**STOCK KING**

Diajukan untuk memenuhi tugas mata kuliah *Algorithm and Programming*



|  |  |
| --- | --- |
| Farrel Najib Anshary | 2201804165 |
| Samuel Jason Santosa | 2201811353 |
| Fernando Gunawan | 2201801081 |
| Calvin Lazuardi | 2201814600 |
| Malvin | 2201799442 |

**PROGRAM STUDI INFORMATIKA**

**STTK BINA NUSANTARA BANDUNG**

**2018**

# **DAFTAR ISI**

[**DAFTAR ISI** 2](#_Toc532753554)

[**BAB 1 PENDAHULUAN** 3](#_Toc532753555)

[**BAB 2 ANALISA DAN PERANCANGAN** 4](#_Toc532753556)

[**2.1.** **Analisa** 4](#_Toc532753557)

[**2.1.1 Analisa Proses** 4](#_Toc532753558)

[**2.1.2 Analisa Struktur Proses** 4](#_Toc532753559)

[**2.2.** **Perancangan** 5](#_Toc532753560)

[**2.2.1 Perancangan Struktur Menu** 5](#_Toc532753561)

[**2.2.2 Perancangan Prosedural** 6](#_Toc532753562)

[**BAB 3 IMPLEMENTASI DAN PENGUJIAN** 14](#_Toc532753563)

[**3.1.** **Implementasi** 15](#_Toc532753564)

[**3.1.1.** **Implementasi Antarmuka** 15](#_Toc532753565)

[**3.1.2.** **Source Code** 15](#_Toc532753566)

[**3.2.** **Pengujian** 15](#_Toc532753567)

[**DAFTAR PUSTAKA** 18](#_Toc532753568)

[Ir. Yuniar Supardi. 2017. Semua Bisa Menjadi Programmer Python Basic. Jakarta: Gramedia. 18](#_Toc532753569)

[**LAMPIRAN** 18](#_Toc532753570)

# **BAB 1 PENDAHULUAN**

Gudang adalah suatu sistem logistik yang di mana berfungsi untuk menyimpan suatu produk dan menyediakan semua informasi stok yang disimpan di gudang dan mudah di akses oleh yang berkepentingan.

Dimana dalam manajemen pergudangan diperlukan saluran distribusi yang baik dari proses pemasukan dan pengeluaran di dalam gudang. Untuk dapat menyimpan barang di dalam gudang , perusahaan atau pengusaha harus mempunyai suatu aplikasi yang dapat menunjang stok, pemasukan barang, pengeluaran barang, dan tata letak barang yang dimana dapat menunjang semua kegiatan dalam gudang.

Stock King adalah aplikasi yang dapat menjawab kebutuhan pemilik gudang. Permasalahan utama para pemilik gudang adalah dalam mengawasi jumlah keluar atau masuk barang agar tidak terjadi kesalahan data. Maka dari itu kami membuat sebuah aplikasi yang kami namakan ‘Stock King’. Kami menawarkan kemudahan dalam beberapa hal dalam aplikasi ini. Contohnya adalah kemudahan dalam memantau kegiatan *input output* barang hingga mengawasi jumlah ketersediaan barang di dalam gudang, dan juga kemudahan dalam memantau ketersediaan stok barang di dalam gudang.

# **BAB 2 ANALISA DAN PERANCANGAN**

## **Analisa**

Analisa adalah proses mengidentifikasi permasalahan, hambatan, serta cara untuk mengetahui kebutuhan sistem yang akan dibangun.

### **2.1.1 Analisa Proses**

Sebuah aplikasi gudang yang baik adalah aplikasi yang dapat memantau seluruh pemasukan , pengeluaran barang dan juga dapat memantau stok barang yang di gudang.

Pemasukan barang adalah proses untuk (meng-input) barang yang akan di masukkan ke gudang dan diletakan sesuai kategori barang.

Pengeluaran barang adalah proses untuk (meng-output) barang yang akan di keluarkan dari gudang dan secara otomatis mengurangi stok di gudang.

Memantau stok adalah di mana kegiatan supervisor / kepala gudang memantau ketersediaan stok di gudang, dan juga bisa memantau di mana letak suatu barang yang di atur saat pemasukan barang.

### **2.1.2 Analisa Struktur Proses**

Struktur proses dari Stock King ini digambarkan seperti pada Gambar

## **Perancangan**

Perancangan adalah proses pengembangan spesifikasi sistem baru berdasarkan hasil analisis.

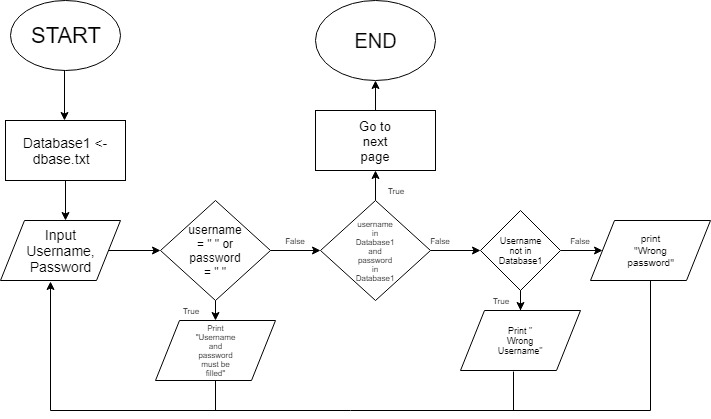
### **2.2.1 Perancangan Struktur Menu**

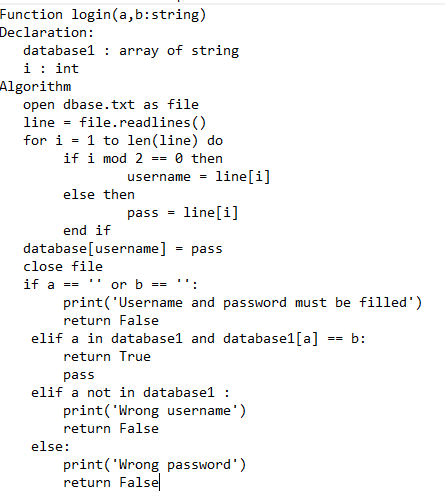
Adapun struktur antarmuka untuk Stock King adalah sebagai berikut.

### 

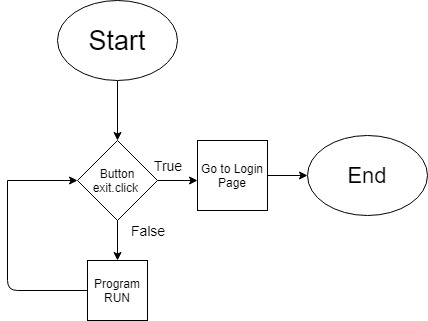
### **2.2.2 Perancangan Prosedural**

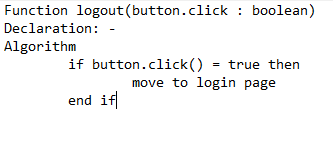
#### 2.2.2.1 Login



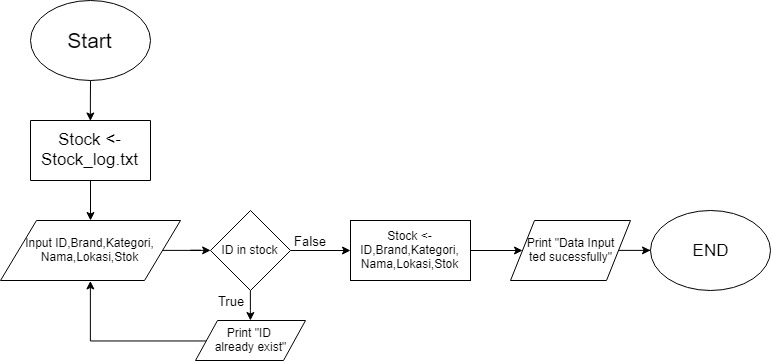


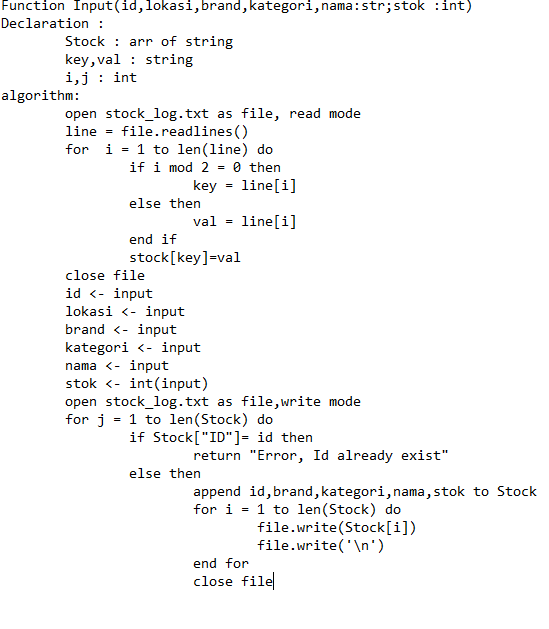
#### 2.2.2.2 Logout



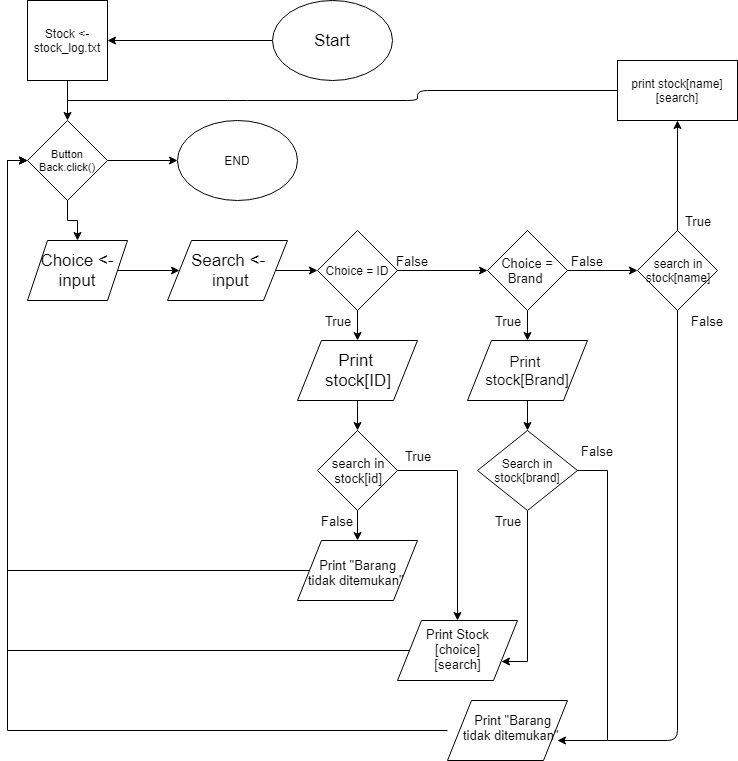


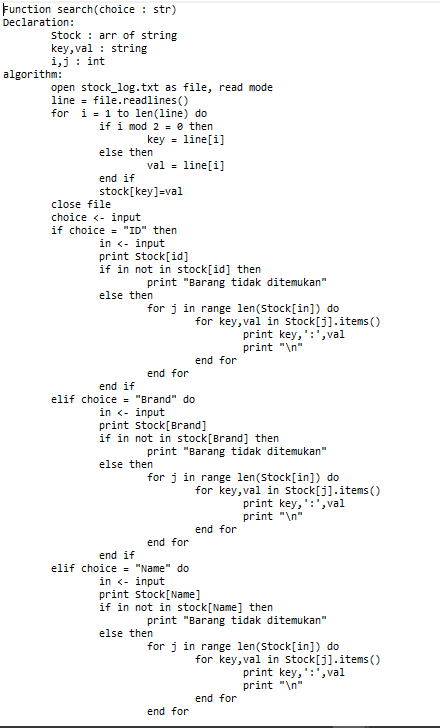
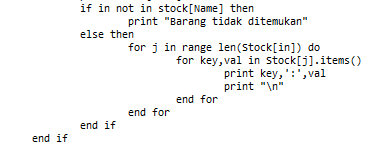
2.2.2.3 Input



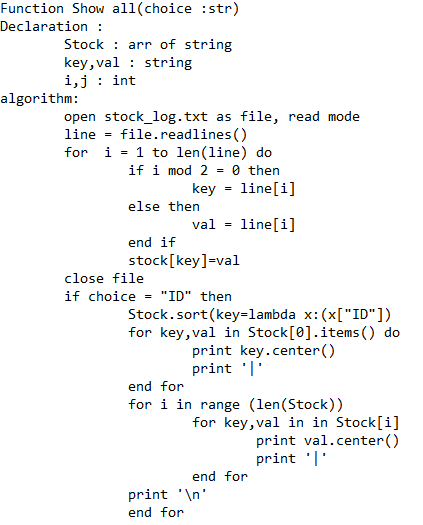
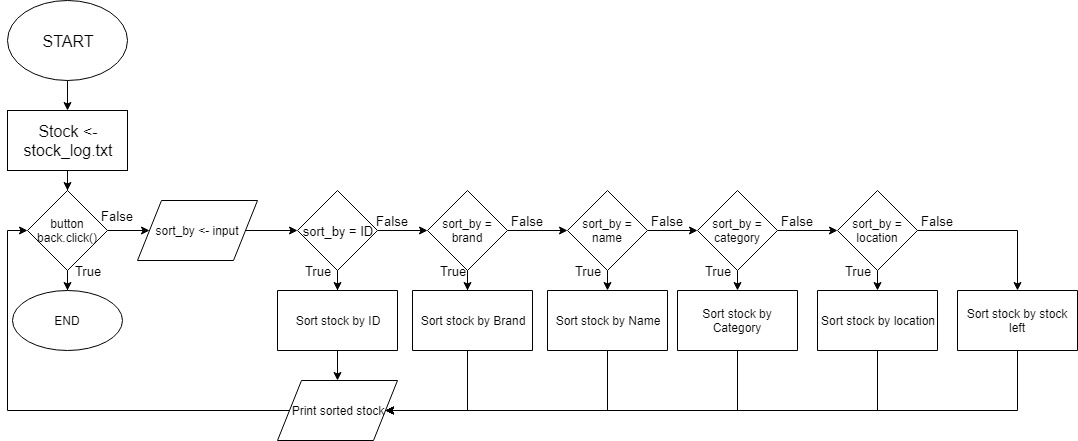


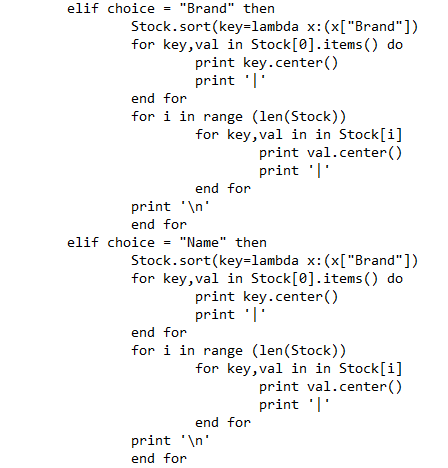
#### 2.2.2.4 Search Item

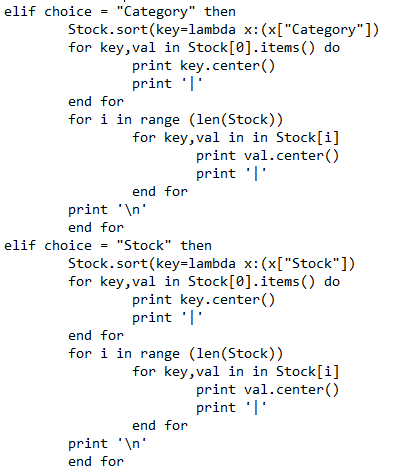


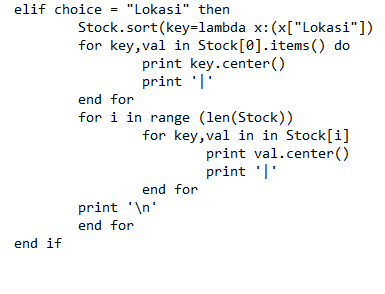


#### 2.2.2.5 Show All Item

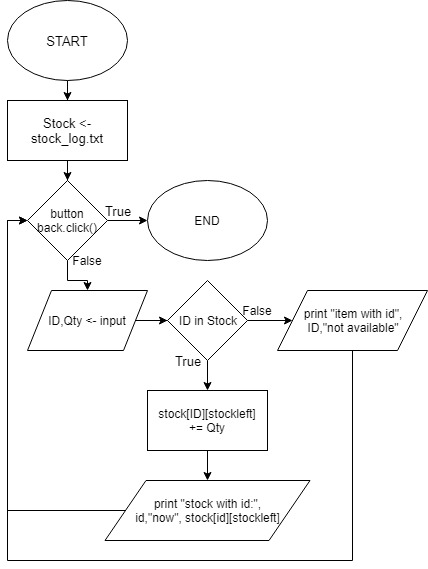


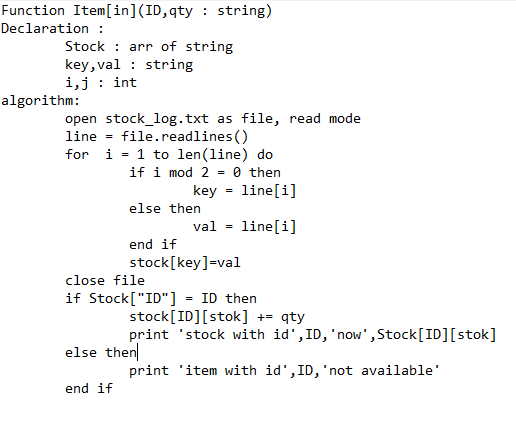




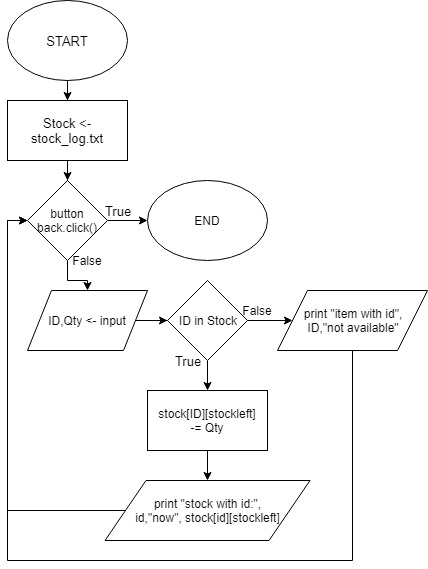


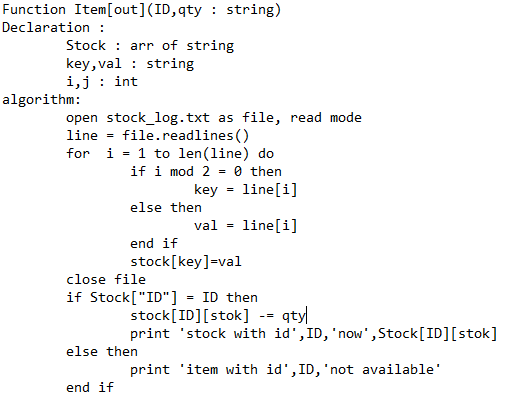
#### 2.2.2.6 Item[IN]



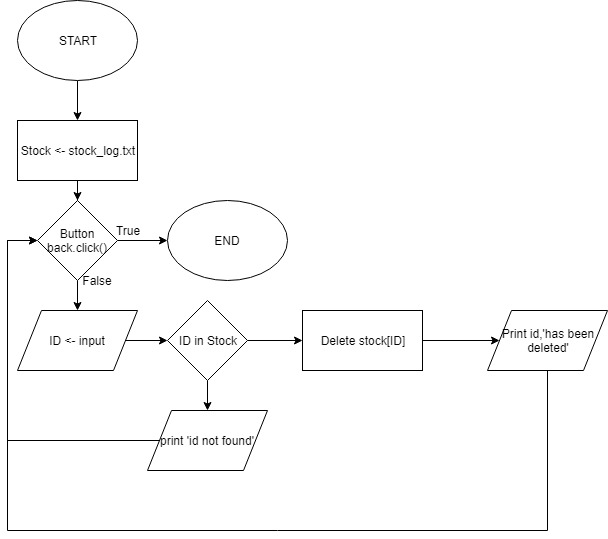


#### 2.2.2.7 Item[OUT]





#### 2.2.2.8 Delete Item



FalseFunction Item[out](ID,qty : string)

Declaration :

Stock : arr of string

key,val : string

i,j : int

algorithm:

open stock\_log.txt as file, read mode

line = file.readlines()

for i = 1 to len(line) do

if i mod 2 = 0 then

key = line[i]

else then

val = line[i]

end if

stock[key]=val

close file

if Stock["ID"] = ID then

stock[ID][stok] -= qty

print 'stock with id',ID,'now',Stock[ID][stok]

else then

print 'item with id',ID,'not available'

end ifunction Item[out](ID,qty : string)

Declaration :

Stock : arr of string

key,val : string

i,j : int

algorithm:

open stock\_log.txt as file, read mode

line = file.readlines()

for i = 1 to len(line) do

if i mod 2 = 0 then

key = line[i]

else then

val = line[i]

end if

stock[key]=val

close file

if Stock["ID"] = ID then

stock[ID][stok] -= qty

print 'stock with id',ID,'now',Stock[ID][stok]

else then

print 'item with id',ID,'not available'

end ife

TrueFunction Item[out](ID,qty : string)

Declaration :

Stock : arr of string

key,val : string

i,j : int

algorithm:

open stock\_log.txt as file, read mode

line = file.readlines()

for i = 1 to len(line) do

if i mod 2 = 0 then

key = line[i]

else then

val = line[i]

end if

stock[key]=val

close file

if Stock["ID"] = ID then

stock[ID][stok] -= qty

print 'stock with id',ID,'now',Stock[ID][stok]

else then

print 'item with id',ID,'not available'

end ifunction Item[out](ID,qty : string)

Declaration :

Stock : arr of string

key,val : string

i,j : int

algorithm:

open stock\_log.txt as file, read mode

line = file.readlines()

for i = 1 to len(line) do

if i mod 2 = 0 then

key = line[i]

else then

val = line[i]

end if

stock[key]=val

close file

if Stock["ID"] = ID then

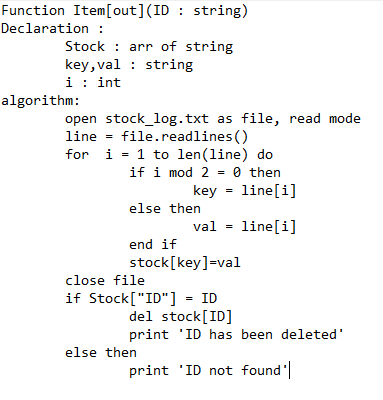
stock[ID][stok] -= qty

print 'stock with id',ID,'now',Stock[ID][stok]

else then

print 'item with id',ID,'not available'

end ife



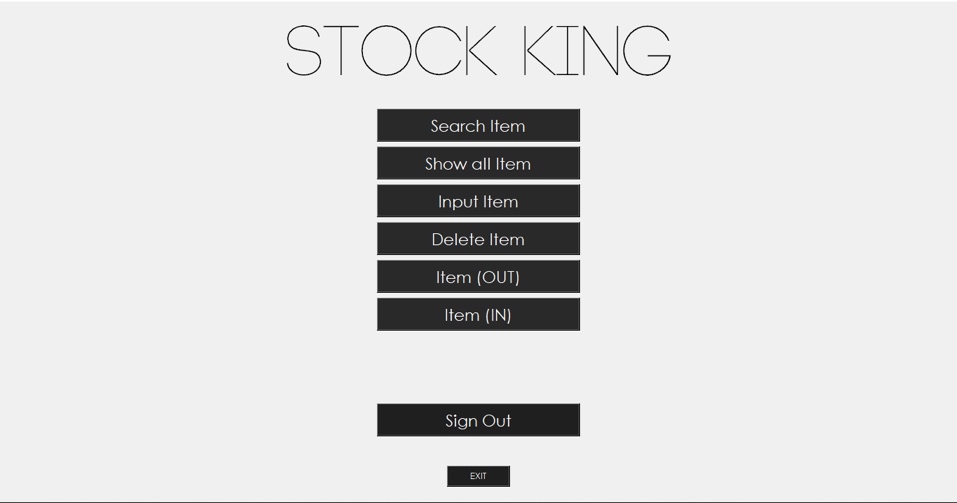
# **BAB 3 IMPLEMENTASI DAN PENGUJIAN**

## **Implementasi**

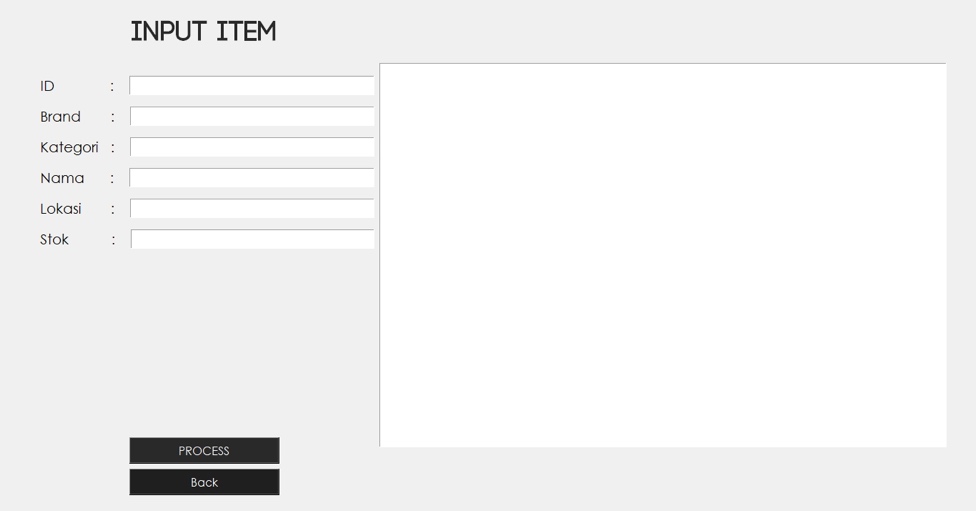
Implementasi merupakan kegiatan penerapan program yang telah dibuat berdasarkan hasil perancangan.

### **Implementasi Antarmuka**

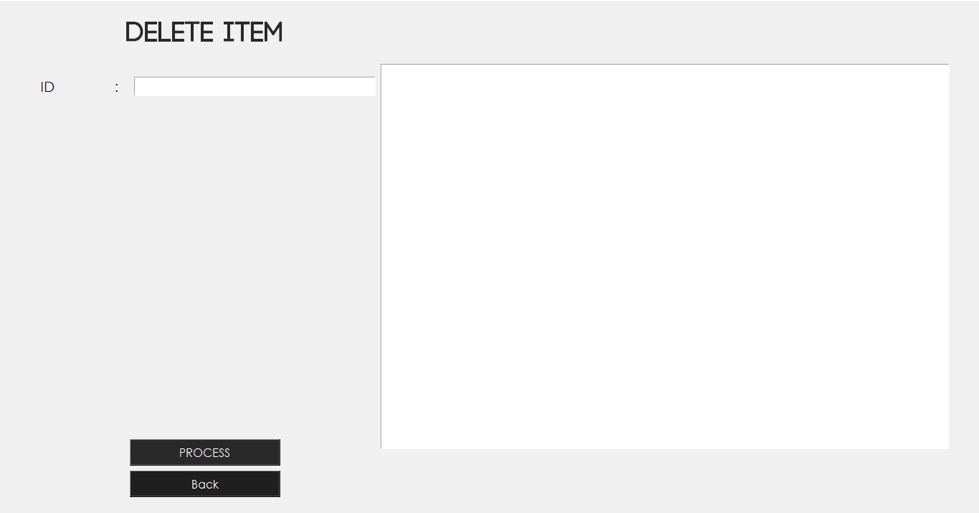
🡪 Main Menu



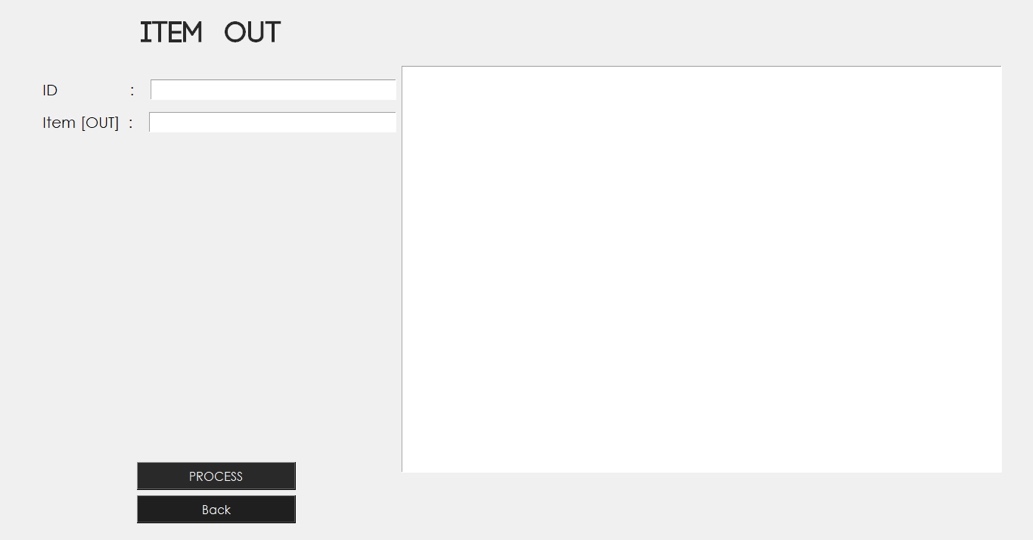
🡪 Input Item



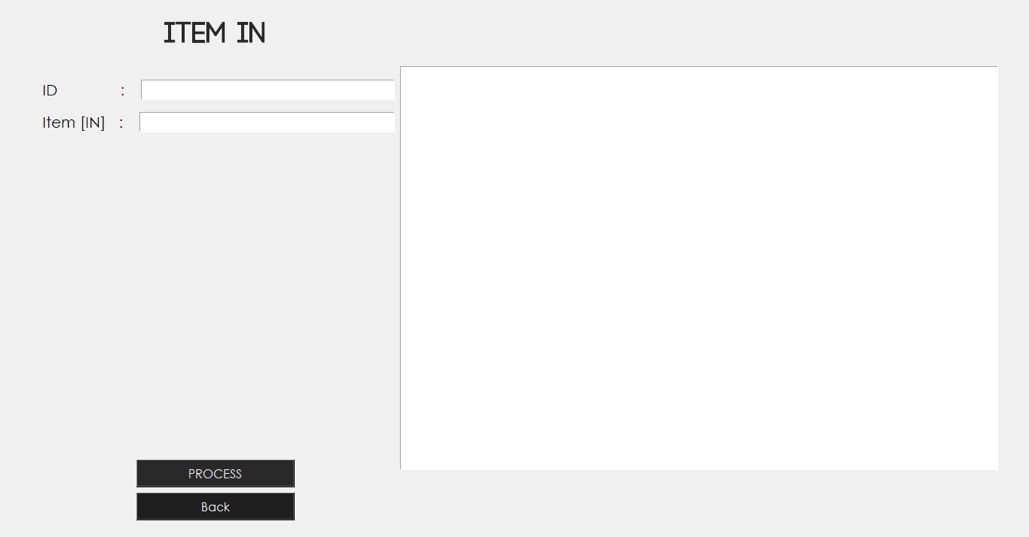
🡪 Delete Item



🡪 Item Out



🡪 Item In



#### 

### **Source Code**

Source code disertakan di lampiran

## **Pengujian**

Pengujian adalah proses untuk menjamin bahwa sistem yang dibangun telah sesuai dengan rancangan yang dibuat. Pengujian menggunakan metode GUI. Hasil pengujian dapat dilihat pada Tabel

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Menu** | **Pengujian** | **Hasil** | |
| 1 | Halaman Login | Masukkan ID dan Pass lalu pilih Log In | Masuk ke halaman utama | |
| Pilih menu ‘Exit to Desktop’ | Program berhenti | |
| 2 | Halaman utama | Pilih ‘Search Item’ | Masuk ke halaman ‘Search Item’ | |
| Pilih ‘Show All Items’ | Masuk ke halaman ‘Show All Items’ | |
| Pilih ‘Input Item’ | Masuk ke halaman ‘Input Item’ | |
| Pilih 'Delete Item’ | Masuk ke halaman 'Delete Item’ | |
| Pilih ‘Item [Out]’ | Masuk ke halaman ‘Item [Out]’ | |
| Pilih ‘Item [In]’ | Masuk ke halaman ‘Item [In]’ | |
| Pilih ‘Exit’ | Program berhenti | |
| 3 | Seach Item | Pilih ‘Search by Kategori’ kemudian ketik kategori yang ingin dicari lalu klik ‘Search’ | Program akan menampilkan kategori yang ada di dalam database dan kemudian pilihan ‘Search by’ berubah menjadi ‘Search by Brand’ | |
| Pilih ‘Search by Brand’ kemudian ketik merek yang ingin dicari lalu klik ‘Search’ | Program akan menampilkan merek yang ada di dalam database dan kemudian pilihan ‘Search by’ berubah menjadi ‘Search by ID’ | |
| Pilih ‘Search by ID’ kemudian ketik ID dari item yang ingin dicari lalu klik ‘Search’ | Program akanMenampilkan detail item spesifik dari ID yang dicari | |
| Pilih ‘Exit’ | Program berhenti | |
| 4 | Show All Item | Pilih ‘Sort by ID’ lalu klik ‘CONFIRM’ | Program akan menampilkan semua item yang ada di database beserta detailnya yang diurutkan berdasarkan ID | |
| Pilih ‘Sort by BRAND’ lalu klik ‘CONFIRM’ | Program akan menampilkan semua item yang ada di database beserta detailnya yang diurutkan berdasarkan merek | |
| Pilih ‘Sort by Kategori’ lalu klik ‘CONFIRM’ | Program akan menampilkan semua item yang ada di database beserta detailnya yang diurutkan berdasarkan kategori barnag | |
| Pilih ‘Sort by Nama’ lalu klik ‘CONFIRM’ | Program akan menampilkan semua item yang ada di database beserta detailnya yang diurutkan berdasarkan nama lengkapnya | |
| Pilih ‘Sort by Lokasi’ lalu klik ‘CONFIRM’ | Program akan menampilkan semua item yang ada di database beserta detailnya yang diurutkan berdasarkan lokasi | |
| Pilih ‘Sort by Stok’ lalu klik ‘CONFIRM’ | Program akan menampilkan semua item yang ada di database beserta detailnya yang diurutkan berdasarkan ketersediaan stok | |
| Pilih ‘Back’ | Kembali ke halaman utama | |
| 5 | Input Item | Masukkan ID, merek, kategori, nama, lokasi, dan stok barang yang baru masuk tetapi belum terdaftar dalam database lalu klik ‘PROCESS’ | Data yang dimasukkan akan tersimpan ke dalam database | |
| Pilih ‘Back’ | Kembali ke halaman utama | |
| 6 | Delete Item | Masukkan ID dari barang yang ingin dihapus dari database kemudian klik ‘PROCESS’ | Data dari ID yang dimasukkan akan terhapus dari database |
| Pilih ‘Back’ | Kembali ke halaman utama |
| 7 | Item [OUT] | Masukkan ID dari barang yang akan dikurangi jumlah stoknya lalu masukkan jumlah barang yang berkurang dari stok lalu klik ‘PROCESS’ | Jumlah stok dalam database akan berkurang sesuai dengan angka yang diinputkan user |
| Pilih ‘Back’ | Kembali ke halaman utama |
| 8 | Item [IN] | Masukkan ID dari barang yang akan ditambah jumlah stoknya lalu masukkan jumlah barang yang bertambah dari stok lalu klik ‘PROCESS’ | Jumlah stok dalam database akan bertambah sesuai dengan angka yang diinputkan user |
| Pilih ‘Back’ | Kembali ke halaman utama |

**DAFTAR PUSTAKA**

Yuniar Supardi. 2017. Semua Bisa Menjadi Programmer Python Basic. Jakarta: Gramedia.

# 

# 

**LAMPIRAN**

**from tkinter import \***

**import tkinter.messagebox as tm**

**from tkinter.font import Font**

**import string**

**#import search**

**class FullScreenApp(object):**

**def \_\_init\_\_(self, master, \*\*kwargs):**

**self.master=master**

**pad=3**

**self.\_geom='200x200+0+0'**

**master.geometry("{0}x{1}+0+0".format(**

**master.winfo\_screenwidth()-pad, master.winfo\_screenheight()-pad))**

**master.bind('<Escape>',self.toggle\_geom)**

**def toggle\_geom(self,event):**

**geom=self.master.winfo\_geometry()**

**print(geom,self.\_geom)**

**self.master.geometry(self.\_geom)**

**self.\_geom=geom**

**class SampleApp(Tk):**

**def \_\_init\_\_(self, \*args, \*\*kwargs):**

**Tk.\_\_init\_\_(self, \*args, \*\*kwargs)**

**container = Frame(self)**

**container.pack(side="top", expand=True)**

**container.grid\_rowconfigure(0, weight=1)**

**container.grid\_columnconfigure(0, weight=1)**

**self.frames = {}**

**for F in (LoginFrame, MainFrame, Search, ShowAll, InputItem, BarangKeluar, BarangMasuk, DeleteItem):**

**page\_name = F.\_\_name\_\_**

**frame = F(parent=container, controller=self)**

**self.frames[page\_name] = frame**

**frame.grid(row=0, column=0, sticky="nsew")**

**self.show\_frame("LoginFrame")**

**def show\_frame(self, page\_name):**

**frame = self.frames[page\_name]**

**frame.tkraise()**

**class LoginFrame(Frame):**

**def \_\_init\_\_(self, parent, controller):**

**Frame.\_\_init\_\_(self, parent)**

**self.controller = controller**

**label1= Label(self, text="Stock-King", font=("Code-Light", 75)).pack(side="top")**

**Label(self, text=" ", font=("Century Gothic", 20)).pack(side="top")**

**label2= Label(self, text="Account Name", font=("Century Gothic", 15)).pack(side="top")**

**self.myvar1=StringVar()**

**username = Entry(self, width=20, font=("Century Gothic", 20), textvariable=self.myvar1).pack(side="top")**

**Label(self, text=" ", font=("Century Gothic", 20)).pack(side="top")**

**label3 = Label(self, text="Password", font=("Century Gothic", 15)).pack(side="top")**

**self.myvar2=StringVar()**

**password = Entry(self, width=20, font=("Century Gothic", 20), show='\*', textvariable=self.myvar2).pack(side="top")**

**Label(self, text=" ", font=20).pack()**

**Label(self, text=" ", font=20).pack()**

**button1= Button(self, text="LOGIN", font=("Code-Light", 20), background="gray12", foreground="White", command=self.login1).pack(side=TOP)**

**Label(self, text=" ", font=("Century Gothic", 20)).pack(side="top")**

**button2 = Button(self, text="Exit Dekstop", font=("Century Gothic", 10), background="gray12", foreground="White",**

**command=self.quit).pack(side="top")**

**def login1(self):**

**a= self.myvar1.get()**

**b= self.myvar2.get()**

**file = open('dbase.txt', 'r+')**

**database1 = {}**

**username = password = ''**

**line = file.readlines()**

**for i in range(len(line)):**

**if i % 2 == 0:**

**username = line[i].rstrip()**

**else:**

**password = line[i].rstrip()**

**database1[username] = password**

**file.close()**

**if a == '' or b == '':**

**tm.showerror('Fill in the Blank','Username and password must be filled')**

**return False**

**elif a in database1 and database1[a] == b:**

**self.controller.show\_frame("MainFrame")**

**return True**

**elif a not in database1:**

**tm.showerror('Wrong username', "Incorrect Username")**

**return False**

**else:**

**tm.showerror('Wrong password', "Incorrect Password")**

**return False**

**class MainFrame(Frame):**

**def \_\_init\_\_(self, parent, controller):**

**Frame.\_\_init\_\_(self, parent)**

**self.controller = controller**

**x = Font(family="Century Gothic", size=17)**

**Label(self, text="Stock-King", font=("Code-Light", 75)).pack(side="top")**

**Label(self, text=" ", font=("Code-Light")).pack(side="top")**

**Button(self, text='Search Item', width=20, font=x, command=lambda: controller.show\_frame("Search"),background="gray16",foreground="White").pack(side=TOP)**

**Label(self, text=" ", font=("Code-Light",1)).pack(side="top")**

**Button(self, text="Show all Item", width=20, font=x, command= lambda: controller.show\_frame("ShowAll"),background="gray16",foreground="White").pack(side=TOP)**

**Label(self, text=" ", font=("Code-Light", 1)).pack(side="top")**

**Button(self, text="Input Item", width=20, font=x, command=lambda: controller.show\_frame("InputItem"),background="gray16",foreground="White").pack(**

**side=TOP)**

**Label(self, text=" ", font=("Code-Light", 1)).pack(side="top")**

**Button(self, text="Delete Item", width=20, font=x, command=lambda: controller.show\_frame("DeleteItem"),background="gray16",foreground="White").pack(**

**side=TOP)**

**Label(self, text=" ", font=("Code-Light", 1)).pack(side="top")**

**Button(self, text="Item (OUT)", width=20, font=x, command=lambda: controller.show\_frame("BarangKeluar"),background="gray16",foreground="White").pack(**

**side=TOP)**

**Label(self, text=" ", font=("Code-Light", 1)).pack(side="top")**

**Button(self, text="Item (IN)", width=20, font=x, command=lambda: controller.show\_frame("BarangMasuk"),background="gray16",foreground="White").pack(**

**side=TOP)**

**Button(self, text='EXIT', font=("Century Gothic", 10),background="gray12",foreground="White", width=10, command=self.quit).pack(side=BOTTOM)**

**Label(self, text=" ").pack(side=BOTTOM)**

**Label(self, text=" ").pack(side=BOTTOM)**

**Button(self, text='Sign Out', font=("Century Gothic",17),background="gray12",foreground="White",width=20, command=lambda: controller.show\_frame("LoginFrame")).pack(side=BOTTOM)**

**Label(self, text=" ").pack(side=BOTTOM)**

**Label(self, text=" ").pack(side=BOTTOM)**

**Label(self, text=" ").pack(side=BOTTOM)**

**class Search(Frame):**

**def \_\_init\_\_(self, parent, controller):**

**Frame.\_\_init\_\_(self, parent)**

**self.controller = controller**

**self.varoption=StringVar()**

**self.varoption.set('')**

**button2 = Button(self, text='Back', width=10, font=("Century Gothic",12),background="gray12",foreground="White", command=self.back).pack(side=TOP)**

**Label(self, text=' ').pack(side=TOP)**

**Label(self, text='CATEGORY',font=("Century Gothic",20)).pack(side=TOP)**

**option=OptionMenu(self, self.varoption, "ID", "Brand", "Kategori").pack(side=TOP)**

**Label(self, text="Input base of your choice", font=("Century Gothic", 15)).pack(side="top")**

**self.myvar1 = StringVar()**

**self.searching = Entry(self, width=20, font=("Century Gothic", 20), textvariable=self.myvar1)**

**self.searching.pack(side="top")**

**Label(self, text=' ').pack(side=TOP)**

**self.button1 = Button(self, text="SEARCH",background="gray16",foreground="White", font=("Century Gothic", 17),**

**command=self.search12).pack(side="top")**

**Label(self, text=' ').pack(side=TOP)**

**self.kotak = Text(master=self)**

**self.kotak.pack(side=BOTTOM)**

**opsi = self.varoption.get()**

**def search12(self):**

**self.search()**

**self.searching.delete(0, END)**

**def back(self):**

**self.varoption.set('')**

**self.kotak.delete('1.0',END)**

**self.searching.delete(0, END)**

**self.controller.show\_frame('MainFrame')**

**def search(self):**

**file = open("stock\_log.txt", "r+")**

**self.stok = []**

**line = file.readlines()**

**key = val = ""**

**temp = {}**

**for i in range(len(line)):**

**if i % 2 == 0:**

**key = line[i].rstrip()**

**else:**

**val = line[i].rstrip()**

**temp[key] = val**

**if (i + 1) % 12 == 0:**

**self.stok.append(temp)**

**temp = {}**

**file.close()**

**opsi=self.varoption.get()**

**if opsi == 'ID':**

**self.kotak.delete('1.0', END)**

**inputan = self.myvar1.get().upper()**

**self.kotak.insert(END, "ID yang Tersedia: \n")**

**for i in range(len(self.stok)):**

**self.kotak.insert(END,self.stok[i]["ID"])**

**self.kotak.insert(END, '\n')**

**# self.kotak.insert(END,"Barang ditemukan! \n")**

**self.searchID(inputan)**

**elif opsi == "Brand":**

**self.kotak.delete('1.0', END)**

**brand = self.myvar1.get().upper()**

**brandunik = []**

**self.kotak.insert(END, "Brand yang Tersedia: \n")**

**for i in range(len(self.stok)):**

**if self.stok[i]["BRAND"] not in brandunik:**

**self.kotak.insert(END,self.stok[i]["BRAND"])**

**self.kotak.insert(END, '\n')**

**brandunik.append(self.stok[i]["BRAND"])**

**self.kotak.insert(END,"ID Barang dengan Brand ")**

**self.kotak.insert(END, brand)**

**self.kotak.insert(END, " :\n")**

**self.searchBrand(brand)**

**elif opsi == "Kategori":**

**self.kotak.delete('1.0', END)**

**kategori = self.myvar1.get().upper()**

**kategoriunik = []**

**self.kotak.insert(END, "Kategori yang Tersedia: \n")**

**for i in range(len(self.stok)):**

**if self.stok[i]["Kategori"] not in kategoriunik:**

**self.kotak.insert(END,self.stok[i]["Kategori"])**

**self.kotak.insert(END, '\n')**

**kategoriunik.append(self.stok[i]["Kategori"])**

**self.kotak.insert(END,"Brand barang dalam kategori ")**

**self.kotak.insert(END, kategori)**

**self.kotak.insert(END, " :\n")**

**self.searchKategori(kategori)**

**else:**

**tm.showerror("ERROR","Please choose one")**

**def searchID(self, inputan):**

**ditemukan = False**

**for j in range(len(self.stok)):**

**if self.stok[j]["ID"] == inputan:**

**ditemukan = True**

**if ditemukan == True:**

**self.kotak.insert(END, "Barang ditemukan! \n")**

**for key, val in self.stok[j].items():**

**self.kotak.insert(END, (key, ":", val))**

**self.kotak.insert(END, '\n')**

**mereksama = self.stok[j]["BRAND"]**

**break**

**if ditemukan == False:**

**self.kotak.insert(END, "Barang tidak ditemukan\n")**

**tm.showinfo('OOPS', 'Not Found!!')**

**return 0**

**self.kotak.insert(END, "ID dengan brand yang sama :\n")**

**adasama = False**

**L = 1**

**for k in range(len(self.stok)):**

**if k == j:**

**continue**

**else:**

**if self.stok[k]["BRAND"] == mereksama:**

**adasama = True**

**self.kotak.insert(END,("%d." % (L), self.stok[k]["ID"]))**

**self.kotak.insert(END, '\n')**

**L += 1**

**if adasama == False:**

**self.kotak.insert(END, "-")**

**def searchKategori(self, kategori):**

**ditemukan = False**

**brandunik = []**

**for j in range(len(self.stok)):**

**if self.stok[j]["Kategori"] == kategori and self.stok[j]["BRAND"] not in brandunik:**

**self.kotak.insert(END,self.stok[j]["BRAND"])**

**self.kotak.insert(END, '\n')**

**brandunik.append(self.stok[j]["BRAND"])**

**ditemukan = True**

**if ditemukan == False:**

**self.kotak.insert(END,"Barang tidak ditemukan")**

**self.kotak.insert(END, '\n')**

**tm.showinfo('OOPS', 'Not Found!!')**

**return 0**

**else:**

**self.varoption.set('Brand')**

**def searchBrand(self, brand):**

**ditemukan = False**

**for j in range(len(self.stok)):**

**if self.stok[j]["BRAND"] == brand:**

**self.kotak.insert(END,self.stok[j]["ID"])**

**self.kotak.insert(END, '\n')**

**ditemukan = True**

**if ditemukan == False:**

**self.kotak.insert(END,"Barang tidak ditemukan")**

**self.kotak.insert(END, '\n')**

**tm.showinfo('OOPS', 'Not Found!!')**

**return 0**

**else:**

**self.varoption.set('ID')**

**class ShowAll(Frame):**

**def \_\_init\_\_(self, parent, controller):**

**Frame.\_\_init\_\_(self, parent)**

**self.controller = controller**

**button2=Button(self, text="Back", width=10, font=("Century Gothic",12),background="gray12",foreground="White", command=self.back).pack(side=TOP)**

**Label(self, text=' ').pack(side=TOP)**

**self.kotak1=Text(master=self, width=160)**

**self.kotak1.pack(side=BOTTOM)**

**self.option = StringVar()**

**self.option.set('')**

**Label(self, text=' ').pack(side=TOP)**

**Label(self, text='SORT BY', font=("Century Gothic",18)).pack(side=TOP)**

**option = OptionMenu(self, self.option, "ID", "Brand", "Kategori", "Nama", "Lokasi", "Stok").pack(side=TOP)**

**Label(self, text=' ').pack(side=TOP)**

**Label(self, text=' ').pack(side=TOP)**

**button1=Button(self, text="CONFIRM", font=("Century Gothic",15),background="gray16",foreground="White", command=self.show).pack(side='top')**

**def back(self):**

**self.kotak1.delete('1.0',END)**

**self.option.set('')**

**self.controller.show\_frame('MainFrame')**

**def show(self):**

**opsi=self.option.get()**

**file = open("stock\_log.txt", "r+")**

**self.stok = []**

**line = file.readlines()**

**key = val = ""**

**temp = {}**

**for i in range(len(line)):**

**if i % 2 == 0:**

**key = line[i].rstrip()**

**else:**

**val = line[i].rstrip()**

**temp[key] = val**

**if (i + 1) % 12 == 0:**

**self.stok.append(temp)**

**temp = {}**

**file.close()**

**if opsi == 'ID':**

**self.kotak1.delete('1.0', END)**

**self.sorted\_ID()**

**elif opsi == 'Brand':**

**self.kotak1.delete('1.0', END)**

**self.sorted\_BRAND()**

**elif opsi == 'Kategori':**

**self.kotak1.delete('1.0', END)**

**self.sorted\_Kategori()**

**elif opsi == 'Nama':**

**self.kotak1.delete('1.0', END)**

**self.sorted\_Nama()**

**elif opsi == 'Lokasi':**

**self.kotak1.delete('1.0', END)**

**self.sorted\_Lokasi()**

**elif opsi == 'Stok':**

**self.kotak1.delete('1.0', END)**

**self.sorted\_Stok()**

**else:**

**tm.showerror("ERROR","Please choose one")**

**def sorted\_ID(self):**

**self.stok.sort(key=lambda x: (x["ID"]))**

**self.kotak1.insert(END, '|')**

**for key, val in self.stok[0].items():**

**self.kotak1.insert(END,'{:^25}'.format(key))**

**self.kotak1.insert(END, '|')**

**self.kotak1.insert(END,"\n")**

**self.kotak1.insert(END, "\_" \* 157, "\n")**

**self.kotak1.insert(END, "\n")**

**for i in range(len(self.stok)):**

**self.kotak1.insert(END, '|')**

**for key, val in self.stok[i].items():**

**self.kotak1.insert(END,'{:^25}'.format(str(val)))**

**self.kotak1.insert(END, '|')**

**self.kotak1.insert(END,"\n")**

**def sorted\_BRAND(self):**

**self.stok.sort(key=lambda x: (x["BRAND"]))**

**self.kotak1.insert(END, '|')**

**for key, val in self.stok[0].items():**

**self.kotak1.insert(END, '{:^25}'.format(key))**

**self.kotak1.insert(END, '|')**

**self.kotak1.insert(END, "\n")**

**self.kotak1.insert(END,"\_"\*157,"\n")**

**self.kotak1.insert(END, "\n")**

**for i in range(len(self.stok)):**

**self.kotak1.insert(END, '|')**

**for key, val in self.stok[i].items():**

**self.kotak1.insert(END, '{:^25}'.format(str(val)))**

**self.kotak1.insert(END, '|')**

**self.kotak1.insert(END, "\n")**

**def sorted\_Kategori(self):**

**self.stok.sort(key=lambda x: (x["Kategori"]))**

**self.kotak1.insert(END, '|')**

**for key, val in self.stok[0].items():**

**self.kotak1.insert(END, '{:^25}'.format(key))**

**self.kotak1.insert(END, '|')**

**self.kotak1.insert(END, "\n")**

**self.kotak1.insert(END, "\_" \* 157, "\n")**

**self.kotak1.insert(END, "\n")**

**for i in range(len(self.stok)):**

**self.kotak1.insert(END, '|')**

**for key, val in self.stok[i].items():**

**self.kotak1.insert(END, '{:^25}'.format(str(val)))**

**self.kotak1.insert(END, '|')**

**self.kotak1.insert(END, "\n")**

**def sorted\_Nama(self):**

**self.stok.sort(key=lambda x: (x["Nama"]))**

**self.kotak1.insert(END, '|')**

**for key, val in self.stok[0].items():**

**self.kotak1.insert(END, '{:^25}'.format(key))**

**self.kotak1.insert(END, '|')**

**self.kotak1.insert(END, "\n")**

**self.kotak1.insert(END, "\_" \* 157, "\n")**

**self.kotak1.insert(END, "\n")**

**for i in range(len(self.stok)):**

**self.kotak1.insert(END, '|')**

**for key, val in self.stok[i].items():**

**self.kotak1.insert(END, '{:^25}'.format(str(val)))**

**self.kotak1.insert(END, '|')**

**self.kotak1.insert(END, "\n")**

**def sorted\_Lokasi(self):**

**self.stok.sort(key=lambda x: (x["Lokasi"]))**

**self.kotak1.insert(END, '|')**

**for key, val in self.stok[0].items():**

**self.kotak1.insert(END, '{:^25}'.format(key))**

**self.kotak1.insert(END, '|')**

**self.kotak1.insert(END, "\n")**

**self.kotak1.insert(END, "\_" \* 157, "\n")**

**self.kotak1.insert(END, "\n")**

**for i in range(len(self.stok)):**

**self.kotak1.insert(END, '|')**

**for key, val in self.stok[i].items():**

**self.kotak1.insert(END, '{:^25}'.format(str(val)))**

**self.kotak1.insert(END, '|')**

**self.kotak1.insert(END, "\n")**

**def sorted\_Stok(self):**

**for i in range(len(self.stok)):**

**self.stok[i]["Stok"] = int(self.stok[i]["Stok"])**

**self.stok.sort(key=lambda x: (x["Stok"]))**

**self.kotak1.insert(END, '|')**

**for key, val in self.stok[0].items():**

**self.kotak1.insert(END, '{:^25}'.format(key))**

**self.kotak1.insert(END, '|')**

**self.kotak1.insert(END, "\n")**

**self.kotak1.insert(END, "\_" \* 157, "\n")**

**self.kotak1.insert(END, "\n")**

**for i in range(len(self.stok)):**

**self.kotak1.insert(END, '|')**

**for key, val in self.stok[i].items():**

**self.kotak1.insert(END, '{:^25}'.format(str(val)))**

**self.kotak1.insert(END, '|')**

**self.kotak1.insert(END, "\n")**

**class InputItem(Frame):**

**def \_\_init\_\_(self, parent, controller):**

**Frame.\_\_init\_\_(self, parent)**

**self.controller = controller**

**self.kotak1 = Text(master=self)**

**self.kotak1.pack(fill=X, ipady=75, ipadx=75, side=RIGHT)**

**#self.myvar1 = StringVar()**

**Button(self, text="Back", width=20, font=("Century Gothic",12),background="gray12",foreground="White",**

**command=self.back).pack(**

**side=BOTTOM)**

**self.initUI()**

**def initUI(self):**

**frame1 = Frame(self)**

**frame1.pack(fill=X)**

**self.myvar1=StringVar()**

**Label(frame1, text='INPUT ITEM', font=("Code-Bold", 30), justify="left",foreground="gray16").pack()**

**Label(frame1, text=' ', font=("Century Gothic", 15), justify="left").pack()**

**Label(frame1, text='ID :', font=("Century Gothic", 15), justify="left").pack(side=LEFT, padx=12, pady=7)**

**self.entry1=Entry(frame1, width=20, font=("Century Gothic", 15), textvariable=self.myvar1)**

**self.entry1.pack(fill=X, padx=7,ipadx=30, expand=True)**

**frame2 = Frame(self)**

**frame2.pack(fill=X)**

**self.myvar2 = StringVar()**

**Label(frame2, text='Brand :', font=("Century Gothic", 15), justify="left").pack(side=LEFT, padx=12, pady=7)**

**self.entry2=Entry(frame2, width=20, font=("Century Gothic", 15), textvariable=self.myvar2)**

**self.entry2.pack(fill=X, padx=7,ipadx=30, expand=True)**

**frame3 = Frame(self)**

**frame3.pack(fill=X)**

**self.myvar3 = StringVar()**

**Label(frame3, text='Kategori :', font=("Century Gothic", 15), justify="left").pack(side=LEFT, padx=12, pady=7)**

**self.entry3=Entry(frame3, width=20, font=("Century Gothic", 15), textvariable=self.myvar3)**

**self.entry3.pack(fill=X, padx=7,ipadx=30, expand=True)**

**frame4 = Frame(self)**

**frame4.pack(fill=X)**

**self.myvar4 = StringVar()**

**Label(frame4, text='Nama :', font=("Century Gothic", 15), justify="left").pack(side=LEFT, padx=12, pady=7)**

**self.entry4=Entry(frame4, width=20, font=("Century Gothic", 15), textvariable=self.myvar4)**

**self.entry4.pack(fill=X, padx=7,ipadx=30, expand=True)**

**frame5 = Frame(self)**

**frame5.pack(fill=X)**

**self.myvar5 = StringVar()**

**Label(frame5, text='Lokasi :', font=("Century Gothic", 15), justify="left").pack(side=LEFT, padx=12, pady=7)**

**self.entry5=Entry(frame5, width=20, font=("Century Gothic", 15), textvariable=self.myvar5)**

**self.entry5.pack(fill=X, padx=7,ipadx=30, expand=True)**

**frame6 = Frame(self)**

**frame6.pack(fill=X)**

**self.myvar6 = StringVar()**

**Label(frame6, text='Stok :', font=("Century Gothic", 15), justify="left").pack(side=LEFT, padx=12, pady=7)**

**self.entry6=Entry(frame6, width=20, font=("Century Gothic", 15), textvariable=self.myvar6)**

**self.entry6.pack(fill=X, padx=7,ipadx=30, expand=True)**

**Label(self, text=" ", font=('Code-Light', 1)).pack(side=BOTTOM)**

**Button(self, text="PROCESS", width=20, font=("Century Gothic",12),background="gray16",foreground="White",**

**command=self.search12).pack(**

**side=BOTTOM)**

**def back(self):**

**self.kotak1.delete('1.0',END)**

**self.entry1.delete(0, END)**

**self.entry2.delete(0, END)**

**self.entry3.delete(0, END)**

**self.entry4.delete(0, END)**

**self.entry5.delete(0, END)**

**self.entry6.delete(0, END)**

**self.controller.show\_frame('MainFrame')**

**def search12(self):**

**self.test()**

**self.entry1.delete(0, END)**

**self.entry2.delete(0, END)**

**self.entry3.delete(0, END)**

**self.entry4.delete(0, END)**

**self.entry5.delete(0, END)**

**self.entry6.delete(0, END)**

**def test(self):**

**self.ID = self.myvar1.get()**

**self.BRAND = self.myvar2.get()**

**self.Kategori = self.myvar3.get()**

**self.Nama = self.myvar4.get()**

**self.Lokasi = self.myvar5.get()**

**try:**

**self.Stok = int(self.myvar6.get())**

**self.Stok = self.myvar6.get()**

**self.inputBarang()**

**except ValueError:**

**tm.showerror('ERROR','Please Input correct Number')**

**def inputBarang(self):**

**self.kotak1.delete('1.0',END)**

**ID=self.ID.upper()**

**Lokasi=self.Lokasi.upper()**

**BRAND=self.BRAND.upper()**

**Kategori=self.Kategori.upper()**

**Nama=self.Nama.title()**

**Stok=self.Stok**

**self.newdata = {}**

**file = open("stock\_log.txt", "r+")**

**self.stok = []**

**line = file.readlines()**

**key = val = ""**

**temp = {}**

**for i in range(len(line)):**

**if i % 2 == 0:**

**key = line[i].rstrip()**

**else:**

**val = line[i].rstrip()**

**temp[key] = val**

**if (i + 1) % 12 == 0:**

**self.stok.append(temp)**

**temp = {}**

**file.close()**

**ditemukan = False**

**for j in range(len(self.stok)):**

**if self.stok[j]["ID"] == ID:**

**ditemukan = True**

**if ditemukan == True:**

**tm.showerror('ERROR', 'ID already exist!')**

**if ditemukan==False:**

**self.newdata["ID"] = ID**

**self.newdata["BRAND"] = BRAND**

**self.newdata["Kategori"] = Kategori**

**self.newdata["Nama"] = Nama**

**self.newdata["Lokasi"] = Lokasi**

**self.newdata["Stok"] = Stok**

**self.stok.append(self.newdata)**

**file = open("stock\_log.txt", "w")**

**for i in range(len(self.stok)):**

**for j in self.stok[i]:**

**file.write(j)**

**file.write('\n')**

**file.write(self.stok[i][j])**

**file.write('\n')**

**file.close()**

**self.kotak1.insert(END,"Barang dengan\n")**

**self.kotak1.insert(END, "ID : "+ID+"\n")**

**self.kotak1.insert(END, "Brand : " + BRAND + "\n")**

**self.kotak1.insert(END, "Kategori : " + Kategori + "\n")**

**self.kotak1.insert(END, "Nama : " + Nama + "\n")**

**self.kotak1.insert(END, "Lokasi : " + Lokasi + "\n")**

**self.kotak1.insert(END, "Stok : " + Stok + "\n")**

**self.kotak1.insert(END,"Data inputted successfully\n")**

**class BarangKeluar(Frame):**

**def \_\_init\_\_(self, parent, controller):**

**Frame.\_\_init\_\_(self, parent)**

**self.controller = controller**

**self.kotak1 = Text(master=self)**

**self.kotak1.pack(fill=X, ipady=75, ipadx=75, side=RIGHT)**

**Button(self, text="Back", width=20, font=("Century Gothic",12),background="gray12",foreground="White",**

**command=self.back).pack(**

**side=BOTTOM)**

**self.initUI()**

**def initUI(self):**

**frame1 = Frame(self)**

**frame1.pack(fill=X)**

**self.myvar1 = StringVar()**

**Label(frame1, text='item (out)', font=("Code-Bold", 30), justify="left",foreground="gray16").pack()**

**Label(frame1, text=' ', font=("Century Gothic", 15), justify="left").pack()**

**Label(frame1, text='ID :', font=("Century Gothic", 15), justify="left").pack(side=LEFT, padx=12, pady=7)**

**self.entry1 = Entry(frame1, width=20, font=("Century Gothic", 15), textvariable=self.myvar1)**

**self.entry1.pack(fill=X, padx=7, ipadx=30, expand=True)**

**frame2 = Frame(self)**

**frame2.pack(fill=X)**

**self.myvar2 = StringVar()**

**Label(frame2, text='Item [OUT] :', font=("Century Gothic", 15), justify="left").pack(side=LEFT, padx=12, pady=7)**

**self.entry2 = Entry(frame2, width=20, font=("Century Gothic", 15), textvariable=self.myvar2)**

**self.entry2.pack(fill=X, padx=7, ipadx=30, expand=True)**

**Label(self, text=" ", font=('Code-Light', 1)).pack(side=BOTTOM)**

**Button(self, text="PROCESS", width=20, font=("Century Gothic",12),background="gray16",foreground="White",**

**command=self.search12).pack(**

**side=BOTTOM)**

**def search12(self):**

**self.test()**

**self.entry1.delete(0, END)**

**self.entry2.delete(0, END)**

**def back(self):**

**self.kotak1.delete('1.0',END)**

**self.entry1.delete(0, END)**

**self.entry2.delete(0, END)**

**self.controller.show\_frame('MainFrame')**

**def test(self):**

**ID = self.myvar1.get()**

**try:**

**keluar = int(self.myvar2.get())**

**self.barangKeluar(ID, keluar)**

**except ValueError:**

**tm.showerror('ERROR','Please Input correct Number')**

**def barangKeluar(self,ID, keluar):**

**file = open("stock\_log.txt", "r+")**

**stok = []**

**line = file.readlines()**

**key = val = ""**

**temp = {}**

**for i in range(len(line)):**

**if i % 2 == 0:**

**key = line[i].rstrip()**

**else:**

**val = line[i].rstrip()**

**temp[key] = val**

**if (i + 1) % 12 == 0:**

**stok.append(temp)**

**temp = {}**

**file.close()**

**exist = False**

**for i in range(len(stok)):**

**if stok[i]["ID"] == ID:**

**exist = True**

**if exist == False:**

**self.kotak1.insert(END,'Item with ID '+ID+'\n' )**

**self.kotak1.insert(END,"Not Available")**

**tm.showinfo('OOPS', "Not Found")**

**else:**

**for i in range(len(stok)):**

**if stok[i]['ID'] == ID:**

**index = i**

**stoklama = stok[index]["Stok"]**

**stokbaru = int(stoklama) - keluar**

**if stokbaru < 0:**

**tm.showerror('ERROR',"Stok kurang")**

**else:**

**if stokbaru==0:**

**tm.showinfo("INFO","Barang telah habis")**

**stok[index]["Stok"] = str(stokbaru)**

**file = open("stock\_log.txt", "w")**

**for i in range(len(stok)):**

**for j in stok[i]:**

**file.write(j)**

**file.write('\n')**

**file.write(stok[i][j])**

**file.write('\n')**

**file.close()**

**self.kotak1.insert(END,"Stock item with ID "+ID+"\n")**

**self.kotak1.insert(END,"Now "+str(stokbaru)+"\n")**

**class BarangMasuk(Frame):**

**def \_\_init\_\_(self, parent, controller):**

**Frame.\_\_init\_\_(self, parent)**

**self.controller = controller**

**self.kotak1 = Text(master=self)**

**self.kotak1.pack(fill=X, ipady=75, ipadx=75, side=RIGHT)**

**Button(self, text="Back", width=20, font=("Century Gothic",12),background="gray12",foreground="White",**

**command=self.back).pack(**

**side=BOTTOM)**

**self.initUI()**

**def initUI(self):**

**frame1 = Frame(self)**

**frame1.pack(fill=X)**

**self.myvar1 = StringVar()**

**Label(frame1, text='item in', font=("Code-Bold", 30), justify="left",foreground="gray16").pack()**

**Label(frame1, text=' ', font=("Century Gothic", 15), justify="left").pack()**

**Label(frame1, text='ID :', font=("Century Gothic", 15), justify="left").pack(side=LEFT, padx=12, pady=7)**

**self.entry1 = Entry(frame1, width=20, font=("Century Gothic", 15), textvariable=self.myvar1)**

**self.entry1.pack(fill=X, padx=7, ipadx=30, expand=True)**

**frame2 = Frame(self)**

**frame2.pack(fill=X)**

**self.myvar2 = StringVar()**

**Label(frame2, text='Item [IN] :', font=("Century Gothic", 15), justify="left").pack(side=LEFT, padx=12, pady=7)**

**self.entry2 = Entry(frame2, width=20, font=("Century Gothic", 15), textvariable=self.myvar2)**

**self.entry2.pack(fill=X, padx=7, ipadx=30,expand=True)**

**Label(self, text=" ", font=('Code-Light', 1)).pack(side=BOTTOM)**

**Button(self, text="PROCESS", width=20, font=("Century Gothic",12),background="gray16",foreground="White",**

**command=self.search12).pack(**

**side=BOTTOM)**

**def search12(self):**

**self.test()**

**self.entry1.delete(0, END)**

**self.entry2.delete(0, END)**

**def back(self):**

**self.kotak1.delete('1.0',END)**

**self.entry1.delete(0, END)**

**self.entry2.delete(0, END)**

**self.controller.show\_frame('MainFrame')**

**def test(self):**

**ID = self.myvar1.get()**

**try:**

**Masuk = int(self.myvar2.get())**

**self.barangMasuk(ID, Masuk)**

**except ValueError:**

**tm.showerror('ERROR', 'Please Input correct Number')**

**def barangMasuk(self,ID, masuk):**

**file = open("stock\_log.txt", "r+")**

**stok = []**

**line = file.readlines()**

**key = val = ""**

**temp = {}**

**for i in range(len(line)):**

**if i % 2 == 0:**

**key = line[i].rstrip()**

**else:**

**val = line[i].rstrip()**

**temp[key] = val**

**if (i + 1) % 12 == 0:**

**stok.append(temp)**

**temp = {}**

**file.close()**

**exist = False**

**for i in range(len(stok)):**

**if stok[i]["ID"] == ID:**

**exist = True**

**if exist == False:**

**self.kotak1.insert(END, 'Item with ID ' + ID + '\n')**

**self.kotak1.insert(END, "Not Available")**

**tm.showinfo('OOPS', "Not Found")**

**else:**

**for i in range(len(stok)):**

**if stok[i]['ID'] == ID:**

**index = i**

**stoklama = stok[index]["Stok"]**

**stokbaru = int(stoklama) + masuk**

**if stokbaru < 0:**

**tm.showerror('ERROR',"Input invalid")**

**else:**

**stok[index]["Stok"] = str(stokbaru)**

**file = open("stock\_log.txt", "w")**

**for i in range(len(stok)):**

**for j in stok[i]:**

**file.write(j)**

**file.write('\n')**

**file.write(stok[i][j])**

**file.write('\n')**

**file.close()**

**self.kotak1.insert(END, "Stock item with ID " + ID + "\n")**

**self.kotak1.insert(END, "Now " + str(stokbaru)+"\n")**

**class DeleteItem(Frame):**

**def \_\_init\_\_(self, parent, controller):**

**Frame.\_\_init\_\_(self, parent)**

**self.controller = controller**

**self.kotak1 = Text(master=self)**

**self.kotak1.pack(fill=X, ipady=75, ipadx=75, side=RIGHT)**

**Button(self, text="Back", width=20, font=("Century Gothic",12),background="gray12",foreground="White",**

**command=self.back).pack(**

**side=BOTTOM)**

**self.initUI()**

**def initUI(self):**

**frame1 = Frame(self)**

**frame1.pack(fill=X)**

**self.myvar1 = StringVar()**

**Label(frame1, text='delete ITEM', font=("Code-Bold", 30),foreground="gray16", justify="left").pack()**

**Label(frame1, text=' ', font=("Century Gothic",15), justify="left").pack()**

**Label(frame1, text='ID :', font=("Century Gothic", 15), justify="left").pack(side=LEFT, padx=12, pady=7)**

**self.entry1 = Entry(frame1, width=20, font=("Century Gothic", 15), textvariable=self.myvar1)**

**self.entry1.pack(fill=X, padx=7, ipadx=30, expand=True)**

**Label(self, text=" ", font=('Code-Light', 1)).pack(side=BOTTOM)**

**Button(self, text="PROCESS", width=20, font=("Century Gothic",12),background="gray16",foreground="White",**

**command=self.search12).pack(**

**side=BOTTOM)**

**def search12(self):**

**self.test()**

**self.entry1.delete(0, END)**

**def back(self):**

**self.kotak1.delete('1.0',END)**

**self.entry1.delete(0, END)**

**self.controller.show\_frame('MainFrame')**

**def test(self):**

**self.kotak1.delete('1.0', END)**

**self.deleteBarang()**

**def deleteBarang(self):**

**ID = self.myvar1.get()**

**file = open("stock\_log.txt", "r+")**

**stok = []**

**line = file.readlines()**

**key = val = ""**

**temp = {}**

**for i in range(len(line)):**

**if i % 2 == 0:**

**key = line[i].rstrip()**

**else:**

**val = line[i].rstrip()**

**temp[key] = val**

**if (i + 1) % 12 == 0:**

**stok.append(temp)**

**temp = {}**

**file.close()**

**exist = False**

**for i in range(len(stok)):**

**if stok[i]['ID'] == ID:**

**exist = True**

**for key, val in stok[i].items():**

**self.kotak1.insert(END, (key, ":", val))**

**self.kotak1.insert(END, '\n')**

**stok.pop(i)**

**self.kotak1.insert(END, 'Has Been Deleted')**

**break**

**if exist == False:**

**tm.showerror("ERROR!", "ID not found!")**

**file = open("stock\_log.txt", "w")**

**for i in range(len(stok)):**

**for j in stok[i]:**

**file.write(j)**

**file.write('\n')**

**file.write(stok[i][j])**

**file.write('\n')**

**file.close()**

**if \_\_name\_\_ == "\_\_main\_\_":**

**app = SampleApp()**

**app.title("Stock - King")**

**app.geometry("600x450+900+300")**

**full = FullScreenApp(app)**

**app.mainloop()**