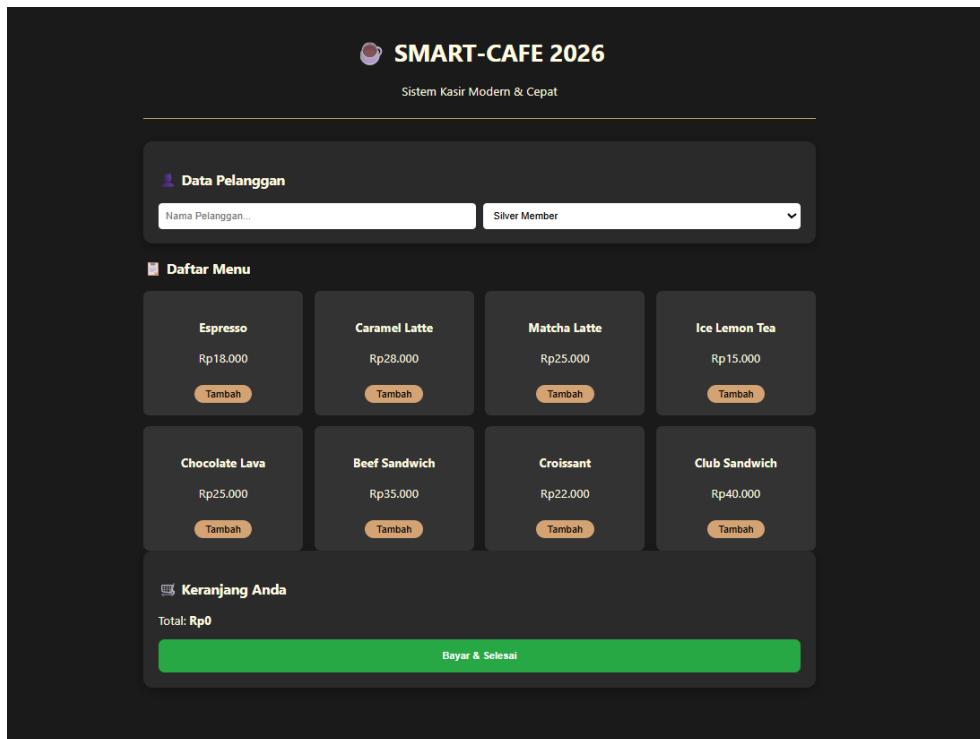


TUGAS PROJEK

SISTEM MANAJEMEN TRANSAKSI SMART-CAFE BERBASIS LOYALTY POINT MEMBER

Disusun untuk memenuhi tugas projek Pemrograman Berorientasi Objek

Dosen Pengampu : Ali Tarmuji, S.T., M.Cs.



Disusun oleh:

NAMA : Bima Ghiramustika Putra
NIM : 2400018112
KELAS : B

LINK GITHUB :

https://github.com/Bimaghira/PBO2025-TP02-BIMA_GHIRAMUSTIKA_PUTRA-2400018112

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS AHMAD DAHLAN

2025/2026

Pendahuluan

1.1 Latar Belakang

Di era digital tahun 2026, efisiensi pelayanan di sektor kuliner telah menjadi standar utama bagi kepuasan pelanggan. Banyak kafe saat ini masih menghadapi tantangan dalam mengelola antrean dan pencatatan transaksi yang cepat, terutama ketika harus menangani variasi menu makanan dan minuman yang beragam secara manual. Ketidakakuratan dalam pencatatan pesanan dan perhitungan total belanja tidak hanya merugikan dari sisi finansial, tetapi juga dapat menurunkan kepercayaan pelanggan terhadap profesionalisme pengelolaan kafe.

Selain aspek kecepatan transaksi, strategi retensi pelanggan melalui program loyalitas menjadi faktor krusial dalam persaingan bisnis modern. Sistem konvensional seringkali gagal dalam mengintegrasikan perhitungan poin member secara real-time, sehingga pemberian apresiasi kepada pelanggan tetap (seperti tipe Silver dan Gold) menjadi tidak konsisten. Oleh karena itu, pengembangan aplikasi SMART-CAFE 2026 dirancang untuk mendigitalisasi proses transaksi sekaligus mengotomatisasi manajemen poin menggunakan prinsip Pemrograman Berorientasi Objek, guna menciptakan layanan yang lebih transparan dan terorganisir.

1.2 Rumusan Masalah

1. Bagaimana cara mempercepat proses input transaksi makanan dan minuman secara akurat?
2. Bagaimana mengimplementasikan sistem loyalitas poin yang berbeda antara tipe member Silver dan Gold secara otomatis?
3. Bagaimana cara menghasilkan bukti transaksi (struk) yang dapat tersimpan secara permanen?

1.3 Tujuan Penelitian

1. Membangun aplikasi kasir berbasis Java yang mampu menangani transaksi banyak produk sekaligus.
2. Menerapkan konsep Pemrograman Berorientasi Objek (OOP) untuk membedakan perhitungan poin pelanggan.
3. Menyediakan fitur penyimpanan struk otomatis ke dalam format file teks (.txt).

1.4 Manfaat Penelitian

1. Bagi Pengelola: Memudahkan rekapitulasi penjualan dan pengelolaan data pelanggan.
2. Bagi Pelanggan: Mendapatkan kepastian perhitungan poin yang transparan sesuai tipe keanggotaan.
3. Bagi Pengembang: Menguji efektivitas konsep inheritance dan polymorphism dalam kasus nyata.

1.5 Batasan Masalah

1. Aplikasi berjalan pada antarmuka baris perintah (CLI/Console).
2. Data produk (master data) bersifat statis di dalam kode (hardcoded).
3. Belum tersedia basis data (database) eksternal; penyimpanan data hanya berupa file teks per transaksi.

A.Ruang Lingkup

1. Input nama pelanggan dan pemilihan tipe member (Silver/Gold).
2. Pemilihan menu dari daftar yang tersedia (4 minuman dan 4 makanan).
3. Manajemen keranjang (tambah dan hapus item).
4. Perhitungan total biaya dan poin member.
5. Cetak struk ke layar dan file .txt.

B. Rancangan Aplikasi

Aplikasi dibangun menggunakan bahasa Java dengan arsitektur modular. Komponen utama meliputi:

- Produk: Sebagai kelas induk untuk item yang dijual.
- Member: Sebagai kelas abstrak untuk manajemen pelanggan.
- Main Logic: Mengatur alur transaksi menggunakan ArrayList sebagai penampung sementara (keranjang).

1.Rancangan Pengguna User

1. Kasir (Administrator/Operator)

- Peran: Pemegang kendali penuh atas jalannya aplikasi.
- Aksi:
 - Menginput identitas pelanggan dan menentukan tipe member (Silver/Gold).
 - Melakukan input pesanan ke dalam sistem berdasarkan permintaan pelanggan (Menu 1).
 - Melakukan koreksi atau penghapusan item jika pelanggan membatalkan pesanan (Menu 2).
 - Mengeksekusi pencetakan struk untuk mengakhiri transaksi (Menu 3).

2. Pelanggan (Viewer)

- Peran: Subjek yang menerima layanan dan poin.
- Aksi:
 - Melakukan pengecekan atau melihat daftar pesanan yang telah diinput oleh kasir melalui tampilan layar "ISI KERANJANG".
 - Memastikan bahwa nama dan tipe member yang dimasukkan sudah sesuai sebelum proses bayar.
 - Melihat rincian perolehan poin dan total belanja pada struk akhir.

2.Rancangan Fungsional Sistem

Secara teknis, fungsionalitas yang tersedia untuk "melihat" pesanan pelanggan adalah sebagai berikut:

- Input Pelanggan : Menangkap nama dan tipe member.
- Pilih Menu : Menampilkan daftar produk beserta harga secara real-time.
- Update Keranjang : Menghapus item dari daftar belanja sebelum dibayar.
- Auto-Point : Menghitung poin secara otomatis berdasarkan total belanja.
- Save Transaction : Mengekspor data belanja ke file teks.

3.Rancangan Struktur Kelas

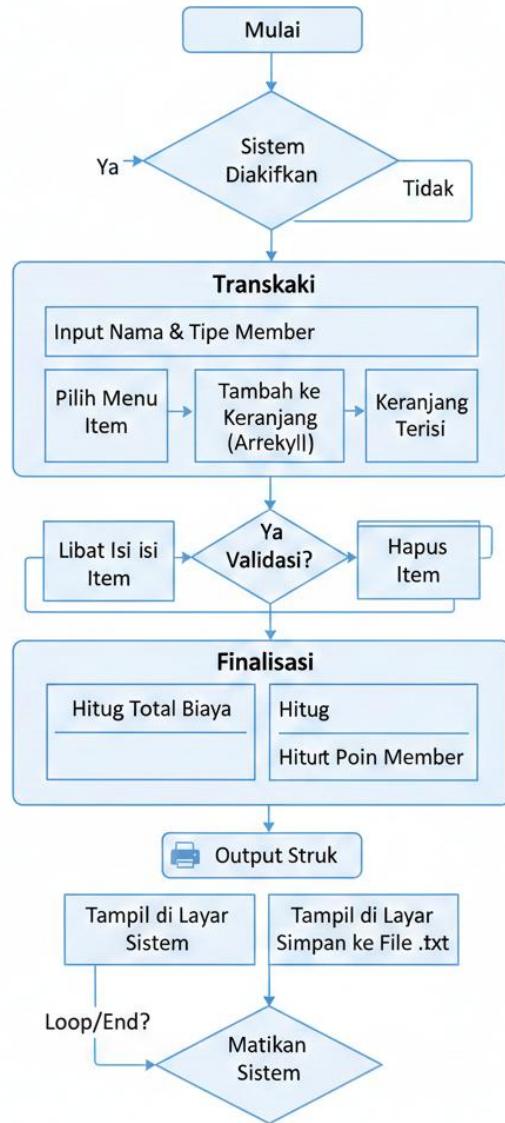
Sistem ini menggunakan prinsip OOP yang terdiri dari:

- Produk (Parent): Memiliki atribut nama dan harga.
- Makanan & Minuman (Children): Mewarisi sifat dari kelas Produk.
- Member (Abstract): Memiliki metode abstrak hitungPoin().
- MemberSilver & MemberGold (Concrete): Mengimplementasikan logika perhitungan poin yang berbeda.

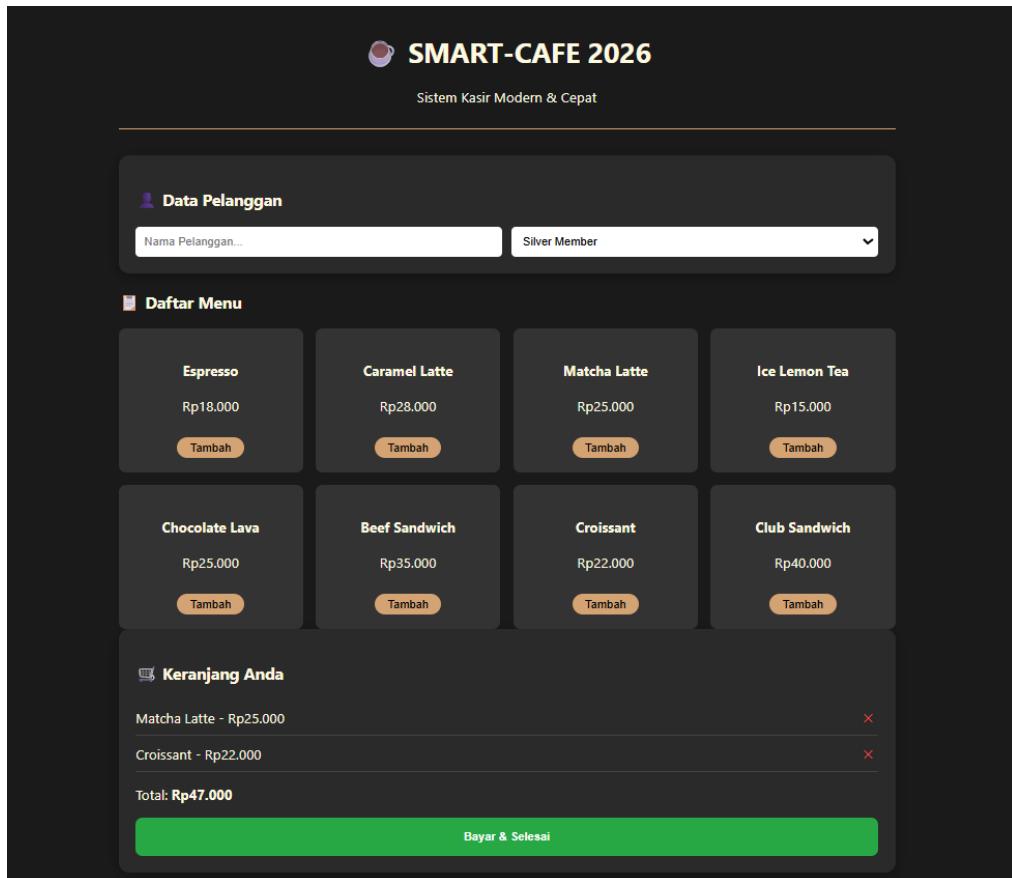
4.Rancangan Alur Aplikasi

1. Start: Sistem diaktifkan.
2. Identifikasi: Input nama pelanggan dan tipe member.
3. Transaksi:
 - User memilih menu.
 - Item masuk ke ArrayList keranjang.
4. Validasi: User dapat melihat isi keranjang atau menghapus item.
5. Finalisasi: Sistem menghitung total biaya dan poin sesuai tipe member.
6. Output: Struk ditampilkan di layar dan disimpan ke file .txt.
7. Loop/End: Kembali ke menu
8. awal atau matikan sistem.

5. Diagram Rancangan Aplikasi



6. Desain Antar Muka Aplikasi



C. Code Program

Main.java

```
import java.util.Scanner;
import java.util.ArrayList;
import java.io.FileWriter;
import java.io.IOException;

public class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        boolean sistemBerjalan = true;

        // Data Produk Master yang diperbanyak
        Produk[] menu = {
            new Minuman("Espresso", 18000),
            new Minuman("Caramel Latte", 28000),
            new Minuman("Matcha Latte", 25000),
            new Minuman("Ice Lemon Tea", 15000),
            new Makanan("Chocolate Lava", 25000),
            new Makanan("Beef Sandwich", 35000),
            new Makanan("Croissant", 22000),
            new Makanan("Club Sandwich", 40000)
        };

        while (sistemBerjalan) { // LOOP UTAMA PELANGGAN
            ArrayList<Produk> keranjang = new ArrayList<>(); // Reset keranjang untuk
            pelanggan baru

            System.out.println("\n=====");
            System.out.println(" WELCOME TO SMART-CAFE 2026 ");
            System.out.println("=====");
            System.out.println("1. Input Transaksi Baru");
            System.out.println("0. Matikan Sistem");
            System.out.print("Pilih: ");
            int opsiSistem = input.nextInt();
            input.nextLine(); // Consume newline

            if (opsiSistem == 0) {
                sistemBerjalan = false;
                System.out.println("Sistem dimatikan. Terima kasih.");
                break;
            }

            System.out.print("Input Nama Pelanggan : ");
            String nama = input.nextLine();
            System.out.print("Tipe Member (1.Silver / 2.Gold): ");
            int tipe = input.nextInt();
```

```

// Polimorfisme Member
Member member = (tipe == 2) ? new MemberGold(nama) : new
MemberSilver(nama);

boolean prosesTransaksi = true;
while (prosesTransaksi) { // LOOP INTERNAL TRANSAKSI
    System.out.println("\n--- DASHBOARD TRANSAKSI: " +
member.getNama().toUpperCase() + " ---");
    System.out.println("1. Tambah Pesanan");
    System.out.println("2. Lihat / Hapus Pesanan");
    System.out.println("3. Bayar & Selesai");
    System.out.print("Pilih menu: ");
    int menuUtama = input.nextInt();

    switch (menuUtama) {
        case 1:
            System.out.println("\nDAFTAR MENU CAFE:");
            for (int i = 0; i < menu.length; i++) {
                System.out.printf("%d. %-15s | Rp%,.0f\n", (i+1), menu[i].getNama(),
menu[i].getHarga());
            }
            System.out.print("Pilih item (1-" + menu.length + "): ");
            int p = input.nextInt();
            if (p >= 1 && p <= menu.length) {
                keranjang.add(menu[p-1]);
                System.out.println("✓ Berhasil dimasukkan.");
            }
        break;

        case 2:
            if (keranjang.isEmpty()) {
                System.out.println("! Keranjang kosong.");
            } else {
                System.out.println("\nKERANJANG " + member.getNama() + ":" );
                for (int i = 0; i < keranjang.size(); i++) {
                    System.out.println((i+1) + ". " + keranjang.get(i).getNama());
                }
                System.out.print("Hapus No (0 batal): ");
                int hapus = input.nextInt();
                if (hapus > 0 && hapus <= keranjang.size()) {
                    keranjang.remove(hapus-1);
                }
            }
        break;

        case 3:
            if (keranjang.isEmpty()) {
                System.out.println("! Tidak ada pesanan untuk dibayar.");
            } else {
                simpanDanTampilkanStruk(member, keranjang);
            }
    }
}

```

```

        prosesTransaksi = false; // Keluar ke menu pelanggan selanjutnya
    }
    break;
}
}
}
input.close();
}

public static void simpanDanTampilkanStruk(Member m, ArrayList<Produk> list) {
    double total = 0;
    StringBuilder isiStruk = new StringBuilder();

    isiStruk.append("\n*****\n");
    isiStruk.append("      SMART-CAFE 2026      \n");
    isiStruk.append("*****\n");
    isiStruk.append("Pelanggan : ").append(m.getNama()).append("\n");
    isiStruk.append("Tipe   : ").append(m.getTipe()).append("\n");
    isiStruk.append("-----\n");

    for (Produk p : list) {
        isiStruk.append(String.format("%-18s : Rp%,10.0f\n", p.getNama(), p.getHarga()));
        total += p.getHarga();
    }

    int perolehanPoin = m.hitungPoin(total);
    m.tambahPoin(perolehanPoin);

    isiStruk.append("-----\n");
    isiStruk.append(String.format("TOTAL BELANJA   : Rp%,10.0f\n", total));
    isiStruk.append("POIN DIDAPAT   : ").append(perolehanPoin).append(" pts\n");
    isiStruk.append("TOTAL SALDO POIN   : ").append(m.getPoin()).append(" pts\n");
    isiStruk.append("*****\n");

    System.out.println(isiStruk.toString());

    try (FileWriter writer = new FileWriter("Struk_" + m.getNama() + ".txt")) {
        writer.write(isiStruk.toString());
        System.out.println("✓ Struk tersimpan: Struk_" + m.getNama() + ".txt");
    } catch (IOException e) {
        System.out.println("! Gagal simpan file.");
    }
}

int perolehanPoin = m.hitungPoin(total);
m.tambahPoin(perolehanPoin);

System.out.println("-----");
System.out.printf("TOTAL BELANJA   : Rp%,10.0f\n", total);

```

```

        System.out.println("-----");
        System.out.println("POIN YANG DIDAPAT : " + perolehanPoin + " pts");
        System.out.println("TOTAL SALDO POIN : " + m.getPoin() + " pts");
        System.out.println("*****");
        System.out.println("    TERIMA KASIH ATAS      ");
        System.out.println("    KUNJUNGAN ANDA!      ");
        System.out.println("*****\n");
    }
}

```

Produk.java

```

class Produk {
    private String nama;
    private double harga;
    public Produk(String n, double h) { this.nama = n; this.harga = h; }
    public String getNama() { return nama; }
    public double getHarga() { return harga; }
}

```

Makanan.java

```

class Makanan extends Produk {
    public Makanan(String nama, double harga) {
        super(nama, harga);
    }
}

// Sub-kelas: Minuman
class Minuman extends Produk {
    public Minuman(String nama, double harga) {
        super(nama, harga);
    }
}

```

Member.java

```
abstract class Member {  
    private String nama;  
    private int poin;  
  
    public Member(String nama) {  
        this.nama = nama;  
        this.poin = 0;  
    }  
  
    public String getNama() { return nama; }  
    public int getPoin() { return poin; }  
    public void tambahPoin(int p) { this.poin += p; }  
  
    public abstract String getTipe();  
    public abstract int hitungPoin(double totalBelanja);  
}
```

MemberSilver.java

```
class MemberSilver extends Member {  
    public MemberSilver(String n) { super(n); }  
    @Override public String getTipe() { return "Silver Member"; }  
    @Override public int hitungPoin(double total) { return (int) (total / 10000); }  
}
```

MemberGold.java

```
class MemberGold extends Member {  
    public MemberGold(String n) { super(n); }  
    @Override public String getTipe() { return "Gold Member"; }  
    @Override public int hitungPoin(double total) { return (int) (total / 10000) * 2; }  
}
```

D. Output Program

```
=====
 WELCOME TO SMART-CAFE 2026
=====
 1. Input Transaksi Baru
 0. Matikan Sistem
 Pilih: 1
```

Pertama tama akan muncul output seperti pada gambar, jika ingin menginputkan transaksi pilih 1

```
Input Nama Pelanggan : BANU
Tipe Member (1.Silver / 2.Gold): 1
```

Setelah itu menginputkan nama pelanggan dan tipe membernya

```
--- DASHBOARD TRANSAKSI: BANU ---
 1. Tambah Pesanan
 2. Lihat / Hapus Pesanan
 3. Bayar & Selesai
Pilih menu: 1
```

Setelah itu akan muncul menu seperti di gambar, jika ingin menambahkan pesanan pilih 1

DAFTAR MENU CAFE:

- | | |
|-------------------|----------|
| 1. Espresso | Rp18.000 |
| 2. Caramel Latte | Rp28.000 |
| 3. Matcha Latte | Rp25.000 |
| 4. Ice Lemon Tea | Rp15.000 |
| 5. Chocolate Lava | Rp25.000 |
| 6. Beef Sandwich | Rp35.000 |
| 7. Croissant | Rp22.000 |
| 8. Club Sandwich | Rp40.000 |

Pilih item (1-8): 3

✓ Berhasil dimasukkan.

Lalu akan muncul menu dan kasir memilih menu yang dipesan oleh pelanggan

--- DASHBOARD TRANSAKSI: BANU ---

1. Tambah Pesanan
2. Lihat / Hapus Pesanan
3. Bayar & Selesai

Pilih menu: 1

Lalu jika sudah menambahakan pesanan pertama akan muncul tampilan seperti di awal lagi dan jika ingin pelanggan ingin menambah pesanan pilih 1

DAFTAR MENU CAFE:

- | | |
|-------------------|----------|
| 1. Espresso | Rp18.000 |
| 2. Caramel Latte | Rp28.000 |
| 3. Matcha Latte | Rp25.000 |
| 4. Ice Lemon Tea | Rp15.000 |
| 5. Chocolate Lava | Rp25.000 |
| 6. Beef Sandwich | Rp35.000 |
| 7. Croissant | Rp22.000 |
| 8. Club Sandwich | Rp40.000 |

Pilih item (1-8): 8

✓ Berhasil dimasukkan.

Setelah itu masukkan pilihan sesuai pilihan pelanggan

--- DASHBOARD TRANSAKSI: BANU ---

1. Tambah Pesanan
2. Lihat / Hapus Pesanan
3. Bayar & Selesai

Pilih menu: 2

Setelah itu akan muncul tampilan menu seperti sebelumnya, untuk mengecek kembali pesanan apakah sudah sesuai apa belum bisa pilih menu

KERANJANG BANU:

1. Matcha Latte
2. Club Sandwich

Hapus No (0 batal): 0

Jika sudah pilih 2 lalu enter akan muncul sesuai yang di pesan oleh pelanggan (menu ini berfungsi untuk mengecek apakah pesanan sudah sesuai yang dipesan apa belum), jika sudah sesuai klik 0

```
--- DASHBOARD TRANSAKSI: BANU ---
```

1. Tambah Pesanan
2. Lihat / Hapus Pesanan
3. Bayar & Selesai

```
Pilih menu: 3
```

Selanjutnya akan ke tampilan sebelumnya, dan jika pesanan sudah selesai maka langsung ke pembayaran pilih 3

```
*****
SMART-CAFE 2026
*****
Pelanggan : BANU
Tipe       : Silver Member
-----
Matcha Latte      : Rp    25.000
Club Sandwich     : Rp    40.000
-----
TOTAL BELANJA     : Rp    65.000
POIN DIDAPAT      : 6 pts
TOTAL SALDO POIN   : 6 pts
*****
```

```
✓ Struk tersimpan: Struk_BANU.txt
```

Setelah itu akan muncul struk pembelian untuk pelanggan dan struk tersebut atas nama pelanggan yang sudah di inputkan di awal tadi supaya mempermudah untuk pengecekan jika pesanan sudah siap

```

=====
WELCOME TO SMART-CAFE 2026
=====

1. Input Transaksi Baru
0. Matikan Sistem

Pilih: 0

Sistem dimatikan. Terima kasih.

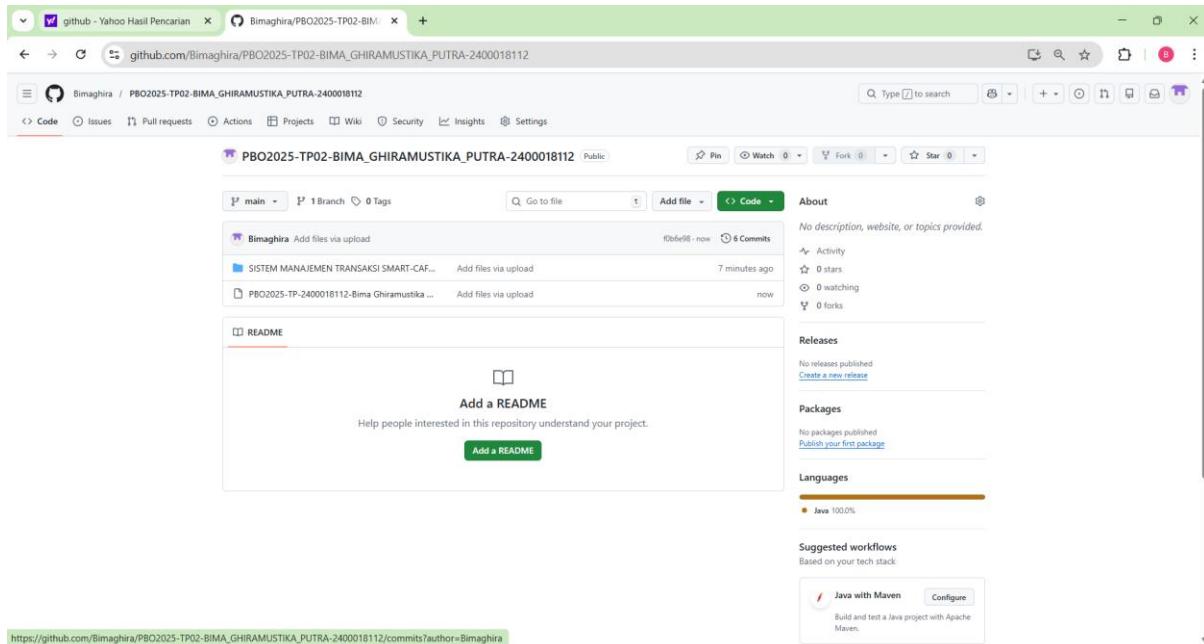
Process finished with exit code 0
|

```

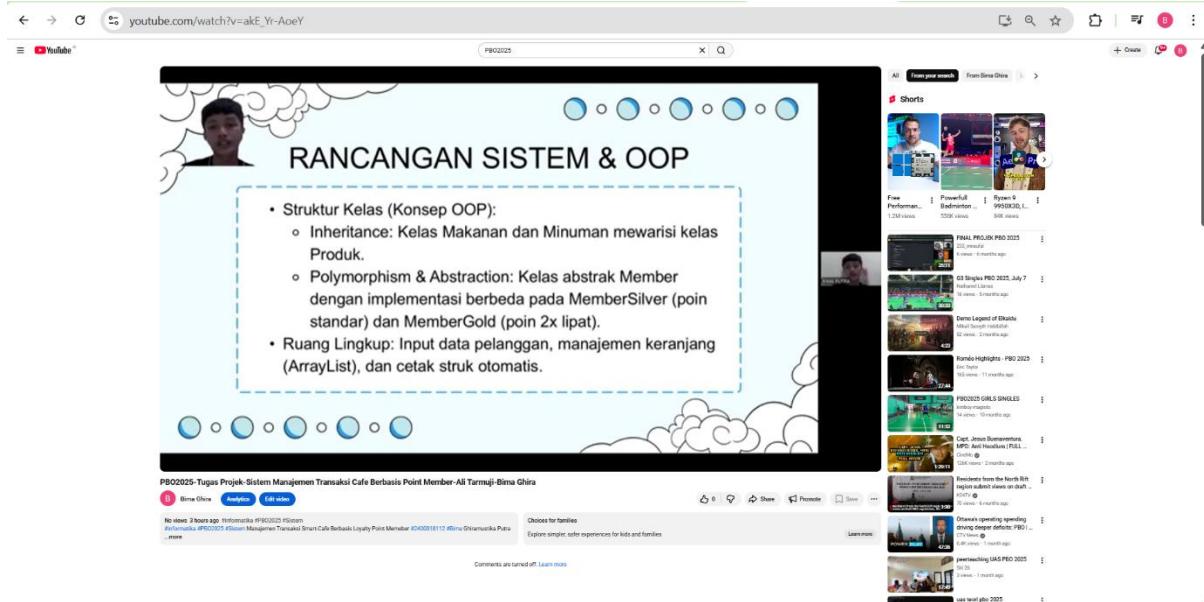
Setelah muncul struk pembelian akan muncul menu seperti di gambar yang mana jika sudah tidak apa pelanggan yang memesan maka pilih 0 untuk mengakhiri namun jika ada pelanggan selanjutnya makan pilih 1

E. Bukti Publikasi

Github:https://github.com/Bimaghira/PBO2025-TP02-BIMA_GHIRAMUSTIKA_PUTRA-2400018112



Youtube : https://youtu.be/akE_Yr-AoeY?si=N9tTejYNnWQ5lwM



F. Analisis Pengerjaan Projek

Proyek ini diselesaikan dalam durasi 5 hari kerja dengan tahapan sebagai berikut:

- Hari 1: Perancangan struktur kelas (Class Diagram) dan pendefinisian atribut dasar pada kelas Produk.
- Hari 2: Implementasi konsep Inheritance pada sub-kelas Makanan dan Minuman.
- Hari 3: Pengembangan sistem loyalitas menggunakan Abstract Class Member serta logika Polymorphism untuk tipe Silver dan Gold.
- Hari 4: Pembuatan logika utama transaksi pada Main.java, termasuk manajemen keranjang menggunakan ArrayList.
- Hari 5: Implementasi fitur File I/O untuk penyimpanan struk otomatis dan tahap pengujian sistem (debugging).

Kesulitan yang Dihadapi: Hambatan utama dalam pengerjaan terletak pada sinkronisasi logika perhitungan poin yang berbeda secara otomatis berdasarkan objek member yang dipilih saat runtime. Selain itu, penanganan input scanner untuk menghindari kesalahan pembacaan baris baru (newline) serta manajemen penghapusan item dalam daftar pesanan memerlukan ketelitian agar indeks data tetap akurat.

Sumber Projek:

- Logika Program: Berdasarkan prinsip Pemrograman Berorientasi Objek (OOP) Java.

- Data Master: Daftar menu dan harga yang didefinisikan secara statis dalam kode program Main.java.
- Library Utama: Menggunakan Java Standard Library seperti java.util.Scanner, java.util.ArrayList, dan java.io.FileWriter.

G. Kesimpulan

Aplikasi SMART-CAFE 2026 dirancang untuk mendigitalisasi proses kasir dan manajemen loyalitas pelanggan guna meningkatkan efisiensi layanan serta akurasi perhitungan poin member. Sistem ini menggunakan paradigma Object-Oriented Programming (OOP) dengan menerapkan konsep Inheritance pada kategori produk serta Polymorphism untuk membedakan perolehan poin antara Member Silver (1x poin) dan Member Gold (2x poin) secara otomatis. Melalui antarmuka kasir yang interaktif, pengguna dapat mengelola keranjang belanja yang terdiri dari 8 pilihan menu makanan dan minuman, melakukan pembaruan pesanan, hingga menghasilkan struk transaksi digital yang tersimpan secara otomatis dalam format file .txt. Secara keseluruhan, program ini berhasil mengintegrasikan struktur kode modular yang aman melalui Encapsulation untuk menciptakan solusi manajemen kafe yang transparan, profesional, dan terdokumentasi dengan baik.