

FLOWCHART DAN PSEUDOCODE



Overview

Algoritma dapat dituliskan ke dalam berbagai bentuk, namun struktur yang rapi dan mengikuti aturan tertentu akan membuat algoritma lebih mudah untuk dibaca dan dipahami. Selanjutnya, algoritma yang telah tersusun rapi akan diimplementasikan ke bahasa pemrograman.



Tujuan

- ☑ Mengenal bentuk pengambilan keputusan menggunakan flowchart
- ☑ Mengenal operasi boolean
- ☑ Mengenal bentuk pengulangan menggunakan flowchart
- ☑ Memahami tujuan penggunaan pseudocode dalam menyusun algoritma

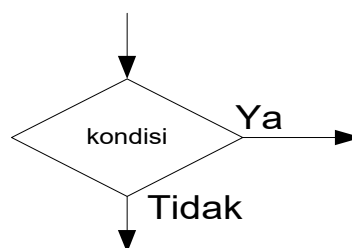
2.1 Flowchart

Seperti telah dijelaskan pada bab sebelumnya bahwa flowchart digunakan untuk menggambarkan algoritma atau proses. Flowchart disusun menggunakan simbol-simbol, maka dapat memberikan gambaran yang efektif dan jelas tentang prosedur logika.

Dalam hal melakukan koreksi atau analisis dari suatu permasalahan, flowchart dapat dengan mudah untuk dilihat dan dikomunikasikan. Hal ini dikarenakan flowchart disusun atas simbol-simbol yang mengikuti suatu standar tertentu.

2.1.1 Pengambilan Keputusan

Pengambilan keputusan perlu dilakukan apabila harus menentukan satu pilihan dari (minimal) dua pilihan yang ada. Dalam hal mengambil keputusan, perlu diketahui kondisi yang sedang dihadapi. Kondisi ini bisa berupa pernyataan boolean atau proses perbandingan. Dalam flowchart, simbol yang digunakan untuk pengambilan keputusan adalah berbentuk belah ketupat.



Gambar 2.1 Simbol pengambilan keputusan

Simbol pengambilan keputusan hanya memiliki satu buah input dan dua buah output yang digunakan untuk memfasilitasi hasil dari pengujian kondisi, yaitu “Ya” atau “Tidak”, “True” atau “False”.

Dalam melakukan pengujian kondisi, terdapat beberapa notasi yang dapat digunakan, misalnya menggunakan notasi relasional :

Tabel 2.1 Notasi relasional

>	Lebih besar dari
<	Kurang dari
≥	Lebih besar atau sama dengan
≤	Kurang dari atau sama dengan
<>	Tidak sama dengan

Dalam proses pengambilan keputusan, kadang kala terdapat beberapa syarat sekaligus. Untuk menangani hal ini dapat digunakan ekspresi aljabar boolean. Aljabar boolean merupakan kalkulus logika yang digunakan untuk menentukan nilai kebenaran dari suatu ekspresi logika. Teknik aljabar ini dikembangkan oleh George Boole pada tahun 1930an, sebagai penghargaan atas penemuannya maka aljabar ini diberi nama sesuai dengan nama belakang beliau.

Dalam aljabar boolean terdapat tiga buah operasi dasar, yaitu : AND, OR, NOT ketiga-tiganya dapat digunakan secara independen atau dapat digunakan sekaligus. Keluaran (output) dari aljabar ini adalah nilai benar (TRUE) atau salah (FALSE).

Berikut ini adalah tabel yang menunjukkan ketiga hasil operasi aljabar boolean :

Tabel 2.2 Tabel Kebenaran AND

<i>X</i>	<i>Y</i>	<i>Hasil</i>
T	T	T
T	F	F
F	T	F
F	F	F

Tabel 2.3 Tabel Kebenaran OR

<i>X</i>	<i>Y</i>	<i>Hasil</i>
T	T	T
T	F	T
F	T	T
F	F	F

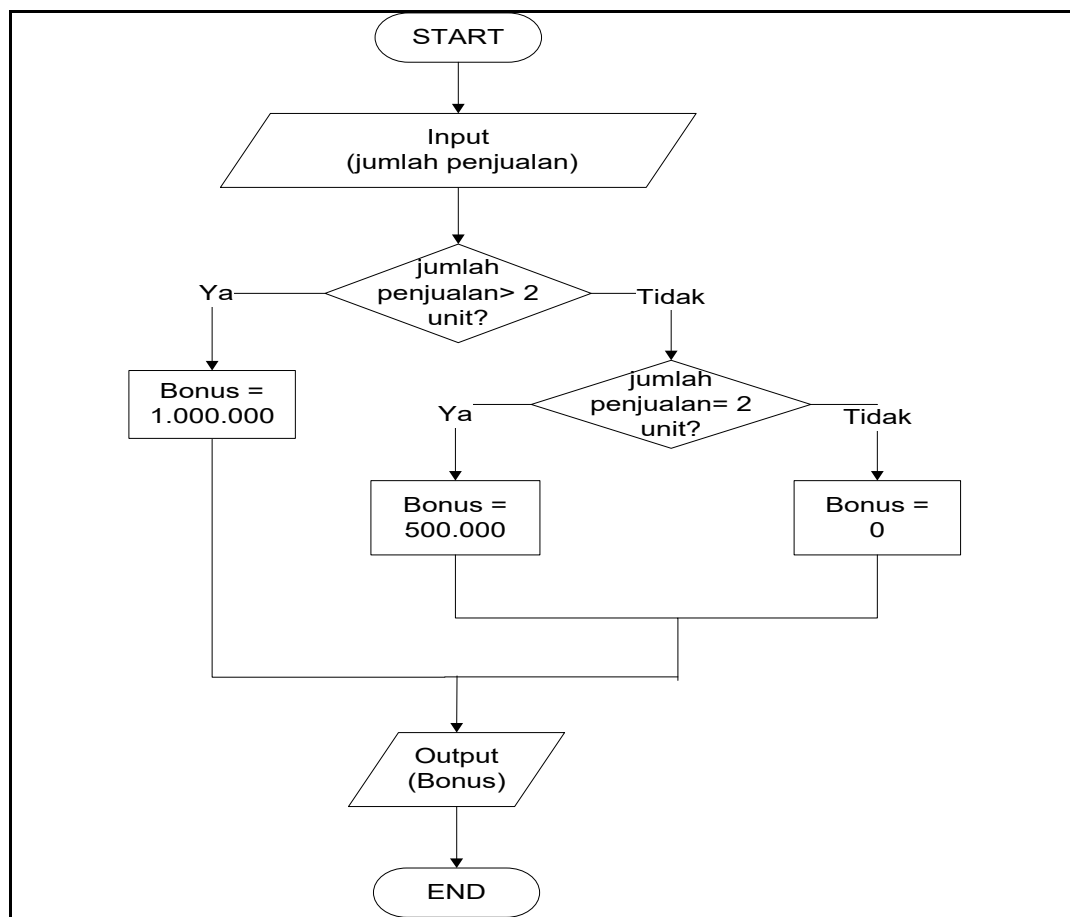
Tabel 2.4 Tabel Kebenaran XOR

<i>X</i>	<i>Hasil</i>
T	F
F	T

Contoh 2.1

Pemimpin sebuah perusahaan otomotif perlu menentukan besarnya bonus yang akan diberikan kepada para pegawainya yang bekerja sebagai account executive. Jika terdapat pegawai yang dalam bulan ini telah menjual mobil lebih dari dua unit, maka akan mendapatkan bonus sebesar Rp 1.000.000,- kemudian pegawai yang bisa menjual mobil tepat dua buah maka, akan mendapatkan bonus Rp 500.000,- namun jika pegawai yang dalam bulan ini penjualannya kurang dari dua unit maka, pegawai tersebut tidak mendapatkan bonus.

Jika kita gambarkan persoalan di atas menggunakan flowchart maka, akan menjadi seperti berikut :



Gambar 2.2 Flowchart penghitungan bonus

2.1.2 Pengulangan Proses

Dalam beberapa kasus, seringkali terdapat proses yang harus dilakukan secara berulang-ulang, sebagai contoh yang paling sederhana adalah proses berjalan kaki. Untuk bisa mencapai tujuan, kita harus melangkahakan kaki secara berulang-ulang supaya dapat menempuh jarak tertentu dan akhirnya sampai tujuan.

Pada kasus yang berhubungan dengan pengolahan informasi menggunakan komputer, terdapat proses-proses yang harus dilakukan secara berulang, mulai dari input data, proses dan output. Program yang baik adalah program yang bisa mengoptimalkan kinerja komputer, dengan cara menggunakan kembali program atau sekumpulan program dengan proses tertentu. Atau dengan kata lain terdapat bagian program yang dapat dipanggil/digunakan secara berulang-ulang. Hal ini akan mempermudah pekerjaan programmer dalam menghasilkan solusi.

Contoh 2.2

Seorang staff IT diminta untuk menampilkan data dari sebuah tabel dimana di dalamnya terdapat seratus baris data. Jika staff tersebut harus menampilkan satu per satu, tentunya akan membutuhkan banyak kode program dan program akan menjadi tidak efektif. Bagaimana cara menyelesaikan persoalan staff IT tersebut?

Solusi:

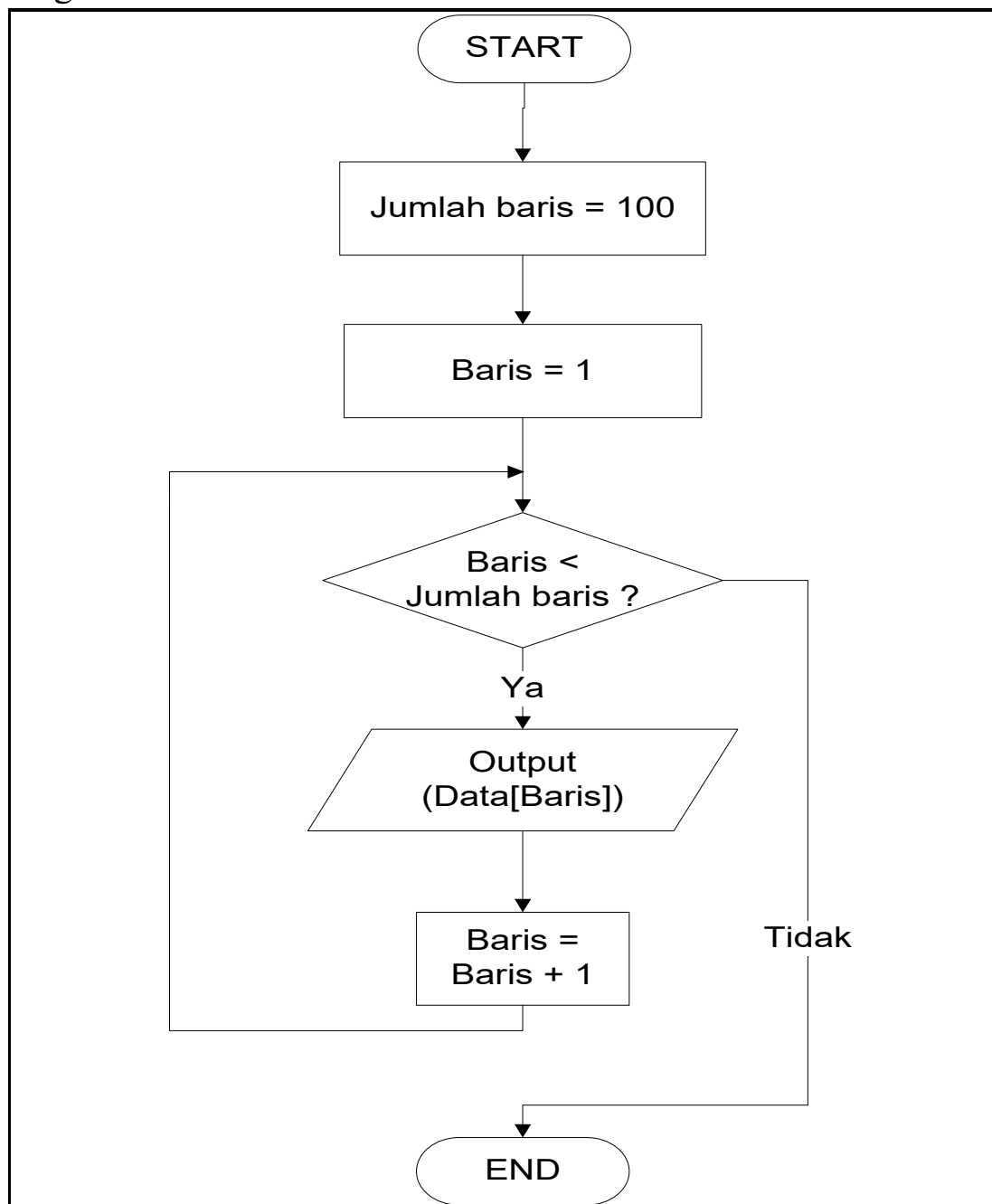
Dalam kasus ini yang diminta adalah bagaimana menampilkan data sebanyak 100 baris tanpa harus menggunakan proses output sebanyak 100 kali. Metode yang digunakan adalah pengulangan.

Dalam proses pengulangan terdapat 3 (tiga) hal penting, yaitu:

- ↳ Inisialisasi (penentuan kondisi/nilai awal)
- ↳ Proses
- ↳ Kondisi berhenti

Untuk kasus menampilkan data, dapat ditentukan bahwa jumlah baris yang akan dibaca adalah 100. Baris akan dibaca mulai dari baris pertama (baris = 1). Proses yang dilakukan adalah membaca dan menampilkan isinya ke layar (output). Pembacaan akan berhenti jika baris yang dibaca sudah mencapai baris ke-100.

Jika digambarkan menggunakan flowchart maka, akan tampak sebagai berikut:



Gambar 2.3 Flowchart untuk menampilkan 100 baris data

Dari gambar dapat dilihat bahwa proses output data hanya muncul satu kali, namun karena proses output dilakukan secara berulang-ulang maka, pembacaan terhadap 10 baris data dapat dilakukan.

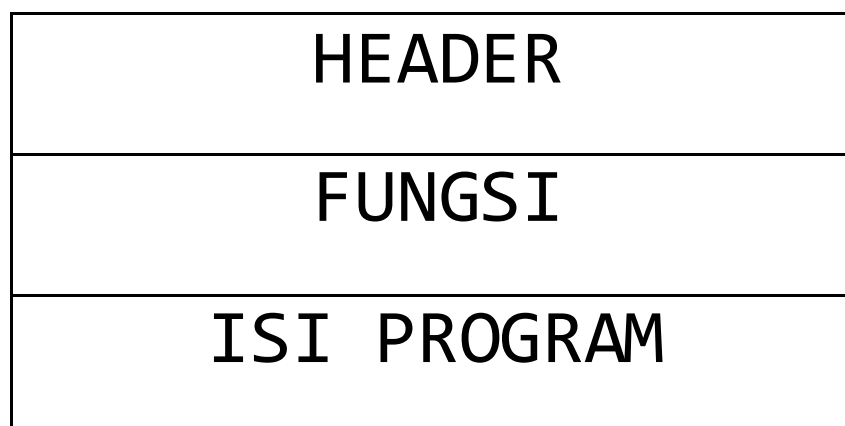
2.2 Pseudocode

Pada bab 1 telah dijelaskan sebelumnya mengenai keuntungan dalam menuangkan logika dan algoritma menggunakan *pseudocode*. Dalam menyelesaikan kasus yang besar dan kompleks, misalnya membuat aplikasi untuk menangani proses bisnis sebuah perusahaan maka, yang paling cocok digunakan dalam menuliskan algoritma adalah *pseudocode*.

Sesungguhnya tidak ada aturan baku dalam penulisan *pseudocode*, namun karena banyaknya bahasa pemrograman yang beredar saat ini maka, aturan penulisan pseudocode diarahkan untuk menyerupai aturan penulisan bahasa pemrograman tertentu. Dalam buku ini akan digunakan aturan penulisan pseudocode yang mendekati bahasa pemrograman Pascal.

2.3 Struktur algoritma

Struktur algoritma yang digunakan mengacu pada struktur pemrograman bahasa C++ yang terdiri dari 3 (tiga) bagian, yaitu :



Gambar 2.4 Struktur program

Pada bagian **Judul**, digunakan sebagai tempat untuk mencantumkan nama atau judul program. Terdapat aturan penulisan judul, yakni:

- ↳ Tidak diawali dengan angka atau karakter selain alphabet
- ↳ Tidak terdapat karakter spasi atau karakter selain alphabet kecuali karakter underscore ‘_’ (sebagai pengganti karakter spasi).

Contoh:

Algoritma berhitung;	Benar
Algoritma konversi bilangan;	Salah
Algoritma perhitungan_pajak;	Benar
Algoritma 2bilangan;	Salah
Algoritma *kecil;	Salah

Pada bagian **deklarasi**, digunakan sebagai tempat untuk mencantumkan variabel, konstanta, dan *record*. Mengingat cara eksekusi kode program dilakukan berurut dari atas ke bawah maka, deklarasi diletakkan di awal program setelah bagian judul. Hal-hal yang dideklarasikan pada bagian ini digunakan sebagai ‘reservasi’ alokasi *memory* untuk penyimpanan data dan akan digunakan selama program bekerja.



Rangkuman

- ☑ Flowchart digunakan untuk menggambarkan algoritma atau proses
- ☑ Dalam melakukan pengujian kondisi digunakan notasi relasional
- ☑ Simbol dalam flowchart yang digunakan untuk pengambilan keputusan adalah belah ketupat
- ☑ Analisa yang benar sangat dibutuhkan saat membuat flowchat
- ☑ Aljabar boolean merupakan kalkulus logika yang digunakan untuk menentukan nilai kebenaran dari suatu ekspresi logika
- ☑ Program yang baik adalah program yang bisa mengoptimalkan kinerja komputer, dengan cara menggunakan kembali program atau sekumpulan program dengan proses tertentu
- ☑ Pseudocode cocok digunakan untuk menuliskan algoritma dengan kasus yang kompleks dan berskala besar