KEMENTERIAN
PENDIDIKAN DASAR
DAN MENENGAH

**MODUL 1 BIDANG KECERDASAN ARTIFISIAL**
PEMANDU TALENTA JENJANG SMP/SEDERAJAT

# PENGENALAN LOGIKA, ALGORITMA, DAN *PSEUDOCODE*

BINA TALENTA INDONESIA 2025

# KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, Modul Pemandu Talenta: Pengenalan Logika, Algoritma, dan *Pseudocode* ini dapat diselesaikan. Modul ini disusun sebagai panduan bagi pemandu talenta pada program Bina Talenta Indonesia tahun 2025 yang diselenggarakan oleh Pusat Prestasi Nasional (Puspresnas), Kementerian Pendidikan Dasar dan Menengah (Kemendikdasmen), dan Universitas Negeri Surabaya (Unesa).

Program ini merupakan perwujudan nyata dari visi "Pendidikan Bermutu untuk Semua," sebuah komitmen yang menjamin hak setiap warga negara untuk mendapatkan layanan pendidikan berkualitas, inklusif, dan relevan dengan tuntutan masa depan. Program Bina Talenta Indonesia ini bertujuan untuk meningkatkan kompetensi dan prestasi murid serta menguatkan kapasitas guru sebagai pemandu talenta.

Kritik dan saran yang membangun dari pembaca sangat kami harapkan demi penyempurnaan modul ini di masa mendatang. Kami berharap modul ini dapat membantu pemandu talenta dalam membimbing murid berprestasi dan mempersiapkan generasi yang unggul dan berkarakter.

Terima kasih kepada semua pihak yang telah membantu dan memberi kesempatan dalam penyusunan modul ini.

Surabaya, Oktober 2025

Penulis

# DAFTAR ISI


KATA PENGANTAR 2

DAFTAR ISI 3

A. DESKRIPSI MODUL 4

1. Tujuan Pelatihan 4

2. Indikator Capaian Pelatihan 4

3. Pokok Bahasan 4

4. Alur Penyampaian Materi 4

B. MATERI DAN AKTIVITAS 6

1. Pemanasan 6

2. Algoritma, *Flowchart*, dan *Pseudocode* 6

3. Logika Percabangan dan Perulangan 8

AKTIVITAS 1.1: Menyusun algoritma untuk masalah kehidupan sehari-hari 9

AKTIVITAS 1.2: Menyusun algoritma dan *flowchart* dengan logika percabangan dan perulangan 10

AKTIVITAS 1.3: Algoritma dan *flowchart* permasalah kompleks 11

4. Konversi simbol flowchart, blok Scratch, dan sintaks Python 12

AKTIVITAS 1.4: Menentukan ganjil atau genap 13

AKTIVITAS 1.5: Merapikan pakaian 15

AKTIVITAS 1.6: Petani, kambing, serigala, dan sayur kubis 16

AKTIVITAS 1.7: Robot pembersih sampah 17

AKTIVITAS 1.8: Mengenali hewan 18

5. Refleksi 20

C. LATIHAN 21

DAFTAR PUSTAKA 23

# A. DESKRIPSI MODUL

Modul ini berfokus pada penguatan dasar-dasar berpikir komputasional, mencakup konsep-konsep seperti algoritma dan struktur logika. Modul ini melatih peserta untuk memformulasikan alur penyelesaian masalah menggunakan diagram alir (*flowchart*) dan *pseudocode*. Setelah memahami modul ini peserta diharapkan mampu merancang solusi yang terstruktur dan sistematis sebelum diubah menjadi kode yang dapat dieksekusi oleh komputer.

## 1. Tujuan Pelatihan

a. Memahami konsep dasar algoritma dan logika sebagai alat untuk penyelesaian masalah.
b. Membuat penyelesaian masalah menggunakan *flowchart* dan *pseudocode*.
c. Menerapkan struktur percabangan dan perulangan untuk menyelesaikan kasus sederhana dan kompleks.

## 2. Indikator Capaian Pelatihan

a. Mampu menjelaskan perbedaan antara algoritma, *flowchart*, dan *pseudocode*.
b. Mampu menyusun algoritma untuk penyelesaian masalah sehari-hari.
c. Mampu menulis *flowchart* dan *pseudocode* untuk algoritma berurutan, percabangan, dan perulangan sederhana.

## 3. Pokok Bahasan

a. Konsep algoritma, *flowchart*, dan *pseudocode*.
b. Struktur kontrol percabangan dan perulangan.
c. Latihan menyusun algoritma dan *flowchart* dengan logika percabangan dan perulangan.

## 4. Alur Penyampaian Materi

| Topik | Pembelajaran Mendalam | Aktivitas | Setoran | JP |
|---|---|---|---|---|
| Pengertian, konsep algoritma, logika, | Memahami | Pengajar menjelaskan konsep algoritma, logika, dan *pseudocode* | | 1 |

| | | | | |
|---|---|---|---|---|
| *pseudocode* | Menerapkan | Peserta mengidentifikasi penggunaan algoritma dalam kehidupan sehari-hari dan menyusunnya dalam urutan perintah yang runtut dan logis. | AKTIVITAS 1.1 | |
| Logika percabangan dan perulangan dalam kehidupan sehari-hari | Memahami | Peserta menjelaskan logika percabangan dan perulangan dalam kehidupan sehari-hari | | 1 |
| | Menerapkan | Peserta mengidentifikasi logika kondisi pada kasus tertentu dan dapat menyusun dalam bentuk algoritma atau *flowchart*. | AKTIVITAS 1.2<br>AKTIVITAS 1.3 | |
| Penerapan algoritma dalam penyelesaian masalah | Memahami | Pengajar memberikan contoh-contoh nyata penggunaan algoritma | | 1 |
| | Menerapkan | Peserta mengidentifikasi dan menyelesaikan masalah. | AKTIVITAS 1.4<br>AKTIVITAS 1.5<br>AKTIVITAS 1.6<br>AKTIVITAS 1.7<br>AKTIVITAS 1.8 | |
| | Merefleksi | | Jurnal Refleksi | |

# B. MATERI DAN AKTIVITAS

## 1. Pemanasan

Di kehidupan sehari-hari, pernahkah kita diminta untuk melakukan sesuatu yang berurutan? Atau melakukan sesuatu yang berdasarkan kondisi tertentu? Atau bahkan melakukan sesuatu yang berulang-ulang? Aktivitas-aktivitas sehari-hari yang kita lakukan secara berurutan, berdasarkan kondisi tertentu, atau berulang-ulang adalah persoalan yang dapat diselesaikan dengan algoritma.

Bayangkan kita akan membuat secangkir teh, bagaimanakah prosesnya?

1. Siapkan cangkir, bubuk teh, gula, dan air panas.
2. Masukan bubuk teh dan gula ke dalam cangkir.
3. Tuangkan air panas secukupnya.
4. Aduk hingga bubuk teh dan gula larut.
5. Teh siap diminum.

Contoh yang lain ketika kita ingin menyeberang pada perlintasan jalan, bagaimanakah prosesnya?

1. Lihat lampu lalu lintas pejalan kaki.
2. Jika lampu berwarna hijau, maka kita boleh menyeberang.
3. Jika lampu berwarna merah, maka kita menunggu di tepi jalan atau tidak boleh menyeberang.

Bayangkan ketika kita akan mencuci tumpukan piring kotor, bagaimanakah prosesnya?

1. Siapkan sabun cuci piring dan pastikan air kran mengalir.
2. Ambil piring kemudian cuci dengan sabun.
3. Bilas piring dengan air kran mengalir.
4. Keringkan piring yang telah dibilas.
5. Ulangi langkah kedua hingga tidak ada lagi piring kotor.

## 2. Algoritma, *Flowchart*, dan *Pseudocode*

Algoritma adalah kumpulan langkah-langkah logis berurutan yang digunakan untuk memecahkan suatu masalah. Algoritma harus bersifat terbatas, artinya langkah-langkah dalam algoritma harus terhenti pada kondisi tertentu. Selain itu algoritma harus jelas, terdapat

masukan (*input*) dan luaran (*output*). Pada pembahasan sebelumnya telah ditunjukkan contoh-contoh algoritma dalam kehidupan sehari-hari seperti proses membuat teh, menyeberang pada perlintasan jalan, dan mencuci tumpukan piring kotor. Tidak terbatas pada contoh-contoh tersebut, aktivitas-aktivitas yang kita lakukan sehari-hari merupakan kumpulan proses atau langkah-langkah yang disebut algoritma.

*Flowchart* adalah diagram yang membantu dalam memvisualisasikan langkah-langkah dalam algoritma. *Flowchart* dibuat berdasarkan simbol-simbol yang memiliki fungsi atau makna tersendiri. Simbol-simbol tersebut disajikan pada Tabel 1.1 di bawah ini. Pembuatan *flowchart* dapat dilakukan secara *online* melalui *website* https://app.diagrams.net/.

Tabel 1.1 Simbol-simbol dasar *flowchart*

| Simbol | Makna |
|---|---|
|  | Simbol untuk memulai dan mengakhiri proses. |
|  | Simbol untuk masukan (*input*) atau luaran (*output*) proses. |
|  | Simbol untuk menunjukkan proses yang terjadi. |
|  | Simbol untuk struktur percabangan. |
|  | Simbol untuk menunjukkan arah alur proses. |

Selain menggunakan *flowchart*, algoritma juga dapat dituliskan menggunakan campuran antara bahasa manusia sehari-hari dan bahasa pemrograman. Penulisan algoritma semacam ini disebut dengan *pseudocode*. *Pseudocode* memiliki makna gabungan dari "pseudo" (semu) dan "code". *Pseudocode* tidak memiliki aturan penulisan yang kaku. *Psudocode* dapat ditulis berdasarkan bahasa pemrograman yang akan diimplementasikan. Contoh *pseudocode* untuk menghitung luas lingkaran adalah sebagai berikut,

```
PROGRAM HitungLuasLingkaran
DEKLARASI:
pi, jariJari, Luas: float
pi ← 3.14
```

```
ALGORITMA:
Luas ← pi * jariJari * jariJari
```

Contoh di bawah ini adalah *pseudocode* untuk menentukan sebuah nilai masuk dalam kriteria "LULUS" atau "TIDAK LULUS",

```
PROGRAM kelulusanMurid
DEKLARASI:
nilai: float
ALGORITMA:
input nilai
if nilai > 60:
   print("LULUS")
else:
   print("TIDAK LULUS")
```

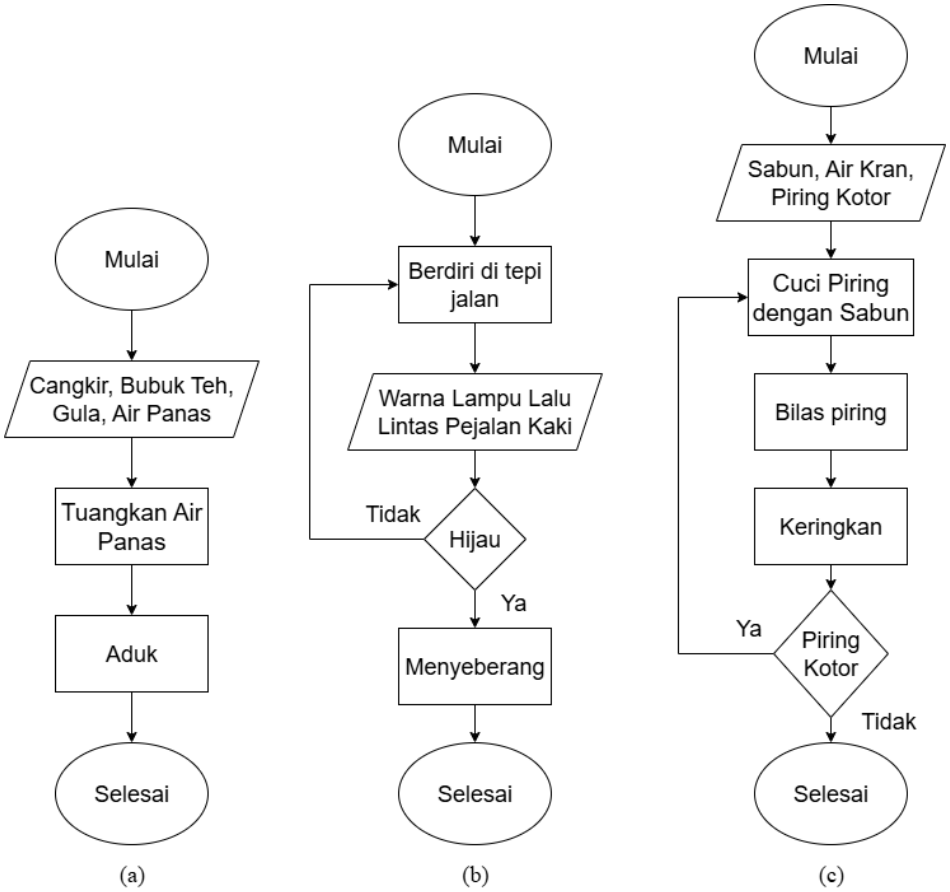
*Pseudocode* penentuan nilai di atas bisa juga dituliskan sebagai berikut

```
PROGRAM kelulusanMurid
BACA nilai
JIKA nilai > 60:
   CETAK "LULUS"
SELAIN ITU:
   CETAK "TIDAK LULUS"
```

## 3. Logika Percabangan dan Perulangan

Logika percabangan adalah salah satu struktur logika yang memungkinkan algoritma untuk membuat keputusan berdasarkan syarat atau kondisi tertentu. Dalam kehidupan sehari-hari kita tidak jarang dihadapkan pada pilihan "Jika A, maka lakukan B, jika tidak A, maka lakukan C". Tanpa struktur logika percabangan program hanya akan menjalankan langkah-langkah secara berurutan. Sedangkan logika perulangan atau dalam istilah koding dikenal sebagai iterasi (*looping*) merupakan salah satu struktur logika yang memungkinkan algoritma untuk menjalankan satu blok instruksi yang sama secara berulang-ulang tanpa harus menulis blok instruksi tersebut berkali-kali.

Pada pembahasan sebelumnya kita telah membuat algoritma untuk membuat secangkir teh. Penggambaran algoritma tersebut dalam bentuk *flowchart* ditunjukkan oleh Gambar 1.1a berikut.




Gambar 1.1. *Flowchart* (a) membuat secangkir teh, (b) menyeberang jalan, dan (c) mencuci piring kotor.

*Flowchart* untuk masalah menyeberang pada perlintasan jalan dan mencuci tumpukan piring ditunjukkan pada Gambar 1.1b dan 1.1c. Perbedaan dari Gambar 1.1a, 1.1b, dan 1.1c di atas adalah alur penyelesaian masalah. Gambar 1.1a menggunakan alur sekuensial (berurutan) untuk menyelesaikan masalahnya, sedangkan Gambar 1.1b dan 1.1c menggunakan kombinasi percabangan dan perulangan.

**AKTIVITAS 1.1**

**Menyusun algoritma untuk masalah kehidupan sehari-hari**

Tujuan

1. Peserta mampu mengenali algoritma pada aktivitas kehidupan sehari-hari.

2. Peserta dapat membuat algoritma untuk penyelesaian masalah kehidupan sehari-hari.

Penjelasan:

1. Tuliskan contoh kegiatan sehari-hari yang dilakukan secara berurutan.
2. Tuliskan algoritma dari langkah-langkah dalam melakukan kegiatan tersebut serta gambarkan *flowchart*-nya.
3. Peserta diminta untuk melengkapi *flowchart* dari kegiatan "Memasak nasi menggunakan *rice cooker*" di bawah ini. *Flowchart* yang telah dibuat diunggah ke LMS.

Jawaban:

| KEGIATAN: Memasak nasi menggunakan *rice cooker* |
|---|
| ALGORITMA:<br>1. Siapkan beras, panci *rice cooker*, dan air mengalir.<br>2. Masukkan beras ke dalam panci *rice cooker*.<br>3. Cuci beras dengan air mengalir.<br>4. Tambahkan air ke dalam panci *rice cooker* sesuai takaran.<br>5. Masukan panci ke dalam *rice cooker*, kemudian tutup.<br>6. Colokkan kabel *rice cooker* ke stop kontak dan tekan tombol "cook".<br>7. Tunggu hingga nasi matang. |
| *FLOWCHART*: |

**AKTIVITAS 1.2**

**Menyusun algoritma dan *flowchart* dengan logika percabangan dan perulangan**

Tujuan

1. Peserta belajar untuk mengenali logika percabangan dan perulangan.
2. Peserta belajar menyusun *flowchart* untuk permasalahan yang melibatkan logika percabangan dan perulangan

Penjelasan:

Tuliskan algoritma dan gambarkan *flowchart* dari penyelesaian permasalahan kehidupan sehari-hari yang melibatkan kondisi tertentu. Peserta diberi waktu 30 menit untuk mengerjakan

AKTIVITAS 1.2 ini. Setelah waktu pengerjaan selesai beberapa peserta mempresentasikan hasil pekerjaannya.

Jawaban:

| KEGIATAN: |
| --- |
| ALGORITMA: |
| FLOWCHART: |

**AKTIVITAS 1.3**

**Algoritma dan *flowchart* permasalahan kompleks**

Berdasarkan pembelajaran yang telah dilakukan pada modul ini, peserta diminta untuk membuat persoalan yang lebih kompleks dan solusinya. Penyelesaian persoalan harus memuat logika percabangan dan perulangan. Peserta diminta untuk membuat algoritma dan *flowchart*-nya. Hasil Jawaban pada aktivitas 1.3 ini diunggah di dalam LMS.

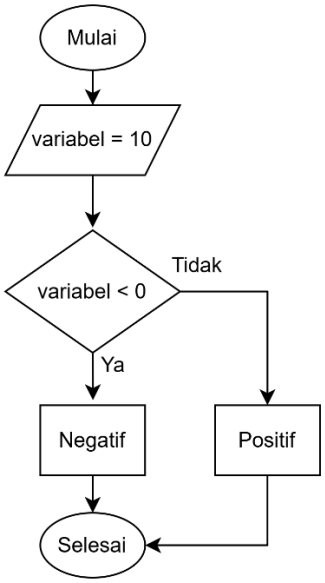
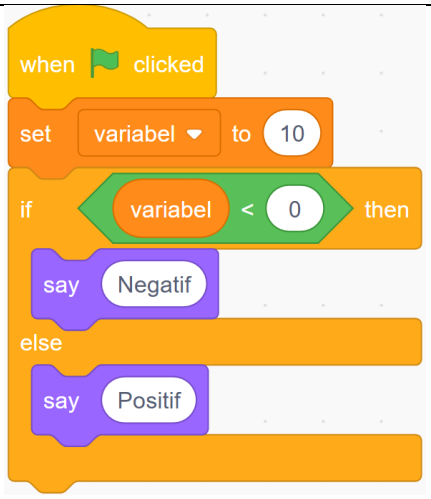
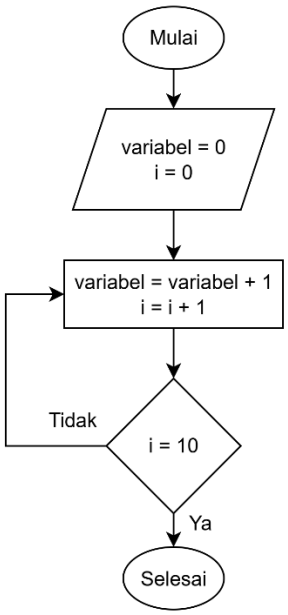
Jawaban:

| Judul: |
| --- |
| Deskripsi: |
| Algoritma: |
| Implementasi *flowchart*: |

## 4. Konversi simbol *flowchart*, blok Scratch, dan sintaks Python

Konversi dari simbol *flowchart* ke dalam kode blok Scratch dan sintaks Python ditunjukkan oleh Tabel 1.2 di bawah ini. Contoh-contoh di dalam Tabel 1.2 melibatkan percabangan dengan `if-else` dan perulangan dengan `while` dan `for`.

Tabel 1.2. Beberapa contoh konversi dari flowchart, blok scratch, dan Python.

<table>
<tr><th>Flowchart</th><th>Blok Scratch</th><th>Python</th></tr>
<tr>
<td>
</td>
<td>
</td>
<td><pre>variabel = 10
if variabel < 0:
    print(“Negatif”)
else:
    print(“Positif”)</pre></td>
</tr>
<tr>
<td rowspan="2">
</td>
<td rowspan="2">
</td>
<td><pre>variabel = 0
i = 0
while i < 10:
    variabel = variabel + 1
    i = i + 1</pre></td>
</tr>
<tr>
<td><pre>for variabel in range(1,
11):
    print(variabel)</pre></td>
</tr>
</table>

**AKTIVITAS 1.4**

**Menentukan ganjil atau genap**

Pada aktivitas ini peserta diminta untuk membuat algoritma untuk menentukan nilai *input* berupa bilangan ganjil atau genap. Peserta juga diminta untuk membuat *pseudocode*, *flowchart*, blok Scratch, dan kode algoritma tersebut dengan bahasa pemrograman Python.

Jawaban:

Algoritma

1. Membaca bilangan bulat masukan dari *user* yang kemudian disimpan ke dalam variabel bilangan.
2. Menghitung sisa bagi dari bilangan dengan 2.
3. Periksa nilai sisa dengan kondisi jika sisa sama dengan 0, maka tampilkan "genap".
4. Jika sisa tidak sama dengan 0, maka tampilkan "ganjil".
5. Kembali ke langkah pertama jika ingin memasukkan nilai baru. Jika tidak akhiri program.

*Pseudocode*

```
PROGRAM ganjil_genap
DEKLARASI:
bilangan: integer
sisa: integer
keputuran: string
ALGORITMA:
while True:
   bilangan <- input user
   sisa <- bilangan mod 2
   if sisa == 0:
        print("genap")
   else:
        print("ganjil")
   keputusan = input("apakah mau memasukkan nilai lain (y/n)?")
   if keputusan == "n"
        break
```

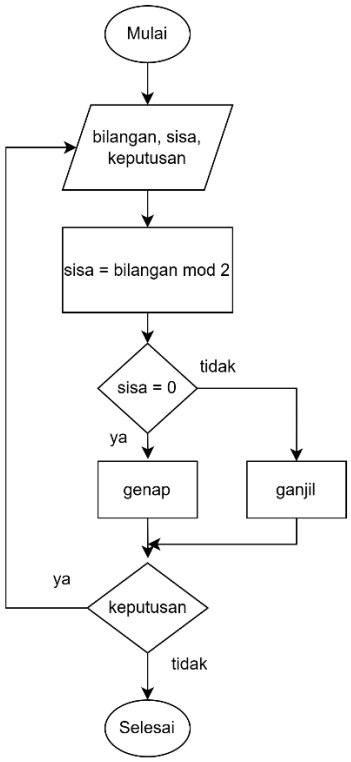
Bahasa pemrograman Python

```
while True:
    bilangan = int(input("Masukan nilai: "))
    sisa = bilangan % 2

    if sisa == 0:
        print("genap")
    else:
        print("ganjil")

    keputusan = input("apakah mau memasukan nilai lain (y/n)?")
    if keputusan == "n":
        break
```

*Flowchart*




Blok Scratch

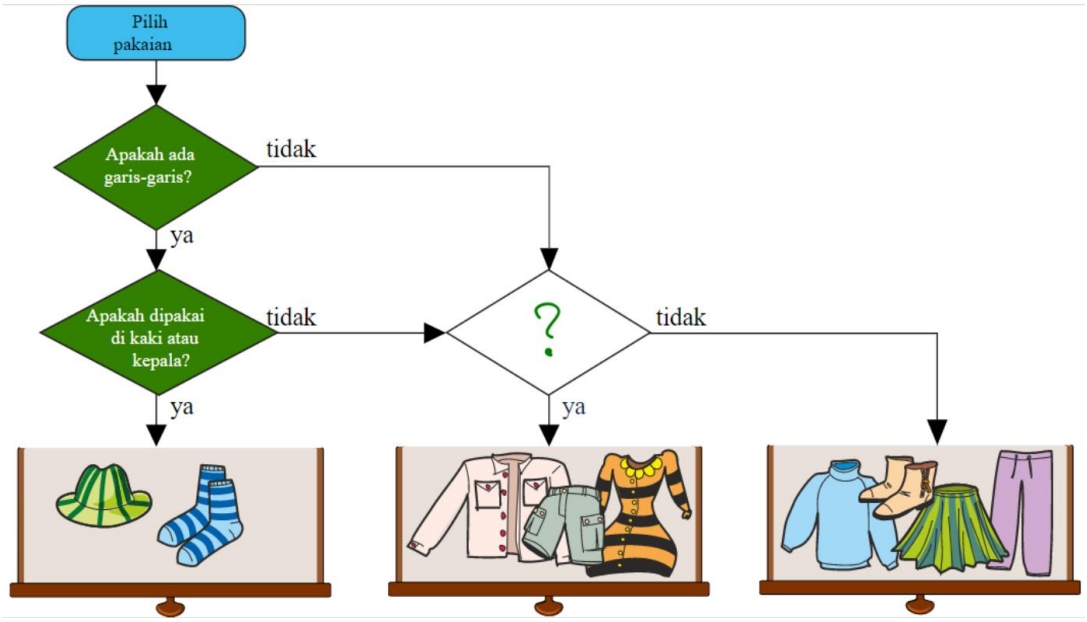


**AKTIVITAS 1.5**

**Merapikan pakaian**

Diadaptasi dari soal Bebras Penggalang dengan kode soal 2020-NZ-01

Anto menata pakaian ke dalam laci berdasarkan sebuah aturan dalam *flowchart*. Dia memulai dengan memilih pakaian, lalu Anto mengikuti anak panah pada diagram alir yang ditunjukkan oleh Gambar 1.2 di bawah ini. Ketika menemui sebuah kondisi Anto mengikuti panah berdasarkan jawaban "ya" atau "tidak". Dia melanjutkan bertanya dan mengikuti anak panah sampai pakaian tiba di laci.




Gambar 1.2. *Flowchart* mengelompokkan jenis pakaian.

Setelah semua pakaian masuk ke dalam laci-laci seperti yang ditunjukkan pada Gambar 1.2 di atas, pertanyaan apa (?) yang kurang pada diagram alir di atas?

A. Apakah lengan panjang?

B. Apakah ada kancingnya?

C. Apakah ada resletingnya?

D. Apakah ada sakunya?

Jelaskan alasan Anda memilih jawaban tersebut!

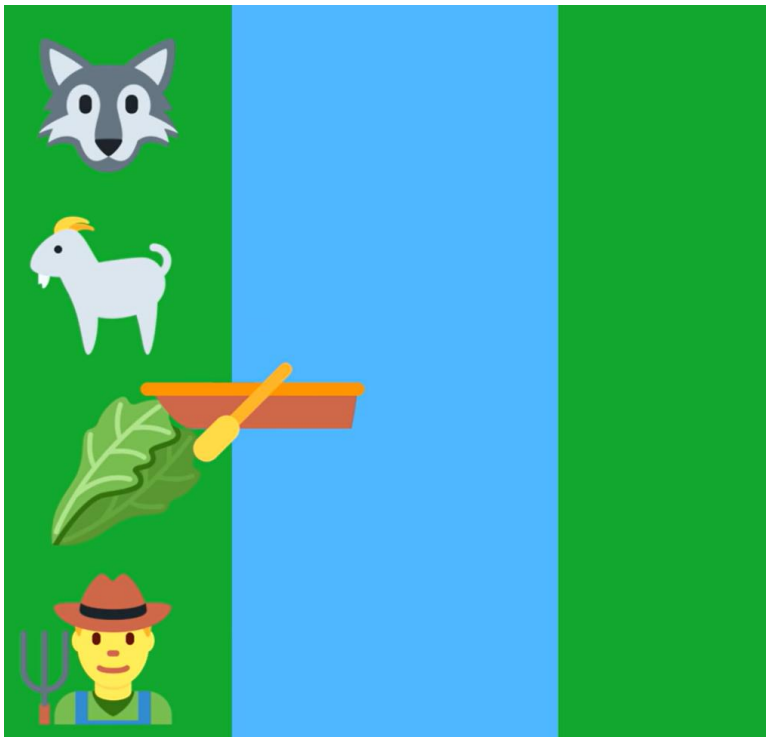
**AKTIVITAS 1.6**

**Petani, kambing, serigala, dan sayur kubis**

Seorang petani berdiri di pinggir sungai bersama serigala, kambing, dan sayur kubis (Gambar 1.3). Mereka ingin menyeberangi Sungai tersebut dengan menggunakan perahu. Sayangnya perahu hanya bisa menampung petani dan satu penumpang lagi (serigala, kambing, atau sayur kubis). Jika ditinggal hanya berdua:

- Serigala akan memakan kambing, atau
- kambing akan memakan sayur kubis.

Petani harus membantu mereka menyeberangi sungai dengan selamat tanpa ada yang dimakan.



Gambar 1.3. Petani dan teman-temannya ingin menyeberangi sungai.

Jawaban:

1. Petani menyeberangkan kambing.
2. Petani kembali.
3. Petani menyeberangkan serigala, saat sampai tukar serigala dengan kambing.
4. Petani kembali
5. Petani menukar kambing dengan sayur kubis.

6. Petani menyeberangkan sayur kubis.

7. Petani kembali.

8. Petani menyeberangkan kambing.

Apakah ada solusi lain?

**AKTIVITAS 1.7**

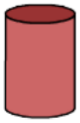
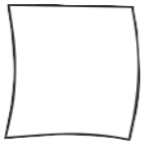
**Robot pembersih sampah**

Diadaptasi dari soal Bebras Penggalang dengan kode soal I-2019-DE-04

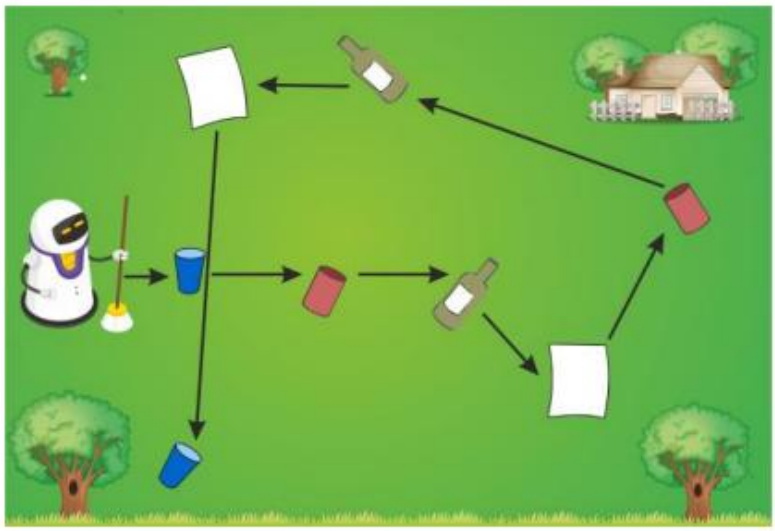


Gambar 1.4. Robot memungut sampah yang berserakan.

Setelah acara pertunjukan pentas seni, beberapa penonton meninggalkan sampah-sampah di lapangan. Untungnya ada robot pembersih sampah yang secara otomatis akan mengangkut sampah yang jaraknya terdekat terlebih dahulu. Dari gambar di bawah ini sampah terakhir yang akan dipungut oleh robot adalah

| A | B | C | D |
|---|---|---|---|
|  |  |  |  |

Jawaban:



Perilaku robot dikendalikan oleh program komputer yang terdiri dari instruksi-instruksi yang ditulis dalam bahasa yang dikenali oleh komputer. Robot mengambil data melalui kamera dan sensor dengan mengenali objek di sekitarnya dan jarak masing-masing objek. Algoritma akan menggerakkan robot untuk mengambil objek dengan jarak terdekat.

Pada aplikasi nyata, robot pengumpul sampah dapat membantu mengumpulkan benda yang berbahaya bagi manusia. Demikian juga robot otonom dapat membantu dalam banyak skenario berbahaya lainnya seperti bencana alam, radiasi zat kimia, lingkungan ekstrem, dan lain sebagainya.

**AKTIVITAS 1.8**

**Mengenali hewan**

Soal Bebras Penggalang dengan kode soal I-2019-PK-07

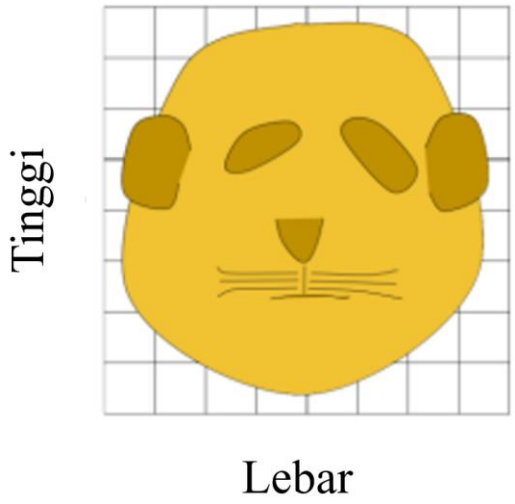
Raja bebras ingin mendapatkan informasi tentang semua hewan penghuni hutan kekuasaannya. Untuk itu, ia mengundang semua penghuni hutan untuk datang ke istana. Sang Raja membeli sebuah mesin pengenal hewan, yang bekerja berdasarkan wajah. Mesin tersebut bekerja mengenali wajah berdasarkan aturan berikut:

| Karakteristik | Kelinci | Berang-berang | Beruang | Kucing |
|---|---|---|---|---|
| Tinggi telinga | ½ tinggi kepala | ¼ tinggi kepala | ¼ tinggi kepala | ½ tinggi kepala |
| Lebar kumis* | Lebar kepala | ½ lebar kepala | ½ lebar kepala | Lebar kepala |
| Lebar kepala | ½ tinggi kepala | ½ tinggi kepala | Tinggi kepala | Tinggi kepala |

*Catatan:

1. Lebar kumis diukur dengan menambahkan bagian kiri dan bagian kanan.
2. Semua ukuran adalah lebar maksimum dan tinggi maksimum.

Hewan yang harus diidentifikasi adalah sebagai berikut:




Pilihlah salah satu dari 4 hewan yang dituliskan pada tabel di atas

a. Kelinci
b. Berang-berang
c. Beruang
d. Kucing

Jawaban:

Gambar tersebut memiliki karakteristik Beruang, karena

- Lebar kepala sama dengan tinggi kepala
- Panjang kumis adalah setengah dari lebar kepala
- Telinga adalah seperempat dari tinggi kepala

Ini salah satu contoh aplikasi dari Pembelajaran Mesin (*Machine Learning*) yang merupakan domain Kecerdasan Artifisial untuk mengenali gambar. Salah satu contoh penerapan nyata Pembelajaran Mesin adalah *Autonomous Driving*, di mana algoritma Pembelajaran Mesin dilatih untuk mengenali garis dan tanda di sepanjang jalan serta semua benda yang mungkin ditemui oleh mobil.

## 5. Refleksi

1. Dalam skala 1 sampai 5, seberapa yakin Anda memahami konsep-konsep pada modul ini? (1 = Sangat Tidak yakin, 2 = Tidak Yakin, 3 = Ragu-ragu, 4 = Yakin, 5 = Sangat yakin)

2. Jelaskan satu cara pemahaman algoritma dan logika dapat membantu Anda dalam kehidupan sehari-hari, baik untuk proses sekuensial, proses dengan logika percabangan atau proses dengan perulangan.

# C. LATIHAN

1. Algoritma paling tepat didefinisikan sebagai...

   A. Aturan sintaksis dalam bahasa pemrograman Python atau Java.

   B. Susunan huruf dan angka yang menjadi bahasa mesin.

   C. Serangkaian langkah-langkah logis, terperinci, dan terbatas untuk menyelesaikan masalah.

   D. Hasil akhir (*output*) dari sebuah proses komputasi.

2. Dalam notasi *flowchart*, simbol yang digunakan untuk menyatakan proses pengambilan keputusan atau percabangan (kondisi) adalah...

   A. Persegi (*Rectangle*)

   B. Oval/*Terminator*

   C. Jajar Genjang (*Parallelogram*)

   D. Belah Ketupat (*Diamond*)

3. Perhatikan urutan *pseudocode* berikut:

```
BACA Usia
JIKA Usia > 17 MAKA
    TULIS "Boleh membuat KTP"
SELAIN ITU MAKA
    TULIS "Belum boleh membuat KTP"
```

   *Pseudocode* di atas menggunakan struktur logika...

   A. Sekuensial

   B. Perulangan

   C. Percabangan

   D. Semua jawaban salah

4. *Pseudocode* berfungsi sebagai jembatan antara bahasa manusia dan bahasa pemrograman. Istilah "Pseudo" sendiri berarti...

   A. Nyata atau Aktual.

   B. Semu atau Tiruan.

   C. Logis atau Sistematis.

   D. Visual atau Grafis.

5. Perhatikan dua algoritma tentang pengaturan lampu lalu lintas di bawah ini

```
(1) if lampu menyala then
       if lampu merah then
          berhenti
       else
          jalan

(2) if lampu menyala then
       if lampu merah then
          berhenti
    else
       jalan
```

Pada kondisi apa kedua algoritma menghasilkan output yang berbeda? Algoritma mana yang menurut Anda lebih sesuai menggambarkan lampu lalu lintas?

# D. DAFTAR PUSTAKA


Ascher, M., 1990. A river-crossing problem in cross-cultural perspective. *Mathematics Magazine*, *63*(1), pp.26-29.

Munir, R., 2011. Algoritma dan Pemrograman dalam bahasa Pascal dan C. Bandung: Informatika.

Munir, R. dan Lidya, L. 2015. Algoritma dan Pemrograman dalam bahasa Pascal, C, dan C++, Edisi Keenam. Bandung: Informatika.

Tim Olimpiade Komputer Indonesia. 2020. Tantangan Bebras Indonesia 2020 – Kategori Penggalang. NBO Bebras Indonesia.

Tim Olimpiade Komputer Indonesia. 2019. Tantangan Bebras Indonesia 2019 – Kategori Penggalang. NBO Bebras Indonesia.

