Nama : I Putu Eka Nesa Suarsa

NIM : 2201010116

Kelas : C

Project : Beli Penjor

Deskripsi : Aplikasi BeliPenjor, aplikasi ini dibuat untuk mempermudah pencatatan pemesanan dan pembelian penjor, serta pengelolaan data barang penjor menjelang Hari Raya Galungan dan Kuningan di toko kami. Dengan dibuatnya aplikasi ini, diharapkan dapat membantu pencatatan transaksi di toko secara lebih efisien.

Pada menu utama aplikasi, terdapat dua pilihan menu yaitu pilihan Pembelian Penjor dan Input Data Barang Penjor:

1. Pembelian Penjor:
   Fitur ini memungkinkan pengguna untuk menambahkan data pembelian penjor, termasuk nama pembeli, harga penjor, jumlah penjor yang dibeli, tanggal pembelian, tanggal pengiriman, nomor telepon pembeli, dan alamat pembeli.
   Fitur ini berfungsi untuk mencatat setiap transaksi pembelian penjor. Dengan pencatatan yang rapi, distribusi pengiriman penjor menjadi lebih mudah, dan pembeli dapat dihubungi dengan cepat sehingga penjor dapat dikirim ke alamat yang tepat.
2. Input Data Barang Penjor:
   Fitur ini memungkinkan pengguna untuk menambahkan data barang penjor, termasuk ID barang, nama barang, harga beli, harga jual, jumlah pembelian awal, jumlah terjual, stok tersedia (jumlah pembelian awal dikurangi jumlah terjual), dan laba per barang. Fitur ini berguna untuk melihat harga barang secara cepat, memonitor persentase penjualan barang, dan menampilkan laba penjualan per ID barang.

Aplikasi ini memerlukan basis data berupa SQL Database untuk menyimpan semua data yang diinputkan, sehingga mempermudah akses, update, maupun penghapusan data. Dengan demikian, pencatatan dan pengelolaan transaksi di toko kami dapat berjalan lebih lancar dan efisien.

Teori Yang Digunakan :

1. Jform1 (data barang)
   a. Java Swing: Java Swing adalah bagian dari Java yang digunakan untuk membangun aplikasi dengan antarmuka pengguna grafis (GUI). Beberapa komponen yang sering digunakan termasuk:
      - JFrame: Jendela utama aplikasi.
      - JTable: Tabel untuk menampilkan data.
      - JButton: Tombol yang bisa diklik.

- JLabel: Label untuk menampilkan teks atau gambar.
- JTextField: Kotak teks untuk input pengguna.

b. JDBC (Java Database Connectivity): JDBC adalah alat yang memungkinkan Java untuk berinteraksi dengan database. Dalam aplikasi ini, JDBC digunakan untuk melakukan operasi:
   - Create (Membuat data barang)
   - Read (Membaca data barang)
   - Update (Memperbarui data barang)
   - Delete (Menghapus) pada data barang di database.

c. Desain Pola (Design Pattern): Beberapa desain pola yang digunakan dalam aplikasi ini termasuk:
   - Model-View-Controller (MVC):
     - Model: Mengacu pada data (database dan operasi CRUD).
     - View: Antarmuka pengguna (JFrame dan komponen GUI lainnya).
     - Controller: Logika aplikasi yang menghubungkan model dan view (event handling pada tombol).
     - Mengapa Menggunakan MVC
       - Memisahkan logika aplikasi, antarmuka pengguna, dan data, membuat kode lebih mudah dipelihara dan diperbarui.
       - Membagi aplikasi menjadi tiga komponen utama memungkinkan pengembangan yang modular dan kolaboratif.
       - Pengujian unit lebih mudah dilakukan karena logika aplikasi terpisah dari antarmuka pengguna.
     - Tujuan Dalam Sistem
       - Meningkatkan struktur dan keterbacaan kode.
       - Memudahkan pengembangan dan pemeliharaan aplikasi.
       - Memungkinkan pengembangan dan pengujian komponen secara terpisah.
     - Singleton :
       - Pola desain ini digunakan untuk memastikan hanya ada satu instance koneksi ke database yang digunakan.

d. Event Handling : Event handling adalah konsep yang digunakan untuk menangani tindakan pengguna, seperti klik tombol. Dalam aplikasi ini, berbagai tindakan tombol seperti btnBARU, btnUBAH, btnHAPUS, dan btnTUTUP memiliki event listener yang terkait dengannya.
   - Penanganan Kesalahan (Exception Handling): Exception handling digunakan untuk menangani kesalahan yang terjadi selama runtime. Dalam aplikasi ini, exception handling digunakan saat berinteraksi dengan database dan saat membaca gambar dari file. Mengapa Menggunakan Error Handling:

- Kehandalan Aplikasi: Menangani kesalahan dengan baik memastikan aplikasi tetap berjalan meskipun terjadi kesalahan.
- Pengalaman Pengguna yang Lebih Baik: Memberikan umpan balik yang informatif kepada pengguna saat terjadi kesalahan.
- Debugging Mudah: Memudahkan pengembang dalam mengidentifikasi dan memperbaiki bug.

- Tujuan dalam Sistem :
  - Meningkatkan stabilitas dan kehandalan aplikasi
  - Memastikan aplikasi dapat menangani kesalahan dengan cara yang tidak mengganggu pengguna.
  - Mempermudah proses debug dan pengembangan
  - Mempermudah proses debug dan pengembangan.

e. Image Handling : Untuk menampilkan gambar di GUI, aplikasi ini menggunakan BufferedImage dan ImageIcon untuk memuat dan menampilkan gambar barang.

f. Graphical User Interface / GUI : GUI adalah antarmuka yang memungkinkan pengguna untuk berinteraksi dengan aplikasi melalui elemen visual seperti tombol, tabel, dan form.

g. Mengapa Menggunakan GUI:
   - Mengapa Menggunakan GUI :
     - Memungkinkan pengguna untuk berinteraksi dengan aplikasi melalui elemen visual yang intuitif Memastikan aplikasi dapat menangani kesalahan dengan cara yang tidak mengganggu pengguna.
     - Mempermudah GUI membuat aplikasi lebih mudah digunakan, terutama bagi pengguna non-teknis.
     - Memberikan tampilan yang menarik dan profesional pada aplikasi.

h. Tujuan dalam Sistem:
   - Meningkatkan pengalaman pengguna dengan antarmuka yang ramah pengguna.
   - Memudahkan pengguna dalam mengelola data produk melalui interaksi visual.
   - Menyediakan visualisasi data yang lebih baik.

2. Jform2 (data pembeli)
   a. Java Swing: Java Swing adalah bagian dari Java yang digunakan untuk membangun aplikasi dengan antarmuka pengguna grafis (GUI). Beberapa komponen yang sering digunakan termasuk:

- JFrame: Jendela utama aplikasi.
- JTable: Tabel untuk menampilkan data.
- JButton: Tombol yang bisa diklik.
- JLabel: Label untuk menampilkan teks atau gambar.

- JTextField: Kotak teks untuk input pengguna.

b. JDBC (Java Database Connectivity): JDBC adalah alat yang memungkinkan Java untuk berinteraksi dengan database. Dalam aplikasi ini, JDBC digunakan untuk melakukan operasi:
   - Create (membuat data pembeli di database)
   - Read (melihat data pembeli di database)
   - Update (memperbaharui data pembeli di database)
   - Delete (Menghapus) pada data pembeli di database.

c. Desain Pola (Design Pattern): Beberapa desain pola yang digunakan dalam aplikasi ini termasuk:
   - Model-View-Controller (MVC):
     - Model: Mengacu pada data (database dan operasi CRUD).
     - View: Antarmuka pengguna (JFrame dan komponen GUI lainnya).
     - Controller: Logika aplikasi yang menghubungkan model dan view (event handling pada tombol).
     - Mengapa Menggunakan MVC
       - Pemeliharaan Mudah: Memisahkan logika aplikasi, antarmuka pengguna, dan data, membuat kode lebih mudah dipelihara dan diperbarui.
       - Modularitas: Membagi aplikasi menjadi tiga komponen utama memungkinkan pengembangan yang modular dan kolaboratif.
       - Pengujian yang Lebih Baik: Pengujian unit lebih mudah dilakukan karena logika aplikasi terpisah dari antarmuka pengguna.
     - Tujuan Dalam Sistem
       - Meningkatkan struktur dan keterbacaan kode.
       - Memudahkan pengembangan dan pemeliharaan aplikasi.
       - Memungkinkan pengembangan dan pengujian komponen secara terpisah.
     - Singleton :
       - Pola desain ini digunakan untuk memastikan hanya ada satu instance koneksi ke database yang digunakan.

d. Penanganan Peristiwa (Event Handling): Event handling adalah konsep yang digunakan untuk menangani tindakan pengguna, seperti klik tombol. Dalam aplikasi ini, berbagai tindakan tombol seperti btnBARU, btnUBAH, btnHAPUS, dan btnTUTUP memiliki event listener yang terkait dengannya.
   - (Exception Handling): Exception handling digunakan untuk menangani kesalahan yang terjadi selama runtime. Dalam aplikasi ini, exception handling digunakan saat berinteraksi dengan database dan saat membaca gambar dari file. Mengapa Menggunakan Error Handling:

        - Menangani kesalahan dengan baik memastikan aplikasi tetap berjalan meskipun terjadi kesalahan.
        - Memberikan umpan balik yang informatif kepada pengguna saat terjadi kesalahan.
        - Memudahkan pengembang dalam mengidentifikasi dan memperbaiki bug.
    - Tujuan dalam Sistem :
        - Meningkatkan stabilitas dan kehandalan aplikasi
        - Memastikan aplikasi dapat menangani kesalahan dengan cara yang tidak mengganggu pengguna.
        - Mempermudah proses debug dan pengembangan
        - Mempermudah proses debug dan pengembangan.

e. Image Handling : Untuk menampilkan gambar di GUI, aplikasi ini menggunakan BufferedImage dan ImageIcon untuk memuat dan menampilkan gambar barang.

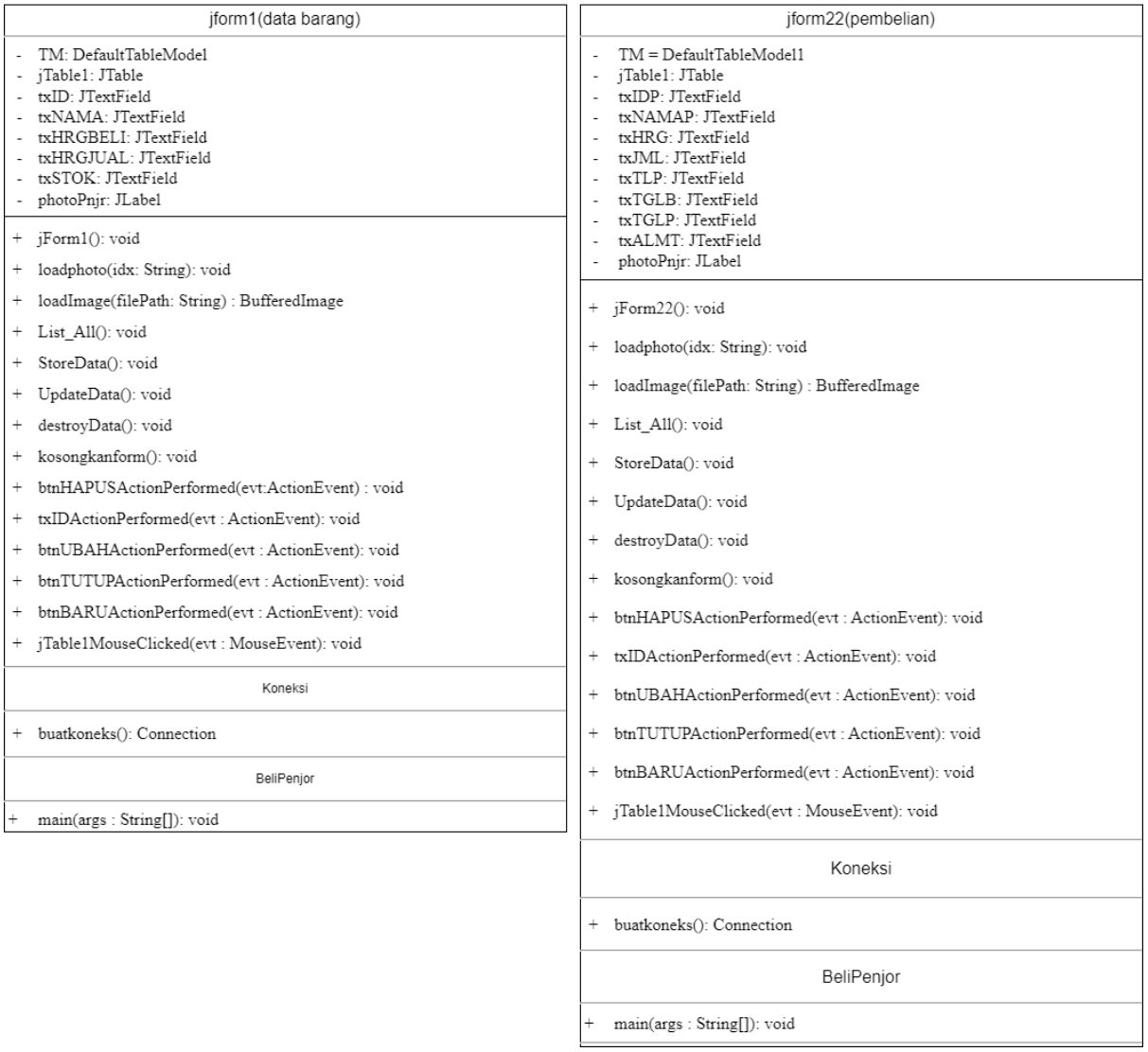
f. Graphical User Interface / GUI : GUI adalah antarmuka yang memungkinkan pengguna untuk berinteraksi dengan aplikasi melalui elemen visual seperti tombol, tabel, dan form.

g. Mengapa Menggunakan GUI:
    - Mengapa Menggunakan GUI :
        - Memungkinkan pengguna untuk berinteraksi dengan aplikasi melalui elemen visual yang intuitif Memastikan aplikasi dapat menangani kesalahan dengan cara yang tidak mengganggu pengguna.
        - Mempermudah GUI membuat aplikasi lebih mudah digunakan, terutama bagi pengguna non-teknis.
        - Memberikan tampilan yang menarik dan profesional pada aplikasi.

h. Tujuan dalam Sistem:
    - Meningkatkan pengalaman pengguna dengan antarmuka yang ramah pengguna.
    - Memudahkan pengguna dalam mengelola data produk melalui interaksi visual.
    - Menyediakan visualisasi data yang lebih baik.

UML Class :

UML Class BeliPenjor



-