PETA KONSEP

A Membuat Algoritma

B Menerapkan Algoritma

A Membuat Algoritma

01 Pengertian Algoritma

Algoritma didefinisikan sebagai kumpulan instruksi yang terstruktur dan dirancang untuk memecahkan masalah tertentu atau melakukan tugas tertentu.

Jenis tahapan Algoritma

a. Mulai
b. Input
c. Pengolahan
d. Output

A

Membuat Algoritma

02 Karakteristik Algoritma

a. Akurasi

b. Efisien

c. Fleksibilitas

d. Kesederhanaan

e. Keandalan

f. Kestabilan

g. Kemudahan Pemeliharaan

h. Dokumentasi

i. Keamanan

A Membuat Algoritma

03 Membuat Algoritma

Langkah langkahnya untuk membuat algoritma yang baik

1. Identifikasi Masalah
2. Analisis Masalah
3. Desain Algoritma
4. Pilih Alat dan Teknologi yang Sesuai
5. Implementasikan Algoritma
6. Uji Algoritma
7. Optimalkan Algoritma
8. Dokumentasikan Algoritma
9. Pemeliharaan dan Pembaruan Algoritma

A Membuat Algoritma

04 Algoritma Percabangan

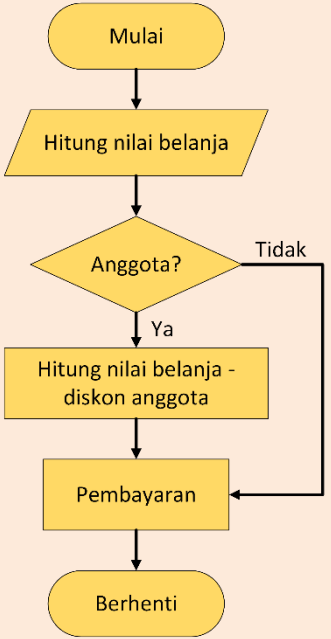
Algoritma percabangan atau algoritma pengambilan keputusan adalah algoritma yang memiliki beberapa alternatif proses yang dapat dipilih berdasarkan kondisi tertentu.

Proses mana yang akan dijalankan akan dipilih sesuai dengan kondisi yang ada, sehingga algoritma dapat membuat keputusan yang tepat dan akurat dalam menangani berbagai situasi.

a. Kondisi Satu Percabangan

Kondisi yang dijalankan jika suatu persyaratan dipenuhi.

Contoh:
Koperasi sekolah memberi diskon 5% bagi anggota, sedangkan non-anggota membayar penuh.
Algoritma pembayaran ini ditunjukkan pada gambar.

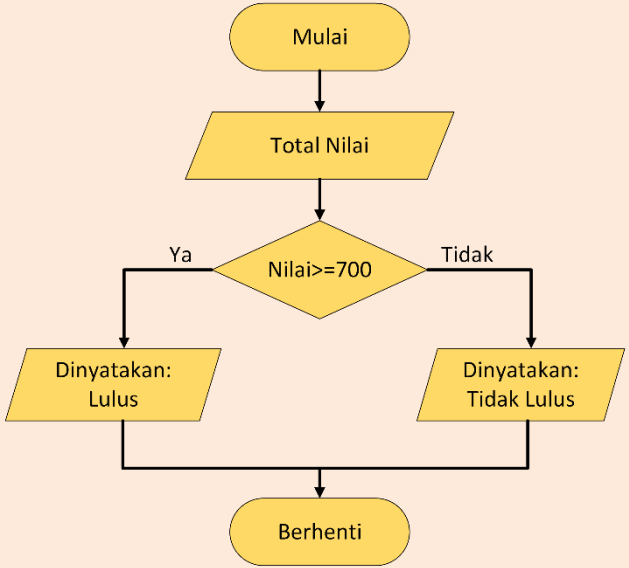


04 Algoritma Percabangan

b. Kondisi Dua Percabangan

ketika algoritma harus memilih salah satu dari dua pilihan berbeda berdasarkan kondisi tertentu.

Contoh, dalam seleksi peserta didik baru, jika nilai ≥ 700, maka diterima. Jika nilai < 700, maka tidak diterima.

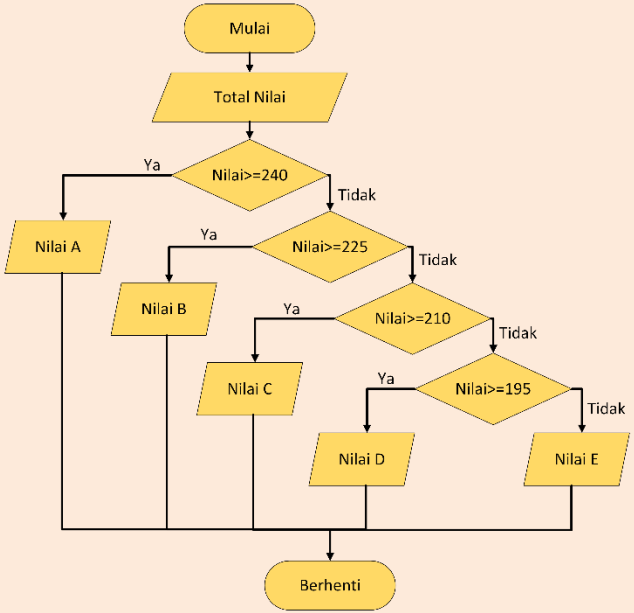


A Membuat Algoritma

04 Algoritma Percabangan

c. Kondisi Tiga atau Lebih Percabangan

Algoritma dengan tiga atau lebih percabangan dapat digunakan, misalnya saat guru memberi nilai huruf A–E berdasarkan total nilai siswa. Setiap rentang nilai memiliki cabang tersendiri, dan flowchart-nya akan menampilkan lima percabangan sesuai kondisi input.



B Menerapkan Algoritma

01 Menggunakan Phython

Python adalah bahasa pemrograman dengan sintaks yang sederhana, sehingga mudah dipelajari. Meskipun sederhana, Python dapat digunakan untuk berbagai tujuan, seperti pengembangan web, artificial intelligence, machine learning, dan analisis data. Selain itu, Python dapat berjalan di berbagai platform, seperti Windows, Mac, Linux, dan Raspberry Pi.

Python bisa dijalankan di berbagai aplikasi seperti PyCharm, VS Code, IDLE, Jupyter Notebook, Anaconda dan Google Colaboratory yang dapat diakses online di colab.research.google.com.

B Menerapkan Algoritma

01 Menggunakan Phython

a. Aturan Dasar Sintaks Python

(1) Sintaks Python

Dapat dijalankan dengan cara menuliskan langsung di baris perintah.

```
1  print("Hello dunia, saya seorang programmer")
```

Perintah di atas dapat dijalankan secara langsung dan jika dijalankan akan menghasilkan keluaran sebagai berikut.

```
Hello dunia, saya seorang programmer
```

B Menerapkan Algoritma

01 Menggunakan Phython

a. Aturan Dasar Sintaks Python

(2) Indentasi

Indentasi adalah spasi di awal baris kode untuk menandai blok perintah. Di Python, indentasi wajib agar program bisa berjalan dengan benar. Berikut contohnya.

```
1  nilai_akhir = int(input("Masukkan total nilai? "))
2  try:
3      if nilai_akhir >= 700:
4          print("Lulus")
5      else:
6          print("Tidak Lulus")
7  except ValueError:
8        print("Input tidak valid. Mohon masukkan
9  nilai dalam bentuk angka.")
```

B Menerapkan Algoritma

01 Menggunakan Phython

a. Aturan Dasar Sintaks Python

(3) Komentar

Dalam pemrograman, komentar digunakan untuk memberi catatan atau penjelasan pada kode. Ditulis dengan tanda pagar (#) dan tidak akan dijalankan oleh Python.

Komentar beberapa baris ditulis dengan menambahkan tanda pagar (#) di awal setiap baris.

```
1  # Ini Adalah komentar
2  # yang terdiri dari beberapa baris
3  print("Hello dunia, saya seorang programmer")
```

Cara lain adalah dengan menggunakan tiga tanda kutip ganda di awal dan akhir komentar.

B

Menerapkan Algoritma

02 Menerima Input

Program sering membutuhkan masukan dari pengguna. Di Python, masukan diterima dengan fungsi `input()` dan disimpan dalam variabel.

```
1  nama = input("Siapa nama Anda? ")
2  tgllahir = input("Masukkan tgl lahir Anda? ")
3  sekolah = input("Masukkan sekolah Anda? ")
4  print(f"Selamat siang, {nama}! Kamu lahir tanggal
   {tgllahir} dan bersekolah di {sekolah}.")
```

Output :

```
Siapa nama Anda? Gianna
Masukkan tgl lahir Anda? 19 Desember 2010
Masukkan sekolah Anda? SMP Erlangga Jakarta
Selamat siang, Gianna! Kamu lahir tanggal 19 Desember 2010 dan bersekolah di SMP Erlangga Jakarta.
```

B

Menerapkan Algoritma

03 Menggunakan Variabel

Variabel adalah nama untuk menyimpan data (angka, teks, atau objek) yang nilainya bisa berubah saat program dijalankan. Di Python, variabel tidak perlu dideklarasikan terlebih dahulu, cukup langsung dituliskan saat diberi nilai.
Contoh penggunaan variabel x dan y akan otomatis dikenali saat diberi nilai dalam kode program.

```
1  x = 5
2  y = 2
```

B

Menerapkan Algoritma

03 Menggunakan Variabel

Operator aritmatika Python (x = 5 dan y = 2).

No.	Operasi	Operator	Contoh	Hasil
1	Penjumlahan	+	x + y	7
2	Pengurangan	-	x - y	3
3	Perkalian	*	x * y	10
4	Pembagian	/	x / y	2,5
5	Modulus	%	x % y	1
6	Eksponensial	**	x ** y	25
7	Pembagian bulat	//	x // y	2

B

Menerapkan Algoritma

03 Menggunakan Variabel

Operasi modulus (%) menghitung sisa hasil pembagian, sedangkan pembagian bulat (//) membagi lalu membulatkan hasilnya ke bawah. Berikut contoh sintaks variabel dan operasi.

```
1 x = 5
2 y = 2
3 print(x + y)
4 print(x - y)
5 print(x * y)
6 print(x / y)
7 print(x % y)
8 print(x ** y)
9 print(x // y)
```

```
7
3
10
2.5
1
25
2
```

B

Menerapkan Algoritma

03 Menggunakan Variabel

Nama variabel di Python bisa berupa huruf, kata, atau gabungan huruf dan angka, tetapi harus mengikuti aturan berikut:

a. Dimulai dengan huruf atau *underscore* (_).
b. Tidak boleh diawali angka.
c. Hanya boleh berisi huruf, angka, dan underscore (A–z, 0–9, _).
d. Bersifat case-sensitive (Nama ≠ NAMA ≠ nama).
e. Tidak boleh menggunakan kata kunci Python.
f. Tidak boleh ada spasi; gunakan gabungan huruf atau underscore, misalnya: NamaVariabel, namaVariabel, atau Nama_Variabel.

B

Menerapkan Algoritma

04 Tipe Data

Setiap variabel menyimpan nilai dengan tipe data tertentu, seperti teks, angka, tanggal, atau nilai benar/salah.

No.	Tipe Data	Nama Tipe Data
1	Teks	str
2	Numerik	int, float, complex
3	Urutan	list, tuple, range
4	Pemetaan	dict
5	Kumpulan	set, frozenset
6	Boolean	bool
7	Biner	bytes, bytearray, memoryview
8	Tidak ada tipe data	NoneType

B

Menerapkan Algoritma

05 Operator Python

a. Operator Penugasan

Digunakan untuk memberikan nilai-nilai kepada variabel-variabel.

No.	Operator	Contoh	Operasi Sebenarnya
1	=	x = 5	x = 5
2	+=	x += 3	x = x + 3
3	-=	x -= 3	x = x - 3
4	*=	x *= 3	x = x * 3
5	/=	x /= 3	x = x / 3
6	%=	x %= 3	x = x % 3
7	//=	x //= 3	x = x // 3
8	**=	x **= 3	x = x ** 3

b. Operator Perbandingan

Digunakan untuk membandingkan dua nilai, dengan hasil berupa True atau False.

No.	Nama	Operator	Contoh
1	Sama dengan	==	`x == y`
2	Tidak sama dengan	!=	`x != y`
3	Lebih besar dari	>	`x > y`
4	Lebih kecil dari	<	`x < y`
5	Lebih besar dari atau sama dengan	>=	`x >= y`
6	Lebih kecil dari atau sama dengan	<=	`x <= y`

c. Operator Logika

Digunakan untuk menggabungkan dua atau lebih kondisi yang ingin diuji.

Operator	Keterangan	Contoh
and	Bernilai true jika kedua statemen bernilai true	`x < 5 and x < 10`
or	Bernilai true jika salah satu statemen bernilai true	`x < 5 or x < 4`
not	Mengembalikan nilai kebalikan dari statemen yang ada	`not (x < 5 and x < 10)`

B Menerapkan Algoritma

06 Menerapkan Kondisi Percabangan

Contoh program Kalkulator BMI untuk menghitung nilai BMI dan menentukan kategori dari nilai BMI dapat dilakukan dengan langkah langkah sebagai berikut.

a. Identifikasi masalah

- Bagaimana menghitung nilai BMI dan menentukan kategori nilai.
- Input apa saja yang dibutuhkan.
- Output apa saja yang diinginkan.

b. Analisis Kebutuhan

Untuk menghitung nilai BMI, program membutuhkan input tinggi badan dan berat badan, lalu memprosesnya dengan rumus BMI sehingga menghasilkan output berupa nilai BMI dan kategorinya.

B Menerapkan Algoritma

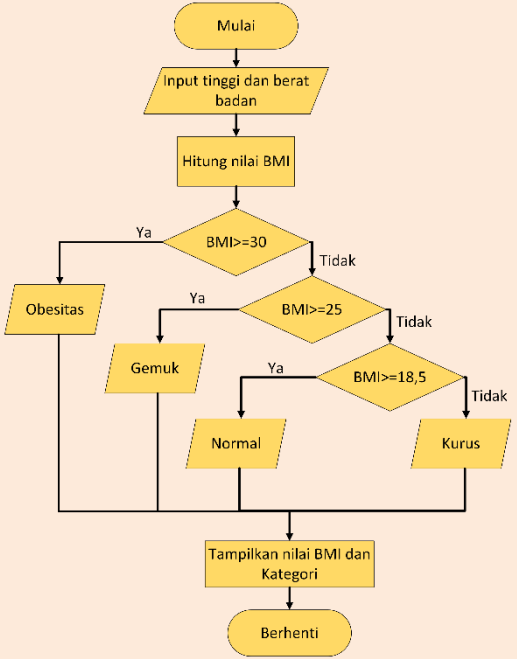
06 Menerapkan Kondisi Percabangan

c. Mendesain algoritma

Algoritma kalkulator BMI dimulai dari input tinggi dan berat badan, menghitung nilai BMI, menentukan kategori, lalu menampilkan hasilnya. Berikut dalam bentuk flowchart.

d. Pilih teknologi

Algoritma di atas akan diimplementasikan menggunakan bahasa pemrograman Python.



e. Implementasi

Kode program untuk implementasi akan seperti sintak berikut.

```
1   print("Selamat Datang di Kalkulator BMI")
2   tinggi = float(input("Masukkan tinggi badan (cm): "))
3   berat = float(input("Masukkan berat badan (kg): "))
4   tinggi = tinggi * 0.01

5   bmi = berat / (tinggi ** 2)

6   OBESITAS = 30
7   GEMUK = 25
8   NORMAL = 18.5

9   if bmi >= OBESITAS:
10   kategori = "Obesitas"
11  elif bmi >= GEMUK:
12   kategori = "Gemuk"
13  elif bmi >= NORMAL:
14   kategori = "Normal"
15  else:
16   kategori = "Kurus"
17  print(f"Nilai BMI anda adalah: {bmi:.2f}")
18  print(f"Anda termasuk: {kategori}")
```

f. Uji Coba dan Evaluasi

Pengujian dan evaluasi algoritma dilakukan dengan menjalankan program menggunakan berbagai input untuk melihat hasil yang ditampilkan.

g. Dokumentasi

Pada proses ini dibuat dokumentasi untuk algoritma yang sudah diimplementasikan.

```
Selamat Datang di Kalkulator BMI
Masukkan tinggi badan (cm): 169
Masukkan berat badan (kg): 70
Nilai BMI anda adalah: 24.51
Anda termasuk: Normal
```