

**TANBU (Tandur Buah): Sistem Kasir untuk Toko Perkebunan Buah
Rumahan**



Disusun oleh:

Aslam Thariq Akbar Akrami – A11.2021.13224

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS DIAN NUSWANTORO

Daftar Isi

I. Latar Belakang.....	3
II. Analisa Kebutuhan Dan Kebutuhan Sistem	4
2.1 Analisa Kebutuhan	4
A. Kebutuhan Pengguna	4
B. Kebutuhan Perangkat Keras	4
C. Kebutuhan Perangkat Lunak	4
2.2 Perancangan	5
A. Use Case Diagram	5
B. Class Diagram.....	5
C. Activity Diagram	6
D. Database.....	7
E. Desain Input dan Output	7
III. Implementasi Sistem.....	10
IV. Penutup	11
V. Lampiran Code Program	12
5.1 Koneksi.....	12
5.2 Frm Login.....	13
5.3 frmMenu.....	14
5.4 frmBarang.....	16
5.5 frmKonsumen.....	22
5.6 frmTransaksi.....	26
5.7 selectBarang	30
5.8 frmLapPenjualan	32
5.9 jExporttoExcel.....	36
5.10 frmBackUp	37

I. Latar Belakang

Dalam era teknologi digital yang semakin maju, sektor bisnis, khususnya toko perkebunan buah rumahan, dituntut untuk mengadopsi inovasi guna meningkatkan efisiensi proses penjualan dan manajemen data. Penjualan buah segar yang berkualitas memerlukan sistem kasir yang terintegrasi untuk mempermudah pemilik toko dalam mengelola persediaan, transaksi, dan laporan penjualan. Oleh karena itu, laporan ini bertujuan untuk mengembangkan aplikasi sistem kasir TANBU (Tandur Buah) berbasis Java dengan menggunakan platform NetBeans, yang akan memberikan solusi terpadu dalam mengoptimalkan proses bisnis dan menghadirkan manfaat yang signifikan bagi toko perkebunan buah rumahan.

Aplikasi TANBU hadir dengan sejumlah fitur yang dirancang untuk meningkatkan efisiensi dan meningkatkan performa bisnis. Fitur "Input Barang" memungkinkan pemilik toko untuk dengan mudah memasukkan data barang seperti nama, harga, stok, dan keterangan lainnya. Dengan demikian, pemantauan persediaan buah-buahan menjadi lebih efisien dan pengelolaan inventaris dapat dilakukan dengan lebih terstruktur. Selain itu, fitur "Input Konsumen" hadir untuk menghadirkan pengalaman belanja yang lebih personal. Aplikasi TANBU mencatat data konsumen, seperti nama, alamat, nomor telepon, dan informasi lainnya, sehingga pemilik toko dapat memahami preferensi dan kebutuhan pelanggan dengan lebih mendalam. Data ini dapat digunakan untuk mengoptimalkan strategi pemasaran dan meningkatkan kepuasan pelanggan. Halaman transaksi menjadi fitur utama dalam aplikasi TANBU. Melalui halaman ini, kasir dapat dengan mudah memilih buah-buahan yang dibeli oleh konsumen, menginputkan jumlah pembayaran, dan sistem akan secara otomatis menghitung kembalian yang harus diberikan. Setiap transaksi yang terekam dalam database akan memberikan gambaran yang komprehensif tentang penjualan harian, memudahkan pemantauan performa toko, dan memberikan informasi yang berharga dalam pengambilan keputusan bisnis. Laporan menjadi instrumen penting dalam mengukur performa bisnis dan menganalisis data. Aplikasi TANBU menyediakan fitur laporan yang terstruktur dan terorganisir dengan rapi. Laporan mencakup data persediaan barang, informasi konsumen, serta laporan penjualan yang dapat disesuaikan berdasarkan rentang waktu. TANBU juga memberikan perhatian pada fitur "Backup Database" yang memungkinkan pemilik toko untuk mencadangkan data secara berkala, melindungi informasi berharga dari kemungkinan kehilangan atau kerusakan, dan menjaga kehandalan data bisnis.

Melalui pendekatan pengembangan perangkat lunak berbasis metode Agile, penelitian ini berupaya untuk menciptakan aplikasi TANBU yang adaptif dan efisien. Implementasi aplikasi sistem kasir TANBU di toko perkebunan buah rumahan diharapkan dapat memberikan manfaat, termasuk peningkatan efisiensi penjualan, meningkatkan kualitas pelayanan kepada konsumen, dan meningkatkan kinerja bisnis secara keseluruhan. TANBU diharapkan dapat menjadi solusi bagi toko perkebunan buah rumahan dalam menghadapi tantangan bisnis di era digital yang membutuhkan efisiensi dalam bekerja.

II. Analisa Kebutuhan Dan Kebutuhan Sistem

2.1 Analisa Kebutuhan

A. Kebutuhan Pengguna

Daftar kebutuhan pengguna (user requirements) dari sistem kasir TANBU (Tandur Buah) untuk toko perkebunan buah rumahan, antara lain :

1. Manajemen data produk
2. Manajemen data penjual
3. Pencatatan transaksi
4. Laporan data produk
5. Laporan data penjual
6. Laporan data transaksi / penjualan
7. Cadangan database (backup database)

B. Kebutuhan Perangkat Keras

Konfigurasi Perangkat Keras (Hardware) :

1. Processor: Intel Pentium IV 1,8 GHz
2. Memory (RAM): 256 MB
3. Harddisk: 80 GB
4. 1 (satu) buah printer struk
5. UPS (Uninterruptible Power Supply)

C. Kebutuhan Perangkat Lunak

Digunakan untuk mendukung sistem informasi penjualan secara komputerisasi, antara lain:

1. Bahasa Pemrograman:

Menggunakan bahasa pemrograman Java dengan pertimbangan sebagai berikut:

- a. Bahasa pemrograman Java mempunyai fasilitas dalam penyediaan fungsi perintah dan kecepatan eksekusi.
- b. Sebagai compiler tanpa bantuan software lainnya.

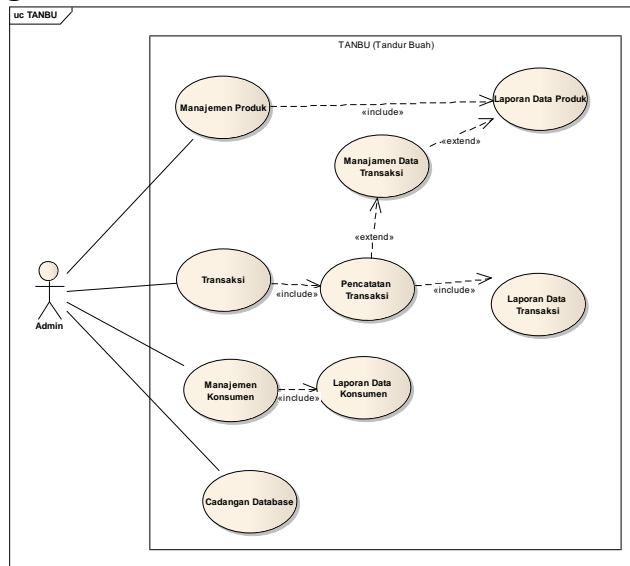
2. Text Editor: NetBeans 6.5

3. Software Aplikasi dan Software Windows:

Perangkat Lunak Aplikasi yang dibutuhkan yaitu Software Ms Office yaitu Excel sebagai alat bantu untuk digunakan oleh bagian-bagian lain untuk pengembangan pengetahuan dan ketrampilan karyawan dalam mengoperasikan komputer.

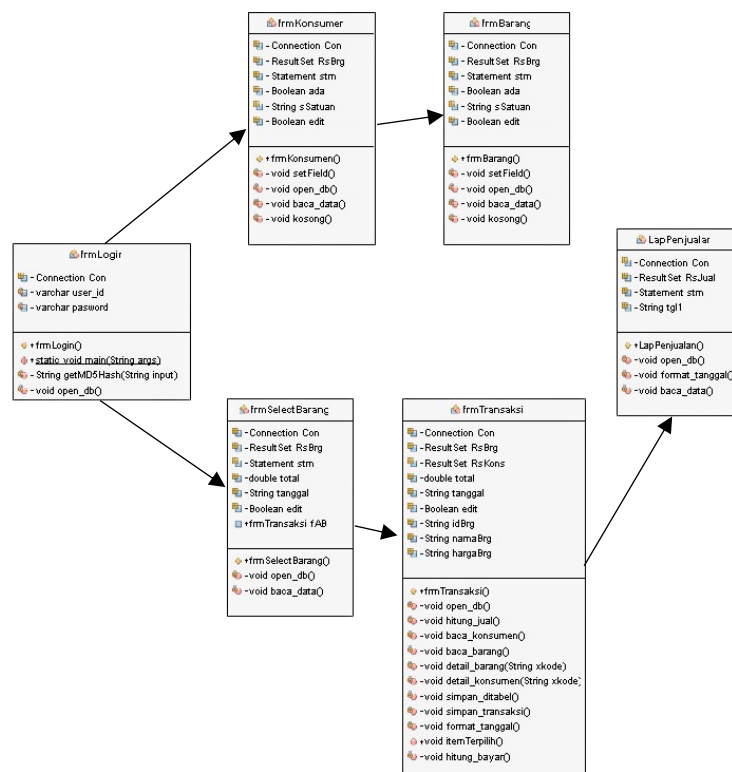
2.2 Perancangan

A. Use Case Diagram

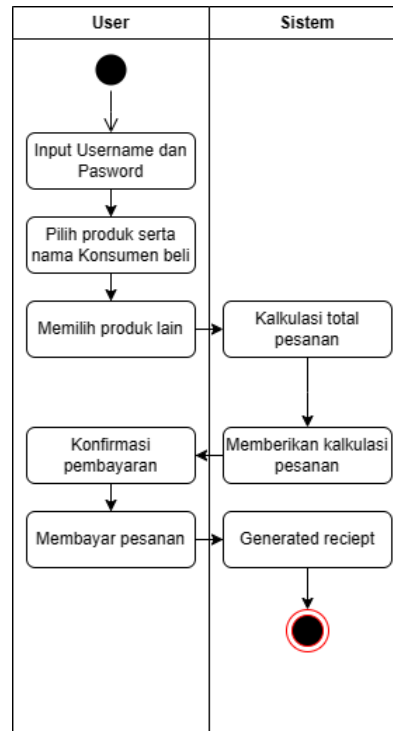


Aplikasi TANBU memiliki use case penting seperti Manajemen Data Produk, Manajemen Data Penjual, Pencatatan Transaksi, Laporan Data Produk, Laporan Data Penjual, Laporan Data Transaksi, dan Cadangan Database (Backup Database). Fitur-fitur ini memungkinkan Admin mengelola data produk dan penjual, Kasir mencatat transaksi penjualan, dan Admin melihat serta mencetak laporan data untuk pengambilan keputusan bisnis yang lebih baik. Cadangan database juga memastikan keamanan data bisnis dari kemungkinan kehilangan atau kerusakan.

B. Class Diagram



C. Activity Diagram



Dalam diagram activity dengan dua sisi, digambarkan interaksi antara pengguna (user) dan sistem dalam proses pencatatan transaksi di aplikasi TANBU (Tandur Buah) untuk toko perkebunan buah rumahan. Pada sisi pengguna (user) di sebelah kiri, proses dimulai dengan "Start Transaction" ketika pengguna memulai transaksi. Selanjutnya, pengguna memilih produk yang akan dibeli, dapat menambahkan produk lain atau menghapus produk dari transaksi. Setiap tindakan pengguna pada sisi kiri akan menimbulkan respons dari sistem pada sisi kanan. Misalnya, sistem akan menghitung total pembayaran dan memberikan struk pembayaran (Generate Receipt) setelah pengguna melakukan konfirmasi pembayaran. Proses berakhir dengan "End Transaction" yang menandakan transaksi telah selesai. Diagram activity ini memberikan gambaran yang lebih jelas tentang langkah-langkah yang diambil oleh pengguna dan bagaimana sistem merespons tindakan pengguna dalam proses pencatatan transaksi di aplikasi TANBU.

D. Database

tugasakhirpbo user <ul style="list-style-type: none">id : int(11)username : varchar(100)nama : varchar(100)role : varchar(50)password : varchar(255)	tugasakhirpbo konsumen <ul style="list-style-type: none">kd_kons : varchar(6)nm_kons : varchar(30)alm : varchar(50)kota : varchar(20)kd_pos : varchar(5)phone : varchar(15)email : varchar(30)	tugasakhirpbo djual <ul style="list-style-type: none">no_jual : varchar(10)kd_brg : char(6)harga_jual : floatjml_jual : int(4)
tugasakhirpbo barang <ul style="list-style-type: none">kd_brg : varchar(6)nm_brg : varchar(30)satuan : varchar(10)harga_jual : doublebiaya_pengadaan : doublestok : int(5)stok_min : int(5)	tugasakhirpbo jual <ul style="list-style-type: none">no_jual : varchar(10)kd_kons : char(6)tgl_jual : date	tugasakhirpbo lapenjualan <ul style="list-style-type: none">no_jual : varchar(10)tgl_jual : datenm_kons : varchar(30)kd_brg : varchar(6)nm_brg : varchar(30)harga_jual : doublejml_jual : int(4)totjual : double

E. Desain Input dan Output

a. Login

TANBU
TANDUR BUAH

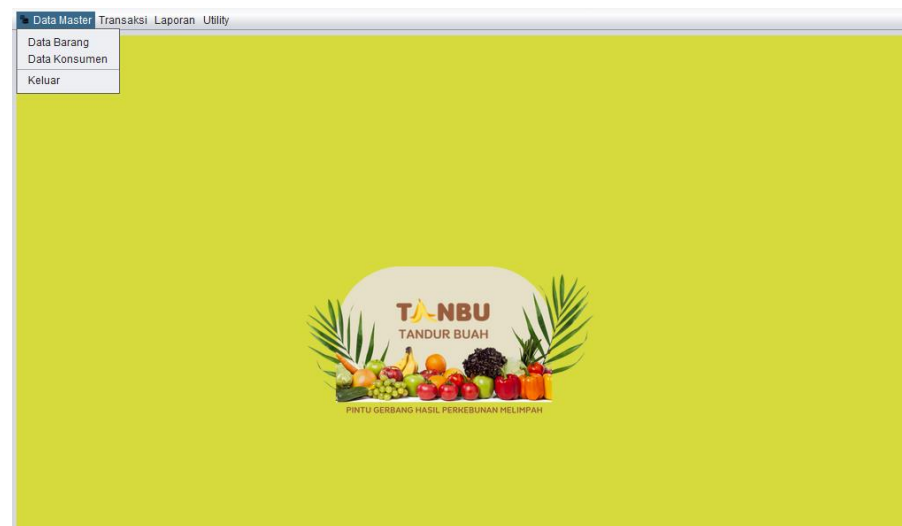
PINTU GERBANG HASIL PERKEBUNAN MELIMPAH

USER NAME

PASSWORD

LOGIN

b. Menu



c. Data Barang

TANBU

TANDUR BUAH

DATA BARANG

KODE PRODUK

B32

NAMA BUAH

Anggur

SATUAN

Kg

HARGA JUAL

35000.0

BIAYA PENGADAAN

32000.0

STOK

15

STOK MINIMAL

6

Kode	Nama	Satuan	Harga Jual	Biaya Pengadaan	Stok	Stok Min
B31	Mangga	Kg	26000.0	24000.0	10	5
B32	Anggur	Kg	35000.0	32000.0	15	6
B33	Apel	Kg	28000.0	26000.0	15	7
B34	Jeruk	Kg	24000.0	22000.0	15	8
B35	Pisang	Kg	18000.0	16000.0	20	10
B36	Nanas	Buah	30000.0	28000.0	8	4
B37	Strawberry	Gram	60000.0	55000.0	500	200
B38	Melon	Buah	45000.0	42000.0	10	5
B39	Kiwi	Buah	50000.0	47000.0	12	6
B40	Pepaya	Kg	22000.0	20000.0	25	12
B41	Semangka	Buah	18000.0	16000.0	30	15
B42	Pisang	Buah	25000.0	23000.0	20	10

TAMBAH

SIMPAN

KOREKSI

HAPUS

BATAL

KELUAR

d. Data Konsumen

TANBU

TANDUR BUAH

DATA KONSUMEN

KODE KONSUMEN

K000

NAMA KONSUMEN

Aslam

ALAMAT

Jl. Ki godek No.13

KOTA

Demak

KODE POS

59565

TELEPON

081325080083

EMAIL

aslamtharq02Gmail.com

Kode	Nama Kons...	Alamat	Kota	Kode Pos	Phone	Email
K000	Aslam	Jl. Ki godek ...	Demak	59565	081325080...	aslamtharq...
K001	Budi Santoso	Jl. Mawar N...	Jakarta Sela...	12345	081234567...	budi.santos...
K002	Siti Rahayu	Jl. Melati No...	Surabaya	67890	082345678...	siti.rahayu@...
K003	Agus Wijaya	Jl. Anggrek ...	Bandung	23456	085678901...	agus.wijaya...
K004	Davi Fitriani	Jl. Cempak ...	Yogyakarta	56789	087890123...	davifitriani...

TAMBAH

SIMPAN

KOREKSI

HAPUS

BATAL

KELUAR

e. Transaksi

TANBU

TANDUR BUAH

TRANSAKSI

No Jual6

Kode KonsumenK000

Tgl. Jual2023/07/16

Nama KonsumenAslam

B40

Pecava

22000.0

2

44000.0

Hapus Item

Pilih Barang

B40

Kd Barang	Nama Barang	Harga Barang	Jumlah	Total
B38	Melon	45000.0	3	135000.0
B40	Pepaya	22000.0	2	44000.0

njualan

: 2023-7-16

Nama Barang	Harga	Jml	Total
Melon	45000.0	3	135000.0
Pepaya	22000.0	2	44000.0

Total : 179000
Bayar : 20000
Kembali : 21000

TOTAL

179000.0

BAYAR

200000

KEMBALI

21000.0

TAMBAH

SIMPAN

BATAL

CETAK

KELUAR

f. Laporan Penjualan

TANBU

TANDUR BUAH

PENJUALAN

Tanggal2023-07-05

s/d2023-07-22

Cari

No Jual	Tanggal	Nama Konsu...	Kode Barang	Nama Barang	Harga Satuan	Jumlah	Total
1	2023-07-13	Aslam	B36	Nanas	30,000	2	60000.0
1	2023-07-13	Aslam	B40	Pepaya	22,000	1	22000.0
2	2023-07-13	Budi Santoso	B42	Pir	35,000	4	140000.0
2	2023-07-13	Budi Santoso	B56	Mangga Muda	28,000	1	28000.0
3	2023-07-13	Budi Santoso	B39	Kiwi	50,000	5	250000.0
3	2023-07-13	Budi Santoso	B38	Melon	45,000	2	90000.0
4	2023-07-13	Dewi Fitriani	B31	Mangga	26,000	5	130000.0
5	2023-07-13	Budi Santoso	B37	Strawberry	60,000	3	180000.0
5	2023-07-13	Budi Santoso	B33	Apel	28,000	2	56000.0

EXPORT

g. Laporan Stok Data Barang

TANBU

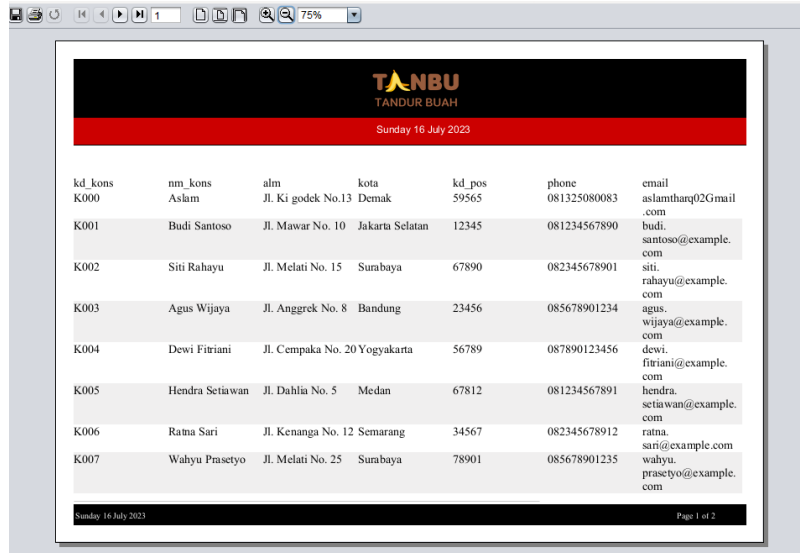
TANDUR BUAH

Sunday 16 July 2023

KODE BARANG	NAMA	SATUAN	HARGA JUAL	BIAYA	STOK	STOK MIN
B31	Mangga	Kg	26000.0	24000.0	10	5
B32	Anggur	Kg	35000.0	32000.0	15	6
B33	Apel	Kg	28000.0	26000.0	15	7
B34	Jenak	Kg	24000.0	22000.0	15	8
B35	Pisang	Kg	18000.0	16000.0	20	10
B36	Nanas	Buah	30000.0	28000.0	8	4
B37	Strawberry	Gram	60000.0	55000.0	500	200
B38	Melon	Buah	45000.0	42000.0	10	5
B39	Kiwi	Buah	50000.0	47000.0	12	6
B40	Pepaya	Kg	22000.0	20000.0	25	12
B41	Semangka	Buah	18000.0	16000.0	30	15
B42	Pir	Buah	35000.0	32000.0	20	10
B43	Markisa	Kg	40000.0	37000.0	8	4
B44	Durian	Kg	60000.0	55000.0	5	2
B45	Sinak	Buah	45000.0	42000.0	15	7
B46	Rambutan	Kg	28000.0	26000.0	20	10
B47	Belimbing	Buah	35000.0	32000.0	10	5
B48	Manggis	Kg	40000.0	37000.0	12	6
B49	Aprikot	Kg	50000.0	47000.0	8	4

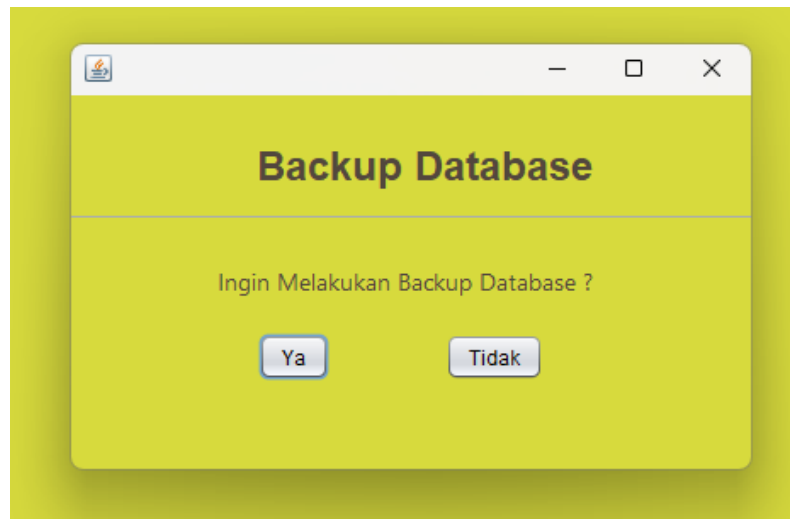
Sunday 16 July 2023Page 1 of 3

h. Laporan data Konsumen



kd_kons	nm_kons	alm	kota	kd_pos	phone	email
K000	Aslam	Jl. Ki godek No.13	Demak	59565	081325080083	aslamtharq02Gmail.com
K001	Budi Santoso	Jl. Mawar No. 10	Jakarta Selatan	12345	081234567890	budi.santoso@example.com
K002	Siti Rahayu	Jl. Melati No. 15	Surabaya	67890	082345678901	siti.rahayu@example.com
K003	Agus Wijaya	Jl. Anggrek No. 8	Bandung	23456	085678901234	agus.wijaya@example.com
K004	Dewi Fitriani	Jl. Cempaka No. 20	Yogyakarta	56789	087890123456	dewi.fitriani@example.com
K005	Hendra Setiawan	Jl. Dahlia No. 5	Medan	67812	081234567891	hendra.setiawan@example.com
K006	Ratna Sari	Jl. Kenanga No. 12	Semarang	34567	082345678912	ratna.sari@example.com
K007	Wahyu Prasetyo	Jl. Melati No. 25	Surabaya	78901	085678901235	wahyu.prasetyo@example.com

i. Backup Database



III. Implementasi Sistem

Implementasi sistem dalam aplikasi penjualan mencakup beberapa frame utama, yaitu `frmTransaksi`, `frmBarang`, dan `frmKonsumen`. Pada `frmTransaksi`, pengguna dapat melakukan transaksi penjualan dengan memasukkan data barang yang dibeli dan informasi konsumen. Setelah transaksi selesai, data tersebut akan tersimpan dalam database untuk tujuan pencatatan dan analisis bisnis.

Selanjutnya, pada `frmBarang`, sistem memungkinkan pengguna untuk mengelola inventaris barang. Dengan menggunakan formulir yang disediakan, pengguna dapat menambahkan, mengedit, atau menghapus data barang seperti kode barang, nama barang, harga beli, harga jual, dan stok barang. Semua informasi ini akan diintegrasikan dengan database aplikasi sehingga pengguna dapat dengan mudah melacak dan mengelola persediaan barang.

Seluruh data konsumen yang terkait dengan proses penjualan dapat dikelola melalui frame `frmKonsumen`. Pengguna dapat mengisi formulir untuk menambahkan, mengedit, atau menghapus data konsumen seperti nomor konsumen, nama, alamat, dan nomor telepon. Data konsumen ini akan terekam dalam database aplikasi, yang membantu dalam mengidentifikasi pelanggan dan memudahkan proses transaksi di masa depan.

Dengan keseluruhan frame yang telah diimplementasikan dalam aplikasi penjualan, sistem ini menyediakan solusi yang lengkap untuk manajemen penjualan yang efisien dan efektif. Pengguna dapat melakukan transaksi penjualan, melihat laporan penjualan, mencadangkan data, mengelola inventaris barang, serta mengelola data konsumen dengan mudah. Hal ini memberikan nilai tambah bagi proses bisnis dan pengambilan keputusan yang lebih baik bagi pengguna. Dengan adanya fitur-fitur ini, aplikasi penjualan ini diharapkan dapat meningkatkan produktivitas dan efisiensi usaha dagang yang menggunakan sistem ini.

IV. Penutup

Dalam laporan ini, telah berhasil dikembangkan sebuah aplikasi penjualan yang bertujuan untuk meningkatkan efisiensi dan manajemen proses penjualan serta inventaris barang. Aplikasi ini dilengkapi dengan beberapa fitur utama, seperti transaksi penjualan, manajemen inventaris, dan pengelolaan data konsumen. Implementasi aplikasi ini menggunakan bahasa pemrograman Java dan database MySQL, sehingga memungkinkan integrasi data yang efisien dan dapat diakses dengan mudah.

Salah satu fitur kunci dalam aplikasi penjualan ini adalah kemampuannya untuk mencatat dan mengelola data transaksi penjualan dengan akurat. Hal ini memungkinkan para pengguna, seperti toko atau usaha kecil untuk memiliki kontrol yang lebih baik dalam manajemen inventaris serta mengelola stok barang secara baik. Selain itu, fitur manajemen inventaris juga membantu mengurangi risiko kehilangan stok dan kelebihan persediaan, sehingga dapat meningkatkan efisiensi operasional.

Aplikasi penjualan ini juga dilengkapi dengan fitur laporan penjualan yang dihasilkan oleh aplikasi, sehingga dapat mendukung pengambilan keputusan yang lebih baik dalam mengelola usaha.

Meskipun aplikasi penjualan ini memiliki potensi untuk memberikan manfaat yang besar bagi para pengguna, namun tetap memiliki beberapa batasan. Pengembangan lebih lanjut dan pemeliharaan rutin diperlukan untuk menjaga kualitas dan kinerja aplikasi ini agar tetap optimal. Secara keseluruhan, aplikasi penjualan ini diharapkan dapat memberikan kontribusi positif dalam efisiensi operasional dan manajemen bisnis, terutama bagi usaha kecil dan menengah. Dengan terus mengembangkan dan meningkatkan aplikasi ini sesuai dengan kebutuhan pasar dan perkembangan teknologi, diharapkan aplikasi ini dapat membantu para pelaku usaha dalam menghadapi tantangan bisnis yang semakin kompleks dan menuntut.

V. Lampiran Code Program

5.1 Koneksi

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class KoneksiMysql {
    String url, usr, pwd, dbn;

    public KoneksiMysql(String dbn) {
        this.url = "jdbc:mysql://localhost/" + dbn;
        this.usr = "root"; // sesuaikan dengan user localhost
        this.pwd = ""; // sesuaikan dengan password localhost
    }

    public KoneksiMysql(String host, String user, String pass,
String dbn) {
        this.url = "jdbc:mysql://" + host + "/" + dbn;
        this.usr = user;
        this.pwd = pass;
    }

    public Connection getConnection() {
        Connection con = null;
        try {
            // Class.forName("com.mysql.jdbc.Driver");
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection(this.url, this.usr,
this.pwd);
        } catch (ClassNotFoundException e) {
            System.out.println("Error #1 : " + e.getMessage());
            System.exit(0);
        } catch (SQLException e) {
            System.out.println("Error #2 : " + e.getMessage());
            System.exit(0);
        }
        return con;
    }

    public static void main(String args[]) {
        KoneksiMysql kon = new KoneksiMysql("tugasakhirpbo"); //
sesuaikan dengan nama database anda
        Connection c = kon.getConnection();
    }
}
```

5.2 Frm Login

```
import java.sql.*;
import javax.swing.JOptionPane;
import java.security.MessageDigest;

public class frmLogin extends javax.swing.JFrame {

    Connection Con;

    public frmLogin() {
        initComponents();
        open_db();
    }

    // Method enkripsi password MD5
    private String getMD5Hash(String input) {
        try {
            MessageDigest md = MessageDigest.getInstance("MD5");
            byte[] messageDigest = md.digest(input.getBytes());
            StringBuilder sb = new StringBuilder();
            for (byte b : messageDigest) {
                sb.append(String.format("%02x", b));
            }
            return sb.toString();
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }

    // Method membuka database server
    private void open_db() {
        try {
            KoneksiMysql kon = new KoneksiMysql("localhost",
"root", "", "tugasakhirpbo");
            Con = kon.getConnection();
        } catch (Exception e) {
            System.out.println("Error: " + e);
        }
    }

    private void btnLoginActionPerformed(java.awt.event.ActionEvent
evt) {
        String username = txtUsr.getText();
        String password =
getMD5Hash(String.valueOf(pswd.getPassword()));

        try {
```

```

        // Query SQL untuk memeriksa username dan password pada
tabel pengguna
        String sql = "SELECT * FROM user WHERE username = ? AND
password = ?";
        PreparedStatement stmt = Con.prepareStatement(sql);
        stmt.setString(1, username);
        stmt.setString(2, password);
        ResultSet rs = stmt.executeQuery();

        if (rs.next()) {
            String role = rs.getString("role");
            frmMenu menu = new frmMenu(role);
            menu.setVisible(true);
            this.dispose();
        } else {
            JOptionPane.showMessageDialog(this, "Username /
Password Tidak Cocok!");
        }

        stmt.close();
        rs.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new frmLogin().setVisible(true);
        }
    });
}

// ...
}

```

5.3 frmMenu

```

import java.sql.Connection;
import net.sf.jasperreports.engine.*;
import net.sf.jasperreports.view.*;

public class frmMenu extends javax.swing.JFrame {

    Connection Con;
    private String role;

    public frmMenu() {

```

```

        initComponents();
    }

    public frmMenu(String role) {
        initComponents();
        this.role = role;
        setAccessBasedOnRole();
    }

    // ...

    private void
mnLapBarangActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            KoneksiMysql kon = new KoneksiMysql("localhost",
"root", "", "tugasakhirpbo");
            Con = kon.getConnection();
            String reportPath = "Cherry_Landscape.jrxml";
            JasperReport jr =
JasperCompileManager.compileReport(reportPath);
            JasperPrint jp = JasperFillManager.fillReport(jr, null,
Con);

            JasperViewer.viewReport(jp);
            Con.close();
        } catch (Exception ex) {
            ex.printStackTrace();
        }
        // TODO add your handling code here:
    }

    private void
mnLapKonsumenActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            KoneksiMysql kon = new KoneksiMysql("localhost",
"root", "", "tugasakhirpbo");
            Con = kon.getConnection();
            String reportPath = "Cherry_Landscape_1.jrxml";
            JasperReport jr =
JasperCompileManager.compileReport(reportPath);
            JasperPrint jp = JasperFillManager.fillReport(jr, null,
Con);

            JasperViewer.viewReport(jp);
            Con.close();
        } catch (Exception ex) {
            ex.printStackTrace();
        }
        // TODO add your handling code here:
    }
}

```

```

        private void
mnLapPenjualanActionPerformed(java.awt.event.ActionEvent evt) {
    new LapPenjualan().setVisible(true);
    // TODO add your handling code here:
}

private void setAccessBasedOnRole() {
    if (role.equals("admin")) {
        // Tampilkan semua menu
        mnMaster.setEnabled(true);
        mnTransaksi.setEnabled(true);
        mnLaporan.setEnabled(true);
        mnUtility.setEnabled(true);
    } else if (role.equals("user")) {
        // Hanya tampilkan menu transaksi
        mnMaster.setEnabled(false);
        mnTransaksi.setEnabled(true);
        mnLaporan.setEnabled(false);
        mnUtility.setEnabled(false);
    }
}

// ...
}

```

5.4 frmBarang

```

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JComboBox;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class frmBarang extends javax.swing.JFrame {

    Connection Con;
    ResultSet RsBrg;
    Statement stm;
    Boolean ada = false;
    String sSatuan;
    Boolean edit = false;
    private Object[][] dataTable = null;
    private String[] header = { "Kode", "Nama", "Satuan", "Harga
Jual", "Biaya Pengadaan", "Stok", "Stok Min" };

```



```

public frmBarang() {
    initComponents();
    open_db();
    baca_data();
    aktif(false);
    setTombol(true);
}

@SuppressWarnings("unchecked")
private void initComponents() {
    // Method untuk menginisialisasi komponen GUI
    // ...

    // Implementasi komponen lainnya...

    // ...
}

private void tblBrgMouseClicked(java.awt.event.MouseEvent evt)
{
    // Method untuk menangani event klik pada tabel
    setField();
}

private void cmdHapusActionPerformed(java.awt.event.ActionEvent
evt) {
    // Method untuk menangani event klik tombol Hapus
    try {
        String sql = "delete from barang where kd_brg='" +
txtKode.getText() + "'";
        stm.executeUpdate(sql);
        baca_data();

        edit = false; // set ulang edit agar form tidak masuk
ke mode edit
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

// Implementasi method-method lainnya...
// ...

public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    // ...
}

```

```

        // Method untuk membuat dan menampilkan form GUI
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new frmBarang().setVisible(true);
            }
        });
    }

    private void setField() {
        // Method untuk memindahkan data dari tabel ke form
        int row = tblBrg.getSelectedRow();
        txtKode.setText((String) tblBrg.getValueAt(row, 0));
        txtNama.setText((String) tblBrg.getValueAt(row, 1));
        cmbSatuan.setSelectedItem((String) tblBrg.getValueAt(row,
2));
        String harga_jual = Double.toString((Double)
tblBrg.getValueAt(row, 3));
        txtHargaJual.setText(harga_jual);
        String biaya_pengadaan = Double.toString((Double)
tblBrg.getValueAt(row, 4));
        txtHargaBeli.setText(biaya_pengadaan);
        String stok = Integer.toString((Integer)
tblBrg.getValueAt(row, 5));
        txtStok.setText(stok);
        String stokmin = Integer.toString((Integer)
tblBrg.getValueAt(row, 6));
        txtStokMin.setText(stokmin);
    }

    private void open_db() {
        // Method untuk membuka koneksi ke database
        try {
            KoneksiMysql kon = new KoneksiMysql("localhost",
"root", "", "tugasakhirpbo");
            Con = kon.getConnection();
        } catch (Exception e) {
            System.out.println("Error : " + e);
        }
    }

    private void baca_data() {
        // Method untuk membaca data dari tabel barang dan
memuatnya ke dalam tabel
        try {
            stm =
Con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
            RsBrg = stm.executeQuery("select * from barang");

```

```

        ResultSetMetaData meta = RsBrg.getMetaData();
        int col = meta.getColumnCount();
        int baris = 0;
        while (RsBrg.next()) {
            baris = RsBrg.getRow();
        }

        dataTable = new Object[baris][col];
        int x = 0;
        RsBrg.beforeFirst();

        while (RsBrg.next()) {
            dataTable[x][0] = RsBrg.getString("kd_brg");
            dataTable[x][1] = RsBrg.getString("nm_brg");
            dataTable[x][2] = RsBrg.getString("satuan");
            dataTable[x][3] = RsBrg.getDouble("harga_jual");
            dataTable[x][4] =
RsBrg.getDouble("biaya_pengadaan");
            dataTable[x][5] = RsBrg.getInt("stok");
            dataTable[x][6] = RsBrg.getInt("stok_min");
            x++;
        }

        tblBrg.setModel(new DefaultTableModel(dataTable,
header));
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

private void kosong() {
    // Method untuk mengosongkan isian data pada form
    txtKode.setText("");
    txtNama.setText("");
    cmbSatuan.setSelectedIndex(0);
    txtHargaJual.setText("");
    txtHargaBeli.setText("");
    txtStok.setText("");
    txtStokMin.setText("");
}

private void aktif(boolean x) {
    // Method untuk mengaktifkan atau menonaktifkan isian data
pada form
    txtKode.setEditable(x);
    txtNama.setEditable(x);
    cmbSatuan.setEnabled(x);
    txtHargaJual.setEditable(x);

```

```

        txtHargaBeli.setEditable(x);
        txtStok.setEditable(x);
        txtStokMin.setEditable(x);
    }

    private void setTombol(boolean t) {
        // Method untuk mengatur status tombol
        cmdTambah.setEnabled(t);
        cmdKoreksi.setEnabled(t);
        cmdHapus.setEnabled(t);
        cmdSimpan.setEnabled(!t);
        cmdBatal.setEnabled(!t);
        cmdKeluar.setEnabled(t);
    }

    private void initComponents() {
        // (Inisialisasi komponen GUI)

        // Event klik pada tombol Tambah
        cmdTambah.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                cmdTambahActionPerformed(evt);
            }
        });

        // Event klik pada tombol Koreksi
        cmdKoreksi.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                cmdKoreksiActionPerformed(evt);
            }
        });

        // Event klik pada tombol Hapus
        cmdHapus.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                cmdHapusActionPerformed(evt);
            }
        });

        // Event klik pada tombol Simpan
        cmdSimpan.addActionListener(new
java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent
    evt) {
        cmdSimpanActionPerformed(evt);
    }
    });

    // Event klik pada tombol Batal
    cmdBatal.addActionListener(new
    java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
    evt) {
        cmdBatalActionPerformed(evt);
        }
    });

    // Event klik pada tombol Keluar
    cmdKeluar.addActionListener(new
    java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
    evt) {
        cmdKeluarActionPerformed(evt);
        }
    });

    // (Implementasi komponen lainnya)
    }

    private void
    cmdTambahActionPerformed(java.awt.event.ActionEvent evt) {
        // Method untuk menangani event klik tombol Tambah
        kosong();
        aktif(true);
        setTombol(false);
        edit = false;
    }

    private void
    cmdKoreksiActionPerformed(java.awt.event.ActionEvent evt) {
        // Method untuk menangani event klik tombol Koreksi
        if (!txtKode.getText().equals("")) {
            aktif(true);
            setTombol(false);
            txtKode.setEditable(false);
            cmbSatuan.setEnabled(true);
            edit = true;
        } else {
            JOptionPane.showMessageDialog(null, "Pilih kode barang
    yang akan dikoreksi!");

```

```

    }
}

private void
cmdSimpanActionPerformed(java.awt.event.ActionEvent evt) {
    // Method untuk menangani event klik tombol Simpan
    if (!txtKode.getText().equals("") &&
!txtNama.getText().equals("") && !txtHargaJual.getText().equals("")
&& !txtHargaBeli.getText().equals("") &&
!txtStok.getText().equals("") && !txtStokMin.getText().equals(""))
    {
        if (edit == false) {
            simpan_data();
        } else if (edit == true) {
            koreksi_data();
        }
        kosong();
        aktif(false);
        setTombol(true);
        baca_data();
    } else {
        JOptionPane.showMessageDialog(null, "Data barang belum
lengkap! Lengkapi data barang terlebih dahulu.");
    }
}

private void cmdBatalActionPerformed(java.awt.event.ActionEvent
evt) {
    // Method untuk menangani event klik tombol Batal
    kosong();
    aktif(false);
    setTombol(true);
}

private void
cmdKeluarActionPerformed(java.awt.event.ActionEvent evt) {
    // Method untuk menangani event klik tombol Keluar
    this.dispose();
}
}

```

5.5 frmKonsumen

```

public class frmKonsumen extends javax.swing.JFrame {
    // (Deklarasi variabel dan method lainnya)

    private void initComponents() {

```

```

        // (Inisialisasi komponen GUI)

        // Event klik pada tombol Tambah
        cmdTambah.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                cmdTambahActionPerformed(evt);
            }
        });

        // Event klik pada tombol Koreksi
        cmdKoreksi.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                cmdKoreksiActionPerformed(evt);
            }
        });

        // Event klik pada tombol Hapus
        cmdHapus.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                cmdHapusActionPerformed(evt);
            }
        });

        // Event klik pada tombol Simpan
        cmdSimpan.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                cmdSimpanActionPerformed(evt);
            }
        });

        // Event klik pada tombol Batal
        cmdBatal.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                cmdBatalActionPerformed(evt);
            }
        });

        // Event klik pada tombol Keluar

```

```

        cmdKeluar.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                cmdKeluarActionPerformed(evt);
            }
        });

        // (Implementasi komponen lainnya)
    }

    // Method untuk menangani event klik tombol Tambah
    private void cmdTambahActionPerformed(java.awt.event.ActionEvent
evt) {
        aktif(true);
        setTombol(false);
        kosong();
    }

    // Method untuk menangani event klik tombol Koreksi
    private void
cmdKoreksiActionPerformed(java.awt.event.ActionEvent evt) {
        edit = true;
        aktif(true);
        setTombol(false);
        txtKodeKonsumen.setEditable(false);
    }

    // Method untuk menangani event klik tombol Hapus
    private void cmdHapusActionPerformed(java.awt.event.ActionEvent
evt) {
        try {
            String sql = "delete from konsumen where kd_kons='" +
txtKodeKonsumen.getText() + "'";
            stm.executeUpdate(sql);
            baca_data();
            edit = false; // set ulang edit agar form tidak masuk ke
mode edit
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }

    // Method untuk menangani event klik tombol Simpan
    private void cmdSimpanActionPerformed(java.awt.event.ActionEvent
evt) {
        String tKodeKons = txtKodeKonsumen.getText();
        String tNama = txtNama.getText();

```



```

        String tAlamat = txtAlamat.getText();
        String tKota = txtKota.getText();
        String tKodePos = txtKodePos.getText();
        String tTelepon = txtTelepon.getText();
        String tEmail = txtEmail.getText();

        try {
            if (edit == true) {
                stm.executeUpdate("update konsumen set nm_kons='" +
tNama + "'," + "alm='" + tAlamat + "',kota='" + tKota + "',kd_pos='"
+ tKodePos + "',phone='" + tTelepon + "',email='" + tEmail + "'
where kd_kons='" + tKodeKons + "'");
            } else {
                stm.executeUpdate("INSERT into konsumen(kd_kons,
nm_kons, alm, kota, kd_pos, phone, email) " + "VALUES(' " + tKodeKons
+ "'," + tNama + "'," + tAlamat + "'," + tKota + "'," + tKodePos
+ "'," + tTelepon + "'," + tEmail + "')");
            }

            tblBrg.setModel(new DefaultTableModel(dataTable,
header));
            baca_data();
            aktif(false);
            setTombol(true);
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }

    // Method untuk menangani event klik tombol Batal
    private void cmdBatalActionPerformed(java.awt.event.ActionEvent
evt) {
        kosong();
        aktif(false);
        setTombol(true);
    }

    // Method untuk menangani event klik tombol Keluar
    private void cmdKeluarActionPerformed(java.awt.event.ActionEvent
evt) {
        dispose();
    }

    // (Bagian kode selanjutnya)
}

```

5.6 frmTransaksi

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.text.MessageFormat;
import java.util.Calendar;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import net.sf.jasperreports.engine.*;
import net.sf.jasperreports.view.JasperViewer;

public class frmTransaksi extends JFrame {

    Connection Con;
    // Semua variabel dan kode lainnya...

    // Konstruktor frmTransaksi dan kode lainnya...

    // Semua method lainnya...

    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new frmTransaksi().setVisible(true);
            }
        });
    }

    // method membuka database server, user, pass, database
    disesuaikan
    private void open_db() {
        try {
            KoneksiMysql kon = new KoneksiMysql("localhost", "root",
            "", "tugasakhirpbo");
            Con = kon.getConnection();
            // System.out.println("Berhasil ");
        } catch (Exception e) {
            System.out.println("Error : " + e);
        }
    }

    // method hitung penjualan dan kode lainnya...

    // method baca data konsumen dan kode lainnya...

    private void format_tanggal() {
        String DATE_FORMAT = "yyyy-MM-dd";
    }
}
```

```

        java.text.SimpleDateFormat sdf = new
java.text.SimpleDateFormat(DATE_FORMAT);
        Calendar c1 = Calendar.getInstance();
        int year = c1.get(Calendar.YEAR);
        int month = c1.get(Calendar.MONTH) + 1;
        int day = c1.get(Calendar.DAY_OF_MONTH);
        tanggal = Integer.toString(year) + "-" +
Integer.toString(month) + "-" + Integer.toString(day);
    }

    // Method untuk generate laporan menggunakan JasperReports
    public void generateReport() {
        try {
            // Lokasi file .jrxml laporan
            String sourceFileName =
"lokasi_laporan/transaksi_report.jrxml";

            // Membuat objek JasperReport dari file .jrxml
            JasperReport jasperReport =
JasperCompileManager.compileReport(sourceFileName);

            // Membuat objek JasperPrint dengan menggunakan
JasperReport dan data dari database
            JasperPrint jasperPrint =
JasperFillManager.fillReport(jasperReport, null, Con);

            // Tampilkan laporan menggunakan JasperViewer
            JasperViewer.viewReport(jasperPrint);
        } catch (JRException e) {
            e.printStackTrace();
        }
    }

    // Method untuk mendapatkan koneksi ke database
    public Connection getConnection() {
        // Kode untuk mendapatkan koneksi ke database
        // Gantilah dengan kode koneksi database Anda sendiri
        return Con;
    }

    // Method untuk menampilkan laporan dalam bentuk dialog
    private void tampilkanLaporan() {
        format_tanggal();
        String ctk = "Nota Penjualan\nNo:" + txtNoJual.getText() +
"\nTanggal : " + tanggal;
        ctk = ctk + "\n"

```

```

+ "-----"
-----";
    ctk = ctk + "\n" + "Kode\tNama Barang\t\tHarga\tJml\tTotal";
    ctk = ctk + "\n"
+ "-----"
-----";

    for (int i = 0; i < tblJual.getRowCount(); i++) {
        String xkd = (String) tblJual.getValueAt(i, 0);
        String xnama = (String) tblJual.getValueAt(i, 1);
        double xhrgh = (Double) tblJual.getValueAt(i, 2);
        int xjml = (Integer) tblJual.getValueAt(i, 3);
        double xtota = (Double) tblJual.getValueAt(i, 4);
        ctk = ctk + "\n" + xkd + "\t" + xnama + "\t\t" + xhrgh +
"\t" + xjml + "\t" + xtota;
    }

    ctk = ctk + "\n"
+ "-----"
-----";

    ctk = ctk + "\n\t\t\t\t\t" + "Total : " +
txtTotal.getText();
    ctk = ctk + "\n\t\t\t\t\t" + "Bayar : " +
txtBayar.getText();
    ctk = ctk + "\n\t\t\t\t\t" + "Kembali : " +
txtKembali.getText();
    ctk = ctk + "\n"
+ "-----"
-----";

    text.setText(ctk);

    String headerField = "";
    String footerField = "";
    MessageFormat header = new MessageFormat(headerField);
    MessageFormat footer = new MessageFormat(footerField);
    boolean interactive = true; // interactiveCheck.isSelected();
    boolean background = true; // backgroundCheck.isSelected();
    PrintingTask task = new PrintingTask(header, footer,
interactive);

    if (background) {
        task.execute();
    } else {
        task.run();
    }

```

```

    }
}

// Inner class PrintingTask untuk mencetak laporan
private class PrintingTask extends SwingWorker<Object, Object> {
    private final MessageFormat headerFormat;
    private final MessageFormat footerFormat;
    private final boolean interactive;
    private volatile boolean complete = false;
    private volatile String message;

    public PrintingTask(MessageFormat header, MessageFormat
footer, boolean interactive) {
        this.headerFormat = header;
        this.footerFormat = footer;
        this.interactive = interactive;
    }

    @Override
    protected Object doInBackground() {
        try {
            complete = text.print(headerFormat, footerFormat,
true, null, null, interactive);
            message = "Printing " + (complete ? "complete" :
"canceled");
        } catch (PrinterException ex) {
            message = "Sorry, a printer error occurred";
        } catch (SecurityException ex) {
            message = "Sorry, cannot access the printer due to
security reasons";
        }
        return null;
    }

    @Override
    protected void done() {
        showMessage(!complete, message);
    }
}

private void showMessage(boolean isError, String message) {
    if (isError) {
        JOptionPane.showMessageDialog(null, message, "Error",
JOptionPane.ERROR_MESSAGE);
    } else {
        JOptionPane.showMessageDialog(null, message,
"Information", JOptionPane.INFORMATION_MESSAGE);
    }
}

```

```
}  
}
```

5.7 selectBarang

```
import java.sql.Connection;  
import java.sql.ResultSet;  
import java.sql.ResultSetMetaData;  
import java.sql.SQLException;  
import java.sql.Statement;  
import javax.swing.JOptionPane;  
import javax.swing.table.DefaultTableModel;  
  
public class frmSelectBarang extends javax.swing.JFrame {  
  
    // Semua variabel dan kode sebelumnya...  
  
    public frmSelectBarang() {  
        initComponents();  
        baca_data();  
    }  
  
    // Kode GUI lainnya...  
  
    private void tblBrgMouseClicked(java.awt.event.MouseEvent evt) {  
        int row = tblBrg.getSelectedRow();  
        int tabelBarang = tblBrg.getSelectedRow();  
        fAB.idBrg = tblBrg.getValueAt(tabelBarang, 0).toString();  
        fAB.namaBrg = tblBrg.getValueAt(tabelBarang, 1).toString();  
        fAB.hargaBrg = tblBrg.getValueAt(tabelBarang, 3).toString();  
        fAB.itemTerpilih();  
        this.dispose();  
    }  
  
    // Kode lainnya...  
  
    public static void main(String args[]) {  
        java.awt.EventQueue.invokeLater(new Runnable() {  
            public void run() {  
                new frmSelectBarang().setVisible(true);  
            }  
        });  
    }  
  
    // Kode lainnya...  
  
    private void open_db() {  
        try {
```

```

        KoneksiMysql kon = new KoneksiMysql("localhost", "root",
"", "tugasakhirpbo");
        Con = kon.getConnection();
    } catch (Exception e) {
        System.out.println("Error : " + e);
    }
}

private void baca_data() {
    open_db();
    try {
        stm =
Con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
        RsBrg = stm.executeQuery("select * from barang order by
kd_brg");

        ResultSetMetaData meta = RsBrg.getMetaData();
        int col = meta.getColumnCount();
        int baris = 0;
        while (RsBrg.next()) {
            baris = RsBrg.getRow();
        }

        dataTable = new Object[baris][col];
        int x = 0;
        RsBrg.beforeFirst();
        while (RsBrg.next()) {
            dataTable[x][0] = RsBrg.getString("kd_brg");
            dataTable[x][1] = RsBrg.getString("nm_brg");
            dataTable[x][2] = RsBrg.getString("satuan");
            dataTable[x][3] = RsBrg.getDouble("harga_jual");
            dataTable[x][4] =
RsBrg.getDouble("biaya_pengadaan");
            dataTable[x][5] = RsBrg.getInt("stok");
            dataTable[x][6] = RsBrg.getInt("stok_min");
            x++;
        }
        tblBrg.setModel(new DefaultTableModel(dataTable,
header));
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

// Variables declaration - do not modify
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;

```

```

private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable tblBrg;
// End of variables declaration
}

```

5.8 frmLapPenjualan

```

package TUGASUASPB0;

import java.io.File;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;

public class LapPenjualan extends JFrame {
    Connection Con;
    ResultSet RsJual;
    Statement stm;
    String tgl1, tgl2;
    private Object[][] dataTable = null;
    private String[] header = { "No Jual", "Tanggal", "Nama
Konsumen", "Kode Barang", "Nama Barang", "Harga Satuan",
        "Jumlah", "Total" };
    DefaultTableModel tableModel = new DefaultTableModel(new
Object[][] {}, header);

    JLabel jLabel1 = new JLabel("Tanggal");
    JLabel jLabel2 = new JLabel("s/d");
    JButton cmdCari = new JButton("Cari");
    JDateChooser dtAwal = new JDateChooser();
    JDateChooser dtAkhir = new JDateChooser();
    JTable tblLapJual = new JTable();

    JButton btn_Export = new JButton("Export");

```



```

public LapPenjualan() {
    setSize(800, 500);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    setTitle("Laporan Penjualan");
    getContentPane().setBackground(new Color(215, 218, 61));
    setLayout(null);
    setVisible(true);

    setLocationRelativeTo(null);

    add(jLabel1);
    jLabel1.setBounds(20, 20, 100, 20);

    add(dtAwal);
    dtAwal.setBounds(150, 20, 120, 20);

    add(jLabel2);
    jLabel2.setBounds(280, 20, 30, 20);

    add(dtAkhir);
    dtAkhir.setBounds(310, 20, 120, 20);

    add(cmdCari);
    cmdCari.setBounds(450, 20, 80, 20);

    add(new JScrollPane(tblLapJual));
    tblLapJual.setBounds(20, 60, 750, 300);

    add(btn_Export);
    btn_Export.setBounds(20, 370, 100, 20);

    cmdCari.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            baca_data();
        }
    });

    btn_Export.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            btn_ExportActionPerformed(e);
        }
    });

    java.text.SimpleDateFormat sdf = new
java.text.SimpleDateFormat("yyyy-MM-dd");
    dtAwal.setDateFormatString("yyyy-MM-dd");
    dtAkhir.setDateFormatString("yyyy-MM-dd");
    open_db();
}

```

```

    }

    private void open_db() {
        try {
            KoneksiMysql kon = new KoneksiMysql("localhost", "root",
            "", "tugasakhirpbo");
            Con = kon.getConnection();
        } catch (Exception e) {
            System.out.println("Error : " + e);
        }
    }

    private void format_tanggal() {
        String DATE_FORMAT = "yyyy-MM-dd";
        java.text.SimpleDateFormat sdf = new
        java.text.SimpleDateFormat(DATE_FORMAT);
        Calendar c1 = Calendar.getInstance();
        int year = c1.get(Calendar.YEAR);
        int month = c1.get(Calendar.MONTH) + 1;
        int day = c1.get(Calendar.DAY_OF_MONTH);

        Date tglAwal = dtAwal.getDate();
        Date tglAkhir = dtAkhir.getDate();
        tgl1 = sdf.format(tglAwal);
        tgl2 = sdf.format(tglAkhir);

        System.out.println("Tanggal : " + tgl1);
        System.out.println("Tanggal2 : " + tgl2);
    }

    private void baca_data() {
        try {
            tgl1 = new SimpleDateFormat("yyyy-MM-
            dd").format(dtAwal.getDate());
            tgl2 = new SimpleDateFormat("yyyy-MM-
            dd").format(dtAkhir.getDate());
            stm =
            Con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_READ_ONLY);
            RsJual = stm.executeQuery("select * from lappenjualan
            where date(tgl_jual)>=" + "date('" + tgl1
            + "') and date(tgl_jual)<=" + "date('" + tgl2 +
            "') order by tgl_jual desc");
            ResultSetMetaData meta = RsJual.getMetaData();
            int col = meta.getColumnCount();
            int baris = 0;
            if (RsJual.last()) {
                baris = RsJual.getRow();
            }
        }
    }

```

```

        RsJual.beforeFirst();
    }
    CustomFormat cf = new CustomFormat();
    dataTable = new Object[baris][col];
    int x = 0;
    while (RsJual.next()) {
        dataTable[x][0] = RsJual.getString("no_jual");
        dataTable[x][1] = RsJual.getDate("tgl_jual");
        dataTable[x][2] = RsJual.getString("nm_kons");
        dataTable[x][3] = RsJual.getString("kd_brg");
        dataTable[x][4] = RsJual.getString("nm_brg");
        dataTable[x][5] = String.format("%.0f",
RsJual.getDouble("harga_jual"));
        dataTable[x][6] = RsJual.getInt("jml_jual");
        dataTable[x][7] = RsJual.getDouble("totjual");
        x++;
    }
    tblLapJual.setModel(new DefaultTableModel(dataTable,
header));
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e);
    } finally {
        if (RsJual != null) {
            try {
                RsJual.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        if (stm != null) {
            try {
                stm.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}

private void btn_ExportActionPerformed(ActionEvent evt) {
    try {
        ExporttoExcel ex = new ExporttoExcel(tblLapJual, new
File("D:\\TUGASUASPB0\\Data Penjualan\\DataPenjualan.xls"));
        JOptionPane.showMessageDialog(null, "Sukses Export
data.....");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

```

```

    }

    private static class CustomFormat {

        public CustomFormat() {
        }

    }

    public static void main(String[] args) {
        new LapPenjualan();
    }
}

```

5.9 jExporttoExcel

```

import java.io.File;
import java.io.FileWriter;
import javax.swing.JTable;
import javax.swing.table.TableModel;

public class ExporttoExcel {
    public ExporttoExcel(JTable table, File file) {
        try {
            TableModel tableModel = table.getModel();
            FileWriter fOut = new FileWriter(file);
            for (int i = 0; i < tableModel.getColumnCount(); i++) {
                fOut.write(tableModel.getColumnName(i) + "\t");
            }
            fOut.write("\n");
            for (int i = 0; i < tableModel.getRowCount(); i++) {
                for (int j = 0; j < tableModel.getColumnCount();
j++) {
                    fOut.write(tableModel.getValueAt(i,
j).toString() + "\t");
                }
                fOut.write("\n");
            }
            fOut.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

5.10 frmBackUp

```
import java.io.IOException;
import java.util.logging.*;
import javax.swing.*;

public class frmBackUp extends javax.swing.JFrame {

    public frmBackUp() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
Code"> //GEN-BEGIN: initComponents
    private void initComponents() {

        cmdYa = new javax.swing.JButton();
        cmdTidak = new javax.swing.JButton();
        jSeparator1 = new javax.swing.JSeparator();
        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON
_CLOSE);
        setBackground(new java.awt.Color(255, 204, 204));
        getContentPane().setLayout(new
org.netbeans.lib.awtextra.AbsoluteLayout());

        cmdYa.setText("Ya");
        cmdYa.addActionListener(new java.awt.event.ActionListener()
{
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                cmdYaActionPerformed(evt);
            }
        });
        getContentPane().add(cmdYa, new
org.netbeans.lib.awtextra.AbsoluteConstraints(110, 140, -1, -1));

        cmdTidak.setText("Tidak");
        cmdTidak.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                cmdTidakActionPerformed(evt);
            }
        });
    }
}
```

```

        getContentPane().add(cmdTidak, new
org.netbeans.lib.awtextra.AbsoluteConstraints(220, 140, -1, -1));
        getContentPane().add(jSeparator1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(0, 70, 400, 10));

        jPanel1.setBackground(new java.awt.Color(215, 218, 61));

        jLabel1.setFont(new java.awt.Font("Showcard Gothic", 1,
24));
        jLabel1.setForeground(new java.awt.Color(82, 71, 61));
        jLabel1.setText("Backup Database");

        jLabel2.setFont(new java.awt.Font("Segoe UI", 0, 14));
        jLabel2.setForeground(new java.awt.Color(82, 71, 61));
        jLabel2.setText("Ingin Melakukan Backup Database ?");

        javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout
t.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(86, 86, 86)
                    .addGroup(jPanel1Layout.createParallelGroup(javax.sw
ing.GroupLayout.Alignment.TRAILING)
                        .addComponent(jLabel2)
                        .addComponent(jLabel1))
                    .addGap(85, Short.MAX_VALUE))
                );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout
t.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(25, 25, 25)
                    .addComponent(jLabel1)
                    .addGap(42, 42, 42)
                    .addComponent(jLabel2)
                    .addGap(102, Short.MAX_VALUE))
                );

        getContentPane().add(jPanel1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 400, 220));

        pack();
    } // </editor-fold> //GEN-END: initComponents

```

```

        private void cmdYaActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_cmdYaActionPerformed
            try {
                String[] executeCmd = new
String[]{"C:\\xampp\\mysql\\bin\\mysqldump", "-uroot",
"tugasakhipbo", "-r", "C:\\NetBeans
SDK\\BackupDB\\tugasakhipbo.sql"};
                Process runtime = Runtime.getRuntime().exec(executeCmd);
                int prosescomplete = runtime.waitFor();
                if (prosescomplete == 0) {
                    JOptionPane.showMessageDialog(rootPane, "Backup
sukses!");
                } else {
                    JOptionPane.showMessageDialog(rootPane, "Backup
gagal!");
                }
            } catch (IOException | InterruptedException ex) {
                Logger.getLogger(frmBackUp.class.getName()).log(Level.SE
VERE, null, ex);
            }
        } //GEN-LAST:event_cmdYaActionPerformed

        private void cmdTidakActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_cmdTidakActionPerformed
            dispose();
        } //GEN-LAST:event_cmdTidakActionPerformed

        public static void main(String args[]) {
            try {
                for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                    if ("Nimbus".equals(info.getName())) {
                        javax.swing.UIManager.setLookAndFeel(info.getClas
sName());
                        break;
                    }
                }
            } catch (ClassNotFoundException ex) {
                java.util.logging.Logger.getLogger(frmBackUp.class.getNa
me()).log(java.util.logging.Level.SEVERE, null, ex);
            } catch (InstantiationException ex) {
                java.util.logging.Logger.getLogger(frmBackUp.class.getNa
me()).log(java.util.logging.Level.SEVERE, null, ex);
            } catch (IllegalAccessException ex) {
                java.util.logging.Logger.getLogger(frmBackUp.class.getNa
me()).log(java.util.logging.Level.SEVERE, null, ex);
            } catch (javax.swing.UnsupportedLookAndFeelException ex) {

```

```
        java.util.logging.Logger.getLogger(frmBackUp.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }

    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new frmBackUp().setVisible(true);
        }
    });
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton cmdTidak;
private javax.swing.JButton cmdYa;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JPanel jPanel1;
private javax.swing.JSeparator jSeparator1;
// End of variables declaration//GEN-END:variables
}
```