

LAPORAN DOKUMENTASI TUGAS BESAR

DASAR PEMROGRAMAN 2

Diajukan untuk memenuhi Tugas Besar & UAS mata kuliah Dasar Pemrograman 2.

Dosen : Beri Noviansyah, M.Kom



Dibuat oleh :

Muhamad Rifqi Rahmatulloh (1942538)

IF-F

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN
KOMPUTER(STMIK)"AMIK BANDUNG"**

Jl.Jakarta No.28 Bandung 40272

Telp./Fax.022 727 1136

www.stmik-amikbandung.ac.id

Deskripsi Program

Nama: Program penjualan buku di toko berbasis gui

Deskripsi: Program penjualan buku ini membantu untuk transaksi penjualan buku di toko buku dengan mengelola data yang terdapat di toko buku.

Hak akses:

- Kasir dan pemilik toko

Fitur Halaman Kasir:

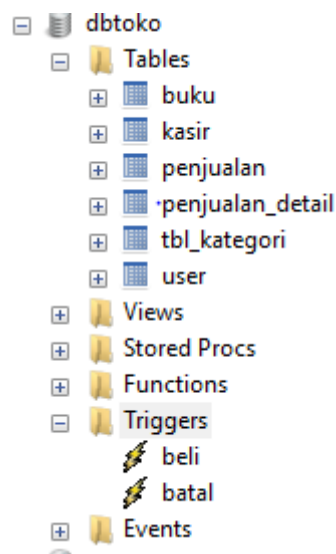
- CRUD DATA BUKU
- CRUD DATA KATEGORI BUKU
- KELOLA DATA PENJUALAN
- CETAK NOTA

Fitur Halaman Pemilik:

- CRUD DATA BUKU
- CRUD DATA KATEGORI BUKU
- CRUD DATA KASIR
- CRUD DATA USER
- MELIHAT DATA PENJUALAN
- REKAP CETAK DATA TRANSAKSI PENJUALAN

Bagian 1. Database

Adapun Nama database proyek saya yaitu dbtoko dan beberapa tabel dan menggunakan trigger untuk mengurangi stok otomatis sebagai berikut:



Gambar 1.1. Tabel

buku	
Columns	
	id_buku, int(11)
	no_isbn, varchar(13)
	judul, varchar(50)
	pengarang, varchar(40)
	penerbit, varchar(40)
	tahun_terbit, varchar(10)
	kategori, varchar(40)
	stok, int(11)
	harga_pokok, int(11)
	harga_jual, int(11)

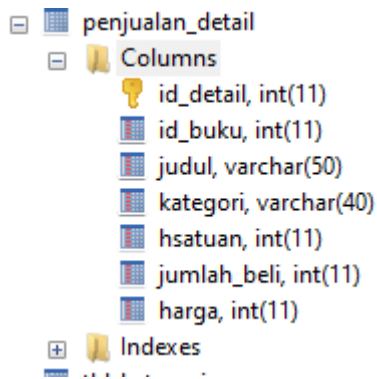
field tabel buku

kasir	
Columns	
	id_kasir, int(11)
	nama_kasir, varchar(40)
	alamat, text
	no_telp, varchar(13)
	email, varchar(40)
	Indexes

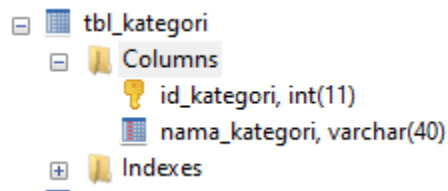
field tabel kasir

penjualan	
Columns	
	id_transaksi, varchar(20)
	id_buku, int(11)
	tanggal, date
	judul, varchar(50)
	kategori, varchar(40)
	hsatuan, int(11)
	jumlah_beli, int(11)
	harga, int(11)
	bayar, int(11)
	kembalian, int(11)
	Indexes

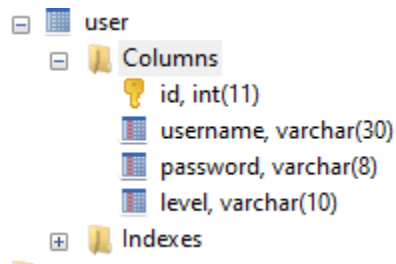
Field tabel penjualan



field tabel penjualan_detail



field tbl_kategori



field tabel user

Bagian2.Dokumentasi Program:

```
public class koneksi {
    Connection koneksi=null;
    public static Connection koneksiDb() {
        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection koneksi = (Connection) DriverManager.getConnection("jdbc:mysql://localhost/dbtoko","root","");
            return koneksi;
        }catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
            return null;
        }
    }
}
```

Sorce code koneksi.java berfungsi sebagai koneksi atau penghubung ke database dbtoko untuk menjadi penghubung agar nanti data di database dapat di tampilkan atuupun disimpan dll.

```

public class Toko {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        login log = new login();
        log.setVisible(true);
    }

}

```

Source code class toko sebagai class main class yang di jalankan pertama saat aplikasi di running di samping berfungsi sebagai class utama dimana di setting untuk running dan frame yang akan di tampilkan pertama adalah frame login

Login



Source code login untuk mengecek data user ada atau tidak di tabel user di databasedengan mengecek username ,password,mengecek level untuk masuk apakah kasir atau penilik.dan memberika pemberitahuan jika password atau

```

Connection kon = koneksi.koneksiDb();

```

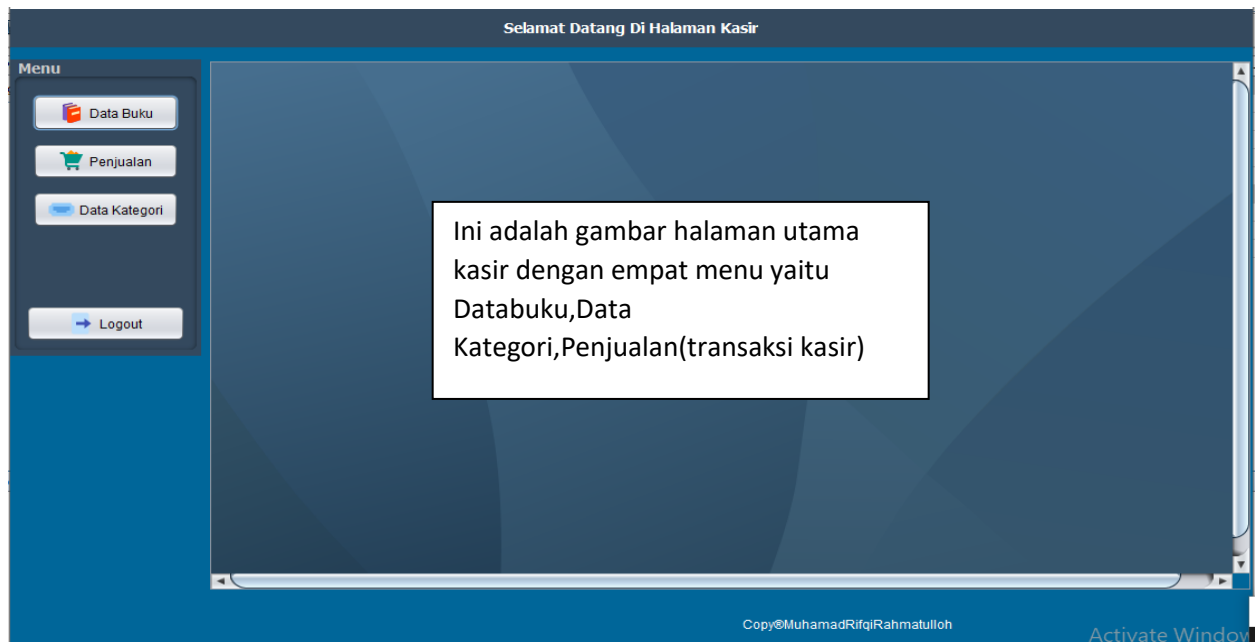
```

try {
    Statement st = kon.createStatement();
    String sql = "SELECT * FROM user where username = '"+txUser.getText()+"' and password = '"+txPass.getText()+"'";
    ResultSet rs = st.executeQuery(sql);

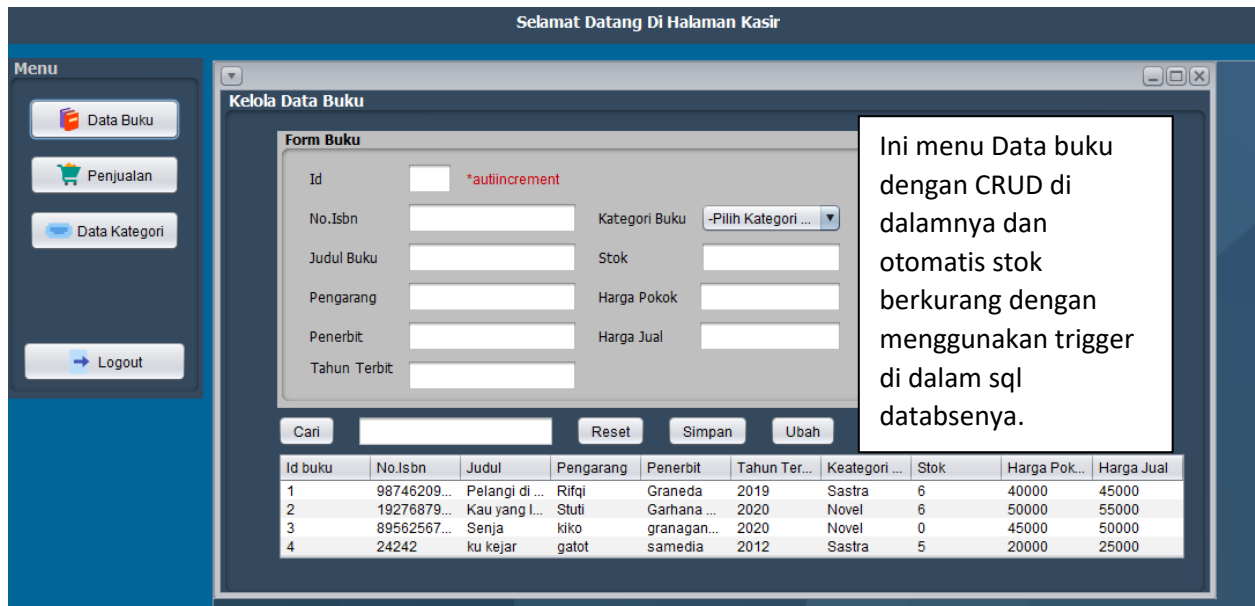
    if(rs.next()){
        if(rs.getString("level").equals("kasir")){
            Menu_kasir fa = new Menu_kasir();
            fa.show();
            this.dispose();
        }else if(rs.getString("level").equals("pemilik")){
            Menu_pemilik fu = new Menu_pemilik();
            fu.show();
            this.dispose();
        }
    }
} else{
    JOptionPane.showMessageDialog(null, "Maaf password atau username anda salah");
    txUser.setText("");
    txPass.setText("");
    txUser.requestFocus();
}
}

```

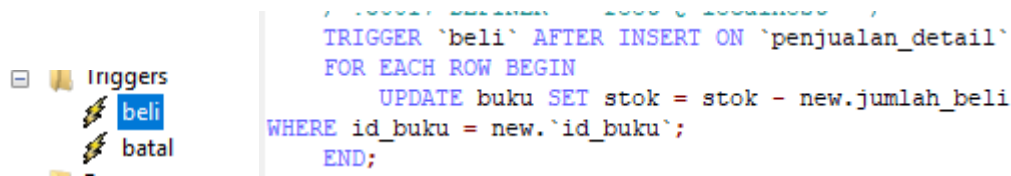
Halaman Kasir



Menu Data Buku



Trigger



Beli adalah nama trigger untuk pengurangan stok di data buku dan batal itu untuk mereturn jika batal transaksi. dan untuk sql querynya after insert setelah insert di penjualan_detail maka update buku stok = stok - new.jumlah_beli dimana id_buku = new.id buku nah disini stok adalah field di tabel buku dan jumlah_beli adalah field di tabel penjualan_detail jadi stok di kurangi inputan di jumlah_beli.

Id Kasir	Nama Kasir	Alamat	No.Telp	Email
1	ade	jl.ibrhim aji no....	0897451892	

Source Code cari data

```
private void cari() {  
    //menampilkan data database kedalam tabel  
    DefaultTableModel model = new DefaultTableModel();  
    model.addColumn("Id Kasir");  
    model.addColumn("Nama Kasir");  
    model.addColumn("Alamat");  
    model.addColumn("No.Telp");  
    model.addColumn("Email");  
    try {  
        String cari = txtcarikasih.getText();  
        String sql = "select * from kasir where nama_kasir like '%" + cari + "%'";  
        java.sql.Statement stm = kon.createStatement();  
        java.sql.ResultSet res = stm.executeQuery(sql);  
        while (res.next()) {  
            model.addRow(new Object[] { res.getString(1), res.getString(2), res.getString(3), res.getString(4) });  
        }  
        tabelkasir.setModel(model);  
    } catch (Exception e) {  
        JOptionPane.showMessageDialog(null, "Data tidak ditemukan");  
    }  
}
```

Source code cari data disini saya menselect menampilkan data yang di cari dimana data yang di cari berdasarkan nama kasir, dan untuk txtcarikasih itu nama variable field cari atau form yang di isi buat nyari datanya.

Source Code button CRUD (create, read, update, delete)

Simpan

```
private void btnsimpanActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    if (txtnamakas.equals(""))  
    {  
        JOptionPane.showMessageDialog(null, "Mohon lengkapi data nama kasir ");  
    } else if (txtalamat.getText().equals("")) {  
        JOptionPane.showMessageDialog(null, "Mohon lengkapi alamat kasir ");  
    } else if (txttelp.getText().length() > 13) {  
        JOptionPane.showMessageDialog(null, "Nomor tidak boleh lebih dari 13 angka");  
    } else if (!txtemail.getText().contains("@") || !txtemail.getText().contains(".")) {  
        JOptionPane.showMessageDialog(null, "email harus berisi @ dan .");  
    } else {  
        try {  
            String sql = "INSERT INTO kasir (nama_kasir,alamat,no_telp,email) VALUES(?, ?, ?, ?)";  
            PreparedStatement pst = kon.prepareStatement(sql);  
            pst.setString(1, txtnamakas.getText());  
            pst.setString(2, txtalamat.getText());  
            pst.setString(3, txttelp.getText());  
            pst.setString(4, txtemail.getText());  
            pst.executeUpdate();  
            JOptionPane.showMessageDialog(null, "Data Berhasil Di Simpan");  
            tampil_kasir();  
            reset();  
        } catch (Exception e) {  
            JOptionPane.showMessageDialog(this, e.getMessage());  
        }  
    }  
}
```

Fungsi ini digunakan untuk menyimpan data yang sudah di inputkan dari form ke dalam database, ini contoh simpan pada data kasir dan fungsi ini juga terdapat di kelola data yang lain yang membedakan hanya jumlah field yang di inputkan saja.


```

private void btnubahActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    try{
        if(tabelkasir.getSelectedRowCount()==0){
            JOptionPane.showMessageDialog(null,"Silahkan pilih baris data yang akan di ubah");
            return ;
        }else{
            String sql="UPDATE kasir SET nama_kasir='"+txtnamakas.getText()+"',alamat='"+txtalamat.getText()+"'";
            PreparedStatement pst=kon.prepareStatement(sql);
            pst.executeUpdate();
            JOptionPane.showMessageDialog(null, "Data Berhasil Di Ubah");
        }
    }catch (Exception e) {
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
    tampil_kasir();
    reset();
}

```

Ini adalah fungsi untuk mengedit data jika kita ada salah input dan biasanya edit data mengacu pada satu field untuk dapat menjadi acuan data mana yang akan di ubah ,misal update kasir set nama,alamat,email where id_kasir nah si id kasir ini di jadikan acuan bahawa nama,alamat dan email dari id kasir mana yang akan di ubah,dan sebelum mengubah data biasanya kita akan memilih baris man yang akan di ubah dengan event mouseclicked pad atabel

Mouse Clicked

```

private void tabelkasirMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    int row = tabelkasir.getSelectedRow();
    txtIdkas.setText(tabelkasir.getModel().getValueAt(row, 0).toString());
    txtnamakas.setText(tabelkasir.getModel().getValueAt(row, 1).toString());
    txtalamat.setText(tabelkasir.getModel().getValueAt(row, 2).toString());
    txttelp.setText(tabelkasir.getModel().getValueAt(row, 3).toString());
    txtemail.setText(tabelkasir.getModel().getValueAt(row, 4).toString());
}

```

Fungsi ini di gunakan untuk menampilkan data pada form field setelah baris data yang akan di ubah di klik dan anda tinggal mengubah field mana yan gsalah inputan data.berikut tampilan setelah baris di klik untuk di ubah

Kelola Data Kasir

Form Kasir

Id *autiincrement

Nama Kasir

Alamat

No.telp

Email

Cari

Id Kasir	Nama Kasir	Alamat	No.Telp	Email
1	ade	jl.ibrahim aji no....	0897451892	ade10@gmail.c...
2	sandi	jl.antapani no.29	08812055409	sandi@gmail.com

Reset Simpan Ubah Hapus

Source Code hapus

```
private void btnhapusActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    if(tabelkasir.getSelectedRowCount()==0){  
        JOptionPane.showMessageDialog(null,"Silahkan pilih baris data yang akan di ubah");  
        return ;  
    }  
    if(JOptionPane.showConfirmDialog(null,"Apakah anda yakin akaan menghapus data ini?")==JOptionPane.OK_OPTION){  
        try{  
            String sql="DELETE FROM kasir WHERE id_kasir='"+txtIdkas.getText()+"'";  
            PreparedStatement pst=kon.prepareStatement(sql);  
            pst.executeUpdate();  
            JOptionPane.showMessageDialog(null,"Data berhasil di hapus");  
        }catch (Exception e) {  
            JOptionPane.showMessageDialog(this, e.getMessage());  
        }  
        tampil_kasir();  
        reset();  
    }  
}
```

Logika nya sama seperti update namun disini kita hnay mengacu pada satu kondisi dan biasanya mengacu ke id disini saya menghapus data di tabel kasir dimana memiliki id kasir sama dengan berapa,dan disini juga berhubungan dengan fungsi mouse diklik jadi anda harus memilih baris data man yang akan di hapus dan jika anda belum emilih baris dan menekan tombol hapus maka aka nada pemberitahuan silahkan seleksi baris yang akan di hapus

Souce code menampilkan data dari database ke dalam tabel

```
private void tampil_kasir(){  
    // membuat tampilan model tabel  
    DefaultTableModel model = new DefaultTableModel();  
    model.addColumn("Id Kasir");  
    model.addColumn("Nama Kasir");  
    model.addColumn("Alamat");  
    model.addColumn("No.Telp");  
    model.addColumn("Email");  
  
    //menampilkan data database kedalam tabel  
    try {  
        String sql = "select * from kasir";  
        java.sql.Statement stm=kon.createStatement();  
        java.sql.ResultSet res=stm.executeQuery(sql);  
        while(res.next()){  
            model.addRow(new Object[] {res.getString(1),res.getString(2),res.getString(3),res.getString(4),res.getString(5)});  
        }  
        tabelkasir.setModel(model);  
    } catch (Exception e) {  
    }  
}  
  
private void reset(){
```

Source tersebut berfungsi untuk menampilkan data dari database ke dalam tabel,dan model.addcoloum itu indik untuk kolomny atau header dan jika jumlah nya berbeda dengan new object maka akan ada kolom yang tidak sejajar dengan data barisnya.

Reset

```
private void reset() {  
    txtIdkas.setText("");  
    txtnamakas.setText("");  
    txtalamat.setText("");  
    txttelp.setText("");  
    txtemail.setText("");  
}
```

Fungsinya untuk menghapus semua data yang sudah di ketik ke dalam text field ketika kita sedang mengunput tetapi membatalkan untuk menginput

Source Code Cari/filter data

```
private void cari() {  
    //menampilkan data database kedalam tabel  
    DefaultTableModel model = new DefaultTableModel();  
    model.addColumn("Id Kasir");  
    model.addColumn("Nama Kasir");  
    model.addColumn("Alamat");  
    model.addColumn("No.Telp");  
    model.addColumn("Email");  
    try {  
        String cari = txtcarikasisir.getText();  
        String sql = "select * from kasir where nama_kasir like '%" + cari + "%'";  
        java.sql.Statement stm = kon.createStatement();  
        java.sql.ResultSet res = stm.executeQuery(sql);  
        while (res.next()) {  
            model.addRow(new Object[] { res.getString(1), res.getString(2), res.getString(3), res.getString(4) });  
        }  
        tabelkasir.setModel(model);  
    } catch (Exception e) {  
        JOptionPane.showMessageDialog(null, "Data tidak ditemukan");  
    }  
}
```

Fungsinya hampir sama dengan menampilkan data namun menampilkan data dengan kondisi tertentu seperti menampilkan beberapa nama atau nama yang mengandung huruf tertentu

Menu Kategori

Menu

- Data Buku
- Penjualan
- Data Kategori
- Logout

Kelola Data Kategori

Form kategori

Id *Auto

Nama Kategori

Cari

Id Kategori	Nama Kategori
1	Novel
10	Komputer
11	Sastra
12	Filasafat

Reset Simpan Ubah Hapus

Menu Data Kategori dengan CRUD di dalamnya dan data ini akan ditampilkan di selected item di data buku.

Menu penjualan

Id bu...	No.Is...	Judul	Peng...	Pene...	Tahu...	Keat...	Stok	Harg...	Harg...
1	9874...	Pela...	Rifqi	Gran...	2019	Sastra	6	40000	45000
2	1927...	Kau...	Stuti	Garh...	2020	Novel	6	50000	55000
3	8956...	Senja	kiko	gran...	2020	Novel	0	45000	50000
4	24242	ku ke...	gatot	sam...	2012	Sastra	5	20000	25000

Id detail	Id buku	Judul	Kategori	harga satuan	jumlah beli	harga
-----------	---------	-------	----------	--------------	-------------	-------

Untuk menu penjualan anda dapat mencari data buku yang di beli pembeli, lalu masukan jumlahnya setelah itu hitung setelah button hitung tambahkan ke daftarbuku di beli lalu masukan buku kedua jika buku yang di beli lebih dari satu dan berbeda, setelah itu press button total semuanya untuk mengetahui total harga buku yang di beli, lalu masukan tunai uang pembeli kembalian akan otomatis menghitung setelah selesai transaksi data akan di simpan ke data transaksi.

Source code menampilkan data buku di penjualan

```
private void tampil_buku(){
// membuat tampilan model tabel
DefaultTableModel model = new DefaultTableModel();
model.addColumn("Id buku");
model.addColumn("No.Isbn");
model.addColumn("Judul");
model.addColumn("Pengarang");
model.addColumn("Penerbit");
model.addColumn("Tahun Terbit");
model.addColumn("Kategori buku");
model.addColumn("Stok");
model.addColumn("Harga Pokok");
model.addColumn("Harga Jual");
//menampilkan data database kedalam tabel
try {
String sql = "select * from buku";
java.sql.Statement stm=kon.createStatement();
java.sql.ResultSet res=stm.executeQuery(sql);
while(res.next()){
model.addRow(new Object[]{res.getString(1),res.getString(2),res.getString(3),res.getString(4),res
, res.getString(6),res.getString(7),res.getString(8),res.getString(9),res.getString
});
tabelbuku.setModel(model);
res.last();
int jumlahdata = res.getRow();
res.first();
} catch (Exception e) {
}
```

Dan untuk menampilkan datanya kita panggil di

```
public Form_penjualan_kasir() {  
    initComponents();  
    tampil_buku();  
    model = new DefaultTableModel();  
  
    tabel.setModel(model);  
    model.addColumn("Id detail");  
    model.addColumn("Id buku");  
    model.addColumn("Judul");  
    model.addColumn("Kategori");  
    model.addColumn("harga satuan");  
    model.addColumn("jumlah beli");  
    model.addColumn("harga");  
    loadData();  
    nofaktur();  
}
```

Dan pada saat transaksi kita tinggal memilih baris dari data buku dan datanya akan di tampilkan di field penjualan data buku yang di pilih seperti di bawah ini kiat hanya menginputkan jumlah buku yang di beli saja karena no nota otomatis.

Id bu...	No.Is...	Judul	Peng...	Pene...	Tahu...	Keat...	Stok	Harg...	Harg...
1	9874...	Pela...	Rifqi	Gran...	2019	Sastra	6	40000	45000
2	1927...	Kau ...	Stuti	Garh...	2020	Novel	6	50000	55000
3	8956...	Senja	kiko	gran...	2020	Novel	20	45000	50000
4	24242	ku ke...	gatot	sam...	2012	Sastra	5	20000	25000

Source code nota otomatis

```
private void nofaktur() {
    try {
        java.sql.Connection c = koneksi.koneksiDb();
        Statement s = c.createStatement();

        String sql = "SELECT * FROM penjualan ORDER by id_transaksi desc";
        ResultSet r = s.executeQuery(sql);

        if (r.next()) {
            String nofak = r.getString("id_transaksi").substring(1);
            String AN = "" + (Integer.parseInt(nofak) + 1);
            String Nol = "";

            if (AN.length() == 1) {
                Nol = "000";
            } else if (AN.length() == 2) {
                Nol = "00";
            } else if (AN.length() == 3) {
                Nol = "0";
            } else if (AN.length() == 4) {
                Nol = "";
            }

            faktur.setText("F" + Nol + AN);
        } else {
            faktur.setText("F0001");
        }
    }
}
```

Sourcode ini kan menampilkan id_transaksi atau nota dan setiap transaksi id nya akan bertambah satu dan saat di tampilkan di field maka akan di tampilkan no nota seperti gambar sebelumnya.

Source code button total

No nota	F0004
Id Buku	1
Judul	Pelangi di pagi hari
Kategori	Sastra
Harga	45000
Jumlah	2
hitung	Total 90000

Seperti pada gambar di samping saat kita meng input jumlah buku yang dibeli, lalu menekan tombol hitung maka akan keluar total dari harga buku dan jumlah buku yang di beli di field total. Dan jika data di semua field dan jumlah beli belum lengkap maka hitung tidak dapat di tampilkan atau di eksekusi.

```
if(faktur.getText().equals("") || idbuku.getText().equals("") || tjudul.getText().equals("")){
    JOptionPane.showMessageDialog(null, "LENGKAPI DATA !", "Senja Abadi Books Store");
}

else{
    String a = tjumlah.getText();
    int aa = Integer.parseInt(a);

    String b = jumlah.getText();
    int bb = Integer.parseInt(b);
    if(aa > bb){
        JOptionPane.showMessageDialog(null, "jumlah melebihi stok", "Senja Abadi Books Store");
        tjumlah.setText("");
    }
    else{
        if(tjumlah.getText().equals("")){
            JOptionPane.showMessageDialog(null, "ISI JUMLAH BELI !");
        }
        else{
            int jumlah, harga, total;

            jumlah = Integer.parseInt(tjumlah.getText().toString());
            harga = Integer.parseInt(hsatuan.getText().toString());
            total = jumlah * harga;

            tttotal.setText(Integer.toString(total));
        }
    }
}
```

Button Tambah ke daftar di beli

Id detail	Id buku	Judul	Kategori	harga satuan	jumlah beli	harga
1	1	Pelangi di pa...	Sastra	45000	2	90000

Ketika kita sudah menghitung total dari item lalu tambahkan ke daftar buku yang di beli dan akan muncul di tabel daftar buku yang di beli, jika buku yang di beli lebih dari satu dan beda judul buku maka input kembali lalu hitung lalu tambah ke daftar item yang di beli seperti pada gambar.

Source code button tambah/sistemnya seperti kita tambah keranjang di online shop

```
String idbukuu = idbuku.getText();
String judull = tjudul.getText();
String kategorii = tkategori.getText();
String hsatuann = hsatuan.getText();
String tjumlaah = tjumlah.getText();
String totall = tttotal.getText();

try {
    java.sql.Connection c = koneksi.koneksiDb();

    String sql = "INSERT INTO penjualan_detail VALUES (?, ?, ?, ?, ?, ?, ?)";

    PreparedStatement p = c.prepareStatement(sql);
    p.setString(1, null);
    p.setString(2, idbukuu);
    p.setString(3, judull);
    p.setString(4, kategorii);
    p.setString(5, hsatuann);
    p.setString(6, tjumlaah);
    p.setString(7, totall);

    p.executeUpdate();
    p.close();
} catch (SQLException e) {
    System.out.println("Terjadi Error");
} finally {
```

Setelah data field di penjualan terisi semua termasuk total data akan di tampung sementara di penjualan detail dengan query insert into penjualan detail dan di tampilkan di tabel daftar buku yang di beli dan ketika disimpan akan ada notifikasi data berhasil di simpan lalu di tampilkan di tabel yang di bawah yaitu daftar buku yang di beli.

Button total,bayar dan kembalian

Total semuanya	Rp. 140000	Bayar	<input type="text" value="150000"/>	Kembalian	<input type="text" value="10000"/>
----------------	------------	-------	-------------------------------------	-----------	------------------------------------

Daftar buku yang di beli						
Id detail	Id buku	Judul	Kategori	harga satuan	jumlah beli	harga
1	1	Pelangi di pa...	Sastra	45000	2	90000
2	3	Senja	Novel	50000	1	50000

Setelah semua buku yang di beli sudah di input semua kita klik button total dan otomatis total keseluruhan dari buku yang di beli akan muncul lalu masukan jumlah uang yang di bayarkan pembeli dan jumlah kembalian akan otomatis muncul.

Source code total

```
try {
    java.sql.Connection c = koneksi.koneksiDb();
    Statement s = c.createStatement();

    String sql = "SELECT SUM(`harga`) AS total FROM penjualan_detail";
    ResultSet r = s.executeQuery(sql);

    while (r.next()) {
        alltotal.setText(r.getString(""+total));
    }
    r.close();
    s.close();
} catch (SQLException e) {
    System.out.println("Terjadi Error");
}
```

Source code menghitung keseluruhan total buku yang di beli,disini say amenggunnakan fungsi sum di query untuk menghitung harga diman harga nama field di database tabel penjualan detai dan menamping nya di as total

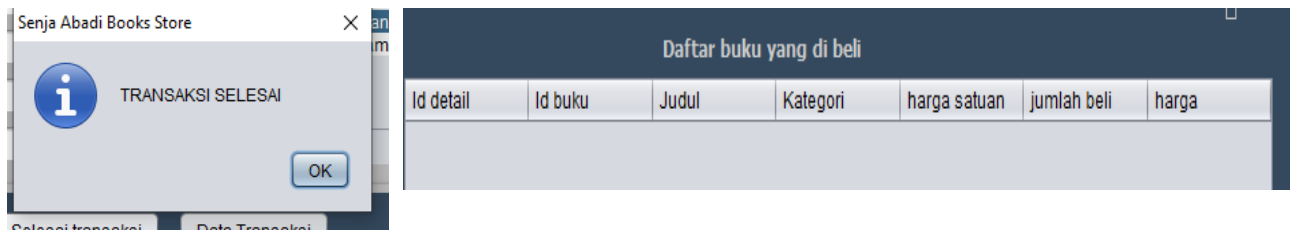
Bayar dan total

```
// TODO add your handling code here:
bayar = Integer.parseInt(String.valueOf(txbbayar.getText()));
total = Integer.parseInt(String.valueOf(alltotal.getText()));
kembali = bayar - total;

txkembalian.setText(Long.toString(kembali));
```

Disini ketika kita meinputkan jumlah bayar maka kembalian akan otomatis muncul di mana kembalian mengambil bauay – total keseluruhan buku yang di beli.

Button selesai transaksi



Setelah transaksi selesai jika tidak ada buku yang di akan di inpu ke daftar beli maka klik selsesai transaksi jika berhasill maka akan muncul pop up transaksi selesai dan data di daftar buku yang di beli akan menjadi kosong kembali.

Source code selesai transaksi

```
if(txkbayar.getText().equals("") || txkembalian.getText().equals("")){
    JOptionPane.showMessageDialog(null, "LENGKAPI DATA !", "Senja Abadi Books Store", JOptionPane.INFORMATION_MESSAGE);
} else {
    String a = txkembalian.getText();
    int ab = Integer.parseInt(String.valueOf(txkembalian.getText()));
    if(ab < 0){
        JOptionPane.showMessageDialog(null, "Uang anda kurang", "Senja Abadi Books Store", JOptionPane.ERROR_MESSAGE);
        txkbayar.setText("");
        txkembalian.setText("");
    } else {
        try {
            Statement s = kon.createStatement();

            String sql = "SELECT * FROM penjualan_detail";
            ResultSet r = s.executeQuery(sql);

            while (r.next()) {
                long millis = System.currentTimeMillis();
                java.sql.Date date = new java.sql.Date(millis);
                System.out.println(date);
                String tgl = date.toString();
                String sqli = "INSERT INTO penjualan VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

                PreparedStatement p = kon.prepareStatement(sqli);
                p.setString(1, faktur.getText());
                p.setString(2, r.getString("id_buku"));
```

Setelah itu maka kan masuk ke source code berikut semua data akan di masukan ke tabel penjualan di database dengan query insert into penjualan.

Menghapus semua data penjualan detail

```
try {
    String sqli ="TRUNCATE `penjualan_detail`;
    java.sql.Connection conn=(java.sql.Connection)koneksi.koneksiDb();
    java.sql.PreparedStatement pst=conn.prepareStatement(sqli);
    pst.execute();
    JOptionPane.showMessageDialog(null, "TRANSAKSI SELESAI", "Senja Abadi Books Store",
    loadData();
    tx2.setText(faktur.getText());
    txbayar.setText("");
    txkembalian.setText("");
    alltotal.setText("");
    nofaktur();
}
```

Fungsi truncate hampir sama dengan query delete namun lebih simple dan tidak di isi kondisi disini data di penjualan detail akan di hapus semua.dan load data untuk menampilkan data penjualan detail karena kita truncate maka ketika kita kllik oke pada pop up transaksi selseai maka yang di tampilkan loaddata tabel kosong.

Button data transaksi di halaman penjualan

```
// TODO add your handling code here:
Datatransaksi.setLocationRelativeTo(null);
tabeltransaksi();
Datatransaksi.setVisible(true);
```

Source cod untuk menampilkan pop up data transaksi dimana tabeltransaksi adalah fungsi untuk menampilkan data transaksi dan Datatransaksi.setVisible(true);menampilkan jdialog Datatransaksi di other komponen di frame form penjualan kasir

Tampilan pop up dat transaksi

The screenshot shows a window titled "TABEL TRANSAKSI" with a search bar labeled "Cari Id Transaksi" and a "Cetak Nota" button. Below is a table of transactions:

Id Trans...	Id Buku	Tanggal	Judul	Kategori	Harga St...	Jumlah ...	Sub Total	Bayar	Kembalian
F0003	2	2020-07-...	Kau yang...	Novel	55000	1	55000	100000	20000
F0003	4	2020-07-...	ku kejar	Sastra	25000	1	25000	100000	20000
F0004	1	2020-07-...	Pelangi ...	Sastra	45000	2	90000	150000	10000
F0004	3	2020-07-...	Senja	Novel	50000	1	50000	150000	10000
F0001	1	2020-07-...	Kau yang...	Sastra	45000	2	90000	100000	10000
F0002	2	2020-07-...	Kau yang...	Novel	55000	1	55000	110000	5000
F0002	4	2020-07-...	ku kejar	Sastra	25000	2	50000	110000	5000

Ketika kita akan mencetak nota maka kita inpukan di cari transaksi lalu cetak nota,karena disini say amemakai order by tgl desc jadi transaksi dengan tanggal terbaru akan tampil di atas.

Data transaksi di menu kasir

The screenshot shows the "Kelola Data Penjualan" menu with a "Form Penjualan" section. A "TABEL TRANSAKSI" popup is open, displaying a list of transactions. The background menu includes fields for "No nota" (F0004), "Id Buku", "Cari Buku", and buttons for "hitung", "Tambah k...", and "Total sem...". At the bottom, there is a "Daftar buku yang di beli" table.

Id Trans...	Id Buku	Tanggal	Judul	Kategori	Harga St...	Jumlah ...	Sub Total	Bayar	Kembalian
F0001	1	2020-07-...	Pelangi ...	Sastra	45000	2	90000	100000	10000
F0002	2	2020-07-...	Kau yang...	Novel	55000	1	55000	110000	5000
F0002	4	2020-07-...	ku kejar	Sastra	25000	2	50000	110000	5000
F0003	2	2020-07-...	Kau yang...	Novel	55000	1	55000	100000	20000
F0003	4	2020-07-...	ku kejar	Sastra	25000	1	25000	100000	20000

Saat anda ingin mencetak nota anda press data transaksi di penjualan maka akan muncul popup data transaksi dan cari no nota yang akan di cetak, lalu cetak nota

View Cetak nota

JasperViewer

Senja Abadi Books Store

Jl.Ibrahim Adji No.28 Kota Bandung
No.Telp 08816764

No.Nota : F0002
Tanggal_transaksi : 7/4/20 12:00 AM

judul	kategori	harga satuan	jumlah_beli	harga
Kau yang Indah	Novel	55000	1	55000
ku kejar	Sastra	25000	2	50000

Total	105000
bayar	110000
kembalian	5000

Halaman Pemilik

Selamat Datang Di Halaman Pemilik

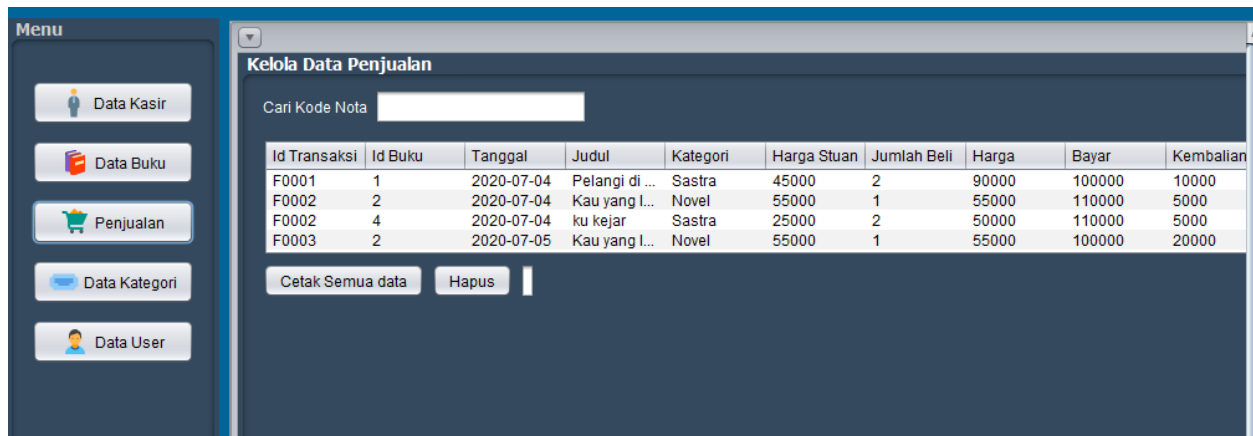
Menu

- Data Kasir
- Data Buku
- Penjualan
- Data Kategori
- Data User
- Logout

Di menu Pemilik hampir sama dengan menu kasir namun ada tambahan pengelolaan data kasir dan data user dimana pemilik dapat mengatur siapa saja yang bias masuk aplikasi sebagai kasir atau pegawainya. dan di setiap menu memiliki fungsi CRUD kecuali data penjualan.

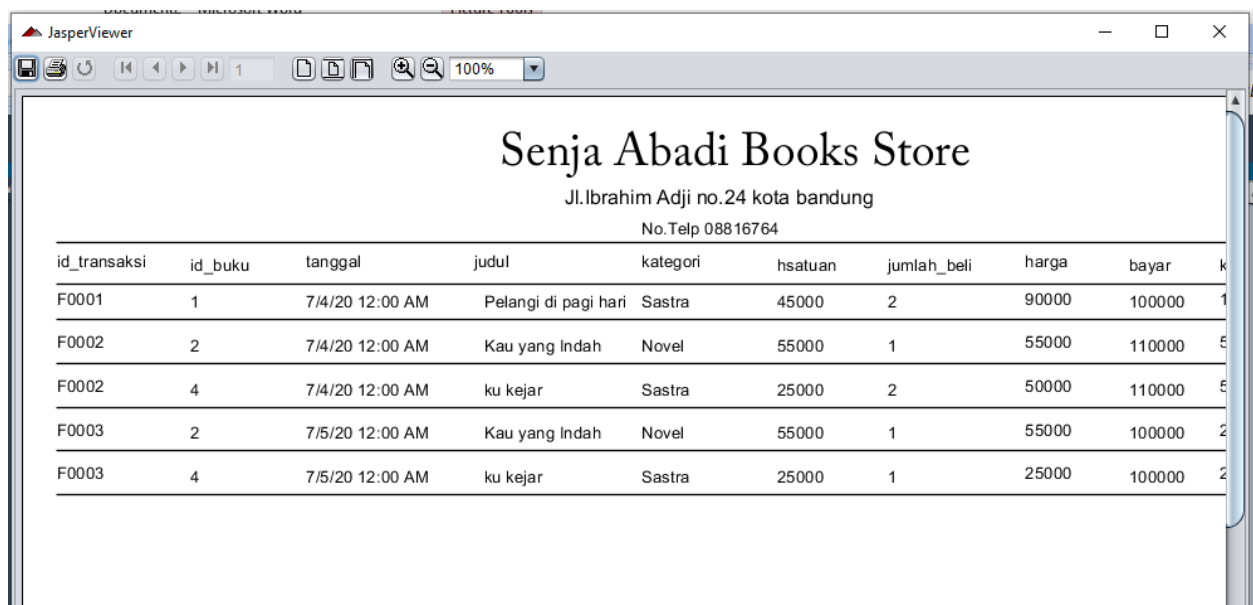
Copy©MuhamadRifqiRahmatulloh

Menu penjualan di halaman pemilik



Disini pemilik dapat melihat data transaksi dan mencetak seluruh dat transaksi untuk di rekap dan disini pemilik dapat menghapus data transaksi ketika seluruh data sudah di rekap, kenapa di penjualan kasir tidak ada button hapus karena takutnya data dapat di manipulasi kasir.

Cetak seluruh data transaksi



Source Code untuk menampilkan cetak nota

```
private void btncetakActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    java.sql.Connection con=null;  
    try {  
        String jdbcDriver ="com.mysql.jdbc.Driver";  
        Class.forName("com.mysql.jdbc.Driver");  
        java.sql.Connection koneksi = DriverManager.getConnection("jdbc:mysql://localhost/dbtoko","root","");  
        Statement stm = koneksi.createStatement();  
  
        try{  
            String report=("C:\\Users\\Rifqi\\Documents\\NetBeansProjects\\toko\\src\\report\\nota.jrxml");  
            HashMap hash = new HashMap();  
            hash.put("nota",txnota.getText());  
            JasperReport JRpt = JasperCompileManager.compileReport(report);  
            JasperPrint JPrint = JasperFillManager.fillReport(JRpt, hash, kon);  
            JasperViewer.viewReport(JPrint, false);  
            Datatransaksi.dispose();  
        }catch (Exception rptexcept){  
            System.out.println("maaf cetak gagal"+rptexcept);  
        }  
    } catch (Exception e) {  
        System.out.println(e);  
    }  
}
```

Fungsi source code ini untuk memanggil data file jrxml atau report dari cetak nota agar muncul atau terhubung dengan program. pertama kita hubungkan dulu dengan koneksi ke database setelah itu masuk bagian report kit memanggil file jrxml dan hashmap berfungsi mengambil parameter atau data apa yang mau di cetak .