Collabits Journal
Vol. 2 No. 2 May 2025 : 103-105
E-ISSN : 1979-5254, P-ISSN : 3062-8601
https://publikasi.mercubuana.ac.id/index.php/collabits

Python Programming Implementation in Food Catalogue Creation Using GUI

Brilian Danuarta[1*], Ifan Prihandi[2]

[1,2] Computer Science, Universitas Mercu Buana, Indonesia

*Coressponden Author: danubrillian@gmail.com

Abstract - In the modern world of programming, the development of user interface (GUI) applications is very important. A simple GUI application can be implemented using the Python programming language and standard modules such as Tkinter, Pandas, and os, as described in this journal. Using this application, users can view the product catalogue, add products to the shopping cart, and make payments. There will be a record of the transaction in an Excel file after the payment is completed. Readers will learn the basic concepts of GUI application development using Python through this practical approach. They will also learn how to use basic features such as file management, user interaction, and data manipulation. This article can serve as a guide for beginners in developing GUI applications using Python.

Keywords:
Application Food Catalogue;
Tkinter;
GUI;

Article History:
Received: 09-04-2025
Revised: 15-05-2025
Accepted: 19-05-2025

Article DOI : 10.22441/collabits.v2i2.27266

1. INTRODUCTION

User interface (GUI) application development has become an important part of the modern programming world as it allows users to interact with software easily and efficiently. Python is one of the most used programming platforms for developing GUI applications as it has many modules and libraries that support the construction of feature-rich and responsive interfaces.

This article describes the development of a simple GUI application built using Python, focusing on the use of Tkinter, Pandas, and os modules. The application aims to help people manage item catalogues, make purchases, and record payment transactions. The user-friendly user interface makes it easy for users to browse the item catalogue, add items to the shopping cart, and initiate the payment process.

This article will discuss in detail every aspect of developing this app in this context, including the code structure, functionality, and implementation of key features. In addition, this article will explain practical steps for using the Tkinter module to build graphical interfaces and how to use the Pandas module to properly manage and transform data.

This case study will teach readers the basic concepts of GUI application development using Python. They will also learn how to use the various modules and libraries available to create useful and effective applications in the real world.

A catalog is a list or book that includes details about different products or services that a company or retailer sells. Images, descriptions, and costs of the products or services are typically included. Typically, the purpose of these catalogs is to persuade prospective clients to buy those products or services. They are available for printing as well as digital access via apps and websites. A variety of products and services, including apparel, electronics, and home furnishings, are available through catalogs.

"Graphical User Interface" (Antarmuka Pengguna Grafis) is the acronym for GUI. This pertains to the visual elements used in computer programs to enable users to interact with the wild through graphic elements such as buttons, menus, and icons. GUI provides a more intuitive way for users to communicate with a lunak penangkat than antarmuka teks based on perintah baris. Examples of GUIs include modern desktop computers like Windows, macOS, and various desktop environments running various Linux distributions, as well as mobile devices like Android and iOS.The graphical user interface (GUI) enables users to perform many tasks such as opening files, launching applications, editing documents, and many more by utilizing input

from keras devices like a mouse, keyboard, or touch screen.

Python is a high-level programming language that is widely used and easy to learn. Python, which was developed by Guido van Rossum and first published in 1991, has grown to become one of the most widely used programming languages worldwide due to its robust syntax, ease of use, and impressive computational power. Python is utilized in various fields such as web development, game development, data analysis, kecerdasan manufacturing, linguistic analysis, and many more.

2. METHOD

The Tkinter module is a standard Python module used to create user interface (GUI) graphics. Use Tkinter to create a window for a website or application.

2.1*Class Catalog App:*

```
class Catalog (variable) master: Any
    def __ini
        self.master = master
        self.master.title("Katalog Barang")

        self.catalog = {"mie goreng": 6000, "mie rebus": 6500, "aqua": 3000, "rokok": 300
        self.cart = {}

        self.create_widgets()
```

This is the main class that creates the barang catalog application. Use the __init__() method to initialize the CatalogApp object. In this case, the main application is customized, and the create_widgets() function is used to generate user interface elements.

2.2*Create_widgets() method:*

```
def create_widgets(self):
    self.master.geometry("800x400")

    self.catalog_frame = tk.Frame(self.master, width=400, bd=2, relief=tk.SOLID)
    self.catalog_frame.pack(side=tk.LEFT, padx=10, pady=10, fill=tk.BOTH, expand=True)

    self.cart_frame = tk.Frame(self.master, width=400, bd=2, relief=tk.SOLID)
    self.cart_frame.pack(side=tk.RIGHT, padx=10, pady=10, fill=tk.BOTH, expand=True)

    for item, price in self.catalog.items():
        item_frame = tk.Frame(self.catalog_frame)
        item_frame.pack(fill=tk.X)

        tk.Label(item_frame, text=f"{item}:", width=15, anchor=tk.W).pack(side=tk.LEFT)
        tk.Label(item_frame, text=f"Rp {price:,.2f}", width=10, anchor=tk.W).pack(side=tk.LEFT)
        tk.Button(item_frame, text="Tambah", command=lambda i=item: self.add_to_cart(i)).pack(side=tk.LEF
```

This method creates GUI elements such as product descriptions and pricing lists. Product catalogs are stored in catalog_frame, whereas product specifications are stored in cart_frame. Every item in the barang catalog has a label, a price, and "Tambah" button to indicate its current state. *Decision Support System* is a computer- based system that can support semi-structured decision making, by utilizing data and then processing it into information in the form of suggestions

2.3*Add_to_cart()and remove_from_cart() methods:*

```
def add_to_cart(self, item):
    if item in self.catalog:
        if item in self.cart:
            self.cart[item] += 1
        else:
            self.cart[item] = 1
        self.update_cart()
    else:
        messagebox.showerror("Error", "Barang tidak ditemukan dalam katalog.")
                            (variable) cart: dict
def remove_from_cart
    if item in self.cart:
        if self.cart[item] > 1:
            self.cart[item] -= 1
        else:
            del self.cart[item]
        self.update_cart()
    else:
        messagebox.showerror("Error", "Barang tidak ditemukan dalam keranjang.")
```

These methods help in the insertion and removal of items from the belanja keranjang. When a user selects the "Tambah" button next to an item in the catalog, add_to_cart() is invoked to add the selected item to the queue. When a user wants to remove an item from the cart, remove_from_cart() can be used to reduce the quantity or remove the item from the cart.

2.4*Reset cart() method:*

```
def reset_cart(self):
    confirmed = messagebox.askyesno("Reset Keranjang", "Anda yakin ingin mereset keranjang ?")
    if confirmed:
        self.cart.clear()
        self.update_cart()
```

This method is activated when the user presses the “Reset Keranjang” button. This opens aconfirmation popup using messagebox.askyesno() to confirm whether the user wants to set a default. If confirmed, the cart will be dissolved.

2.5*Update_Cart() Method:*

```
def update_cart(self):
    self.cart_listbox.delete(0, tk.END)

    for item, quantity in self.cart.items():
        self.cart_listbox.insert(tk.END, f"{item} (x{quantity})")
```

This method is used to adjust the cart shop tampilan every time there is a change in the cart. This creates a new widget for each item in the keranjang and then moves all of the widgets in the cart_frame.

2.6 *Building a Tkinter application:*

```
if __name__ == "__main__":
    root = tk.Tk()
    app = CatalogApp(root)
    app.master.title('Kedai 88')

    root.mainloop()
```

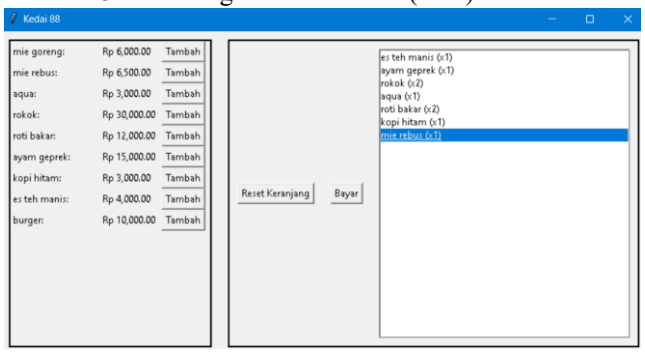
Create an instance of tk.Tk() as the primary window.

Creating a CatalogApp instance with the aforementioned primary key. Use root.title() to set the

JND title.Use PIL to create a GIF image as a universal icon by using Image.open() and ImageTk.PhotoImage(). Append the image to the JP icon using root.tk.call().

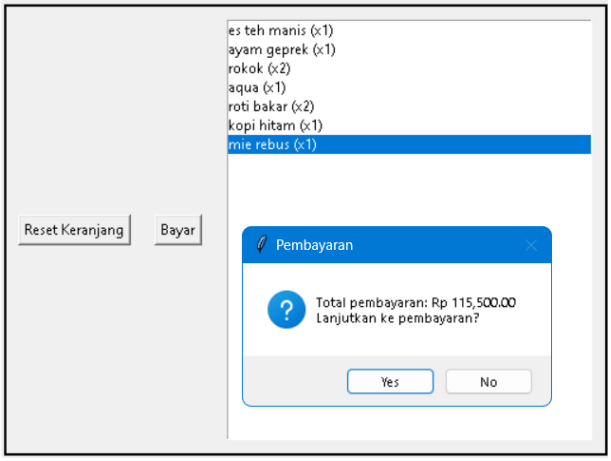
3. RESULT & OUTPUTS

3.1 Creating a user interface (GUI)



➢ The user interface is created by using Tkinter elements such as LabelFrame, Label, Button, and Frame. Utilizing a loop through each item in the catalog, the daftar barang is displayed.
Every object has a "Tambah" button that enables users to attach objects to cart.

➢ 3.2. Payment and receipt of the transaction:



➢ When the user makes a payment, the total amount is determined by the item's price and quantity in the store. Pengguna will be questioned in order to confirm the payment before completing the transaction. After payment confirmation, the transaction will be recorded in an Excel file.

➢ 3.3. Integral Reliances:

	A	B	C	D	E	F
1	Tanggal Waktu	Item	Jumlah	Harga Satuan	Total Harga	
2	2024-05-20 09:16:16	burger	1	10000	10000	
3	2024-05-20 09:16:16	kopi hitam	1	3000	3000	
4	2024-05-20 09:16:40	es teh manis	1	4000	4000	
5	2024-05-20 09:16:40	burger	1	10000	10000	
6	2024-05-20 09:17:22	es teh manis	1	4000	4000	
7	2024-05-20 09:17:22	burger	1	10000	10000	
8	2024-05-20 10:58:34	es teh manis	2	4000	8000	
9	2024-05-20 10:58:34	ayam geprek	1	15000	15000	
10	2024-05-20 10:58:34	aqua	2	3000	6000	
11	2024-05-20 10:58:34	mie rebus	2	6500	13000	
12	2024-05-20 10:58:34	roti bakar	3	12000	36000	

➢ This code uses external modules like datetime, os, and pandas to do operations like copying data to an Excel file and setting the time and date.

4. CONCLUSION

In summary, this code is a clean implementation of a barcode katalog application using Python's Tkinter library. This application allows users to view product descriptions, add items to a belanja keranjang, organize keranjang items, sort keranjang items, and complete payments. To improve this code's quality and functionality, it is advised to increase code complexity, code maintenance, user input validation, user antarmuka, work efficiency, document augmentation, and unit testing. By implementing these strategies, the group can become more robust and able to be manipulated.

BIBLIOGRAPHY

[1] Python, W. (2021). Python. Python releases for windows, 24.

[2] Martinez, W. L. (2011). Graphical user interfaces. Wiley Interdisciplinary Reviews: Computational Statistics, 3(2), 119-133.

[3] Prehanto, Dedy Rahman, and Maya Lailatul Fitriah. "RANCANG BANGUN SISTEM INFORMASI PEMESANAN KATERING BERBASIS WEB (Studi Kasus Pada Katering Jelita)." Inovate: Jurnal Ilmiah Inovasi Teknologi Informasi 1.1 (2016).

[4] PREHANTO, Dedy Rahman; FITRIAH, Maya Lailatul. RANCANG BANGUN SISTEM INFORMASI PEMESANAN KATERING BERBASIS WEB (Studi Kasus Pada Katering Jelita). Inovate: Jurnal Ilmiah Inovasi Teknologi Informasi, 2016, 1.1.

[5] HEYDT, Michael. Learning pandas. Packt Publishing Ltd, 201