

Dasar-Dasar Pemrograman 1

Tugas Pemrograman 04

Deadline: ~~12~~ 13 Desember 2022 23:55 Waktu SCoLe



FAKULTAS
ILMU
KOMPUTER

Tujuan

- Menerjemahkan dari kebutuhan pengguna ke solusi pemrograman.
- Memperoleh pengalaman praktis dari topik-topik yang sudah dipelajari terkait OOP, inheritance, dan GUI.

Pendahuluan

Pak Bambang ingin mengembangkan restorannya menjadi “Kafe Daun-Daun Pacilkom v2.0 🌿”. Pak Bambang tentu tetap ingin konsisten menu di restorannya bebas dari bahan aditif. Akan tetapi kini, Pak Bambang ada keinginan tambahan, yakni aplikasi restorannya bersifat adiktif dan disukai oleh banyak pelanggannya. Untuk itu, Pak Bambang tidak ingin aplikasinya berupa *text-based/Command Line Interface (CLI)*, tetapi lebih ke berbasis Graphical User Interface (GUI). **Sebagai catatan, Anda diwajibkan menerapkan OOP+inheritance dan menggunakan GUI tkinter.**

Penjelasan Program

Pada Tugas Pemrograman 04 kali ini, peserta akan membuat simulasi *Point of Sales* (POS) berbasis OOP, inheritance, dan GUI yang bisa digunakan pembeli untuk memesan makanan, minuman, dan pendamping.

Di “Kafe Daun-Daun Pacikom v2.0 🌿”, terdapat tiga jenis menu, yakni: MEALS, DRINKS, SIDES. Setiap jenis menu tersebut dapat berisi lebih dari satu menu. Tiap jenis menu memiliki **info tambahan** yang berbeda-beda: MEALS memiliki tingkat kegurihan (1-5); DRINKS memiliki tingkat kemanisan (1-5); dan SIDES memiliki tingkat keviralan (1-5).

Pada awal jalannya program, program akan membaca berkas bernama **menu.txt** yang berisi daftar menu, dengan format sebagai berikut:

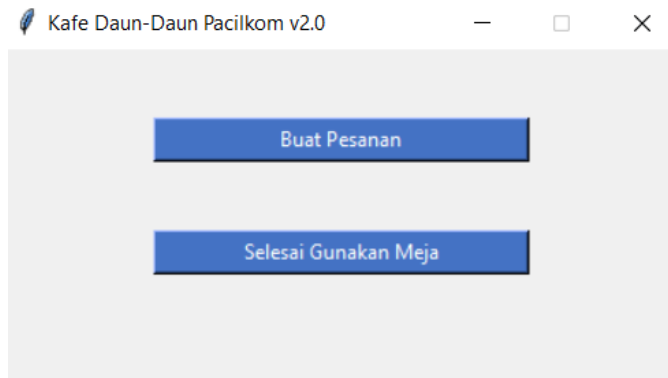
```
===JENIS MENU 1
kode_menu_1;Nama Menu 1;Harga (dalam Rupiah);Info Tambahan 1
kode_menu_2;Nama Menu 2;Harga (dalam Rupiah);Info Tambahan 2
kode_menu_3;Nama Menu 3;Harga (dalam Rupiah);Info Tambahan 3
===JENIS MENU 2
kode_menu_4;Nama Menu 4;Harga (dalam Rupiah);Info Tambahan 4
kode_menu_5;Nama Menu 5;Harga (dalam Rupiah);Info Tambahan 5
```

Contoh isi berkas menu.txt adalah sebagai berikut:

```
===MEALS
M01;Thai Beef Salad;52000;1
M02;Beef Korean Noodle;56000;3
===DRINKS
D01;Green Smoothies;25000;2
D02;Golden Glow;30000;4
===SIDES
T01;Soft Boiled Egg;10000;1
T02;Edamame;10000;3
```

Ketentuan Program

- Baik kode menu maupun nama menu bersifat unik.
- Harga pada menu berupa bilangan bulat positif atau nol.
- Tidak ada menu yang sama dalam satu berkas **menu.txt**.
- menu.txt akan dimuat secara otomatis pada saat program dijalankan dan diasumsikan formatnya selalu valid.
- Nama pembeli bersifat unik.
- Terdapat total 10 meja di “Kafe Daun-Daun Pacilkom v2.0 🌿”.
- Tampilan *landing page* dari programnya adalah sebagai berikut:



Perhatikan bahwa Anda perlu membayangkan bahwa aplikasi GUI ini dijalankan di sebuah komputer yang berada di depan restoran atau berada di kasir dekat pintu masuk. Aplikasi ini berjalan terus dari awal restoran buka hingga restoran tutup. Berikut alur penggunaan aplikasi tersebut:

- Ketika ada pembeli yang datang, pembeli tersebut menekan tombol “Buat Pesanan”, kemudian melihat ketersediaan meja dan memilih menu.
- Ketika seorang pembeli selesai makan, ia kemudian menuju kasir untuk membayar dan kemudian menekan tombol “Selesai Gunakan Meja” pada komputer yang berada di dekat kasir tersebut (tombol ini juga bisa dibantu oleh pelayan saat

pembayaran). Ketika tombol “Selesai Gunakan Meja” ditekan, detail menu yang disantap dan total harga yang harus dibayar akan ditampilkan.



Template

Untuk memudahkan pengerjaan, telah disediakan template yang berisi *class* yang dibutuhkan dan kerangka pengerjaan GUI. Implementasi **tidak wajib menggunakan template**. Template dapat diakses [di sini](#).

Penjelasan masing-masing fitur adalah sebagai berikut:

1. Fitur BUAT PESANAN

BUAT PESANAN adalah fitur untuk pembeli membuat pesanan baru.

Alur Fitur BUAT PESANAN

1. Pembeli dapat klik tombol “[Buat Pesanan](#)” dari tampilan *landing page* untuk menjalankan fitur ini.
2. Program akan menampilkan prompt “**Siapa nama Anda?**” kemudian pembeli dapat memasukkan namanya.

Kafe Daun-Daun Pacilkom v2.0

Siapa nama Anda? Donna

Kembali Lanjut

- Setelah pembeli menekan tombol “Lanjut”, program akan menampilkan daftar menu dengan jumlah pesanan yang masih kosong (= nol) beserta nomor meja yang diberikan secara acak dari meja-meja yang masih tersedia.¹
- Pembeli dapat menginput jumlah pesannya per menu, dengan total harga yang akan ter-*update* secara otomatis di tampilan GUI-nya.

Kafe Daun-Daun Pacilkom v2.0

Nama pemesan: Donna No Meja: 2 **Ubah**

MEALS

Kode	Nama	Harga	Kegurihan	Jumlah
M01	Thai Beef Salad	52000	1	3
M02	Beef Korean Noodle	56000	3	0

DRINKS

Kode	Nama	Harga	Kemanisan	Jumlah
D01	Green Smoothies	25000	2	2
D02	Golden Glow	30000	4	0

SIDES

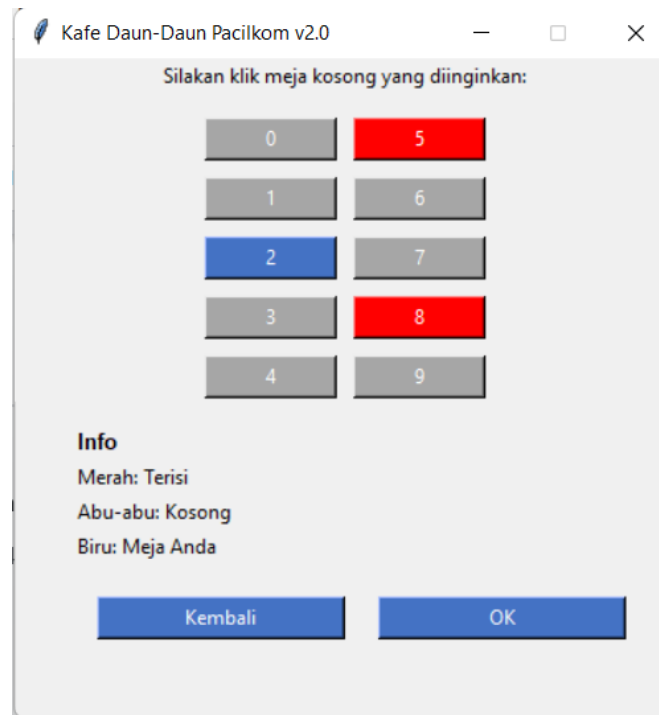
Kode	Nama	Harga	Keviralan	Jumlah
T01	Soft Boiled Egg	10000	1	0
T02	Edamame	10000	3	0

Total harga: 206000

Kembali OK

- Pembeli juga dapat mengubah nomor mejanya jika diperlukan.

¹ Dalam kasus ini, jika ternyata meja sudah penuh, akan muncul pop-up informasi “**Mohon maaf, meja sedang penuh. Silakan datang kembali di lain kesempatan.**”, dan kembali ke tampilan landing.



Hint

Pembuatan tabel pada tkinter dapat dilakukan dengan berbagai cara. Ada yang menggunakan TreeView dan ada juga yang memanfaatkan sifat grid layout dari sebuah container seperti Frame. Berikut adalah contoh kode untuk menghasilkan sebuah tabel menggunakan cara yang kedua. Anda bisa mencoba dan memodifikasi dari potongan program berikut:

```
import tkinter as tk
import tkinter.ttk as ttk
```

```

class Table(tk.Frame):
    def __init__(self, data, master = None):
        super().__init__(master)
        self.pack()
        self.data = data
        self.total_rows = len(self.data)
        self.total_columns = len(self.data[0])
        self.generate_table()

    def generate_table(self):
        for i in range(self.total_rows):
            for j in range(self.total_columns - 1):
                entry = tk.Entry(self, width = 20, fg = 'blue')
                entry.grid(row = i, column = j)
                entry.insert(tk.END, self.data[i][j])
                entry['state'] = 'readonly'

            # kolom paling kanan -> combobox
            values = tuple([k for k in range(10)])
            opsi_jumlah = ttk.Combobox(self, values = values)
            opsi_jumlah.grid(row = i, column = self.total_columns - 1)

if __name__ == "__main__":
    menu_list = [('M01', 'Thai Beef Salad', 52000, 1, 1),
                 ('M02', 'Bakso', 20000, 2, 3),

```

```

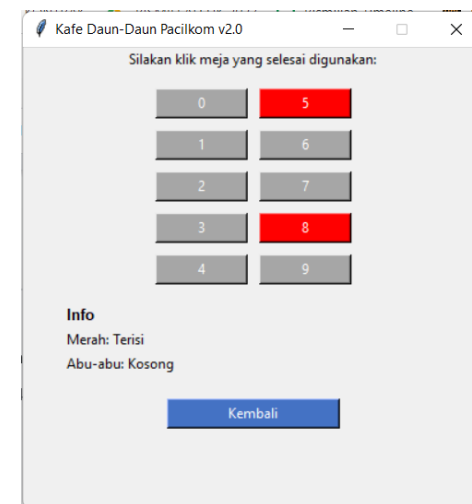
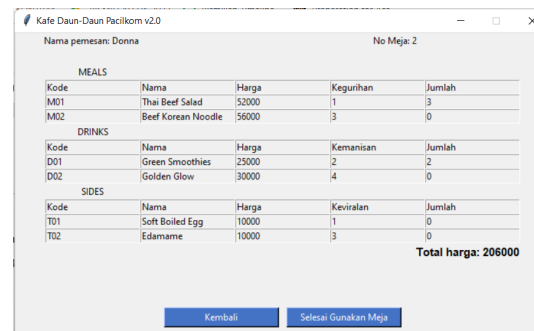
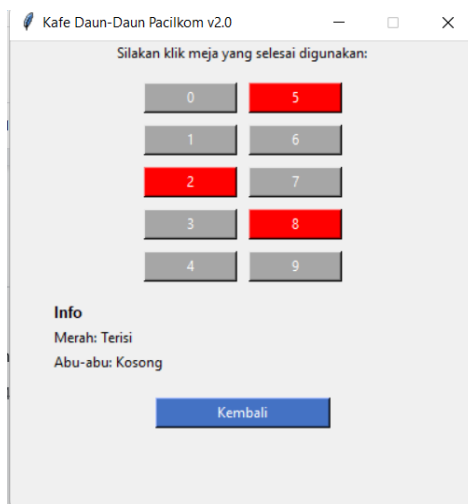
('M03', 'Mie Ayam', 20000, 2, 2),
('M04', 'Semangkok Mecin', 20000, 5, 3)]

gui = tk.Tk()
table = Table(menu_list, gui)
gui.mainloop()

```

2. Fitur SELESAI GUNAKAN MEJA

SELESAI GUNAKAN MEJA adalah fitur untuk pembeli ketika sudah selesai makan. Untuk mengakses fitur ini, pada tampilan *landing page*, pembeli dapat klik “[Selesai Gunakan Meja](#)”. Ketika meja selesai digunakan, nomor meja tersebut akan bisa dipesan oleh pembeli berikutnya.



Komponen Penilaian

Sesuai dengan ketentuan yang ada di [Rubrik Penilaian TP DDP1](#).

Plagiarisme tidak akan ditoleransi. Anda diperbolehkan berdiskusi dengan teman terkait ide implementasi tugas ini. **Harap menuliskan kolaborator jika berdiskusi dengan mahasiswa lain.** Perlu diingat bahwa **implementasi kode dilakukan secara individu.** Tim pengajar akan melakukan *code similarity checking* pada implementasi kode mahasiswa, dan jika terbukti melakukan kecurangan/plagiarisme akan mendapat **sanksi berupa nilai 0** pada TP ini. Perlu diperhatikan bahwa **menuliskan kolaborator tidak akan membebaskan pekerjaan Anda dari *code similarity checking*.**

Pengumpulan

Berkas yang perlu dikumpulkan:

- [Kelas]_[KodeAsdos]_[NPM]_[NamaLengkap]_TP04.py

Kumpulkan berkas .py yang **telah di-zip** dengan format penamaan seperti berikut.

[Kelas]_[KodeAsdos]_[NPM]_[NamaLengkap]_TP04.zip

Contoh:

A_ABC_2206123456_ThamiEndamora_TP04.zip