

STA561 Pemrograman Statistika

Pengantar Algoritma dan Pemrograman

Dr. Agus M Soleh



agusms@apps.ipb.ac.id

Outline

- 1 Apa itu Algoritma dan Pemrograman?
- 2 Menggambarkan Algoritma
 - Diagram Alir (*Flowchart*)
 - Diagram Alir Proses (DFD)
- 3 Struktur Algoritma
 - Runtunan
 - Pemilihan/percabangan
 - Perulangan
- 4 Fungsi dan Prosedur
- 5 Teladan

Apa itu Algoritma dan Pemrograman?

- Algoritma adalah urutan instruksi atau langkah-langkah yang harus dilakukan untuk menyelesaikan masalah
- Setiap instruksi atau langkah dalam algoritma ini haruslah logis dan jelas
- Algoritma harus efisien dari segi tempat dan waktu
- Sekumpulan instruksi yang merupakan penyelesaian masalah disebut sebagai program
- Untuk menyusun program komputer ini diperlukan suatu bahasa yang disebut sebagai bahasa pemrograman
- Penulis program disebut sebagai pemrogram dan kegiatan merancang dan menulis program disebut pemrograman

Apa itu Algoritma dan Pemrograman?

Secara rinci berikut definisi dari Algoritma:

- Langkah-langkah yang dilakukan agar solusi masalah dapat diperoleh.
- Suatu prosedur yang merupakan urutan langkah-langkah yg berintegrasi.
- Suatu metode khusus yang digunakan untuk menyelesaikan suatu masalah yang nyata.

Apa itu Algoritma dan Pemrograman?

Syarat-syarat sebuah algoritma dikatakan baik:

- Algoritma harus efisien dimana prosesnya dapat diselesaikan secepat mungkin dengan frekuensi perhitungan yang sependek mungkin.
- Algoritma yang baik harus dengan mudah diimplementasikan ke perangkat komputer.
- Algoritma yang baik harus mudah dipahami.
- Akurasi tinggi adalah syarat lain dari algoritma yang baik.
- Semakin umum maka semakin baik.
- Algoritma yang baik memiliki langkah-langkah yang jelas dan detail.
- Algoritma yang baik juga harus bisa dikembangkan.

Apa itu Algoritma dan Pemrograman?

algoritma mempunyai lima ciri penting yang meliputi:

- 1 Finiteness (keterbatasan), algoritma harus berakhir setelah mengerjakan sejumlah langkah proses.
- 2 Definiteness (kepastian), setiap langkah harus didefinisikan secara tepat dan tidak berarti ganda.
- 3 Input (masukan), algoritma memiliki nol atau lebih data masukan (input).
- 4 Output (keluaran), algoritma mempunyai nol atau lebih hasil keluaran (output).
- 5 Effectiveness (efektivitas), algoritma harus sangkil (efektif), langkah-langkah algoritma dikerjakan dalam waktu yang wajar.

Apa itu Algoritma dan Pemrograman?

Sifat dari algoritma adalah sebagai berikut:

- 1 Tidak menggunakan simbol atau sintaks dari suatu bahasa pemrograman tertentu.
- 2 Tidak tergantung pada suatu bahasa pemrograman tertentu.
- 3 Notasi-notasinya dapat digunakan untuk seluruh bahasa manapun.
- 4 Algoritma dapat digunakan untuk merepresentasikan suatu urutan kejadian secara logis dan dapat diterapkan di semua kejadian sehari-hari

Menggambar Algoritma

- Algoritma dapat digambarkan dengan banyak notasi:
 - bahasa alamiah
 - pseudocode
 - diagram alir, dlsb
- Diperlukan suatu mekanisme atau alat bantu untuk menuangkan hasil pemikiran mengenai langkah-langkah penyelesaian masalah yang sistematis dan teratur
- Untuk bisa menyusun solusi diperlukan kemampuan problem-solving yang baik
- Salah satu tool (alat) yang dapat digunakan, yakni flowchart
- Skema lain yang dapat digunakan untuk menyusun algoritma adalah pseudocode
- Tujuan dari penggunaan pseudocode adalah supaya:
 - lebih mudah dibaca oleh manusia
 - lebih mudah untuk dipahami
 - lebih mudah dalam menuangkan ide/hasil pemikiran

Diagram Alir (*Flowchart*)

- Diagram alir (bahasa Inggris: flowchart), bagan alir, atau bagan arus adalah sebuah jenis diagram yang mewakili algoritme, alir kerja atau proses, yang menampilkan langkah-langkah dalam bentuk simbol-simbol grafis, dan urutannya dihubungkan dengan panah.
- Diagram alir digunakan untuk mendesain dan mendokumentasi proses atau program sederhana.

Diagram Alir (*Flowchart*)

Notasi dari flowchart adalah:












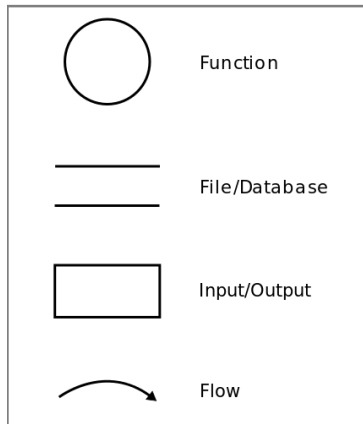
Gambar	Nama	Keterangan
	Garis Alir	Menunjukkan arah aliran algoritma, dari satu proses ke proses berikutnya.
	Terminal	Menunjukkan awal atau akhir sebuah proses.
	Proses / Langkah	Menyatakan kegiatan yang akan terjadi dalam diagram alir.
	Titik Keputusan	Proses / langkah di mana perlu adanya keputusan atau adanya kondisi tertentu. Di titik ini selalu ada dua keluaran untuk melanjutkan aliran kondisi yang berbeda.
	Masukan / Keluaran	Digunakan untuk mewakili data masuk, atau data keluar. Hanya bisa dimulai dari masukan menuju keluaran, bukan sebaliknya.
	Anotasi	Melambungkan komentar tentang suatu atau beberapa bagian dari diagram alir. Tentu saja, komentar tidak memiliki dampak apapun terhadap proses yang berlangsung.
	Predefined Process	Digunakan untuk menunjukkan suatu proses yang begitu kompleks, sehingga tidak bisa dijelaskan di diagram alir ini dan merujuk pada diagram alir yang terpisah.
	Persiapan / Inisialisasi	Menunjukkan operasi yang tidak memiliki efek khusus selain mempersiapkan sebuah nilai untuk langkah / proses berikutnya. Lambang ini juga digunakan untuk menggantikan titik keputusan yang biasanya berbentuk ketupat jika ingin menggunakan pengulangan pada kondisi tertentu.
	Konektor Dalam Halaman	Biasanya digunakan dalam pengulangan. Digunakan untuk menghubungkan satu proses ke proses lainnya, sama halnya seperti tanda panah. Boleh saja lebih dari satu proses yang mengarah kepadanya, namun hanya bisa menghasilkan satu keluaran. Sehingga diagram alir terlihat lebih rapi karena mengurangi tanda panah yang lalu lalang di dalam diagram alir.
	Konektor Luar Halaman	Terkadang, diagram alir tidak muat dalam satu halaman saja. Oleh karena itu, lambang ini berfungsi untuk menghubungkan satu proses ke proses lainnya, sama halnya seperti tanda panah, hanya saja untuk merujuk ke halaman yang berbeda.
	Kontrol / Inspeksi	Menunjukkan proses / langkah di mana ada inspeksi atau pengontrolan.

Diagram Alir Proses (DFD)

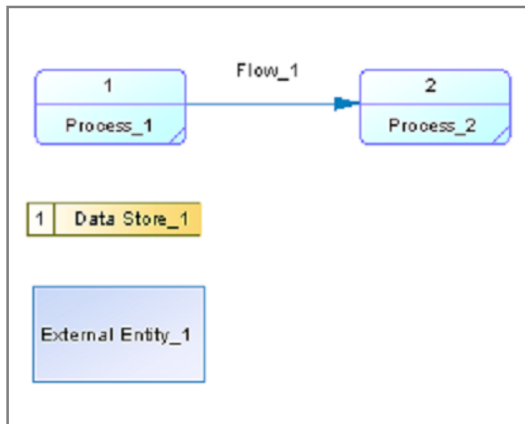
- Diagram alir data (*data flow diagram*, DFD) adalah suatu diagram yang menggambarkan aliran data dari sebuah proses atau sistem (biasanya sistem informasi).
- DFD juga menyediakan informasi mengenai luaran dan masukan dari setiap entitas dan proses itu sendiri.
- DFD tidak memiliki kontrol terhadap alirannya, tidak ada aturan mengenai keputusan maupun pengulangan.
- Komponen DFD terdiri dari:
 - Proses
 - Aliran Data
 - Warehouse
 - Entitas Eksternal

Diagram Alir Proses (DFD)

Notasi dari DFD adalah:



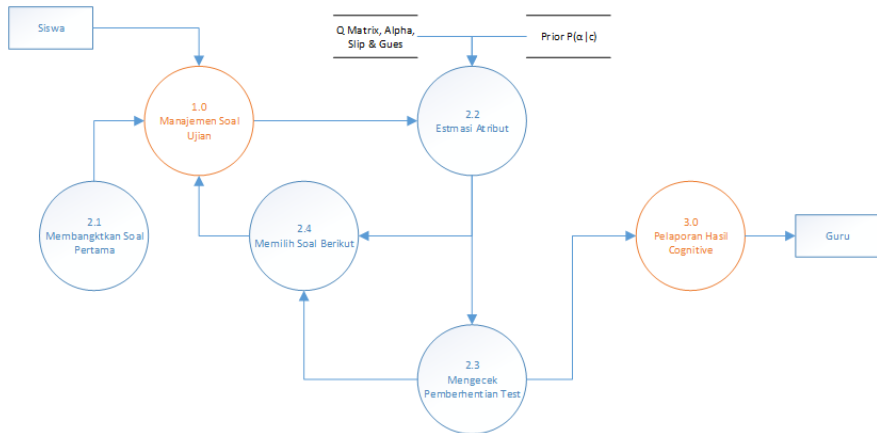
(a)



(b)

Diagram Alir Proses (DFD)

Ilustrasi dari DFD:



Struktur Algoritma

Struktur algoritma terbagi kedalam 3 bagian yaitu:

- Runtunan
- Pemilihan/percabangan
- Perulangan

Runtunan

- Program dengan permasalahan yang sederhana biasanya hanya berisi runtunan instruksi tanpa adanya proses pemilihan atau perulangan
- Contoh program menghitung luas lingkaran:
Untuk menghitung luas lingkaran menggunakan rumus:

$$Luaslingkaran = \pi \times r^2$$

- Algoritma jika dituliskan dalam runtunan bahasa natural adalah sebagai berikut:
 - 1 Mulai
 - 2 Inisialisasi nilai pi sebagai konstanta
 - 3 Masukkan Jari-jari
 - 4 Hitung luas
 - 5 Tampilkan luas lingkaran
 - 6 Selesai

Runtunan

Dalam bentuk pseudocode, algoritma di atas dapat berbentuk sebagai berikut:

```
PROGRAM Luas_Lingkaran
```

```
DEKLARASI
```

```
    pi      : 3.14
```

```
    r       : real
```

```
    luas    : real
```

```
ALGORITMA:
```

```
    baca (r)
```

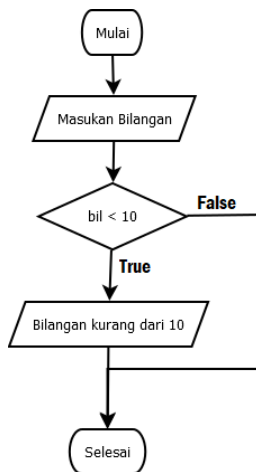
```
    luas <- pi * r * r
```

```
    tulis (luas)
```

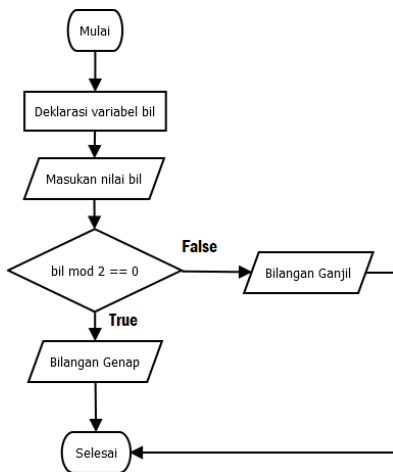

Pemilihan/percabangan

- Seringkali suatu instruksi hanya bisa di kerjakan jika memenuhi suatu persyaratan tertentu
- Struktur pemilihan memungkinkan kita melakukan aksi jika suatu syarat dipenuhi.
- Beberapa pemilihan/percabangan yang mungkin terjadi:
 - Contoh Satu Kasus
 - Contoh Dua Kasus
 - Contoh Tiga Kasus atau lebih

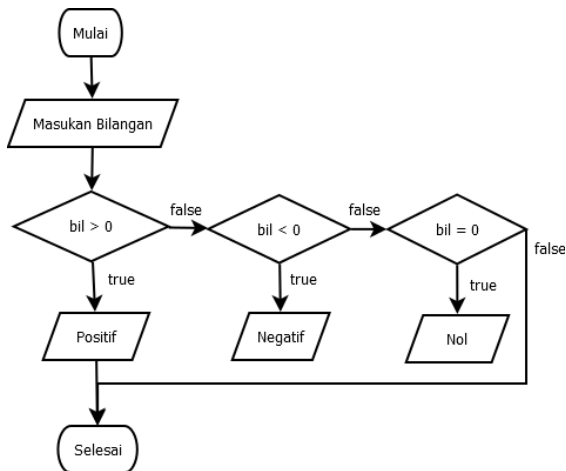
Pemilihan/percabangan: Contoh Satu Kasus



Pemilihan/percabangan: Contoh Dua Kasus



Pemilihan/percabangan: Contoh Tiga Kasus atau Lebih



Perulangan

- Perulangan atau sering disebut *repetition* atau *loop* merupakan pekerjaan yang dilakukan untuk menjalankan suatu perintah/instruksi dengan pola yang sama secara berulang-ulang.
- Ada tiga bentuk dari perulangan ini:
 - Perulangan For: jika jumlah perulangan sudah diketahui
 - Perulangan While: jika jumlah perulangan tidak diketahui dan mungkin tidak pernah ada
 - Perulangan Repeat: jika jumlah perulangan tidak diketahui dan minimal sekali dilakukan

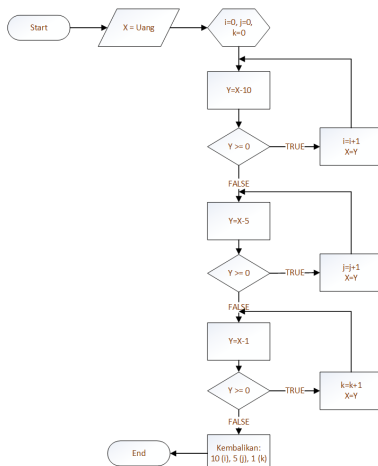
Fungsi dan Prosedur

- Prosedur dan fungsi adalah sebuah program terpisah (sub program diluar program utama) dalam blok tersendiri yang berfungsi untuk menyelesaikan masalah khusus.
- Menggunakan konsep modular yaitu: dengan memecah sebuah program yang rumit menjadi beberapa subprogram yang lebih sederhana.
- Dari segi efisiensi, dapat dilihat dimana prosedur dan fungsi cukup dituliskan sekali namun dapat dipanggil kapan saja atau digunakan berulang-ulang.
- Secara umum, fungsi dan prosedur memiliki banyak persamaan. Perbedaan dari keduanya:
 - suatu fungsi akan mengembalikan sebuah nilai, sedangkan prosedur tidak

Teladan

- Misalkan akan dibuat algoritma kembalian uang dari suatu *vending machine*. Pecahan Uang yang tersedia pada mesin tersebut terdiri dari pecahan 10, 5, dan 1.
- Untuk keperluan ini, kita akan menggunakan diagram alir untuk algoritamanya. Asumsi dalam algoritma ini adalah pecahan 10, 5, dan 1 yang tersedia di *vending machine* tidak terbatas.

Teladan



Teladan

Program R untuk algoritma di atas adalah sebagai berikut:

```
x <- 28 #sejumlah uang yang
      akan dipecah
```

```
i <- 0
```

```
y <- x-10
```

```
while(y>=0) {
```

```
  i <- i+1
```

```
  x <- y
```

```
  y <- y-10
```

```
}
```

```
j <- 0
```

```
y <- x-5
```

```
while(y>=0) {
```

```
  j <- j+1
```

```
  x <- y
```

```
  y <- y-5
```

```
}
```

```
k <- 0
```

```
y <- x-1
```

```
while(y>=0) {
```

```
  k <- k+1
```

```
  x <- y
```

```
  y <- y-1
```

```
}
```

```
print(paste("Uang 10
            sebanyak", i, "lembar"))
```

```
print(paste("Uang 5
            sebanyak", j, "lembar"))
```

```
print(paste("Uang 1
            sebanyak", k, "lembar"))
```

Akhir materi 1...