

LAPORAN FINAL PROJECT
Aplikasi Periksa Stok Barang Warung
Berbasis GUI Tkinter Python



Dosen Pengampu:

Dr. Elly Matul Imah, M.Kom.
Riskyana Dewi Intan Puspitasari, M.Kom.

Diusulkan oleh Kelompok 7:

1. Dhani Aditya Putra Efendi (22031554038)
2. Fadhilah Nuria Shinta (22031554003)
3. Daffa Fazly Rashidan (22031554006)

PRODI SAINS DATA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN
ALAM
UNIVERSITAS NEGERI SURABAYA

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sebuah program umumnya memiliki model eksekusi yang memungkinkan komputer untuk menjalankannya secara langsung. Model ini terdiri dari serangkaian instruksi yang digunakan oleh komputer untuk melakukan tugas tertentu, dan instruksi-instruksi tersebut ditulis dalam bahasa pemrograman, yang merupakan bahasa yang dimengerti oleh komputer. Bahasa pemrograman adalah sebuah sintaks yang digunakan untuk mendefinisikan program komputer, dan melalui bahasa ini, seorang programmer dapat membuat berbagai aplikasi. Contoh-contoh bahasa pemrograman yang umum digunakan adalah Python, C++, dan Java. Dalam tugas ini, kami memilih untuk menggunakan bahasa pemrograman Python karena alasan bahwa Python relatif mudah dipahami dan juga karena mata kuliah Pemrograman Dasar kami menggunakan bahasa tersebut.

Menurut pengertian dari Python Software Foundation (2016), Python adalah bahasa pemrograman interpretatif, berorientasi objek dan semantik yang dinamis. Python memiliki *high-level* struktur data, *dynamic typing* dan *dynamic binding*. Python mempunyai sintaks sederhana dan mudah dipelajari untuk mempermudah pembacaan dan python juga gratis. Python bisa dibilang bahasa pemrograman dengan tujuan umum yang dikembangkan secara khusus untuk membuat source code mudah dibaca. Python juga memiliki *library* yang lengkap sehingga memungkinkan programmer untuk membuat aplikasi yang mutakhir dengan menggunakan *source code* yang tampak sederhana (Ananda et al., 2018). Dalam membuat aplikasi pada python tkinter adalah salah satu alat untuk bahasa pemrograman Python. Python dan tkinter adalah kombinasi sempurna jika Anda ingin membuat GUI dengan cepat dan mudah. Tkinter adalah antarmuka pengguna grafis (GUI) Python standar yang digunakan untuk membuat tampilan aplikasi dengan komponen modul tkinter seperti tombol, kotak teks, label, bingkai, jendela, yang mendukung pembuatan aplikasi GUI dengan sangat baik.

Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi *output* (Jogiyanto, 1999:12). Sementara menurut Hengki W. Pramana, Aplikasi adalah suatu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti *system* perniagaan, game pelayanan masyarakat, periklanan, atau semua proses yang hampir dilakukan manusia. Fungsi

aplikasi adalah untuk mempermudah dalam melakukan suatu hal agar tidak membuang waktu dan tenaga. Aplikasi yang kita rancang saat ini adalah aplikasi “Periksa stok barang warung” di mana aplikasi tersebut dapat membantu seorang manajer dan karyawan untuk mengelola data stok barang. Dalam rangka mendukung pengelolaan stok barang yang lebih baik, proyek aplikasi memeriksa stok barang menggunakan library Tkinter ini dibangun. Tkinter merupakan *library* GUI (*Graphical User Interface*) standar untuk bahasa pemrograman Python. Dengan menggunakan Tkinter, pengembang dapat dengan mudah membuat antarmuka pengguna yang menarik dan intuitif. Tujuan dari proyek ini adalah mengembangkan aplikasi yang dapat membantu perusahaan atau toko dalam mengelola stok barang dengan lebih efisien. Aplikasi ini akan memungkinkan pengguna untuk melihat daftar barang, memeriksa jumlah stok, menambahkan atau mengedit informasi barang, serta mencetak laporan stok yang diperlukan.

BAB II

Landasan Teori

2.1 Python

Python merupakan bahasa pemrograman tingkat tinggi yang sering digunakan dalam berbagai bidang seperti pengembangan perangkat lunak, pengembangan web, analisis data, kecerdasan buatan, dan lain-lain. Salah satu prinsip desain Python adalah fokus pada keterbacaan kode, sehingga memudahkan pengembang untuk menulis kode yang mudah dipahami dan terstruktur (Andhika, 2023).

Python merupakan bahasa pemrograman yang interpretatif, artinya kode Python dieksekusi baris perbaris oleh interpreter Python. Hal ini memungkinkan pengembang untuk melakukan uji coba dan pengujian kode dengan cepat tanpa perlu mengompilasi seluruh program. Selain itu, Python juga memiliki dukungan yang luas untuk berbagai platform dan sistem operasi, termasuk Windows, macOS, dan Linux.

Salah satu keunggulan utama Python adalah ekosistemnya yang kaya dengan pustaka dan kerangka kerja yang siap pakai. Ada banyak pustaka yang tersedia seperti NumPy, Pandas, dan Matplotlib yang digunakan dalam analisis data dan ilmu data. Untuk pengembangan web, terdapat kerangka kerja seperti Django dan Flask. Python juga mendukung pengembangan aplikasi mobile, pengembangan game, kecerdasan buatan, dan masih banyak lagi melalui pustaka-pustaka yang tersedia.

2.2 GUI

Graphical User Interface atau GUI telah mengubah cara pengguna berinteraksi dengan komputer dan aplikasi perangkat lunak. GUI menyajikan antarmuka yang menarik secara visual dan mudah digunakan yang memungkinkan pengguna untuk menjelajahi program, mengakses fitur, dan memanipulasi data dengan mudah. Daripada bergantung pada perintah berbasis teks dan sintaks yang kompleks, GUI memungkinkan pengguna berinteraksi dengan elemen grafis seperti jendela, tombol, menu, dan ikon. Representasi visual ini mempermudah pembelajaran dan memberdayakan pengguna untuk melakukan tugas tanpa perlu mengingat perintah yang kompleks atau bahasa pemrograman.

Selain itu, GUI memungkinkan integrasi elemen multimedia seperti

gambar, video, dan audio, yang meningkatkan pengalaman pengguna dan memfasilitasi komunikasi yang efektif. Representasi visual, ikon, dan grafik dapat menyampaikan informasi dengan lebih efisien daripada teks biasa, sehingga meningkatkan pemahaman dan mempercepat pengambilan keputusan.

Dengan menggunakan GUI, pengguna dapat melihat hasil tindakan mereka secara langsung melalui elemen interaktif seperti tombol yang berubah penampilannya ketika diklik atau bidang teks yang menampilkan masukan secara real-time. Mekanisme umpan balik ini meningkatkan produktivitas pengguna, karena mereka dapat dengan cepat melakukan iterasi dan melakukan penyesuaian berdasarkan petunjuk visual yang diberikan oleh GUI.

2.3 Jump Search

Seperti *Binary Search*, *Jump Search* adalah algoritma pencarian untuk *array* yang sudah diurutkan. Ide dasarnya adalah untuk memeriksa lebih sedikit elemen dengan melompat maju dengan langkah-langkah tetap atau melewati beberapa elemen untuk mencari semua elemen (Nayal, 2016).

Dalam algoritma ini, *array* dibagi menjadi beberapa blok (misalnya, dengan ukuran m). Algoritma ini mencari elemen dalam satu blok, dan jika elemen tidak ditemukan, maka melompat ke blok berikutnya. Ketika algoritma menemukan blok yang berisi elemen yang dicari, ia menggunakan algoritma pencarian linier untuk menemukan indeks yang tepat. Dengan demikian, *Jump Search* mencoba mengurangi jumlah perbandingan yang diperlukan dibandingkan dengan pencarian linier, namun tetap lebih lambat daripada pencarian biner. Ini adalah alternatif yang efisien untuk pencarian linier dalam *array* yang diurutkan.

2.4 Linear Search

Linear search, juga dikenal sebagai *sequential search*, adalah algoritma pencarian sederhana yang digunakan untuk mencari elemen tertentu dalam suatu *list* atau *array* (Alza.2020). Algoritma ini bekerja dengan memeriksa setiap elemen secara berurutan dari awal hingga akhir list sampai elemen yang dicari ditemukan atau seluruh list telah diperiksa.

Proses *linear search* dimulai dengan membandingkan elemen pertama dengan elemen yang dicari. Jika elemen tersebut cocok, pencarian selesai dan elemen tersebut ditemukan. Jika tidak cocok, maka algoritma akan melanjutkan dengan membandingkan elemen kedua dengan elemen yang dicari, dan

seterusnya. Proses ini terus berlanjut sampai elemen yang dicari ditemukan atau seluruh *list* telah diperiksa. Jika elemen yang dicari tidak ada dalam *list*, maka *linear search* akan menyelesaikan pencarian dengan mengunjungi setiap elemen dalam *list*

2.5 Tkinter

Tkinter (Tk Interface) adalah sebuah pustaka standar untuk pembuatan antarmuka GUI (*Graphical User Interface*) dalam bahasa pemrograman Python, yang didasarkan pada Tk GUI Toolkit. Tkinter menyediakan berbagai macam elemen grafis yang memudahkan pembuatan program dengan antarmuka grafis. Setiap Toolkit GUI menyediakan *widget*, yang merupakan objek antarmuka pengguna seperti *button*, *scroll bar*, *listbox*, *check button*, *radio button*, label text Widget ini mengemas implementasi rinci dan memiliki perilaku default yang telah didefinisikan, sehingga mempermudah pengembangan program GUI (Fajri, 2020)

2.6 Data set CSV

CSV adalah suatu format data dalam basis data di mana setiap *record* terpisahkan dengan tanda koma atau titik koma. Data yang berakhir dengan ekstensi CSV umumnya digunakan untuk bertukar data yang berjumlah besar antar aplikasi yang berbeda. Data CSV ini dirancang untuk memudahkan mengekspor data dan mengimpornya ke program lain (Nora, 2020)

BAB III

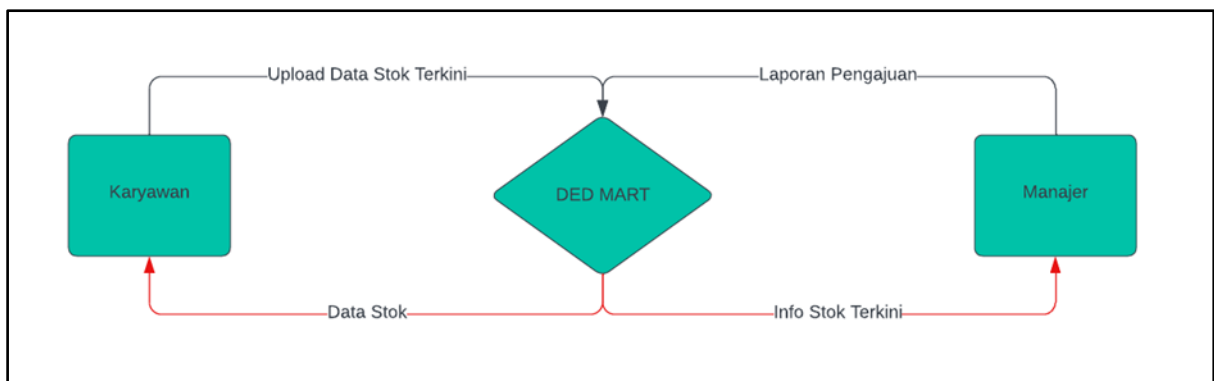
Desain Proyek

3.1 Perancangan Sistem

Perancangan sistem adalah proses merencanakan, merancang, dan menggambarkan struktur, komponen, dan interaksi suatu sistem dengan tujuan untuk mengoptimalkan kinerja sistem, mengidentifikasi kebutuhan dan kendala, serta memfasilitasi pengembangan dan implementasi sistem.

a. Diagram Konteks

Diagram konteks dalam perancangan sistem adalah gambaran grafis yang menunjukkan sistem yang sedang dirancang dan cara interaksinya dengan dunia luar. Diagram ini memberi kita gambaran tentang apa yang ada di dalam sistem dan bagaimana sistem tersebut berhubungan dengan orang, organisasi, atau sistem lain di sekitarnya. Adapun Diagram Konteks dari aplikasi ini adalah sebagai berikut :



Gambar 3.1, Gambar Diagram Konteks

Penjelasan dari diagram konteks di atas Adalah:

Masukan berupa:

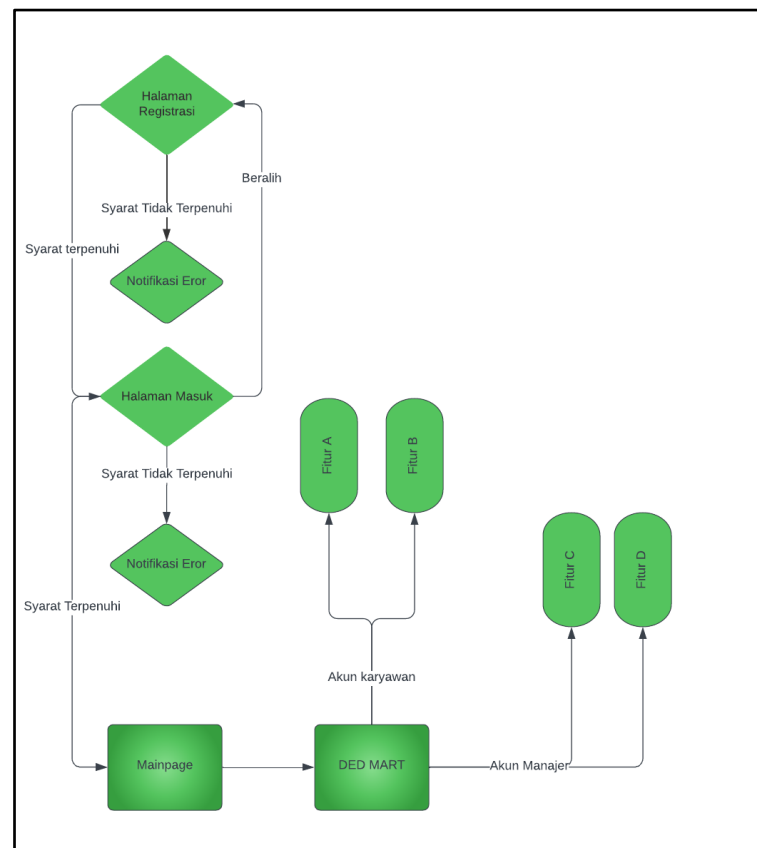
- 1.Karyawan : Berupa input data stok terkini terhadap data stok warung.
- 2.Manajer : Pembuatan Laporan Pengajuan dengan input stok data terkini.

Keluaran berupa:

- 1.Karyawan : Data *template* stok warung awal.
- 2.Manajer : Mendapatkan info stok terkini

b. Diagram Alir Sistem

Merupakan bagian dari perancangan sistem yang menggunakan representasi grafis untuk menunjukkan urutan langkah-langkah kerja atau aliran data yang harus dibuat oleh programmer. Hal ini bertujuan untuk menentukan langkah-langkah kerja mulai dari tahap awal hingga mencapai pembuatan desain yang diperlukan oleh pengguna.



Gambar 3.2, Diagram Alir Sistem

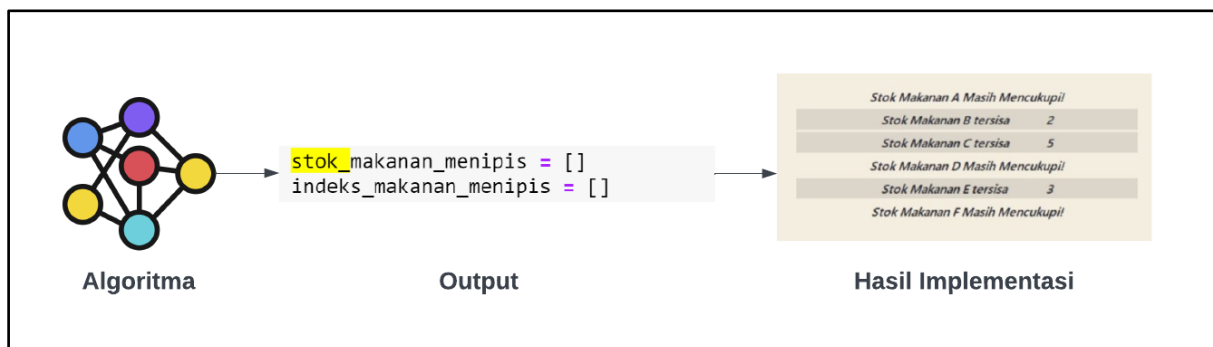
Penjelasan spesifik terkait:

1. Syarat Terpenuhi Halaman Register
 - a) Semua *Entry* pada halaman wajib terisi,
 - b) Email harus menggunakan Email yang berbeda dari Email yang telah didaftarkan sebelumnya,
 - c) Input password harus sama dengan input konfirmasi password
 - d) Nomor telepon pada *Entry* harap dimasukan menggunakan digit angka,
 - e) Entry Jabatan jika ***user* sebelum telah terdaftar sebagai manajer maka selanjutnya tidak diperkenankan adanya manajer kedua,**
 - f) Mewajibkan penggunaan '@' pada Entry email,

2. Syarat terpenuhi Halaman Login
 - Semua Entry pada halaman wajib terisi,
 - Email yang di inputkan telah terdaftar sebelumnya,
 - Password yang di inputkan harus sesuai dengan *password* email yang di inputkan,
3. Pada Halaman Mainpage terdapat macam macam-macam usaha perusahaan yang dimiliki
4. Dan dalam proyek ini kami membuat salah satu usaha berupa ‘DED MART’ yang di dalamnya terdapat masing-masing fitur yang dibedakan melalui akun yang berlabel **karyawan** yang memiliki fitur upload data stok dan download template awal dengan akun yang berlabel **manajer** memiliki fitur berupa check barang untuk mengetahui stok terkini dan fitur pembuat laporan pengajuan dengan input stok terkini.

3.2 Implementasi algoritma

Dalam proyek ini menerapkan algoritma searching yang disebut jump search dan sequential search yang dijadikan 2 perbandingan, yang telah dipelajari dalam mata kuliah struktur data dan algoritma di perkuliahan. Algoritma ini digunakan untuk mencari stok barang yang kurang dari 5 sebagai indikator stok menipis dalam fitur "check barang" yang tersedia bagi akun yang berlabel manajer. Kompleksitas waktu terburuk algoritma jump search adalah $(O(\sqrt{n}))$ sedangkan untuk sequential search sendiri memiliki kompleksitas waktu terburuk berupa $O(n)$. Informasi tentang stok barang yang menipis ini akan ditampilkan kepada manajer, sehingga mereka dapat mengetahui barang-barang mana yang stoknya menipis. Selanjutnya, manajer dapat membuat file pengajuan berdasarkan informasi tersebut.

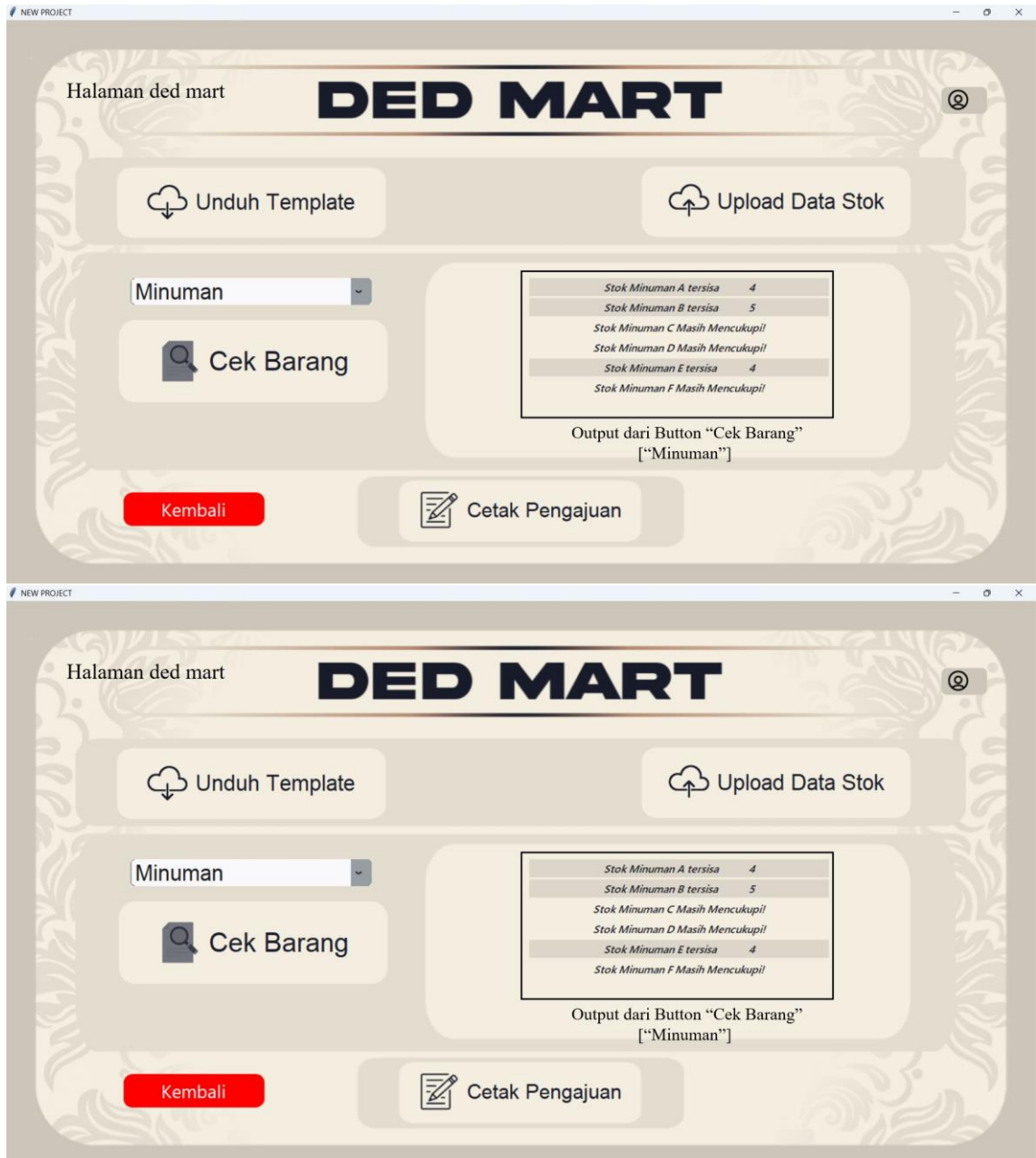


Gambar 3.3, Implementasi Algoritma

BAB IV Implementasi

4.1 Linear Search

- Implementasi linear search disini dapat dilihat dari fitur cek barang.
- User dapat melihat stock makanan, minuman dan snack yang telah di *upload* oleh karyawan dalam fitur 'Upload Data Stock'.





Hasil yang keluar akan urut dari stock (makanan/minuman/snack) A, B, C hingga F

4.2 Jump Search

- Implementasi *Jump Search* pada aplikasi ini dapat dilihat pada fitur cek barang seperti linear search tadi
- Pada algoritma *jump search*, output yang keluar tidak sama seperti linear search.
- Output yang keluar menjadi tidak urut

Output pada algoritma *jump search* :





BAB V

Kesimpulan

Dari proyek yang kami lakukan untuk membandingkan dua algoritma *searching*, yaitu *linear search* dan *jump search*, dapat ditarik kesimpulan bahwa algoritma *linear search* lebih cocok digunakan dalam konteks aplikasi ini. Hal ini disebabkan oleh beberapa faktor, terutama dalam hal implementasi data stok pada dataset.

Penggunaan algoritma *jump search* dalam kasus ini membutuhkan pengurutan data stok terlebih dahulu sebelum dilakukan pencarian. Namun, pengurutan data ini menyebabkan perubahan indeks, yang pada gilirannya dapat menyebabkan kesalahan (error) ketika data yang diurutkan ditampilkan secara terurut pada aplikasi melalui indeks. Ini mengindikasikan bahwa penggunaan algoritma *jump search* tidak cocok dalam konteks ini, mengingat kesulitan dalam menampilkan data yang terurut dengan benar.

Di sisi lain, algoritma *linear search*, yang tidak memerlukan pengurutan data sebelum pencarian, memungkinkan implementasi yang lebih baik dalam hal menampilkan data yang terurut pada aplikasi. Hal ini dapat dilakukan tanpa perubahan indeks yang signifikan. Oleh karena itu, algoritma *linear search* lebih direkomendasikan dalam kasus ini untuk menjaga kestabilan dan keakuratan tampilan data stok pada aplikasi.

Kesimpulan tersebut didasarkan pada observasi bahwa implementasi algoritma *linear search* tanpa pengurutan data awal memberikan hasil yang lebih baik dalam konteks penggunaan aplikasi yang mencakup tampilan data stok secara terurut.

BAB VI

DAFTAR PUSTAKA

- Pratama, Andhika P. (2023). Hashnode. Diambil dari [Pengenalan bahasa pemrograman Python \(hashnode.dev\)](https://hashnode.dev)
- Imamah, F., & Does, A. (2018). Aplikasi chatbot (milki bot) yang terintegrasi dengan web CMS untuk customer service pada UKM MINSU. *Jurnal Cendikia*, 16(2 Oktober), 100-106.
- Jogiyanto, H. M. (1999). Analisis dan Desain Sistem Informasi: pendekatan terstruktur teori. *Praktek dan Aplikasi Bisnis, Andi Offset, Yogyakarta*.
- Alza.(2020). koding.alza.web.id. Diakses tanggal 10 Juni 2023, [Algoritma linear search - Belajar koding untuk pemula \(alza.web.id\)](https://koding.alza.web.id)
- Bastian, Rio. (2021). Algoritma dan Pemrograman 2 Jump Search. Palembang : Studocu.
- Nayal, C. (2016). Jump Search. Retrieved from [Jump Search - GeeksforGeeks](https://www.geeksforgeeks.org/jump-search/)
- Fajri, T. Rahmad Effendi. (2020). Sistem Absensi Berbasis Pengenalan Wajah Secara Real Time menggunakan Metode Fisherface. *Jurnal Nasional Informatika dan Teknologi Jaringan*, Available at <https://doi.org/10.30743/infotekjar.v4i2.2377>
- Asmarani, Nora Galuh. (2020). Apa itu Data CSV?. Diambil dari [Apa Itu Data CSV? \(ddtc.co.id\)](https://ddtc.co.id)
- Porter M (2013), Creating Shared Value as Business Strategy, Harvard Business Review

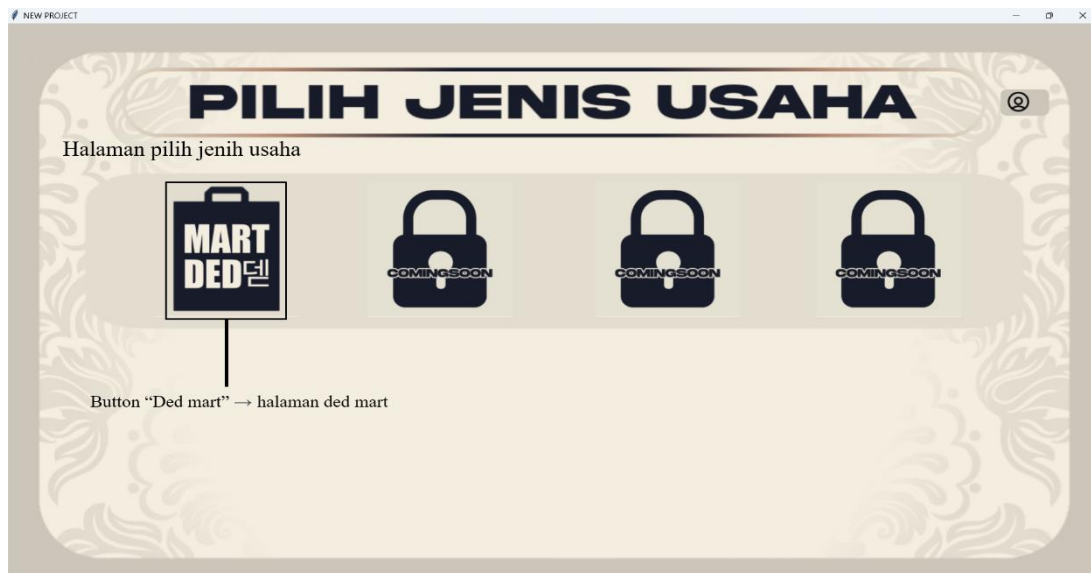
BAB VII Lampiran

1. User Manual Guide

- **Menu register** : User diminta untuk melakukan registrasi terlebih dahulu dengan memasukkan **Nama Depan, Nama belakang, Jabatan, No. Handphone, Email, Password, Konfirmasi password**.
- **Entry email harus menggunakan @ dan No. handphone harus menggunakan digit.**
- Dipastikan bahwa user harus memasukkan alamat email dengan benar, jika salah maka user tidak dapat melakukan registrasi.

- Jika user telah memiliki akun maka bisa langsung masuk ke halaman login
- **Halaman log in** : User diminta mengisi email dan password sesuai yang telah didaftarkan sebelumnya.

- **Home Page** : Setelah berhasil login maka user masuk ke dalam menu home page yang berisi pilihan jenis usaha



- Fitur yang tersedia selanjutnya yakni **fitur untuk karyawan** yang meliputi ‘unduh template’, ‘Entry Data Stok’, dan ‘Cetak Pengajuan’.
- Sedangkan untuk **manajer** yakni fitur ‘Cek Barang’



- **Fitur Entry Stok Data** : User dengan jabatan karyawan dapat mengisi stok data barang yang tersedia hari itu dan nantinya stok tersebut akan masuk ke file csv.
- Setelah mengisi, user dapat menyimpan, lalu kembali atau menyimpan, lalu log out.

NEW PROJECT

ENTRY STOK DATA

Halaman entry stok

Menuju ke halaman register

Logout

Entry data stok

Makanan A	Makanan B	Makanan C	Makanan D	Makanan E	Makanan F
5	6	5	2	5	6
Minuman A	Minuman B	Minuman C	Minuman D	Minuman E	Minuman F
4	5	6	7	4	9
Snack A	Snack B	Snack C	Snack D	Snack E	Snack F
6	4	5	3	4	5

Output dari button simpan akan masuk ke file csv dari template

Simpan

Kembali

-kembali ke halaman sebelumnya

Information!!

Data Telah Disimpan!!

OK

- Pengajuan Stok : Disini karyawan dapat mengajukan jumlah stok barang kepada manajer untuk di tambah di warungnya nanti.
- Stok yang diajukan akan tersimpan dalam format docx

NEW PROJECT

PENGAJUAN STOK

Halaman pengajuan stok

Entry nama pembuat surat

Entry stok yang diminta

Nama

Manajer

Makanan A	Makanan B	Makanan C	Makanan D	Makanan E	Makanan F
88	88	88	88	88	88
Minuman A	Minuman B	Minuman C	Minuman D	Minuman E	Minuman F
88	88	88	88	88	88
Snack A	Snack B	Snack C	Snack D	Snack E	Snack F
88	88	88	88	88	88

Simpan

Kembali

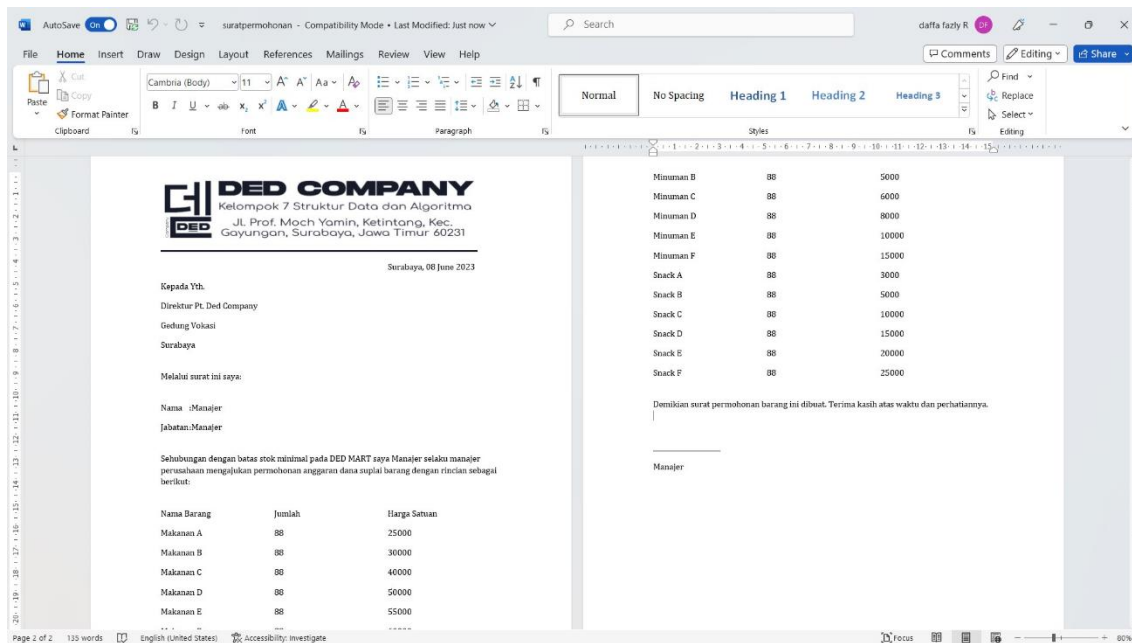
Stok yang telah diminta akan disimpan dengan format .docx

Information!!

Data telah disimpan sebagai Document!!

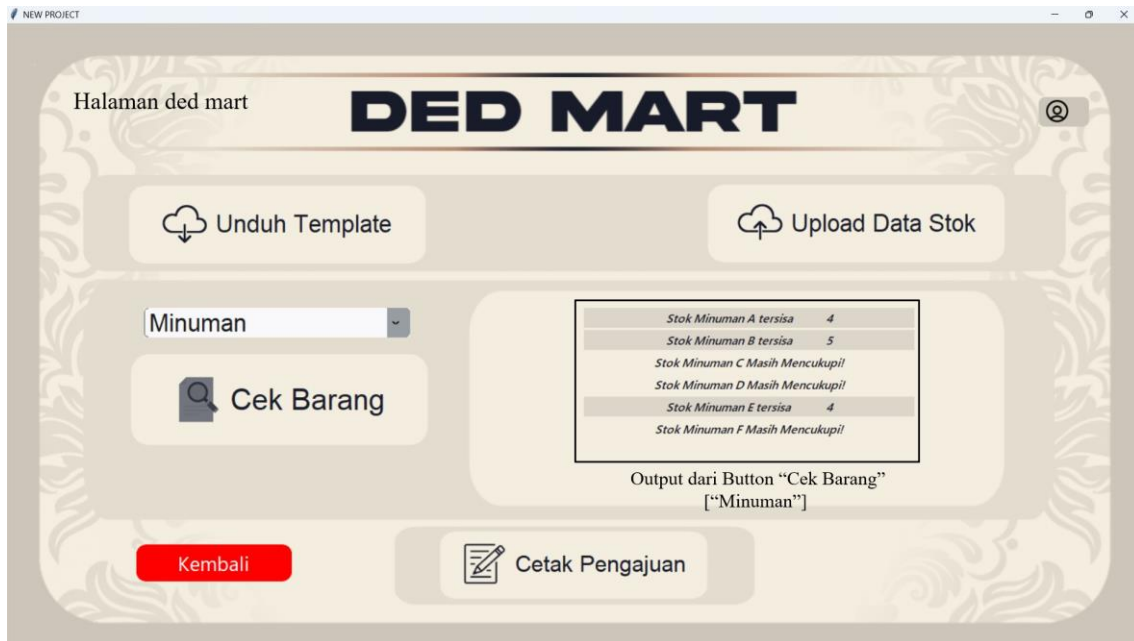
OK

- Berikut adalah tampilan docx untuk cetak pengajuan



- Pada jabatan manajer, di sini user dapat mengecek barang (makanan, minuman, dan snack) yang sebelumnya sudah diupload oleh karyawan
- Hasil dari stok barang yang kurang dari 5 akan muncul pada sebelah kanan
- Jika **stok masih mencukupi**, maka akan muncul tulisan Stok Makanan A Masih Mencukupi





2. Listing Code

Halaman Registrasi

```
def regispage(self):
    def regis2():
        #Andisi mewajibkan semua entry terisi dengan baik
        if self.nama_depan_entry.get() and self.nama_belakang_entry.get() and self.email_entry.get() and self.password_entry.get() and self.jabatan_b
        dataset = pd.read_csv('data_user1.csv')
        # Membuat list seluruh email yg ada di csv
        email = []
        for x in dataset.email:
            email.append(x)

        #jika ada email yang sama pada data set
        if self.email_entry.get() in email:
            self.wajib = Label(self.register_frame, background='#f4eee0', text='Email telah digunakan!', fg='red', font=("yu gothic ui", 12), rel
            self.wajib.place(x=670, y=530)

        else:
            #nomor telfon yang diinputkan harus berupa angka
            nomortelfon=self.nomor_entry.get()
            if nomortelfon.isdigit():
                nomortelfon=[]
                for y in dataset.nomor:
                    nomortelfon.append(y)

            #mengubah widget entry pada nomor harus diinputkan berupa angka
            if self.nomor_entry.get() in nomortelfon:
                self.wajib = Label(self.register_frame, background='#f4eee0', text='Nomor telfon telah digunakan!', fg='red', font=("yu gothi
                self.wajib.place(x=670, y=530)

            else:
                jabatan=[]
                for z in dataset.jabatan:
                    jabatan.append(z)

                if self.jabatan_box.get() == 'Manajer' and 'Manajer' in jabatan:
                    self.wajib = Label(self.register_frame, background='#f4eee0', text='Manajer sudah ada', fg='red', font=("yu gothic u
                    self.wajib.place(x=670, y=530)

            else:
                if "0" in self.email_entry.get():
                    if self.password_entry.get() == self.konfirmasipw_entry.get():
                        #menyimpan data dari index
                        dataset.loc[len(dataset.index)] = [self.nama_depan_entry.get(), self.nama_belakang_entry.get(), self.email_entr
                        dataset.to_csv("data_user1.csv", index=False, header=True)

                        #Frame timpa dari register
                        self.timpa_frame = Frame(self.register_frame, bg='#1d0529', width=1460, height=779)
                        self.timpa_frame.place(x=0,y=0)

                        #Frame
                        self.lg_frame = Frame(self.timpa_frame, bg='#f4eee0', width=1460, height=779)
                        self.lg_frame.place(x=0,y=0)
                        self.bgrn_image = ImageTk.PhotoImage(Image.open('gambar/bgrn2new.png'))
                        self.bgrn2 = Label(self.lg_frame, image=self.bgrn_image, background="#ccc6ba").place(x=0, y=0)

                        #Email
                        self.email_label = Label(self.lg_frame, text="Email", bg="#e2ddce", fg="white",
                                                font=("yu gothic ui", 30, "bold",))
                        self.email_label.place(x=690, y=180)
                        self.email_entry = Entry(self.lg_frame, highlightthickness=0, relief=FLAT, bg="white", fg="#999999",
                                                font=("yu gothic ui ", 30, "bold"))
                        self.email_entry.place(x=495, y=237, width=500)
```

Halaman Login

```
def login():
    #kondisi mewajibkan semua entry terisi dengan baik
    if self.nama_depan_entry.get() and self.nama_belakang_entry.get() and self.email_entry.get() and self.password_entry.get() and self.jabatan_b:
        dataset = pd.read_csv('data_user1.csv')
        # Membuat list seluruh email yg ada di csv
        email = []
        for x in dataset.email:
            email.append(x)

        #jika ada email yang sama pada data set
        if self.email_entry.get() in email:
            self.wajib = Label(self.register_frame, background='#f4eee0', text='Email telah digunakan!', fg='red', font=("yu gothic ui", 12), rel
            self.wajib.place(x=380,y=425)

        else:
            #nomor telfon yang diinputkan harus berupa angka
            nomortelfon=self.nomor_entry.get()
            if nomortelfon.isdigit():
                nomortelfon=[]
                for y in dataset.nomor:
                    nomortelfon.append(y)

                #mengubah widget entry pada nomor harus diinputkan berupa angka
                if self.nomor_entry.get() in nomortelfon:
                    self.wajib = Label(self.register_frame, background='#f4eee0', text='Nomor telfon telah digunakan!', fg='red', font=("yu gothi
                    self.wajib.place(x=670, y=530)

                else:
                    jabatan=[]
                    for z in dataset.jabatan:
                        jabatan.append(z)

                    if self.jabatan_box.get() == 'Manajer' and 'Manajer' in jabatan:
                        self.wajib = Label(self.register_frame, background='#f4eee0', text='Manajer sudah ada', fg='red', font=("yu gothic ui",
                        self.wajib.place(x=380,y=320)

                    else:
                        if "@" in self.email_entry.get():
                            if self.password_entry.get() == self.konfirmasipw_entry.get():
                                #menyimpan data dari index
                                dataset.loc[len(dataset.index)] = [self.nama_depan_entry.get(), self.nama_belakang_entry.get(), self.email_entr
                                dataset.to_csv("data_user1.csv", index=False, header=True)

                                #Frame timpa dari register
                                self.timpa_frame = Frame(self.register_frame, bg='#1d0529', width=1460, height=779)
                                self.timpa_frame.place(x=0,y=0)

                                #Frame
                                self.lg_frame = Frame(self.timpa_frame, bg='#f4eee0', width=1460, height=779)
                                self.lg_frame.place(x=0,y=0)
                                self.bgrn_image = ImageTk.PhotoImage(Image.open('gambar/bgrn2new.png'))
                                self.bgrn2 = Label(self.lg_frame, image=self.bgrn_image, background="#ccc6ba").place(x=0, y=0)

                                #Email
                                self.email_label = Label(self.lg_frame, text="Email", bg="#e2ddce", fg="#181b29",
                                                            font=("yu gothic ui", 30, "bold"))
                                self.email_label.place(x=690, y=180)
                                self.email_entry = Entry(self.lg_frame, highlightthickness=0, relief=FLAT, bg="white", fg="#999999",
                                                            font=("yu gothic ui ", 30, "bold"))
                                self.email_entry.place(x=495, y=237, width=500)
```



```
if self.email_entry.get() and self.password_entry.get():
    # save to csv
    dataset = pd.read_csv('data_user1.csv')

    # Menaut list yg berisi semua email dan pw
    email = []

for i in dataset.email:
    email.append(i)

# Kondisi untuk cek email dan nomor telpon yang sesuai sesuai dengan email di dataset
if self.email_entry.get() in email:

    # Cara untuk mengetahui index dari email yg di input
    self.idx = email.index(self.email_entry.get())

    # Kondisi untuk cek password yg di input sama dgn password dataset
    if self.password_entry.get() == dataset.password[self.idx]:

        #tampilkan next page

        self.tampilan = Frame(self.window, bg="#f4eeeb", width=1468, height=779)
        self.tampilan.place(x=35, y=35)
        self.bgrn_image = ImageTk.PhotoImage(Image.open('gambar/bgprplussha.png'))
        self.bgrn_label = Label(self.tampilan, image=self.bgrn_image, background="#ccccba").place(x=0, y=0)

        def logoutacc():
            def dest():
                self.dons.destroy()
                self.dons = customtkinter.CTkFrame(master=self.tampilan, fg_color="#ccccba", width=200, height=110)
                self.dons.place(x=1250, y=65)
                self.batal_customtkinter.CTkButton(master=self.dons, text="", text_font=('Arial', 30), corner_radius=8, text_color="#81b29", fg_color="white", image=self.img_logoutacc, image_open=('gambar/logoutacc.png'))
                self.logoutacc_customtkinter.CTkButton(master=self.dons, text='Logout', image=self.img_logoutacc, text_font=('Arial', 20), text_color="black", font_weight="bold")
                messagebox.showinfo("Output Berhasil!")
            self.imgacc = ImageTk.PhotoImage(Image.open('gambar/acc.png'))
            self.logoutaccada_customtkinter.CTkButton(master=self.tampilan, text='', image=self.imgacc, text_font=('Arial', 20), text_color="#81b29", font_weight="bold")

        #messagebox.showinfo("Information!!","Opsi ini akan segera hadir")
        def soon():
            messagebox.showinfo("Information!!","Opsi ini akan segera hadir")
            #Tampilkan Usaha Penjualan dan Label
            self.soonimg = ImageTk.PhotoImage(Image.open('gambar/soon.png'))
            self.marketing = ImageTk.PhotoImage(Image.open('gambar/dedmarkt2.png'))
            self.atkded = ImageTk.PhotoImage(Image.open('gambar/atkdcd.png'))
            self.btnmarkt = customtkinter.CTkButton(self.tampilan, text='', text_font=('Arial', 20), corner_radius=25, fg_color="#e2dce", hover_color="#e2dce", text_color="black", font_weight="bold")
            self.btnmarkt.place(x=170, y=205)

            self.btnsoon = customtkinter.CTkButton(master=self.tampilan, text='', image=self.atkded, text_font=('Arial', 20), text_color="#e2dce", text_color="black", font_weight="bold")
            self.btnsoon.place(x=470, y=205)

            self.btnsoon = customtkinter.CTkButton(master=self.tampilan, text='', image=self.soonimg, text_font=('Arial', 20), text_color="#e2dce", text_color="black", font_weight="bold")
            self.btnsoon.place(x=790, y=205)

            self.btnsoon = customtkinter.CTkButton(master=self.tampilan, text='', image=self.soonimg, text_font=('Arial', 20), text_color="#e2dce", text_color="black", font_weight="bold")
            self.btnsoon.place(x=1100, y=205)
```

```
def dodata():
    if dataset.jabatan[isif_ide] == "Karyanan":
        def template():
            data = ('Data Di peroleh ({},{},{},{},{},{},{},{},{})'.format('Makanan' if Makanan A 'Makanan B' 'Makanan C' 'Makanan D' 'Makanan E' 'Makanan F'; 'Minuman' if Minuman A 'Minuman B' 'Minuman C' 'Minuman D' 'Minuman E' 'Minuman F'; 'Stok Makanan' [5,5,5,5,5,5], 'Marga Minuman' [20
                'Snack A', 'Snack B', 'Snack C', 'Snack D', 'Snack E', 'Snack F']; 'Stok Snack' [5,5,5,5,5,5], 'Marga Snack' [20000,30000,40000,
            df.to_csv(data_stok.csv)
            df.to_csv("data_stok.csv", index = False)

        messagebox.showinfo("Information!", "Data Telah Di Simpan!")

    def entry_barang():
        def entry_datastok(frame, x, y, text):
            label = Label(frame, text=text, bg="#f0d0d0", fg="#333333", font=("Roboto Bold", 20, "bold"))
            label.grid(row=y)
            entry = Entry(frame, highlightthickness=0, relief=RIDGE, bg="white", fg="#999999", font=("yu gothic ui", 20, "bold"), justify="ce
            return entry

        def simpan_stok():
            if entry.get():
                dataset = pd.read_csv ("data_stok.csv")

                #membuat variabel waktu saat ini dengan datetime
                a = dt.datetime.now()
                waktu = a.strftime("%Y %B %d")

                #ambil nilai value dari entry
                entry_makanan = [makanan.get() for makanan in self.value_makanan]
                entry_minuman = [minuman.get() for minuman in self.value_minuman]
                entry_snack = [snack.get() for snack in self.value_snack]

                #memasukkan tanggal waktu date
                for i in range(len(dataset)):
                    dataset.loc[i, 'Data Diperbarui'] = waktu

                #memasukkan data makanan baru
                for i in range(len(dataset)):
                    dataset.loc[i, 'Stok Makanan'] = entry_makanan[i]

                #memasukkan data minuman baru
                for i in range(len(dataset)):
                    dataset.loc[i, 'Stok Minuman'] = entry_minuman[i]

                #memasukkan data snack baru
                for i in range(len(dataset)):
                    dataset.loc[i, 'Stok Snack'] = entry_snack[i]

                dataset.to_csv("data_stok.csv", index=False, header=True)
                messagebox.showinfo("Information!", "Data Telah Disimpan!")

            else:
                self.wajib = Label(self.tempo, background="#ffcccc", text="Semua entry harus di isi!",
                                fg="#993333", font=("yu gothic ui", 12, "bold", "italic"), relief=FLAT, width=40, height=1)
                self.wajib.place(x=520, y=520)

        #tampilan halaman
        self.tempo = Frame(self.ltempo, bg="#ffcccc", width=1600, height=775)
        self.ltempo.place(x=0, y=0)
        self.menubar.destroy()
        self.menu.destroy()
```

Halaman Ded mart (untuk manajer)

```
elif dataset.jabatan[self.id]=='Manajer':
    def cek_barang():
        #membaca csv
        def jump_search(nama_kolom, target):
            dataset = pd.read_csv('data_stok.csv')
            dataset = dataset.sort_values(nama_kolom)

            arr = list(dataset[nama_kolom])
            n = len(arr)
            step = int(n**0.5)
            prev = 0
            result_id = []
            result_value = []

            # Langkah melompati himpunan data
            while prev < n and arr[min(step, n) - 1] < target:
                prev = step
                step += int(n**0.5)

            # Pastikan prev tidak melebihi n-1
            prev = min(prev, n - 1)

            # Melakukan binary search pada subhimpunan yang lebih kecil
            while prev >= 0 and arr[prev] >= target:
                prev -= 1

            # Mencari nilai-nilai yang kurang dari target
            while prev >= 0 and arr[prev] < target:
                result_id.append(prev)
                result_value.append(arr[prev])
                prev -= 1

            return result_id, result_value

        dataset = pd.read_csv('data_stok.csv')
        #tipe pencarian yang dipilih untuk menampilkan makanan yang tersedia
        if self.stok_bar_box.get() == 'Makanan':
            #membuat frame stok agar meminimalisir error
            self.frame_cekbarang = Frame(self.frame_cekbarang, bg='black', width=700, height=235).place(x=635, y=385)

            #membuat list dengan hasil pencarian
            stok_makanan = []
            stok_makanan_minipis = []
            indeks_makanan_minipis = []

            #mencari value di setiap kolom dataset
            for i in dataset['Stok Makanan']:
                stok_makanan.append(i)
                print(stok_makanan)

            #mencari stok yang minipis dari value yang diambil dari dataset
            for i in range(len(stok_makanan)):
                if stok_makanan[i] <= 5:
                    stok_makanan_minipis.append(stok_makanan[i])
                    print(stok_makanan_minipis)

            #Mencari Indeks makanan mana yang minipis untuk ditampilkan
            for i in range(len(stok_makanan)):
                if stok_makanan[i] <= 5:
                    indeks_makanan_minipis.append(i)
                    print(indeks_makanan_minipis)

            #menampilkan data
            data = pd.read_csv('data_stok.csv')
```

Fitur unduh template

```
def template():
    data = {'Data Diperbarui': ['', '', '', '', '', ''], 'Makanan': ['Makanan A', 'Makanan B', 'Makanan C', 'Makanan D', 'Makanan E', 'Makanan F'], 'S
    'Minuman': ['Minuman A', 'Minuman B', 'Minuman C', 'Minuman D', 'Minuman E', 'Minuman F'], 'Stok Minuman': [5, 5, 5, 5, 5], 'Harga Minuman': [20
    'Snack': ['Snack A', 'Snack B', 'Snack C', 'Snack D', 'Snack E', 'Snack F'], 'Stok Snack': [5, 5, 5, 5, 5], 'Harga Snack': [20000, 30000, 40000,
    df = pd.DataFrame(data)
    df.to_csv('data_stok.csv', index = False)

    #message box!
    messagebox.showinfo("Information!!", "Data Telah Di Simpan!!")
```

Fitur entry data stok

```
def entry_barang():
    def entry_updatetok(frame, x, y, text):
        label = Label(frame, text=text, bg="#e2ddce", fg="#181b29", font=("Roboto Bold", 20, "bold"))
        label.place(x=x, y=y)
        entry = Entry(frame, highlightthickness=0, relief=RIDGE, bg="white", fg="#999999", font=("yu gothic ui", 20, "bold"), justify='center')
        entry.place(x=x, y=y+35, width=30, height=30)
        return entry

    def simpan_stok():
        if entry.get():
            dataset = pd.read_csv("data_stok.csv")

            #membuat variabel waktu saat ini dengan datetime
            a = dt.datetime.now()
            waktu = a.strftime("%Y %B %d")

            #Ambil nilai value dari Entry
            entry_makanan = [makanan.get() for makanan in self.value_makanan]
            entry_minuman = [minuman.get() for minuman in self.value_minuman]
            entry_snack = [snack.get() for snack in self.value_snack]
            #Memasukan tanggal update data
            for i in range(len(dataset)):
                dataset.loc[i, 'Data Diperbarui'] = waktu

            #memasukan data makanan Baru
            for i in range(len(dataset)):
                dataset.loc[i, 'Stok Makanan'] = entry_makanan[i]

            #Memasukan data Minuman Baru
            for i in range(len(dataset)):
                dataset.loc[i, 'Stok Minuman'] = entry_minuman[i]

            #Memasukan data Snack Baru
            for i in range(len(dataset)):
                dataset.loc[i, 'Stok Snack'] = entry_snack[i]

            dataset.to_csv("data_stok.csv", index=False, header=True)
            messagebox.showinfo("Information!!", 'Data Telah Disimpan!!')
        else:
            self.wajib = Label(self.timpa, background='#f4eee0', text='*Semua entry harus di isi',
                                fg="#191825", font=("yu gothic ui", 12, 'bold', 'italic'), relief=FLAT, width=49, height=1)
            self.wajib.place(x=520, y=520)

    #Latar belakang
    self.timpa = Frame(self, bg='#f4eee0', width=1460, height=779)
    self.timpa.place(x=0, y=0)
    self.bgrn_image = ImageTk.PhotoImage(Image.open('gambar/bgrnupload.png'))
    self.bgrnlabl = Label(self.timpa, image=self.bgrn_image, background="#ccc6ba").place(x=0, y=0)

    def logoutacc():
        def dest():
            # messagebox.showinfo("Logout Berhasil")
            self.dons.destroy()

        self.dons = customtkinter.CTkFrame(master=self.timpa, fg_color='#ccc6ba', width=200, height=110)
        self.dons.place(x=1250, y=65)
        self.batal=customtkinter.CTkButton(master=self.dons, text='+', text_font=('Arial', 30), corner_radius=8, text_color="#181b29", fg_color='white')
        self.batal.place(x=1300, y=75)
        self.img_logout = ImageTk.PhotoImage(Image.open('gambar/logoutacc.png'))
        self.img_logout.place(x=1250, y=100)
```


Fitur cek barang (implementasi algoritma)

```
def cek_barang():
    #membaca csv
    def jump_search(nama_kolom, target):
        dataset = pd.read_csv('data_stok.csv')
        dataset = dataset.sort_values(nama_kolom)

        arr = list(dataset[nama_kolom])
        n = len(arr)
        step = int(n**0.5)
        prev = 0
        result_idx = []
        result_value = []

        # langkah melompati himpunan data
        while prev < n and arr[min(step, n) - 1] < target:
            prev = step
            step += int(n**0.5)

        # Memeriksa prev tidak melebihi n-1
        prev = min(prev, n - 1)

        # Melakukan binary search pada subhimpunan yang lebih kecil
        while prev >= 0 and arr[prev] >= target:
            prev -= 1

        # Mencari nilai-nilai yang kurang dari target
        while prev >= 0 and arr[prev] < target:
            result_idx.append(prev)
            result_value.append(arr[prev])
            prev -= 1

        return result_idx, result_value

    dataset = pd.read_csv('data_stok.csv')
    #fika coba cek yang dipilih untuk menampilkan makanan yang menipis
    if self.stokbar_box.get() == 'Makanan':
        #membuat frame tittle agar meminimalisir error
        self.frame_cekbarang = Frame(self.frame_cekbarang, bg='#f4eeeb', width=700, height=235).place(x=635, y=385)

        #membuat list dengan hasil pencarian
        stok_makanan = []
        stok_makanan_mempis = []
        indeks_makanan_mempis = []

        #memari value di setiap kolom dataset
        for i in dataset['stok Makanan']:
            stok_makanan.append(i)
            print(stok_makanan)

        #mencari stok yang menipis dari value yang diambil dari dataset
        for i in range(len(stok_makanan)):
            if stok_makanan[i] <= 5:
                stok_makanan_mempis.append(stok_makanan[i])
            print(stok_makanan_mempis)

        # Mencari indeks minuman mana yang menipis untuk ditampilkan
        for i in range(len(stok_makanan)):
            if stok_makanan[i] <= 5:
                indeks_makanan_mempis.append(i)
            print(indeks_makanan_mempis)

        #menampilkan data
        for i in range(len(stok_makanan)):
            if i in indeks_makanan_mempis:
```

```
        #menampilkan data
        for i in range(len(stok_makanan)):
            if i in indeks_makanan_mempis:
                if i == 0:
                    self.label_showdata = Label(self.frame_cekbarang, background='#d9d9c9', text=f'Stok Makanan A tersisa{stok_makanan[i]}', fg='*191825', font=('yu gothic ui', 12, 'bold', 'italic'), relief=FLAT, width=40, height=20, place(x=700, y=385))
                elif i == 1:
                    self.label_showdata = Label(self.frame_cekbarang, background='#d9d9c9', text=f'Stok Makanan B tersisa{stok_makanan[i]}', fg='*191825', font=('yu gothic ui', 12, 'bold', 'italic'), relief=FLAT, width=40, height=20, place(x=700, y=415))
                elif i == 2:
                    self.label_showdata = Label(self.frame_cekbarang, background='#d9d9c9', text=f'Stok Makanan C tersisa{stok_makanan[i]}', fg='*191825', font=('yu gothic ui', 12, 'bold', 'italic'), relief=FLAT, width=40, height=20, place(x=700, y=445))
                elif i == 3:
                    self.label_showdata = Label(self.frame_cekbarang, background='#d9d9c9', text=f'Stok Makanan D tersisa{stok_makanan[i]}', fg='*191825', font=('yu gothic ui', 12, 'bold', 'italic'), relief=FLAT, width=40, height=20, place(x=700, y=475))
                elif i == 4:
                    self.label_showdata = Label(self.frame_cekbarang, background='#d9d9c9', text=f'Stok Makanan E tersisa{stok_makanan[i]}', fg='*191825', font=('yu gothic ui', 12, 'bold', 'italic'), relief=FLAT, width=40, height=20, place(x=700, y=505))
                elif i == 5:
                    self.label_showdata = Label(self.frame_cekbarang, background='#d9d9c9', text=f'Stok Makanan F tersisa{stok_makanan[i]}', fg='*191825', font=('yu gothic ui', 12, 'bold', 'italic'), relief=FLAT, width=40, height=20, place(x=700, y=535))
            else:
                if i == 0:
                    self.label_showdata = Label(self.frame_cekbarang, background='#f4eeeb', text=f'Stok Makanan A Masih Mencukupi!', fg='*191825', font=('yu gothic ui', 12, 'bold', 'italic'), relief=FLAT, width=40, height=20, place(x=700, y=385))
                elif i == 1:
                    self.label_showdata = Label(self.frame_cekbarang, background='#f4eeeb', text=f'Stok Makanan B Masih Mencukupi!', fg='*191825', font=('yu gothic ui', 12, 'bold', 'italic'), relief=FLAT, width=40, height=20, place(x=700, y=415))
                elif i == 2:
                    self.label_showdata = Label(self.frame_cekbarang, background='#f4eeeb', text=f'Stok Makanan C Masih Mencukupi!', fg='*191825', font=('yu gothic ui', 12, 'bold', 'italic'), relief=FLAT, width=40, height=20, place(x=700, y=445))
                elif i == 3:
                    self.label_showdata = Label(self.frame_cekbarang, background='#f4eeeb', text=f'Stok Makanan D Masih Mencukupi!', fg='*191825', font=('yu gothic ui', 12, 'bold', 'italic'), relief=FLAT, width=40, height=20, place(x=700, y=475))
                elif i == 4:
                    self.label_showdata = Label(self.frame_cekbarang, background='#f4eeeb', text=f'Stok Makanan E Masih Mencukupi!', fg='*191825', font=('yu gothic ui', 12, 'bold', 'italic'), relief=FLAT, width=40, height=20, place(x=700, y=505))
                elif i == 5:
                    self.label_showdata = Label(self.frame_cekbarang, background='#f4eeeb', text=f'Stok Makanan F Masih Mencukupi!', fg='*191825', font=('yu gothic ui', 12, 'bold', 'italic'), relief=FLAT, width=40, height=20, place(x=700, y=535))

    if self.stokbar_box.get() == 'Minuman':
        #membuat frame tittle agar meminimalisir error
        self.frame_cekbarang = Frame(self.frame_cekbarang, bg='#f4eeeb', width=700, height=235).place(x=635, y=385)

        #membuat list dengan hasil pencarian
        stok_minuman = []
        stok_minuman_mempis = []
        indeks_minuman_mempis = []

        # Mencari value di setiap kolom dataset
        for i in dataset['stok Minuman']:
```

```
def suratpengajuan():  
    self.window = Frame(self.window, bg="#f0eeeb", width=1468, height=779)  
    self.timea.place(x=35, y=35)  
    self.bgrn_image = ImageTk.PhotoImage(Image.open('gambar/bgrnrt.png'))  
    self.bgrnlblai= Label(self.timea, image=self.bgrn_image, background="#ecccba").place(x=0, y=0)  
    def loginacc():  
        def dest():  
            self.dons.destroy()  
            self.dons = customtkinter.CTkFrame(master=self.timea, fg_color="#ecccba", width=200, height=110)  
            self.dons.place(x=1250, y=65)  
            self.lblai=customtkinter.CTkButton(master=self.dons, text='-',text_font=('Arial', 30), corner_radius=8, text_color="#f8bbd9", font_size=16)  
            self.lgnlacc = ImageTk.PhotoImage(Image.open('gambar/loginacc.png'))  
            self.lgnlogacc=customtkinter.CTkButton(master=self.dons, text='loginot', image=self.lgnlogacc,text_font=('Arial', 20), text_color=  
self.lgnacc = ImageTk.PhotoImage(Image.open('gambar/lcc.png'))  
            self.lgnlogaccda=customtkinter.CTkButton(master=self.timea, text='+', image=self.lgnacc,text_font=('Arial', 20), text_color="#f8bbd  
name=StringVar()  
makanan=StringVar()  
makanan=StringVar()  
makanan=StringVar()  
makanan=StringVar()  
makanan=StringVar()  
makanan=StringVar()  
makanan=StringVar()  
  
minuman=StringVar()  
minuman=StringVar()  
minuman=StringVar()  
minuman=StringVar()  
minuman=StringVar()  
minuman=StringVar()  
minuman=StringVar()  
  
snack=StringVar()  
snack=StringVar()  
snack=StringVar()  
snack=StringVar()  
snack=StringVar()  
snack=StringVar()  
  
def suratl():  
    word=document()  
    today = dt.datetime.now().strftime("%d %M %Y")  
    word.add_picture("gambar/koport.png")  
    # word.add_heading("Surat Pengajuan Stok Barang", 0)  
    word.add_paragraph("*****SURAT*****Surabaya, (today)")  
    word.add_paragraph("Kryada Yeh.")  
    word.add_paragraph("Direktur Pt. Des Company")  
    word.add_paragraph("Gedung Vokal")  
    word.add_paragraph("Surabaya")  
  
word.add_paragraph("(Melalui surat ini saya :")  
  
word.add_paragraph("("+nama[1].nama.get()+")")  
word.add_paragraph("(Jabatan:"+kmdr.get()+")")  
  
word.add_paragraph("("+"Sehubungan dengan batas stok minimal pada DED PMSI saya (nama.get()) selaku manajer perusahaan mengajukan  
tabel_header(["Nama Barang", "Jumlah", "Harga Satuan"])  
some_data=[  
    ["Makanan A", makanan.get(), "25000"],  
    ["Makanan B", makanan.get(), "30000"],  
    ["Makanan C", makanan.get(), "40000"],  
    ["Makanan D", makanan.get(), "50000"],  
    ["Makanan E", makanan.get(), "55000"],  
    ["Makanan F", makanan.get(), "60000"],
```