



# Algoritma dan Dasar Pemrograman

## 03. Algoritma & Struktur Keputusan

---

INF-103 Genap 2021-22

Husni

# Outline

- Algoritma, Pseudocode, & Flowchart
- Struktur Keputusan:
  - Struktur If–then–else
  - Operator relasi

# Contoh

- **Contoh 1:** Menentukan Nilai Akhir Mahasiswa
- **Contoh 2:** Mengonversi Panjang
- **Contoh 3:** Kalkulator Luas Persegi Panjang
- **Contoh 4:** Menentukan Nilai Terbesar

# Capaian Pembelajaran

- Menjelaskan (mendeskripsikan) algoritma dalam berbagai bentuk.
- Memahami perbedaan antara algoritma dan pseudocode.
- Membuat diagram alir program (*flowchart*).

# Algoritma, Pseudocode, & Flowchart

# Algoritma, Pseudocode, & Flowchart

- Apa itu algoritma?
  - Serangkaian **instruksi langkah demi langkah** untuk melakukan **tugas tertentu**.
  - Suatu algoritma haruslah:
    - Jernih (**jelas**), **tepat** dan **tidak ambigu**.
    - Memberikan solusi yang benar dalam **semua kasus**, dan **akhirnya berakhir**.
- Apa itu *pseudocode*?
  - Terks bahasa manusia (Inggris) yang **terlihat mirip dengan kode**
    - Tapi itu bukan kode yang sebenarnya (hanya terlihat mirip) .
    - Jadikan *pseudocode* sebagai cara untuk **mengekspresikan algoritma**.
- Apa itu *flowchart*?
  - Suatu **representasi grafis** dari urutan operasi dalam sistem informasi atau program.

# Algoritma, Pseudocode, & Flowchart

- Jelasnya...
  - **Algoritma** adalah **rangkaian langkah** yang diambil untuk **menyelesaikan masalah**, seperti resep berisi langkah-langkah yang diambil untuk membuat makanan!
  - Algoritma dapat diekspresikan dengan banyak cara:
    - **Pseudocode**: bukan “*real code*”, tetapi cara yang lebih formal daripada menuliskan langkah-langkah algoritmik saja
      - Sebagai contoh, mungkin programmer tidak tahu bahasa yang akan digunakan. Jadi, mereka hanya menulis kode-semu selama fase Pemecahan Masalah.
    - **Flowchart**: ini adalah representasi grafis dari algoritma
    - **Kode aktual**: ini dibuat pada fase Implementasi
      - Python, Java, C++, C, PHP, dll.

# Contoh 1: Penentuan Nilai Akhir Mahasiswa

Tuliskan algoritma dan kode-semu untuk menentukan nilai akhir mahasiswa dan perlihatkan apakah itu lulus atau gagal.  
Nilai akhir dihitung sebagai rata-rata dari empat nilai.

# Contoh 1: Penentuan Nilai Akhir Mahasiswa

## Algoritma

Tuliskan algoritma dan kode-semu untuk menentukan nilai akhir mahasiswa dan perlihatkan apakah itu lulus atau gagal. Nilai akhir dihitung sebagai rata-rata dari empat nilai.

### Teks langkah demi langkah:

1. Minta pengguna memasukkan 4 nilai (**Mark1**, **Mark2**, **Mark3**, **Mark4**)
2. Hitung rerata nilai (**Avg**) dengan menjumlahkan semua nilai dan membaginya dengan 4
3. Jika rata-rata (**Avg**) lebih daripada atau sama dengan 60
4.       Cetak “Lulus”
5. Jika tidak
6.       Cetak “Gagal”
7. Akhir blok Jika



# Contoh 1: Penentuan Nilai Akhir Mahasiswa

## Pseudocode

Tuliskan algoritma dan kode-semu untuk menentukan nilai akhir mahasiswa dan perlihatkan apakah itu lulus atau gagal. Nilai akhir dihitung sebagai rata-rata dari empat nilai.

### Pseudocode:

```
1. input Mark1, Mark2, Mark3, Mark4
2. Avg = (Mark1 + Mark2 + Mark3 + Mark4) / 4
3. if Avg >= 60:
4.     print "Lulus"
5. else:
6.     print "Gagal"
7. End if
```

# Contoh 1: Penentuan Nilai Akhir Mahasiswa

## Verifikasi Algoritma

```
1. input Mark1, Mark2, Mark3, Mark4
2. Avg = (Mark1 + Mark2 + Mark3 + Mark4) / 4
3. if Avg >= 60:
4.     print "Lulus"
5. else:
6.     print "Gagal"
7. End if
```

Test 1

Algoritma



Saya menunggu anda memberikan 4 nilai

Mark1 = 80, Mark2 = 90, Mark3 = 95, Mark4 = 85

Pengguna (User)



# Contoh 1: Penentuan Nilai Akhir Mahasiswa

## Verifikasi Algoritma

```
1. input Mark1, Mark2, Mark3, Mark4
2. Avg = (Mark1 + Mark2 + Mark3 + Mark4) / 4
3. if Avg >= 60:
4.     print "Lulus"
5. else:
6.     print "Gagal"
7. End if
```

Test 1

Algoritma



Mark1 = 80, Mark2 = 90, Mark3 = 95, Mark4 = 85  
 $Avg = (80 + 90 + 95 + 85) / 4 = 350 / 4 = 87.5$

Pengguna



# Contoh 1: Penentuan Nilai Akhir Mahasiswa

## Verifikasi Algoritma

```
1. input Mark1, Mark2, Mark3, Mark4
2. Avg = (Mark1 + Mark2 + Mark3 + Mark4) / 4
3. if Avg >= 60:
4.     print "Lulus"
5. else:
6.     print "Gagal"
7. End if
```

Test 1

Algoritma



Mark1 = 80, Mark2 = 90, Mark3 = 95, Mark4 = 85

Avg = 87.5

Avg >= 60 = 87.5 >= 60 = Yes

Pengguna



# Contoh 1: Penentuan Nilai Akhir Mahasiswa

## Verifikasi Algoritma

```
1. input Mark1, Mark2, Mark3, Mark4
2. Avg = (Mark1 + Mark2 + Mark3 + Mark4) / 4
3. if Avg >= 60:
4.     print "Lulus"
5. else:
6.     print "Gagal"
7. End if
```

Test 1

Algoritma



Mark1 = 80, Mark2 = 90, Mark3 = 95, Mark4 = 85  
Avg = 87.5

"Lulus"

Output:  
**Lulus**

Pengguna



# Contoh 1: Penentuan Nilai Akhir Mahasiswa

## Verifikasi Algoritma

```
1. input Mark1, Mark2, Mark3, Mark4
2. Avg = (Mark1 + Mark2 + Mark3 + Mark4) / 4
3. if Avg >= 60:
4.     print "Lulus"
5. else:
6.     print "Gagal"
7. End if
```

Test 1

Algoritma



Mark1 = 80, Mark2 = 90, Mark3 = 95, Mark4 = 85  
Avg = 87.5

Pengguna



# Contoh 1: Penentuan Nilai Akhir Mahasiswa

## Verifikasi Algoritma

```
1. input Mark1, Mark2, Mark3, Mark4
2. Avg = (Mark1 + Mark2 + Mark3 + Mark4) / 4
3. if Avg >= 60:
4.     print "Lulus"
5. else:
6.     print "Gagal"
7. End if
```

Test 2

Algoritma



Saya menantikan anda memberikan 4 nilai

Mark1 = 42, Mark2 = 55, Mark3 = 60, Mark4 = 37

Pengguna



# Contoh 1: Penentuan Nilai Akhir Mahasiswa

## Verifikasi Algoritma

```
1. input Mark1, Mark2, Mark3, Mark4
2. Avg = (Mark1 + Mark2 + Mark3 + Mark4) / 4
3. if Avg >= 60:
4.     print "Lulus"
5. else:
6.     print "Gagal"
7. End if
```

Test 2

Algoritma



Mark1 = 42, Mark2 = 55, Mark3 = 60, Mark4 = 37  
 $Avg = (42 + 55 + 60 + 37) / 4 = 194 / 4 = 48.5$

Pengguna





# Contoh 1: Penentuan Nilai Akhir Mahasiswa

## Verifikasi Algoritma

```
1. input Mark1, Mark2, Mark3, Mark4
2. Avg = (Mark1 + Mark2 + Mark3 + Mark4) / 4
3. if Avg >= 60:
4.     print "Lulus"
5. else:
6.     print "Gagal"
7. End if
```

Test 2

Algoritma



Mark1 = 42, Mark2 = 55, Mark3 = 60, Mark4 = 37

Avg = 48.5

Avg >= 60 = 48.5 >= 60 = No

Pengguna



# Contoh 1: Penentuan Nilai Akhir Mahasiswa

## Verifikasi Algoritma

```
1. input Mark1, Mark2, Mark3, Mark4
2. Avg = (Mark1 + Mark2 + Mark3 + Mark4) / 4
3. if Avg >= 60:
4.     print "Lulus"
5. else:
6.     print "Gagal"
7. End if
```

Test 2

Algoritma



Mark1 = 42, Mark2 = 55, Mark3 = 60, Mark4 = 37  
Avg = 48.5

Pengguna



# Contoh 1: Penentuan Nilai Akhir Mahasiswa

## Verifikasi Algoritma

```
1. input Mark1, Mark2, Mark3, Mark4
2. Avg = (Mark1 + Mark2 + Mark3 + Mark4) / 4
3. if Avg >= 60:
4.     print "Lulus"
5. else:
6.     print "Gagal"
7. End if
```

Test 2

Algoritma



Mark1 = 42, Mark2 = 55, Mark3 = 60, Mark4 = 37  
Avg = 48.5

"Gagal"

Pengguna



Output:  
**Gagal**

# Contoh 1: Penentuan Nilai Akhir Mahasiswa

## Verifikasi Algoritma

```
1. input Mark1, Mark2, Mark3, Mark4
2. Avg = (Mark1 + Mark2 + Mark3 + Mark4) / 4
3. if Avg >= 60:
4.     print "Lulus"
5. else:
6.     print "Gagal"
7. End if
```

Test 2

Algoritma



Mark1 = 42, Mark2 = 55, Mark3 = 60, Mark4 = 37  
Avg = 48.5

Pengguna




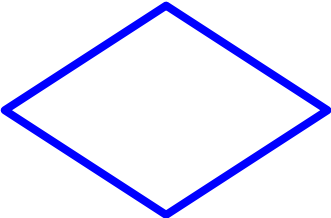



Output:  
Gagal

# Flowchart

- Suatu representasi grafis dari rangkaian operasi dalam suatu sistem informasi atau program komputer.
- Program flowcharts menunjukkan urutan instruksi dalam satu program atau subrutin.
  - menunjukkan logika suatu algoritma
  - menekankan langkah-langkah individu dan interkoneksinya
  - Contoh: mengontrol aliran dari satu tindakan ke tindakan berikutnya.
- Beberapa simbol digunakan untuk menggambarkan setiap jenis dari flowchart.

# Simbol Flowchart

<u>Nama</u>	<u>Simbol</u>	<u>Kegunaan dalam Flowchart</u>
Oval		Menunjukkan awal atau akhir program.
Parallelogram		Menunjukkan operasi input/output.
Rectangle		Menunjukkan suatu proses yang dilakukan. Contoh : penjumlahan, pengurangan, pembagian.
Diamond		Menunjukkan keputusan atau cabang. Program harus dilanjutkan mengikuti satu dari dua rute (Mis.If/Then/Else)
Flow line		Menunjukkan arah aliran logika dalam program

# Flowcharts

- Apakah diagram alir benar-benar diperlukan atau membantu?
  - Di dunia nyata, **programs tidak hanya 1000 baris**.
  - **Program** terdiri dari **ratusan ribu baris kode**.
    - Bahkan Jutaan baris kode.
  - Dapatkah kita hanya menggunakan bahasa Inggris untuk menggambarkan program?
    - Tentu saja mungkin, tetapi kita akan memperoleh sebuah buku!
  - Karena itu, berpikir dengan **flowcharts** adalah **lebih mudah**, cara **lebih jelas** untuk **dengan cepat memahami** apa yang **program akan kerjakan**.

# Flowcharts

- Apakah **flowcharts** sungguh **diperlukan** dan **membantu**?
  - Jadi secara ringkas, **yes**, ini betul-betul membantu.
- Tentu saja, dalam kuliah ini, kita hanya berurusan dengan **program yang jauh lebih kecil**.
  - Apakah kita membutuhkan diagram alir untuk program kecil??
  - Mungkin saja TIDAK.
  - Namun, mahasiswa harus membiasakan diri membuat diagram alir dengan **program lebih kecil dan lebih mudah**.
  - Kemudian, itu akan mudah dilakukan untuk **program yang lebih besar**.



## Contoh 2: Mengkonversi Panjang

Tuliskan suatu Algoritma, Pseudocode, dan gambarkan flowchart untuk mengkonversi panjang dalam kaki (*feet*) ke dalam *centimeter*.

### Algoritma:

1. Masukkan panjang dalam satuan kaki atau *feet* (LFT)
2. Hitung panjang dalam cm (LCM) dengan mengalikan LFT dengan 30
3. Cetak panjang dalam cm (LCM)

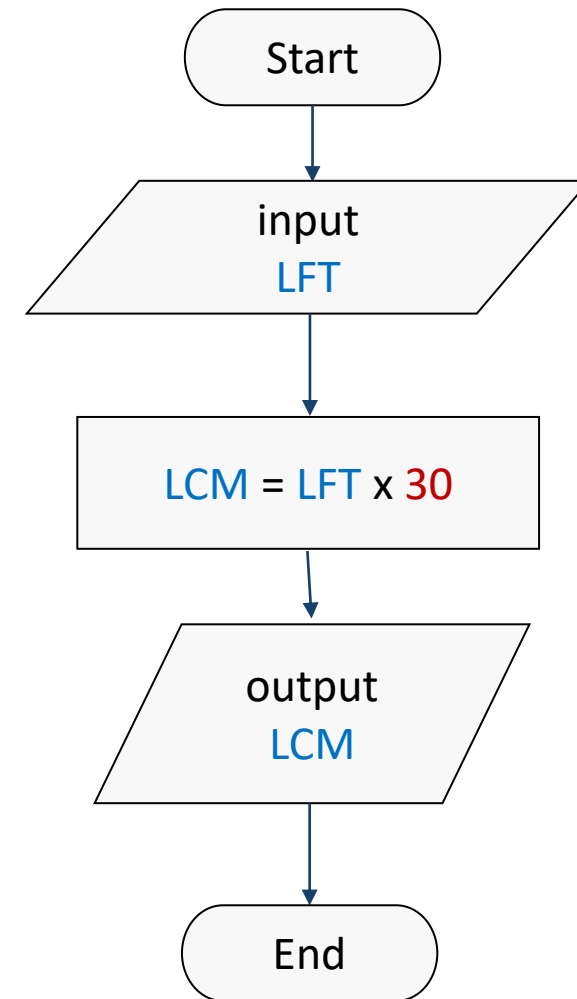
# Contoh 2: Mengkonversi Panjang

## Algoritma

Pseudocode:

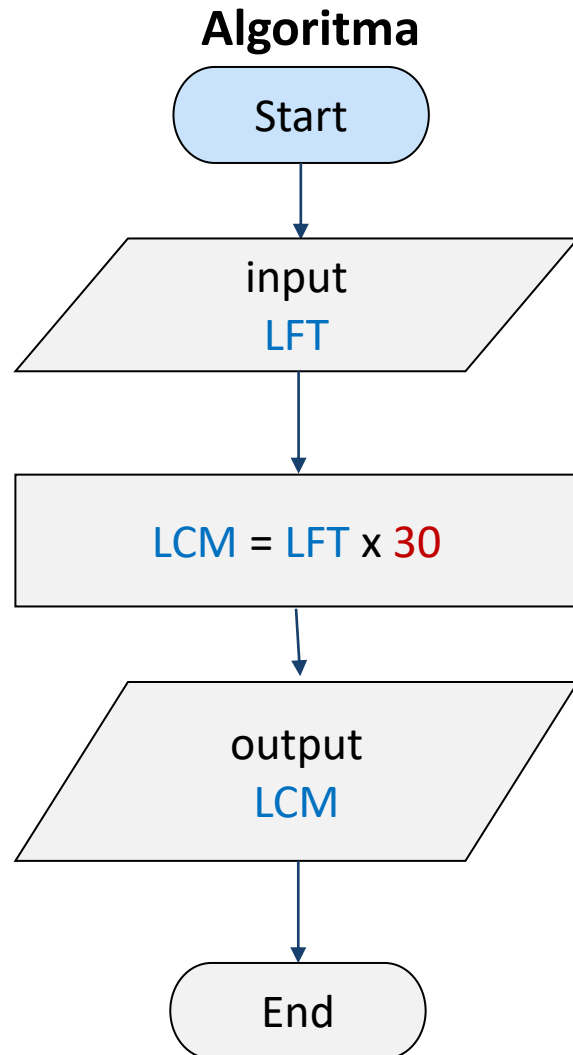
1. LFT = input "Panjang dalam feet"
2. LCM = LFT x 30
3. print LCM

Flowchart:



# Contoh 2: Mengkonversi Panjang

## Verifikasi Algoritma



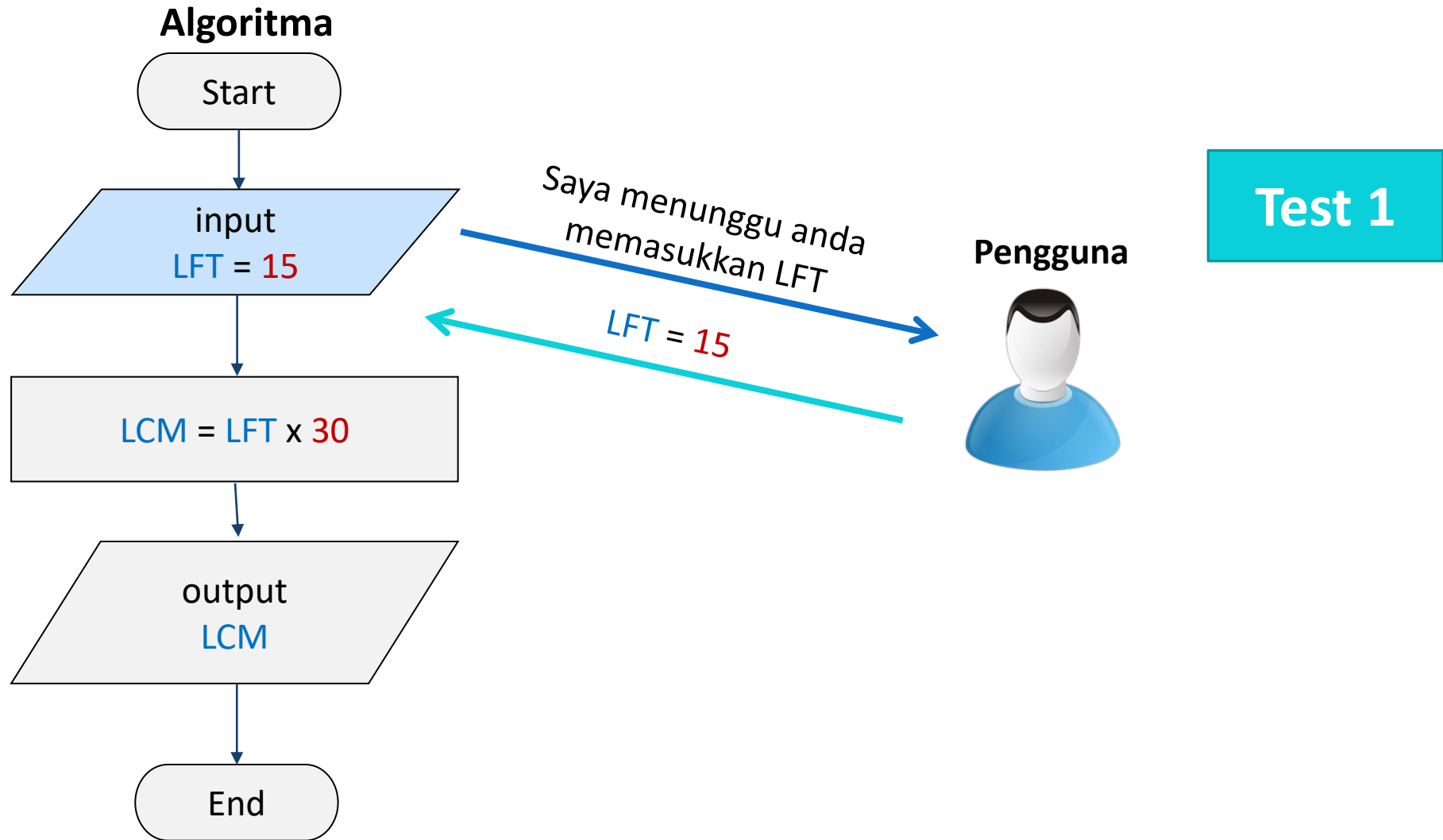
**Pengguna**



**Test 1**

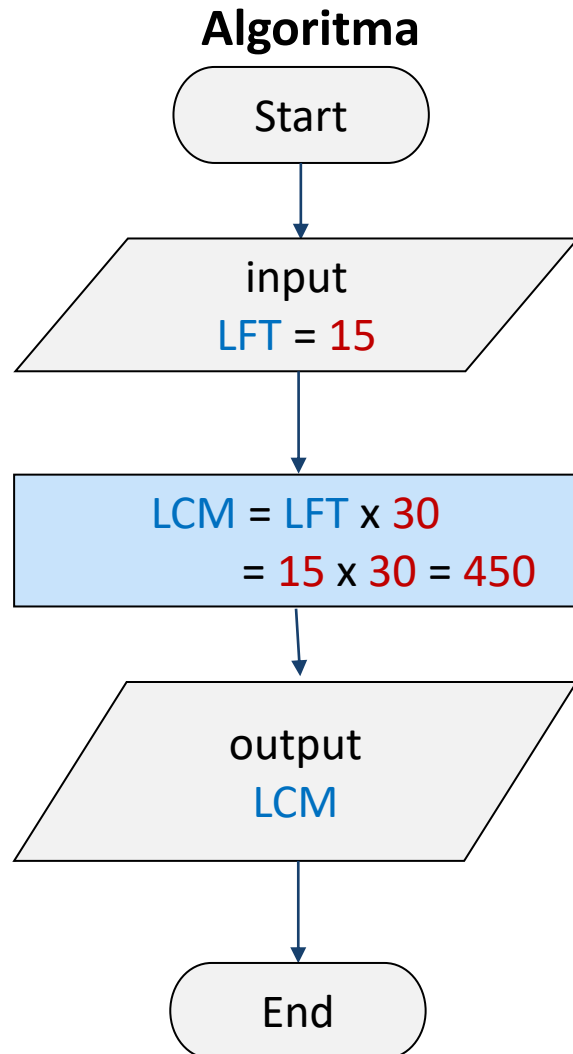
# Contoh 2: Mengkonversi Panjang

## Verifikasi Algoritma



# Contoh 2: Mengkonversi Panjang

## Verifikasi Algoritma



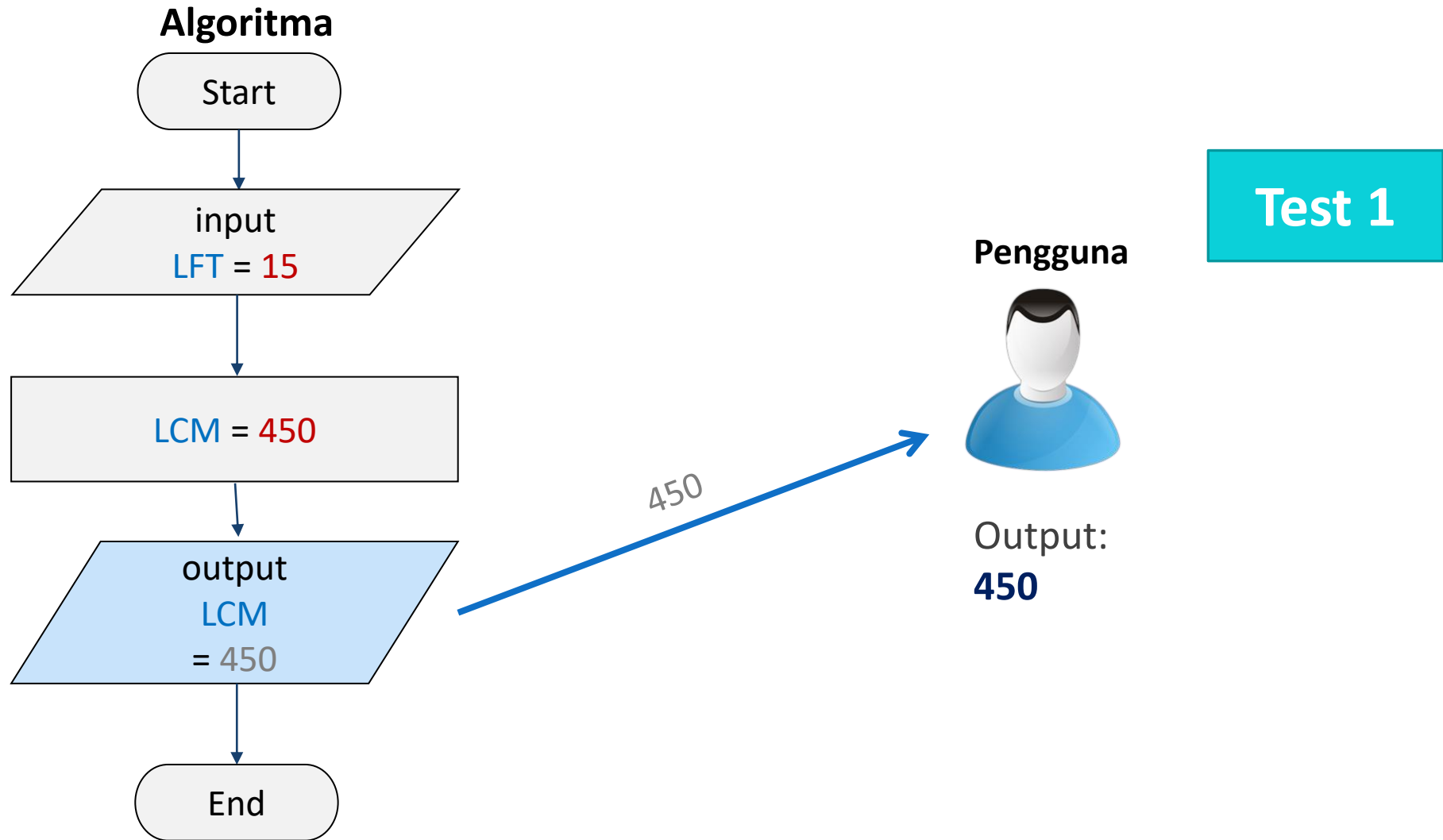
Pengguna



Test 1

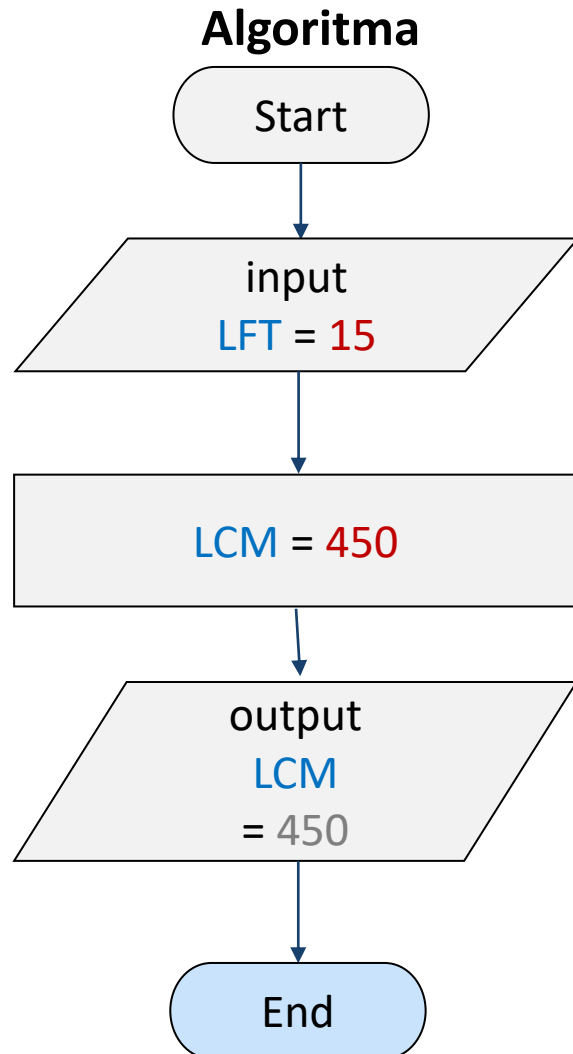
# Contoh 2: Mengkonversi Panjang

## Verifikasi Algoritma



# Contoh 2: Mengkonversi Panjang

## Verifikasi Algoritma



**Pengguna**



Output:  
450

**Test 1**

# Contoh 3: Kalkulator Luas Segi Empat

Tuliskan suatu Algoritma, Pseudocode, dan gambar diagram alir yang akan membaca dua sisi persegi panjang dan menghitung luasnya.

## Algoritma:

1. Masukkan panjang ( $L$ ) dan lebar ( $W$ ) dari suatu segi empat
2. Hitung luas ( $A$ ) dengan mengalikan  $L$  dengan  $W$
3. Cetak  $A$



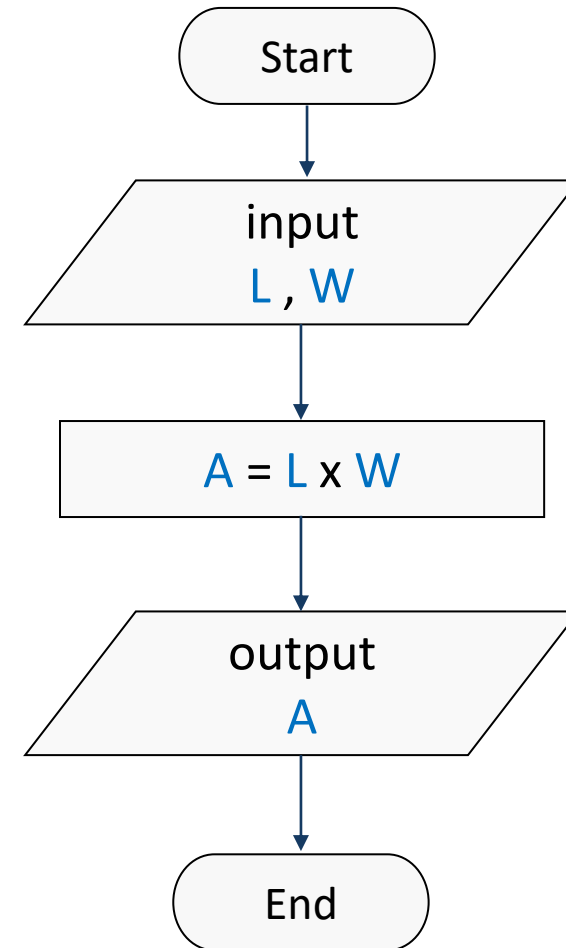
# Contoh 3: Kalkulator Luas Segi Empat

## Algoritma

Pseudocode:

1. input  $L, W$
2.  $A = L \times W$
3. print  $A$

Flowchart:



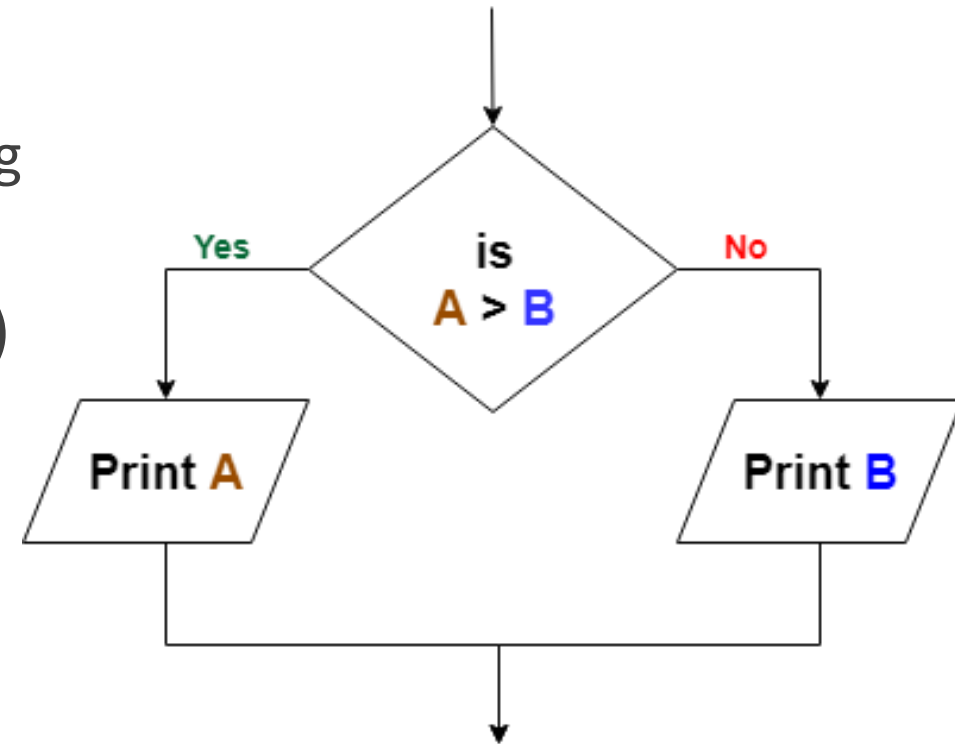
# Struktur Keputusan

Struktur if-then-else  
Operator Relasi

# Struktur Keputusan

- Ekspresi  $A > B$  merupakan suatu ekspresi logis
- Ini mendeskripsikan suatu kondisi yang memerlukan uji (test)
- **if  $A > B$  bernilai true (Jika A lebih besar daripada B)** kita mengambil aksi sisi kiri: cetak nilai dari A
- **if  $A > B$  bernilai false (Jika A tidak lebih besar daripada B)** kita mengambil aksi sisi kanan: cetak nilai dari B

Catatan: Print = Cetak = Output



# Struktur if–then–else

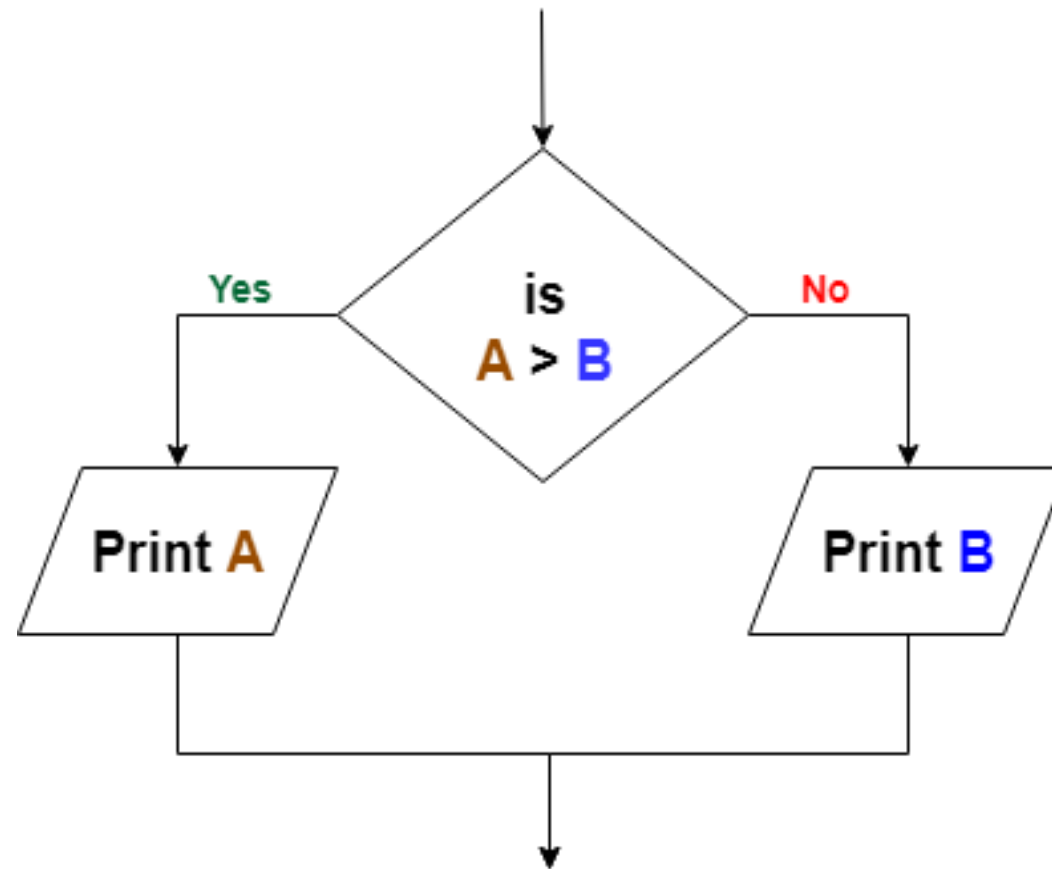
Strukturnya adalah sebagai berikut

```
if condition then
    true alternative
else
    false alternative
end if
```

# Struktur if-then-else

Algoritma dengan flowchart adalah sebagai berikut:

```
if A > B then  
    print A  
else  
    print B  
End if
```



# Operator Relasi

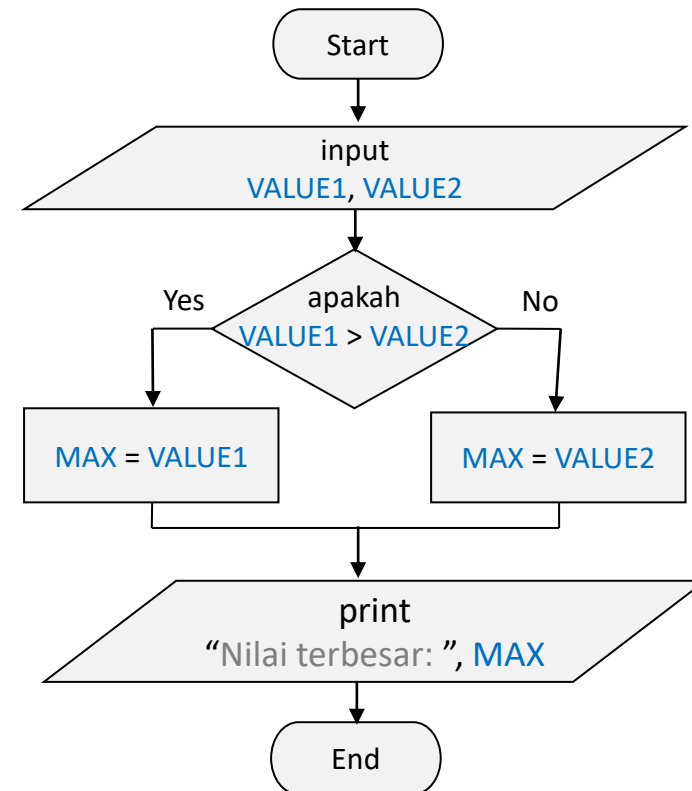
Operator	Deskripsi
>	Lebih besar daripada
<	Kurang (lebih kecil) daripada
==	Sama dengan
$\geq$ Or $\geq$	Lebih besar daripada atau sama dengan
$\leq$ Or $\leq$	Lebih kecil daripada atau sama dengan
$\neq$ Or $\neq$	Tidak sama dengan

# Contoh 4: Menentukan Nilai Terbesar

Tuliskan Pseudocode untuk membaca dua nilai, kemudian temukan nilai terbesar dan cetak nilai tersebut dengan pesan penjelasan.

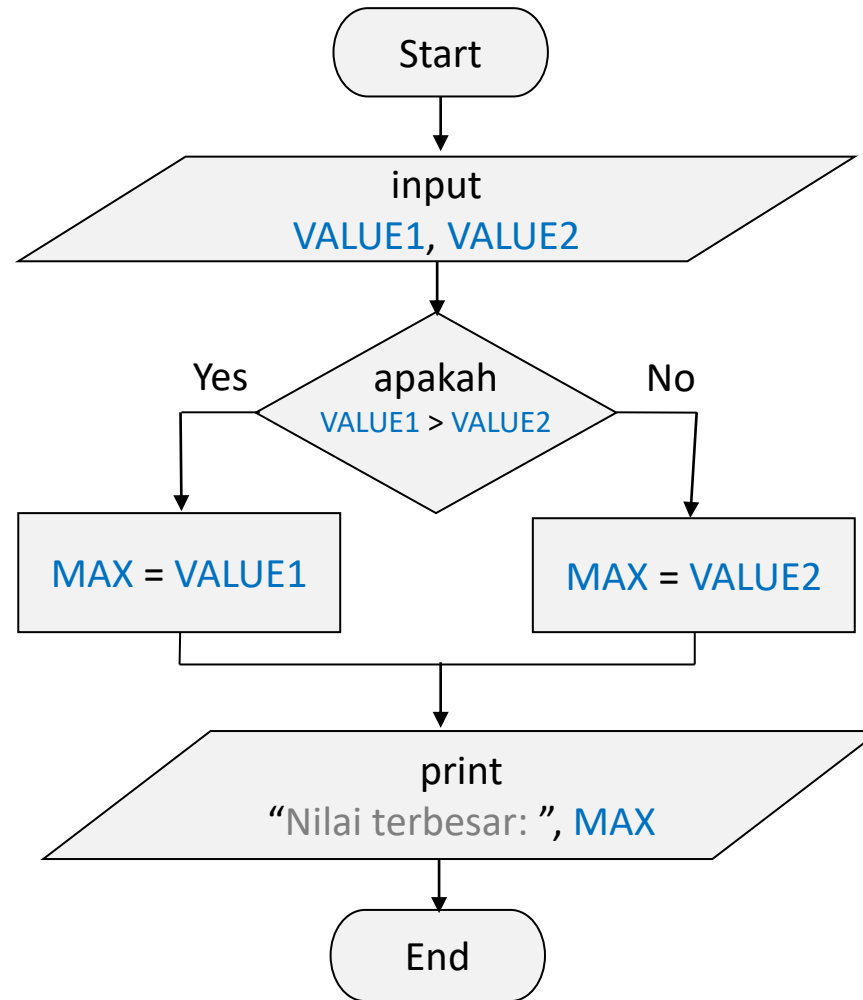
Pseudocode:

```
1. Input VALUE1, VALUE2
2. if (VALUE1 > VALUE2) then
3.     MAX = VALUE1
4. else
5.     MAX = VALUE2
6. endif
7. print "Nilai terbesar: ", MAX
```



# Contoh 4: Menentukan Nilai Terbesar

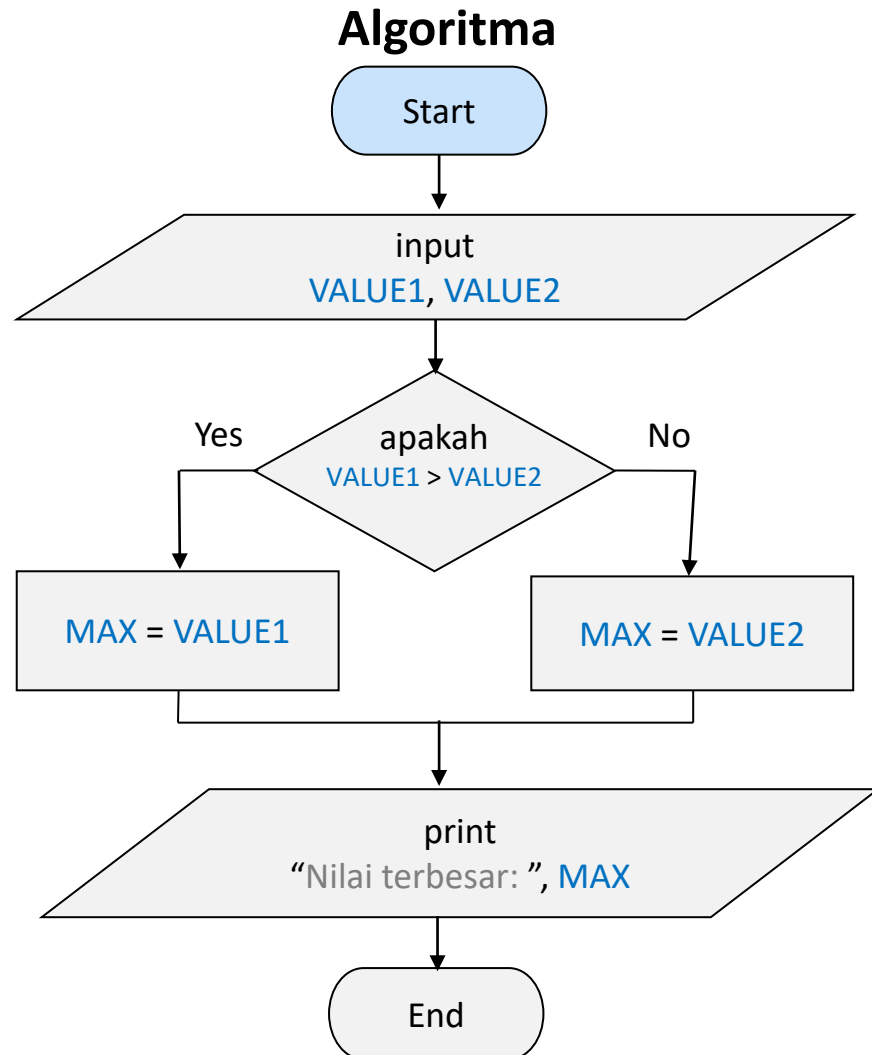
## Algoritma





# Contoh 4: Menentukan Nilai Terbesar

## Verifikasi Algoritma



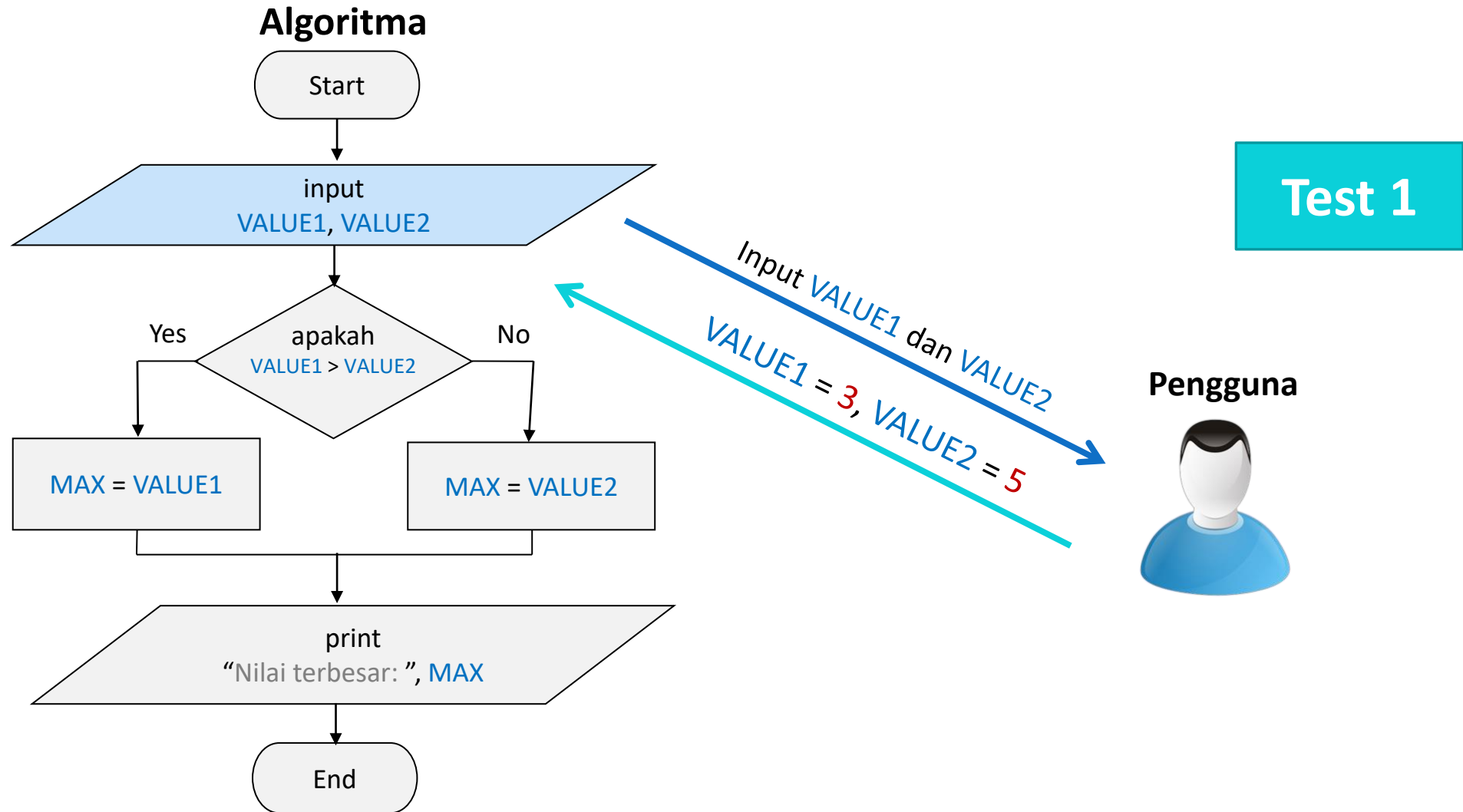
**Test 1**

**Pengguna**



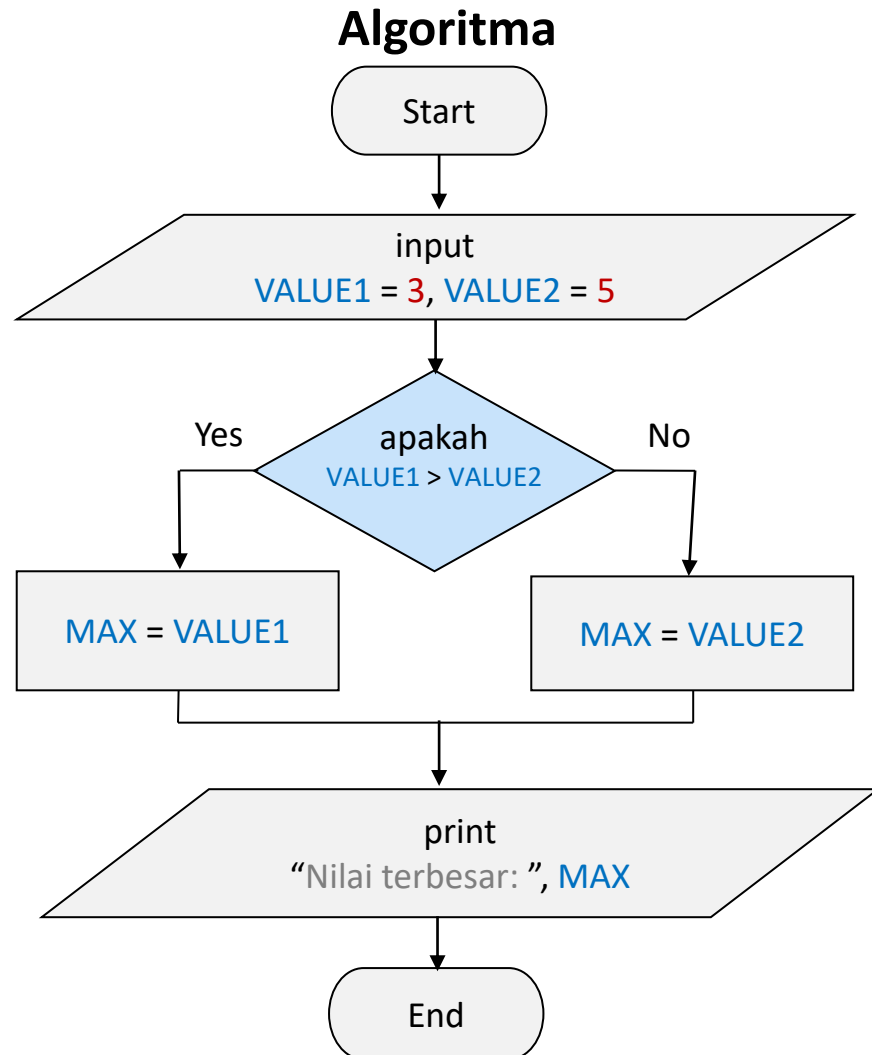
# Contoh 4: Menentukan Nilai Terbesar

## Verifikasi Algoritma



# Contoh 4: Menentukan Nilai Terbesar

## Verifikasi Algoritma



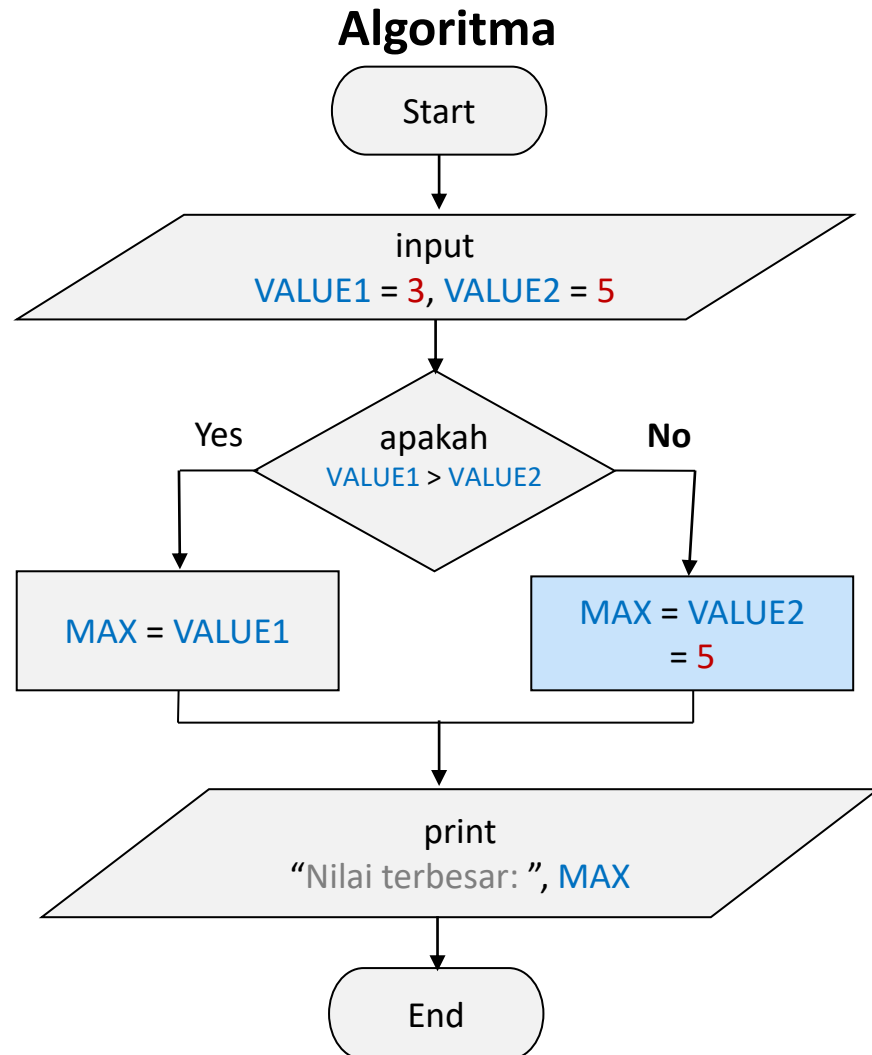
**Test 1**

**Pengguna**



# Contoh 4: Menentukan Nilai Terbesar

## Verifikasi Algoritma



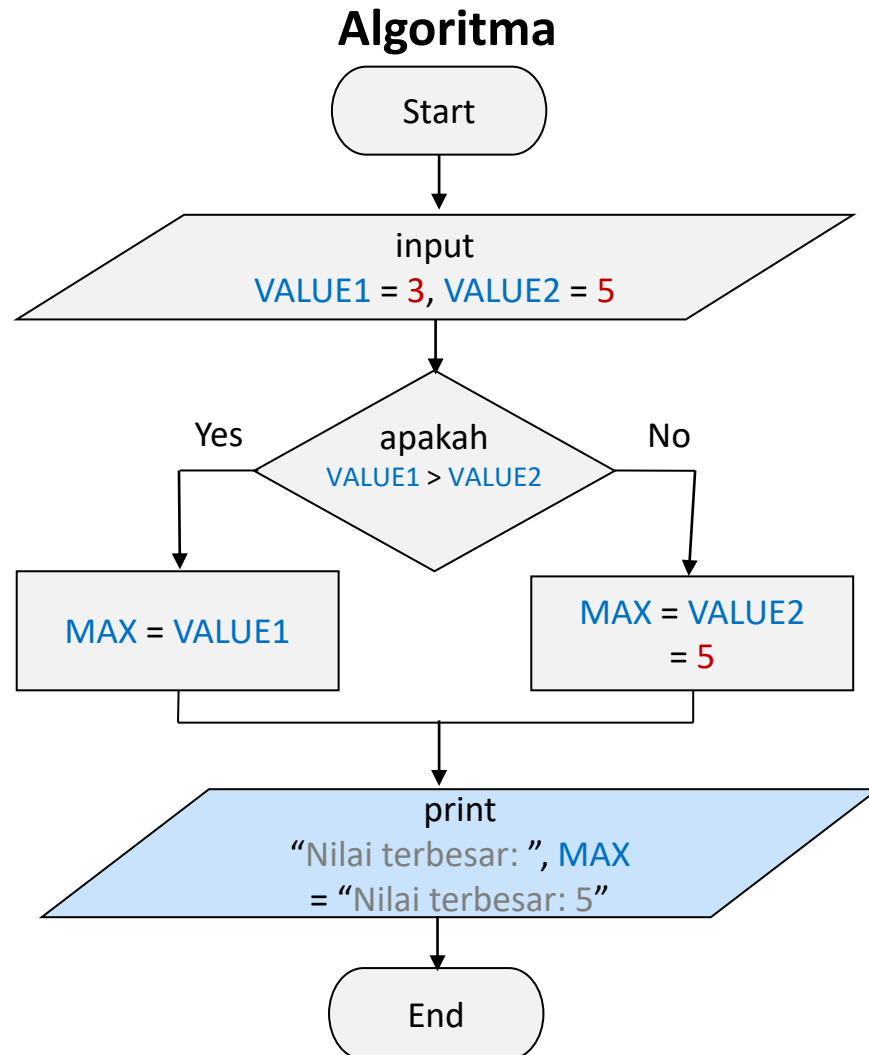
**Test 1**

**Pengguna**



# Contoh 4: Menentukan Nilai Terbesar

## Verifikasi Algoritma



**Test 1**

**Pengguna**

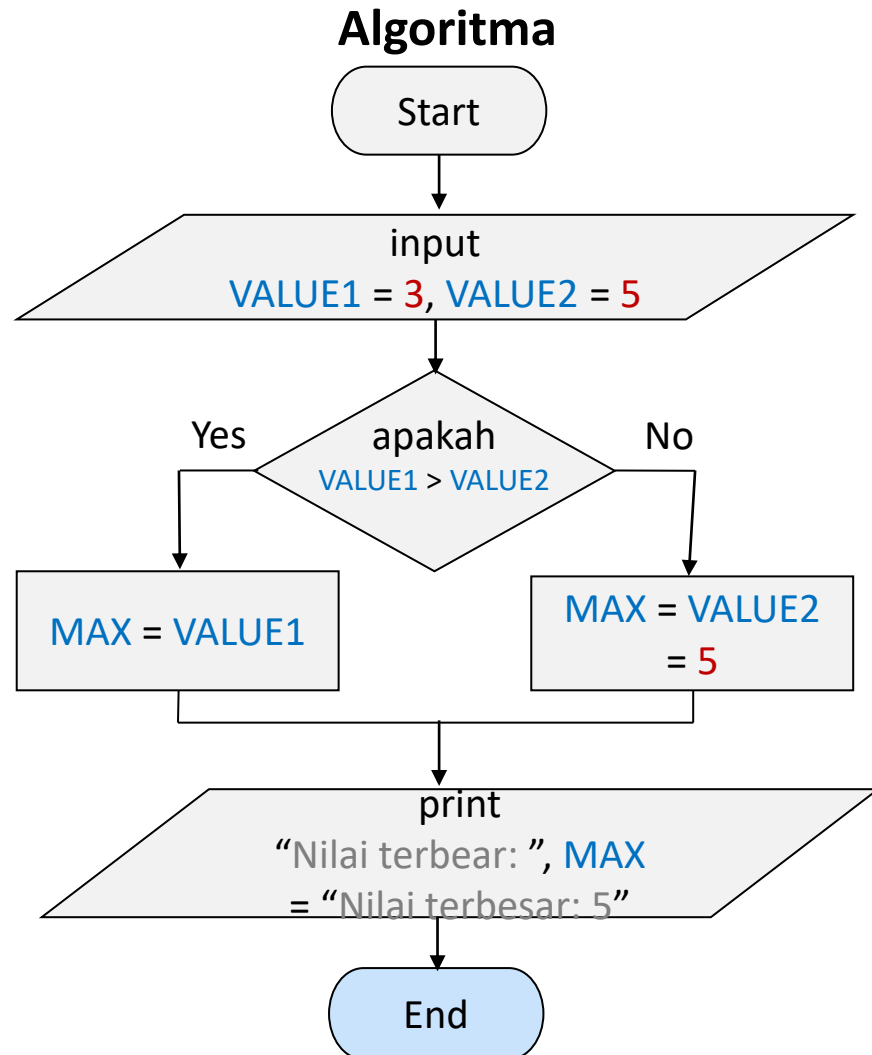


"Nilai terbesar: 5"

Output:  
**Nilai terbesar: 5**

# Contoh 4: Menentukan Nilai Terbesar

## Verifikasi Algoritma



**Test 1**

**Pengguna**



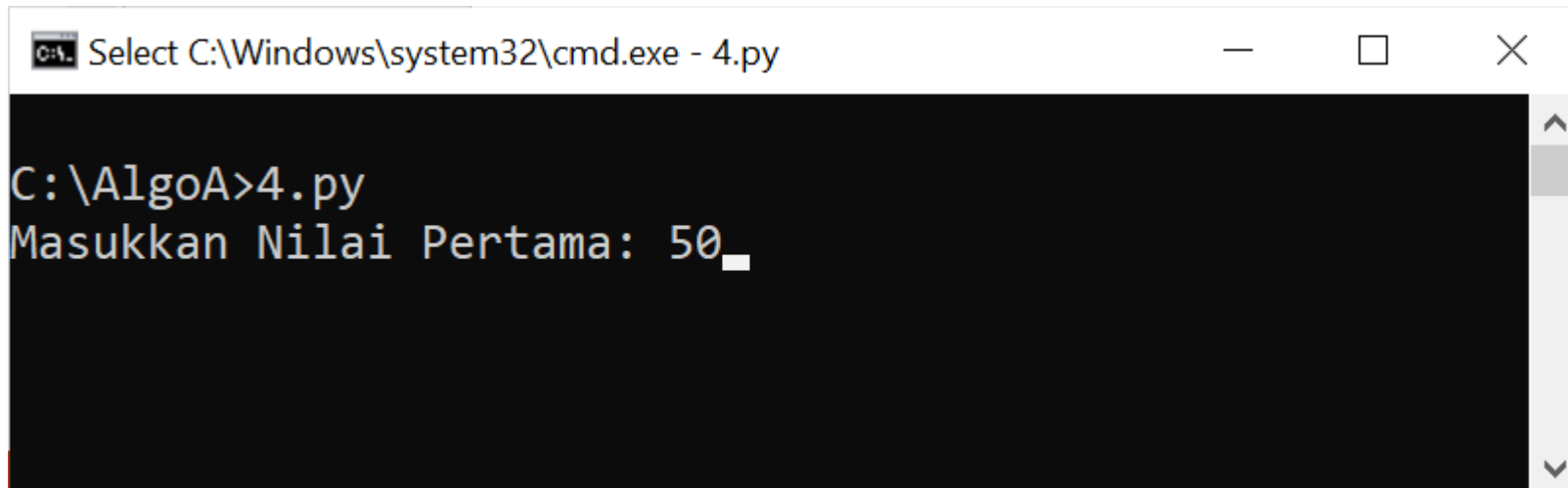
Output:  
Nilai terbesar: 5

# Contoh 4: Menentukan Nilai Terbesar

## Verifikasi Algoritma

```
1. Input VALUE1, VALUE2
2. if (VALUE1 > VALUE2) then
3.     MAX = VALUE1
4. else
5.     MAX = VALUE2
6. endif
7. print "Nilai terbesar: ", MAX
```

Test 2



The screenshot shows a Windows command prompt window titled "Select C:\Windows\system32\cmd.exe - 4.py". The prompt is at "C:\AlgoA>". The user has entered "4.py" and the prompt has moved to the next line. The user has then entered "Masukkan Nilai Pertama: 50\_" and the prompt has moved to the next line. The window has a black background and white text.

```
Select C:\Windows\system32\cmd.exe - 4.py

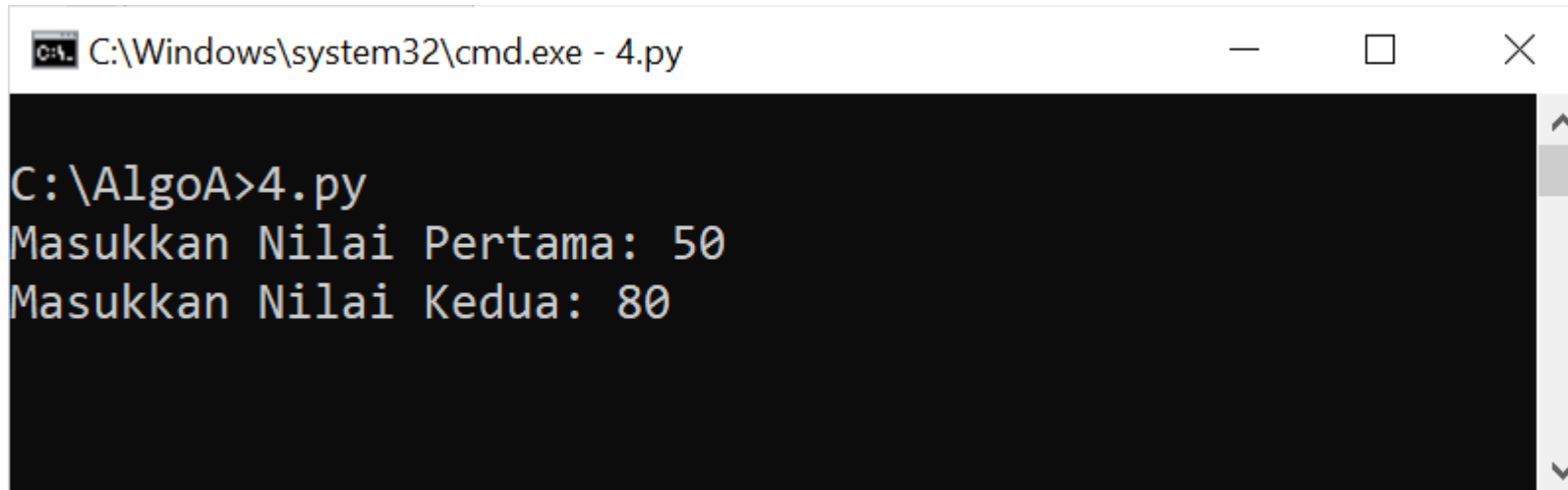
C:\AlgoA>4.py
Masukkan Nilai Pertama: 50_
```

# Contoh 4: Menentukan Nilai Terbesar

## Verifikasi Algoritma

1. Input VALUE1, VALUE2
2. if (VALUE1 > VALUE2) then
3.       MAX = VALUE1
4. else
5.       MAX = VALUE2
6. endif
7. print “Nilai terbesar: ”, MAX

Test 2



```
C:\Windows\system32\cmd.exe - 4.py

C:\AlgoA>4.py
Masukkan Nilai Pertama: 50
Masukkan Nilai Kedua: 80
```

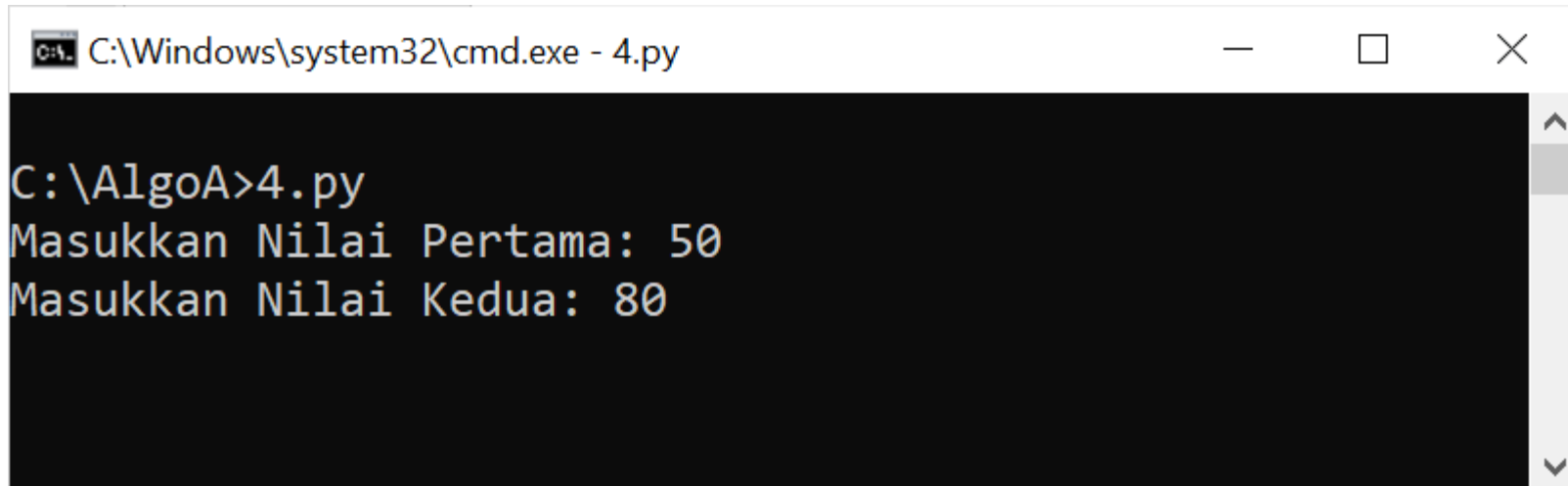


# Contoh 4: Menentukan Nilai Terbesar

## Verifikasi Algoritma

1. Input `VALUE1`, `VALUE2`
2. `if (VALUE1 > VALUE2) then`
3.       `MAX = VALUE1`
4. `else`
5.       `MAX = VALUE2`
6. `endif`
7. `print "Nilai terbesar: ", MAX`

Test 2



```
C:\Windows\system32\cmd.exe - 4.py

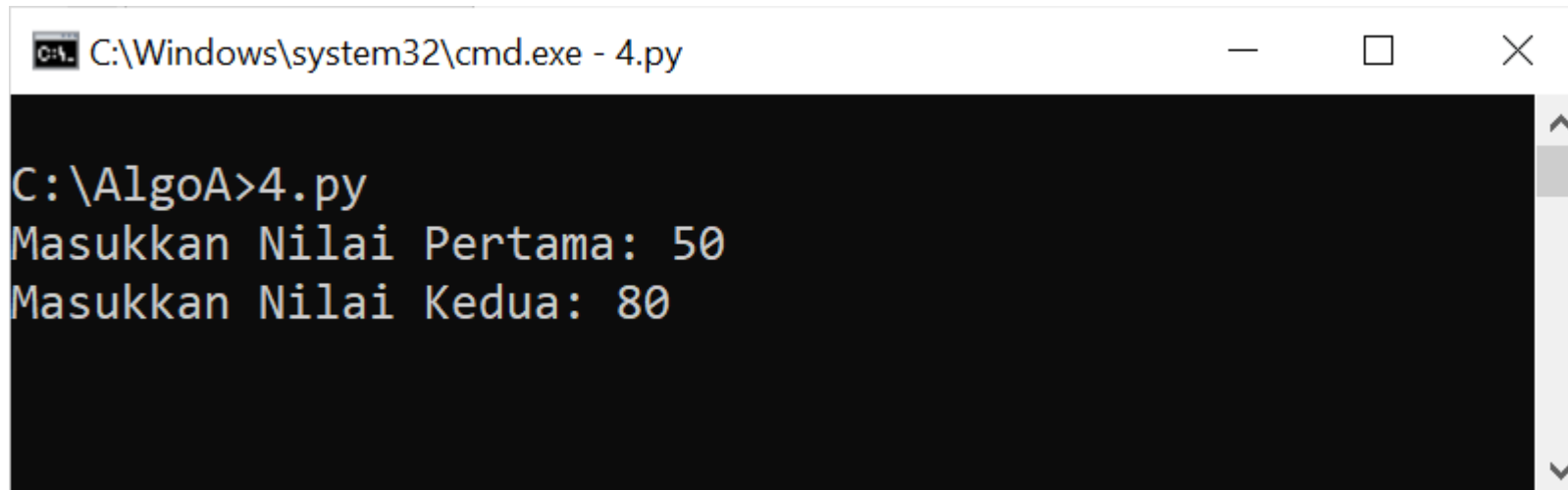
C:\AlgoA>4.py
Masukkan Nilai Pertama: 50
Masukkan Nilai Kedua: 80
```

# Contoh 4: Menentukan Nilai Terbesar

## Verifikasi Algoritma

1. Input `VALUE1`, `VALUE2`
2. if (`VALUE1` > `VALUE2`) then
3.       `MAX` = `VALUE1`
4. else
5.       `MAX` = `VALUE2`
6. endif
7. print “`Nilai terbesar:` ”, `MAX`

Test 2



```
C:\Windows\system32\cmd.exe - 4.py

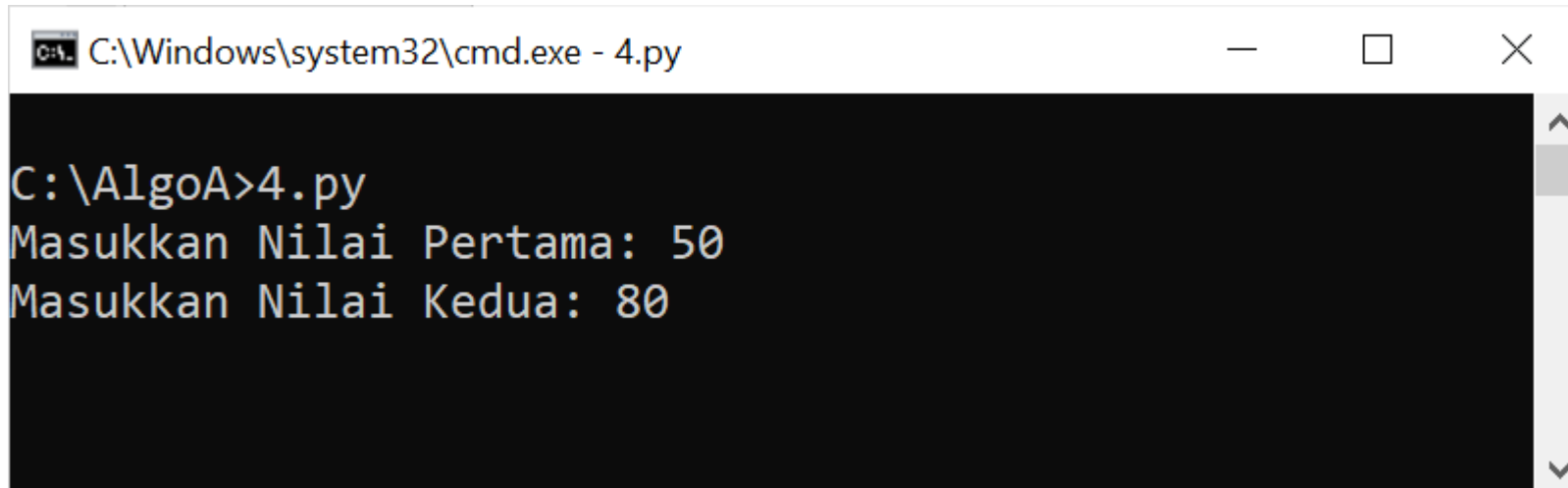
C:\AlgoA>4.py
Masukkan Nilai Pertama: 50
Masukkan Nilai Kedua: 80
```

# Contoh 4: Menentukan Nilai Terbesar

## Verifikasi Algoritma

1. Input `VALUE1`, `VALUE2`
2. if (`VALUE1` > `VALUE2`) then
3.         `MAX` = `VALUE1`
4. else
5.         `MAX` = `VALUE2`
6. endif
7. print “Nilai terbesar: ”, `MAX`

Test 2



```
C:\Windows\system32\cmd.exe - 4.py

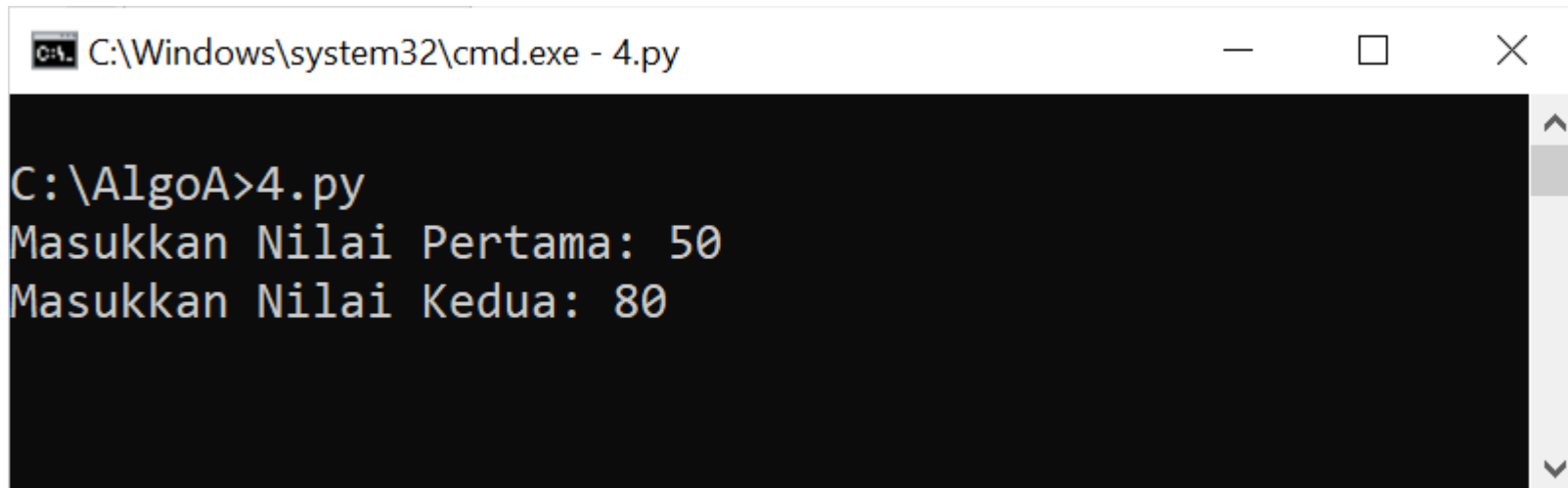
C:\AlgoA>4.py
Masukkan Nilai Pertama: 50
Masukkan Nilai Kedua: 80
```

# Contoh 4: Menentukan Nilai Terbesar

## Verifikasi Algoritma

```
1. Input VALUE1, VALUE2
2. if (VALUE1 > VALUE2) then
3.     MAX = VALUE1
4. else
5.     MAX = VALUE2
6. endif
7. print "Nilai terbesar: ", MAX
```

Test 2



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - 4.py". The prompt is at "C:\AlgoA>". The user has entered "4.py", and the prompt has moved to the next line. The user then entered "Masukkan Nilai Pertama: 50", and the prompt moved to the next line. Finally, the user entered "Masukkan Nilai Kedua: 80", and the prompt moved to the next line. The output of the script is not visible in the screenshot.

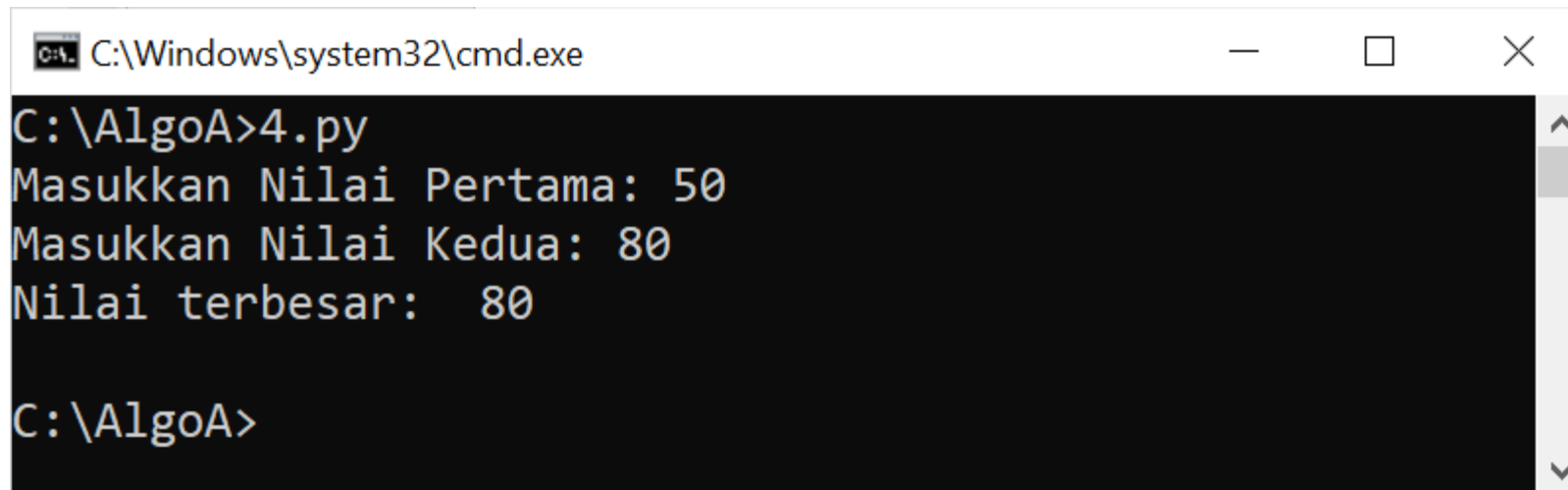
```
C:\Windows\system32\cmd.exe - 4.py
C:\AlgoA>4.py
Masukkan Nilai Pertama: 50
Masukkan Nilai Kedua: 80
```

# Contoh 4: Menentukan Nilai Terbesar

## Verifikasi Algoritma

1. Input `VALUE1`, `VALUE2`
2. if (`VALUE1` > `VALUE2`) then
3.       `MAX` = `VALUE1`
4. else
5.       `MAX` = `VALUE2`
6. endif
7. print “Nilai terbesar: ”, `MAX`

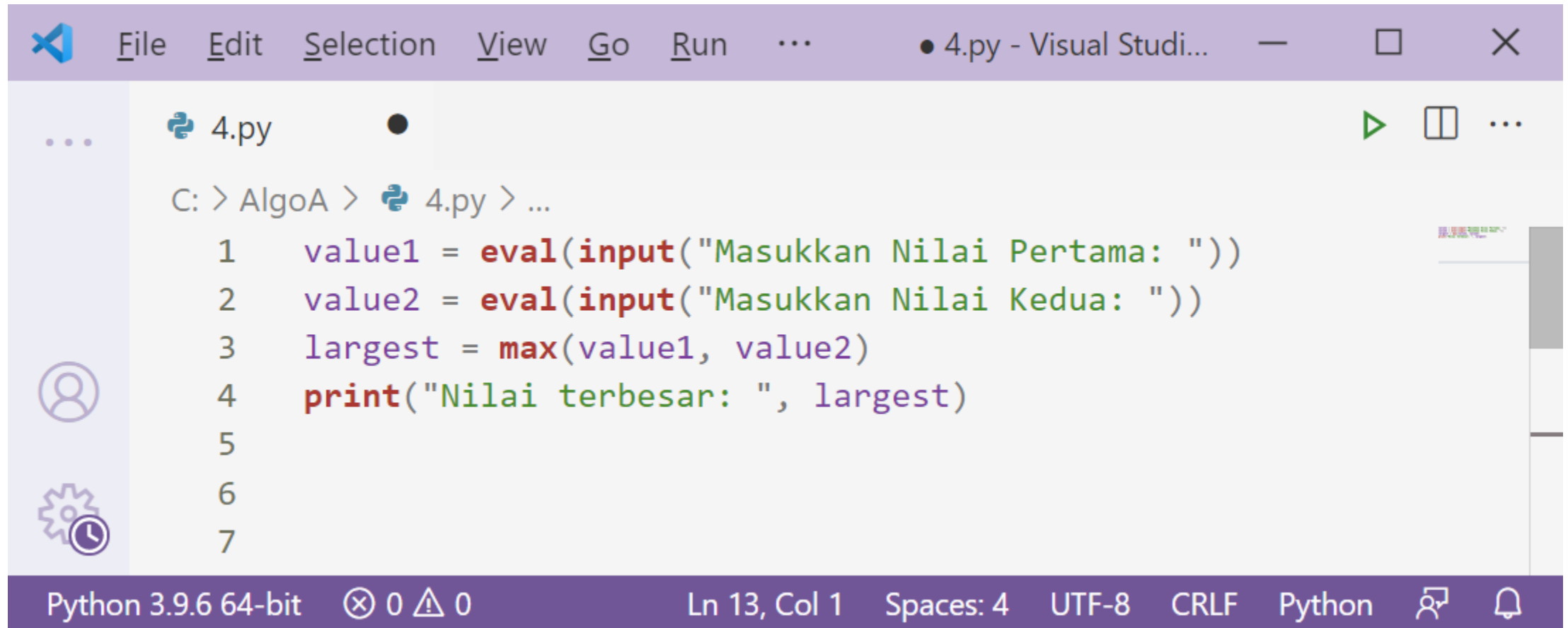
Test 2



```
C:\Windows\system32\cmd.exe
C:\AlgoA>4.py
Masukkan Nilai Pertama: 50
Masukkan Nilai Kedua: 80
Nilai terbesar: 80
C:\AlgoA>
```

# Contoh 4: Menentukan Nilai Terbesar

## Kode Python



The image shows a screenshot of the Visual Studio Code editor interface. The title bar at the top indicates the file is '4.py' in a 'Visual Studi...' window. The menu bar includes 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', and a dropdown menu. The file explorer on the left shows '4.py' as the active file. The editor area displays a Python script with the following code:

```
C: > AlgoA > 4.py > ...  
1  value1 = eval(input("Masukkan Nilai Pertama: "))  
2  value2 = eval(input("Masukkan Nilai Kedua: "))  
3  largest = max(value1, value2)  
4  print("Nilai terbesar: ", largest)  
5  
6  
7
```

The status bar at the bottom provides details about the environment: 'Python 3.9.6 64-bit', '0' errors and '0' warnings, 'Ln 13, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', 'Python', and icons for search and notifications.