

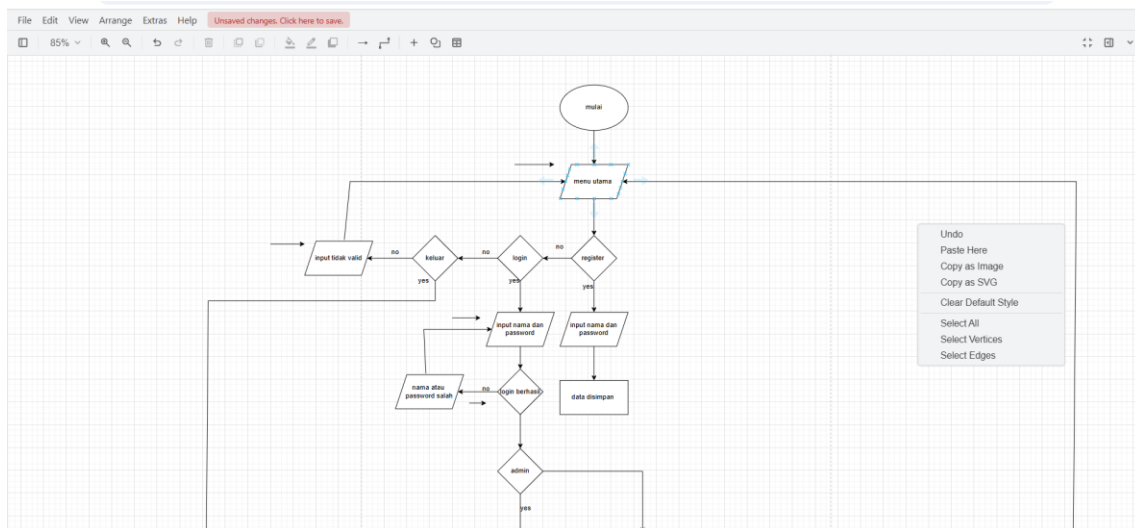
LAPORAN PRAKTIKUM
POSTTEST (8)
ALGORITMA PEMROGRAMAN DASAR



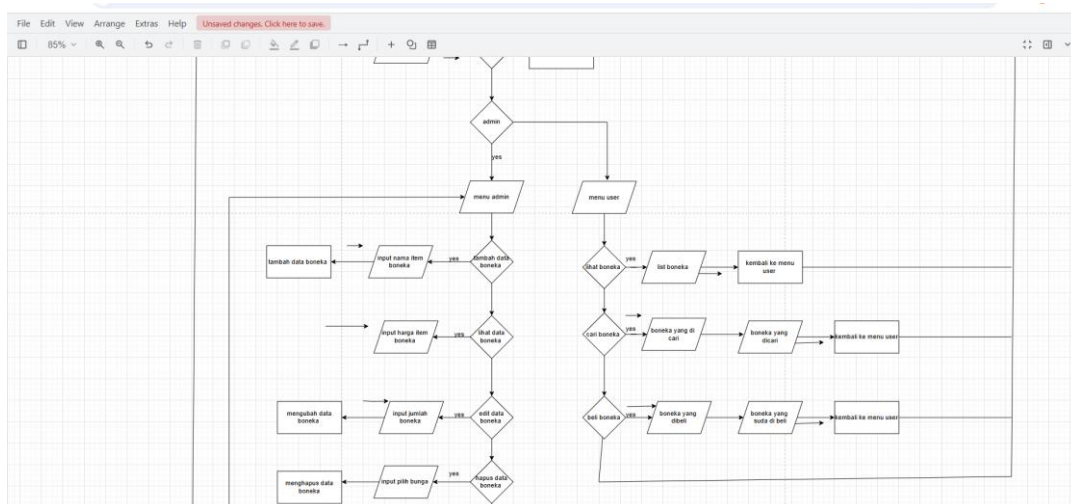
Disusun oleh:
Nama Kirana Cinta Mentari
(2509106106)
Kelas (C1 '25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Gambar flowchart 1.1



Gambar flowchart 1,2

15. dan inputan username dan password tersimpan

16. dan kembali ke menu

17. dan ketika pengguna masuk ke login dengan menggunakan username dan password, yang telah di inputkan pengguna

18. pengguna hanya bisa melihat data

19. dan ketika pengguna keluar, maka akan kembali ke menu utama

20. dan pengguna keluar juga di menu utama

21. selesai

2. Deskripsi Singkat Program

Tujuan program

Program ini dibuat untuk mengelola data stok barang pada sebuah toko boneka. Sistemnya memakai login, sehingga hanya pengguna yang terdaftar bisa masuk. Ada dua jenis akses, yaitu admin dan user. Admin memiliki kendali penuh atas data stok, sedangkan user hanya bisa melihat data.

Fungsi dan manfaat utama program

Sistem Akun (Login & Register)

- *Program menyimpan data akun dalam list akun.*
- *Pengguna bisa membuat akun baru (otomatis jadi user).*
- *Admin bawaan adalah “Kirana Cinta Mentari” dengan password “admin123”.*
- *Login menggunakan username dan password yang dicek satu per satu.*

Hak Akses Berdasarkan Peran

Setelah login, program membedakan menu untuk:

- *Admin (punya kontrol penuh atas stok)*
- *User biasa (hanya menampilkan data)*

Pengelolaan Data Stok

Data stok disimpan dalam list stok, dengan format: [nama, harga, jumlah]

Admin bisa melakukan:

1. *Tambah data Memasukkan nama, harga, dan jumlah barang baru.*
2. *Lihat data Menampilkan semua stok yang tersimpan.*
3. *Edit data Memilih barang berdasarkan nomor lalu mengganti isinya.*
4. *Hapus data Menghapus data stok berdasarkan nomor.*

Menu Khusus untuk User

User hanya mendapatkan dua pilihan:

- *Lihat daftar barang.*
- *Keluar dari menu*

3. Source Code

```
# data.py

akun = {
    "Kirana Cinta Mentari": {"password": "admin123", "role": "admin"},
    "Bunga": {"password": "user123", "role": "user"}
}

stok = {
    "Boneka Teddy": 10,
    "Boneka Doraemon": 8,
    "Boneka Hello Kitty": 5
}

keranjang = {}
```

```
# main.py
import os
from auth import login, daftar

def main():
    while True:
        os.system("cls" if os.name == "nt" else "clear")
        print("=== TOKO BONEKA ===")
        print("1. Masuk")
        print("2. Daftar")
        print("0. Keluar")
        pilihan = input("Pilih: ")

        if pilihan == "1":
            login()
```

```

        elif pilihan == "2":
            daftar()
        elif pilihan == "0":
            print("Terima kasih sudah datang ke Toko Boneka.")
            break
        else:
            print("Pilihan tidak valid.")
            input("Tekan Enter untuk lanjut...")

if __name__ == "__main__":
    main()

# data.py

akun = {
    "Kirana Cinta Mentari": {"password": "admin123", "role": "admin"},
    "Bunga": {"password": "user123", "role": "user"}
}

stok = {
    "Boneka Teddy": 10,
    "Boneka Doraemon": 8,
    "Boneka Hello Kitty": 5
}

keranjang = {}

# stok.py
from prettytable import PrettyTable
from data import stok

def tampilkan_stok():
    print("\n=== DAFTAR STOK BONEKA ===")
    tabel = PrettyTable(["Nama Boneka", "Jumlah (pcs)"])
    for nama, jumlah in stok.items():
        tabel.add_row([nama, jumlah])
    print(tabel)

def tambah_stok(nama_boneka, jumlah):
    if nama_boneka in stok:
        stok[nama_boneka] += jumlah
    else:
        stok[nama_boneka] = jumlah
    print(f"{nama_boneka} berhasil ditambah. Sekarang stoknya: {stok[nama_boneka]} pcs")

```

```

# transaksi.py
from prettytable import PrettyTable
from data import stok, keranjang

def beli_boneka(nama_boneka, jumlah):
    if nama_boneka in stok and stok[nama_boneka] >= jumlah:
        stok[nama_boneka] -= jumlah
        keranjang[nama_boneka] = keranjang.get(nama_boneka, 0) + jumlah
        print(f"Berhasil membeli {jumlah} {nama_boneka}")
    else:
        print("Stok tidak cukup atau boneka tidak ditemukan.")

def lihat_keranjang():
    print("\n=== KERANJANG ANDA ===")
    if keranjang:
        tabel = PrettyTable(["Nama Boneka", "Jumlah"])
        for nama, jumlah in keranjang.items():
            tabel.add_row([nama, jumlah])
        print(tabel)
    else:
        print("Keranjang masih kosong.")

# auth.py
import os
from data import akun
from stok import tampilkan_stok, tambah_stok
from transaksi import beli_boneka, lihat_keranjang

def tampilkan_menu():
    print("\n1. Lihat Stok")
    print("2. Tambah Stok (Admin)")
    print("3. Beli Boneka (User)")
    print("4. Lihat Keranjang")
    print("5. Logout")

def login():
    os.system("cls" if os.name == "nt" else "clear")
    print("=== LOGIN ===")
    username = input("Masukkan nama pengguna: ")
    password = input("Masukkan password: ")

    if username in akun and akun[username]["password"] == password:
        print(f"\nSelamat datang, {username}.")
        input("Tekan Enter untuk lanjut...")
        menu_utama(username)
    else:

```



```

        print("\nLogin gagal. Nama atau password salah.")
        input("Tekan Enter untuk kembali...")

def daftar():
    os.system("cls" if os.name == "nt" else "clear")
    print("=== DAFTAR AKUN BARU ===")
    nama = input("Masukkan nama pengguna baru: ")
    if nama in akun:
        print("Nama sudah digunakan. Coba yang lain.")
        input("Tekan Enter untuk kembali...")
        return
    pw = input("Masukkan password: ")
    akun[nama] = {"password": pw, "role": "user"}
    print("Akun berhasil dibuat. Silakan login.")
    input("Tekan Enter untuk lanjut...")

def menu_utama(username):
    while True:
        os.system("cls" if os.name == "nt" else "clear")
        print(f"=== MENU UTAMA ({username}) ===")
        tampilkan_menu()

        pilihan = input("Pilih menu: ")

        try:
            if pilihan == "1":
                tampilkan_stok()
            elif pilihan == "2":
                if akun[username]["role"] == "admin":
                    nama = input("Masukkan nama boneka: ")
                    jumlah = int(input("Masukkan jumlah tambahan: "))
                    tambah_stok(nama, jumlah)
                else:
                    print("Anda bukan admin, tidak dapat menambah
stok.")
            elif pilihan == "3":
                if akun[username]["role"] == "user":
                    nama = input("Masukkan nama boneka: ")
                    jumlah = int(input("Masukkan jumlah yang ingin
dibeli: "))
                    beli_boneka(nama, jumlah)
                else:
                    print("Admin tidak dapat membeli boneka.")
            elif pilihan == "4":
                lihat_keranjang()
            elif pilihan == "5":

```

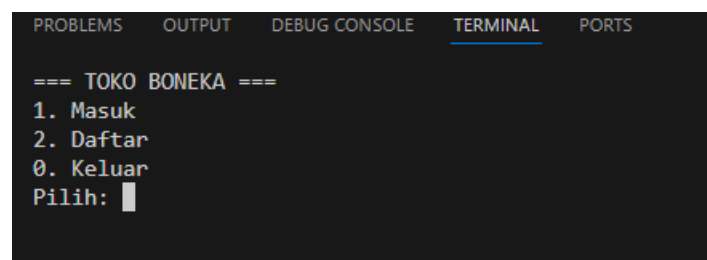
```

        print("Logout berhasil.")
        input("Tekan Enter untuk kembali ke menu awal...")
        break
    else:
        print("Pilihan tidak valid.")
except ValueError:
    print("Input harus berupa angka.")

input("\nTekan Enter untuk melanjutkan...")

```

4. Hasil Output



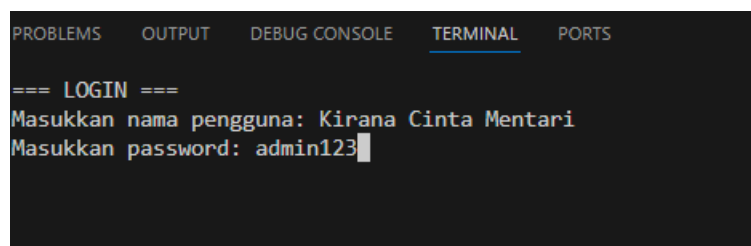
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

=== TOKO BONEKA ===
1. Masuk
2. Daftar
0. Keluar
Pilih: 

```

Gambar 1.1 Tampilan menu awal



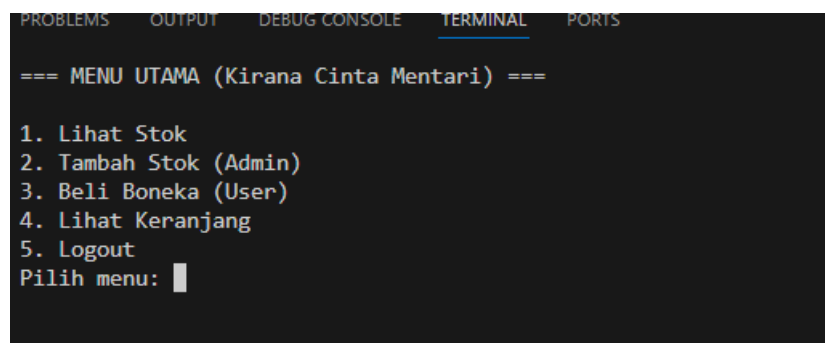
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

=== LOGIN ===
Masukkan nama pengguna: Kirana Cinta Mentari
Masukkan password: admin123

```

Gambar 1.2 Tampilan login



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

=== MENU UTAMA (Kirana Cinta Mentari) ===
1. Lihat Stok
2. Tambah Stok (Admin)
3. Beli Boneka (User)
4. Lihat Keranjang
5. Logout
Pilih menu: 

```

Gambar 1.3 Tampilan menu utama

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

=== MENU UTAMA (Kirana Cinta Mentari) ===

1. Lihat Stok
2. Tambah Stok (Admin)
3. Beli Boneka (User)
4. Lihat Keranjang
5. Logout
Pilih menu: 1

=== DAFTAR STOK BONEKA ===
- Boneka Teddy: 10 pcs
- Boneka Doraemon: 8 pcs
- Boneka Hello Kitty: 5 pcs

Tekan Enter untuk melanjutkan...
```

Gambar 1.4 daftar stok boneka

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

=== MENU UTAMA (Kirana Cinta Mentari) ===

1. Lihat Stok
2. Tambah Stok (Admin)
3. Beli Boneka (User)
4. Lihat Keranjang
5. Logout
Pilih menu: 2
Masukkan nama boneka: boneka tung tung sahur
Masukkan jumlah tambahan: 10
boneka tung tung sahur berhasil ditambah! Jumlah sekarang: 10 pcs

Tekan Enter untuk melanjutkan...
```

Gambar 1.5 tambah stok boneka

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
=== MENU UTAMA (Kirana Cinta Mentari) ===

1. Lihat Stok
2. Tambah Stok (Admin)
3. Beli Boneka (User)
4. Lihat Keranjang
5. Logout
Pilih menu: 3
Admin tidak bisa membeli boneka.

Tekan Enter untuk melanjutkan...
```

Gambar 1.6 beli boneka

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
=== MENU UTAMA (Kirana Cinta Mentari) ===

1. Lihat Stok
2. Tambah Stok (Admin)
3. Beli Boneka (User)
4. Lihat Keranjang
5. Logout
Pilih menu: 4

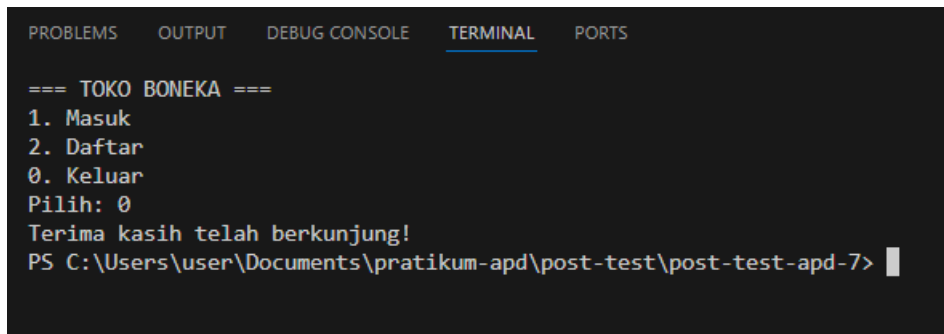
=== KERANJANG ANDA ===
Keranjang kosong.

Tekan Enter untuk melanjutkan...
```

Gamabr 1.7 lihat keranjang

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
=== DAFTAR AKUN BARU ===
Masukkan nama pengguna baru: cinte
Masukkan password: 122
```

Gambar 1.8 daftar akun baru



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

=== TOKO BONEKA ===
1. Masuk
2. Daftar
0. Keluar
Pilih: 0
Terima kasih telah berkunjung!
PS C:\Users\user\Documents\pratikum-apd\post-test\post-test-apd-7>
```

Gambar 1.9 keluar

5. Langkah-langkah GIT

5.1 GIT Add

```
PS C:\partikum-apd\pratikum-apd> git push -u origin main
```

Gambar 5.1 Langkah Git

Perintah git add . digunakan untuk menambahkan semua perubahan file yang ada di dalam folder proyek ke dalam staging area Git. Staging area adalah tempat sementara di mana perubahan file disiapkan sebelum benar-benar disimpan ke dalam riwayat repository melalui perintah git commit

5.2 GIT Commit

```
PS D:\Documents\nahh itu\praktikum apd postest 2> git add .
PS D:\Documents\nahh itu\praktikum apd postest 2> git commit
```

Gambar 5.2 Langkah Git

Commit dalam Git dapat diibaratkan seperti menyimpan catatan atau rekaman atas perubahan yang telah dilakukan pada proyek. Git commit berfungsi untuk menyimpan (merekam) snapshot atau perubahan pada kode atau file di repository Git. Saat kamu melakukan commit, Git akan menyimpan semua perubahan yang sudah kamu staging (dimasukkan ke area staging) dalam sebuah commit object

5. GIT Push

```
PS D:\Documents\nahh itu\praktikum apd postest 2> git push
```

Gambar 5.3 Langkah Git

Git push adalah perintah yang digunakan untuk mengirimkan perubahan atau hasil kerja dari komputer Anda (repository lokal) ke penyimpanan Git yang ada di internet atau server (repository remote), seperti GitHub atau GitLab. Setelah Anda melakukan perubahan dan menyimpannya secara lokal, perintah ini berfungsi agar perubahan tersebut dapat tersimpan secara online dan dapat diakses oleh orang lain yang memiliki akses ke repository tersebut