Universitas Muhammadiyah Malang

LAB REPORT
COURSE: ADVANCED PROGRAMMING
LAB TITLE: FINAL PRACTICUM EXAM

Name:

Muhammad Kharisma Aditya Putra (202410370110200)

Muhammad Diandra Adzka Surya (202410370110024)

Class: Informatika/3-A/5-K

BAB I
PROGRAM KELOLA STOK BARANG GUDANG

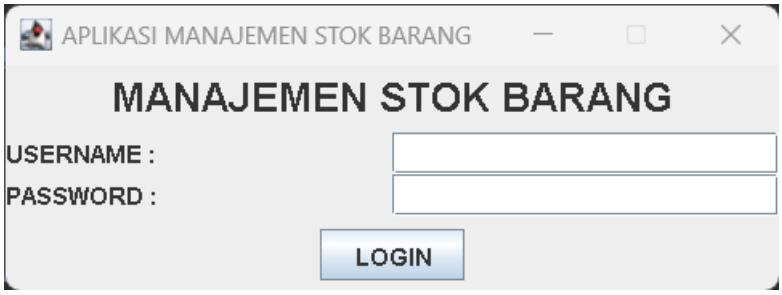
Program Stok Barang Gudang yang kami buat sebagai Tugas Ujian Akhir Praktikum merupakan program berbasis Java Swing yang dirancang untuk membantu proses pengelolaan data stok barang di gudang secara terkomputerisasi. Program ini menyediakan fitur CRUD (Create, Read, Update, dan Delete) yang memungkinkan pengguna untuk menambahkan data barang baru, menampilkan daftar stok barang, memperbarui informasi barang, serta menghapus data barang yang tidak diperlukan. Data stok ditampilkan dalam bentuk tabel (JTable) sehingga memudahkan pengguna dalam melihat dan mengelola informasi barang secara terstruktur.

Selain itu, program ini dilengkapi dengan fitur pencarian data secara dinamis (real-time search) menggunakan JTextField dan TableRowSorter, yang memungkinkan pengguna mencari data barang dengan cepat berdasarkan kata kunci tertentu tanpa perlu menekan tombol pencarian. Setiap perubahan teks pada kolom pencarian akan langsung memfilter data pada tabel, sehingga meningkatkan efisiensi dan kemudahan penggunaan. Dengan antarmuka grafis yang sederhana dan fungsional, program ini diharapkan dapat membantu proses pengelolaan stok barang gudang menjadi lebih efektif, akurat, dan terorganisir.

Untuk mendukung penyimpanan data, program ini menerapkan konsep file handling dengan menggunakan file berformat CSV (Comma Separated Values) sebagai media penyimpanan data stok barang. Setiap data barang yang ditambahkan, diperbarui, atau dihapus melalui fitur CRUD akan disimpan dan dikelola langsung pada file CSV, sehingga data tetap tersimpan meskipun program ditutup. Proses pembacaan dan penulisan data dilakukan secara terstruktur, di mana setiap baris pada file CSV merepresentasikan satu data barang dan setiap kolom berisi atribut barang seperti nama barang, jumlah stok, dan informasi lainnya. Dengan penggunaan file CSV, program ini menjadi lebih ringan, mudah dipelihara, serta tidak bergantung pada sistem basis data eksternal.

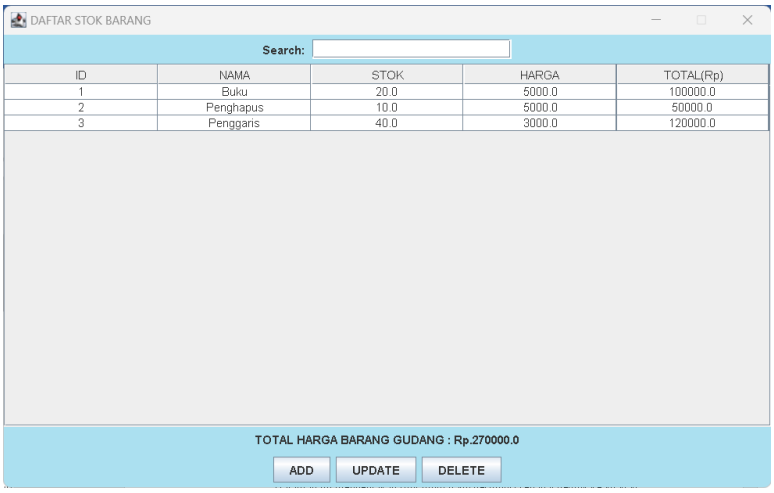
Berikut Tampilan-tampilan pada program yang kami buat.

A. HALAMAN LOGIN



Halaman ini menyediakan fitur login yang berfungsi sebagai bentuk keamanan dimana hanya pengguna yang memiliki akses yang dapat mengakses program ini.

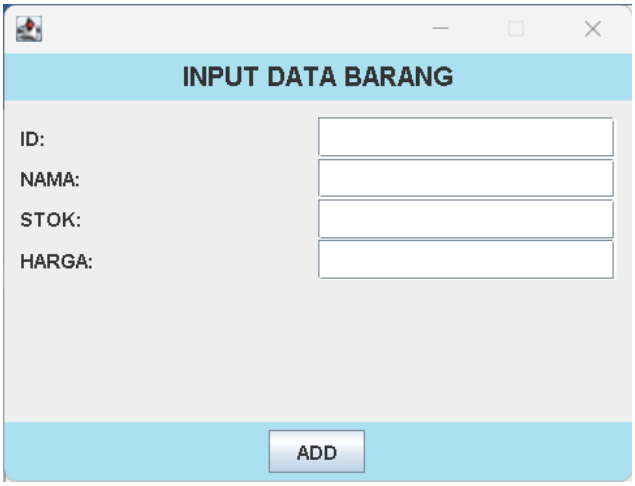
B. HALAMAN TABEL (NAVIGATION PANE)



Setelah pengguna yang memiliki hak akses berhasil melakukan proses login, sistem akan menampilkan halaman Navigasi. Pada halaman ini, pengguna dapat melihat data barang yang disajikan dalam bentuk tabel, yang meliputi ID barang, nama barang, jumlah stok, harga per barang, serta total harga barang. Tabel tersebut dilengkapi dengan fitur pengurutan (sorting) dan pencarian (searching) untuk memudahkan pengguna dalam mengurutkan dan menemukan data tertentu.

Pada bagian bawah halaman, ditampilkan total harga keseluruhan barang gudang, yang merupakan akumulasi nilai seluruh stok barang berdasarkan jumlah dan harga yang tercantum pada data. Selain itu, tersedia pula tombol ADD, UPDATE, dan DELETE yang berfungsi untuk melakukan penambahan, pembaruan, serta penghapusan data barang pada tabel.

C. HALAMAN TAMBAH DATA BARU (ADD DATA)

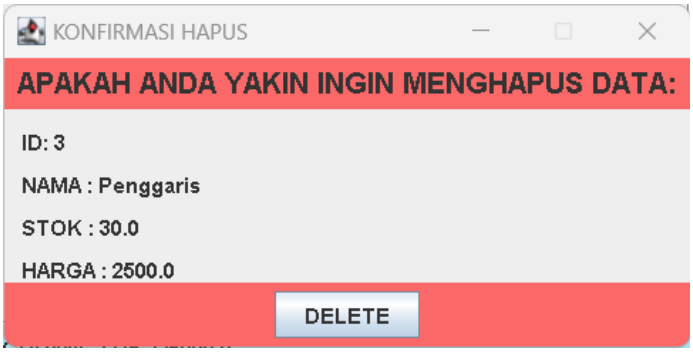


Halaman ini merupakan halaman yang digunakan untuk menambahkan data baru ke dalam tabel. Halaman ini akan ditampilkan ketika pengguna menekan tombol ADD pada Halaman Tabel. Melalui halaman ini, pengguna dapat memasukkan data barang baru sesuai dengan atribut yang telah disediakan. Data yang telah dimasukkan akan ditampilkan pada Halaman Tabel setelah pengguna menekan tombol ADD sebagai konfirmasi penambahan data.

D. HALAMAN UPDATE DATA YANG SUDAH ADA (UPDATE DATA)

Halaman ini merupakan halaman yang digunakan untuk memperbarui data barang yang dipilih pada tabel. Pada halaman ini, pengguna dapat melakukan perubahan terhadap nilai stok dan harga barang. Setelah proses pembaruan dilakukan, sistem akan secara otomatis memperbarui data barang yang terpilih pada tabel sehingga menampilkan nilai terbaru sesuai dengan perubahan yang dilakukan.

E. HALAMAN DELETE DATA



Halaman ini merupakan halaman konfirmasi yang digunakan untuk menghapus data barang yang dipilh pada tabel. Pada halaman ini, pengguna dapat melakukan konfirmasi apakah data yang dipilih ingin dihapus atau tidak. Setelah proses konfirmasi dilakukan, sistem akan secara otomatis memperbarui data barang pada tabel tanpa menampilkan data yang kita hapus

BAB II
PENJELASAN FITUR-FITUR PENTING

A. MENAMPILKAN DATA MENGGUNAKAN TABEL (JTable)

JTable merupakan salah satu komponen Graphical User Interface (GUI) pada Java Swing yang digunakan untuk menampilkan dan mengelola data dalam bentuk tabel, serupa dengan tabel pada Microsoft Excel atau spreadsheet lainnya. Secara bawaan (default), JTable menampilkan data dalam bentuk tipe String. Namun, pada program yang dikembangkan, diperlukan penampilan data dengan tipe Integer dan Double agar sesuai dengan kebutuhan pengolahan data stok barang. Oleh karena itu, dibuat sebuah DefaultTableModel khusus untuk menangani perbedaan tipe data tersebut, sehingga data dapat ditampilkan dan dikelola dengan tipe yang sesuai.

```
String[] columns = {"ID","NAMA","STOK","HARGA","TOTAL(Rp)"};
MyTableModel model = new MyTableModel(columns);
JTable table = new JTable(model);
```

Kami menggunakan custom DefaultTableModel, yaitu MyTableModel, untuk mengatasi permasalahan perbedaan tipe data pada setiap kolom tabel. Class MyTableModel berfungsi untuk mengatur dan menyesuaikan tipe data pada masing-masing kolom sesuai dengan kebutuhan program. Pada program ini, data ditampilkan pada tabel dalam bentuk String, Integer, dan Double, sehingga pengelolaan dan penampilan data dapat dilakukan secara lebih akurat dan terstruktur.

```
public class MyTableModel extends DefaultTableModel {  9 usages   KharismaAditya *

    public MyTableModel(Object[] columns) { super(columns,  rowCount: 0); }

    @Override   KharismaAditya *
    public Class<?> getColumnClass(int columnIndex) {
        return switch (columnIndex) {
            case 0 -> Integer.class;
            case 2 -> Integer.class;
            case 3 -> Double.class;
            case 4 -> Double.class;
            default -> String.class;
        };
    }

    @Override   KharismaAditya
    public boolean isCellEditable(int row, int column) { return column != 0; }
}
```

Pada kode yang ditampilkan, dilakukan pengaturan tipe data pada kolom tertentu sesuai dengan jenis data yang dimuat. Kolom dengan indeks 0 dan 2 diubah ke dalam tipe Integer karena pada kolom tersebut tersimpan data berupa ID barang dan jumlah stok barang.

Selanjutnya, kolom dengan indeks 3 dan 4 diubah ke dalam tipe Double karena kolom tersebut memuat data berupa harga per barang dan total harga barang. Pengaturan tipe data ini bertujuan agar fitur pengurutan data (sorting) pada JTable dapat berjalan dengan benar, khususnya pada data yang berbentuk numerik atau nominal, sehingga hasil pengurutan menjadi lebih akurat.

B. LOAD DATA DARI CSV

Pada program yang kami kembangkan, digunakan file CSV (Comma Separated Values) sebagai media penyimpanan data barang gudang. Pada fungsi Load Data, sistem membaca setiap data yang tersimpan di dalam file CSV untuk kemudian dimuat ke dalam program, sehingga data tersebut dapat dikelola dan diproses lebih lanjut sesuai dengan kebutuhan aplikasi.

```
public static void loadCSV(String path, DefaultTableModel model) {  1 usage   KharismaAditya
    try (BufferedReader br = new BufferedReader(new FileReader(path))) {
        String line;
        boolean header = true;

        while ((line = br.readLine()) != null) {
            String[] data = line.split( regex: ",");

            if (header) {
                header = false;
                continue;
            }
            double total = Double.parseDouble(data[2]) * Double.parseDouble(data[3]);

            Object[] row = new Object[]{
                    Integer.parseInt(data[0]),
                    data[1],
                    Double.parseDouble(data[2]),
                    Double.parseDouble(data[3]),
                    Double.parseDouble(String.valueOf(total)),
            };

            model.addRow(row);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Pada kode tersebut, digunakan BufferedReader untuk membaca isi file CSV secara efisien. Struktur perulangan (loop) pada kode berfungsi untuk membaca data pada setiap baris file secara berurutan. Setiap baris data yang dibaca kemudian dipisahkan ke dalam elemen-elemen data dan disimpan ke dalam variabel data dengan menggunakan tanda koma (,) sebagai pemisah antar elemen, sesuai dengan format file CSV.

```
double total = Double.parseDouble(data[2]) * Double.parseDouble(data[3]);
```

Baris kode tersebut berfungsi untuk melakukan perhitungan total harga barang dengan cara mengalikan jumlah stok dengan harga per barang. Pada file CSV yang digunakan, data total harga barang tidak disimpan secara langsung, sehingga nilai tersebut tidak diambil dari file CSV, melainkan dihitung secara otomatis di dalam program. Perhitungan ini dilakukan dengan mengambil nilai stok dan harga dari data barang yang bersangkutan. Pendekatan ini bertujuan untuk menghindari penyimpanan data yang berlebihan pada file CSV, serta menjaga konsistensi dan efisiensi pengelolaan data.

```
Object[] row = new Object[]{
        Integer.parseInt(data[0]),
        data[1],
        Double.parseDouble(data[2]),
        Double.parseDouble(data[3]),
        Double.parseDouble(String.valueOf(total)),
};

model.addRow(row);
```

Kode tersebut berfungsi untuk menampilkan data yang berasal dari file CSV ke dalam tabel pada Java Swing. Variabel Object[] row digunakan untuk menampung satu baris data yang dibaca dari file CSV sebelum ditampilkan pada JTable. Data yang dimasukkan ke dalam variabel tersebut harus disusun secara berurutan sesuai dengan struktur kolom tabel serta menggunakan tipe data yang telah ditentukan. Selanjutnya, baris kode model.addRow(row) berfungsi untuk menambahkan data yang tersimpan di dalam variabel row ke dalam TableModel, sehingga data tersebut dapat ditampilkan pada tabel.

C. FITUR SEARCH DAN SORTING PADA TABEL

1. SEARCHING

```
JTextField searchField = new JTextField( columns: 20);
searchField.getDocument().addDocumentListener(new javax.swing.event.DocumentListener() {  KharismaAdity
    private void search() {  3 usages  KharismaAditya
        String text = searchField.getText();

        if (text.trim().length() == 0) {
            sorter.setRowFilter(null);
        } else {
            sorter.setRowFilter(
                    RowFilter.regexFilter( regex: "(?i)" + text)
            );
        }
    }

    @Override  KharismaAditya
    public void insertUpdate(javax.swing.event.DocumentEvent e) { search(); }

    @Override  KharismaAditya
    public void removeUpdate(javax.swing.event.DocumentEvent e) { search(); }

    @Override  KharismaAditya
    public void changedUpdate(javax.swing.event.DocumentEvent e) { search(); }
});
```

Kode ini mengimplementasikan fitur pencarian dinamis (real-time search) pada JTable dengan memanfaatkan:

- JTextField sebagai input pencarian
- DocumentListener untuk mendeteksi perubahan teks
- TableRowSorter dan RowFilter untuk memfilter data tabel

Dengan metode ini, data pada JTable akan otomatis tersaring setiap kali pengguna mengetik atau menghapus teks, tanpa memerlukan tombol pencarian tambahan.

Search: Peng

ID	NAMA	STOK ▲	HARGA	TOTAL(Rp)
2	Penghapus	10.0	5000.0	50000.0
3	Penggaris	30.0	2500.0	75000.0

Pada tampilan berikut, diperlihatkan bagaimana fitur searching bekerja, pada TextField diinput text berupa ”Peng” yang secara otomatis sistem akan mencari data dengan nama yang mengandung bagian yang diinput.

2. SORTING

```
TableRowSorter<DefaultTableModel> sorter =
        new TableRowSorter<>(model);
table.setRowSorter(sorter);
```

Kode tersebut berfungsi untuk menangani metode pengurutan data (sorting) pada tabel. Pengurutan yang dimaksud adalah proses mengurutkan data berdasarkan nilai tertentu, baik dari nilai terkecil ke terbesar maupun sebaliknya. Pada contoh yang ditampilkan, sistem melakukan proses pengurutan pada kolom Stok, sehingga tampilan data pada tabel akan disusun berdasarkan nilai stok setiap barang dari yang terkecil ke yang terbesar.

Search:

ID	NAMA	STOK ▲	HARGA	TOTAL(Rp)
2	Penghapus	10.0	5000.0	50000.0
1	Buku	20.0	5000.0	100000.0
3	Penggaris	30.0	2500.0	75000.0

D. FITUR ADD

```
public void addData(DefaultTableModel model, JFrame frame) {  1 usage   KharismaAditya
    try{
        int ID = Integer.parseInt(idInput.getText());
        String NAMA = nameInput.getText();
        double STOK = Double.parseDouble(stokInput.getText());
        double HARGA = Double.parseDouble(priceInput.getText());
        double TOTAL = STOK * HARGA;
        if(isAlready(model,NAMA,idInput.getText())){
            model.addRow(new Object[]{
                    ID,NAMA,STOK,HARGA,TOTAL
            });

            CSVUtil.saveCSV(FILE_PATH, (MyTableModel) model);
            mainApp.updateTotalLabel();
            JOptionPane.showMessageDialog(frame,  message: "Data berhasil ditambah!");
        }else{
            JOptionPane.showMessageDialog(frame,  message: "Data yang anda masukkan sudah ada");
            idInput.setText("");nameInput.setText("");stokInput.setText("");priceInput.setText("");
        }
    }catch(NumberFormatException ex){
        JOptionPane.showMessageDialog(frame,  message: "NILAI STOK DAN HARGA HARUS BERUPA ANGKA");
        idInput.setText("");nameInput.setText("");stokInput.setText("");priceInput.setText("");
    }
}
```

Kode tersebut menunjukkan proses penambahan data barang baru ke dalam sistem. Untuk menambahkan data, pengguna diwajibkan menginput ID barang, nama barang, jumlah stok, serta harga barang melalui field yang telah disediakan, kemudian menekan tombol ADD sebagai konfirmasi. Data yang berhasil ditambahkan akan diproses oleh sistem dan selanjutnya ditampilkan pada tabel melalui mekanisme penambahan baris pada model tabel.

```
model.addRow(new Object[]{
        ID,NAMA,STOK,HARGA,TOTAL
});
```

Kode tersebut berfungsi untuk menambahkan data baru ke dalam tampilan tabel dengan menggunakan metode .addRow(). Setelah data baru berhasil ditambahkan, baris baru akan langsung muncul pada tabel, sehingga pengguna dapat melihat hasil penambahan data secara langsung tanpa perlu memuat ulang tampilan.

```
CSVUtil.saveCSV(FILE_PATH, (MyTableModel) model);
```

Data baru yang ditambahkan ke dalam tabel akan disimpan dengan cara menyimpan tampilan data terbaru pada tabel ke dalam file CSV, sehingga perubahan data dapat tersimpan secara permanen dan tetap tersedia saat program dijalankan kembali.

CSVUtil.saveCSV()

```java
public class CSVutil {  6 usages  KharismaAditya
    public static void saveCSV(String path, MyTableModel model) {  3 usages  KharismaAditya
        try (PrintWriter pw = new PrintWriter(new FileWriter(path))) {
            for (int i = 0; i < model.getColumnCount(); i++) {
                pw.print(model.getColumnName(i));
                if (i < model.getColumnCount() - 1) pw.print(",");
            }
            pw.println();

            for (int i = 0; i < model.getRowCount(); i++) {
                for (int j = 0; j < model.getColumnCount(); j++) {
                    pw.print(model.getValueAt(i, j));
                    if (j < model.getColumnCount() - 1) pw.print(",");
                }
                pw.println();
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

}
```

Baris kode CSVUtil.saveCSV() pada fitur ADD berfungsi untuk menyimpan data terbaru yang ditampilkan pada tabel ke dalam file CSV. Metode saveCSV akan mengambil seluruh data yang terdapat pada tabel, kemudian menyimpannya kembali ke file CSV dengan cara menimpa (overwrite) isi file sebelumnya. Dengan mekanisme ini, setiap kali fungsi tersebut dipanggil, isi file CSV akan selalu diperbarui sesuai dengan kondisi data yang sedang ditampilkan pada tabel. Proses ini memastikan bahwa data baru yang ditambahkan dapat tersimpan secara permanen dan tetap konsisten dengan tampilan tabel saat ini.

E. FITUR UPDATE

```
updButton.addActionListener( ActionEvent e -> {
    try {
        String STOK = StokInput.getText();
        String HARGA = HargaInput.getText();
        double TOTAL = Double.parseDouble(STOK) * Double.parseDouble(HARGA);

        model.setValueAt(Double.parseDouble(STOK), row,  column: 2);
        model.setValueAt(Double.parseDouble(HARGA), row,  column: 3);
        model.setValueAt(TOTAL, row,  column: 4);

        CSVUtil.saveCSV(FILE_PATH, (MyTableModel) model);
        mainApp.updateTotalLabel();
        JOptionPane.showMessageDialog(frame,  message: "Data berhasil diupdate!");
        frame.dispose();

    } catch (NumberFormatException ex) {
        JOptionPane.showMessageDialog(frame,  message: "Nilai dan Kehadiran harus angka!");
    }
});
```

Kode tersebut menunjukkan proses pembaruan (update) data pada baris tabel yang dipilih. Untuk melakukan pembaruan data, pengguna diharuskan menginput nilai stok baru dan harga baru, kemudian menekan tombol UPDATE sebagai konfirmasi perubahan. Metode model.setValueAt() digunakan untuk mengubah nilai pada kolom tertentu di dalam tabel dengan menentukan nilai terbaru, indeks baris yang dipilih, dan indeks kolom yang akan diperbarui. Sebagai contoh, perubahan nilai stok dilakukan dengan memperbarui kolom stok pada baris data yang dipilih, sehingga tabel akan menampilkan nilai terbaru sesuai dengan input pengguna.

```
model.setValueAt(Double.parseDouble(STOK), row,  column: 2);
```

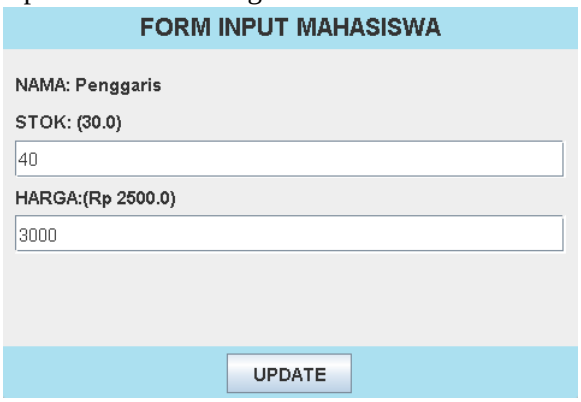
- Double.parseDouble(STOK) adalah nilai Stok baru
- row adalah jenis barang yang ingin kita ubah
- column : 2 adalah letak kolom stok (2 pada kode berarti kolom ke-2)

Contoh Implementasi fitur Update

1. Data sebelum update

3	Penggaris	30.0	2500.0	75000.0

2. Update Stok dan Harga



3. Data sesudah update

3	Penggaris	40.0	3000.0	120000.0

Dan sama seperti sebelumya, CSVUtil.saveCSV() berfungsi untuk menyimpan data terbaharui ke dalam file CSV

F. FITUR DELETE

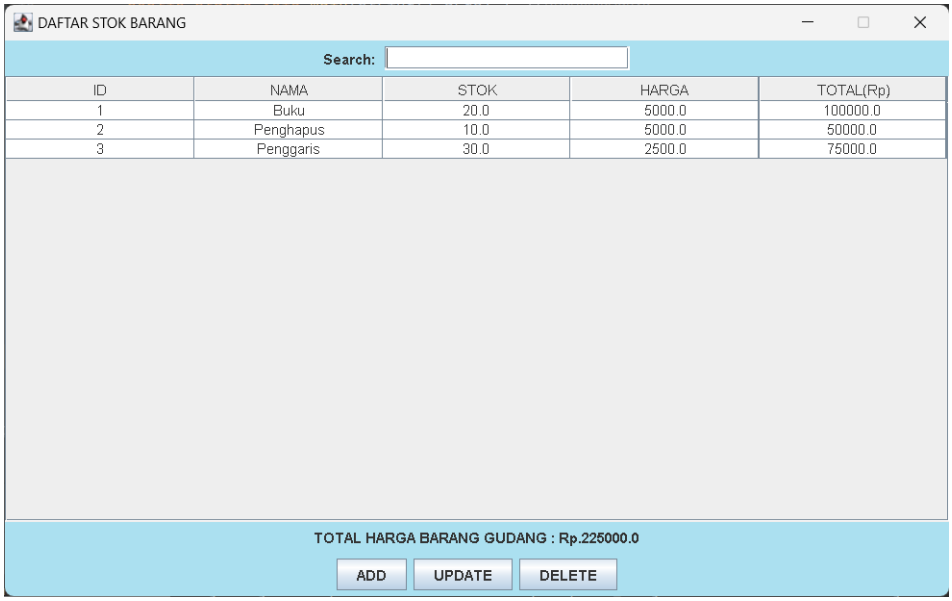
```
model.removeRow(row);
CSVutil.saveCSV(FILE_PATH,(MyTableModel) model);
mainApp.updateTotalLabel();
JOptionPane.showMessageDialog(frame, message: "Data berhasil dihapus!");
frame.dispose();
```

Kode tersebut menunjukkan proses penghapusan data pada baris tabel yang dipilih. Metode model.removeRow(modelRow) digunakan untuk menghapus baris data tertentu berdasarkan indeks baris yang dipilih (modelRow) dari tabel. Setelah proses penghapusan dilakukan, metode CSVUtil.saveCSV() kembali dipanggil untuk menyimpan kondisi data terbaru ke dalam file CSV. Dengan demikian, data yang telah dihapus tidak lagi tersimpan pada file CSV dan perubahan tersebut bersifat permanen.

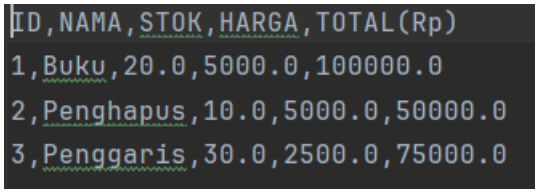
BAB III
UJI FITUR CRUD

A. FITUR ADD

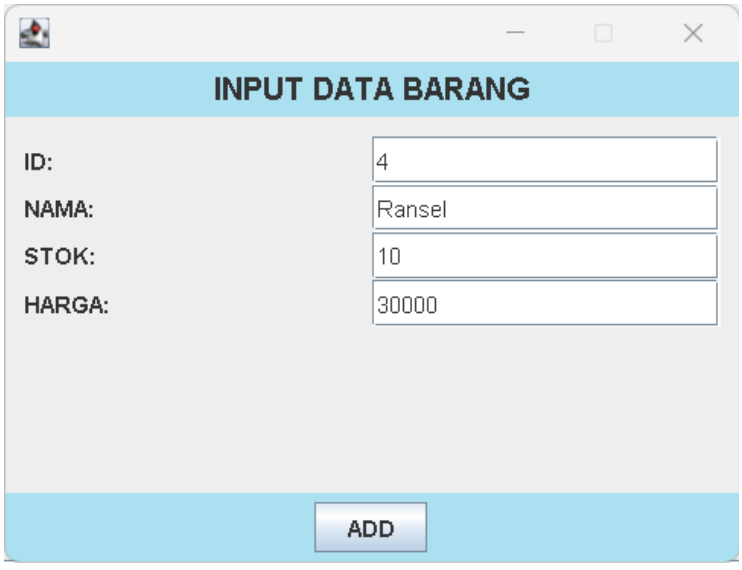
1. Data sebelum
    - Tabel



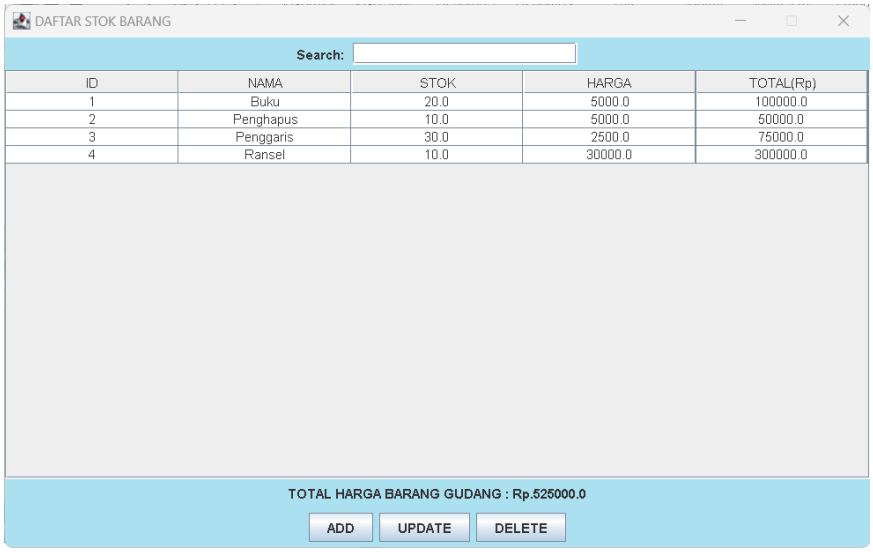
    - CSV



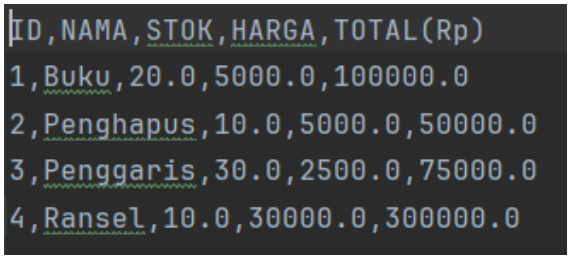
2. Penambahan data
    - ID : 4
    - BARANG : Ransel
    - STOK : 10
    - HARGA : 30.000



3. Data sesudah
    - Tabel

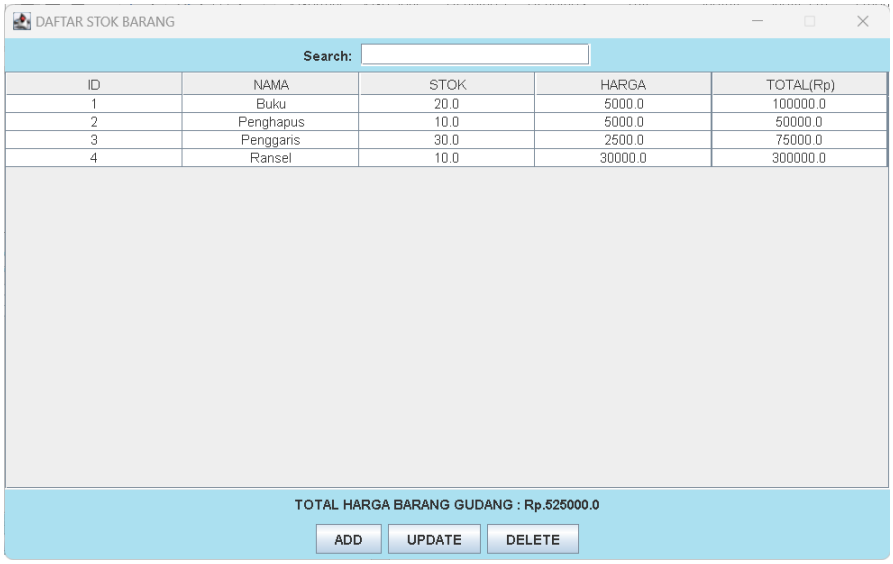


    - CSV



B. FITUR UPDATE

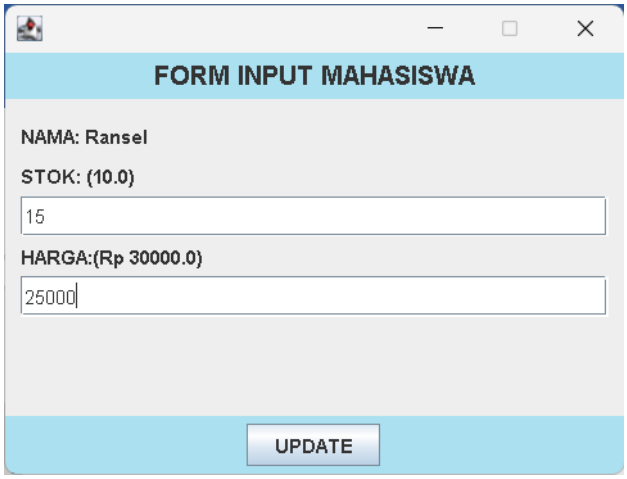
1. Data sebelum
    - Tabel



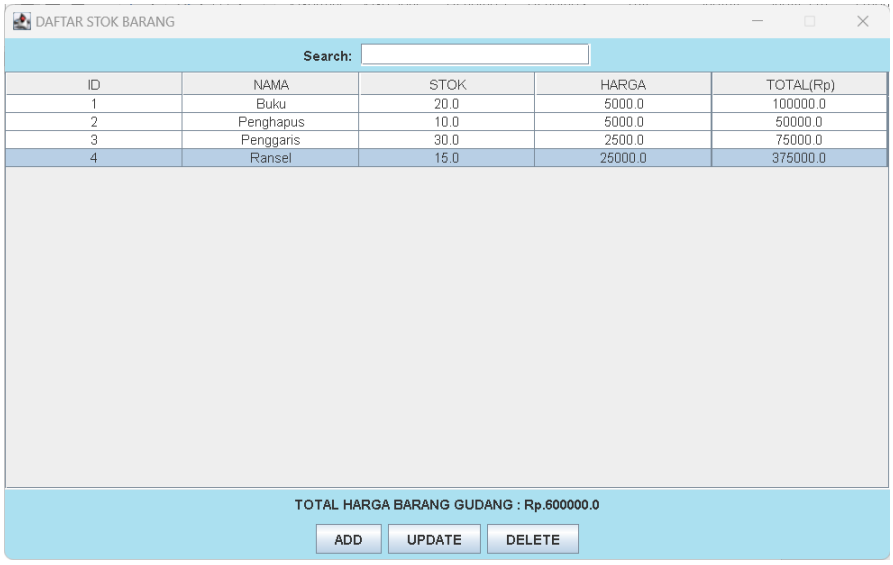
- CSV

```
ID,NAMA,STOK,HARGA,TOTAL(Rp)
1,Buku,20.0,5000.0,100000.0
2,Penghapus,10.0,5000.0,50000.0
3,Penggaris,30.0,2500.0,75000.0
4,Ransel,10.0,30000.0,300000.0
```

2. Ubah data yang sudah ada
   - Data yang diubah : Ransel
   - STOKbaru : 15
   - HARGA baru : 25.000



3. Data sesudah
   - Tabel

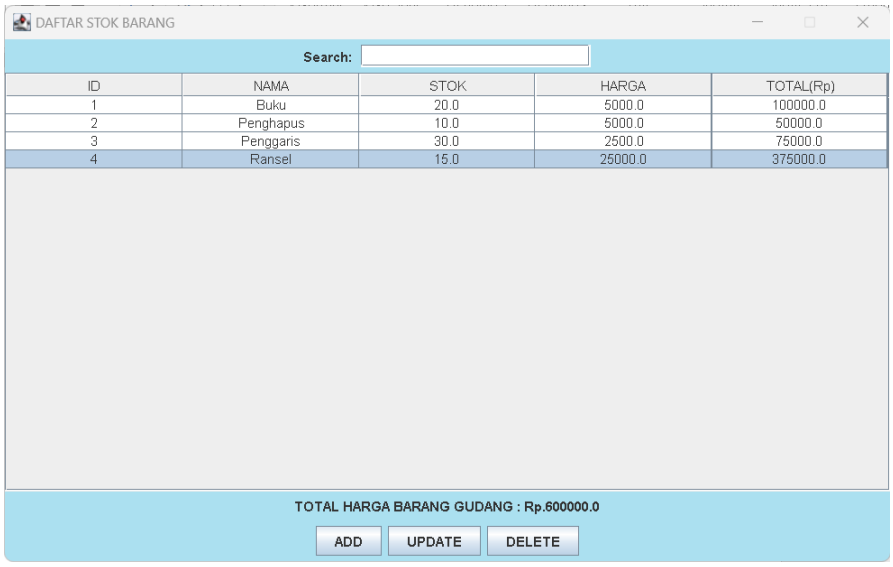


   - CSV

```
ID,NAMA,STOK,HARGA,TOTAL(Rp)
1,Buku,20.0,5000.0,100000.0
2,Penghapus,10.0,5000.0,50000.0
3,Penggaris,30.0,2500.0,75000.0
4,Ransel,15.0,25000.0,375000.0
```

C. FITUR DELETE

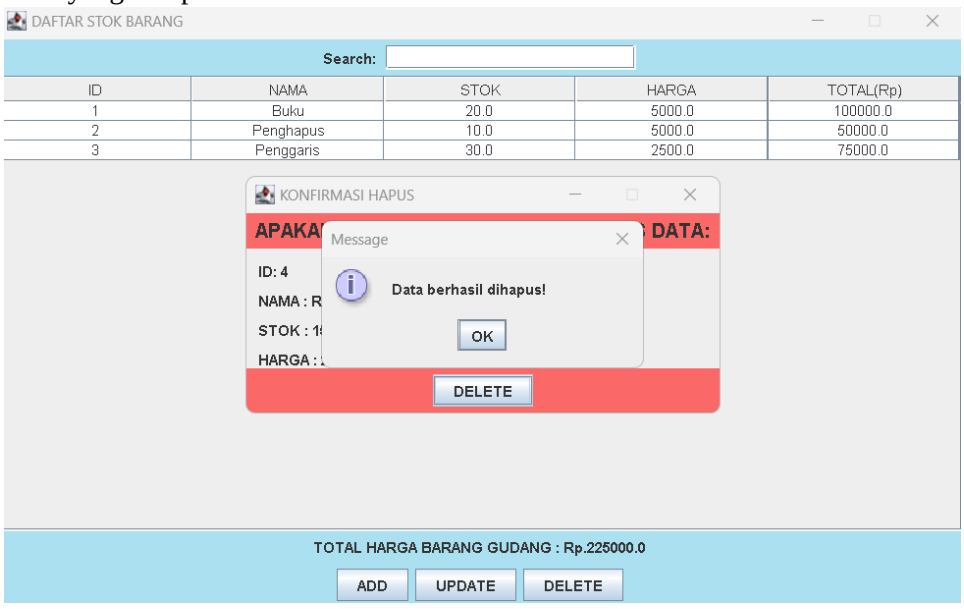
1. Data sebelum
   - Tabel



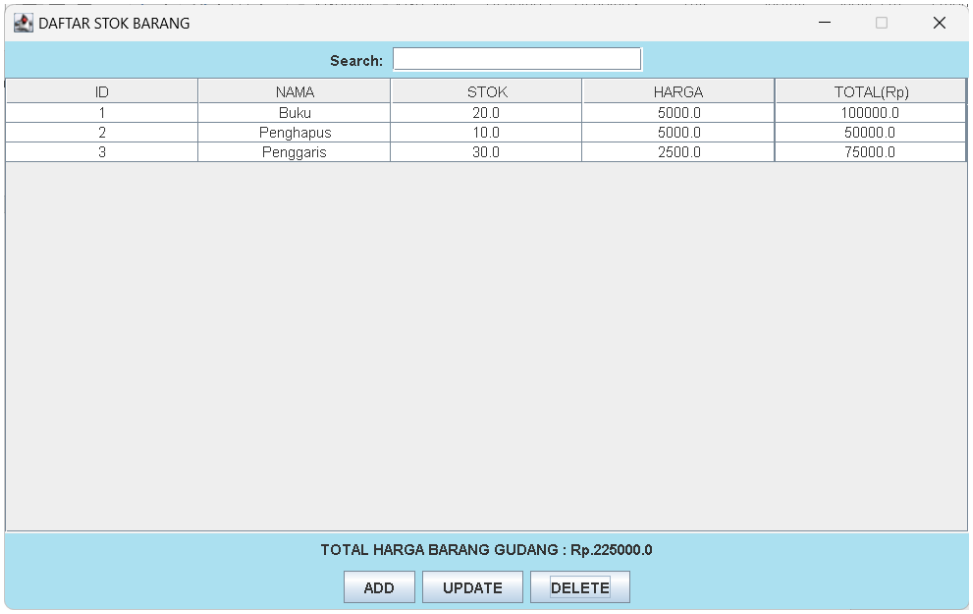
   - CSV
   -

```
ID,NAMA,STOK,HARGA,TOTAL(Rp)
1,Buku,20.0,5000.0,100000.0
2,Penghapus,10.0,5000.0,50000.0
3,Penggaris,30.0,2500.0,75000.0
4,Ransel,15.0,25000.0,375000.0
```

2. Hapus data yang sudah ada
   - Data yang dihapus : Ransel



3. Data sesudah
    - Tabel



    - CSV

```
ID,NAMA,STOK,HARGA,TOTAL(Rp)
1,Buku,20.0,5000.0,100000.0
2,Penghapus,10.0,5000.0,50000.0
3,Penggaris,30.0,2500.0,75000.0
```

BAB IV
PENUTUP

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa program pengelolaan stok barang gudang berbasis Java Swing dengan penyimpanan data menggunakan file CSV telah berhasil diimplementasikan dengan baik. Program mampu menjalankan seluruh fitur utama, yaitu penambahan data (ADD), pembaruan data (UPDATE), penghapusan data (DELETE), serta penampilan data pada tabel yang dilengkapi dengan fitur pencarian dan pengurutan sehingga memudahkan pengguna dalam mengelola stok barang secara terkomputerisas.

Melalui pengujian fitur CRUD yang melibatkan perubahan data pada tabel dan file CSV, terlihat bahwa setiap operasi yang dilakukan pada antarmuka aplikasi tercermin secara konsisten pada file penyimpanan, sehingga integritas data tetap terjaga. Ke depannya, program ini masih dapat dikembangkan lebih lanjut, misalnya dengan menambahkan fitur validasi input, pengelolaan pengguna, serta migrasi penyimpanan data ke sistem basis data seperti MySQL agar aplikasi menjadi lebih **skalabel** dan andal untuk digunakan dalam skala yang lebih besar.