

Representasi Algoritma: Diagram Alir & Pseudokode

Menuliskan langkah penyelesaian masalah secara terstruktur, mudah dipahami manusia, dan siap diubah menjadi kode program.



Tujuan Pembelajaran

Setelah mempelajari materi ini, kamu mampu:

- 1 Menjelaskan pengertian diagram alir (flowchart) dan pseudokode sebagai representasi algoritma
- 2 Membaca (menelusuri / tracing) diagram alir dan pseudokode
- 3 Menuliskan solusi program sederhana dalam bentuk diagram alir maupun pseudokode
- 4 Membedakan kapan menggunakan diagram alir dan kapan menggunakan pseudokode

KATA KUNCI

Algoritma

Diagram Alir

Pseudokode

Percabangan

Perulangan

Apa Itu Algoritma?

Algoritma adalah kumpulan instruksi yang **terstruktur dan terbatas**, yang dapat diimplementasikan untuk menyelesaikan suatu permasalahan tertentu — termasuk dalam bentuk program komputer.

Ibaratnya, algoritma adalah sebuah 'resep' untuk menyelesaikan masalah — sama seperti resep memasak suatu hidangan.

Menulis algoritma ≠ menulis program:

- Program ditulis agar dipahami **mesin**
- Algoritma ditulis agar dipahami **manusia**

2 CARA MEREPRESENTASIKAN ALGORITMA



Diagram Alir

Notasi visual berupa simbol & gambar



Pseudokode

Kode semu, mendekati bahasa pemrograman

Mengenai Diagram Alir

Diagram alir adalah notasi algoritma yang menggunakan **gambar atau simbol** sebagai penjelas dari suatu algoritma.

Dibuat dalam bentuk aliran simbol yang dapat ditelusuri dari titik permulaan (START) hingga titik akhir (END) menggunakan simbol standar ANSI/ISO.



Mudah dipahami — proses dapat dibaca & ditelusuri dengan cepat

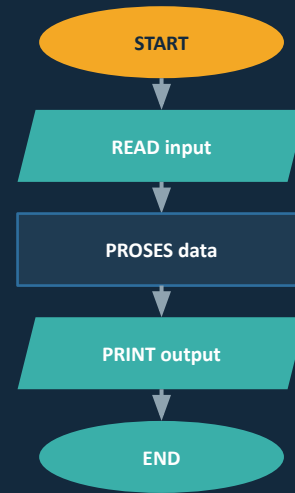


Berarah jelas — mengikuti garis alir (flowline) dari awal ke akhir



Modular — dapat memuat subprogram untuk proses yang kompleks

ALUR DASAR



Simbol-Simbol Diagram Alir

Tabel 2.1 — Simbol standar ANSI/ISO beserta maknanya



Garis Alir (Flowline)

Arah aliran program dari awal hingga akhir.



Terminator

Titik awal (START) atau titik akhir (END) suatu program.



Proses

Kegiatan komputasi, misalnya operasi aritmatika.



Percabangan

Titik keputusan; salah satu cabang dilalui berdasarkan kondisi.



Masukan / Keluaran

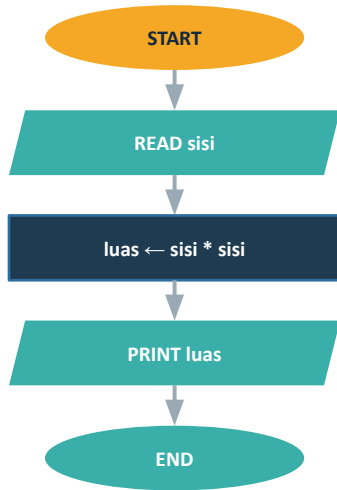
Titik program menerima data (input) atau menghasilkan informasi (output).



Subprogram

Kegiatan/proses lain yang telah didefinisikan sebelumnya.

Menghitung Luas Persegi



Cara Baca Diagram

1. Mulai dari START, ikuti arah panah.
2. Baca nilai sisi, simpan ke variabel sisi.
3. Hitung luas = sisi × sisi.
4. Cetak (PRINT) nilai luas, lalu selesai (END).

INPUT

sisi

PROSES

luas = sisi * sisi

OUTPUT

luas

PSEUDOKODE

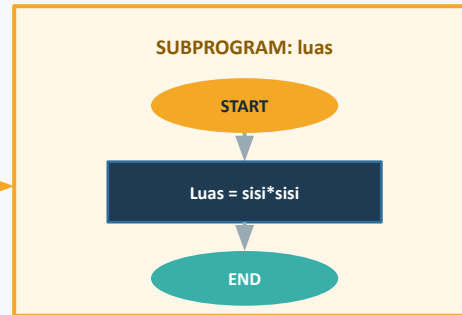
BACA sisi

luas ← sisi * sisi

CETAK luas

Subprogram: Luas Permukaan Kubus

Proses kompleks dapat memanggil subprogram yang sudah dibuat sebelumnya (abstraksi & dekomposisi).



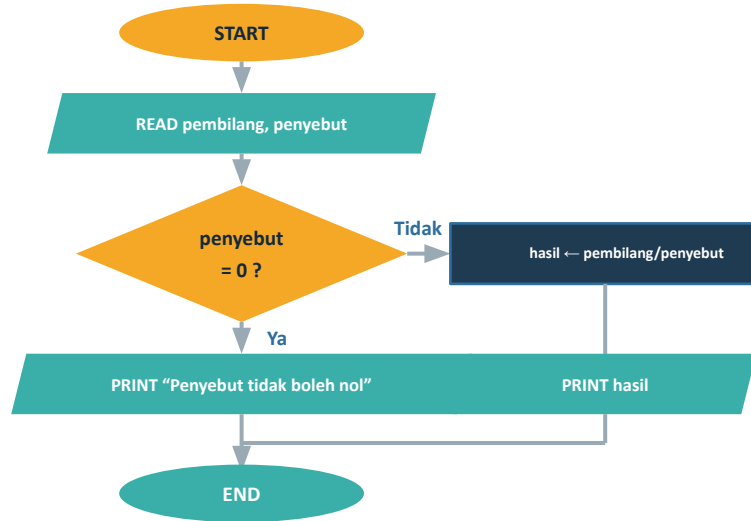
PSEUDOKODE — PROGRAM UTAMA & SUBPROGRAM

Utama Input: sisi | Proses: luas_permukaan $\leftarrow$ luas(sisi) * 6 |
Output: luas_permukaan

Subprogram Input: sisi | Proses: luas $\leftarrow$ sisi * sisi | Output:
luas

Percabangan: Membagi Bilangan

Simbol percabangan (diamond) mencegah pembagian dengan nol — salah satu cabang dipilih berdasarkan kondisi.



PSEUDOKODE

BACA pembilang, penyebut

IF penyebut = 0

CETAK "Penyebut tidak
boleh nol"

ELSE

hasil ← pembilang / penyebut

CETAK hasil

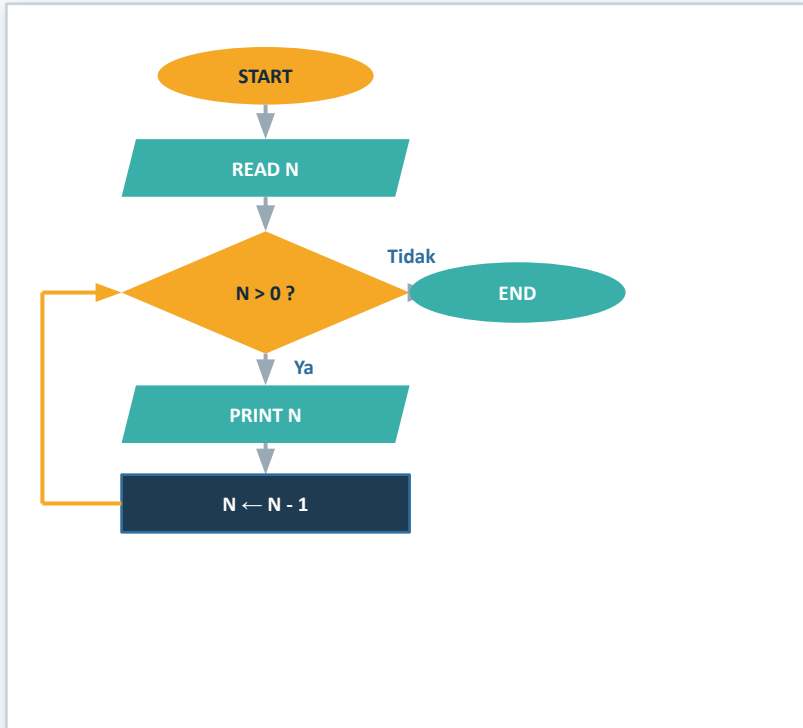
Kasus Uji

$10 \div 2 \rightarrow \text{hasil} = 5$

$8 \div 0 \rightarrow \text{"Penyebut tidak boleh nol"}$

Perulangan: Hitung Mundur dari N ke 1

Aliran dapat kembali ke simbol sebelumnya (loop) selama kondisi berhenti belum terpenuhi.



PSEUDOKODE

BACA N

WHILE N > 0

CETAK N

N ← N - 1

Contoh Penelusuran (N = 5)

N: 5 → cetak 5 → N=4

N: 4 → cetak 4 → N=3

N: 3 → cetak 3 → N=2

N: 2 → cetak 2 → N=1

N: 1 → cetak 1 → N=0

N: 0 → loop berhenti

Menelusuri (Tracing) Diagram Alir

Teknik membaca algoritma langkah demi langkah menggunakan lembar kerja 3 kolom: Masukan, Nilai Variabel, dan Keluaran.

1

Siapkan lembar kerja

Tuliskan data yang diolah & nama variabel;
kosongkan bagian keluaran

2

Telusuri dari awal

Ikuti diagram alir mulai dari simbol START

3

Simbol masukan (READ)

Coret data yang dibaca, letakkan ke variabel yang sesuai

4

Simbol proses (assignment)

Catat nilai baru hasil ekspresi ke variabel terkait

5

Simbol keluaran (PRINT)

Tuliskan isi variabel ke kolom keluaran

6

Terminator (END)

Penelusuran selesai — nilai keluaran akhir didapat

Mengenai Pseudokode

Pseudokode (kode semu) adalah bahasa buatan manusia yang **informal** untuk merepresentasikan algoritma menggunakan konvensi gabungan bahasa pemrograman.

Tujuannya menutupi kekurangan diagram alir dengan memberi langkah lebih detail & dekat dengan bahasa pemrograman.

PRINSIP PENULISAN

- Satu baris untuk satu pernyataan (statement)
- Indentasi menunjukkan hierarki blok pernyataan

KATA TERBATAS (RESERVED WORD)

BACA

CETAK

IF

THEN

ELSE

WHILE

FOR EACH

RETURN

Kata-kata ini ditulis dengan huruf kapital sebagai konvensi umum penulisan bahasa pemrograman.

Contoh pemanggilan fungsi:

```
largest ← L[0]
FOR EACH item in L, DO
  IF item > largest THEN
    largest ← item
```

Struktur Umum Pseudokode

Setiap pseudokode idealnya memuat tiga bagian: Input, Proses, dan Output — agar mudah diuji dan diubah menjadi kode program.



Contoh Lengkap — Algoritma Membagi Bilangan

Algoritma Membagi Bilangan

Input: Pembilang dan Penyebut

Proses:

BACA pembilang, penyebut

IF penyebut = 0

 PRINT "Penyebut tidak boleh nol"

ELSE

 hasil ← pembilang / penyebut

 CETAK hasil

Output: Hasil pembagian tercetak

Mengapa struktur ini penting?

- Memudahkan pembuatan **kasus uji** (test case)
- Format Input/Output harus konsisten agar **autograder** dapat menilai dengan benar
- Menjadi jembatan langsung menuju **kode program**

Diagram Alir vs Pseudokode



DIAGRAM ALIR

- Berbentuk visual / grafis dengan simbol standar
- Lebih mudah dipahami pemula & non-programmer
- Cocok untuk alur proses yang relatif sederhana
- Menjadi kurang praktis untuk algoritma yang panjang & kompleks
- Baik untuk presentasi/diskusi & dokumentasi visual

{ } PSEUDOKODE

- Berbentuk teks, mendekati bahasa pemrograman
- Lebih ringkas & cepat ditulis untuk logika kompleks
- Mudah menuliskan struktur bersarang (nested) & rekursif
- Lebih mudah dikonversi langsung menjadi kode program
- Tidak ada aturan baku (sifatnya informal)

Ayo Berlatih!

10.2-01-U

Menelusuri Diagram Alir

Telusuri 4 diagram alir yang telah dipelajari dengan kasus baru

10.2-02-U

Menelusuri Pseudokode

Terapkan teknik penelusuran yang sama pada pseudokode

10.2-03-U

Menulis Algoritma

Susun diagram alir/pseudokode dari deskripsi: mesin bakso otomatis

Rangkuman



Diagram Alir

Representasi visual algoritma memakai simbol standar ANSI/ISO (terminator, proses, percabangan, I/O, subprogram) yang dihubungkan garis alir.



Pseudokode

Representasi teks informal yang mendekati bahasa pemrograman, ditulis dengan struktur Input–Proses–Output serta reserved word (IF, WHILE, FOR, dst).



Penelusuran (Tracing)

Teknik membaca algoritma langkah-demi-langkah dengan mencatat nilai variabel — berlaku untuk diagram alir maupun pseudokode.



Memilih Representasi

Diagram alir unggul untuk visualisasi & pemula; pseudokode unggul untuk logika kompleks & konversi cepat ke kode program.

LATIHAN, 1 KELOMPOK = 2 SISWA

1. Buatlah algoritma dengan pseudocode dan flowchart untuk menyeduh segelas teh manis hangat, mulai dari menyiapkan bahan hingga siap diminum. (Bahan yang tersedia: teh celup, gula, air panas, gelas, dan sendok. Pastikan algoritmamumu memiliki input, proses, dan output yang jelas.)
2. Buatlah algoritma dengan pseudocode dan flowchart sederhana untuk proses penarikan uang tunai di mesin ATM, mulai dari memasukkan kartu hingga uang keluar atau transaksi ditolak. (Pertimbangkan kemungkinan: PIN salah, saldo tidak mencukupi, dan transaksi berhasil. Pastikan algoritmamumu mencakup ketiga kemungkinan tersebut)
3. Buatlah algoritma dengan pseudocode dan flowchart untuk menghitung luas persegi panjang, jika diketahui panjang dan lebarnya. (dalam 1 pseudocode dan 1 flowchart)
4. Buatlah algoritma dengan pseudocode dan flowchart untuk menentukan apakah sebuah bilangan bulat yang diinput merupakan bilangan genap atau ganjil.

Terima Kasih

Selamat berlatih menyusun diagram alir dan pseudokode untuk memecahkan masalahmu sendiri!
