

## BAB 1

### Pengenalan Algoritma



#### 1.1 Tujuan

1. Dapat menjelaskan dan menggunakan Algoritma, Flowchart dan Pseudocode dalam membuat program
2. Mengenal lingkungan dan perlengkapan untuk membuat Algoritma, Flowchart dan Pseudocode

## **1.2 Pengantar**

### **1.2.1 Algoritma**

Algoritma adalah sekumpulan instruksi yang merupakan penyelesaian masalah itu dinamakan program. Agar program dapat dilaksanakan oleh komputer, program tersebut harus ditulis dalam suatu bahasa yang dimengerti oleh komputer. Bahasa komputer yang digunakan dalam menulis program dinamakan bahasa pemrograman. Urutan langkahlangkah yang sistematis untuk menyelesaikan sebuah masalah dinamakan algoritma. Jadi algoritma adalah urutan logis pengambilan keputusan untuk pemecahan masalah. Kata logis merupakan kata kunci. Langkah-langkah tersebut harus logis, ini berarti nilai kebenarannya harus dapat ditentukan, benar atau salah. Urutan langkah-langkah yang sistematis dan logis untuk menyelesaikan suatu permasalahan.









Ciri-ciri algoritma:

- Langkah tersebut akan berhenti dan benar
- Mempunyai bentuk yang sederhana sehingga efektif
- Langkah-langkahnya jelas dan pasti

### **1.2.2 Flowchart**











Kadang-kadang perlu digambarkan bagaimana arus data dari algoritma yang sudah dibuat, terutama kalau algoritma sudah cukup kompleks. Untuk itu algoritma dapat disajikan dalam bentuk flowchart (diagram alir). Untuk membantu memahami nalar suatu program digunakan grafik/symbol yang mengekspresikan kegiatan-kegiatan dalam sebuah program.

Tabel 1. 1 Simbol Dasar Flowchart

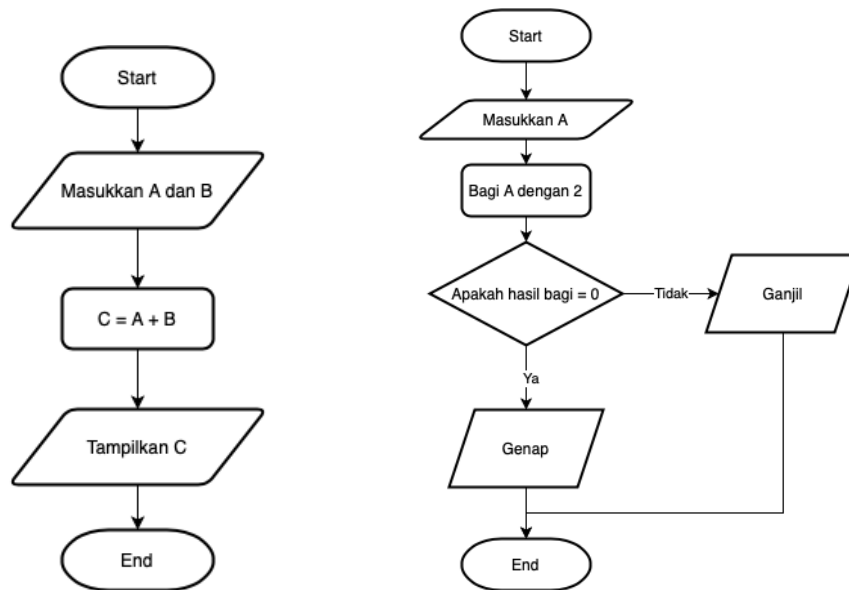
Simbol	Fungsi	Simbol	Fungsi
	Start/End		Input/Output
	Arah aliran		Konektor dalam satu halaman
	Konektor dengan halaman berbeda		Percabangan/ Pengambilan keputusan
	Proses		Sub proses

Tabel 1.1 merupakan daftar simbol-simbol flowchart dasar yang paling sering digunakan, sedangkan Tabel 1.2 adalah daftar simbol-simbol flowchart lain yang dapat digunakan untuk menggambarkan aliran suatu algoritma.

Tabel 1. 2 Simbol Flochart Lainnya

Simbol	Fungsi	Simbol	Fungsi
	Preparation		Dokumen
	Internal storage		Manual input
	Operasi manual		Merger
	Multiple document		Catatan/ Keterangan
	Penyimpanan data		Summing junction

Contoh flowchart untuk menghitung penjumlahan dan perhitungan bilangan genap dan ganjil dapat dilihat pada Gambar 1.1

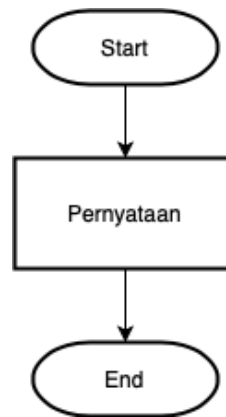


Gambar 1. 1 Flowchart Penjumlahan dan Hitungan Bilangan Ganjil Genap

Ada beberapa aturan yang harus dipahami dalam membuat flowchart, di antaranya adalah :

- Tidak ada kaidah yang baku.
- Flowchart = gambaran hasil analisa suatu masalah
- Flowchart dapat bervariasi antara satu pemrogram dengan pemrograman lainnya.
- Hindari pengulangan proses yang tidak perlu dan logika yang berbelit sehingga jalannya proses menjadi singkat.
- Jalannya proses digambarkan dari atas ke bawah dan diberikan tanda panah untuk memperjelas.

- Selalu dimulai dengan "Start/Begin" dan diakhiri dengan "Finish/End" seperti pada Gambar 1.2.



Gambar 1. 2 Penggunaan Start dan End

### 1.2.3 Pseudocode

Pseudocode berasal dari kata pseudo dan code yang artinya kode semu atau menyerupai kode program sebenarnya. Itu berarti pseudocode bukanlah kode program yang sebenarnya, melainkan menggunakan suatu bahasa pemrograman tertentu. Pseudocode berbeda dengan flowchart. Jika flowchart menggunakan simbol-simbol berbentuk gambar untuk menjelaskan alur logika berpikir sehingga dapat digunakan untuk membuat atau menjelaskan suatu program, pseudocode menggunakan bahasa sehingga tidak berupa gambar lagi. Namun, keduanya tetap memiliki tujuan yang sama yaitu membantu menuangkan alur pemikiran ke dalam bentuk tertulis. Dapat dilihat pada Tabel 1.3 untuk memperjelas perbedaan antara flowchart dan pseudocode.

Tabel 1. 3 Perbedaan Algoritma, Flowchart dan Pseudocode

<b>Algoritma</b>	<b>Flowchart</b>	<b>Pseudocode</b>
Menggunakan bahasa tingkat tinggi	Menggunakan simbol berbentuk gambar	Menggunakan bahasa tingkat tinggi yang menyerupai kode program
Tidak standar	Standar	Belum standar
Mudah dibaca	Mudah dimengerti	Mudah dibaca
Tidak dapat langsung dibuat programnya	Tidak dapat langsung dibuat programnya	Bisa langsung dibuat programnya
Masih berupa ide	Berupa rancangan	Sudah hampir berupa implementasi

Tabel 1.4 memberikan contoh algoritma untuk mendapatkan luas persegi panjang dengan pseudocode-nya.

Tabel 1. 4 Algoritma dan Pseudocode Luas Persegi Panjang

<b>Algoritma</b>	<b>Pseudocode</b>
Masukkan Panjang	Input Panjang
Masukkan lebar	Input Lebar
Nilai Luas adalah panjang x lebar	Luas <- panjang x lebar
Tampilkan Luas	Print Luas

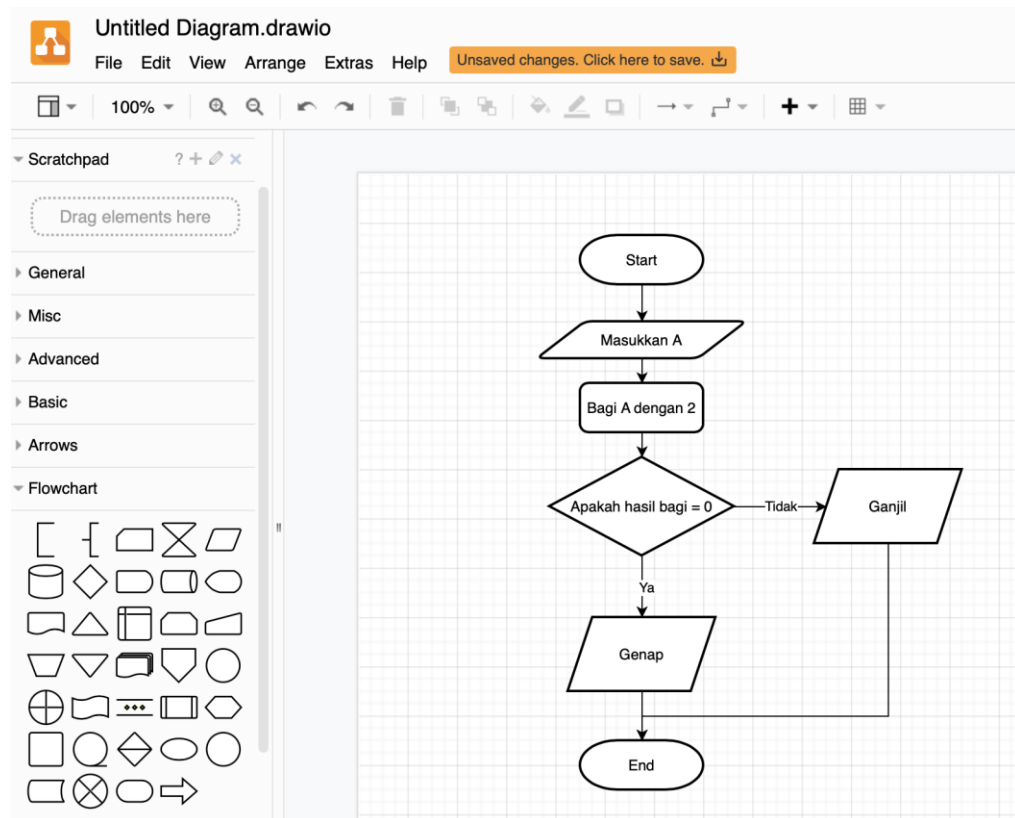
Berikut ini adalah contoh pseudocode untuk Hitungan Bilangan Ganjil Genap

```
1. STORE 'angka' with any value (number)
2.
3. IF 'angka' MOD 2 = 0 THEN
4.   DISPLAY "GENAP"
5. ELSE
6.   DISPLAY "GANJIL"
7. ENDIF
```

## 1.3 Kegiatan Praktikum

### 1.3.1 Kegiatan 1 : Mengetahui Aplikasi Drawio

1. Unduh aplikasi drawio pada alamat <https://www.diagrams.net> . Unduh sesuai dengan sistem operasi yang digunakan.
2. Pasang aplikasi dengan menjalankan file installer Drawio hasil download tadi, ikuti petunjuk yang ada. Jika mengalami kesulitan, mintalah panduan dari asisten praktikum. Aplikasi drawio juga dapat dijalankan secara online melalui halaman <https://draw.io>
3. Setelah selesai, jalankan aplikasi dan pada toolbox sebelah kiri pilihlah *flowchart* seperti pada Gambar 1.3.



Gambar 1. 3 Flowchart pada aplikasi draw.io

4. Mulailah mencoba menggunakan simbol-simbol flowchart dengan meng-klik pada icon yang ditampilkan di toolbox sebelah kiri
5. Jika ada presentasi dari asisten, perhatikan dan ikuti percobaan yang dilakukan.
6. Tulis analisis singkat mengenai hasil praktikum ini.

### 1.3.2 Kegiatan 2: Membuat Flowchart

1. Dengan menggunakan aplikasi Drawio, buatlah flowchart untuk algoritma menghitung:
  - Keliling lingkaran
  - Luas lingkaran
2. Tulis analisa singkat mengenai hasil praktikum ini



### **1.3.3 Kegiatan 3: Membuat Pseudocode**

1. Jalankan aplikasi pengolah kata (Word/WPS/Libre/Open Office) kemudian buatlah pseudocode untuk kasus pada Praktikum 1.3.2.
2. Tulis analisis singkat mengenai hasil praktikum ini.