

CRUD

Point of Sales

Fauzia Arofah, JCDS-0408



INTRODUCTION



Point of sales atau yang lebih familiar dengan sebutan aplikasi kasir merupakan aplikasi yang mengintegrasikan seluruh kegiatan bisnis.

Constraint :

- Inventory (Create, Read, Update, Delete) dan transaksi penjualan.
- User : 'Owner' atau 'Kasir'.
- Transaksi : 2 jenis. (makanan custom dan minuman)

Gambaran Aplikasi

The screenshot displays the Subway mobile app interface. On the left, there are promotional banners for 'Fresh Party Flavours!', 'Party Platters' (starting at 50.00), and 'SUBWAY BREAKFAST' (available until 11:00am). The main section is titled 'How To Order' and includes five steps: 1. Which Sub, Salad or Wrap?, 2. 6-inch or Footlong?, 3. Which Bread & Cheese?, 4. Veggies or Extras?, and 5. Make it a Meal?. To the right of these steps are three menu categories: 'Classics' (Spicy Italian, Steak & Cheese, Chicken Teriyaki, Meatball Marinara, Seafood & Crab, Veggie Delite™), 'Favorites' (Tuna, Turkey Breast & Chicken Slice, Chicken Slice, Roasted Chicken, Roast Beef, Turkey Breast), and 'Premium' (G. Protein, Meat™, Medium B.M.T.™, C. Protein, Club™). Below these is a 'Nutrition Information' table. On the far right, there are sections for 'Breakfast' (Cheese & Egg, Chicken Slice & Egg, Breakfast Sings & Egg, Tuna & Egg) and 'Extras' (Cheese & Egg, Chicken Slice & Egg, Breakfast Sings & Egg, Tuna & Egg). A red box highlights the 'How To Order' steps and the 'Nutrition Information' table. A blue arrow points from the 'Nutrition Information' table to the text on the right.

How To Order

- 1 Which Sub, Salad or Wrap?
- 2 6-inch or Footlong?
- 3 Which Bread & Cheese?
- 4 Veggies or Extras?
- 5 Make it a Meal?

Classics Available in 6-inch & footlong

- Spicy Italian
- Steak & Cheese
- Chicken Teriyaki
- Meatball Marinara
- Seafood & Crab
- Veggie Delite™

Favorites

- Tuna
- Turkey Breast & Chicken Slice
- Chicken Slice
- Roasted Chicken
- Roast Beef
- Turkey Breast

Premium

- G. Protein, Meat™
- Medium B.M.T.™
- C. Protein, Club™

Nutrition Information

ITEM	CALORIES	FAT (g)	SALADS	CALORIES	FAT (g)
6" REGULAR SUBS					
Chicken Slice	284	3.6	Roasted Chicken	143	2.1
Roasted Chicken	280	2.0	Roast Beef	155	4.9
Roast Beef	314	4.9	Turkey Breast	117	1.9
Turkey Breast	240	4.8	Turkey Breast & Chicken Slice	131	2.9
Subway Club™	276	2.0	Subway Club™	181	4.7
Turkey Breast & Chicken Slice	290	2.9	Chicken Teriyaki	182	2.1
Chicken Teriyaki	140	2.4	Veggie Delite™	54	1.1
Veggie Delite™	213	1.2			

Breakfast

- Cheese & Egg
- Chicken Slice & Egg
- Breakfast Sings & Egg
- Tuna & Egg

Extras

- Cheese & Egg
- Chicken Slice & Egg
- Breakfast Sings & Egg
- Tuna & Egg

- Terinspirasi dari menu-menu di Subway.
- Aplikasi ini memuat bahan baku Bread, Protein, Cheese, Sauce, dan Veggie yang dapat menjadi menu custom makanan.
- Aplikasi ini juga memiliki inventory untuk minuman
- Setiap item yang ada memiliki informasi nutrisi

Daftar Inventory:

Berisi nama item pada dictionary beserta info Stock, Fat, Protein, Calories, dan harga.

Dictionary bahan baku makanan: Bread, Protein, Cheese, Sauce, Veggie

Dictionary minuman: Drink

Dictionary tambahan: inventory_all, yang merupakan gabungan dari seluruh inventory



Contoh pada dict bread:
{ "Italian HerbsCheese":
{"stock": 1, "fat": 12.09,
"protein":9.53, "calories": 271,
"harga": 15000}} dll.....



Contoh pada dict protein:
{ "Smoked Beef Salami": {"stock":
100, "fat": 5.11, "protein":2.9,
"calories": 59, "harga": 12000
}} dll.....



Contoh pada dict cheese:
{ "Cream Cheese": {"stock": 100,
"fat":5.06, "protein":1.09,
"calories": 51, "harga": 3000 }
} dll.....



Contoh pada dict sauce:
{ "Mayo": {"stock": 100,
"fat":4.91, "protein":0.13,
"calories": 57, "harga": 2000
}} dll.....



Contoh pada dict veggie:
{ "Lettuce": {"stock": 100,
"fat":0.08, "protein":0.5,
"calories": 8, "harga": 2000
}} dll.....



Contoh pada dict bread:
{ "Mineral Water": {"stock": 100,
"fat":0, "protein":0,
"calories": 0, "harga": 7000 }
} dll.....



```
Inventory_all={}  
inventory_all.update(bread)  
inventory_all.update(protein)  
inventory_all.update(cheese)  
inventory_all.update(sauce)  
inventory_all.update(veggie)  
inventory_all.update(drink)
```



Daftar Function

- **def posisi_jabatan()** -> Fungsi ini meminta user untuk memasukkan jabatan user, yang akan disimpan pada variabel posisi.
- **def main_menu(posisi)** -> Fungsi ini akan menampilkan menu utama yang dapat diakses berdasarkan jabatan (argumen:'posisi').
- **def lanjut(menu)** -> Fungsi ini akan mengkonfirmasi user untuk akses ke menu yg sedang diakses/ ke main_menu/ keluar dari program, sesuai kebutuhan user
- **def display_inventory (pilih_dict,teks)** -> Fungsi ini akan menampilkan tabel inventory berdasarkan dictionary yg dipilih dan nama dictionary tersebut.
- **def exit_program()->** Fungsi ini akan mengakhiri program aplikasi Point of Sales
- **def read_program()** -> Fungsi ini akan menampilkan menu terkait menampilkan inventory bahan baku di gudang.
- **def create_program()** -> Fungsi ini digunakan untuk menambah item di dalam inventory sesuai inputan user.
- **def update_program()** -> Fungsi ini digunakan untuk meng-update informasi stock atau harga pada inventory yang dipilih user.
- **def delete_program()** -> Fungsi ini digunakan untuk menghapus item satu persatu ataupun secara keseluruhan dalam dictionary yang dipilih user.
- **def transaksi()** -> Fungsi ini digunakan untuk proses transaksi penjualan item dari inventory, yang dibagi menjadi jenis makanan atau minuman.
- **def transaksi_makanminum (makan_minum):** -> Fungsi ini akan melakukan proses pemilihan dan penambahan item yang dipilih oleh user ke dalam keranjang belanja dalam transaksi penjualan.

```

=====
Selamat Datang di Aplikasi Point of Sales
=====

Silahkan masukkan profesi Anda: owner
=====

Menu Utama Owner:
1. Menampilkan Inventory Bahan Baku
2. Menambah Inventory Bahan Baku
3. Menghapus Inventory Bahan Baku
4. Update Inventory Bahan Baku
5. Transaksi
6. Log in dengan posisi lain
7. Keluar Program
=====

Masukkan Menu Pilihan Anda (angka): 1
=====

Menampilkan Inventory Gudang:
1. Menampilkan Inventory Bread
2. Menampilkan Inventory Protein
3. Menampilkan Inventory Cheese
4. Menampilkan Inventory Sauce
5. Menampilkan Inventory Veggie
6. Menampilkan Inventory Drink
7. Menampilkan Seluruh Inventory
8. Kembali ke Main Menu
=====

Masukkan Menu Pilihan Anda (angka): 1
=====

DAFTAR INVENTORY ROTI
Index  ||Cal  ||Fat  ||Prot ||Stock ||Harga ||Roti
=====
1      ||271  ||12.09 ||9.53 ||1      ||15000 ||Italian HerbsCheese
2      ||271  ||3.5   ||8.8  ||1      ||15000 ||Italian Bread
3      ||184  ||0.91  ||6.73 ||100    ||17000 ||Whole Wheat Tortilla
4      ||266  ||3.29  ||7.64 ||100    ||10000 ||Flat White Bread
=====

Apakah Anda mau mengakses menu menampilkan inventory kembali (Y/N)? n
=====

```

Contoh running program (1):

- Program akan memanggil fungsi **posisi_jabatan()**. Contoh disamping, user menginputkan **posisi:'Owner'**.
- Fungsi **posisi_jabatan** akan memanggil **main_menu(posisi)**, sehingga tampil menu menu yang bisa diakses oleh 'Owner'.
- User memilih untuk menampilkan inventory bahan baku, maka user menginputkan angka 1 dan program akan memanggil fungsi **read_program()**
- Program menampilkan opsi menu **read_program()**. User ingin melihat inventory bread, maka user menginputkan angka 1 dan program memanggil fungsi **display_inventory(pilih_dict, teks)**. Dimana **pilih_dict = bread**, dan **teksnya = 'roti'**

```
if opsi_menu_read>=1 and opsi_menu_read<=7:  Jika inputan user antara 1-7
```

```
opsi_opsi={ #dict yg berisi tuple (nama_dict, dan nama_dict dlm string)
```

```

1: (bread, 'Roti'),
2: (protein, 'Protein'),
3: (cheese, 'Keju'),
4: (sauce, 'Saus'),
5: (veggie, 'Sayur'),
6: (drink, 'Minuman'),
7: (inventory_all, "Semua Bahan Baku")

```

Opsi_opsi= merupakan dictionary berisi tuple.

Jika User menginput 1, maka :
pilih_dict = opsi_opsi[1][0]= bread (akan memanggil dictionary bread)
teks= opsi_opsi[1][1]= 'Roti'

```
pilih_dict=opsi_opsi[opsi_menu_read][0] #mau mengambil nama dict dalam tuple
```

```
teks=opsi_opsi[opsi_menu_read][1] #untuk mengambil string dict dalam tuple
```

```
display_inventory (pilih_dict,teks)
```

display_inventory(pilih_dict, teks)

menjadi **display_inventory (bread, 'roti')**, dan menghasilkan output table seperti contoh.

Contoh running program (l)(continue...):

- o Setelah program menampilkan daftar inventory, program akan mengkonfirmasi user kelanjutan program dengan memanggil fungsi **lanjut('menampilkan inventory')**.

```
DAFTAR INVENTORY ROTI
Index  ||Cal  ||Fat  ||Prot  ||Stock ||Harga ||Roti
=====
1      ||271  ||12.09 ||9.53  ||1      ||15000 ||Italian HerbsCheese
2      ||271  ||3.5   ||8.8   ||1      ||15000 ||Italian Bread
3      ||184  ||0.91  ||6.73  ||100    ||17000 ||Whole Wheat Tortilla
4      ||266  ||3.29  ||7.64  ||100    ||10000 ||Flat White Bread
=====

Apakah Anda mau mengakses menu menampilkan inventory kembali (Y/N)? n
=====
Apakah Anda mau kembali ke main menu (Y/N)? n
=====

Anda telah keluar dari aplikasi Point of Sales ini. Terimakasih dan Sampai Jumpa
=====
PS D:\fauziah\Purwadhika>
```

```
def lanjut(menu):
    global posisi

    # menu = nama menu yg sedang berjalan, sehingga bisa dipanggil sebagai variabel dinamis
    lanjut_menu_ini= str(input(f"Apakah Anda mau mengakses menu {menu} kembali (Y/N)? ").lower())
    print("=====")
    if lanjut_menu_ini!='y':
        lanjut_main_menu=str(input('Apakah Anda mau kembali ke main menu (Y/N)? ').lower())
        print("=====")
        if lanjut_main_menu!='y':
            exit_program()
        elif lanjut_main_menu=='y':
            main_menu(posisi)
```

Fungsi **lanjut(menu)**. Pada contoh disamping, menu diisi dengan string 'menampilkan inventory'

Konfirmasi untuk looping di fungsi **read_program()** atau tidak?

Jika user tidak mengetik 'Y', maka akan dikonfirmasi lagi, apakah mau ke main menu?

Jika user mengetik 'Y', maka program akan memanggil kembali fungsi **main_menu(posisi)**, dalam contoh disamping posisi masih owner, jd terpanggil **main_menu(owner)**

Jika user tidak mengetik 'Y', maka program akan memanggil fungsi **exit_program()**

```
def exit_program():
    print("=====")
    print("Anda telah keluar dari aplikasi Point of Sales ini. Terimakasih dan Sampai Jumpa")
    print("=====")
    quit()
```

Fungsi **exit_program()**. Berisi pesan bahwa user telah keluar dari program. Lalu akan mengakhiri program berjalan

```
def display_inventory (pilih_dict,teks):
```

Fungsi **display_inventory(pilih_dict,teks).**

```
    print()
```

```
    print(f"DAFTAR INVENTORY {teks.upper()}")
```

```
    print(f'Index\t||Cal\t||Fat\t||Prot\t||Stock\t||Harga\t||{teks.title()}')
```

```
    print("=====
```

Slicing dictionary yang terpilih. Untuk mendapatkan index iterasi, jenis (yg berupa key), dan nutri_inv (berisi info2 dalam dictionary terpilih

```
    for index_, (jenis, nutri_inv) in enumerate(pilih_dict.items()):
        print(f'{index_+1}\t||{nutri_inv["calories"]}\t||{nutri_inv["fat"]}\t||{nutri_inv["protein"]}\t||{nutri_inv["stock"]}\t||{nutri_inv["harga"]}\t||{jenis} ')
    print("=====")
```

```
    print()
```

```
    print()
```


Contoh running program (2):

- Program akan memanggil fungsi **posisi_jabatan()**. Contoh disamping, user menginputkan **posisi:'kasir'**.
- Fungsi **posisi_jabatan()** akan memanggil **main_menu(kasir)**, sehingga tampil menu menu yang bisa diakses oleh 'kasir'. Menu pada **main_menu(kasir)** dan **main_menu(owner)** berbeda.
- User memilih untuk menginput ulang posisi lain, maka user menginput angka 2 dan program memanggil fungsi **posisi_jabatan()**, sehingga user diminta untuk menginputkan kembali posisi jabatan.

```
=====
Selamat Datang di Aplikasi Point of Sales
=====

Silahkan masukkan profesi Anda: kasir
=====

Menu Utama Kasir:
1. Transaksi
2. Log in dengan posisi lain
3. Keluar Program
=====

Masukkan Menu Pilihan Anda (angka): 2
=====

Selamat Datang di Aplikasi Point of Sales
=====

Silahkan masukkan profesi Anda: [ ]
```

Fungsi **posisi_jabatan()**. Pada contoh disamping, menu diisi dengan string 'menampilkan inventory

```
def posisi_jabatan():
```

```
    global posisi
```

```
    print()
```

```
    print("====="
```

```
Selamat Datang di Aplikasi Point of Sales
```

```
=====
```

```
while True:
```

```
    print()
```

```
    # constraint:: insiasi posisi HANYA untuk 'owner' dan 'kasir'
```

```
    posisi= input('Silahkan masukkan profesi Anda: ').lower()
```

```
    print("=====
```

User diminta untuk menginputkan posisi user.

```
    if posisi=='owner' or posisi=='kasir':
```

```
        main_menu(posisi)
```

Jika inputan user antara 'owner' atau 'kasir', maka program akan memanggil fungsi **main_menu(posisi)**, dengan posisi sesuai inputan user.

```
    else :
```

```
        print('Posisi yang Anda Masukkan Salah!')
```

```
        print("=====
```

```
        lanjut=input('Apakah Anda mau login kembali(Y/N)? ').lower()
```

```
        print("=====
```

```
        if lanjut!='y':
```

```
            exit_program()
```

Jika inputan user bukan 'owner' atau 'kasir', maka program akan memanggil mengkonfirmasi user untuk login Kembali atau tidak?

```
        else:
```

```
            posisi_jabatan()
```

Jika user tidak mengetik Y, maka program akan memanggil fungsi **exit_program()**.

Selain itu, program akan memanggil fungsi **posisi_jabatan()** dan meminta inputan posisi Kembali dari user.

Def `create_program()`:

Untuk membuat item baru, dengan catatan nama item= primary key. Jadi tidak boleh sama dengan item yang sudah ada.

```
def create_program():
    global posisi
    while True:
        print('''=====
Menambah Inventory Gudang:

1. Menambah Inventory Bread
2. Menambah Inventory Protein
3. Menambah Inventory Cheese
4. Menambah Inventory Sauce
5. Menambah Inventory Veggie
6. Menambah Inventory Drink
7. Kembali ke Main Menu

=====''')

        opsi_menu_create=(input("Masukkan Menu Pilihan Anda (angka): "))
        if opsi_menu_create.isdigit():
            Validari inputan user harus berupa angka sesuai menu
            opsi_menu_create=int(opsi_menu_create)
            print("=====")
            if opsi_menu_create>=1 and opsi_menu_create<=6:
                opsi_opsi={ #dict yg berisi tuple (nama_dict, dan nama_dict dlm string)
                    1: (bread, 'Roti'),
                    2: (protein, 'Protein'),
                    3: (cheese, 'Keju'),
                    4: (sauce, 'Saus'),
                    5: (veggie, 'Sayur'),
                    6: (drink, 'Minuman')}

                pilih_dict=opsi_opsi[opsi_menu_create][0] #mau mengambil dict dalam tuple sesuai inputan user
                teks=opsi_opsi[opsi_menu_create][1] #untuk mengambil string dict dalam tuple

                display_inventory (pilih_dict,teks)
```

def create_program():

```
    print(f"Masukkan nama {teks.lower()} yang ingin ditambahkan.")
    print("=====")
    tambah_item = str(input("Nama bahan baku tidak boleh sama dengan yang sudah ada: ")).title()
    print("=====")

    if tambah_item in pilih_dict:
        print(f'{tambah_item} sudah ada didalam inventory.')
        print("=====")

    else:
        try:
            tambah_stock=int(input("Masukkan stock item: "))
            tambah_harga= int(input('Masukkan harga item: '))
            tambah_fat=float(input ('Masukan info nutrition
            tambah_prot=float(input ('Masukan info nutrition
            tambah_kal=float(input ('Masukan info nutrition (kalori) item: '))
            print("=====")

            if int(tambah_stock)>=0 and int(tambah_harga)>=0 and float(tambah_fat)>=0 and float(tambah_prot)>=0 and float(tambah_kal)>=0:
                pilih_dict[tambah_item.title()]= {
                    "stock": (tambah_stock),
                    "harga": (tambah_harga),
                    "fat": tambah_fat,
                    "protein": tambah_prot,
                    "calories": (tambah_kal)}

                display_inventory (pilih_dict,teks)

            else:
                print(f'{tambah_item.title()} gagal ditambahkan')
                print('Info stock dan harga harus berupa bilangan bulat lebih dari 0')
                print('Kandungan nilai fat, protein, dan kalori harus lebih dari nol')
                print("=====")

        except:
            print(f'{tambah_item.title()} gagal ditambahkan')
            print('Info stock dan harga harus berupa angka bilangan bulat lebih dari 0')
            print('Kandungan nilai fat, protein, dan kalori harus berupa angka lebih dari nol')
            print("=====")

    elif opsi_menu_create==7:
        main_menu(posisi)
```

Pengecekan inputan user (var:tambah_item), apa bila sudah ada dalam invntory, program akan menampilkan pesan.

Jika inputan user valid, user bisa menginputkan value tambahan lainnya untuk di tambahkan pada inventory. Dengan tipe data yang sudah ditentukan.

Jika inputan value tambahan user valid, maka data baru akan terbentuk,

Jika inputan value tidak sesuai, maka item baru tidak akan di tambahkan ke inventory

Penanganan error inputan value tambahan dari user.

Def create_program():
(..continue.. running
program)

```
DAFTAR INVENTORY ROTI
Index ||Cal ||Fat ||Prot ||Stock ||Harga ||Roti
=====
1 ||271 ||12.09 ||9.53 ||1 ||15000 ||Italian HerbsCheese
2 ||271 ||3.5 ||8.8 ||1 ||15000 ||Italian Bread
3 ||184 ||0.91 ||6.73 ||100 ||17000 ||Whole Wheat Tortilla
4 ||266 ||3.29 ||7.64 ||100 ||10000 ||Flat White Bread
=====
```

Salah satu namaitem yang sudah ada
dalam inventory: 'Italian bread'

Masukkan nama roti yang ingin ditambahkan.

Nama bahan baku tidak boleh sama dengan yang sudah ada: italian bread

Italian Bread sudah ada didalam inventory.

Case 1: Ketika user menginputkan nama
item baru = 'Italian bread', maka akan ada
pesan bahwa Italian bread sudah ada
dalam inventory, dan program tidak
menambahkan item baru.

Apakah Anda mau mengakses menu menambah inventory kembali (Y/N)? y

Menambah Inventory Gudang:

1. Menambah Inventory Bread
2. Menambah Inventory Protein
3. Menambah Inventory Cheese
4. Menambah Inventory Sauce
5. Menambah Inventory Veggie
6. Menambah Inventory Drink
7. Kembali ke Main Menu

Masukkan Menu Pilihan Anda (angka): 1

```
DAFTAR INVENTORY ROTI
Index ||Cal ||Fat ||Prot ||Stock ||Harga ||Roti
=====
1 ||271 ||12.09 ||9.53 ||1 ||15000 ||Italian HerbsCheese
2 ||271 ||3.5 ||8.8 ||1 ||15000 ||Italian Bread
3 ||184 ||0.91 ||6.73 ||100 ||17000 ||Whole Wheat Tortilla
4 ||266 ||3.29 ||7.64 ||100 ||10000 ||Flat White Bread
=====
```

Masukkan nama roti yang ingin ditambahkan.

Nama bahan baku tidak boleh sama dengan yang sudah ada: roti bayam

Case 2: Ketika user menginputkan nama
item baru = 'roti bayam', Nama item roti
bayam belum ada dalam inventory,
sehingga program meminta user
menginput info tambahan lainnya.

Masukkan stock item: 99

Masukkan harga item: 15000

Masukan info nutrition (fat) item: 0.2

Masukan info nutrition (protein) item: 2

Masukan info nutrition (kalori) item: 20

```
DAFTAR INVENTORY ROTI
Index ||Cal ||Fat ||Prot ||Stock ||Harga ||Roti
=====
1 ||271 ||12.09 ||9.53 ||1 ||15000 ||Italian HerbsCheese
2 ||271 ||3.5 ||8.8 ||1 ||15000 ||Italian Bread
3 ||184 ||0.91 ||6.73 ||100 ||17000 ||Whole Wheat Tortilla
4 ||266 ||3.29 ||7.64 ||100 ||10000 ||Flat White Bread
5 ||20.0 ||0.2 ||2.0 ||99 ||15000 ||Roti Bayam
=====
```

Jika semua inputan user valid, item baru
akan masuk ke dalam inventory.

Def update_program():

```
def update_program ():
    global posisi
    while True:
        print("=====")
        print('Mengupdate Inventory Gudang:
1. Mengupdate Inventory Bread
2. Mengupdate Inventory Protein
3. Mengupdate Inventory Cheese
4. Mengupdate Inventory Sauce
5. Mengupdate Inventory Veggie
6. Mengupdate Inventory Drink
7. Kembali ke main menu''')

        print("=====")
        opsi_menu_update=input("Masukkan Menu Pilihan Anda (angka): ")
        if opsi_menu_update.isdigit():
            opsi_menu_update=int(opsi_menu_update)
            print("=====")
            if opsi_menu_update>=1 and opsi_menu_update<=6:
                opsi_opsi={ #dict yg berisi tuple (nama_dict, dan nama_dict dlm string)
                    1: (bread, 'Bread'),
                    2: (protein, 'Protein'),
                    3: (cheese, 'Keju'),
                    4: (sauce, 'Saus'),
                    5: (veggie, 'Sayur'),
                    6: (drink, 'Minuman')}

                pilih_dict=opsi_opsi[opsi_menu_update][0] #mau mengambil nama dict dalam tuple
                teks=opsi_opsi[opsi_menu_update][1] #untuk mengambil string dict dalam tuple

                display_inventory (pilih_dict,teks)

                print('Pilihan Update
1. Update Stock
2. Update Harga''')

                print("=====")
                try:
                    opsi_update=int(input("Masukkan menu update Anda (angka): "))
                    print("=====")

                    if opsi_update==1 or opsi_update==2:
                        if opsi_update==1:
                            update_valuesnya="stock"
                        elif opsi_update==2:
                            update_valuesnya="harga"

                        if opsi_update==1 and index_update<len(pilih_dict):
                            update_key = (list(pilih_dict.keys()))[index_update-1]
                            info_update = int(input(f"Masukkan {update_valuesnya} {update_key} yang Akan Anda Update: "))
                            if info_update>=0:
                                pilih_dict[update_key][update_valuesnya]=info_update
                                print(f'{update_valuesnya.capitalize()} untuk {update_key} berhasil di update menjadi {info_update}')
                                print("=====")
                                display_inventory (pilih_dict,teks)
                            else:
                                print('Gagal mengupdate data!')
                                print(f'{update_valuesnya.capitalize()} update tidak boleh kurang dari nol')
                                print("=====")
                        else:
                            print("Angka yang Anda Masukkan Tidak Valid!")
                            print("=====")

                    except:
                        print("Inputan Anda Tidak Valid!")
                        print("Inputan harus berupa bilangan bulat positif!")
                        print("=====")

                    elif opsi_update!=1 or opsi_update!=2:
                        print("Angka yang Anda Masukkan Tidak Valid!")
                        print("=====")

                    except:
                        print("Inputan Anda Tidak Valid!")
                        print("Inputan harus berupa bilangan bulat positif!")
                        print("=====")
```

Inputan pertama memilih inventory mana yang mau di update

Inputan kedua memilih update info apa stock/ harga?

Inputan ketiga memilih item mana dari inventory yg dipilih, yg mau di update

Inputan keempat menginput info stok/ harga yang mau di update

Jika semua inputan valid, maka info item akan terupdate.

Jika ada salah satu inputan tidak valid, maka akan gagal, dan tidak ada yang di update.

Mengupdate Inventory Gudang:
1. Mengupdate Inventory Bread
2. Mengupdate Inventory Protein
3. Mengupdate Inventory Cheese
4. Mengupdate Inventory Sauce
5. Mengupdate Inventory Veggie
6. Mengupdate Inventory Drink
7. Kembali ke main menu

Def `update_program()`:
(..continue.. running program))

Masukkan Menu Pilihan Anda (angka): 1

DAFTAR INVENTORY BREAD

Inventory awal bread. Stock Italian herbscheese=1

Index	Cal	Fat	Prot	Stock	Harga	Bread
1	271	12.09	9.53	1	15000	Italian HerbsCheese
2	271	3.5	8.8	1	15000	Italian Bread
3	184	0.91	6.73	100	17000	Whole Wheat Tortilla
4	266	3.29	7.64	100	10000	Flat White Bread

Pilihan Update

1. Update Stock
2. Update Harga

Update Stock

Masukkan menu update Anda (angka): 1

Masukkan index inventory yang Akan Anda Update: 1

Masukkan stock Italian HerbsCheese yang Akan Anda Update: 99

Stock untuk Italian HerbsCheese berhasil di update menjadi 99

Inventory bread setelah di update Stock Italian
herbscheese=99

DAFTAR INVENTORY BREAD

Index	Cal	Fat	Prot	Stock	Harga	Bread
1	271	12.09	9.53	99	15000	Italian HerbsCheese
2	271	3.5	8.8	1	15000	Italian Bread
3	184	0.91	6.73	100	17000	Whole Wheat Tortilla
4	266	3.29	7.64	100	10000	Flat White Bread

Apakah Anda mau mengakses menu mengupdate inventory kembali (Y/N)?

Mengupdate Inventory Gudang:
1. Mengupdate Inventory Bread
2. Mengupdate Inventory Protein
3. Mengupdate Inventory Cheese
4. Mengupdate Inventory Sauce
5. Mengupdate Inventory Veggie
6. Mengupdate Inventory Drink
7. Kembali ke main menu

Masukkan Menu Pilihan Anda (angka): 1

DAFTAR INVENTORY BREAD

Inventory awal bread. Harga Italian herbscheese =15000

Index	Cal	Fat	Prot	Stock	Harga	Bread
1	271	12.09	9.53	99	15000	Italian HerbsCheese
2	271	3.5	8.8	1	15000	Italian Bread
3	184	0.91	6.73	100	17000	Whole Wheat Tortilla
4	266	3.29	7.64	100	10000	Flat White Bread

Pilihan Update

1. Update Stock
2. Update Harga

Update Harga

Masukkan menu update Anda (angka): 2

Masukkan index inventory yang Akan Anda Update: 1

Masukkan harga Italian HerbsCheese yang Akan Anda Update: 25000

Harga untuk Italian HerbsCheese berhasil di update menjadi 25000

Inventory bread setelah di update harga Italian
herbscheese=25000

DAFTAR INVENTORY BREAD

Index	Cal	Fat	Prot	Stock	Harga	Bread
1	271	12.09	9.53	99	25000	Italian HerbsCheese
2	271	3.5	8.8	1	15000	Italian Bread
3	184	0.91	6.73	100	17000	Whole Wheat Tortilla
4	266	3.29	7.64	100	10000	Flat White Bread

Apakah Anda mau mengakses menu mengupdate inventory kembali (Y/N)?

```
def delete_program():
    while True:
        print("=====")
        print("'Menghapus Inventory Gudang:")
        1. Menghapus Inventory Bread
        2. Menghapus Inventory Protein
        3. Menghapus Inventory Cheese
        4. Menghapus Inventory Sauce
        5. Menghapus Inventory Veggie
        6. Menghapus Inventory Drink
        7. Menghapus Seluruh Inventory
        8. Kembali ke main menu'''

        opsi_menu_delete=input("Masukkan Menu Pilihan Anda (angka): ")
        print("=====")
        if opsi_menu_delete.isdigit():
            opsi_menu_delete=int(opsi_menu_delete)
            if opsi_menu_delete>1 and opsi_menu_delete<=6:
                opsi_opsi={ #dict yg berisi tuple (nama_dict, dan nama_dict dlm string)
                    1: (bread, 'Roti'),
                    2: (protein, 'Protein'),
                    3: (cheese, 'Keju'),
                    4: (sauce, 'Saus'),
                    5: (veggie, 'Sayur'),
                    6: (drink, 'Minuman')}

                pilih_dict=opsi_opsi[opsi_menu_delete][0] #mau mengambil nama dict dalam tuple
                teks=opsi_opsi[opsi_menu_delete][1] #untuk mengambil string dict dalam tuple

                display_inventory (pilih_dict,teks)

                key_pilih_dict= list(pilih_dict.keys()) #memanggil list dari nama2 bahan baku

                try:
                    index_hapus=int(input("Masukkan index inventory yang Akan Anda Hapus: "))
                    print("=====")

                    if index_hapus>=0 and index_hapus<len(key_pilih_dict):
                        opsi_delete = str((input(f'Anda yakin akan menghapus {key_pilih_dict[index_hapus]}? (Y/N)?')).lower())
                        if opsi_delete== "y":
                            del pilih_dict[key_pilih_dict[index_hapus]]
                            print(f'Inventory {key_pilih_dict[index_hapus]} telah di hapus')
                            print("=====")
                        else:
                            print("Angka (index) yang Anda Masukkan Tidak Valid!")
                            print("=====")
                    except:
                        print("Inputan Anda Masukkan Tidak Valid!")
                        print("=====")
```

Def **delete_program()**:

```
elif opsi_menu_delete==7:
    display_inventory (inventory_all,"Bahan Baku")
    print("=====")
    opsi_delete = str((input(f'Anda yakin akan menghapus (Y/N)?')).lower())
    print("=====")
    if opsi_delete!= "y":
        print(f'Batal Menghapus Seluruh Inventory')
        print("=====")

else:
    bread.clear()
    protein.clear()
    cheese.clear()
    sauce.clear()
    veggie.clear()
    drink.clear()
    inventory_all.clear()
    print(f'Seluruh Inventory Telah di Hapus')
    print("=====")
```

Def **delete_program()**, menu 7.
menghapus seluruh inventory

- Menghapus Inventory Gudang:
1. Menghapus Inventory Bread
 2. Menghapus Inventory Protein
 3. Menghapus Inventory Cheese
 4. Menghapus Inventory Sauce
 5. Menghapus Inventory Veggie
 6. Menghapus Inventory Drink
 7. Menghapus Seluruh Inventory
 8. Kembali ke main menu

Masukkan Menu Pilihan Anda (angka): 1

DAFTAR INVENTORY ROTI

Index	Cal	Fat	Prot	Stock	Harga	Roti
-------	-----	-----	------	-------	-------	------

1	271	12.09	9.53	99	25000	Italian HerbsCheese
2	271	3.5	8.8	1	15000	Italian Bread
3	184	0.91	6.73	100	17000	Whole Wheat Tortilla
4	266	3.29	7.64	100	10000	Flat White Bread

Masukkan index inventory yang Akan Anda Hapus: 4

Anda yakin akan menghapus Flat White Bread? (Y/N)?y

Inventory Flat White Bread telah di hapus

Apakah Anda mau mengakses menu menghapus inventory kembali (Y/N)? y

Menghapus Inventory Gudang:

1. Menghapus Inventory Bread
2. Menghapus Inventory Protein
3. Menghapus Inventory Cheese
4. Menghapus Inventory Sauce
5. Menghapus Inventory Veggie
6. Menghapus Inventory Drink
7. Menghapus Seluruh Inventory
8. Kembali ke main menu

Masukkan Menu Pilihan Anda (angka): 1

DAFTAR INVENTORY ROTI

Index	Cal	Fat	Prot	Stock	Harga	Roti
-------	-----	-----	------	-------	-------	------

1	271	12.09	9.53	99	25000	Italian HerbsCheese
2	271	3.5	8.8	1	15000	Italian Bread
3	184	0.91	6.73	100	17000	Whole Wheat Tortilla

Masukkan index inventory yang Akan Anda Hapus:

Def delete_program():
(..continue.. running program))

- Menghapus Inventory Gudang:
1. Menghapus Inventory Bread
 2. Menghapus Inventory Protein
 3. Menghapus Inventory Cheese
 4. Menghapus Inventory Sauce
 5. Menghapus Inventory Veggie
 6. Menghapus Inventory Drink
 7. Menghapus Seluruh Inventory
 8. Kembali ke main menu

Masukkan Menu Pilihan Anda (angka): 7

DAFTAR INVENTORY BAHAN BAKU

Index	Cal	Fat	Prot	Stock	Harga	Bahan Baku
-------	-----	-----	------	-------	-------	------------

1	271	12.09	9.53	99	25000	Italian HerbsCheese
2	271	3.5	8.8	1	15000	Italian Bread
3	184	0.91	6.73	100	17000	Whole Wheat Tortilla
4	266	3.29	7.64	100	10000	Flat White Bread
5	59	5.11	2.9	100	12000	Smoked Beef Salami
6	130	11.28	5.7	100	10000	Pepperoni
7	105	5.5	6.6	100	15000	Chicken Strips
8	291	22.23	21.11	100	20000	Beef Brisket
9	51	5.06	1.09	100	3000	Cream Cheese
10	85	5.61	7.27	100	7000	Mozarella
11	113	9.28	6.97	100	5000	Cheddar Cheese
12	57	4.91	0.13	100	2000	Mayo
13	58	5.47	0.17	100	3000	Thousand Island
14	44	4.26	0.06	100	3000	Italian Dressing
15	40	4.5	0	100	4000	Sesame oil
16	100	9	1	100	4000	Roasted Sesame
17	8	0.08	0.5	100	2000	Lettuce
18	16	0.06	0.27	100	1000	Tomatoes
19	16	0.08	0.3	100	2000	Cucumber
20	48	0.09	1.06	100	3000	Onion
21	42	4.44	0.3	100	5000	Green Olives
22	0	0	0	100	7000	Mineral Water
23	0	0	0	100	10000	Diet Coke

Anda yakin akan menghapus (Y/N)?y

Seluruh Inventory Telah di Hapus

Seluruh Inventory Telah di Hapus

Apakah Anda mau mengakses menu menghapus inventory kembali (Y/N)? y

Menghapus Inventory Gudang:

1. Menghapus Inventory Bread
2. Menghapus Inventory Protein
3. Menghapus Inventory Cheese
4. Menghapus Inventory Sauce
5. Menghapus Inventory Veggie
6. Menghapus Inventory Drink
7. Menghapus Seluruh Inventory
8. Kembali ke main menu

Masukkan Menu Pilihan Anda (angka): 7

DAFTAR INVENTORY BAHAN BAKU

Index	Cal	Fat	Prot	Stock	Harga	Bahan Baku
-------	-----	-----	------	-------	-------	------------

Def transaksi():

```
def transaksi():
    global keranjang
    global no_urut
    global total_keseluruhan
    global opsi_opsi

    if inventory_all!={}:
        while True:
            keranjang = []
            total_keseluruhan = 0
            print("=====")
            print("=====TRANSAKSI PENJUALAN=====")
            print("=====")

            opsi_opsi= { #dict yg berisi tuple (nama_dict, dan nama_dict dlm string)
                1: (bread, 'Roti'),
                2: (protein, 'Protein'),
                3: (cheese, 'Keju'),
                4: (sauce, 'Saus'),
                5: (veggie, 'Sayur')}

            transaksi_makanminum("makanan")

            pesan_minum=input('Apakah Anda mau memesan minum (Y/N? ')
            print("=====")

            if pesan_minum == 'y':
                opsi_opsi= { #dict yg berisi tuple (nama_dict, dan nama_dict dlm string)
                    1: (drink, 'Minuman')}

                transaksi_makanminum("minuman")
```

```
print("KERANJANG BELANJA ANDA")
print("=====")
no_urut = 1 #assign untuk urutan di rakapan belanja
for dict_dalam_keranjang in keranjang:
    for key_keranjang, value_keranjang in dict_dalam_keranjang.items():
        print(no_urut, ' ', key_keranjang)
        print(' ', value_keranjang)
        print()
        no_urut+=1

print("=====")
print('TOTAL BELANJA      : ', total_keseluruhan)
while True:
    uang_pembayaran = int(input('PEMBAYARAN TUNAI      : '))
    print("=====")
    if uang_pembayaran>total_keseluruhan and uang_pembayaran>0:
        uang_kembali= uang_pembayaran-total_keseluruhan
        print('KEMBALI      : ', uang_kembali)
        print("=====")
        print('PEMBAYARAN BERHASIL, TERIMAKASIH TELAH BERBELANJA')
        print('-----KEDATANGAN ANDA SELALU KAMI TUNGGU : )-----')
        break
    elif uang_pembayaran<total_keseluruhan or uang_pembayaran<0:
        print('UANG ANDA KURANG.')
        print('PASTIKAN UANG PEMBAYARAN MINIMAL: ', total_keseluruhan)
        print()
        break

    lanjut('transaksi')
else:
    print(' Transaksi tidak bisa dilakukan, karena Inventory Anda Kosong.')
    print('Silahkan kembali ke main menu')
    main_menu(posisi)
```



```
def transaksi_makanminum (makan_minum):
```

```
    global keranjang
    global no_urut
    global total_keseluruhan
    global opsi_opsi
```

```
    while True:
```

```
        menu_pesanan=[]
        sum_fat_menu = 0
        sum_prot_menu = 0
        sum_cal_menu = 0
        sum_harga_menu = 0
```

```
    for i in range(1,len(opsi_opsi)+1):
```

```
        dictionary_opsi_opsi[i][0] #mau mengambil nama dict dalam tuple
        teks=opsi_opsi[i][1] #untuk mengambil string dict dalam tuple
```

```
        display_inventory (dictionary_,teks)
```

```
        validasi_index_input=False
```

```
        while validasi_index_input==False:
```

```
            print('Pisahkan dengan koma jika lebih dari 1 pilihan. Contoh:(1,2)')
            print('Input angka (index) berlaku kelipatan, (contoh:2,2,2 untuk pemesana 3 item yg sama) ')
            print('Input angka nol (0), untuk melewati item ')
            index_input_pilihan = (input(f"Pilih index yang diinginkan (ANGKA: ")
```

```
            print("=====")
```

```
            input_pilihan_split = index_input_pilihan.split(',') # hasil inputan dijadikan list-nya ['1','2','3']
```

```
            for r in input_pilihan_split:
```

```
                if int(r)>len(dictionary_) or int(r)<0: #cek apakah index yg diinput tidak lebih dari len(dict)
```

```
                    print("Angka (index) yang Anda Masukkan Tidak Valid!")
```

```
                    validasi_index_input=False
```

```
                    print("Silahkan input kembali!")
```

```
                    print("=====")
```

```
            else:
```

```
                validasi_index_input=True
```

```
        if validasi_index_input==False:
```

```
            continue
```

Def transaksi_makanminum():

```
def transaksi_makanminum (makan_minum):
```

```
    # CEK_STOK UNTUK INPUTAN YG DI INPUT
```

```
    if validasi_index_input==True:
```

```
        count_char = {}
```

```
        for index_input in input_pilihan_split: #kalkulasi yg index dan brp jumlahnya (neu dipesan)
```

```
            if index_input in count_char:
```

```
                count_char[index_input] +=1 #kalo sudah ada, dijumlahkan
```

```
            else:
```

```
                count_char[index_input] = 1
```

```
    #kalau sudah di rekap di count_char, kita abandingkan dengan stok pada dict dictionary_
```

```
    item_jumlah_pembelian_char={} #isinya teks buat rekapian []
```

```
    for x, (y,z) in enumerate(dictionary_.items()): #slicing keys&value di dict dalam dict (dict dictionary_)
```

```
        for keys, value in count_char.items(): #slicing key&value di dic count_char
```

```
            if int(x)==int(keys)-1: #ketika index dictionary_index keys-1 count_char, cek stok
```

```
                if int(value)>int(z["stock"]): #klo stok pesanan di count_char > stok di dictionary_ maka stok tdk cukup
                    print(f'Mohon maaf stock {y} tidak cukup')
```

```
                validasi_index_input=False
```

```
                menu_pesanan=[]
```

```
                print("Silahkan input kembali!")
```

```
                print("=====")
```

```
            else: #stock aman
```

```
                teks= str(value) + ' ' + str(y)
```

```
                item_jumlah_pembelian_char.append(teks) +
```

```
                menu_pesanan.append(item_jumlah_pembelian_char)
```

```
    if validasi_index_input==False:
```

```
        continue
```

```
    if validasi_index_input==True:
```

```
        for index_bekurang, stock_bekurang in count_char.items():
```

```
            for index_dict, (item_dict, stock) in enumerate(dictionary_.items()):
```

```
                if (int(index_bekurang)-1)==(index_dict):
```

```
                    dictionary_[item_dict]['stock']=stock['stock']-stock_bekurang
```

```
                    sum_fat_menu= sum_fat_menu+(float(stock_bekurang))*float(dictionary_[item_dict]['fat'])
```

```
                    sum_prot_menu= sum_prot_menu+(float(stock_bekurang))*float(dictionary_[item_dict]['protein'])
```

```
                    sum_cal_menu= sum_cal_menu+(float(stock_bekurang))*float(dictionary_[item_dict]['calories'])
```

```
                    sum_harga_menu= sum_harga_menu+(stock_bekurang*dictionary_[item_dict]['harga'])
```

Isinya index yang dipilih, banyaknya yg dipesan. contoh: {'5':2}

Mengambil teks. Contoh 2 Italian Hearscheese

Teks dicatat ke dalam item_jumlah_pembelian_char[]

Item_jumlah_pembelian_char dicatat kedalam menu_pesanan[]

Update stok

Kalkulasi informasi gizi dan harga dari item-item yang dipilih dan menjadi satu kesatuan menu custom.

Def `transaksi_makanminum()`:
(...continue)

```
def transaksi_makanminum (makan_minum):  
    teks_rekapan_menu= ''  
    for item in menu_pesanan:  
        → teks_rekapan_menu +=item[0]+' '  Agar teks rekapannya kesamping  
  
    nutrisi_rekap_menu=(f'Fat :{sum_fat_menu:.2f}, Prot:{sum_prot_menu:.2f}, Cal:{sum_cal_menu:.2f}, Harga Item:{sum_harga_menu}')  
  
    print(teks_rekapan_menu[:-1])  
    print(nutrisi_rekap_menu)  
    print('berhasil ditambahkan ke chart.')  
    print("=====  
  
    item_belanja={teks_rekapan_menu[:-1]:nutrisi_rekap_menu}  
    → keranjang.append(item_belanja)  
  
    → total_keseluruhan=total_keseluruhan+sum_harga_menu  
  
    lanjut_belanja=input(f'Apakah Anda mau melanjutkan berbelanja {makan_minum} (Y/N)? ').lower()  
    print("=====  
    if lanjut_belanja!='y':  
        break  
  
#=====
```

Selamat Datang di Aplikasi Point of Sales

Silahkan masukkan profesi Anda: kasir

Def transaksi():
(.continue.. running
program))

Menu Utama Kasir:

1. Transaksi
2. Log in dengan posisi lain
3. Keluar Program

Masukkan Menu Pilihan Anda (angka) 1

-----TRANSAKSI PENJUALAN-----

DAFTAR INVENTOR / ROTI

Index	Cal	Fat	Prot	Stock	Harga	Roti
-------	-----	-----	------	-------	-------	------

1	271	12.09	9.53	1	15000	Italian HerbsCheese
2	271	3.5	8.8	1	15000	Italian Bread
3	184	0.91	6.73	100	17000	Whole Wheat Tortilla
4	266	3.29	7.64	100	10000	Flat White Bread

Pisahkan dengan koma jika lebih dari 1 pilihan. Contoh:(1,2)

Input angka (index) berlaku kelipatan, (contoh:2,2,2 untuk pemesana 3 item yg sama)

Input angka nol (0), untuk melewati item)

Pilih index yang diinginkan (ANGKA): 1

DAFTAR INVENTORY PROTEIN

Index	Cal	Fat	Prot	Stock	Harga	Protein
-------	-----	-----	------	-------	-------	---------

1	59	5.11	2.9	100	12000	Smoked Beef Salami
2	130	11.28	5.7	100	10000	Pepperoni
3	105	5.5	6.6	100	15000	Chicken Strips
4	291	22.23	21.11	100	20000	Beef Brisket

Pisahkan dengan koma jika lebih dari 1 pilihan. Contoh:(1,2)

Input angka (index) berlaku kelipatan, (contoh:2,2,2 untuk pemesana 3 item yg sama)

Input angka nol (0), untuk melewati item)

Pilih index yang diinginkan (ANGKA): 1

DAFTAR INVENTOR / KEJU

Index	Cal	Fat	Prot	Stock	Harga	Keju
-------	-----	-----	------	-------	-------	------

1	51	5.06	1.09	100	3000	Cream Cheese
2	85	5.61	7.27	100	7000	Mozarella
3	113	9.28	6.97	100	5000	Cheddar Cheese

Pisahkan dengan koma jika lebih dari 1 pilihan. Contoh:(1,2)

Input angka (index) berlaku kelipatan, (contoh:2,2,2 untuk pemesana 3 item yg sama)

Input angka nol (0), untuk melewati item)

Pilih index yang diinginkan (ANGKA): 1

DAFTAR INVENTOR SAUS

Index	Cal	Fat	Prot	Stock	Harga	Saus
-------	-----	-----	------	-------	-------	------

1	57	4.91	0.13	100	2000	Mayo
2	58	5.47	0.17	100	3000	Thousand Island
3	44	4.26	0.06	100	3000	Italian Dressing
4	40	4.5	0	100	4000	Sesame oil
5	100	9	1	100	4000	Roasted Sesame

Pisahkan dengan koma jika lebih dari 1 pilihan. Contoh:(1,2)

Input angka (index) berlaku kelipatan, (contoh:2,2,2 untuk pemesana 3 item yg sama)

Input angka nol (0), untuk melewati item)

Pilih index yang diinginkan (ANGKA): 1

DAFTAR INVENTOR SAYUR

Index	Cal	Fat	Prot	Stock	Harga	Sayur
-------	-----	-----	------	-------	-------	-------

1	8	0.08	0.5	100	2000	Lettuce
2	6	0.06	0.27	100	1000	Tomatoes
3	6	0.08	0.3	100	2000	Cucumber
4	48	0.09	1.06	100	3000	Onion
5	42	4.44	0.3	100	5000	Green Olives

Pisahkan dengan koma jika lebih dari 1 pilihan. Contoh:(1,2)

Input angka (index) berlaku kelipatan, (contoh:2,2,2 untuk pemesana 3 item yg sama)

Input angka nol (0), untuk melewati item)

Pilih index yang diinginkan (ANGKA): 1

1 Italian HerbsCheese 1 Smoked Beef Salami 1 Cream Cheese 1 Mayo 1 Lettuce

Fat :27.25, Prot:14.15, Cal:446.00, Harga Item:34000

berhasil ditambahkan ke chart.

Apakah Anda mau melanjutkan berbelanja makanan (Y/N)? y

DAFTAR INVENTOR ROTI

Index	Cal	Fat	Prot	Stock	Harga	Roti
-------	-----	-----	------	-------	-------	------

1	271	12.09	9.53	0	15000	Italian HerbsCheese
2	271	3.5	8.8	1	15000	Italian Bread
3	184	0.91	6.73	100	17000	Whole Wheat Tortilla
4	266	3.29	7.64	100	10000	Flat White Bread

Pisahkan dengan koma jika lebih dari 1 pilihan. Contoh:(1,2)

Input angka (index) berlaku kelipatan, (contoh:2,2,2 untuk pemesana 3 item yg sama)

Input angka nol (0), untuk melewati item)

Pilih index yang diinginkan (ANGKA): 2

DAFTAR INVENTORY SAYUR

Index	Cal	Fat	Prot	Stock	Harga	Sayur
-------	-----	-----	------	-------	-------	-------

1	8	0.08	0.5	99	2000	Lettuce
2	6	0.06	0.27	99	1000	Tomatoes
3	6	0.08	0.3	100	2000	Cucumber
4	48	0.09	1.06	100	3000	Onion
5	42	4.44	0.3	100	5000	Green Olives

Pisahkan dengan koma jika lebih dari 1 pilihan. Contoh:(1,2)

Input angka (index) berlaku kelipatan, (contoh:2,2,2 untuk pemesana 3 item yg sama)

Input angka nol (0), untuk melewati item)

Pilih index yang diinginkan (ANGKA): 3

1 Whole Wheat Tortilla 1 Chicken Strips 1 Cheddar Cheese 1 Italian Dressing 1 Cucumber

Fat :20.03, Prot:20.66, Cal:452.00, Harga Item:42000

berhasil ditambahkan ke chart.

Apakah Anda mau melanjutkan berbelanja makanan (Y/N)? n

Apakah Anda mau memesan minum (Y/N)? y

DAFTAR INVENTORY MINUMAN

Index	Cal	Fat	Prot	Stock	Harga	Minuman
-------	-----	-----	------	-------	-------	---------

1	0	0	0	100	7000	Mineral Water
2	0	0	0	100	10000	Diet Coke

Pisahkan dengan koma jika lebih dari 1 pilihan. Contoh:(1,2)

Input angka (index) berlaku kelipatan, (contoh:2,2,2 untuk pemesana 3 item yg sama)

Input angka nol (0), untuk melewati item)

Pilih index yang diinginkan (ANGKA): 1,1

2 Mineral Water

Fat :0.00, Prot:0.00, Cal:0.00, Harga Item:14000

berhasil ditambahkan ke chart.

Apakah Anda mau melanjutkan berbelanja minuman (Y/N)? n

KERANJANG BELANDA ANDA

1. 1 Italian HerbsCheese 1 Smoked Beef Salami 1 Cream Cheese 1 Mayo 1 Lettuce

Fat :27.25, Prot:14.15, Cal:446.00, Harga Item:34000

2. 1 Italian Bread 1 Pepperoni 1 Mozarella 1 Thousand Island 1 Tomatoes

Fat :25.92, Prot:22.21, Cal:550.00, Harga Item:36000

3. 1 Whole Wheat Tortilla 1 Chicken Strips 1 Cheddar Cheese 1 Italian Dressing 1 Cucumber

Fat :20.03, Prot:20.66, Cal:452.00, Harga Item:42000

4. 2 Mineral Water

Fat :0.00, Prot:0.00, Cal:0.00, Harga Item:14000

TOTAL BELANDA : 126000

PEMBAYARAN TUNAI : 130000

KEMBALI : 4000

PEMBAYARAN BERHASIL, TERIMA KASIH TELAH BERBELANJA

-----KEDATANGAN ANDA SELALU KAMI TUNGGU :-----

Apakah Anda mau mengakses menu transaksi kembali (Y/N)?

Def transaksi():
(..continue.. running
program))

```
*=====
Selamat Datang di Aplikasi Point of Sales
=====

Silahkan masukkan profesi Anda: owner
=====

Menu Utama Owner:
1. Menampilkan Inventory Bahan Baku
2. Menambah Inventory Bahan Baku
3. Menghapus Inventory Bahan Baku
4. Update Inventory Bahan Baku
5. Transaksi
6. Log in dengan posisi lain
7. Keluar Program
=====

Masukkan Menu Pilihan Anda (angka): 3
=====

Menghapus Inventory Gudang:
1. Menghapus Inventory Bread
2. Menghapus Inventory Protein
3. Menghapus Inventory Cheese
4. Menghapus Inventory Sauce
5. Menghapus Inventory Veggie
6. Menghapus Inventory Drink
7. Menghapus Seluruh Inventory
0. Kembali ke main menu
=====

Masukkan Menu Pilihan Anda (angka): 7
=====

DAFTAR INVENTORY BAHAN BAKU
Index  ||Cal  ||Fat  ||Prot ||Stock ||Harga ||Bahan Baku
=====
1      ||271  ||12.09||9.53 ||0      ||15000 ||Italian HerbsCheese
2      ||271  ||3.5   ||8.8  ||0      ||15000 ||Italian Bread
3      ||184  ||0.91  ||6.73 ||99     ||17000 ||Whole Wheat Tortilla
4      ||266  ||3.29  ||7.64 ||100    ||10000 ||Flat White Bread
5      ||59   ||5.11  ||2.9  ||99     ||12000 ||Smoked Beef Salami
6      ||130  ||11.28 ||5.7  ||99     ||10000 ||Pepperoni
7      ||185  ||5.5   ||6.6  ||99     ||15000 ||Chicken Strips
8      ||291  ||22.23 ||21.11||100    ||20000 ||Beef Brisket
9      ||51   ||5.06  ||1.09 ||99     ||3000  ||Cream Cheese
10     ||85   ||5.61  ||7.27 ||99     ||7000  ||Mozarella
11     ||113  ||9.28  ||6.97 ||99     ||5000  ||Cheddar Cheese
12     ||57   ||4.91  ||0.13 ||99     ||2000  ||Mayo
13     ||58   ||5.47  ||0.17 ||99     ||3000  ||Thousand Island
14     ||44   ||4.26  ||0.06 ||99     ||3000  ||Italian Dressing
15     ||40   ||4.5   ||0     ||100    ||4000  ||Sesame oil
16     ||100  ||9     ||1     ||100    ||4000  ||Roasted Sesame
17     ||8     ||0.08  ||0.5  ||99     ||2000  ||Lettuce
18     ||6     ||0.06  ||0.27 ||99     ||1000  ||Tomatoes
19     ||6     ||0.08  ||0.3  ||99     ||2000  ||Cucumber
20     ||48   ||0.09  ||1.06 ||100    ||3000  ||Onion
21     ||42   ||4.44  ||0.3  ||100    ||5000  ||Green Olives
22     ||0     ||0     ||0     ||98     ||7000  ||Mineral Water
23     ||0     ||0     ||0     ||100    ||10000 ||Diet Coke
=====
```

Anda yakin akan menghapus (Y/N)?y

Seluruh Inventory Telah di Hapus

Apakah Anda mau mengakses menu menghapus inventory kembali (Y/N)? n

Apakah Anda mau kembali ke main menu (Y/N)? y

Menu Utama Owner:

1. Menampilkan Inventory Bahan Baku
2. Menambah Inventory Bahan Baku
3. Menghapus Inventory Bahan Baku
4. Update Inventory Bahan Baku
5. Transaksi
6. Log in dengan posisi lain
7. Keluar Program

Masukkan Menu Pilihan Anda (angka): 5

Transaksi tidak bisa dilakukan, karena Inventory Anda Kosong.
Silahkan kembali ke main menu

Menu Utama Owner:

1. Menampilkan Inventory Bahan Baku
2. Menambah Inventory Bahan Baku
3. Menghapus Inventory Bahan Baku
4. Update Inventory Bahan Baku
5. Transaksi
6. Log in dengan posisi lain
7. Keluar Program

Masukkan Menu Pilihan Anda (angka): 0

Terima
Kasih

