LAPORAN PRAKTIKUM

POSTTEST 7

ALGORITMA PEMROGRAMAN DASAR

Disusun oleh:

Sufi Ridho Utomo (2509106101)

Kelas (C1’25)

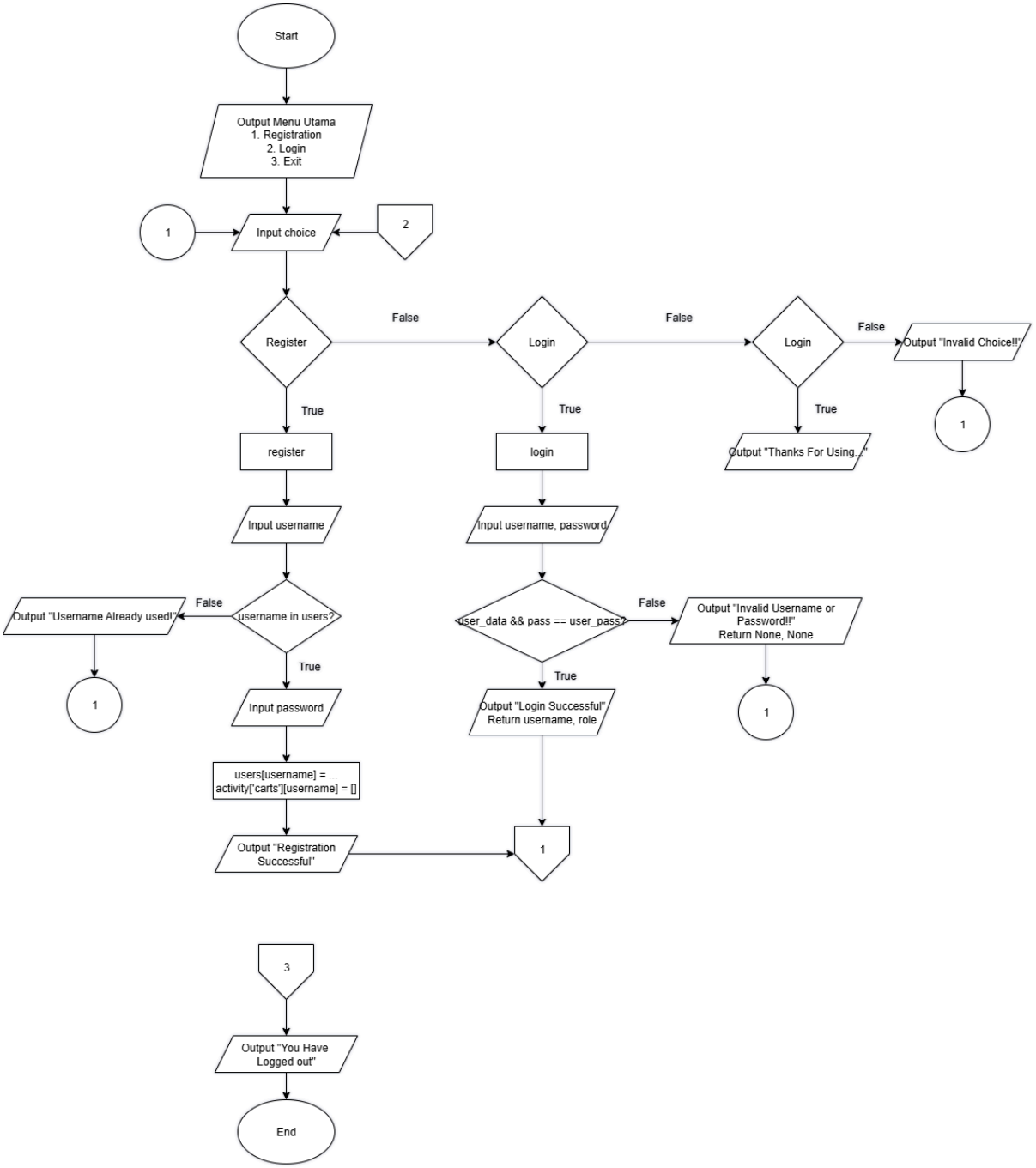
PROGRAM STUDI INFORMATIKA

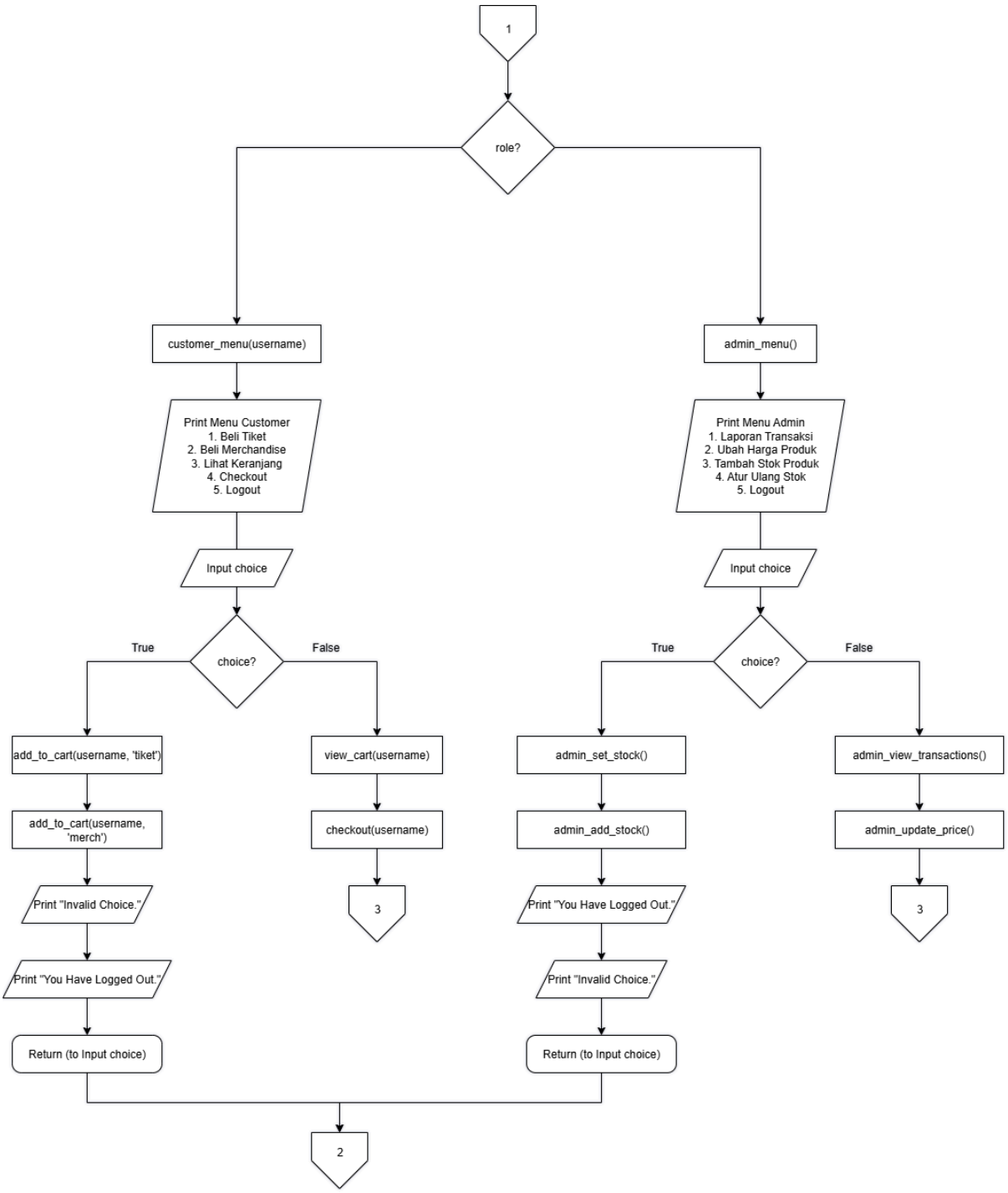
UNIVERSITAS MULAWARMAN

SAMARINDA

2025

1. Flowchart


Start
Output Menu Utama
1. Registration
2. Login
3. Exit
1
Input choice
2
Register
False
Login
False
Login
False
Output "Invalid Choice!!"
1
True
True
True
Output "Thanks For Using..."
register
login
Input username
Input username, password
Output "Username Already used!"
False
username in users?
True
1
Input password
user_data && pass == user_pass?
False
Output "Invalid Username or Password!!"
Return None, None
True
1
Output "Login Successful"
Return username, role
users[username] = ...
activity['carts'][username] = []
Output "Registration Successful"
1
3
Output "You Have Logged out"
End



Gambar 1.0 Flowchart

2. Deskripsi Program

Program ini dibuat sebagai bentuk aplikasi yang dapat digunakan pengguna untuk membeli sebuah tiket dan merchandise pada sebuah balapan 24 Jam Nürburgring Nordschleife yang ada di Jerman dan mempermudah pengguna untuk mengakses pembelian tiket dimanapun dan kapanpun pengguna berada tanpa harus datang langsung ke tempat penjualan tiketnya dan aplikasi ini cukup praktis pada penggunaannya yang dimana pengguna cukup registrasi dan login untuk membeli tiket tersebut dengan mudah dan cepat.

3. Source Code

```
import os
os.system("cls")

users = {
    'Admin': {'password': 'Sufi123', 'role': 'admin'},
    'Sufi': {'password': 'Sufi321', 'role': 'customer'}
}

products = {
    'tickets': [
        [1, "Event Ticket (Thur-Sun)", 1500000, 100],
        [2, "Weekend Ticket (Fri-Sun)", 1300000, 200],
        [3, "Race Ticket", 1150000, 300],
        [4, "Day Ticket", 700000, 350],
        [5, "Paddock Access Add-on", 1000000, 75]
    ],
    'merchandise': [
        [1, "N24H Official T-Shirt", 1200000, 100],
        [2, "N24H Cap", 600000, 120],
        [3, "N24H SunGlasses", 2700000, 75],
        [4, "Wall Clock (NBR)", 700000, 100],
        [5, "Scale Model Car 1:43", 1300000, 75]
    ]
}

activity = {
    'carts': {
        'Sufi': []
    },
    'transactions': []
}


def register():
    print("\n ==| User Registration |==")

    username = input("New Username: ")

    if username in users:
        print("\nUsername Already used!")
        return

    password = input("New Password: ")
    users[username] = {'password': password, 'role': 'customer'}
    activity['carts'][username] = []
```

```
    print(f"Registration Successful: {username}")

def login():
    print("\n ==| Login Admin/Customer |==")
    print("+-------------------------------------------------------+")
    username = input("Username: ")
    password = input("Password: ")
    print(" ")

    user_data = users.get(username)

    if user_data and user_data['password'] == password:
        print(f"Login Successful, Welcome {username}!")
        return username, user_data['role']

    print("Invalid Username or Password!!")
    return None, None

def display_products(product_list, title):
    print(f"\n ==| {title} |==")

print("+----------+-------------------------------------------------------
---+-----------------------------------+----------------+")
    print("| ID   | Nama Produk                   | Harga            | Stok
|")

print("+----------+-------------------------------------------------------
---+-----------------------------------+----------------+")
    for item in product_list:
        print(f"| {item[0]:^3} | {item[1]:<28} | Rp {item[2]:>11,} |
{item[3]:^5} |")

print("+----------+-------------------------------------------------------
---+-----------------------------------+----------------+")

def add_to_cart(username, product_type):
    product_list = products['tickets'] if product_type == 'tiket' else
products['merchandise']
    title = "Daftar Tiket" if product_type == 'tiket' else "Daftar
Merchandise"
    display_products(product_list, title)

    item_id_str = input("Enter The Product ID You Want To Buy: ")

    if not item_id_str.isdigit():
        print("Invalid ID Input, Must Be A Number.")
        return
```

```
    item_id = int(item_id_str)
    selected_item = None
    for item in product_list:
        if item[0] == item_id:
            selected_item = item
            break

    if not selected_item:
        print("ID Product Not Found.")
        return

    quantity_str = input(f"Amount '{selected_item[1]}' What You Want To Buy:
")

    if not quantity_str.isdigit() or int(quantity_str) <= 0:
        print("Invalid Number, Number Must Be A Greater Than 0.")
        return

    quantity = int(quantity_str)

    if quantity > selected_item[3]:
        print(f"Insufficient Stock. Remaining Stock: {selected_item[3]}")
        return

    activity['carts'][username].append({'item': selected_item, 'quantity':
quantity, 'type': product_type})
    print(f"Successfully Added {quantity} x {selected_item[1]} To Cart.")

def view_cart(username):
    print("\n ==| Your Shopping Cart |==")
    user_cart = activity['carts'].get(username, [])

    if not user_cart:
        print("Your Cart Is Empty.")
        return


print("+------------+-------------------------------------------------------
-----+--------------------+-----------------------------------+")
    print("| No.  | Nama Produk                  | Jumlah   | Subtotal
|")

print("+------------+-------------------------------------------------------
-----+--------------------+-----------------------------------+")

    total_price = 0
    for i, cart_item in enumerate(user_cart, 1):
        item_details = cart_item['item']
```

```
        quantity = cart_item['quantity']
        subtotal = item_details[2] * quantity
        total_price += subtotal
        print(f"| {i:^4} | {item_details[1]:<28} | {quantity:^8} | Rp
{subtotal:>11,} |")


print("+------------+------------------------------------------------------------
-----+------------------+-----------------------------------+")
    total_str = f"Rp {total_price:,}"
    print(f"| Total Pembayaran:{total_str: >52} |")

print("+------------------------------------------------------------------------
------------------------------------------------------------------+")


def checkout(username):
    user_cart = activity['carts'].get(username, [])

    if not user_cart:
        print("Your Cart Is Empty. There Is Nothing To Checkout.")
        return

    print("\n ==| Checkout Cart |==")
    print("+-----------------------------------------+")

    for cart_item in user_cart:
        item_id = cart_item['item'][0]
        product_list = products['tickets'] if cart_item['type'] == 'tiket'
else products['merchandise']

        current_item_stock = -1
        for db_item in product_list:
            if db_item[0] == item_id:
                current_item_stock = db_item[3]
                break

        if cart_item['quantity'] > current_item_stock:
            print(f"Sorry, Stock For '{cart_item['item'][1]}' Has Run
Out/Decreased. Transaction Cancelled.")
            return

    total_price = 0
    for cart_item in user_cart:
        item_details = cart_item['item']
        quantity = cart_item['quantity']
        item_details[3] -= quantity
        total_price += item_details[2] * quantity
```

```
    activity['transactions'].append({'customer': username, 'items':
user_cart, 'total': total_price})
    activity['carts'][username] = []
    print(f"Transaction Successful, Total Payment: Rp{total_price}.
Thanks!")

def admin_view_transactions():
    print("\n ==| Report Of All Transactions |==")

print("+-------------------------------------------------------------------
+")

    if not activity['transactions']:
        print("None Transaction.")
        return

    for i, trans in enumerate(activity['transactions'], 1):
        print(f"Transaction #{i}")
        print(f"  Customer: {trans['customer']}")
        print(f"  Total    : Rp{trans['total']:,}")
        print("  Detail Item:")
        for cart_item in trans['items']:
            item = cart_item['item']
            quantity = cart_item['quantity']
            print(f"    - {item[1]} (x{quantity})")
        print("-" * 30)

def admin_select_product():
    choice = input("Select Product Category (1: Ticket, 2: Merchandise): ")
    if choice == '1':
        product_list = products['tickets']
        display_products(product_list, "Ticket List")
    elif choice == '2':
        product_list = products['merchandise']
        display_products(product_list, "Merchandise List")
    else:
        print("Invalid Category.")
        return None

    item_id_str = input("Enter The Product To Be Set: ")
    if not item_id_str.isdigit():
        print("Invalid ID Input, Must Be A Number.")
        return None

    item_id = int(item_id_str)
    for item in product_list:
        if item[0] == item_id:
```

```
            return item

    print("Product Not Found.")
    return None

def admin_update_price():
    print("\n ==| Change Price |==")
    print("+---------------------------------------+")
    item_to_manage = admin_select_product()

    if not item_to_manage:
        return

    print(f"Change The Price For: '{item_to_manage[1]}'")
    print(f"Current Price: Rp{item_to_manage[2]}")

    new_price_str = input("Enter New Price: ")
    if not new_price_str.isdigit() or int(new_price_str) < 0:
        print("Invalid Price, Must Be A Non-Negative Number.")
        return

    item_to_manage[2] = int(new_price_str)
    print("Price Changed Successfully")

def admin_add_stock():
    print("\n ==| Add Product Stock |==")
    print("+-------------------------------------------------+")
    item_to_manage = admin_select_product()

    if not item_to_manage:
        return

    print(f"Add Stock To: '{item_to_manage[1]}'")
    print(f"Currrent Stock: {item_to_manage[3]}")

    amount_str = input("Stock Quantity Increased: ")
    if not amount_str.isdigit() or int(amount_str) <= 0:
        print("Invalid Number, Must Be A Positive Number.")
        return

    item_to_manage[3] += int(amount_str)
    print(f"Stock Added Successfully. New Stock: {item_to_manage[3]}")

def admin_set_stock():
    print("\n ==| Product Stock Reset |==")
    print("+-------------------------------------------------+")
    item_to_manage = admin_select_product()
```

```
    if not item_to_manage:
        return

    print(f"Reset Stock For: '{item_to_manage[1]}'")
    print(f"Current Stock: {item_to_manage[3]}")

    new_stock_str = input("Enter New Stock Quantity: ")
    if not new_stock_str.isdigit() or int(new_stock_str) < 0:
        print("Invalid Number, Must Be A Non-Negative Number.")
        return

    item_to_manage[3] = int(new_stock_str)
    print(f"Stock Successfully Reset, New Stock: {item_to_manage[3]}")

def customer_menu(username):
    print("\n ==| Customer Menu |==")
    print("+----------------------------------------+")
    print("| 1. Beli Tiket       |")
    print("| 2. Beli Merchandise |")
    print("| 3. Lihat Keranjang  |")
    print("| 4. Checkout         |")
    print("| 5. Logout           |")
    print("+----------------------------------------+")
    choice = input("Enter Your Choice: ")

    if choice == '1':
        add_to_cart(username, 'tiket')
        customer_menu(username)
    elif choice == '2':
        add_to_cart(username, 'merch')
        customer_menu(username)
    elif choice == '3':
        view_cart(username)
        customer_menu(username)
    elif choice == '4':
        checkout(username)
        customer_menu(username)
    elif choice == '5':
        print("You Have Logged Out.")
        return
    else:
        print("Invalid Choice.")
        customer_menu(username)

def admin_menu():
    print("\n ==| Admin Menu |==")
    print("+-----------------------------------------------------+")
    print("| 1. Laporan Transaksi       |")
```

```
    print("| 2. Ubah Harga Produk       |")
    print("| 3. Tambah Stok Produk      |")
    print("| 4. Atur Ulang Stok Produk  |")
    print("| 5. Logout                  |")
    print("+----------------------------------------------------+")
    choice = input("Enter Your Choice: ")

    if choice == '1':
        admin_view_transactions()
        admin_menu()
    elif choice == '2':
        admin_update_price()
        admin_menu()
    elif choice == '3':
        admin_add_stock()
        admin_menu()
    elif choice == '4':
        admin_set_stock()
        admin_menu()
    elif choice == '5':
        print("You Have Logged Out.")
        return
    else:
        print("Invalid Choice.")
        admin_menu()

def main_menu():
    print("\n ==| Welcome To Nürburgring |==")
    print("+------------------------------------------------------------+")
    print("|       1. Registration        |")
    print("|          2. Login            |")
    print("|          3. Exit             |")
    print("+------------------------------------------------------------+")
    choice = input("Enter Your Choice: ")

    if choice == '1':
        register()
        main_menu()
    elif choice == '2':
        username, role = login()
        if username and role == 'customer':
            customer_menu(username)
        elif username and role == 'admin':
            admin_menu()
        main_menu()
    elif choice == '3':
        print("Thanks For Using Nürburgring")
        return
```

```
    else:
        print("Invalid Choice!!, Try Again")
        main_menu()

if __name__ == "__main__":
    main_menu()
```

Code Blocks 1.1 Source Code

4. Hasil Output

4.1 Hasil Output Registrasi

```
 ==| Welcome To Nürburgring |==
+-------------------------------+
|       1. Registration         |
|          2. Login             |
|          3. Exit              |
+-------------------------------+
Enter Your Choice: 1

 ==| User Registration |==
New Username: Tono
New Password: heru
Registration Successful: Tono
```

Code Blocks 1.2 Output Registrasi

4.2 Hasil Output Login Customer

```
 ==| Welcome To Nürburgring |==
+-------------------------------+
|       1. Registration         |
|          2. Login             |
|          3. Exit              |
+-------------------------------+
Enter Your Choice: 2

 ==| Login Admin/Customer |==
+-----------------------------+
Username: Tono
Password: heru

Login Successful, Welcome Tono!
```

Code Blocks 1.3 Output Login Customer

4.2 Hasil Output Login Admin

```
 ==| Welcome To Nürburgring |==
+-------------------------------+
|       1. Registration         |
|          2. Login             |
|          3. Exit              |
+-------------------------------+
Enter Your Choice: 2

 ==| Login Admin/Customer |==
+-----------------------------+
Username: Admin
Password: Sufi123

Login Successful, Welcome Admin!
```

Code Blocks 1.4 Output Login Admin

4.2 Hasil Output Admin Menu

```
 ==| Admin Menu |==
+-----------------------------+
| 1. Laporan Transaksi        |
| 2. Ubah Harga Produk        |
| 3. Tambah Stok Produk       |
| 4. Atur Ulang Stok Produk   |
| 5. Logout                   |
+-----------------------------+
Enter Your Choice: 2

 ==| Change Price |==
+---------------------+
Select Product Category (1: Ticket, 2: Merchandise): 2

 ==| Merchandise List |==
+------+-------------------------------+--------------------+----------+
| ID   | Nama Produk                   | Harga              | Stok     |
+------+-------------------------------+--------------------+----------+
|  1   | N24H Official T-Shirt         | Rp   1,200,000 |  100     |
|  2   | N24H Cap                      | Rp     600,000 |  120     |
```

```
|  3  | N24H SunGlasses              | Rp   2,700,000 |  75   |
|  4  | Wall Clock (NBR)             | Rp     700,000 |  100  |
|  5  | Scale Model Car 1:43         | Rp   1,300,000 |  75   |
+-----+------------------------------+----------------+-------+
Enter The Product To Be Set: 3
Change The Price For: 'N24H SunGlasses'
Current Price: Rp2700000
Enter New Price: 2300000
Price Changed Successfully
```

Code Blocks 1.5 Output Login Admin Menu

4.2 Hasil Output Customer Menu

```
 ==| Customer Menu |==
+-----------------------+
| 1. Beli Tiket         |
| 2. Beli Merchandise   |
| 3. Lihat Keranjang    |
| 4. Checkout           |
| 5. Logout             |
+-----------------------+
Enter Your Choice: 1

 ==| Daftar Tiket |==
+-----+------------------------------+----------------+-------+
| ID  | Nama Produk                  | Harga          | Stok  |
+-----+------------------------------+----------------+-------+
|  1  | Event Ticket (Thur-Sun)      | Rp   1,500,000 |  100  |
|  2  | Weekend Ticket (Fri-Sun)     | Rp   1,300,000 |  200  |
|  3  | Race Ticket                  | Rp   1,150,000 |  300  |
|  4  | Day Ticket                   | Rp     700,000 |  350  |
|  5  | Paddock Access Add-on        | Rp   1,000,000 |  75   |
+-----+------------------------------+----------------+-------+
Enter The Product ID You Want To Buy: 5
Amount 'Paddock Access Add-on' What You Want To Buy: 2
Successfully Added 2 x Paddock Access Add-on To Cart.
```

Code Blocks 1.6 Output Login Customer Menu

5. Langkah-langkah GIT

5.1 GIT Add

```
PS C:\Users\acer\Documents\Praktikum_APD_2025_C1> git add .
```

Gambar 1.3 GIT Add

GIT Add digunakan untuk menambahkan file ke *staging area* atau tempat penyimpanan sebelum disimpan secara permanen.

5.2 GIT Commit

```
PS C:\Users\acer\Documents\Praktikum_APD_2025_C1> git commit -m
"Tugas_APD_C1"
[main b88dcd8] Tugas_APD_C1
```

Gambar 1.4 GIT Commit

GIT Commit digunakan untuk menimpan perubahan file yang telah diubah pada GIT *Add.*

5.3 GIT Push

```
PS C:\Users\acer\Documents\Praktikum_APD_2025_C1> git push -u origin
main
Enumerating objects: 13, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 3.77 KiB | 3.77 MiB/s, done.
Total 7 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To https://github.com/SufiRidhoUtomo/Praktikum_APD_2025_C1.git
   3f69734..cb26c39  main -> main
branch 'main' set up to track 'origin/main'.
```

Gambar 1.5 GIT Push

GIT Push digunakan untuk mengirim perubahan file dari *local repository* ke *Remote* untuk diproses oleh *Remote* untuk dihubungkan menuju *online Repository.*