdiri atas nama algoritma dan penjelasan tentang algoritma tersebut.
 - Hasil pemikiran konseptual.
 - Bagian inti dari suatu algoritma.
 - Bagian untuk mendefinisikan semua nama yang dipakai dalam algoritma.
- 7) Apa yang dimaksud dengan Kepala Algoritma?
- Bagian yang terdiri atas nama algoritma dan penjelasan tentang algoritma tersebut.
 - Hasil pemikiran konseptual.
 - Bagian inti dari suatu algoritma.
 - Bagian untuk mendefinisikan semua nama yang dipakai dalam algoritma.
- 8) Arti gambar flowchart ini adalah
- start
 - input
 - proses
 - kondisional
- 
- 9) Simbol di samping ini mempunyai arti
- start
 - input
 - proses
 - kondisional
- 

10) Arti gambar *flowchart* ini adalah

- A. start
- B. input
- C. proses
- D. kondisional

Cocokkanlah jawaban Anda dengan Kunci Jawaban Tes Formatif 1 yang terdapat di bagian akhir modul ini. Hitunglah jawaban yang benar. Kemudian, gunakan rumus berikut untuk mengetahui tingkat penguasaan Anda terhadap materi Kegiatan Belajar 1.

$$\text{Tingkat penguasaan} = \frac{\text{Jumlah Jawaban yang Benar}}{\text{Jumlah Soal}} \times 100\%$$

Arti tingkat penguasaan: 90 - 100% = baik sekali
80 - 89% = baik
70 - 79% = cukup
< 70% = kurang

Apabila mencapai tingkat penguasaan 80% atau lebih, Anda dapat meneruskan dengan Kegiatan Belajar 2. **Bagus!** Jika masih di bawah 80%, Anda harus mengulangi materi Kegiatan Belajar 1, terutama bagian yang belum dikuasai.

KEGIATAN BELAJAR 2**Pengantar Bahasa Pemrograman**

Seperti yang telah disebutkan dalam Kegiatan Belajar 1, bahwa agar program dapat dilaksanakan oleh komputer, maka program tersebut harus ditulis dalam suatu bahasa yang dimengerti oleh komputer. Sebagaimana dalam kehidupan, kita hanya memberikan perintah kepada orang lain dalam bahasa yang ia mengerti. Karena komputer merupakan mesin, maka program harus ditulis dalam bahasa yang khusus dibuat untuk berkomunikasi dengan komputer. Bahasa komputer yang digunakan dalam menulis program dinamakan bahasa pemrograman. Bahasa pemrograman terdiri dari bahasa pemrograman tingkat rendah (*low level language*) yaitu bahasa pemrograman yang cukup sulit dimengerti oleh penggunanya karena orientasi bahasanya yang lebih dekat ke bahasa mesin dan bahasa pemrograman tingkat tinggi (*high level language*) yaitu bahasa pemrograman dengan menulis program dalam bahasa awam (bahasa Inggris sehari-hari dengan aturan tertentu) yang kemudian oleh komputer diterjemahkan ke dalam bahasa komputer (bahasa mesin). Untuk lebih jelasnya mengenai bahasa pemrograman, maka akan dijelaskan contoh bahasa pemrograman dengan menggunakan Pascal dan C++.

A. BAHASA PEMROGRAMAN PASCAL

Pascal adalah bahasa tingkat tinggi (*high level language*) yang orientasinya pada segala tujuan yang dirancang oleh Profesor Niklaus Wirth dari Technical University di Zurich, Switzerland. Nama Pascal diambil sebagai penghargaan terhadap Blaise Pascal, ahli matematik dan filosofi terkenal abad 17 dari Perancis. Profesor Niklaus Wirth memperkenalkan kompiler bahasa Pascal pertama kali untuk komputer CDC 6000 (Control Data Corporation) yang dipublikasikan pada tahun 1971 dengan tujuan untuk membantu mengajar program komputer secara sistematis, khususnya untuk memperkenalkan pemrograman terstruktur (*structured programming*). Jadi Pascal adalah bahasa yang ditujukan untuk membuat program terstruktur.

Standar Pascal adalah bahasa Pascal yang didefinisikan oleh K. Jensen dan Niklaus Wirth. Penerapan nyata dari standar Pascal banyak yang berbeda seperti apa yang telah didefinisikan oleh K. Jensen dan Niklaus Wirth.

Standar Pascal di Eropa didefinisikan oleh ISO (*International Standards Organization*) dan di Amerika oleh kerjasama antara ANSI (*American National Standard Institute*) dengan IEEE (*Institute of Electronic Engineers*). Beberapa versi dari Pascal yang telah beredar di pasaran di antaranya yaitu UCSD Pascal (*University of California at San Diego Pascal*), MS-Pascal (Microsoft Pascal), Apple Pascal dan Turbo Pascal.

Pada kegiatan belajar ini maka yang digunakan adalah Turbo Pascal for Windows versi 1.5 yang merupakan *copyright* dari Borland Internasional Inc.

1. Mengoperasikan Turbo Pascal for Windows

Turbo Pascal for Windows (TPW) merupakan sebuah program Pascal yang dijalankan pada sistem operasi windows. Untuk dapat mengoperasikannya maka diperlukan perangkat keras dan lunak sebagai berikut.

- a. Satu unit computer yang terdiri dari CPU, Monitor, Keyboard dan Mouse.
- b. Program Turbo Pascal for Windows.

Untuk dapat bekerja dengan Pascal, terlebih dahulu kita harus menginstal program Pascal ke dalam *hardisk*. Setelah diinstal, maka program Pascal dapat dijalankan dengan cara sebagai berikut.

- a. Klik *Start* pada jendela awal windows, kemudian klik *Programs* dan klik TPW 1.5 (bila menginstal program Turbo Pascal for Windows versi 1.5)

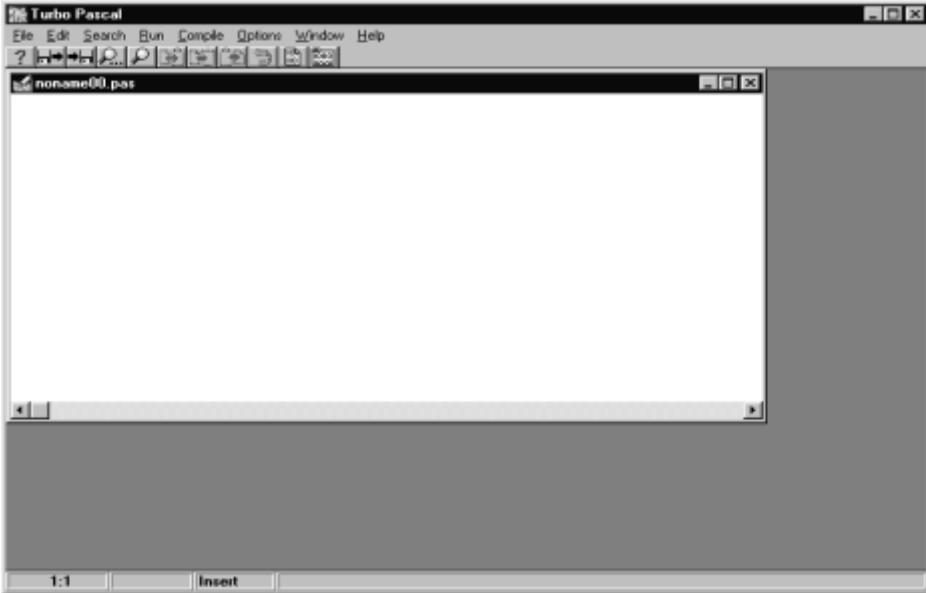


Gambar 1.2
Menu Start Programs untuk memilih TPW 1.5

- b. Setelah keluar tampilan program Pascal, maka kita dapat membuat suatu program baru dengan memilih menu File kemudian pilih New.



Gambar 1.3
Tampilan Program TPW 1.5



Gambar 1.4
Tampilan New Program TPW 1.5

- c. Setelah selesai membuat suatu program, maka kita dapat menjalankan program yang telah dibuat benar atau masih salah dengan memilih menu *Run*.
- d. Jika program masih belum benar, maka akan keluar pesan sesuai dengan kesalahan yang dibuat.
- e. Jika program telah benar maka akan keluar hasil Run-nya.
- f. Simpanlah hasil program yang telah dibuat dengan memilih menu *File, Save As*, beri nama sesuai yang diinginkan.
- g. Untuk keluar dari program Pascal ini cukup dengan memilih menu *File* kemudian *Exit* atau dengan memilih tanda  pada sudut jendela.

2. Struktur Program

Struktur lengkap program Pascal terdiri dari empat bagian, yaitu kepala program, bagian deklarasi, subprogram dan badan program. Di bawah ini merupakan contoh program PASCAL dengan struktur yang lengkap.

Program Contoh: {Program untuk menghitung luas lingkaran}

USES

 wincrt;

LABEL

 Stop;

CONST

 phi=3.14;

TYPE

Bilangan=real;

VAR

Jari-jari, luas : bilangan;

PROCEDURE LuasLingkaran;

 Begin

 Luas:=phi* jari-jari*jari-jari;

 End;

BEGIN

 Writeln("Menghitung Luas Lingkaran");

 Writeln;

 Write("Jari-jari : ");Readln(jari-jari);

 Luas Lingkaran;

 Goto Stop;

 Stop;

 Begin

 Write("Tekan sembarang tombol !");readln;

 End;

END.

Catatan:

- Program Pascal dalam penulisan huruf tidak membedakan antara huruf kapital dan huruf lainnya.
- Kalimat di dalam tanda kurung {...} digunakan sebagai komentar atau hanya untuk menjelaskan sesuatu, tidak diproses dalam program.
- Setiap satu perintah diakhiri tanda titik koma (;).
- Pernyataan yang lebih dari satu perintah harus diawali dengan BEGIN dan diakhiri dengan END.

- Penjelasan setiap perintah akan diuraikan pada bagian berikutnya.

a. *Kepala program*

Kepala Program berisi nama dari judul program. Bentuk umumnya adalah sebagai berikut.

```
PROGRAM nama[(input,output)];
```

Keterangan:

Nama : berisi nama program yang singkat, mengikuti aturan penamaan (tidak menggunakan tanda baca atau spasi).

Input, output : bersifat opsional (pilihan), bila dituliskan input saja berarti operasi yang dilakukan hanya bersifat masukan. Jika output saja berarti operasi yang dilakukan hanya bersifat keluaran. Jika keduanya yaitu input dan output maka operasi yang dilakukan dapat bersifat masukan atau keluaran. Hal ini biasanya digunakan bila berkaitan dengan pengelolaan file.

b. *Bagian deklarasi*

Dalam bahasa pemrograman Pascal semua variabel, jenis variabel, prosedur dan sebagainya akan digunakan dalam badan program yang harus dinyatakan terlebih dahulu. Bagian deklarasi dituliskan secara terurut dan dikelompokkan sebagai berikut.

- 1) USES
- 2) LABEL
- 3) CONST
- 4) TYPE
- 5) VAR

Penjelasan yang lebih rinci akan dijelaskan pada bagian Deklarasi (halaman 1.23 sampai 1.26).

c. *Subprogram*

Pascal adalah bahasa pemrograman terstruktur, yaitu programnya terdiri dari program utama dan program-program bagian. Program bagian inilah yang disebut subprogram. Di dalam Pascal, subprogram dituliskan sebelum program utama dan dituliskan dengan:

- 1) FUNCTION, yaitu subprogram yang berupa fungsi.
- 2) PROCEDURE, yaitu subprogram yang berupa program kecil.

Untuk penjelasan yang lebih rinci tentang *function* dan *procedure*, lihat uraian pada Modul 4.

d. Badan Program

Badan program berisi program utama di mana program Pascal tidak akan menghasilkan apa-apa tanpa badan dan kepala program. Sedangkan bagian deklarasi dan program bagian (subprogram) adalah opsional (pilihan), yaitu digunakan bila diperlukan saja. Badan Program diawali dengan BEGIN dan diakhiri dengan END disertai tanda titik (.). Di dalam badan program dapat digunakan BEGIN dan END yang lain, tetapi END yang bukan akhir dari badan program diakhiri dengan tanda titik koma (;).

3. Deklarasi Program

Di bawah ini merupakan pernyataan yang terdapat dalam bagian deklarasi.

a. USES

Uses merupakan suatu perintah yang digunakan untuk memakai suatu unit tertentu dari Turbo Pascal. Unit pada Turbo Pascal adalah kumpulan-kumpulan instruksi yang merupakan prosedur dan fungsi yang diciptakan dalam Turbo Pascal. Bentuk umumnya adalah sebagai berikut.

USES daftar kata tercadang;

Keterangan:

Daftar kata tercadang: berisi kata-kata tercadang (*reserval word*) yang tersedia dalam program unit seperti wincrt.

b. LABEL

Label adalah pendeklarasian kata-kata atau tanda-tanda yang digunakan untuk menyatakan suatu tempat pada program. Label dapat berisi apa saja, kecuali tanda-tanda operasi aritmatik, seperti perkalian, penambahan, pengurangan, pembagian, dan lain-lain. Bentuk Umum dari label adalah sebagai berikut.

LABEL *identifier*;

Keterangan:

Identifier: berupa bilangan asli yang kurang dari 999 atau berupa teks nama yang singkat.

c. *CONST*

Perintah Const adalah perintah untuk mendefinisikan suatu konstanta. Tata cara penulisan konstanta adalah sebagai berikut.

- 1) Sebelum mendefinisikan konstanta, Anda harus memberikan kata Const yang menyatakan awal pendefinisian konstanta.
- 2) Konstanta ditulis dalam tata penulisan :
Konstanta = suatu besaran
- 3) Konstanta tidak hanya berlaku untuk bilangan tetapi bisa juga berupa suatu string.
- 4) Dalam konstanta boleh terdapat operasi aritmatik seperti perkalian, penambahan, pengurangan, pembagian, dan lain-lain.
- 5) Nama konstanta tidak boleh mengandung spasi dan tanda aritmatik, serta tidak boleh dimulai dengan bilangan.

Contoh:

Const

Jumlah = 70;

Total = 'Tujuh Puluh';

Hasil = 70*4;

d. *TYPE*

Type adalah perintah yang berguna untuk mendefinisikan suatu tipe yang akan dimasukkan ke dalam suatu variabel.

Contoh:

Type

Bilangan = real;

e. *VAR*

Kata *var* merupakan tanda dimulainya pendefinisian variabel-variabel yang akan digunakan pada program. Tata cara penulisan suatu variabel adalah:

- 1) Pendefinisian variabel dimulai dengan kata VAR
- 2) Untuk mendefinisikan suatu variabel, Anda harus memperhatikan tata cara penulisan pendefinisian variabel yaitu:
nama_var1, nama_var2, ...: tipe_variabel;
- 3) Dari tata penulisan di atas, Anda dapat mendefinisikan beberapa variabel sekaligus bila tipenya sama.

- 4) Nama_variabel tidak boleh mengandung spasi dan tanda aritmatik serta tidak boleh dimulai dengan bilangan.
- 5) Nama variabel tidak boleh sama dengan nama perintah, directive ataupun operator yang telah ditentukan oleh Turbo Pascal.

Berikut ini adalah beberapa contoh pendefinisian variabel yang benar.

Var

Jari-jari, luas: bilangan;
 i: integer;
 S: String;

Sedangkan contoh penulisan variabel yang salah adalah sebagai berikut.

Var

I J : integer; (mengandung spasi/kurang koma)
 Str : String; (mengandung nama perintah)
 A*B : Real; (mengandung tanda aritmatik)

Adapun tipe-tipe variabel yang telah disediakan oleh Turbo Pascal adalah sebagai berikut.

Tabel 1.2
Tipe-Tipe Variabel dalam Pascal

No.	Tipe Variabel	Keterangan
1.	Integer	Bilangan bulat yang mempunyai batas bilangan antara -32768 hingga 32767
2.	Shortint	Bilangan bulat yang mempunyai batas bilangan antara -128 hingga 127
3.	Longint	Bilangan bulat yang mempunyai batas bilangan antara -2147483648 hingga 2147483649
4.	Word	Bilangan Bilangan bulat yang mempunyai batas bilangan antara 0 hingga 65535
5.	Byte	Bilangan bulat yang mempunyai batas bilangan antara 0 hingga 255
6.	Real	Bilangan real yang mempunyai batas bilangan antara $\pm 2.9 \cdot 10^{-39}$ hingga $\pm 1.7 \cdot 10^{38}$
7.	Single	Bilangan real yang mempunyai batas bilangan antara $\pm 1.5 \cdot 10^{-45}$ hingga $\pm 1.7 \cdot 10^{45}$
8.	Double	Bilangan real yang mempunyai batas bilangan antara $\pm 5.0 \cdot 10^{-324}$ hingga $\pm 1.7 \cdot 10^{308}$

No.	Tipe Variabel	Keterangan
9.	Extended	Bilangan real yang mempunyai batas bilangan antara $\pm 5.0 \cdot 10^{-4951}$ hingga $\pm 1.1 \cdot 10^{4932}$
10.	Boolean	Type variabel yang hanya mempunyai dua kemungkinan, yaitu True atau False
11.	Char	Type variabel yang hanya dapat menampung satu karakter saja
12.	String	Type variabel yang dapat menampung serangkaian huruf-huruf karakter (n buah karakter)
13	Text	Karakter dari string

4. Masukan dan Keluaran (Input dan Output)

Masukan atau input adalah informasi yang dibutuhkan oleh komputer. Sumbernya berasal dari pengguna komputer melalui keyboard, mouse dan lain-lain. Sedangkan keluaran atau output adalah informasi hasil pengolahan komputer berdasarkan masukan (input) data.

a. Masukan (Input)

Dalam Pascal, perintah untuk masukan data adalah READ dan READLN. Perbedaan kedua pernyataan ini terletak pada sifatnya dalam menangani kelebihan data.

REAL : membaca sesuai dengan ukuran variabelnya yaitu membaca satu atau lebih variabel berupa integer, char, real atau string dari masukan standar.

READLN : membaca sampai dengan akhir baris, yaitu membaca nol, atau lebih variabel berupa integer, char, real atau string dari masukan standar.

b. Keluaran (output)

Perintah untuk menampilkan data ke monitor atau layar adalah dengan menggunakan WRITE dan WRITELN. Perbedaan kedua pernyataan ini adalah sebagai berikut.

WRITE : setelah mencetak atau menampilkan data yang diberikan, kursor akan diletakkan di samping hasil yang ditampilkan. Hal ini menyebabkan bila ingin menampilkan data lagi maka akan terletak di samping hasil tampilan data yang pertama.

WRITELN: setelah mencetak atau menampilkan data yang diberikan, kursor akan dipindahkan ke bawah satu baris dengan posisi horizontal pada awal baris. Jadi bila kita melakukan proses

cetak setelah perintah `Writeln`, kata-kata yang dicetak atau ditampilkan akan terdapat di bawah kata-kata yang dicetak atau ditampilkan sebelumnya.

Untuk memformat suatu data pada layar, maka Anda dapat menambahkan tanda titik dua (`:`) diikuti dengan bilangan yang menyatakan lebar ruang untuk menampilkan data.

Data : n

Format di atas menyebabkan *data* ditampilkan rata kanan pada ruang dengan ukuran sebesar *n* karakter. Contoh:

```
Writeln('12345678901234567890');
Writeln('Jakarta ' : 10);
Writeln('Benar' : 7);
Writeln('1234.5678' : 7);
```

Bila dieksekusi maka hasilnya adalah:

```
12345678901234567890
      Jakarta
      Benar
1.2E+03
```

Khusus untuk bilangan real, maka format berikut dapat digunakan:

Data : lebar_total : lebar_pecahan

Dalam hal ini *lebar_total* dipakai mengatur lebar ruang untuk data dan *lebar_pecahan* dipakai untuk menentukan jumlah digit bagi nilai pecahannya. Contoh :

```
Writeln('12345678901234567890');
Writeln('1234.5678 ' : 12:5);
Writeln('1234.5678' : 13:5);
Writeln('1234.5678' : 13:4);
Writeln('1234.5678' : 13:0);           {Tanpa bagian pecahan}
```

Bila dieksekusi maka hasilnya adalah:

```
12345678901234567890
1234.56780
1234.56780
1234.5678
1235
```

Sebagai catatan untuk mencetak suatu karakter atau kata-kata di suatu tempat tertentu, maka di dalam bahasa pemrograman Pascal digunakan perintah gotoxy dengan tata penulisan sebagai berikut :

Gotoxy(X,Y)

dimana X adalah posisi horisontal yang dapat diisi dengan angka dari 1 hingga 80, sedangkan Y adalah posisi vertikal yang dapat diisi dengan angka dari 1 hingga 25. Sebagai contoh :

```
gotoxy(15,2); writeln('Bahasa Pemrograman');
gotoxy(17,5); writeln('Turbo Pascal');
```

Hasil potongan program di atas adalah pada lokasi (15,2) akan tercetak kalimat dari writeln yang pertama yaitu Bahasa Pemrograman, sedangkan pada lokasi (17,5) akan tercetak kalimat dari writeln kedua yaitu Turbo Pascal.

5. Ekspresi Matematika

Operasi aljabar dan operasi logika yang digunakan dalam Pascal adalah sebagai berikut.

Tabel 1.3
Operator Aljabar

Operator Aljabar	Keterangan
+	Jumlah
-	Kurang
*	Kali
/	Bagi
DIV	Hasil Bagi (dibulatkan ke bawah ke bilangan terdekat)
MOD	Sisa Hasil Bagi (sisa dari hasil proses pembagian)

Tabel 1.4
Operator Logika

Operator Logika	Keterangan
AND	Dan
OR	Atau
XOR	Atau Eksklusif
NOT	Komplemen
MOD	Sisa Hasil Bagi

6. Fungsi

Pascal menyediakan beberapa fungsi standar yang agak berbeda dengan FUNCTION. Anda dapat membuat suatu fungsi yang dibutuhkan tetapi tidak disediakan oleh Pascal dengan membuat sendiri fungsi tadi dalam subprogram yang dinamakan dengan FUNCTION. Adapun beberapa fungsi standar yang telah disediakan dalam Pascal adalah sebagai berikut.

Tabel 1.5
Fungsi dalam Pascal

Fungsi	Keterangan	Type Variabel
INC	Penjumlahan dengan 1	Integer
DEC	Pengurangan dengan 1	Integer
TRUNC	Pembulatan bilangan ke bawah	Integer
ROUND	Pembulatan bilangan ke atas	Integer
INT	Fungsi yang berguna untuk mengambil bilangan bulat dari suatu bilangan pecahan	Real
FRAC	Fungsi yang berguna untuk mengambil bilangan di belakang koma dari suatu bilangan pecahan	Real
ABS	Harga mutlak	Real, Integer
Phi	Konstanta phi (3.14)	Real
SQR	Pangkat dua	Real
SQRT	Akar pangkat dua	Real
LN	Fungsi logaritma	Real
EXP	Fungsi eksponensial	Real
SIN	Fungsi sinus	Real
COS	Fungsi Cosinus	Real
ARCTAN	Fungsi Arcangen	Real

7. File pada Pascal

Turbo Pascal menyediakan tiga macam file yaitu:

- a. File bertipe
- b. File teks
- c. File tak bertipe

Penjelasan masing-masing file adalah sebagai berikut.

a. File Bertipe

Merupakan jenis file yang mengharuskan Anda menyebutkan tipe komponen file sebelum mengaksesnya. Tipe data yang digunakan dapat berupa integer, char, string atau bahkan record. Komponen adalah istilah

untuk elemen dari file bertipe. Langkah-langkah dalam operasi file bertipe adalah sebagai berikut.

1) Mengaitkan variabel file dengan nama file pada disk

Langkah ini akan menentukan file mana yang akan diproses. Pada pemrosesan file (termasuk file bertipe), Anda akan selalu mengacu suatu file melalui variabel file. Itulah sebabnya mengapa pertama-tama yang harus Anda lakukan adalah mengaitkan variabel file dengan nama file. Anda perlu memanggil **Assign** untuk keperluan ini. **Assign** adalah prosedur yang bentuk deklarasinya adalah sebagai berikut.

```
Procedure Assign(Var f, NamaFile : String);
```

Dengan f adalah variabel file dan NamaFile adalah nama file fisik pada disk.

2) Membuka dan Menutup file

Suatu file harus dibuka terlebih dahulu sebelum dapat diakses. Hal ini dapat diibaratkan bila Anda bermaksud membaca buku, Anda pasti membuka buku itu terlebih dahulu dan setelah membaca atau menulis buku itu, maka Anda akan menutup buku itu kembali. Begitu pula pada file, Anda perlu menutup file setelah selesai memproses file. Ada dua hal yang perlu diperhatikan dalam membuka file bertipe:

- a) Bila file belum ada, Anda perlu menciptakannya dan sekaligus membukanya.
- b) Bila file sudah ada, Anda cukup membukanya saja.

Menciptakan file dapat dilakukan dengan menggunakan prosedur **Rewrite**. Bentuk deklarasi prosedur ini adalah:

```
Procedure Rewrite(Var f);
```

Bila memanggil **Rewrite**, ada dua kemungkinan yang terjadi:

- a) Bila file sudah ada, isi file akan hilang. Oleh karena itu berhati-hatilah menggunakan Rewrite karena siapa tahu Anda tidak bermaksud membuang data.
- b) Bila file belum ada, file akan diciptakan dan sesudah itu file sudah dalam keadaan terbuka.

Bila Anda ingin membuka file yang sudah ada dan barangkali telah ada isinya di mana Anda tidak ingin menghapus isi file, maka Anda perlu memakai prosedur **Reset**, bukannya **Rewrite**. Namun harus menjamin file yang akan dibuka memang sudah ada isinya. Bentuk deklarasi Reset adalah sebagai berikut.

```
Reset(Var f);
```

Untuk menutup suatu file, cukup dengan memanggil prosedur **Close**. Bentuk deklarasi prosedur ini adalah:

```
Close(Var f);
```

Contohnya :

```
Assign(FileInt, 'BIL.DAT');
```

```
Rewrite(FileInt);           {Menciptakan dan membuka file}
```

```
{-----Proses terhadap file -----}
```

```
Close(FilInt);           {Menutup file}
```

3) Merekam Data

Untuk merekam data maka perlu digunakan prosedur **Write**. Bentuk **Write** adalah:

```
Write(f, Var1[,Var2,...,Varn]);
```

Dengan f adalah variabel file bertipe dan Var1,...,Varn adalah variabel bertipe sesuai dengan komponen file bertipe.

4) Membaca file

Data yang telah Anda rekam dapat dibaca dengan menggunakan **Read**. Bentuk prosedur ini adalah:

```
Procedure Read(f, V1, [,V2,...,Vn]);
```

Dengan f adalah variabel file dan V1,...,V2 adalah variabel yang sesuai dengan tipe komponen file.

5) Memeriksa akhir file

Akhir file dijumpai setelah data yang terakhir. Bila Anda menjumpai akhir file maka harus segera menghentikan pembacaan (Kalau tidak, akan terjadi kesalahan yang menyebabkan program terhenti). Untuk mengetahui

akhir file sudah dijumpai atau belum, dapat menggunakan fungsi bernama **Eof**. Bentuk deklarasi fungsi ini:

```
Function Eof(Var f) : Boolean;
```

Dengan *f* adalah variabel file. Hasil fungsi ini adalah True, jika posisi file berada pada akhir file dan False jika posisi file tidak berada pada akhir file.

Turbo Pascal menyediakan prosedur bernama **Seek**, yang berguna untuk memindahkan posisi file ke suatu posisi pada file. Deklarasinya berbentuk:

```
Seek(Var f, N:LongInt);
```

Dengan *f* adalah variabel file dan *N* adalah komponen *N*. Dalam hal ini komponen pertama dari file diberi nilai 0.

Salah satu manfaat **Seek** adalah untuk melakukan penambahan komponen terhadap suatu file yang sudah berisi. Caranya adalah dengan meletakkan posisi file ke akhir file. Dalam hal ini file perlu dibuka dengan Reset (bukan Rewrite). Perintahnya adalah:

```
Seek(FileInt, FileSize(FileInt));
```

Perintah di atas menempatkan posisi file ke komponen yang sama dengan jumlah komponen pada file. Karena perintah ini maka posisi file akan berada pada akhir file. Selanjutnya kalau terdapat perintah Write ke file, data akan menempati komponen terakhir.

b. File Teks

Sebagai contoh dari file teks adalah file yang digunakan untuk menyimpan program. File berekstensi PAS yang sering dibuat adalah contoh file teks. Semua file yang Anda buat dengan editor teks, seperti NotePad, WordPad termasuk editor Turbo Pascal sendiri adalah file teks. Ciri khas dari file teks adalah karakter yang menyebabkan adanya baris baru. Itulah sebabnya elemen pada file teks adalah baris.

c. File Tak Bertipe

File tak bertipe merupakan cara mengakses file tanpa menyebutkan tipe file. Anda dapat mengakses file apa saja melalui file tak bertipe yaitu:

- 1) Menyalin sebarang file.
- 2) Mengganti nama file.

- 3) Menghapus nama file.
- 4) Memeriksa keberadaan suatu file.
- 5) Menampilkan isi sebarang file.
- 6) Mengganti data yang ada pada sebarang file dengan basis karakter.
- 7) Mencari kata-kata tertentu pada sebarang file.

B. BAHASA PEMROGRAMAN C++

1. Sejarah C++

Bahasa Pemrograman C++ merupakan pengembangan dari bahasa pemrograman C yang diciptakan satu dekade setelah C, diciptakan oleh Bjarne Stroustrup. Tanda ++ berasal dari nama operator penaikan pada bahasa C.

Keistimewaan yang sangat berarti pada C++ adalah karena bahasa ini mendukung pemrograman yang berorientasi objek (PBO atau OOP/*Object Oriented Programming*). Tujuan utama pembuatan C++ adalah untuk meningkatkan produktivitas pemrogram dalam membuat aplikasi. Pemrograman Berorientasi Objek (PBO) dan C++ dapat mengurangi kekompleksitasan, terutama pada program yang besar yang terdiri dari 10.000 baris atau lebih.

2. Pemrograman Berorientasi Objek (PBO)

Pada bahasa yang berorientasi objek, data yang melekat dalam suatu objek biasa disebut variabel instan. Istilah ini pada bahasa C++ disebut dengan data atau anggota data. Adapun fungsi yang melekat pada suatu objek disebut fungsi anggota (*member function*). Fungsi ini merupakan satu-satunya cara untuk mengakses terhadap anggota data dari objek. Untuk membaca anggota data harus memanggil fungsi anggota, dengan kata lain data bersifat tersembunyi bagi fungsi-fungsi di luar fungsi anggota. Istilah yang umum untuk fungsi anggota pada bahasa pemrograman berorientasi objek adalah metode. Adapun pemanggilan fungsi anggota sering disebut pengiriman pesan ke objek. Namun istilah yang terakhir ini tidak biasa digunakan pada C++.

Perbedaan Istilah pada PBO dan C++ adalah:

- a. pada PBO disebut variabel instans, sedangkan pada C++ disebut data (anggota data).
- b. pada PBO disebut metoda, sedangkan pada C++ disebut fungsi anggota.

3. Objek dan Kelas

Dalam terminologi PBO, objek sebenarnya adalah anggota dari kelas. Dengan kata lain, kelas adalah kumpulan objek yang sama. Sebagai contoh, misalnya kumpulan penyanyi dangdut seperti: Inul Daratista, Iis Dahlia, Ikke Nurjanah, Cici Paramida, Rita Sugiarto, mereka secara sendiri-sendiri dapat diibaratkan sebagai anggota kelas, adapun "penyanyi dangdut" menyatakan kelas.

4. Karakteristik PBO

Ada 3 karakteristik utama dari bahasa yang berorientasi objek, yaitu:

a. *Encapsulation*

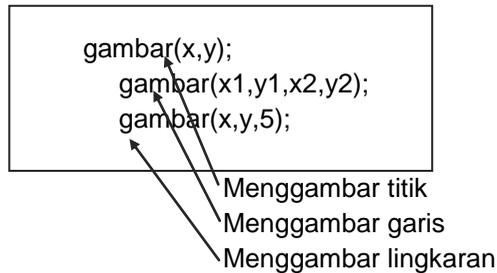
Encapsulation (pengkapsulan) adalah pengemasan data dan fungsi dalam wadah bernama objek. Data tidak lagi diperlakukan sebagai komponen kedua setelah fungsi, melainkan mempunyai kedudukan yang sama dengan fungsi.

b. *Inheritance*

Inheritance (pewarisan) adalah sifat dalam bahasa berorientasi objek yang memungkinkan sifat-sifat dari suatu kelas diturunkan ke kelas yang lain.

c. *Polymorphism*

Polymorphism (polimorfisme) adalah suatu konsep yang menyatakan sesuatu yang sama dapat memiliki berbagai bentuk dan perilaku yang berbeda. Salah satu jenis polimorfisme pada C++ dapat dikenakan pada fungsi atau operator yang dikenal dengan istilah *overloading*. Sebagai contoh, Anda dapat membuat suatu fungsi bernama gambar. Kalau Anda menyertakan posisi x dan y maka yang digambar adalah titik, sedangkan kalau yang disertakan adalah dua pasang x dan y maka hasilnya berupa garis. Adapun kalau yang disertakan adalah posisi x dan y serta jari-jari, yang digambar adalah lingkaran.



5. Program C++

Program C++ dapat ditulis menggunakan sebarang editor teks, seperti EDIT (milik DOS), Wordstar, SideKick ataupun menggunakan editor bawaan dari kompilernya.

Contoh sebuah program C++ sebagai berikut.

```
#include <iostream.h>
void main ()
{
    cout << "Selamat menggunakan program C++. \n";
}
```

Penjelasan contoh program di atas adalah:

#include <iostream.h> : header file untuk input dan output stream

void: menyatakan bahwa fungsi main tidak memiliki nilai balik

main: fungsi yang akan dieksekusi pertama kali oleh C++

() : menyatakan bahwa fungsi main tidak memiliki argumen

{ : tanda yang mengawali program C++

} : tanda yang mengakhiri program C++

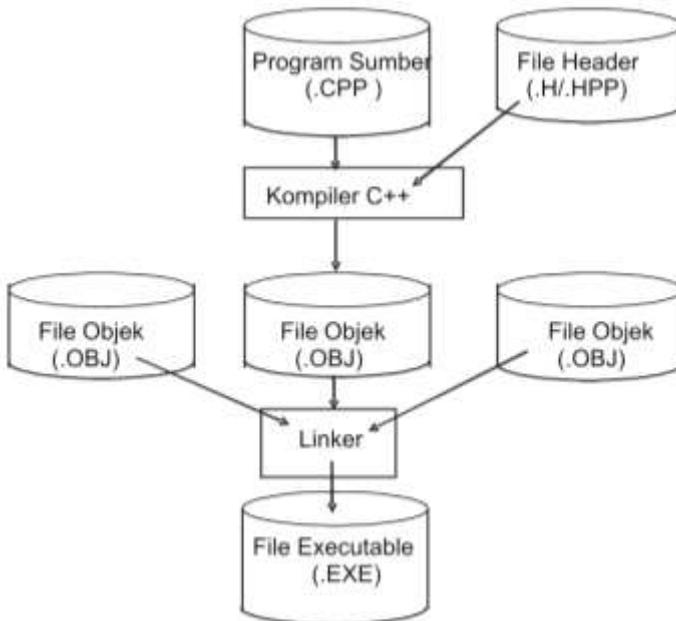
cout << "Selamat menggunakan program C++. \n"; : sebuah contoh pernyataan yang akan menampilkan tulisan yang terletak pada sepasang tanda petik ganda ke layar.

Program C++ biasa ditulis dengan nama ekstensi .CPP (dari kata C plus plus). Agar program ini bisa dijalankan (dieksekusi), program harus dikompilasi terlebih dulu dengan menggunakan kompilernya. Pada saat pengkompilasian, program sumber (.CPP) bersama file-file header (berekstensi .H atau .HPP) akan diterjemahkan oleh kompilernya menjadi

kode objek (.OBJ). File objek ini berupa file dalam format biner (berkode 0 dan 1).

Selanjutnya file objek bersama-sama dengan file objek lain serta file pustaka (.LIB) dikaitkan menjadi satu oleh linker. Hasilnya berupa file yang bersifat *executable*. File ini yang bisa dijalankan dari sistem operasi secara langsung.

Proses pembentukan file *executable* seperti gambar berikut.



Gambar 1.5
Proses pembentukan file executable

6. Kompiler C++

Saat ini banyak kompiler C++ yang beredar di pasaran. Sebagai contoh Microsoft Corporation, sebagai perusahaan pembuat perangkat lunak yang sangat termashur, mengeluarkan produk kompiler C++ berupa Microsoft C/ C++ serta visual C++. Adapun Borland International yang sangat populer dengan Turbo Pascal mengeluarkan Turbo C++ dan Borland C++. Kedua kompiler terakhir inilah yang mendasari contoh-contoh program dalam modul ini.

Pengenalan Program C++

- a. Fungsi main ()
Program C++ tidak pernah lepas dari fungsi, sebab fungsi adalah salah satu dasar penyusun blok pada C++. Sebuah program C++ minimal mengandung sebuah fungsi, yaitu fungsi main (). Fungsi ini menjadi awal dan akhir eksekusi program C++.
- b. Pernyataan
Baris:
cout << “Selamat menggunakan program C++.\n”; merupakan contoh sebuah pernyataan. Pada contoh di atas, pernyataan digunakan untuk menampilkan tulisan yang terletak pada sepasang tanda petik ganda yang disebut konstanta string.
- c. cout
Pengenal cout merupakan sebuah objek di dalam C++. Objek ini disediakan untuk mengarahkan data ke standard output (normalnya adalah layar). Misalnya:
cout << “Selamat menggunakan program C++\n”
- d. #include <iostream.h>
Baris:
#include <iostream.h>
bukanlah suatu pernyataan. Oleh sebab itu tidak ada tanda titik koma yang diperlukan. Baris tersebut menginstruksikan kepada kompiler untuk menyisipkan file lain (pada contoh di atas yaitu iostream.h) saat program dikompilasi. Dalam hal ini file-file yang berakhiran dengan .h disebut file header, yaitu file-file yang berisi berbagai deklarasi, seperti fungsi, variabel dan sebagainya.
- e. clrscr () untuk menghapus layar
Perintah untuk menghapus layar biasa dilibatkan dalam program. Pernyataan yang diperlukan untuk keperluan ini adalah: clrscr ();

7. Tipe Data

Tipe data dasar pada C++ meliputi: char, int, short, long, float, double dan long double. Tipe data yang berhubungan dengan bilangan bulat adalah char, int, short dan long. Sedangkan lainnya berhubungan dengan bilangan pecahan. Ukuran memori yang diperlukan untuk masing-masing tipe data sangat bergantung pada perangkat keras dari komputer yang digunakan.

Karena itu jangkauan bilangan dari masing-masing tipe data juga dapat berlainan.

Tabel 1.6
Ukuran Berbagai Tipe Dasar pada C++

Tipe Data	Ukuran Memori	Jangkauan Nilai	Jumlah Digit Presisi
Char	1 byte	-128 hingga +127	-
Int	2 byte	-32768 hingga +32767	-
Long	4 byte	-2.147.438.648 hingga 2.147.438.647	-
Float	4 byte	3.4×10^{-38} hingga $3.4 \times 10^{+38}$	6.7
Double	8 byte	1.7×10^{-308} hingga $1.7 \times 10^{+308}$ hingga	15 - 16
Long double	10 byte	3.4×10^{-4932} hingga $1.1 \times 10^{+4932}$ hingga	19

8. Manipulator

Manipulator pada umumnya digunakan untuk mengatur tampilan data. Misalnya untuk mengatur agar suatu nilai ditampilkan dengan lebar 10 karakter dan diatur rata kanan terhadap lebar tersebut.

Tabel 1.7
Manipulator pada C++

Manipulator	Keterangan
endl	Menyisipkan <i>newline</i> dan mengirimkan isi penyangga keluaran ke piranti keluaran
ends	Menyisipkan karakter null
flush	Mengirimkan isi penyangga keluaran ke piranti keluaran
dec	Mengkonversi ke bilangan basis sepuluh
hex	Mengkonversi ke bilangan basis 16 (heksademisal)
oct	Mengkonversi ke bilangan basis 8 (oktal)
setbase(int n)	Mengkonversi ke bilangan basis n ($n = 8, 10$ atau 16)
setw(int n)	Mengatur lebar field untuk suatu nilai sebesar n karakter
setfill(int c)	Menyetel karakter pemenuh berupa c
setprecision(int n)	Menyetel presisi bilangan pecahan sebesar n digit
setiosflags(long f)	Menyetel format yang ditentukan oleh f
resetiosflag(long f)	Menghapus format yang ditentukan oleh f .

Manipulator **setiosflags()** merupakan manipulator yang dapat dipakai untuk mengontrol sejumlah tanda format yang tercantum dalam tabel berikut.

Tabel 1.8
Tanda format untuk **setiosflags()** dan **resetiosflags()**

Nama Tanda Format	Keterangan
<code>ios::left</code>	Menyetel rata kiri terhadap lebar field yang diatur melalui setw()
<code>ios::right</code>	Menyetel rata kanan terhadap lebar field yang diatur melalui setw()
<code>ios::scientific</code>	Memformat keluaran dalam notasi eksponensial
<code>ios::fixed</code>	Memformat keluaran dalam bentuk notasi desimal
<code>ios::dec</code>	Memformat keluaran dalam basis 10 (desimal)
<code>ios::oct</code>	Memformat keluaran dalam basis 8 (oktal)
<code>ios::hex</code>	Memformat keluaran dalam basis 16 (heksadesimal)
<code>ios::uppercase</code>	Memformat huruf pada notasi heksadesimal dalam bentuk huruf kapital
<code>ios::showbase</code>	Menampilkan awalan 0x untuk bilangan heksadesimal atau 0 (nol) untuk bilangan oktal
<code>ios::showpoint</code>	Menampilkan titik desimal pada bilangan pecahan yang tidak memiliki bagian pecahan
<code>ios::showpos</code>	Untuk menampilkan tanda + pada bilangan positif

9. Fungsi **getch()** dan **getche()**

Fungsi **getch()** dan **getche()** berguna untuk membaca sebuah karakter tanpa perlu menekan Enter. Selain itu, fungsi ini juga dapat dipakai untuk membaca tombol seperti Spasi, Tab ataupun Enter. Bentuk pemakaiannya:

```
karakter = getch();
```

```
karakter = getche();
```

Kode tombol yang ditekan akan diberikan ke variabel karakter yang bertipe `char`. Sebagai catatan apabila fungsi **getch()** atau **getche()** digunakan, file header bernama `conio.h` perlu disertakan dalam program. Perbedaan kedua fungsi tersebut adalah **getch()** tidak menampilkan karakter dari tombol yang ditekan, sedangkan **getche()** menampilkan karakter dari tombol yang akan ditekan.

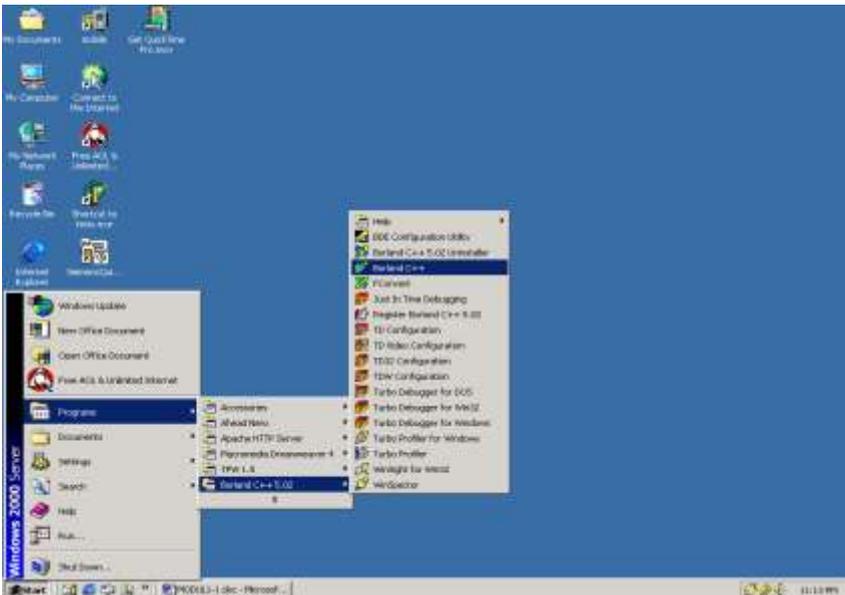
10. Mengoperasikan Borland C++

Borland C++ 5.02 merupakan sebuah program C++ yang dijalankan pada sistem operasi windows. Untuk dapat mengoperasikannya, maka diperlukan perangkat keras dan lunak sebagai berikut.

- a. Satu unit computer yang terdiri dari CPU, Monitor, Keyboard dan Mouse.
- b. Program Borland C++5.02.

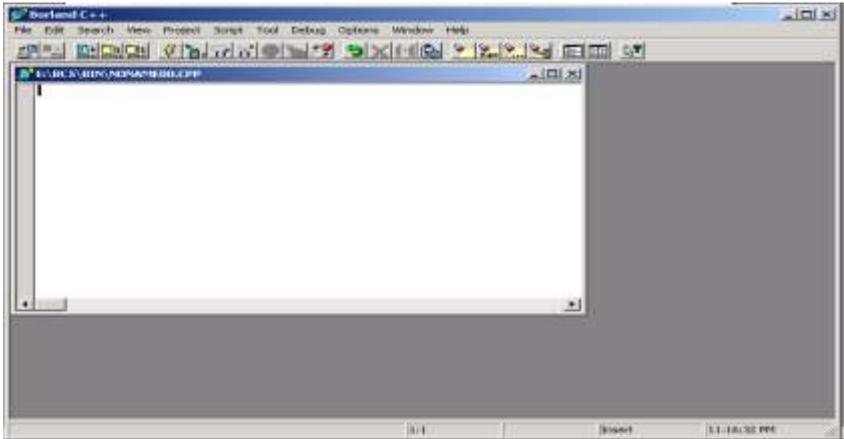
Untuk dapat bekerja dengan C++, terlebih dahulu kita harus menginstal program C++ ke dalam hardisk. Setelah diinstal, maka program C++ dapat dijalankan dengan cara sebagai berikut:

- a. Klik *Start* pada jendela awal windows, kemudian klik *Programs* dan klik Borland C++5.02 (bila menginstal program Borland C++5.02)



Gambar 1.6
Menu Start Programs untuk memilih Borland C++ 5.02

Setelah keluar tampilan program C++, maka kita dapat membuat suatu program baru.



Gambar 1.7
Tampilan New Programs Borland C++ 5.02

- b. Setelah selesai membuat suatu program, maka kita dapat menjalankan program yang telah dibuat benar atau masih salah dengan memilih menu *Project* kemudian pilih *Build all*.



Gambar 1.8
Menu Project *Build all* Borland C++ 5.02

- c. Jika program masih belum benar, maka akan keluar pesan sesuai dengan kesalahan yang dibuat.
- d. Jika program telah benar maka akan keluar hasil Run-nya.
- e. Simpanlah hasil program yang telah dibuat dengan memilih menu *File, Save As*, beri nama sesuai yang diinginkan.
- f. Untuk keluar dari program Pascal ini cukup dengan memilih menu *File* kemudian *Exit* atau dengan memilih tanda  pada sudut kanan jendela.



LATIHAN

Untuk memperdalam pemahaman Anda mengenai materi di atas, kerjakanlah latihan berikut!

- 1) Ada dua jenis bahasa pemrograman, sebutkan dan jelaskan!
- 2) Siapakah perancang bahasa pemrograman Pascal?
- 3) Bagaimanakah susunan struktur program Pascal?
- 4) Apa tujuan utama pembuatan C++?
- 5) Apa perbedaan istilah pada PBO dan C++?

- 6) Apa yang dimaksud dengan encapsulation (pengkapsulan) dalam bahasa berorientasi objek?
- 7) Jelaskan pengertian inheritance (pewarisan) dalam bahasa berorientasi objek?
- 8) Jelaskan pengertian polimorphism dalam bahasa berorientasi objek?

Petunjuk Jawaban Latihan

Jawaban untuk latihan di atas dapat Anda temui pada teori yang diberikan pada Kegiatan Belajar 2 tentang Bahasa Pemrograman Pascal dan C++. Sekarang periksalah baik-baik hasil kerja Anda. Kunci Jawaban berikut ini dapat Anda gunakan sebagai bahan perbandingan. Jawaban yang benar dari soal-soal di atas adalah sebagai berikut.

- 1) a. bahasa pemrograman tingkat rendah (*low level language*) yaitu bahasa pemrograman yang cukup sulit dimengerti oleh penggunanya karena orientasi bahasanya yang lebih dekat ke bahasa mesin
b. bahasa pemrograman tingkat tinggi (*high level language*) yaitu bahasa pemrograman dengan menulis program dalam bahasa awam (bahasa Inggris sehari-hari dengan aturan tertentu) yang kemudian oleh komputer diterjemahkan ke dalam bahasa komputer (bahasa mesin).
- 2) Perancang bahasa Pascal adalah Profesor Niklaus Wirth
- 3) Struktur lengkap program Pascal terdiri dari empat bagian, yaitu kepala program, bagian deklarasi, subprogram dan badan program.
- 4) Tujuan utama pembuatan C++ adalah untuk meningkatkan produktivitas pemrogram dalam membuat aplikasi.
- 5) Perbedaan Istilah pada PBO dan C++ adalah:
 - a. pada PBO disebut variabel instans, sedangkan pada C++ disebut data (anggota data).
 - b. pada PBO disebut metode, sedangkan pada C++ disebut fungsi anggota.
- 6) *Encapsulation* (pengkapsulan) adalah pengemasan data dan fungsi dalam wadah bernama objek. Data tidak lagi diperlakukan sebagai komponen kedua setelah fungsi, melainkan mempunyai kedudukan yang sama dengan fungsi.

- 7) *Inheritance* (pewarisan) adalah sifat dalam bahasa berorientasi objek yang memungkinkan sifat-sifat dari suatu kelas diturunkan ke kelas yang lain.
- 8) *Polymorphism* (polimorfisme) adalah suatu konsep yang menyatakan sesuatu yang sama dapat memiliki berbagai bentuk dan perilaku yang berbeda.



RANGKUMAN

Rangkuman berikut adalah intisari dari pembahasan Kegiatan Belajar 2. Pahami rangkuman berikut agar Anda lebih memahami materi yang baru saja diberikan.

1. Bahasa komputer yang digunakan dalam menulis program dinamakan bahasa pemrograman.
2. Bahasa pemrograman terdiri dari bahasa pemrograman tingkat rendah (*low level language*) yaitu bahasa pemrograman yang cukup sulit dimengerti oleh penggunanya karena orientasi bahasanya yang lebih dekat ke bahasa mesin dan bahasa pemrograman tingkat tinggi (*high level language*) yaitu bahasa pemrograman dengan menulis program dalam bahasa awam (bahasa Inggris sehari-hari dengan aturan tertentu) yang kemudian oleh komputer diterjemahkan ke dalam bahasa komputer (bahasa mesin).
3. Pascal adalah bahasa tingkat tinggi (*high level language*) yang orientasinya pada segala tujuan.
4. Struktur lengkap program Pascal terdiri dari empat bagian, yaitu kepala program, bagian deklarasi, subprogram dan badan program.
5. Bahasa Pemrograman C++ merupakan pengembangan dari bahasa pemrograman C yang diciptakan satu dekade setelah C, diciptakan oleh Bjarne Stroustrup. Tanda ++ berasal dari nama operator kenaikan pada bahasa C.
6. Tujuan utama pembuatan C++ adalah untuk meningkatkan produktivitas pemrogram dalam membuat aplikasi.
7. Pada bahasa yang berorientasi objek, data yang melekat dalam suatu objek biasa disebut variabel instan.
8. Ada 3 karakteristik utama dari bahasa yang berorientasi objek, yaitu: *encapsulation*, *inheritance* dan *polimorphism*.

**TES FORMATIF 2**

Pilihlah satu jawaban yang paling tepat!

- 1) Yang dimaksud dengan bahasa pemrograman tingkat rendah adalah bahasa pemrograman
 - A. yang cukup sulit dimengerti oleh penggunanya karena orientasi bahasanya yang lebih dekat ke bahasa mesin
 - B. yang cukup mudah dimengerti oleh penggunanya karena orientasi bahasanya yang lebih dekat ke bahasa mesin
 - C. dengan menulis program dalam bahasa awam (bahasa Inggris sehari-hari dengan aturan tertentu)
 - D. dengan menulis program dalam bahasa mesin

- 2) Bahasa pemrograman tingkat tinggi adalah bahasa pemrograman
 - A. yang cukup sulit dimengerti oleh penggunanya karena orientasi bahasanya yang lebih dekat ke bahasa mesin
 - B. yang cukup mudah dimengerti oleh penggunanya karena orientasi bahasanya yang lebih dekat ke bahasa mesin
 - C. dengan menulis program dalam bahasa awam (bahasa Inggris sehari-hari dengan aturan tertentu)
 - D. dengan menulis program dalam bahasa mesin

- 3) Struktur lengkap program Pascal terdiri dari
 - A. kepala program, deklarasi, subprogram, teks pernyataan
 - B. kepala program, deklarasi, badan program
 - C. kepala, deklarasi, deskripsi
 - D. kepala program, deklarasi, subprogram, badan program

- 4) Kepala Program dalam Pascal berisi
 - A. subprogram
 - B. deklarasi dari judul program
 - C. nama dari judul program
 - D. badan program

- 5) Perintah yang digunakan untuk mendefinisikan suatu konstanta adalah...
 - A. USES
 - B. LABEL
 - C. CONST
 - D. VAR

- 6) Pernyataan yang sifatnya membaca sampai dengan akhir baris, yaitu membaca nol, atau lebih variabel berupa integer, char, real atau string dari masukan standar pada bahasa Pascal disebut pernyataan
- A. Writeln
 - B. Readln
 - C. Write
 - D. Read
- 7) Pencipta Bahasa Pemrograman C++ adalah
- A. Bjarne Stroustrup.
 - B. Niklaus Wirth
 - C. Benyamin Franklin
 - D. K. Jensen
- 8) Program C++ biasa ditulis dengan nama ekstensi
- A. .PAS
 - B. .TXT
 - C. .CPP
 - D. .DAT

Cocokkanlah jawaban Anda dengan Kunci Jawaban Tes Formatif 2 yang terdapat di bagian akhir modul ini. Hitunglah jawaban yang benar. Kemudian, gunakan rumus berikut untuk mengetahui tingkat penguasaan Anda terhadap materi Kegiatan Belajar 2.

$$\text{Tingkat penguasaan} = \frac{\text{Jumlah Jawaban yang Benar}}{\text{Jumlah Soal}} \times 100\%$$

Arti tingkat penguasaan: 90 - 100% = baik sekali

80 - 89% = baik

70 - 79% = cukup

< 70% = kurang

Apabila mencapai tingkat penguasaan 80% atau lebih, Anda dapat meneruskan dengan modul selanjutnya. **Bagus!** Jika masih di bawah 80%, Anda harus mengulangi materi Kegiatan Belajar 2, terutama bagian yang belum dikuasai.

Kunci Jawaban Tes Formatif

Tes Formatif 1

- 1) B , Lihat pendahuluan A
- 2) C , Lihat pendahuluan B
- 3) D , Lihat pendahuluan Kegiatan Belajar 1
- 4) A , Lihat uraian A
- 5) D , Lihat uraian A.2
- 6) C , Lihat uraian A.3
- 7) A , Lihat uraian A.1
- 8) D , Lihat Tabel 1.1
- 9) A , Lihat Tabel 1.1
- 10) C , Lihat Tabel 1.1

Tes Formatif 2

- 1) A , Lihat pendahuluan Kegiatan Belajar 2
- 2) C , Lihat pendahuluan Kegiatan Belajar 2
- 3) D , Lihat uraian A.1
- 4) C , Lihat uraian A.1.a
- 5) C , Lihat uraian A.2
- 6) B , Lihat uraian A.3.a
- 7) A , Lihat uraian B.1
- 8) C , Lihat uraian B.5

Daftar Pustaka

Abdul Kadir. (1997). *Pemrograman C++*. Yogyakarta: Andi Offset.

Abdul Kadir. (1999). *Pemrograman Pascal Buku 1*. Yogyakarta: Andi Offset.

Deitel. (2001). *C++ How To Program*. Prentice Hall.

Insap Santosa. (1996). *Struktur Data Menggunakan Turbo Pascal 6.0*. Yogyakarta: Andi Offset.

Rinaldi Munir. (1999). *Algoritma dan Pemrograman Jilid 1*. Penerbit ITB.