

# "Teknik Coding Inovatif: Memanfaatkan Higher Order Functions untuk Aplikasi yang Skalabel dan Mudah Dikelola"

Natasya Ega Lina Marbun<sup>1</sup>, Khusnun Nisa<sup>2</sup>, Presilia<sup>3</sup>, Irvan Alfaritzi<sup>4</sup>, Syalaisha Andini Putriansyah<sup>5</sup>  
Program Studi Sains Data, Fakultas Sains, Institut Teknologi Sumatera.

Email: [natasya.122450024@student.itera.ac.id](mailto:natasya.122450024@student.itera.ac.id), [khusnun.122450078@student.itera.ac.id](mailto:khusnun.122450078@student.itera.ac.id),  
[presilia.122450081@student.itera.ac.id](mailto:presilia.122450081@student.itera.ac.id), [irvan.122450093@student.itera.ac.id](mailto:irvan.122450093@student.itera.ac.id), [syalaisha.122450111@student.itera.ac.id](mailto:syalaisha.122450111@student.itera.ac.id)

## 1. Pendahuluan

Di era digital ini, kebutuhan akan aplikasi yang skalabel dan mudah dikelola semakin meningkat, aplikasi yang dapat dengan mudah diubah ukurannya untuk memenuhi permintaan yang berubah dan juga mudah dipelihara oleh tim pengembangan. Aplikasi seperti ini penting untuk bisnis di era digital saat ini, karena memungkinkan mereka untuk beradaptasi dengan cepat terhadap perubahan pasar dan kebutuhan pelanggan.

Hal ini terjadi karena didorong oleh pertumbuhan data dan pengguna yang cukup signifikan, serta tuntutan akan fleksibilitas dan kelincahan dalam pengembangan aplikasi. Salah satu teknik coding yang dapat menjawab tantangan ini adalah *Higher-Order Functions*. Metode ini merupakan fungsi orde tinggi, yang mampu untuk menerima fungsi lain sebagai argumen atau mengembalikan fungsi lain sebagai hasil. Kemampuan ini membuka berbagai peluang untuk menulis kode yang lebih ringkas, fleksibel, dan mudah diuji, sehingga berkontribusi pada pengembangan aplikasi yang skalabel dan mudah dikelola.

Penerapan *High Order Function* seperti pemfilteran, pemetaan, pengurangan, penggabungan, dan lainnya sangat bermanfaat dalam pengembangan aplikasi skalabel dan mudah dikelola. Kemampuan HOF untuk menerima dan menghasilkan fungsi lain memungkinkan pengembang untuk menulis kode yang lebih ringkas, fleksibel, dan mudah diuji, sehingga berkontribusi pada pengembangan aplikasi yang lebih efisien dan berkelanjutan. Dengan memahami konsep dan manfaat HOF, para pengembang dapat memanfaatkan kekuatannya untuk membangun aplikasi yang memenuhi tuntutan era digital yang dinamis ini.

## 2. Metode

### 2.1. Map

Map termasuk kedalam fungsi bawaan pada pemrograman Python yang digunakan untuk menerapkan fungsi tertentu ke setiap elemen dari sebuah urutan dan mengembalikan hasilnya dalam bentuk iterator. Pada pengolahan data di Pandas, fungsi map dapat digunakan pada kolom tertentu karena map hanya bekerja pada series atau array satu dimensi.

### 2.2. Filter

Digunakan untuk memfilter elemen-elemen dalam courses berdasarkan fungsi `filtered_func` (2021). Dalam program ini, `filter()` digunakan untuk memfilter kursus-kursus berdasarkan durasinya, yaitu kursus-kursus yang memiliki durasi 10 jam atau lebih.

### 2.3. Lambda functions

Digunakan dalam ekspresi lambda di map dan sebagai argumen untuk `filtered_func`. Fungsi lambda digunakan dalam program ini untuk membuat fungsi anonim yang mengevaluasi apakah

durasi suatu kursus lebih besar atau sama dengan 10 jam. Fungsi lambda ini kemudian digunakan sebagai argumen untuk fungsi `filter_by`.

## 2.4. Nested Functions

Fungsi-fungsi `multiplier` dan `filter_func` didefinisikan di dalam fungsi-fungsi lain (`multiply_by` dan `filter_by`), sehingga bersifat nested atau bersarang.

## 2.5. Zip

Digunakan untuk menggabungkan dua iterable menjadi satu, dalam kasus ini digunakan untuk menggabungkan dua iterator, yaitu `courses` dan `discounted_prices`, agar dapat diakses bersamaan dalam loop `for`. Ini memungkinkan kita untuk mencetak informasi kursus beserta harga diskonnya.

## 2.6 Lambda course : course.duration >= 10

Digunakan sebagai kriteria untuk melakukan filter pada kursus-kursus berdasarkan durasi yang ditentukan.

# 3. Pembahasan

## 3.1. Kode Program

### 3.1.1. HOF untuk manipulasi data

```
# Definisikan Higher Order Functions yang akan digunakan
def multiply_by(factor):
    def multiplier(number):
        return number * factor
    return multiplier

def filter_by(criteria):
    def filter_func(item):
        return criteria(item)
    return filter_func
```

Gambar 1. Pembuatan HOF untuk Manipulasi Data.

Pada gambar 1 menjelaskan tentang fungsi tingkat tinggi (*Higher-order function*) bernama `'multiply_by'`. Fungsi ini menerima satu parameter `'factor'` yang merupakan angka dengan mana nilai yang diterima oleh fungsi hasilnya akan dikalikan. Fungsi `'multiply_by'` akan mengembalikan fungsi dalam fungsi (`'multiplier'`) yang kemudian dapat digunakan untuk mengalikan nilai dengan faktor tertentu. Selanjutnya `'filter_by'`, mengambil satu parameter `'criteria'` yang seharusnya adalah suatu fungsi yang digunakan untuk menentukan apakah suatu item memenuhi kriteria tertentu atau tidak. Fungsi `'filter_by'` mengembalikan fungsi dalam fungsi (`'filter_func'`). Fungsi dalam fungsi tersebut menerima satu parameter `'item'` dan mengembalikan hasil dari pemanggilan `'criteria(item)'`. Yang dimana memungkinkan untuk membuat fungsi filter yang dapat menentukan apakah suatu item harus disertakan dalam hasil filtrasi berdasarkan kriteria yang ditentukan.

### 3.1.2. Pembuatan aplikasi pembelajaran

```
# Buat aplikasi pembelajaran yang skalabel dan mudah dikelola
class Course:
    def __init__(self, name, price, duration):
        self.name = name
        self.price = price
        self.duration = duration

    def __str__(self):
        return f"{self.name} ({self.duration} jam) - Rp {self.price}"

courses = [
    Course("Python Programming", 200000, 8),
    Course("Web Development", 300000, 10),
    Course("Machine Learning", 500000, 12),
    Course("Data Science", 400000, 10),
]
```

Gambar 2. Aplikasi pembelajaran yang skalabel dan mudah dikelola

Pada gambar 2 menjelaskan kelas *'Course'* yang merepresentasikan kursus dalam sebuah aplikasi pembelajaran. Setiap objek *'Course'* memiliki atribut nama (*'name'*), harga (*'price'*), dan durasi (*'duration'*). Metode *'\_\_str\_\_'* digunakan untuk menghasilkan representasi string yang menyajikan informasi tentang kursus dengan format yang mudah dibaca. Selain itu, kode juga membuat daftar (*'courses'*) yang berisi beberapa objek *'Course'* yang mewakili kursus-kursus yang tersedia dalam aplikasi pembelajaran ini. Dengan ini landasan dari aplikasi pembelajaran yang dapat mengelola informasi tentang kursus-kursus yang ditawarkan. Aplikasi ini dapat diubah dan disesuaikan dengan menambahkan fungsionalitas tambahan seperti pendaftaran kursus, pencarian kursus, dan lain sebagainya.

### 3.1.3. Pendefinisial Objek

```
# Definisikan Higher Order Functions yang akan digunakan
multiplier_func = multiply_by(0.9) # Diskon 10%
filtered_func = filter_by(lambda course: course.duration >= 10) # Filter kursus dengan durasi 10 jam atau lebih

# Menggunakan Higher Order Functions untuk manipulasi data kursus
discounted_prices = map(multiplier_func, map(lambda course: course.price, courses))
filtered_courses = filter(filtered_func, courses)
```

Gambar 3. Pendefinisian objek HOF

Pada Gambar 3. terdapat pendefinisian 4 objek, yaitu objek *multiplier\_func*, *filtered\_func*, *discounted\_prices* dan *filtered\_courses*. *Multiplier\_func* mendefinisikan parameter yang akan digunakan dalam fungsi *"multiply\_by"* dimana fungsi akan melakukan perhitungan potongan harga/diskon. Objek *filtered\_func* menggunakan fungsi *filter\_by* untuk memfilter variabel *course* yang memiliki nilai lebih dari 10. Fungsi *map* pada saat mendeklarasikan objek *discounted prices* digunakan untuk melakukan operasi dari objek *multiplier\_func* dengan data yang berasal dari iterasi pada list *course*. Fungsi *filter* digunakan untuk memfilter list pada *courses* dengan melakukan operasi yang terdapat pada objek *filtered\_func* ke setiap item dalam list *courses*.

### 3.1.4. Penggunaan Aplikasi

```
# Menampilkan hasil
print("Kursus yang telah didiskon:")
for course, discounted_price in zip(courses, discounted_prices):
    course.price = discounted_price
    print(course)

print("\nKursus dengan durasi 10 jam atau lebih:")
for course in filtered_courses:
    print(course)
```

Gambar 4. Penggunaan Aplikasi HOF

Penerapan high order function dapat diterapkan menggunakan kode pada Gambar 4. Kode tersebut melakukan 2 iterasi, iterasi pertama akan mencetak kursus yang telah diberi diskon dan iterasi kedua akan menampilkan nama kursus yang memiliki durasi kursus lebih dari 10jam. Pada iterasi pertama terdapat fungsi `zip()` untuk menggabungkan list course dengan harga diskon.

#### **4. Kesimpulan**

Dengan demikian, dapat disimpulkan bahwa teknik coding inovatif menggunakan Higher Order Functions (HOF) merupakan solusi yang sangat relevan dalam pengembangan aplikasi skalabel dan mudah dikelola di era digital saat ini. Melalui penerapan HOF, pengembang dapat menghasilkan kode yang lebih efisien dan fleksibel, memungkinkan aplikasi untuk beradaptasi dengan cepat terhadap perubahan lingkungan bisnis dan kebutuhan pengguna. Selain itu, pemahaman mendalam tentang konsep HOF memungkinkan pengembang untuk memanfaatkan potensi penuhnya dalam membangun solusi yang inovatif dan efektif.

Secara keseluruhan, penerapan HOF tidak hanya membantu meningkatkan produktivitas pengembangan aplikasi, tetapi juga memberikan fondasi yang kuat untuk menciptakan solusi yang berkualitas tinggi dan siap menghadapi tantangan di era digital yang dinamis. Dengan demikian, para pengembang perangkat lunak dapat memanfaatkan keunggulan HOF untuk menciptakan aplikasi yang tangguh, mudah dielola, dan dapat bersaing dalam pasar yang terus berubah.

#### **5. Daftar Pustaka**

Flanagan (2018). *Python Pocket Reference: Python in Your Pocket* (5th ed.). O'Reilly Media.

Anggara (2021). Pengembangan sistem informasi manajemen persediaan barang dagang berbasis web menggunakan library XSS filtering. In *Seri Prosiding Seminar Nasional Dinamika Informatika* (Vol. 5, No. 1).

Baihaqi (2021). *Pemrograman Berbasis Event dengan PyQt5* (Vol. 1). Zahira Media Publisher.

Yuniarti, W. D. (2019). *Dasar-dasar pemrograman dengan python*. Deepublish.

Herho, S. H. S. (2017). *Tutorial Pemrograman Python 2 Untuk Pemula*.

Hutton, G. (1992). Higher-order functions for parsing. *Journal of functional programming*, 2(3), 323-343.