

1. FormHome.java

```
package id.ac.pos.gudang.Form;

import id.ac.pos.gudang.Dialog.Admin.DialogRecycleBin;
import id.ac.pos.gudang.Dialog.DialogBackup;
import id.ac.pos.gudang.Dialog.DialogLaporan;
import id.ac.pos.gudang.Dialog.DialogUbahProduk;
import id.ac.pos.gudang.Panel.PanelPemesanan;
import id.ac.pos.gudang.Panel.PanelPenerimaan;
import id.ac.pos.gudang.Panel.PanelPengembalian;
import id.ac.pos.gudang.Panel.PanelPengiriman;
import id.ac.pos.gudang.dao.ProdukDAO;
import id.ac.pos.gudang.daoimpl.ProdukDAOImpl;
import id.ac.pos.gudang.entity.Produk;
import id.ac.pos.gudang.tablemodel.ProdukTM;
import java.awt.Dialog;
import java.awt.CardLayout;
import javax.swing.JFrame;
import javax.swing.*;
import java.awt.event.KeyEvent;
import java.io.IOException;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.table.TableRowSorter;

/**
 *
```

```

* @author Oyoy
*/

public final class FormHome extends javax.swing.JFrame {

    /**
     * Creates new form FormHome
     */
    private Produk produk;
    private ProdukDAO dao;
    ArrayList<Produk> arrayProduk;
    TableRowSorter sorter;

    public FormHome() {

    }

    FormHome(String nama, String nik) {
        initComponents();
        this.setExtendedState(JFrame.MAXIMIZED_BOTH);
        refresh();
        int tahun;
        tahun = dao.getTahunSekarang();
        fieldTahunPrangko.setText(Integer.toString(tahun));
        fieldTahunProdukDokumenFilateli.setText(Integer.toString(tahun));
        fieldTahunProdukKemasan.setText(Integer.toString(tahun));
        fieldTahunProdukMSSS.setText(Integer.toString(tahun));
        fieldTahunProdukMerchandise.setText(Integer.toString(tahun));
        fieldTahunProdukPrisma.setText(Integer.toString(tahun));
        fieldTahunProdukSHPSHPSS.setText(Integer.toString(tahun));
    }

```

```
SelamatDatang.setText("SELAMAT DATANG, " + nama.toUpperCase());  
this.nik.setText(nik);  
}
```

```
private void refresh() {  
    autoincrementProduk();  
    getDataProduk();  
}
```

```
public void autoincrementProduk() {  
    autoincrementPrangko();  
    autoincrementMS_SS();  
    autoincrementSHP_SHPSS();  
    autoincrementKemasan();  
    autoincrementMerchandise();  
    autoincrementPrisma();  
    autoincrementDokumenFilateli();  
}
```

```
private void autoincrementPrangko() {  
    String kosong = null;  
    dao = new ProdukDAOImpl();  
    String jenisProduk = "PR";  
  
    String kode_prangko = dao.getIdProduk(jenisProduk);  
    if (kode_prangko == null) {  
        kode_prangko = "PR000000000";  
    }  
}
```

```
String sub_nomor_string = kode_prangko.substring(2, 11);
int sub_nomor_int = Integer.parseInt(sub_nomor_string);
sub_nomor_string = String.valueOf(sub_nomor_int);
int panjang = sub_nomor_string.length();
switch (panjang) {
    case 1:
        kosong = "000000000";
        break;
    case 2:
        kosong = "00000000";
        break;
    case 3:
        kosong = "0000000";
        break;
    case 4:
        kosong = "000000";
        break;
    case 5:
        kosong = "00000";
        break;
    case 6:
        kosong = "0000";
        break;
    case 7:
        kosong = "000";
        break;
    case 8:
        kosong = "00";
        break;
    case 9:
        kosong = "0";
        break;
}
```

```

        break;
    case 9:
        kosong = null;
        break;
    default:
        break;
    }

    sub_nomor_int = sub_nomor_int + 1;
    sub_nomor_string = String.valueOf(sub_nomor_int);
    kode_prangko = "PR" + kosong + sub_nomor_string;
    fieldKodeProdukPrangko.setText(kode_prangko);
}

private void autoincrementMS_SS() {
    Object jenisMS_SS = ComboJenisMS_SS.getSelectedItem();
    String kosong = null;
    String jenisProduk = null;

    dao = new ProdukDAOImpl();
    if (jenisMS_SS == "MS") {
        jenisProduk = "MS";
        //Tambahkan pilihan item untuk buah

    } else if (jenisMS_SS == "SS") {
        jenisProduk = "SS";
    }

    String kodeMS_SS = dao.getIdProduk(jenisProduk);

```

```
if (kodeMS_SS == null) {  
    kodeMS_SS = "" + jenisProduk + "000000000";  
}  
  
String sub_nomor_string = kodeMS_SS.substring(2, 11);  
int sub_nomor_int = Integer.parseInt(sub_nomor_string);  
sub_nomor_string = String.valueOf(sub_nomor_int);  
int panjang = sub_nomor_string.length();  
switch (panjang) {  
    case 1:  
        kosong = "000000000";  
        break;  
    case 2:  
        kosong = "00000000";  
        break;  
    case 3:  
        kosong = "0000000";  
        break;  
    case 4:  
        kosong = "000000";  
        break;  
    case 5:  
        kosong = "00000";  
        break;  
    case 6:  
        kosong = "0000";  
        break;  
    case 7:  
        kosong = "000";  
        break;  
    case 8:  
        kosong = "00000";  
        break;  
    case 9:  
        kosong = "000000";  
        break;  
    case 10:  
        kosong = "0000000";  
        break;  
    case 11:  
        kosong = "00000000";  
        break;  
    case 12:  
        kosong = "000000000";  
        break;  
}
```

```

        break;
    case 8:
        kosong = "0";
        break;
    case 9:
        kosong = null;
        break;
    default:
        break;
}

sub_nomor_int = sub_nomor_int + 1;
sub_nomor_string = String.valueOf(sub_nomor_int);
kodeMS_SS = jenisProduk + kosong + sub_nomor_string;
fieldKodeProdukMS_SS.setText(kodeMS_SS);
}

```

```

private void autoincrementKemasan() {
    String kosong = null;
    dao = new ProdukDAOImpl();
    String jenisProduk = "KM";

    String kode_prangko = dao.getIdProduk(jenisProduk);
    if (kode_prangko == null) {
        kode_prangko = "KM0000000000";
    }

    String sub_nomor_string = kode_prangko.substring(2, 11);
    int sub_nomor_int = Integer.parseInt(sub_nomor_string);
    sub_nomor_string = String.valueOf(sub_nomor_int);
}

```

```
int panjang = sub_nomor_string.length();
switch (panjang) {
    case 1:
        kosong = "00000000";
        break;
    case 2:
        kosong = "0000000";
        break;
    case 3:
        kosong = "000000";
        break;
    case 4:
        kosong = "00000";
        break;
    case 5:
        kosong = "0000";
        break;
    case 6:
        kosong = "000";
        break;
    case 7:
        kosong = "00";
        break;
    case 8:
        kosong = "0";
        break;
    case 9:
        kosong = null;
```



```

        break;
    default:
        break;
    }
    sub_nomor_int = sub_nomor_int + 1;
    sub_nomor_string = String.valueOf(sub_nomor_int);
    kode_prangko = "KM" + kosong + sub_nomor_string;
    fieldKodeProdukKemasan.setText(kode_prangko);
}

private void autoincrementMerchandise() {
    String kosong = null;
    dao = new ProdukDAOImpl();
    String jenisProduk = "MC";

    String kode_merchandise = dao.getIdProduk(jenisProduk);
    if (kode_merchandise == null) {
        kode_merchandise = "MC000000000";
    }
    String sub_nomor_string = kode_merchandise.substring(2, 11);
    int sub_nomor_int = Integer.parseInt(sub_nomor_string);
    sub_nomor_string = String.valueOf(sub_nomor_int);
    int panjang = sub_nomor_string.length();
    switch (panjang) {
        case 1:
            kosong = "000000000";
            break;
        case 2:

```

```

        kosong = "0000000";
        break;
    case 3:
        kosong = "000000";
        break;
    case 4:
        kosong = "00000";
        break;
    case 5:
        kosong = "0000";
        break;
    case 6:
        kosong = "000";
        break;
    case 7:
        kosong = "00";
        break;
    case 8:
        kosong = "0";
        break;
    case 9:
        kosong = null;
        break;
    default:
        break;
}

sub_nomor_int = sub_nomor_int + 1;
sub_nomor_string = String.valueOf(sub_nomor_int);

```

```
    kode_merchandise = "MC" + kosong + sub_nomor_string;  
    fieldKodeProdukMerchandise.setText(kode_merchandise);  
}
```

```
private void autoincrementPrisma() {  
    String kosong = null;  
    dao = new ProdukDAOImpl();  
    String jenisProduk = "PS";  
  
    String kode_prisma = dao.getIdProduk(jenisProduk);  
    if (kode_prisma == null) {  
        kode_prisma = "PS0000000000";  
    }  
    String sub_nomor_string = kode_prisma.substring(2, 11);  
    int sub_nomor_int = Integer.parseInt(sub_nomor_string);  
    sub_nomor_string = String.valueOf(sub_nomor_int);  
    int panjang = sub_nomor_string.length();  
    switch (panjang) {  
        case 1:  
            kosong = "000000000";  
            break;  
        case 2:  
            kosong = "00000000";  
            break;  
        case 3:  
            kosong = "0000000";  
            break;  
        case 4:
```

```

        kosong = "00000";
        break;
    case 5:
        kosong = "0000";
        break;
    case 6:
        kosong = "000";
        break;
    case 7:
        kosong = "00";
        break;
    case 8:
        kosong = "0";
        break;
    case 9:
        kosong = null;
        break;
    default:
        break;
}

sub_nomor_int = sub_nomor_int + 1;
sub_nomor_string = String.valueOf(sub_nomor_int);
kode_prisma = "PS" + kosong + sub_nomor_string;
fieldKodeProdukPrisma.setText(kode_prisma);
}

public String cariData(String tabCari) {
    Object pilihanCari = "";

```

```

if (tabCari.compareTo("PR") == 0) {
    pilihanCari = comboCariPrangko.getSelectedItem();
} else if (tabCari.compareTo("kemasan") == 0) {
    pilihanCari = comboJenisKemasan.getSelectedItem();
} else if (tabCari.compareTo("shp") == 0) {
    pilihanCari = comboSHP_SHPSS.getSelectedItem();
} else if (tabCari.compareTo("ss") == 0) {
    pilihanCari = comboMS_SS.getSelectedItem();
} else if (tabCari.compareTo("merchandise") == 0) {
    pilihanCari = comboMerchandise.getSelectedItem();
} else if (tabCari.compareTo("prisma") == 0) {
    pilihanCari = comboPrisma.getSelectedItem();
} else if (tabCari.compareTo("df") == 0) {
    pilihanCari = comboDokumenFilateli.getSelectedItem();
}

```

```
String jenisCari = null;
```

```

if (pilihanCari == "Id Produk") {
    jenisCari = "id_produk";
} else if (pilihanCari == "Nama Produk") {
    jenisCari = "nama_produk";
} else if (pilihanCari == "Nominal") {
    jenisCari = "nominal";
} else if (pilihanCari == "Stok") {
    jenisCari = "stok";
} else if (pilihanCari == "Biaya Cetak") {
    jenisCari = "biaya_cetak";
}

```

```

    } else if (pilihanCari == "Tahun") {
        jenisCari = "tahun";
    }

    return jenisCari;

}

public String cariPilihan(String tab) {
    String pilihanCari = "";

    if (tab.compareTo("prangko") == 0) {
        pilihanCari = "PR";
    } else if (tab.compareTo("kemasan") == 0) {
        pilihanCari = "KM";
    } else if (tab.compareTo("shp") == 0) {
        pilihanCari = "SHP";
    } else if (tab.compareTo("ss") == 0) {
        pilihanCari = "SS";
    } else if (tab.compareTo("merchandise") == 0) {
        pilihanCari = "MC";
    } else if (tab.compareTo("prisma") == 0) {
        pilihanCari = "PS";
    } else if (tab.compareTo("df") == 0) {
        pilihanCari = "DF";
    }

    return pilihanCari;
}

```

```
}
```

```
public String cariJenis(String tab) {  
    String idJenis = "";  
  
    if (tab.compareTo("prangko") == 0) {  
        idJenis = "PR";  
    } else if (tab.compareTo("kemasan") == 0) {  
        idJenis = "KM";  
    } else if (tab.compareTo("shp") == 0) {  
        idJenis = "SHP";  
    } else if (tab.compareTo("ss") == 0) {  
        idJenis = "SS";  
    } else if (tab.compareTo("merchandise") == 0) {  
        idJenis = "MC";  
    } else if (tab.compareTo("prisma") == 0) {  
        idJenis = "PS";  
    } else if (tab.compareTo("df") == 0) {  
        idJenis = "DF";  
    }  
  
    return idJenis;  
}
```

```
private void autoincrementDokumenFilateli() {  
    String kosong = null;  
    dao = new ProdukDAOImpl();  
    String jenisProduk = "DF";
```

```
String kode_dokumen_filateli = dao.getIdProduk(jenisProduk);
if (kode_dokumen_filateli == null) {
    kode_dokumen_filateli = "DF000000000";
}

String sub_nomor_string = kode_dokumen_filateli.substring(2, 11);
int sub_nomor_int = Integer.parseInt(sub_nomor_string);
sub_nomor_string = String.valueOf(sub_nomor_int);
int panjang = sub_nomor_string.length();

switch (panjang) {
    case 1:
        kosong = "000000000";
        break;
    case 2:
        kosong = "00000000";
        break;
    case 3:
        kosong = "0000000";
        break;
    case 4:
        kosong = "000000";
        break;
    case 5:
        kosong = "00000";
        break;
    case 6:
        kosong = "0000";
        break;
}
```



```

        case 7:
            kosong = "00";
            break;
        case 8:
            kosong = "0";
            break;
        case 9:
            kosong = null;
            break;
        default:
            break;
    }

    sub_nomor_int = sub_nomor_int + 1;
    sub_nomor_string = String.valueOf(sub_nomor_int);
    kode_dokumen_filateli = "DF" + kosong + sub_nomor_string;
    fieldKodeProdukDokumenFilateli.setText(kode_dokumen_filateli);
}

private void resetField() {

    fieldNamaProdukPrangko.setText("");
    fieldNamaProdukMSSS.setText("");
    fieldNamaProdukSHPSHPSS.setText("");
    fieldNamaProdukMerchandise.setText("");
    fieldNamaProdukKemasan.setText("");
    fieldNamaProdukDokumenFilateli.setText("");
    fieldNamaProdukPrisma.setText("");
}

```

```

        fieldNominalPrangko.setText("");
        fieldNominalProdukMSSS.setText("");
        fieldNominalProdukSHPSHPSS.setText("");
        fieldNominalProdukMerchandise.setText("");
        fieldNominalProdukKemasan.setText("");
        fieldNominalProdukDokumenFilateli.setText("");
        fieldNominalProdukPrisma.setText("");

        fieldBiayaCetakPrangko.setText("");
        fieldBiayaCetakProdukMSSS.setText("");
        fieldBiayaCetakSHPSHPSS.setText("");
        fieldBiayaCetakProdukMerchandise.setText("");
        fieldBiayaCetakProdukKemasan.setText("");
        fieldBiayaCetakDokumenFilateli.setText("");
        fieldBiayaCetakProdukPrisma.setText("");

        int tahun;

        tahun = dao.getTahunSekarang();
        fieldTahunPrangko.setText(Integer.toString(tahun));
        fieldTahunProdukDokumenFilateli.setText(Integer.toString(tahun));
        fieldTahunProdukKemasan.setText(Integer.toString(tahun));
        fieldTahunProdukMSSS.setText(Integer.toString(tahun));
        fieldTahunProdukMerchandise.setText(Integer.toString(tahun));
        fieldTahunProdukPrisma.setText(Integer.toString(tahun));
        fieldTahunProdukSHPSHPSS.setText(Integer.toString(tahun));
    }

    private void autoincrementSHP_SHPSS() {

```

```

Object jenisSHP_SHPSS = ComboJenisSHP_SHPSS.getSelectedItem();
String kosong = null;
String jenisProduk = null;
String sub_nomor_string = null;

dao = new ProdukDAOImpl();
if (jenisSHP_SHPSS == "SHP") {
    jenisProduk = "SHP";
    //Tambahkan pilihan item untuk buah

} else if (jenisSHP_SHPSS == "SHPSS") {
    jenisProduk = "SHPSS";
}

String kodeSHP_SHPSS = dao.getIdProduk(jenisProduk);
if (kodeSHP_SHPSS == null) {
    if (jenisSHP_SHPSS == "SHP") {
        kodeSHP_SHPSS = "" + jenisProduk + "00000000";
        //Tambahkan pilihan item untuk buah

    } else if (jenisSHP_SHPSS == "SHPSS") {
        kodeSHP_SHPSS = "" + jenisProduk + "000000";
    }

}

if (jenisSHP_SHPSS == "SHP") {
    sub_nomor_string = kodeSHP_SHPSS.substring(3, 11);
}

```

