

Nama Kelompok :

Sundari 201410370311022

Kopsahwati Khotima 201410370311025

Basri Basren 201410370311030

IdentitasRuangKelas
Private String NamaRuang; Private Int LokasiRuang; Private Int ProgramStudi;
String getNamaRuang() Void setNamaRuang(String NamaRuang) int getLocationRuang() Void setLocationRuang(int LokasiRuang) Int getProgramStudi() Void setProgramStudi(int ProgramStudi)

PrepCode :

Deklarasi **String** untuk menampung nama dari ruangan diberi nama variabel **NamaRuang**

Deklarasi **Int** untuk menampung pilihan lokasi dari ruangan diberi nama variabel

LokasiRuang

Deklarasi **int** untuk menampung pilihan program studi diberi nama variabel **ProgramStudi**

Deklarasi Method **getNamaRuang()** tipe data **String** untuk mengembalikan nilai **NamaRuang**

Deklarasi Method **setNamaRuang(String NamaRuang)** tipe data void untuk menampung nilai **NamaRuang**.

Deklarasi Method **getLocationRuang()** tipe data **Int** untuk mengembalikan nilai **LokasiRuang**

Deklarasi Method **setLokasiRuang(int LokasiRuang)** tipe data void untuk menampung nilai **LokasiRuang**.

Deklarasi Method **getProgramStudi ()** tipe data **int** untuk mengembalikan nilai **ProgramStudi**

Deklarasi Method **setProgramStudi (int ProgramStudi)** tipe data void untuk menampung nilai **ProgramStudi**.

IdentitasRuangKelasTest

```
import java.io.FileWriter;
public class IdentitasRuangkelastest extends IndukClass{
    IdentitasRuangkelas IDKelas = new IdentitasRuangkelas();
    public void Input(String Namaruang,String Lokasiruang, String Programstudi){

    @Override
        public double Sesuai() {
            return 0;
        }

        @Override
        public double Analisis() {
            return 0;
        }

        @Override
        public void Save() {
            try{
                FileWriter Writer = new FileWriter("Identitas.txt");
                Writer.write("Nama Ruangan = "+Namaruang);
                Writer.write("Lokasi Ruangan = "+Lokasiruang);
                Writer.write("Program Studi = "+Programstudi);
                Writer.close();
            }catch (Exception ex){
                ex.printStackTrace();
            }
        }
    }
```

import java.io.FileWriter; berfungsi untuk menampung writer
public class IdentitasRuangkelastest extends IndukClass{ berikut adalah class IdentitasRuangkelastest yang mewariskan atau mengambil method - method yang ada di IndukClass ke kelas IdentitasRuangkelastest
IdentitasRuangkelas IDKelas = new IdentitasRuangkelas(); berikut adalah membuat object IDKelas dari class IdentitasRuangkelastest

public void Input(String Namaruang,String Lokasiruang, String Programstudi){ berikut adalah method inputan yang mana mengimputkan Namaruang, Lokasiruang dan Programstudi

@Override

```
public double Sesuai() {  
    return 0
```

berikut mengOverride daSesuai dan nilai kembaliannya adalah 0

@Override

```
public double Analisis() {  
    return 0;
```

berikut mengOverride method Analisis dan nilai kembaliannya adalah 0

@Override

```
public void Save() {  
    try{  
        FileWriter Writer = new FileWriter("Identitas.txt");  
        Writer.write("Nama Ruangan = "+Namaruang);  
        Writer.write("Lokasi Ruangan = "+Lokasiruang);  
        Writer.write("Program Studi = "+Programstudi);  
        Writer.close();  
    }catch (Exception ex){  
        ex.printStackTrace();
```

Berikut mengOverride method save lalu try catch yang mana hasil tulisan atau inputan user itu dapat di tampung atau di save di method save tersebut

KodisiRuang

```
Private int Panjang;  
Private int Lebar;  
Private Int JumlahKursi;  
Private Int JumlaPintu;  
Private Int JumlahJendela;
```

```
int getPanjang()  
Void setPanjang(int Panjang)
```

```
int getLebar()  
Void setLebar(int Lebar)
```

```
Int getJumlahKursi ()  
Void setJumlahKursi (int JumlahKursi)
```

```
Int getJumlaPintu ()  
Void setJumlaPintu (int JumlaPintu)
```

```
Int get JumlahJendela ()  
Void set JumlahJendela (int JumlahJendela)
```

PrepCode :

Deklarasi **Private Int** untuk menampung panjang ruangan dari ruangan diberi nama variable **Panjang**

Deklarasi **Private int** untuk menampung lebar dari ruangan diberi nama variable **Lebar**

Deklarasi **Private Int** untuk menampung jumlah dari kursi diberi nama variable **JumlahKursi**

Deklarasi **Private int** untuk menampung jumlah dari pintu diberi nama variable **JumlahPintu**

Deklarasi **Private Int** untuk menampung jumlah dari jendela diberi nama variable **JumlahJendela**

Deklarasi method **getPanjang()** tipedata **int** untuk mengembalikan nilai **Panjang**

Deklarasi method **setPanjang(Int Panjang)** tipe data void untuk menampung nilai **Panjang**

Deklarasi method **getLebar()** tipe data **Int** untuk mengembalikan nilai **Lebar**.

Deklarasi method **setLebar (int Lebar)** tipe data void untuk menampung nilai **Lebar**.

Deklarasi method **getJumlahKursi ()** tipe data **int** untuk mengembalikan nilai **JumlahKursi**

Deklarasi method **setJumlahKursi (int JumlahKursi);** tipe data void untuk menampung nilai **JumlahKursi**.

Deklarasi method **getJumlahPintu ()** tipe data **int** untuk mengembalikan nilai **JumlahPintu**

Deklarasi method **setJumlahPintu (int JumlahPintu)** tipe data void untuk menampung nilai **JumlahPintu**.

Deklarasi method **getJumlahJendela ()** tipe data **int** untuk mengembalikan nilai **JumlahJendela**

Deklarasi method **setJumlahJendela (int JumlahJendela);** tipe data void untuk menampung nilai **JumlahJendela**.

KodisiRuangTest

```
import java.io.FileWriter;
```

```
public class KondisiRuangtest extends IndukClass implements ClassInterface{
```

```
    KondisiRuang KondisiKelas = new KondisiRuang();
```

```
    public static double KondisiRuang_Sesuai = 0.0;
```

```
    public double Panjang, Lebar , Luas, Rasio;
```

```
    public int jumlahkursi, jumlahpintu, jumlahjendela;
```

```
    public String bentuk;
```

```
    public void Input(double Panjang, double Lebar, int jumlahkursi, int jumlahpintu, int jumlahjendela) {
```

```
        @Override
```

```
        public double luas(){
```

```
            Luas = Panjang*Lebar;
```

```
            return Luas;
```

```
        }
```

```
        @Override
```

```
public String bentuk(){  
    if(Panjang != Luas){  
        bentuk = "Persegi Panjang";  
    }else{  
        bentuk = "Bukan Persegi panjang";  
    }  
    return bentuk;  
}
```

```
@Override  
public double rasio() {  
    Rasio = Luas/ jumlahkursi;  
    return Rasio;  
}
```

```
@Override  
public double Analisis() {  
    if(Panjang != Luas){  
        KondisiRuang_Sesuai ++;  
    }  
    if(Rasio >= 0.5){  
        KondisiRuang_Sesuai++;  
    }  
    if(jumlahpintu >= 2){  
        KondisiRuang_Sesuai++;  
    }  
    if(jumlahjendela >= 1){  
        KondisiRuang_Sesuai ++;  
    }  
    return KondisiRuang_Sesuai;  
}
```

```
@Override  
public double Sesuai() {  
    return KondisiRuang_Sesuai/4;  
}
```

```
@Override  
public void Save() {  
    try{  
        FileWriter Writer = new FileWriter("Kondisi.txt");
```

```

        Writer.write("Panjang Ruangan = "+Panjang);
        Writer.write("Lebar Ruangan = "+Lebar);
        Writer.write("jumlah Kursi = "+jumlahkursi);
        Writer.write("Jumlah Pintu = "+jumlahpintu);
        Writer.write("Jumlah Jendela = "+jumlahjendela);
        Writer.write("Luas Ruangan = "+Luas);
        Writer.write("Rasio Ruangan = "+Rasio);
        Writer.write("Bentuk Ruangan = "+bentuk);
        Writer.close();
    }catch (Exception ex){
        ex.printStackTrace();
    }
}

```

import java.io.FileWriter; Befungsi untuk menamapung writer
public class KondisiRuangtest extends IndukClass implements ClassInterface{ berikut adalah class KondisiRuangtes yang mewariskan atau mengambil method - method yang ada di IndukClass ke kelas KondisiRuangtes
KondisiRuang KondisiKelas = new KondisiRuang(); Membuat object KondisiKelas dari kelas KondisiRuang

public static double KondisiRuang_Sesuai = 0.0; Membuat Variabel baru double yang mana KondisiRuang_Sesuai = 0

public double Panjang, Lebar, Luas, Rasio; Membuat variabel double Panjang, Lebar, Luas, dan Rasio

public int jumlahkursi, jumlahpintu, jumlahjendela; Membuat variabel int jumlahkursi, jumlahpintu, dan jumlahjendela

public String bentuk; Mendeklarasi Variabel bentuk dengan menggunakan String

public void Input(double Panjang, double Lebar, int jumlahkursi, int jumlahpintu, int jumlahjendela) { Method void Input yang berisi double panjang, double lebar, int jumlahkursi, int jumlahpintu, dan int jumlahjendela

@Override

```

    public double luas(){
        Luas = Panjang*Lebar;
        return Luas;
    }

```

MengOverride method luas yang mana method luas tersebut menghitung luas yaitu luas = panjang*lebar ;
 dan memberikan nilai kembalian yang mana nilai kembaliannya hasil luas

@Override

```

    public String bentuk(){
        if(Panjang != Luas){

```

```

        bentuk = "Persegi Panjang";
    }else{
        bentuk = "Bukan Persegi panjang";
    }
    return bentuk;
}

```

MengOverride method bentuk yang mana method bentuk tersebut menghitung bentuk
 bentuk = jika panjang tidak sama dengan luas maka bentuk persegi panjang dan jika
 bentuk sama dengan luas maka bentuk bukan persegi panjang
 dan memberi nilai kembalian yang mana nilai kembalian hasil dari bentuk yang suda di
 hitung tersebut.

@Override

```

    public double rasio() {
        Rasio = Luas/ jumlahkursi;
        return Rasio;
    }

```

MengOverride method rasio dan memiliki nilai kembalian yaitu rasio

@Override

```

    public double Analisis() {
        if(Panjang != Luas){
            KondisiRuang_Sesuai ++;
        }
        if(Rasio >= 0.5){
            KondisiRuang_Sesuai++;
        }
        if(jumlahpintu >= 2){
            KondisiRuang_Sesuai++;
        }
        if(jumlahjendela >= 1){
            KondisiRuang_Sesuai ++;
        }
        return KondisiRuang_Sesuai;
    }

```

MengOverride method Analisis yang mana menAnalisi
 jika panjang tidak sama dengan luas maka KondisiRuang_Sesuai++
 jika Rasio >= 0.5 maka KondisiRuang_Sesui++
 jika jumlahpintu >=2 maka KondisiRuang_Sesuai++
 jika jumlahjendela >=1 maka KondisiRuang_Sesuai++
 dan memberi nilai kembalian hasil yang sudah di Analisis

@Override

```

    public double Sesuai() {

```



```
    return KondisiRuang_Sesuai/4;  
}
```

MengOverride method Sesuai dan memiliki nilai kembalian KonRuang_sesui/4;

@Override

```
public void Save() {  
    try{  
        FileWriter Writer = new FileWriter("Kondisi.txt");  
        Writer.write("Panjang Ruangan = "+Panjang);  
        Writer.write("Lebar Ruangan = "+Lebar);  
        Writer.write("jumlah Kursi = "+jumlahkursi);  
        Writer.write("Jumlah Pintu = "+jumlahpintu);  
        Writer.write("Jumlah Jendela = "+jumlahjendela);  
        Writer.write("Luas Ruangan = "+Luas);  
        Writer.write("Rasio Ruangan = "+Rasio);  
        Writer.write("Bentuk Ruangan = "+bentuk);  
        Writer.close();  
    }catch (Exception ex){  
        ex.printStackTrace();  
    }  
}
```

MengOverride method save dan didalamnya ada try catch yang mana try catch tersebut dapat menampung tulisan atau inputan dari user

SaranaRuangan

```
Private int JumlahSteker;  
Private int KondisiSteker;  
Private Int PosisiSteker;
```

```
Private Int JumlahKabelLCD;  
Private Int KondisiKabelLCD;  
Private Int PosisiKabelLCD;
```

```
Private int JumlahLampu;  
Private int KondisiLampu;  
Private int PosisiLampu;
```

```
int getJumlahSteker()  
void setJumlahSteker(int JumlahSteker)
```

```
int getKondisiSteker()
void setKondisiSteker(int KondisiSteker)

int getPosisiSteker()
void setPosisiSteker(int PosisiSteker)

int getJumlahKabelLCD()
void setJumlahKabelLCD(int JumlahKabelLCD)

int getKondisiKabelLCD()
void setKondisiKabelLCD(int KondisiKabelLCD)

int getPosisiKabelLCD()
void setPosisiKabelLCD(int PosisiKabelLCD)

int getJumlahLampu()
void setJumlahLampu(int JumlahLampu)

int getKondisiLampu()
void setKondisiLampu(int KondisiLampu)

int getPosisiLampu()
void setPosisiLampu(int PosisiLampu)
```

PrepCode :

Deklarasi **Private Int** untuk menampung jumlah steker diberi nama variable **JumlahSteker**

Deklarasi **Private int** untuk menampung kondisi steker diberi nama variable
KondisiSteker

Deklarasi **Private int** untuk menampung posisi steker dari ruangan diberi nama variable
PosisiSteker

Deklarasi **Private Int** untuk menampung jumlah kabel LCD diberi nama variable
JumlahKabelLCD

Deklarasi **Private int** untuk menampung Kondisi kabel LCD diberi nama variable
KondisiKabelLCD

Deklarasi **Private Int** untuk menampung Posisi kabel LCD diberi nama variable
PosisiKabelLCD

Deklarasi **Private Int** untuk menampung jumlah lampu diberi nama variable

JumlahLampu

Deklarasi **Private int** untuk menampung kondisi lampu diberi nama variable **KondisiLampu**

Deklarasi **Private int** untuk menampung posisi lampu dari ruangan diberi nama variable

PosisiLampu

Deklarasi method **getJumlahSteker()** tipe data **int** untuk mengembalikan nilai

JumlahSteker

Deklarasi method **setJumlahSteker(int JumlahSteker)** tipe data void untuk mengeset

JumlahSteker

Deklarasi method **getKondisiSteker()** tipe data **Int** untuk mengembalikan nilai

KondisiSteker

Deklarasi method **setKondisiSteker(int KondisiSteker)** tipe data void untuk mengeset

KondisiSteker

Deklarasi method **getPosisiSteker()** tipe data **int** untuk mengembalikan nilai **PosisiSteker**

Deklarasi method **setPosisiSteker(int PosisiSteker)** tipe data void untuk mengeset

PosisiSteker

Deklarasi method **getJumlahKabelLCD()** tipe data **int** untuk mengembalikan nilai

JumlahLCD

Deklarasi method **setJumlahKabelLCD(int JumlahKabelLCD)** tipe data void untuk mengeset **JumlahKabelLCD**

Deklarasi method **getKondisiKabelLCD()** tipe data **Int** untuk mengembalikan nilai

KondisiKabelLCD

Deklarasi method **setKondisiKabelLCD(int KondisiKabelLCD)** tipe data void untuk mengeset **KondisiKabelLCD**

Deklarasi method **getPosisiKabelLCD()** tipe data **int** untuk mengembalikan nilai

PosisiKabelLCD

Deklarasi method **setPosisiKabelLCD(int PosisiKabelLCD)** tipe data void untuk mengeset

PosisiKabelLCD

Deklarasi method **getJumlahLampu()** tipe data **int** untuk mengembalikan nilai

JumlahLampu

Deklarasi method **setJumlahLampu(int JumlahLampu)** tipe data void untuk mengeset

JumlahLampu

Deklarasi method **getKondisiLampu()** tipe data **Int** untuk mengembalikan nilai

KondisiLampu

Deklarasi method **setKondisiLampu(int KondisiLampu)** tipe data void untuk mengeset

KondisiLampu

Deklarasi method **getPosisiLampu()** tipe data **int** untuk mengembalikan nilai **PosisiLampu**

Deklarasi method **setPosisiLampu(int PosisiLampu)** tipe data void untuk mengeset **PosisiLampu**

SaranaTest

```
import java.io.FileWriter;
```

```
public class SaranaTest extends IndukClass{  
    SaranaRuangan KondisiKelas2 = new SaranaRuangan();  
    public static double SaranaRuang_Sesuai = 0.0;  
    public int JumlahSteker, KondisiStekerBaik, KondisiStekerBuruk, PosisiStekerDD,  
    PosisiStekerPR;  
    public int JumlahLampu, KondisiLampuBaik, KondisiLampuBuruk, PosisiLampuA;  
    public int JumlahKLCD, KondisiKLCDBaik, KondisiKLcdburuk, PosisiKLcddd;
```

```
    public void input(int JumlahSteker,int KondisiStekerBaik, int KondisiStekerBuruk, int  
    PosisiStekerDD, int PosisiStekerPR,  
        int JumlahLampu, int KondisiLampuBaik, int KondisiLampuBuruk, int  
    PosisiLampuA,  
        int JumlahKLCD, int KondisiKLCDBaik, int KondisiKLcdburuk, int PosisiKLcddd){  
        this.JumlahSteker = JumlahSteker;  
        this.KondisiStekerBaik = KondisiStekerBaik;  
        this.KondisiStekerBuruk = KondisiStekerBuruk;  
        this.PosisiStekerDD = PosisiStekerDD;  
        this.PosisiStekerPR = PosisiStekerPR;  
        this.JumlahLampu = JumlahLampu;  
        this.KondisiLampuBaik = KondisiLampuBaik;  
        this.KondisiLampuBuruk = KondisiLampuBuruk;  
        this.PosisiLampuA = PosisiLampuA;  
        this.JumlahKLCD = JumlahKLCD;  
        this.KondisiKLCDBaik = KondisiKLCDBaik;  
        this.KondisiKLcdburuk = KondisiKLcdburuk;  
        this.PosisiKLcddd = PosisiKLcddd;  
    }  
}
```

```

@Override
public double Analisis() {
    try{
        if(JumlahSteker>=4){
            SaranaRuang_Sesuai++;
        }
        if(KondisiStekerBaik == JumlahSteker && KondisiStekerBaik >= 4){
            SaranaRuang_Sesuai++;
        }
        if(PosisiStekerDD + PosisiStekerPR == JumlahSteker && PosisiStekerDD +
PosisiStekerPR >= 4){
            SaranaRuang_Sesuai++;
        }
        if(JumlahLampu>=18){
            SaranaRuang_Sesuai++;
        }
        if(KondisiLampuBaik == JumlahLampu && KondisiLampuBaik >= 18){
            SaranaRuang_Sesuai++;
        }
        if(PosisiLampuA == JumlahLampu && PosisiLampuA >= 18){
            SaranaRuang_Sesuai++;
        }
        if(JumlahKLCD>= 1){
            SaranaRuang_Sesuai++;
        }
        if(KondisiKLCDBaik == JumlahKLCD && KondisiKLCDBaik >= 1){
            SaranaRuang_Sesuai++;
        }
        if(PosisiKLCDDD == JumlahKLCD && PosisiKLCDDD >=1){
            SaranaRuang_Sesuai++;
        }
    }catch(Exception e){
        System.out.println("ERROR");
    }
    return SaranaRuang_Sesuai;
}

```

```

@Override
public double Sesuai() {
    return SaranaRuang_Sesuai/9;
}

```

@Override

public void Save() {

try{

FileWriter Writer = new FileWriter("SaranaKelas.txt");

Writer.write("Masukan Jumlah Steker = "+JumlahSteker);

Writer.write("Masukan Kondisi Steker Baik = "+KondisiStekerBaik);

Writer.write("Masukan Kondisi Steker Rusak = "+KondisiStekerBuruk);

Writer.write("Masukan Posisi Steker Dekat Dosen = "+PosisiStekerDD);

Writer.write("Masukan Posisi Steker Pojok Ruang = "+PosisiStekerPR);

Writer.write("Masukan Jumlah Kabel LCD = "+JumlahKLCD);

Writer.write("Masukan Kondisi Kabel LCD Baik = "+KondisiKLCDBaik);

Writer.write("Masukan Kondisi Kabel LCD Rusak = "+KondisiKLCDBuruk);

Writer.write("Masukan Posisi Kabel LCD Dekat Dosen = "+PosisiKLCDDD);

Writer.write("Masukan Jumlah Lampu = "+JumlahLampu);

Writer.write("Masukan Kondisi Lampu Baik = "+KondisiLampuBaik);

Writer.write("Masukan Kondisi Lampu Baik = "+KondisiLampuBuruk);

Writer.write("Masukan Posisi Lampu = "+PosisiLampuA);

}catch(Exception ex){

ex.printStackTrace();

}

import java.io.FileWriter; Befungsi untuk menamapung writer

public class SaranaTest extends IndukClass{ Method dari class SaranaTest mewariskan atau memanggil method - method yang ada di kelas IndukClass

SaranaRuangan KondisiKelas2 = new SaranaRuangan(); Membuat object baru Kondisikelas2 dari kelas SaranaRuangan

public static double SaranaRuang_Sesuai = 0.0; Membuat variabel SaranaRuang_Sesuai=0.0;

public int JumlahSteker, KondisiStekerBaik, KondisiStekerBuruk, PosisiStekerDD, PosisiStekerPR; Membuat variabel int JumlahStaker, KondisiStekerBaik, KondisiStekerBuruk, PosisiStekerDD, dan PosisiStekerPR;

public int JumlahLampu, KondisiLampuBaik, KondisiLampuBuruk, PosisiLampuA; Membuat variabelpublic int JumlahLampu, KondisiLampuBaik, KondisiLampuBuruk, dan PosisiLampuA;

public int JumlahKLCD, KondisiKLCDBaik, KondisiKLCDBuruk, PosisiKLCDDD; Membuat variabel public int JumlahKLCD, KondisiKLCDBaik, KondisiKLCDBuruk, PosisiKLCDDD;

```

    public void input(int JumlahSteker,int KondisiStekerBaik, int KondisiStekerBuruk, int
    PosisiStekerDD, int PosisiStekerPR,
        int JumlahLampu, int KondisiLampuBaik, int KondisiLampuBuruk, int
    PosisiLampuA,
        int JumlahKLCD, int KondisiKLCDBaik, int KondisiKLCDBuruk, int PosisiKLCDDD){

```

Method input yang mana berisi variabel int JumlahSteker,int KondisiStekerBaik, int KondisiStekerBuruk, int PosisiStekerDD, int PosisiStekerPR,int JumlahLampu, int KondisiLampuBaik, int KondisiLampuBuruk, int PosisiLampuA,int JumlahKLCD, int KondisiKLCDBaik, int KondisiKLCDBuruk, int PosisiKLCDDD

```

    this.JumlahSteker = JumlahSteker;
    this.KondisiStekerBaik = KondisiStekerBaik;
    this.KondisiStekerBuruk = KondisiStekerBuruk;
    this.PosisiStekerDD = PosisiStekerDD;
    this.PosisiStekerPR = PosisiStekerPR;
    this.JumlahLampu = JumlahLampu;
    this.KondisiLampuBaik = KondisiLampuBaik;
    this.KondisiLampuBuruk = KondisiLampuBuruk;
    this.PosisiLampuA = PosisiLampuA;
    this.JumlahKLCD = JumlahKLCD;
    this.KondisiKLCDBaik = KondisiKLCDBaik;
    this.KondisiKLCDBuruk = KondisiKLCDBuruk;
    this.PosisiKLCDDD = PosisiKLCDDD;
}

```

@Override

```

public double Analisis() {
    try{
        if(JumlahSteker>=4){
            SaranaRuang_Sesuai++;
        }
    }
}

```

MengOverride method Analisis yang mana didalamnya ada jika jumlahSteker >=4 maka SaranaRuangan_Sesuai ++;

```

        if(KondisiStekerBaik == JumlahSteker && KondisiStekerBaik >= 4){
            SaranaRuang_Sesuai++;
        }
    }
}

```

jika KondisiStekerBaik== JumlahSteker dan dan KondisiStekerBaik >=4 maka SaranaRuangan_Sesuai++

```

        if(PosisiStekerDD + PosisiStekerPR == JumlahSteker && PosisiStekerDD +
    PosisiStekerPR >= 4){
            SaranaRuang_Sesuai++;
        }
    }
}

```

jika PosisiStekerDD ditambah PosisiStekerPR samadengan samadengan jumlahSteker dan dan PosisiStekerDD ditambah PosisiStekerPR >= 4 maka SaranaRuangan_Sesuai++

```
if(JumlahLampu>=18){  
    SaranaRuang_Sesuai++;  
}
```

jika jumlahLampu >=18 maka SaranaRuangan_Sesuai++

```
if(KondisiLampuBaik == JumlahLampu && KondisiLampuBaik >= 18){  
    SaranaRuang_Sesuai++;  
}
```

jika KondisiLampuBaik samadengan samadengan jumlahLampu dan dan KondisiLampuBaik>=18 maka SaranaRuangan_Sesuai++

```
if(PosisiLampuA == JumlahLampu && PosisiLampuA >= 18){  
    SaranaRuang_Sesuai++;  
}
```

jika PosisiLampuA samadengan samadengan JumlahLampu dan dan PosisiLampuA >= 18 maka SaranaRuangan_Sesuai++;

```
if(JumlahKLCD>= 1){  
    SaranaRuang_Sesuai++;  
}
```

jika JumlahKLCD>= 1 maka SaranaRuangan_Sesuai++;

```
if(KondisiKLCDBaik == JumlahKLCD && KondisiKLCDBaik >= 1){  
    SaranaRuang_Sesuai++;  
}
```

jika KondisiKLCDBaik samadengan samadengan JumlahKLCD dan dan KondisiKLCDBaik >= 1 maka SaranaRuangan_Sesuai++;

```
if(PosisiKLCDDD == JumlahKLCD && PosisiKLCDDD >=1){  
    SaranaRuang_Sesuai++;  
}
```

jika PosisiKLCDDD samadengan samadengan JumlahKLCD dan dan PosisiKLCDDD >=1) maka SaranaRuangan_Sesuai++;

```
}catch(Exception e){  
    System.out.println("ERROR");  
}
```

catch menampilkan "ERROR"

```
return SaranaRuang_Sesuai;  
}
```

Memberikan nilai kembalian SaaranaRuangan_Sesuai

@Override


```

public double Sesuai() {
    return SaranaRuang_Sesuai/9;
}

```

MenOverride method Sesuai dan memberikan nilai kembalian SaranaRuang_Sesuai

@Override

```

public void Save() {
    try{
        FileWriter Writer = new FileWriter("SaranaKelas.txt");
        Writer.write("Masukan Jumlah Steker = "+JumlahSteker);
        Writer.write("Masukan Kondisi Steker Baik = "+KondisiStekerBaik);
        Writer.write("Masukan Kondisi Steker Rusak = "+KondisiStekerBuruk);
        Writer.write("Masukan Posisi Steker Dekat Dosen = "+PosisiStekerDD);
        Writer.write("Masukan Posisi Steker Pojok Ruang = "+PosisiStekerPR);
        Writer.write("Masukan Jumlah Kabel LCD = "+JumlahKLCD);
        Writer.write("Masukan Kondisi Kabel LCD Baik = "+KondisiKLCDBaik);
        Writer.write("Masukan Kondisi Kabel LCD Rusak = "+KondisiKLCDBuruk);
        Writer.write("Masukan Posisi Kabel LCD Dekat Dosen = "+PosisiKLCDDD);
        Writer.write("Masukan Jumlah Lampu = "+JumlahLampu);
        Writer.write("Masukan Kondisi Lampu Baik = "+KondisiLampuBaik);
        Writer.write("Masukan Kondisi Lampu Baik = "+KondisiLampuBuruk);
        Writer.write("Masukan Posisi Lampu = "+PosisiLampuA);
    }catch(Exception ex){
        ex.printStackTrace();
    }
}

```

MengOverride method save dan pembuntan saving objek yang di try catch menulis atau yang di inputkan oleh user.

SaranaRuangan2

```

private int JumlahKipasAngin;
private int KondisiKipasAngin;
private int PosisiKipasAngin;

private int JumlahAC;

```

```
private int KondisiAC;  
private int PosisiAC;
```

```
private int SSID;  
private int Bandwith;
```

```
private int JumlahCCTV;  
private int KondisiCCTV;  
private int PosisiCCTV;
```

```
int getJumlahKipasAngin()  
void setJumlahKipasAngin(int JumlahKipasAngin)  
int getKondisiKipasAngin()  
void setKondisiKipasAngin(int KondisiKipasAngin)  
int getPosisiKipasAngin()  
void setPosisiKipasAngin(int PosisiKipasAngin)
```

```
int getJumlahAC()  
void setJumlahAC(int JumlahAC)  
int getKondisiAC()  
void setKondisiAC(int KondisiAC)  
int getPosisiAC()  
void setPosisiAC(int PosisiAC)
```

```
int getJumlahCCTV()  
void setJumlahCCTV(int JumlahCCTV)  
int getKondisiCCTV()  
void setKondisiCCTV(int KondisiCCTV)  
int getPosisiCCTV()  
void setPosisiCCTV(int PosisiCCTV)
```

```
int getSSID()  
void setSSID(int SSID)  
int getBandwith()  
void setBandwith(int Bandwith)
```

PrepCode :

Deklarasi **Private Int** untuk menampung jumlah kipas angina diberi nama variable **JumlahKipasAngin**

Deklarasi **Private int** untuk menampung kondisi kipas angin diberi nama variable **KondisiKipasAngin**

Deklarasi **Private int** untuk menampung posisi kipas angin dari ruangan diberi nama variable **PosisiKipasAngin**

Deklarasi **Private Int** untuk menampung jumlah AC diberi nama variable **JumlahAC**

Deklarasi **Private int** untuk menampung Kondisi AC diberi nama variable **KondisiAC**

Deklarasi **Private Int** untuk menampung Posisi AC diberi nama variable **PosisiAC**

Deklarasi **Private Int** untuk menampung SSID diberi nama variable **SSID**

Deklarasi **Private int** untuk menampung Bandwith diberi nama variable **Bandwith**

Deklarasi **Private Int** untuk menampung jumlah CCTV diberi nama variable **JumlahCCTV**

Deklarasi **Private int** untuk menampung Kondisi CCTV diberi nama variable **KondisiCCTV**

Deklarasi **Private Int** untuk menampung Posisi CCTV diberi nama variable **PosisiCCTV**

Deklarasi method **getJumlahKipasAngin()** typedata **int** untuk mengembalikan nilai **JumlahKipasAngin**

Deklarasi method **setKipasAngin (int JumlahKipasAngin)** tipe data void untuk mengeset **JumlahKipasAngin**

Deklarasi method **getKondisiKipasAngin()** typedata **Int** untuk mengembalikan nilai **KondisiKipasAngin**

Deklarasi method **setKondisiKipasAngin(int KondisiSKipasAngin)** tipe data void untuk mengeset **KondisiKipasAngin**

Deklarasi method **getPosisiKipasAngin()** typedata **int** untuk mengembalikan nilai **PosisiKipasAngin**

Deklarasi method **setPosisiKipasAngin(int PosisiKipasAngin)** tipe data void untuk mengeset **PosisiKipasAngin**

Deklarasi method **getJumlahAC()** typedata **int** untuk mengembalikan nilai **JumlahAC**

Deklarasi method **setJumlahAC(int JumlahAC)** tipe data void untuk mengeset **JumlahAC**

Deklarasi method **getKondisiAC()** typedata **Int** untuk mengembalikan nilai **KondisiAC**

Deklarasi method **setKondisiAC(int KondisiAC)** tipe data void untuk mengeset **KondisiAC**

Deklarasi method **getPosisiAC()** typedata **int** untuk mengembalikan nilai **PosisiAC**

Deklarasi method **setPosisiAC(int PosisiAC)** tipe data void untuk mengeset **PosisiAC**

Deklarasi method **getSSID()** typedata **int** untuk mengembalikan nilai **SSID**

Deklarasi method **setSSID(int SSID)** tipe data void untuk mengeset **SSID**

Deklarasi method **getBandwith()** typedata **Int** untuk mengembalikan nilai **Bndwith**

Deklarasi method **setBandwith(int Bandwith)** tipe data void untuk mengeset **Bandwith**

Deklarasi method **getJumlahCCTV()** typedata **int** untuk mengembalikan nilai **JumlahCCTV**

Deklarasi method **setJumlahCCTV(int JumlahCCTV)** tipe data void untuk mengeset **JumlahCCTV**

Deklarasi method **getKondisiCCTV()** tipe data **Int** untuk mengembalikan nilai **KondisiCCTV**
Deklarasi method **setKondisiCCTV(int KondisiCCTV)** tipe data void untuk mengeset **KondisiCCTV**
Deklarasi method **getPosisiCCTV()** tipe data **int** untuk mengembalikan nilai **PosisiCCTV**
Deklarasi method **setPosisiCCTV(int PosisiCCTV)** tipe data void untuk mengeset **PosisiCCTV**

SaranaTest2

```
import java.io.FileWriter;  
import java.io.IOException;  
import javax.swing.JOptionPane;
```

```
public class SaranaTest2 extends IndukClass {
```

```
    public int JumlahKA, KondisiKABaik, KondisiKABuruk, PosisiKAAatap;  
    public int JumlahAC, KondisiACBaik, KondisiACBuruk, PosisiACBelakang,  
    PosisiACSamping;  
    public static double SaranaRuang2_Sesuai = 0.0;
```

```
    public void input(int JumlahKA,int KondisiKABaik, int KondisiKABuruk, int PosisiKAAatap,  
        int JumlahAC, int KondisiACBaik, int KondisiACBuruk, int PosisiACBelakang, int  
    PosisiACSamping,  
        int JumlahCCTV, int KondisiCCTVBaik, int KondisiCCTVBuruk, int  
    PosisiCCTVBelakang, int PosisiCCTVDepan,  
        String SSID, int Bandwith){  
        this.JumlahKA = JumlahKA;  
        this.KondisiKABaik = KondisiKABaik;  
        this.KondisiKABuruk = KondisiKABuruk;  
        this.PosisiKAAatap = PosisiKAAatap;  
        this.JumlahAC = JumlahAC;  
        this.KondisiACBaik = KondisiACBaik;  
        this.KondisiACBuruk = KondisiACBuruk;  
        this.PosisiACBelakang = PosisiACBelakang;  
        this.PosisiACSamping = PosisiACSamping;  
        this.JumlahCCTV = JumlahCCTV;  
        this.KondisiCCTVBaik = KondisiCCTVBaik;  
        this.KondisiCCTVBuruk = KondisiCCTVBuruk;  
        this.PosisiCCTVBelakang = PosisiCCTVBelakang;
```

```

this.PosisiCCTVDepan = PosisiCCTVDepan;
this.SSID = SSID;
this.Bandwith = Bandwith;
}

@Override
public double Analisis() {
try{
if(JumlahKA >=2){
SaranaRuang2_Sesuai++;
}
if(KondisiKABaik == JumlahKA && KondisiKABaik >= 2){
SaranaRuang2_Sesuai++;
}
if(PosisiKAAtap == JumlahKA && PosisiKAAtap >= 2){
SaranaRuang2_Sesuai++;
}
if(JumlahAC>=1){
SaranaRuang2_Sesuai++;
}
if(KondisiACBaik == JumlahAC && KondisiACBaik >= 1){
SaranaRuang2_Sesuai++;
}
if(PosisiACBelakang +PosisiACSamping== JumlahAC && PosisiACBelakang
+PosisiACSamping >= 1){
SaranaRuang2_Sesuai++;
}
if(JumlahCCTV>= 2){
SaranaRuang2_Sesuai++;
}
if(KondisiCCTVBaik == JumlahCCTV && KondisiCCTVBaik >= 2){
SaranaRuang2_Sesuai++;
}
if(PosisiCCTVBelakang+ PosisiCCTVDepan == JumlahCCTV && PosisiCCTVBelakang
+ PosisiCCTVDepan >=2){
SaranaRuang2_Sesuai++;
}
if("UMM Hotspot".equals(SSID)){
SaranaRuang2_Sesuai++;
}
if("UMM Hospott".equals(SSID) && Bandwith>=1 ){

```

```
SaranaRuang2_Sesuai++;  
}  
}catch(Exception e){  
System.out.println("ERROR");  
}
```

```
return SaranaRuang2_Sesuai;  
}
```

```
@Override  
public double Sesuai() {  
return SaranaRuang2_Sesuai/11;  
}
```

```
@Override  
public void Save() {  
try{  
FileWriter Writer = new FileWriter("KondisiKelas3.txt");  
Writer.write("Masukan Jumlah Kipas Angin = "+JumlahKA);  
Writer.write("Masukan Kondisi Kipas Angin Baik = "+KondisiKABaik);  
Writer.write("Masukan Kondisi Kipas Angin Buruk = "+KondisiKABuruk);  
Writer.write("Masukan Posisi Kipas Angin = "+PosisiKAAtap);  
Writer.write("Masukan Jumlah AC = "+JumlahAC);  
Writer.write("Masukan Kondisi AC Baik = "+KondisiACBaik);  
Writer.write("Masukan Kondisi AC Buruk = "+KondisiACBuruk);  
Writer.write("Masukan Posisi AC Belakang = "+PosisiACBelakang);  
Writer.write("Masukan Posisi AC Samping = "+PosisiACSamping);  
Writer.write("Masukan SSID = "+SSID);  
Writer.write("Masukan Bandwith = "+Bandwith);  
Writer.write("Masukan Jumlah CCTV = "+JumlahCCTV);  
Writer.write("Masukan Kondisi CCTV Baik = "+KondisiCCTVBaik);  
Writer.write("Masukan Kondisi CCTV Buruk = "+KondisiCCTVBuruk);  
Writer.write("Masukan Posisi CCTV Belakang = "+PosisiCCTVBelakang);  
Writer.write("Masukan Posisi CCTV Depan = "+PosisiCCTVDepan);  
}catch(Exception ex){  
ex.printStackTrace();  
}  
}
```

```
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.JOptionPane;
```

public class SaranaTest2 extends IndukClass { Kelas SaranaTest2 mewariskan atau mengambil method - method dari super class IndukClass ke SaranaTest2

public int JumlahKA, KondisiKABaik, KondisiKABuruk, PosisiKAAtap; Membuat variabel baru int JumlahKA, KondisiKABaik, KondisiKABuruk, PosisiKAAtap;

public int JumlahAC, KondisiACBaik, KondisiACBuruk, PosisiACBelakang, PosisiACSamping; Membuat variabel int JumlahAC, KondisiACBaik, KondisiACBuruk, PosisiACBelakang, PosisiACSamping;

public static double SaranaRuang2_Sesuai = 0.0; Membuat variabel SaranaRuang2_Sesuai=0.0:

public static double SaranaRuang2_Sesuai = 0.0;

Mambuat variabel double SaranaRuang2_Sesuai = 0.

```
public void input(int JumlahKA,int KondisiKABaik, int KondisiKABuruk, int PosisiKAAtap,
    int JumlahAC, int KondisiACBaik, int KondisiACBuruk, int PosisiACBelakang, int
    PosisiACSamping,
    int JumlahCCTV, int KondisiCCTVBaik, int KondisiCCTVBuruk, int
    PosisiCCTVBelakang, int PosisiCCTVDepan){
    this.JumlahKA = JumlahKA;
    this.KondisiKABaik = KondisiKABaik;
    this.KondisiKABuruk = KondisiKABuruk;
    this.PosisiKAAtap = PosisiKAAtap;
    this.JumlahAC = JumlahAC;
    this.KondisiACBaik = KondisiACBaik;
    this.KondisiACBuruk = KondisiACBuruk;
    this.PosisiACBelakang = PosisiACBelakang;
    this.PosisiACSamping = PosisiACSamping;
    this.JumlahCCTV = JumlahCCTV;
    this.KondisiCCTVBaik = KondisiCCTVBaik;
    this.KondisiCCTVBuruk = KondisiCCTVBuruk;
    this.PosisiCCTVBelakang = PosisiCCTVBelakang;
    this.PosisiCCTVDepan = PosisiCCTVDepan;
}
```

Method input di dalamnya ada berbagai variabel diantaranya int JumlahKA,int KondisiKABaik, int KondisiKABuruk, int PosisiKAAtap,

```
int JumlahAC, int KondisiACBaik, int KondisiACBuruk, int PosisiACBelakang, int
PosisiACSamping,
int JumlahCCTV, int KondisiCCTVBaik, int KondisiCCTVBuruk, int
PosisiCCTVBelakang, int PosisiCCTVDepan)
```

```
@Override
```

```
public double Analisis() {
```

```
MengOverride method Analisis
```

```
try{
```

```
if(JumlahKA >=2){
```

```
SaranaRuang2_Sesuai++;
```

```
}
```

```
Jika JumlahKA lebih dari samadengan 2 maka SaranaRuang2_Sesuai
```

```
if(KondisiKABaik == JumlahKA && KondisiKABaik >= 2){
```

```
SaranaRuang2_Sesuai++;
```

```
}
```

```
Jika KondisiKABaik samadengan samadengan JumlahKA dan dan KondisiKABaik lebih
besar samadengan 2 maka SaranaRuang2_Sesuai
```

```
if(PosisiKAAatap == JumlahKA && PosisiKAAatap >= 2){
```

```
SaranaRuang2_Sesuai++;
```

```
}
```

```
Jika PosisiKAAatap samadengan samadengan JumlahKA dan dan PosisiKAAatap lebih besar
samadengan 2 maka SaranaRuang2_Sesuai
```

```
if(JumlahAC>=1){
```

```
SaranaRuang2_Sesuai++;
```

```
}
```

```
Jika JumlahAC lebih besar samadengan 1 maka SaranaRuang2_Sesuai
```

```
if(KondisiACBaik == JumlahAC && KondisiACBaik >= 1){
```

```
SaranaRuang2_Sesuai++;
```

```
}
```

```
Jika KondisiACBaik sama dengan sama dengan JumlahAC dan dan KondisiACBaik lebih dari
sama dengan 1 maka SaranaRuang2_Sesuai
```

```
if(PosisiACBelakang +PosisiACSamping== JumlahAC && PosisiACBelakang
+PosisiACSamping >= 1){
```

```
SaranaRuang2_Sesuai++;
```

```
}
```

```
Jika PosisiACBelakang di tambah PosisiACSamping sama dengan sama dengan JumlahAC
dan dan PosisiACBelakang di tambah PosisiACSamping lebih dari sama dengan 1 maka
SaranaRuang2_Sesuai
```

```
if(JumlahCCTV>= 2){
```

```
SaranaRuang2_Sesuai++;
```



```

    }
    Jika JumlahCCTV lebih dari sama dengan 2 maka JumlahCCTV
        if(KondisiCCTVBaik == JumlahCCTV && KondisiCCTVBaik >= 2){
            SaranaRuang2_Sesuai++;
        }
    Jika KondisiCCTVBaik sama dengan sama dengan JumlahCCTV dan dan KondisiCCTVBaik
    lebih dari sama dengan 2 maka SaranaRuang2_Sesuai
        if(PosisiCCTVBelakang+ PosisiCCTVDepan == JumlahCCTV && PosisiCCTVBelakang
+ PosisiCCTVDepan >=2){
            SaranaRuang2_Sesuai++;
        }
    Jika PosisiCCTVBelakang+ PosisiCCTVDepan == JumlahCCTV && PosisiCCTVBelakang +
    PosisiCCTVDepan >=2 maka SaranaRuang2_Sesuai
        }catch(Exception e){
            System.out.println("ERROR");
        }

        return SaranaRuang2_Sesuai;
    }
    Memiliki nilai kembalian SaranaRuang2_Sesuai

    @Override
    public double Sesuai() {
        return SaranaRuang2_Sesuai/11;
    }
}

MengOverride method sesuai dan memiliki nilai kembalian SaranaRuang2_Sesuai/11
@Override
public void Save() {
    try{
        FileWriter Writer = new FileWriter("KondisiKelas3.txt");
        Writer.write("Masukan Jumlah Kipas Angin = "+JumlahKA);
        Writer.write("Masukan Kondisi Kipas Angin Baik = "+KondisiKABaik);
        Writer.write("Masukan Kondisi Kipas Angin Buruk = "+KondisiKABuruk);
        Writer.write("Masukan Posisi Kipas Angin = "+PosisiKAAtap);
        Writer.write("Masukan Jumlah AC = "+JumlahAC);
        Writer.write("Masukan Kondisi AC Baik = "+KondisiACBaik);
        Writer.write("Masukan Kondisi AC Buruk = "+KondisiACBuruk);
        Writer.write("Masukan Posisi AC Belakang = "+PosisiACBelakang);
        Writer.write("Masukan Posisi AC Samping = "+PosisiACSamping);
    }
}

```

```

        Writer.write("Masukan SSID =" +SSID);
        Writer.write("Masukan Bandwith =" +Bandwith);
        Writer.write("Masukan Jumlah CCTV =" +JumlahCCTV);
        Writer.write("Masukan Kondisi CCTV Baik =" +KondisiCCTVBaik);
        Writer.write("Masukan Kondisi CCTV Buruk =" +KondisiCCTVBuruk);
        Writer.write("Masukan Posisi CCTV Belakang =" +PosisiCCTVBelakang);
        Writer.write("Masukan Posisi CCTV Depan =" +PosisiCCTVDepan);
    }catch(Exception ex){
        ex.printStackTrace();
    }
}

```

mengoverride method save yang mana di dalamnya ada try catch nya

LingkunganRuangKelas

```

private int KondisiLantai;
private int KondisiDinding;
private int KondisiAtap;
private int KondisiPintu;
private int KondisiJendela;

```

```

public int getKondisiLantai()
public void setKondisiLantai(int KondisiLantai)
public int getKondisiDinding
public void setKondisiDinding(int KondisiDinding)
public int getKondisiAtap()
public void setKondisiAtap(int KondisiAtap)
public int getKondisiPintu()
public void setKondisiPintu(int KondisiPintu)
public int getKondisiJendela()
public void setKondisiJendela(int KondisiJendela)

```

PrepCode :

Deklarasi **Private Int** untuk menampung Kondisi lantai diberi nama variable **KondisiLantai**

Deklarasi **Private int** untuk menampung kondisi dinding diberi nama variable

KondisiDinding

Deklarasi **Private int** untuk menampung kondisi atap dari ruangan diberi nama variable **KondisiAtap**

Deklarasi **Private Int** untuk menampung kondisi pintu diberi nama variable **KondisiPintu**

Deklarasi **Private int** untuk menampung kondisi jendela diberi nama variable **KondisiJendela**

Deklarasi method **getKondisiLantai()** typedata **int** untuk mengembalikan nilai **KondisiLantai**

Deklarasi method **setKondisiLantai(int KondisiLantai)** tipe data void untuk mengeset **KondisiLantai**

Deklarasi method **getKondisiDinding()** typedata **Int** untuk mengembalikan nilai **KondisiDinding**

Deklarasi method **setKondisiDinding(int KondisiDinding)** tipe data void untuk mengeset **KondisiDinding**

Deklarasi method **getKondisiAtap()** typedata **int** untuk mengembalikan nilai **KondisiAtap**

Deklarasi method **setKondisiAtap(int KondisiAtap)** tipe data void untuk mengeset **KondisiAtap**

Deklarasi method **getKondisiPintu()** typedata **int** untuk mengembalikan nilai **KondisiPintu**

Deklarasi method **setKondisiPintu(int KondisiPintu)** tipe data void untuk mengeset **KondisiPintu**

Deklarasi method **getKondisiJendela()** typedata **Int** untuk mengembalikan nilai **KondisiJendela**

Deklarasi method **setKondisiJendela(int KondisiJendela)** tipe data void untuk mengeset **KondisiJendela**

LingkunganTest extends IndukClass

```
import java.io.FileWriter;  
import java.io.IOException;
```

```
public class LingkunganTest extends IndukClass{  
    LingkunganRuangKelas LingkunganKelas =new LingkunganRuangKelas();  
    public static double Lingkungan_Sesuai = 0.0;  
    public String KondisiLantai, KondisiDinding, KondisiAtap, KondisiPintu,  
    KondisiJendela;
```

```
    public void Input(String KondisiLantai, String KondisiDinding, String KondisiAtap, String  
    KondisiPintu, String KondisiJendela) {  
        this.KondisiLantai = KondisiLantai;  
        this.KondisiDinding = KondisiDinding;
```

```
this.KondisiAtap = KondisiAtap;  
this.KondisiPintu = KondisiPintu;  
this.KondisiJendela = KondisiJendela;  
}
```

```
@Override  
public double Analisis() {  
    try{  
        if("Bersih".equals(KondisiLantai)){  
            Lingkungan_Sesuai++;  
        }  
        if("Bersih".equals(KondisiDinding)){  
            Lingkungan_Sesuai++;  
        }  
        if("Bersih".equals(KondisiAtap)){  
            Lingkungan_Sesuai++;  
        }  
        if("Bersih".equals(KondisiPintu)){  
            Lingkungan_Sesuai++;  
        }  
        if("Bersih".equals(KondisiJendela)){  
            Lingkungan_Sesuai++;  
        }  
    }catch(Exception e){  
        System.out.println("ERROR");  
    }  
    return Lingkungan_Sesuai;  
}
```

```
@Override  
public double Sesuai() {  
    return Lingkungan_Sesuai/5;  
}
```

```
@Override  
public void Save() {  
    try{  
        FileWriter Writer = new FileWriter("Lingkungan.txt");  
        Writer.write("Kondisi Lantai Ruangan = " + KondisiLantai+"\t");  
        Writer.write("Kondisi Dinding Ruangan = " + KondisiDinding+"\t");  
        Writer.write("Kondisi Atap Ruangan = " + KondisiAtap+"\t");  
        Writer.write("Kondisi Pintu Ruangan = " + KondisiPintu+"\t");  
    }  
}
```

```

        Writer.write("Kondisi Jendela Ruangan = " + KondisiJendela+"\t");
        Writer.close();
    }catch(IOException ex){
        ex.printStackTrace();
    }
}
}

```

```

import java.io.FileWriter;
import java.io.IOException;

```

public class LingkunganTest extends IndukClass{ kelas LingkunganTest mewariskan atau menurunkan method method yang ada di super class IndukClass ke class LingkunganTest

```

    LingkunganRuangKelas LingkunganKelas =new LingkunganRuangKelas();

```

Membuat object baru LingkunaganKelas dari kelas LingkunganRuangKelas

```

    public static double Lingkungan_Sesuai = 0.0; Membuat variabel

```

```

    Lingkungan_Sesuai = 0.0

```

```

    public String KondisiLantai, KondisiDinding, KondisiAtap, KondisiPintu,
    KondisiJendela;

```

Membuat variabel String KondisiLantai, KondisiDinding, KondisiAtap, KondisiPintu, KondisiJendela

```

public void Input(String KondisiLantai, String KondisiDinding, String KondisiAtap, String
    KondisiPintu, String KondisiJendela) {

```

```

    this.KondisiLantai = KondisiLantai;

```

```

    this.KondisiDinding = KondisiDinding;

```

```

    this.KondisiAtap = KondisiAtap;

```

```

    this.KondisiPintu = KondisiPintu;

```

```

    this.KondisiJendela = KondisiJendela;

```

```

}

```

Membuat method input yang di dalamnya ada variabel String KondisiLantai, String KondisiDinding, String KondisiAtap, String KondisiPintu, String KondisiJendela

```

    @Override

```

```

    public double Analisis() {

```

MengOverride method Analisis

```

        try{
            if("Bersih".equals(KondisiLantai)){
                Lingkungan_Sesuai++;
                Jika bersih equals KondisiLantai maka Lingkungan_Sesuai++
            }
            if("Bersih".equals(KondisiDinding)){
                Lingkungan_Sesuai++;
                Jika bersih equals KondisiDinding maka Lingkungan_Sesuai++
            }
            if("Bersih".equals(KondisiAtap)){
                Lingkungan_Sesuai++;
            }
            Jika bersih equals KondisiAtap maka Lingkungan_Sesuai++

            if("Bersih".equals(KondisiPintu)){
                Lingkungan_Sesuai++;
            }
            Jika bersih equals KondisiPintu maka Lingkungan_Sesuai++

            if("Bersih".equals(KondisiJendela)){
                Lingkungan_Sesuai++;
            }
            Jika bersih equals KondisiJendela maka Lingkungan_Sesuai++
        }catch(Exception e){
            System.out.println("ERROR");
        }
        return Lingkungan_Sesuai;
    }
    Memiliki nilai kembalian Lingkungan_Sesuai

```

```

    @Override
    public double Sesuai() {
        return Lingkungan_Sesuai/5;
    }

    @Override
    public void Save() {
        try{
            FileWriter Writer = new FileWriter("Lingkungan.txt");
            Writer.write("Kondisi Lantai Ruangan = " + KondisiLantai+"\t");
            Writer.write("Kondisi Dinding Ruangan = " + KondisiDinding+"\t");

```

```

        Writer.write("Kondisi Atap Ruangan = " + KondisiAtap+"\t");
        Writer.write("Kondisi Pintu Ruangan = " + KondisiPintu+"\t");
        Writer.write("Kondisi Jendela Ruangan = " + KondisiJendela+"\t");
        Writer.close();
    }catch(IOException ex){
        ex.printStackTrace();
    }
}

```

}

Mengoverride method save didalamnya terdapat try catch yang menampung method method untuk di inputkan user

KebersihanRuangKelas

```

private int SirkulasiUdara;
private int nilaiPencahayaan;
private int Kelembapan;
private int suhu;

public void setSirkulasiUdara(int SirkulasiUdara)
public int getSirkulasiUdara()

public void setnilaiPencahayaan(int
nilaiPencahayaan)
public int getnilaiPencahayaan()

public void setKelembapan(int Kelembapan)
public int getKelembapan()

public void setsuhu(int suhu)
public int getsuhu()

```

PrepCode :

Deklarasi **Private Int** untuk menampung Kondisi sirkulasi udara diberi nama variable **SirkulasiUdara**

Deklarasi **Private int** untuk menampung nilai pencahayaan diberi nama variable **nilaiPencahayaan**

Deklarasi **Private int** untuk menampung nilai kelembapan dari ruangan diberi nama variable **Kelembapan**

Deklarasi **Private Int** untuk menampung nilai suhu diberi nama variable **suhu**

Deklarasi method **get SirkulasiUdara ()** typedata **int** untuk mengembalikan nilai **SirkulasiUdara**

Deklarasi method **set SirkulasiUdara (int SirkulasiUdara)** tipe data void untuk mengeset **SirkulasiUdara**

Deklarasi method **getnilaiPencahayaan ()** typedata **Int** untuk mengembalikan nilai **nilaiPencahayaan**

Deklarasi method **setnilaiPencahayaan (int nilaiPencahayaan)** tipe data void untuk mengeset **nilaiPencahayaan**

Deklarasi method **getKelembapan ()** typedata **int** untuk mengembalikan nilai **Kelembapan**

Deklarasi method **setKelembapan (int Kelembapan)** tipe data void untuk mengeset **Kelembapan**

Deklarasi method **getsuhu ()** typedata **Int** untuk mengembalikan nilai **suhu**

Deklarasi method **setsuhu (int suhu)** tipe data void untuk mengeset **suhu**

KebersihanTest extends IndukClass

```
import java.io.FileWriter;  
import java.io.IOException;
```

```
public class KebersihanTest extends IndukClass{  
    KebersihanRuangKelas KebersihanKelas = new KebersihanRuangKelas();  
    public static double Kebersihan_Sesuai = 0.0;  
    public String SirkulasiUdara;  
    public int Pencahayaan, Kelembapan, Suhu;
```

```
public void Input(String SirkulasiUdara, int Pencahayaan,int Kelembapan, int Suhu) {  
    this.SirkulasiUdara = SirkulasiUdara;
```



```
this.Pencahayaayan = Pencahayaayan;  
this.Kelembapan = Kelembapan;  
this.Suhu = Suhu;  
}
```

```
@Override  
public double Analisis(){  
try{  
if(SirkulasiUdara.equals("Lancar")){  
Kebersihan_Sesuai++;  
}  
if(Pencahayaayan >= 250 && Pencahayaayan <=350){  
Kebersihan_Sesuai++;  
}  
if(Kelembapan >= 70 && Kelembapan<= 80){  
Kebersihan_Sesuai++;  
}  
if(Suhu >= 25 && Suhu <= 35){  
Kebersihan_Sesuai++;  
}  
}catch(Exception e){  
System.out.println("ERROR");  
}  
return Kebersihan_Sesuai;  
}
```

```
@Override  
public double Sesuai() {  
return Kebersihan_Sesuai/4;  
}
```

```
@Override  
public void Save() {  
try{  
FileWriter Writer = new FileWriter("Kebersihan.txt");  
Writer.write("Sirkulasi Udara Ruangan = " + SirkulasiUdara+"\t");  
Writer.write("Pencahayaayan Ruangan = " + Pencahayaayan+"\t");  
Writer.write("Kelembapan Ruangan= " + Kelembapan+"\t");  
Writer.write("Suhu Ruangan= " + Suhu+"\t");  
Writer.close();  
}catch(IOException ex){  
ex.printStackTrace();  
}
```

```
}  
}
```

```
import java.io.FileWriter;  
import java.io.IOException;
```

public class KebersihanTest extends IndukClass{ Mewariskan atau menurunkan dari method method yang ada di superclass dari class IndukClass ke class subclass

KebersihanTest

KebersihanRuangKelas KebersihanKelas = new KebersihanRuangKelas(); Membuat object KebersihanKelas dari class KebersihanRuangKelas

public static double Kebersihan_Sesuai = 0.0; Membuat variabel Kebersihan_Sesuai sama dengan 0.0

public String SirkulasiUdara; Membuat variabel String SirkulasiUdara
public int Pencahayaan, Kelembapan, Suhu; Membuat variabel int Pencahayaan, Kelembapan, Suhu

```
public void Input(String SirkulasiUdara, int Pencahayaan,int Kelembapan, int Suhu) {  
    this.SirkulasiUdara = SirkulasiUdara;  
    this.Pencahayaan = Pencahayaan;  
    this.Kelembapan = Kelembapan;  
    this.Suhu = Suhu;  
}
```

Method input yang di dalamnya ada variabel String SirkulasiUdara, int Pencahayaan,int Kelembapan, int Suhu

```
@Override  
public double Analisis(){  
Mengoverride method Analisis  
    try{  
        if(SirkulasiUdara.equals("Lancar")){  
            Kebersihan_Sesuai++;  
        }  
    }
```

Jika SirkuitUdara equals Lancar maka Kebersihan_Sesuai++

```
        if(Pencahayaan >= 250 && Pencahayaan <=350){  
            Kebersihan_Sesuai++;  
        }  
    }
```

Jika Pencahayaan lebih besar sama dengan 250 dan dan Pencahayaan kurang dari sama dengan 350 maka Kebersihan_Sesuai++

```
if(Kelembapan >= 70 && Kelembapan<= 80){  
    Kebersihan_Sesuai++;  
}
```

Jika Kelembapan lebih dari sama dengan 70 dan dan Kelembapan kurang dari sma dengan 80 maka Kebersihan_Sesuai+

```
if(Suhu >= 25 && Suhu <= 35){  
    Kebersihan_Sesuai++;  
}
```

Jika Suhu lebih besar sama dengan 25 dan dan Suhu kurang dari sama dengan 35 maka Kebersihan_Sesuai++

```
}catch(Exception e){  
    System.out.println("ERROR");  
}  
return Kebersihan_Sesuai;  
}
```

Memiliki nilai kembaliam Kebersiha _Sesuai

```
@Override  
public double Sesuai() {  
    return Kebersihan_Sesuai/4;  
}
```

MengOverride method Sesuai dan memiliki nilai kembalian Kebersihan_Sesuai/4

@Override

```
public void Save() {  
    try{  
        FileWriter Writer = new FileWriter("Kebersihan.txt");  
        Writer.write("Sirkulasi Udara Ruangan = " + SirkulasiUdara+"\n");  
        Writer.write("Pencahayaan Ruangan = " + Pencahayaan+"\n");  
        Writer.write("Kelembapan Ruangan= " + Kelembapan+"\n");  
        Writer.write("Suhu Ruangan= " + Suhu+"\n");  
        Writer.close();  
    }catch(IOException ex){  
        ex.printStackTrace();  
    }  
}
```

MengOverride method save dan di dalam method save terdapat try catch yang fungsinya menampung atau menulis atau menginputkan yang user inputkan

KenyamananRuangan
<pre>private int Kebisingan; private int Bau; private int Kebocoran; private int Kerusakan; private int Keausan;</pre>
<pre>public void setKebisingan(int Kebisingan) public int getKebisingan () public void setBau(int Bau) public int getBau () public void setKebocoran(int Kebocoran) public int getKebocoran () public void setKerusakan(int Kerusakan) public int getKerusakan () public void setKeausan(int Keausan) public int getKeausan ()</pre>

PrepCode :

Deklarasi **Private Int** untuk menampung Kondisi kebisingan diberi nama variable **Kebisingan**

Deklarasi **Private int** untuk menampung kondisi bau diberi nama variable **Bau**

Deklarasi **Private int** untuk menampung kondisi kebocoran dari ruangan diberi nama variable **Kebocoran**

Deklarasi **Private Int** untuk menampung kondisi kerusakan diberi nama variable **Kerusakan**

Deklarasi **Private int** untuk menampung kondisi keausan diberi nama variable **Keausan**

Deklarasi method **getKebisingan()** tipe data **int** untuk mengembalikan nilai **Kebisingan**

Deklarasi method **setKebisingan(int Kebisingan)** tipe data void untuk mengeset **Kebisingan**

Deklarasi method **getBau()** tipe data **Int** untuk mengembalikan nilai **Bau**

Deklarasi method **setBau(int Bau)** tipe data void untuk mengeset **Bau**

Deklarasi method **getKebocoran()** tipe data **int** untuk mengembalikan nilai **Kebocoran**

Deklarasi method **setKebocoran(int Kebocoran)** tipe data void untuk mengeset **Kebocoran**

Deklarasi method **getKerusakan()** tipe data **int** untuk mengembalikan nilai **Kerusakan**

Deklarasi method **setKerusakan(int Kerusakan)** tipe data void untuk mengeset **Kerusakan**

Deklarasi method **getKeausan()** tipe data **Int** untuk mengembalikan nilai **Keausan**

Deklarasi method **setKeausan(int Keausan)** tipe data void untuk mengeset **Keausan**

KenyamananTest extends IndukClass

```
import java.io.FileWriter;  
import java.io.IOException;
```

```
public class KenyamananTest extends IndukClass{  
    KenyamananRuangan Kenyamanan = new KenyamananRuangan();  
    public static double Kenyamanan_Sesuai = 0.0;  
    public String Kebisingan,Bau,Kebocoran,Kerusakan,Keausan;
```

```
public void Input(String Kebisingan,String Bau,String Kebocoran,String Kerusakan,String  
Keausan) {  
    this.Kebisingan = Kebisingan;  
    this.Bau = Bau;  
    this.Kebocoran = Kebocoran;  
    this.Kerusakan = Kerusakan;  
    this.Keausan = Keausan;  
}
```

@Override

```
public double Analisis() {  
    try{  
        if("Tidak Bising".equals(Kebisingan)){  
            Kenyamanan_Sesuai++;  
        }  
        if("Tidak Bau".equals(Bau)){  
            Kenyamanan_Sesuai++;
```

```

    }
    if("Tidak Bocor".equals(Kebocoran)){
        Kenyamanan_Sesuai++;
    }
    if("Tidak Rusak".equals(Kerusakan)){
        Kenyamanan_Sesuai++;
    }
    if("Tidak Aus".equals(Keausan)){
        Kenyamanan_Sesuai++;
    }
} catch (Exception e){
    System.out.println("ERROR");
}
return Kenyamanan_Sesuai;
}

```

```

@Override
public double Sesuai() {
    return Kenyamanan_Sesuai/5;
}

```

Override

```

public void Save() {
    try{
        FileWriter Writer = new FileWriter("Kenyamanan.txt");
        Writer.write("Kebisingan Ruangan = " + Kebisingan+"\t");
        Writer.write("Bau Ruangan = " + Bau+"\t");
        Writer.write("Kebocoran Ruangan= " + Kebocoran+"\t");
        Writer.write("Kerusakan Ruangan= " + Kerusakan+"\t");
        Writer.write("Keausan Ruangan= " + Keausan+"\t");
        Writer.close();
    } catch (IOException ex){
        ex.printStackTrace();
    }
}
}

```

```

import java.io.FileWriter;
import java.io.IOException;

```

public class KenyamananTest extends IndukClass{ Mewariskan atau menurunkan method
method dari superclass IndukClass ke subclas KenyamananTest

 KenyamananRuangan Kenyamanan = new KenyamananRuangan(); Membuat
object kenyamanan dari class KenyamananRuangan

 public static double Kenyamanan_Sesuai = 0.0; Membuat variabel
Kenyamanan_Sesuai = 0.0

 public String Kebisingan,Bau,Kebocoran,Kerusakan,Keausan; Membuat variabel
String Kebisingan,Bau,Kebocoran,Kerusakan,Keausan

```
public void Input(String Kebisingan,String Bau,String Kebocoran,String Kerusakan,String
Keausan) {
    this.Kebisingan = Kebisingan;
    this.Bau = Bau;
    this.Kebocoran = Kebocoran;
    this.Kerusakan = Kerusakan;
    this.Keausan = Keausan;
}
```

Method Input yang mana didalamnya terdapat variabel String Kebisingan,String
Bau,String Kebocoran,String Kerusakan,String Keausan

 @Override

 public double Analisis() {

MengOverride method Analisis

```
try{
    if("Tidak Bising".equals(Kebisingan)){
        Kenyamanan_Sesuai++;
    }
}
```

Jika Tidak Bising equals Kebisingan maka Kenyamanan_Sesuai

```
if("Tidak Bau".equals(Bau)){
    Kenyamanan_Sesuai++;
}
```

Jika Tidak Bau equals Bau maka Kenyamanan_Sesuai

```
if("Tidak Bocor".equals(Kebocoran)){
    Kenyamanan_Sesuai++;
}
```

Jika Tidak Bocor equals Kebocoran maka Kenyamanan_Sesuai

```
if("Tidak Rusak".equals(Kerusakan)){
    Kenyamanan_Sesuai++;
}
```

Jika Tidak Rusak equals Kerusakan maka Kenyamanan_Sesuai

```

        if("Tidak Aus".equals(Keausan)){
            Kenyamanan_Sesuai++;
        }

```

Jika Tidak Aus equals Keausan maka Kenyamanan_Sesuai

```

        }catch(Exception e){
            System.out.println("ERROR");
        }
        return Kenyamanan_Sesuai;
    }

```

```

    @Override
    public double Sesuai() {
        return Kenyamanan_Sesuai/5;
    }

```

MengOverride method Sesuai dan memiliki nilai kembalian Kenyamanan_Sesuai/5

@Override

```

    public void Save() {
        try{
            FileWriter Writer = new FileWriter("Kenyamanan.txt");
            Writer.write("Kebisingan Ruangan = " + Kebisingan+"\t");
            Writer.write("Bau Ruangan = " + Bau+"\t");
            Writer.write("Kebocoran Ruangan= " + Kebocoran+"\t");
            Writer.write("Kerusakan Ruangan= " + Kerusakan+"\t");
            Writer.write("Keausan Ruangan= " + Keausan+"\t");
            Writer.close();
        }catch(IOException ex){
            ex.printStackTrace();
        }
    }

```

MengOverride method save yang didalamnya ada try catch nya untuk menampung method method yang di tulis

KeamananRuangKelas

```

private int kekokohan;
private int kunci;
private int bahaya;

```



```
public void setkekokohan(int
kekokohan)
public int getkekokohan()

public void setkunci(int kunci)
public int getkunci()

public void setbahaya(int bahaya)
public int getbahaya()
```

PrepCode :

Deklarasi **Private Int** untuk menampung Kondisi kekokohan diberi nama variable **kekokohan**

Deklarasi **Private int** untuk menampung kondisi kunci diberi nama variable **kunci**

Deklarasi **Private int** untuk menampung kondisi bahaya dari ruangan diberi nama variable **bahaya**

Deklarasi method **getkekokohan ()** tipedata **int** untuk mengembalikan nilai **kekokohan**

Deklarasi method **setkekokohan(int kekokohan)** tipe data void untuk mengeset **kekokohan**

Deklarasi method **getkunci()** tipedata **Int** untuk mengembalikan nilai **kunci**

Deklarasi method **setkunci(int kunci)** tipe data void untuk mengeset **kunci**

Deklarasi method **getbahaya()** tipedata **int** untuk mengembalikan nilai **bahaya**

Deklarasi method **setbahaya(int bahaya)** tipe data void untuk mengeset **bahaya**

KeamananTest extends IndukClass

```
import java.io.FileWriter;
import java.io.IOException;

public class KeamananTest extends IndukClass{
    KeamananRuangKelas KeamananRuang = new
KeamananRuangKelas();
    public static double Keamanan_Sesuai = 0.0;
```

```
public String Kekokohan, Kunci, Bahaya;
```

```
public void Input(String Kekokohan, String Kunci, String Bahaya){  
    this.Kekokohan = Kekokohan;  
    this.Kunci = Kunci;  
    this.Bahaya = Bahaya;  
}
```

```
public double Analisis(){  
    try{  
        switch (Kekokohan) {  
            case "Kokoh":  
                Keamanan_Sesuai++;  
                break;  
            case "Tidak Kokoh":  
                break;  
        }  
        switch (Kunci) {  
            case "Ada":  
                Keamanan_Sesuai++;  
                break;  
            case "Tidak Ada":  
                break;  
        }  
        switch (Bahaya) {  
            case "Aman":  
                Keamanan_Sesuai++;  
                break;  
            case "Tidak Aman":  
                break;  
        }  
    }catch(Exception e){  
        System.out.println("ERRORooo");  
    }  
    return Keamanan_Sesuai/3;  
}
```

```
@Override  
public double Sesuai() {  
    return Keamanan_Sesuai/3;  
}
```

@Override

```
public void Save() {  
    try{  
        FileWriter Writer = new FileWriter("Keamanan.txt");  
        Writer.write("Kekokohan Ruangan = " + this.Kekokohan+"\n");  
        Writer.write("Kunci Pintu dan Jendela = " + this.Kunci+"\n");  
        Writer.write("Keamanan Ruangan= " + this.Bahaya+"\n");  
        Writer.close();  
    }catch(IOException ex){  
        ex.printStackTrace();  
    }  
}
```

```
import java.io.FileWriter;  
import java.io.IOException;
```

```
public class KeamananTest extends IndukClass{  
    mewariskan method method yang ada di superclass Indukclass ke subclass KeamananTest  
    KeamananRuangKelas KeamananRuang = new KeamananRuangKelas();  
    Membuat object keamanan rauang dari class KeamananRuangKelas  
    public static double Keamanan_Sesuai = 0.0;  
    membuat variabel Keamanan_Sesuai = 0.0  
    public String Kekokohan, Kunci, Bahaya;  
    membuat variabel String Kekokohan, Kunci, Bahaya
```

```
public void Input(String Kekokohan, String Kunci, String Bahaya){  
    this.Kekokohan = Kekokohan;  
    this.Kunci = Kunci;  
    this.Bahaya = Bahaya;  
}
```

Method input didalamnya terdapat variabel String Kekokohan, String Kunci, String Bahaya

```
public double Analisis(){  
    mengOverride Analisis  
    try{  
        switch (Kekokohan) {  
            case "Kokoh":  
                Keamanan_Sesuai++;  
                break;  
            case "Tidak Kokoh":
```

```

        break;
    }
    switch (Kunci) {
    case "Ada":
        Keamanan_Sesuai++;
        break;
    case "Tidak Ada":
        break;
    }
    switch (Bahaya) {
    case "Aman":
        Keamanan_Sesuai++;
        break;
    case "Tidak Aman":
        break;
    }
    }catch(Exception e){
    System.out.println("ERRORooo");
    }
    return Keamanan_Sesuai/3;
    }

```

Memiliki nilai kembalian Keamanan_Sesuai

```

@Override
public double Sesuai() {
    return Keamanan_Sesuai/3;
}

```

MengOverride method Keamanan_Sesuai/3

@Override

```

public void Save() {
    try{
        FileWriter Writer = new FileWriter("Keamanan.txt");
        Writer.write("Kekokohan Ruangan = " + this.Kekokohan+"\t");
        Writer.write("Kunci Pintu dan Jendela = " + this.Kunci+"\t");
        Writer.write("Keamanan Ruangan= " + this.Bahaya+"\t");
        Writer.close();
    }catch(IOException ex){
        ex.printStackTrace();
    }
}

```

mengoverride method save yang didalamnya terdapat try catch untuk menyimpan inputan user

Login

```
public class Login {
    public String NamaCreat;
    public String PasswordCreat;
    public String Passwordconf;
    public String NamaLogin;
    public String PasswordLogin;
    public boolean Next;

    public void SetMater(String Nama, String Password, String ConfPass){
        Next = false;
        this.NamaCreat = Nama;
        this.PasswordCreat = Password;
        this.Passwordconf = ConfPass;
        try{
            if(this.NamaCreat != null && this.PasswordCreat != null &&
                this.Passwordconf != null && this.PasswordCreat.equals(Passwordconf)){
                Next = true;
            }
        }catch(Exception e){
            System.out.println("ERROR");
        }
    }

    public void SetLogin(String Nama, String Password) {
        Next = false;
        this.NamaLogin = Nama;
        this.PasswordLogin = Password;
        try{
            if(this.NamaLogin!= null && this.PasswordLogin != null
                &&this.NamaLogin.equals(NamaCreat) && this.PasswordLogin.equals(PasswordCreat)){
                Next = true;
            }
        }catch(Exception e){
            System.out.println("ERROR");
        }
    }
}
```

```
public class LogIn {  
    Class log in  
    public String NamaCreat;  
    deklarasi variabel String NamaCreat  
    public String PasswordCreat;  
    deklarasi variabel String PasswordCreat  
    public String Passwordconf;  
    deklarasi variabel Passwordconf  
    public String NamaLogin;  
    mendeklarasi variabel NamaLogin  
    public String PasswordLogin;  
    mendeklarasi variabel PasswordLogin  
    public boolean Next;  
    mendeklarasi variabel Next
```

```
    public void SetMater(String Nama, String Password, String ConfPass){  
        Next = false;  
        this.NamaCreat = Nama;  
        this.PasswordCreat = Password;  
        this.Passwordconf = ConfPass;  
        try{  
            if(this.NamaCreat != null && this.PasswordCreat != null &&  
                this.Passwordconf != null && this.PasswordCreat.equals(Passwordconf)){  
                Next = true;  
            }  
        }catch(Exception e){  
            System.out.println("ERROR");  
        }  
    }  
  
    public void SetLogin(String Nama, String Password) {  
        Next = false;  
        this.NamaLogin = Nama;  
        this.PasswordLogin = Password;  
        try{  
            if(this.NamaLogin!= null && this.PasswordLogin != null  
                &&this.NamaLogin.equals(NamaCreat) && this.PasswordLogin.equals(PasswordCreat)){  
                Next = true;
```

```
}  
}catch(Exception e){  
System.out.println("ERROR");  
}
```

AllTest

```
public class AllTest {  
    public static double rata_rata;  
  
    public static double Rata_rata (double Kondisi,double Sarana, double sarana2,  
    double Lingkungan, double Kebersihan, double Kenyamanan, double Keamanan){  
        rata_rata = (Kondisi +Sarana +sarana2 +Lingkungan + Kebersihan + Kenyamanan +  
        Keamanan)/7*100;  
        System.out.println("Jumlah rata_rata Nilai Sesuai = " +rata_rata +" %");  
        return rata_rata;  
  
    public static void main(String[] args){  
        TampilanAwal Awal = new TampilanAwal();  
        Awal.setVisible(true);  
        IndukClass kondisi = new KondisiRuangtest();  
        IndukClass SaranaRuang = new SaranaTest ();  
        IndukClass SaranaRuang2 = new SaranaTest2();  
        IndukClass lingkungan = new LingkunganTest();  
        IndukClass kebersihan = new KebersihanTest();  
        IndukClass kenyamanan = new KenyamananTest();  
        IndukClass keamanan = new KeamananTest();  
  
        }  
}
```

```
public class AllTest {  
    public static double rata_rata;  
    method rata rata  
  
    public static double Rata_rata (double Kondisi,double Sarana, double sarana2,
```

```
double Lingkungan, double Kebersihan, double Kenyamanan, double Keamanan){  
rata_rata = (Kondisi +Sarana +sarana2 +Lingkungan + Kebersihan + Kenyamanan +  
Keamanan)/7*100;
```

method menghitung rata rata dari Kondisi +Sarana +sarana2 +Lingkungan + Kebersihan +
Kenyamanan + Keamanan)/7*100;

```
System.out.println("Jumlah rata_rata Nilai Sesuai = " +rata_rata +" %");
```

Menampilkan jumlah atau hasil sesuai di tambah rata rata

```
return rata_rata;
```

memberi nilai kembalian rata rata

```
public static void main(String[] args){
```

```
TampilanAwal Awal = new TampilanAwal();
```

Membuat object awal dari TampilanAwal

```
Awal.setVisible(true);
```

Awal dot memberikan nilai Visible (benar)

```
IndukClass kondisi = new KondisiRuangtest();
```

Membuat object kondisi dari KondisiRuangtest

```
IndukClass SaranaRuang = new SaranaTest ();
```

Membuat object SaranaRuang

```
IndukClass SaranaRuang2 = new SaranaTest2();
```

Membuat object SaranaRuang2

```
IndukClass lingkungan = new LingkunganTest();
```

Membuat object Lingkungan

```
IndukClass kebersihan = new KebersihanTest();
```

Membuat object kebersihan

```
IndukClass kenyamanan = new KenyamananTest();
```

Membuat object kenyamanan

```
IndukClass keamanan = new KeamananTest();
```

Membuat object keamanan

```
}
```

```
}
```

IndukClass

```
import java.util.Scanner;
```



```

public abstract class IndukClass {
    Scanner input = new
Scanner(System.in);
    public abstract double Analisis();
    public abstract void Save();
    public abstract double Sesuai();
}

```

```

import java.util.Scanner;

```

```

public abstract class IndukClass {
ini adalah class superclass yang mana akan menurunkan method method nya ke subclass
subclass yang ada
    Scanner input = new Scanner(System.in);
Membuat object input dari Scanner
    public abstract double Analisis();
method Analisis
    public abstract void Save();
method Save
    public abstract double Sesuai();
method Sesuai
}

```

ClassInterfac

```

public interface ClassInterface {

    public double luas();
    public String bentuk();
    public double rasio();
}

```

```

public interface ClassInterface {

```

ini class interface yang mana memiliki method menghitung dan dari method method ini lalau di turunkan ke class kondisikelas

```
public double luas();  
method menghitung luas  
public String bentuk();  
method menghitung bentuk  
public double rasio();  
method menghitung rasio  
}
```

ClassTampilanAwal GUI

```
TampilanIdentitas TID = new TampilanIdentitas();  
TampilanLihatData LData = new TampilanLihatData();  
boolean menu = true;  
public TampilanAwal() {  
    initComponents();  
    Menu.setVisible(false);  
    Profil.setVisible(false);  
    SetAkun.setVisible(false);  
}
```

```
private void TombolMenuMouseClicked(java.awt.event.MouseEvent evt) {  
    if(menu == true){  
        Menu.setVisible(true);  
        menu = false;  
    }else if(menu == false){  
        Menu.setVisible(false);  
        menu = true;  
    }  
}
```

```
Login login = new Login();  
boolean AkunLogin = false;
```

```
private void LoginButtonActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    if(Username.getText() != "" || Password.getText() != ""){
```

```

login.SetLogin(Username.getText(), Password.getText());
if(login.Next == true){
    Login.setVisible(false);
    AkunLogin = true;
}
}
User.setText(Username.getText());
Pass.setText(Password.getText());
}

private void CreatAkunActionPerformed(java.awt.event.ActionEvent evt) {

    SetAkun.setVisible(true);
    Login.setVisible(false);
}

private void CreatAkunButtonActionPerformed(java.awt.event.ActionEvent evt) {
    if(SetUsername.getText() != "" && SetPassword.getText() != "" &&
ConfirmPassword.getText() != ""){
        login.SetMater(SetUsername.getText(), SetPassword.getText(),
ConfirmPassword.getText());
        if(login.Next == true){
            Login.setVisible(true);
            SetAkun.setVisible(false);
        }
    }
}

private void TanggalActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:
}

private void InputDataActionPerformed(java.awt.event.ActionEvent evt) {

    TID.Input();
    Menu.setVisible(false);
    TombolMenu.setVisible(true);
}

```

```
private void LihatDataActionPerformed(java.awt.event.ActionEvent evt) {

    LData.Data();
    Menu.setVisible(false);
    TombolMenu.setVisible(true);
}

private void AkunActionPerformed(java.awt.event.ActionEvent evt) {

    Profil.setVisible(true);
    Menu.setVisible(false);
}

private void OKButtonActionPerformed(java.awt.event.ActionEvent evt) {

    Profil.setVisible(false);
    TombolMenu.setVisible(true);
}

private void LogoutMouseClicked(java.awt.event.MouseEvent evt) {

    Username.setText("");
    Password.setText("");
    Login.setVisible(true);
    Menu.setVisible(false);
    Profil.setVisible(false);
    SetAkun.setVisible(false);
}

private void MinimazeActionPerformed(java.awt.event.ActionEvent evt) {

    setState(ICONIFIED);
}

private void ExitActionPerformed(java.awt.event.ActionEvent evt) {

    System.exit(0);
}

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.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

            java.util.logging.Logger.getLogger(TampilanAwal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

            java.util.logging.Logger.getLogger(TampilanAwal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

            java.util.logging.Logger.getLogger(TampilanAwal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

            java.util.logging.Logger.getLogger(TampilanAwal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new TampilanAwal().setVisible(true);
            }
        });
    }
}

```

```

TampilanIdentitas TID = new TampilanIdentitas();
Membuat object TID dari TampilanIdentitas
    TampilanLihatData LData = new TampilanLihatData();
Membuat object Ldata dari TampilanLihatData
    boolean menu = true;
boolean menu samadengan benar

```

```

    public TampilanAwal() {
method tampilan Awal
        initComponents();
        Menu.setVisible(false);
Menu memberikan nilai Visible (benar)
        Profil.setVisible(false);
Profil memberikan nilai Visible (salah)
        SetAkun.setVisible(false);
Memberikan nilai akun dan memberikan nilai Visible
    }

```

```

private void TombolMenuMouseClicked(java.awt.event.MouseEvent evt) {
method TombolMenuMouseClicked
    if(menu == true){
        Menu.setVisible(true);
        menu = false;
jika menu samadengan ssamadengan benar
menu dot memberikan nilai Visible (benar)
menu samadengan salah
    }else if(menu == false){
        Menu.setVisible(false);
        menu = true;
jika maka menu samadengan samadengan salah
menu memberikan nilai Visible (salah)
menu samadengan salah
    }
    }
    Login login = new Login();
Membuat object login dari Login
    boolean AkunLogin = false;
boolean AkunLogin samadengan salah
    private void LoginButtonActionPerformed(java.awt.event.ActionEvent evt) {
method LoginButtonAction
        if(Username.getText() != "" || Password.getText() != ""){
            login.SetLogin(Username.getText(), Password.getText());
            if(login.Next == true){
                Login.setVisible(false);
                AkunLogin = true;
            }
        }
    }
    Jika Username menampilkan Text tidak sama dengan || Password menampilkan text
    Login memberikan nilai Login (Username mebampilkan text , Password menampilkan text

```

Jika login dot Next samadengan samadengan benar

Login memberikan nilai Visible (salah)

Akun login samadengan benar

```
}
    User.setText(Username.getText());
    Pass.setText>Password.getText());
}
```

User memberikan nilai Text (Username.getText())

Pass memberikan nilai Text (Password.getText());

```
private void CreatAkunActionPerformed(java.awt.event.ActionEvent evt) {
    SetAkun.setVisible(true);
    Login.setVisible(false);
}
```

Method CreatAkunActionPerformed

```
private void CreatAkunButtonActionPerformed(java.awt.event.ActionEvent evt) {
```

method CreatAkunButtonActionPerformed

```
    if(SetUsername.getText() != "" && SetPassword.getText() != "" &&
    ConfirmPassword.getText() != ""){
```

```
        login.SetMater(SetUsername.getText(), SetPassword.getText(),
    ConfirmPassword.getText());
```

```
        if(login.Next == true){
            Login.setVisible(true);
            SetAkun.setVisible(false);
        }
    }
```

```
}
```

```
private void TanggalActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    // TODO add your handling code here:
```

Method TanggalActionPerformed

```
}
```

```
private void InputDataActionPerformed(java.awt.event.ActionEvent evt) {
```

method InputDataActionPerformed

```
    TID.Input();
```

```

        Menu.setVisible(false);
        TombolMenu.setVisible(true);
    }

    private void LihatDataActionPerformed(java.awt.event.ActionEvent evt) {
method LihatDataActionPerformed
        LData.Data();
        Menu.setVisible(false);
        TombolMenu.setVisible(true);
    }

    private void AkunActionPerformed(java.awt.event.ActionEvent evt) {
method AkunActionPerformed
        Profil.setVisible(true);
        Menu.setVisible(false);
    }

    private void OKButtonActionPerformed(java.awt.event.ActionEvent evt) {
method OKButtonActionPerformed
        Profil.setVisible(false);
        TombolMenu.setVisible(true);
    }

    private void LogoutMouseClicked(java.awt.event.MouseEvent evt) {
method LogoutMouseClicked
        Username.setText("");
        Password.setText("");
        Login.setVisible(true);
        Menu.setVisible(false);
        Profil.setVisible(false);
        SetAkun.setVisible(false);
    }

    private void MinimazeActionPerformed(java.awt.event.ActionEvent evt) {
method MinimazeActionPerformed gunanya untuk me Minimaze tampilan gui
        setState(ICONIFIED);
    }

    private void ExitActionPerformed(java.awt.event.ActionEvent evt) {
method ExitActionPerformed gunanya untuk me mengExit tampilan gui
        System.exit(0);
    }

```



```

    }

    public static void main(String[] args) {
method main(

        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(TampilanAwal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(TampilanAwal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(TampilanAwal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(TampilanAwal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new TampilanAwal().setVisible(true);
            }
        });
    }
}

```

```
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.table.DefaultTableModel;
```

```
public class TampilanIdentitas extends javax.swing.JFrame {
```

```
    IdentitasRuangkelas SetID = new IdentitasRuangkelas();
    IdentitasRuangkelastest ID = new IdentitasRuangkelastest();
    KondisiRuang SetKondisi = new KondisiRuang();
    KondisiRuangtest Kondisi = new KondisiRuangtest();
    TampilanSarana Sarana = new TampilanSarana();
```

```
    public TampilanIdentitas() {
        initComponents();
        Identitas.setVisible(true);
        TombolIdentitas.setVisible(false);
        Tabel.setVisible(false);
    }
```

```
    private void NextIdentitasActionPerformed(java.awt.event.ActionEvent evt) {
        try{
            SetID.setNamaruang(TNamaRuangan.getText());
            SetID.setLokasiruang(CLokasiRuang.getSelectedItem().toString());
            SetID.setProgramstudi(CprogramStudi.getSelectedItem().toString());
            ID.Input(SetID.getNamaruang(), SetID.getLokasiruang(), SetID.getProgramstudi());

            SetKondisi.setPanjang(Double.parseDouble(TPanjangRuang.getText()));
            SetKondisi.setLebar(Double.parseDouble(TLebarRuang.getText()));
            SetKondisi.setJumlahkursi(Integer.parseInt(TJumlahKursi.getText()));
            SetKondisi.setJumlahpintu(Integer.parseInt(TJumlahPintu.getText()));
            SetKondisi.setJumlahjendela(Integer.parseInt(TJumlahJendela.getText()));
            Kondisi.Input(SetKondisi.getPanjang(), SetKondisi.getLebar(),
                SetKondisi.getJumlahkursi(),
            SetKondisi.getJumlahpintu(),SetKondisi.getJumlahjendela());
            Kondisi.luas();
            Kondisi.bentuk();
            Kondisi.rasio();

            Kondisi.Analisis();
            Kondisi.Sesuai();
```

```

        System.out.println(Kondisi.KondisiRuang_Sesuai);
        System.out.println(SetID.getNamaruang());
        System.out.println(SetID.getLokasiruang());
        System.out.println(SetID.getProgramstudi());

        DefaultTableModel modelID = (DefaultTableModel) TableIdentitas.getModel();
        modelID.addRow(new
Object[]{SetID.getNamaruang(),SetID.getLokasiruang(),SetID.getProgramstudi(),
        SetKondisi.getPanjang(),SetKondisi.getLebar(),SetKondisi.getJumlahkursi(),
SetKondisi.getJumlahpintu(),SetKondisi.getJumlahjendela(),Kondisi.Luas,Kondisi.Rasio,Ko
ndisi.bentuk});

        Tabel.setVisible(true);
        TombolIdentitas.setVisible(true);
        Identitas.setVisible(false);
    }catch(Exception e){
        System.out.println("ERROR");
    }
}

private void SJumlahKursiStateChanged(javax.swing.event.ChangeEvent evt) {

    TJumlahKursi.setText(String.valueOf(SJumlahKursi.getValue()));
}

private void SJumlahPintuStateChanged(javax.swing.event.ChangeEvent evt) {

    TJumlahPintu.setText(String.valueOf(SJumlahPintu.getValue()));
}

private void SJumlahJendelaStateChanged(javax.swing.event.ChangeEvent evt) {

    TJumlahJendela.setText(String.valueOf(SJumlahJendela.getValue()));
}

private void SaveActionPerformed(java.awt.event.ActionEvent evt) {
Sarana.Sarana();
    ID.Save();
    Kondisi.Save();
}

```

```

TNamaRuangan.setText("");
CLokasiRuang.setSelectedIndex(0);
CprogramStudi.setSelectedIndex(0);
TPanjangRuang.setText("");
TLebarRuang.setText("");
TJumlahKursi.setText("0");
TJumlahPintu.setText("0");
TJumlahJendela.setText("0");
this.setVisible(false);
Identitas.setVisible(true);
Tabel.setVisible(false);
TombolIdentitas.setVisible(false);
}

```

```

import java.io.FileWriter;
import java.io.IOException;
import javax.swing.table.DefaultTableModel;

public class TampilanIdentitas extends javax.swing.JFrame {
method TampilanIdentitas menurunkan atau mewariskan dari JFrame

    IdentitasRuangkelas SetID = new IdentitasRuangkelas();
Membuat object SetID dari IdentitasRuangkelas
    IdentitasRuangkelastest ID = new IdentitasRuangkelastest();
Membuat object ID dari IdentitasRuangkelastest
    KondisiRuang SetKondisi = new KondisiRuang();
Membuat object SetKondisi dari KondisiRuang
    KondisiRuangtest Kondisi = new KondisiRuangtest();
Membuat object Kondisi dari KondisiRuangtest
    TampilanSarana Sarana = new TampilanSarana();
Membuat object Sarana dari TampilanSarana

    public TampilanIdentitas() {
membuat Object TampilanIdentitas
        initComponents();
        Identitas.setVisible(true);

```

```
TombolIdentitas.setVisible(false);
Tabel.setVisible(false);
}
```

```
private void NextIdentitasActionPerformed(java.awt.event.ActionEvent evt) {
method NextIdentitasActionPerformed
    try{
        SetID.setNamaruang(TNamaRuangan.getText());
        SetID.setLokasiruang(CLokasiRuangan.getSelectedItem().toString());
        SetID.setProgramstudi(CprogramStudi.getSelectedItem().toString());
        ID.Input(SetID.getNamaruang(), SetID.getLokasiruang(), SetID.getProgramstudi());

        SetKondisi.setPanjang(Double.parseDouble(TPanjangRuangan.getText()));
        SetKondisi.setLebar(Double.parseDouble(TLebarRuangan.getText()));
        SetKondisi.setJumlahkursi(Integer.parseInt(TJumlahKursi.getText()));
        SetKondisi.setJumlahpintu(Integer.parseInt(TJumlahPintu.getText()));
        SetKondisi.setJumlahjendela(Integer.parseInt(TJumlahJendela.getText()));
        Kondisi.Input(SetKondisi.getPanjang(), SetKondisi.getLebar(),
            SetKondisi.getJumlahkursi(),
SetKondisi.getJumlahpintu(),SetKondisi.getJumlahjendela());
        Kondisi.luas();
        Kondisi.bentuk();
        Kondisi.rasio();

        Kondisi.Analisis();
        Kondisi.Sesuai();
        System.out.println(Kondisi.KondisiRuang_Sesuai);
        System.out.println(SetID.getNamaruang());
        System.out.println(SetID.getLokasiruang());
        System.out.println(SetID.getProgramstudi());

        DefaultTableModel modelID = (DefaultTableModel) TableIdentitas.getModel();
        modelID.addRow(new
Object[] {SetID.getNamaruang(),SetID.getLokasiruang(),SetID.getProgramstudi(),
            SetKondisi.getPanjang(),SetKondisi.getLebar(),SetKondisi.getJumlahkursi(),
SetKondisi.getJumlahpintu(),SetKondisi.getJumlahjendela(),Kondisi.Luas,Kondisi.Rasio,Kon
disi.bentuk});

        Tabel.setVisible(true);
    }
}
```

```

        TombolIdentitas.setVisible(true);
        Identitas.setVisible(false);
    }catch(Exception e){
        System.out.println("ERROR");
    }
}

private void SJumlahKursiStateChanged(javax.swing.event.ChangeEvent evt) {
method SJumlahKursiStateChanged
    TJumlahKursi.setText(String.valueOf(SJumlahKursi.getValue()));
}

private void SJumlahPintuStateChanged(javax.swing.event.ChangeEvent evt) {
method SJumlahPintuStateChanged
    TJumlahPintu.setText(String.valueOf(SJumlahPintu.getValue()));
}

private void SJumlahJendelaStateChanged(javax.swing.event.ChangeEvent evt) {
method SJumlahJendelaStateChanged
    TJumlahJendela.setText(String.valueOf(SJumlahJendela.getValue()));
}

private void SaveActionPerformed(java.awt.event.ActionEvent evt) {
method SaveActionPerformed
Sarana.Sarana();
    ID.Save();
    Kondisi.Save();
    TNamaRuangan.setText("");
    CLokasiRuang.setSelectedIndex(0);
    CprogramStudi.setSelectedIndex(0);
    TPanjangRuang.setText("");
    TLebarRuang.setText("");
    TJumlahKursi.setText("0");
    TJumlahPintu.setText("0");
    TJumlahJendela.setText("0");
    this.setVisible(false);
    Identitas.setVisible(true);
    Tabel.setVisible(false);
    TombolIdentitas.setVisible(false);
}

```

Tampilan Keadaan

```
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class TampilanKeadaan extends javax.swing.JFrame {

    KeamananRuangKelas SetKeamanan = new KeamananRuangKelas() ;
    KeamananTest Keamanan = new KeamananTest();
    KebersihanRuangKelas SetKebersihan = new KebersihanRuangKelas();
    KebersihanTest Kebersihan = new KebersihanTest();
    KenyamananRuangKelas SetKenyamanan = new KenyamananRuangKelas();
    KenyamananTest Kenyamanan = new KenyamananTest();
    LingkunganRuangKelas SetLingkungan = new LingkunganRuangKelas();
    LingkunganTest Lingkungan = new LingkunganTest();

    public TampilanKeadaan() {
        initComponents();
        Keadaan.setVisible(true);
        TombolKeadaan.setVisible(false);
        Tabel.setVisible(false);
    }

    private void LantaiBersihActionPerformed(java.awt.event.ActionEvent evt) {

        if(LantaiBersih.isSelected()){
            LantaiKotor.setSelected(false);
            SetLingkungan.setKondisiLantai("Bersih");
        }
    }

    private void LantaiKotorActionPerformed(java.awt.event.ActionEvent evt) {

        if(LantaiKotor.isSelected()){
            LantaiBersih.setSelected(false);
            SetLingkungan.setKondisiLantai("Kotor");
        }
    }

    private void DindingBersihActionPerformed(java.awt.event.ActionEvent evt) {
```

```

        if(DindingBersih.isSelected()){
            DindingKotor.setSelected(false);
            SetLingkungan.setKondisiDinding("Bersih");
        }
    }

    private void DindingKotorActionPerformed(java.awt.event.ActionEvent evt) {

        if(DindingKotor.isSelected()){
            DindingBersih.setSelected(false);
            SetLingkungan.setKondisiDinding("Kotor");
        }
    }

    private void AtapBersihActionPerformed(java.awt.event.ActionEvent evt) {

        if(AtapBersih.isSelected()){
            AtapKotor.setSelected(false);
            SetLingkungan.setKondisiAtap("Bersih");
        }
    }

    private void AtapKotorActionPerformed(java.awt.event.ActionEvent evt) {

        if(AtapKotor.isSelected()){
            AtapBersih.setSelected(false);
            SetLingkungan.setKondisiAtap("Kotor");
        }
    }

    private void PintuBersihActionPerformed(java.awt.event.ActionEvent evt) {

        if(PintuBersih.isSelected()){
            PintuKotor.setSelected(false);
            SetLingkungan.setKondisiPintu("Bersih");
        }
    }

    private void PintuKotorActionPerformed(java.awt.event.ActionEvent evt) {

        if(PintuKotor.isSelected()){

```



```

        PintuBersih.setSelected(false);
        SetLingkungan.setKondisiPintu("Kotor");
    }
}

private void JendelaBersihActionPerformed(java.awt.event.ActionEvent evt) {
    if(JendelaBersih.isSelected()){
        JendelaKotor.setSelected(false);
        SetLingkungan.setKondisiJendela("Bersih");
    }
}

private void JendelaKotorActionPerformed(java.awt.event.ActionEvent evt) {

    if(JendelaKotor.isSelected()){
        JendelaBersih.setSelected(false);
        SetLingkungan.setKondisiJendela("Kotor");
    }
}

private void LancarActionPerformed(java.awt.event.ActionEvent evt) {

    if(Lancar.isSelected()){
        TidakLancar.setSelected(false);
        SetKebersihan.setSirkulasiUdara("Lancar");
    }
}

private void TidakLancarActionPerformed(java.awt.event.ActionEvent evt) {

    if(TidakLancar.isSelected()){
        Lancar.setSelected(false);
        SetKebersihan.setSirkulasiUdara("Tidak Lancar");
    }
}

private void JPencapaianStateChanged(javax.swing.event.ChangeEvent evt) {

    Pencapaian.setText(String.valueOf(JPencapaian.getValue()));
}

```

```
private void JKelembapanStateChanged(javax.swing.event.ChangeEvent evt) {  
  
Kelembapan.setText(String.valueOf(JKelembapan.getValue()));  
}  
  
private void JSuhuStateChanged(javax.swing.event.ChangeEvent evt) {  
  
Suhu.setText(String.valueOf(JSuhu.getValue()));  
}  
  
private void TidakBisingActionPerformed(java.awt.event.ActionEvent evt) {  
  
if(TidakBising.isSelected()){  
    Bising.setSelected(false);  
    SetKenyamanan.setKebisingan("Tidak Bising");  
}  
}  
  
private void BisingActionPerformed(java.awt.event.ActionEvent evt) {  
  
if(Bising.isSelected()){  
    TidakBising.setSelected(false);  
    SetKenyamanan.setKebisingan("Bising");  
}  
}  
  
private void TidakBauActionPerformed(java.awt.event.ActionEvent evt) {  
  
if(TidakBau.isSelected()){  
    Bau.setSelected(false);  
    SetKenyamanan.setBau("Tidak Bau");  
}  
}  
  
private void BauActionPerformed(java.awt.event.ActionEvent evt) {  
  
if(Bau.isSelected()){  
    TidakBau.setSelected(false);  
    SetKenyamanan.setBau("Bau");  
}  
}
```

```

private void TidakBocorActionPerformed(java.awt.event.ActionEvent evt) {

    if(TidakBocor.isSelected()){
        Bocor.setSelected(false);
        SetKenyamanan.setKebocoran("Tidak Bocor");
    }
}

private void BocorActionPerformed(java.awt.event.ActionEvent evt) {
    if(Bocor.isSelected()){
        TidakBocor.setSelected(false);
        SetKenyamanan.setKebocoran("Bocor");
    }
}

```

```

import java.io.FileWriter;
import java.io.IOException;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class TampilanKeadaan extends javax.swing.JFrame {
    method TampilanKeadaan mewariskan dari JFrame

        KeamananRuangKelas SetKeamanan = new KeamananRuangKelas() ;
    Membuat object SetKeamanan dari KeamananRuangKelas
        KeamananTest Keamanan = new KeamananTest();
    Membuat object SetKeamanan dari KeamananTest
        KebersihanRuangKelas SetKebersihan = new KebersihanRuangKelas();
    Membuat object dari KebersihanRuangKelas
        KebersihanTest Kebersihan = new KebersihanTest();
    Membuat object Kebersihan dari KebersihanTest
        KenyamananRuangan SetKenyamanan = new KenyamananRuangan();
    Membuat object SetKenyamanan dari KenyamananRuangan
        KenyamananTest Kenyamanan = new KenyamananTest();
    Membuat object Kenyamanan dari KenyamananTest
        LingkunganRuangKelas SetLingkungan = new LingkunganRuangKelas();

```

Membuat object SetLingkungan dari LingkunganRuangKelas
LingkunganTest Lingkungan = new LingkunganTest();
Membuat object Lingkungan dari LingkunganTest

TampilanLihatData

```
public class TampilanLihatData extends javax.swing.JFrame {

    TampilanIdentitas Ide = new TampilanIdentitas();
    TampilanSarana Sar = new TampilanSarana();
    TampilanSarana2 Sar2 = new TampilanSarana2();
    TampilanKeadaan Kea = new TampilanKeadaan();

    public TampilanLihatData() {
        initComponents();
        TabelSarana2.setVisible(false);
        TabelSarana.setVisible(false);
        TabelID.setVisible(false);
        TabelKeadaan.setVisible(false);
    }

    private void ExitActionPerformed(java.awt.event.ActionEvent evt) {
        System.exit(0);
    }

    private void RSaranaActionPerformed(java.awt.event.ActionEvent evt) {

        if(RSarana.isSelected()){
            TabelSarana.setVisible(true);
            TabelSarana2.setVisible(false);
            TabelID.setVisible(false);
            TabelKeadaan.setVisible(false);

            RID.setSelected(false);
            RSarana2.setSelected(false);
            RKeadaan.setSelected(false);
        }
    }
}
```

```
private void RSarana2ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    if(RSarana2.isSelected()){  
        TabelSarana2.setVisible(true);  
        TabelSarana.setVisible(false);  
        TabelID.setVisible(false);  
        TabelKeadaan.setVisible(false);
```

```
        RID.setSelected(false);  
        RSarana.setSelected(false);  
        RKeadaan.setSelected(false);  
    }  
}
```

```
private void RKeadaanActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    if(RKeadaan.isSelected()){  
        TabelKeadaan.setVisible(true);  
        TabelSarana.setVisible(false);  
        TabelSarana2.setVisible(false);  
        TabelID.setVisible(false);
```

```
        RID.setSelected(false);  
        RSarana.setSelected(false);  
        RSarana2.setSelected(false);  
    }  
}
```

```
private void RIDActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    if(RID.isSelected()){  
        TabelID.setVisible(true);  
        TabelSarana.setVisible(false);  
        TabelSarana2.setVisible(false);  
        TabelKeadaan.setVisible(false);
```

```
        RSarana2.setSelected(false);  
        RSarana.setSelected(false);  
        RKeadaan.setSelected(false);  
    }
```

```
}  
  
public void Data() {
```

```
public class TampilanLihatData extends javax.swing.JFrame {  
method TampilanLihatData di wariskan oleh javax.swing.JFrame
```

```
    TampilanIdentitas Ide = new TampilanIdentitas();
```

Membuat object Ide dari TampilanIdentitas

```
    TampilanSarana Sar = new TampilanSarana();
```

Membuat object Sar dari TampilanSarana

```
    TampilanSarana2 Sar2 = new TampilanSarana2();
```

Membuat object Sar2 dari TampilanSarana2

```
    TampilanKeadaan Kea = new TampilanKeadaan();
```

Membuat object Kea dari TampilanKeadaan

```
    public TampilanLihatData() {  
        initComponents();  
        TabelSarana2.setVisible(false);  
        TabelSarana.setVisible(false);  
        TabelID.setVisible(false);  
        TabelKeadaan.setVisible(false);  
    }
```

TampilanSarana

```
import java.io.IOException;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import javax.swing.table.DefaultTableModel;  
  
public class TampilanSarana extends javax.swing.JFrame {  
  
    SaranaTest SaranaTest = new SaranaTest();  
    SaranaRuangan SetSarana = new SaranaRuangan();  
    TampilanSarana2 Sarana2 = new TampilanSarana2();
```

```
public TampilanSarana() {  
    initComponents();  
    Sarana.setVisible(true);  
    TombolSarana.setVisible(false);  
    Tabel.setVisible(false);  
}
```

```
private void NextSarana1ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try{  
        SetSarana.setJumlahSteker(Integer.parseInt(TJumlahSteker.getText()));  
        SetSarana.setKondisiSteker(Integer.parseInt(TKondisiStekerB.getText()));  
        SetSarana.setKondisiStekerR(Integer.parseInt(TKondisiStekerR.getText()));  
        SetSarana.setPosisiStekerDD(Integer.parseInt(TPosisiStekerDD.getText()));  
        SetSarana.setPosisiStekerPR(Integer.parseInt(TPosisiStekerPJ.getText()));  
  
        SetSarana.setJumlahLampu(Integer.parseInt(TJumlahLampu.getText()));  
        SetSarana.setKondisiLampu(Integer.parseInt(TKondisiLampuB.getText()));  
        SetSarana.setKondisiLampuR(Integer.parseInt(TKondisiLampuR.getText()));  
        SetSarana.setPosisiLampuL(Integer.parseInt(TPosisiLampuL.getText()));  
  
        SetSarana.setJumlahKabelLCD(Integer.parseInt(TJumlahLCD.getText()));  
        SetSarana.setKondisiKabelLCD(Integer.parseInt(TKondisiLCDB.getText()));  
        SetSarana.setKondisiKabelLCDR(Integer.parseInt(TKondisiLCDR.getText()));  
        SetSarana.setPosisiKabelLCDDD(Integer.parseInt(TPosisiLCDDD.getText()));  
  
        SaranaTest.input(SetSarana.getJumlahSteker(), SetSarana.getKondisiSteker(),  
        SetSarana.getKondisiStekerR(),  
        SetSarana.getPosisiStekerDD(),SetSarana.getPosisiStekerPR(),  
        SetSarana.getJumlahLampu(), SetSarana.getKondisiLampu(),  
        SetSarana.getKondisiLampuR(),SetSarana.getPosisiLampuL(),  
        SetSarana.getJumlahKabelLCD(), SetSarana.getKondisiKabelLCD(),  
        SetSarana.getKondisiKabelLCDR(), SetSarana.getPosisiKabelLCDDD());  
        SaranaTest.Analisis();  
        SaranaTest.Sesuai();  
  
        DefaultTableModel modelSteker = (DefaultTableModel) TableSteker.getModel();  
        modelSteker.addRow(new  
        Object[]{SetSarana.getJumlahSteker(),SetSarana.getKondisiSteker(),SetSarana.getKondisi  
        StekerR(),
```

```

SetSarana.getPosisiStekerDD(),SetSarana.getPosisiStekerPR(),SetSarana.getJumlahKabelLCD(),SetSarana.getKondisiKabelLCD(),
    SetSarana.getKondisiKabelLCDDR(),SetSarana.getPosisiKabelLCDDD());});

    DefaultTableModel modelLampu = (DefaultTableModel) TableLampu.getModel();
    modelLampu.addRow(new
Object[] {SetSarana.getJumlahSteker(),SetSarana.getKondisiSteker(),SetSarana.getKondisi
StekerR(),
    SetSarana.getPosisiStekerDD()});
Tabel.setVisible(true);
Sarana.setVisible(false);
TombolSarana.setVisible(true);

} catch (Exception e) {
System.out.println("ERROR");
}
}

```

```

import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.table.DefaultTableModel;

public class TampilanSarana extends javax.swing.JFrame {
method TampilanSarana diwariskan dari javax.swing.JFrame
    SaranaTest SaranaTest = new SaranaTest();
Membuat object SaranaTest dari SaranaTest
    SaranaRuangan SetSarana = new SaranaRuangan();
Membuat object SetSarana dari SaranaRuangan
    TampilanSarana2 Sarana2 = new TampilanSarana2();
Membuat object Sarana2 dari TampilanSarana2
    public TampilanSarana() {
        initComponents();
        Sarana.setVisible(true);
        TombolSarana.setVisible(false);
        Tabel.setVisible(false);
    }
}

```


TampilanSarana2

```
import java.io.FileWriter;
import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class TampilanSarana2 extends javax.swing.JFrame {

    SaranaTest2 Sarana2 = new SaranaTest2();
    SaranaRuangan2 SetSarana2 = new SaranaRuangan2();
    TampilanKeadaan Keadaan = new TampilanKeadaan();

    public TampilanSarana2() {
        initComponents();
        Sarana_2.setVisible(true);
        Tabel.setVisible(false);
        TombolSarana2.setVisible(false);
    }

    private void NextSarana2ActionPerformed(java.awt.event.ActionEvent evt) {

        try{
            SetSarana2.setJumlahKipasAngin(Integer.parseInt(TJumlahKipas.getText()));
            SetSarana2.setKondisiKipasAngin(Integer.parseInt(TKondisiKipasB.getText()));
            SetSarana2.setKondisiKipasAnginBuruk(Integer.parseInt(TKondisiKipasR.getText()));
            SetSarana2.setPosisiKipasAngin(Integer.parseInt(TPosisiKipasA.getText()));

            SetSarana2.setJumlahAC(Integer.parseInt(TJumlahAC.getText()));
            SetSarana2.setKondisiAC(Integer.parseInt(TKondisiACB.getText()));
            SetSarana2.setKondisiACBuruk(Integer.parseInt(TKondisiACR.getText()));
            SetSarana2.setPosisiACBelakang(Integer.parseInt(TPosisiACB.getText()));
            SetSarana2.setPosisiACSamping(Integer.parseInt(TPosisiACS.getText()));

            SetSarana2.setSSID(CSSID.getSelectedItemAt().toString());
            SetSarana2.setBandwith(Integer.parseInt(TBandwith.getText()));
```

```
SetSarana2.setJumlahCCTV(Integer.parseInt(TJumlahCCTV.getText()));
SetSarana2.setKondisiCCTV(Integer.parseInt(TKondisiCCTVB.getText()));
SetSarana2.setKondisiCCTVBuruk(Integer.parseInt(TKondisiCCTVR.getText()));
SetSarana2.setPosisiCCTVBelakang(Integer.parseInt(TPosisiCCTVB.getText()));
SetSarana2.setPosisiCCTVDepan(Integer.parseInt(TPosisiCCTVD.getText()));
```

```
Sarana2.input(SetSarana2.getJumlahKipasAngin(),
SetSarana2.getKondisiKipasAngin(), SetSarana2.getKondisiKipasAnginBuruk(),
SetSarana2.getPosisiKipasAngin(),
    SetSarana2.getJumlahAC(), SetSarana2.getKondisiAC(),
SetSarana2.getKondisiACBuruk(),SetSarana2.getPosisiACBelakang(),
SetSarana2.getPosisiACSampling(),
    SetSarana2.getJumlahCCTV(), SetSarana2.getKondisiCCTV(),
SetSarana2.getKondisiCCTVBuruk(),
SetSarana2.getPosisiCCTVBelakang(),SetSarana2.getPosisiCCTVDepan(),
    SetSarana2.getSSID(),SetSarana2.getBandwith());
Sarana2.Analisis();
Sarana2.Sesuai();
```

```
DefaultTableModel modelAC = (DefaultTableModel) TableAC.getModel();
modelAC.addRow(new Object[]{SetSarana2.getJumlahAC(),
SetSarana2.getKondisiAC(),
    SetSarana2.getKondisiACBuruk(),SetSarana2.getPosisiACBelakang(),
SetSarana2.getPosisiACSampling()});
```

```
DefaultTableModel modelCCTV = (DefaultTableModel) TableCCTV.getModel();
modelCCTV.addRow(new Object[]{SetSarana2.getJumlahCCTV(),
SetSarana2.getKondisiCCTV(),
    SetSarana2.getKondisiCCTVBuruk(),SetSarana2.getPosisiCCTVDepan(),
SetSarana2.getPosisiCCTVBelakang()});
```

```
DefaultTableModel modelKA = (DefaultTableModel) TableKipas.getModel();
modelKA.addRow(new Object[]{SetSarana2.getJumlahKipasAngin(),
SetSarana2.getKondisiKipasAngin(),
    SetSarana2.getKondisiKipasAnginBuruk(),
SetSarana2.getPosisiKipasAngin(),SetSarana2.getSSID(),SetSarana2.getBandwith()});
```

```
Tabel.setVisible(true);
Sarana_2.setVisible(false);
TombolSarana2.setVisible(true);
```

```
    }catch(Exception e){  
        System.out.println("ERROR");  
    }  
}
```

```
private void SJumlahACStateChanged(javax.swing.event.ChangeEvent evt) {
```

```
    TJumlahAC.setText(String.valueOf(SJumlahAC.getValue()));  
    SKondisiACB.setMaximum(SJumlahAC.getValue());  
    SPosisiACB.setMaximum(SJumlahAC.getValue());  
}
```

```
private void SKondisiACBStateChanged(javax.swing.event.ChangeEvent evt) {
```

```
    SKondisiACB.setMaximum(SJumlahAC.getValue());  
    SKondisiACR.setMaximum(SJumlahAC.getValue() - SKondisiACB.getValue());  
    SKondisiACR.setMinimum(0);  
    TKondisiACB.setText(String.valueOf(SKondisiACB.getValue()));  
    TKondisiACR.setText(String.valueOf(SKondisiACR.getValue()));  
}
```

```
private void SKondisiACRStateChanged(javax.swing.event.ChangeEvent evt) {
```

```
    TKondisiACR.setText(String.valueOf(SKondisiACR.getValue()));  
}
```

```
private void SPosisiACBStateChanged(javax.swing.event.ChangeEvent evt) {
```

```
    SPosisiACB.setMaximum(SJumlahAC.getValue());  
    SPosisiACS.setMaximum(SJumlahAC.getValue() - SPosisiACB.getValue());  
    SPosisiACS.setMinimum(0);  
    TPosisiACB.setText(String.valueOf(SPosisiACB.getValue()));  
    TPosisiACS.setText(String.valueOf(SPosisiACS.getValue()));  
}
```

```
private void SPosisiACSStateChanged(javax.swing.event.ChangeEvent evt) {
```

```
    TPosisiACS.setText(String.valueOf(SPosisiACS.getValue()));
```

```
}
```

```
private void SJumlahKipasStateChanged(javax.swing.event.ChangeEvent evt) {
```

```
TJumlahKipas.setText(String.valueOf(SJumlahKipas.getValue()));
```

```
SKondisiKipasB.setMaximum(SJumlahKipas.getValue());
```

```
}
```

```
private void SKondisiKipasBStateChanged(javax.swing.event.ChangeEvent evt) {
```

```
SKondisiKipasB.setMaximum(SJumlahKipas.getValue());
```

```
SKondisiKipasR.setMaximum(SJumlahKipas.getValue() -  
SKondisiKipasB.getValue());
```

```
SKondisiKipasR.setMinimum(0);
```

```
TKondisiKipasB.setText(String.valueOf(SKondisiKipasB.getValue()));
```

```
TKondisiKipasR.setText(String.valueOf(SKondisiKipasR.getValue()));
```

```
}
```

```
private void SKondisiKipasRStateChanged(javax.swing.event.ChangeEvent evt) {
```

```
TKondisiKipasR.setText(String.valueOf(SKondisiKipasR.getValue()));
```

```
}
```

```
private void SPosisiKipasAStateChanged(javax.swing.event.ChangeEvent evt) {
```

```
SPosisiKipasA.setMaximum(SJumlahKipas.getValue());
```

```
TPosisiKipasA.setText(String.valueOf(SPosisiKipasA.getValue()));
```

```
}
```

```
private void SJumlahCCTVStateChanged(javax.swing.event.ChangeEvent evt) {
```

```
TJumlahCCTV.setText(String.valueOf(SJumlahCCTV.getValue()));
```

```
SKondisiCCTVB.setMaximum(SJumlahCCTV.getValue());
```

```

SPosisiCCTVD.setMaximum(SJumlahCCTV.getValue());
    }

    private void SKondisiCCTVBStateChanged(javax.swing.event.ChangeEvent evt) {

SKondisiCCTVB.setMaximum(SJumlahCCTV.getValue());
SKondisiCCTVR.setMaximum(SJumlahCCTV.getValue() - SKondisiCCTVB.getValue());
    SKondisiCCTVR.setMinimum(0);
TKondisiCCTVB.setText(String.valueOf(SKondisiCCTVB.getValue()));
TKondisiCCTVR.setText(String.valueOf(SKondisiCCTVR.getValue()));
    }

    private void SKondisiCCTVRStateChanged(javax.swing.event.ChangeEvent evt) {

TKondisiCCTVR.setText(String.valueOf(SKondisiCCTVR.getValue()));
    }

    private void SPosisiCCTVDStateChanged(javax.swing.event.ChangeEvent evt) {

SPosisiCCTVD.setMaximum(SJumlahCCTV.getValue());
SPosisiCCTVB.setMaximum(SJumlahCCTV.getValue() - SPosisiCCTVD.getValue());
    SPosisiCCTVB.setMinimum(0);
TPosisiCCTVD.setText(String.valueOf(SPosisiCCTVD.getValue()));
TPosisiCCTVB.setText(String.valueOf(SPosisiCCTVB.getValue()));
    }

    private void SPosisiCCTVBStateChanged(javax.swing.event.ChangeEvent evt) {

TPosisiCCTVB.setText(String.valueOf(SPosisiCCTVB.getValue()));
    }


    private void SBandwithStateChanged(javax.swing.event.ChangeEvent evt) {

TBandwith.setText(String.valueOf(SBandwith.getValue()));
    }

    private void SaveActionPerformed(java.awt.event.ActionEvent evt) {

```

```

        Keadaan.Keadaan();
        Sarana2.Save();
        this.setVisible(false);
        Tabel.setVisible(false);
        Sarana_2.setVisible(true);
        TombolSarana2.setVisible(false);
    }

    private void EditActionPerformed(java.awt.event.ActionEvent evt) {

        Sarana_2.setVisible(true);
        Tabel.setVisible(false);
        TombolSarana2.setVisible(false);
    }

    public void Sarana2() {

        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
                javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

            java.util.logging.Logger.getLogger(TampilanSarana2.class.getName()).log(java.util.logging.
                Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

            java.util.logging.Logger.getLogger(TampilanSarana2.class.getName()).log(java.util.logging.
                Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

            java.util.logging.Logger.getLogger(TampilanSarana2.class.getName()).log(java.util.logging.
                Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

            java.util.logging.Logger.getLogger(TampilanSarana2.class.getName()).log(java.util.logging.
                Level.SEVERE, null, ex);
        }
    }

```

```
}  
//</editor-fold>  
//</editor-fold>  
  
this.setVisible(true);  
}
```

```
import java.io.FileWriter;  
import java.io.IOException;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import javax.swing.JOptionPane;  
import javax.swing.table.DefaultTableModel;  
  
public class TampilanSarana2 extends javax.swing.JFrame {  
method TampilanSarana2 diwariskan oleh javax.swing.JFrame  
  
    SaranaTest2 Sarana2 = new SaranaTest2();  
Membuat object Sarana2 dari SaranaTest2  
    SaranaRuangan2 SetSarana2 = new SaranaRuangan2();  
Membuat object SetSarana2 dari SaranaRuangan2  
    TampilanKeadaan Keadaan = new TampilanKeadaan();  
Membuat object Keadaan dari TampilanKeadaan  
    public TampilanSarana2() {  
method TampilanSarana2  
        initComponents();  
        Sarana_2.setVisible(true);  
        Tabel.setVisible(false);  
        TombolSarana2.setVisible(false);  
    }
```