

# **MAKALAH TUGAS PEMOGRAMAN FUNGSIONAL**

Makalah ini disusun untuk memenuhi Tugas Akhir Mata kuliah Pemograman Fungsional

Dosen Pengampu : Wildan Suharso, S.Kom., M.Kom



**DISUSUN OLEH :**

Kelompok 8

Faishal Akbar Syahputra      202110370311018

M Daffa Raihan Syalwa      202110370311022

Rayhan Arve SMFPF      202110370311025

**UNIVERSITAS MUHAMMADIYAH MALANG**

**MALANG**

**2023**

## **BAB 1**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Toko Sembako adalah hal yang sering kita temui dalam kehidupan masyarakat ini. Kita dapat melihat banyak sekali toko sembako saat pergi ke pasar. Toko sembako menyediakan berbagai macam barang kebutuhan sehari-hari kita. Toko sembako tentunya memerlukan pendataan barang-barang yang akan dijual.

Program ini merupakan implementasi sederhana dari sebuah sistem manajemen barang untuk toko sembako. Tujuan utama dari program ini adalah membantu pemilik toko dalam mengelola daftar barang, melakukan operasi perhitungan, dan menyajikan informasi secara visual melalui diagram batang.

#### **1.2 Tujuan**

Tujuan dari Program ini adalah :

1. Program ini memungkinkan pengguna untuk menambahkan barang baru ke dalam daftar. Data yang dimasukkan melibatkan nama barang, jenis satuan (KG/Liter), harga per satuan, dan jumlah stok barang.
2. Program ini dirancang untuk memberikan pemahaman yang baik tentang cara kerja fungsi-fungsi fungsional, ekspresi lambda, high-order functions, filter, dan konsep-konsep lain dalam pemrograman Python.
3. Program menyajikan informasi yang berguna, seperti total penjualan, total pendapatan, barang mahal, dan diagram batang harga barang.

#### **1.3 Data dan Gambaran Program**

Program ini merupakan penerapan program kasir yang memiliki fokus utama pada input penjualan barang. Di mana memiliki fitur utama dapat melihat barang apa yang

terjual, harga barang, dan mem-filter barang yang mahal. Barang yang dikategorikan mahal dapat dilihat dari harga barang tersebut, apabila harga satuan barangnya lebih dari Rp5.000, maka otomatis akan dikategorikan sebagai barang mahal. Semua fitur dalam program ini, mengimplementasikan konsep fungsional python, perulangan, pure functions, lambda expressions, list comprehension, generator, higher-order functions, filter, reduce, inner functions, serta library matplotlib untuk dapat menampilkan grafik.

```
Masukkan nama barang (ketik 'selesai' untuk mengakhiri): gula
Masukkan satuan barang (KG/Liter): kg
Masukkan harga barang per KG/Liter: 10000
Masukkan jumlah barang: 2
Ingin melanjutkan input barang? (yes/no): yes
Masukkan nama barang (ketik 'selesai' untuk mengakhiri): garam
Masukkan satuan barang (KG/Liter): kg
Masukkan harga barang per KG/Liter: 6000
Masukkan jumlah barang: 1
Ingin melanjutkan input barang? (yes/no): no

--- OUTPUT ---

Semua Barang:
Nama: jajan, Stok: 3.0 kg, Harga: 20,000
Nama: Beras, Stok: 5.0 kg, Harga: 30,000
Nama: gula, Stok: 2.0 kg, Harga: 10,000
Nama: garam, Stok: 1.0 kg, Harga: 6,000

Total Penjualan: 236,000
Total Pendapatan: 236,000
Barang Mahal: ['jajan', 'Beras', 'gula', 'garam']
Semua Nama Barang (Generator): ['jajan', 'Beras', 'gula', 'garam']

Barang Mahal (Filter):
Nama: jajan, Stok: 3.0 kg, Harga: 20,000
Nama: Beras, Stok: 5.0 kg, Harga: 30,000

Total Stok Setelah Penambahan Stok:
11.0
```

#### 1.4 Algoritma Program

Program ini merupakan implementasi sederhana dari sistem manajemen barang untuk toko sembako. Berikut adalah penjelasan mengenai algoritma program tersebut:

## **Fungsi-fungsi Utama**

### **1. Tambah Barang :**

Meminta pengguna untuk memasukkan informasi barang seperti nama, jenis satuan (KG/Liter), dan harga per unit. Kemudian mengkonversi jumlah barang ke dalam format numerik dan menyesuaikan stok dan harga berdasarkan jenis barang, dan menambahkan data barang baru ke dalam daftar “barang\_list” dan mengembalikan daftar tersebut.

### **2. Tampilkan Semua Barang :**

Menampilkan semua barang dalam daftar, termasuk nama, stok, jenis satuan, dan harga.

### **3. Hitung Total Penjualan :**

Menggunakan fungsi “sum” dan list comprehension untuk menghitung total penjualan dengan menjumlahkan hasil perkalian stok dan harga untuk setiap barang.

### **4. Hitung Total Pendapatan :**

Menggunakan lambda expression dan fungsi `sum` untuk menghitung total pendapatan dengan cara yang serupa seperti total penjualan.

### **5. Lambda Barang Mahal :**

Menggunakan lambda expression untuk membuat fungsi lambda yang mengembalikan daftar barang yang memiliki harga di atas suatu ambang batas tertentu (*threshold*).

### **6. Generate Semua Nama Barang :**

Menggunakan generator untuk menghasilkan semua nama barang dalam bentuk iterator.

### **7. Apply Operasi pada Semua Barang :**

Menggunakan higher-order function untuk menerapkan suatu operasi pada semua barang dalam daftar.

### **8. Filter Barang Mahal :**

Menggunakan fungsi `filter` untuk menyaring barang-barang yang dianggap "mahal" berdasarkan suatu kriteria.

## **9. Hitung Total Stok :**

Menggunakan lambda expression, fungsi “reduce”, dan fungsi untuk menghitung total stok dari semua barang dalam daftar.

## **10. Membuat Fungsi Tambah Stok:**

Menggunakan inner function sebagai bagian dari higher-order function untuk membuat fungsi tambah stok.

## **11. Tampilkan Diagram Batang :**

Menggunakan library Matplotlib untuk membuat diagram batang dari harga barang.

# **Penggunaan Program**

## **1. Input Barang:**

Program meminta pengguna untuk memasukkan informasi barang secara interaktif, dengan opsi untuk menghentikan input dengan mengetikkan 'selesai'.

## **2. Menampilkan Informasi:**

Program menampilkan semua barang, total penjualan, total pendapatan, barang mahal, dan semua nama barang menggunakan fungsi-fungsi yang telah dibuat.

## **3. Manipulasi Data Barang:**

Pengguna dapat menambahkan stok, melakukan filter terhadap barang mahal, dan melihat total stok setelah penambahan stok.

## **4. Visualisasi Data:**

Program menyajikan diagram batang harga barang menggunakan Matplotlib.

Program ini dirancang untuk memberikan pemahaman yang baik tentang konsep-konsep fungsional, ekspresi lambda, higher-order functions, filter, dan visualisasi data menggunakan Python.

## BAB 2

## FITUR

### 2.1 Pure Function

Merupakan fungsi dalam bahasa python yang tidak boleh mengubah, dan tidak boleh terikat dengan variabel lain yang berada diluar fungsi. Pada program ini fungsi yang menerapkan pure function terdapat pada fungsi “semua\_barang” yang mana isinya hanya bergantung sesuai dengan apa yang di inisiasikan, dan selalu ada nilainya meskipun ditambahkan nilai lain.

```
1  semua_barang = [  
2      ("jajan", "kg", 3, 20000),  
3      ("Beras", "kg", 5, 30000)  
4      ]
```

```
Masukkan nama barang (ketik 'selesai' untuk mengakhiri): gula  
Masukkan satuan barang (KG/Liter): kg  
Masukkan harga barang per KG/Liter: 10000  
Masukkan jumlah barang: 2  
Ingin melanjutkan input barang? (yes/no): yes  
Masukkan nama barang (ketik 'selesai' untuk mengakhiri): garam  
Masukkan satuan barang (KG/Liter): kg  
Masukkan harga barang per KG/Liter: 6000  
Masukkan jumlah barang: 1  
Ingin melanjutkan input barang? (yes/no): no
```

--- OUTPUT ---

```
Semua Barang:  
Nama: jajan, Stok: 3.0 kg, Harga: 20,000  
Nama: Beras, Stok: 5.0 kg, Harga: 30,000  
Nama: gula, Stok: 2.0 kg, Harga: 10,000  
Nama: garam, Stok: 1.0 kg, Harga: 6,000
```

## 2.2 Lambda dan List

Lambda merupakan fungsi dalam python yang bersifat anon, sehingga tampak lebih sederhana, tetapi tidak menghilangkan fungsi utamanya. Sedangkan list comprehension merupakan fungsi pada python yang juga dapat meringkas penulisan kode, yang digunakan untuk membuat daftar data.

```
1 # Lambda Expression (Modul2) & Reduce (Modul3) Menghitung Total Pendapatan
2 hitung_pendapatan = lambda barang_list: '{:,.0f}'.format(sum([barang[2] * barang[3] for barang in barang_list]))
3
4 # Lambda (Modul2) & Menampilkan Barang Mahal
5 barang_mahal = lambda barang_list, threshold: [barang[0] for barang in barang_list if barang[3] > threshold]
```

```
1 def hitung_total_stok(barang_list):
2     return reduce(lambda x, y: x + y[2], barang_list, 0)
```

## 2.3 Generator

Merupakan fungsi khusus dalam python yang tidak mengembalikan nilai tunggal. Dalam program ini, generator mengembalikan barang yang berada dalam list barang. Sehingga output menampilkan seluruh nama-nama barang.

```
1 def generate_semua_nama_barang(barang_list):
2     for barang in barang_list:
3         yield barang[0]
```

```
Semua Nama Barang (Generator): ['jajan', 'Beras', 'gula', 'garam']
```

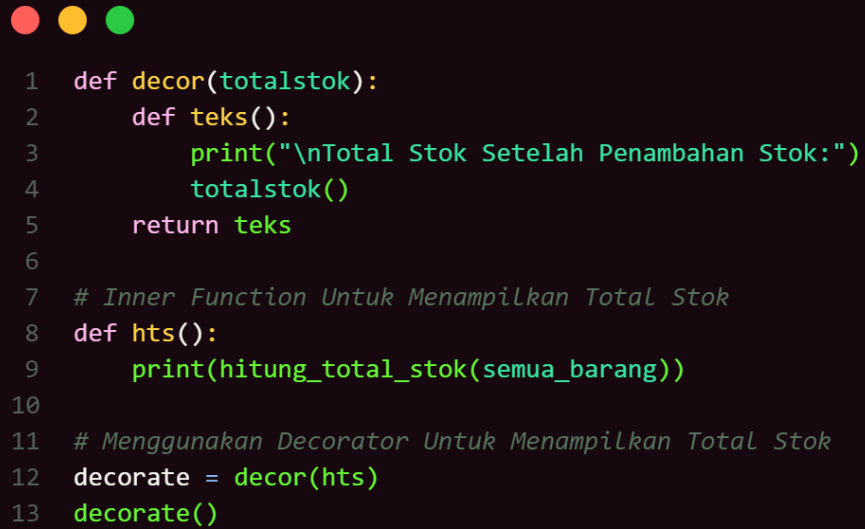
## 2.4 Inner Function

Inner function atau bisa disebut nested function merupakan fungsi yang memiliki fungsi lagi di dalamnya. Inner function digunakan untuk membatasi dari semua yang terjadi di luar fungsi induk. Pada program ini, terdapat 3 fungsi yang menerapkan inner function.

```
1 def buat_fungsi_tambah_stok(jumlah):
2     def tambah_stok(barang):
3         return (barang[0], barang[1], barang[2] + jumlah, barang[3])
4     return tambah_stok
```

```
1 semua_barang = apply_operasi_pada_semua_barang(semua_barang, buat_fungsi_tambah_stok(0))
```





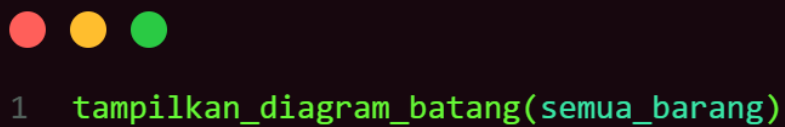
```

1  def decor(totalstok):
2      def teks():
3          print("\nTotal Stok Setelah Penambahan Stok:")
4          totalstok()
5      return teks
6
7  # Inner Function Untuk Menampilkan Total Stok
8  def hts():
9      print(hitung_total_stok(semua_barang))
10
11 # Menggunakan Decorator Untuk Menampilkan Total Stok
12 decorate = decor(hts)
13 decorate()

```

## 2.5 Data Visualization

Merupakan visualisasi data dalam bentuk chart, visualisasi ini dibuat dengan menggunakan library matplotlib, dan ditampilkan dalam *pop up*.



```

1  tampilkan_diagram_batang(semua_barang)

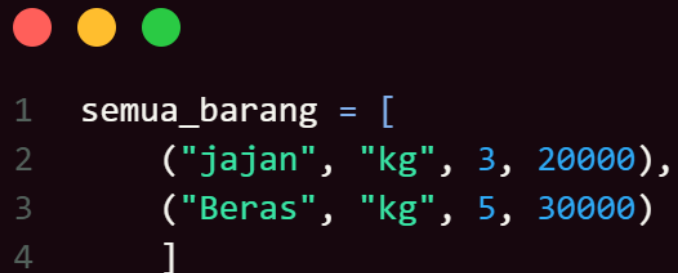
```

## BAB III

### LOCAL DAN GLOBAL

#### 3.1 Global Variabel

Merupakan variabel yang di inisiasi di luar fungsi-fungsi yang berada dalam satu program, sehingga variabel tersebut dapat diakses dan digunakan oleh fungsi lain yang berada dalam program tersebut.



```
1  semua_barang = [  
2      ("jajan", "kg", 3, 20000),  
3      ("Beras", "kg", 5, 30000)  
4  ]
```

#### 3.2 Local Variabel

Sedangkan lokal variabel merupakan variabel yang di inisiasi di dalam fungsi-fungsi yang berada dalam program, sehingga penggunaan variabel tersebut terbatas hanya pada fungsi tersebut dan tidak bisa digunakan oleh fungsi lainnya.

```

1 def tambah_barang(barang_list, nama, jenis, harga_per_unit):
2     # Input Jumlah Barang
3     jumlah = float(input("Masukkan jumlah barang: "))
4
5     # Inisialisasi Variabel Stok & Harga
6     stok, harga = 0, 0.0
7
8     # Cek Jenis Barang (Kg/Liter Atau Bukan)
9     if jenis.lower() == 'kg' or jenis.lower() == 'liter':
10        stok = jumlah
11        # Mengonversi Harga Per Unit Ke Dalam Format Numerik
12        harga = float(harga_per_unit.replace('.', '').replace(',', '.'))
13    else:
14        stok = int(jumlah)
15        harga = int(harga_per_unit)
16
17    # Menambahkan Barang Baru Ke Dalam Daftar Barang
18    barang_list.append((nama, jenis, stok, harga))
19
20    # Explicitly return barang_list
21    return barang_list

```

```

1 def hitung_total_penjualan(barang_list):
2     total_penjualan = sum([barang[2] * barang[3] for barang in barang_list])
3     return '{:,.0f}'.format(total_penjualan)

```

```

1 hitung_pendapatan = lambda barang_list: '{:,.0f}'.format(sum([barang[2] * barang[3] for barang in barang_list]))

```

```

1 barang_mahal = lambda barang_list, threshold: [barang[0] for barang in barang_list if barang[3] > threshold]

```

```

1 def generate_semua_nama_barang(barang_list):
2     for barang in barang_list:
3         yield barang[0]


```

## BAB 4


### HIGH ORDER FUNCTION

Merupakan fungsi yang dapat digunakan kembali secara terus menerus. Fungsi ini juga dapat di assign ke fungsi lain, dan juga dapat digunakan untuk menerima parameter.

High order function ini mengembalikan sebuah fungsi lain sebagai *return value*.



```
1 def apply_operasi_pada_semua_barang(barang_list, operasi):  
2     return [operasi(barang) for barang in barang_list]
```

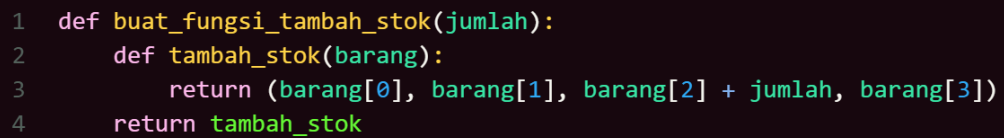


```
1 semua_barang = apply_operasi_pada_semua_barang(semua_barang, buat_fungsi_tambah_stok(0))
```

## BAB 5

### CLOSURE

Closure merupakan fungsi python, yang mana di dalam fungsi tersebut, terdapat fungsi lain yang dapat mengakses variabel di luar fungsi itu sendiri.



```
1 def buat_fungsi_tambah_stok(jumlah):
2     def tambah_stok(barang):
3         return (barang[0], barang[1], barang[2] + jumlah, barang[3])
4     return tambah_stok
```

## BAB 6

### DECORATOR

Merupakan fungsi yang sesuai dengan namanya, yaitu untuk menghias return value yang sudah ada. Decorator dapat digunakan untuk memodifikasi atau menambahkan fungsionalitas ke fungsi yang ada tanpa mengubah implementasi inti dari fungsi tersebut.

```
1  def decor(totalstok):
2      def teks():
3          print("\nTotal Stok Setelah Penambahan Stok:")
4          totalstok()
5      return teks
6
7  # Inner Function Untuk Menampilkan Total Stok
8  def hts():
9      print(hitung_total_stok(semua_barang))
10
11 # Menggunakan Decorator Untuk Menampilkan Total Stok
12 decorate = decor(hts)
13 decorate()
```

## **BAB 7**

### **LIBRARY DAN GUI**

#### **7.1 Penggunaan Library**

Dalam pengembangan program manajemen Toko Sembako ini, beberapa library Python digunakan untuk meningkatkan fungsionalitas dan mempermudah implementasi. Dua library yang secara khusus digunakan adalah:

##### **7.1.1 Matplotlib**

Matplotlib adalah library untuk membuat visualisasi dalam bentuk grafik dan plot. Pada program ini, Matplotlib digunakan untuk membuat histogram frekuensi harga masing-masing barang yang ada di toko sembako. Visualisasi ini memberikan pemahaman yang lebih baik tentang harga dari masing-masing barang.

##### **7.2.2 Reduce**

Reduce adalah fungsi bawaan Python yang berada di dalam modul functools. Fungsi reduce digunakan untuk mengaplikasikan suatu fungsi biner secara kumulatif ke elemen-elemen dari iterable (seperti list) untuk menghasilkan satu nilai akhir. Dengan kata lain, reduce menggabungkan elemen-elemen secara berurutan dengan menggunakan fungsi yang ditentukan. Visualisasi ini memberikan pemahaman yang lebih baik tentang urutan harga dari masing-masing barang dari yang termahal sampai yang termurah.