

Algoritma & Struktur Data

Kode MK: 0807003

Semester: Ganjil 2022/2023

Pertemuan Ke-1 & 2

Apa itu Algoritma?

Apa Definisi Algoritma?

“Urutan langkah-langkah untuk menyelesaikan persoalan” [MUN16]

“Deretan langkah-langkah komputasi yang mentransformasikan data masukan menjadi keluaran” [COR90]

“Deretan instruksi yang jelas untuk memecahkan persoalan, yaitu untuk memperoleh keluaran yang diinginkan dari suatu masukan dalam jumlah waktu yang terbatas” [LEV03]

Renungan: QS. An Nisa: 101 - 102

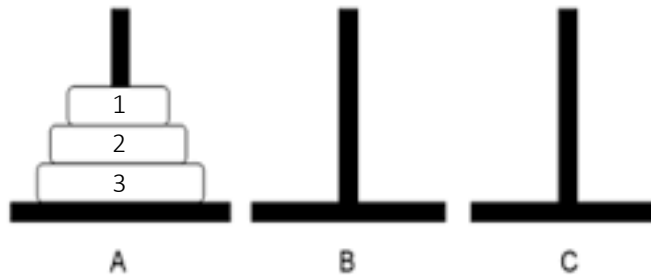
Dan apabila kamu bepergian di bumi, maka tidaklah berdosa kamu meng-qasar shalat jika kamu takut diserang orang kafir. Sesungguhnya orang kafir itu adalah musuh yang nyata bagimu. (101)

Dan apabila engkau (Muhammad) berada di tengah-tengah mereka (sahabatmu) lalu engkau hendak melaksanakan shalat bersama-sama mereka, maka hendaklah segolongan dari mereka berdiri (salat) besertamu dan menyandang senjata mereka, kemudian apabila mereka (yang shalat bersamamu) sujud (telah menyempurnakan satu rakaat) maka hendaklah mereka pindah dari belakangmu (untuk menghadapi musuh) dan hendaklah datang golongan yang lain yang belum shalat, lalu mereka shalat denganmu, dan hendaklah mereka bersiap-siaga dan menyandang senjata mereka. Orang kafir ingin agar kamu lengah terhadap senjatamu dan harta bendamu, lalu mereka menyerbu kamu sekaligus. Dan tidak mengapa kamu meletakkan senjata-senjatamu, jika kamu mendapatkan suatu kesusahan karena hujan atau karena kamu sakit, dan bersiap-siagalah kamu. Sungguh, Allah telah menyediakan azab yang menghinakan bagi orang-orang kafir itu. (102)

Instruksi & Aksi

Sebuah algoritma berisi sekumpulan **instruksi** yang menggambarkan langkah-langkah penyelesaian suatu persoalan. Instruksi adalah perintah untuk melakukan **aksi** tertentu.

Menara Hanoi



Jika piringan ada 3, berapa kali jumlah perpindahan?

Sumber gambar: <https://goo.gl/images/orMxmz>

PROGRAM Menara Hanoi

{Diberikan tiga buah batang besi (A, B, dan C) dan 3 cakram yang berada di batang A sedemikian sehingga cakram yang besar selalu berada di bawah cakram yang lebih kecil. Algoritma Menara Hanoi berusaha memindahkan cakram dari batang A ke B dengan bantuan batang C dengan syarat hanya satu cakram yang bisa dipindahkan dalam satu waktu dan cakram besar harus selalu berada di bawah cakram yang lebih kecil}

ALGORITMA:

1. Pindahkan cakram 1 ke batang B. {batang B berisi cakram 1}
2. Pindahkan cakram 2 ke batang C. {batang B masih berisi cakram 1}
3. Pindahkan cakram 1 ke batang C. {batang B tidak berisi cakram apapun}
4. Pindahkan cakram 3 ke batang B. {batang B berisi cakram 3}
5. Pindahkan cakram 1 ke batang A. {batang B masih berisi cakram 3}
6. Pindahkan cakram 2 ke batang B. {batang B berisi cakram 3 dan cakram 2 secara berurut dari bawah ke atas}
7. Pindahkan cakram 1 ke batang B. {batang B berisi cakram 3, cakram 2, dan cakram 1 secara berurut dari bawah ke atas}

Aksi "memindahkan cakram"

Sekumpulan instruksi

Karakteristik Algoritma

Menurut Donald E. Knuth dalam *Art of Computer Programming* [KNU73], algoritma harus memiliki 5 ciri penting sebagai berikut:

- ➔ Harus berhenti
- ➔ Setiap langkah harus didefinisikan tepat / tidak ambigu / bermakna ganda
- ➔ Memiliki 0 atau lebih masukan (*input*)
- ➔ Memiliki 0 atau lebih keluaran (*output*)
- ➔ Harus efisien / dapat dikerjakan dalam waktu yang masuk akal

Notasi Algoritma

Notasi I: menggunakan kalimat Deskriptif

Notasi II: menggunakan *flowchart*

Notasi III: menggunakan pseudo-code

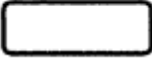


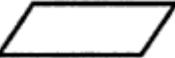

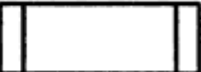


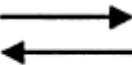
Notasi I: Kalimat Deskriptif

Notasi ini menyatakan langkah- langkah algoritma dalam bentuk kalimat deskriptif. Masalah penukaran isi gelas A dan B dapat kita selesaikan dengan urutan algoritma sebagai berikut.

- Siapkan 1 Gelas kosong C
- Tuangkan isi gelas A (Kopi) kegelas C
- Tuangkan isi gelas B (Teh) kegelas A
- Tuangkan isi gelas C (Kopi) kegelas B

Notasi II: Flowchart

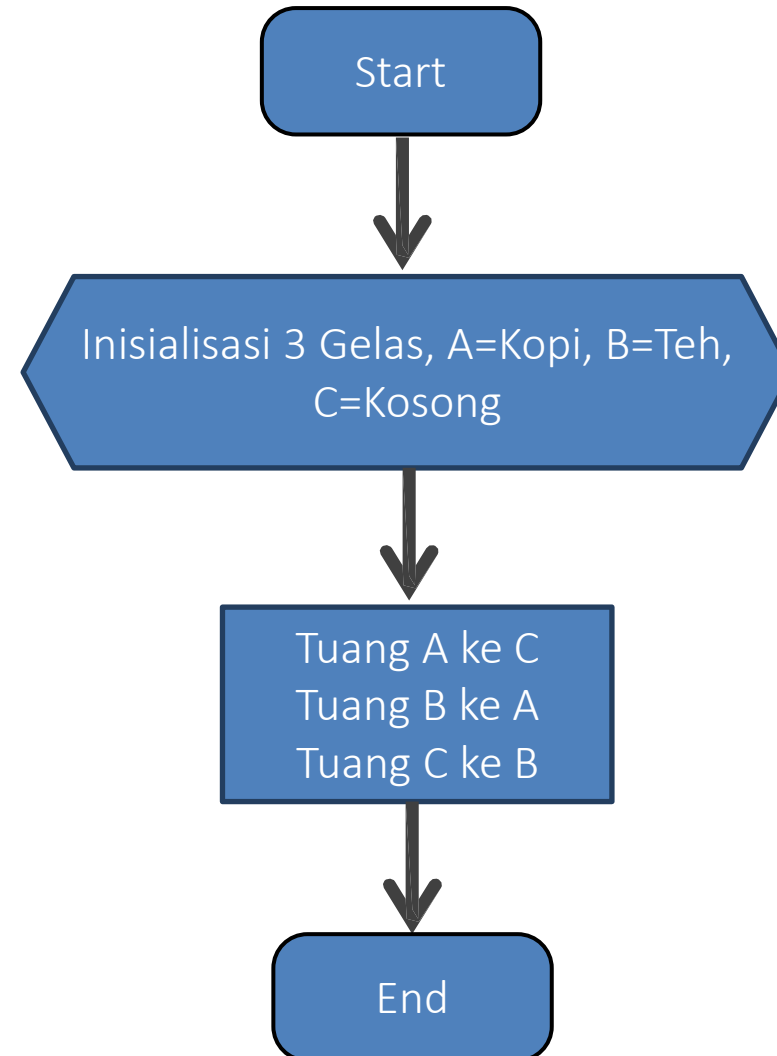
Notasi-notasi pada *flowchart*

-  : (*terminal symbol*), menunjukkan awal dan akhir dari program
-  : (*preparation symbol*), memberikan nilai awal pada suatu variabel atau *counter*
-  : (*processing symbol*), menunjukkan pengolahan aritmatika dan pemindahan data
-  : (*input/output symbol*), menunjukkan proses input atau output
-  : (*decision symbol*), mewakili operasi perbandingan logika
-  : (*predefined process symbol*), proses yang ditulis sebagai subprogram, yaitu prosedur/fungsi
-  : (*connector symbol*), penghubung pada halaman yang sama
-  : (*off page connector symbol*), penghubung pada halaman yang berbeda
-  : arah proses

Notasi II: Flowchart

Merupakan sebuah bagan alir yang menggambarkan aliran instruksi dari sebuah algoritma dalam bentuk geometri.

Flowchart dari Langkah Penyelesaian Masalah



Notasi III: Pseudo-code

Pseudo artinya adalah pura-pura menyerupai. Dengan demikian pseudo-code adalah sebuah kode/tanda/ceritera yang menyerupai penjelasan cara untuk menyelesaikan masalah, ditulis dalam bahasa yang mendekati Bahasa pemrograman tingkat tinggi atau merupakan campuran bahasa manusia dengan sebuah Bahasa pemrograman.

Pseudo code dari ilustrasi masalah yang disampaikan sebelumnya:

```
Deklarasi :  
Gelas_A, Gelas_B, Gelas_C : string  
  
Algoritma:  
Gelas_A="Kopi"  
Gelas_B="Teh"  
Gelas_C="Kosong"  
Gelas_C = Gelas_A  
Gelas_A=Gelas_B  
Gelas_B=Gelas_C
```

Algoritma Euclidean dalam 3 Notasi

Notasi I: menggunakan kalimat Deskriptif

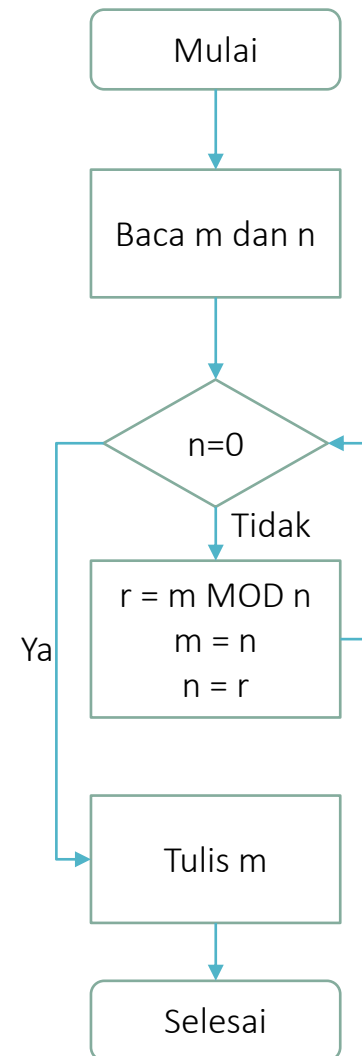
*{Diberikan dua buah bilangan bulat tak-negatif m dan n ($m \geq n$). **Algoritma Euclidean mencari pembagi bersama terbesar**, gcd (greatest common divisor), dari kedua bilangan tersebut, yaitu bilangan bulat positif terbesar yang habis membagi m dan n }*

ALGORITMA:

1. Jika $n=0$, maka m adalah jawabannya; stop.
tetapi jika $n \neq 0$, lanjutkan ke langkah 2.
2. Bagilah m dengan n dan misalkan r adalah sisanya.
3. Ganti nilai m dengan nilai n , dan ganti nilai n dengan r , lalu ulangi kembali ke langkah 1.

Algoritma Euclidean dalam 3 Notasi

Notasi II: menggunakan flowchart



Algoritma Euclidean dalam 3 Notasi

Notasi III: menggunakan pseudo-code

PROGRAM Euclidean

{Diberikan dua buah bilangan bulat tak-negatif m dan n ($m \geq n$). Program ini mencari pembagi bersama terbesar, gcd (greatest common divisor), dari kedua bilangan tersebut, yaitu bilangan bulat positif terbesar yang habis membagi m dan n }

DEKLARASI:

m, n : **integer** {dua bilangan bulat yang akan dicari gcd nya}
 r : **integer** {sisa hasil pembagian}

ALGORITMA:

```
read( $m, n$ )      {asumsi:  $m \geq n$ }  
while  $n \neq 0$  do  
     $r \leftarrow m \bmod n$       {bagi  $m$  dengan  $n$  dan simpan sisanya dalam  $r$ }  
     $m \leftarrow n$       {ganti nilai  $m$  dengan  $n$ }  
     $n \leftarrow r$       {ganti nilai  $n$  dengan  $r$ }  
end while  
{kondisi akhir pengulangan:  $n = 0$ }  
  
write( $m$ )
```

Struktur Teks Algoritma

Notasi algoritma dalam kuliah ini menggunakan notasi Pseudo-Code, dengan struktur sebagai berikut:

PROGRAM Nama Program

{Penjelasan mengenai Initial State, Final State, dan Proses yang dilakukan untuk menyelesaikan persoalan, termasuk apa masukan dan keluarannya. Initial State (I.S.) adalah kondisi sebelum algoritma dijalankan. Final State (F.S.) adalah kondisi yang diharapkan setelah algoritma selesai dijalankan}

DEKLARASI:

{Bagian ini berisi deklarasi semua nama tipe, nama konstanta, nama peubah (variabel), nama prosedur, dan nama fungsi}

ALGORITMA:

{Bagian ini adalah inti dari algoritma yang berisi langkah-langkah penyelesaian persoalan. Semua teks yang tidak dituliskan di antara tanda kurung kurawal buka dan kurung kurawal tutup harus dianggap sebagai notasi algoritma}

Konstruksi Dasar Algoritma

Instruksi & Aksi

Tiga Konstruksi Dasar

Tiga Konstruksi Dasar



Runtunan (sequence)



Pemilihan (selection)



Pengulangan (repetition)

Struktur Runtunan

Terdiri atas satu atau lebih instruksi

Instruksi dikerjakan secara berurutan (sekuensial)

Urutan pengerjaan instruksi menggambarkan logika berpikir.

Urutan pengerjaan yang salah menunjukkan logika berpikir yang salah sehingga hasil akhir algoritma pun salah.

Jika urutan pengerjaan diubah, hasil akhir dari algoritma mungkin berubah



Runtunan (sequence)

Struktur Runtunan

Tinjau algoritma berikut:

ALGORITMA Menghitung harga barang setelah diskon

1. Input *harga barang*.
2. Input p . { p adalah persentase diskon}
3. Hitung *potongan harga* = *harga barang* * p
4. Hitung *harga barang setelah diskon* = *harga barang* – *potongan harga*
5. Tulis *harga barang setelah diskon*.



Runtunan (sequence)

Bagaimana jika instruksi ke-2 dan ke-3 bertukar posisi??

Struktur Pemilihan

Instruksi dalam algoritma dapat berisi **aksi** yang hanya bisa dijalankan jika **memenuhi kondisi** tertentu.

Bentuk instruksi **semacam** ini dinamakan **struktur pemilihan**.

```
if kondisi then  
  aksi
```

kondisi adalah persyaratan yang bernilai benar atau salah
Aksi hanya dikerjakan apabila *kondisi* bernilai benar

Struktur Pemilihan

Struktur pemilihan dapat pula berbentuk lebih umum agar memungkinkan pelaksanaan aksi alternatif apabila kondisi bernilai salah.

```
if kondisi then  
    aksi 1  
else  
    aksi 2
```

else artinya “jika tidak”.

Jika *kondisi* terpenuhi, maka *aksi 1* yang dijalankan, tetapi *jika tidak* terpenuhi, maka *aksi 2* yang dijalankan

Struktur Pemilihan

ALGORITMA Sikap yang Dilakukan Bila Menemui Lampu Lalu Lintas

if lampu traffic light berwarna merah **then**

Hentikan kendaraan Anda

else

if lampu traffic light berwarna kuning **then**

jalan dengan hati-hati

else

jalan terus

ALGORITMA Sikap yang Dilakukan Bila Menemui Lampu Lalu Lintas

if lampu traffic light berwarna merah **then**

Hentikan kendaraan Anda **else if** lampu traffic light berwarna kuning **then**

jalan dengan hati-hati **else** jalan terus

Struktur Pemilihan

Dua buah algoritma ini tidak sepenuhnya sama, tetapi keduanya menjelaskan sikap yang harus dilakukan bila menemui lampu lalu lintas di jalan raya

(1) **if** traffic light menyala **then**

if lampu merah **then**

 berhenti

else

 jalan

(2) **if** traffic light menyala **then**

if lampu merah **then**

 berhenti

else

 jalan

Pada keadaan apa kedua algoritma di atas menggambarkan perilaku yang berbeda?

*Yakni pada keadaan **traffic light padam** dan **traffic light menyala selain warna merah***

Mengapa demikian?

Algoritma mana yang lebih memuaskan?

Struktur Pengulangan

Struktur pengulangan dibutuhkan untuk memudahkan penulisan instruksi yang mesti dilakukan berulang kali, secara **efisien dan efektif**

Struktur pengulangan dituliskan dalam bentuk notasi pengulangan. Beberapa notasi pengulangan yang dapat digunakan:

repeat N times

for

repeat-until

while

repeat N times dan *for*

Digunakan untuk melakukan perulangan aksi yang sama dengan syarat bahwa banyaknya pengulangan sudah **diketahui**.

Notasi *repeat N times* mirip fungsinya dengan notasi *for*

ALGORITMA Menulis kalimat 1000 kali:

1. Tulis “saya berjanji tidak akan terlambat lagi masuk ke kelas Algoritma & Struktur Data”
2. Tulis “saya berjanji tidak akan terlambat lagi masuk ke kelas Algoritma & Struktur Data”
- .
- .
999. Tulis “saya berjanji tidak akan terlambat lagi masuk ke kelas Algoritma & Struktur Data”
1000. Tulis “saya berjanji tidak akan terlambat lagi masuk ke kelas Algoritma & Struktur Data”

*Apakah algoritma tersebut efektif?
Apakah algoritma tersebut efisien?*

repeat N times dan for

ALGORITMA Menulis kalimat 1000 kali:

repeat 1000 times

Tulis "saya berjanji tidak akan terlambat lagi masuk ke kelas Algoritma & Struktur Data"

ALGORITMA Menulis kalimat 1000 kali:

for *i* dari 1 sampai 1000 **do**

Tulis "saya berjanji tidak akan terlambat lagi masuk ke kelas Algoritma & Struktur Data"

Selingan...

NIM	Nama	Alamat	Telepon
13501001	Darmawan	Jl. Taman Poris No 1A	250023040 ← Row ke 1
13500030	Tizar	Jl. Sumur Bandung No 20	259023040 ← Row ke 2
...
13502098	Yusi	Jl. Sangkuriang No. 45	250234000 ← Row ke n

Bagaimana algoritma untuk mencari informasi alamat dan nomor telepon Mahasiswa dengan $NIM = x$?

Gunakan konstruksi dasar yang sudah Anda pelajari sampai detik ini

repeat-until

Repeat-until → ulangi-sampai

Digunakan untuk melakukan proses perulangan **sampai** tercapai kondisi tertentu. Jika kondisi yang ditetapkan telah terpenuhi, perulangan dihentikan.

Struktur pengulangan repeat-until dapat digunakan meski **jumlah pengulangan tidak diketahui** sebelum pengulangan dilakukan.

Instruksi untuk melakukan aksi pada badan pengulangan dilakukan **minimal sekali**

Bentuk umumnya:

repeat

Aksi

until *kondisi*

repeat-until

NIM	Nama	Alamat	Telepon	
13501001	Darmawan	Jl. Taman Poris No 1A	250023040	← Row ke 1
13500030	Tizar	Jl. Sumur Bandung No 20	259023040	← Row ke 2
...	
13502098	Yusi	Jl. Sangkuriang No. 45	250234000	← Row ke n

ALGORITMA Mencari Data Alamat dan Telepon Mahasiswa Ber-NIM x

1. Tinjau row pertama dari tabel

2. **repeat**

if NIM pada row table = x **then**

 tulis Alamat dan Telepon dari row tersebut

else

 tinjau row berikutnya di dalam table

until NIM yang dicari sudah ditemukan atau akhir table sudah terdeteksi

while

while → selagi atau selama

Digunakan untuk melakukan proses perulangan **selama** kondisi tertentu masih terpenuhi. Jika kondisi yang ditetapkan **tidak** terpenuhi, maka perulangan tidak dilakukan/dihentikan.

Struktur pengulangan *while* dapat digunakan meski **jumlah pengulangan tidak diketahui** sebelum pengulangan dilakukan.

Instruksi untuk melakukan aksi pada badan pengulangan **mungkin saja tidak dilakukan sama sekali**.

Bentuk umumnya:

While kondisi do
Aksi

while

NIM	Nama	Alamat	Telepon	
13501001	Darmawan	Jl. Taman Poris No 1A	250023040	← Row ke 1
13500030	Tizar	Jl. Sumur Bandung No 20	259023040	← Row ke 2
...	
13502098	Yusi	Jl. Sangkuriang No. 45	250234000	← Row ke n

ALGORITMA Mencari Data Alamat dan Telepon Mahasiswa Ber-NIM x

1. Tinjau row pertama dari tabel
2. **while** NIM yang dicari belum ditemukan dan belum akhir tabel **do**
 - if** NIM pada row table = x **then**
 - tulis Alamat dan Telepon dari row tersebut
 - else**
 - tinjau row berikutnya di dalam table

while

Tinjau algoritma berikut:

ALGORITMA Menghitung GCD dari dua bilangan, m dan n .

1. Input m .
2. Input n . {asumsi $n \leq m$ }
3. **while** $n \neq 0$ **do**
4. bagilah m dengan n , sisanya dimasukkan ke r
5. ganti nilai m dengan n
6. ganti nilai n dengan r
7. **end while**
8. tulis m

Bagaimana jika instruksi ke-5 dan ke-6 bertukar posisi??

Ringkasan perbandingan antara berbagai notasi perulangan

Kriteria	repeat N times for i dari 1 sampai N	while <i>kondisi</i> do <i>aksi</i>	repeat <i>aksi</i> until <i>kondisi</i>
Jumlah maksimal perulangan	N kali	Tidak perlu diketahui sebelum struktur perulangan ini dijalankan	Tidak perlu diketahui sebelum struktur perulangan dijalankan
Eksekusi aksi pada badan/ <i>body</i> perulangan	<i>Pasti</i> dilakukan sebanyak N kali	<i>Mungkin saja</i> tidak dieksekusi sama sekali	<i>Pasti</i> minimal sekali dieksekusi
Kapan perulangan berhenti?	Setelah N kali	Pada saat <i>kondisi</i> bernilai salah	Pada saat <i>kondisi</i> bernilai benar

Tugas Mandiri

Tiga buah cakram yang masing-masing berdiameter berbeda mempunyai lubang di titik pusatnya. Ketiga cakram tsb dimasukkan pada sebuah batang besi A sedemikian sehingga cakram yang berdiameter lebih besar selalu terletak di bawah cakram yang berdiameter lebih kecil.

*Tulis algoritma dalam **notasi pseudo-code** untuk memindahkan seluruh cakram tsb ke batang besi B; setiap kali hanya satu cakram yang boleh dipindahkan, tetapi pada setiap perpindahan tidak boleh ada cakram yang lebih besar berada di atas cakram kecil.*

Batang besi C dapat dipakai sebagai tempat peralihan dengan tetap memegang aturan yang telah disebutkan.

Ilustrasi cakram dan batang A, B, dan C akan digambarkan pada slide materi

Jazakumullah khairan
katsiiraa...

Sekian dan Terimakasih