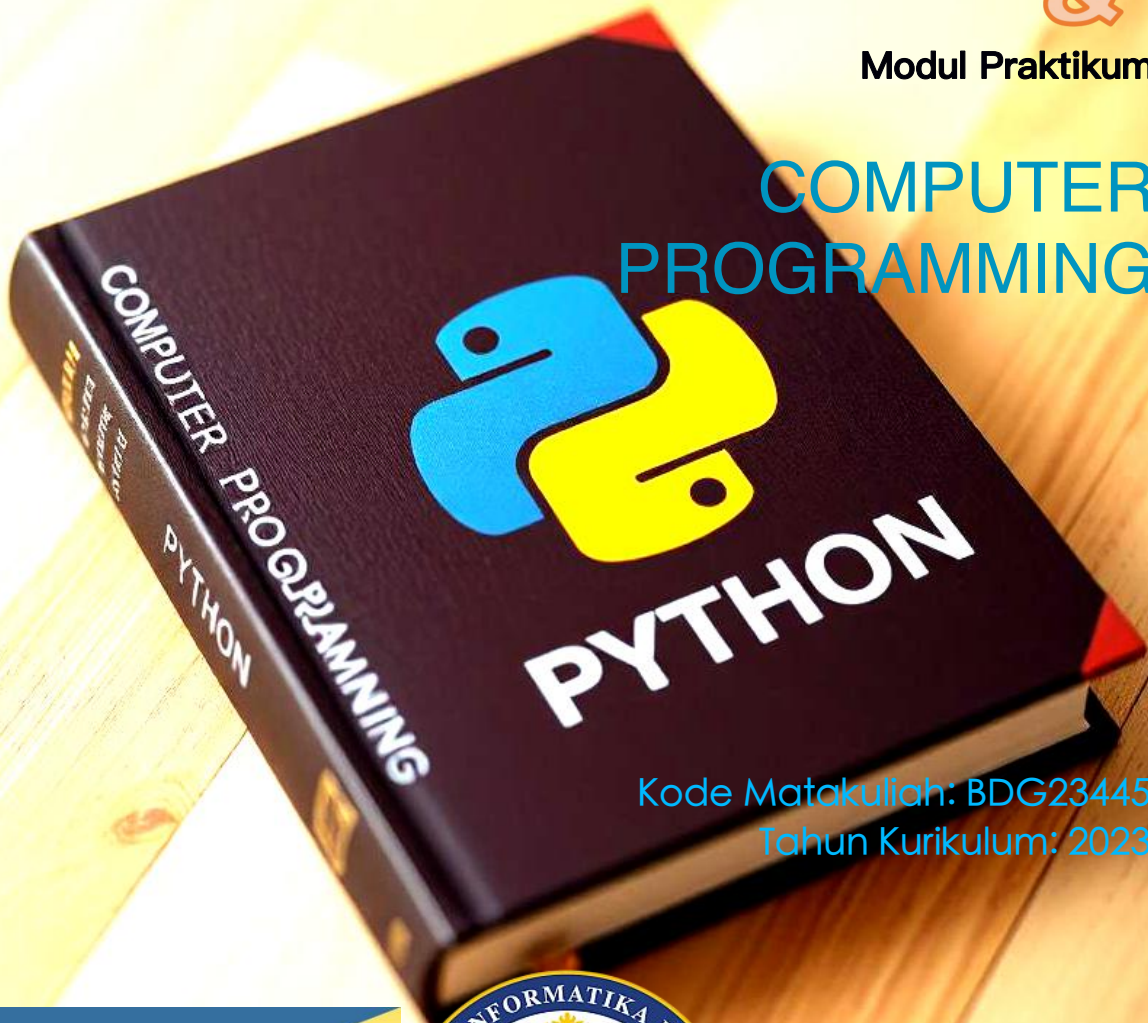




COMPUTER PROGRAMMING



Kode Matakuliah: BDG23445
Tahun Kurikulum: 2023



Penyusun:
Bayu Nugroho, S.Kom., M.Eng



FAKULTAS EKONOMI & BISNIS
INSTITUT INFORMATIKA DAN BISNIS DARMAJAYA
2025

Modul 2

Basic syntax & structure in python programming

1. Aturan penulisan kode dalam Python (indentasi, komentar)

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Python Indentation

Indentation refers to the spaces at the beginning of a code line.

Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important.

Python uses indentation to indicate a block of code.

Example:

```
if 5 > 2:  
    print("Five is greater than two!")
```

Python Comments

Python has commenting capability for the purpose of in-code documentation.

Comments start with a #, and Python will render the rest of the line as a comment:

Example

Comments in Python

```
#This is a comment.  
print("Hello, World!")
```

Multiline Comments

Python does not really have a syntax for multiline comments.

To add a multiline comment you could insert a # for each line:

Or, not quite as intended, you can use a multiline string.

Since Python will ignore string literals that are not assigned to a variable, you can add a multiline string (triple quotes) in your code, and place your comment inside it:

Example

```
"""
This is a comment
written in
more than just one line
"""
print("Hello, World!")
```

2. Penggunaan variabel dan pengaturan nama variabel yang Benar

Python Variables

In Python, variables are created when you assign a value to it:

Example

Variables in Python:

```
x = 5
y = "Hello, World!"
```

3. Penggunaan fungsi dasar print(), input() dan variabel

Python print() Function

Definition and Usage

The print() function prints the specified message to the screen, or other standard output device.

The message can be a string, or any other object, the object will be converted into a string before written to the screen.

Example:

Print a message onto the screen:

```
print("Hello World")
```

Syntax

```
print(object(s), sep=separator, end=end, file=file, flush=flush)
```

Parameter Values

Parameter	Description
<i>object(s)</i>	Any object, and as many as you like. Will be converted to string before printed
<i>sep='separator'</i>	Optional. Specify how to separate the objects, if there is more than one. Default is ' '
<i>end='end'</i>	Optional. Specify what to print at the end. Default is '\n' (line feed)
<i>file</i>	Optional. An object with a write method. Default is sys.stdout
<i>flush</i>	Optional. A Boolean, specifying if the output is flushed (True) or buffered (False). Default is False

Print more than one object:

```
print("Hello", "how are you?")
```

Example

Print a tuple:

```
print('Enter your name:')  
x = input()  
print('Hello, ' + x)
```

Python input() Function

The `input()` function allows user input.

Example

Ask for the user's name and print it:

```
x = ("apple", "banana", "cherry")  
print(x)
```

Syntax

```
input(prompt)
```

Parameter Values

Parameter	Description
<i>prompt</i>	A String, representing a default message before the input.

More Examples

Use the prompt parameter to write a message before the input:

```
x = input('Enter your name:')  
print('Hello, ' + x)
```

4. Mode eksekusi: Script dan interactive mode

Python syntax can be executed by writing directly in the Command Line:

```
>>> print("Hello, World!")  
Hello, World!
```

Or by creating a python file on the server, using the .py file extension, and running it in the Command Line:

```
C:\Users\Your Name>python myfile.py
```

5. Penulisan komentar dalam kode dan pentingnya dokumentasi kode

One line comment in python

```
#This is a comment.  
print("Hello, World!")
```

Multiline comment in python

```
"""  
This is a comment  
written in  
more than just one line  
"""  
print("Hello, World!")
```

The Importance of Documentation in Programming

Documentation plays a paramount role in the field of programming, serving as a vital means of communication and knowledge preservation (medium.com). It encompasses a variety of written materials that assist programmers in understanding, maintaining, and collaborating on software projects. In this article, we will explore the significance of documentation in programming, highlighting its benefits and the various types that exist. Additionally, we will delve into best practices for creating effective documentation, discuss the tools and resources available to programmers, address common challenges, and examine real-world case studies to showcase the profound impact that documentation can have. Finally, we will conclude by pondering the future outlook for documentation in programming and its continuous role in fostering efficient and sustainable software development processes.

What is Documentation in Programming?

Documentation in programming refers to the process of creating written or visual materials that explain and describe the code, its functionality, and how to use it. It serves as a guide for developers, users, and other stakeholders to understand and interact with the software effectively.

Why is Documentation Important in Programming?

Documentation is essential in programming for a variety of reasons. Firstly, it improves the overall understandability and maintainability of the code. It helps developers quickly grasp the purpose and functionality of different parts of the codebase, making it easier to make changes or fix bugs.

Secondly, documentation facilitates collaboration and knowledge sharing among team members. It allows developers to communicate their intentions, decisions, and best practices to others, promoting consistency and reducing confusion.

Finally, documentation streamlines the onboarding process for new developers joining a project. It provides a comprehensive resource for them to familiarize themselves with the code, reducing the time and effort required to get up to speed.