

PETA KONSEP

A

Translasi Pemrograman Visual

B

Mengenal dan Menggunakan Pemrograman Berbasis Teks

A

Translasi Pemrograman Visual

01

Menggunakan EduBlock

Langkah-langkah membangun program untuk menghitung dan menggambar garis lurus.

1. Masuk halaman EduBlock dengan menggunakan alamat URL app.edublocks.org.
2. Buat akun baru di EduBlock.
3. Buatlah proyek baru, beri nama proyek tersebut, misalnya Garis Lurus.

4. Tambahkan blok kode berikut.


```
# Start code here
import math
from turtle import *
turtle = Turtle()
```

A Translasi Pemrograman Visual

01 Menggunakan EduBlock

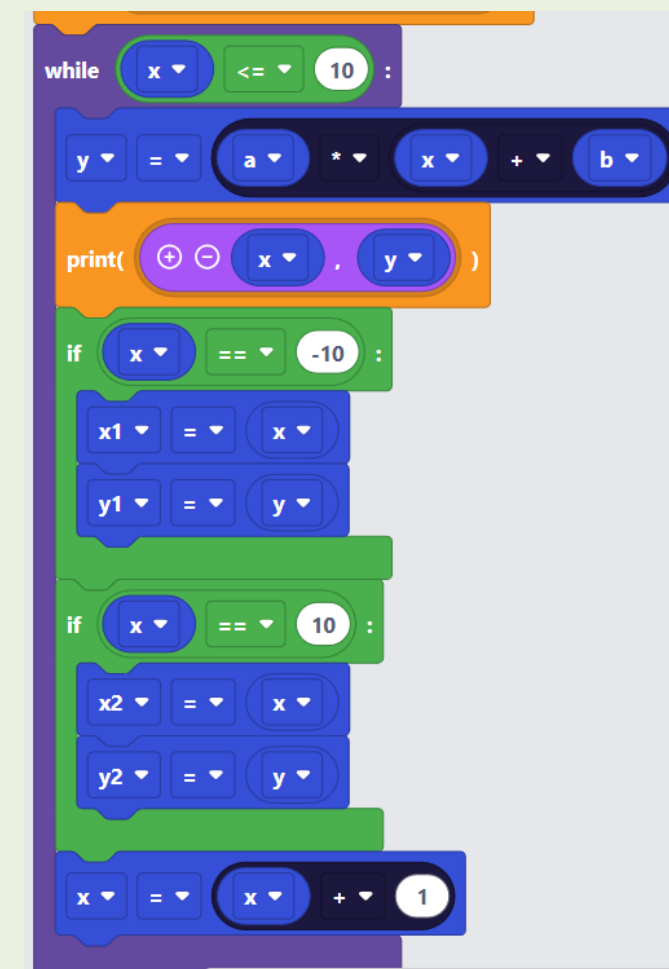
Langkah-langkah membangun program untuk menghitung dan menggambar garis lurus.

- 5 Tambahkan blok kode input, pengatur nilai, dan penampil pesan



```
a = int( input( "Masukkan nilai a " ) )
b = int( input( "Masukkan nilai b " ) )
x = -10
print( "Titik (x, y) yang dilewati adalah: " )
```

- 6 Masukkan blok kode seperti berikut.



```
while x <= 10 :
  y = a * x + b
  print( x , y )
  if x == -10 :
    x1 = x
    y1 = y
  if x == 10 :
    x2 = x
    y2 = y
  x = x + 1
```

A


Translasi Pemrograman Visual

01

Menggunakan EduBlock

Langkah-langkah membangun program untuk menghitung dan menggambar garis lurus.

7 Tambahkan blok kode berikut.



```
turtle .pen up ()
turtle .goto( -180 , 0 )
turtle .pen down ()
turtle . forward ( 360 )
turtle .pen up ()
turtle .goto( 0 , -180 )
turtle . left ( 90 )
turtle .pen down ()
turtle . forward ( 360 )
turtle .pen up ()
```

8 Tambahkan blok kode terakhir.



```
turtle .goto( 10*x1 , 10*y1 )
turtle .pencolor( + "red" )
turtle .pen down ()
turtle .goto( 10*x2 , 10*y2 )
turtle .pen up ()
```

A Translasi Pemrograman Visual

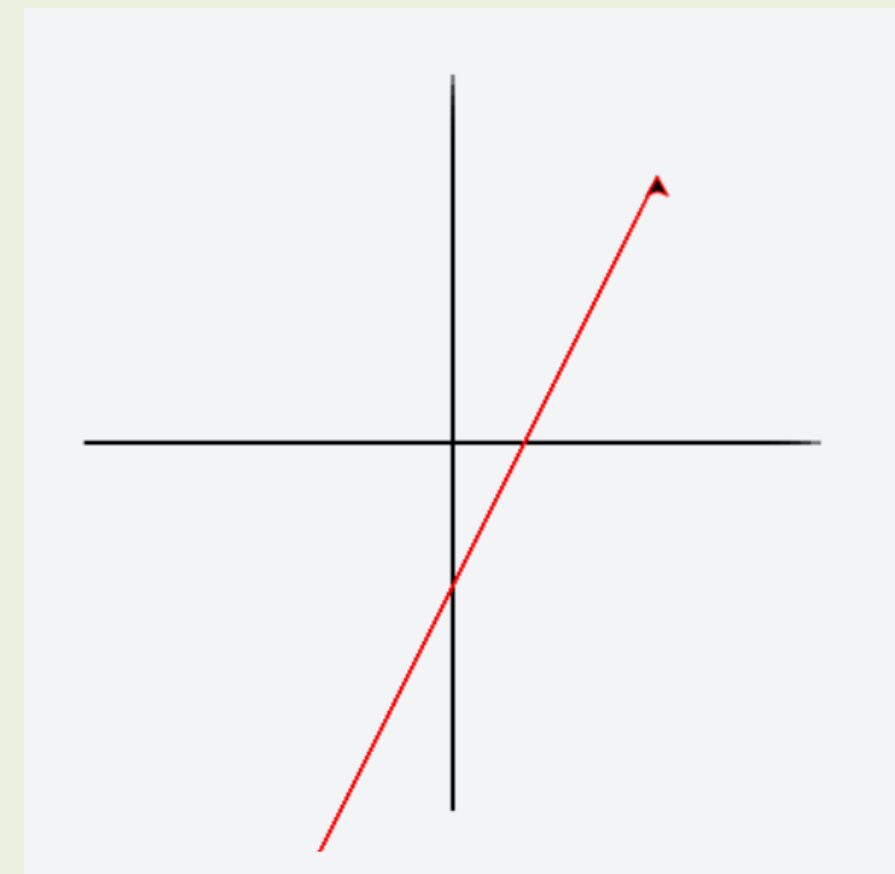
01 Menggunakan EduBlock

Langkah-langkah membangun program untuk menghitung dan menggambar garis lurus.

● Output teks program

```
Masukkan nilai a 2
Masukkan nilai b -7
Titik (x, y) yang dilewati adalah:
-10 -27
-9 -25
-8 -23
-7 -21
-6 -19
-5 -17
-4 -15
-3 -13
-2 -11
-1 -9
0 -7
1 -5
2 -3
3 -1
4 1
5 3
6 5
7 7
8 9
9 11
10 13
```

● Output visual program

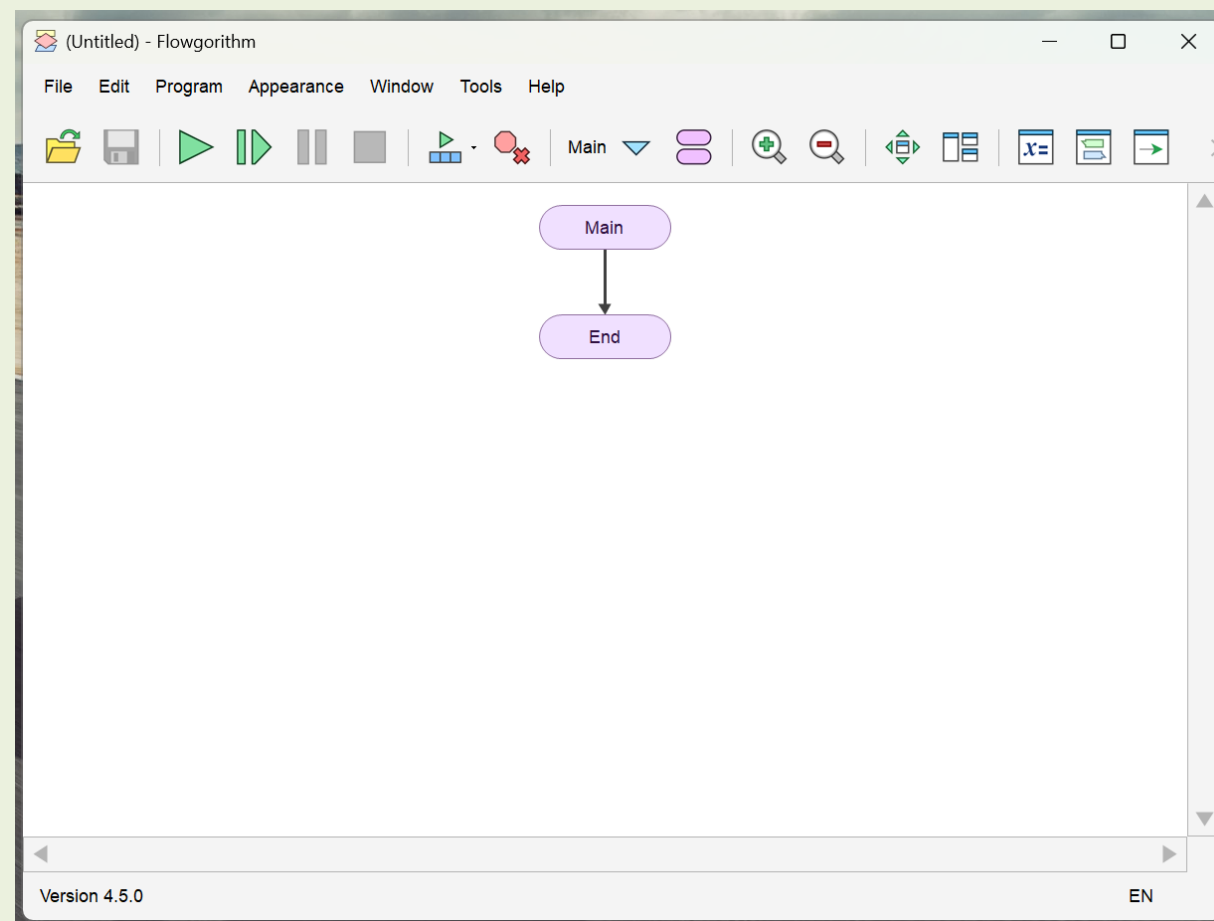


A

Translasi Pemrograman Visual

02

Menggunakan Flowgorithm



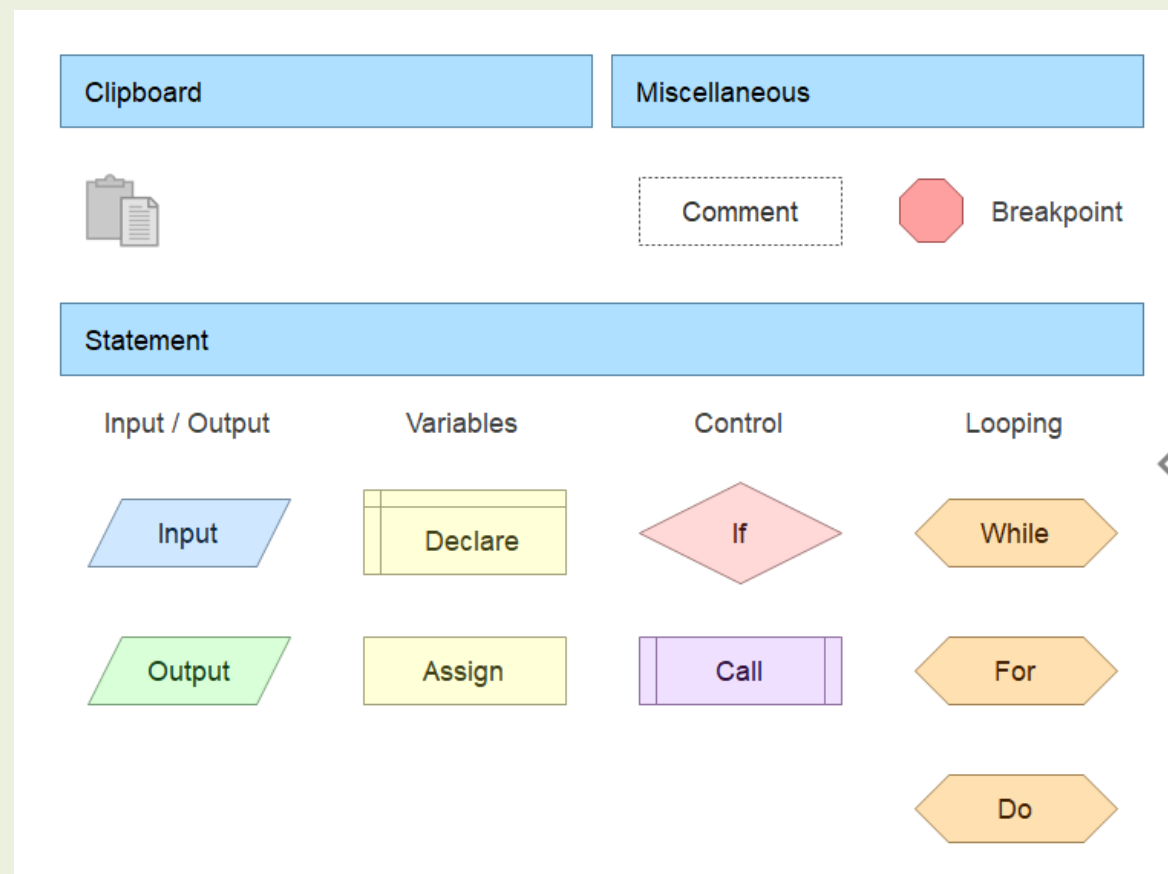
Flowgorithm adalah Bahasa pemrograman visual yang merepresentasikan logika program menggunakan simbol-simbol *flowchart*.

A

Translasi Pemrograman Visual

02

Menggunakan Flowgorithm



Simbol dasar dan fungsinya

Input: Memasukkan masukan

Output: Menampilkan keluaran

Declare: Deklarasi variable yang digunakan

Assign: Memberikan nilai ke variable tertentu

If: Membuat percabangan

While, For, dan Do: Membuat perulangan

Call: Memanggil subprogram lain

A

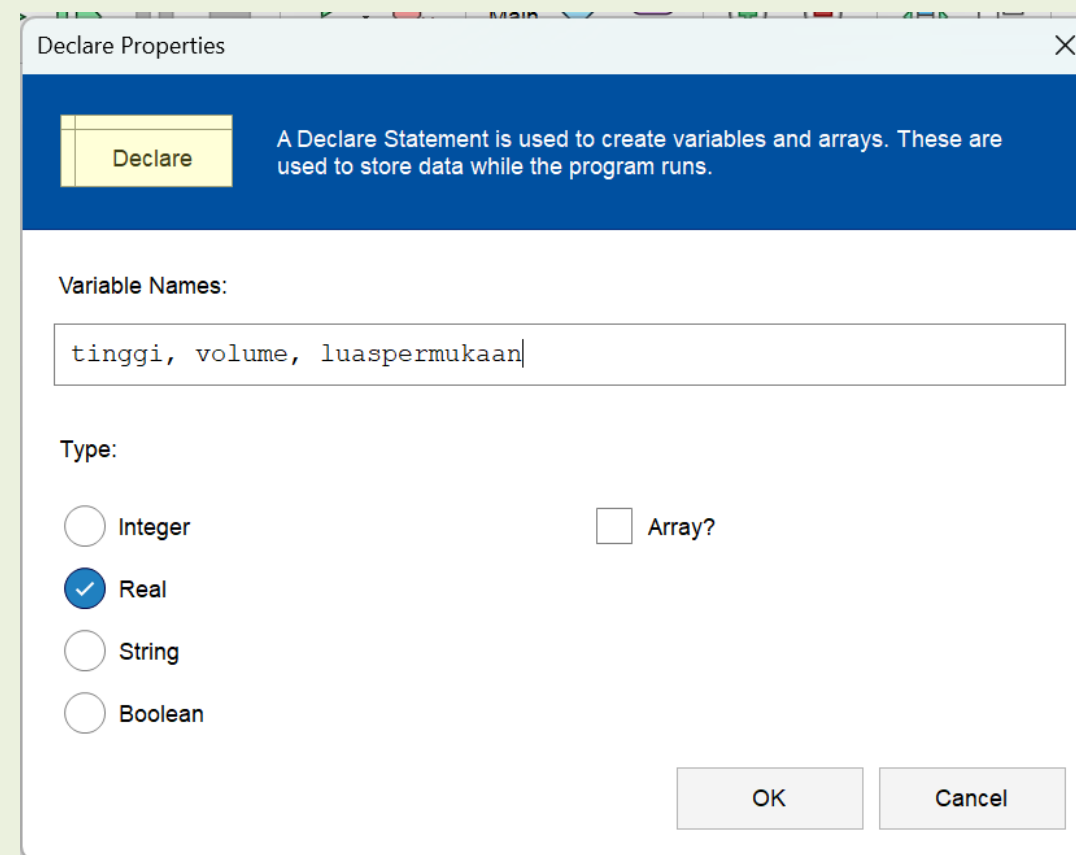
Translasi Pemrograman Visual

02

Menggunakan Flowgorithm

Langkah-Langkah membuat program yang menghitung volume dan luas tabung di Flowgorithm

- Buat proyek baru di Flowgorithm
- Klik panah, lalu pilih simbol Declare yang muncul.
- Klik simbol Declare, masukkan nama variabel, pilih tipe data, klik "OK"



A

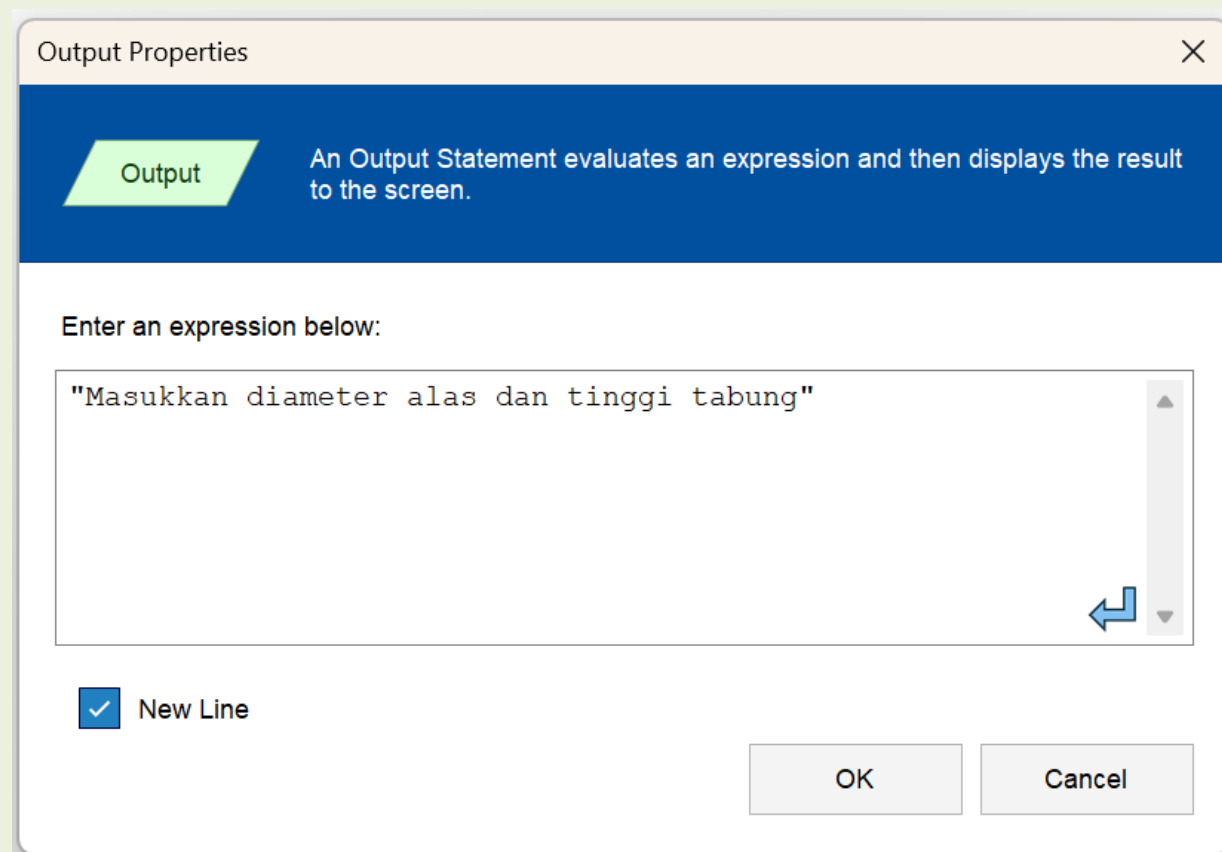
Translasi Pemrograman Visual

02

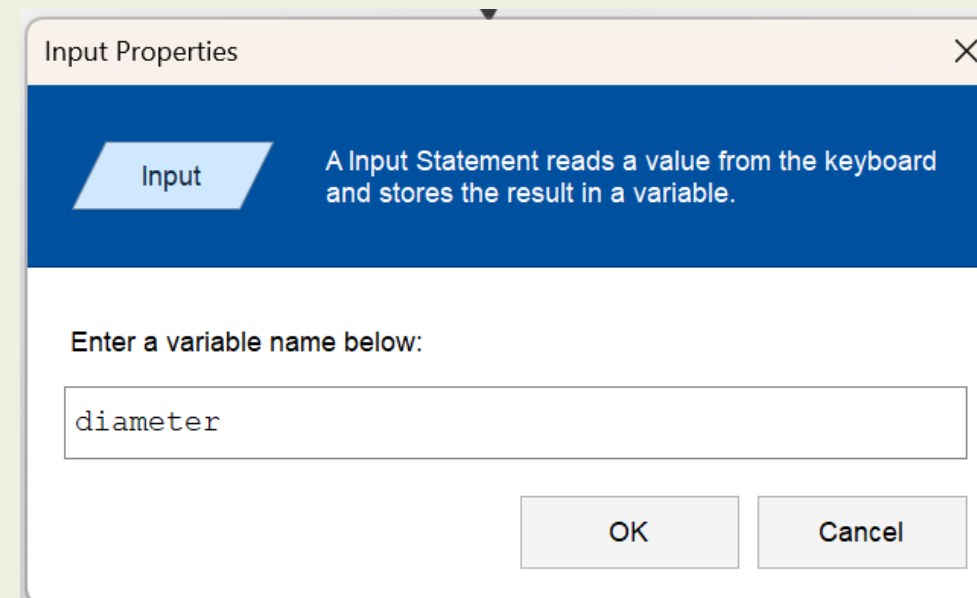
Menggunakan Flowgorithm

Langkah-Langkah membuat program yang menghitung volume dan luas tabung di Flowgorithm

- Klik tanda panah di bawah Declare, tambahkan Output dan masukkan pesan yang ingin ditampilkan.



- Klik tanda panah di bawah Output dan tambahkan Input. Masukkan variabel diameter



- Dengan cara yang sama, tambahkan simbol input untuk variabel tinggi.

A

Translasi Pemrograman Visual

02

Menggunakan Flowgorithm

Langkah-Langkah membuat program yang menghitung volume dan luas tabung di Flowgorithm

- Tambahkan simbol Assign dan masukkan rumus volume.
- Tambahkan simbol Assign untuk rumus luas permukaan.

Assignment Properties

Assign An Assignment Statement calculates an expression and then stores the result in a variable.

Variable: volume = Expression: 3.14*(diameter/2)^2*tinggi|

OK Cancel

Assignment Properties

Assign An Assignment Statement calculates an expression and then stores the result in a variable.

Variable: luaspermukaan = Expression: (2*3.14*(diameter/2)^2)+(3.14*diameter*tinggi|)

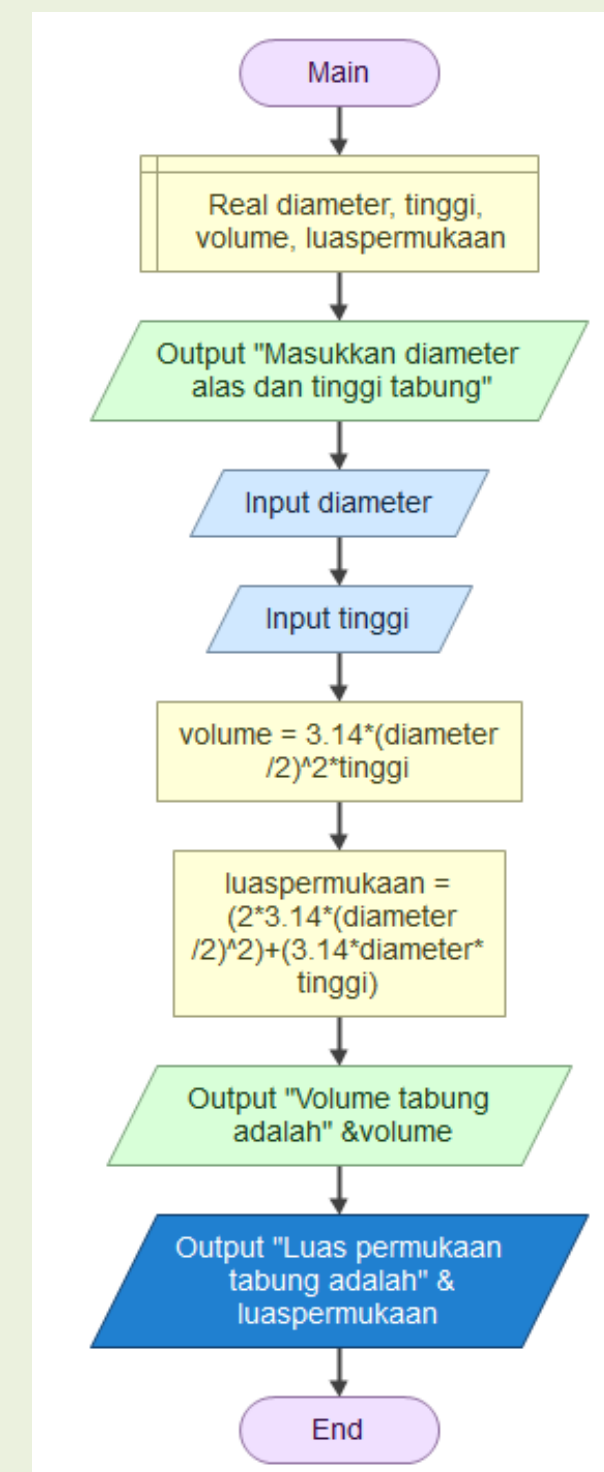
OK Cancel

A Translasi Pemrograman Visual

02 Menggunakan Flowgorithm

- Kode program keseluruhan
- Tampilan program ketika dijalankan.

```
Masukkan diameter alas dan tinggi tabung  
20  
25  
Volume tabung adalah7850  
Luas permukaan tabung adalah2198
```



A

Translasi Pemrograman Visual

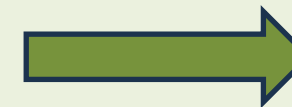
03

Translasi Konsep antar Visual Programming

a. Translasi baris demi baris

- Buat variabel yang diperlukan (X1, X2, Y1, Y2, jarakX, jarakY, dan jarak)
- Buat blok input x1, y1, x2, dan y2.

```
# Start code here
X1 = int( input( "Masukkan nilai koordinat X titik 1 (X1) " ) )
Y1 = int( input( "Masukkan nilai koordinat Y titik 1 (Y1) " ) )
X2 = int( input( "Masukkan nilai koordinat X titik 2 (X2) " ) )
Y2 = int( input( "Masukkan nilai koordinat Y titik 2 (Y2) " ) )
```



```
jarak = 0
jarak
jarakX
jarakY
X1
X2
Y1
Y2
```

A

Translasi Pemrograman Visual

03

Translasi Konsep antar Visual Programming

a. Translasi baris demi baris

- Tambahkan blok perhitungan dan impor fungsi Math.



```
import math
jarakX = X2 + X1
jarakY = Y2 + Y1
jarak = math.sqrt(jarakX * jarakX + jarakY * jarakY)
```

- Tambahkan blok Output hasil penghitungan.



```
print("Jarak kedua titik adalah:")
print(jarak)
```

A

Translasi Pemrograman Visual

03

Translasi Konsep antar Visual Programming

a. Translasi baris demi baris

● Kode program keseluruhan

```
# Start code here
import math
X1 = int( input( "Masukkan nilai koordinat X titik 1 (X1) " ) )
Y1 = int( input( "Masukkan nilai koordinat Y titik 1 (Y1) " ) )
X2 = int( input( "Masukkan nilai koordinat X titik 2 (X2) " ) )
Y2 = int( input( "Masukkan nilai koordinat Y titik 2 (Y2) " ) )
jarakX = X2 + X1
jarakY = Y2 + Y1
jarak = math.sqrt( jarakX * jarakX + jarakY * jarakY )
print( "Jarak kedua titik adalah:" )
print( jarak )
```

● Tampilan program ketika dijalankan

Text Output

```
Masukkan nilai koordinat X titik
1 (X1)5
Masukkan nilai koordinat Y titik
1 (Y1)9
Masukkan nilai koordinat X titik
2 (X2)1
Masukkan nilai koordinat Y titik
2 (Y2)6
Jarak kedua titik adalah:
5.0
```

A

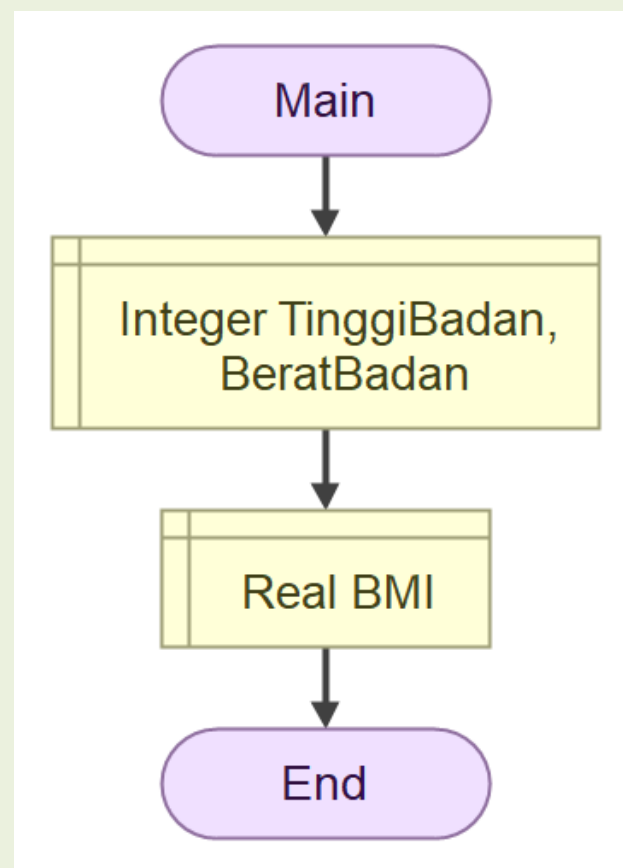
Translasi Pemrograman Visual

03

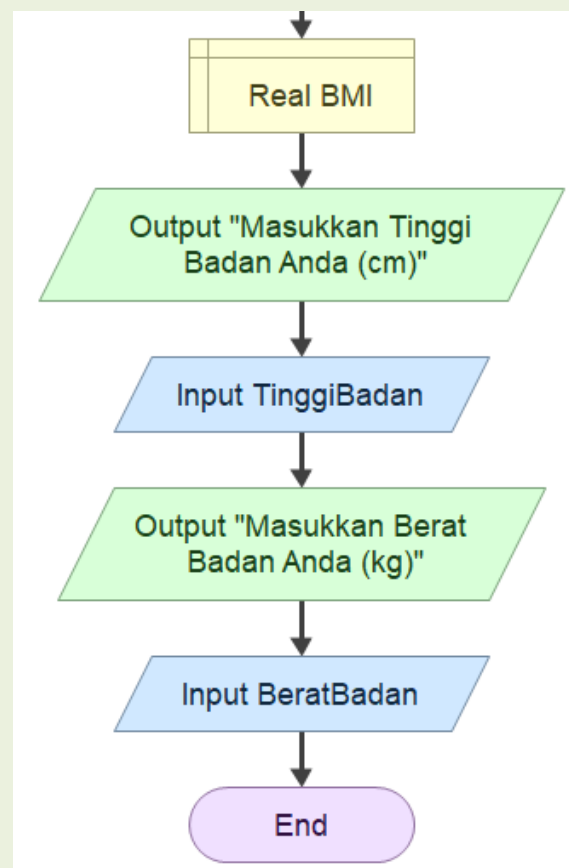
Translasi Konsep antar Visual Programming

b. Mentranslasikan subprogram

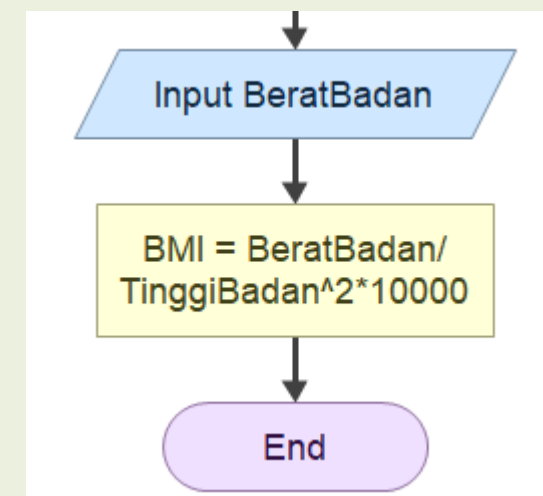
● Subprogram Mulai



● Subprogram Input Data



● Subprogram Utama



A

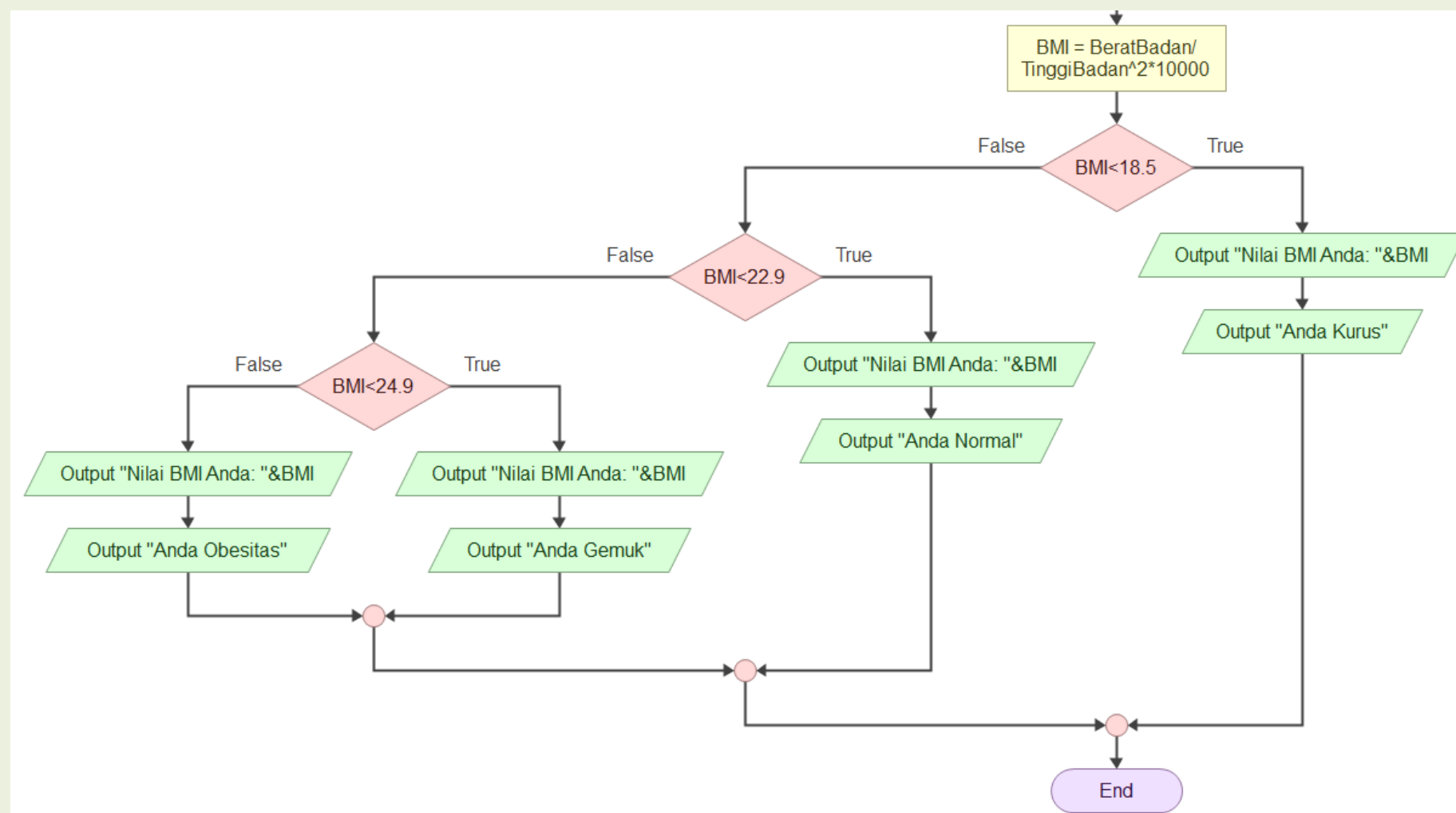
Translasi Pemrograman Visual

03

Translasi Konsep antar Visual Programming

b. Mentranslasikan subprogram

- Subprogram Tampilkan Hasil



B

Mengenal dan Menggunakan Pemrograman Berbasis Teks

01

Mengenal Python

a. Aturan dasar sintaks Python

- (1) Menuliskan sintaks langsung di baris perintah.
- (2) Indentasi (ruang kosong untuk mengatur blok perintah)
- (3) Komentar (#)

b. Menerima Input

Menggunakan fungsi **input()**

Contoh:

```
Nama = input("Siapa nama anda? ")
```

B

Mengenal dan Menggunakan Pemrograman Berbasis Teks

02

Menggunakan Variabel

Aturan pemberian nama variabel

- (1) Nama variabel harus dimulai dengan huruf atau karakter underscore (_).
- (2) Nama variabel tidak boleh diawali dengan angka.
- (3) Nama variabel hanya dapat terdiri dari huruf, angka, dan underscore (A-z, 0-9, dan _).
- (4) Nama variabel bersifat sensitif terhadap huruf besar dan kecil (case sensitif).
- (5) Nama variabel tidak boleh menggunakan kata-kata kunci yang sudah digunakan oleh Python.
- (6) Nama variabel tidak boleh mengandung spasi. Dengan kata lain, tidak boleh terdiri dari dua kata.

B**Mengenal dan Menggunakan Pemrograman Berbasis Teks****02****Menggunakan Variabel****a. Tipe data**

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**Mengenal dan Menggunakan Pemrograman Berbasis Teks****03****Operator Python****a. Operator penugasan**

No	Operator	Contoh	Operator 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

No	Nama	Operator	Contoh
1.	Sama dengan	<code>==</code>	<code>x = 5</code>
2.	Tidak sama dengan	<code>!=</code>	<code>x = x + 3</code>
3.	Lebih besar dari	<code>></code>	<code>x = x - 3</code>
4.	Lebih kecil dari	<code><</code>	<code>x = x * 3</code>
5.	Lebih besar dari atau sama dengan	<code>>=</code>	<code>x = x / 3</code>
6.	Lebih kecil dari atau sama dengan	<code><=</code>	<code>x = x % 3</code>

c. Operator logika

Operator	Keterangan	Contoh
<code>and</code>	Bernilai True jika kedua statement bernilai True	<code>x < 5 and x < 10</code>
<code>or</code>	Bernilai True jika salah satu statement bernilai True	<code>x < 5 or x < 4</code>
<code>not</code>	Mengembalikan nilai kebalikan dari statement yang ada	<code>(x < 5 and x < 10)</code>

B

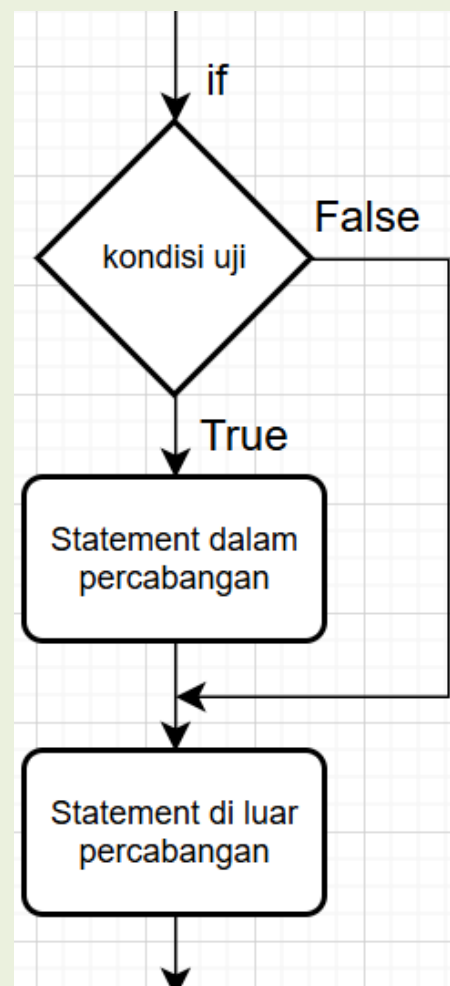
Mengenal dan Menggunakan Pemrograman Berbasis Teks

04

Kondisi Percabangan

a. Kondisi satu percabangan

Flowchart



Sintaks

```
KodePass = int(input("Masukkan Kode Pass! "))  
if KodePass == 123:  
    print("Kode Pass Anda benar")  
print("Ini adalah akhir program")
```

B

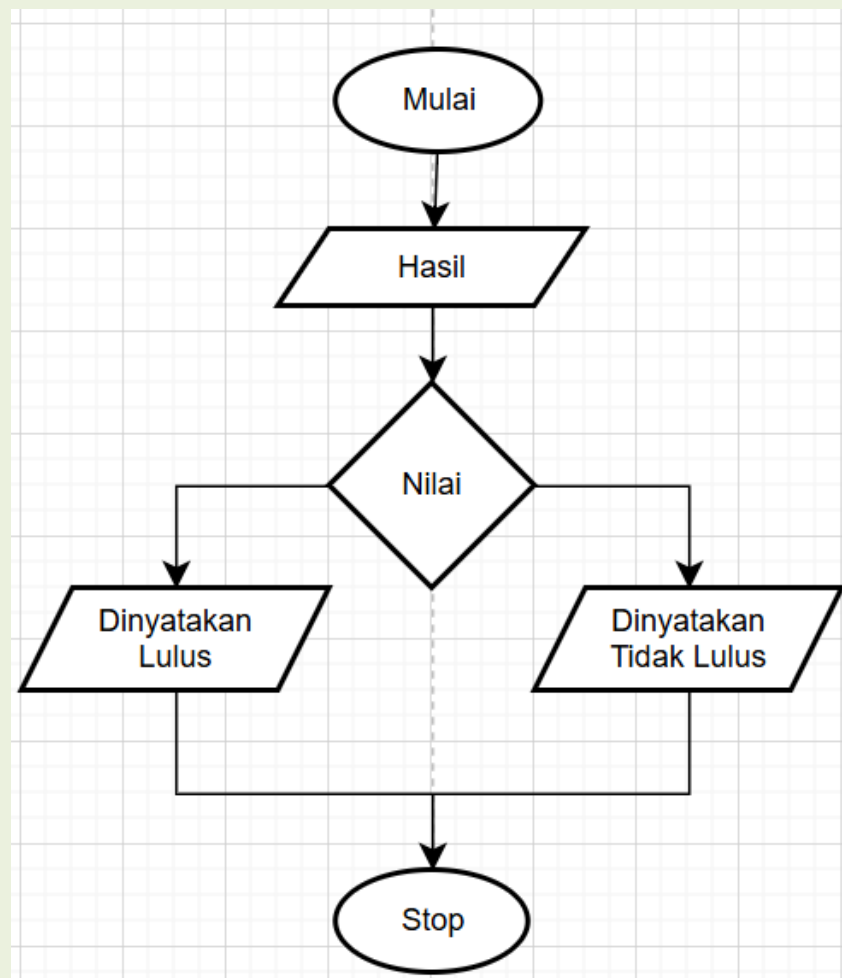
Mengenal dan Menggunakan Pemrograman Berbasis Teks

04

Kondisi Percabangan

b. Kondisi dua percabangan

Flowchart



Sintaks

```
totalNilai = int(input("Masukkan total nilai!"))
if totalNilai >= 700:
    print("Lulus")
else:
    print("Tidak Lulus")
```

B

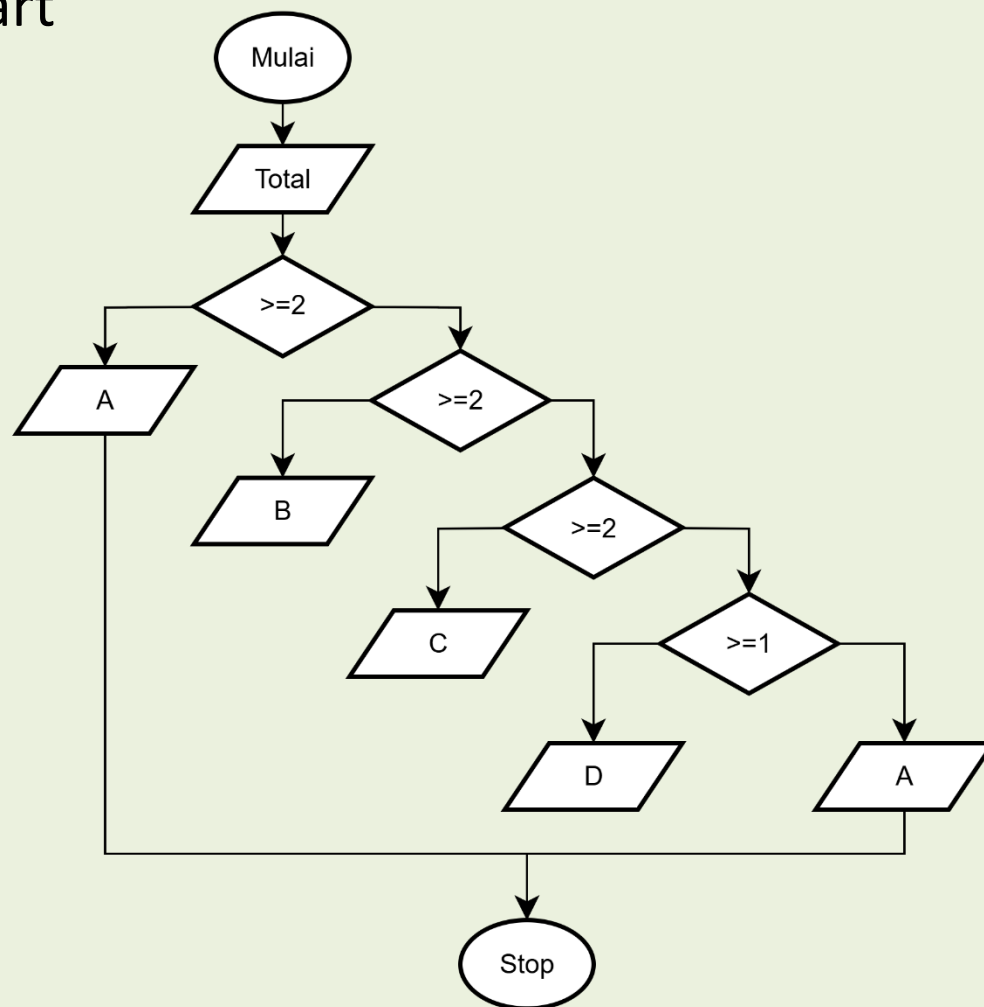
Mengenal dan Menggunakan Pemrograman Berbasis Teks

04

Kondisi Percabangan

c. Kondisi tiga percabangan atau lebih

Flowchart



Sintaks

```
totalNilai = int(input("Masukkan total nilai! "))
if totalNilai >= 240:
    print("A")
elif totalNilai >=225:
    print("B")
elif totalNilai >=210:
    print("C")
elif totalNilai >=195:
    print("D")
else:
    print("E")
```

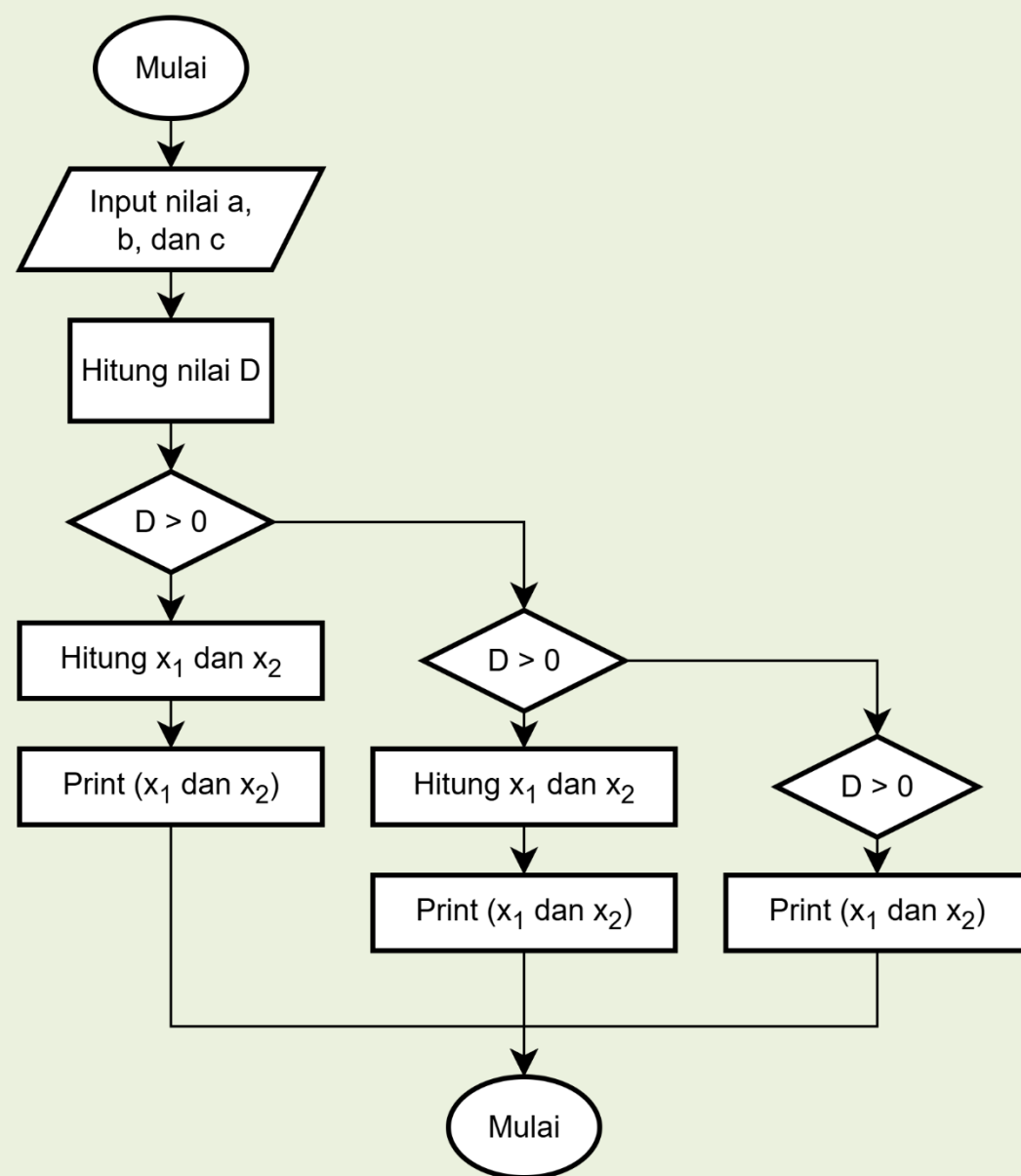
B

Mengenal dan Menggunakan Pemrograman Berbasis Teks

05

Menarapkan Kondisi Percabangan

Flowchart



Sintaks

```
import math
a = int(input("Masukkan nilai a! "))
b = int(input("Masukkan nilai b! "))
c = int(input("Masukkan nilai c! "))
D = b*b - (4*a*c)
if D > 0:
    akarD = math.sqrt(D)
    x1 = (-b + akarD) / (2*a)
    x2 = (-b - akarD) / (2*a)
    print("Akar x1 = ", x1)
    print("Akar x2 = ", x2)
elif D == 0:
    x1 = -b / (2*a)
    x2 = -b / (2*a)
    print("Akar x1 = ", x1)
    print("Akar x2 = ", x2)
else:
    print("Akar x1 imajiner")
    print("Akar x2 imajiner")
```

B

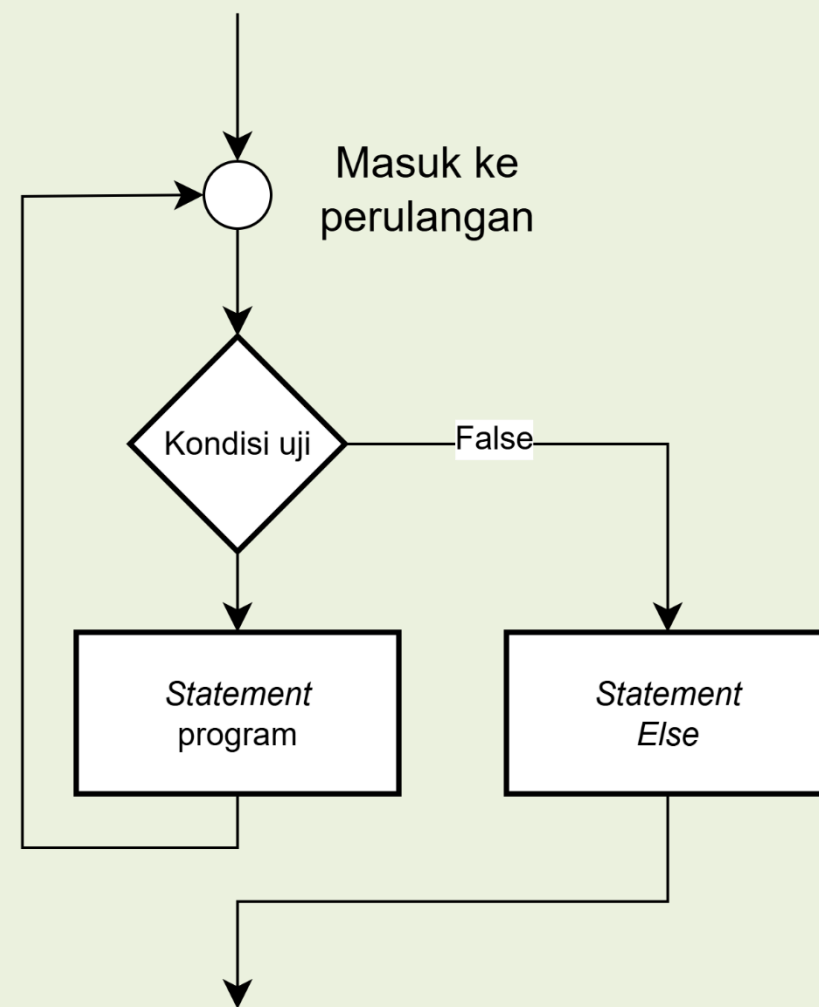
Mengenal dan Menggunakan Pemrograman Berbasis Teks

06

Menggunakan Perulangan

a. Menggunakan *statement* “while”

Flowchart



Sintaks

```
i = 1
while i < 8:
    print(i)
    i = i + 1
else:
    print("Proses iterasi sudah selesai")
```

B

Mengenal dan Menggunakan Pemrograman Berbasis Teks

06

Menggunakan Perulangan

b. Perulangan “for”

Menggunakan fungsi “range”

Contoh sintaks:

```
for x in range(5):  
    print(x)
```

Hasil:

```
0  
1  
2  
3  
4
```

Menggunakan fungsi “range else”

Contoh sintaks:

```
for x in range(5):  
    print(x)  
else:  
    print("Proses  
perulangan sudah 5 kali")
```

Hasil:

```
0  
1  
2  
3  
4  
Proses perulangan  
sudah 5 kali
```

B

Mengenal dan Menggunakan Pemrograman Berbasis Teks

06

Menggunakan Perulangan

b. Perulangan “for”

Perulangan di dalam daftar

Contoh sintaks:

```
Buah = ["Apel",  
"Belimbing",  
"Cempedak",  
"Durian", "Enau"]  
for x in Buah:  
    print(x)
```

Hasil:

```
Apel  
Belimbing  
Cempedak  
Durian  
Enau
```

Perulangan dengan selang tertentu

Contoh sintaks:

```
for x in range(0, 40, 5):  
    print(x)
```

Hasil:

```
0  
5  
10  
15  
20  
25  
30  
35
```

B

Mengenal dan Menggunakan Pemrograman Berbasis Teks

07

Menerapkan Perulangan

- Menghitung Deret

Sintaks program:

```
a = int(input("Masukkan bilangan pertama! "))
b = int(input("Masukkan selisih antar bilangan! "))
x = int(input("Masukkan bilangan terbesar! "))
an = a
jumlah = 0
i = 1
while an <= x:
    print("Bilangan ke", i, "=", an)
    jumlah = jumlah + an
    print("Jumlah bilangan ke", i, "=", jumlah)
    an = an + b
    i = i + 1
```

Hasil program:

```
Masukkan bilangan pertama! 3
Masukkan selisih antar bilangan! 4
Masukkan bilangan terbesar! 30
Bilangan ke 1 = 3
Jumlah bilangan ke 1 = 3
Bilangan ke 2 = 7
Jumlah bilangan ke 2 = 10
Bilangan ke 3 = 11
Jumlah bilangan ke 3 = 21
Bilangan ke 4 = 15
Jumlah bilangan ke 4 = 36
Bilangan ke 5 = 19
Jumlah bilangan ke 5 = 55
Bilangan ke 6 = 23
Jumlah bilangan ke 6 = 78
Bilangan ke 7 = 27
Jumlah bilangan ke 7 = 105
```