TASK 1 ALGORITHMS AND PROGAMMING
AUTOMATIC PIZZA ORDERING PROGRAM WITH PYTHON

Group name:

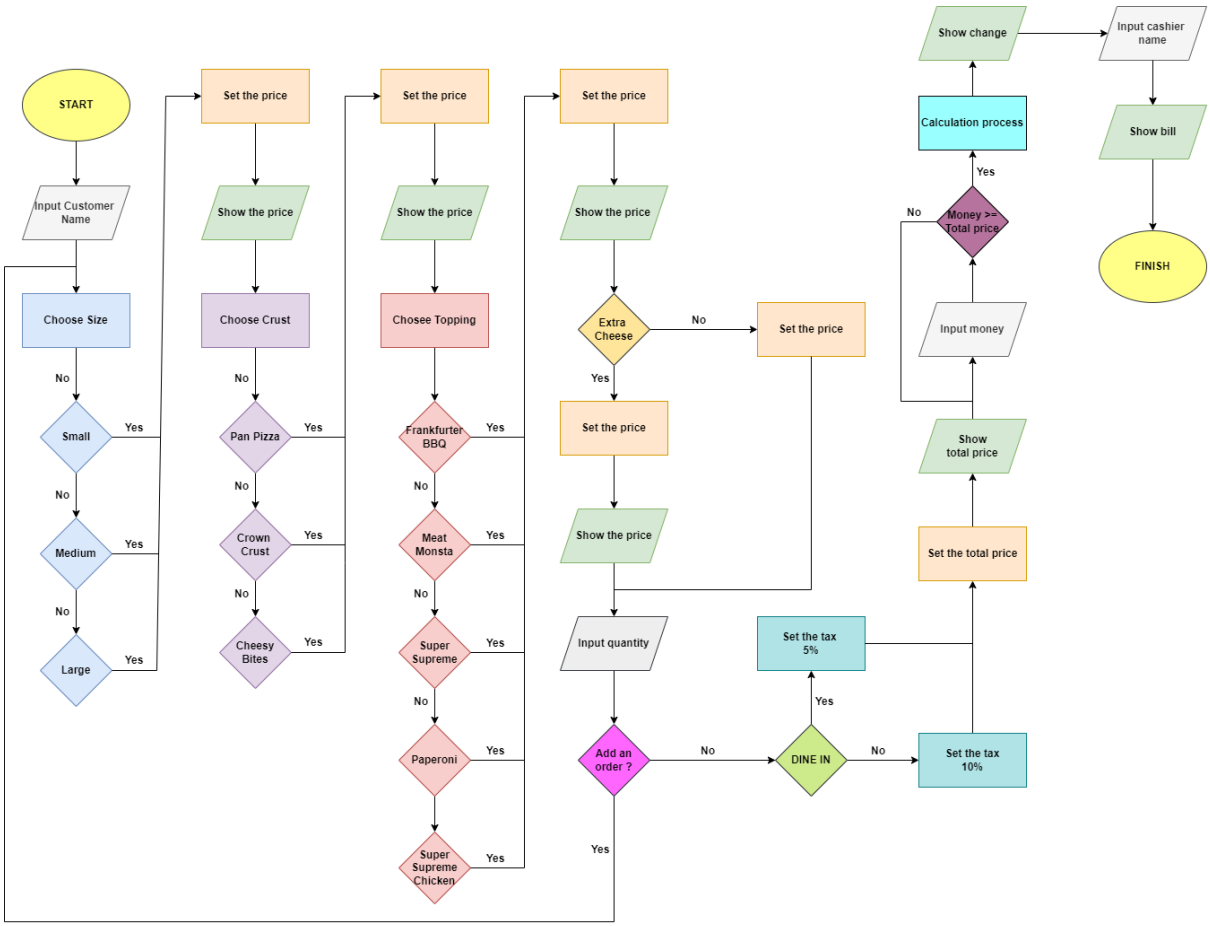
Fitrya Chalifatus Zahro - 24091397117

Fikro Nabila - 24091397110

INFORMATICS MANAGEMENT STUDY PROGRAM
FACULTY OF VOCATIONAL
STATE UNIVERSITY OF SURABAYA

2024

A. Flowchart



Gambar 1. Flowchart

- **Flowchart Analysis:**
    1. The process begins by inputting the customer's name to record the order.
    2. Pizza size selection: The customer chooses one of the three pizza sizes: Small, Medium, or Large.
    3. The system displays the price according to the selected size.
    4. Crust selection: The customer selects one of the three available crust types: Pan Pizza, Crown Crust, or Cheesy Bites.
    5. The system displays the price according to the selected crust type : Frankfurter BBQ, Meat Monsta, Super Supreme, Paperoni, or Super Supreme Chicken
    6. Topping selection: The customer chooses one of the five available topping options.
    7. The system displays the price according to the selected topping.
    8. Extra cheese option: The customer can choose whether or not to add extra cheese. If yes, the price for the extra cheese is added to the total.

9. Input order quantity: If the customer wants to add another order, the process will return to the pizza selection step. If not, the process moves to the next step.
10. Dine-in selection: If the customer chooses to dine in, the system adds a 5% tax.
11. Takeaway selection: If the customer chooses takeaway, the tax added is 10%.
12. Total price determination: After all selections are made, the system calculates and displays the total price, including the tax.
13. Payment: The customer inputs the money to pay for the order. If the amount given is equal to or greater than the total price, the process continues to the change calculation step.
14. If the money provided is insufficient, the customer must enter more money until the total is met.
15. Change calculation: If the amount given is more than the total price, the system calculates and displays the change.
16. Cashier name input: The cashier's name is input into the system, and the system will display the receipt.
17. Finish: After all processes are completed, the transaction is closed, and the order is ready to be served or taken away.

B. Snippets Code

1. Input

```python
import datetime
welcome = print(f"{"*" * 38}\n{"==== WELCOME TO D'PIZZA NYELL ====":^38}\n{"*" * 38}")
client_name = input("What's your name?")
print(f"Hallo {client_name}! Please input your order:")

price = 0;
while True:
    size = input(f"{"Size Pizza:"}\n{"S = Small":<10} {"[10000]":>23}\n{"M = Medium":<10}{"[15000]":>24}\n{"L = Large":<10}{"[20000]":>24}\nChoose Size: ").upper()
    if size == "S":
        price += 10000
        fix = print(f"{"Small Size":<32}[{price}]")
        size_name = f"{"Small Size":<30}{10000}"
        break
    elif size == "M":
        price += 15000
        fix = print(f"{"Medium Size":<32}[{price}]")
        size_name = f"{"Medium Size":<30}{15000}"
        break
    elif size == "L":
        price += 20000
        fix = print(f"{"Large Size":<32}[{price}]")
        size_name = f"{"Large Size":<30}{20000}"
        break
    else:
        print("Invalid order. Repeat your order, please!")

while True:
    print(f"{"Crust Type:"}\n{"1 = Pan Pizza":<10} {"[20000]":>20}\n{"2 = Crown Crust":<10}{"[25000]":>19}\n{"3 = Cheesy Bites":<10}{"[30000]":>18}")
    crust = int(input("Choose Crust: "))
    if crust == 1:
        price += 20000
        fix = print(f"{"Pan Pizza":<32}[{price}]")
        crust_name = f"{"Pan Pizza":<30}{20000}"
        break
    elif crust == 2:
        price += 25000
        fix = print(f"{"Crown Crust":<32}[{price}]")
        crust_name = f"{"Crown Crust":<30}{25000}"
        break
    elif crust == 3:
        price += 30000
        fix = print(f"{"Cheesy Bites":<32}[{price}]")
        crust_name = f"{"Cheesy Bites":<30}{30000}"
        break
    else:
        print("Invalid order. Repeat your order, please!")

```

```
new_order = input("Add an orders?(yes/no): ").lower()
if new_order == "yes":
    price1 = 0;
    quantity_price1 = 0;
    while True:
        print("Please input your order:")
        size1 = input(f"{"Size Pizza:"}\n{"S = Small":<10} {"[10000]":>23}\n{"M = Medium":<10}{"[15000]":>24}\n{"L = Large":<10}{"[20000]":>24}\nChoose Size: ").upper()
        if size1 == "S":
            price1 += 10000
            fix1 = print(f"{"Small Size":<32}[{price1}]")
            size_name1 = f"{"Small Size":<30}{10000}"
            break
        elif size1 == "M":
            price1 += 15000
            fix1 = print(f"{"Medium Size":<32}[{price1}]")
            size_name1 = f"{"Medium Size":<30}{15000}"
            break
        elif size1 == "L":
            price1 += 20000
            fix1 = print(f"{"Large Size":<32}[{price1}]")
            size_name1 = f"{"Large Size":<30}{20000}"
            break
        else:
            print("Invalid order. Repeat your order, please!")

    while True:
        print(f"{"Crust Type:"}\n{"1 = Pan Pizza":<10} {"[20000]":>20}\n{"2 = Crown Crust":<10}{"[25000]":>19}\n{"3 = Cheesy Bites":<10}{"[30000]":>18}")
        crust1 = int(input("Choose Crust: "))
        if crust1 == 1:
            price1 += 20000
            fix1 = print(f"{"Pan Pizza":<32}[{price1}]")
            crust_name1 = f"{"Pan Pizza":<30}{20000}"
            break
        elif crust1 == 2:
            price1 += 25000
            fix1 = print(f"{"Crown Crust":<32}[{price1}]")
            crust_name1 = f"{"Crown Crust":<30}{25000}"
            break
        elif crust1 == 3:
            price1 += 30000
            fix1 = print(f"{"Cheesy Bites":<32}[{price1}]")
            crust_name1 = f"{"Cheesy Bites":<30}{30000}"
            break
        else:
            print("Invalid order. Repeat your order, please!")

```

```
    while True:
        topping1 = input(f"{"Variant Toppings:"}\n{"F = Frankfurter BBQ":<18} {"[28000]":>14}\n{"M = Meat Monsta":<18}{"[35000]":>19}\n{"S = Super Supreme":<18}{"[42000]":>17}\n{"P = Paperoni":<18}{"[49000]":>22}\n{"C = Super Supreme Chicken":<18}{"[56000]":>9}\nChoose Topping: ").upper()
        if topping1 == "F":
            price1 += 28000
            fix1 = print(f"{"Frankfurter BBQ":<32}[{price1}]")
            topping_name1 = f"{"Frankfurter BBQ":<30}{28000}"
            break
        elif topping1 == "M":
            price1 += 35000
            fix1 = print(f"{"Meat Monsta":<32}[{price1}]")
            topping_name1 = f"{"Meat Monsta":<30}{35000}"
            break
        elif topping1 == "S":
            price1 += 42000
            fix1 = print(f"{"Super Supreme":<32}[{price1}]")
            topping_name1 = f"{"Super Supreme":<30}{42000}"
            break
        elif topping1 == "P":
            price1 += 49000
            fix1 = print(f"{"Paperoni":<32}[{price1}]")
            topping_name1 = f"{"Paperoni":<30}{49000}"
            break
        elif topping1 == "C":
            price1 += 56000
            fix1 = print(f"{"Super Supreme Chicken":<32}[{price1}]")
            topping_name1 = f"{"Super Supreme Chicken":<30}{56000}"
            break
        else:
            print("Invalid order. Repeat your order, please!")

    cheese1 = input("Extra Cheese?(yes/no): ")
    if cheese1.lower() == "yes":
        price1 += 13000
        fix1 = print(f"{"Extra Cheese":<32}[{price1}]")
        cheese_name1 = f"{"Extra Cheese":<30}{13000}"
    else:
        price1 += 0
        cheese_name1 = "No Extra Cheese"

    quantity1 = int(input("Quantity: "))
    quantity_price1 = 0;
    if quantity1 >= 1:
        quantity_price1 = price1*quantity1
        print(quantity1)
    else:
        print(quantity1)
else:
    quantity_price1 = 0

quantity_price2 = quantity_price+quantity_price1
dine_status = "";
price_status = 0;
dine_in = input("Dine in?(yes/no): ").lower()
if dine_in == "yes":
    price_status = quantity_price2 * (5 / 100)
    dine_status = "<< Dine in >>"
else:
    price_status = quantity_price2 * (10 / 100)
    dine_status = "<< Take Away >>"

```

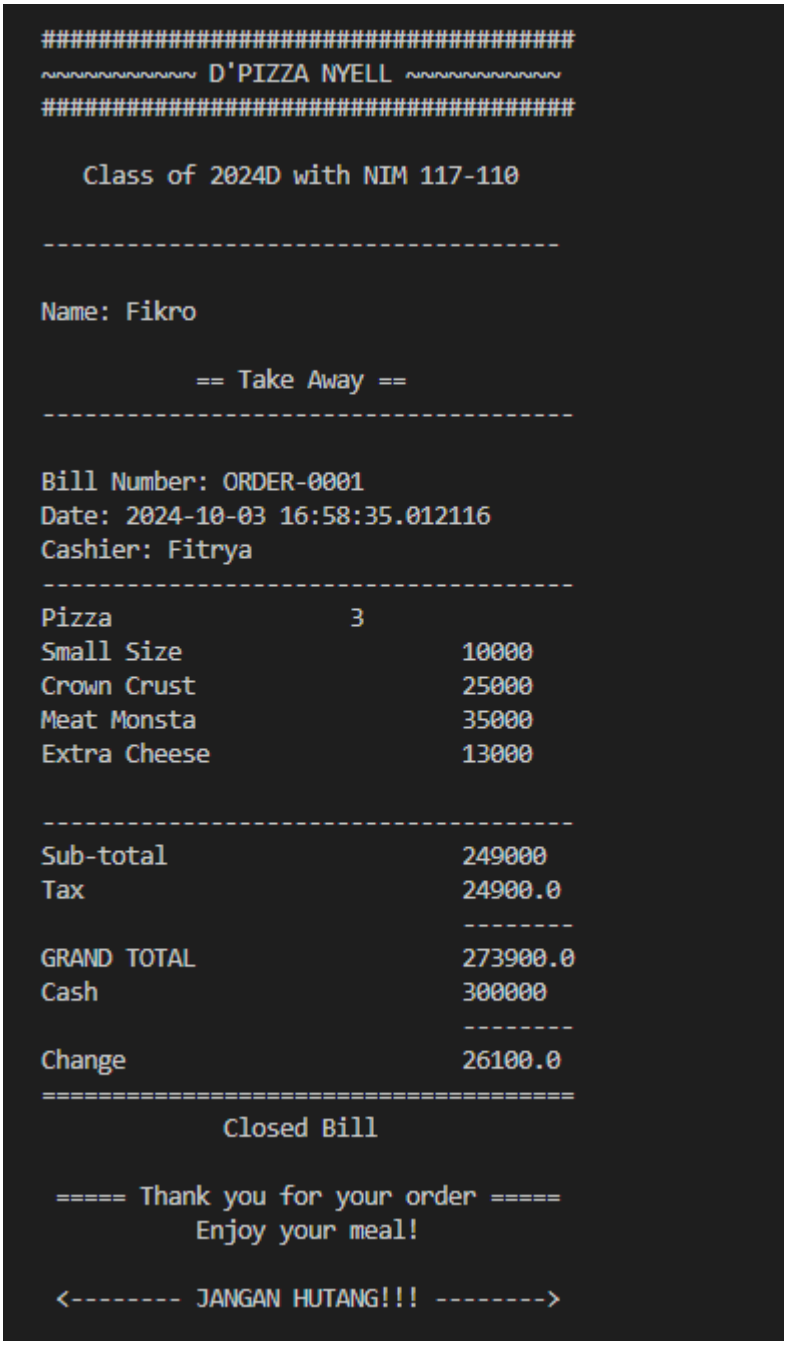
```
def order_numb():
    order_count = 1
    order_number = f"ORDER-{order_count:04d}"
    order_count += 1
    return order_number

ordernumbdef = order_numb()

def bill_number():
    order_num = ordernumbdef
    print(f"Bill Number: {order_num}")

def prod_time():
    prod = datetime.datetime.now()
    print(f"Date: {prod}")

grandtotal = quantity_price2+price_status

print(f"{"Sub-total":<32}{quantity_price2}\n{"Tax":<32}{price_status}\n{"Grand-total":<32}{grandtotal}")

while True:
    cash = int(input("Money: "))
    change = 0;
    if cash >= grandtotal:
        change = cash-grandtotal
        break
    else:
        print("Not enough payments")

print(f"{"Change:":<31} {change}")
cashier = input("Cashier name? ")

#Bill Order
def bill():
    print(f"\n{"#" * 38}\n{"~~~~~~~~~~ D'PIZZA NYELL ~~~~~~~~~~":^38}\n{"#" * 38}\n ")
    print(f"{"Class of 2024D with NIM 117-110":^38}\n \n-------------------------------------\n ")
    print(f"Name: {client_name}\n ")
    print(f"{dine_status:^38}\n{"-" * 38}\n ")
    bill_number()
    prod_time()
    print(f"Cashier: {cashier}\n{"-" * 38}")
    print(f"Pizza{quantity:^35}")
    print(f"{size_name}\n{crust_name}\n{topping_name}\n{cheese_name}")
    if new_order == "yes":
        print(f"Pizza{quantity1:^35}")
        print(f"{size_name1}\n{crust_name1}\n{topping_name1}\n{cheese_name1}")
    print(f" \n{"-" * 38}\n{"Sub-total":<30}{quantity_price2}")
    print(f"{"Tax":<30}{price_status}\n{"-"*8:>38}")
    print(f"{"GRAND TOTAL":<30}{grandtotal}")
    print(f"{"Cash":<30}{cash}\n{"-"*8:>38}")
    print(f"{"Change":<30}{change}")
    print(f"{"=" * 38}\n{"Closed Bill":^38}\n \n{"===== Thank you for your order =====":^38}\n{"Enjoy your meal!":^38}\n \n{"<-------- JANGAN HUTANG!!! -------->":^38}\n ")

bill()
```

Picture 2. Source Code

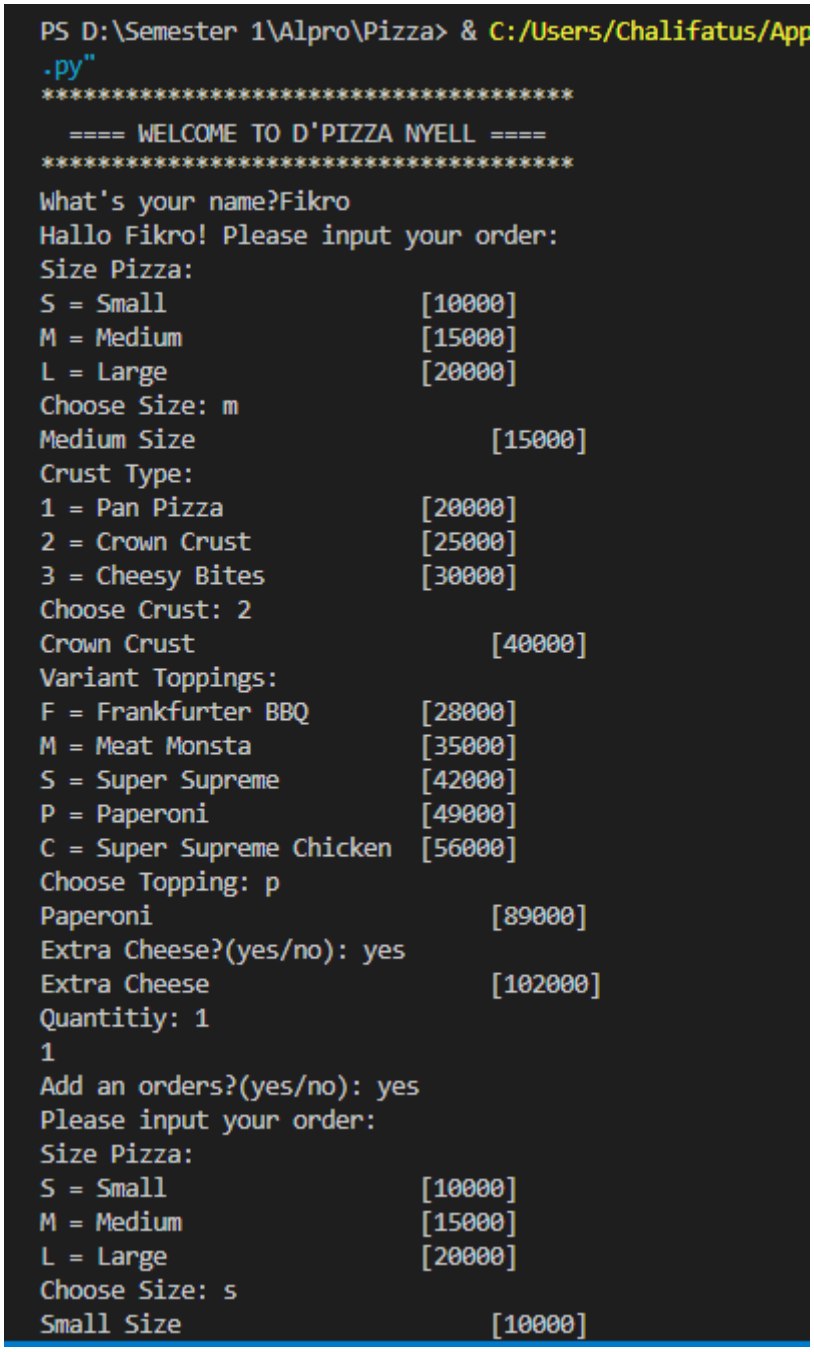
2. Output

```
PS D:\Semester 1\Alpro\Pizza> & C:/Users/Chalifatus/AppData/Local
.py"
****************************************
  ==== WELCOME TO D'PIZZA NYELL ====
****************************************
What's your name?Fikro
Hallo Fikro! Please input your order:
Size Pizza:
S = Small                   [10000]
M = Medium                  [15000]
L = Large                   [20000]
Choose Size: s
Small Size                       [10000]
Crust Type:
1 = Pan Pizza               [20000]
2 = Crown Crust             [25000]
3 = Cheesy Bites            [30000]
Choose Crust: 2
Crown Crust                      [35000]
Variant Toppings:
F = Frankfurter BBQ         [28000]
M = Meat Monsta             [35000]
S = Super Supreme           [42000]
P = Paperoni                [49000]
C = Super Supreme Chicken   [56000]
Choose Topping: m
Meat Monsta                      [70000]
Extra Cheese?(yes/no): yes
Extra Cheese                     [83000]
Quantitiy: 3
3
Add an orders?(yes/no): no
Dine in?(yes/no): no
Sub-total                        249000
Tax                              24900.0
Grand-total                      273900.0
Money: 300000
Change:                          26100.0
Cashier name? Fitrya
```

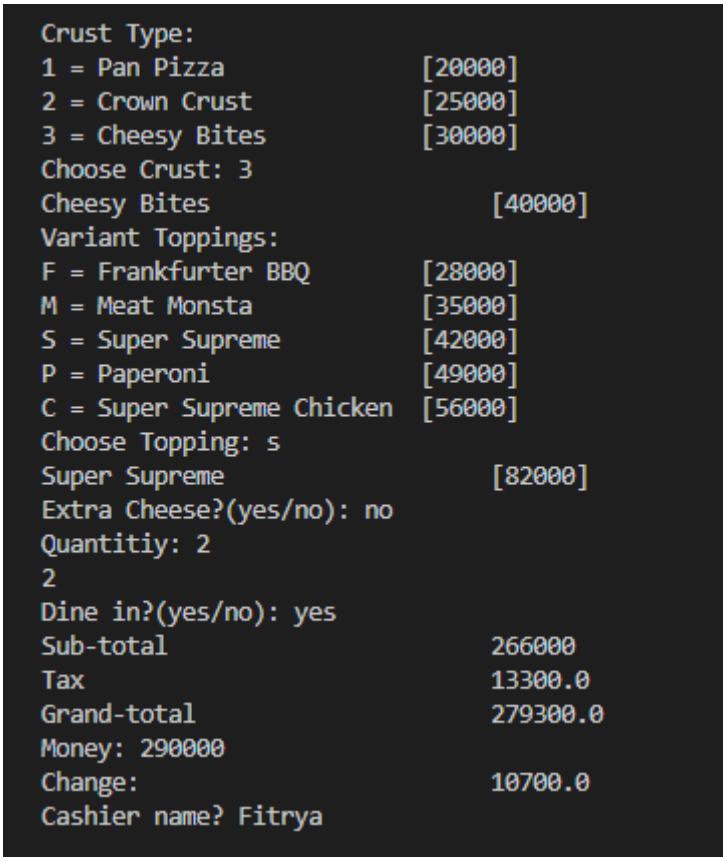
Picture 3. Pizza ordering view (1 order with extra cheese)



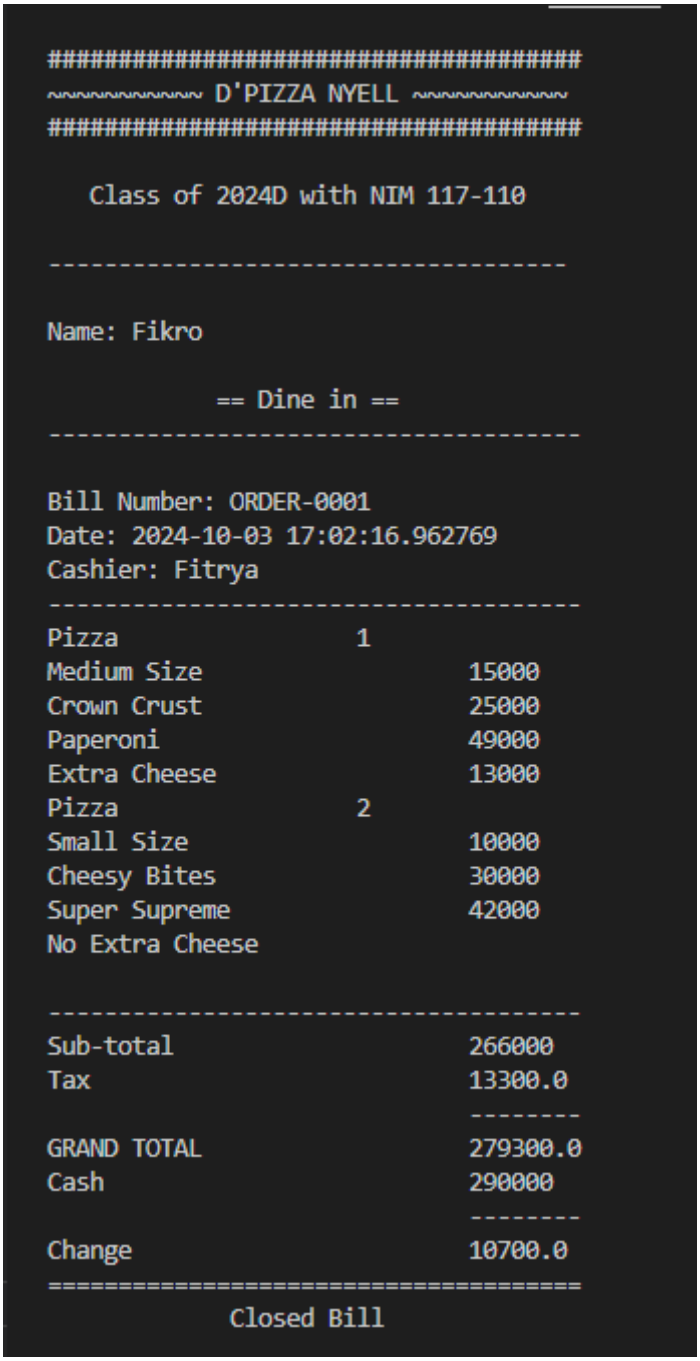
Picture 4. Bill of purchase view (1 order with extra cheese)



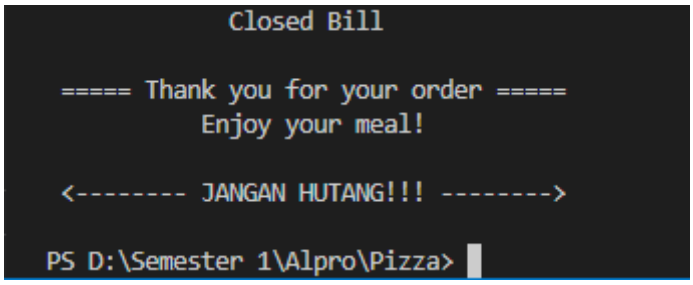
Picture 5. Pizza ordering view (2 order and no extra cheese)



Picture 6. Pizza ordering view (2 order and no extra cheese)



Picture 7. Bill of purchase view (2 order and no extra cheese)



Picture 8 . Final view of bill cover (2 order and no extra cheese)

C. Explanation of Source Code

1. Explanation of Each Code Section

```
pizza.py > ...
1   #import datetime is used to add a datetime module that functions to manipulate dates and times.
2   import datetime
```

Picture 9. Functions to manipulate dates and times

The above code is used to import the datetime module which allows us to manipulate dates and times. This module is useful for recording order times and creating receipts.

```
3   #The code below is used to display a welcome greeting, enter a name, and greet the customer.
4   welcome = print(f"{"*" * 38}\n{"==== WELCOME TO D'PIZZA NYELL ====":^38}\n{"*" * 38}")
5   client_name = input("What's your name?")
6   print(f"Hallo {client_name}! Please input your order:")
```

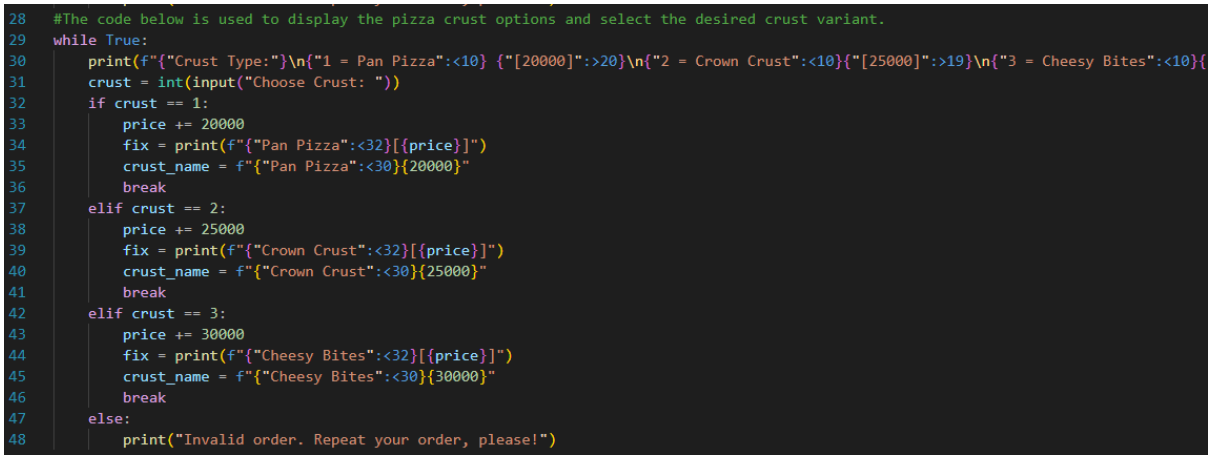
Picture 10. Welcome greeting, enter a name, and greet the customer

The code above displays a welcome to the customer and asks the customer to enter his name, then the customer will be greeted and asked to enter the order.

```
7   #The code below is used to display the size options and select the desired size.
8   price = 0;
9   while True:
0       size = input(f"{"Size Pizza:"}\n{"S = Small":<10} {"[10000]":>23}\n{"M = Medium":<10}{"[15000]":>24}\n{"L = Large":<10}{"[20000]":>
1       if size == "S":
2           price += 10000
3           fix = print(f"{"Small Size":<32}[{price}]")
4           size_name = f"{"Small Size":<30}{10000}"
5           break
6       elif size == "M":
7           price += 15000
8           fix = print(f"{"Medium Size":<32}[{price}]")
9           size_name = f"{"Medium Size":<30}{15000}"
0           break
1       elif size == "L":
2           price += 20000
3           fix = print(f"{"Large Size":<32}[{price}]")
4           size_name = f"{"Large Size":<30}{20000}"
5           break
6       else:
7           print("Invalid order. Repeat your order, please!")
```

Picture 11. Size options and desired size

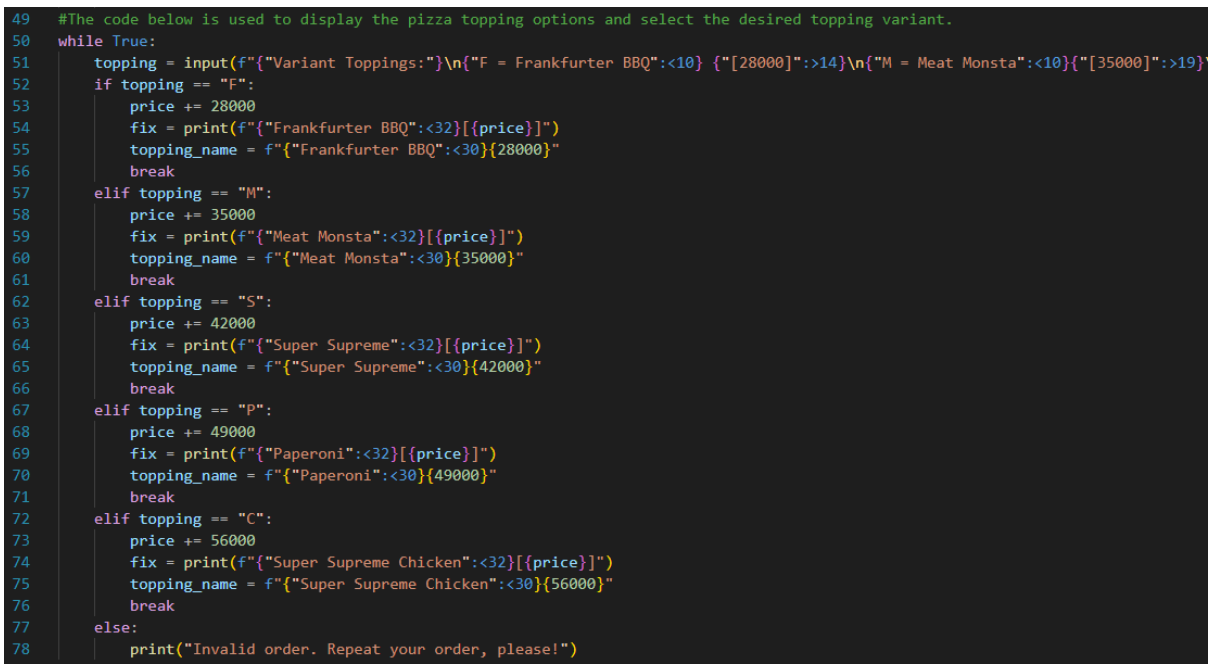
The code above displays the pizza size options along with the price and the customer is asked to enter the pizza size options using the codes already listed (S), (M), (L). If the customer does not enter the code as stated in the pizza size options, the customer will be asked to select the size again until the option entered is appropriate.



```
#The code below is used to display the pizza crust options and select the desired crust variant.
while True:
    print(f"{"Crust Type:"}\n{"1 = Pan Pizza":<10} {"[20000]":>20}\n{"2 = Crown Crust":<10}{"[25000]":>19}\n{"3 = Cheesy Bites":<10}{
    crust = int(input("Choose Crust: "))
    if crust == 1:
        price += 20000
        fix = print(f"{"Pan Pizza":<32}[{price}]")
        crust_name = f"{"Pan Pizza":<30}{20000}"
        break
    elif crust == 2:
        price += 25000
        fix = print(f"{"Crown Crust":<32}[{price}]")
        crust_name = f"{"Crown Crust":<30}{25000}"
        break
    elif crust == 3:
        price += 30000
        fix = print(f"{"Cheesy Bites":<32}[{price}]")
        crust_name = f"{"Cheesy Bites":<30}{30000}"
        break
    else:
        print("Invalid order. Repeat your order, please!")
```

Picture 12. Crust options and crust variant

The code above displays a selection of pizza crusts along with the price and the customer is asked to enter the choice of pizza crust using the code that has been listed (1), (2), (3). If the customer does not enter the code as listed on the pizza crust option, then the customer will be asked to select the pizza crust again until the option entered is appropriate.



```
#The code below is used to display the pizza topping options and select the desired topping variant.
while True:
    topping = input(f"{"Variant Toppings:"}\n{"F = Frankfurter BBQ":<10} {"[28000]":>14}\n{"M = Meat Monsta":<10}{"[35000]":>19}
    if topping == "F":
        price += 28000
        fix = print(f"{"Frankfurter BBQ":<32}[{price}]")
        topping_name = f"{"Frankfurter BBQ":<30}{28000}"
        break
    elif topping == "M":
        price += 35000
        fix = print(f"{"Meat Monsta":<32}[{price}]")
        topping_name = f"{"Meat Monsta":<30}{35000}"
        break
    elif topping == "S":
        price += 42000
        fix = print(f"{"Super Supreme":<32}[{price}]")
        topping_name = f"{"Super Supreme":<30}{42000}"
        break
    elif topping == "P":
        price += 49000
        fix = print(f"{"Paperoni":<32}[{price}]")
        topping_name = f"{"Paperoni":<30}{49000}"
        break
    elif topping == "C":
        price += 56000
        fix = print(f"{"Super Supreme Chicken":<32}[{price}]")
        topping_name = f"{"Super Supreme Chicken":<30}{56000}"
        break
    else:
        print("Invalid order. Repeat your order, please!")
```

Picture13. Pizza topping options

The code above displays a selection of pizza toppings along with the price and the customer is asked to enter the pizza crust topping using the code that has been listed (F), (M), (S), (P), (C). If the customer does not enter the code as stated in the pizza topping options, then the customer will be asked to select the pizza topping again until the option entered is appropriate.

```
#The code below is used to display a question to the customer whether or not to add extra cheese.
cheese = input("Extra Cheese?(yes/no): ")
if cheese.lower() == "yes":
    price += 13000
    fix = print(f"{"Extra Cheese":<32}[{price}]")
    cheese_name = f"{"Extra Cheese":<30}{13000}"
else:
    price += 0
    cheese_name = "No Extra Cheese"
```

Picture 14. Extra Cheese

In the code above, customers are asked to choose whether they want to add extra cheese or not with the “yes” or “no” option menu.

```
#The code below is used to display the order quantity fill.
quantity = int(input("Quantitiy: "))
quantity_price = 0;
if quantity >= 1:
    quantity_price = price*quantity
    print(quantity)
else:
    print(quantity)
```

Picture 15. Order quantity fill

The code above prompts the customer to enter the quantity or number of pizzas they want to order.

```
#The code below is used to display the question whether to add an order or not.
new_order = input("Add an orders?(yes/no): ").lower()
if new_order == "yes":
    price1 = 0;
    quantity_price1 = 0;
    while True:
        print("Please input your order:")
        size1 = input(f"{"Size Pizza:"}\n{"S = Small":<10} {"[10000]":>23}\n{"M = Medium":<10}{"[15000
        if size1 == "S":
            price1 += 10000
            fix1 = print(f"{"Small Size":<32}[{price1}]")
            size_name1 = f"{"Small Size":<30}{10000}"
            break
        elif size1 == "M":
            price1 += 15000
            fix1 = print(f"{"Medium Size":<32}[{price1}]")
            size_name1 = f"{"Medium Size":<30}{15000}"
            break
        elif size1 == "L":
            price1 += 20000
            fix1 = print(f"{"Large Size":<32}[{price1}]")
            size_name1 = f"{"Large Size":<30}{20000}"
            break
        else:
            print("Invalid order. Repeat your order, please!")
```

Picture 16. Question whether to add an order or not

The code above asks the customer whether or not to add the order with the options “yes” or “no”. If the customer adds an order then the customer will be directed to enter the order again starting from the size choice to the order quantity.

```
#The code below is used to display the question whether the order is dine-in or takeaway.
quantity_price2 = quantity_price+quantity_price1
dine_status = "";
price_status = 0;
dine_in = input("Dine in?(yes/no): ").lower()
if dine_in == "yes":
    price_status = quantity_price2 * (5 / 100)
    dine_status = "== Dine in =="
else:
    price_status = quantity_price2 * (10 / 100)
    dine_status = "== Take Away =="
```

Picture 17. Question is dine-in or takeaway

The code above displays a question to the customer whether the order should be eaten on the spot or taken away. If the customer chooses to eat in, a tax of 5% will be charged. If the customer chooses takeaway, the tax will be 10%.

```
#The code below is used to display the payment amount.
print(f"{"Sub-total":<32}{quantity_price2}\n{"Tax":<32}{price_status}\n{"Grand-total":<32}{grandtotal}")
```

Picture 18. The payment amount

The code above displays the sum of the order price, tax, and the overall total price.

```
#The code below is used to input payment and calculate change.
while True:
    cash = int(input("Money: "))
    change = 0;
    if cash >= grandtotal:
        change = cash-grandtotal
        break
    else:
        print("Not enough payments")
print(f"{"Change:":<31} {change}")
```

Picture 19. Payment and calculate change

The code above prompts the customer to enter the amount of money paid. Then the system will automatically calculate the amount of money paid and will be matched with the total price of the payment. If the amount of money is less than the total price to be paid, then the customer will be asked to make another payment until the payment price is met. Conversely, if the amount of money is more than the total price to be paid, the system will automatically calculate the change.

```
231   #The code below is used to enter the cashier's name.
232   cashier = input("Cashier name? ")
233
```

Picture 20. Cashier's name

The code above prompts the customer to enter the name of the staff manning the cash register.

```
202   #The code below is used to display the order number automatically.
203   def order_numb():
204       order_count = 1
205       order_number = f"ORDER-{order_count:04d}"
206       order_count += 1
207       return order_number
208
209   ordernumbdef = order_numb()
210   def bill_number():
211       order_num = ordernumbdef
212       print(f"Bill Number: {order_num}")
```

Picture 21. Order number automatically

The above code is used to automatically add the order number to the bill.

```
213   #The code below is used to display the order date and time automatically.
214   def prod_time():
215       prod = datetime.datetime.now()
216       print(f"Date: {prod}")
```

Picture 22. Order date and time automatically

The code above is used to automatically add the order date and time to the bill.

```
234   #The code below is used to display the order bill.
235   def bill():
236       print(f"\n{"#" * 38}\n{"~~~~~~~~~~~~ D'PIZZA NYELL ~~~~~~~~~~~~":^38}\n{"#" * 38}\n ")
237       print(f"{"Class of 2024D with NIM 117-110":^38}\n \n-----------------------------------\n ")
238       print(f"Name: {client_name}\n ")
239       print(f"{dine_status:^38}\n{"-" * 38}\n ")
240       bill_number()
241       prod_time()
242       print(f"Cashier: {cashier}\n{"-" * 38}")
243       print(f"Pizza{quantity:^35}")
244       print(f"{size_name}\n{crust_name}\n{topping_name}\n{cheese_name}")
245       if new_order == "yes":
246           print(f"Pizza{quantity1:^35}")
247           print(f"{size_name1}\n{crust_name1}\n{topping_name1}\n{cheese_name1}")
248       print(f" \n{"-" * 38}\n{"Sub-total":<30}{quantity_price2}")
249       print(f"{"Tax":<30}{price_status}\n{"-"*8:>38}")
250       print(f"{"GRAND TOTAL":<30}{grandtotal}")
251       print(f"{"Cash":<30}{cash}\n{"-"*8:>38}")
252       print(f"{"Change":<30}{change}")
253       print(f"{"=" * 38}\n{"Closed Bill":^38}\n \n{"===== Thank you for your order =====":^38}\n{"Enjoy your meal!":^38}\n \n{"<-----
254
255   bill()
```

Picture 23. Order bill

The code above is used to display a bill that contains the customer's name, order number, order date and time, cashier staff name, order details, price and payment details.

2. Explanation of Syntax

```
import datetime
```

Picture 24. Import

The **import** function is used to import a module or package into a code script that allows the programmer to use functions, classes, or variables defined in the imported module.

```
print(f"{"Small Size":<32}[{price}]")
```

Picture 25. Print and f-string

The **print()** function is used to display the result of the code or output to the terminal. **f-string** is used to insert an expression or variable into a string in an easier and cleaner way and is marked with the letter f or F before the string quotes. **:<32** is used to set text alignment in an output. Usually used in string formats, especially f-string.

```
int(input("Choose Crust: "))
```

Picture 26. Integer and input

The **input()** function is used to take input from the user via the console. The **int()** function is used to convert a value or data into an integer data type.

```
9   while True:
```

Picture 27. While true

while True: is used to create a loop that will continue to run without stopping, until the loop is explicitly stopped with a **break** command or through certain conditioning that causes the loop to no longer execute.

```
81  if cheese.lower() == "yes":
82      price += 13000
83      fix = print(f"{"Extra Cheese":<32}[{price}]")
84      cheese_name = f"{"Extra Cheese":<30}{13000}"
85  else:
86      price += 0
87      cheese_name = "No Extra Cheese"
```

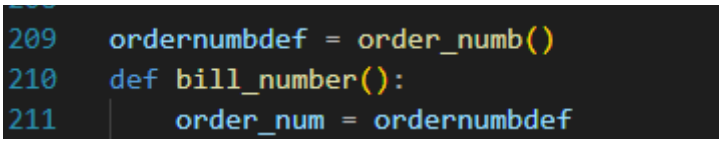
Picture 28. if and else

if else is used to perform decision making based on a given condition. It allows the program to execute a specific block of code if the specified condition is **True**, and a different block of code if the condition is **False**. **.lower()** digunakan untuk mengubah semua karakter dalam sebuah

string menjadi huruf kecil (lowercase).Ini berguna untuk membuat perbandingan string menjadi tidak sensitif terhadap huruf besar dan kecil.

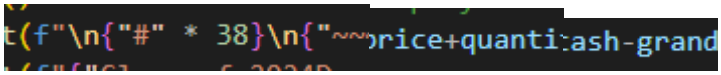
Picture 29. upper

.upper() is used to convert all letters in a string to uppercase. This is useful when you want to ensure that the text inside a string is displayed in uppercase format, regardless of how the string was originally written.



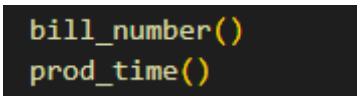
Picture 30. Ordernumbdef

ordernumbdef is a variable, variables are used to store data that can be used and manipulated in the program. **def** is used to define functions, which allows the organization of code and the invocation of repeated blocks of code kali untuk tugas yang sama.



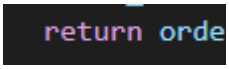
Picture 31. Code +-*

code +-* is used to perform arithmetic operations that add, subtract, or shift a value.



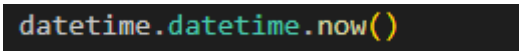
Picture 32. Bill_number

bill_number() is the name of a function that is used to call or access the function in the program.



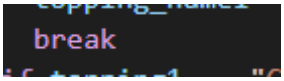
Picture 33. Return

return is used in functions to return the value of the function to its caller. When the return is executed, the function will stop, and the value specified after the return will be sent back.



Picture 34. Datetime.datetime.now()

datetime.datetime.now() is used to get the current date and time inside a Python program. This function returns a datetime object that contains information about the year, month, day, hour, minute, second, and microsecond.



Picture 35. Break

break is used in Python to stop execution of a loop (such as for or while) immediately. When break is executed, the program exits the loop immediately, and resumes execution on the next line after the loop.

D. Conclusion

The above program is a program that we created for automatic pizza ordering that allows customers to order pizza independently or commonly called self service. The customer is asked to enter a name, choose a pizza size, choose a pizza crust, choose pizza toppings, extra cheese, order quantity, and choose between dine-in or take-out. After that, the program calculates the total price, including tax, and requests payment. Information about the order, including the order number, date and time, and transaction details, is displayed in the form of a receipt. The program also handles additional orders if the user wants to order more than one pizza, and finally displays the full receipt after payment.

Some of the main features of the program we created are:

1. Selection of pizza size (Small, Medium, Large).
2. Selection of pizza crust type (Pan Pizza, Crown Crust, Cheesy Bites).
3. Selection of pizza toppings with various options.
4. Option to add extra cheese.
5. Calculation of total price and tax based on the selections or orders entered.

6. Clear and informative order receipt generation.
7. The program also provides an interactive and simple user experience in ordering pizza.

E. Link Github

https://github.com/fitryatkj3/Kelompok1-Alpro.git