

**PROPOSAL TUGAS BESAR ALGORITMA DAN PEMROGRAMAN
APLIKASI GENERATOR *INVOICE* PENJUALAN**



SI-46-08
Disusun oleh:

Sheila Deninta Maharani
1202223324

**PROGRAM STUDI S1 SISTEM INFORMASI
FAKULTAS REKAYASA INDUSTRI
UNIVERSITAS TELKOM
2023**

DAFTAR PUSTAKA

BAB I PENDAHULUAN.....	2
1.1 Latar Belakang	2
1.2 Rumusan Masalah	2
1.3 Tujuan.....	3
1.4 Batasan Masalah.....	3
BAB II PEMBAHASAN.....	3
2.1 Deskripsi Program.....	3
2.2 Design Program.....	3
2.3 Flowchart Program.....	16
DAFTAR PUSTAKA.....	18

BAB I PENDAHULUAN

1.1 Latar Belakang

Dalam dunia bisnis, pencatatan dan pengelolaan transaksi penjualan adalah aspek penting yang memengaruhi kelangsungan dan kesuksesan sebuah perusahaan. Pencatatan transaksi yang efisien dan akurat memungkinkan perusahaan untuk memantau aliran pendapatan, mengelola stok, melakukan analisis keuangan, dan menyediakan layanan pelanggan yang baik. Namun, masih banyak bisnis, terutama yang berukuran kecil atau menengah, yang menghadapi tantangan dalam hal pencatatan transaksi penjualan.

Metode manual dalam pencatatan transaksi penjualan, seperti menggunakan buku catatan atau lembar Excel, sering kali memakan waktu, rawan kesalahan, dan sulit dilacak. Kesalahan manusia dalam menginput data atau menghitung total transaksi dapat mengakibatkan ketidakakuratan dalam laporan keuangan, kesalahan pengiriman barang, atau kesalahan faktur. Selain itu, dengan adanya pertumbuhan bisnis dan volume transaksi yang semakin besar, metode manual menjadi tidak efisien dan tidak praktis.

Untuk mengatasi tantangan ini, banyak bisnis beralih ke penggunaan aplikasi penjualan otomatis. Aplikasi ini menyediakan solusi yang lebih terstruktur, efisien, dan akurat dalam pencatatan dan pengelolaan transaksi penjualan. Namun, masih ada bisnis yang belum mengadopsi aplikasi tersebut karena keterbatasan pengetahuan teknologi, biaya implementasi, atau kurangnya sumber daya IT.

Dalam konteks ini, pengembangan program aplikasi GUI (Graphical User Interface) Python untuk *generate invoice* menjadi sangat relevan dan penting. Python adalah bahasa pemrograman yang bersifat modular dan dapat diperluas melalui impor modul ekstensi, dengan distribusi Python menyertakan beragam ekstensi standar untuk berbagai operasi seperti manipulasi string, ekspresi reguler mirip Perl, pembuat Graphical User Interface (GUI), utilitas web, layanan sistem operasi, alat pemecahan masalah, dan sebagainya (Sanner, M. F., 1999). Dengan menggunakan Python, pengembang dapat menciptakan antarmuka pengguna yang intuitif dan mudah digunakan dalam bentuk GUI.

Aplikasi ini akan memberikan manfaat signifikan bagi bisnis dalam hal efisiensi, produktivitas, dan penghematan waktu. Dengan menggunakan aplikasi generator faktur penjualan otomatis, bisnis dapat menghilangkan proses manual yang memakan waktu dan rawan kesalahan. Pengguna dapat dengan mudah mengisi informasi transaksi, termasuk data pelanggan, produk atau layanan yang dibeli, harga, dan jumlah, melalui antarmuka yang disediakan. Program akan secara otomatis menghasilkan *invoice* dengan detail transaksi dan informasi pelanggan yang relevan.

Selain itu, aplikasi ini juga memungkinkan bisnis untuk mengelola data pelanggan dengan lebih baik. Informasi pelanggan seperti nama, alamat, nomor telepon, dan informasi kontak lainnya dapat disimpan dan diakses dengan mudah. Ini memungkinkan pengguna untuk menghasilkan faktur dengan cepat dan akurat dengan memanfaatkan data pelanggan yang tersedia.

Secara keseluruhan, pengembangan program aplikasi GUI (Graphical User Interface) Python untuk *generate invoice* merupakan langkah penting dalam mengotomatisasi proses pencatatan transaksi penjualan. Dengan menggunakan aplikasi ini, bisnis akan mendapatkan keuntungan dalam hal efisiensi, akurasi, dan penghematan waktu, yang pada gilirannya akan meningkatkan produktivitas, kepuasan pelanggan, dan keberlanjutan bisnis secara keseluruhan.

1.2 Rumusan Masalah

Dalam pencatatan transaksi penjualan secara manual, terdapat beberapa masalah, antara lain:

1. Memakan waktu yang lama dan membutuhkan tenaga yang besar
2. Berpotensi terjadinya kesalahan pada pencatatan transaksi
3. Kesulitan dalam membuat laporan keuangan yang akurat dan cepat

1.3 Tujuan

Tujuan dari pembuatan aplikasi generator fraktur penjualan otomatis adalah untuk membantu mempermudah dan mempercepat pencatatan transaksi penjualan serta meningkatkan akurasi laporan keuangan.

1.4 Batasan Masalah

Pada pembuatan aplikasi generator fraktur penjualan otomatis ini, terdapat beberapa batasan masalah, antara lain:

1. Aplikasi hanya dapat digunakan untuk bisnis skala kecil dan menengah.
2. Aplikasi hanya dapat menghasilkan fraktur penjualan dalam format docx.
3. Aplikasi tidak dapat melakukan integrasi dengan sistem informasi bisnis yang sudah ada.

BAB II PEMBAHASAN

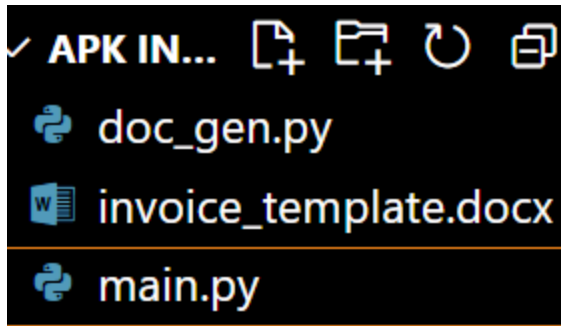
2.1 Deskripsi Program

Program aplikasi generator invoice penjualan adalah sebuah program Python yang menggunakan *Graphical User Interface* (GUI) yang memungkinkan pengguna untuk menginput data yang terkait dengan pencetakan *invoice* penjualan. Pengguna dapat mengisi nomor *invoice*, nama pelanggan, produk yang dibeli beserta harga dan jumlahnya, serta tanggal pemesanan dan jatuh tempo pembayaran barang. Harga-harga yang diinputkan akan secara otomatis dijumlahkan dan ditampilkan di GUI. Setelah semua data terisi, pengguna dapat menekan tombol "Generate Invoice" untuk mencetak invoice dengan menggunakan data yang telah diinputkan dalam bentuk file dokumen Word. Template dari invoice dapat disesuaikan sesuai dengan kebutuhan bisnis pengguna. Pada program ini juga terdapat fungsi yang dapat menghitung pajak dari pembelian barang pelanggan secara otomatis dan langsung terisi di dalam dokumen *invoice*-nya.

Apabila terjadi kesalahan dalam penginputan produk, pengguna dapat menggunakan fitur "hapus produk" untuk menghapus produk yang salah. Selain itu, jika pengguna ingin membatalkan proses pencetakan invoice atau ingin membuat invoice baru, pengguna dapat menekan tombol "New Invoice" yang akan menghapus semua data yang telah dimasukkan dalam aplikasi. Selain fitur "hapus item", program ini juga menerapkan fitur tombol "Enter" keyboard sebagai opsi kedua dari tombol "Add item" yang berfungsi untuk menambahkan item ke tampilan aplikasi GUI.

Dengan menggunakan program ini, pengguna dapat dengan mudah dan cepat mencetak invoice penjualan dengan menggunakan data yang relevan. Hal ini akan membantu bisnis dalam menghemat waktu dan tenaga yang sebelumnya dibutuhkan untuk mencatat transaksi penjualan secara manual. Dengan adanya fitur GUI, pengguna juga dapat dengan mudah berinteraksi dengan program dan mengelola data yang dimasukkan. Program ini memberikan solusi yang efisien dan efektif dalam mempermudah proses pencatatan dan pencetakan invoice penjualan.

2.2 Design Program



Program ini terdiri dari tiga file yang saling berhubungan. Berikut adalah deskripsi dari masing-masing file:

1. File **main.py**

File ini berfungsi sebagai file utama program yang mengatur logika antarmuka pengguna. File ini menggunakan modul “tkinter” untuk membuat antarmuka pengguna grafis. Selain itu, file ini mengimpor modul-modul lain yang diperlukan seperti `ttk`, `docxtpl`, `tkcalendar`, dan `datetime`.

Untuk modul “tk” atau Themed Tkinter, modul ini merupakan bagian dari library `tkinter` yang menyediakan antarmuka pengguna yang lebih kaya secara visual. Modul ini mengimplementasikan tema yang dapat digunakan untuk memodifikasi tampilan elemen-elemen antarmuka pengguna seperti tombol, label, entri, dan lainnya. Sedangkan, modul “`docxtpl`” adalah pustaka Python yang digunakan untuk menggabungkan data dengan template dokumen Microsoft Word (.docx) untuk membuat dokumen yang dihasilkan secara dinamis. Modul ini memungkinkan untuk mengisi variabel-variabel yang ditentukan dalam template dokumen dengan data yang diberikan, seperti teks, tabel, gambar, dan lainnya. Hal ini dapat menghasilkan dokumen yang dipersonalisasi berdasarkan template yang telah ditentukan sebelumnya. Selain kedua modul tersebut, terdapat 2 modul yang berkesinambungan yaitu “`tkcalendar`” dan “`datetime`”. Modul “`tkcalendar`” adalah pustaka Python yang menyediakan widget kalender yang dapat digunakan dalam aplikasi berbasis Tkinter. Modul ini memungkinkan Anda untuk menampilkan dan memilih tanggal dengan mudah menggunakan antarmuka kalender. Widget “`DateEntry`” yang disediakan oleh `tkcalendar` memungkinkan pengguna untuk memasukkan tanggal dengan cara yang intuitif dan memastikan format tanggal yang benar. Sedangkan, modul “`datetime`” adalah bagian dari library standar Python yang menyediakan fungsi dan kelas untuk memanipulasi dan bekerja dengan objek tanggal dan waktu. Modul ini memungkinkan Anda untuk mengambil informasi tentang tanggal dan waktu saat ini, melakukan operasi matematika pada tanggal dan waktu, memformat tanggal dan waktu menjadi string, dan banyak lagi. `datetime` sangat berguna dalam pemrosesan tanggal dan waktu, termasuk penghitungan selisih waktu, manipulasi tanggal, pengaturan zona waktu, dan sebagainya.

Modul – modul di atas digunakan untuk menjalankan beberapa syntax dan fungsi terkait fitur – fitur di program ini. Pada program ini, pengguna dapat memasukkan informasi seperti nama, nomor telepon, tanggal order, tanggal jatuh tempo, nomor invoice, serta menambahkan, menghapus, dan melihat daftar item invoice. Program ini juga menghitung subtotal, pajak penjualan, dan total secara otomatis. Selain itu, program juga memberikan fitur untuk menghasilkan invoice dalam format dokumen Word.

Untuk menjalankan semua fitur tersebut diperlu fungsi – fungsi seperti :

1. Fungsi `clear_item()` yang digunakan untuk menghapus dan mengatur ulang nilai-nilai pada elemen-elemen entri dan spinbox pada antarmuka pengguna.
2. Variabel `invoice_list` digunakan untuk menyimpan daftar item invoice yang ditambahkan oleh pengguna.
3. Fungsi `add_item()` yang digunakan untuk menambahkan item ke dalam daftar penjualan. Data kuantitas, deskripsi barang, dan harga diambil dari elemen-elemen entri dan spinbox pada antarmuka pengguna. Jika ada data yang kosong, maka akan muncul pesan peringatan. Setelah itu, item penjualan ditambahkan ke dalam treeview dan daftar penjualan. Jumlah subtotal juga diupdate.
4. Fungsi `new_invoice()` yang digunakan untuk mengatur ulang semua elemen antarmuka pengguna dan membersihkan daftar penjualan.
5. Fungsi `generate_invoice()` yang digunakan untuk menghasilkan invoice penjualan. Data seperti nama, nomor telepon, tanggal order, tanggal jatuh tempo, nomor invoice, daftar item penjualan, subtotal, pajak penjualan, dan total diambil dari elemen-elemen entri, spinbox, dan daftar penjualan pada antarmuka pengguna. Invoice tersebut disimpan sebagai file dengan nama yang unik menggunakan `DocxTemplate` dan `save()`. Setelah itu, pesan konfirmasi ditampilkan dan fungsi `new_invoice()` dipanggil untuk memulai invoice baru.
6. Fungsi `delete_item()` yang digunakan untuk menghapus item dari daftar penjualan. Item yang dipilih dihapus dari daftar penjualan dan treeview, dan subtotal diperbarui.
7. Fungsi `on_enter(event)` yang digunakan untuk menambahkan item ke daftar penjualan ketika tombol Enter ditekan.

Selain itu, berbagai elemen antarmuka pengguna seperti label, entri, spinbox, tombol, dan treeview dibuat dan ditempatkan sesuai dengan tata letak yang telah ditentukan di file ini. Termasuk juga pengaturan gaya dan tampilan antarmuka pengguna dilakukan dengan menggunakan `ttk.Style()`. Lalu pada akhirnya, `window.mainloop()` digunakan untuk menjalankan aplikasi dan mempertahankan antarmuka pengguna hingga ditutup. Berikut ini implementasi dalam bentuk *source codes*-nya :

```
import tkinter
from tkinter import *
from tkinter import ttk
from docxtpl import DocxTemplate
from tkcalendar import DateEntry
import datetime
from tkinter import messagebox

def clear_item():
    kuantitas_spinbox.delete(0, tkinter.END)
    kuantitas_spinbox.insert(0, "1")
    deskripsiBarang_entry.delete(0, tkinter.END)
    price_spinbox.delete(0, tkinter.END)
    price_spinbox.insert(0, "0.0")
```

```

invoice_list = []
def add_item():
    kuantitas = int(kuantitas_spinbox.get())
    deskripsiBarang = deskripsiBarang_entry.get()
    price = float(price_spinbox.get())
    if not kuantitas or not deskripsiBarang or not price:
        messagebox.showwarning("Peringatan", "Mohon isi terlebih dahulu datanya!")
    return
    line_total = kuantitas*price
    invoice_item = [kuantitas, deskripsiBarang, price, line_total]
    tree.insert('',0, values=invoice_item)
    clear_item()

    invoice_list.append(invoice_item)
    subtotal = sum([item[3] for item in invoice_list])
    subtotal_label.config(text=f'Total: {subtotal:.2f}')

def new_invoice():
    first_name_entry.delete(0, tkinter.END)
    last_name_entry.delete(0, tkinter.END)
    phone_entry.delete(0, tkinter.END)
    no_invoice_entry.delete(0, tkinter.END)
    tgl_order_entry.set_date(datetime.date.today())
    tgl_jatuhtempo_entry.delete(0, tkinter.END)
    clear_item()
    tree.delete(*tree.get_children())

    invoice_list.clear()
    subtotal_label.config(text='Total: 0.0')

def generate_invoice():
    if len(invoice_list) == 0:
        messagebox.showwarning("Peringatan", "Tolong isi data terlebih dahulu!")
    return
    doc = DocxTemplate("invoice_template.docx")
    name = first_name_entry.get()+last_name_entry.get()
    phone = phone_entry.get()
    tgl_order = tgl_order_entry.get()
    tgl_jatuhtempo = tgl_jatuhtempo_entry.get()
    no_invoice = no_invoice_entry.get()
    subtotal = sum(item[3] for item in invoice_list)
    salestax = 0.1
    total = subtotal*(1+salestax)

```

```

        doc.render({"name":name,
                    "phone":phone,
                    "tgl_order": tgl_order,
                    "tgl_jatuhtempo": tgl_jatuhtempo,
                    "no_invoice" : no_invoice,
                    "invoice_list": invoice_list,
                    "subtotal":subtotal,
                    "salestax":str(salestax*100)+"%",
                    "total":total})

        doc_name = "new_invoice" + name +
datetime.datetime.now().strftime("%Y-%m-%d-%H%M%S") + ".docx"
        doc.save(doc_name)

        messagebox.showinfo("Invoice Complete", "Invoice Complete")

        new_invoice()

```

```

def delete_item():
    selected_items = tree.selection()
    for selected_item in selected_items:
        item_values = tree.item(selected_item)['values']
        for i in invoice_list:
            if i[1] == item_values[1]:
                invoice_list.remove(i)
                break
        tree.delete(selected_item)
    subtotal = sum(item[3] for item in invoice_list)
    subtotal_label.config(text=f'Total: {subtotal:.2f}')

```

```

def on_enter(event):
    add_item()

```

```

window = tkinter.Tk()
window.title("Aplikasi Penjualan PT Sheila Jaya")

```

```

style = ttk.Style(window)
style.theme_use("clam")

```

```

frame = tkinter.Frame(window)
frame.pack(padx=20, pady=10)

```

```

subtotal_label = tkinter.Label(frame, text='Total: 0.0', font=("Arial",
12, "bold"))

```



```

subtotal_label.grid(row=9, column=2)

title_label = tkinter.Label(frame, text="Aplikasi Penjualan PT Sheila
Jaya", font=("Arial", 16, "bold"))
title_label.grid(row=0, column=0, columnspan=3, pady=10)

first_name_label = tkinter.Label(frame, text="First Name")
first_name_label.grid(row=1, column=0)
last_name_label = tkinter.Label(frame, text="Last Name")
last_name_label.grid(row=1, column=1)

first_name_entry = tkinter.Entry(frame)
last_name_entry = tkinter.Entry(frame)
first_name_entry.grid(row=2, column=0)
last_name_entry.grid(row=2, column=1)

phone_label = tkinter.Label(frame, text="Phone")
phone_label.grid(row=1, column=2)
phone_entry = tkinter.Entry(frame)
phone_entry.grid(row=2, column=2)

tgl_order_label = tkinter.Label(frame, text="Tanggal Order")
tgl_order_label.grid(row=3, column=0)
tgl_order_entry = DateEntry(frame, date_pattern='dd/mm/yyyy')
tgl_order_entry.grid(row=4, column=0)

no_invoice_label = tkinter.Label(frame, text="Nomor Invoice")
no_invoice_label.grid(row=3, column=1)
no_invoice_entry = tkinter.Entry(frame)
no_invoice_entry.grid(row=4, column=1)

tgl_jatuhtempo_label = tkinter.Label(frame, text="Tanggal Jatuh Tempo")
tgl_jatuhtempo_label.grid(row=3, column=2)
tgl_jatuhtempo_entry = DateEntry(frame, date_pattern='dd/mm/yyyy')
tgl_jatuhtempo_entry.grid(row=4, column=2)

kuantitas_label = tkinter.Label(frame, text="Kuantitas")
kuantitas_label.grid(row=5, column=0)
kuantitas_spinbox = tkinter.Spinbox(frame, from_=1, to=100)
kuantitas_spinbox.grid(row=6, column=0)

deskripsiBarang_label = tkinter.Label(frame, text="Deskripsi Barang")
deskripsiBarang_label.grid(row=5, column=1)
deskripsiBarang_entry = tkinter.Entry(frame)
deskripsiBarang_entry.grid(row=6, column=1)

```

```

price_label = tkinter.Label(frame, text="Harga Satuan")
price_label.grid(row=5, column=2)
price_spinbox = tkinter.Spinbox(frame, from_=0.0, to=1000000.0,
increment=500.0)
price_spinbox.grid(row=6, column=2)

add_item_button = tkinter.Button(frame, text="Add item",
command=add_item)
add_item_button.grid(row=7, column=1, pady=5)
add_item_button.bind('<Return>', on_enter)

delete_item_button = tkinter.Button(frame, text = "Delete item",
command = delete_item)
delete_item_button.grid(row=7, column=2, pady=5)

columns = ('kuantitas', 'deskripsiBarang', 'price', 'total')
tree = ttk.Treeview(frame, columns=columns, show="headings")
tree.heading('kuantitas', text='Kuantitas')
tree.heading('deskripsiBarang', text='Deskripsi Barang')
tree.heading('price', text='Harga Satuan')
tree.heading('total', text="Total")

tree.grid(row=8, column=0, columnspan=3, padx=20, pady=10)
style = ttk.Style()
style.configure("Treeview.Heading", background="#DEB887",
foreground="black")

save_invoice_button = tkinter.Button(frame, text="Generate Invoice",
command=generate_invoice)
save_invoice_button.grid(row=10, column=0, columnspan=3, sticky="news",
padx=20, pady=5)
new_invoice_button = tkinter.Button(frame, text="New Invoice",
command=new_invoice)
new_invoice_button.grid(row=11, column=0, columnspan=3, sticky="news",
padx=20, pady=5)

window.configure(bg='#DEB887')
frame.configure(bg='#FFE4C4')
title_label.configure(bg='#FFE4C4')
first_name_label.configure(bg='#FFE4C4')
last_name_label.configure(bg='#FFE4C4')
phone_label.configure(bg='#FFE4C4')
tgl_order_label.configure(bg='#FFE4C4')
no_invoice_label.configure(bg='#FFE4C4')

```

```

tgl_jatuhtempo_label.configure(bg='#FFE4C4')
kuantitas_label.configure(bg='#FFE4C4')
deskripsiBarang_label.configure(bg='#FFE4C4')
price_label.configure(bg='#FFE4C4')
add_item_button.configure(bg='#DEB887', foreground="black")
delete_item_button.configure(bg='#DEB887', foreground="black")
save_invoice_button.configure(bg='#DEB887', foreground="black")
new_invoice_button.configure(bg='#DEB887', foreground="black")
title_label.configure(bg='#FFE4C4')
subtotal_label.configure(borderwidth=2, relief="ridge")

kuantitas_spinbox.focus()
deskripsiBarang_entry.focus()
price_spinbox.focus()
window.bind('<Return>', on_enter)

image = tkinter.PhotoImage(file="D:/2. SHEILA KULIAH TELKOM/APK INVOICE
GENERATOR PENJUALAN/logo_sheila_jaya1.png")
image = image.subsample(4)
logo_label = tkinter.Label(frame, image=image)
logo_label.grid(row=0, column=0)
logo_label.configure(bg='#FFE4C4')

window.mainloop()

```

2. File **doc_gen.py**

File “doc_gen.py” bertanggung jawab untuk menghasilkan file *invoice* berdasarkan template yang telah ditentukan. File ini menggunakan modul “DocxTemplate” dari “docxtpl” untuk menghasilkan invoice penjualan. Objek doc dibuat dengan menggunakan template *invoice* yang disebut “invoice_template.docx”. Data penjualan berupa daftar item invoice disimpan dalam variabel `invoice_list`. Objek doc di-render dengan mengganti placeholder di dalam template dengan data yang diperoleh dari input-an *user*. Data yang diisi antara lain nama, nomor telepon, tanggal order, tanggal jatuh tempo, nomor invoice, daftar item invoice, subtotal, pajak penjualan, dan total. Hasil invoice yang telah diisi kemudian disimpan sebagai file “new_invoice.docx”. Berikut ini implementasi dari *source codes*-nya:

```

from docxtpl import DocxTemplate

doc = DocxTemplate("invoice_template.docx")

invoice_list = [[2, "pen", 0.5, 1],
                [1, "paper pack", 5, 5],
                [2, "notebook", 2, 4]]

doc.render({"name": "john",

```


```

"phone": "555-5555",
"tgl_order": "27/01/2023",
"tgl_jatuhtempo": "27/02/2023",
"no_invoice": "140",
"invoice_list": invoice_list,
"subtotal": 10,
"salestax": "10%",
"total": 9))
doc.save("new_invoice.docx")

```

3. File **invoice template.docx**

File “invoice_template.docx” berfungsi sebagai template invoice. Template ini telah disiapkan dengan format dan tata letak yang telah ditentukan. Bagian-bagian seperti nama, nomor telepon, tanggal order, tanggal jatuh tempo, nomor invoice, daftar item invoice, subtotal, pajak penjualan, dan total telah ditandai sebagai placeholder yang akan diisi dengan data saat menghasilkan invoice. Placeholder yang digunakan seperti “{{name}}”, “{{phone}}”, “{{tgl_order}}”, “{{tgl_jatuhtempo}}”, “{{no_invoice}}”, “{{item[0]}}”, “{{item[1]}}”, “{{item[2]}}”, “{{item[3]}}”, “{{%tr for item in invoice_list%}}”, “{{%tr endfor%}}”, “{{subtotal}}”, “{{salestax}}”, dan “{{total}}” digunakan untuk menandai posisi di mana data penjualan akan dimasukkan saat menghasilkan invoice. Berikut ini implemnetasinya :



PT SHEILA JAYA
PRODUKSI KONVEKSI

Jl. Pramuka No.41, Nganguk, Mlati
Lor, Kec. Kota Kudus, Kabupaten
Kudus, Jawa Tengah 59319

INVOICE

Order Date : {{ tgl_order }}

Due Date Payment : {{ tgl_jatuhtempo }}

Invoice Number : {{ no_invoice }}

To: {{ name }}

Phone: {{ phone }}

Kuantitas	Deskripsi Barang	Harga Satuan	Line Total
[[%tr for item in invoice_list %]]			
[[item[0]]]	[[item[1]]]	[[item[2]]]	[[item[3]]]
[[%tr endfor %]]			
Subtotal			[[subtotal]]
Sales Tax			[[salestax]]
Total			[[total]]

Thank you for your business!

Payment Informations :
Bank Rakyat Indonesia

No. Rekening : 1020901901xx

a.n PT Sheila Jaya

Regards,

Harry Styles

4. Hasil program aplikasi GUI Python “Generator Invoice”

Aplikasi Penjualan PT Sheila Jaya

PT SHEILA JAYA

First Name: Last Name: Phone:

Tanggal Order: Nomor Invoice: Tanggal Jatuh Tempo:

Kuantitas: Deskripsi Barang: Harga Satuan:

Kuantitas	Deskripsi Barang	Harga Satuan	Total
Total: 0.0			

Aplikasi Penjualan PT Sheila Jaya

PT SHEILA JAYA

First Name: Zayn Last Name: Malik Phone: 7324880190

Tanggal Order: 07/06/2023 Nomor Invoice: 01 Tanggal Jatuh Tempo: 09/06/2023

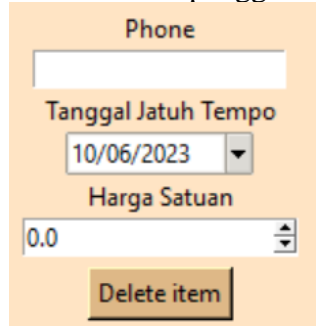
Kuantitas: 1 Deskripsi Barang: pocari Harga Satuan: 7000.0

Kuantitas	Deskripsi Barang	Harga Satuan	Total
1	pocari	7000.0	7000.0
1	beng-beng	2000.0	2000.0
Total: 9000.00			

Jika salah satu item dihapus, maka tampilan total harga akan diperbarui :

Kuantitas	Deskripsi Barang	Harga Satuan	Total
1	pocari	7000.0	7000.0
Total: 7000.00			

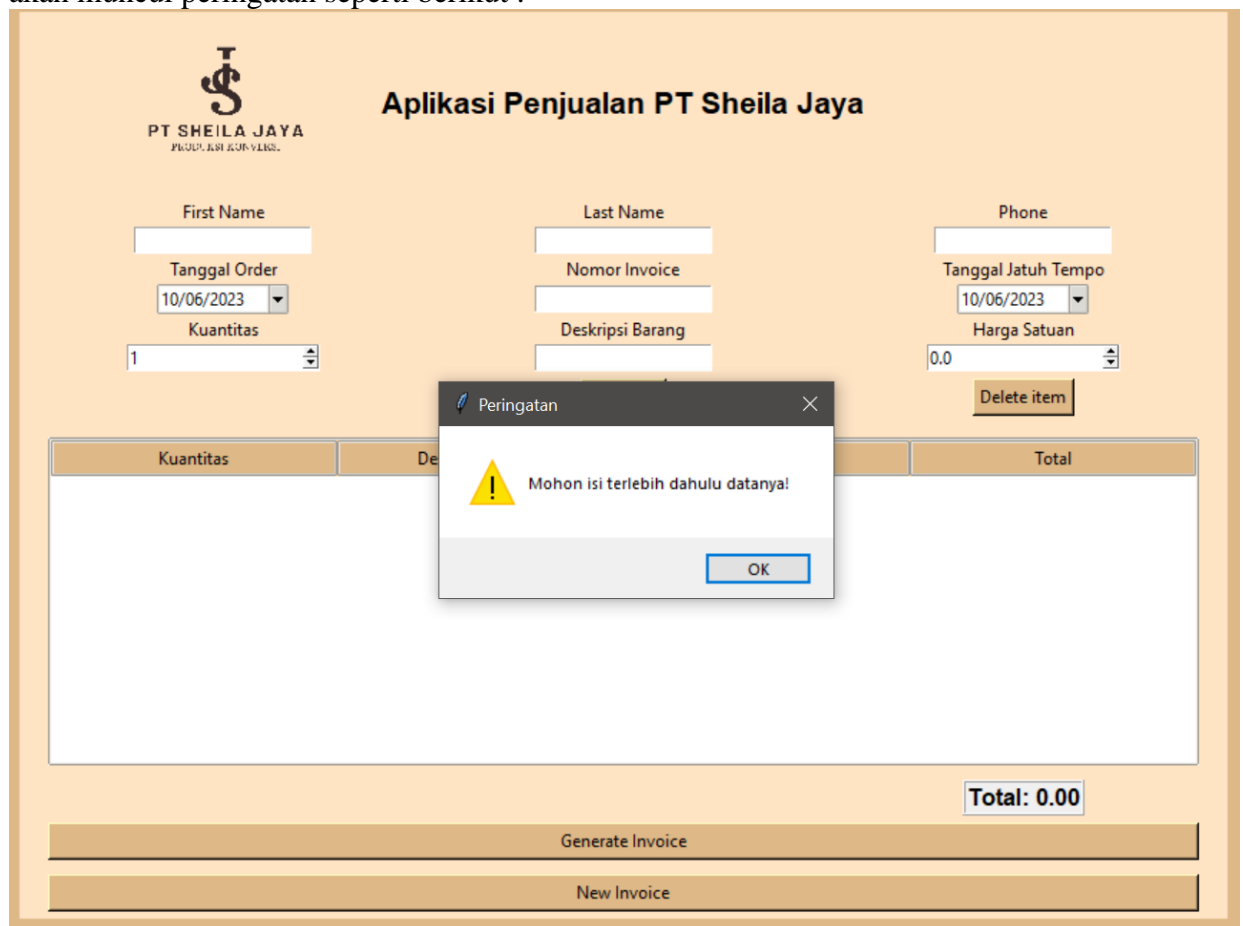
Pada *spin box* untuk “Harga Satuan” memiliki *increment* sebesar 500 agar memudahkan pengguna dalam menambahkan harga.



Phone

Tanggal Jatuh Tempo
10/06/2023
Harga Satuan
0.0
Delete item

Jika data kosong, lalu menekan tombol “Add item” dan “Generate Invoice”, maka akan muncul peringatan seperti berikut :



PT SHEILA JAYA
PRODUKSI KURVING

Aplikasi Penjualan PT Sheila Jaya

First Name

Tanggal Order
10/06/2023
Kuantitas
1

Last Name

Nomor Invoice

Deskripsi Barang

Phone

Tanggal Jatuh Tempo
10/06/2023
Harga Satuan
0.0
Delete item

Peringatan

! Mohon isi terlebih dahulu datanya!

OK

Kuantitas	De	Total

Total: 0.00

Generate Invoice

New Invoice

First Name

Tanggal Order
10/06/2023
Kuantitas
1

Last Name

Nomor Invoice

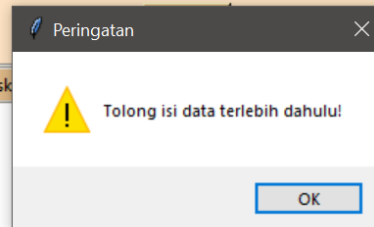
Deskripsi Barang

Phone

Tanggal Jatuh Tempo
10/06/2023
Harga Satuan
0.0

Delete item

Kuantitas	Deskripsi	Total




Total: 0.00

Generate Invoice

New Invoice

5. Contoh hasil *invoice*-nya dalam bentuk dokumen



PT SHEILA JAYA
PRODUKSI KONVERSI

Jl. Pramuka No.41, Nganguk, Mlati
Lor, Kec. Kota Kudus, Kabupaten
Kudus, Jawa Tengah 59319

INVOICE

Order Date : 27/05/2023
Due Date Payment : 27/05/2023
Invoice Number : 01

To: Sheila Deninta
Phone: Maharani


Kuantitas	Deskripsi Barang	Harga Satuan	Line Total
1	baju tidur	160000.0	160000.0
1	tank-top	75000.0	75000.0
1	manset turtleneck	50000.0	50000.0
10	celana dalam katun	20000.0	200000.0
1	kaos oblong	25000.0	25000.0
Subtotal			510000.0
Sales Tax			10.0%
Total			561000.0

Thank you for your business!

Payment Informations :
Bank Rakyat Indonesia
No. Rekening : 1020901901xx
a.n PT Sheila Jaya

Regards,

Harry Styles



PT SHEILA JAYA
PRODUKSI KONVERSI

Jl. Pramuka No.41, Nganguk, Mlati
Lor, Kec. Kota Kudus, Kabupaten
Kudus, Jawa Tengah 59319

INVOICE

Order Date : 07/06/2023
Due Date Payment : 09/06/2023
Invoice Number : 02

To: ZaynMalik
Phone: 7324880190

Kuantitas	Deskripsi Barang	Harga Satuan	Line Total
1	beng-beng	2000.0	2000.0
1	pocari	7000.0	7000.0
Subtotal			9000.0
Sales Tax			10.0%
Total			9900.0

Thank you for your business!

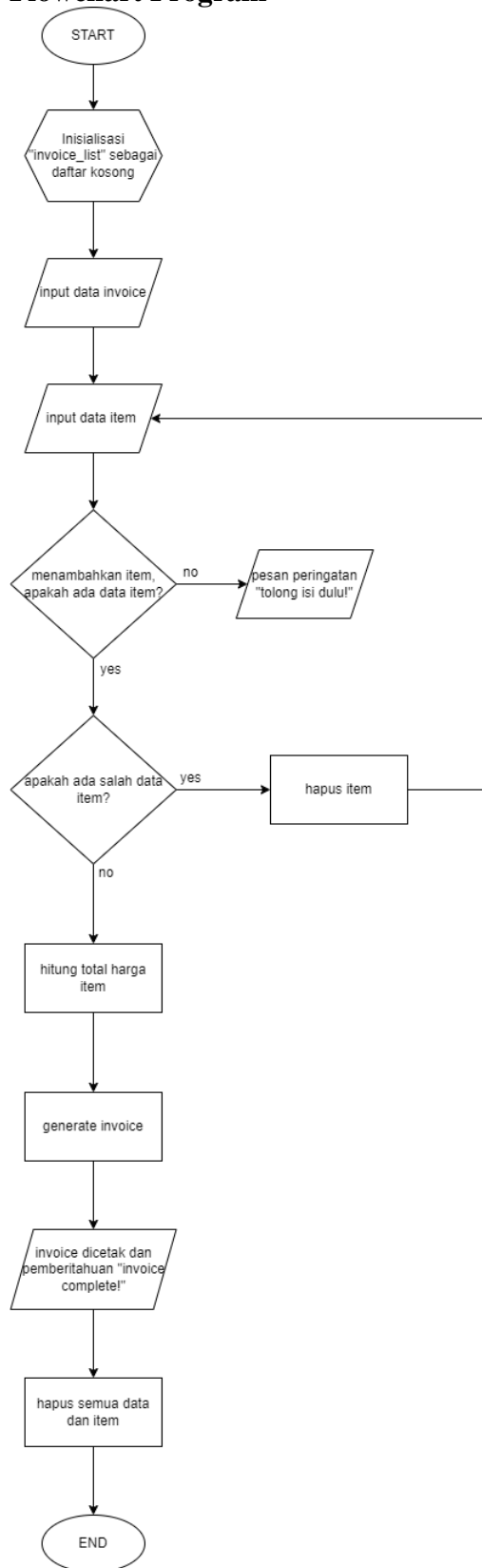
Payment Informations :
Bank Rakyat Indonesia
No. Rekening : 1020901901xx
a.n PT Sheila Jaya

Regards,

Harry Styles

Pada 2 gambar di atas menunjukkan bahwa perhitungan pajak pembelian telah dihitung dan ditulis secara langsung di dokumen invoice berupa .docx.

2.3 Flowchart Program



Berikut adalah urutan dan komponen yang perlu digambarkan dalam flowchart untuk program aplikasi “Invoice Generator” :

- a. Mulai (Simbol: Terminal) : menandakan awal dari program.
- b. Inisiasi “invoice_list” (Simbol : *Decision*) : sebagai penyimpanan data item pembelian dengan daftar kosong terlebih dahulu
- c. Input Data (Simbol: Output) : pengguna memasukkan data seperti nama, nomor telepon, tanggal order, nomor *invoice*, tanggal jatuh tempo, dan tanggal pemesanan.
- d. Tambah Item (Simbol: *Decision*) : data item yang dimasukkan oleh pengguna seperti kuantitas, deskripsi barang, dan harga satuan disimpan dalam daftar *invoice* dan diberikan kondisi jika pengguna mengosongi informasi item ini, maka akan ada peringatan “Mohon isi terlebih dahulu datanya!”.
- e. Penentuan Kondisi Kesalahan Data (Simbol: *Decision*) : jika pengguna mengalami kesalahan peng-input-an data, maka pengguna bisa memilih kondisi dengan menggunakan fitur “Delete Item” untuk menghapus data item yang salah. Kemudian melakukan peng-input-an data baru lagi.
- f. Hitung Total (Simbol: Proses) : jumlah subtotal dihitung berdasarkan harga satuan dan kuantitas item dalam daftar *invoice*.
- g. Generate Invoice (Simbol: Proses) : data *customer* dan daftar item *invoice* digabungkan dengan template dokumen menggunakan modul `docxtpl` untuk menghasilkan invoice.
- h. Tampilan Pesan dan Hasil Cetak *Invoice* (Simbol: Output) : pesan yang memberi tahu pengguna bahwa invoice telah selesai diisi dan dicetak.
- i. Hapus Data (Simbol: Proses) : data yang ada di antarmuka aplikasi dihapus dan aplikasi dikembalikan ke keadaan awal.
- j. Selesai (Simbol: Terminal) : menandakan akhir dari program.

Komponen yang perlu digambarkan dalam flowchart adalah:

- ✓ Simbol Terminal: digunakan untuk menunjukkan awal dan akhir program.
- ✓ Simbol Proses: digunakan untuk menggambarkan langkah-langkah pemrosesan data.
- ✓ Simbol Input/Output: digunakan untuk menunjukkan operasi input atau output data, seperti pengguna memasukkan data atau menampilkan pesan.
- ✓ Panah: menghubungkan simbol-simbol dan menunjukkan urutan aliran program.

Dalam flowchart ini, simbol terminal digambarkan sebagai bentuk *elips*, simbol *decision* digambarkan dengan bentuk segi enam, simbol proses digambarkan sebagai bentuk persegi panjang, dan simbol input/output digambarkan sebagai bentuk jajar genjang. Panah digunakan untuk menghubungkan simbol-simbol dan menunjukkan aliran program dari satu langkah ke langkah berikutnya.

DAFTAR PUSTAKA

Modul ALPRO 23. (2023). Bandung: DASPRO Team.

Sanner, M. F. (1999). Python: a programming language for software integration and development. *J Mol Graph Model*, 17(1), 57-61.