

# Dasar Pemrograman

Tim Ajar Dasar Pemrograman 2025

# Pengantar



# Deskripsi MK

## Deskripsi Mata Kuliah Dasar Pemrograman

Dasar Pemrograman memberikan **pengetahuan dan pemahaman** konsep dasar algoritma dan dasar pemrograman sehingga dalam mata kuliah ini mahasiswa **memiliki dasar untuk menyelesaikan permasalahan-permasalahan logika** dengan menggunakan flowchart dan pseudocode.

# Tujuan MK

## Di akhir kuliah mahasiswa mampu:

- Mahasiswa mengenal konsep dasar algoritma serta mampu menganalisis permasalahan sederhana ke dalam bentuk algoritma
- Mahasiswa mampu memahami dan menerapkan konsep-konsep dasar pemrograman seperti tipe data, variabel, dan struktur dasar dalam pemrograman.
- Mahasiswa mampu memahami manajemen kontrol versi serta struktur pemilihan dan perulangan untuk menyelesaikan permasalahan sederhana.
- Mahasiswa mampu memahami pengolahan data dengan array
- Mahasiswa mampu memahami penggunaan fungsi untuk modularitas dalam pemrograman

# Pokok Bahasan

1. Dasar Pemrograman (Konsep Algoritma)
2. Tipe Data, Variabel, Konstanta, Input-Output
3. Operator, Input-Output
4. Git
5. Pemilihan
6. Perulangan
7. Array
8. Fungsi/Prosedur

# Pengantar – Kontrak Kuliah...(1)

Pertemuan	Materi
Minggu – 1	Dasar Pemrograman
Minggu – 2	Operator, Input-Output
Minggu – 3	Tipe data, Variabel, Input – Output, Sequence
<b>Minggu - 4</b>	<b>Kuis 1</b>
Minggu – 5	Git
Minggu – 6	Pemilihan
<b>Minggu – 7</b>	<b>UTS</b>
<b>Minggu – 8</b>	<b>CM</b>
Minggu – 9	Perulangan 1
Minggu – 10	Perulangan 2 (Nested Loop)
Minggu – 11	Array 1
Minggu – 12	Array 2

# Pengantar – Kontrak Kuliah...(2)

Pertemuan	Materi
<b>Minggu – 13</b>	<b>Kuis - 2</b>
Minggu – 14	Fungsi 1
Minggu – 15	Fungsi 2
<b>Minggu – 16</b>	<b>CM</b>
<b>Minggu – 17</b>	<b>UAS</b>

- **Bobot Kuis : 20 %**
- **Bobot Tugas : 20 %**
- **Bobot UTS : 30%**
- **Bobot UAS : 30%**

# **Pertemuan 1**

## **DASAR PEMROGRAMAN**

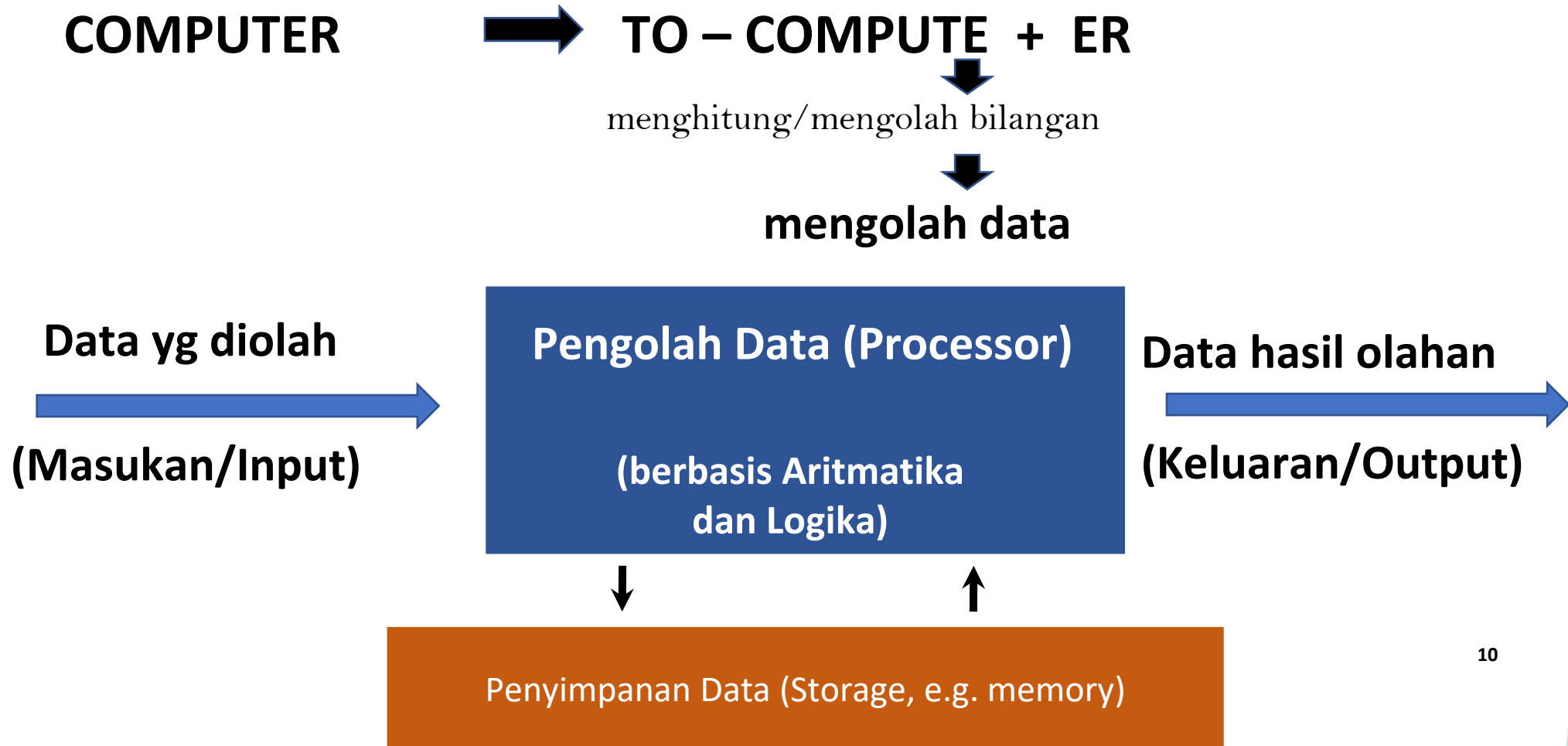


# Tujuan

**Setelah menempuh materi ini, mahasiswa hendaknya mampu:**

Mahasiswa mengenal konsep dasar algoritma serta mampu menganalisis permasalahan sederhana ke dalam bentuk algoritma

# Introduction - What Is a Computer?



# Program Komputer

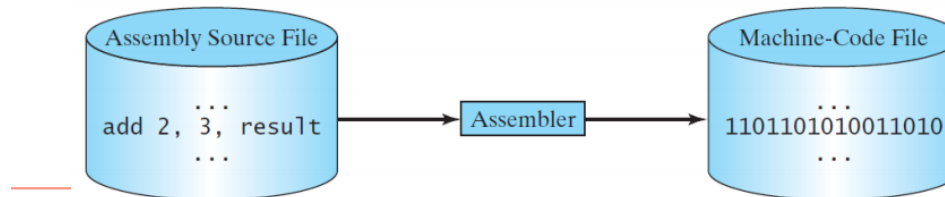
- Program Komputer atau yang sering disebut perangkat lunak (software) merupakan instruksi untuk computer.
- Anda dapat memberikan perintah kepada Komputer melalui program.
- Tanpa adanya program, Komputer adalah mesin kosong yang tidak bisa diberikan intruksi.
- Karena Komputer tidak mengerti Bahasa manusia maka cara berkomunikasi dengan computer adalah dengan menggunakan *computer languages*
- **Program ditulis dengan menggunakan bahasa pemrograman.**

# Computer Languages

Machine Language - Assembly Language - High Level Language

- Machine Language : program ditulis dalam Bahasa biner.
  - Misalnya, untuk menjumlahkan dua angka, instruksi dalam biner seperti ini:
  - 1101101010011010
- Assembly Language : lebih mudah dibandingkan Bahasa mesin.
  - Komputer tidak mengerti Bahasa assembly sehingga terdapat assembler yang menterjemahkan Bahasa assembly ke Bahasa mesin
  - Contoh :

ADDF3 R1, R2, R3





# Computer Languages

Machine Language - Assembly Language - High Level Language

- High Level Language : mirip dengan Bahasa inggris dan mudah dipelajari oleh programmer
  - Contoh : java, C++, C, C#, Python, ds

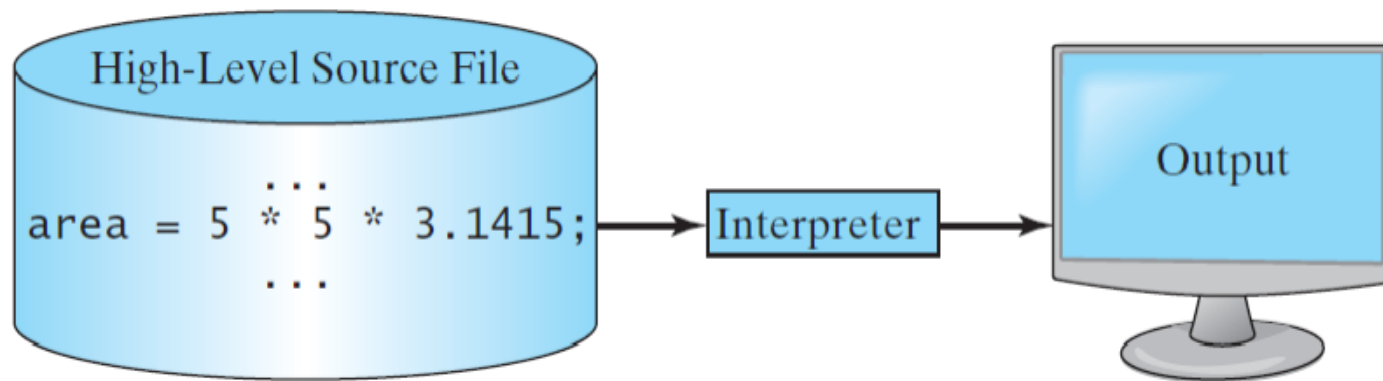
```
1 public class MyFirstJava{  
2  
3     public static void main(String[] args){  
4  
5         System.out.println("Hello Word");  
6  
7     }  
8 }
```

# Interpreting/Compiling Source Code

- Program yang ditulis dalam Bahasa tingkat tinggi (high-level language) disebut *source code* atau *source program*
- Karena program tidak mengerti Bahasa *source code* maka *source code* harus diterjemahkan ke dalam Bahasa mesin untuk bisa di eksekusi oleh computer
- Translasi / penterjemahan source code ke Bahasa mesin dapat dilakukan menggunakan tools pemrograman lain yang disebut interpreter atau compailer.

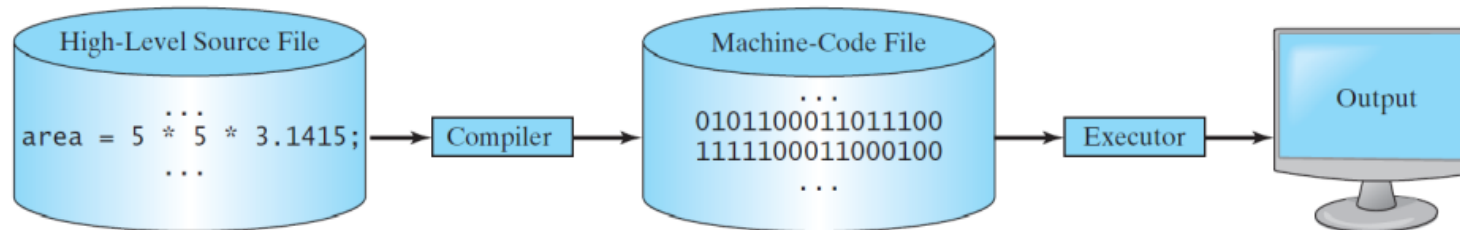
# Interpreting Source Code

- Interpreter membaca satu pernyataan (statement) dari source code lalu menterjemahkannya ke dalam Bahasa mesin dan kemudian langsung mengeksekusinya.
- cara kerja interpreter:



# Compiling Source Code

- Compiler menterjemahkan seluruh source code menjadi sebuah file yang berisi Bahasa mesin dan kemudian file yang berisi Bahasa mesin tersebut yang akan di eksekusi.
- cara kerja compiler:





# Bahasa Pemrograman Java

- Java sudah ada sejak **1995**, masih dipakai hingga saat ini di industri besar.
- Java Bahasa pemrograman yang **ketat dengan aturan** (class, main, dan tipe data harus jelas).
  - **Struktur program terorganisir**
  - Jika sudah bisa Java, pindah ke Bahasa pemrograman lain akan lebih gampang.
- **Contoh Aplikasi menggunakan Java**
  - **Android Apps** → Awalnya **semua aplikasi Android ditulis dengan Java** sebelum Kotlin populer, Contoh: WhatsApp, Spotify (awal pengembangan).
  - **Backend Web** → Spring Framework banyak dipakai e-commerce.
  - **Bank** → Hadoop (framework big data) ditulis dalam Java.
  - **Big Data** → Hadoop, Apache Spark berjalan di JVM.
  - **IoT** → Banyak perangkat IoT (smart card, smart meter) menggunakan Java.

# Algoritma

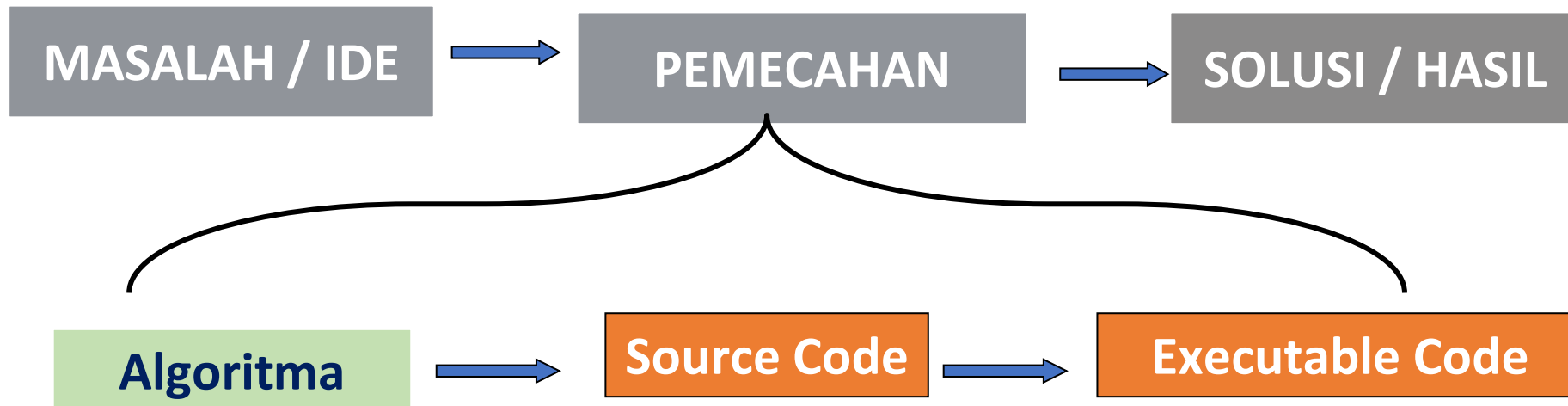
# Algoritma...(1)

- Proses pengolahan data membutuhkan **algoritma**
- Algoritma : urutan langkah-langkah logis penyelesaian masalah yang disusun secara **sistematis** dan **logis**.
- Sistematis : urutan langkah-langkah yang disusun harus jelas, memiliki pola tertentu
- Logis : masuk akal, bisa dibaca dan diterima oleh akal
- Bisa dinilai benar salahnya

# Algoritma...(2)

- **Pemrograman komputer**

- Algoritma diimplementasikan dalam **program komputer**
- Satu set instruksi atau langkah-langkah yang dijalankan dengan komputer untuk menyelesaikan suatu masalah.





# Penilaian Algoritma

- Hasilnya harus benar, sesuai dengan yang diinginkan
- Seberapa baik hasil dari algoritma tersebut
- Efisiensi algoritma → waktu dan memori

Dimungkinkan antara orang satu dengan yang lain, dari **masalah yang sama**, akan mendapatkan **algoritma yang berbeda**. Semuanya dianggap benar, asal hasil yang dikeluarkan sesuai dengan harapan. Namun demikian, diantara perbedaan algoritma tersebut, bisa dipilih mana yang lebih **efisien**

# Struktur Dasar Algoritma

## 1. Struktur runtutan/sekuensial :

Digunakan untuk program yang pernyataannya sequential atau urutan.

## 2. Struktur pemilihan

Digunakan untuk program yang menggunakan pemilihan atau penyeleksian kondisi.

## 3. Struktur perulangan

Digunakan untuk program yang pernyataannya akan dieksekusi berulang-ulang.

# Kriteria Algoritma

- Input
  - Terdapat inputan bernilai nol ataupun lebih banyak sebagai sumber yang berasal dari luar program.
- Output
  - Output minimal terdiri dari satu kuantitas hasil.
- *Definiteness*
  - Setiap instruksi yang diberikan harus jelas dan tidak ambigu.
- *Finiteness*
  - Jika kumpulan instruksi algoritma ditelusuri, maka tahapan algoritma akan berakhir setelah sejumlah langkah yang terbatas.
- *Effectiveness*
  - Setiap instruksi harus cukup mendasar sehingga mudah untuk dilakukan sesuai dengan kebutuhan

# Contoh Algoritma

Masalah : Bagaimana cara membuat segelas teh

- Input : Teh, Air, gula
- Output : Segelas teh
- Data lain : gelas, teko listrik
- Proses
  - Tuang air ke teko Listrik
  - Panaskan air menggunakan teko listrik
  - Masukkan teh ke gelas
  - Tuang air panas ke gelas
  - Tambah gula ke gelas
  - Aduk



# Computational Thinking

# Computational Thinking ???

- Computational Thinking (CT) adalah sebuah cara berfikir untuk memecahkan persoalan, merancang sistem, memahami perilaku manusia.
- CT merupakan dasar konsep informatika.
- CT berarti berfikir untuk menciptakan dan menggunakan beberapa tingkatan abstraksi, mulai memahami persoalan sehingga mengusulkan pemecahan solusi yang efektif dan efisien.



# Ada empat teknik kunci dalam Computational Thinking

## Dekomposisi

memecah masalah yang lebih besar/kompleks menjadi bagian-bagian yang lebih kecil/ sederhana.

## Abstraksi

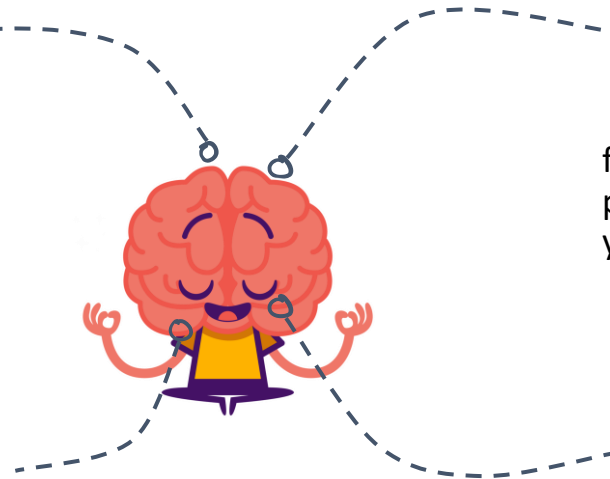
fokus pada informasi penting saja, dan mengabaikan detail yang kurang relevan.

## Pattern/Pola


mencari/menemukan pola/kesamaan antar masalah maupun dalam masalah tersebut.

## Algoritma

Membangun langkah-langkah solusi dari masalah



# Contoh soal CT

- **Berang-berang menggunakan bohlam biner untuk mengirim pesan.**  
 Untuk mengirim pesan, mereka menggunakan tabel dan aturan berikut.  
 Berang-berang mengirim pesan sebagai berikut:
-  **Tabel Kode (A–Z):**  
 (Setiap huruf direpresentasikan oleh kode biner 8-bit, seperti dalam tabel di sebelah kiri.)

A	01000001	J	01001010	S	01010011
B	01000010	K	01001011	T	01010100
C	01000011	L	01001100	U	01010101
D	01000100	M	01001101	V	01010110
E	01000101	N	01001110	W	01010111
F	01000110	O	01001111	X	01011000
G	01000111	P	01010000	Y	01011001
H	01001000	Q	01010001	Z	01011010
I	01001001	R	01010010		



Rules:

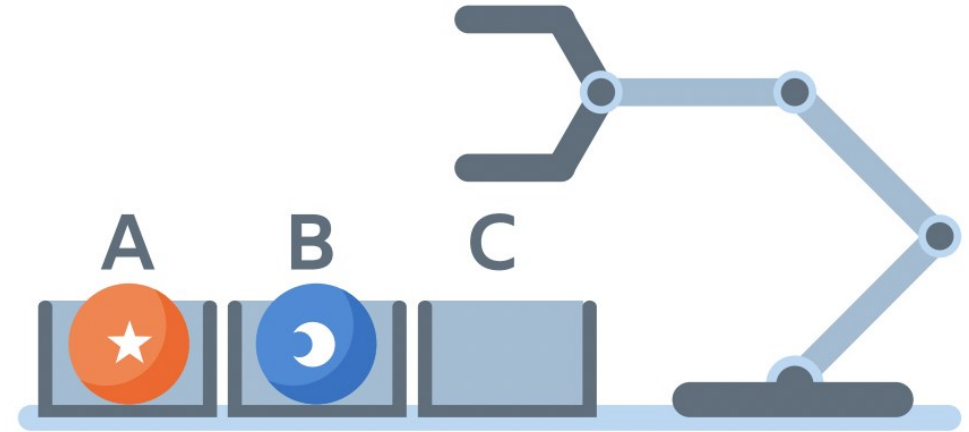


## Question

Which message did they send?

# Contoh soal CT- Runtutan

- Tim Berang-berang membangun prototipe lengan robot. Prototipe tersebut akan diuji dengan cara berikut: Mereka menempatkan lengan tersebut di atas meja dengan dua bola: satu di dalam nampan A, dan satu lagi di dalam nampan B. Nampan C kosong. Lengan robot mengikuti Algoritma dengan urutan langkah-langkah berikut:
  1. Mengambil bola di dalam nampan A dan meletakkannya ke dalam nampan C.
  2. Mengambil bola di dalam nampan B dan meletakkannya kedalam nampan A.
  3. Mengambil bola di dalam nampan C dan meletakkannya kedalam nampan B.



Ketika lengan robot selesai, manakah pernyataan berikut yang benar?

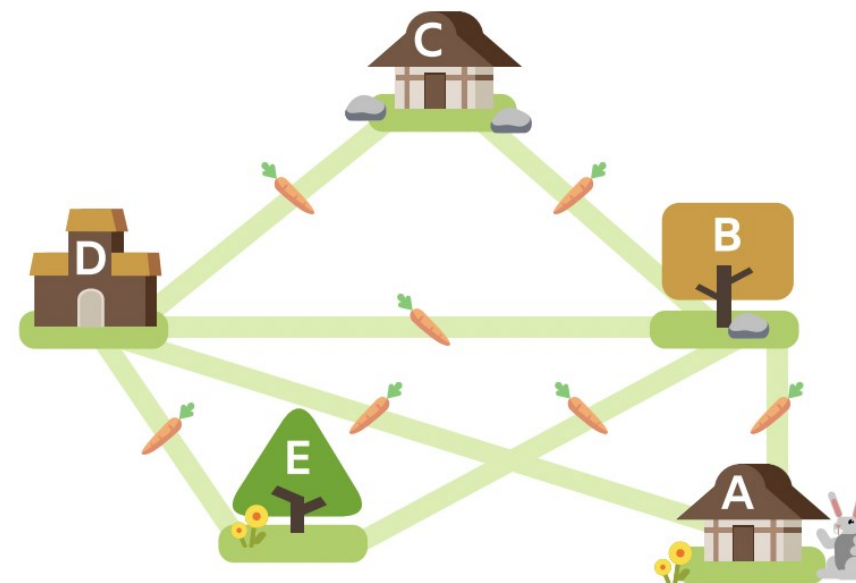
A. Ada dua bola di nampan A	B. Tidak ada yang berubah. Setiap bola kembali ke tempat awal
C. Nampan A kosong	D. Nampan C kosong

# Contoh soal CT - Pilihan

Rumah kelinci kecil ada di **A**.

Dimulai dari rumahnya, dia akan mengambil wortel. Wortel-wortel tersebut ditunjukkan dalam gambar di samping.

Kelinci kecil ingin mengambil semua wortel dengan kecepatan tercepat dan kembali ke rumahnya. Bisakah Anda membantu kelinci kecil menemukan rute optimal?



Manakah Rute berikut ini yang paling optimal?

- A. ABCDEBDA
- B. ADEBADCBDA
- C. ABDABCDEBA
- D. ADEBCDABA

# Contoh soal CT - Perulangan

- Havel memiliki sebuah **mesin jahit yang bisa diprogram**.  
Mesin ini bisa menjahit dengan jahitan “+” atau jahitan “x”.  
Mesin juga bisa **memindahkan kain maju 1 langkah**.
- Mesin bisa menjahit kedua jahitan pada tempat yang sama (urutan tidak masalah). Jika keduanya ditumpuk, hasilnya menjadi “\*”.
- Program mesin terdiri dari urutan karakter: +, x, dan →.
- Saat mesin dijalankan, program akan **berulang terus** selama pedal ditekan.

- Jika program yang dijalankan:

+ → + x → x →

- Maka hasil jahitannya adalah:

+\*x+\*x+\*x+\*x+\*x+\*x+\*x+

- Jika havel ingin menghasilkan pola jahitan berikut ini :

x\*x\*x\*x\*x\*x\*x\*x\*x\*x\*x\*x\*x\*x\*x

- Maka Program seperti apa yang harus dijalankan havel?

# Algoritma - Runtutan

Bu Siti memiliki sebuah kolam ikan berbentuk **persegi panjang** dengan ukuran panjang 20 m dan lebar 10 m. Di dalam kolam tersebut, terdapat sebuah **pulau kecil berbentuk persegi** dengan sisi 4 m yang digunakan untuk menanam pohon.

Berapakah luas kolam yang benar-benar terisi air? Buatlah algoritmanya.

Jawab :

- **Input** : Panjang kolam = 20 m, lebar kolam = 10 m, sisi pulau = 4 m
- **Output** : Luas kolam yang berisi air
- **Proses** :
  - Input Panjang kolam, lebar kolam, sisi pulau
  - Hitung Luas kolam (persegi panjang)  
 $= \text{panjang} \times \text{lebar}$   
 $= 20 \times 10 = 200 \text{ m}^2$
  - Hitung Luas pulau (persegi)  
 $= \text{sisi} \times \text{sisi}$   
 $= 4 \times 4 = 16 \text{ m}^2$
  - Hitung Luas kolam berisi air = luas kolam – luas pulau  
 $= 200 - 16 = 184 \text{ m}^2$
- **Hasil**: Luas kolam berisi air = **184 m<sup>2</sup>**



# Algoritma - Pemilihan

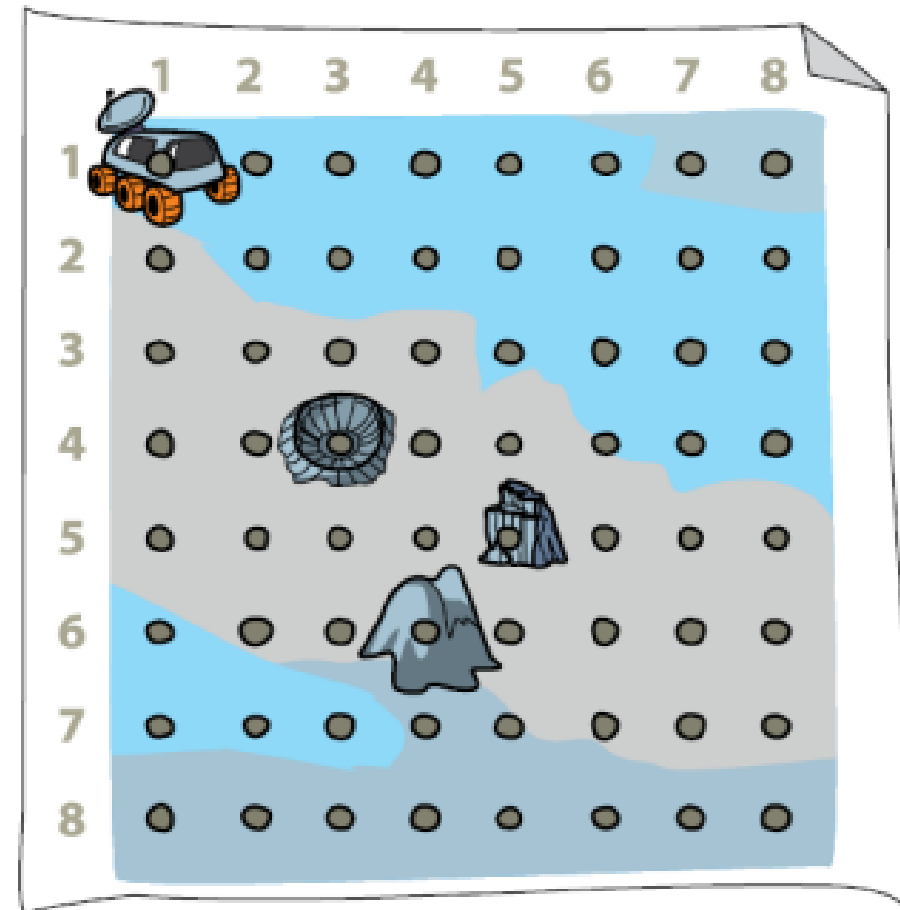
Tarif parkir motor Rp2.000 untuk 2 jam pertama. Setelah itu, setiap jam terdapat biaya tambahan sebesar Rp1.000. Buatlah algoritma untuk menghitung biaya parkir.

Jawab :

- Input : lama parkir (jam)
- Proses :
  - Masukkan lama parkir
  - Jika  $\text{lama} \leq 2$ , maka  $\text{biaya} = 2000$
  - Jika  $\text{lama} > 2$ , maka  $\text{biaya} = 2000 + (\text{lama} - 2) \times 1000$
  - Output biaya parkir
- Output : biaya parkir
- Data Lain : -

# Algoritma - Perulangan

- Penjelajah bulan yang dikendalikan berang-berang bergerak dari satu arah ke yang lain menggunakan peta di bawah ini. mulai (1,1) sampai (8,8). Angka pertama adalah baris, yang kedua - adalah kolom jumlah. Dimanakah letak objek kawah, batu, dan gunung? Buatlah Algoritmanya :



# Contoh Studi Kasus Perulangan

- Input : Peta (baris, kolom), jumlah baris, jumlah kolom
- Output : Posisi Gunung
- Proses :
  1. Lakukan pengecekan secara berulang-ulang mulai dari baris pertama untuk kolom ke-1 sampai dengan ke-8
    - Jika menemukan objek simpan informasi posisi baris dan kolom
  2. Lakukan pengecekan secara berulang-ulang mulai dari baris kedua untuk kolom ke-1 sampai dengan ke-8
    - Jika menemukan objek simpan informasi posisi baris dan kolom
  3. Lakukan pengecekan secara berulang-ulang mulai dari baris ketiga untuk kolom ke-1 sampai dengan ke-8
    - Jika menemukan objek simpan informasi posisi baris dan kolom
  4. Lakukan pengecekan secara berulang-ulang mulai dari baris keempat untuk kolom ke-1 sampai dengan ke-8
    - Jika menemukan objek simpan informasi posisi baris dan kolom
- Lakukan perulangan sampai baris ke 8

# Ada Pertanyaan



# ???

# Latihan

# Latihan 1

- Berdasarkan gambar berikut ini, berapa banyak huruf **R** yang ada?  
Buatlah Algoritmanya!

	0	1	2	3
0	r	R	B	r
1	a	r	R	z
2	M	R	r	R
3	R	O	t	r

# Latihan 2

- Buatlah Agoritma untuk mengkonversi Nilai Angka ke nilai Huruf seperti pada table berikut ini

Nilai Angka	Nilai Mutu		
	Nilai Huruf	Nilai Setara	Kualifikasi
$80 < N \leq 100$	A	4	Sangat Baik
$73 < N \leq 80$	B+	3,5	Lebih dari Baik
$65 < N \leq 73$	B	3	Baik
$60 < N \leq 65$	C+	2,5	Lebih dari Cukup
$50 < N \leq 60$	C	2	Cukup
$39 < N \leq 50$	D	1	Kurang
$N \leq 39$	E	0	Gagal