

**MAKALAH**  
**PARADIGMA PEMROGRAMAN PYTHON SEBAGAI BAHASA PEMROGRAMAN**

Disusun Guna Memenuhi Tugas Mata Kuliah Pemrograman Terstruktur

Dosen Pengampu:

**Muhammad Hasbi, S.T., M.T.**



**OLEH:**

**FADHIL RAHMAN HAKIM (3202316018)**

PROGRAM STUDI TEKNIK INFORMATIKA JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI PONTIANAK

PONTIANAK

2023

## **KATA PENGANTAR**

Puji syukur Saya panjatkan kepada Tuhan Yang Maha Esa, karena atas limpahan rahmatnya penyusun dapat menyelesaikan makalah ini tepat waktu tanpa ada halangan yang berarti dan sesuai dengan harapan.

Ucapan terima kasih Saya sampaikan kepada bapak ... Muhammad Hasbi, sebagai dosen pengampu mata kuliah ... Pemrograman Terstruktur yang telah membantu memberikan arahan dan pemahaman dalam penyusunan makalah ini.

Saya menyadari bahwa dalam penyusunan makalah ini masih banyak kekurangan karena keterbatasan Saya. Maka dari itu penyusun sangat mengharapkan kritik dan saran untuk menyempurnakan makalah ini. Semoga apa yang ditulis dapat bermanfaat bagi semua pihak yang membutuhkan.

Pontianak, 11 Oktober 2023

# DAFTAR ISI

<b>DAFTAR ISI .....</b>	<b>3</b>
<b>BAB I .....</b>	<b>1</b>
1.1 Latar Belakang: .....	1
1.2 Struktur Makalah: .....	1
1.3 Tujuan: .....	1
<b>BAB II .....</b>	<b>2</b>
2.1 Pemrograman Fungsional .....	2
2.2 Pemrograman Berorientasi Objek (Object-Oriented Programming - OOP) .....	2
2.3 Pemrograman Deklaratif .....	2
2.4 Pemrograman Imperatif .....	2
<b>BAB III .....</b>	<b>3</b>
3.1 Metode Pemrograman Berorientasi Objek (OOP) .....	3
3.1.1 Definisi Kelas: .....	3
3.1.2 Pembuatan Objek: .....	3
3.1.3 Pewarisan (Inheritance): .....	3
3.1.4 Polimorfisme: Polimorfisme memungkinkan objek untuk berperilaku berbeda sesuai dengan kelas yang mereka miliki. ....	3
3.2 Contoh Metode penulisan program dalam Python .....	4
3.3 Penanganan Kasus Khusus .....	4
3.3.1 Metode Tanpa Kembalian: .....	4
3.3.2 Metode dengan Argumen Opsional: .....	4
3.3.3 Rekursi: .....	4
<b>BAB IV .....</b>	<b>5</b>
4.1 Paradigma Pemrograman dalam Python: .....	5
4.2 Metode: .....	5
<b>DAFTAR PUSTAKA .....</b>	<b>6</b>

# **BAB I**

## **Pendahuluan**

Bahasa pemrograman Python telah meraih popularitas luar biasa dan menjadi salah satu yang paling sering digunakan di seluruh dunia. Pertama kali dirilis pada tahun 1991 oleh Guido van Rossum, Python telah menjadi pilihan utama bagi pengembang perangkat lunak, ilmuwan data, peneliti kecerdasan buatan, dan berbagai profesional di berbagai bidang teknologi. Popularitasnya terletak pada sintaksis yang mudah dipahami, fleksibilitasnya, serta kemampuannya untuk mendukung beragam paradigma pemrograman.

### **1.1 Latar Belakang:**

Python dirancang sebagai bahasa pemrograman umum yang mudah digunakan dan dipahami. Python terus berkembang dan ditambahkan fitur baru untuk memenuhi kebutuhan pengembangan perangkat lunak modern. Python menjadi bahasa yang populer di berbagai bidang, seperti pengembangan web, pemrosesan data, kecerdasan buatan, pemodelan matematika, dan pengembangan perangkat lunak.

### **1.2 Struktur Makalah:**

Makalah ini akan menjelaskan empat paradigma pemrograman utama yang digunakan dalam Python. Keempat paradigma tersebut adalah pemrograman berorientasi objek, pemrograman fungsional, pemrograman imperatif, dan pemrograman deklaratif. makalah akan membahas metode dan penanganan kasus khusus pada sebuah paradigma, serta bagaimana paradigma-paradigma ini bisa diterapkan dalam pengembangan software..

### **1.3 Tujuan:**

Tujuan makalah ini adalah untuk menjelaskan paradigma pemrograman yang digunakan dalam Python secara komprehensif. Memahami paradigma pemrograman Python adalah kunci untuk menguasai bahasa python secara menyeluruh dan mampu menggunakan python dengan efektif dalam aspek pengembangan program.

## **BAB II**

### **Paradigma Pemrograman dalam Python**

Python, sebagai bahasa pemrograman serbaguna, menggabungkan berbagai paradigma pemrograman yang memungkinkan pengembang untuk menggunakan pendekatan yang sesuai dengan tugas yang dihadapi. Paradigma pemrograman adalah cara berpikir dan metode pengorganisasian kode untuk mencapai tujuan tertentu dalam pengembangan perangkat lunak. Dalam Python, terdapat beberapa paradigma pemrograman utama yang akan kita bahas: pemrograman berorientasi objek, pemrograman fungsional, pemrograman imperatif, dan pemrograman deklaratif.

#### **2.1 Pemrograman Fungsional**

Python juga mendukung paradigma pemrograman fungsional. Ini berarti Anda dapat menganggap fungsi sebagai objek, mengirimkannya sebagai argumen, dan bahkan menyimpannya dalam struktur data. Python memiliki fungsi lambda, dan Anda dapat menggunakan fungsi seperti `map()`, `filter()`, dan `reduce()` untuk melakukan operasi pada data.

#### **2.2 Pemrograman Berorientasi Objek (Object-Oriented Programming - OOP)**

Python mendukung pemrograman berorientasi objek dengan baik. Dalam OOP, kode dikelompokkan menjadi objek yang memiliki atribut dan metode. Python memungkinkan definisi kelas dan pembuatan objek dengan mudah, serta pewarisan dan polimorfisme.

#### **2.3 Pemrograman Deklaratif**

Python juga mendukung paradigma pemrograman deklaratif. Ini terlihat dalam bahasa yang mudah dibaca dan ekspresif. Anda dapat menyatakan apa yang harus dilakukan, bukan bagaimana melakukannya. Contoh nyata adalah dalam SQL, yang digunakan untuk mengakses database.

#### **2.4 Pemrograman Imperatif**

Pemrograman imperatif adalah pendekatan yang digunakan dalam pengembangan perangkat lunak konvensional. Python mendukung struktur pengendalian seperti perulangan dan percabangan, yang memungkinkan penggunaan paradigma ini untuk menulis kode prosedural.

## **Kesimpulan**

Python adalah bahasa pemrograman yang sangat serbaguna karena mendukung berbagai paradigma pemrograman. Ini memungkinkan programmer untuk memilih pendekatan yang paling sesuai dengan tugas yang sedang mereka kerjakan. Memahami paradigma pemrograman yang mendasari Python adalah kunci untuk menjadi seorang pengembang Python yang lebih baik dan efisien dalam pengembangan perangkat lunak.

## **BAB III**

### **Metode**

Metode merujuk pada sebagian dari program komputer yang terdiri dari serangkaian perintah atau instruksi yang dirancang untuk mengeksekusi tugas atau operasi tertentu. Dalam paradigma pemrograman, istilah "metode" seringkali juga disebut sebagai "fungsi" atau "prosedur." Penggunaan metode memungkinkan kita untuk mengelompokkan kode, melakukan penggunaan kembali kode, dan menciptakan abstraksi, yang semuanya berperan penting dalam merancang dan mengorganisir program secara lebih efisien.

### **3.1 Metode Pemrograman Berorientasi Objek (OOP)**

Pemrograman berorientasi objek (OOP) adalah paradigma pemrograman yang menganggap masalah sebagai kumpulan objek yang berinteraksi satu sama lain. Python mendukung OOP dengan baik, dan dalam konteks ini, metode-metode penting yang terkait adalah:

**3.1.1 Definisi Kelas:** Anda dapat membuat kelas yang mewakili objek dan mendefinisikan atribut dan metode mereka.

**3.1.2 Pembuatan Objek:** Dengan kelas, Anda dapat membuat objek sesuai dengan desain kelas tersebut.

**3.1.3 Pewarisan (Inheritance):** Python mendukung pewarisan, yang memungkinkan Anda mengambil atribut dan metode dari kelas yang sudah ada ke dalam kelas baru.

**3.1.4 Polimorfisme:** Polimorfisme memungkinkan objek untuk berperilaku berbeda sesuai dengan kelas yang mereka miliki.

## 3.2 Contoh Metode penulisan program dalam Python

```
# Definisikan sebuah kelas sederhana
class Hitung:
    # Metode untuk menambahkan dua angka
    def tambah(self, angka1, angka2):
        hasil = angka1 + angka2
        return hasil

# Membuat objek dari kelas Hitung
kalkulator = Hitung()

# Memanggil metode "tambah" untuk menjumlahkan dua angka
hasil_penjumlahan = kalkulator.tambah(5, 3)

# Menampilkan hasil penjumlahan
print("Hasil penjumlahan 5 + 3 adalah:", hasil_penjumlahan)
```

Dalam contoh ini, kita mendefinisikan sebuah kelas sederhana yang disebut Hitung dengan satu metode, yaitu tambah, yang digunakan untuk menambahkan dua angka. Kemudian, kita membuat objek kalkulator dari kelas Hitung, memanggil metode tambah untuk menjumlahkan dua angka (dalam contoh ini, 5 dan 3), dan menampilkan hasil penjumlahannya.

## 3.3 Penanganan Kasus Khusus

**3.3.1 Metode Tanpa Kembalian:** Beberapa metode hanya melakukan tindakan tanpa mengembalikan nilai apapun, seperti mencetak keluaran ke layar atau mengubah keadaan internal objek.

**3.3.2 Metode dengan Argumen Opsional:** Metode dalam Python dapat memiliki argumen opsional yang memungkinkan pemanggilan metode dengan atau tanpa argumen tambahan.

**3.3.3 Rekursi:** Rekursi adalah konsep di mana metode memanggil dirinya sendiri untuk menyelesaikan tugas tertentu. Ini berguna untuk pemecahan masalah yang dapat dibagi menjadi submasalah serupa

## **BAB IV**

### **Kesimpulan**

Makalah ini mengulas paradigma pemrograman yang menjadi dasar bahasa pemrograman Python, serta menitikberatkan pada konsep metode dan hasil dalam konteks pengembangan perangkat lunak. Pada kesimpulan ini, kami akan menggambarkan secara singkat poin-poin penting yang telah disajikan dalam makalah ini.

#### **4.1 Paradigma Pemrograman dalam Python:**

Python adalah bahasa pemrograman yang mendukung beragam paradigma pemrograman, termasuk pemrograman berorientasi objek, fungsional, imperatif, dan deklaratif. Keberagaman paradigma ini menjadikan Python sebagai bahasa yang sangat serbaguna dan relevan dalam berbagai konteks pengembangan perangkat lunak.

Python mendukung berbagai paradigma pemrograman, termasuk pemrograman berorientasi objek, fungsional, imperatif, dan deklaratif. Pemahaman ini memungkinkan pengembang untuk memilih pendekatan yang sesuai dengan tujuan dan kompleksitas proyek.

#### **4.2 Metode:**

Metode adalah unit dalam program yang terdiri dari serangkaian perintah atau instruksi yang didesain untuk menjalankan tugas spesifik.

Penggunaan metode dalam pemrograman memberikan manfaat seperti pengelompokan logika, kemampuan penggunaan ulang kode, dan pencapaian tingkat abstraksi yang lebih baik dalam proses pengembangan perangkat lunak.

Dengan pemahaman yang kuat mengenai paradigma pemrograman, penerapan metode yang efisien, dan manajemen yang sesuai, developer dapat membuat pemrograman menjadi lebih efektif, mudah, dan sesuai dengan kebutuhan. Python, sebagai bahasa pemrograman yang mendukung beragam paradigma, akan memberikan fleksibilitas kepada para developer.



## DAFTAR PUSTAKA

Python.org. (n.d.). About Python. <https://www.python.org/about/>.

Lutz, M. (2013). Learning Python (5th ed.). O'Reilly Media.

Python Software Foundation. (2021). The Python Language Reference.

Downey, A. B. (2015). Think Python: How to Think Like a Computer Scientist. O'Reilly Media.

Van Rossum, G. (1995). Python Reference Manual. Python Software Foundation.