

Bab 2

Notasi Penulisan Algoritma

Pokok Bahasan

1. Jenis penulisan algoritma
2. Tata aturan penulisan algoritma dengan bahasa deskriptif
3. Tata aturan penulisan algoritma dengan pseudocode
4. Notasi flowchart
5. Tata aturan penulisan algoritma dengan flowchart

Tujuan

1. Menjelaskan tentang pilihan cara penulisan algoritma.
2. Mengetahui tata aturan penulisan algoritma dengan bahasa deskriptif.
3. Memahami tata aturan penulisan algoritma dengan pseudocode.
4. Mengetahu berbagai notasi flowchart.
5. Mengetahui cara penulisan algoritma dengan flowchart.

Algoritma berisi langkah-langkah penyelesaian masalah. Notasi algoritma merupakan hal dasar yang harus diketahui oleh setiap orang yang ingin membuat suatu program, karena dalam notasi algoritma inilah terdapat kerangka-kerangka suatu program. Deskripsi langkah-langkah dalam algoritma tidak mengacu pada sintaks bahasa pemrograman apapun dan tidak tergantung pada spesifikasi komputer yang mengeksekusinya. Tidak ada aturan baku dalam menuliskan algoritma, yang penting mudah dibaca dan menggunakan bahasa yang mudah dipahami. Meskipun demikian untuk menghindari kekeliruan, ketaatan terhadap notasi perlu diperhatikan. Terdapat tiga cara yang umum digunakan dalam menuliskan algoritma yaitu:

1. Kalimat deskriptif
2. Pseudocode
3. Flowchart

Uraian berikutnya akan membahas lebih detail tentang notasi penulisan algoritma disertai dengan contoh.

2.1 Kalimat deskriptif

Notasi penulisan algoritma dengan menggunakan bahasa deskriptif biasa juga disebut dengan notasi alami. Dilakukan dengan cara menuliskan instruksi-instruksi yang harus dilaksanakan dalam bentuk untaian kalimat deskriptif dengan menggunakan bahasa yang jelas. Dasar dari notasi bahasa deskriptif adalah Bahasa Inggris, namun dapat dimodifikasi dengan bahasa sehari-hari termasuk Bahasa Indonesia. Karena tidak ada aturan baku dalam menuliskan algoritma dengan notasi deskriptif maka tiap orang dapat membuat aturan penulisan dan notasi algoritma sendiri. Hal ini dapat dimengerti karena teks algoritma tidak sama dengan teks program. Program adalah implementasi algoritma dalam notasi bahasa pemrograman tertentu. Namun, agar notasi algoritma mudah ditranslasi ke dalam notasi bahasa pemrograman, maka sebaiknya notasi algoritma tersebut berkoresponden dengan notasi bahasa pemrograman pada umumnya. Kata kerja adalah jenis kata yang biasa digunakan dalam penulisan bahasa deskriptif, contohnya tulis, baca, hitung, tampilkan, ulangi, bandingkan, dan lain-lain.

Notasi jenis ini cocok untuk algoritma yang pendek. Tapi untuk masalah algoritma yang panjang, notasi ini kurang efektif. Cara penulisan algoritma dengan notasi bahasa deskriptif paling mudah dibuat, namun demikian cara ini paling sulit untuk diterjemahkan ke dalam bahasa pemrograman. Pada dasarnya teks algoritma dengan bahasa deskriptif disusun oleh tiga bagian utama yaitu:

1. Bagian judul (header)
2. Bagian deklarasi (kamus)
3. Bagian deskripsi

Setiap bagian disertai dengan komentar untuk memperjelas maksud teks yang dituliskan. Komentar adalah kalimat yang diapit oleh pasangan tanda kurung kurawal ('{' dan '}').

2.1.1 Judul Algoritma

Merupakan bagian yang terdiri atas nama algoritma dan penjelasan (spesifikasi) tentang algoritma tersebut. Dibagian ini juga digunakan untuk menentukan apakah teks algoritma yang dibuat tersebut adalah program, prosedur, atau fungsi. Nama algoritma sebaiknya singkat namun cukup menggambarkan apa yang dilakukan oleh algoritma tersebut.

Di bawah nama algoritma disertai dengan penjelasan singkat (intisari) tentang apa yang dilakukan oleh algoritma. Penjelasan dibawah nama algoritma sering dinamakan juga spesifikasi algoritma yang dituliskan dalam kurung kurawal ({}). Algoritma harus ditulis sesuai dengan spesifikasi yang didefinisikan. Gambar 2.1 adalah contoh judul algoritma menghitung luas lingkaran yang disertai dengan penjelasan singkat.

<p>Algoritma Luas_Lingkaran ← Judul Algoritma</p> <p>{Menghitung luas lingkaran untuk ukuran jari-jari tertentu. Algoritma menerima masukan jari-jari lingkaran, menghitung luasnya, lalu cetak luasnya ke piranti keluaran} ← Spesifikasi</p>

Gambar 2.1. Contoh bagian judul algoritma.

2.1.2 Bagian Deklarasi

Di dalam algoritma, deklarasi atau kamus adalah bagian untuk mendefinisikan semua nama yang dipakai di dalam algoritma. Nama tersebut dapat berupa nama variabel, nama konstanta, nama tipe, nama prosedur atau nama fungsi. Semua nama tersebut baru dapat digunakan di dalam algoritma jika telah didefinisikan terlebih dahulu didalam bagian deklarasi. Penulisan sekumpulan nama dalam bagian deklarasi sebaiknya dikelompokkan menurut jenisnya. Pendefinisian nama konstanta sekaligus memberikan nilai konstanta. Pendefinisian nama fungsi atau prosedur sekaligus dengan pendefinisian spesifikasi dan parameternya. Gambar 2.2 adalah bentuk umum bagian deklarasi. Sedangkan gambar 2.3 adalah contoh bagian deklarasi algoritma menghitung luas lingkaran.

<p>Deklarasi :</p> <p>{Nama Type variabel yang bukan tipe data dasar}</p> <p>type waktu:<hh:mm:ss: integer> {Type waktu terdiri dari 3 data masukan yaitu "hh" sebagai jam, "mm" sebagai menit, dan "ss" sebagai detik}</p> <p>{Nama konstanta, harus menyebutkan tipe data dan nilai}</p> <p>constant PHI: real = 3.141592653589793</p> <p>constant E: real = 2.718281828459045</p> <p>{Nama variabel yang menggunakan tipe data dasar}</p> <p>nama : String {suatu nilai yang merupakan kumpulan karakter}</p> <p>ketemu : boolean {suatu nilai logika (true atau false)}</p> <p>beratBadan : real {suatu nilai bilangan pecahan}</p> <p>jumlahAnak : integer {suatu nilai bilangan bulat}</p> <p>{Nama fungsi, mendefinisikan domain, nama, jumlah, tipe dan jumlah parameter, serta tipe data keluaran}</p> <p>function tambah(x:int, y:int): int</p> <p>{proses: menambahkan dua nilai data dan hasil penambahan sebagai nilai keluaran fungsi}</p>
--

Gambar 2.2. Bentuk umum bagian deklarasi algoritma.

```
Deklarasi :  
jari_jari = real {tipe data bilangan pecahan}  
luas = real {tipe data bilangan pecahan}  
PHI = 3.14
```

Gambar 2.3. Contoh bagian deklarasi algoritma.

2.1.3 Bagian deskripsi.

Deskripsi adalah bagian inti dari struktur algoritma. Bagian ini berisi uraian langkah-langkah penyelesaian masalah. Langkah-langkah ini dituliskan dengan notasi yang lazim dalam penulisan algoritma. Setiap langkah algoritma dibaca dari langkah paling atas hingga langkah paling bawah. Urutan penulisan menentukan urutan pelaksanaan perintah. Seperti telah dijelaskan di bab satu bahwa penyusun atau struktur dasar algoritma adalah langkah-langkah. Suatu Algoritma dapat terdiri dari tiga struktur dasar, yaitu runtunan, pemilihan dan pengulangan. Ketiga jenis langkah tersebut membentuk konstruksi suatu algoritma. Pada bagian deskripsi inilah letak tiga struktur algoritma tersebut.

Gambar 2.3 adalah contoh bagian deskripsi algoritma menghitung luas lingkaran. Gambar 2.4 adalah contoh algoritma menghitung luas lingkaran yang dituliskan menggunakan kalimat deskriptif secara lengkap.

```
Deskripsi :  
1. Baca jari_jari  
2. Hitung luas = jari_jari * jari_jari * PHI  
3. Tampilkan luas ke layar  
4. Selesai
```

Gambar 2.4. Contoh bagian deskripsi algoritma.

```
Algoritma Luas Lingkaran  
{Menghitung luas lingkaran untuk ukuran jari-jari tertentu.  
Algoritma menerima masukan jari-jari lingkaran, menghitung  
luasnya, lalu cetak luasnya ke piranti keluaran}  
  
Deklarasi :  
jari_jari = real {tipe data bilangan pecahan}  
luas = real {tipe data bilangan pecahan}  
PHI = 3.14  
  
Deskripsi:  
1. Baca jari_jari  
2. Hitung luas = PHI*jari_jari * jari_jari  
3. Tampilkan luas ke layar  
4. Selesai
```

Gambar 2.5. Contoh penulisan algoritma menggunakan kalimat deskriptif.

2.2 Pseudocode

Pseudocode adalah cara penulisan algoritma yang menyerupai bahasa pemrograman tingkat tinggi. Pseudocode menggunakan bahasa yang hampir menyerupai bahasa pemrograman. Biasanya pseudo-code menggunakan bahasa yang mudah dipahami secara universal dan juga lebih ringkas dari pada algoritma. Pseudocode berisi deskripsi dari algoritma pemrograman komputer yang menggunakan struktur sederhana dari beberapa bahasa pemrograman tetapi bahasa tersebut hanya di tujukan agar dapat di baca manusia. Sehingga pseudocode tidak dapat dipahami oleh komputer. Supaya notasi pseudocode bisa dipahami oleh komputer maka harus diterjemahkan terlebih dahulu menjadi sintaks bahasa pemrograman komputer tertentu.

Dalam pseudocode, tidak ada sintaks standar yang resmi. Karena itu, pseudocode ini dapat diterapkan dalam berbagai bahasa pemrograman. Disarankan untuk menggunakan keyword yang umum digunakan seperti : if, then, else, while, do, repeat, for, dan lainnya. Keuntungan menggunakan notasi pseudocode adalah kemudahan mentranslasi ke notasi bahasa pemrograman, karena terdapat korespondensi antara setiap pseudocode dengan notasi bahasa pemrograman. Tabel 2.1. menunjukkan perbandingan beberapa kata yang biasa digunakan dalam penulisan algoritma dengan menggunakan kalimat deskriptif dan pseudocode.

Tabel 2.1. Perbandingan beberapa kata yang biasa digunakan dalam penulisan algoritma dengan menggunakan kalimat deskriptif dan pseudocode.

Kalimat Deskriptif	Pseudocode
Masukkan panjang	Input panjang
	Read panjang
	Baca panjang
Hitung luas dengan rumus panjang x lebar	luas \leftarrow panjang * lebar
Tampilkan luas	Output luas
	Print luas
	Write luas
Jika sudah selesai, cetak luas	If kondisi_selesai == true then print luas
Nilai B ditambah 5	B \leftarrow B+5
Jika nilai A lebih kecil dari 5 maka nilai B dibagi 3	If A<5 then B \leftarrow B/3

Jika nilai A lebih besar dari nilai B maka tampilkan A, jika A lebih kecil dari B maka tampilkan nilai B	If A>B then print A else print B
--	----------------------------------

Struktur penulisan pseudocode secara umum sama dengan struktur penulisan algoritma dengan menggunakan kalimat deskriptif yaitu dimulai dari judul/header, deklarasi/kamus dan diakhiri dengan deskripsi. Meskipun tidak ada sintaks khusus dalam penulisan pseudocode, tetapi terkadang pseudocode dituliskan dengan menggunakan style atau gaya penulisan dari beberapa bahasa pemrograman yang ada, seperti Fortran, Pascal, C dan lain-lain. Gambar 2.5 adalah contoh penulisan pseudocode dengan menggunakan gaya penulisan beberapa bahasa pemrograman. Gambar 2.6 adalah contoh pseudocode menentukan bilangan terbesar dari 3 masukan bilangan. Sedangkan Gambar 2.7 adalah contoh pseudocode konversi nilai angka menjadi nilai huruf.

Fortran style	Pascal style	C style
<pre> program TikTok do i=0to100 set flag to true if i is divisible by 3 print "Tik" set flag to false if i is divisible by 5 print "Tok" set flag to false if flag, print i print a newline end do </pre>	<pre> procedure TikTok for i:=0to100 do set flag to true; if i is divisible by 3 then print "Tik"; set flag to false; if i is divisible by 5 then print "Tok"; set flag to false; if flag, print i; print a newline; end </pre>	<pre> void function TikTok for(i=0;i<=100;i++){ set flag to true; if i is divisible by 3 print "Tik"; set flag to false; if i is divisible by 5 print "Tok"; set flag to false; if flag, print i; print a newline; } </pre>

Gambar 2.5. Contoh pseudocode menggunakan gaya penulisan Fortran, Pascal dan C.

<p>Algoritma : Bilangan Maksimum {Dibaca tiga buah bilangan dari piranti masukan. Carilah bilangan bulat maksimum diantara ketiga bilangan tersebut}</p> <p>Deklarasi : Bil1,Bil2,Bil3 : integer {bilangan yang dicari maksimumnya} Max : integer {variabel bantu}</p> <p>Deskripsi : 1. Read (Bil1,Bil2) 2. If Bil1 >= Bil2 then 3. Bil1 = Max 4. Else Bil2 = Max 5. Read (Bil3) 6. If Bil3 >= Max then 7. Bil3 = Max 8. Write (Max)</p>

Gambar 2.6 Contoh pseudocode menentukan bilangan terbesar dari 3 masukan bilangan.

Algoritma Konversi Nilai Angka ke Huruf

{Dibaca tiga buah bilangan dari piranti masukan. Carilah dan tampilkan bilangan bulat maksimum diantara ketiga bilangan tersebut}

Deklarasi:

nama dan nim = String

nilai = integer

Deskripsi:

```
1. Read (nama)
2. Read (nim)
3. Read (nilai)
4. If (nilai < 45) then
5.     Grade = E
6. Else if (nilai >= 45) and (nilai < 60) then
7.     Grade = D
8. Else if (nilai >= 60) and (nilai < 70) then
9.     Grade = C
10. Else if (nilai >= 70) and (nilai < 80) then
11.     Grade = B
12. Else
13.     Grade = A
14. Write (nama)
15. Write (NIM)
16. Write (nilai)
17. Selesai
```

Gambar 2.7. Contoh pseudocode konversi nilai angka menjadi nilai huruf.

2.3 Flowchart

Flowchart adalah cara penulisan algoritma dengan menggunakan notasi grafis. Flowchart merupakan gambar atau bagan yang memperlihatkan urutan atau langkah-langkah dari suatu program dan hubungan antar proses beserta pernyataannya. Gambaran ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan antara proses digambarkan dengan garis penghubung. Dengan menggunakan flowchart akan memudahkan kita untuk melakukan pengecekan bagian-bagian yang terlupakan dalam analisis masalah. Disamping itu flowchart juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram yang bekerja dalam tim suatu proyek. Flowchart

menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian.

Pada dasarnya terdapat berbagai macam flowchart, diantaranya yaitu Flowchart Sistem (System Flowchart), Flowchart Paperwork / Flowchart Dokumen (Document Flowchart), Flowchart Skematik (Schematic Flowchart), Flowchart Program (Program Flowchart), Flowchart Proses (Process Flowchart). Untuk keperluan pembuatan program maka digunakan Flowchart Program.

Flowchart program menggambarkan urutan instruksi yang digambarkan dengan simbol tertentu untuk memecahkan masalah dalam suatu program. Dalam flowchart program mengandung keterangan yang lebih rinci tentang bagaimana setiap langkah program atau prosedur seharusnya dilaksanakan. Flowchart ini menunjukkan setiap langkah program atau prosedur dalam urutan yang tepat saat terjadi. Programmer menggunakan flowchart program untuk menggambarkan urutan instruksi dari program komputer. Analis Sistem menggunakan flowchart program untuk menggambarkan urutan tugas-tugas pekerjaan dalam suatu prosedur atau operasi.

Dalam pembuatan flowchart program tidak ada rumus atau patokan yang bersifat mutlak. Karena flowchart merupakan gambaran hasil pemikirandalam menganalisis suatu masalah yang nantinya akan diubah menjadi program komputer. Sehingga flowchart yang dihasilkan dapat bervariasi antara satu pemrogram dengan yang lainnya. Namun demikian terdapat beberapa anjuran yang harus diperhatikan, yaitu :


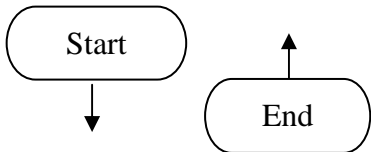

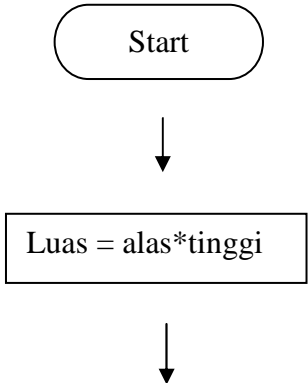
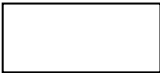
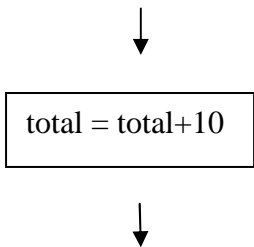
1. Flowchart digambarkan di suatu halaman dimulai dari sisi atas ke bawah dan dari sisi kiri ke kanan.
2. Aktivitas yang digambarkan harus didefinisikan dengan menggunakan bahasa dan simbol yang tepat dan definisi ini harus dapat dimengerti oleh pembacanya.
3. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas. Hanya terdapat satu titik awal dan satu titik akhir.
4. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja, misalkan MENGHITUNG NILAI RATA-TARA.
5. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
6. Lingkup dan range dari aktifitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu digambarkan pada flowchart yang sama. Simbol konektor harus digunakan dan

percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.

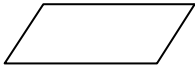
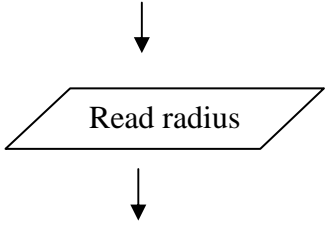
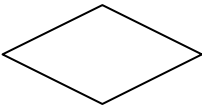
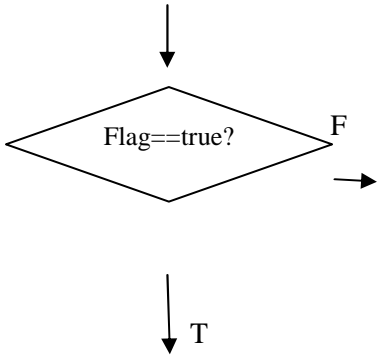
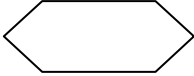
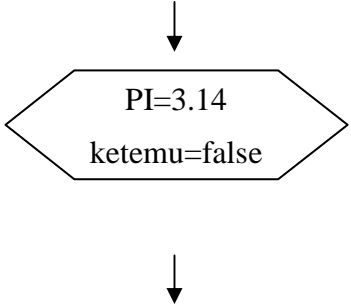
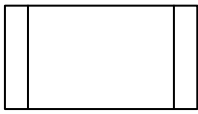
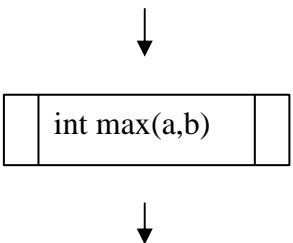


7. Gunakan simbol-simbol flowchart yang standar.

Simbol-simbol flowchart yang biasanya dipakai adalah simbol-simbol flowchart standar yang dikeluarkan oleh ANSI dan ISO. Tabel 2.2 merupakan beberapa simbol flowchart yang digunakan dalam menggambar suatu flowchart:

Tabel 2.2.Simbol-simbol Flowchart

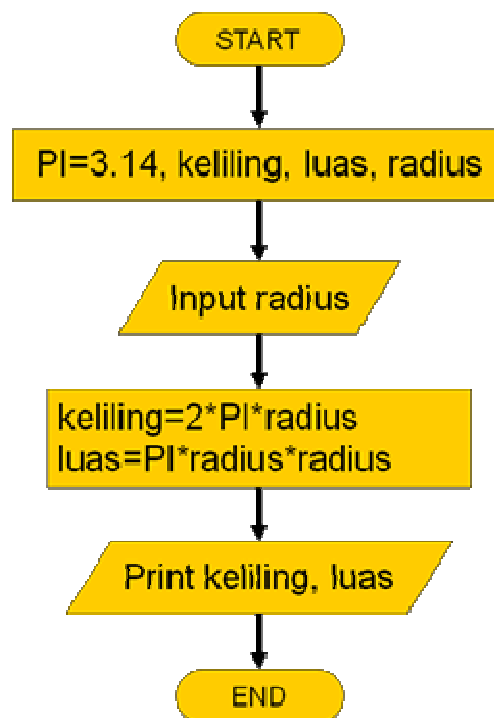
SIMBOL	NAMA	FUNGSI	CONTOH
	Terminator	Simbol Awal (Start) / Simbol Akhir (End)	
	Flow Line	Simbol aliran / penghubung	
	Proses	Perhitungan / pengolahan	

Tabel 2.2.Simbol-simbol Flowchart (lanjutan)

SIMBOL	NAMA	FUNGSI	CONTOH
	Input / Output Data	Mempresentasikan pembacaan data (read) / penulisan (write).	
	Decision	Simbol pernyataan pilihan, berisi suatu kondisi yang selalu menghasilkan 2 nilai keluaran yaitu benar atau salah	
	Preparation	Inisialisasi / pemberian nilai awal	
	Predefined Process (subprogram)	Proses menjalankan sub program / fungsi / prosedur	
	On Page Connector	Penghubung Flow chart pada satu halaman	

			↓ (A)
⬇	Off Page Connector	Penghubung Flow chart pada halaman berbeda	⬇ 2 ↓ 2

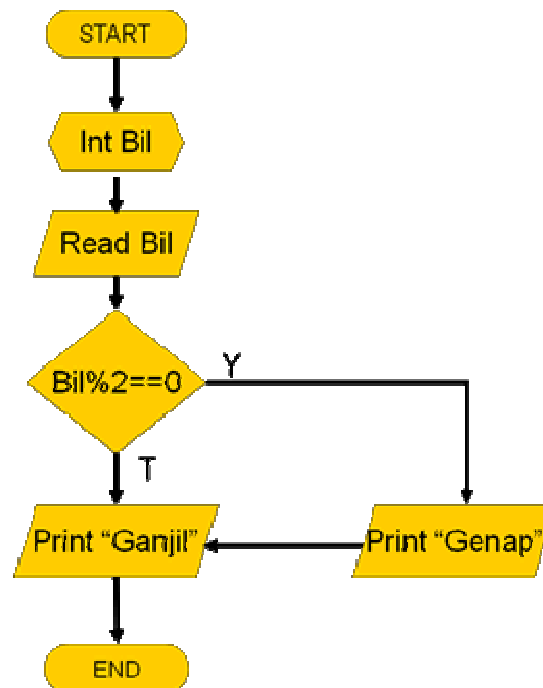
Untuk memahami lebih dalam mengenai flowchart ini, dibuat sebuah kasus sederhana. Misalnya buatlah sebuah rancangan program dengan menggunakan flowchart untuk menentukan keliling dan luas lingkaran. Perumusan untuk menentukan luas lingkaran adalah: $\text{luas} = \text{PI} \times \text{radius} \times \text{radius}$, dan keliling lingkaran adalah $\text{keliling} = 2 \times \text{PI} \times \text{radius}$, dengan PI adalah sebuah konstanta 3.14. Flowchart permasalahan ini dapat dilihat di Gambar 2.8.



Gambar 2.8. Flowchart luas dan keliling lingkaran

Selanjutnya akan dibuat contoh flowchart yang mengandung percabangan atau decision. Misalnya untuk permasalahan penentuan apakah suatu bilangan termasuk bilangan ganjil atau genap. Cara menyelesaikan permasalahan ini adalah dengan membagi bilangan

dengan angka 2. Jika nilai sisa pembagian nya adalah 0 maka bilangan tersebut adalah bilangan genap, jika nilai sisa pembagiannya adalah 1 maka bilangan tersebut adalah bilangan ganjil. Operasi aritmatika yang digunakan untuk menentukan nilai sisapembagian adalah perasi modulo (%). Flowchart permasalahan ini dapat dilihat di Gambar 2.9. Dalam hal ini Bil adalah bilangan yang akan di cek status ganjil atau genapnya.



Gambar 2.9. Flowchart Penentuan Bilangan Ganjil-Genap

2.4 Latihan Soal

1. Buatlah algoritma untuk menentukan nilai terbesar dari bilangan bulat yang dibaca dari keyboard dan menuliskan hasilnya ke layar! ALgoritma dibuat dalam bentuk kalimat deskriptif, pseudocode dan flowchart.
2. Buat algoritma dalam bahasa deskriptif dan flowchart untuk kegiatan mengambil dan menabung uang di bank melalui teller!
3. Buat algoritma dalam bahasa deskriptif dan pseudocode untuk menentukan apakah suatu bilangan merupakan bilangan genap atau ganjil!
4. Membuat flowchart untuk proses pembuatan kopi yang rasa manisnya tepat
5. 2. membuat flowchart mengurutkan 3 bilangan acak
6. Buat algoritma untuk menghitung nilai N suku dari deret aritmatika berikut: