

# Dasar Pemrograman

Tim Ajar Dasar Pemrograman 2024

# 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 [C2, A3] – Pekan 1-2
- Mahasiswa memahami dan mampu menjelaskan tentang Tipe Data, Variabel, Input-output, Sequence, Operator (Penugasan Aritmatika, Penugasan Gabungan, Increment, Decrement, Relasional, Logika, Kondisional, Bitwise, Casting) dan mampu menerapkannya dalam penulisan algoritma [C4, A3] – Pekan 3
- Mahasiswa mampu menjelaskan tentang konsep pemilihan sederhana dan bersarang, serta mampu menuliskan algoritma pemilihan untuk menyelesaikan studi kasus dengan menggunakan flowchart [C4, A3, P2] – Pekan 5-6
- Mahasiswa mampu memahami algoritma perulangan sederhana dan Array 1, serta mampu menggambarkan permasalahan studi kasus dengan menggunakan flowchart [C4, A3, P2] – Pekan 7-8
- Mahasiswa mampu memahami konsep array 2 dimensi dan perulangan bersarang, serta mampu menyelesaikan studi kasus dengan penggunaan Array – Pekan 10-11
- Mahasiswa mampu menjelaskan konsep fungsi iteratif dan rekursif, membuat / mendeklarasikan fungsi, pemanggilan fungsi serta mampu menerapkannya dalam menyusun algoritma penyelesaian masalah [C4, A3, P2] – Pekan 13-14

# Pokok Bahasan

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



# Pengantar – Kontrak Kuliah...(1)

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

# Pengantar – Kontrak Kuliah...(2)

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

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

# **Pertemuan 1**

## **DASAR PEMROGRAMAN**

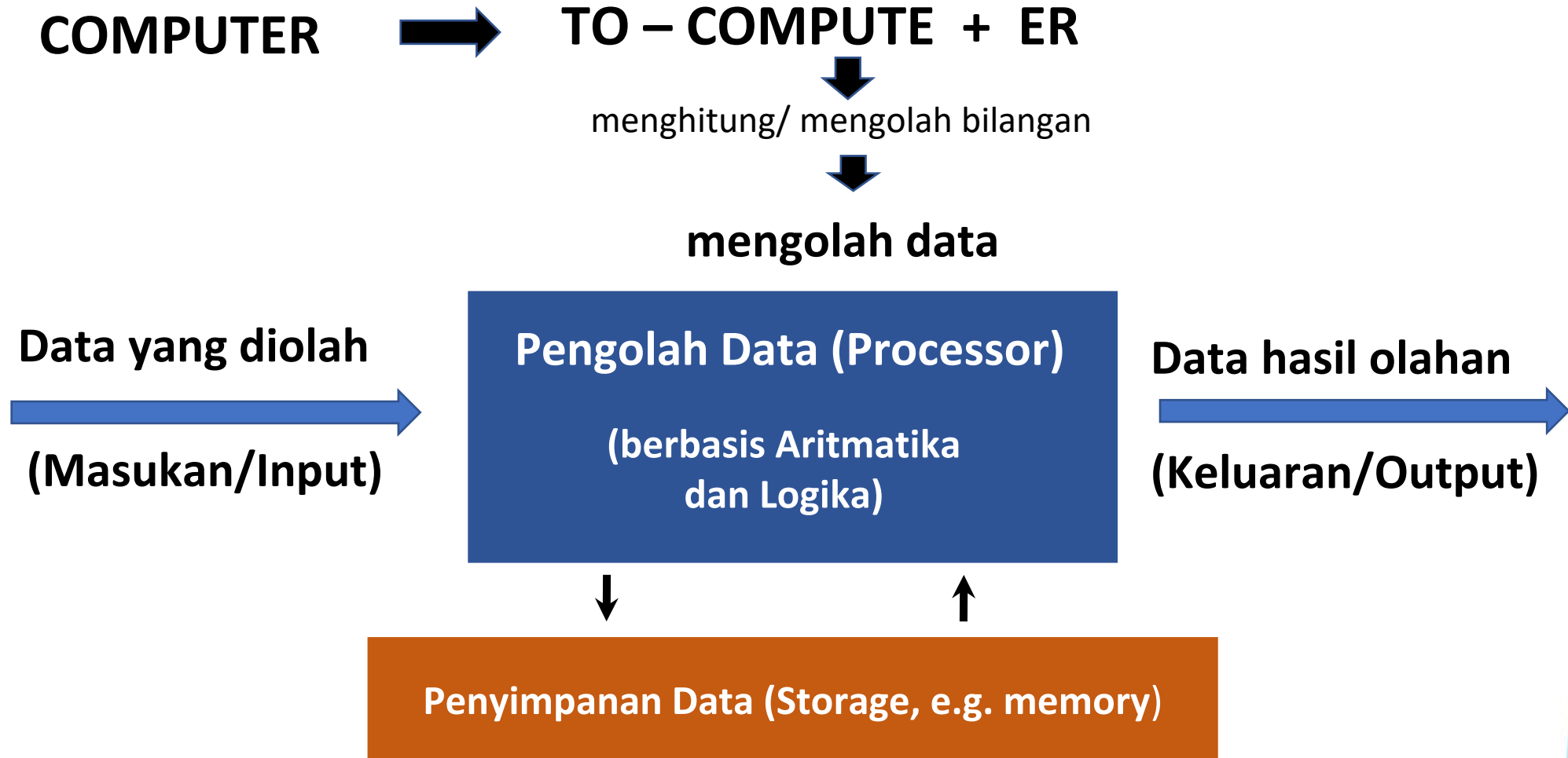


# Tujuan

**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

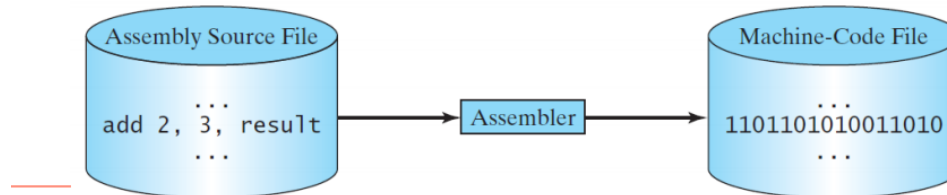
- Program komputer atau yang sering disebut perangkat lunak (*software*) merupakan instruksi untuk komputer.
- Anda dapat memberikan perintah kepada komputer melalui program.
- Tanpa adanya program, komputer adalah mesin kosong yang tidak bisa diberikan instruksi.
- Karena komputer tidak mengerti bahasa manusia maka cara berkomunikasi dengan komputer 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 menerjemahkan 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, dsb

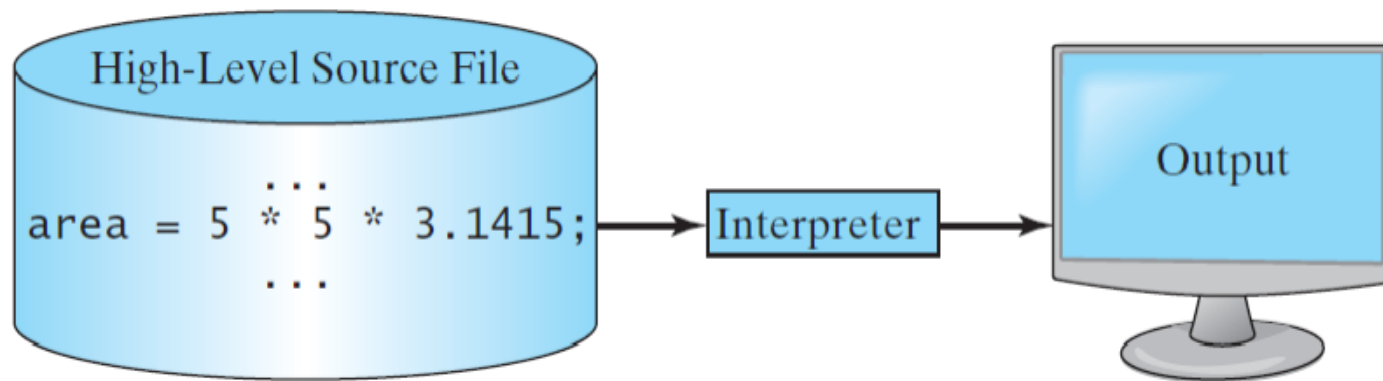
```
1 public class MyFirstJava{  
2  
3     public static void main(String[] args){  
4  
5         System.out.println("Hello Word");  
6  
7     }  
8 }
```

# 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 komputer
- Translasi/ penterjemahan *source code* ke bahasa mesin dapat dilakukan menggunakan *tools* pemrograman lain yang disebut interpreter atau *compiler*.

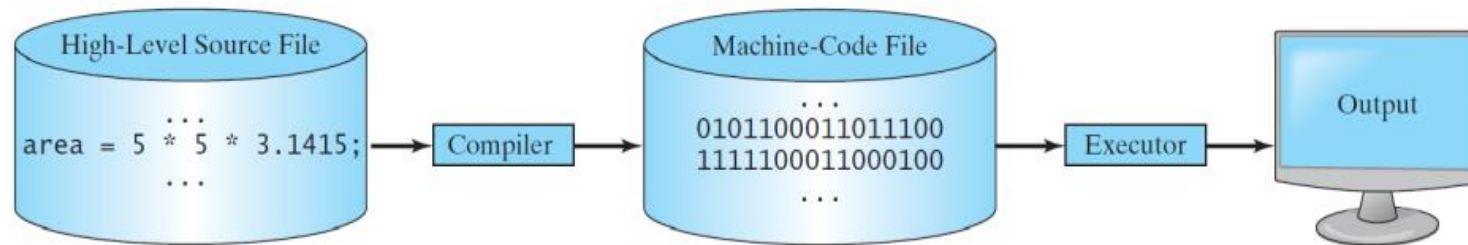
# Interpreting Source Code

- 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 dieksekusi.
- Cara kerja *compiler*:





# Algoritma

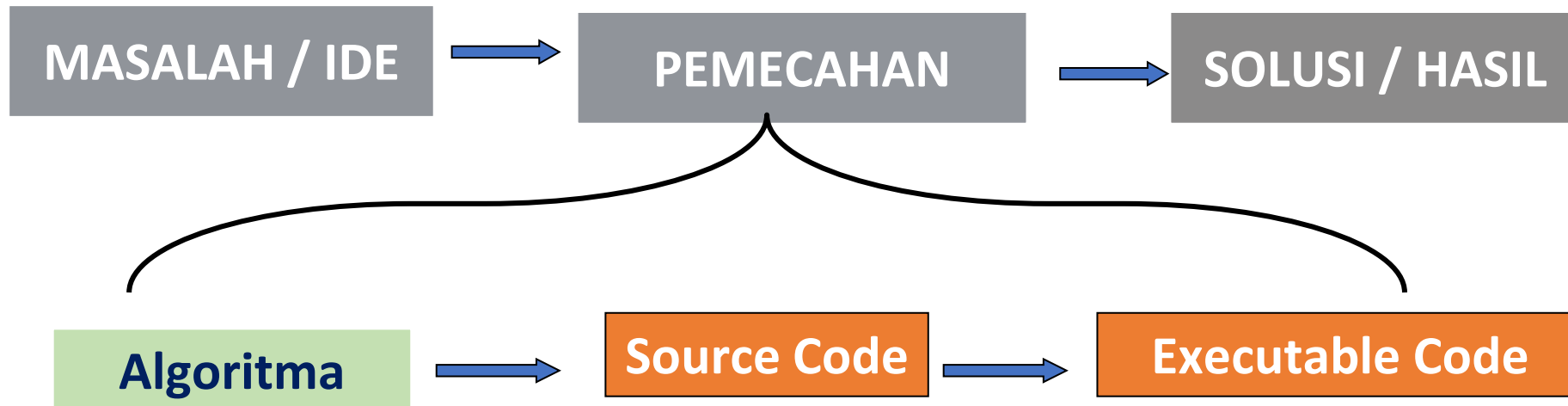
# Algoritma...(1)

- 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. Pernyataan dieksekusi secara berurutan satu demi satu, dari atas ke bawah.

## 2. Struktur pemilihan

Digunakan untuk program yang menggunakan pemilihan atau penyeleksian kondisi.

## 3. Struktur perulangan

Digunakan untuk program yang pernyataannya akan dieksekusi berulang-ulang.

# Karakteristik Algoritma

- **Input**
  - Terdapat inputan bernilai nol ataupun lebih banyak sebagai sumber yang berasal dari luar program. Input digunakan untuk memproses data dan menghasilkan keluaran yang diinginkan
- **Output**
  - Algoritma harus menghasilkan minimal satu nilai sebagai keluaran yang merupakan hasil dari pemrosesan input sesuai dengan tujuan algoritma.
- **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 dengan menggunakan sumber daya yang tersedia

# Contoh Algoritma

Masalah: menyetrika 5 buah pakaian

- Input: pakaian
- Output: pakaian yang sudah disetrika
- Data lain: setrika, alas setrika
- Proses
  - Siapkan alas untuk menyetrika
  - Hubungkan kabel setrika ke listrik
  - Saat setrika sudah panas mulailah menyetrika
  - Jika sudah disetrika kemudian lipat pakaian tersebut
  - Ulangi langkah menyetrika sampai semua pakaian habis
  - Lipat alas dan cabut kabel setrika dari listrik

# 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.

# Teknik Kunci dalam Computational Thinking

## Dekomposisi

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

## Abstraksi

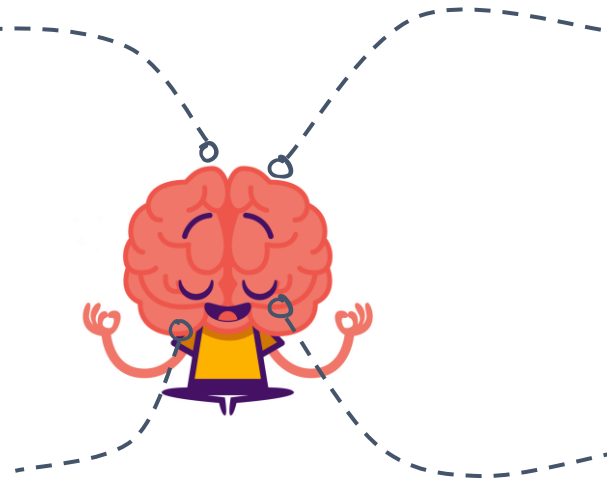
fokus pada informasi penting saja, dan mengabaikan detail yang kurang relevan.

## Pattern/Pola

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

## Algoritma

Membangun langkah-langkah solusi dari masalah



# Contoh soal CT

- Si berang-berang ingin membuat kode rahasia untuk berkomunikasi. Kode ini dibuat dengan mengganti setiap huruf dengan angka sesuai dengan tabel berikut:

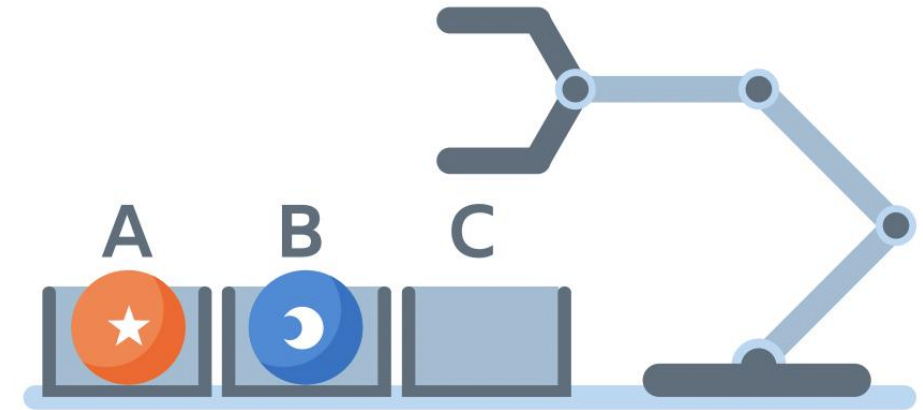
A	B	C	D	E	F	G	H	I	J	K	L	M
	2			5							12	13
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
		16						22				26

Jika kode yang dibuat oleh ninja tersebut adalah sebagai berikut, pesan apa yang ingin disampaikan oleh si berang-berang?

Kode	9	12, 15, 22, 5	10, 20, 9
Isi pesan			

# 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 ke dalam nampan B.



Ketika lengan robot selesai, manakah pernyataan berikut yang benar?

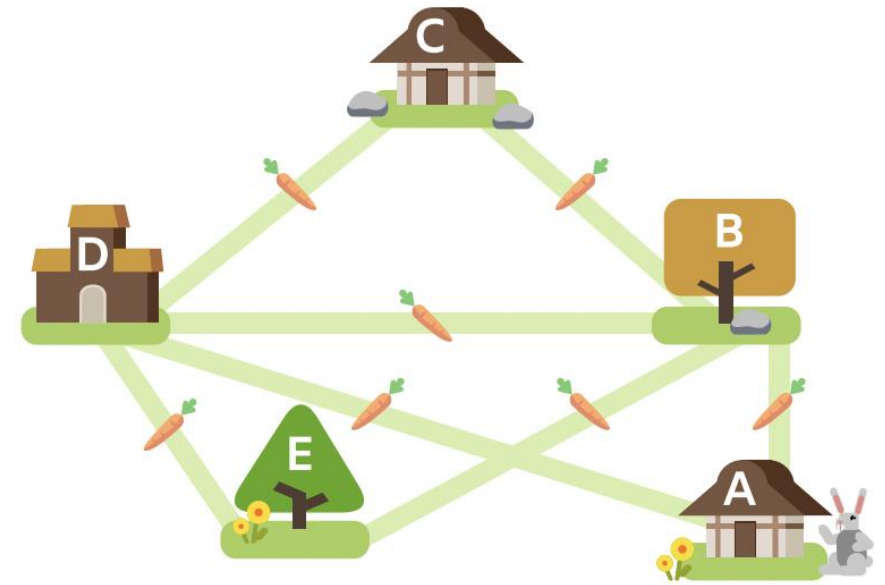
A. Ada dua bola di nampan A	B. Tidak ada yang berubah. Setiap bola kembali ke tempat awal
C. Nampan A kosong	D. Nampan C kosong

# 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?

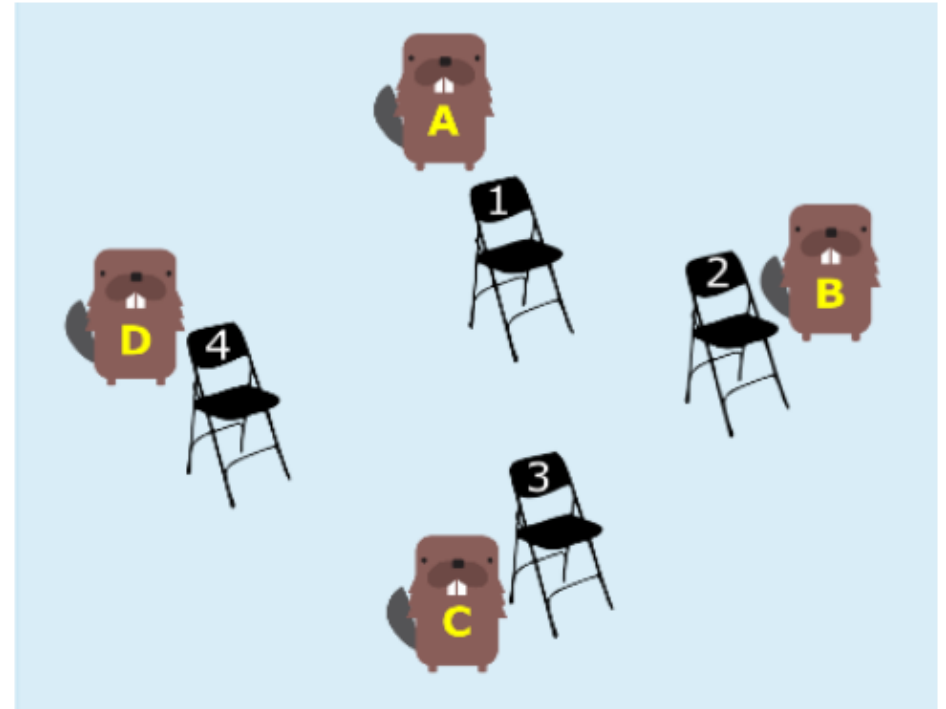


Manakah Rute berikut ini yang paling optimal?

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

# Contoh soal CT - Perulangan

- Sebuah kelompok berjumlah 4 berang-berang memainkan "kursi-musik" yaitu berpindah kursi saat musik dimainkan.
- Saat musik dimulai, setiap berang-berang harus berpindah ke kursi searah dengan putaran jarum jam. Satu kursi dapat ditempati oleh lebih dari satu berang-berang.
- Pada setiap putaran, berang-berang (A) akan berpindah tiga (3) kursi. Berang-berang (C) akan berpindah dua (2) kursi, sedangkan berang-berang (B, D) hanya akan berpindah satu (1) kursi. Semua berang-berang berpindah searah jarum jam.



Jika pada awalnya posisi masing-masing sebagaimana terlihat pada gambar di atas, kursi mana yang kosong pada putaran ke-3?

**A. 1**

**A. 2**

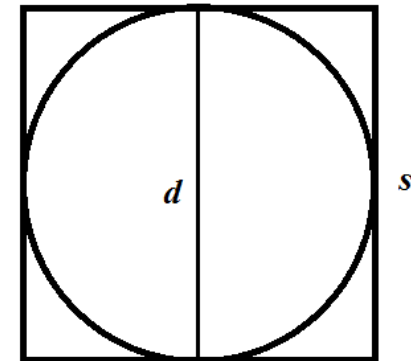
**A. 3**

**A. 4**

# Algoritma - Runtutan

Pak ahmad memiliki lahan berbentuk persegi dengan sisi 100m, di dalam lahan pak ahmad terdapat taman bunga berbentuk lingkaran. Berapakah luas lahan pak ahmad yang tidak ditanami bunga? Buatlah algoritmanya.

- Jawab:
- Input: Lahan dengan sisi = 100m
- Output: luas lahan yang tidak ditanami bunga
- Proses:
  - Luas keseluruhan lahan (persegi)
    - $= \text{sisi} \times \text{sisi} = 10.000 \text{ m}$
  - Luas ditanam bunga (lingkaran)
    - $= \pi * r * r = 3.14 * 50 * 50 = 7850 \text{ m}$
  - Luas yang tidak ditanami bunga = luas persegi – luas lingkaran
    - $10.000 - 7850 = 2.150 \text{ m}$



# Algoritma - Pemilihan

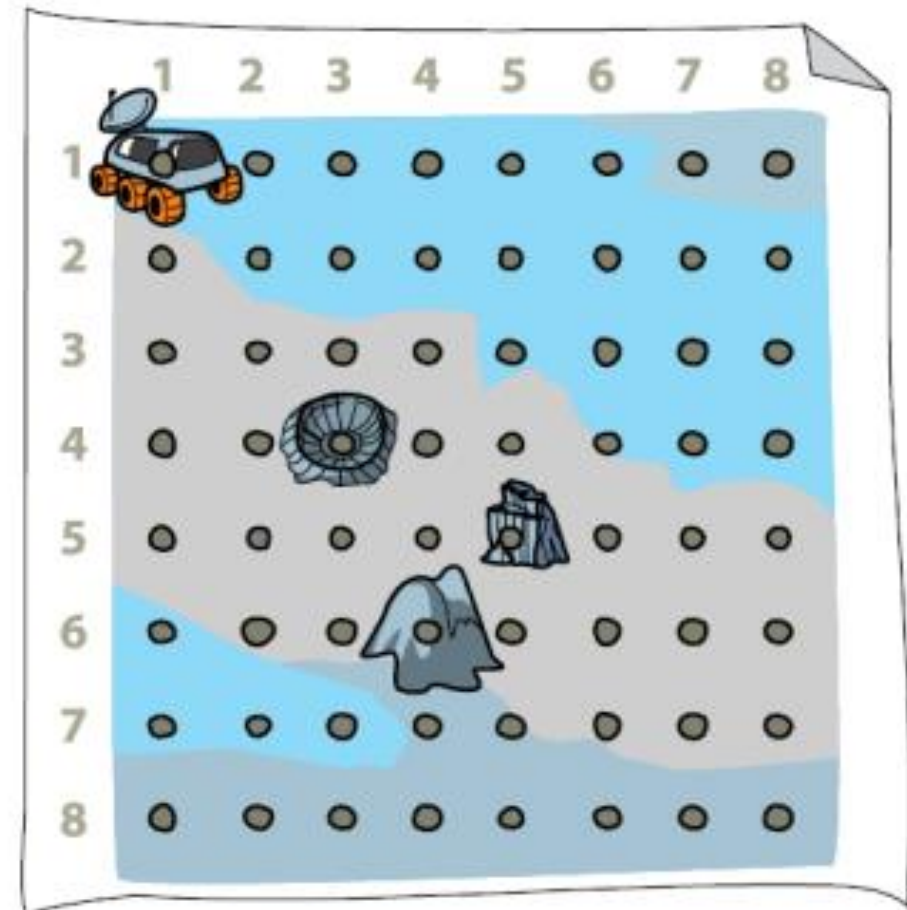
Saat ujian akhir terdapat ketentuan bahwa jika mahasiswa memiliki nilai di bawah 70 maka mahasiswa tersebut mengikuti ujian ulang. Andi ternyata mendapatkan nilai 90. apakah Andi mengikuti ujian ulang? Buatlah algoritmanya.

- Jawab :
- Input : nilai mahasiswa
- Proses :
  - Masukkan nilai mahasiswa
  - Jika nilai mahasiswa  $< 70$ , maka mahasiswa tersebut harus mengikuti ujian ulang
  - Jika nilai mahasiswa  $> 70$ , maka mahasiswa tersebut tidak perlu ujian ulang
  - Output ujian ulang / tidak
- Output: remidi / tidak
- 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. 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

Doni memiliki dua gelas yang berisi air dengan warna berbeda:

- Gelas A berisi air merah.
- Gelas B berisi air biru.

Doni ingin menukar isi kedua gelas tersebut sehingga:

- Gelas A berisi air biru.
- Gelas B berisi air merah.

Namun, Doni hanya memiliki satu gelas kosong tambahan, yaitu gelas C, yang bisa digunakan untuk membantu pertukaran ini. Buat algoritma bagaimana cara Doni menukar isi kedua gelas A dan B tersebut

# Latihan 2

- Anda memiliki sebuah daftar angka berikut: [4, 7, 2, 9, 3]. Buatlah algoritma untuk mencari angka yang terkecil dalam daftar tersebut.

# Latihan 3

- 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 4

- Buatlah algoritma untuk menghitung nilai akhir yang terdiri dari nilai tugas, UTS dan UAS dengan proporsi 30%, 30%, dan 40%. Lalu mengkonversi nilai akhir tersebut yang berupa angka ke nilai huruf seperti pada tabel berikut:

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