

GUI PENGELOLAAN STOK GUDANG MENGUNAKAN PYTHON

Mata Kuliah : Pemrograman Berbasis Objek Praktek



Disusun Oleh :

Risang Haryo Pamungkas Bimani Sakti
(5230411290)

**FAKULTAS SAINS DAN TEKNOLOGI
PROGRAM STUDI INFORMATIKA
UNIVERSITAS TEKNOLOGI YOGYAKARTA
2024/2025**

LATAR BELAKANG

Gudang rumah adalah suatu bangunan atau ruang yang dirancang khusus untuk menyimpan berbagai barang atau peralatan yang tidak digunakan terlalu sering dalam kehidupan sehari-hari. Dalam gudang rumah, pastinya semua barang harus terorganisir dan rapi dalam penempatannya. Namun, sering kali orang lupa meletakkan barang tertentu itu di mana atau bahkan lupa barang apa saja yang telah disimpan di dalam gudang tersebut.

Dari permasalahan yang timbul, saya berinisiatif untuk membuat program yang dapat mencatat semua aktivitas yang pernah dilakukan oleh pengguna di gudang tersebut agar dapat memantau barang apa saja yang telah tersimpan di gudang rumah milik pengguna.

PEMBAHASAN

1. Impor library

```
1 from tkinter import ttk, messagebox, filedialog
2 import tkinter as tk
3 import pandas as pd
4 import datetime as dt
```

kode di atas digunakan untuk mengimpor library yang dibutuhkan dalam program ini.

2. Buat kelas stok gudang

```
1 class StuffStock:
2     def __init__(self, window):
3         '''Untuk menginisialisasi beberapa hal yang dibutuhkan secara langsung'''
4         self.window = window
5         self.window.title("Aplikasi Penyimpanan Stok Barang")
6         self.window.geometry("700x600")
7         self.window.configure(bg="#242150")
8         self.category_dict = {"Furnitur": "Sekat A", "Elektronik": "Sekat B", "Perkakas": "Sekat C", "Bahan Material": "Sekat D", "Lainnya": "Sekat E"}
9         self.unit_list = ["Kilogram", "Meter", "Liter", "Unit", "Kardus", "Sak"]
```

kode di atas digunakan untuk membuat class stuffStock yang didalamnya langsung menginisialisasi beberapa hal, disini init saya gunakan untuk mengatur window.

3. Buat fungsi Inisialisasi semua variabel entry

```
1 def initVars() -> None:
2     '''Fungsi untuk menginisialisasi semua variabel yang digunakan untuk menampung inputan user secara real time di entry form'''
3     self.name_var = tk.StringVar()
4     self.category_var = tk.StringVar()
5     self.qty_var = tk.StringVar()
6     self.unit_var = tk.StringVar()
7     self.location_var = tk.StringVar()
8     self.category_var.trace_add("write", generateLocate)
9     self.qty_var.trace_add("write", validateData)
```

4. Buat fungsi untuk memvalidasi input pengguna harus bertipe data float

```
1 def isfloat(value) -> None:
2     '''Fungsi untuk memvalidasi input pengguna harus bertipe data float'''
3     try:
4         float(value)
5         return True
6     except ValueError:
7         return False
```

5. Buat fungsi untuk mengosongkan formulir

```
1 def resetForm() -> None:
2     '''Fungsi untuk mengosongkan formulir'''
3     for var in (self.name_var, self.category_var, self.qty_var, self.unit_var, self.location_var):
4         var.set('')
```

6. Buat fungsi untuk memberikan penanda jika validasi data salah

```
1 def validateData(*args) -> None:
2     '''Fungsi untuk memberikan penanda jika jumlah tidak berupa angka'''
3     self.qty_entry.config(bg='white' if (isfloat(self.qty_var.get()) or self.qty_var.get() == "") else 'pink')
```

7. Buat fungsi untuk menyingkronkan kategori barang dan lokasi penempatan

```
1 def generateLocate(*args) -> None:
2     '''Fungsi untuk mendapatkan value dari Lokasi berdasarkan kategori barang'''
3     category = self.category_var.get()
4     if category and category in self.cateogry_dict:
5         location = self.cateogry_dict[category]
6         self.location_var.set(f"{location}")
```

8. Buat fungsi untuk menambahkan data ke daftar gudang

```
1 def addData():
2     '''Fungsi untuk menambahkan barang ke gudang'''
3     name = self.name_var.get()
4     category = self.category_var.get()
5     qty = self.qty_var.get()
6     unit = self.unit_var.get()
7
8     # Untuk memeriksa apakah ada nama yang sama lalu memberikan pesan tertentu
9     for item in self.stuff_table.get_children():
10         item_values = self.stuff_table.item(item, 'values')
11         if item_values[0].lower() == name.lower():
12             messagebox.showwarning("", f"{name} Sudah Ada")
13             return
14
15     # Untuk memeriksa apakah semua form sudah terisi
16     if name and category and isfloat(qty) and unit:
17         float(qty)
18         location = self.cateogry_dict[category]
19         datetime = dt.date.today()
20         self.stuff_table.insert("", "end", values=(name, category, qty, unit, location, datetime))
21         resetForm() # Untuk mengosongkan form
22     else:
23         messagebox.showerror("Error", "Pastikan Formulir Terisi Semua dan Jumlah Barang Berupa Angka")
```

9. Buat fungsi untuk menghapus data yang dipilih dari daftar gudang

```
1 def deleteData():
2     '''Fungsi untuk menghapus barang atau beberapa barang yang dipilih'''
3     selected_item = self.stuff_table.selection()
4     if not selected_item:
5         messagebox.showerror("Error", "Tidak ada barang yang dipilih")
6         return
7
8     # Untuk konfirmasi apakah yakin ingin menghapus barang ini?
9     confirm = messagebox.askyesno("Konfirmasi", "Yakin Ingin Menghapus Barang Ini?")
10    if confirm == False:
11        return
12    for item in selected_item:
13        self.stuff_table.delete(item)
14    messagebox.showinfo("Info", "Barang Berhasil Dihapus")
```

10. Fungsi untuk mengubah data yang dipilih dari daftar gudang

```
1 def editData():
2     '''Fungsi untuk memilih data yang hendak diedit
3     (disarankan untuk memilih salah satu barang saja karena jika memilih lebih dari satu,
4     maka yang akan diambil adalah barang yang dipilih pertama kali)'''
5     selected_item = self.stuff_table.selection()
6     if not selected_item:
7         messagebox.showerror("Error", "Tidak ada barang yang dipilih")
8         return
9
10    # Memastikan yang bisa diedit adalah barang yang pertama kali dipilih
11    item_values = self.stuff_table.item(selected_item[0], option='values')
12    variables = [self.name_var, self.category_var, self.qty_var, self.unit_var, self.location_var]
13    for i, var in enumerate(variables):
14        var.set(item_values[i])
15    self.id_selected_item = selected_item[0]
```

```

1 def updateData():
2     '''Untuk memperbaharui barang yang telah dipilih'''
3     # menyingkronkan barang yang dipilih sesuai dengan fungsi editData
4     if not hasattr(self, 'id_selected_item') or not self.id_selected_item:
5         messagebox.showerror("", "Tidak Ada Data Yang Harus Diupdate")
6         return
7
8     # Sama seperti fungsi addData tetapi mengganti value barang yang dipilih
9     name = self.name_var.get()
10    category = self.category_var.get()
11    qty = self.qty_var.get()
12    unit = self.unit_var.get()
13
14    # Untuk memeriksa apakah ada nama yang sama lalu memberikan pesan tertentu
15    for item in self.stuff_table.get_children():
16        item_values = self.stuff_table.item(item, 'values')
17        if item_values[0].lower() == name.lower():
18            messagebox.showwarning("", f"{name} Sudah Ada")
19            return
20
21    if name and category and isfloat(qty) and unit:
22        float(qty)
23        location = self.categry_dict[category]
24        datetime = dt.date.today()
25        self.stuff_table.item(self.id_selected_item, values=(name, category, qty, unit, location, datetime))
26        self.id_selected_item = None
27        resetForm()
28        messagebox.showinfo("", "Berhasil mengubah barang")
29    else:
30        messagebox.showerror("Error", "Pastikan Formulir Terisi Semua dan Jumlah Barang Berupa Angka")

```

11. Fungsi untuk mengekspor file ke spreadsheet

```

1 def exportFile():
2     '''Untuk mengekspor file ke bentuk spreadsheet'''
3     # Menambahkan semua item di tabel ke tipe data list
4     data = []
5     for value in self.stuff_table.get_children():
6         values = self.stuff_table.item(value, 'values')
7         data.append(values)
8
9     # Memeriksa jika data kosong maka memunculkan pesan tertentu
10    if data == []:
11        messagebox.showerror("", "Tidak Ada Data Yang Diekspor")
12        return
13
14    df = pd.DataFrame(data, columns=("Nama", "Kategori", "Jumlah", "Satuan", "Lokasi", "Tanggal"))
15    file_path = filedialog.asksaveasfilename(defaultextension=".csv", filetypes=[("Excel", "*.xlsx"), ("CSV", "*.csv")], title="Simpan File")
16
17    if file_path:
18        if file_path.endswith(".csv"):
19            df.to_csv(file_path, index=False)
20        else:
21            df.to_excel(file_path, index=False)
22        messagebox.showinfo("Sukses", f"Data berhasil diekspor ke {file_path}")

```

12. Fungsi untuk mengimpor file dari penyimpanan lokal

```
1 def importFile():
2     '''Untuk mengimpor file Excel atau CSV'''
3     file_path = filedialog.askopenfilename(defaultextension=".csv", filetypes=[("CSV", "*.csv"), ("Excel", "*.xlsx")], title="Cari File")
4
5     if file_path:
6         if file_path.endswith(".csv"):
7             df = pd.read_csv(file_path)
8         else:
9             df = pd.read_excel(file_path)
10    else:
11        return
12
13    for item in self.stuff_table.get_children():
14        self.stuff_table.delete(item)
15
16    for i, value in df.iterrows():
17        self.stuff_table.insert("", "end", values=tuple(value))
18
19    messagebox.showinfo("Sukses", f"Data berhasil diimpor")
```

13. Fungsi untuk membuat widget stok gudang

```
1 def createWidget(self):
2     ## Mengganti Tema
3     style = ttk.Style().theme_use('clam')
4
5     ## Mengonfigurasi Grid
6     self.window.grid_columnconfigure(0, weight=1)
7     self.window.grid_rowconfigure(3, weight=1)
8
9     ## Label Judul
10    title_label = tk.Label(self.window, text="Selamat Datang di Gudang", bg="#242158", font=("Helvetica", 16, "bold"), foreground="white")
11    title_label.grid(row=0, column=0, pady=8)
12
13    ## Membuat Frame Bagian Atas
14    top_frame = tk.Frame(self.window, bg="#213a58", relief="solid", borderwidth=1)
15    top_frame.grid(row=1, column=0, sticky="nsw", padx=10, pady=10)
16    top_frame.columnconfigure(0, weight=1)
17    top_frame.columnconfigure(1, weight=3)
18
19    # Menginisialisasi semua var
20    initVars()
21
22    # Nama Barang
23    name_label = tk.Label(top_frame, text="Nama Barang : ", bg="#213a58", font=("Helvetica", 10, "bold"), foreground="white")
24    name_label.grid(row=0, column=0, sticky="w", padx=4, pady=4)
25    self.name_entry = tk.Entry(top_frame, textvariable=self.name_var)
26    self.name_entry.grid(row=0, column=1, sticky="ew", padx=4, pady=4)
27
28    # Kategori Barang
29    category_label = tk.Label(top_frame, text="Kategori Barang : ", bg="#213a58", font=("Helvetica", 10, "bold"), foreground="white")
30    category_label.grid(row=1, column=0, sticky="w", padx=4, pady=4)
31    self.category_option = ttk.Combobox(top_frame, textvariable=self.category_var, values=list(self.category_dict.keys()), state="readonly")
32    self.category_option.grid(row=1, column=1, sticky="ew", padx=4, pady=4)
33
34    # Jumlah Barang
35    qty_label = tk.Label(top_frame, text="Jumlah Barang : ", bg="#213a58", font=("Helvetica", 10, "bold"), foreground="white")
36    qty_label.grid(row=2, column=0, sticky="w", padx=4, pady=4)
37    self.qty_entry = tk.Entry(top_frame, textvariable=self.qty_var)
38    self.qty_entry.grid(row=2, column=1, sticky="ew", padx=4, pady=4)
39
40    # Satuan Barang
41    unit_label = tk.Label(top_frame, text="Satuan Barang : ", bg="#213a58", font=("Helvetica", 10, "bold"), foreground="white")
42    unit_label.grid(row=3, column=0, sticky="w", padx=4, pady=4)
43    self.unit_option = ttk.Combobox(top_frame, textvariable=self.unit_var, values=self.unit_list, state="readonly")
44    self.unit_option.grid(row=3, column=1, sticky="ew", padx=4, pady=4)
45
46    # Lokasi Penempatan Barang
47    location_label = tk.Label(top_frame, text="Lokasi Penempatan : ", bg="#213a58", font=("Helvetica", 10, "bold"), foreground="white")
48    location_label.grid(row=4, column=0, sticky="w", padx=4, pady=4)
49    self.location_entry = tk.Entry(top_frame, textvariable=self.location_var, state="readonly")
50    self.location_entry.grid(row=4, column=1, sticky="ew", padx=4, pady=4)
51
```



```

1  ## Membuat Frame Bagian Tengah
2  middle_frame = tk.Frame(self.window, bg="#242158")
3  middle_frame.grid(row=2, column=0, sticky="nesw", padx=10)
4
5  # Tombol Taruh Gudang
6  self.submit_btn = ttk.Button(middle_frame, text="Taruh", command=addData)
7  self.submit_btn.pack(side='right')
8
9  # Tombol Update Barang yang Dipilih
10 self.update_btn = ttk.Button(middle_frame, text="Update", command=updateData)
11 self.update_btn.pack(side='right', padx=4)
12
13 # Tombol Edit Barang yang Dipilih
14 self.edit_btn = ttk.Button(middle_frame, text="Edit", command=editData)
15 self.edit_btn.pack(side='right')
16
17 # Tombol Hapus barang yang dipilih
18 self.delete_btn = ttk.Button(middle_frame, text="Hapus", command=deleteData)
19 self.delete_btn.pack(side='right', padx=4)
20
21 # Tombol Export ke Spreadsheet
22 self.export_btn = ttk.Button(middle_frame, text="Ekspor File", command=exportFile)
23 self.export_btn.pack(side='left')
24
25 # Tombol Import ke Python
26 self.import_btn = ttk.Button(middle_frame, text="Impor File", command=importFile)
27 self.import_btn.pack(side='left', padx=4)

```

```

1  ## Membuat Frame Bagian Bawah
2  bottom_frame = tk.Frame(self.window, bg="#213a58", relief="solid", borderwidth=1)
3  bottom_frame.grid(row=3, column=0, sticky="nesw", padx=10, pady=10)
4
5  # Label Tabel
6  title_label = tk.Label(bottom_frame, text="Daftar Barang", bg="#213a58", font=("Helvetica", 12, "bold"), foreground="white")
7  title_label.pack(fill="both")
8
9  # Daftar Barang Heading
10 self.stuff_table = ttk.Treeview(bottom_frame, columns=("Nama", "Kategori", "Jumlah", "Satuan", "Lokasi", "Tanggal"), show="headings")
11 self.stuff_table.heading("Nama", text="Nama")
12 self.stuff_table.heading("Kategori", text="Kategori")
13 self.stuff_table.heading("Jumlah", text="Jumlah")
14 self.stuff_table.heading("Satuan", text="Satuan")
15 self.stuff_table.heading("Lokasi", text="Lokasi")
16 self.stuff_table.heading("Tanggal", text="Tanggal")
17
18 # Daftar Barang Body
19 self.stuff_table.column("Nama", anchor="w", width=150)
20 self.stuff_table.column("Kategori", anchor="w", width=50)
21 self.stuff_table.column("Jumlah", anchor="e", width=10)
22 self.stuff_table.column("Satuan", anchor="w", width=10)
23 self.stuff_table.column("Lokasi", anchor="w", width=50)
24 self.stuff_table.column("Tanggal", anchor="w", width=50)
25 self.stuff_table.pack(fill="both", expand=True, padx=12, pady=12)

```

14. Buat objek stok gudang kemudian panggil fungsi createWidget()



Note :

- 1. Fungsi nomor 3-12 berada di dalam fungsi createWidget() supaya tidak dapat diubah dari luar kelas.*
- 2. Penjelasan ada di dalam kode.*

IMPLEMENTASI

1. Memasukkan data barang.

Selamat Datang di Gudang

Nama Barang :	<input type="text" value="Kardus Besar"/>
Kategori Barang :	<input type="text" value="Lainnya"/>
Jumlah Barang :	<input type="text" value="20"/>
Satuan Barang :	<input type="text" value="Unit"/>
Lokasi Penempatan :	<input type="text" value="Sekat E"/>

Ekspor File

Impor File

Hapus

Edit

Update

Taruh

Daftar Barang

Nama	Kategori	Jumlah	Satuan	Lokasi	Tanggal
------	----------	--------	--------	--------	---------

Isi formulir di atas lalu klik tombol taruh.

Selamat Datang di Gudang

Nama Barang :

Kategori Barang :

▼

Jumlah Barang :

Satuan Barang :

▼

Lokasi Penempatan :

Ekspor File

Impor File

Hapus

Edit

Update

Taruh

Daftar Barang

Nama	Kategori	Jumlah	Satuan	Lokasi	Tanggal
Kardus Besar	Lainnya	20	Unit	Sekat E	2024-11-28

Maka akan tampil di daftar barang.

2. Mengubah data di daftar barang.

Daftar Barang					
Nama	Kategori	Jumlah	Satuan	Lokasi	Tanggal
Kardus Besar	Lainnya	20	Unit	Sekat E	2024-11-28

Pilih data yang ingin diubah

Ekspor File

Impor File

Hapus

Edit

Update

Taruh

Klik tombol edit.

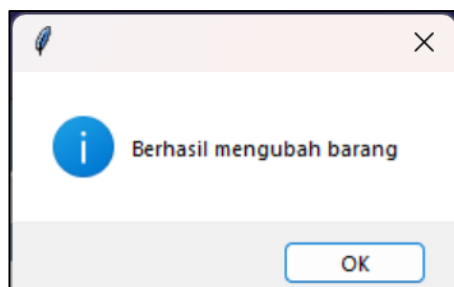
Nama Barang :	Kardus Besar
Kategori Barang :	Lainnya
Jumlah Barang :	20
Satuan Barang :	Unit
Lokasi Penempatan :	Sekat E

Di formulir akan muncul data yang telah dipilih.

Nama Barang :	Kardus Besar 100x200x300cm
Kategori Barang :	Lainnya
Jumlah Barang :	20
Satuan Barang :	Unit
Lokasi Penempatan :	Sekat E

Ekspor File
Impor File
Hapus
Edit
Update
Taruh

Ubah isi dari data yang telah dipilih, kemudian klik update.



Daftar Barang					
Nama	Kategori	Jumlah	Satuan	Lokasi	Tanggal
Kardus Besar 100x200x300cm	Lainnya	20	Unit	Sekat E	2024-11-28

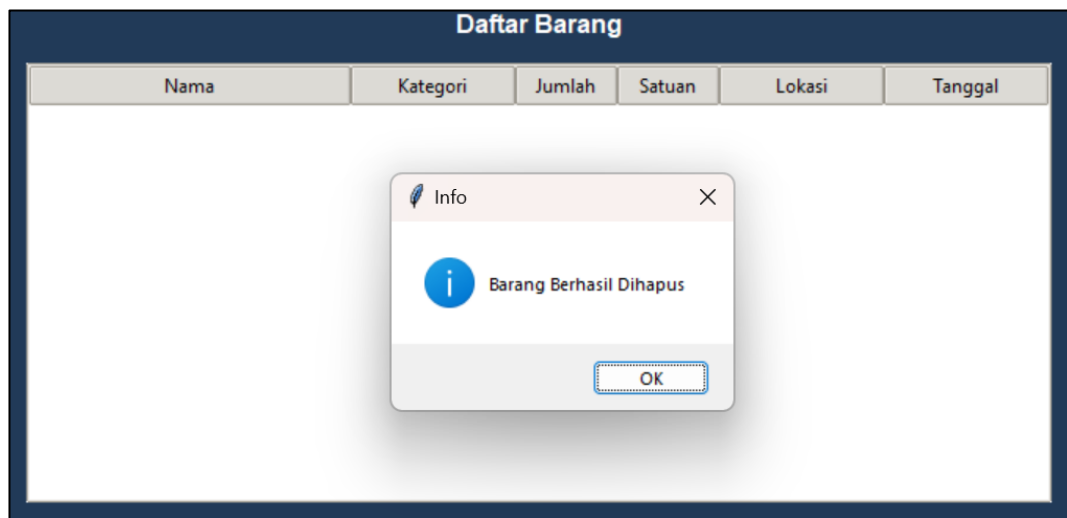
Maka akan muncul notifikasi seperti di atas dan data pun telah diperbarui.

3. Menghapus data dari daftar barang.

Ekspor File
Impor File
Hapus
Edit
Update
Taruh

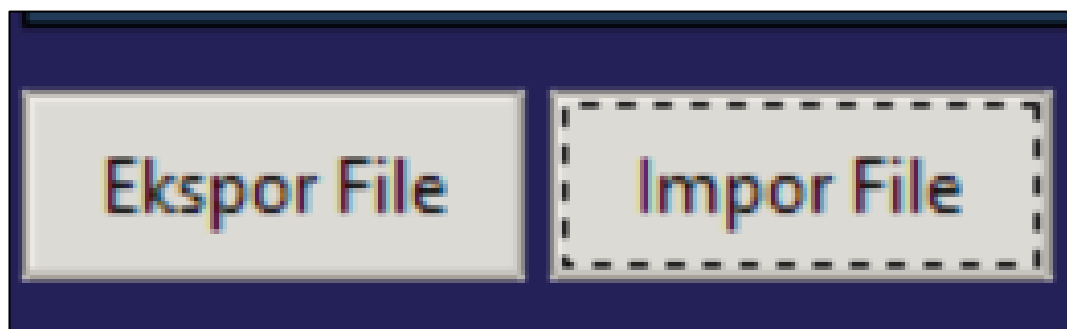
Daftar Barang					
Nama	Kategori	Jumlah	Satuan	Lokasi	Tanggal
Kardus Besar 100x200x300cm	Lainnya	20	Unit	Sekat E	2024-11-28

Pilih data yang hendak dihapus, kemudian klik hapus.

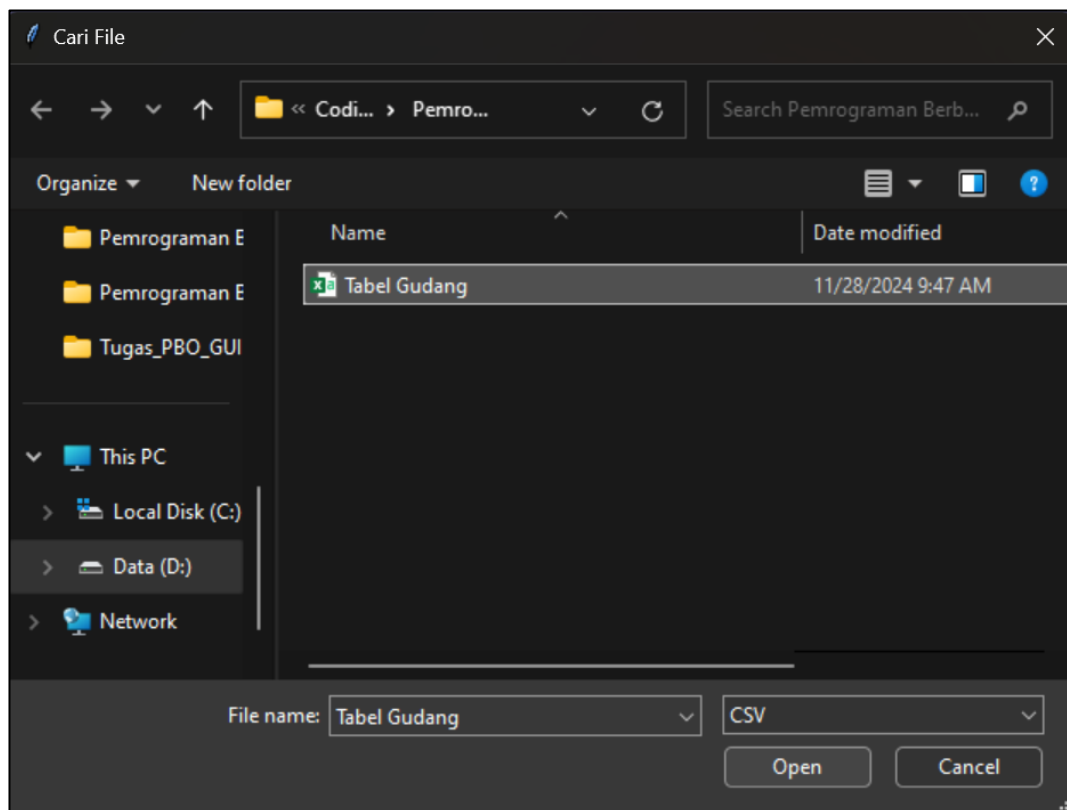


Maka data akan terhapus dan akan muncul notifikasi seperti di atas.

4. Impor File dari Penyimpanan Lokal



Klik tombol impor file.

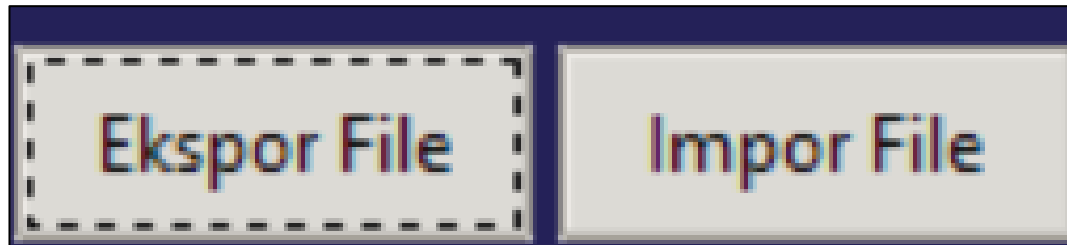


Pilih file yang hendak diimpor, klik open.

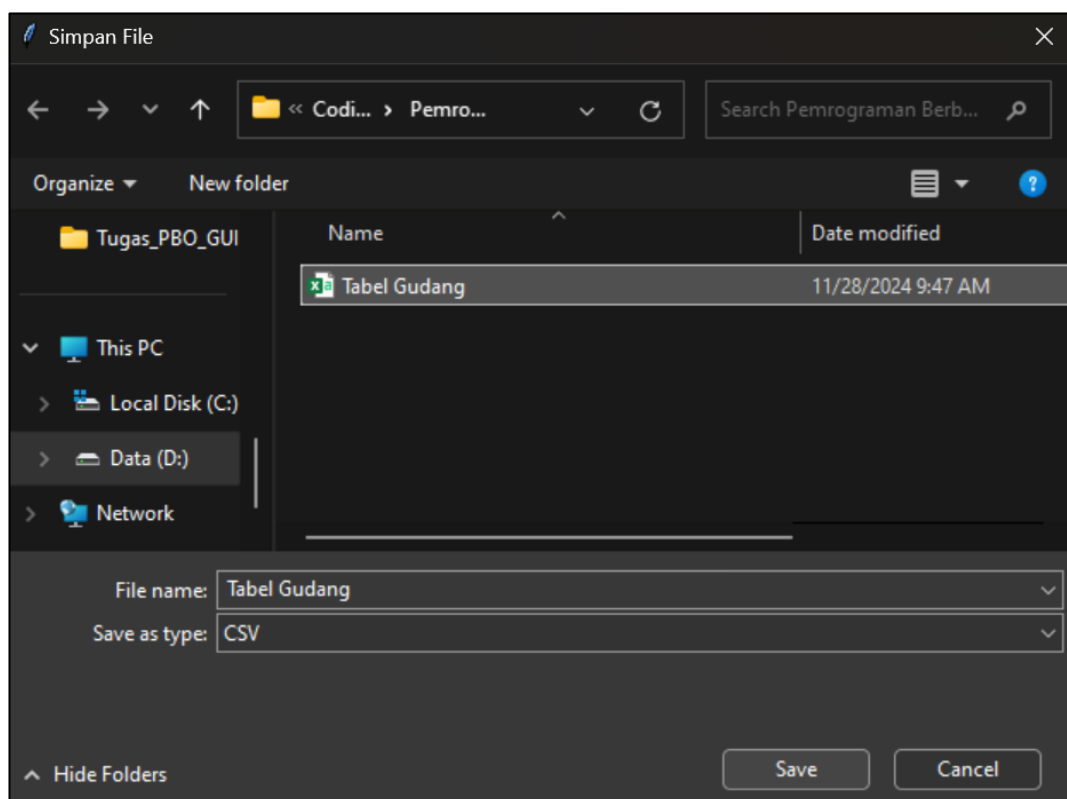


Maka data yang ada di file tersebut akan tampil di tabel GUI dan akan muncul notifikasi seperti di atas.

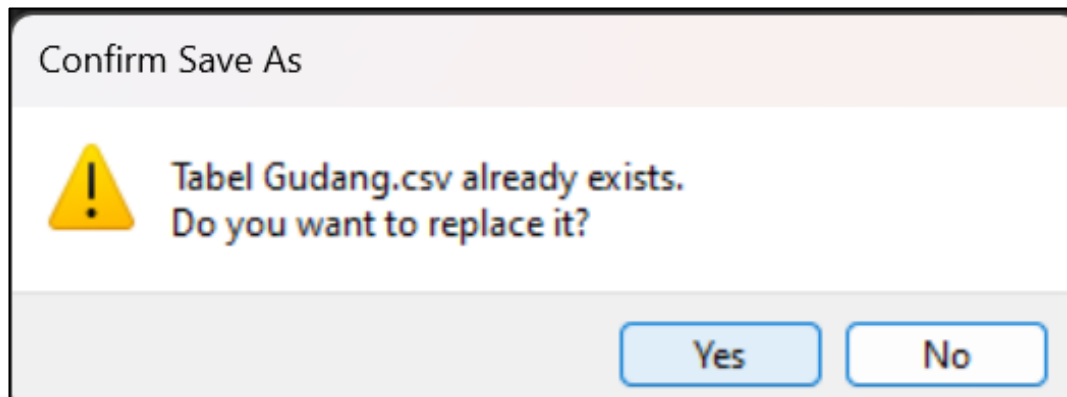
5. Ekspor File ke Penyimpanan Lokal



Klik tombol ekspor file.



Pilih direktori yang untuk menempatkan file, jika ingin menimpa file maka klik file yang hendak ditimpa lalu klik save.



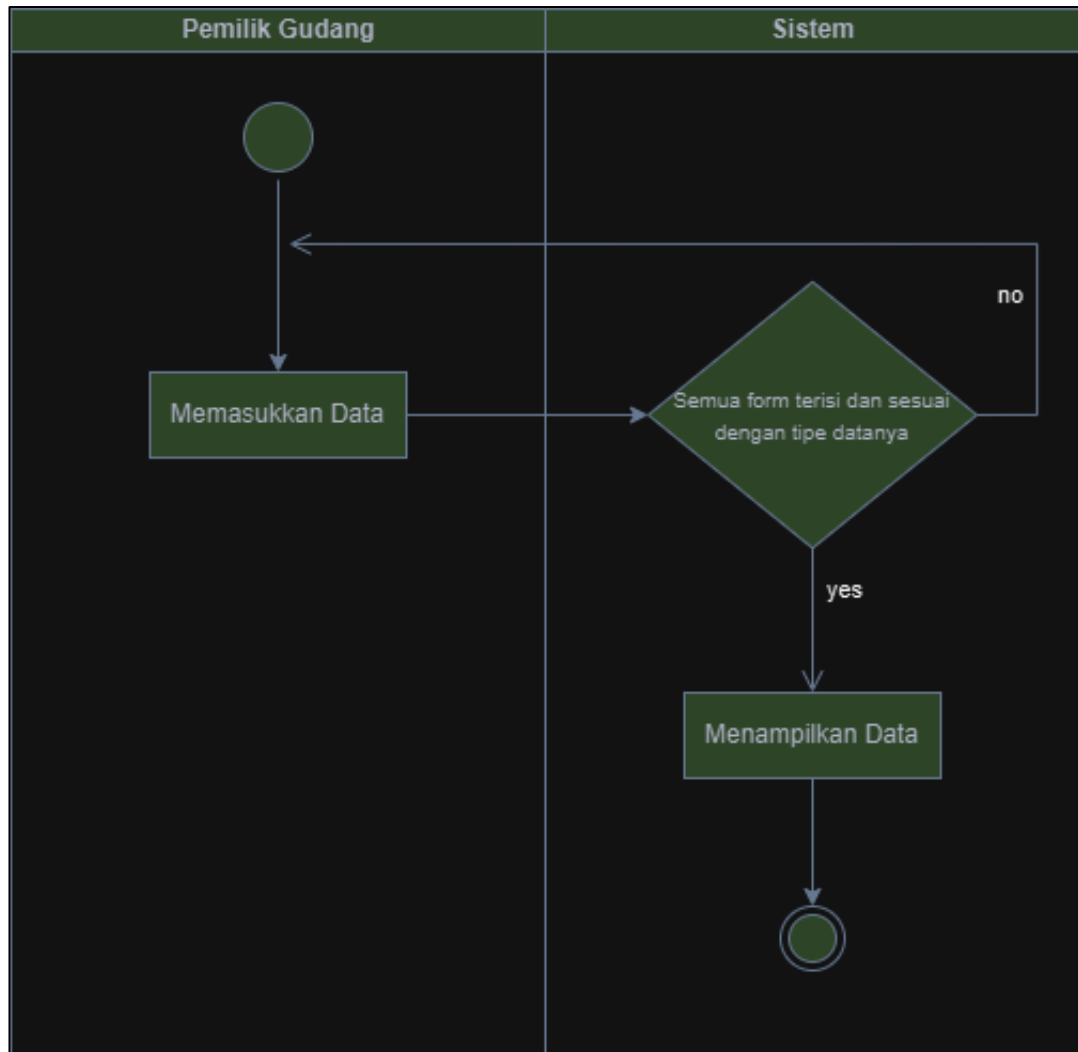
Maka akan muncul warning seperti di atas, jika ingin lanjut menimpa file maka klik yes.



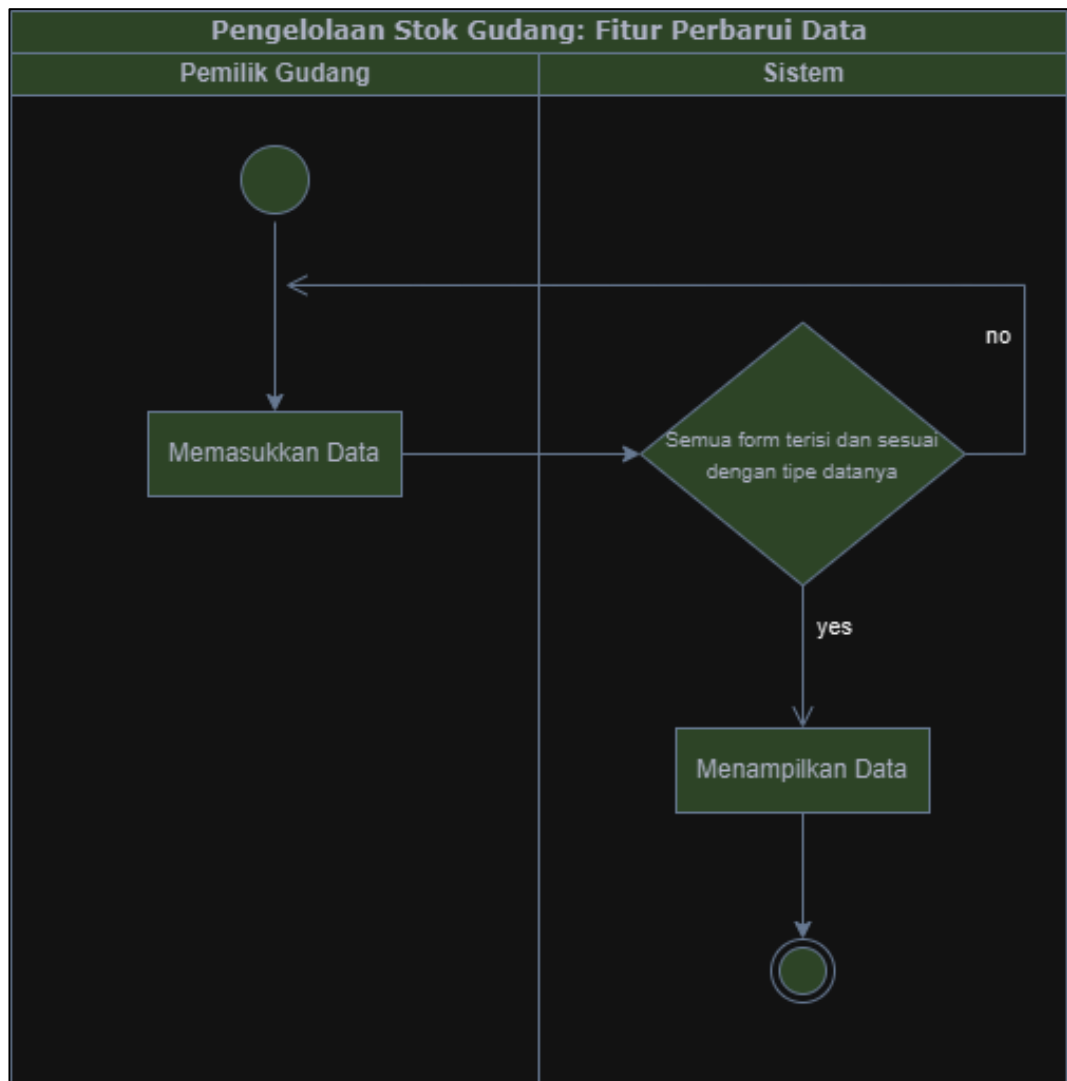
Setelah itu akan muncul notifikasi seperti di atas yang menandakan berhasil ekspor file.

Activity Diagram

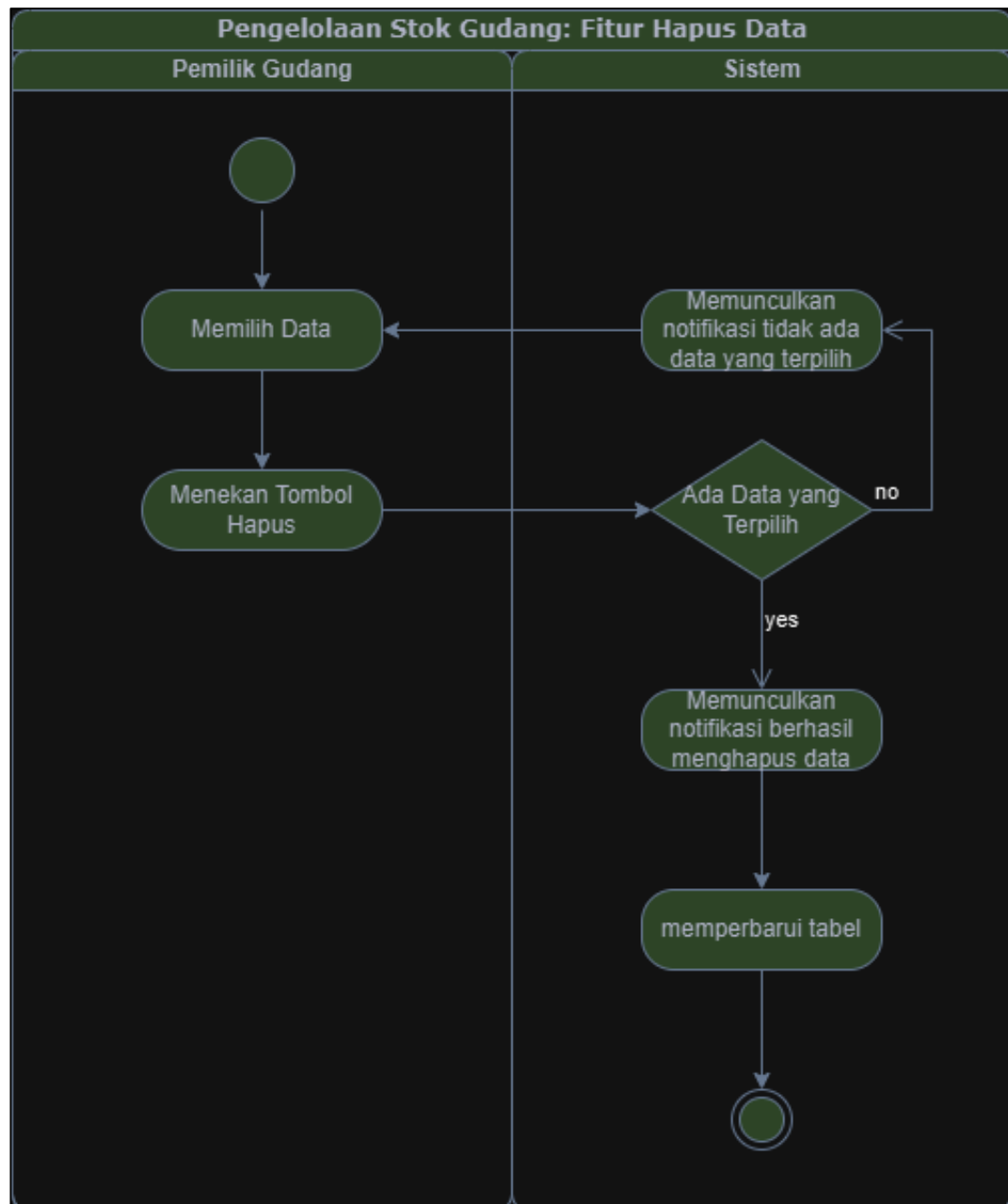
1. Fitur Tambah Data



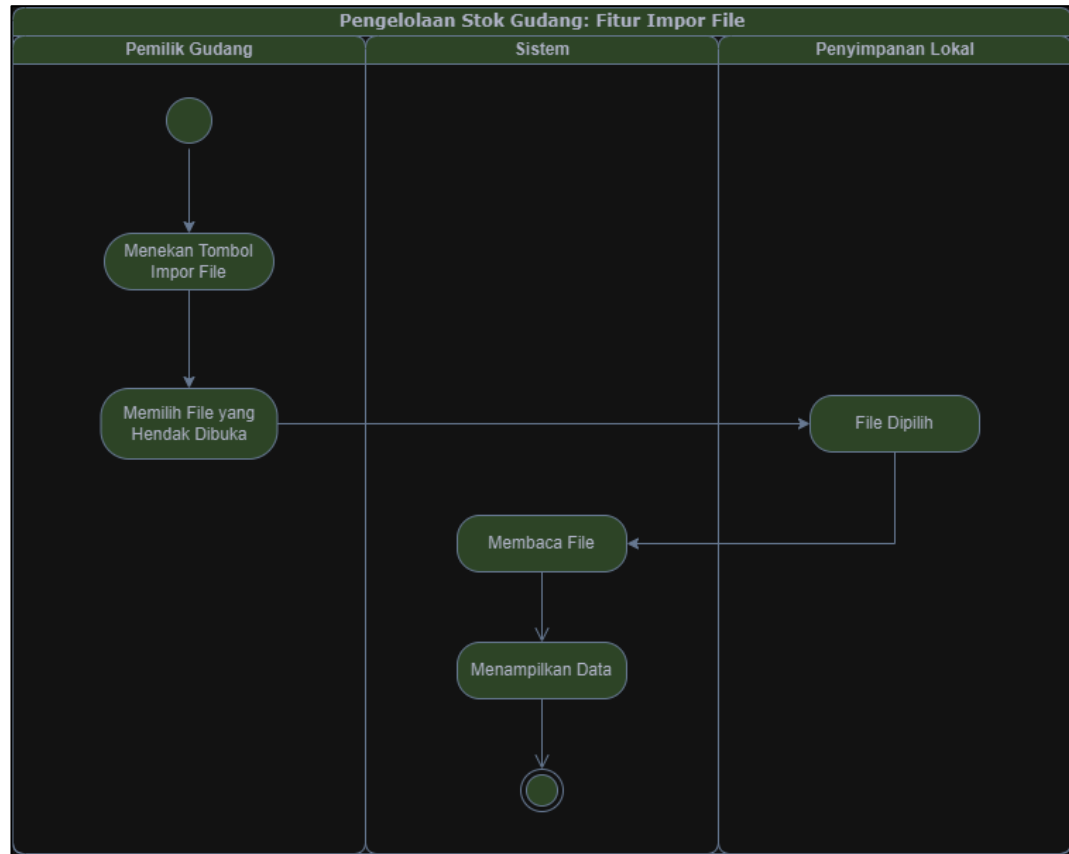
2. Fitur Perbarui Data



3. Fitur Hapus Data



4. Fitur Impor File



5. Fitur Ekspor Data

