

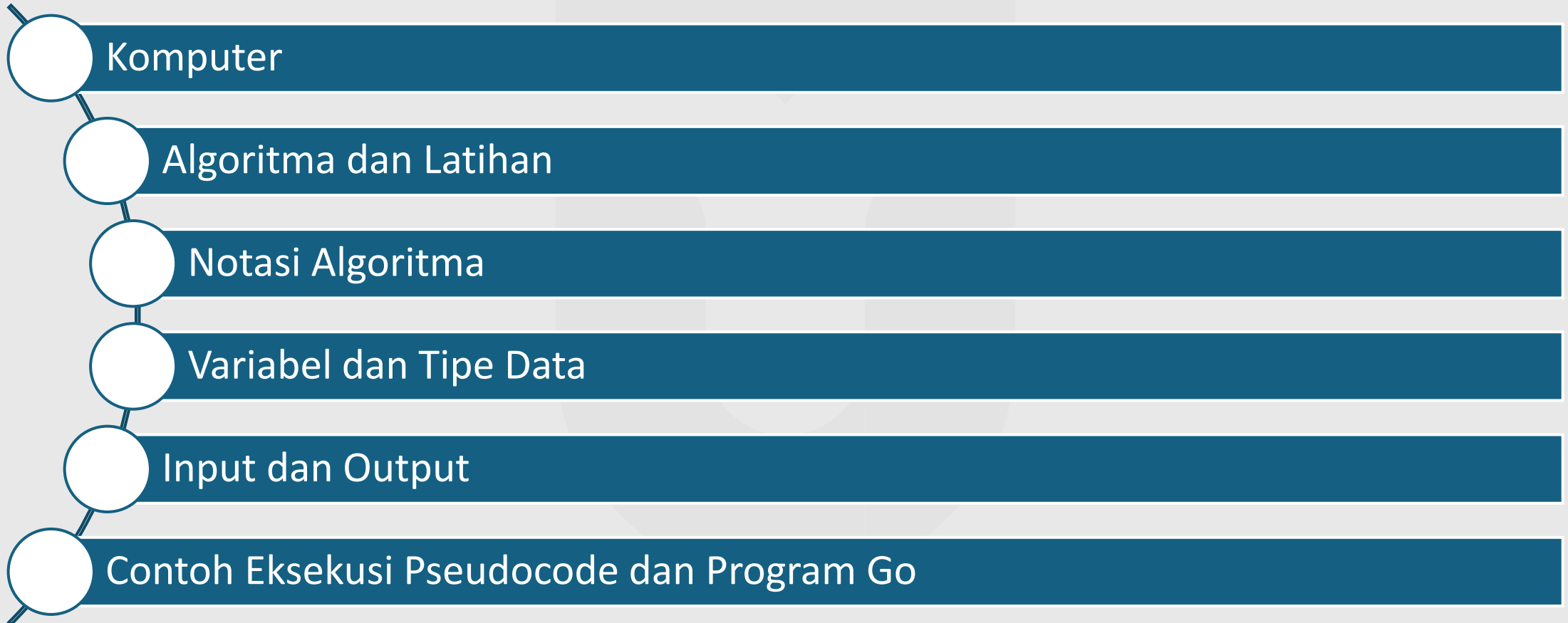
# Pengenalan Paradigma Pemrograman, Konsep Tipe Data dan I/O

ALGORITMA & PEMROGRAMAN 1 (CAK1BAB3)

Pertemuan 01 - Prodi S1 Informatika ,  
Fakultas Informatika, Universitas Telkom



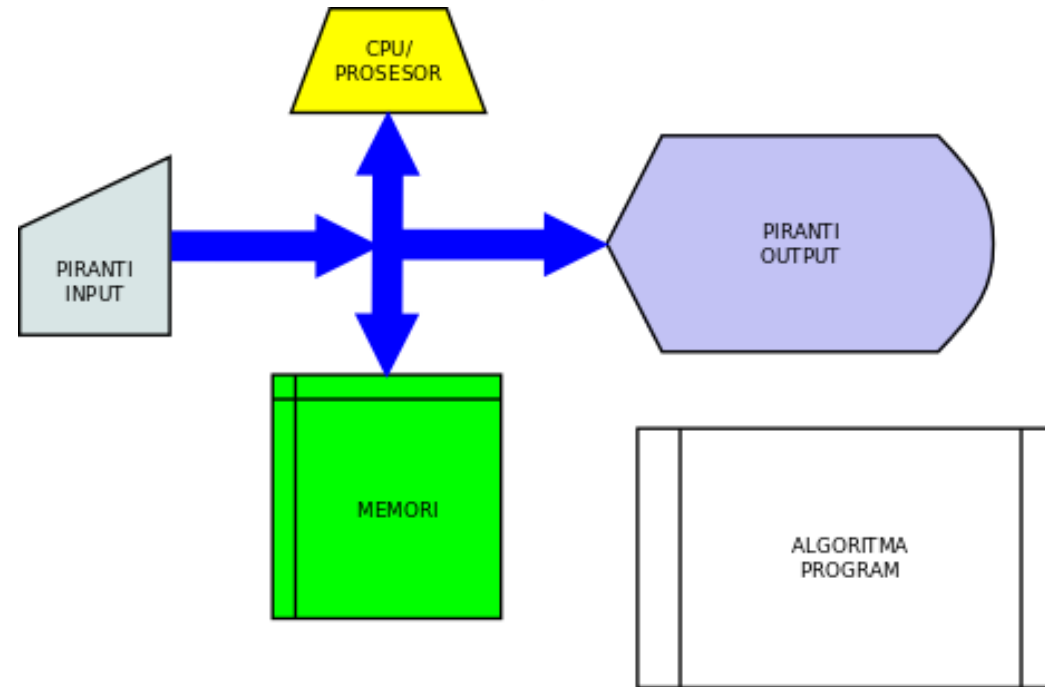
# Outline





# Model Komputer

# Model Kompute & RAM



## Piranti Input

- Perangkat yang digunakan computer untuk menerima data dari pengguna

## Memori

- Untuk penyimpanan data yang akan atau dihasilkan dari suatu proses

## Prosesor

- Eksekutor setiap perintah/instruksi yang diberikan melalui suatu program komputer

## Piranti Output

- Perangkat yang digunakan untuk menampilkan data atau informasi

## Program Komputer

- Kumpulan instruksi atau tugas yang dieksekusi oleh komputer

# Paradigma Pemrograman

## Pemrograman

- Proses **membangun** sebuah **program komputer** yang dapat dieksekusi oleh komputer untuk **tugas tertentu**.

## Program Komputer

- **Program komputer** berisi **kumpulan instruksi** untuk **memanfaatkan sumber daya yang ada pada komputer** seperti keyboard, monitor, kamera, printer, hardisk, CPU, GPU, dan lain sebagainya **untuk melaksanakan tugas tertentu**.

## Bahasa Pemrograman

- Media berkomunikasi dengan komputer yang terdiri dari **lexis (kosakata)** serta **syntax (aturan)** penyusunan kata-kata tersebut untuk menjadi sebuah **instruction (kalimat)** yang valid.

# Paradigma Pemrograman

- Belajar pemrograman  $\neq$  Belajar bahasa pemrograman
- Belajar memprogram adalah belajar tentang strategi pemecahan masalah, metodologi dan sistematika pemecahan masalah tersebut kemudian menuangkannya dalam suatu notasi yang disepakati bersama. Lebih bersifat pemahaman persoalan, analisis, dan sintesis.
- Belajar bahasa pemrograman adalah belajar memakai suatu bahasa berdasarkan tata bahasa dan memanfaatkan instruksi-instruksi yang dapat dipakai. Lebih bersifat keterampilan

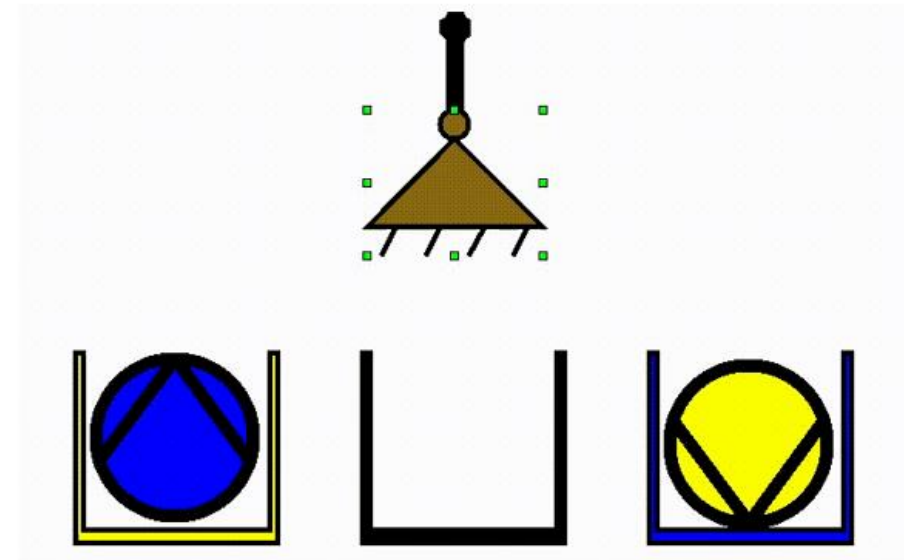
# Algoritma

# Algoritma (Animasi)

## Definisi:

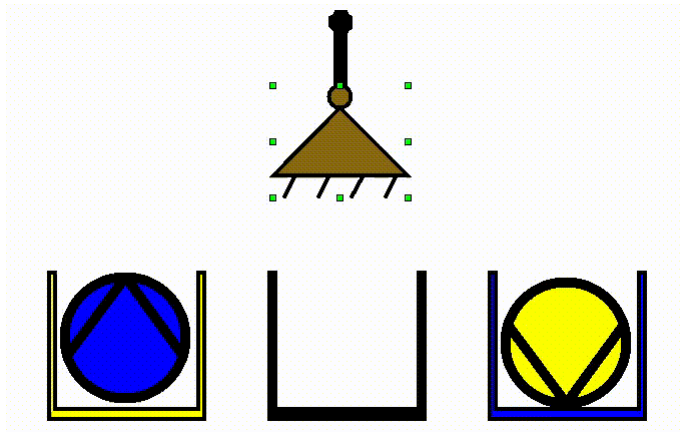
Algoritma adalah urutan instruksi yang jika diikuti dapat digunakan untuk mencapai suatu tujuan tertentu atau menyelesaikan suatu masalah.

## Contoh Kasus:





# (Animasi)



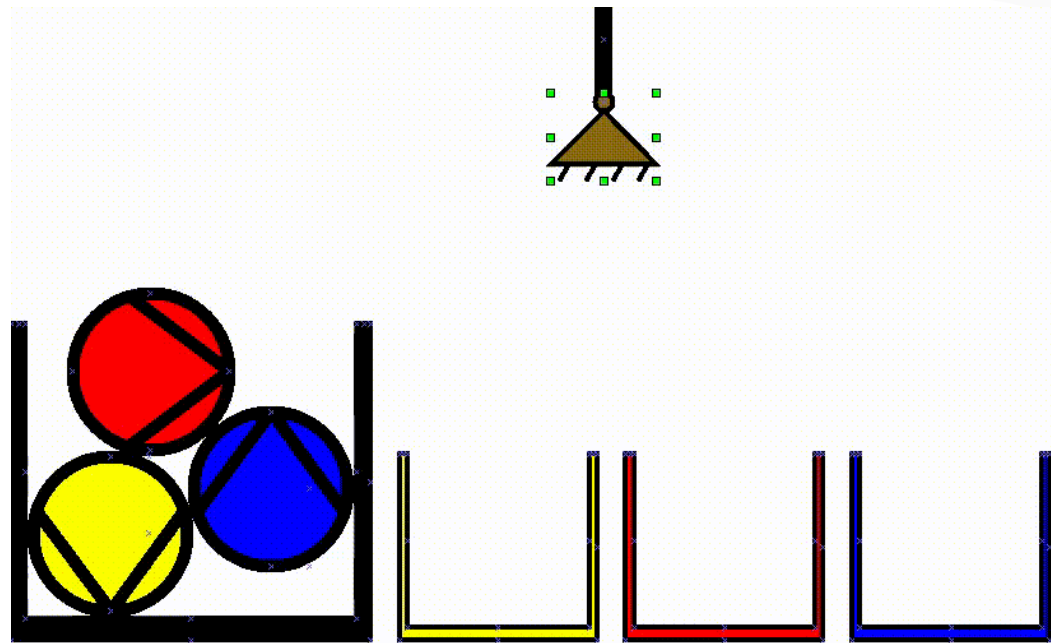
## Tukar Isi Dua Kotak:

1. Ambil bola dari kotak kuning dan masukan kedalam kotak hitam
2. Ambil bola dari kotak biru dan masukan kedalam kotak kuning
3. Ambil bola dari kotak hitam dan masukan kedalam kotak biru

# Algoritma

- Urutan instruksi yang jika diikuti akan mencapai suatu tujuan tertentu
- Algoritma harus mempunyai nama yang sesuai
- Instruksi adalah:
  - Instruksi dasar yang sudah langsung dipahami oleh pelaksana
  - Algoritma bernama yang sudah didefinisikan sebelumnya
- Selalu menghasilkan efek neto, yaitu beda antara status akhir terhadap status awal
- Eksekusi secara sekuensial dari instruksi pertama dan berakhir pada instruksi terakhir

# Contoh : Rotasi isi Kotak (Animasi)



## Algoritma:

1. Ambil bola dari kotak hitam dan masukan kedalam kotak biru
2. Ambil bola dari kotak hitam dan masukan kedalam kotak kuning
3. Ambil bola dari kotak hitam dan masukan kedalam kotak merah
4. Tukar isi kotak kuning dan merah
5. Tukar isi kotak merah dan biru

# Contoh : Pemindahan isi Kotak

## Pindahkan 100 bola: ■

1. Ambil bola dari kotak hitam dan masukan kedalam kotak oranye
2. Ambil bola dari kotak hitam dan masukan kedalam kotak oranye
3. Ambil bola dari kotak hitam dan masukan kedalam kotak oranye
4. Ambil bola dari kotak hitam dan masukan kedalam kotak oranye
5. ...

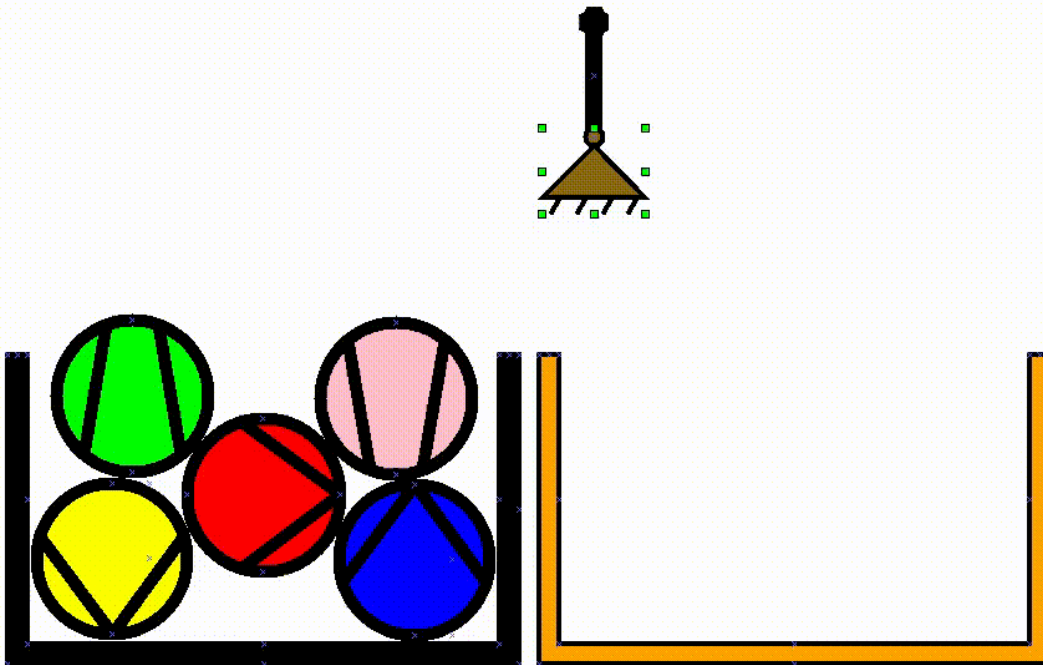
## Pindahkan 100 bola: ✓

1. Ulangi 100 kali:
  - a) Ambil bola dari kotak hitam dan masukan kedalam kotak oranye

# Contoh : Pemindahan isi Kotak (Animasi)

## Algoritma:

1. Ulangi sampai kotak hitam kosong:
  - a) Ambil bola dari kotak hitam dan masukan kedalam kotak oranye



# Contoh : Pembagian Isi Kotak

## Pindahkan bola kuning: ■

1. Ambil bola dari kotak hitam dan masukan kedalam kotak kuning

## Pindahkan bola merah: ■

1. Ambil bola dari kotak hitam dan masukan kedalam kotak merah

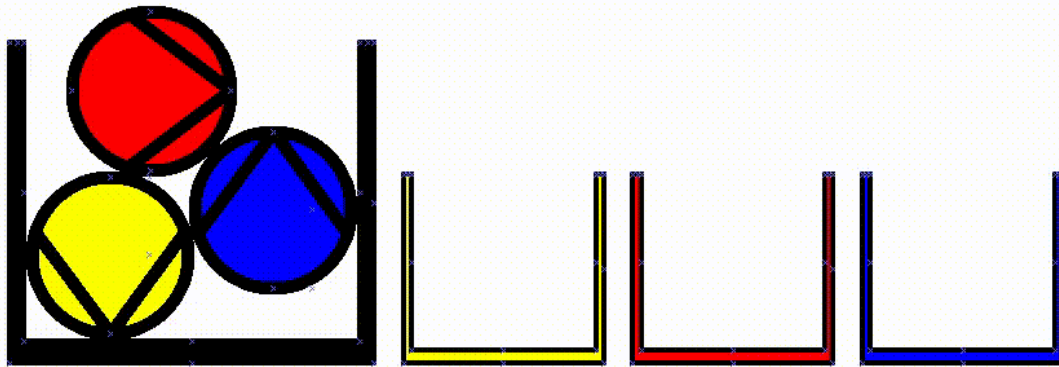
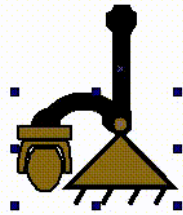
## Pindahkan bola biru: ■

1. Ambil bola dari kotak hitam dan masukan kedalam kotak biru

## Pindahkan bola sesuai warna: ✓

1. Ambil bola dari kotak hitam
2. Bergantung warna bola:
  - a) Jika kuning, masukan kedalam kotak kuning
  - b) Jika merah, masukan kedalam kotak merah
  - c) Jika biru, masukan kedalam kotak biru

# Contoh : Pembagian Isi Kotak (Animasi)



## Bagikan Isi Kotak:

1. Ulangi sampai kotak hitam kosong:
  - a) Ambil bola dari kotak hitam
  - b) Bergantung warna bola:
    - Jika kuning, masukan kedalam kotak kuning
    - Jika merah, masukan kedalam kotak merah
    - Jika biru, masukan kedalam kotak biru

# Latihan Soal



# Soal 1. Lemari Baju

Buatlah langkah-langkah atau algoritma untuk menghitung volume lemari baju di kamar anda.

Asumsi: lemari berbentuk balok, peralatan yang tersedia adalah penggaris, pensil dan kertas.

## Soal 2. Membuat Kopi

Terdapat sebuah dispenser air panas, sebuah sendok, wadah berisi gula dan wadah berisi kopi, dan beberapa cangkir kertas. Tuliskan langkah-langkah atau algoritma untuk membuat kopi hingga ada bahan yang habis.

Asumsi: air panas dan cangkir selalu tersedia.

## Soal 3. Bola Terurut

Terdapat sebuah kotak berwarna hitam di atas meja, sehingga kita tidak bisa melihat isi dari kotak. Salah satu sisinya terdapat lubang seukuran pergelangan tangan untuk bisa mengambil benda yang ada di dalamnya. Di dalam bola terdapat tiga buah bola berukuran sama, dan masing-masing bola terdapat label dengan angka tertentu.

Tuliskan langkah-langkah untuk menyusun tiga bola tersebut terurut berdasarkan label angkanya di luar kotak.

Asumsi: setiap pengambilan hanya boleh satu bola. Diperbolehkan memberi asumsi lain yang dirasa perlu.

## Soal 4. Bola Terberat

Terdapat empat buah bola yang tampilannya sama. Berat dari masing-masing bola mungkin berbeda. Tersedia sebuah timbangan seperti gambar di bawah, di mana setiap wadah timbangan hanya cukup untuk satu bola.

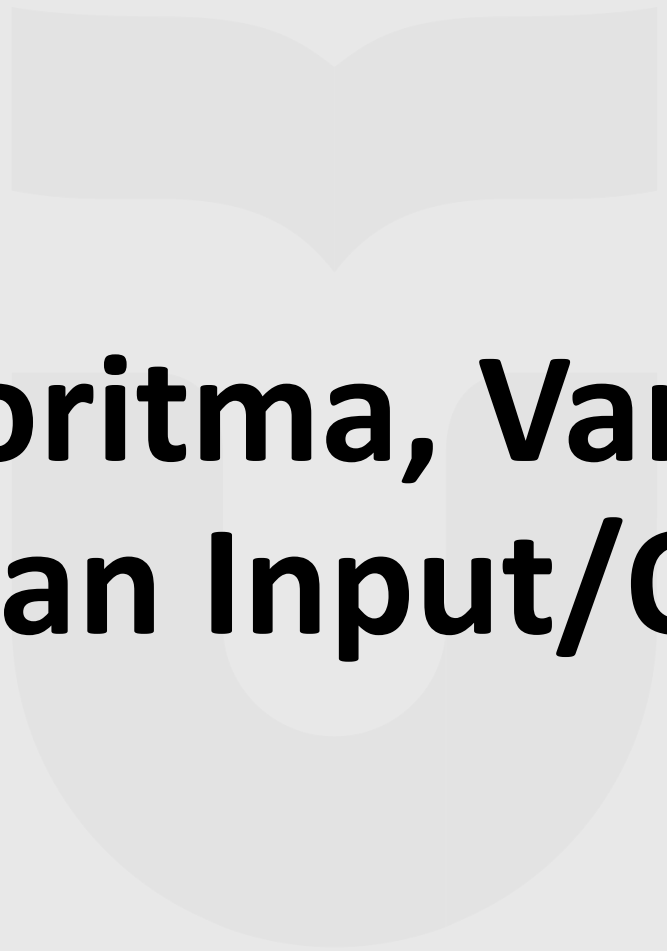
Tuliskan langkah-langkah untuk mencari bola yang paling berat dari 4 bola tersebut.



## Soal 5. Koper

Misalnya anda adalah petugas di bandara yang ditugaskan untuk menghitung rata-rata berat koper yang akan dimasukkan ke dalam pesawat salah satu maskapai penerbangan. Masalahnya komputer yang digunakan sedang rusak, sehingga anda diminta menghitung manual menggunakan kalkulator. Tuliskan algoritma untuk persoalan tersebut!

Asumsi: terdapat sejumlah  $N$  penumpang



# Notasi Algoritma, Variabel, Tipe Data dan Input/Output

# Notasi Algoritma

- Algoritma  $\neq$  Kode Program
- **Pseudocode** adalah notasi yang **disepakati** untuk menuliskan algoritma sehingga mudah dibaca, dipahami dan diimplementasikan menjadi kode program.
- Penulisan algoritma terdiri dari 3 bagian:
  - Judul
  - Deklarasi kamus
  - Instruksi Algoritma

# Contoh Penulisan Pseudocode

Contoh Notasi Pseudocode	Contoh Notasi Bahasa Go
<pre> <b>program</b> &lt;nama program&gt; {deskripsi dari program yang dibuat} <b>kamus/deklarasi</b>     {deklarasi variabel dan tipe data yang digunakan}     ... <b>algoritma</b>     {berisi kumpulan instruksi algoritma tertentu}     ...     ... <b>endprogram</b> </pre>	<pre> <b>package</b> main /* deskripsi dari program yang dibuat */ <b>func</b> main(){     /*deklarasi variabel dan tipe data yang digunakan*/     ...     ...     /*berisi kumpulan instruksi algoritma tertentu*/     ...     ... } </pre>

Komentar tidak akan dianggap sebagai suatu instruksi

Penulisan komentar pada menggunakan {...} atau /\* .... \*/



# Aturan Nama Program

- Dalam bahasa pemrograman, setiap nama mempunyai aturan penulisan. Pada teks algoritma tidak ada aturan ketat.
- Pemilihan nama harus interpretatif, tidak menimbulkan kerancuan, jika singkat harus disertai penjelasan.  
contoh:  $x-y$  akan membingungkan sebab mungkin " $x$  minus  $y$ "
- Nama merupakan satu kesatuan leksikal, maka sebuah nama harus dituliskan secara utuh (tidak dipisah blank) supaya satu nama dapat dibedakan dengan nama lainnya (besaran leksikal lain)

# Tipe Data

Untuk menggunakan suatu tipe data, harus disediakan yang namanya **variable** dengan tipe data yang dimaksud.

<b>Integer</b>	Bilangan bulat, contoh: 1, 1234, -45
<b>Boolean</b>	Representasi logika, yaitu: true, false
<b>Real</b>	Bilangan berkoma, contoh: 3.14
<b>String</b>	Teks, contoh: CII-1A3 Pengenalan Pemrograman
<b>Character</b>	himpunan simbol yang terdefinisi oleh suatu enumerasi, contoh: 'a' direpresentasikan oleh bilangan 97 pada ASCII

# Variabel

Suatu **tempat/lokasi di dalam memori komputer** yang diberi nama

01

## Nama Variable

Dimulai dengan huruf dan dapat diikuti dengan sejumlah huruf, angka, atau garisbawah. **Contoh: ketemu, found, rerata, mhs1, data2,**

02

## Tipe Data Variable

Data yang tersimpan dalam suatu variabel dapat mempunyai tipe apa saja selama diperkenankan oleh bahasa pemrograman yang digunakan.

Tipe data yang umum tersedia adalah **integer, real atau float, boolean, karakter, string, dan alamat memori.**

03

## Akses Nilai Variable

Dengan menyebutkan nama variabel, nilai data yang tersimpan dalam variabel tersebut dapat diambil.

04

## Akses Lokasi atau Alamat Variable

### Pada Bahasa Pemrograman\*

**Lokasi** atau **informasi alamat** dari variabel dapat diperoleh dengan memberikan prefix **&** pada nama variabel tersebut.

Contoh: **&ketemu** akan menghasilkan alamat memori dari variabel **ketemu**

# Deklarasi Variabel



- Variabel harus dideklarasikan dulu sebelum digunakan.
- Variabel juga harus diinisialisasi dulu (diisi data) agar nilai yang tersimpan diketahui dengan jelas dan eksekusi algoritma menjadi terprediksi.



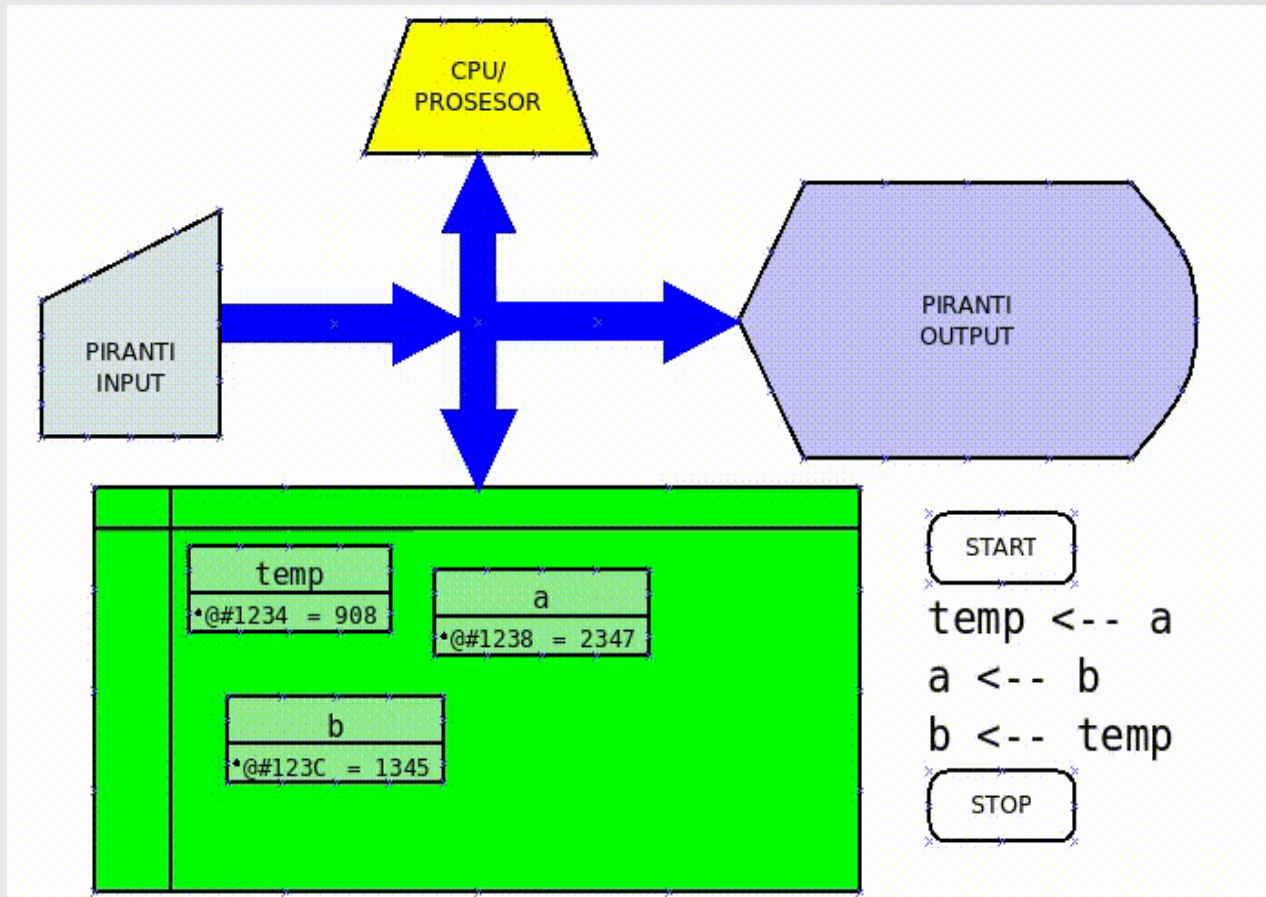
## Deklarasi variable pada kamus

Notasi Pseudocode	Notasi Bahasa Go
<b>kamus</b> suhu : <b>real</b> umur : <b>integer</b> a1, a2, a3 : <b>char</b> nama : <b>string</b> lulus : <b>boolean</b>	<b>var {</b> suhu <b>float64</b> umur <b>int</b> a1, a2, a3 <b>rune</b> nama <b>string</b> lulus <b>bool</b> <b>}</b>

# Assignment Nilai

- **Variabel** adalah suatu lokasi memori yang telah diberi **nama**
- **Assignment** adalah instruksi menyalin data ke dalam **variabel**
- Notasi assignment adalah  
    **<variable> = <value>** atau  
    **<variable> ← <value>**
- Dimana **<variable>** adalah nama dari suatu variabel
- Dan **value** adalah nilai yang akan disimpan ke lokasi tersebut
- Contoh:  
    **data ← 43**  
    **total = total + 37**

# (Animasi)



*Salah satu algoritma yang sangat mendasar*

## Swap isi variabel a dan b:

Dengan bantuan satu variabel temporer untuk menyimpan sementara isi salah satu variabel

Alamat memori dari suatu variabel disimbolkan oleh .@#

# Input & Output

- Instruksi **input/read/scan** : mengambil data dari piranti masukan dan menyimpannya ke suatu lokasi memori

**input**(<daftar variabel>)

- Instruksi **output** : mengirim data ke piranti keluaran

**output**(<daftar data>)

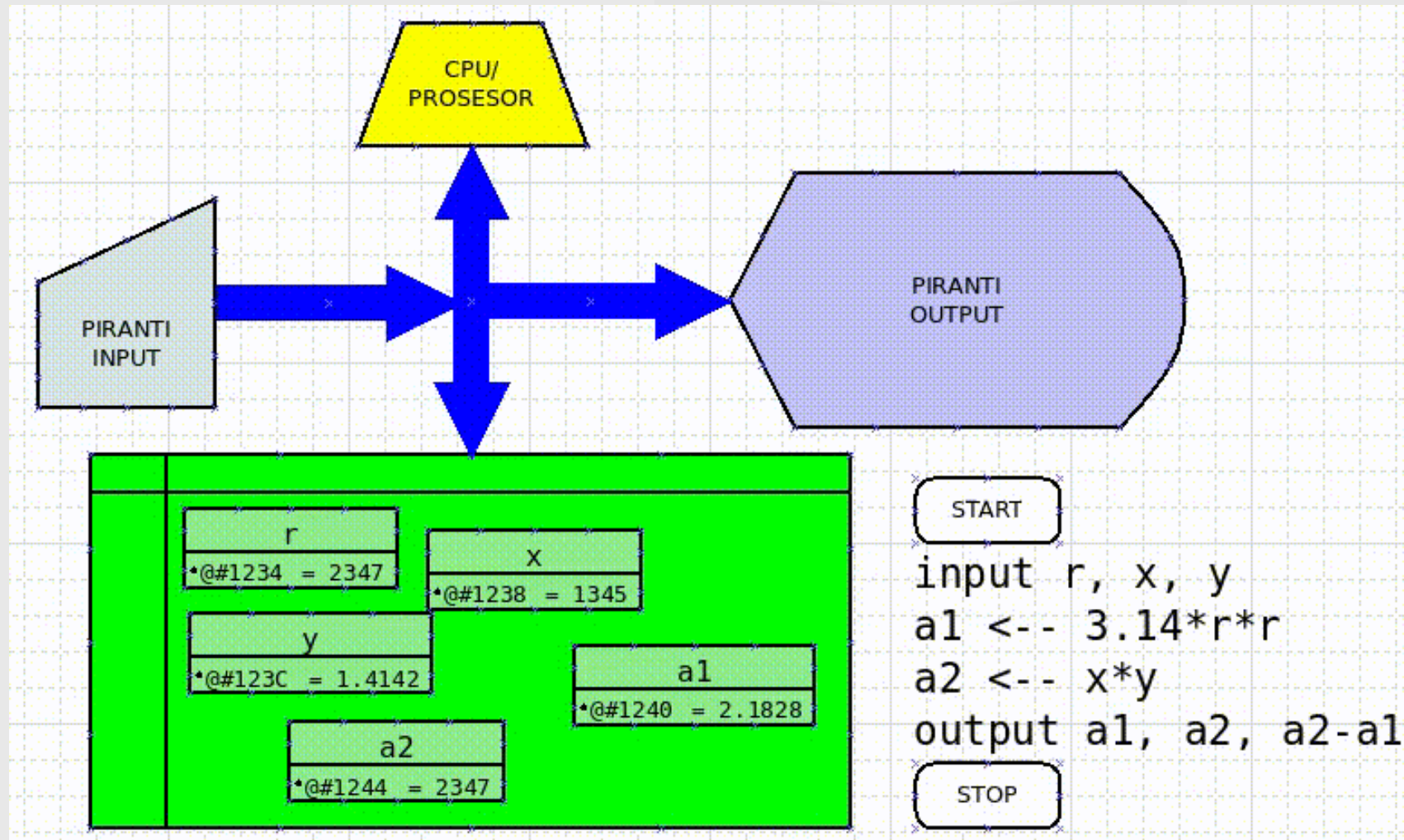
variasi kata kunci: **print/write**

## Contoh

**input** (var\_a, var\_b, var\_c)

**output**("Selamat Datang", var\_a, "S1-IF", var\_b, 3.14, var\_c)

# (Animasi)



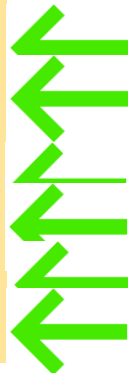


# Ilustrasi Variabel dan Alamat Memori

Apabila diasumsikan sebuah algoritma dapat dieksekusi layaknya kode program maka

## Kode Program

```
kamus
    suhu : real
    umur : integer
algoritma
    suhu ← 36.3
    umur ← 10
    print(suhu, umur)
    print(&suhu, &umur)
```



## Memory / RAM

name	umur
value	10
address	0x05cf

## Piranti Output (Monitor)

36.3 10

0x0f84 0x05cf

name	suhu
value	36.3
address	0x0f84

# Ilustrasi Tracing Program (Animasi)

CONSOLE

OUTPUT

INPUT

MEMORY

PROGRAM

```
b1=-NOT-INITIALIZED-
b2=-NOT-INITIALIZED-
h=-NOT-INITIALIZED-
w=-NOT-INITIALIZED-
a1=-NOT-INITIALIZED-
a2=-NOT-INITIALIZED-
```

```
program BedaLuas
kamus
    b1, b2, h, w : integer
    a1, a2 : integer
algoritma
    read b1, b2, h, w
    a1 <- (b1+b2)*h/2
    a2 <- h*w
    write a1, a2, (a2-a1)
endprogram
```

# Contoh Program Go: Edit, Kompilasi dan Eksekusi (Animasi)

```
[CCH1A4 go]$
```

# Terima Kasih 😊

