PROGRAMMING BASICS ASSIGNMENT PAPER

Submitted to fulfill the First Semester Final Exam Assignment in the Basic Programming course.

Lecturer:
Elin Panca Saputra S.kom., M.kom

Compiled by:

Swandaru Tirta Sandhika (17231105)

FACULTY OF ENGINEERING AND INFORMATICS INFORMATION TECHNOLOGY STUDY PROGRAM 2023/2024

BINA SARANA INFORMATIKA UNIVERSITY
jl. kramat raya no 98 Special region of Jakarta

2023

FOREWORD

Assalamualaikum wr, wb

the author's praise and gratitude for the presence of allah SWT finally we completed our paper entitled the ***purchase of ball merchandise*** can be completed. we express our deepest gratitude to Mr. Elin Panca Saputra S.kom., M.kom as a lecturer in the Basic Programming course.

The author is fully aware that in writing this paper we still have many shortcomings and are far from perfect. for that we invite readers to provide criticism and suggestions so that we can improve and minimize errors in the future. to come.

Hopefully the writing of this paper can provide benefits for all readers. Thank you. Wassalamualaikum wr, wb

Jakarta, December 13, 2023

Swandaru Tirta Sandhika
Author

CHAPTER I
INTRODUCTION

1.1 Definitions on Programming Basics

Program Application and *Data Science* technology-based applications are a growing program area in data analysis in the industry today. for this reason, an early introduction to basic programming technology with *Python is* needed. To students up to the level of students who are still in school or for students who take science and data science study programs. *Python* programming is a multiplatform programming language that is widely used in current technology applications, especially *software engineering and Data Analytics.* ***the*** purpose of making a *ball merchandise purchase* program to introduce as well as become one of the requirements for taking basic programming courses using *Python.*

1.2 Purpose of the paper

The purpose of writing this paper is to provide a better understanding of the basics of programming using python. this paper will discuss several concepts Basic concepts of Python Programming, History of the Python programming language, Basic structures in python and technology-based models that use python language.

1.3 Scope and topic coverage

The topic in this paper discusses the creation of simple applications using basic Python programs in general. this paper will not discuss too broadly complex, but uses a simple case study approach to prove that it can be done carefully.

1.4 Research Methodology

This paper is written based on a simple research approach from various recognized sources in the field of basic programming, the methodology used in writing this paper is to conduct literature studies and simple research in this field. in addition, this paper also follows the structure of academic writing commonly used in writing scientific papers.

1.5 Research methodology with SDLC (System Development Life Cycle)

SDLC / System Development Life Cycle is a systematic process for developing information systems. SDLC divides the system development process into several stages, so that this system can be managed properly.

The following is an illustration of the use of research methodology using SDLC (System development Life Cycle) technique:

The benefits of SDLC are:

- Helps ensure that the information or scope that is built meets the needs of the user.
- help ensure that the system it aims to build functions properly.
- helps reduce the risk of system development failure

Another use of SDLC is to help develop the system with the project team or as the main *roadmap* in developing the environment in the following systems:

- Understand user needs (society, market/industry, surrounding community)
- develop an effective design system that attracts investors into the development of application systems
- Implement the system correctly (fixed code multiple t i m e s after deployment or maintenance)
- test the system to ensure it works properly.
- apply the system to production so that products can be tested repeatedly

Stages in developing a simple application for ordering ball merchandise using SDLC:

1. **Program planning**

(our group discussed about the application program that we want to create and develop)

2. **program design**

including mock-up and structural design and optimization of its use.

3. **development**

(Our group developed this program based on our natural ideas and future demand)

4. **testing**

we do testing first when we finish creating our program and after it runs according to the design we add other features such as digital pay codes using other payments.

5. **Implementation**

in implementing this, we as a development can implement it into the cashier system and forward it for launching to consumers or entrepreneurs to distribute the program application.

6. **Maintenance**

This maintenance program or system is created so that the production system that we develop and make can be optimized again in it and is able to serve our consumers and users in the experience they get when using our products.

SDLC is a commonly used method for developing a scope of our systems, this method helps ensure that the systems developed and built meet the needs of users and function for the wider community.

CHAPTER II
PROGRAMMING BASICS USING PYTHON

2.1 History of python programming language development

a. History of Python Development

Python was created and developed by Guido Van Rossum, a programmer from the Netherlands. Its creation took place in the city of Amsterdam, Netherlands in 1990. In 1995 Python was developed again to be more compatible by Guido Van Rossum. Furthermore, in early 2000, there was an update to the Python version until it reached Version 3 until now. The choice of the name Python itself was taken from a fairly famous television show called Monty Python Flying Circus which is Guido van Rossum's favorite circus show.

b. Pros and Cons of Python

Many people are interested in using Python because it is considered easy to learn, even by beginners. The code is easy to read and can perform many complex functions easily due to the large number of standard libraries. Program development in Python can also be done quickly and using less code. Python can even make very complicated programs easy. Python itself supports multi-platform and multi-system and has an automatic memory management system like Java.

Unfortunately Python is quite slow to run. It also lacks support for Android and IOS platform development. Python also has limitations with database access. In addition, Python is not suitable for memory-intensive tasks and multi-core / multi-processor work.

That's a glimpse of information about the Python programming language that you can see. For developers or novice programmers, it is highly recommended to learn this programming language because it is considered easy. Python can also be applied in making software, applications on smartphones, GUI programs, CLI programs, Internet of Things, games and others. Hopefully this information is useful and adds to your insight.

2.2 Basic structure of python

a. Statement

python uses indentation as a way to mark up blocks of code. statements that have the same identity together as part of a single block of code are indented using four spaces a tab, improving readability and ensuring

that the structure derived from the flow control value statement (if, while, and for) can be clearly identified by looking at the example program below:

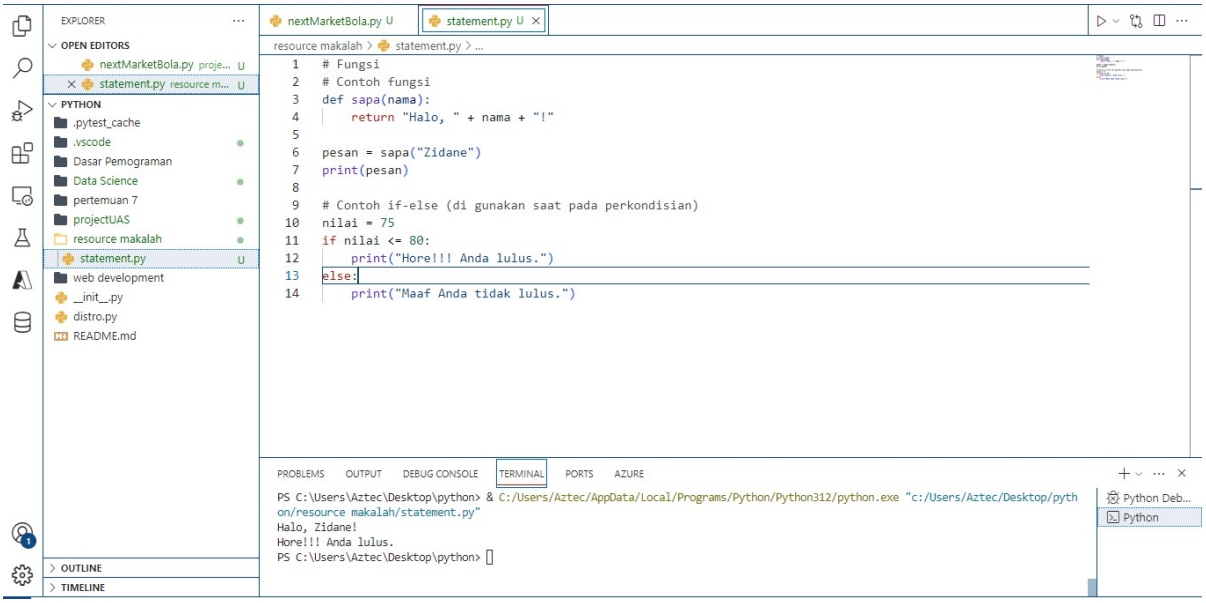


Figure 1.0 (example of basic program structure using if-else statement indentation)

In the figure, functions are blocks of code that can be invoked by passing certain arguments. They can help organize code into smaller, reusable parts. Function definitions start with the keyword **def.**

b. the function of each variable and how it works in a computer

2.3 main functions in creating simple applications with python

1. **main:** the place where the program code runs (*Running).* main will be executed inside the ongoing program
2. **If name == "_main": :** Ensures that the main function is only executed if this file is run as the main program, not if it is imported as a module file into another python program.
3. **Hashlib:** this library provides functions to calculate the hash (checksum) of data and this hashing technique is the process of converting data (such as strings or files) into unique hash values. used to store passwords in a secure way to ensure data integrity.
4. **datetime:** provides classes for manipulating and formatting dates and times and developers can create date and time objects and perform time subtraction and addition operations.

5. **getpass: The** *getpass* library provides functions to request users to enter passwords or information without displaying the input on the screen, which is useful for securing sensitive information such as passwords.
6. **Self: A** name convention used to define methods in python classes referring to instances of objects used referring to objects or methods used.
7. **class: a** blueprint for creating objects in python to define the attributes and methods that the objects will have.
8. **add_item:** a function or method that **adds** items to a programming structure such as a list or dictionary.
9. **for: The** basic structure of a looping program repeats a series of statements for each element in a sequence (list, tuple, or string)
10. **if/elif: The** basic structure of a branching program used to execute
11. **return: returns** the function call value or returns a string or tuple value
12. **while True: An** infinite loop control that will execute a block of code whose value is considered true *True* is a boolean constant that is always true, so the loop will continue to run endlessly
13. **break:** used to stop loop execution or exit the loop early if a condition is met

In this library, there is no need to add it again because it is directly available in the python.exe program.

CHAPTER III

CREATING A SIMPLE PYTHON APPLICATION: SOCCER MERCHANDISE PURCHASE & DATA SCIENCE IMPLEMENTATION

3.1 Definition of Data Science

a. Data Science is a combination of a number of sciences consisting of statistics, mathematics and business knowledge. The goal is to extract knowledge or information from data. usually people who are proficient in the field of data science use machine learning algorithms. which is useful in processing text data, video images, photos, audio and others that produce artificial intelligence systems, artificial intelligence systems are designed to meet all human needs for various tasks that are too difficult for humans. utilized by analysis and users in business to design the right strategy to solve a problem or achieve a goal.
b. **implementation in using simple data science**

(continued using evidence in the data science program)

C. Football merchandise purchase application program

In the scope of creating this simple application program, python for the ball merchandise purchase project may vary depending on the needs and goals of the business. from the review of the General Objectives that can be based on this project are:

1. **Order processing:**

Facilitate the order process Merchandise balls for customers or customers can easily choose the products they want, put customers in the shopping cart

2. **Management investment**

easing the process of ordering ball merchandise for customers, the goal is to ensure that customers can easily select the products they want, place them in the shopping cart and make the purchase process smooth.

3. **Payment Integration**

provide online payment options, which are safe and convenient for customers. the goal is to make it easier for customers to make payments, whether through barcode scanning, credit, bank transfers or other electronic payment methods.

4. **Promotion and Tax**

The use of promo codes (*Discounts)* to attract customers and drive sales. The goal is to increase product appeal and improve customer retention.

3.2 Source code:

Library

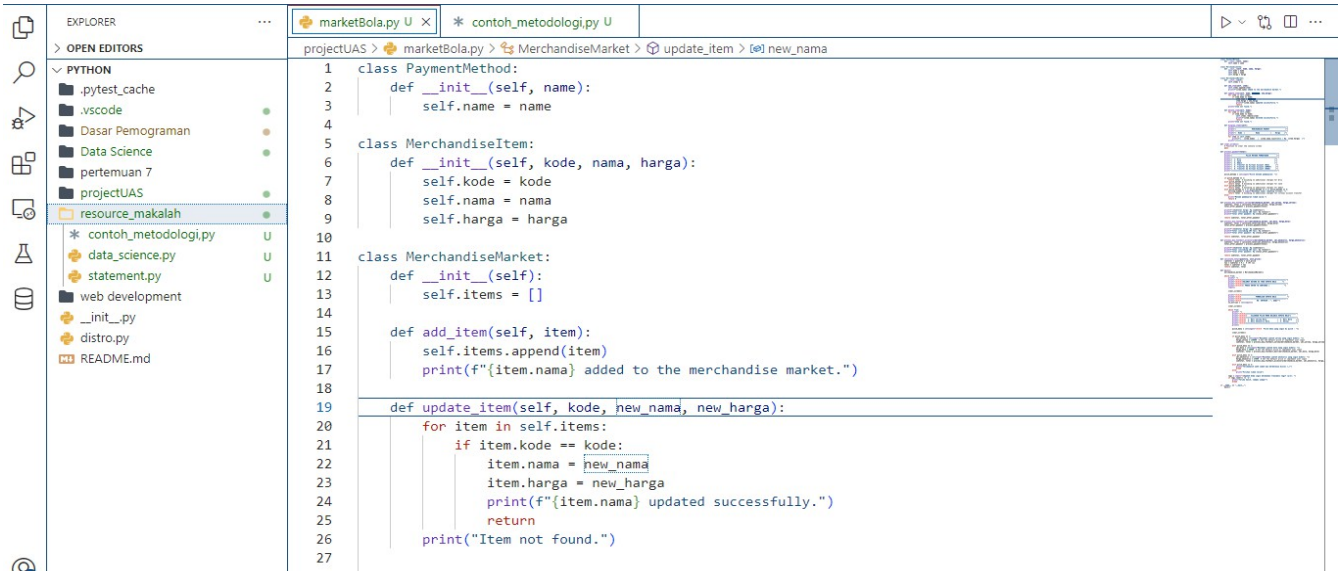


fig 1.0 using oop class in python application program

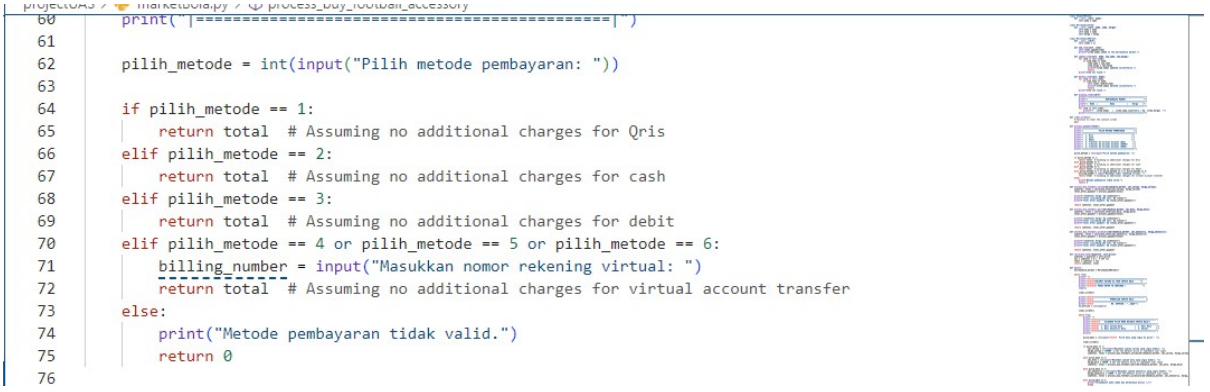
In the initial part we started with a collection of python libraries to make it easier to create programs. using the *Class function* to declare the Class function in our program as a standard for defining data and behavior that will form an object as well as syntax. we also use Def as entering the main value or value in a program to initialize the program at the beginning. we also use the def main function to add lines / long lines to the application program *Purchase of soccer merchandise* to separate the header and footer by using the same sign as the

Use the display function to display a list or display of products

Fig 1.1 Display_items, Clear_screen,

in this program using the display items function is used to display a list of items / lists onto the screen, this function can be used to be used to display any list of items in this application program, our group uses a lot of display items to help process data which will be processed again in the branching function, in this case study of making simple applications in *purchasing* ball *merchandise* there is a *Clear_Screen function this function* is used to delete the previous page when input is successfully entered in the program before displaying the new output.

c. nested IF function

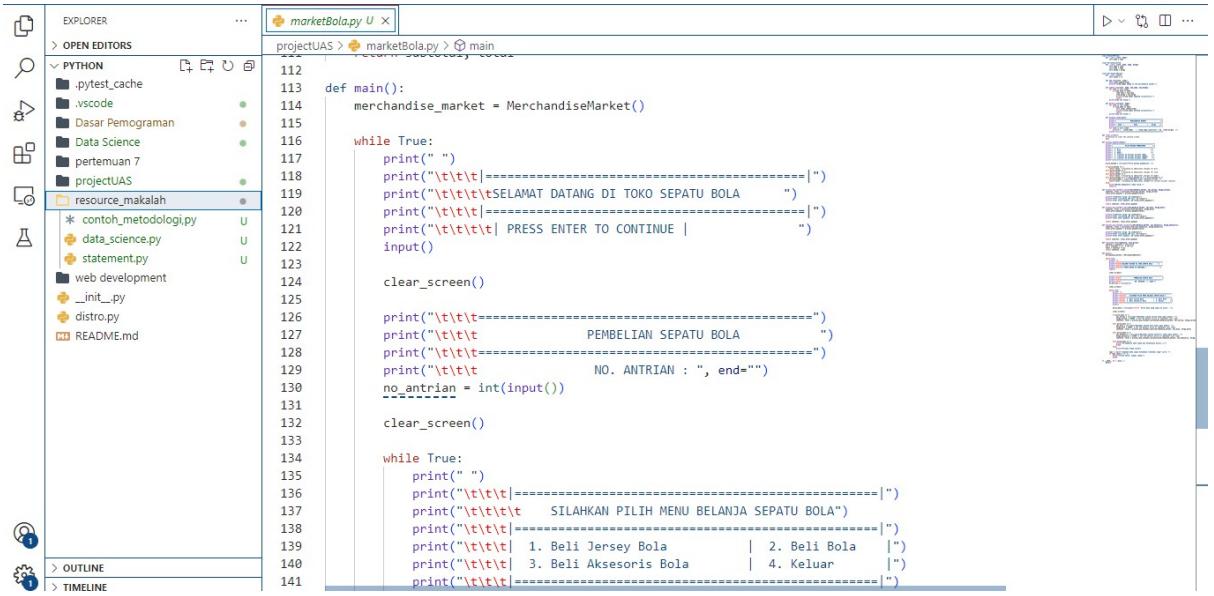


```
print("|================================================|")

pilih_metode = int(input("Pilih metode pembayaran: "))

if pilih_metode == 1:
    return total  # Assuming no additional charges for Qris
elif pilih_metode == 2:
    return total  # Assuming no additional charges for cash
elif pilih_metode == 3:
    return total  # Assuming no additional charges for debit
elif pilih_metode == 4 or pilih_metode == 5 or pilih_metode == 6:
    billing_number = input("Masukkan nomor rekening virtual: ")
    return total  # Assuming no additional charges for virtual account transfer
else:
    print("Metode pembayaran tidak valid.")
    return 0
```

in this program the if elif function in our program is used to test conditions and perform actions based on these conditions. this function can be used to create a more flexible program and can adapt to various case studies. this function can be used to create a more flexible program.

and can adapt to various conditions and there are several *return* commands used to stop the execution of the function if certain conditions are met, so in this sense it can be concluded that it can stop the function after the first condition is met.

d. initialization on def main, while True and break commands



case study in using the def main command can be defined as the main function in python programming and this function will be executed when the program is run in this def main function can be explained that it is used to

- displays the main menu
- receive input from the user
- process input from the user
- display the processing result

Break

The break command is used to stop the execution of loops in the creation of a simple application for ordering ball merchandise and can be implemented to stop looping on the choice of item addition options and item shopping calculations.

CHAPTER IV

CONCLUSION &

SUGGESTION

In the Presentation of Basic Programming for the creation of a *soccer booking Merchandise* application delivered by Brother Swandaru Tirta Sandhika and brother Muhammad Zidane on December 19, 2023 at Bina Sarana informatics University, found a solution in this Presentation so that in the future it can be interpreted into a real project. in this paper, discusses various basic concepts of Python programming such as:

- Variables
- Data type
- Math Operations
- Flow control
- Function

This presentation also discusses how to create a simple application for ordering Football Merchandise using python and this application has features:

1. Main menu
2. Merchandise List
3. Merchandise ordering
4. Merchandise payment
5. out

This simple application uses the concepts of basic python programming concepts and OOP (object oriented Programming) as an additional *class* to identify the case studies discussed that have been discussed in the presentation.

Suggestions for Basic programming papers using Python technology

a. more detailed explanation

the explanation of the use of simple applications using the python programming language can still be more detailed explained and many data types or strings can be represented in papers or presentations.

b. more diverse examples

in this presentation, it can be completed again which is more diverse including implementation or using other case studies as additional references / resources in the presentation of this program. the group should also include more complex functions of some parameters used in returning variable values or values.

c. discussion between groups

provide opportunities for participants to ask questions about the material presented and discuss in class 17.1b.07, overall this presentation is focused on the soccer booking merchandise application has provided useful information for participants. and this paper also this presentation can be used as a reference to learn programming using the python technology language.

Here's our application programming source code:

```
class PaymentMethod:

    def init(self, name):

        self.name = name



class MerchandiseItem:

    def init(self, code, name, price):

        self.code = code

        self.name = name

        self.price = price



class MerchandiseMarket:

    def init(self):
```

```
    self.items = []


def add_item(self, item):
    self.items.append(item)
    print(f"{item.name} added to the merchandise market.")


def update_item(self, code, new_name, new_price):
    for item in self.items:
        if item.code == code:
            item.name = new_name
            item.price = new_price
            print(f"{item.name} updated
            successfully.") return
    print("Item not found.")


def delete_item(self, code):
    for item in self.items:
        if item.code == code: self.items.remove(item)
            print(f"{item.name} deleted
            successfully.")
```

```
                return
        print("Item not found.")

    def display_items(self):
        print("|=================================================|")
        print("                |MERCHANDISE                    MARKET|")
        print("|=================================================|")
        print("|Code |Name     |                 Price          |")
        print("|=================================================|")
        for item in self.items:
            print(f"| {item.code} | {item.name.ljust(23)} | Rp. {item.price} | ")
        print("|=================================================|")

def clear_screen():
    # Function to clear the console screen
    pass

def process_payment(total):
    print("|=============================================|")
    print("|PICK     PAYMENT METHOD|                     ")
```

```
print("|=============================================|")
print("| 1. Qris                                     |")
print("| 2. Cash                                     |")
print("| 3.                                          |")
print("| 4. Transfer to Virtual Account (BCA)|       ")
print("| 5. Transfer to Virtual Account (GOPAY)    |")
print("| 6. Transfer to Virtual Account (DANA)       |")
print("|=============================================|")

select_method = int(input("Select payment method: "))

if select_method == 1:
    return total # Assuming no additional charges for Qris
elif select_method == 2:
    return total # Assuming no additional charges for cash
elif select_method == 3:
    return total # Assuming no additional charges for debit
elif select_method == 4 or select_method == 5 or select_method ==
    6: billing_number = input("Enter virtual account number: ")
    return total # Assuming no additional charges for virtual account transfer
```

```
    Else:
        print("Invalid payment method.") return 0


def process_buy_football_jersey(merchandise_market, jml_jersey, price_jersey):
    subtotal, total = calculate_total(jml_jersey, price_jersey)
    total_after_payment = process_payment(total)

    print(f "Subtotal price: Rp {subtotal}") print(f
    "Total (including 10% tax): Rp {total}")
    print(f "Total after payment: Rp {total_after_payment}")

    return subtotal, total_after_payment

def process_buy_football_ball(merchandise_market, jml_balls,
    price_balls): subtotal, total = calculate_total(jml_balls, price_balls)
    total_after_payment = process_payment(total)

    print(f "Subtotal price: Rp {subtotal}") print(f
    "Total (including 10% tax): Rp {total}")
```

```
    print(f "Total after payment: Rp {total_after_payment}")


    return subtotal, total_after_payment


def process_buy_football_accessory(merchandise_market, jml_accessories, price_accessories):
    subtotal, total = calculate_total(jml_accessories, price_accessories)
    total_after_payment = process_payment(total)


    print(f "Subtotal price: Rp {subtotal}") print(f
    "Total (including 10% tax): Rp {total}")
    print(f "Total after payment: Rp {total_after_payment}")


    return subtotal, total_after_payment


def calculate_total(quantity, unit_price):
    subtotal = quantity * unit_price
    tax = subtotal * 0.1 # 10% tax
    total = subtotal + tax
    subtotal return, total
```

```
def main():
    merchandise_market = MerchandiseMarket()

    while True:
        print(" ")
        print("\t\t\t|=============================================|")
        print("\t\t\t\tWelcome TO THE BALL SHOES SHOP                  ")
        print("\t\t|=============================================|")
        print("\t\t\t\t| PRESS ENTER TO CONTINUE |              ")
        input()

        clear_screen()

        print("\t\t\t===============================================")
        print("\t\t\t              BALL SHOE PURCHASE               ")
        print("\t\t\t===============================================")
        print("\t\t\t               NO. QUEUE : ", end="")
        no_antrian = int(input())

        clear_screen()
```

```
    while True:
        print(" ")
        print("\t\t\t|=======================================================|")
        print("\t\t\t    SELECT BALL SHOPPING MENU")
        print("\t\t\t|=======================================================|")
        print("\t\t\t|. Buy Ball Jerseys              2. Buy Balls|     ")
        print("\t\t\t|. Buy Ball Accessories          4. Exit|          ")
        print("\t\t\t|=======================================================|")
        print()

        select_menu = int(input("\t\t\t Select the menu you want to select: "))

        clear_screen()

        if select_menu == 1:
            jml_jersey = int(input("Enter the number of jerseys you want to buy: "))
            price_jersey = 150000 # Set the default price or implement user input
                    subtotal, total = process_buy_football_jersey(merchandise_market, jml_jersey,
price_jersey)
```

```
        elif select_menu == 2:
            jml_bola = int(input("Enter the number of balls you want to buy: "))
            price_bola = 50000 # Set the default price or implement user input
            subtotal, total = process_buy_football_ball(merchandise_market, jml_ball, price_ball)

        elif select_menu == 3:
            jml_aksesoris = int(input("Enter the number of accessories you want to buy: "))
            price_aksesoris = 25000 # Set the default price or implement user input
              subtotal, total = process_buy_football_accessory(merchandise_market, jml_accessories,
price_accessories)

        elif select_menu == 4:
            print("Thank you for shopping here >_<") break
        Else:
            print("Invalid selection")

    again = input("\nDo you want to do the transaction again? (y/n): ") if
    lagi.lower() !='y':
```

```
            print("Thanks, see you!") break


if name_____== "main":

    main()
```

LITERATURE

[1] *Sharma, P., & Singh, D. (2015, April 1). Comparative Study of Various SDLC Models on Different Parameters. International Journal of Engineering Research, 4(4), 188-191. https://doi.org/10.17950/ijer/v4s4/405,*

[2] *M., L., & V., K. (2014, September 9). TEACHING ALGORITHMIZATION AND PROGRAMMING USING PYTHON LANGUAGE. Information Technologies in Education, 013-023. https://doi.org/10.14308/ite000493*

[3] *Ramady, G. (2020, May 7). DESIGNING A DESKTOP-BASED SOCCER TICKET BOOKING APPLICATION PROGRAM USING VISUAL STUDIO 2010. INFORMATION (Journal of Informatics And Systems Information), 12(1), 54–66. https://doi.org/10.37424/informasi.v12i1.49*

[4] *Mozef, E., & Kurniati, S. (2022, March 30). Application of measuring ball speed in table tennis match videos using OpenCV and Python. JITEL (Scientific Journal of Telecommunications, Electronics, and Electrical Power), 2(1), 11-24. https://doi.org/10.35313/jitel.v2.i1.2022.11-24*