

Mata Kuliah	:	Dasar Pemrograman
Bobot Sks	:	2
Dosen Pengembang	:	Riad Sahara, S.Si, M.T Syahid Abdullah, S.Si, M.Kom
Tutor	:	Syahid Abdullah, S.Si, M.Kom
Capaian Pembelajaran Mata Kuliah	:	1. Mahasiswa mampu memahami Konsep Algoritma dan Pemrograman
Kompetensi Akhir di Setiap Tahap (Sub- Cpmk)		1. Mahasiswa memahami terminologi algoritma 2. Mahasiswa memahami struktur dasar algoritma 3. Mahasiswa memahami bentuk penulisan algoritma dan menerapkannya pada suatu masalah
Minggu Perkuliahan Online Ke-		2

JUDUL TOPIK – Algoritma

Algoritma

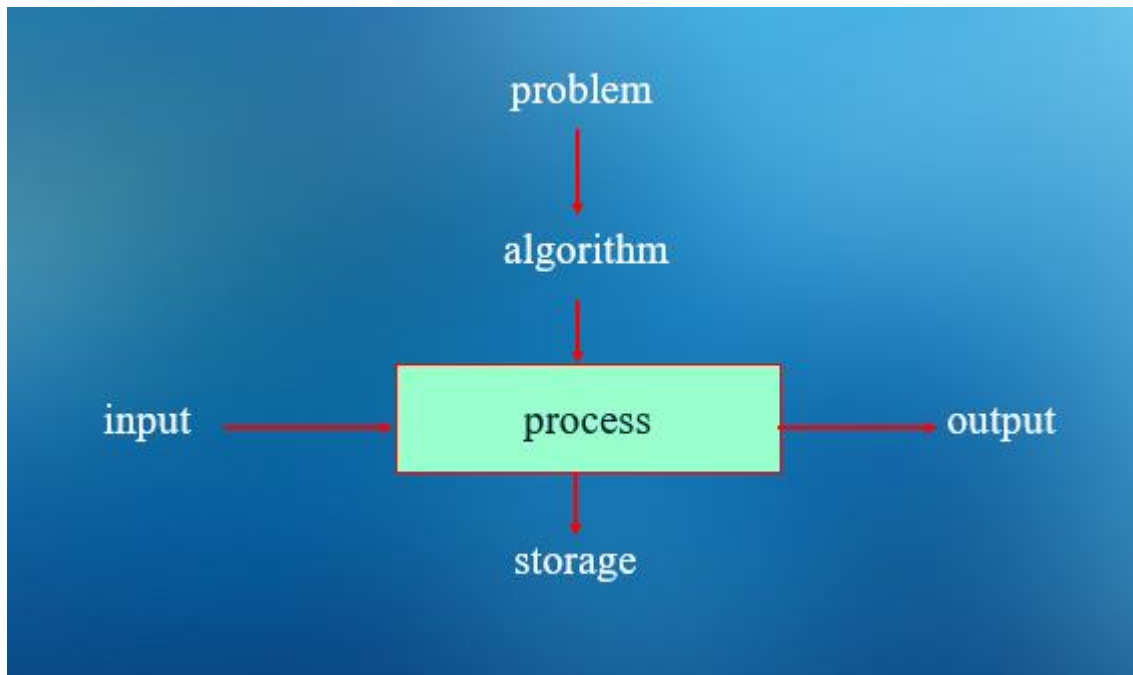
Definisi

- Algoritma merupakan urutan langkah demi langkah yang terbatas, diskrit, memiliki instruksi yang tidak ambigu untuk memecahkan masalah tertentu
 - Memiliki data input dan diharapkan untuk menghasilkan data output
 - Setiap instruksi dapat dilakukan dalam jumlah waktu yang terbatas dengan cara deterministik
- Secara sederhana, algoritma merupakan serangkaian instruksi untuk memecahkan suatu masalah (menyelesaikan pekerjaan)
- Masalah bisa dalam bentuk apapun:
 - Bisnis

- Dapatkan part dari Vancouver ke Ottawa besok pagi
- Alokasikan tenaga kerja untuk memaksimalkan keuntungan
- Hidup
 - Saya lapar. Bagaimana cara memesan pizza?
 - Jelaskan bagaimana cara mengikat tali sepatu pada anak usia 5 tahun
- Kita berurusan dengan masalah pengolahan data diterjemahkan ke program yang dapat dijalankan pada komputer
- Karena kita hanya bisa input, simpan, proses & output data pada komputer, instruksi dalam algoritma kita akan terbatas pada keempat fungsi ini

Representasi Algoritmik

- Input
 - Get information *Get (input command)*
- Storage
 - Store information *Given/Result*
Intermediates/Set
- Process
 - Arithmetic *Let (assignment command)*
 - Repeat instructions *Loop*
 - Branch conditionals *If*
- Output
 - Give information *Give (output command)*



Deskripsi Algoritma

- Pahami masalah sebelum memecahkannya
 - Identifikasi & beri nama setiap Input (**Given**)
 - Identifikasi & beri nama setiap Output (**Result**)
 - Tetapkan nama Algoritma (**Name**)
 - Gabungkan ketiga bagian informasi menjadi sebuah pernyataan resmi (**Definition**):
 - **Result := Name(Given)**

Method

- Setelah deskripsi algoritma telah disiapkan, berikutnya adalah memecahkan masalahnya
- Kita kembangkan serangkaian instruksi (terbatas) yang ketika dijalankan akan melakukan komputasi terhadap **Result** yang diinginkan dari **Given** (**Method**)

Assignment Command

- **Syntax**
 $X = 5Y + 16$

Sebelah kiri tanda “=” menunjukkan nama variabel, dan sebelah kanan merupakan nilai atau ekspresi.

Setiap variabel mengacu pada lokasi yang unik dalam memori komputer yang berisi nilai.

- **Interpretasi**

Sebuah assignment dijalankan dalam dua langkah:

1-evaluasi ekspresi ditemukan di sisi kanan.

2-pengaturan nilai yang dikembalikan dalam sel memori yang sesuai dengan variabel.

- **Contoh**

- `Let SideSize=15`
- `Let Area=SideSize*SideSize`

Assignment: Mathematics VS Computer Science

- Instruksi berikut adalah sama pada matematika

$$A = B \qquad B = A$$

Tidak dalam computer science

Let A = B tidak sama dengan **Let B = A**

- Dalam matematika kita mengenal relasi

Sebuah relasi $B = A + 1$ merupakan pernyataan yang selalu benar setiap saat

Pada computer science, kita menggunakan assignment. Kita dapat memiliki:

$$A = 5$$

$$B = A + 1$$

$$A = 2$$

Relasi $B = A + 1$ bernilai benar setelah instruksi kedua dan sebelum instruksi ketiga

- Instruksi $A = A + 3$ bernilai salah pada matematika.

Pada computer science, **Let A = A + 3** berarti: nilai A yang baru merupakan penjumlahan antara nilai A yang lama dengan 3.

- Instruksi $A + 5 = 3$ diperbolehkan dalam matematika.

Let $A + 5 = 3$ tidak memiliki arti pada computer science karena sebelah kiri “=” haruslah nama variabel

Input Command

- Syntax
Get variable
Variabel diperoleh dari **Given**
- Interpretasi
Di sini, user harus memberikan sebuah nilai. Nilai yang diberikan, ditugaskan pada variabel.
- Contoh
Get Size_Side

Output Command

- Syntax
Give variable
Variabel diperoleh dari **Result**
- Interpretasi
Nilai variabel ditampilkan.
- Contoh
Give Area

Algoritma 1

- Tulis algoritma untuk menemukan jumlah dari tiga bilangan yang diberikan

```
NAME           : SUM3
GIVENS         : N1, N2, N3
RESULTS        : Total
DEFINITION     : Total := SUM3 (N1, N2, N3)
-----
METHOD         :
    Get N1
    Get N2
    Get N3
    Let Total = N1 + N2 + N3
    Give Total
```

Algoritma 2

- Tulis algoritma untuk menemukan hasil dari operasi pembagian jika diberikan dua bilangan X dan Y

```
NAME      : Division
GIVENS    : X, Y
RESULTS   : Quotient
DEFINITION: Quotient := Division(X, Y)
-----
METHOD    :
    Get X
    Get Y

    Let Quotient = X/Y

    Give Quotient
```

Algoritma 3

- Tulis algoritma untuk menemukan hasil penjumlahan dan perkalian dari dua bilangan yang diberikan

```
NAME      : SumTimes
GIVENS    : Num1, Num2
RESULTS   : Total, Product
DEFINITION: Total & Product := SumTimes (Num1, Num2)
-----
METHOD    :
    Get Num1
    Get Num2

    Let Total = Num1 + Num2
    Let Product = Num1 * Num2

    Give Total
    Give Product
```

Algoritma 4

- Tentukan jumlah dan rata-rata dari tiga bilangan yang diberikan

```
NAME      :AVG3
GIVENS    :Num1, Num2, Num3
RESULTS   :Sum , Average
DEFINITION:Sum & Average := AVG3 (Num1, Num2, Num3)
-----
METHOD    :
    Get Num1
    Get Num2
    Get Num3

    Let Sum = Num1 + Num2 + Num3
    Let Average = Sum /3

    Give Sum
    Give Average
```

Intermediate

- Terkadang dalam sebuah algoritma, kita perlu memiliki variabel untuk menyimpan nilai sementara.
- Ini adalah variabel perantara yang pada Deskripsi Algoritma disebut sebagai Intermediate.

Algoritma 5

- Tulis algoritma untuk menukar nilai dari dua bilangan

```
NAME      : Swap
GIVENS    : X, Y
RESULTS   : X, Y
INTERMEDIATES: Temp
DEFINITION: Swap (X, Y)
```

METHOD:

```
  Get X
  Get Y
  Let Temp = X
  Let X = Y
  Let Y = Temp
  Give X
  Give Y
```

Algoritma 6

- Tulis algoritma yang menambahkan dua bilangan yang diberikan (X dan Y) dan mengembalikan jumlah tersebut dalam variabel X


```

NAME      : AddXY
GIVENS    : X, Y
RESULTS   : X
INTERMEDIATES: None
DEFINITION: AddXY (X, Y)
-----
METHOD:
    Get X
    Get Y

    Let X = X+ Y

    Give X

```

Summary

Name	Nama formal sebuah algoritma
Given	Nama dari nilai yang diberikan untuk suatu masalah
Result	Nama dari nilai yang dihasilkan dari suatu masalah
Intermediate	Nama dari variabel perantara yang digunakan dalam algoritma
Definition	Definisi formal dari algoritma, menggunakan format Name , Given , dan Result
Method	Rangkaian langkah demi langkah dari instruksi untuk menyelesaikan masalah

Menelusuri Algoritma

- Tujuan menelusuri algoritma adalah untuk memastikan bahwa ia bekerja
- Ini adalah tes yang harus diselesaikan sebelum menulis kode komputer

- Penelusuran meliputi:
 - Mengeksekusi urutan instruksi dengan sampel serangkaian nilai
 - Menghitung nilai dari setiap variabel setelah setiap instruksi dieksekusi
 - Memeriksa hasil yang benar
- Langkah 1: Beri nomor setiap instruksi dalam algoritma
- Langkah 2: Buat tabel
 - Kolom pertama dari tabel menunjukkan instruksi yang telah dieksekusi
 - Kolom selanjutnya berisi daftar semua variabel algoritma (**Given, Result, Intermediate**)
- Langkah 3: Lengkapi tabel
 - Setiap kolom tabel mewakili variabel
 - Mulai di baris pertama dari algoritma. Identifikasi apa yang akan terjadi pada setiap variabel setelah instruksi dieksekusi
 - Ubah nilai setiap perubahan instruksi dan tinggalkan semua kolom lain kosong

Penelusuran 1

- Telusuri Algoritma 4 menggunakan bilangan 24, 31, dan 35

	METHOD	Line	Num1	Num2	Num3	Sum	Avg
(1)	Get Num1	1	24				
(2)	Get Num2	2		31			
(3)	Get Num3	3			35		
(4)	Let Sum=Num1+Num2+Num3	4				90	
(5)	Let Average=Sum/3	5					30
(6)	Give Sum	6 output 90					
(7)	Give Average	7 output 30					

Penelusuran 2

- Telusuri Algoritma 5 dimana bilangan X dan Y bernilai 25 dan 88

	METHOD	Line	X	Y	Temp
(1)	Get X	1	25		
(2)	Get Y	2		88	
(3)	Let Temp=X	3			25
(4)	Let X=Y	4	88		
(5)	Let Y=Temp	5		25	
(6)	Give X	6 output 88			
(7)	Give Y	7 output 25			

Evaluasi

1. Tulis algoritma penjumlahan 5 bilangan. Telusuri algoritma jika diberikan bilangan 1, 3, 5, 7, dan 9.
2. Tulis algoritma untuk menghitung rata-rata dari 5 bilangan. Telusuri algoritma jika diberikan bilangan 1, 3, 5, 7, dan 9.
3. Tulis algoritma untuk mengalikan 3 bilangan. Telusuri algoritma jika diberikan bilangan 2, 9, dan 6.

Daftar Pustaka

Goodrich, Michael, Roberto Tamassia, and David Mount. *Data structures and algorithms in C++*. John Wiley & Sons, 2011.

Mehlhorn, Kurt, and Peter Sanders. *Algorithms and data structures: The basic toolbox*. Springer, 2010.