

# Memorization dengan Function Decorators untuk Resep Kue

Asa Do'a Uyi (122450005), Try Yani Rizki Nur Rohmah (122450020), Marleta Cornelia Leander (122450092), Sofyan Fauzi Dzaki Arif (122450116), Amalia Melani Putri (122450122)

## 1. Pendahuluan

*Function decorators* merupakan fungsi untuk mengolah dan mengubah fungsi lain tanpa perlu mengubah kodenya secara langsung. Salah satu pengembangan metode *function decorators* adalah *memorization*. *Memorization* pada *function decorators* dapat digunakan dalam pemanggilan argumen yang hasilnya dapat langsung disimpan tanpa perlu melakukan perhitungan ulang.

Pada laporan ini, kami akan menerapkan *memorization* pada *function decorators* untuk membuat resep kue lapis. Penerapan ini berisi langkah-langkah yang diperlukan untuk membuat kue lapis, yang mana fungsi *memorization* akan mengurangi tingkat *overhead* dari komputasi yang terjadi ketika terjadi pengulangan langkah-langkah yang sama.

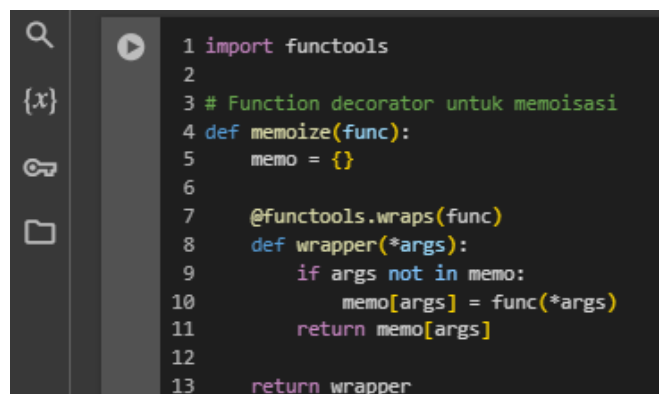
## 2. Metode

Metode yang digunakan dalam *function decorators* adalah :

1. Memodifikasi fungsi lain sebagai argumen (*first-class object*).
2. Memodifikasi prototype asli dari objek pembungkus dan bisa memodifikasi dengan metode baru seperti *number.prototype*, *string.prototype* dan *boolean.prototype*.
3. Memeriksa kelas, operator *instanceof* memungkinkan untuk memeriksa suatu objek milik class tertentu.

## 3. Pembahasan

### 3.1. Function decorator untuk memoisasi.



```
1 import functools
2
3 # Function decorator untuk memoisasi
4 def memoize(func):
5     memo = {}
6
7     @functools.wraps(func)
8     def wrapper(*args):
9         if args not in memo:
10             memo[args] = func(*args)
11         return memo[args]
12
13     return wrapper
```

Fungsi dari `memoize(func)` digunakan sebagai dekorator yang mengimplementasikan memoisasi. Fungsi ini menggunakan dictionary 'memo' untuk menyiapkan hasil dari pemanggilan fungsi yang telah dilakukan sebelumnya.

### 3.2. Fungsi-fungsi langkah-langkah dalam resep kue lapis.

```
# Fungsi-fungsi langkah-langkah dalam resep kue lapis
@memoize
def langkah_pertama():
    print("1. Menyiapkan adonan dasar kue lapis")
    return "Adonan dasar kue lapis disiapkan"

@memoize
def langkah_kedua():
    print("2. Memasak adonan dasar pertama")
    return "Adonan dasar pertama dimasak"

@memoize
def langkah_ketiga():
    print("3. Menyiapkan adonan kedua")
    return "Adonan kedua disiapkan"

@memoize
def langkah_keempat():
    print("4. Memasak adonan kedua")
    return "Adonan kedua dimasak"

@memoize
def langkah_kelima():
    print("5. Menumpuk adonan dasar dan adonan kedua secara bergantian")
    return "Adonan dasar dan adonan kedua ditumpuk"

@memoize
def langkah_keenam():
    print("6. Memanggang kue lapis hingga matang")
    return "Kue lapis matang dan siap disajikan"
```

Dan setiap langkah-langkah dalam pembuatan kue lapis (misalnya 'langkah pertama', 'langkah\_kedua', dan seterusnya.) di dekorasikan dengan '@memoize'. Ini berarti setiap kali fungsi tersebut dipanggil dengan argumen tertentu, hasilnya akan disimpan di dalam 'memo' jika belum ada sebelumnya, dan diambil dari 'memo' jika pemanggilan fungsi dengan argumen yang sama telah dilakukan sebelumnya.

### 3.3. Menggunakan fungsi-fungsi memoisasi untuk membuat kue lapis.

```
46 # Menggunakan fungsi-fungsi memoisasi untuk membuat kue lapis
47 print("--- Langkah-langkah Pembuatan Kue Lapis ---")
48 hasil_langkah_pertama = langkah_pertama()
49 print(hasil_langkah_pertama) # Output: Adonan dasar kue lapis disiapkan
50
51 hasil_langkah_kedua = langkah_kedua()
52 print(hasil_langkah_kedua) # Output: Adonan dasar pertama dimasak
53
54 hasil_langkah_ketiga = langkah_ketiga()
55 print(hasil_langkah_ketiga) # Output: Adonan kedua disiapkan
56
57 hasil_langkah_keempat = langkah_keempat()
58 print(hasil_langkah_keempat) # Output: Adonan kedua dimasak
59
60 hasil_langkah_kelima = langkah_kelima()
61 print(hasil_langkah_kelima) # Output: Adonan dasar dan adonan kedua ditumpuk
62
63 hasil_langkah_keenam = langkah_keenam()
64 print(hasil_langkah_keenam) # Output: Kue lapis matang dan siap disajikan
65
```

Penggunaan fungsi-fungsi langkah-langkah memoisasi untuk membuat kue lapis dilakukan setelah definisi fungsi-fungsi tersebut. Langkah-langkah tersebut dipanggil secara berurutan dan hasilnya dicetak.

### 3.4. Penggunaan Memoisasi

```
66 # Karena beberapa langkah sudah dijalankan sebelumnya,
67 # hasilnya langsung diambil dari memoisasi
68 print("--- Penggunaan Memoisasi ---")
69 hasil_langkah_pertama_lagi = langkah_pertama()
70 print(hasil_langkah_pertama_lagi) # Output: Adonan dasar kue lapis disiapkan
71
72 hasil_langkah_kedua_lagi = langkah_kedua()
73 print(hasil_langkah_kedua_lagi) # Output: Adonan dasar pertama dimasak
74
```

Dalam blok “penggunaan Memoisasi”, beberapa langkah-langkah yang telah dipanggil sebelumnya dipanggil lagi. Karena hasilnya telah disimpan dalam ‘memo’, maka hasilnya langsung diambil dari ‘memo’ tanpa perlu menjalankan ulang fungsi-fungsi tersebut.

## 4. Kesimpulan

Pada laporan ini, kami menerapkan metode *memorization* dengan *function decorators* untuk melakukan peningkatan yang efisiensi terhadap langkah-langkah pembuatan kue lapis. Dapat dilihat pemanfaatan *function decorators* mengurangi pengulangan langkah-langkah yang tidak diperlukan. Kemudian, *memorization* membantu pengoptimalan *function decorators*. Dari hal ini, disimpulkan jika penggunaan *memorization* dengan *function decorators* cukup efektif dan efisien.

Link Colab: <https://colab.research.google.com/drive/1lh-9f-mbsXrZfrBiF0w5YlvRho4JeDxe?usp=sharing>