LAPORAN PRAKTIKUM

POSTTEST 5

ALGORITMA PEMROGRAMAN DASAR

Disusun oleh:

Sufi Ridho Utomo (2509106101)

Kelas (C1’25)

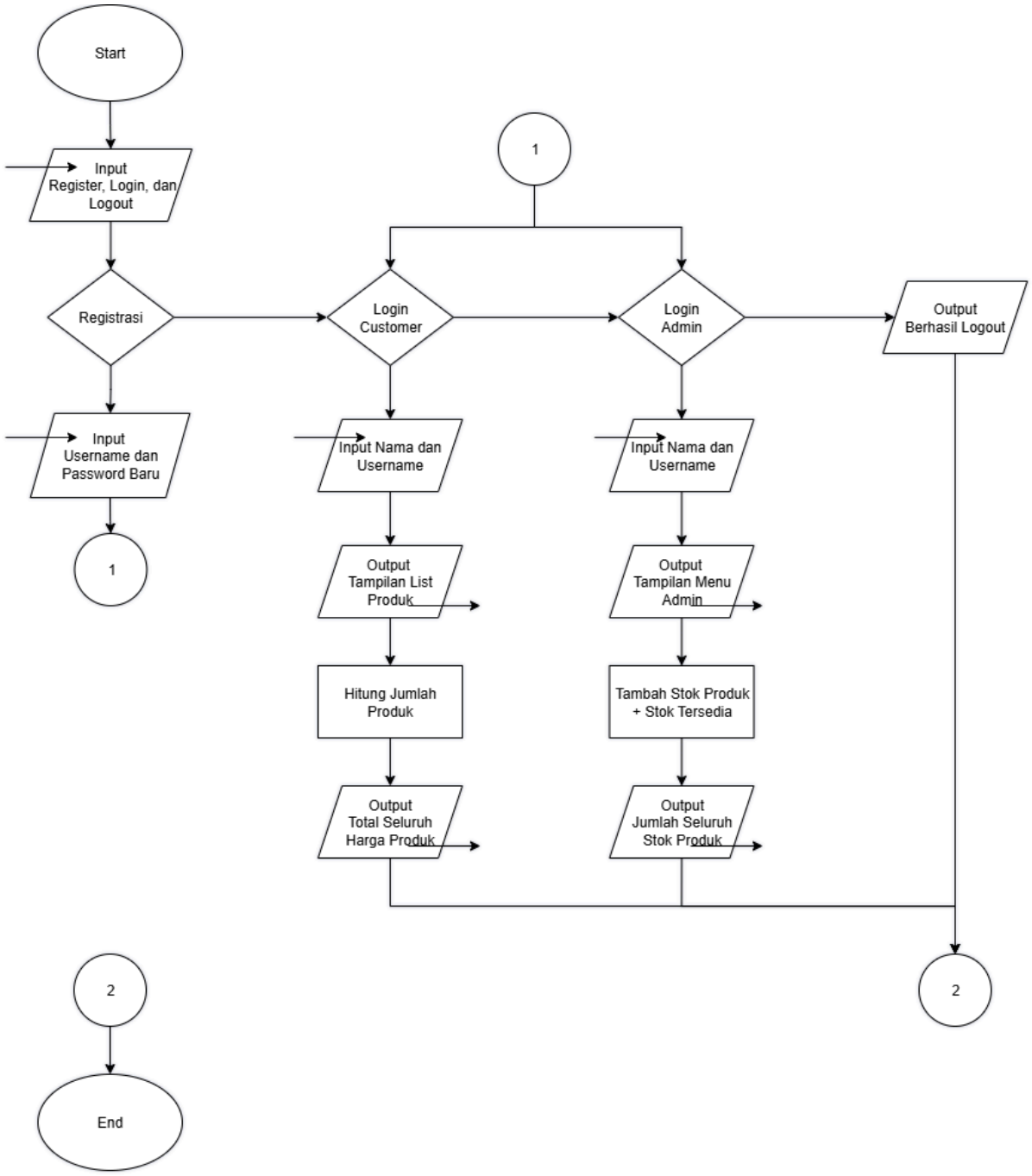
PROGRAM STUDI INFORMATIKA

UNIVERSITAS MULAWARMAN

SAMARINDA

2025

1. Flowchart



Gambar 1.0 Flowchart

2. Deskripsi Program

Program ini dibuat untuk mempermudah pelanggan untuk membeli tiket dan merch Nürburgring Nordschleife dengan cepat dan mudah dipahami.

3. Source Code

```
users = {
    'Admin': {'password': 'Sufi123', 'role': 'admin'},
    'Sufi': {'password': 'Sufi456', 'role': 'customer'}
}

carts = {
    'Sufi': []
}

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]
]

transactions = []

def main():
    while True:
        print(" ")
        print(""" ==| Welcome To Nürburgring |==
+-------------------------------------------------------------+
|       1. Registration        |
|          2. Login            |
|          3. Logout           |
+-------------------------------------------------------------+""")
        choice = input("Enter Your Choice: ")

        if choice == '1':
            register()
```

```
        elif choice == '2':
            username, role = login()
            if username and role == 'customer':
                customer_menu(username)
            elif username and role == 'admin':
                admin_menu()
        elif choice == '3':
            print("Thanks For Using Nürburgring")
            break
        else:
            print("Invalid Choice!!, Try Again")

if __name__ == "__main__":
    main()
def register():
    print(" ")
    print(" ==| User Registration |==")
    username = input("New Username: ")

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

    password = input("New Password: ")
    users[username] = {'password': password, 'role': 'customer'}
    carts[username] = []
    print(f"Registration Successful: {username}")

def login():
    print(" ")
    print(" ==| 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(" ")
    print(" ==| Ticket/Merchandise List |==")
```

```
print("+-------------------------------------------------------------+")
    print("ID | Ticket and Merch                 | Price        | Stock")
    print("-" * 55)
    for item in product_list:
        # Format agar tampilan rapi
        print(f"{item[0]:<2} | {item[1]:<25} | Rp{item[2]:<10} | {item[3]}")
    print("-" * 55)

def add_to_cart(username, product_type):
    product_list = tickets if product_type == 'tiket' else merchandise
    title = "Ticket List" if product_type == 'tiket' else "Merchandise List"
    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

    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(" ")
    print(" ==| Your Shopping Cart |==")
```

```
    print("+-------------------------------------------------+")
    user_cart = carts.get(username, [])

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

    total_price = 0
    print("No | Produk Name                 | Amount | Subtotal")
    print("-" * 55)
    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:<2} | {item_details[1]:<25} | {quantity:<6} |
Rp{subtotal}")

    print("-" * 55)
    print(f"Total Spending: Rp{total_price}")

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

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

    print(" ")
    print(" ==| Checkout Cart |==")
    print("+-----------------------------------------+")
    for cart_item in user_cart:
        item_id = cart_item['item'][0]
        product_list = tickets if cart_item['type'] == 'tiket' else
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

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

def customer_menu(username):
    while True:
        print(" ")
        print(" ==| Customer Menu |==")
        print("+-----------------------------------------+")
        print("1. View & Buy Ticket")
        print("2. View & Buy Merchandise")
        print("3. View Shopping Cart")
        print("4. Checkout")
        print("5. Logout")
        choice = input("Enter Your Choice: ")

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


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

print("+-----------------------------------------------------------------------
+")
    if not transactions:
        print("None Transaction.")
        return
    for i, trans in enumerate(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 = tickets
        display_products(product_list, "Ticket List")
    elif choice == '2':
        product_list = 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(" ")
    print(" ==| 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(" ")
    print(" ==| 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(" ")
    print(" ==| 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 admin_menu():
    while True:
        print(" ")
        print(" ==| Admin Menu |==")
        print("+-----------------------------------+")
        print("1. View Transaction Reports")
        print("2. Change Product Price")
```

```
        print("3. Added New Stock")
        print("4. Product Stock Reset")
        print("5. Logout")
        choice = input("Enter Your Choice: ")

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

Gambar 1.1 Source Code

4. Hasil Output

4.1 Hasil Output Registrasi

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

 ==| User Registration |==
New Username: Bintang
New Password: bintang cihuy
Registration Successful: Bintang
```

Gambar 1.2 Output Registrasi

4.2 Hasil Output Login Customer

```
 ==| Welcome To Nürburgring |==
+-------------------------------------------------------------+
```

```
|       1. Registration        |
|          2. Login            |
|          3. Logout           |
+------------------------------------------------------------+
Enter Your Choice: 2

 ==| Login Admin/Customer |==
+----------------------------------------------------------+
Username: Bintang
Password: Bintang Cihuy

Login Successful, Welcome Bintang!
```

Gambar 1.3 Output Login Customer

4.2 Hasil Output Login Admin

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

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

Login Successful, Welcome Admin!
```

Gambar 1.4 Output Login Admin

4.2 Hasil Output Admin Menu

```
 ==| Admin Menu |==
+-----------------------------------+
1. View Transaction Reports
2. Change Product Price
3. Added New Stock
4. Product Stock Reset
```

```
5. Logout
Enter Your Choice: 3

 ==| Add Product Stock |==
+-------------------------------------------------+
Select Product Category (1: Ticket, 2: Merchandise): 1

 ==| Ticket/Merchandise List |==
+-----------------------------------------------------------+
ID | Ticket and Merch                | Price        | Stock
-----------------------------------------------------
1  | Event Ticket (Thur-Sun)   | Rp1500000    | 100
2  | Weekend Ticket (Fri-Sun)  | Rp1300000    | 200
3  | Race Ticket               | Rp1150000    | 300
4  | Day Ticket                | Rp700000     | 350
5  | Paddock Access Add-on     | Rp1000000    | 75
-----------------------------------------------------
Enter The Product To Be Set: 4
Add Stock To: 'Day Ticket'
Currrent Stock: 350
Stock Quantity Increased: 375
Stock Added Successfully. New Stock: 725
```

Gambar 1.5 Output Login Admin Menu

4.2 Hasil Output Customer Menu

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

 ==| Login Admin/Customer |==
+-----------------------------------------------------+
Username: Bintang
Password: Bintang Cihuy

Login Successful, Welcome Bintang!

 ==| Customer Menu |==
```

```
+----------------------------------------+
1. View & Buy Ticket
2. View & Buy Merchandise
3. View Shopping Cart
4. Checkout
5. Logout
Enter Your Choice: 2

 ==| Ticket/Merchandise List |==
+-------------------------------------------------------------+
ID | Ticket and Merch                | Price        | Stock
-------------------------------------------------------------
1  | N24H Official T-Shirt     | Rp1200000    | 100
2  | N24H Cap                  | Rp600000     | 120
3  | N24H SunGlasses           | Rp2700000    | 75
4  | Wall Clock (NBR)          | Rp700000     | 100
5  | Scale Model Car 1:43      | Rp1300000    | 75
-------------------------------------------------------------
Enter The Product ID You Want To Buy: 2
Amount 'N24H Cap' What You Want To Buy: 4
Successfully Added 4 x N24H Cap To Cart.

 ==| Customer Menu |==
+----------------------------------------+
1. View & Buy Ticket
2. View & Buy Merchandise
3. View Shopping Cart
4. Checkout
5. Logout
Enter Your Choice: 3

 ==| Your Shopping Cart |==
+-----------------------------------------------------+
No | Produk Name                | Amount | Subtotal
-------------------------------------------------------
1  | N24H Cap                   | 4      | Rp2400000
-------------------------------------------------------
Total Spending: Rp2400000

 ==| Customer Menu |==
+----------------------------------------+
1. View & Buy Ticket
```

```
2. View & Buy Merchandise
3. View Shopping Cart
4. Checkout
5. Logout
Enter Your Choice: 4

 ==| Checkout Cart |==
+----------------------------------------+
Transaction Successful, Total Payment: Rp2400000. Thanks!
```

Gambar 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
"Post_Test_APD_5"
[main 897ef25] Post_Test_APD_5
```

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: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 3.66 KiB | 535.00 KiB/s, done.
Total 9 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
```

```
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To https://github.com/SufiRidhoUtomo/Praktikum_APD_2025_C1.git
   0962ca0..7762900  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.*