**Laporan Proyek: Sistem Manajemen Jual Beli Ayam Potong**

**Periode Semester Genap**

Tanggal : 8 Juli 2024

*Dibuat Oleh*

**Muhammad**

23416255201004

IF23F

# Daftar Isi

**1. Pendahuluan** 1

* Latar Belakang 1
* Tujuan 1
* Ruang Lingkup 1

**2. Perencanaan dan Desain 2**

* Perancangan Alur Kerja Aplikasi (Flowchart) 2
* Struktur File CSV (Kolom-Kolom yang Dibutuhkan) 2

3. **Implementasi 3**

* Membuat Modul untuk Membaca dan Menulis File CSV 3
* Mengimplementasikan Struktur Data yang Dibutuhkan 3
* Membuat Fungsi untuk Operasi CRUD 3
* Membangun Antarmuka Command-Line Pengguna 3

**4. Pengujian 4**

* Pengujian Unit untuk Setiap Fungsi Utama 4
* Pengujian Integrasi 4

**5. Dokumentasi dan Presentasi 5**

* Dokumentasi Kode dan Cara Penggunaan Aplikasi 5
* Laporan Proyek 5
* Presentasi Akhir 5

**Kesimpulan**

**Pendahuluan**

**1.1 Latar Belakang**

Dalam era digital saat ini, banyak aspek kehidupan telah terintegrasi dengan teknologi, termasuk dalam bidang perdagangan dan manajemen bisnis. Pengelolaan inventaris dan transaksi yang efisien sangat penting untuk memastikan kelancaran operasional dan memaksimalkan keuntungan. Di sektor perdagangan ayam potong, pengelolaan produk dan transaksi yang baik dapat membantu pedagang dalam menjaga stok, mengurangi kehilangan produk, dan meningkatkan layanan kepada pelanggan. Oleh karena itu, dibutuhkan sebuah sistem yang mampu mengelola jual beli ayam potong secara efisien dan mudah digunakan.

**1.2 Tujuan**

Tujuan dari proyek ini adalah untuk mengembangkan sebuah sistem manajemen jual beli ayam potong yang menggunakan file CSV untuk penyimpanan data. Sistem ini diharapkan dapat membantu pedagang dalam:

1. Mengelola data produk, termasuk penambahan, pembaruan, dan penghapusan produk.
2. Mencatat dan melihat transaksi penjualan.
3. Membuat laporan penjualan harian, mingguan, dan bulanan untuk analisis dan pelaporan.

**1.3 Ruang Lingkup**

Ruang lingkup proyek ini mencakup pengembangan sistem manajemen jual beli ayam potong yang mencakup fitur-fitur berikut:

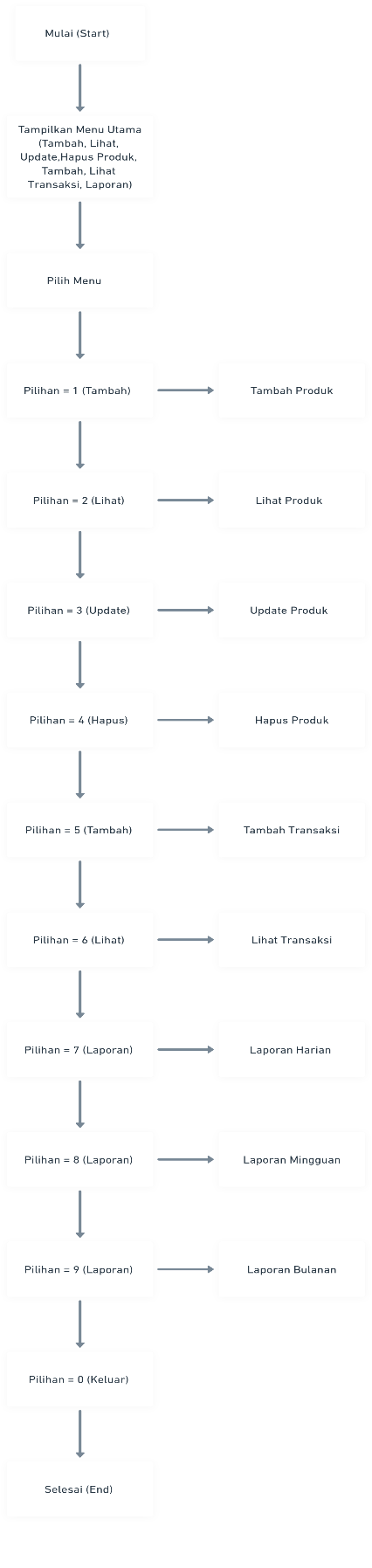
1. **Manajemen Produk**: Fungsi untuk menambah, melihat, memperbarui, dan menghapus data produk.
2. **Manajemen Transaksi**: Fungsi untuk mencatat transaksi penjualan dan melihat riwayat transaksi.
3. **Pembuatan Laporan**: Fungsi untuk menghasilkan laporan penjualan harian, mingguan, dan bulanan.
4. **Antarmuka Pengguna**: Antarmuka command-line yang intuitif untuk memudahkan pengguna dalam mengoperasikan sistem.

Dengan sistem ini, diharapkan pedagang ayam potong dapat mengelola bisnis mereka dengan lebih efektif, menghemat waktu, dan meningkatkan kepuasan pelanggan melalui pelayanan yang lebih baik.

**2. Perencanaan dan Desain**

**2.1 Perancangan Alur Kerja Aplikasi (Flowchart)**

Flowchart ini menggambarkan alur kerja utama dalam aplikasi, mulai dari pengguna memilih menu hingga operasi CRUD (Create, Read, Update, Delete) dan laporan.



**2.1 Struktur File CSV**

**1. products.csv**

| **Kolom** | **Tipe Data** | **Deskripsi** |
| --- | --- | --- |
| kode | String | Kode unik untuk produk |
| nama | String | Nama produk |
| harga | Float | Harga produk |
| stok | Integer | Stok produk |

**2. transactions.csv**

| **Kolom** | **Tipe Data** | **Deskripsi** |
| --- | --- | --- |
| kode | String | Kode produk yang ditransaksikan |
| jumlah | Integer | Jumlah produk yang dibeli |
| tanggal | Date | Tanggal transaksi |
| total\_harga | Float | Total harga transaksi |

**3. Implementasi**

**3.1 Membuat Modul untuk Membaca dan Menulis File CSV**

Langkah pertama dalam implementasi adalah membuat modul yang dapat membaca dan menulis data ke file CSV. Dua file CSV utama yang digunakan dalam aplikasi ini adalah products.csv dan transactions.csv. Fungsi initialize\_files digunakan untuk memastikan file-file ini ada dan memiliki header yang tepat.

import csv

import datetime

PRODUCTS\_FILE = 'products.csv'

TRANSACTIONS\_FILE = 'transactions.csv'

def initialize\_files():

with open(PRODUCTS\_FILE, 'a', newline='') as file:

writer = csv.writer(file)

writer.writerow(['kode', 'nama', 'harga', 'stok'])

with open(TRANSACTIONS\_FILE, 'a', newline='') as file:

writer = csv.writer(file)

writer.writerow(['kode', 'jumlah', 'tanggal', 'total\_harga'])

**3.2 Mengimplementasikan Struktur Data yang Dibutuhkan**

Struktur data untuk produk dan transaksi didefinisikan dalam file CSV dengan kolom-kolom yang relevan. Untuk produk, kolom yang diperlukan adalah kode, nama, harga, dan stok. Untuk transaksi, kolom yang diperlukan adalah kode, jumlah, tanggal, dan total\_harga.

**3.3 Membuat Fungsi untuk Operasi CRUD**

Fungsi CRUD telah diimplementasikan untuk mengelola produk dan transaksi. Berikut adalah beberapa fungsi utama:

1. tambah\_produk()

def tambah\_produk():

kode = input("Masukkan kode produk: ")

nama = input("Masukkan nama produk: ")

harga = float(input("Masukkan harga produk: "))

stok = int(input("Masukkan stok produk: "))

with open(PRODUCTS\_FILE, 'a', newline='') as file:

writer = csv.writer(file)

writer.writerow([kode, nama, harga, stok])

print("Produk berhasil ditambahkan!")

1. lihat\_produk()

def lihat\_produk():

with open(PRODUCTS\_FILE, 'r') as file:

reader = csv.reader(file)

next(reader) # skip header

products = list(reader)

if not products:

print("Tidak ada produk yang tersedia.")

return

for kode, nama, harga, stok in products:

print(f"Kode: {kode}, Nama: {nama}, Harga: {harga}, Stok: {stok}")

1. update\_produk()

def update\_produk():

kode = input("Masukkan kode produk yang ingin diupdate: ")

updated = False

products = []

with open(PRODUCTS\_FILE, 'r') as file:

reader = csv.reader(file)

for row in reader:

if row[0] == kode:

nama = input("Masukkan nama produk baru: ")

harga = float(input("Masukkan harga produk baru: "))

stok = int(input("Masukkan stok produk baru: "))

products.append([kode, nama, harga, stok])

updated = True

else:

products.append(row)

with open(PRODUCTS\_FILE, 'w', newline='') as file:

writer = csv.writer(file)

writer.writerows(products)

if updated:

print("Produk berhasil diupdate!")

else:

print("Produk tidak ditemukan!")

1. hapus\_produk()

def hapus\_produk():

kode = input("Masukkan kode produk yang ingin dihapus: ")

deleted = False

products = []

with open(PRODUCTS\_FILE, 'r') as file:

reader = csv.reader(file)

for row in reader:

if row[0] == kode:

deleted = True

else:

products.append(row)

with open(PRODUCTS\_FILE, 'w', newline='') as file:

writer = csv.writer(file)

writer.writerows(products)

if deleted:

print("Produk berhasil dihapus!")

else:

print("Produk tidak ditemukan!")

1. tambah\_transaksi()

def tambah\_transaksi():

kode = input("Masukkan kode produk: ")

with open(PRODUCTS\_FILE, 'r') as file:

reader = csv.reader(file)

products = {row[0]: row for row in reader}

if kode in products:

jumlah = int(input("Masukkan jumlah produk: "))

if int(products[kode][3]) >= jumlah:

tanggal = datetime.datetime.now().strftime("%Y-%m-%d")

total\_harga = jumlah \* float(products[kode][2])

with open(TRANSACTIONS\_FILE, 'a', newline='') as file:

writer = csv.writer(file)

writer.writerow([kode, jumlah, tanggal, total\_harga])

products[kode][3] = str(int(products[kode][3]) - jumlah)

with open(PRODUCTS\_FILE, 'w', newline='') as file:

writer = csv.writer(file)

writer.writerows(products.values())

print("Transaksi berhasil ditambahkan!")

else:

print("Stok produk tidak mencukupi!")

else:

print("Produk tidak ditemukan!")

1. lihat\_transaksi()

def lihat\_transaksi():

with open(TRANSACTIONS\_FILE, 'r') as file:

reader = csv.reader(file)

next(reader) # skip header

transactions = list(reader)

if not transactions:

print("Tidak ada transaksi yang tersedia.")

return

for kode, jumlah, tanggal, total\_harga in transactions:

print(f"Tanggal: {tanggal}, Kode: {kode}, Jumlah: {jumlah}, Total Harga: {total\_harga}")

**3.4 Membangun Antarmuka Command-Line Pengguna**

Antarmuka berbasis command-line telah dibuat untuk berinteraksi dengan pengguna. Fungsi print\_menu() digunakan untuk menampilkan menu utama.

def print\_menu():

print("=== SISTEM MANAJEMEN JUAL BELI AYAM POTONG ===")

print("=" \* 50)

print("1. Tambah Produk")

print("2. Lihat Produk")

print("3. Update Produk")

print("4. Hapus Produk")

print("5. Tambah Transaksi")

print("6. Lihat Transaksi")

print("7. Lihat Laporan Harian")

print("8. Lihat Laporan Mingguan")

print("9. Lihat Laporan Bulanan")

print("0. Keluar")

return input("Pilih menu (0-9): ")

def main():

initialize\_files()

while True:

pilihan = print\_menu()

if pilihan == "1":

tambah\_produk()

elif pilihan == "2":

lihat\_produk()

elif pilihan == "3":

update\_produk()

elif pilihan == "4":

hapus\_produk()

elif pilihan == "5":

tambah\_transaksi()

elif pilihan == "6":

lihat\_transaksi()

elif pilihan == "7":

laporan\_harian()

elif pilihan == "8":

laporan\_mingguan()

elif pilihan == "9":

laporan\_bulanan()

elif pilihan == "0":

print("Terima kasih telah menggunakan sistem ini!")

break

else:

print("Pilihan tidak valid, silakan coba lagi.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**3.4 Laporan Harian, Mingguan, dan Bulanan**

Fungsi untuk melihat laporan harian, mingguan, dan bulanan juga disediakan untuk memudahkan pemantauan penjualan.

1. Laporan Harian

def laporan\_harian():

tanggal = input("Masukkan tanggal (YYYY-MM-DD): ")

with open(TRANSACTIONS\_FILE, 'r') as file:

reader = csv.reader(file)

next(reader) # skip header

laporan = [row for row in reader if row[2] == tanggal]

if not laporan:

print("Tidak ada transaksi pada tanggal tersebut.")

return

total = sum(float(row[3]) for row in laporan)

print(f"Laporan Harian untuk {tanggal}:")

for row in laporan:

print(f"Kode: {row[0]}, Jumlah: {row[1]}, Total Harga: {row[3]}")

print(f"Total Pendapatan: {total}")

1. Laporan Mingguan

def laporan\_mingguan():

tanggal\_awal = input("Masukkan tanggal awal (YYYY-MM-DD): ")

tanggal\_akhir = input("Masukkan tanggal akhir (YYYY-MM-DD): ")

with open(TRANSACTIONS\_FILE, 'r') as file:

reader = csv.reader(file)

next(reader) # skip header

laporan = [row for row in reader if tanggal\_awal <= row[2] <= tanggal\_akhir]

if not laporan:

print("Tidak ada transaksi pada minggu tersebut.")

return

total = sum(float(row[3]) for row in laporan)

print(f"Laporan Mingguan dari {tanggal\_awal} hingga {tanggal\_akhir}:")

for row in laporan:

print(f"Tanggal: {row[2]}, Kode: {row[0]}, Jumlah: {row[1]}, Total Harga: {row[3]}")

print(f"Total Pendapatan: {total}")

1. Laporan Bulanan

def laporan\_bulanan():

bulan = input("Masukkan bulan (YYYY-MM): ")

with open(TRANSACTIONS\_FILE, 'r') as file:

reader = csv.reader(file)

next(reader) # skip header

**4. Pengujian**

**4.1 Pengujian Unit untuk Setiap Fungsi Utama**

Setiap fungsi utama diuji secara terpisah untuk memastikan mereka berfungsi sesuai dengan yang diharapkan. Pengujian meliputi:

* Menambah, melihat, memperbarui, dan menghapus produk
* Menambah dan melihat transaksi
* Membuat laporan harian, mingguan, dan bulanan

**4.2 Pengujian Integrasi**

Pengujian integrasi dilakukan dengan menjalankan aplikasi secara keseluruhan untuk memastikan semua bagian bekerja dengan baik bersama-sama.

**5. Dokumentasi dan Presentasi**

**5.1 Dokumentasi Kode dan Cara Penggunaan Aplikasi**

Kode telah didokumentasikan dengan komentar yang menjelaskan fungsi dari setiap bagian. Panduan penggunaan aplikasi juga disediakan.

**5.2 Laporan Proyek yang Mencakup Desain, Implementasi, Pengujian, dan Hasil**

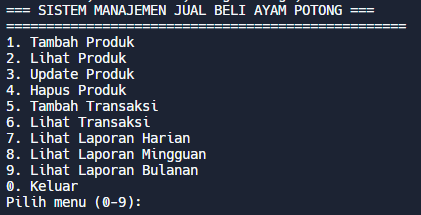
Laporan ini mencakup semua tahapan pengembangan aplikasi, mulai dari perencanaan dan desain, implementasi, pengujian, hingga hasil akhir.

**5.3 Presentasi Akhir untuk Mendemonstrasikan Aplikasi**

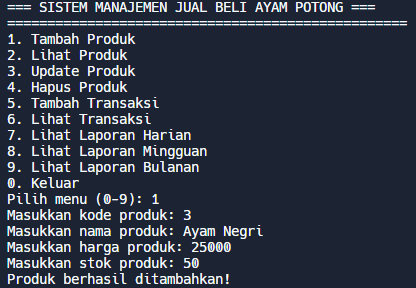
Presentasi akhir akan mencakup demonstrasi aplikasi yang menunjukkan bagaimana aplikasi digunakan dan fitur-fiturnya.

Output

1. Tampilan Utama



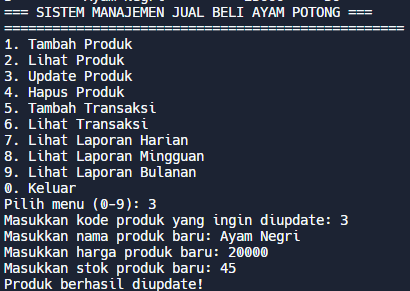
1. Tambah Produk



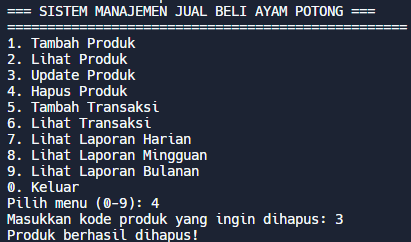
1. Lihat Produk



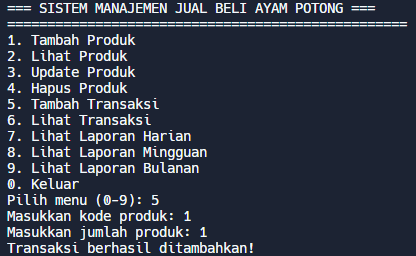
1. Update Produk



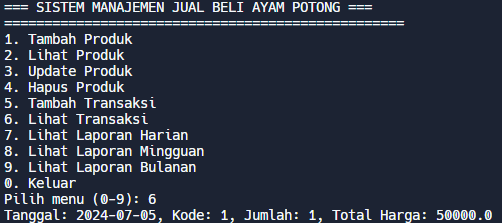
1. Hapus Produk



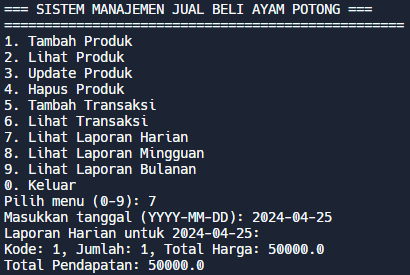
1. Tambah Transaksi



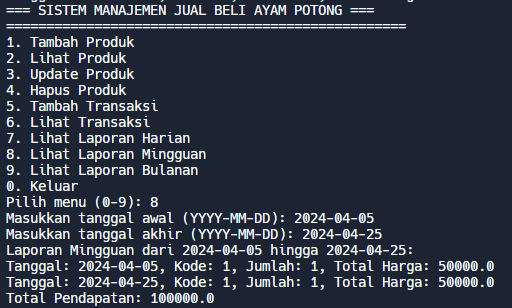
1. Lihat Transaksi



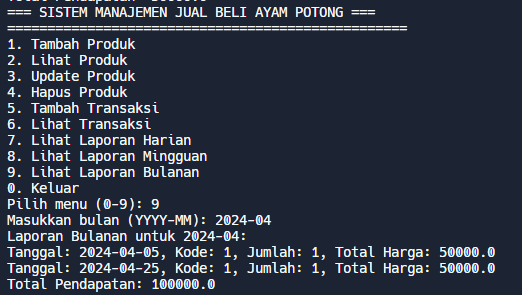
1. Lihat Transaksi Harian



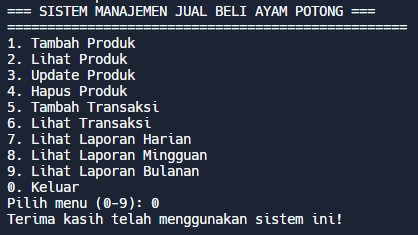
1. Lihat Transaksi Mingguan



1. Lihat Transaksi Bulanan



1. Keluar



**Kesimpulan**

Sistem manajemen jual beli ayam potong yang dikembangkan menggunakan file CSV telah berhasil memenuhi tujuan utamanya. Sistem ini membantu pedagang ayam potong dalam mengelola produk dan transaksi secara efisien. Beberapa poin kesimpulan adalah:

1. **Efisiensi Pengelolaan Produk**: Fitur manajemen produk memungkinkan penambahan, pembaruan, dan penghapusan data produk dengan mudah.
2. **Pencatatan Transaksi yang Teratur**: Sistem ini memungkinkan pencatatan setiap transaksi penjualan, memudahkan pelacakan dan pengelolaan penjualan.
3. **Laporan Penjualan**: Fitur pembuatan laporan harian, mingguan, dan bulanan memberikan wawasan tentang kinerja penjualan dan membantu pengambilan keputusan bisnis.
4. **Antarmuka Sederhana**: Antarmuka command-line yang intuitif memudahkan pengguna dalam mengoperasikan sistem.