



LAPORAN PRAKTIKUM PROGRAMA KOMPUTER

DISUSUN OLEH :

Aditya Agung Nugroho	I0322002
Angelique Yoanita	I0322014
Charisma Cintani	I0322027
Chezava Deananta	I0322028



LAPORAN

PRAKTIKUM PROGRAM KOMPUTER



“FOOD OREDERING”

Disusun Oleh:

KELOMPOK 02

- | | |
|--------------------------------|----------|
| 1. Aditya Agung Nugroho | I0322002 |
| 2. Angelique Yoanita | I0322015 |
| 3. Charisma Cintani Adhi Surya | I0322027 |
| 4. Chezava Deananta | I0322028 |

5.

PROGRAM STUDI S1 TEKNIK INDUSTRI FAKULTAS TEKNIK
UNIVERSITAS SEBELAS MARET
SURAKARTA

2023



BAB I

DESKRIPSI MASALAH

Di tengah perkembangan teknologi yang cukup pesat ini menuntut semuanya untuk terus beradaptasi dengan adanya berbagai perubahan. Salah satunya adalah restoran yang dituntut untuk terus dapat menciptakan berbagai inovasi yang dapat memudahkan pelanggan dan restoran tersebut dalam bertransaksi. Sebuah restoran harus mampu untuk memenuhi keinginan pelanggan dengan baik, cepat dan tepat, salah satunya dengan memiliki sistem pencatatan (kasir) yang baik. Hal tersebut dapat dilakukan dengan mempertimbangkan aspek penggunaan teknologi terbaru, seperti aplikasi *mobile* atau perangkat lunak khusus, yang dapat mengotomatiskan proses pencatatan pesanan, pengelolaan inventaris, dan pembayaran. Dalam upaya meningkatkan efisiensi dan kualitas layanan, restoran juga dapat mengintegrasikan sistem pencatatan dengan sistem lainnya, seperti sistem manajemen pemesanan online yang memungkinkan pelanggan untuk memesan makanan secara *online*.

Selain itu, pengembangan sistem pencatatan yang baik juga mencakup pelatihan karyawan restoran dalam penggunaan sistem tersebut. Restoran perlu memastikan bahwa karyawan memiliki pemahaman yang baik tentang penggunaan sistem, termasuk kemampuan untuk mengelola pesanan dengan efisien, mengatasi masalah teknis, dan memberikan dukungan pelanggan yang memadai.

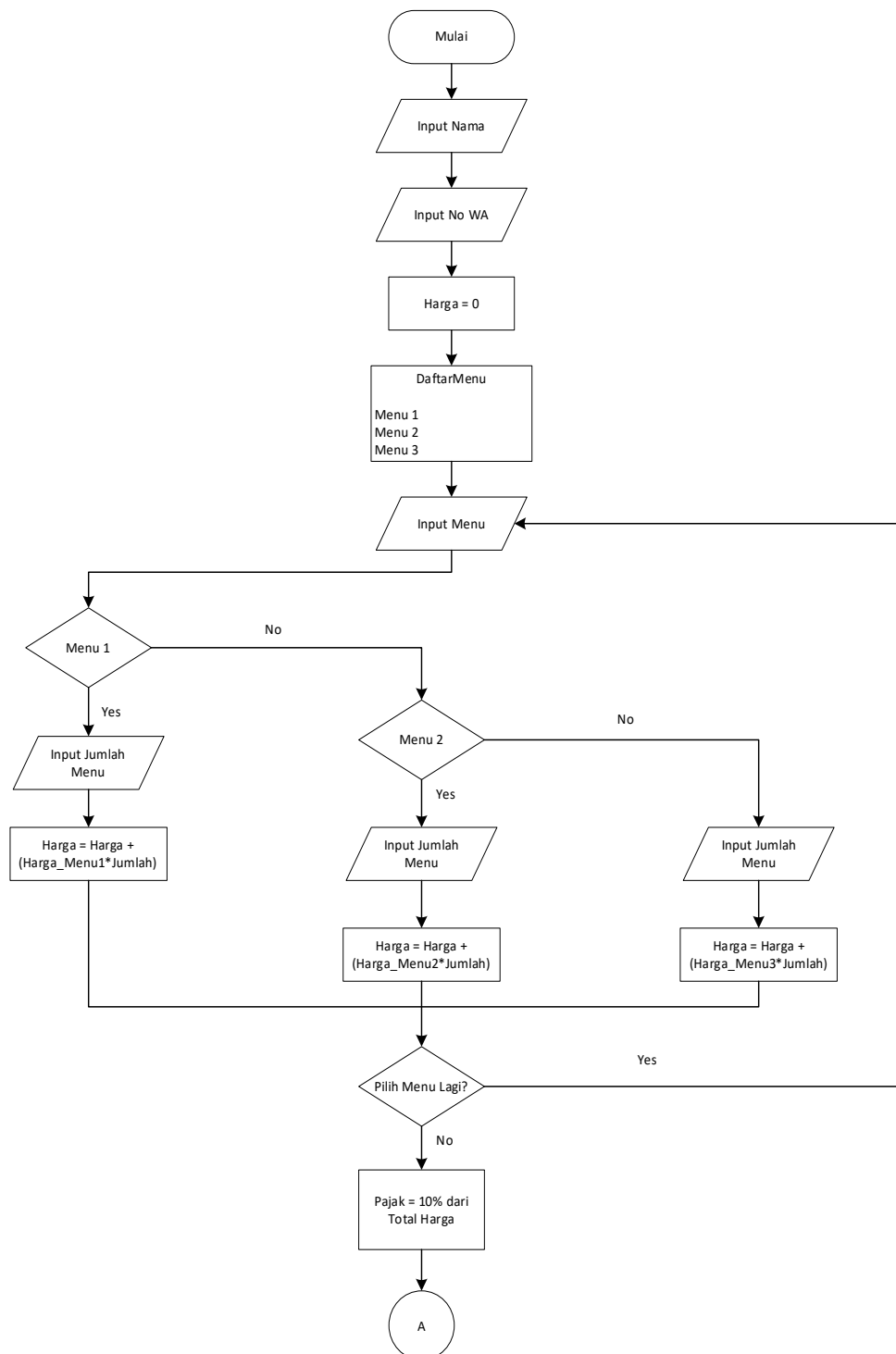
Dengan mengembangkan sistem pencatatan atau kasir yang baik, restoran dapat meningkatkan produktivitas, meningkatkan kepuasan pelanggan, dan mengoptimalkan pengelolaan operasional mereka. Selain itu, mereka akan dapat beradaptasi dengan perubahan teknologi yang terus berkembang dan tetap bersaing dalam industri restoran yang semakin kompetitif.

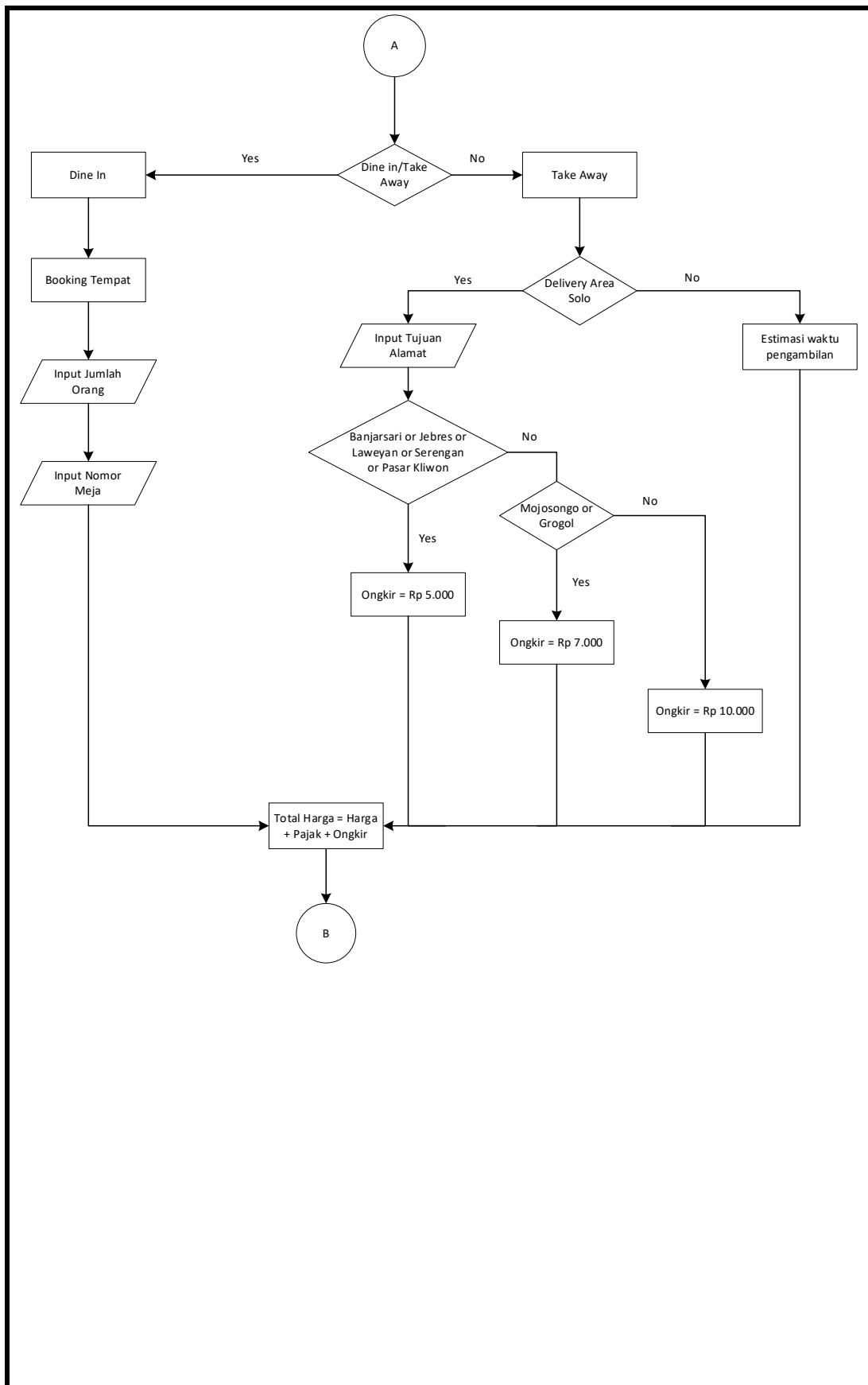


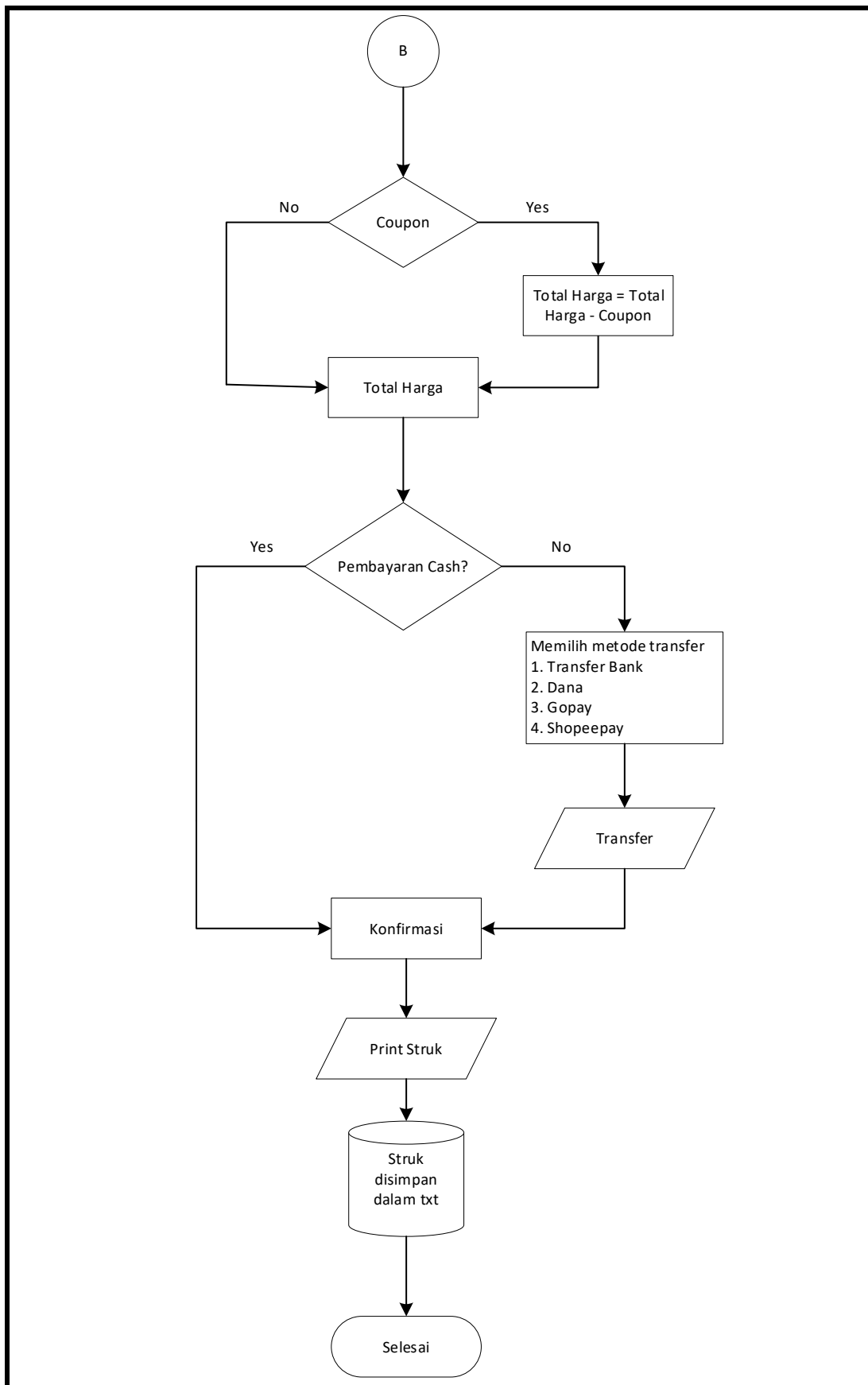
BAB II

FLOWCHART PEMROGRAMAN BESERTA PENJELASANNYA

2.1 Flowchart









2.2 Penjelasan Flowchart

Pada program kasir berikut ini, pembeli diminta untuk *input* nama dan nomor telepon dari *user* itu sendiri. Harga dimulai dari 0 rupiah. Kemudian *user* diminta untuk *input* menu yang diinginkan. Setelah *input* menu yang diinginkan, program akan menghitung harga dengan rumus $\text{harga} + (\text{harga menu yang dipilih} * \text{jumlah})$. Program akan mengenakan pajak pada total pesanan yang telah dipilih oleh *user*.

Program akan memberikan 3 opsi kepada *user* yaitu, *dine in*, *take away*, dan yang terakhir adalah pilihan untuk keluar dari program kasir. Jika *user* memilih untuk *dine in*, maka program akan memberi perintah agar *user booking* tempat dan *input* jumlah orang dan *input* nomor meja yang dipilih. Di saat *user* memilih *take away*, program akan memberi pilihan yaitu *take away* atau *delivery*. Jika memilih *delivery* maka program akan meminta *user* untuk input tujuan alamat pengiriman. Daerah Banjarsari, Jebres, Laweyan, Pasar Kliwon akan mendapat ongkos kirim sama yaitu sebesar 5000 rupiah. Daerah Mojosongo dan Grogol akan mendapat ongkir 7000 rupiah. Sedangkan Daerah Colomadu dan Ngemplak akan mendapatkan ongkos kirim sebesar 10000 rupiah. Selain daerah – daerah tersebut maka tidak melayani *delivery*. Namun disaat *user* memilih opsi *take away* maka program akan memberikan estimasi waktu dari pembuatan menu makanan yang telah dipilih *user*. Setelah itu program akan menghitung total harga dengan menjumlahkan harga menu, pajak dan harga ongkos kirim.

Dalam proses pembayaran, program akan memberikan opsi mengenai kupon. Bila *user* memiliki kupon maka bisa di *input* ke dalam program dan jika kupon yang di *input user* tersedia maka total harga akan dikurangi dengan jumlah kupon yang telah dipilih. Program akan memberikan pilihan kepada *user* yaitu pembayaran dengan metode tunai atau *cash* dan pembayaran dengan metode *transfer* yang mencakup *transfer bank*, *Dana*, *Gopay* dan *Shopeepay*. Jika memilih metode *transfer* maka *user* dapat memilih metode *transfer* yang diinginkan dan dapat melakukan transaksi. Setelah melakukan transaksi maka program akan meminta *user* untuk mengkonfirmasi kembali. Setelah konfirmasi maka struk akan di print dan tersimpan di dalam file txt dan program kasir selesai.



BAB III

KODE PROGRAM DI PYTHON

3.1. Fungsi Daftar Menu

```
namefile = 'daftar_menu.csv'

def daftarmenu():
    data = []
    with open (namefile) as filecsv:
        readcsv = csv.reader (filecsv,delimiter = ',')
        data = list(readcsv)
    headers = data[0]
    tablefmt = "grid"
    numalign = "center"
    for row in data:
        del row[3]
    table = tabulate(data[1:], headers=headers, tablefmt=tablefmt, numalign=numalign)
    print(table)
```

Program di atas adalah sebuah program yang membaca dan menampilkan isi dari *file* CSV yang berisi daftar menu. Nama *file* CSV yang digunakan disimpan dalam variabel `namefile`. Program ini menggunakan modul `csv` untuk membaca *file* CSV. Pertama, program membuka *file* CSV menggunakan pernyataan `with open(namefile) as filecsv`. Kemudian, menggunakan fungsi `csv.reader(filecsv, delimiter=',')`, program membaca isi *file* CSV dan menyimpannya dalam variabel `readcsv`. Data yang dibaca dari *file* CSV kemudian diubah menjadi *list* menggunakan fungsi `list(readcsv)`, dan disimpan dalam variabel `data`.

Selanjutnya, program mengambil baris pertama dari data yang merupakan *header* kolom. Variabel `headers` menyimpan *header* tersebut. Variabel `tablefmt` digunakan untuk mengatur format tampilan tabel, sedangkan `numalign` digunakan untuk mengatur perataan angka dalam tabel. Program ini juga melakukan penghapusan elemen ke-3 pada setiap baris data dengan menggunakan pernyataan `del row [3]` untuk menghapus kolom yang tidak diperlukan.

Setelah itu, program menggunakan fungsi `tabulate` untuk menghasilkan tabel yang akan ditampilkan. Tabel tersebut terdiri dari data mulai dari baris kedua, dengan header yang telah ditentukan sebelumnya. Format tabel dan perataan angka



diatur sesuai dengan nilai dari `tablefmt` dan `numalign`. Hasil tabel tersebut kemudian ditampilkan menggunakan pernyataan `print(table)`. Dengan demikian, program ini membaca *file* CSV yang berisi daftar menu, menghapus kolom yang tidak diperlukan, dan menampilkan isi daftar menu dalam bentuk tabel.

3.2. Fungsi Nomor Meja

```
def database_meja(database):
    with open("data_meja.csv", "w", newline="") as file:
        writer = csv.writer(file)
        writer.writerow(["Nomor Meja"])
        for nomor_meja in database:
            writer.writerow([nomor_meja])

def baca_database():
    database = []
    try:
        with open("data_meja.csv", "r") as file:
            reader = csv.reader(file)
            next(reader) # Melompati baris header
            for row in reader:
                nomor_meja = int(row[0])
                database.append(nomor_meja)
    except FileNotFoundError:
        return database
    return database

def pesan_nomor_meja():
    global nomor_meja
    database = baca_database()
    while True:
        try:
            nomor_meja = int(input("Masukkan nomor meja: "))
            break
        except ValueError:
            print("input salah")
    while nomor_meja in database or nomor_meja > 10:
        if nomor_meja in database:
            print("Nomor meja telah dipesan. Silakan pilih nomor meja lain.")
        elif nomor_meja > 10:
            print("Meja tidak tersedia.")
        nomor_meja = int(input("Masukkan nomor meja: "))
    database.append(nomor_meja)
```



```
database_meja(database)
print("Nomor meja berhasil dipesan.")

def tampilkan_nomor_meja_tersedia():
    database = baca_database()
    nomor_meja_tersedia = [nomor for nomor in range(1, 11) if nomor not in database]
    print("Nomor meja yang tersedia: ")
    print(nomor_meja_tersedia)
```

Program di atas merupakan program sederhana untuk mengelola *database* nomor meja dalam sebuah restoran. Program ini terdiri dari beberapa fungsi yang saling berinteraksi untuk membaca, menambah, dan menampilkan nomor meja yang tersedia.

Fungsi pertama adalah `'database_meja'` yang bertujuan untuk menyimpan nomor meja ke dalam file CSV. Fungsi ini menerima argumen `'database'` yang berisi daftar nomor meja yang ingin disimpan. Fungsi akan membuka *file* "data_meja.csv" dan menulis nomor meja ke dalam *file* tersebut.

Fungsi kedua adalah `'baca_database'` yang digunakan untuk membaca *file* CSV yang berisi nomor meja. Fungsi ini akan membuka *file* "data_meja.csv" dan membaca nomor meja yang tersimpan di dalamnya. Nomor meja tersebut kemudian akan dimasukkan ke dalam sebuah *list* yang akan dikembalikan oleh fungsi ini.

Fungsi ketiga adalah `'pesan_nomor_meja'` yang bertugas untuk meminta *input* nomor meja dari pengguna. Fungsi akan memanggil fungsi `'baca_database'` untuk mendapatkan nomor meja yang telah dipesan sebelumnya. Selanjutnya, pengguna akan diminta untuk memasukkan nomor meja yang diinginkan. Jika nomor meja yang dimasukkan tidak berupa angka, maka akan muncul pesan kesalahan. Selanjutnya, fungsi akan memeriksa apakah nomor meja tersebut telah dipesan sebelumnya atau apakah nomor meja tersebut melebihi jumlah maksimum meja yang tersedia. Jika kondisi tersebut terpenuhi, maka akan muncul pesan kesalahan dan pengguna diminta untuk memasukkan nomor meja yang lain. Jika nomor meja valid, nomor meja akan ditambahkan ke dalam database menggunakan fungsi `'database_meja'` dan pesan berhasil akan ditampilkan.

Fungsi terakhir adalah `'tampilkan_nomor_meja_tersedia'` yang digunakan untuk menampilkan nomor meja yang masih tersedia. Fungsi ini akan memanggil



fungsi `baca_database` untuk mendapatkan nomor meja yang telah dipesan sebelumnya. Selanjutnya, fungsi akan membuat list `nomor_meja_tersedia` yang berisi nomor meja yang belum dipesan. List tersebut akan ditampilkan sebagai *output*.

Dengan menggunakan program ini, pengguna dapat memesan nomor meja yang tersedia dan melihat nomor meja yang masih tersedia di restoran tersebut. Program ini juga menyimpan data nomor meja yang telah dipesan dalam file CSV untuk penggunaan berikutnya.

3.3. Fungsi Ongkos Kirim

```
def hitung_ongkir_menu(kecamatan, totalharga):
    global ongkir
    ongkir = 0
    kecamatan = kecamatan.lower()
    if kecamatan == "jebres" or kecamatan == "banjarsari" or kecamatan == "laweyan"
or kecamatan == "serengan" or kecamatan == "pasar kliwon":
        ongkir = 5000
    elif kecamatan == "mojosongo" or kecamatan == "grogol":
        ongkir = 7000
    elif kecamatan == "colomadu" or kecamatan == "ngemplak":
        ongkir = 10000
    else:
        print("Kami Hanya Melayani Delivery Area Solo")
        ongkir == 0
    total_biaya = ongkir + totalharga
    return total_biaya
```

Program di atas adalah sebuah fungsi yang digunakan untuk menghitung biaya ongkir berdasarkan kecamatan pengiriman dan total harga pesanan. Fungsi `hitung_ongkir_menu` memiliki dua parameter, yaitu `kecamatan` yang merupakan string yang menyatakan kecamatan pengiriman, dan `totalharga` yang merupakan total harga pesanan.

Pertama, fungsi ini menginisialisasi variabel `ongkir` dan mengatur nilainya menjadi 0. Kemudian, kecamatan yang dimasukkan oleh pengguna akan diubah menjadi huruf kecil menggunakan metode `lower()` agar lebih fleksibel dalam membandingkan kondisi.



Selanjutnya, dilakukan serangkaian kondisi dengan menggunakan pernyataan *if-elif-else*. Jika kecamatan pelanggan adalah "jebres", "banjarsari", "laweyan", "serengan", atau "pasar kliwon", maka biaya ongkir akan ditetapkan sebesar 5000. Jika kecamatan pelanggan adalah "mojosongo" atau "grogol", maka biaya ongkir akan ditetapkan sebesar 7000. Jika kecamatan pelanggan adalah "colomadu" atau "ngemplak", maka biaya ongkir akan ditetapkan sebesar 10000. Jika kecamatan pelanggan tidak termasuk dalam kecamatan-kecamatan tersebut, maka akan ditampilkan pesan bahwa pengiriman hanya tersedia di daerah Solo, dan nilai ongkir akan tetap 0. Selanjutnya, total biaya dihitung dengan menambahkan nilai ongkir dengan total harga pesanan. Nilai total biaya ini akan dikembalikan oleh fungsi sebagai hasil eksekusi.

Dengan menggunakan program ini, pengguna dapat menghitung biaya ongkir berdasarkan kecamatan pengiriman dan total harga pesanan. Program ini mempermudah pengguna dalam mengestimasi total biaya yang harus dibayarkan, termasuk biaya ongkir, berdasarkan kecamatan pengiriman yang ditentukan.

3.4. Fungsi Kode Kupon

```
def kodekupon(kode, totalharga):  
    global kupon  
    kupon = 0  
    if kode == "caca0214":  
        kupon = 5000  
    elif kode == "caca2728":  
        kupon = 7000  
    else:  
        kupon = 0  
    hargakupon = totalharga - kupon  
    return hargakupon
```

Program di atas adalah sebuah fungsi dengan nama 'kodekupon' yang digunakan untuk menerapkan diskon kupon pada total harga pembelian. Fungsi ini memiliki dua parameter, yaitu 'kode' (kode kupon) dan 'totalharga' (total harga pembelian). Di awal fungsi, terdapat deklarasi variabel global 'kupon' yang



diinisialisasi dengan nilai 0. Variabel ini digunakan untuk menyimpan nilai diskon dari kupon yang diberikan.

Selanjutnya, terdapat struktur pengkondisian `if-elif-else` untuk memeriksa nilai `kode` yang diberikan. Jika nilai `kode` adalah "caca0214", maka nilai `kupon` akan diatur menjadi 5000. Jika nilai `kode` adalah "caca2728", maka nilai `kupon` akan diatur menjadi 7000. Jika nilai `kode` tidak sama dengan kedua nilai tersebut, maka nilai `kupon` akan tetap 0. Setelah nilai `kupon` ditentukan, program menghitung nilai `hargakupon` dengan mengurangi nilai `kupon` dari `totalharga`. Hal ini dilakukan untuk mengaplikasikan diskon kupon pada total harga pembelian. Terakhir, fungsi mengembalikan nilai `hargakupon` sebagai hasil dari perhitungan diskon kupon.

3.5. Fungsi Metode Pembayaran Transfer

```
def metode_transfer(norek_resto, no_telepon, totalharga):
    valid_input = False
    while not valid_input:
        nomor_rekening = input("Input nomor rekening tujuan: ")
        if nomor_rekening == norek_resto or nomor_rekening == no_telepon:
            valid_harga = False
            while not valid_harga:
                totalbayar = float(input("Masukan total pembayaran: "))
                if totalbayar == totalharga:
                    valid_harga = True
                    valid_input = True
                else:
                    print("Tidak valid")
            else:
                print("nomor rekening tidak valid")
    return True
```

Program di atas merupakan sebuah fungsi dengan nama `metode_transfer` yang digunakan untuk melakukan transfer pembayaran. Fungsi ini memiliki tiga parameter, yaitu `norek_resto` (nomor rekening tujuan), `no_telepon` (nomor telepon), dan `totalharga` (total harga pembayaran). Pada awalnya, variabel `valid_input` diinisialisasi sebagai *False* untuk mengontrol perulangan validasi *input*. Kemudian, program memasuki perulangan *while* yang berlangsung selama *input* yang



diberikan belum valid. Di dalam perulangan tersebut, pengguna diminta untuk memasukkan nomor rekening tujuan melalui `nomor_rekening = input("Input nomor rekening tujuan: ")`. Jika nomor rekening yang dimasukkan sama dengan `norek_resto` atau `no_telepon`, program melanjutkan dengan memvalidasi jumlah pembayaran.

Program kemudian memasuki perulangan *while* bersarang yang akan berjalan selama jumlah pembayaran belum valid. Di dalamnya, pengguna diminta untuk memasukkan total pembayaran melalui `totalbayar = float(input("Masukkan total pembayaran: "))`. Jika jumlah pembayaran yang dimasukkan sama dengan `totalharga`, maka jumlah pembayaran dianggap valid dan perulangan selesai. Namun, jika jumlah pembayaran yang dimasukkan tidak sama dengan `totalharga`, program akan menampilkan pesan "Tidak valid" dan mengulang perulangan pembayaran.

Jika nomor rekening yang dimasukkan tidak sama dengan `norek_resto` atau `no_telepon`, program akan menampilkan pesan "nomor rekening tidak valid" dan mengulang perulangan utama. Setelah pengguna memasukkan nomor rekening tujuan dan jumlah pembayaran yang valid, fungsi akan mengembalikan *True*. Secara keseluruhan, program ini memastikan bahwa pengguna memberikan nomor rekening tujuan yang valid dan jumlah pembayaran yang sesuai sebelum melanjutkan proses transfer pembayaran.

3.6. Fungsi Pajak

```
def harga_pajak1(totalharga):  
    pajak = totalharga * 0.1  
    totalpajak = totalharga + pajak  
    return totalpajak  
  
def harga_pajak2(biaya_total):  
    pajak = biaya_total * 0.1  
    totalpajak = biaya_total + pajak  
    return totalpajak
```

Program di atas terdiri dari dua fungsi, yaitu `'harga_pajak1'` dan `'harga_pajak2'`, yang digunakan untuk menghitung total harga pesanan setelah



dikenakan pajak. Fungsi `harga_pajak1` menerima parameter `totalharga`, yang merupakan total harga pesanan sebelum dikenakan pajak. Di dalam fungsi ini, dilakukan perhitungan pajak dengan mengalikan `totalharga` dengan 0.1 (atau 10% dalam bentuk desimal) dan hasilnya disimpan dalam variabel `pajak`. Selanjutnya, total harga setelah pajak dihitung dengan menambahkan `totalharga` dengan `pajak`, dan hasilnya disimpan dalam variabel `totalpajak`. Nilai `totalpajak` ini kemudian dikembalikan sebagai output dari fungsi.

Fungsi `harga_pajak2` memiliki parameter `biaya_total`, yang merupakan total biaya pesanan setelah dikenakan ongkir. Fungsi ini memiliki langkah-langkah yang serupa dengan fungsi sebelumnya. Pajak dihitung dengan mengalikan `biaya_total` dengan 0.1, dan hasilnya disimpan dalam variabel `pajak`. Kemudian, total harga setelah pajak dihitung dengan menambahkan `biaya_total` dengan `pajak`, dan hasilnya disimpan dalam variabel `totalpajak`. Nilai `totalpajak` dikembalikan sebagai output dari fungsi.

Kedua fungsi tersebut berfungsi untuk menghitung total harga pesanan setelah dikenakan pajak dengan persentase 10% atau 0.1. Pengguna dapat menggunakan fungsi `harga_pajak1` jika mereka hanya memiliki informasi tentang total harga pesanan sebelum dikenakan ongkir atau pajak. Sedangkan, jika pengguna sudah memiliki total biaya pesanan setelah dikenakan ongkir, mereka dapat menggunakan fungsi `harga_pajak2` untuk menghitung total harga pesanan setelah dikenakan pajak.

Program ini memberikan kemudahan bagi pengguna untuk menghitung total harga pesanan yang sudah termasuk pajak. Dengan menggunakan fungsi-fungsi ini, pengguna dapat dengan mudah menghitung total pembayaran yang harus mereka bayarkan.

3.7. Fungsi Kasir

```
def programkasir():  
    with open(namefile, newline='') as filecsv:  
        reader = csv.reader(filecsv)  
        datamenu = list(reader)  
        nama = str(input("Ketik Nama Anda: "))  
        while True:  
            try:
```



```
        nomor_telepon = int(input("Masukkan nomor telepon: "))
        break
    except ValueError:
        print("Input harus berupa angka. Silakan coba lagi.")
print("Nomor telepon yang dimasukkan:", nomor_telepon)
daftarmenu()
totalharga = 0
listjmlpesan= []
wktestimasi = 0
print ("")
print ("Masukkan KODE berikut pada 'Kode Makanan dipesan' apabila selesai
        memilih menu:")
print ("Input (1) untuk DINE IN")
print ("Input (2) untuk TAKE AWAY")
print ("Input (3) untuk Keluar dari Program Kasir")
print ("")
pilihanmenu = input("Kode makanan dipesan: ")
while pilihanmenu != ("3"):
    match = 0
    for i in datamenu:
        if pilihanmenu == i[0]:
            match += 1
            while True :
                try:
                    jml_pesan = int(input("Jumlah dipesan: "))
                    break
                except ValueError:
                    print ("Input salah. Masukkan jumlah pesanan")
            hargamenu = int(i[2])*jml_pesan
            totalharga += hargamenu
            print (i[1],"\t Rp",hargamenu)
            entry = [i[1], jml_pesan,hargamenu]
            listjmlpesan.append(entry)
            print (listjmlpesan)
            estimasi = int(i[3])*jml_pesan
            wktestimasi += estimasi

#Unuk pilihan menu DINE IN
if pilihanmenu == "1":
    match += 1
    totalpajak = harga_pajak1(totalharga)
    while True:
        try:
            orang = int(input("Jumlah orang: "))
            break
        except ValueError:
```




```
        print("Input salah. Masukkan jumlah orang")
    tampilkan_nomor_meja_tersedia()
    pesan_nomor_meja()
    kode = str(input("KODE KUPON (input 0 jika tidak ada): "))
    harga_akhir = kodekupon(kode, totalpajak)
    tanggal = datetime.date.today()
    tanggal_pesan = tanggal.strftime("%d/%m/%Y")
    nama_file = "data_struk.txt"
    with open(nama_file, 'a') as file:
        file.write("\n=====\\n")
        file.write("===== N O T A =====\\n")
        file.write("===== C A C A - R E S T O =====\\n")
        file.write("=====\\n")
        file.write("Tanggal      : {}\\n".format(tanggal_pesan))
        file.write("Nama         : {}\\n".format(nama))
        file.write("No WA        : +62 {}\\n".format(nomor_telepon))
        file.write("Pesanan      : \\n".format(len(listjmlpesan)))
        for pesanan in listjmlpesan:
            file.write(str(pesanan) + "\\n")
        file.write("Nomor Meja   : {}\\n".format(nomor_meja))
        file.write("Jumlah orang : {}\\n".format(orang))
        file.write("Harga menu   : Rp {}\\n".format(totalharga))
        file.write("Pajak        : 10%\\n")
        file.write("Kupon        : Rp {}\\n".format(kupon))
        file.write("Total        : Rp {}\\n".format(harga_akhir))
        file.write("===== Selamat Datang Kembali =====\\n")
        file.write("=====\\n")
    print("\\n=====")
    print("===== N O T A =====")
    print("===== C A C A - R E S T O =====")
    print("=====")
    print("Nama         :", nama)
    print("No WA        : +62", nomor_telepon)
    print("Pesanan      : ".format(len(listjmlpesan)))
    for pesanan in listjmlpesan:
        print (pesanan)
    print("Nomor Meja   :", nomor_meja)
    print("Jumlah orang :", orang)
    print("Harga menu   : Rp", totalharga)
    print("Pajak        : 10%")
    print("Kupon        : Rp", kupon)
    print("Total        : Rp", harga_akhir)
    print("===== Selamat Datang Kembali =====")
    print("=====")
    print("")
    print("Silahkan pilih Metode Pembayaran")
```



```
print("[1] Cash")
print("[2] Transfer >> BCA : 123456789")
print("Dana/Spay/Gopay : 081328023385")
choice = str(input(">> "))
if choice == "1":
    print("-----")
    print("Mohon untuk membayar dengan uang pas")
    print("-----")
    exit()
elif choice == "2":
    norek_resto = "123456789"
    no_telepon = "081328023385"
    totalbayar = harga_akhir
    if metode_transfer(norek_resto, no_telepon, totalbayar):
        print("-----")
        print(" Transaksi berhasil ")
        print("-----")
        exit()
    else:
        print("-----")
        print(" Transaksi gagal ")
        print("-----")

#Untuk pilihan menu TAKE AWAY
elif pilihanmenu == "2":
    match += 1
    print("""Silahkan Pilih Menu Berikut
    Pilih [1] untuk DELIVERY (Only Area Solo)
    Pilih [2] untuk NON DELIVERY""")
    while True:
        try:
            opsi = int(input("[1] atau [2]: "))
            break
        except ValueError:
            print('Input salah')

#Opsii DELIVERY
if opsi == 1:
    alamat = str(input('Input alamat lengkap: '))
    kecamatan = str(input("input kecamatan: "))
    biaya_total = hitung_ongkir_menu(kecamatan, totalharga)
    totalpajak = harga_pajak2(biaya_total)
    kode = str(input("KODE KUPON (input 0 jika tidak ada): "))
    harga_akhir = kodekupon(kode, totalpajak)
    tanggal = datetime.date.today()
    tanggal_pesan = tanggal.strftime("%d/%m/%Y")
```



```
nama_file = "data_struk.txt"
with open(nama_file, 'a') as file:
    file.write("\n===== \n")
    file.write("===== N O T A ===== \n")
    file.write("===== C A C A - R E S T O ===== \n")
    file.write("===== \n")
    file.write("Tanggal      : {} \n".format(tanggal_pesanan))
    file.write("Nama         : {} \n".format(nama))
    file.write("No WA        : +62 {} \n".format(nomor_telepon))
    file.write("Alamat       : {} \n".format(alamat))
    file.write("Pesanan      : \n".format(len(listjmlpesanan)))
    for pesanan in listjmlpesanan:
        file.write(str(pesanan))
    file.write("\nHarga Menu   : Rp {} \n".format(totalharga))
    file.write("Ongkir        : Rp {} \n".format(ongkir))
    file.write("Pajak         : 10% \n")
    file.write("Kupon         : Rp {} \n".format(kupon))
    file.write("Total Harga   : Rp {} \n".format(harga_akhir))
    file.write("===== Selamat Datang Kembali ===== \n")
    file.write("===== \n")

print("\n===== ")
print("===== N O T A ===== ")
print("===== C A C A - R E S T O ===== ")
print("===== ")
print("Nama          :", nama)
print("No WA         : +62", nomor_telepon)
print("Alamat        : ", alamat)
print("Pesanan       : ".format(len(listjmlpesanan)))
for pesanan in listjmlpesanan:
    print (pesanan)
print("Harga Menu   : Rp", totalharga)
print("Ongkir        : Rp", ongkir)
print("Pajak         : 10%")
print("Kupon         : Rp", kupon)
print("Total Harga   : Rp", harga_akhir)
print("===== Selamat Datang Kembali =====")
print("===== ")
print("")
print("Silahkan pilih Metode Pembayaran")
print("[1] Cash      ")
print("[2] Transfer >> BCA          : 123456789")
print("                Dana/Spay/Gopay : 081328023385")
choice = str(input(">> "))
if choice == "1":
    print("-----")
    print("Mohon untuk membayar dengan uang pas")
```



```
print("-----")
exit()
elif choice == "2":
    norek_resto = "123456789"
    no_telepon = "081328023385"
    totalbayar = harga_akhir
    if metode_transfer(norek_resto, no_telepon, totalbayar):
        print("-----")
        print(" Transaksi berhasil ")
        print("-----")
        exit()
    else:
        print("-----")
        print(" Transaksi gagal ")
        print("-----")

#Opsi NON DELIVERY
elif opsi == 2:
    totalpajak = harga_pajak1(totalharga)
    kode = str(input("KODE KUPON (input 0 jika tidak ada): "))
    harga_akhir = kodekupon(kode, totalpajak)
    tanggal = datetime.date.today()
    tanggal_pesanan = tanggal.strftime("%d/%m/%Y")
    nama_file = "data_struk.txt"
    with open(nama_file, 'a') as file:
        file.write("\n===== \n")
        file.write("===== N O T A ===== \n")
        file.write("===== C A C A - R E S T O ===== \n")
        file.write("===== \n")
        file.write("Tanggal      : {} \n".format(tanggal_pesanan))
        file.write("Nama         : {} \n".format(nama))
        file.write("No WA        : +62 {} \n".format(nomor_telepon))
        file.write("Pesanan      : \n".format(len(listjmlpesan)))
        for pesanan in listjmlpesan:
            file.write(str(pesanan) + " \n")
        file.write("\nHarga Menu   : Rp {} \n".format(totalharga))
        file.write("pajak        : 10% \n")
        file.write("Kupon        : Rp {} \n".format(kupon))
        file.write("Total Harga   : Rp {} \n".format(harga_akhir))
        file.write("----- \n")
        file.write(f"ambil pesanan anda dalam {estimasi} menit \n")
        file.write("----- \n")
        file.write("===== Selamat Datang Kembali ===== \n")
        file.write("===== \n")
    print("\n===== ")
    print("===== N O T A ===== ")
```



```
print("===== C A C A - R E S T O =====")
print("=====")
print("Nama          :", nama)
print("No WA         : +62", nomor_telepon)
print("Pesanan        :".format(len(listjmlpesan)))
for pesanan in listjmlpesan:
    print (pesanan)
print("Harga Menu   : Rp", totalharga)
print("pajak         : 10%")
print("Kupon          : Rp", kupon)
print("Total Harga   : Rp", harga_akhir)
print("-----")
print(f"silahkan ambil pesanan anda dalam {wktestimasi} menit")
print("-----")
print("===== Selamat Datang Kembali =====")
print("=====")
print("")
print("Silahkan pilih Metode Pembayaran")
print("[1] Cash          ")
print("[2] Transfer >> BCA          : 123456789")
print("          Dana/Spay/Gopay : 081328023385")
choice = str(input(">> "))
if choice == "1":
    print("-----")
    print("Mohon untuk membayar dengan uang pas")
    print("-----")
    exit()
elif choice == "2":
    norek_resto = "123456789"
    no_telepon = "081328023385"
    totalbayar = harga_akhir
    if metode_transfer(norek_resto, no_telepon, totalbayar):
        print("-----")
        print(" Transaksi berhasil ")
        print("-----")
        exit()
    else:
        print("-----")
        print(" Transaksi gagal ")
        print("-----")
else:
    print("Input tidak valid")

if match == ("a"):
    print ("Input salah, ulangi input")
pilihanmenu = input("Kode makanan dipesan: ")
```



Fungsi program kasir berfungsi untuk *user* menginput makanan dan minuman yang dipilih, beserta dengan layanan yang diberikan hingga struk muncul. Program membaca data menu dari 'data_menu.csv'. Selanjutnya *user* diminta untuk menginput nama dan nomor telepon. Program memanggil fungsi 'datamenu' untuk menampilkan daftar menu yang tersedia. Mendeklarasikan variable 'totalharga' untuk menghitung total harga pesanan dan variable 'listjmlpesan' untuk menyimpan data pesanan *user* dalam *list*.

Pilihan [1] DINE IN, program meminta *user* untuk memasukkan jumlah orang dan program mengimport fungsi 'tampilkan_nomor_meja_tersedia' untuk menampilkan nomor meja yang tersedia. Program mengimport fungsi 'pesan_nomor_meja' untuk menginput nomor meja yang tersedia dan memasukkan kode kupon jika ada. Program mengimport modul *datetime* untuk menampilkan tanggal hari ini dan fungsi 'harga_pajak' untuk menghitung pajak. Harga akhir dari pesanan dihitung dari total pesanan, pajak, kupon (jika ada). Program akan menampilkan struk pesanan yang berisi tanggal pemesanan, nama, nomor telepon, *list* pesanan, nomor meja, jumlah orang, harga menu, pajak, kupon, dan harga akhir. Selanjutnya struk akan disimpan dalam 'data_struk.txt'.

Pilihan [2] TAKE AWAY, program ini terdapat dua pilihan menu yaitu *Delivery* dan *Non Delivery*. Ketika *user* memilih *Delivery*, maka program akan meminta *user* untuk menginput alamat lengkap. Program mengimport fungsi 'hitung_ongkir_menu' dengan menginput kecamatan untuk menentukan harga ongkir. Program mengimport modul *datetime* untuk menampilkan tanggal hari ini dan fungsi 'harga_pajak' untuk menghitung pajak. Harga akhir dari pesanan dihitung dari total pesanan, ongkir, pajak, kupon (jika ada). Program akan menampilkan struk pesanan yang berisi tanggal pemesanan, nama, nomor telepon, alamat, *list* pesanan, harga menu, ongkir, pajak, kupon, dan harga akhir.

Ketika *user* memilih program *Non Delivery*, maka *user* tidak perlu untuk menginput alamat. Program mengimport modul *datetime* untuk menampilkan tanggal hari ini dan fungsi 'harga_pajak' untuk menghitung pajak. Harga akhir dari pesanan dihitung dari total pesanan, pajak, kupon (jika ada). Program akan



menampilkan struk pesanan yang berisi tanggal pemesanan, nama, nomor telepon, list pesanan, harga menu, pajak, kupon, harga akhir, dan estimasi pesanan. Estimasi pesanan berguna untuk memberikan *user* informasi mengenai waktu makanan dan minuman dibuat, program mengambil data estimasi dari csv 'daftar_menu.csv'.

Pilihan [3] KELUAR merupakan pilihan untuk keluar dari program kasir. *Output* dari pilihan *Dine In*, *Delivery*, dan *Non-Delivery* adalah sebuah struk. Setelah program menampilkan struk, maka *user* diminta untuk melakukan pembayaran. Program memberikan pilihan kepada *user* untuk memilih pembayaran *cash* atau *transfer*. Apabila *user* memilih metode *transfer*, *user* harus memasukkan nomor rekening restoran dan memasukkan total harga yang harus dibayarkan. Program akan menampilkan "Transaksi Berhasil" dan mengakhiri program.

3.8. Fungsi Main Program

```
import modulfood as mf
while True:
    print("=====")
    print("===== SELAMAT DATANG =====")
    print("===== CACA RESTO =====")
    print("=====")
    print("=====Jl. Utara Pasar Gedhe No.34, Sudioprajan,=====")
    print("=====Kec. Jebres, Kota Surakarta, Jawa Tengah=====")
    print("=====")
    print("=====SILAKAN PILIH MENU BERIKUT !=====")
    print("")
    print("                Klik (1) Program Kasir                ")
    print("                Klik (2) Tampilkan Menu                ")
    print("                Klik (3) Keluar Program                ")
    mulai = input("> ")
    if mulai == "1":
        print("=====")
        print("===== CACA RESTO =====")
        print("=====")
        print("")
        mf.programkasir()

    elif mulai == "2":
        print("=====")
        print("===== CACA RESTO =====")
        print("=====")
```



```
mf.daftarmenu()
elif mulai == ("3"):
    print ("")
    print ("PROGRAM TERHENTI!")
    print ("")
    break
else:
    print ("")
    print ("Input SALAH!")
    print ("PROGRAM TERHENTI!")
    print ("")
    break
```

Program di atas merupakan sebuah program yang menyajikan tampilan menu dan pilihan aksi untuk restoran fiktif bernama "CACA RESTO". Program ini menggunakan modul bernama "modulfood" yang telah diimpor sebagai mf. Pada awal program, terdapat sebuah perulangan *while True* yang berjalan secara terus-menerus sehingga menu akan terus ditampilkan hingga program dihentikan. Setiap iterasi perulangan, program akan mencetak tampilan menu dan pilihan aksi restoran "CACA RESTO" yang terdiri dari Program Kasir (input "1"), Tampilkan Menu (input "2"), dan Keluar Program (input "3").

Jika pengguna memilih "1", program akan mencetak tampilan "CACA RESTO" dan memanggil fungsi programkasir() dari modul modulfood untuk menjalankan program kasir. Jika pengguna memilih "2", program akan mencetak tampilan "CACA RESTO" dan memanggil fungsi daftarmenu() dari modul modulfood untuk menampilkan menu makanan. Jika pengguna memilih "3", program akan mencetak pesan "PROGRAM TERHENTI!" dan menghentikan perulangan menggunakan pernyataan *break*.

Jika pengguna memasukkan input selain "1", "2", atau "3", program akan mencetak pesan "Input SALAH!" dan "PROGRAM TERHENTI!", kemudian menghentikan perulangan. Secara keseluruhan, program ini menyediakan antarmuka untuk memilih antara menjalankan program kasir, menampilkan daftar menu, atau keluar dari program. Pilihan pengguna akan menentukan tindakan selanjutnya yang akan dijalankan.



BAB IV

HASIL RUNNING PYTHON

4.1. Output Daftar Menu

```
=====
===== CACA RESTO =====
=====
```

Kode	Menu	Harga
101	Nasi Ayam Dada	20000
102	Nasi Ayam Paha Atas	20000
103	Nasi Ayam Paha Bawah	15000
104	Nasi Ayam Sayap	15000
105	Burger	23000
201	Es Teh	3000
202	Air Mineral	2000
203	Cola	7000
204	Es Jeruk	3000
205	Es Milo	7000

```
=====
```



4.2. Output Dine In

```
=====
===== N O T A =====
===== C A C A - R E S T O =====
=====
Nama           : Nona Tua
No WA          : +62 84126482917
Pesanan        :
['Nasi Ayam Dada', 1, 20000]
['Nasi Ayam Paha Atas', 2, 40000]
['Es Teh', 3, 9000]
Nomor Meja     : 3
Jumlah orang   : 3
Harga menu     : Rp 69000
Pajak          : 10%
Kupon          : Rp 7000
Total          : Rp 68900.0
===== Selamat Datang Kembali =====
=====
```

4.3. Output Delivery

```
=====
===== N O T A =====
===== C A C A - R E S T O =====
=====
Nama           : Chelsea
No WA          : +62 81725290476
Alamat         : Jebres, Surakarta
Pesanan        :
['Nasi Ayam Paha Bawah', 1, 15000]
['Nasi Ayam Sayap', 1, 15000]
['Cola', 2, 14000]
Harga Menu     : Rp 44000
Ongkir         : Rp 5000
Pajak          : 10%
Kupon          : Rp 5000
Total Harga    : Rp 48900.0
===== Selamat Datang Kembali=====
=====
```



4.4. Output Non Delivery

```
=====
===== N O T A =====
===== C A C A - R E S T O =====
=====
Nama          : Anton
No WA         : +62 81251257920
Pesanan       :
['Burger', 2, 46000]
Harga Menu    : Rp 46000
pajak         : 10%
Kupon         : Rp 7000
Total Harga   : Rp 43600.0
-----
silahkan ambil pesanan anda dalam 14 menit
-----
===== Selamat Datang Kembali =====
=====
```