



# BAB 3

---

## PENULISAN DAN NOTASI ALGORITMA PEMROGRAMAN

# **BAB 3**

# **PENULISAN DAN NOTASI**

# **ALGORITMA PEMROGRAMAN**

## **Capaian Pembelajaran**

Setelah mempelajari materi dalam bab ini, Mahasiswa diharapkan mampu menganalisis sebuah kasus operasional bisnis dan menyampaikan pendapat dalam bentuk flowchart dan pseudocode.

## **Pokok Bahasan**

1. Penulisan Kalimat Deskriptif
2. Pseudocode
3. Notasi Flowchart
4. Implementasi Notasi Algoritma pada beragam kasus bisnis.

## **Evaluasi Pembelajaran**

Soal Latihan Penulisan dan Notasi Algoritma Pemrograman

---

## Pre Test

### Penulisan dan Notasi Algoritma Pemrograman

1. Menurut Anda, apa fungsi dari algoritma ?
2. Apa yang Anda ketahui tentang penulisan Algoritma menggunakan kalimat deskriptif? Jelaskan!
3. Apa yang Anda ketahui tentang pseudocode? Jelaskan!
4. Menurut Anda, apa kegunaan dari Flowchart?
5. Menurut Anda, apa perbedaan dari pseudocode dan flowchart?

Algoritma dapat didefinisikan sebagai kumpulan instruksi yang berurutan secara sistematis, yang menggambarkan langkah - langkah yang perlu diikuti untuk menyelesaikan permasalahan tertentu dengan bantuan komputer. Instruksi dalam algoritma merupakan perintah untuk melakukan aksi tertentu, misalnya instruksi menulis hasil perhitungan, membaca masukan, memanggil sebuah prosedur, dan sebagainya. Instruksi algoritma dinyatakan sebagai pernyataan yang dapat disajikan dalam bentuk tulisan. Tulisan pernyataan instruksi algoritma haruslah menggunakan bahasa yang mudah dibaca dan dipahami oleh manusia. Penyajian penulisan algoritma ini dapat dilakukan menggunakan untaian kalimat deskriptif, pseudocode dan flowchart. Bab 3 ini akan menjelaskan tentang bagaimana penggunaan dan penyajian notasi algoritma pemrograman dalam bentuk kalimat deskriptif, pseudocode, dan flowchart. Selain itu, di bab ini juga akan menjelaskan beberapa contoh penerapan notasi algoritma pada beragam kasus bisnis.

### 3.1 Penulisan Kalimat Deskriptif

Pada dasarnya, sebuah algoritma merupakan langkah - langkah sistematis yang dapat digunakan untuk penyelesaian permasalahan tertentu. Penulisan algoritma dapat disajikan dalam bentuk kalimat deskriptif. Penulisan notasi algoritma menggunakan kalimat deskriptif disebut dengan notasi alami.

Penulisan instruksi - instruksi dalam notasi algoritma deskriptif dilakukan dalam bentuk kalimat deskriptif dengan menggunakan bahasa yang jelas. Notasi algoritma deskriptif ini biasanya digunakan pada algoritma yang pendek karena apabila untuk algoritma yang panjang notasi deskriptif kurang efektif. Penulisan instruksi pada algoritma deskriptif dapat menggunakan bahasa sehari - hari, seperti Bahasa Indonesia atau Bahasa Inggris. Tidak aturan yang baku untuk menuliskan algoritma deskriptif. Namun, penulisan algoritma menggunakan kalimat deskriptif ini dapat menimbulkan ambiguitas, dimana kalimat dapat mengandung makna ganda sehingga dapat terjadi kesalahpahaman saat membaca algoritma tersebut. Algoritma yang pendek akan lebih efektif jika ditulis menggunakan untaian kalimat deskriptif ini.

### 3.1.1 Penggunaan Notasi Kalimat Deskriptif

Algoritma dengan notasi kalimat deskriptif biasanya menggunakan berbagai kata kerja dalam bahasa manusia, seperti mulai, tulis, baca, tampilkan, jika, maka, ulangi. Adapun menggunakan bahasa manusia, namun dalam Bahasa Inggris, seperti read, print, write, if, end, dsb. Tidak ada ketentuan baku dalam penulisan notasi kalimat deskriptif, penulis bebas dalam menuliskan notasi, yang penting tulisan tersebut jelas dan mudah dimengerti oleh pembaca. Dalam penulisannya, notasi kalimat deskriptif terbagi menjadi 3 bagian, antara lain: judul, deklarasi, dan deskripsi. Bagian judul merupakan bagian yang terdiri dari nama algoritma dan penjelasan tentang algoritma yang akan dibuat. Bagian deklarasi mendefinisikan semua nama yang digunakan pada algoritma, seperti variable, tipe data, fungsi, konstanta. Bagian deskripsi mendefinisikan uraian langkah - langkah penyelesaian masalah.

Adapun contoh dari penulisan notasi kalimat deskriptif adalah sebagai berikut:

Contoh 1:

```
Judul:  
Algoritma_Perkalian  
{perkalian 2 bilangan bulat. Algoritma memasukkan 2 bilangan bulat, kemudian  
menghitung hasil perkalian dari kedua angka tersebut, dan mencetak hasil  
perkalian tersebut}  
Deklarasi:  
bil, bi2, hasilKali = int  
Deskripsi:  
Input Bil, Bi2  
hitung hasilKali = bil x bi2  
Tampilkan hasilKali  
Selesai
```

Contoh 2:

```
Judul:  
Algoritma_hitungDiskon
```

```
{menghitung harga setelah diskon dengan memasukkan harga awal produk dan  
besaran diskonnya}  
Deklarasi:  
hargaAwal, besaranDiskon(%), hargaDiskon = int  
Deskripsi:  
Input hargaAwal, besaranDiskon(%)  
Hitung hargaDiskon = hargaAwal-(hargaAwal x besaranDiskon(%))  
Tampilkan hargaDiskon  
Selesai.
```

Kedua contoh algoritma diatas dapat menunjukkan bahwa penggunaan notasi kalimat deskriptif sangatlah sederhana, mudah dibaca dan dipahami. Penyelesaian permasalahan yang sederhana lebih mudah menggunakan notasi kalimat deskriptif ini, namun jika penyelesaian masalah terbilang kompleks dan rumit, akan lebih sulit terselesaikan menggunakan notasi kalimat deskriptif, dan dapat menggunakan notasi algoritma lainnya, seperti pseudocode atau flowchart.

### 3.2 Pseudocode

Algoritma dapat ditulis menggunakan notasi pseudocode. Penulisan menggunakan notasi pseudocode ini memerlukan penjabaran yang selaras dengan logika, sehingga menjadikan algoritma mudah dibaca dan dipahami. Pseudocode berasal dari kata pseudo yang berarti imitasi atau mirip atau menyerupai dan code yang berarti kode program. Dengan kata lain, pseudocode dalam sebuah algoritma bukanlah kode program sebenarnya, hanya mirip saja, sehingga tidak dapat langsung dieksekusi oleh komputer. Pseudocode harus melalui proses translasi terlebih dulu. Translasi merupakan proses penerjemahan pseudocode ke dalam bahasa pemrograman tertentu. Terdapat berbagai macam bahasa pemrograman yang dapat dieksekusi oleh komputer, seperti Bahasa Java, Bahasa Pascal, Bahasa C++, dan sebagainya. Sama halnya dengan kalimat deskriptif, penulisan algoritma menggunakan pseudocode juga terbilang mudah. Secara umum, tidak ada aturan atau notasi pseudocode yang baku untuk menuliskan algoritma. Notasi yang digunakan juga menggunakan bahasa manusia, seperti read, print, write, if, end.

Hal ini berbeda dengan notasi bahasa pemrograman yang harus mematuhi aturan - aturan tertentu agar program dapat berjalan. Proses penerjemahan atau translasi dari pseudocode ke kode program mudah dilakukan karena terdapat korespondensi antara notasi pseudocode dan notasi bahasa pemrograman atau dengan kata lain, sebelum mengenal bahasa pemrograman, sebaiknya kita mengetahui terlebih dulu bagaimana alur alogaritma suatu program menggunakan pseudocode.

### 3.2.1 Pentingnya Menggunakan Pseudocode

Pada dasarnya, pseudocode merupakan notasi algoritma yang dapat dijadikan acuan untuk membuat sebuah program. Pseudocode ini bukanlah kode pemrograman sebenarnya, namun penulisannya menggunakan bahasa tiruan dari bahasa pemrograman. Pseudocode dapat memecahkan permasalahan dengan kalimat sederhana, sehingga dapat mudah dibaca dan dipahami oleh manusia. Sedangkan, bahasa pemrograman merupakan bahasa yang nantinya akan dibaca oleh software dalam pembuatan program. Pada intinya, pseudocode penting digunakan untuk menuliskan algoritma dengan cara yang lebih sederhana.

Berikut adalah beberapa uraian tentang pentingnya menggunakan pseudocode:

- a. Memberikan kemudahan dalam pemahaman  
Konsep dan struktur pseudocode dapat dipahami dengan mudah, sehingga akan memberikan kemudahan dalam membuat rancangan alur algoritma yang benar dan tepat. Dengan demikian, pseudocode dapat memberikan pemahaman alogaritma yang baik sebelum mempelajari bahasa pemrograman yang lebih kompleks.
- b. Menjadi sarana penghubung yang baik  
Pseudocode merupakan notasi alogaritma yang memberikan kemudahan bagi para pemula yang sedang ingin merancang sebuah progam. Pseudocode mampu menjadi penghubung yang baik, yang mana mampu menyambungkan logika dengan program yang sedang dirancang.



- c. Dapat digunakan sebagai dokumentasi.

Penulisan notasi algoritma menggunakan pseudocode dapat menjadi sarana dokumentasi. Hal ini dikarenakan pseudocode yang sudah dibuat dapat disimpan dan dapat dikembangkan lagi apabila akan melakukan peningkatan pada alur alogaritma pemrograman.

- d. Penerjemah flowchart.

Notasi Pseudocode dapat digunakan oleh programmer pemula untuk menerjemahkan flowchart ke kode pemrograman dengan lebih efisien dan lebih efektif.

- e. Memudahkan Proses Pengembangan Sistem

Pseudocode dapat membantu dalam proses pengembangan aplikasi, karena menggunakan bahasa sederhana dan struktur sederhana sehingga mudah dibaca dan mudah dimodifikasi.

### 3.2.2 Ciri - ciri Pseudocode

Adapun ciri - ciri atau karakteristik yang dimiliki oleh pseudocode. Ciri - ciri ini perlu diketahui untuk mengenal lebih jauh tentang pseudocode.

Berikut uraian dari ciri - ciri tersebut:

- a. Pseudocode menggunakan bahasa yang tidak rumit, yang mana mudah dibaca dan dipahami oleh manusia.
- b. Pseudocode tidak memiliki aturan atau standar baku yang harus diikuti oleh pembuatnya
- c. Pseudocode menggunakan sintaks dari suatu program, seperti  $\leftarrow$ ,  $<$ ,  $>$ ,  $<=$ ,  $>=$ , dll
- d. Pseudocode ditulis berupa langkah - langkah yang berurutan untuk menyelesaikan permasalahan.

### 3.2.3 Notasi Pseudocode

Penggunaan notasi pada pseudocode memiliki tujuan untuk memahami fungsi-fungsi dan algoritma di sebuah kode pemrograman. Pada pembahasan sebelumnya, telah dipahami bahwa pseudocode tidak memiliki aturan yang baku



dalam penggunaan notasi. Namun, pada umumnya pseudocode menggunakan beberapa notasi sebagai berikut:

**INPUT:** notasi ini digunakan untuk memasukkan isi, seperti klik atau tombol. Contoh notasi untuk INPUT adalah Read, Get, Obtain.

**OUTPUT:** notasi ini digunakan untuk menampilkan hasil dari INPUT dan proses. Contoh notasi untuk OUTPUT adalah Print, Write, Display, Show.

**COMPUTE:** notasi ini digunakan untuk memproses atau menghitung masukan. Contoh notasi untuk COMPUTE adalah Compute, Calculate, Determine.

**DECISION:** notasi ini digunakan untuk menunjukkan adanya percabangan pada algoritma. Contoh notasi untuk DECISION adalah If-then-else, Case.

**ITERATION:** notasi ini digunakan untuk menunjukkan adanya perulangan pada algoritma. Contoh notasi untuk ITERATION adalah While, For, Repeat-Until. Notasi While digunakan untuk sebuah perulangan yang memiliki iterasi awal. For digunakan untuk sebuah perulangan perhitungan iterasi. Sedangkan, Repeat-Until digunakan sebuah perulangan yang memiliki kondisi akhir.

### 3.2.4 Penggunaan notasi Pseudocode

Tujuan utama dari dibuatnya pseudocode adalah agar manusia dapat membaca atau menyelesaikan permasalahan dengan sebuah algoritma. Pseudocode berisi instruksi - instruksi sistematis yang berguna untuk menyelesaikan suatu permasalahan.

Seorang programmer membutuhkan pseudocode untuk digunakan sebagai dasar algoritma pengembangan sebuah program. Untuk membuat sebuah program, pseudocode diubah / ditranslasi ke bahasa pemrograman. Terdapat berbagai bahasa pemrograman yang dapat digunakan untuk mengembangkan suatu program, seperti Bahasa Java, Bahasa Pascal, Bahasa C++, dan sebagainya. Untuk pembahasan lebih lanjut, bahasa pemrograman yang akan digunakan pada buku ini hanyalah bahasa Java saja. Seperti pada pembahasan sebelumnya, dapat

diketahui bahwa penggunaan notasi pseudocode sangatlah mudah, karena tidak ada aturan yang perlu diikuti. Berikut ini diuraikan contoh - contoh penggunaan notasi pseudocode:

Contoh 1: Instruksi (pernyataan): Tulis kalimat "Semangat belajar, ya!"

Maka, notasi pseudocodenya dapat ditulis:

```
write (Semangat belajar, ya!)
```

Notasi pseudocode ditulis dengan sederhana, tidak memikirkan bagaimana tampilan dari luaran yang diinginkan. Sebaliknya, jika dalam bahasa pemrograman, terdapat instruksi yang dapat digunakan untuk menampilkan luaran dengan berbagai format. Misalnya, "Semangat belajar, ya!" ingin ditampilkan dalam satu baris atau antara "semangat belajar" dan "ya!" ditampilkan menggunakan koma atau hanya spasi saja. Tampilan luaran tersebut dapat diperoleh dengan menggunakan aturan aturan yang berlaku didalam bahasa pemrograman.

Notasi Write berfungsi untuk mencetak kalimat "Semangat belajar, ya!", Notasi Write di dalam algoritma berkoresponden dengan print dan println dalam bahasa Java. Pada bahasa pemrograman Java antara setiap instruksi dipisahkan dengan tanda ";" (semicolon).

Dengan demikian, translasi Write (Semangat belajar, ya!) ke bahasa Java adalah:

Notasi pseudocode:

```
write (Semangat belajar, ya!)
```

Notasi Bahasa Java:

```
System.out.print ("Semangat belajar, ya!");
```

Contoh 2: Instruksi (pernyataan): Memasukkan nilai m ke dalam n

Maka, notasi pseudocodenya dapat ditulis:

```
n ← m
```

Notasi "←" pada pseudocode memiliki arti yaitu mengisi nilai m ke dalam variable n. Translasi notasi "←" dalam bahas Java adalah "=". Sehingga,

$n \leftarrow m$  ditranslansi ke Bahasa Java menjadi:  $n=m$

Dalam penulisannya, pseudocode juga memiliki struktur yang sama dengan notasi kalimat deskriptif. Struktur pseudocode dapat diuraikan sebagai berikut:

### Judul

Bagian judul berisi nama program yang akan dibuat. Judul dapat menyatakan penjelasan singkat tentang program yang akan dibuat. Biasanya bagian ini penulisannya diawali dengan kata "PROGRAM", kemudian diikuti dengan nama programnya. Umumnya, nama program terdiri dari satu kata, singkat, namun menggambarkan apa yang dilakukan oleh program. Setelah menyatakan nama program, maka dapat diberikan uraian singkat yang mendeskripsikan program dibawah nama program. Berikut adalah contoh penulisannya:

```
PROGRAM LuasLingkaran
{Program membaca Panjang jari-jari (r) sebuah lingkaran, menghitung luas
lingkaran, lalu mencetak luas tersebut ke layar.}
```

### Deklarasi

Bagian deklarasi digunakan untuk menyatakan semua nama yang digunakan di dalam algoritma. Nama yang dideklarasikan dapat berupa nama konstanta, nama peubah, nama tipe data, nama prosedur atau fungsi. Dalam algoritma, semua nama harus dikenali terlebih dulu sebelum mereka digunakan. Bagian deklarasi ini dapat dikosongkan apabila tidak ada penggunaan nama didalam bagian isi algoritma. Berikut adalah contoh penulisannya:

### DEKLARASI:

```
const pi=3.14 {konstantan}
r: real      {jari - jari lingkaran dalam satuan cm}
L: real      {luas lingkaran dalam satuan [(cm) ^2]}
```

## Isi Algoritma

Bagian isi merupakan bagian utama dari algoritma, berisi sekumpulan instruksi atau langkah - langkah yang sistematis, kondisional, atau berulang untuk menjalankan program dengan menggunakan nama - nama yang telah dideklarasikan pada bagian deklarasi. Berikut adalah contoh penulisannya:

### ALGORITMA:

```
read(r)      {masukkan jari-jari lingkaran}
L ← pi*r*r {hitung luas lingkaran}
write(L)     {tampilkan luas lingkaran ke layar}.
```

## 3.3 Notasi Flowchart

Flowchart merupakan standar dasar pemodelan yang berfungsi untuk menggambarkan alur logika dalam algoritma pemrograman. Flowchart ditampilkan dalam bentuk diagram yang mana dapat merepresentasikan urutan dari operasi yang dilakukan dari awal sampai akhir untuk penyelesaian algoritma pemrograman, sehingga mendapatkan hasil. Penggunaan flowchart akan membuat algoritma pemrograman menjadi lebih mudah dimengerti, karena algoritma ditampilkan dalam simbol - simbol gambar yang terstandar. Tujuan penggunaan flowchart adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, ringkas dan jelas dengan menggunakan simbol - simbol yang mudah dipahami.

### 3.3.1 Jenis - jenis Flowchart

Flowchart dapat dibagi menjadi 5 jenis, yaitu flowchart sistem, flowchart dokumen, flowchart skematik, flowchart program, dan flowchart proses. dan program flowchart. Penjelasan dari ke-5 jenis flowchart tersebut dapat diuraikan sebagai berikut:

- a. Flowchart sistem merupakan diagram alir yang menggambarkan arus pekerjaan secara keseluruhan dari suatu sistem. Flowchart sistem

menjabarkan urutan - urutan dari prosedur - prosedur yang ada didalam sistem.

- b. Flowchart dokumen merupakan diagram alir yang menggambarkan arus dari laporan atau formulir.
- c. Flowchart skematik merupakan diagram alir yang hampir sama dengan flowchart sistem, yaitu menggambarkan arus prosedur yang ada didalam sistem. Perbedaanya adalah dalam pembuatannya, flowchart skematik tidak hanya menggunakan simbol namun juga menggunakan gambar-gambar computer dan peralatan lainnya yang digunakan. Gambar digunakan untuk memudahkan komunikasi kepada orang yang kurang paham terkait diagram alir yang dibuat.
- d. Flowchart program merupakan diagram alir yang menggambarkan urutan atau langkah - langkah logika dari suatu prosedur pemecahan masalah. Untuk flowchart program ini dapat dibuat dengan menggunakan metode penggambaran flowchart sebagai berikut:
  - Conceptual flowchart, menggambarkan alur pemecahan masalah secara global
  - Detail flowchart, menggambarkan alur pemecahan masalah secara rinci.
- e. Flowchart proses merupakan diagram alir yang menggambarkan rekayasa industrial dengan cara merinci dan menganalisis langkah-langkah dalam suatu prosedur atau sistem.

### 3.3.2 Pentingnya Penggunaan Flowchart

Flowchart memiliki peranan penting bagi perusahaan di berbagai bidang. Keberadaan flowchart digunakan oleh perusahaan sebagai acuan untuk menjalankan suatu proses atau aktivitas bisnis. Flowchart juga memiliki fungsi yang cukup penting didalam proses pengembangan sebuah program. Penggunaan flowchart menjadi sangat penting karena beberapa alasan sebagai berikut:



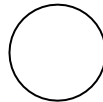
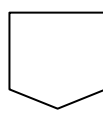
- a. Flowchart dapat memperjelas proses yang kompleks  
Sebuah program yang rumit dan kompleks pasti sulit untuk dipahami. Pengembang program membutuhkan flowchart untuk berkomunikasi dengan pelanggannya. Pelanggan merupakan orang awam yang perlu dijelaskan tentang algoritma dari berjalannya sebuah program. Dengan flowchart, maka kerumitan akan menjadi lebih sederhana, pelanggan lebih mudah menerima penjelasan yang disampaikan.
- b. Flowchart dapat meningkatkan efisiensi  
Flowchart dapat digunakan untuk meninjau ulang alur kerja sebuah program. Sebuah program masih dapat diperbaiki atau dioptimalkan dengan penyederhanaan proses maupun pengalihan alur melalui flowchart. Dengan demikian, flowchart dapat memberikan efisiensi dalam memperbaiki dan mengoptimalkan sebuah program.
- c. Flowchart dapat mengetahui proses pada program yang dijalankan  
Dengan menggunakan flowchart, pengembang atau pelanggan dapat melihat bagaimana sebuah proses berjalan ketika program mulai diaktifkan. Hal ini dikarenakan simbol - simbol flowchart yang dapat memberikan keterangan secara jelas dan detil mengenai apa yang terjadi didalam sebuah program. Misalnya, flowchart untuk perhitungan harga diskon, maka akan terdapat simbol input yang berisi harga awal, besaran diskon dan juga ada simbol proses yang digunakan untuk melakukan perhitungan harga diskon, serta terdapat simbol output untuk menampilkan harga diskon.
- d. Flowchart dapat digunakan sebagai acuan proses pengembangan  
Sebelum melakukan koding pada program, para tim pengembang membuat flowchart terlebih dulu. Notasi flowchart digunakan dalam proses perancangan sebuah program. Dalam penerapannya, flowchart ini digunakan sebagai dasar atau acuan untuk melakukan koding, sehingga pengembang akan lebih mudah memahami alur algoritma dari program yang akan dikembangkan.

### 3.3.3 Simbol Flowchart

Flowchart adalah penggambaran algoritma secara diagram yang bertujuan untuk menyelesaikan masalah dengan menggunakan sekumpulan simbol. Simbol - simbol flowchart dapat dibagi menjadi 3 kelompok, yaitu flow direction symbols, processing symbols, dan input/output symbols. Setiap simbol memiliki fungsinya masing-masing. Berikut uraian dan penjelasan simbol - simbol flowchart:

a. Flow direction symbols



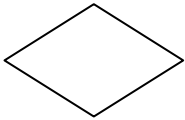



Simbol - simbol yang termasuk dalam kelompok flow direction, merupakan simbol yang digunakan untuk menghubungkan simbol satu dengan yang lain. Simbol ini disebut juga dengan connecting line. Berikut adalah simbol - simbol flow direction beserta fungsinya:

Simbol	Fungsi
Simbol Arus 	Menyatakan jalannya arus suatu proses
Simbol komunikasi link 	Menyatakan transmisi data dari satu lokasi ke lokasi lain
Simbol connector 	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
Simbol <i>offline connector</i> 	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda





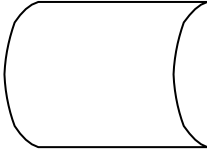


b. *Processing symbols*

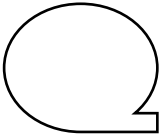
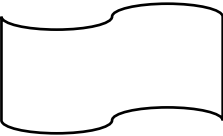
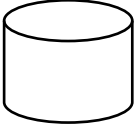
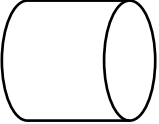
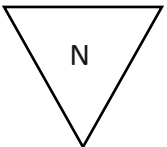
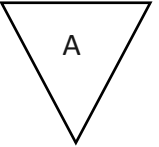
Simbol - simbol yang termasuk dalam kelompok *processing symbols*, merupakan simbol yang digunakan untuk menunjukkan jenis operasi pengolahan dalam suatu proses / prosedur. Berikut adalah simbol - simbol *processing symbols* beserta fungsinya:

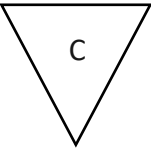
Simbol	Fungsi
Simbol Proses 	Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer
Simbol manual 	Menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer
Simbol decision 	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya / tidak
Simbol <i>predefined process</i> 	Menyatakan suatu fungsi atau sub program
Simbol preparation 	Menyatakan nilai awal variable atau digunakan untuk algoritma perulangan
Simbol <i>terminal</i> 	Menyatakan permulaan (awal) atau akhir suatu program

c. *Input / Output symbols*

Simbol - simbol yang termasuk dalam kelompok *input / output symbols*, merupakan simbol yang digunakan untuk menunjukkan jenis peralatan yang digunakan sebagai media input dan output. Berikut adalah simbol - simbol *input / output symbols* beserta fungsinya:

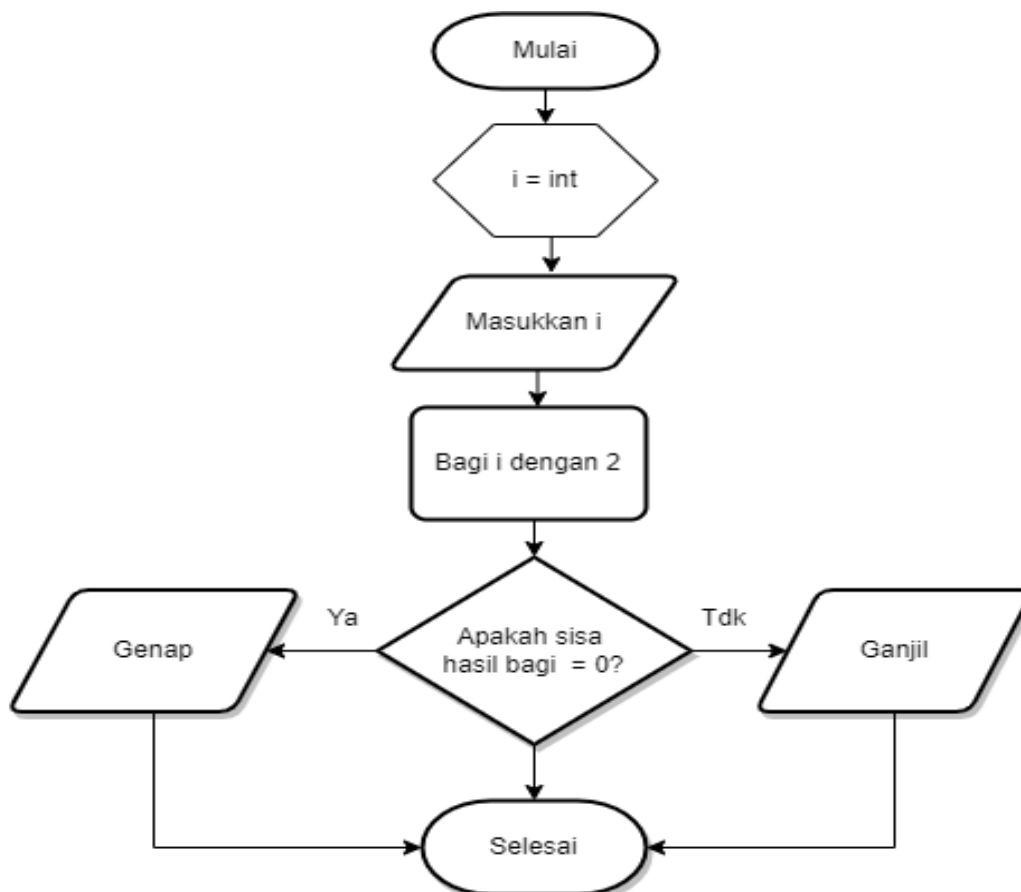
Simbol	Fungsi
<p>Simbol <i>input/output</i></p> 	Menyatakan proses input atau output tanpa tergantung jenis peralatannya
<p>Simbol <i>punched card</i></p> 	Menyatakan input berasal dari kartu atau output ditulis ke kartu
<p>Simbol <i>disk storage</i></p> 	Menyatakan input berasal dari disk atau output disimpan ke disk
<p>Simbol document</p> 	Mencetak keluaran dalam bentuk dokumen (melalui printer)
<p>Simbol display</p> 	Mencetak keluaran dalam layar monitor

Simbol	Fungsi
Simbol <i>magnetic tape</i> 	Menyatakan output yang menggunakan pita magnetic
Simbol <i>punched tape</i> 	Menyatakan input dan output menggunakan kertas berlubang
Simbol <i>hardisk storage</i> 	Menyatakan input output menggunakan hard disk
Simbol <i>magnetic drum</i> 	Menyatakan input output menggunakan drum magnetic
Simbol <i>offline storage</i> 	Menyatakan file non komputer yang diarsip urut angka
Simbol <i>offline storage</i> 	Menyatakan file non komputer yang diarsip urut huruf

Simbol	Fungsi
Simbol <i>offline storage</i> 	Menyatakan file non komputer yang diarsip urut tanggal

Contoh flowchart:

Setelah mengetahui berbagai simbol yang dapat digunakan untuk membuat flowchart, maka selanjutnya dibagian ini akan diberikan contoh penggunaan simbol - simbol tersebut untuk membuat flowchart sederhana.



Flowchart diatas merupakan flowchart yang digunakan untuk menentukan apakah bilangan yang dimasukkan merupakan bilangan ganjil atau genap. Flowchart diatas diawali dengan memasukkan suatu bilangan kemudian bilangan dibagi dengan 2, selanjutnya dilakukan proses pengecekan apakah bilangan tersebut ganjil atau genap dengan menggunakan simbol decision. Apabila bilangan tersebut dibagi dengan 2 hasilnya 0 atau habis tidak tersisa, maka bilangan tersebut termasuk bilangan genap, dan sebaliknya apabila masih ada sisanya maka bilangan tersebut merupakan bilangan ganjil.

Pada sub bab selanjutnya, akan dibahas beberapa penerapan notasi algoritma pada beragam kasus bisnis sederhana. Notasi algoritma akan dijabarkan dalam bentuk notasi kalimat deskriptif, pseudocode, dan flowchart.

### 3.4 Implementasi Notasi Algoritma pada Beragam Kasus Bisnis

Pada sub bab ini akan diberikan beberapa contoh penerapan notasi algoritma di berbagai studi kasus bisnis. Setiap studi kasus akan diselesaikan menggunakan notasi algoritma penulisan kalimat deskriptif, pseudocode, dan flowchart.

#### 3.4.1 Algoritma perhitungan total pembayaran

Membuat algoritma perhitungan total pembayaran pembelian sepatu merk Nevada. Dengan kondisi terdapat diskon sepatu Nevada sebesar 50%, jika pelanggan membeli sepatu merk tersebut dengan total pembelanjaan lebih dari sama dengan Rp. 500.000,-. Harga mula sepatu yang akan dibeli oleh pelanggan adalah Rp.256.000,-, dan pelanggan membeli sepatu merk Nevada sebanyak 2 pasang, sehingga untuk menghitung total pembayaran yang harus dibayar pelanggan dapat menggunakan algoritma sebagai berikut:

Notasi kalimat deskriptif:

```
Algoritma_hitungTotalPembayaran
{menghitung total pembayaran pembelian sepatu merk Nevada dengan memasukkan
harga mula produk dan besaran diskon}
Deklarasi:
hargaMula, besaranDiskon(%), hargaDiskon = int
```

Deskripsi:

```
Input namaProduk, hargaMula, besaranDiskon(%)
Input jumlahproduk
Hitung pembayaran = hargaMula produk * jumlahproduk
Jika pembayaran >= 500.000 maka diberikan discount 50%, Selain itu tidak
mendapat discount
Hitung total pembayaran = pembayaran - discount
Cetak total pembayaran.
```

Notasi pseudocode:

```
PROGRAM PerhitunganTotalPembayaran
```

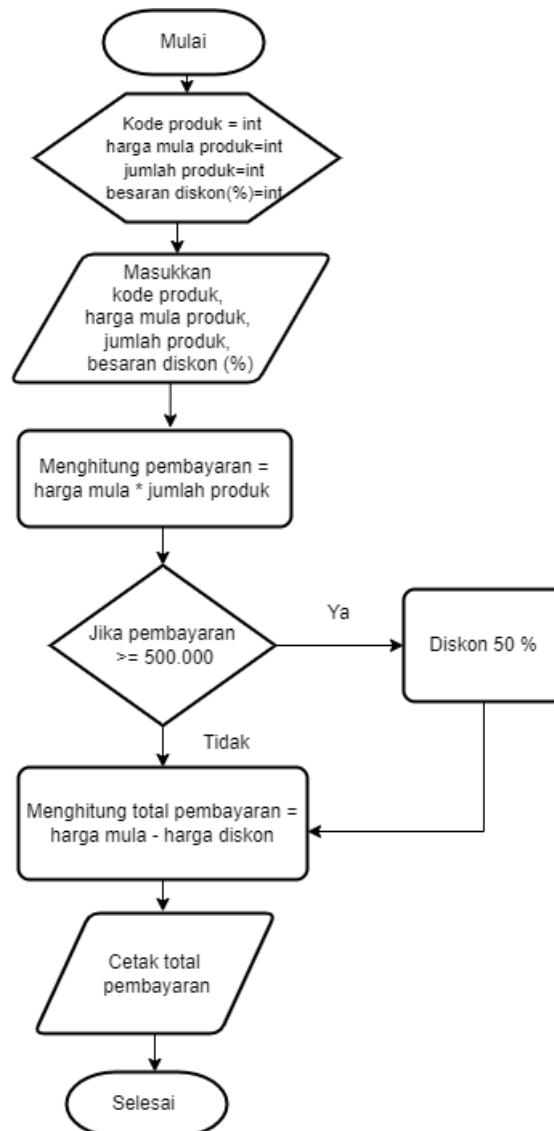
```
DEKLARASI
```

```
    k = string {nama produk}
    m = int {harga mula produk}
    n = int {jumlah produk}
    d = int {besaran diskon}
    p = int {hitung pembayaran}
    s = int {hitung harga discount}
    H = int {hitung total pembayaran}
```

```
ALGORITMA
```

```
read (k)      {masukkan kode produk}
read (m) {masukkan harga mula produk}
read (n)      {masukkan jumlah produk}
read(d)       {masukkan besaran discount (%) }
p ← m*n
Jika p >= 500.000
maka H ← p - (p*d)
write(H)      {tampilkan total pembayaran}
End
```

Notasi flowchart:



### 3.4.2 Algoritma mengecek ketersediaan barang alat tulis di gudang

Algoritma cek ketersediaan map bening yang ada ditoko. Dengan kondisi jika jumlah map bening kurang dari sama dengan 40 buah, maka akan dilakukan pembelian kepada supplier. Berdasarkan studi kasus, maka dapat diselesaikan dengan notasi algoritma sebagai berikut:

Notasi kalimat deskriptif:

Algoritma\_HitungKetersediaanBarang

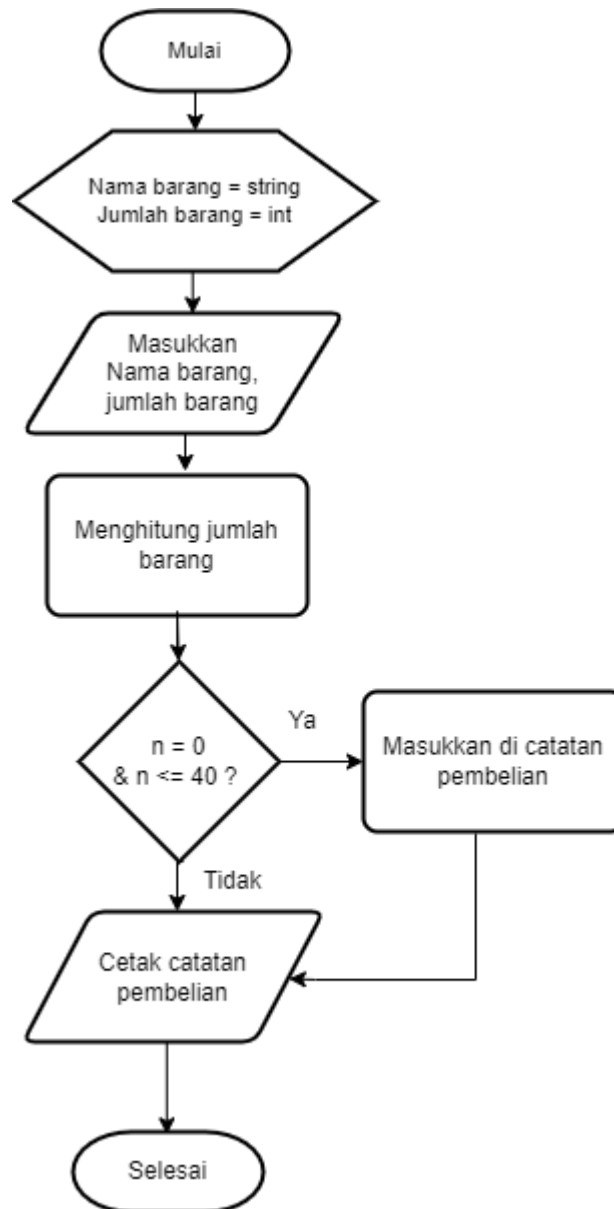


```
{menghitung jumlah ketersediaan barang}
Deklarasi:
Nama barang = string, jumlah barang = int, catatan pembelian = string
Deskripsi:
Input Nama barang
Melakukan cek ketersediaan barang
Hitung jumlah barang
Jika jumlah barang <= 40 buah, maka dimasukkan dalam catatan pembelian
Cetak catatan pembelian
Catatan pembelian siap digunakan acuan untuk membeli ke supplier.
```

## Notasi pseudocode

```
PROGRAM CekKetersediaanStok
DEKLARASI
    m=string (nama barang)
    n=int {jumlah barang}
    c=string {catatan}
ALGORITMA
read (m)    {masukkan nama barang}
read (n)   {masukkan jumlah barang}
jika n = 0 dan n <= 40
Maka c = m
write (c) {cetak Catatan pembelian}
End
```

Notasi flowchart



### 3.4.3 Algoritma menghitung gaji karyawan

Algoritma menghitung gaji yang akan diberikan kepada karyawan, dimana gaji yang akan diterima mencakup gaji pokok, tunjangan, dan transportasi. Kasus tersebut dapat diselesaikan dengan algoritma sebagai berikut:

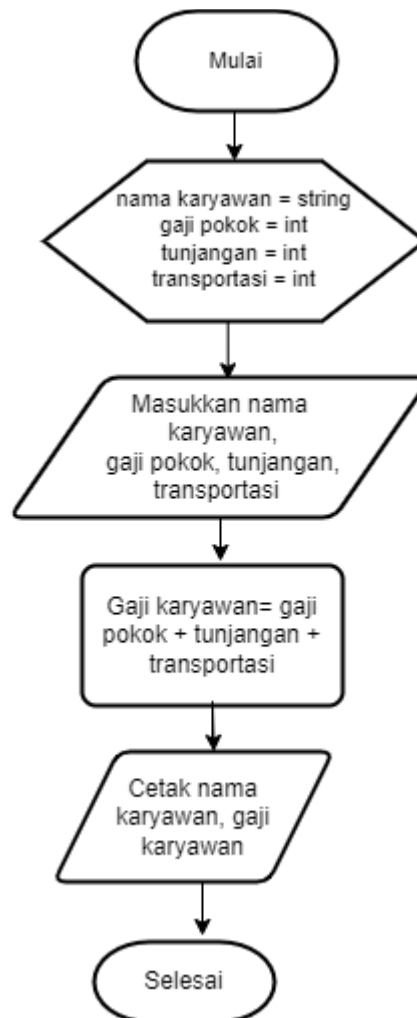
## Notasi kalimat deskriptif:

```
Algoritma_hitungGaji
{menghitung gaji yang akan diberikan kepada karyawan}
Deklarasi:
namakaryawan = string, gajipokok = int, tunjangan = int, transportasi = int,
gajikaryawan = int
Deskripsi:
Input namakaryawan
Input gajipokok
Input tunjangan
Input transportasi
gajikaryawan = gajipokok+tunjangan+transportasi
Cetak namakaryawan diikuti dengan besar gajikaryawan
```

## Notasi pseudocode

```
PROGRAM gajikaryawan
DEKLARASI
    a=string (nama karyawan)
    b=int {gaji pokok}
    c=int {tunjangan}
    d=int {transportasi}
    e=int {gaji karyawan}
ALGORITMA
read (a) {masukkan nama karyawan}
read (b) {masukkan gaji pokok}
read (c) {masukkan tunjangan}
read (d) {masukkan transportasi}
e ← b + c + d
write (a,e) {cetak namakaryawan, gajikaryawan}
End
```

Notasi flowchart



---

## POST TEST

### Soal tentang Materi Penulisan dan Notasi Algoritma Pemrograman

1. Jelaskan apa itu pseudocode dan flowchart!
2. Jelaskan keuntungan dari penggunaan notasi pseudocode dan flowchart!
3. Sebutkan 5 simbol flowchart yang Anda ketahui dan jelaskan fungsi dari ke-5 simbol tersebut!
4. Tulislah algoritma menghitung keliling lingkaran dengan menggunakan notasi kalimat deskriptif, pseudocode dan flowchart!
5. Sebuah bis melaju dengan kecepatan  $x$  km/jam. Jika bis tersebut berjalan selama  $t$  jam, maka tulislah algoritma untuk menghitung jarak ( $s$ ) yang telah ditempuh bis! Rumus menghitung jarak ( $s$ ) adalah kecepatan  $\times$  waktu!