LAPORAN AKHIR
PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK
Periode VI

Disusun Oleh :

NAMA : Alex Gilang S
NPM : 06.2019.1.07133

FAKULTAS TEKNIK ELEKTRO DAN TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI ADHI TAMA SURABAYA
SURABAYA
2020

LAPORAN AKHIR
PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK
Periode VI

Disusun Oleh:

ALEX GILANG S 06.2019.1.07133
ERIC WAHYU AMIRUDDIN

FAKULTAS TEKNIK ELEKTRO DAN TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI ADHI TAMA SURABAYA
SURABAYA
2020

LEMBAR PENGESAHAN

PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

Periode VI

Di susun oleh :

Alex Gilang Suherman

06.2019.1.07133

Tanggal 15/01/2021

Dosen Pembimbing	**Asisten Pembimbing**
Nanang Fakhrur R, S.ST., M.Kom	**Eric Wahyu Amiruddin**
NIP 122093	NPM 06.2018.1.07004

NAMA : Alex Gilang S.
NPM : 06.2019.1.07133

BIMBINGAN DOSEN
PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

No.	TANGGAL	POKOK BAHASAN	PARAF DOSEN PEMBIMBING

Surabaya, 15 Januari 2020
Dosen Pembimbing

Nanang Fakhrur R, S.ST., M.Kom
NIP 122093

NAMA : Alex Gilang S.
NPM : 06.2019.1.07133

BIMBINGAN ASISTEN
PRAKTIKUM PEMRGRAMAN BERORIENTASI OBJEK

No.	TANGGAL	POKOK BAHASAN	PARAF ASISTEN PEMBIMBING

Surabaya, 15 Januari 2020
Asisten Pembimbing

Eric Wahyu Amiruddin

RESMI
MODUL 1

LABORATORIUM REKAYASA PERANGKAT LUNAK
FAKULTAS TEKNIK ELEKTRO DAN

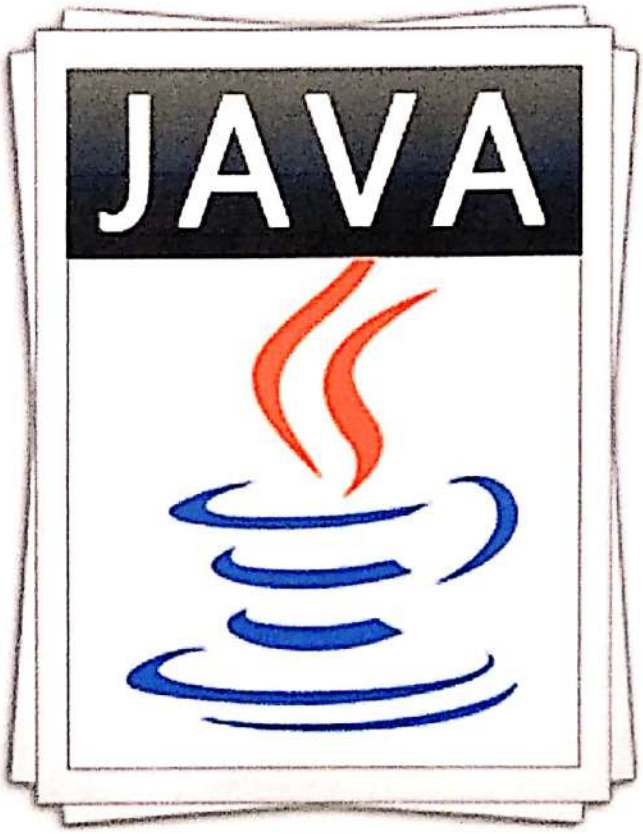
TEKNOLOGI INFORMASI

INSTITUT TEKNOLOGI ADHI TAMA

SURABAYA

2020

LAPORAN RESMI



PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
PERIODE VI - Tahun 2020/2021

NAMA	:	Alex Gilangs
NPM	:	06.2019.1.02133
MODUL	:	1 (Satu

Dipindai dengan CamScanner

Tugas Resmi

Soal Resmi

Membuat Class Diagram

Langkah Pertama

Membuat Class anggota

Source Code

```
package alex07133_rentalcamera;

public class Alex07133_anggota {
    String
Alex07133_id_anggota,Alex07133_nama,Alex07133
_notelp;

    public Alex07133_anggota(String
Alex07133_id_anggota,String
Alex07133_nama,String Alex07133_notelp){

this.Alex07133_id_anggota=Alex07133_id_anggot
a;
```

Screenshot

```
0.ASTRID
1.SABRINA
2.ALEX
Petugas Rental = 2
Data berhasil disimpan
```

Analisa

Digunakan untuk data anggota rental

Tugas Resmi

Langkah Kedua

Membuat class kamera

Source Code

```
package alex07133_rentalcamera;


public class Alex07133_camera {
    static String[]
Alex07133_CAMERA_CANON={"CANON 3000D","CANON
1500D","CANON 200D","CANON 750D"};
}
```

Screenshot

```
0. CANON 3000D
1. CANON 1500D
2. CANON 200D
3. CANON 750D
Pilih Kamera = 2
```

Analisa

Untuk Pemilihan kameranya

Tugas Resmi

Langkah Ketiga

Membuat class pelanggan

Source Code

```
package alex07133_rentalcamera;
import java.util.Date;

public class Alex07133_pelanggan {
    String
Alex07133_id_pelanggan,Alex07133_nama,Alex071
33_notelp;
    Date Alex07133_tglpinjam,
Alex07133_tglkembali;
    int Alex07133_indexkamera;
    int Alex07133_indexpetugas;

    public Alex07133_pelanggan(String
```

Screenshot

```
DATA KE 0
ID = 12
Nama = rayhan
Notelp = 08122359138
Kamera yang Dipinjam = CANON 200D
Tanggal Peminjaman Kamera (dd/mm/yyyy) = 06-11-2011
Tanggal Pengembalian Kamera (dd/mm/yyyy) = 06-12-2012
Petugas Rental = ALEX
```

Analisa

Untuk format penyimpanan data pelanggan

Tugas Resmi

Langkah Ke-empat

Membuat class RentalCamera

Source Code

```
package alex07133_rentalcamera;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.Scanner;

public class Alex07133_RentalCamera {
    static
ArrayList<Alex07133_anggota>dataPetugas = new
ArrayList();
    static
ArrayList<Alex07133_pelanggan>dataPelanggan =
new ArrayList();
```

Screenshot

```
Data Rental
Pilihan :
1. Input Data
2. Lihat Data
3. Ubah Data
4. Hapus Data
5. Exit
Pilih : 1
Input ID Pelanggan : 12
Input Nama Pelanggan : rayhan
Input Notelp Pelanggan : 08122359138
Input Tanggal Peminjaman Kamera (mm/dd/yyyy) = 18/11/2012
Input Tanggal Pengembalian Kamera (mm/dd/yyyy) = 19/11/2011
0. CANON 3000D
1. CANON 1500D
2. CANON 200D
3. CANON 750D
```

Analisa

Untuk membuat menu dan output input data

RESMI
MODUL 2

LABORATORIUM REKAYASA PERANGKAT LUNAK
FAKULTAS TEKNIK ELEKTRO DAN

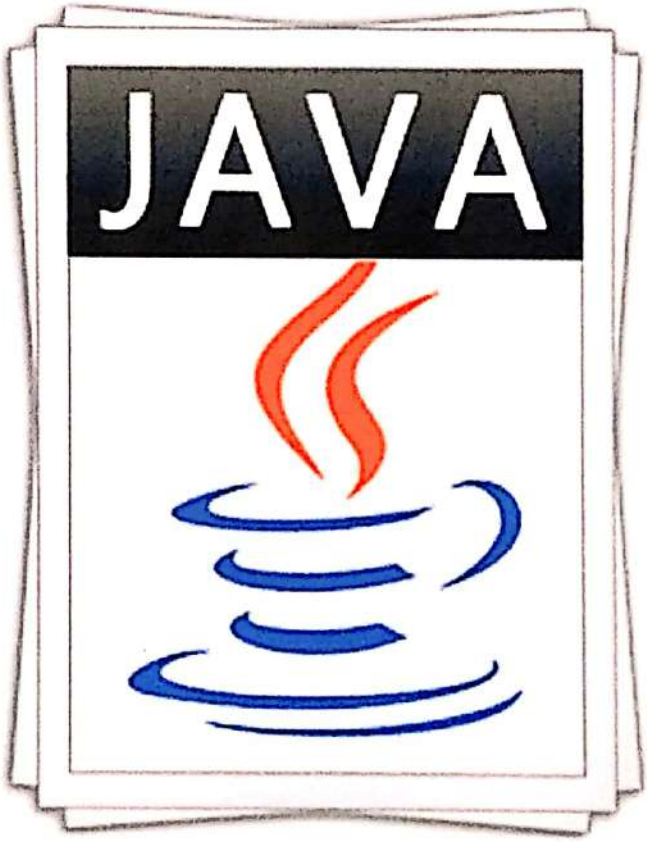
TEKNOLOGI INFORMASI

INSTITUT TEKNOLOGI ADHI TAMA

SURABAYA

2020

LAPORAN RESMI



PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
PERIODE VI - Tahun 2020/2021

NAMA	:	Alex Gilang S.
NPM	:	06.2019.1.07133
MODUL	:	2 (dua)

Dipindai dengan CamScanner

Tugas Resmi

Soal Resmi

Menerapkan Relasi Antar Objek

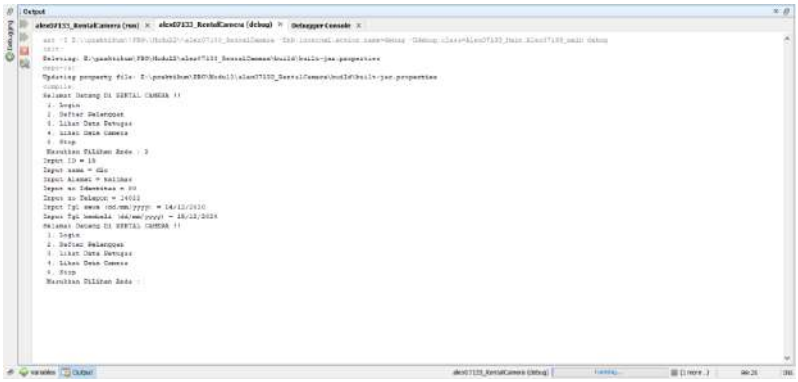
Langkah Pertama

Membuat Relasi Antar Objek

Source Code

```
package Alex07133_Entity;
import java.text.SimpleDateFormat;
import java.util.Date;
public class Alex07133_Rentalcamera {
protected String
id,nama,alamat,noIdentitas,notelp;

public Alex07133_Rentalcamera(String
id,String nama, //konstruktor,parameter
    String alamat, String noIdentitas,String
notelp){
    this.id=id:
```

Screenshot



Analisa

Antar class bisa saling terhubung

Tugas Resmi

Langkah Kedua

Membuat pewarisan dan Enkapsulasi

Source Code

```
public class Alex07133_PelangganEntity
extends Alex07133_Rentalcamera
protected Date tgl_sewa,tgl_kembali;
//atribut
    public Alex07133_PelangganEntity(String
id,String nama,
    String alamat, String noIdentitas,String
notelp,Date tgl_sewa,Date tgl_kembali){

super(id,nama,alamat,noIdentitas,notelp);//me
manggil construktor dari superclass
        this.tgl_sewa=tgl_sewa;
        this.tgl_kembali=tgl_kembali;
```

Screenshot

```
 Masukkan Pilihan Anda : 2
Input ID = 15
Input nama = dio
Input Alamat = kalimas
Input no Identitas = 80
Input no Telepon = 14022
Input Tgl sewa (dd/mm/yyyy) = 14/12/2020
Input Tgl kembali (dd/mm/yyyy) = 15/12/2020
```

Analisa

Output yang dihasilkan dari melalui enkapsulasi protected string dan juga pewarisan dari kelas Alex07133_RentalCamera

Tugas Resmi

Langkah Ketiga

Membuat Polimorphisme

Source Code

```
@Override   //pendefinisian ulang method yang
sama pada subclass
    public String getId() {
        return id;
    }
```

Screenshot

```
Selamat Datang Di RENTAL CAMERA !!
 1. Login
 2. Daftar Pelanggan
 3. Lihat Data Petugas
 4. Lihat Data Camera
 0. Stop
 Masukkan Pilihan Anda : 1
1. Login petugas
2. Login pelanggan
 Pilih : 2
ID : 15
Nama : dio
```

Analisa

pendefinisian ulang method yang sama pada subclass

RESMI
MODUL 3

LABORATORIUM REKAYASA PERANGKAT LUNAK
FAKULTAS TEKNIK ELEKTRO DAN

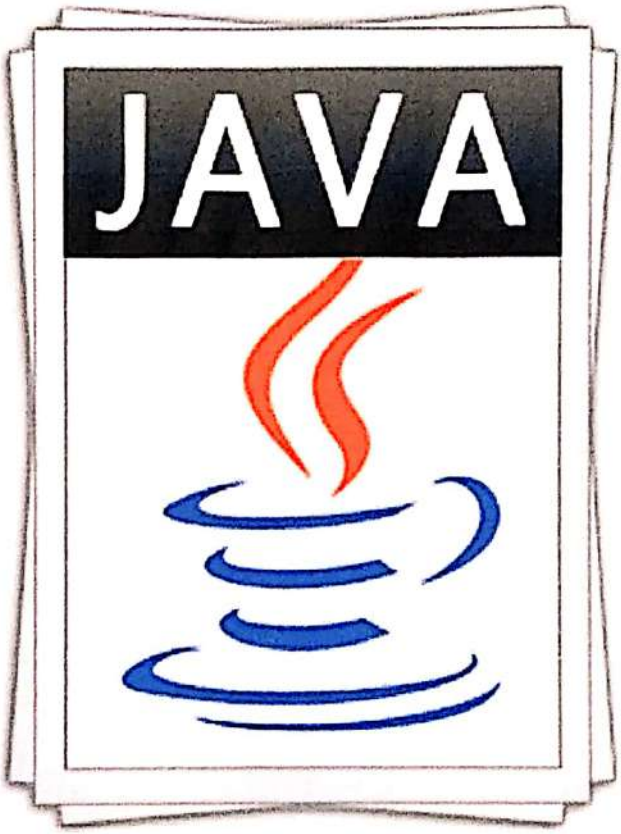
TEKNOLOGI INFORMASI

INSTITUT TEKNOLOGI ADHI TAMA

SURABAYA

2020

LAPORAN RESMI



PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
PERIODE VI - Tahun 2020/2021

NAMA : Alex Gilang S.
NPM : 06.2019.1.07133
MODUL : 3 (tiga)

Dipindai dengan CamScanner

Tugas Resmi

Soal Praktikum

1. *Melanjutkan program dari modul sebelumnya, tetap menerapkan CRUD pada program*
2. *Terapkan Exception Handling Pada program anda*

Langkah Pertama

Membuat package Controller, package Entity, dan package Model.

Membuat class dengan nama Alex07133_RentalCamera pada package Entity

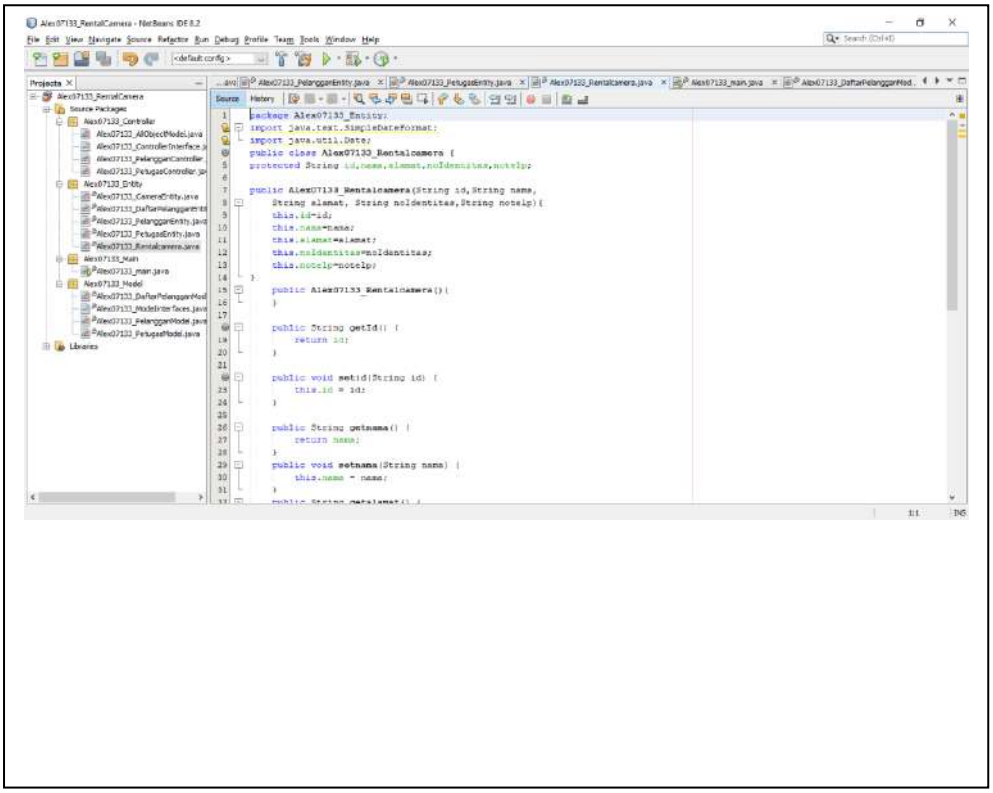
Source Code

```
package Entity;
import java.text.SimpleDateFormat;
import java.util.Date;
public class Alex07133_RentalCamera {
    protected String
id,nama,alamat,noIdentitas,notelp;

public Sabrina07239_RentalCamera(String id,String
nama, String alamat, String noIdentitas,String
notelp){
    this.id=id;
    this.nama=nama;
    this.alamat=alamat;
    this.noIdentitas=noIdentitas;
    this.notelp=notelp;
}
    public String getId() {
        return id;
    }
    public void setid(String id) {
        this.id = id;
    }
    public String getnama() {
        return nama;
    }
    public void setnama(String nama) {
        this.nama = nama;
    }
    public String getalamat() {
        return alamat;
    }
    public void setalamat(String alamat) {
```

Tugas Resmi

```
    public void setnoIdentitas(String noIdentitas) {
        this.noIdentitas = noIdentitas;
    }
    public String getnotelp() {
        return notelp;
    }
    public void setnotelp(String notelp) {
        this.notelp = notelp;
    }

}
```

Screenshot



Analisa

Pembuatan class ini bertujuan agar class ini menjadi class induk (parent class) untuk penerapan inheritance pada projek
Terdapat enkapsulasi dalam projek, yaitu menggunakan protected pada properti. Protected dapat diakses oleh class itu sendiri dan class turunan

Tugas Resmi

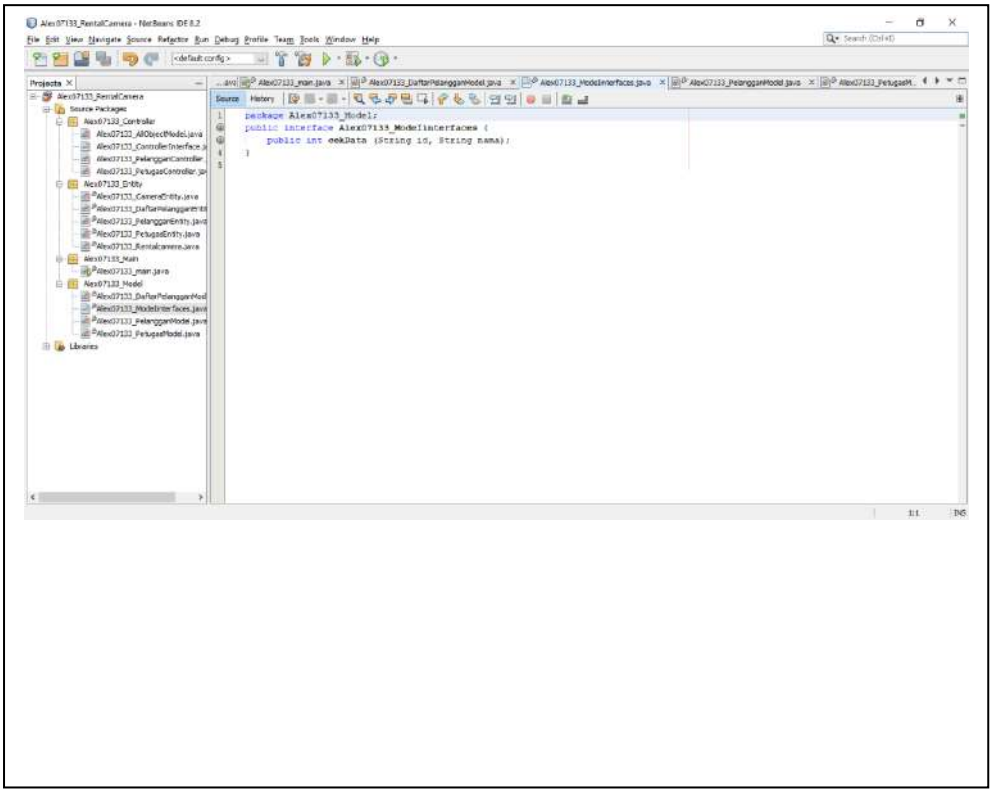
Langkah Kedua

Membuat kelas interface dengan nama Alex07133_ModeIinterfaces pada package Model

Source Code

```
package Model;
public interface Alex07133_ModeIinterfaces {
public int cekData (String id, String nama);
}
```

Screenshot



Analisa

Kelas interface hanya terdiri dari method-method kosong (abstract, tanpa body method). Interface biasanya lebih digunakan untuk mengelompokkan behavior-behavior menjadi grup behavior. Kelas interface menggambarkan kemampuan, bukan entitas/konsep objek.

Tugas Resmi

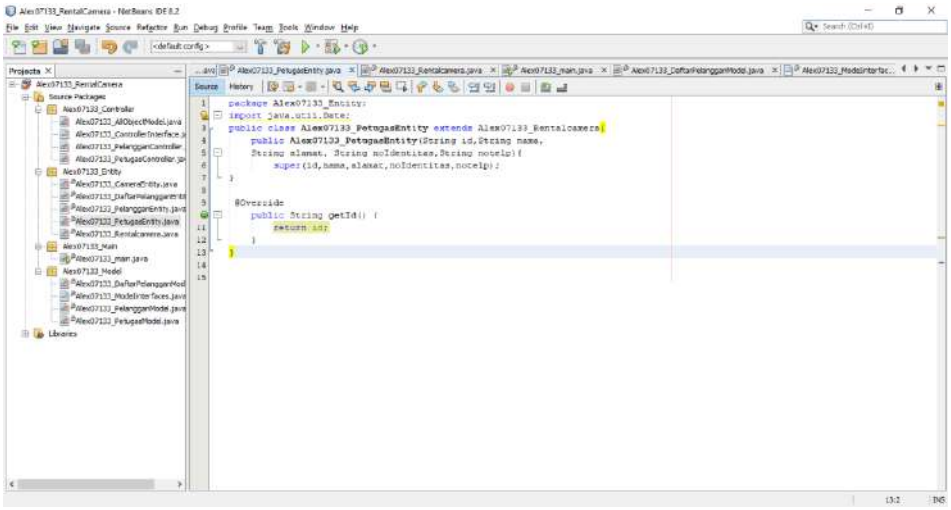
Langkah Ketiga

Membuat class Alex07133_PetugasEntity pada package Entity

Source Code

```
package Entity;
import java.util.Date;
public class Alex07133_PetugasEntity extends
Alex07133_RentalCamera{
    public Alex07133_PetugasEntity(String id,String
nama,
    String alamat, String noIdentitas,String notelp){
    super(id,nama,alamat,noIdentitas,notelp);
}
 @Override
    public String getId() {
        return id;
    }
}
```

Screenshot



Tugas Resmi

Analisa

Terdapat inheritance yang diterapkan pada class ini, dimana class induknya adalah Alex07133_RentalCamera. Ditandai dengan keyword extends sebelum nama class induk disebutkan, keyword extens harus ditambahkan pada definisi class yang menjadi subclass. Selain itu terdapat keyword super untuk memanggil contructor dari class induk. Access modifier dari class induk akan diwariskan pada subclass ketika inheritance dilakukan.
Terdapat enkapsulasi yang diterapkan pada class ini, yaitu public. Public mempunyai sifat dapat diakses oleh semua class.
Terdapat polymorphism yang diterapkan pada class ini yaitu method overriding, method overriding digunakan untuk pendefinisian ulang method yang sama pada subclass, yang ditandai dengan keyword @Override.

Tugas Resmi

Langkah Keempat

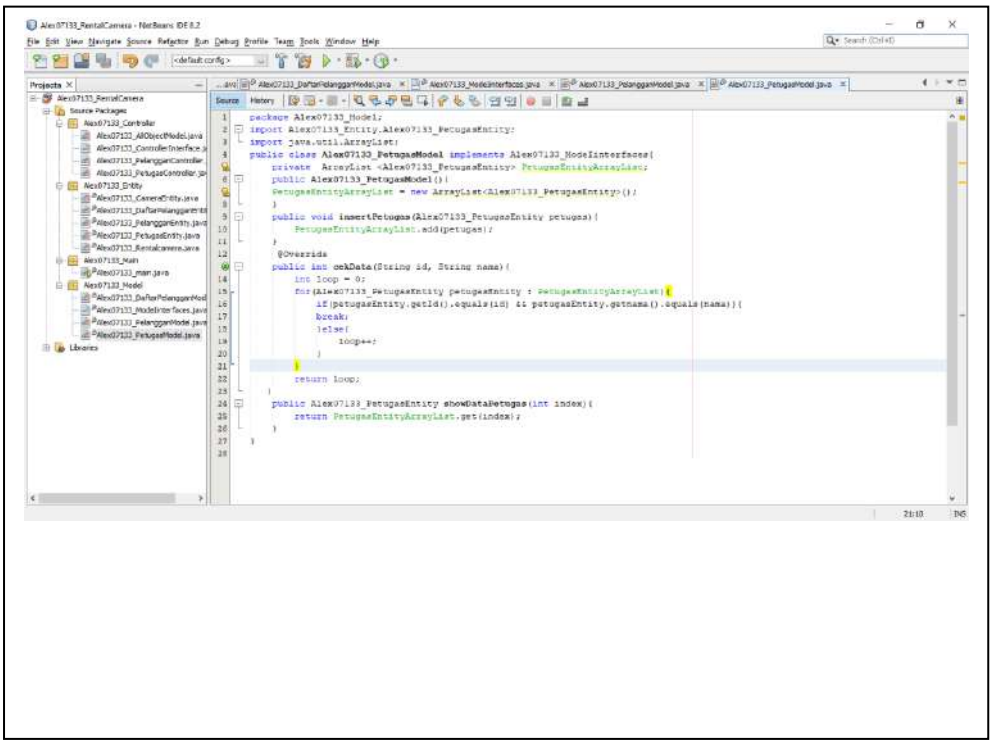
Membuat class Alex07133_PetugasModel pada package Model

Source Code

```
package Alex07133_Model;
import Alex07133_Entity.Alex07133_PetugasEntity;
import java.util.ArrayList;
public class Alex07133_PetugasModel implements
Alex07133_ModeIinterfaces{
    private  ArrayList <Alex07133_PetugasEntity>
PetugasEntityArrayList;
    public Alex07133_PetugasModel(){
    PetugasEntityArrayList = new
ArrayList<Alex07133_PetugasEntity>();
    }
    public void insertPetugas(Alex07133_PetugasEntity
petugas){
        PetugasEntityArrayList.add(petugas);
    }
     @Override
    public int cekData(String id, String nama){
        int loop = 0;
        for(Alex07133_PetugasEntity petugasEntity :
PetugasEntityArrayList){
            if(petugasEntity.getId().equals(id) &&
petugasEntity.getnama().equals(nama)){
            break;
            }else{
                loop++;
            }
        }
        return loop;
   }
    public Alex07133_PetugasEntity
showDataPetugas(int index){
        return PetugasEntityArrayList.get(index);
    }
}
```

Tugas Resmi

Screenshot



Analisa

Terdapat implementasi yang diterapkan pada class, class ini mengimplementasi dari class Alex07133_ModeIInterface. Class melakukan implementasi agar class bisa mengadopsi behavior-behavior dari class Alex07133ModeIInterface.
Terdapat enkapsulasi yang diterapkan pada class ini, yaitu private dan public. Private memiliki sifat bahwa properti dan method hanya bisa diakses oleh class itu sendiri, sedangkan public dapat diakses oleh semua class.
Terdapat polymorphism yang diterapkan pada class ini yaitu method overriding, method overriding digunakan untuk pendefinisian ulang method yang sama pada subclass, yang ditandai dengan keyword @Override.

Tugas Resmi

Langkah Kelima

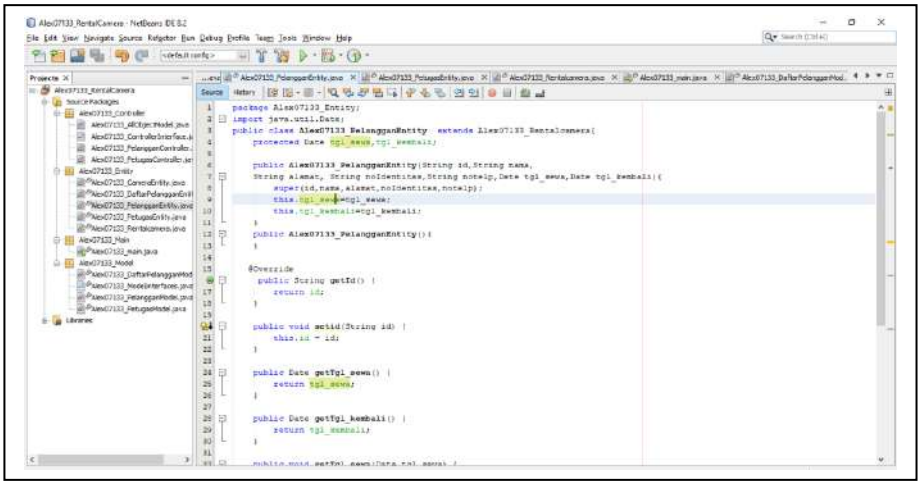
Membuat class Alex07133_PelangganEntity pada package Entity

Source Code

```
package Alex07133_Entity;
import java.util.Date;
public class Alex07133_PelangganEntity  extends
Alex07133_Rentalcamera{
    protected Date tgl_sewa,tgl_kembali;

    public Alex07133_PelangganEntity(String id,String
nama,
    String alamat, String noIdentitas,String
notelp,Date tgl_sewa,Date tgl_kembali){
        super(id,nama,alamat,noIdentitas,notelp);
        this.tgl_sewa=tgl_sewa;
        this.tgl_kembali=tgl_kembali;
    }
    public Alex07133_PelangganEntity(){
    }

   @Override
     public String getId() {
        return id;
    }

    public void setid(String id) {
        this.id = id;
    }

    public Date getTgl_sewa() {
        return tgl_sewa;
    }

    public Date getTgl_kembali() {
        return tgl_kembali;
    }

    public void setTgl_sewa(Date tgl_sewa) {
        this.tgl_sewa = tgl_sewa;
```

Tugas Resmi

Screenshot



Analisa

Terdapat inheritance yang diterapkan pada class ini, dimana class induknya adalah Alex07133RentalCamera. Ditandai dengan keyword extends sebelum nama class induk disebutkan, keyword extens harus ditambahkan pada definisi class yang menjadi subclass. Selain itu terdapat keyword super untuk memanggil contructor dari class induk. Access modifier dari class induk akan diwariskan pada subclass ketika inheritance dilakukan.

Terdapat enkapsulasi yang diterapkan pada class ini, yaitu public dan protected. Public mempunyai sifat dapat diakses oleh semua class. Sedangkan protected, properti dan method bisa diakses oleh class itu sendiri dan subclass

Terdapat polymorphism yang diterapkan pada class ini yaitu method overriding, method overriding digunakan untuk pendefinisian ulang method yang sama pada subclass, yang ditandai dengan keyword @Override.

Tugas Resmi

Langkah Keenam

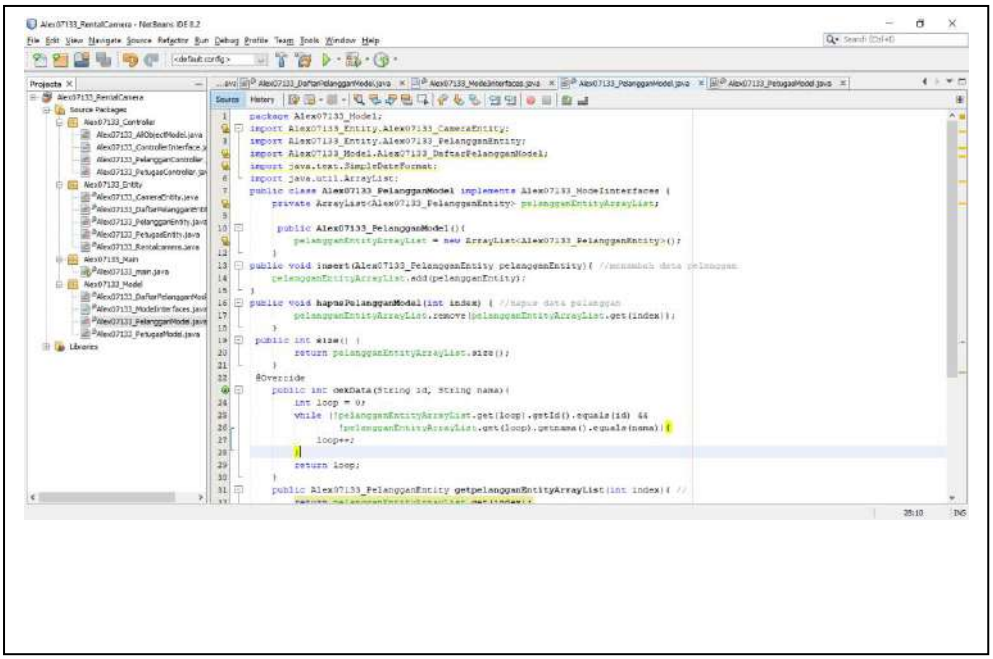
Membuat class Alex07133_PelangganModel pada package Model

Source Code

```
package Alex07133_Model;
import Alex07133_Entity.Alex07133_CameraEntity;
import Alex07133_Entity.Alex07133_PelangganEntity;
import
Alex07133_Model.Alex07133_DaftarPelangganModel;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
public class Alex07133_PelangganModel implements
Alex07133_ModeIinterfaces {
    private ArrayList<Alex07133_PelangganEntity>
pelangganEntityArrayList;

     public Alex07133_PelangganModel(){
        pelangganEntityArrayList = new
ArrayList<Alex07133_PelangganEntity>();
    }
public void insert(Alex07133_PelangganEntity
pelangganEntity){ //menambah data pelanggan
    pelangganEntityArrayList.add(pelangganEntity);
}
public void hapusPelangganModel(int index) { //hapus
data pelanggan

pelangganEntityArrayList.remove(pelangganEntityArrayL
ist.get(index));
    }
 public int size() {
        return pelangganEntityArrayList.size();
    }
 @Override
    public int cekData(String id, String nama){
        int loop = 0;
        while
(!pelangganEntityArrayList.get(loop).getId().equals(i
d) &&

!pelangganEntityArrayList.get(loop).getnama().equals(
nama)){
            loop++;
        }
```

Tugas Resmi

Screenshot



Analisa

Terdapat implementasi yang diterapkan pada class, class ini mengimplementasi dari class Alex07133_ModeIinterface. Class melakukan implementasi agar class bisa mengadopsi behavior-behavior dari class Alex07133_ModeIinterface.
Terdapat enkapsulasi yang diterapkan pada class ini, yaitu private dan public. Private memiliki sifat bahwa properti dan method hanya bisa diakses oleh class itu sendiri, sedangkan public dapat diakses oleh semua class.
Terdapat polymorphism yang diterapkan pada class ini yaitu method overriding, method overriding digunakan untuk pendefinisian ulang method yang sama pada subclass, yang ditandai dengan keyword @Override.

Tugas Resmi

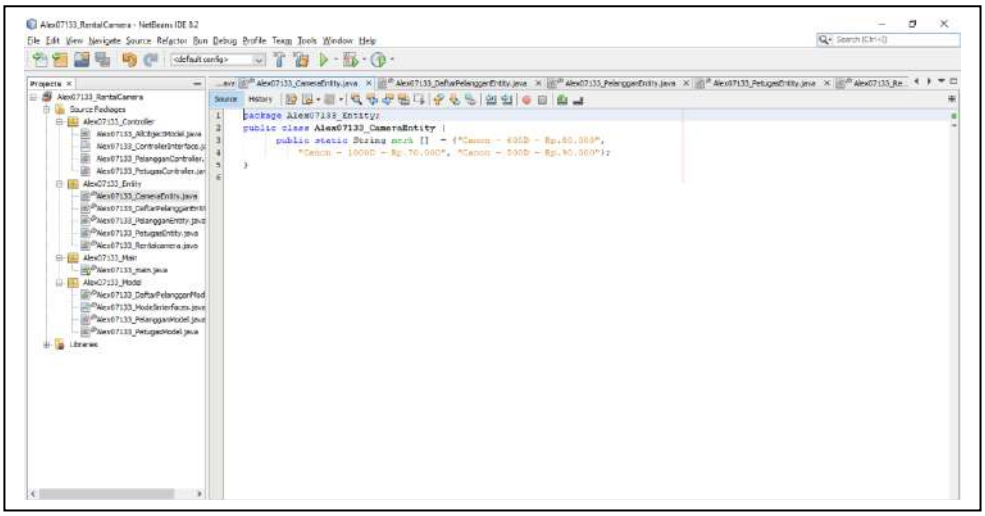
Langkah Ketujuh

Membuat class Alex07133_CameraEntity pada package Entity

Source Code

```
package Entity;
public class Alex07133_CameraEntity {
      public static String merk []  = {"Canon - 600D
- Rp.80.000", "Canon - 1000D - Rp.70.000", "Canon -
500D - Rp.90.000"}; //atribut
}
```

Screenshot



Analisa

Class terdapat data array yang berisi daftar pilihan camera, bertipe data String.
Class tidak menerapkan inheritance dan polymorphism.
Terdapat enkapsulasi yang diterapkan pada class ini, yaitu public. Public memiliki sifat bahwa properti dan method bisa diakses oleh semua class.

Tugas Resmi

Langkah Kedelapan

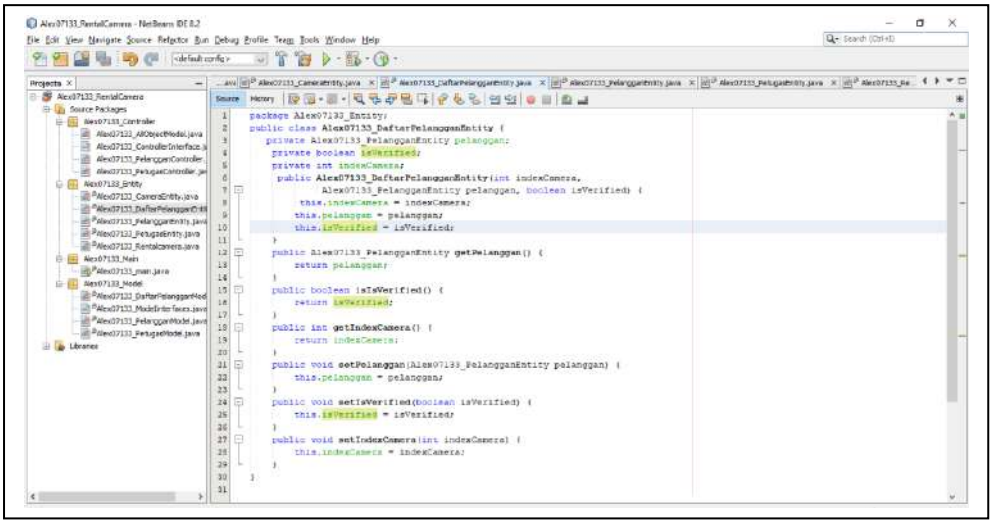
Membuat class Alex07133_DaftarPelangganEntity pada package Entity

Source Code

```
package Alex07133_Entity;
public class Alex07133_DaftarPelangganEntity {
   private Alex07133_PelangganEntity pelanggan;
    private boolean isVerified;
    private int indexCamera;
     public Alex07133_DaftarPelangganEntity(int
indexCamera,
             Alex07133_PelangganEntity pelanggan,
boolean isVerified) {
         this.indexCamera = indexCamera;
        this.pelanggan = pelanggan;
        this.isVerified = isVerified;
    }
    public Alex07133_PelangganEntity getPelanggan() {
        return pelanggan;
    }
    public boolean isIsVerified() {
        return isVerified;
    }
    public int getIndexCamera() {
        return indexCamera;
    }
    public void
setPelanggan(Alex07133_PelangganEntity pelanggan) {
        this.pelanggan = pelanggan;
    }
    public void setIsVerified(boolean isVerified) {
        this.isVerified = isVerified;
    }
    public void setIndexCamera(int indexCamera) {
        this.indexCamera = indexCamera;
    }
}
```

Tugas Resmi

Screenshot



Analisa

Terdapat enkapsulasi yang diterapkan pada class ini, yaitu private dan public. Private memiliki sifat bahwa properti dan method hanya bisa diakses oleh class itu sendiri, sedangkan public dapat diakses oleh semua class.
Terdapat variable untuk menampung data pelanggan, serta isVerified untuk mengonfirmasi status pelanggan nantinya, dan juga indexCamera untuk menginput data pilihan camera.

Tugas Resmi

Langkah Kesembilan

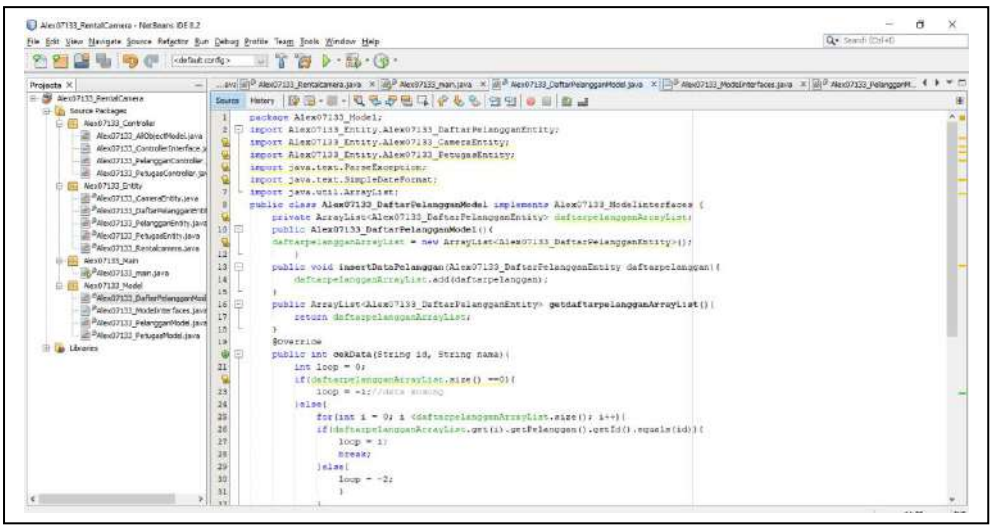
Membuat class Alex07133_DaftarPelangganModel pada package Model

Source Code

```
package Alex07133_Model;
import
Alex07133_Entity.Alex07133_DaftarPelangganEntity;
import Alex07133_Entity.Alex07133_CameraEntity;
import Alex07133_Entity.Alex07133_PetugasEntity;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
public class Alex07133_DaftarPelangganModel
implements Alex07133_ModeIinterfaces {
    private
ArrayList<Alex07133_DaftarPelangganEntity>
daftarpelangganArrayList;
    public Alex07133_DaftarPelangganModel(){
    daftarpelangganArrayList = new
ArrayList<Alex07133_DaftarPelangganEntity>();
        }
    public void
insertDataPelanggan(Alex07133_DaftarPelangganEntity
daftarpelanggan){

daftarpelangganArrayList.add(daftarpelanggan);
    }
    public ArrayList<Alex07133_DaftarPelangganEntity>
getdaftarpelangganArrayList(){
        return daftarpelangganArrayList;
    }
    @Override
    public int cekData(String id, String nama){
        int loop = 0;
        if(daftarpelangganArrayList.size() ==0){
            loop = -1;//data kosong
        }else{
            for(int i = 0; i
<daftarpelangganArrayList.size(); i++){

if(daftarpelangganArrayList.get(i).getPelanggan().get
Id().equals(id)){
                loop = i;
                break;
```

Tugas Resmi

Screenshot



Analisa

Terdapat implementasi yang diterapkan pada class, class ini mengimplementasi dari class Alex07133_ModeIinterface. Class melakukan implementasi agar class bisa mengadopsi behavior-behavior dari class Alex07133_ModeIinterface.
Terdapat enkapsulasi yang diterapkan pada class ini, yaitu private dan public. Private memiliki sifat bahwa properti dan method hanya bisa diakses oleh class itu sendiri, sedangkan public dapat diakses oleh semua class.
Terdapat polymorphism yang diterapkan pada class ini yaitu method overriding, method overriding digunakan untuk pendefinisian ulang method yang sama pada subclass, yang ditandai dengan keyword @Override.

Tugas Resmi

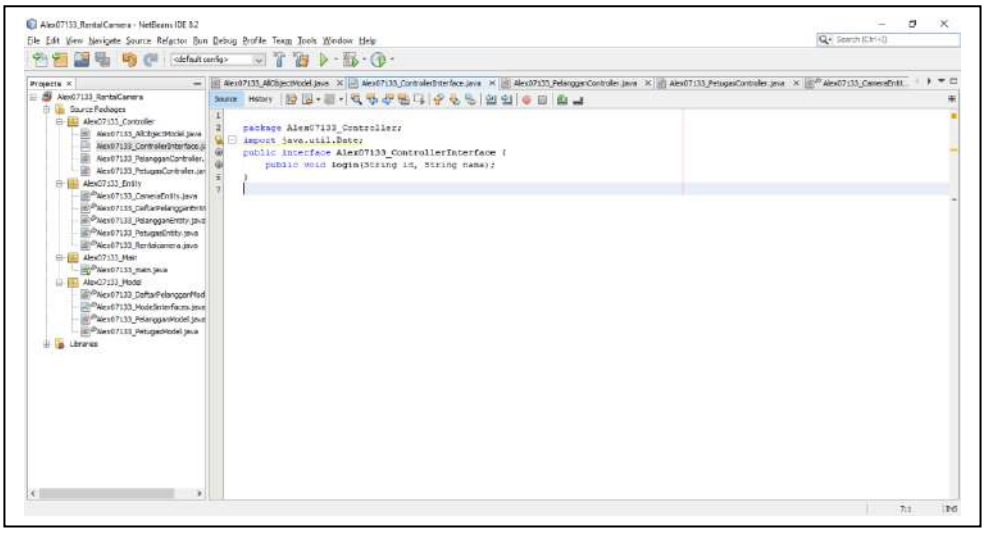
Langkah Kesepuluh

Membuat class Alex07133_ControllerInterface pada package Controller

Source Code

```
package Controller;

import java.util.Date;
public interface Alex07133_ControllerInterface {
    public void login(String id, String nama);
}
```

Screenshot



Analisa

Kelas interface hanya terdiri dari method-method kosong (abstract, tanpa body method). Interface biasanya lebih digunakan untuk mengelompokkan behavior-behavior menjadi grup behavior. Kelas interface menggambarkan kemampuan, bukan entitas/konsep objek.

Tugas Resmi

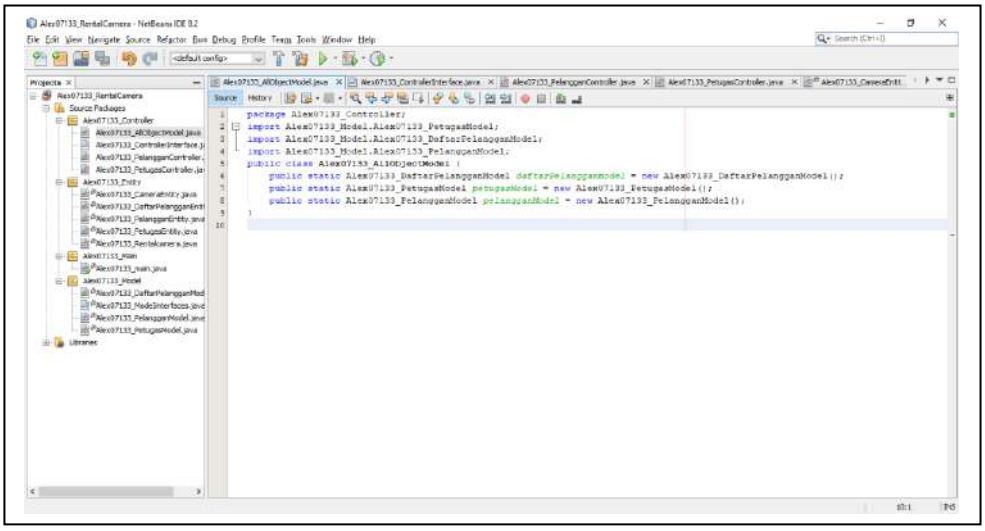
Langkah Kesebelas

Membuat class Alex07133_AllObjectModel pada package Controller

Source Code

```
package Alex07133_Controller;
import Alex07133_Model.Alex07133_PetugasModel;
import
Alex07133_Model.Alex07133_DaftarPelangganModel;
import Alex07133_Model.Alex07133_PelangganModel;
public class Alex07133_AllObjectModel {
    public static Alex07133_DaftarPelangganModel
daftarPelangganmodel = new
Alex07133_DaftarPelangganModel();
    public static Alex07133_PetugasModel petugasModel
= new Alex07133_PetugasModel();
    public static Alex07133_PelangganModel
pelangganModel = new Alex07133_PelangganModel();
}
```

Screenshot



Analisa

Kelas yang diciptakan untuk menjadi penghubung antara class-class yang berada pada package Model dan main program

Tugas Resmi

Langkah Kedua belas

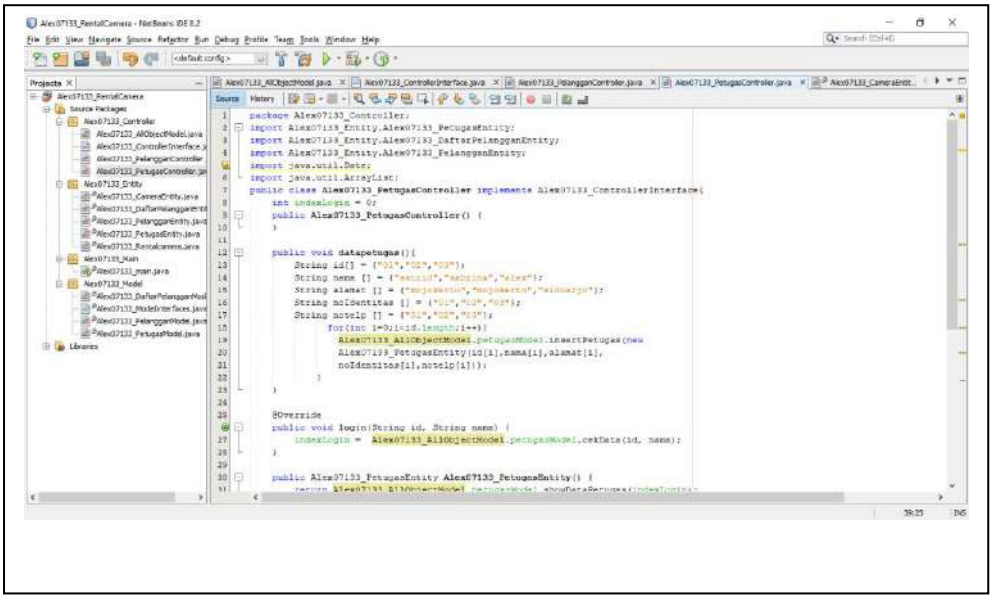
Membuat class Alex07133_PetugasController pada package Controller

Source Code

```
package Alex07133_Controller;
import Alex07133_Entity.Alex07133_PetugasEntity;
import
Alex07133_Entity.Alex07133_DaftarPelangganEntity;
import Alex07133_Entity.Alex07133_PelangganEntity;
import java.util.Date;
import java.util.ArrayList;
public class Alex07133_PetugasController implements
Alex07133_ControllerInterface{
    int indexLogin = 0;
    public Alex07133_PetugasController() {
    }

    public void datapetugas(){
        String id[] = {"01","02","03"};
        String nama [] = {"astrid","sabrina","alex"};
        String alamat [] =
{"mojokerto","mojokerto","sidoarjo"};
        String noIdentitas [] = {"01","02","03"};
        String notelp [] = {"01","02","03"};
              for(int i=0;i<id.length;i++){

Alex07133_AllObjectModel.petugasModel.insertPetugas(n
ew

Alex07133_PetugasEntity(id[i],nama[i],alamat[i],
                noIdentitas[i],notelp[i]));
            }
    }

    @Override
    public void login(String id, String nama) {
        indexLogin =
Alex07133_AllObjectModel.petugasModel.cekData(id,
nama);
    }
```

Tugas Resmi

Screenshot



Analisa

Terdapat implementasi yang diterapkan pada class, class ini mengimplementasi dari class Alex07133_ControllerInterface. Class melakukan implementasi agar class bisa mengadopsi behavior-behavior dari class Alex07133_ControllerInterface.

Terdapat polymorphism yang diterapkan pada class ini yaitu method overriding, method overriding digunakan untuk pendefinisian ulang method yang sama pada subclass, yang ditandai dengan keyword @Override.

Tugas Resmi

Langkah Ketiga belas

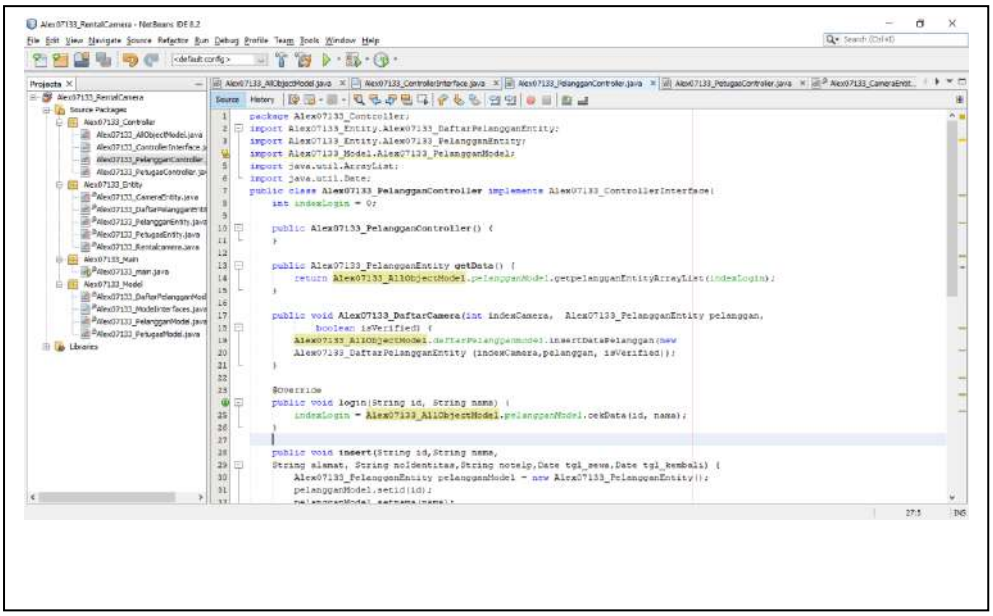
Membuat class Alex07133_PelangganController pada package Controller

Source Code

```
package Alex07133_Controller;
import
Alex07133_Entity.Alex07133_DaftarPelangganEntity;
import Alex07133_Entity.Alex07133_PelangganEntity;
import Alex07133_Model.Alex07133_PelangganModel;
import java.util.ArrayList;
import java.util.Date;
public class Alex07133_PelangganController implements
Alex07133_ControllerInterface{
    int indexLogin = 0;

    public Alex07133_PelangganController() {
    }

    public Alex07133_PelangganEntity getData() {
        return
Alex07133_AllObjectModel.pelangganModel.getpelangganE
ntityArrayList(indexLogin);
    }

    public void Alex07133_DaftarCamera(int
indexCamera,  Alex07133_PelangganEntity pelanggan,
            boolean isVerified) {

Alex07133_AllObjectModel.daftarPelangganmodel.insertD
ataPelanggan(new
        Alex07133_DaftarPelangganEntity
(indexCamera,pelanggan, isVerified));
    }

    @Override
    public void login(String id, String nama) {
        indexLogin =
Alex07133_AllObjectModel.pelangganModel.cekData(id,
nama);
    }

    public void insert(String id,String nama,
    String alamat, String noIdentitas,String
notelp,Date tgl_sewa,Date tgl_kembali) {
```

Tugas Resmi

Screenshot



Analisa

Terdapat implementasi yang diterapkan pada class, class ini mengimplementasi dari class Alex07133_ControllerInterface. Class melakukan implementasi agar class bisa mengadopsi behavior-behavior dari class Alex07133_ControllerInterface.

Terdapat polymorphism yang diterapkan pada class ini yaitu method overriding, method overriding digunakan untuk pendefinisian ulang method yang sama pada subclass, yang ditandai dengan keyword @Override.

Tugas Resmi

Langkah Keempat belas

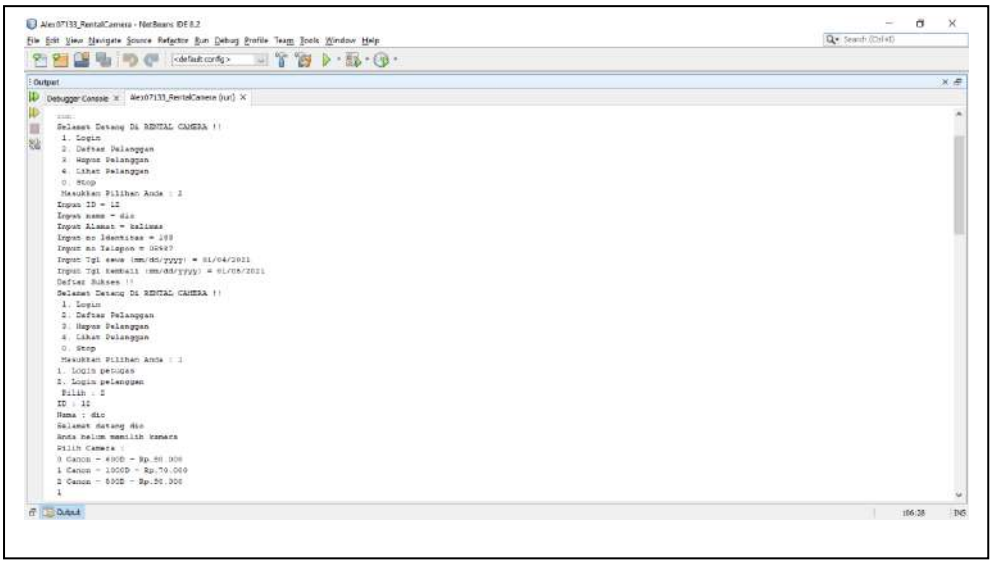
Mengisi code-code pada main class

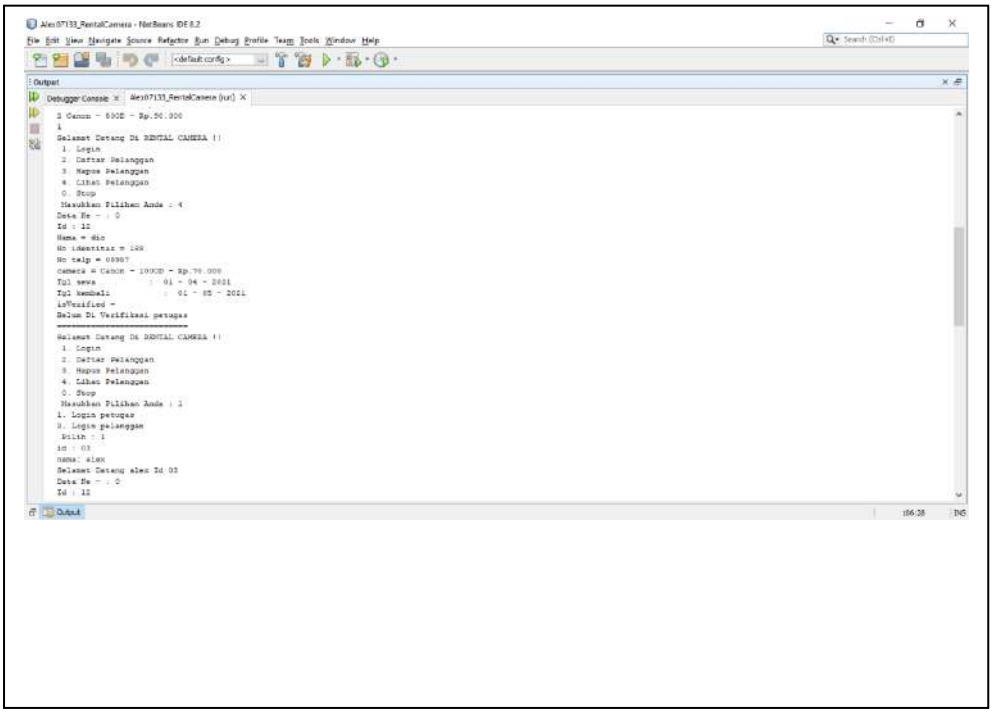
Source Code

```
package Alex07133_Main;
import Alex07133_Entity.Alex07133_CameraEntity;
import
Alex07133_Entity.Alex07133_DaftarPelangganEntity;
import
Alex07133_Model.Alex07133_DaftarPelangganModel;
import
Alex07133_Controller.Alex07133_PetugasController;
import Alex07133_Controller.Alex07133_AllObjectModel;
import
Alex07133_Controller.Alex07133_PelangganController;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;

public class Alex07133_main {

    private static Alex07133_PetugasController
petugasModel = new Alex07133_PetugasController();
    private static Alex07133_PelangganController
pelangganModel = new Alex07133_PelangganController();
    private static Scanner input = new Scanner
(System.in);

    public static void main(String[] args){
        Alex07133_main data = new Alex07133_main();
        data.viewMenu();
    }

    void viewMenu(){
        int loop = 0;
        int pilih = 0;
        int pilLogin = 0;
        petugasModel.datapetugas();
        do{
           System.out.print("Selamat Datang Di RENTAL
CAMERA !!" +
                            "\n 1. Login" +
                            "\n 2. Daftar Pelanggan"
+
```

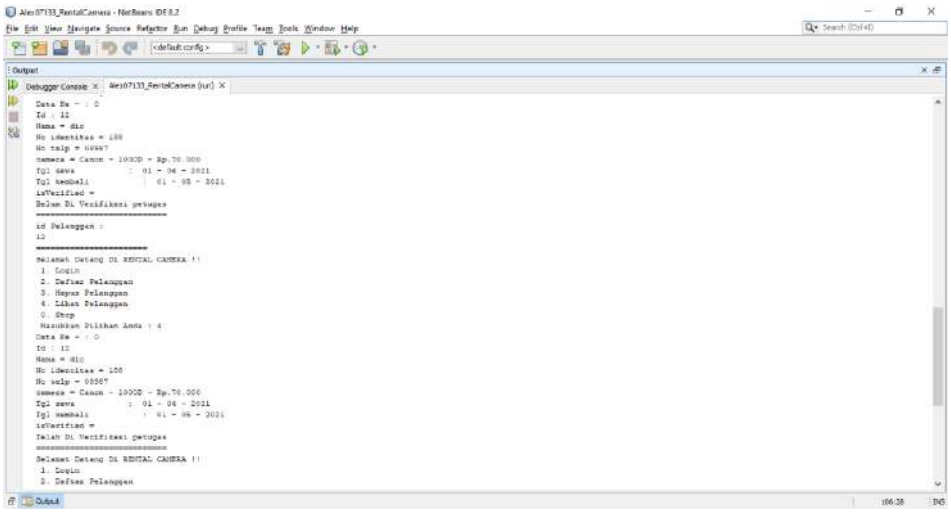
Tugas Resmi

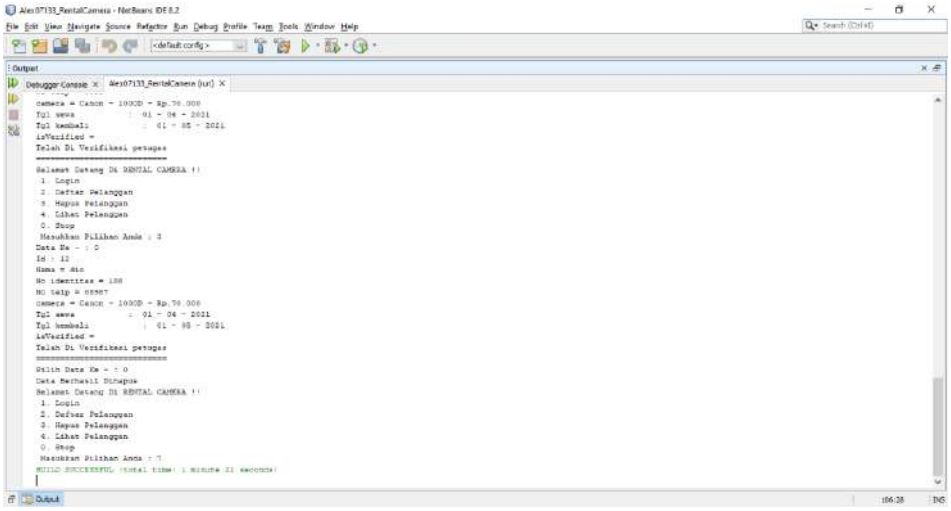
Screenshot





Tugas Resmi


Alex07133_RentalCamera - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Output
Debugger Console
Alex07133_RentalCamera (run)
Data Ke - : 0
Id : 12
Nama = dio
No identitas = 188
No telp = 08987
camera = Canon - 1000D - Rp.70.000
Tgl sewa : 01 - 04 - 2021
Tgl kembali : 01 - 05 - 2021
isVerified =
Belum Di Verifikasi petugas
id Pelanggan :
12
Selamat Datang Di RENTAL CAMERA !!
1. Login
2. Daftar Pelanggan
3. Hapus Pelanggan
4. Lihat Pelanggan
0. Stop
Masukkan Pilihan Anda : 4
Data Ke - : 0
Id : 12
Nama = dio
No identitas = 188
No telp = 08987
camera = Canon - 1000D - Rp.70.000
Tgl sewa : 01 - 04 - 2021
Tgl kembali : 01 - 05 - 2021
isVerified =
Telah Di Verifikasi petugas
Selamat Datang Di RENTAL CAMERA !!
1. Login
2. Daftar Pelanggan
Output
186:28
INS


Alex07133_RentalCamera - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Output
Debugger Console
Alex07133_RentalCamera (run)
camera = Canon - 1000D - Rp.70.000
Tgl sewa : 01 - 04 - 2021
Tgl kembali : 01 - 05 - 2021
isVerified =
Telah Di Verifikasi petugas
Selamat Datang Di RENTAL CAMERA !!
1. Login
2. Daftar Pelanggan
3. Hapus Pelanggan
4. Lihat Pelanggan
0. Stop
Masukkan Pilihan Anda : 3
Data Ke - : 0
Id : 12
Nama = dio
No identitas = 188
No telp = 08987
camera = Canon - 1000D - Rp.70.000
Tgl sewa : 01 - 04 - 2021
Tgl kembali : 01 - 05 - 2021
isVerified =
Telah Di Verifikasi petugas
Pilih Data Ke - : 0
Data Berhasil Dihapus
Selamat Datang Di RENTAL CAMERA !!
1. Login
2. Daftar Pelanggan
3. Hapus Pelanggan
4. Lihat Pelanggan
0. Stop
Masukkan Pilihan Anda : 0
BUILD SUCCESSFUL (total time: 1 minute 21 seconds)
Output
186:28
INS

Tugas Resmi

Analisa

Pada main program, semua inputan terdapat di dalamnya lalu akan diteruskan ke Controller untuk memberi perintah ke Model, kemudian dari Model akan dikembalikan Controller untuk dikembalikan ke main program lagi agar menampilkan output program. Semua code pada main program tidak memanggil method di Model, tetapi method di Controller. Pada main program juga tidak terdapat ArrayList karena ArrayList berada di Model.

Terdapat enkapsulasi yang diterapkan pada class ini, yaitu private. Private memiliki sifat bahwa properti dan method hanya bisa diakses oleh class itu sendiri.

Terdapat penggunaan Exception Handling pada method register(), dimana Exception Handling bertujuan agar apabila terdapat kesalahan pengisian inputan program tidak akan langsung terhenti, melainkan akan menampilkan pesan error dan akan otomatis kembali ke tampilan pilihan menu, lalu bisa melakukan pengisian data ulang.

RESMI
MODUL 4

LABORATORIUM REKAYASA PERANGKAT LUNAK
FAKULTAS TEKNIK ELEKTRO DAN

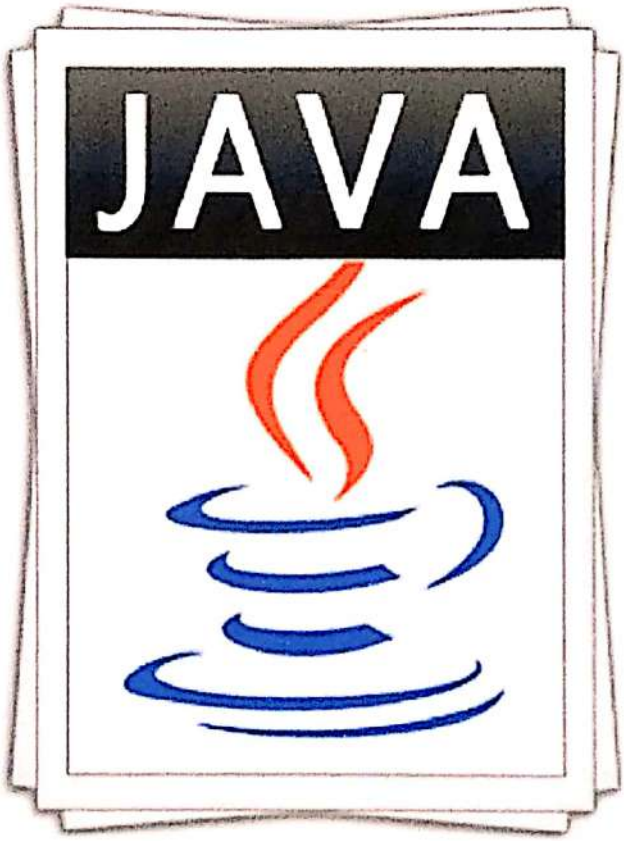
TEKNOLOGI INFORMASI

INSTITUT TEKNOLOGI ADHI TAMA

SURABAYA

2020

LAPORAN RESMI



PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
PERIODE VI - Tahun 2020/2021

NAMA	:	Alex Gilang S.
NPM	:	06.2019.1.07133
MODUL	:	4 (empat)

Dipindai dengan CamScanner

Tugas Resmi

Soal Praktikum

- *Terapkan GUI sesuai dengan project masing-masing.*
- *Terapkan CRUD di GUI project kalian.*
- *Terapkan JOptionPane ke GUI kalian.*

Langkah Pertama

Membuat package view untuk mengatur tampilan gui lalu menambahkan beberapa class salah satunya class Alex07133_PetugasGUI

Source Code

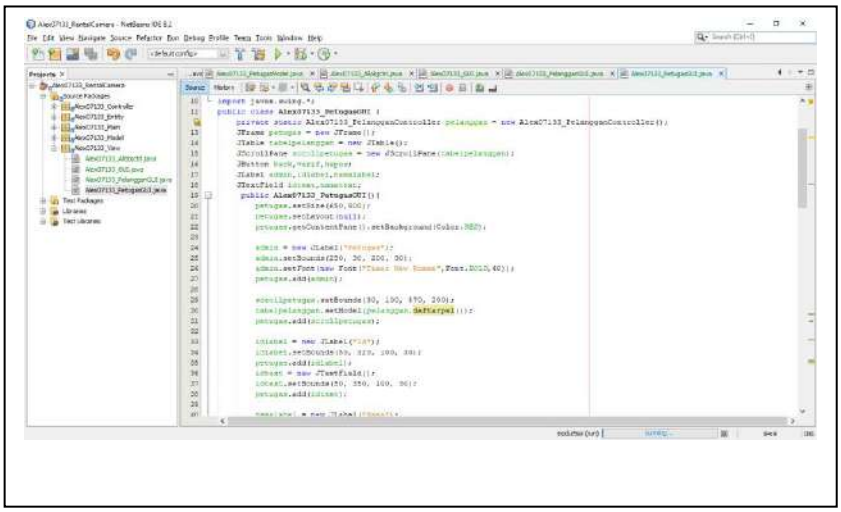
```
package Alex07133_View;
import Alex07133_Controller.Alex07133_AllObjectModel;
import
Alex07133_Controller.Alex07133_PelangganController;
import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.*;
public class Alex07133_PetugasGUI {
    private static Alex07133_PelangganController
pelanggan = new Alex07133_PelangganController();
    JFrame petugas = new JFrame();
    JTable tabelpelanggan = new JTable();
    JScrollPane scrollpetugas = new
JScrollPane(tabelpelanggan);
    JButton back,verif,hapus;
    JLabel admin,idlabel,namalabel;
    JTextField idtext,namatext;
     public Alex07133_PetugasGUI(){
        petugas.setSize(650,500);
        petugas.setLayout(null);

petugas.getContentPane().setBackground(Color.RED);

        admin = new JLabel("Petugas");
        admin.setBounds(250, 30, 200, 30);
        admin.setFont(new Font("Timer New
Roman",Font.BOLD,40));
```

Tugas Resmi

```
scrollpetugas.setBounds(30, 100, 570, 200);
        tabelpelanggan.setModel(pelanggan.daftarpel());
        petugas.add(scrollpetugas);

        idlabel = new JLabel("Id");
        idlabel.setBounds(50, 320, 100, 30);
        petugas.add(idlabel);
        idtext = new JTextField();
        idtext.setBounds(50, 350, 100, 30);
        petugas.add(idtext);

        namalabel = new JLabel("Nama");
        namalabel.setBounds(200, 320, 100, 30);
        petugas.add(namalabel);
        namatext = new JTextField();
        namatext.setBounds(200, 350, 100, 30);
        petugas.add(namatext);

        verif = new JButton("verif");
        verif.setBounds(350, 350, 100, 30);
        verif.setBackground(Color.GRAY);
        petugas.add(verif);

        back = new JButton("back");
        back.setBounds(50, 400, 100, 30);
        back.setBackground(Color.GRAY);
        petugas.add(back);

        hapus = new JButton("Hapus");
        hapus.setBounds(350, 410, 100, 30);
        hapus.setBackground(Color.GRAY);
        petugas.add(hapus);

        petugas.setLocationRelativeTo(null);
        petugas.setVisible(true);

petugas.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        back.addActionListener(new ActionListener(){
            @Override
            public void actionPerformed(ActionEvent
ae){
                petugas.dispose();
                Alex07133_GUI men = new
Alex07133_GUI();//nama objek men
            }
        });
```

Tugas Resmi

Screenshot



Analisa

Pada tahap Pembuatan class Alex07133_PetugasGUI ini berfungsi untuk mengsahilkan tampilan pada saat kita menjalankan program dan masuk ke dalam menu bagian admin

Tugas Resmi

Langkah Kedua

Membuat class Alex07133_PelangganGUI untuk memberikan tampilan GUI di dalam menu pelanggan

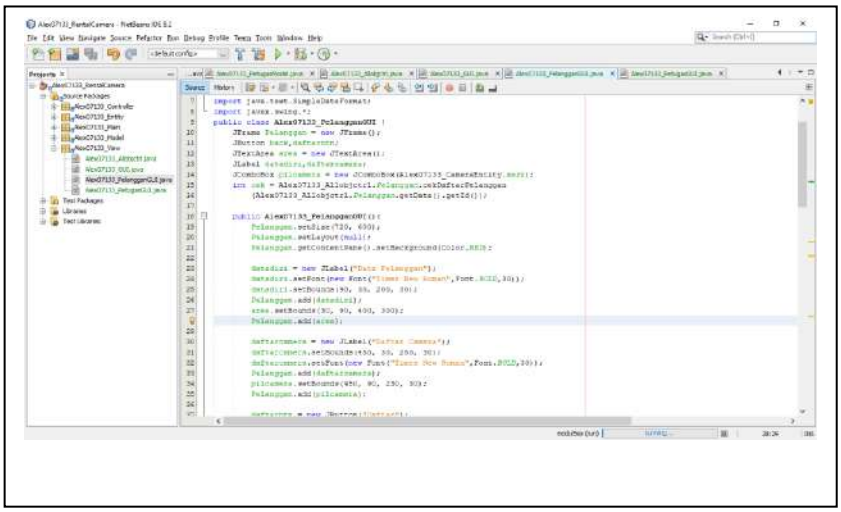
Source Code

```
package Alex07133_View;
import Alex07133_Entity.Alex07133_CameraEntity;
import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.text.SimpleDateFormat;
import javax.swing.*;
public class Alex07133_PelangganGUI {
    JFrame Pelanggan = new JFrame();
    JButton back,daftarbtn;
    JTextArea area = new JTextArea();
    JLabel datadiri,daftarcamera;
    JComboBox pilcamera = new
JComboBox(Alex07133_CameraEntity.merk);
    int cek =
Alex07133_Allobjctrl.Pelanggan.cekDaftarPelanggan

(Alex07133_Allobjctrl.Pelanggan.getData().getId());

    public Alex07133_PelangganGUI(){
        Pelanggan.setSize(720, 600);
        Pelanggan.setLayout(null);

Pelanggan.getContentPane().setBackground(Color.RED);

        datadiri = new JLabel("Data Pelanggan");
        datadiri.setFont(new Font("Timer New
Roman",Font.BOLD,30));
        datadiri.setBounds(90, 30, 200, 30);
        Pelanggan.add(datadiri);
        area.setBounds(30, 90, 400, 300);
        Pelanggan.add(area);

        daftarcamera = new JLabel("Daftar Camera");
        daftarcamera.setBounds(450, 30, 250, 30);
        daftarcamera.setFont(new Font("Times New
Roman",Font.BOLD,30));
        Pelanggan.add(daftarcamera);
        pilcamera.setBounds(450, 90, 230, 30);
```

```
        Pelanggan.add(pilcamera);

        daftarbtn = new JButton("Daftar");
        daftarbtn.setBounds(500, 200, 100, 30);
        daftarbtn.setBackground(Color.GRAY);
        Pelanggan.add(daftarbtn);

        back = new JButton("Back");
        back.setBounds(30, 500, 100, 30);
        back.setBackground(Color.GRAY);
        Pelanggan.add(back);

        Pelanggan.setVisible(true);
        Pelanggan.setLocationRelativeTo(null);

Pelanggan.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE
);
        if(cek==-1){ //cek data, apabila kita belum
daftar praktkum
            JOptionPane.showMessageDialog(null, "Anda
Belum Daftar Camera", "Information",
JOptionPane.INFORMATION_MESSAGE);
        }else if(cek==-2){
            JOptionPane.showMessageDialog(null,"Anda
Belum Daftar Camera", "Information",
JOptionPane.INFORMATION_MESSAGE);
        }else{
            area.setText(datapelanggan());
        }
        daftarbtn.addActionListener(new
ActionListener(){
        @Override
        public void actionPerformed(ActionEvent ae){
            int indexcamera =
pilcamera.getSelectedIndex();

Alex07133_Allobjctrl.Pelanggan.Alex07133_DaftarCamera

(indexcamera,Alex07133_Allobjctrl.Pelanggan.getData(),
false);
            area.setText(datapelanggan());
        }
        });

        back.addActionListener(new ActionListener(){
            @Override
```

Tugas Resmi

Screenshot



Analisa

Padal class Alex07133_PelangganGUI disni kita membuat tampilan gui pada menu ketika kita masuk menu masuk sebagai praktikan

Tugas Resmi

Langkah Ketiga

Membuat class Alex07133_GUI untuk memberikan tampilan GUI di menu awal

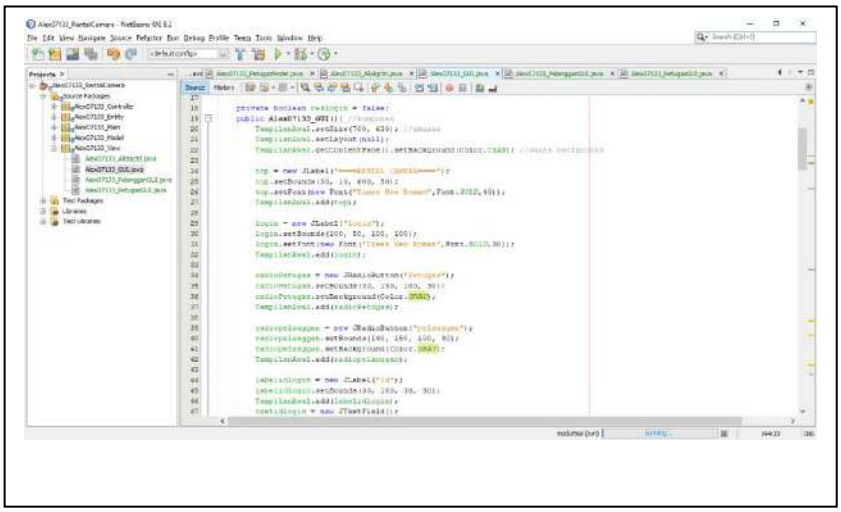
Source Code

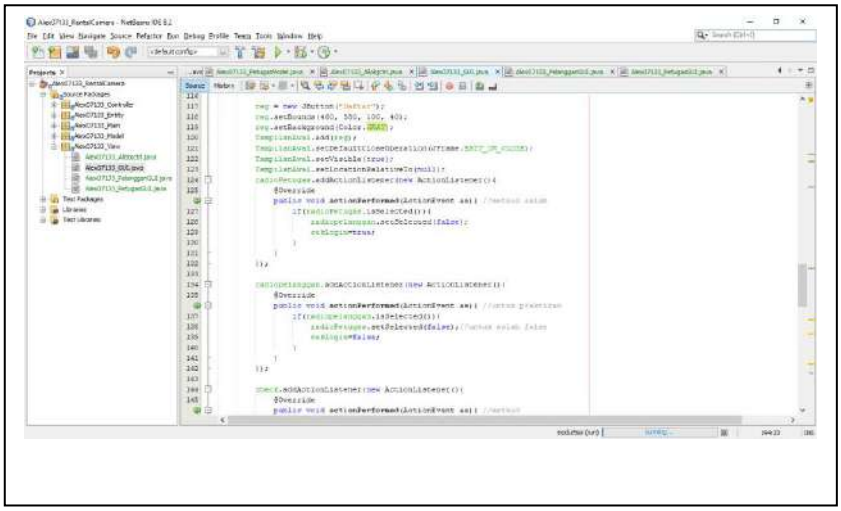
```
 package Alex07133_View;
import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Date;
import javax.swing.*;
public class Alex07133_GUI {
    JFrame TampilanAwal = new JFrame();//tampilan awal
    JLabel login,daftar,top;
    JRadioButton radioPetugas,radiopelanggan; //tombol
pilihan
    JTextField
textidlogin,textnamalogin,textalamat,textnoidentitas,te
xtnotelp,texttglsewa,

texttglkembali,textiddaftar,textnamadaftar;
    JLabel
labelidlogin,labelnamalogin,labelalamat,labelnoidentita
s,labelnotelp,labeltglsewa,

labeltglkembali,labeliddaftar,labelnamadaftar;
    JButton check,reg; //tombol

    private boolean ceklogin = false;
    public Alex07133_GUI(){ //komponen
        TampilanAwal.setSize(700, 630); //ukuran
        TampilanAwal.setLayout(null);

TampilanAwal.getContentPane().setBackground(Color.CYAN)
; //warna background

        top = new JLabel("====RENTAL CAMERA====");
        top.setBounds(50, 10, 600, 50);
        top.setFont(new Font("Times New
Roman",Font.BOLD,40));
        TampilanAwal.add(top);

        login = new JLabel("login");
        login.setBounds(100, 50, 100, 100);
        login.setFont(new Font("Times New
```

```
 TampilanAwal.add(top);

        login = new JLabel("login");
        login.setBounds(100, 50, 100, 100);
        login.setFont(new Font("Times New
Roman",Font.BOLD,30));
        TampilanAwal.add(login);

        radioPetugas = new JRadioButton("Petugas");
        radioPetugas.setBounds(30, 150, 100, 30);
        radioPetugas.setBackground(Color.GRAY);
        TampilanAwal.add(radioPetugas);

        radiopelanggan = new JRadioButton("pelanggan");
        radiopelanggan.setBounds(140, 150, 100, 30);
        radiopelanggan.setBackground(Color.GRAY);
        TampilanAwal.add(radiopelanggan);

        labelidlogin = new JLabel("Id");
        labelidlogin.setBounds(30, 180, 30, 30);
        TampilanAwal.add(labelidlogin);
        textidlogin = new JTextField();
        textidlogin.setBounds(30, 210, 200, 30);
        TampilanAwal.add(textidlogin);

        labelnamalogin = new JLabel("Nama");
        labelnamalogin.setBounds(30, 240, 100, 30);
        TampilanAwal.add(labelnamalogin);
        textnamalogin = new JTextField();
        textnamalogin.setBounds(30, 270, 200, 30);
        TampilanAwal.add(textnamalogin);

        check = new JButton("check");
        check.setBounds(90, 320, 100, 40);
        check.setBackground(Color.GRAY);
        TampilanAwal.add(check);

        daftar = new JLabel("Daftar");
        daftar.setFont(new Font("Times New
Roman",Font.BOLD,30));
        daftar.setBounds(440, 50, 150, 100);
        TampilanAwal.add(daftar);

        labeliddaftar = new JLabel("Id");
        labeliddaftar.setBounds(400, 150, 30, 30);
        TampilanAwal.add(labeliddaftar);
```

Tugas Resmi

Screenshot





Analisa

Padal class Alex07133_GUI disni kita membuat tampilan gui pada menu awal ketika kita menjalankan program

Tugas Resmi

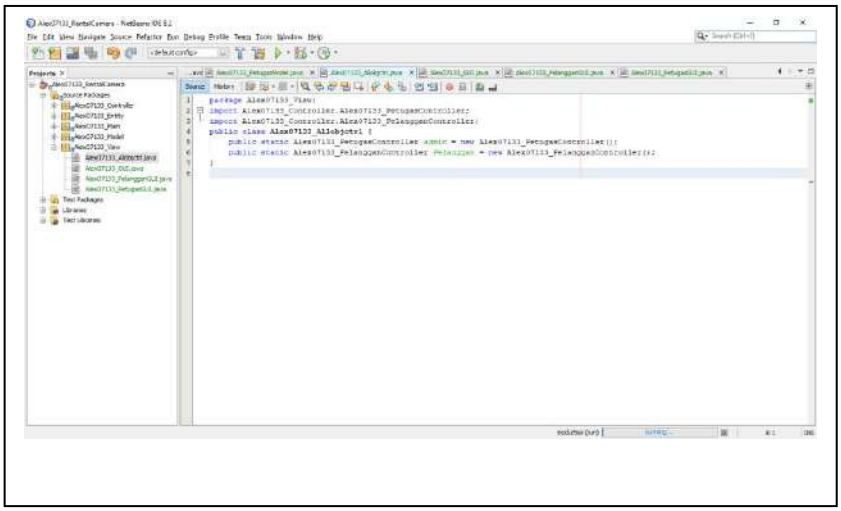
Langkah Keempat

Membuat class Alex07133_Allobjctrl untuk menggabungkan class GUI agar terhubung ke dalam Controller

Source Code

```
package Alex07133_View;
import
Alex07133_Controller.Alex07133_PetugasController;
import
Alex07133_Controller.Alex07133_PelangganController;
public class Alex07133_Allobjctrl {
    public static Alex07133_PetugasController admin =
new Alex07133_PetugasController();
    public static Alex07133_PelangganController
Pelanggan = new Alex07133_PelangganController();
}
```

Screenshot



Analisa

Padal class Alex07133_Allobjctrl disni kita membuat tampilan gui supaya dapat terhubung ke kontroller

Tugas Resmi

Langkah Kelima

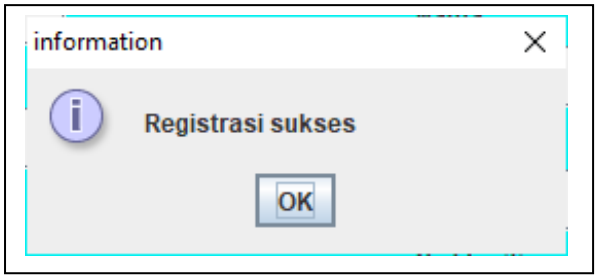
Menambahkan JOptionPane pada GUI

Source Code

```
@Override
            public void actionPerformed(ActionEvent
ae){ //method
                if(ceklogin==true){
                    try{

Alex07133_Allobjctrl.admin.datapetugas(); // login
untuk petugas

Alex07133_Allobjctrl.admin.login(textidlogin.getText(),
textnamalogin.getText());
                        String nama =
Alex07133_Allobjctrl.admin.Alex07133_PetugasEntity().ge
tnama(); //memasukkan id,nama, jika benar

JOptionPane.showMessageDialog(null, "Selamat datang
"+nama,"information", JOptionPane.INFORMATION_MESSAGE);
                        Alex07133_PetugasGUI petugas =
new Alex07133_PetugasGUI();
                        TampilanAwal.dispose();
                    }catch(Exception exception){ //jika
id/nama salah makan akan muncul pesan

JOptionPane.showMessageDialog(null, "id atau nama
salah", "information",
JOptionPane.INFORMATION_MESSAGE);
                    }
                        kosong();//method kosong, jika
datanya kosong
                }else{
                    try{ //login untuk pelanggan

Alex07133_Allobjctrl.Pelanggan.login(textidlogin.getTex
t(),textnamalogin.getText());
                       String nama =
Alex07133_Allobjctrl.Pelanggan.getData().getnama();
//memasukkan id,nama, jika benar

JOptionPane.showMessageDialog(null, "Selamat datang
"+nama,"information", JOptionPane.INFORMATION_MESSAGE);
                       Alex07133_PelangganGUI  pel =
```

Tugas Resmi

Screenshot



Analisa

Joptionpane disni saya gunakan ketika berhasil mendaftarkan pelanggan dan juga masuk pelanggan pada saat berhasil registrasi

Tugas Resmi

Langkah Keenam

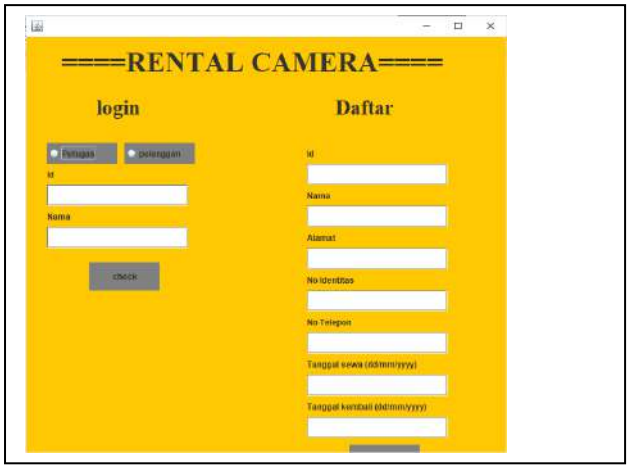
Menerapkan CRUD pada program GUI

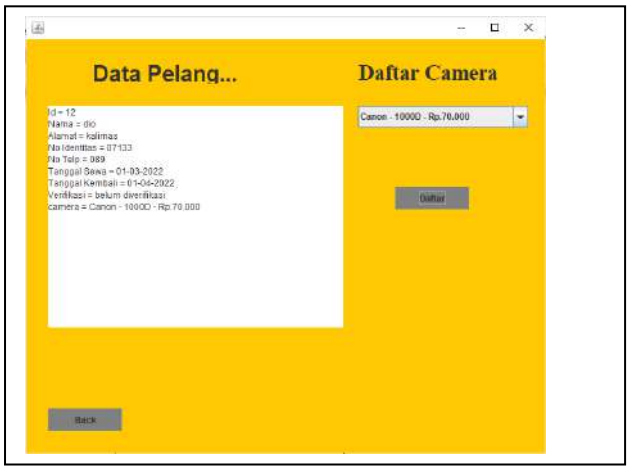
Source Code

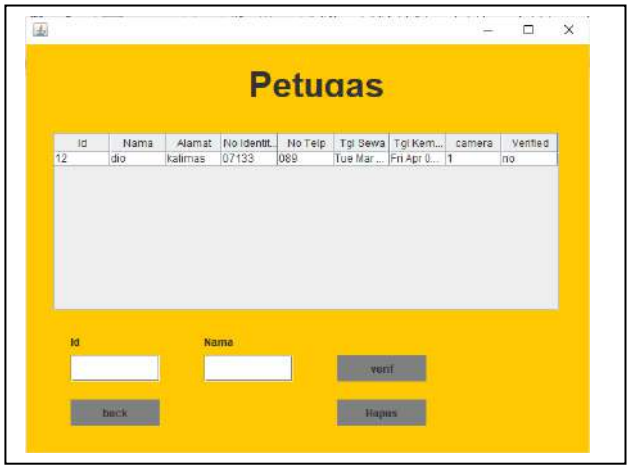
```
        verif.addActionListener(new ActionListener(){
            @Override
            public void actionPerformed(ActionEvent
ae){ //untuk verifikasi id, dan nama
                String id = idtext.getText();
                String nama = namatext.getText();
                int index =
Alex07133_AllObjectModel.daftarPelangganmodel.cekData(i
d, nama); //dicek

Alex07133_Allobjctrl.admin.updateIsVerified(index,
Alex07133_Allobjctrl.Pelanggan.showDaftarPelanggan(inde
x)

.getIndexCamera(),Alex07133_Allobjctrl.Pelanggan.showDa
ftarPelanggan(index).getPelanggan());

tabelpelanggan.setModel(pelanggan.daftarpel());
            }
        });

        hapus.addActionListener(new ActionListener(){
            @Override
            public void actionPerformed(ActionEvent
ae){ //untuk verifikasi id, dan nama
                String id = idtext.getText();
                int index =
Alex07133_AllObjectModel.daftarPelangganmodel.alldatape
langgan().size(); //dicek

Alex07133_Allobjctrl.Pelanggan.deleteDataPelanggan(inde
x);

tabelpelanggan.setModel(pelanggan.daftarpel());
            }
        });

        tabelpelanggan.addMouseListener(new
MouseAdapter(){
            @Override
```

Tugas Resmi

Screenshot







Analisa

Proses CRUD dapat di akses melalui GUI dengan tampilan dan support untuk mouse klik (ActionListener)

PRAKTIKUM
MODUL 1

LABORATORIUM REKAYASA PERANGKAT LUNAK
FAKULTAS TEKNIK ELEKTRO DAN
TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI ADHI TAMA
SURABAYA
2020

LAPORAN PRAKTIKUM

PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
PERIODE VI - Tahun 2020/2021

NAMA	:	Alex Gilang S.
NPM	:	06.2019.1.07133
MODUL	:	1 (Satu)

Dipindai dengan CamScanner

LAPORAN PENDAHULUAN

PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
PERIODE VI - Tahun 2020/2021

NAMA	:	Alex Gibang. S
NPM	:	06.2019.1.07133
MODUL	:	I (satu)

Tugas Pertanyaan

Modul I
PENGENALAN JAVA

kerjakan langkah-langkah berikut dan buktikan hasilnya dengan cetak screenshot saat program dijalankan pada laporan tugas pendahuluan dan tunjukkan hasil-nya pada saat praktikum

Pada bab ini kita akan membuat program sistem informasi laboratorium sederhana dengan fitur daftar dan login.

Langkah-langkah pengerjaan:

1. Buatlah file baru dengan nama NPM06992 Soal pendahuluan 1 (NPM 06992 sesuaikan dengan NPM Pratikan)
2. Pastikan file yang telah terbuat, nama kelas dan nama file harus sama.

```
1. Public class NPM06992SoalPendahuluan1 {
2. }
```

3. Buatlah sebuah Void Main di dalam class yang sudah dibuat tersebut.

```
1. Public class NPM06992SoalPendahuluan1 {
2.  Public static Void main (string [] args) {
3.  }
4. }
```

4. Buat class Aslab Entity yang digunakan untuk menyimpan data asistenlab dengan kode sebagai berikut:

Class AslabEntity

```
1. Package NPM06992 soal Pendahuluan1;
2. Import java.util.Date;
3. Public class Aslab Entity {
4.  String npm, nama, pasword, notele;
5.  Date Tanggal lahir;
```

Tugas Pertanyaan

```
6. Public AsistenEntity (String npm, String nama, String Password,
   String notelp, Date tanggallahir) {
7.   this.npm = npm;
8.   this.nama = nama;
9.   this.Password = Password;
10.  this.notelp = notelp;
11.  this.Tanggallahir = Tanggallahir;
12. }
13. String getNpm () {
14.     return this.npm;
15. }
16. String getNama () {
17.     return this.nama;
18. }
19. String getPassword () {
20.     return this.Password;
21. }
22. String getNoTelp () {
23.     return this.notelp;
24. }
25. Date getDate () {
26.     return this.Tanggallahir;
27. } }
```

5. Buatlah class Pada Praktikum Entity yang digunakan untuk menyimpan data Praktikum dengan kode sebagai berikut :

Class Praktikum Entity

```
1. Public class PraktikumEntity {
2. static String [] Praktikum = {"PBO", "SO", "PETRU"};
3. }
```

6. Buatlah Class Praktikan Entity yang digunakan untuk menyimpan data Praktikan dengan kode sebagai berikut :

Tugas Pertanyaan

Class PraktikanEntity

```
1. Package NPM06592Soal Pendahuluan 1;
2. Import java.Util.Date;
3. Public class Praktikan Entity {
4.    String npm, nama, Password, notelp, kelas;
5.    Date Tanggal lahir;
6.    Int Indexpraktikum;
7. Int Indexaslab;
8.
9. Public praktikanEntity (String npm, String nama, String
Password, String notelp, Date tangal lahir, Int Indexprak,
int Indexaslab, String kelas ){
10.    this.npm=npm;
11.    this.nama=nama;
12.    this.Password=Password;
13.    this.notelp=notelp;
14.    this.Tanggallahir=Tanggallahir;
15.    this.Index Praktikum=Indexprak;
16.    this.Index aslab=Indexaslab;
17.    this.kelas=kelas;
18. }
19. String getNPM(){
20.    return this.npm;
21. }
22. String getNama(){
23.    return this.nama;
24. }
25. String getPassword(){
26.    return this.password;
27. }
28. String getNoTelp(){
29.    return this.notelp;
```

Tugas Pertanyaan

```
30. }
31. Date getDate() {
32  return this.TanggalLahir;
33 }
34. int getIndexPraktikum () {
35. return IndexPraktikum;
36. }
37. }
```

7. Pada main class yang sudah di buat sebelumnya tulis kode seperti berikut:

Class main

```
1. Package NPM06992soalPendahuluan1;
2.
3. Import java.text.SimpleDataFormat;
4. Import java.util.Arraylist;
5. Import java.util.Date;
6. Import java.util.Scanner;
7.
8. Public class NPM06992soalPendahuluan1 {
9.  Static Array list <asLabEntity> dataAslab = new Arraylist();
10. Static Array list <PraktikanEntity> dataPraktikan = new Arraylist();
11. Static Scanner Input = new Scanner (System.In);
12.
13. Public static Void main (String [] args) {
14. Syeem.out.Println ("selamat Datang di Laboratorium
INFORMATIKA ITATS!! ");
15. Int Pil;
16. DataAslab();
17. Do {
18. System.out.Println ("1. Daftar ");
19. System.out.Println ("2. login ");
20. System.out.Println ("3. Exit ");
```

Tugas Pertanyaan

```
System.out.println ("Pilih = ");
Pil = Input. nextInt ();
Switch (Pil) {
  Case 1 :
        daftar ();
        break;
  Case 2 :
  System.out. Print ("NPM = ");
  String npm = input .next ();
  System. out. Print ("Password = ");
  String Pass = Input .next();
  View (npm, Pass);
        break;
  }
  } while (Pil != 3);
}
Static void daftar () {
  System. out. Print ("Input NPM = ");
  String npm = Input. next ();
  system. out. print ("Input Nama = ");
  String nama = Input. next();
  String .out. print ("Input Password = ");
  String Pass = Input. next();
  system. out. print ("Input no Telp = ");
  String no telp = Input. next();
  Sytem. out. Print ("Input tangal lahir (dd/mm/yyyy)
    = ");
  Date tanggal = new Date (input. next());
  System. out. Print ("Kelas = ");
  String kelas = Input. next();
  For (int i=0; i < dataAslab. siz(); i++) {
  System. out. Println (i+". "+dataAslab. get(i). nama);
```

Tugas Pertanyaan

```
52. }
53. System.out.print ("Pilih Aslab : ");
54. Int Index Aslab= Input.nextInt ();
55. for(int i=0; i< PraktikumEntity.Praktikum.lenght; i++);
56. System.out.Println (i+". "+ Praktikum Entity.Praktikum
    [i]);
57. }
58. System.out.print ("pilih Praktikum : ");
59. Int indexprak= Input.next Int ();
60. data Praktikan.add (new Praktikan Entity (npm, nama, pass,
    notelp, tanggal, Index Prak, Index Asiab, kelas));
61. }
62. Static void View (String npm, String password)
63. for(Int i=0; i< data Praktikan.size (); i++){
64. If(npm.equals (data Praktikan.get(i).get NPM ()) &&
    Password.equals (datapraktikan.get (i).get Password())) {
65. System.out.Println ("NPM = "+data prak tikan.get(i)
    .get NPM());
66. System.out.Println ("nama= "+ data Praktikan.get(i)
    .get Nama());
67. System.out.Print Ln ("Tanggal lahir = "+new simple Date
    format ("dd-MM-yyyy").format (data Praktikum.get(i)
    .get Date()));
68. System.out.Print Ln ("No Telp= "+datapraktikan.get (i)
    .get notelP()));
69. System.out.Print Ln ("nama Asisten = "+ data Aslab.get
    (data Praktikan .get(i). Index asleb).get nama()));
```

Tugas Pertanyaan

```
70.  System.out.Println("Praktikum = " +
PraktikumEntity.Praktikum [data praktikan.
get(i).index praktikum]);
71. } else {
72.  System.out.Println(" nama atau Password
salah");
73.  }
74. }
75. }
76. static void DataAslab () {
77. String npmaslab [] = {"01", "02", "03", "04", "05", "06", "07"
};
78. String password Aslab [] = {"01", "02", "03", "04", "05", "06",
"07"};
79. String nama Aslab [] = {"michael", "fadlan", "odila",
fitria", "sita", "eric", "fernanda"};
80. String notelpAslab[] = {"01", "02", "03", "04", "05",
06", "07"};
81. String tgl lahir Aslab [] = {"01/05/2000", "01/12/2000",
03/3/2000", "23/6/2000", "11/3/2000", "13/9/2000", "22/12/
2000"};
82. for (int i=0 ; i < npm Aslab.length ; i++) {
83. data Aslab.Add (new Aslab Entity (npmAslab [i], nama Aslab
[i], Password Aslab [i],
84. notelp Aslab [i], new Date (tgl lahir Aslab [i]) ));
85.  }
86. }
87. }
8. Jalankan dan periksa kembali programnya.
```

Tugas Praktikum

Soal Praktikum

Membuat Class Diagram

Langkah Pertama

Membuat Class anggota

Source Code

```
package alex07133_rentalcamera;

public class Alex07133_anggota {
    String
Alex07133_id_anggota,Alex07133_nama,Alex07133
_notelp;

    public Alex07133_anggota(String
Alex07133_id_anggota,String
Alex07133_nama,String Alex07133_notelp){

this.Alex07133_id_anggota=Alex07133_id_anggot
a;
```

Screenshot

```
0.ASTRID
1.SABRINA
2.ALEX
Petugas Rental = 2
Data berhasil disimpan
```

Analisa

Digunakan untuk data anggota rental

Tugas Praktikum

Langkah Kedua

Membuat class kamera

Source Code

```
package alex07133_rentalcamera;


public class Alex07133_camera {
    static String[]
Alex07133_CAMERA_CANON={"CANON 3000D","CANON
1500D","CANON 200D","CANON 750D"};
}
```

Screenshot

```
0. CANON 3000D
1. CANON 1500D
2. CANON 200D
3. CANON 750D
Pilih Kamera = 2
```

Analisa

Untuk Pemilihan kameranya

Tugas Praktikum

Langkah Ketiga

Membuat class pelanggan

Source Code

```
package alex07133_rentalcamera;
import java.util.Date;

public class Alex07133_pelanggan {
    String
Alex07133_id_pelanggan,Alex07133_nama,Alex071
33_notelp;
    Date Alex07133_tglpinjam,
Alex07133_tglkembali;
    int Alex07133_indexkamera;
    int Alex07133_indexpetugas;

    public Alex07133_pelanggan(String
```

Screenshot

```
DATA KE 0
ID = 12
Nama = rayhan
Notelp = 08122359138
Kamera yang Dipinjam = CANON 200D
Tanggal Peminjaman Kamera (dd/mm/yyyy) = 06-11-2011
Tanggal Pengembalian Kamera (dd/mm/yyyy) = 06-12-2012
Petugas Rental = ALEX
```

Analisa

Untuk format penyimpanan data pelanggan

Tugas Praktikum

Langkah Ke-empat

Membuat class RentalCamera

Source Code

```
package alex07133_rentalcamera;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.Scanner;

public class Alex07133_RentalCamera {
    static
ArrayList<Alex07133_anggota>dataPetugas = new
ArrayList();
    static
ArrayList<Alex07133_pelanggan>dataPelanggan =
new ArrayList();
```

Screenshot

```
Data Rental
Pilihan :
1. Input Data
2. Lihat Data
3. Ubah Data
4. Hapus Data
5. Exit
Pilih : 1
Input ID Pelanggan : 12
Input Nama Pelanggan : rayhan
Input Notelp Pelanggan : 08122359138
Input Tanggal Peminjaman Kamera (mm/dd/yyyy) = 18/11/2012
Input Tanggal Pengembalian Kamera (mm/dd/yyyy) = 19/11/2011
0. CANON 3000D
1. CANON 1500D
2. CANON 200D
3. CANON 750D
```

Analisa

Untuk membuat menu dan output input data

Laporan Sementara

Soal Praktikum

Buatlah aplikasi sesuai project masing-masing praktikan dengan menerapkan:

1. Terapkan Object Array (ArrayList) pada Project masing-masing.

2. Buatlah CRUD (Create, Read, Update, Delete) pada project masing masing.

3. Terapkan Entity pada Object Array.

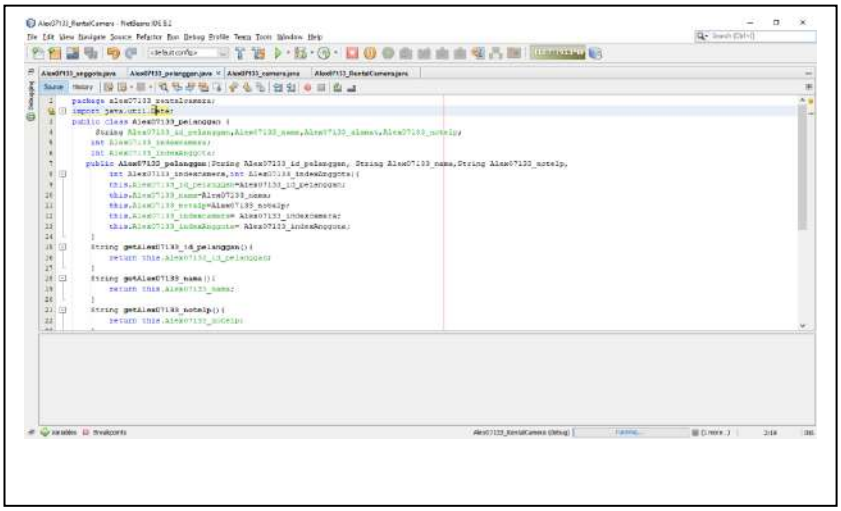
Langkah Pertama

Membuat beberapa class

Source Code

```
package alex07133_rentalcamera;
import java.util.Date;
public class Alex07133_pelanggan {
     String
Alex07133_id_pelanggan,Alex07133_nama,Alex071
33_alamat,Alex07133_notelp;
    int Alex07133_indexcamera;
    int Alex07133_indexAnggota;
   public Alex07133_pelanggan(String
Alex07133_id_pelanggan, String
Alex07133_nama,String Alex07133_notelp,
        int Alex07133_indexcamera,int
Alex07133_indexAnggota){
```

Screenshot



Analisa

Dari sini kita hanya membuat kerangkanya setelah itu ke main class untuk eksekusi program

Laporan Sementara

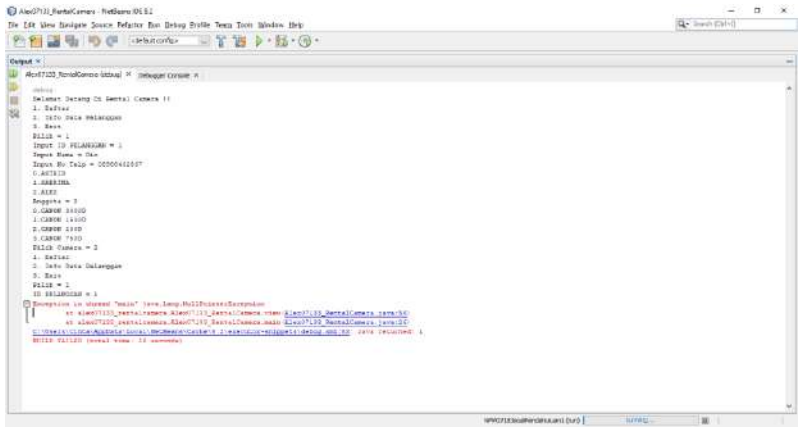
Langkah Kedua

setelah membuat class tadi, sekarang membuat main class

Source Code

```
package alex07133_rentalcamera;
import java.util.ArrayList;
import java.util.Scanner;
public class Alex07133_RentalCamera {
    static
ArrayList<Alex07133_anggota>dataAnggota = new
ArrayList();
    static
ArrayList<Alex07133_pelanggan>dataPelanggan =
new ArrayList();
    static Scanner input = new
Scanner(System.in);
    private static Object
```

Screenshot



Analisa

Program ini hanya menginputkan atau mendaftarkan sebuah data pelanggan yang mau menyewa kamera

PRAKTIKUM
MODUL 2

LABORATORIUM REKAYASA PERANGKAT LUNAK
FAKULTAS TEKNIK ELEKTRO DAN
TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI ADHI TAMA
SURABAYA
2020

LAPORAN PRAKTIKUM

Java

PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
PERIODE VI - Tahun 2020/2021

NAMA	:	Alex Gilang S.
NPM	:	06-2019.1.07133
MODUL	:	2 (dua)

Dipindai dengan CamScanner

LAPORAN PENDAHULUAN

PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
PERIODE VI - Tahun 2020/2021

NAMA : Alex Gilang S.
NPM : 06.2019.1.07133
MODUL : 2 (Dua)

Tugas Pertanyaan

MODUL II

RELASI ANTAR OBJECT DASAR

Kerjakan langkah-langkah berikut dan buktikan hasilnya dengan cetak screenshot saat program dijalankan pada laporan Tugas Pendahuluan dan ditunjukkan hasilnya pada saat praktikum

Aturan pengerjaan = setiap nama class harus diakhiri NPM dengan format nama Class NPM. Contoh : Main 07083.

Langkah-langkah pengerjaan :

1. Buat Proyek Java Application baru
2. Buat package Entity dan package model
3. Buat class Abstract dengan nama mahasiswaAbstractEntity Seperti code di bawah ini pada package Entity:

Class Mahasiswa Abstract Entity

```
1. Package Entity;
2. Import Java.Util.Date;
3. Public Abstract class MahasiswaAbstractEntity {
4. Protected String npm, Password, nama, no_telp;
5. protected Date tglLahir;
6. Public MahasiswaAbstractEntity (String npm, String Password,
   String nama, String no_telp, Date tglLahir){
7. this.npm= npm;
8. this.nama= nama;
9. this.Password = Password;
10 this.notelp = no_telp;
11. this.tglLahir= tglLahir; }
12. Public MahasiswaAbstractEntity () { }
13. Public abstract String getNpm ();
14. Public Void set Npm (String npm) { this.npm= npm ; }
15. Public String get Password () { return password ; }
16. Public Void Set Password (String password) {
    this.Password = Password ; }
```

Tugas Pertanyaan

17. Public String getNama() { return nama; }
18. Public Void setNama (String nama) { this.nama=nama;
    }
19. Public String getNo_telp() { return no_telp; }
20. Public Void setNo_telp (String no_telp) { this.no_telp
    = no_telp; }
21. Public Date getTglLahir() { return tglLahir; }
22. Public Void setTglLahir (Date tglLahir) {
    this.tglLahir = tglLahir; }
23. } }

4. Kemudian buatlah kelas Interfaces dengan nama class ModelInterfaces. Jika sudah, tuliskan code seperti dibawah pada package model:

Class ModelInterfaces

1. Package Model;
2. Public interface ModelInterfaces {
3. Public Void View ();
4. Public int cekData (String nam, String password); }

5. Buatlah Class AslabEntity dengan code sebagai berikut pada package Entity:

Class AslabEntity

1. Package Entity;
2. Import java.Util.Date;
3. Public class AslabEntity extends Mahasiswa Abstract Entity {
4. Private String laboratorium;
5. Public AslabEntity (String npm, String password, String nama, String no_telp, Date tglLahir, String lab) {
6. super (npm, password, nama, no_telp, tglLahir);
7. this.laboratorium = lab; }

Tugas Pertanyaan

```
8. @Override
9. Public string getnpm () {return npm ; }
10. Public string getLaboratorium () { return laboratorium;}
11. Public void setLaboratorium (string laboratorium) {
12. this.laboratorium = laboratorium; } }
```

6. Buatlah class Baru dengan nama Aslab Model pada Package model dengan code sebagai berikut:

Class Aslab Model

```
1. Package model;
2. Import modul2npm. Aslab Entity;
3. Import java.text. SimpleDate format;
4. import java.util. Array'List ;
5. Public class AslabModel implements ModelInterfaces {
6. Private ArrayList <Aslab Entity> aslab Entity ArrayList;
7. Public Aslab Model () {
8. aslab Entity Array List = new Array List <Aslab Entity> (); }
9. Public void Insert Aslab (Aslab Entity aslab) {
10. aslab Entity Array List .add (aslab); }
11. @ Override
12. Public void View () {
13. for (Aslab Entity aslab Entity : aslab Entity Array List) {
14. System.Out. Print (aslab Entity .get Npm());
15. System .out.Print (aslab Entity. get Nama());
16. System. Out.Print (aslab Entity. get Password ());
17. System. out. Print (aslab Entity. get No_telp());
18. System.out. Print. (new Simple Dateformat ("dd-MM-yyyy").
format (aslab Entity . get tgl Lahir ()));
19. System . out. Prin In ()); }
20 }
21 Public int Cek Data (string npm, string Password) {
```

Tugas Pertanyaan

```
22. int loop = 0;
23. for ( AslabEntity a : aslabEntityArrayList ){
24. if(aslabEntity.getNpm().equals(npm) && aslabEntity
.getPassword().equals(password)){ break;
25. } else { loop++; } }
26. return loop; }
27. public AslabEntity showDataAslab (int Index) {
return aslabEntityArrayList.get(index); }
28. }
```

7. Kemudian buatlah class baru dengan nama Praktikan Entity pada package Entity dengan Code seperti dibawah ini:

```
1. package Entity;
2. import Java.util.Date;
3. Public class Praktikan Entity extends Mahasiswa
Abstract Entity {
4. private String kelas;
5. Public Praktikan Entity (String npm, String Password,
String nama, String no_telp, Date tglLahir, String
kelas) {
6. super( npm, password, nama, no_telp, tglLahir);
7. this.kelas = kelas; }
8. @Override
9. Public String getNpm() { return npm; }
10. Public String getKelas() { return kelas; }
11. Public Void setKelas (String kelas) {
12 this.kelas = kelas; }
13. }
```

8. Buatlah class baru dengan nama PraktikanModel pada Package Model dengan code seperti dibawah ini:

Tugas Pertanyaan

Class Praktikan Model

```
1. Package model;
2. import modul2npm.Praktikan Entity
3. import Java.util.ArrayList;
4. Public class Praktikan Model implements ModelInterfaces
   {
5. Private ArrayList <Praktikan Entity> Praktikan Entity
   Arraylist;
6. Public Praktikan Model() {
7. Praktikan Entity ArrayList = new Array List <Praktikan
   Entity>(); }
8. Public void Insert( Praktikan Entity Praktikan Entity) {
9. Praktikan Entity ArrayList .add (Praktikan Entity); }
10. @ Override
11. Public Void View() {
12. For (Praktikan Entity praktikan Entity : Praktikan
   Entity ArrayList ) {
13. System.out.Println ("==========22222 ");
14. System.out.Println("NPM = "+ Praktikan Entity .getNpm
   ()+"\n nama : "+ Praktikan Entity .getNama()+"\n
   Password "+ Praktikan Entity .getPassword()+"\n No.telp"
   +Praktikan Entity.getNo_telp()+"\n Tanggal Lahir "+
   Praktikan Entity .get TglLahir());
15. System.out.Println("=========== ===== "); }
16. }
17. Public int cekData(String npm, String password) { int loop=0;
   }
18. While (!Praktikan Entity Array List .get (loop).getNpm().equals
   (npm) && !Praktikan Entity ArrayList.get(loop).getPassWord().
   equals (Password)) { loop ++; }
19. return loop; }
20. Public Praktikan Entity getPraktikan Entity ArrayList
```

Tugas Pertanyaan

(int index) {
21. return praktikanEntityArrayList.get(index); }
22. }

9. Buatlah class baru dengan nama PraktikumEntity pada Package Entity dengan code seperti dibawah:
Class PraktikumEntity
1. Package Entity
2. Public class PraktikumEntity {
3. Public static String nama[] = {"PBO", "SO", "PETRUK"};
4. Public static String periode; }

10. Buatlah class baru dengan nama DaftarPrakEntity pada Package Entity dengan code sebagai berikut:
Class DaftarPrakEntity
1. Package Entity
2. Public class DaftarPrakEntity {
3. Private PraktikanEntity pratikan;
4. Private boolean isVerified;
5. Private int indexPraktikum;
6. Public DaftarPrakEntity (int indexpraktikum, PraktikanEntity Pratikan, boolean isVerified) {
7. this.Pratikan = Pratikan;
8. this.isVerified = isVerified;
9. this.indexPraktikum = indexPraktikum; }
10. Public PraktikanEntity getPratikan () { return pratikan;
11. }
12. Public Void setPratikan (PraktikanEntity Pratikan) {
13. this.Pratikan = Pratikan; }
14. Public boolean isVerified () { return Isverifien; }
15. Public Void setVerified (boolean verifien) {
16. isVerified = verified; }

Tugas Pertanyaan

17. Public int getIndexPrak() {
18. return indexPraktikum; }
19. }

11. Buatlah class baru dengan nama Daftarprak model pada package Model dengan code seperti dibawah ini:

Class DaftarPrakModel

1. Package Model;
2. import modul2npm.DaftarPrakEntity;
3. import modul2npm.PraktikumEntity;
4. import java.text.ParseException;
5. import java.text.SimpleDateFormat;
6. import java.util.Date;
7. Public class DaftarprakModel implements ModelInterfaces {
8. Private ArrayList<DaftarPrakEntity> DaftarPrakArrayList;
9. Public DaftarPrakModel() {
10. daftarPrakArrayList = new ArrayList<DaftarPrakEntity>(); }
11. Public Void insertDataDaftarprak(DaftarprakEntity daftarPrak) {
12. daftarPrakArrayList.add(daftarPrak); }
13. Public ArrayList<DaftarPrakEntity> getDaftarPrakArrayList() { return daftarPrakArrayList; }
14. @Override
15. Public Void View() {
16. for (DaftarprakEntity daftarprak : daftarPrakArrayList) {
17. System.out.Println("==========");
18. System.out.Print("NPM= "+daftarPrak.getPraktikan

Tugas Pertanyaan

```
(). getNpm()+ "\n nama = "+daftarPrak.getPratikan().
getNama()+ "\n Password "+daftarPrak.getPratikan().
getPassword()+ "\n No.Telp : "+daftarPrak.getPratikan().
getNo_telp()+ "\n Tanggal Lahir="+ new SimpleDateFormat
("dd-mm-yyyy").format(daftarPrak.getPratikan().
getTglLahir()))
19.+ "\n Praktikum: "+ PraktikumEntity nama[daftar
Prak.getIndexPrak()] +"\n isVerified = ");
20. if (daftarPrak.isVerified()== false){
21 System.out.Println("Belum Diverifikasi Admin")
22. }else
23. System.out.Println("Telah Diverifikasi"); }
24. System.out.Println("==========================");}
25. }
26. Public int cekData(String npm, String password){
27 int loop=0;
28. if (daftarPrakArrayList.size()==0){
29. loop=-1 ; // data kosong
30. }else{
31. for(int i=0; i<daftarPrakArrayList.size(); i++){
32. if(daftarPrakArrayList.get(i).getpratikan().get
Npm().equals(npm)){ loop=i;
33. break; }else{
34. loop=-2; } }
35. } return loop; }
36. Public Void updateIsVerified(int index,Daftarprak
Entity daftarprakEntity){
37. daftarprakArrayList.set(index, daftarprakEntity);}
38. }
```

12. Kemudian isikan code seperti dibawah ini pada class main

Tugas Pertanyaan

Class Main

```
1. Package project;
2. Import mod 12npm ;
3. Import model2npm *;
4. Import Entity *;
5. Import java.util.Date;
6. Import java.text.SimpleDateFormat;
7. Import java.util.Scanner;
8. Public class main{
9. Private static Aslab model aslabModel = new AslabModel();
10. Private static scanner input=new Scanner(System.in);
11. Private static PraktikanModel praktikanModel= new
Praktikan model ();
12. Private DaftarprakModel daftarprak Model = new Daftar
Prak model();
13. Private static int pilprak;
14. static int cekpraktikan;
15. static int cek Aslab;
16. Public Static Void main (String[] args){ int loop=0;
17. dataAslab();
18. do {
19. int pilih=0;
20. System.out.print("Selamat Datang di Laboratorium Informa
21fika ITATS 11" + "\n 1. Login " + "\n 2.Daftar Pratikan" +
22" \n 0. STOP" + "\n Masukkan Pilihan Anda:"); Pilih=input
23.next int(); if(pilih == 1) {
24. int pilLogin=0;
25. System.out.print("1. Login Aslab "+"\n2. Login pratikan
" + "\n Pilih= ");
26. pilLogin =Input. next int();
27. IF(pilLogin==1) { loginAdmin(); }
28. else { login(); }
```

Tugas Pertanyaan

```
29. }else if (pilih ==2) { register();
30. }else if (pilih ==3) { aslabModelView();
31. }else if (pilih == 4) { }else { break };
32. }
33. } while (loop!=1); }
34. Static void dataAslab(){
35. String npmAslab[]={"01","02","03","04","05","06","07"};
36. String passwordAslab[]={"01","02","03","04","05","06","07"};
37. String namaAslab[]={"Michael","Alan","Edit","Fitria","Sita",
    "Eric","Fernanda"};
38. String noTelpAslab[]={"01","02","03","04","05","06","07"};
39. String tglLahirAslab[]={"04/05/2000","1/12/2000","03/3/
    2000","23/6/2000","11/12/2000","13/9/2000","22/12/2000"};
40. String laboratorium[]={"RPL","SO","BASPROG","BASPROG",
    "SO","RPL","RPL"};
41. for(int i=0; i<npmAslab.length; i++){
42. aslabModel.InsertAslab(new AslabEntity(npmAslab[i],
    PasswordAslab[i], namaAslab[i], noTelpAslab[i], new
    Date(tglLahirAslab[i]), laboratorium[i]));}
43. }
44. Static Void register(){
45. System.out.Print("input NPM = ");
46. String npm = input.next();
47. system.out.Print("Input Nama = ");
48. String nama = input.next();
49. system.out.Print("input Password = ");
50. String pass = input.next();
51. System.out.Print("input notelp = ");
52. String notelp = input.next();
53. System.out.print("Input tgl Lahir dd/mm/yyyy=
    ");
54. Date tanggal = new Date(input.next());
```

Tugas Pertanyaan

```
55. System.out.Print ("kelas :");
56. String kelas = input.next();
57. PraktikanModel insert(new praktikanEntity(npm, Pass,
    nama, notelp, tanggal, kelas)); }
58. Static void login() { system.out.Print("NPM: ");
59. String npm = input.next();
60. System.out.Print ("password : ");
61. String Password = input.next();
62 Cekpraktikan = PraktikanModel.Cek Data(npm, Pass
    word));
63. System.out.Println("Selamat datang " + praktikanModel
    .get praktikanEntity Array List(cekpraktikan).get
    nama());
64. int cekpraktikum = daftar prak model.Cek Data (Npm,
    Password); if(cekPraktikum == -1) {
65. System.out.Println ("Anda Belum dafta Praktikum");
66. daftar Prak(); }
67. else if (cekpraktikum == -2) {
68. System.out.Println ("Anda belum dafta praktikum");
69. daftarprak(); } else {
70. System.out.Println ("nama= " + daftarprak Model.Show
    Daftarprak(cekpraktikan).get Pratikan().get Nama());
71. System.out.Println("NPM = " + daftarprak Model show
    Daftarprak(cekpraktikan).get Pratikan().get Npm());
72. System.out.Println ("No Telp = " + daftarprakModel.Show
    Daftarprak(cekpraktikan).get Pratikan().getNo_telp());
73. System.out.Println ("praktikum= " + praktikum Entity.
    nama [daftarprakModel.showDaftarprak(cekpraktikan).get
    IndexPrak()]);
74. System.out.Println("is Verified = " + daftarprakModel.Show
    Daftar Prak(cekPraktikan).isVerified()); }
75. }
```

Tugas Pertanyaan

```
76. static void daftarprak (){
77. system.out.println("pilih praktikum = ");
78. for(int i=0; i<praktikumEntity.nama.length; i++){
79. System.out.println(i+". "+praktikumEntity.nama[i]); }
80. System.out.print ("pilih praktikum = "); pilprak=input.next int();
81. daftarprak Model.InsertDataDaftarPrak (new DaftarpraKEntity
(pilprak, praktikan Model.get Praktikan Entity Array List
(cekpraktikan), false)); }
82. Static Void Login Admin () {
83. System.out.print ("NPM = "); String npm = Input.next();
84. System.out.print ("Password = ");
85. String Password = input.next();
86. cekAslab = aslab Model.cekData (npm, Password);
87. System.out.Println("selamat Datang"+aslab Model.show Data
Aslab (cek Aslab).get Nama());
89 praktikan Model.view(); Update Isverified; }
90. Static Void update Isverified (){
91. System.out.Print ("NPM Praktikan = "); String npm= Input.next();
92. int Index = daftarprakModel.cekData(npm, null);
93. daftarprak Model.Update Isverified (index, new Daftar
PrakEntity (pilPrak, praktikan Model.get Praktikan
Entity Array List (index), true)); }
94. }
```

B. Jalankan dan periksa kembali programnya.

Tugas Praktikum

Soal Praktikum

Menerapkan Relasi Antar Objek

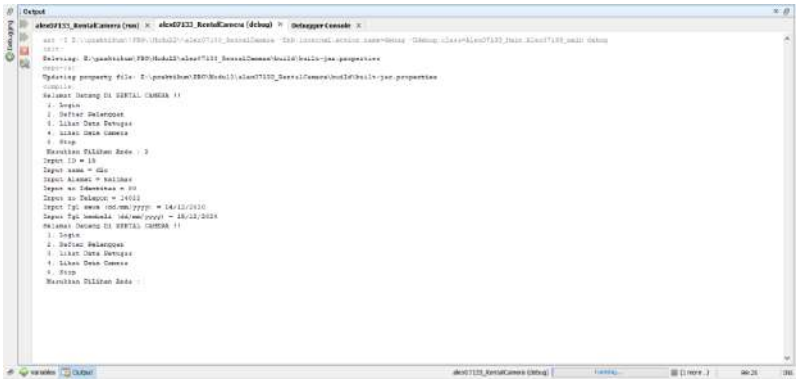
Langkah Pertama

Membuat Relasi Antar Objek

Source Code

```
package Alex07133_Entity;
import java.text.SimpleDateFormat;
import java.util.Date;
public class Alex07133_Rentalcamera {
protected String
id,nama,alamat,noIdentitas,notelp;

public Alex07133_Rentalcamera(String
id,String nama, //konstruktor,parameter
    String alamat, String noIdentitas,String
notelp){
    this.id=id:
```

Screenshot



Analisa

Antar class bisa saling terhubung

Tugas Praktikum

Langkah Kedua

Membuat pewarisan dan Enkapsulasi

Source Code

```
public class Alex07133_PelangganEntity
extends Alex07133_Rentalcamera
protected Date tgl_sewa,tgl_kembali;
//atribut
    public Alex07133_PelangganEntity(String
id,String nama,
    String alamat, String noIdentitas,String
notelp,Date tgl_sewa,Date tgl_kembali){

super(id,nama,alamat,noIdentitas,notelp);//me
manggil construktor dari superclass
        this.tgl_sewa=tgl_sewa;
        this.tgl_kembali=tgl_kembali;
```

Screenshot

```
 Masukkan Pilihan Anda : 2
Input ID = 15
Input nama = dio
Input Alamat = kalimas
Input no Identitas = 80
Input no Telepon = 14022
Input Tgl sewa (dd/mm/yyyy) = 14/12/2020
Input Tgl kembali (dd/mm/yyyy) = 15/12/2020
```

Analisa

Output yang dihasilkan dari melalui enkapsulasi protected string dan juga pewarisan dari kelas Alex07133_RentalCamera

Tugas Praktikum

Langkah Ketiga

Membuat Polimorphisme

Source Code

```
@Override   //pendefinisian ulang method yang
sama pada subclass
    public String getId() {
        return id;
    }
```

Screenshot

```
Selamat Datang Di RENTAL CAMERA !!
 1. Login
 2. Daftar Pelanggan
 3. Lihat Data Petugas
 4. Lihat Data Camera
 0. Stop
 Masukkan Pilihan Anda : 1
1. Login petugas
2. Login pelanggan
 Pilih : 2
ID : 15
Nama : dio
```

Analisa

pendefinisian ulang method yang sama pada subclass

Laporan Sementara

Soal Praktikum

1. Terapkan Package pada project anda.
2. Menerapkan Relasi Antar Object (Minimal 3 relasi).
3. Menerapkan Inheritance(Pewarisan) pada project kalian.
4. Terapkan Enkapsulasi.
5. Terapkan Polimorphism.

Langkah Pertama

menambahkan polimorphisme

Source Code

```
package Alex07133_Entity;

public abstract class Alex07133_anggota {
    protected String
Alex07133_id_anggota,Alex07133_nama,Alex07133
_notelp;

    public Alex07133_anggota(String
Alex07133_id_anggota,String
Alex07133_nama,String Alex07133_notelp){

this.Alex07133_id_anggota=Alex07133_id_anggot
a;
```

Screenshot

```
0.ASTRID
1.SABRINA
2.ALEX
Petugas Rental = 2
Data berhasil disimpan
```

Laporan Sementara

Analisa

Digunakan untuk data anggota rental

Laporan Sementara

Langkah Kedua

Membuat class kamera

Source Code

```
package alex07133_rentalcamera;


public class Alex07133_camera {
    static String[]
Alex07133_CAMERA_CANON={"CANON 3000D","CANON
1500D","CANON 200D","CANON 750D"};
}
```

Screenshot

```
0. CANON 3000D
1. CANON 1500D
2. CANON 200D
3. CANON 750D
Pilih Kamera = 2
```

Analisa

Untuk Pemilihan kameranya

Laporan Sementara

Langkah Ketiga

Membuat class pelanggan

Source Code

```
package alex07133_rentalcamera;
import java.util.Date;

public class Alex07133_pelanggan {
    String
Alex07133_id_pelanggan,Alex07133_nama,Alex071
33_notelp;
    Date Alex07133_tglpinjam,
Alex07133_tglkembali;
    int Alex07133_indexkamera;
    int Alex07133_indexpetugas;

    public Alex07133_pelanggan(String
```

Screenshot

```
DATA KE 0
ID = 12
Nama = rayhan
Notelp = 08122359138
Kamera yang Dipinjam = CANON 200D
Tanggal Peminjaman Kamera (dd/mm/yyyy) = 06-11-2011
Tanggal Pengembalian Kamera (dd/mm/yyyy) = 06-12-2012
Petugas Rental = ALEX
```

Analisa

Untuk format penyimpanan data pelanggan

Laporan Sementara

Langkah Ke-empat

Membuat class RentalCamera

Source Code

```
package alex07133_rentalcamera;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.Scanner;

public class Alex07133_RentalCamera {
    static
ArrayList<Alex07133_anggota>dataPetugas = new
ArrayList();
    static
ArrayList<Alex07133_pelanggan>dataPelanggan =
new ArrayList();
```

Screenshot

```
Data Rental
Pilihan :
1. Input Data
2. Lihat Data
3. Ubah Data
4. Hapus Data
5. Exit
Pilih : 1
Input ID Pelanggan : 12
Input Nama Pelanggan : rayhan
Input Notelp Pelanggan : 08122359138
Input Tanggal Peminjaman Kamera (mm/dd/yyyy) = 18/11/2012
Input Tanggal Pengembalian Kamera (mm/dd/yyyy) = 19/11/2011
0. CANON 3000D
1. CANON 1500D
2. CANON 200D
3. CANON 750D
```

Analisa

Untuk membuat menu dan output input data

PRAKTIKUM
MODUL 3

LABORATORIUM REKAYASA PERANGKAT LUNAK
FAKULTAS TEKNIK ELEKTRO DAN
TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI ADHI TAMA
SURABAYA
2020

LAPORAN PRAKTIKUM

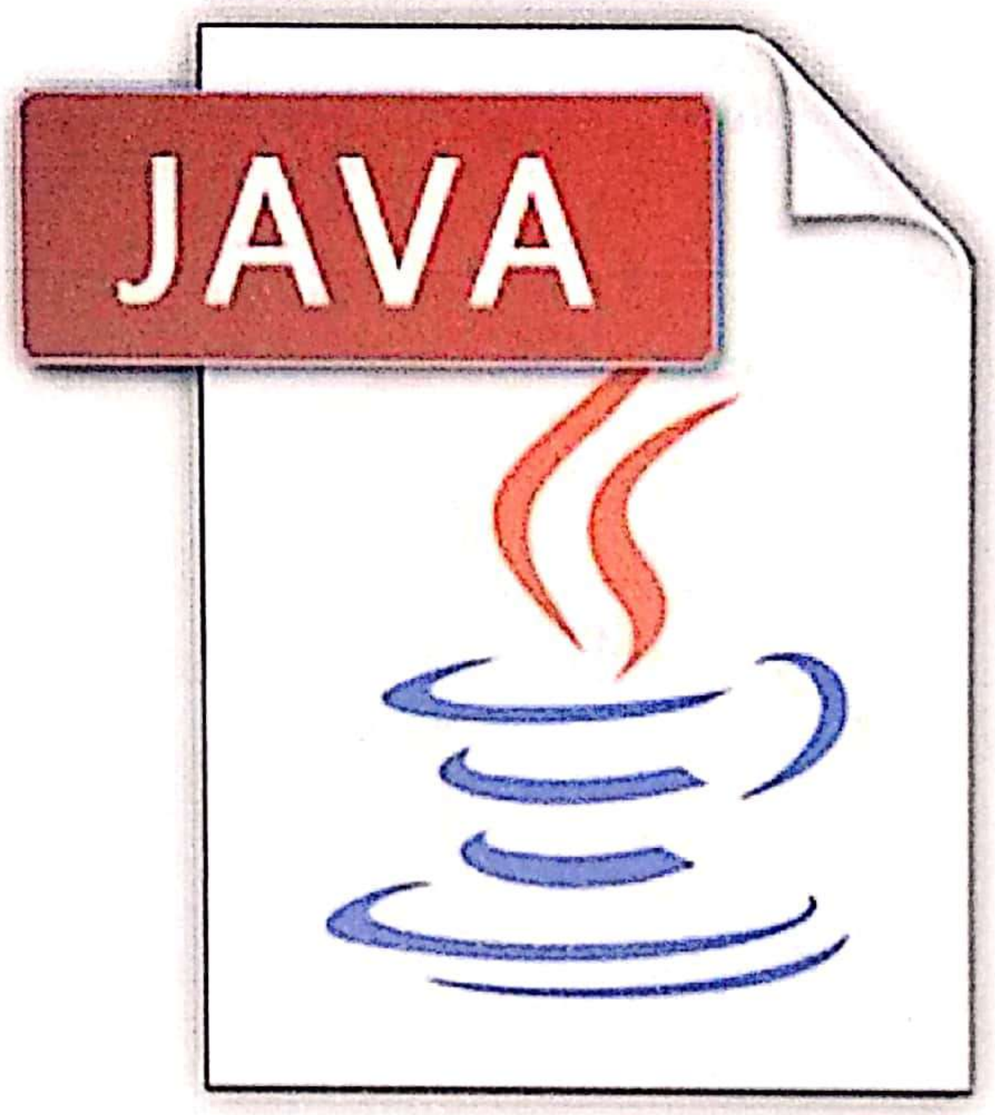
Java

PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
PERIODE VI - Tahun 2020/2021

NAMA	:	Alex Gilang S.
NPM	:	06.2019.1.07133
MODUL	:	3 (tiga)

Dipindai dengan CamScanner

LAPORAN PENDAHULUAN



PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
PERIODE VI - Tahun 2020/2021

NAMA : Alex Gilang S.
NPM : 06.2019.1.07133
MODUL : 3 (Tiga)

Dipindai dengan CamScanner

Tugas Pertanyaan

MODUL III

RELASI ANTAR OBJECT LANJUTAN

Kerjakan langkah-langkah berikut dan buktikan hasilnya dengan cetak screenshot atau program dijalankan pada Laporan tugas pendahuluan dan tunjukkan hasilnya pada saat praktikum.

Pada bab ini kita akan melanjutkan sistem informasi pendaftaran praktikum dari modul sebelumnya.

Aturan pengerjaan : Setiap nama class harus diakhiri npm dengan format nama class npm contoh ; Main 07015

Selanjutnya, silahkan membuat package class sesuai struktur sebagai berikut.

```
|-- controller
|----- Admin controller.java
|----- AllObject Model.java
|----- Controller Interface.java
|----- Praktikan controller.java
```

Keterangan : |-- : package

|----- : Isi class dari package

Setelah itu kita siapkan logika programnya. Pertama-tama kita menulis kode pada package controller, Didalam package ini terdapat 4 class. Masukkan code berikut pada setiap kelasnya

Controller interface.java

```
1. package controller;
2.
3. import java.util.Date;
4
5. public interface controller interface {
6.     public void login (String npm, String
       password);
7. }
```


Dipindai dengan CamScanner

Tugas Pertanyaan

AdminController.Java

```
1. package controller;
2.
3. import Entity.AslabEntity;
4. import Entity.DaftarPrakEntity;
5. import Entity.PraktikanEntity;
6. import Java.util.ArrayList;
7. import Java.util.Date;
8.
9. public class AdminController implements ControllerInterface {
10.   int Index Login = 0;
11.
12.   public AdminController () {
13.   }
14.
15. public void dataAslab() {
16.   String npmAslab [] = {"01", "02", "03", "04",
    "05", "06", "07"};
17.   String passwordAslab [] = {"01", "02", "03",
    "04", "05", "06", "07"};
18.   String namaAslab [] = {"Michael", "Alan",
    "Odil", "Fitria", "Sita", "Eric", "Fernanda"};
19.   String notelpAslab [] = {"01", "02", "03",
    "04", "05", "06", "07"};
20.   String tglLahirAslab [] = {"04/05/2000",
    "01/12/2000", "03/3/2000", "23/06/2000",
    "11/3/2000", "13/9/2000", "22/12/2000"};
21.   String laboratorium [] = {"RPL", "SO",
    "BASPROG", "BASPROG", "RPL", "SO", "RPL"};
22.   for (int i = 0; i < npmAslab.length; i++) {
23.   AllObjectModel.AslabModel.InsertAslab
```


Dipindai dengan CamScanner

Tugas Pertanyaan

```
(new AsiabEntity (npmAsiab[i], passwordAsiab[i], nama
Asiab[i], notelpAsiab[i], new Date (tglLahirAsiab[i]),
laboratorium[i]));
24.         }
25.     }
26.
27.     public void viewAsiab () {
28.         AllObjectModel.asiabModel.view();
29.     }
30.
31.     public void listpendaftarpraktikum () {
32.         AllObjectModel.daftarpraktModel.view();
33.     }
34.
35.     @ Override
36.         public void login (String npm, String password)
37.         {
38.             IndexLogin =
            AllObjectModel.asiabModel.cekData (npm,
 password);
39.         }
40.         public void updateIsVerified (int index, int index
prak, PraktikanEntity praktikanEntity) {
41.         AllObjectModel.daftarpraktModel.updateIsVerified
(index, new DaftarpraktEntity (indexprak, praktikanEntity,
true));
42.         }
43.         Public ArrayList <DaftarpraktEntity>
            cekdaftarpraktmodel () {
44.         return
```


Dipindai dengan CamScanner

Tugas Pertanyaan

```
AllObjectModel.daftarpraktModel.getDaftarprak
ArrayList();
}
}
```

Praktikancontroller.java

```
package controller;

import Entity.DaftarpraktEntity;
import Entity.PraktikanEntity;
import java.util.Date;

public class PraktikanController implements ControllerInterface {
    int IndexLogin = 0;
    public PraktikanController() {
    }

    public PraktikanEntity getData() {
        return
        AllObjectModel.praktikanModel.getPraktikanEntityArrayList
        (IndexLogin);
    }
    public void daftarpraktikum (int indexpraktikum, PraktikanEntity praktikanEntity, boolean isVerified) {
        AllObjectModel.daftarpraktModel.insertDataDaftarprak(
        new DaftarpraktEntity(indexpraktikum, PraktikanEntity,
        isVerified));
}

@Override
```


Dipindai dengan CamScanner

Tugas Pertanyaan

```
21. public void login ( String npm, String password) {
22.     indexLogin =
    AllObjectModel. prakTikanModel. cekData ( npm, password);
23. }
24.
25.     public void insert ( String npm, String password,
 String nama, String no_telp, Date tglLahir, String kelas ){
26.         AllObject Model. prakTikanModel. insert ( new
 PrakTikanEntity (npm, password, no_telp, nama, tglLahir,
 kelas ) );
27.     }
28.     Public prakTikanEntity prakTikanEntity () {
29.         return
        AllObject Model. prakTikanModel. get prakTikanEntity Array
 List (indexLogin) ;
30.     }
31.
32.     public int cekDaftarprak ( String npm) {
33.         int cek =
    AllObject Model. daftarprakModel. cekData ( npm, null);
34.         return cek;
35.     }
36.     public DaftarprakEntity showDaftarprak (int index) {
37.         return
    AllObject Model. daftarprakModel. showDaftarprak (index);
38.     }
39. }
```


Dipindai dengan CamScanner

Tugas Pertanyaan

AllObjectModel.java

```
1. package controller;
2.
3. import Model.AsiabModel;
4. import Model.DaftarpraktModel;
5. import Model.PraktikanModel;
6.
7. public class AllObjectModel {
8.     public static DaftarpraktModel daftarpraktModel =
new DaftarpraktModel();
9.     public static AsiabModel asiabmodel = new Asiab
Model();
10.    public static PraktikanModel praktikanModel =
new PraktikanModel();
11. }
```

Selanjutnya pada main class akan berbeda dari modul sebelumnya. Tulis kode ini pada Main.java dengan merubah isinya sebagai berikut:

Main.Java

```
1. package Project;
2.
3. import Entity.PraktikumEntity;
4. import Controller.AdminController;
5. import Controller.AllObjectModel;
6. import Controller.PraktikanController;
7. import java.text.SimpleDateFormat;
8. import java.util.Date;
9. import java.util.Scanner;
10.
11. public class Main {
12.     private static AdminController admin = new
AdminController();
```


Dipindai dengan CamScanner

Tugas Pertanyaan

```
13. private static scanner input = new scanner ( system. in);
14. private static praktikancontroller praktikan = new
praktikan controller ();
15.
16. public static void main ( String [] args) {
17.         int loop = 0;
18.         admin. daftarslab ();
19.         do {
20.                 int pilih = 0;
21.         System. out. print ( selamat Datang di laboratorium Informatika ITATS !!. " +
22.                         " \n 1. login " +
23.                         " \n 2. Daftar praktikan " +
24.                         " \n 0. Stop " +
25.                         " \n Masukkan pilihan Anda: ");
26.         pilih = input. nextint ();
27.         if ( pilih == 1) {
28.                 int pilLogin = 0;
29.                 int pilLogin = 0;
30.         System. out. print ( " 1. login Aslab " + " \n 2. login
praktikan " + " \n pilih: " );
31.         pilLogin = input. nextint ();
31.         if ( pilLogin == 1) {
33.                 loginAslab ();
34.         } else {
35.                 login praktikan ();
36.         }
37.
38.         } else if ( pilih == 2) {
39.                 registerPraktikan ();
40.         } else if ( pilih == 3) {
```


Dipindai dengan CamScanner

Tugas Pertanyaan

```
            admin.viewAslab();
                } else if (pilih == 4) {
                } else {
                    break;
                }
            } while (loop != 1);
        }

    Static void loginAslab() {
        System.out.print(" NPM: ");
        String npm = input.next();
        System.out.print("password: ");
        String password = input.next();
        try {
            admin.login(npm, password);
            System.out.println(" Selamat Datang " +
admin.aslabEntity().getNama() + " Dari lab " + admin.
aslabEntity().getLaboratorium());
            if (admin.cekdaftarpraktModel().size()
== 0) {
            System.out.println(" Data masih kosong !! ");
            } else {
                viewDaftarprak();
                updateIsVerified();
            }
        } catch (Exception exception) {
            System.out.println(" NPM atau password
anda salah !! ");
            }
        }
```


Dipindai dengan CamScanner

Tugas Pertanyaan

```
67.     Static void loginPraktikan () {
68.         System. out. print (" NPM : ");
69.         String npm = input. next ();
70.         System. out. print (" password : ");
71.         String password = input. next ();
72.         praktikan. login( npm, password );
73.         System. out. println (" Selamat Datang"
 + praktikan. praktikanEntity (). getNama ());
74.         int cek = praktikan. cekDaftarprak
( praktikan. praktikanEntity (). get npm ());
75.         if ( cek == -1 ) {
76.             System. out. println (" Anda Belum
 Terdaftar praktikum !! ");
77.             register Praktikum ();
78.         } else if ( cek == -2 ) {
79.             System. out. println (" Anda Belum
 Terdaftar praktikum !! ");
80.             registerpraktikum ();
81.         } else {
82.             System. out. println (" ==========
 ========== ");
83.             System. out. print (" NPM : " + Praktikan.
 showDaftarprak (cek). get Praktikan (). get npm() + " \n
 Nama : " + praktikan. showDaftarprak (cek). getpraktikan()
 . getNama () + " \n password " + praktikan. showDaftar prak
 (cek). get praktikan () . get password () + " \n No. telp " +
 praktikan. showDaftarprak (cek). get Praktikan (). get No.
 telp () + " \n Tanggal Lahir : " + new SimpleDateFormat ("
 dd - MM - yyyy "). format ( praktikan. showDaftarprak (cek)
 . get praktikan (). get Tgl Lahir ())
84.                 + " \n praktikum : " + praktikum
```


Dipindai dengan CamScanner

Tugas Pertanyaan

```
Entity.nama [praktikan.showDaftarprak(cek).getIndex
prak()] + "\n isVerified : ");
85.                 if (praktikan.showDaftarprak(cek).isVerified
() == false) {
86.                 System.out.println(" Belum di verifikasi Admin");
87.                 }
88.                 else {
89.                 System.out.println(" Telah di verifikasi ");
90.                 }
91.                 System.out.println(" Kelas : " + praktikan.
showDaftarprak(cek).getPraktikan().getKelas());
92.                 System.out.println("=========================
==========================");
93.                 }
94.                 }
95.
96.         Static void registerPraktikan() {
97.                 try {
98.                 System.out.print(" Masukkan NPM : ");
99.                 String npm = input.next();
100.                System.out.print(" Masukkan password: ");
101.                String password = input.next();
102.                System.out.print(" Masukkan Nama : ");
103.                String nama = input.next();
104.                System.out.print(" Masukkan No Telepon : ");
105.                String notelp = input.next();
106.                System.out.print(" Masukkan Tanggal Lahir
(dd/MM/yyyy) : ");
107.                Date tglLahir = new Date(input.next());
108.                System.out.print(" Masukkan Kelas Anda : ");
109.                String kelas = input.next();
```


Dipindai dengan CamScanner

Tugas Pertanyaan

```
110.        Praktikan.insert(npm, password, nama, notelp,
tglLahir, kelas);
111.        System.out.println("Daftar Sukses !!");
112.        } catch (Exception exception) {
113.        System.out.println("Format pengisian salah !!");
114.        }
115.    }
116.
117.    Static void registerPraktikum() {
118.        System.out.println("pilih praktikum: ");
119.        for (int i=0; i< PraktikumEntity.namaLength; i++) {
120.            System.out.println((i)+" " + praktikumEntity.namaLength[i]);
121.            int pilprak = input.nextInt();
122.
123.        Praktikan.daftarPraktikum(pilprak, praktikan, getData(), false);
124.        admin.listPendaftarPraktikum();
125.        }
126.        Static void viewDaftarprak() {
127.            admin.lisPendaftarPraktikum();
128.        }
129.
130.        Static void updateIsVerified() {
131.            System.out.print("Masukkan NPM: ");
132.            String NPM = input.next();
133.        int index = AllobjectModel.daftarprakModel.cekData(npm, null);
```


Dipindai dengan CamScanner

Tugas Pertanyaan

```
        admin.updateIsVerified(index, praktikan.showDaftarPrak(index).getIndexPrak(), praktikan.showDaftarPrak(index).getPraktikan();
        }
    }
```

- Jalankan dan periksa kembali programnya.

Dipindai dengan CamScanner

Tugas Praktikum

Soal Praktikum

1. *Melanjutkan program dari modul sebelumnya, tetap menerapkan CRUD pada program*
2. *Terapkan Controller Pada program anda*
    1) *Alur :*
        1. *View (Main) memberi inputan*
        2. *Controller menerima inputan dan memberi perintah ke model*
        3. *Model memerima perintah dan mengembalikan ke Controller*
        4. *Controller mengembalikan ke View (Main)*
    2) *Peraturan :*
        1. *Tidak boleh ada System.out.println(); di Controller*
        2. *Tidak boleh ada inputan di dalam Controller*
        3. *Array List harus berada di Model*
3. *Terapkan Exception Handling Pada program anda*
4. *Pada Model dan Controller (Menggunakan Parameter) semua inputan ada di class main*
5. *Petunjuk lebih lengkap ada di dalam lampiran File*
    1) *Peraturan pada package Model :*
        1. *Tidak boleh ada System.out.println(); di Model*
        2. *Tidak boleh ada inputan di dalam Model*
        3. *Array List harus berada di Model*
    2) *Peraturan pada main program :*
        1. *Tidak boleh ada Array List di main/view*
        2. *Tidak boleh memanggil Method Model di main, harus melalui Controller*
        3. *Harus menerapkan Enkapsulasi*
6. *Harap Dibaca keterangan pada comment di setiap Class*

Tugas Praktikum

Langkah Pertama

Membuat package Controller, package Entity, dan package Model.

Membuat class dengan nama Alex07133_RentalCamera pada package Entity

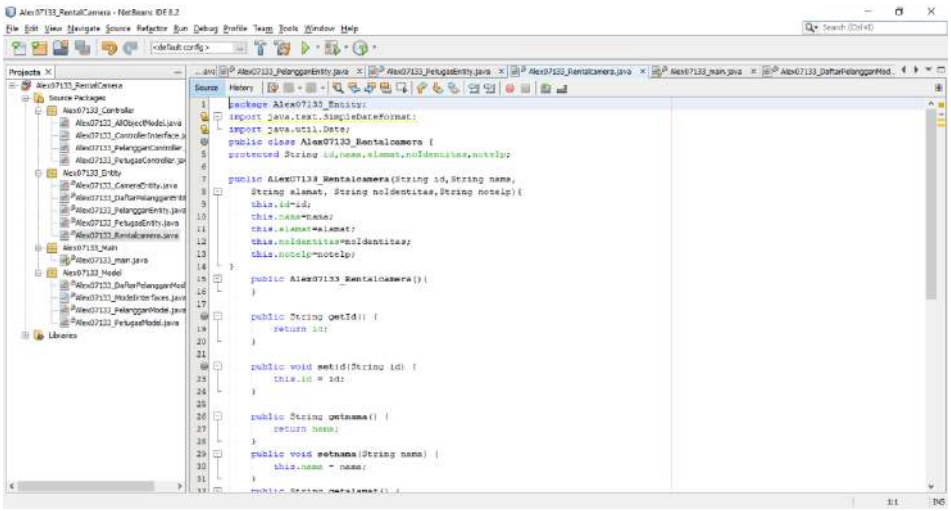
Source Code

```
package Entity;
import java.text.SimpleDateFormat;
import java.util.Date;
public class Alex07133_RentalCamera {
    protected String
id,nama,alamat,noIdentitas,notelp;

public Sabrina07239_RentalCamera(String id,String
nama, String alamat, String noIdentitas,String
notelp){
    this.id=id;
    this.nama=nama;
    this.alamat=alamat;
    this.noIdentitas=noIdentitas;
    this.notelp=notelp;
}
    public String getId() {
        return id;
    }
    public void setid(String id) {
        this.id = id;
    }
    public String getnama() {
        return nama;
    }
    public void setnama(String nama) {
        this.nama = nama;
    }
    public String getalamat() {
        return alamat;
    }
    public void setalamat(String alamat) {
        this.alamat = alamat;
    }
    public String getnoIdentitas() {
        return noIdentitas;
    }
```

Tugas Praktikum

```
    public void setnoIdentitas(String noIdentitas) {
        this.noIdentitas = noIdentitas;
    }
    public String getnotelp() {
        return notelp;
    }
    public void setnotelp(String notelp) {
        this.notelp = notelp;
    }

}
```

Screenshot



Analisa

Pembuatan class ini bertujuan agar class ini menjadi class induk (parent class) untuk penerapan inheritance pada projek
Terdapat enkapsulasi dalam projek, yaitu menggunakan protected pada properti. Protected dapat diakses oleh class itu sendiri dan class turunan

Tugas Praktikum

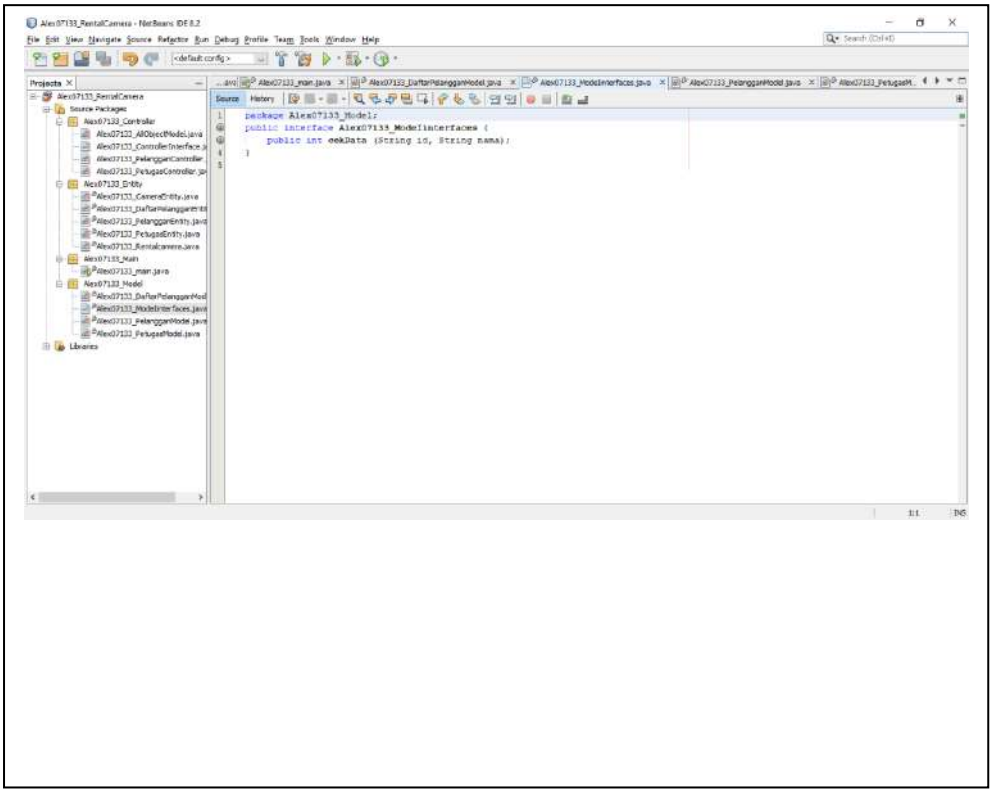
Langkah Kedua

Membuat kelas interface dengan nama Alex07133_ModeIinterfaces pada package Model

Source Code

```
package Model;
public interface Alex07133_ModeIinterfaces {
public int cekData (String id, String nama);
}
```

Screenshot



Analisa

Kelas interface hanya terdiri dari method-method kosong (abstract, tanpa body method). Interface biasanya lebih digunakan untuk mengelompokkan behavior-behavior menjadi grup behavior. Kelas interface menggambarkan kemampuan, bukan entitas/konsep objek.

Tugas Praktikum

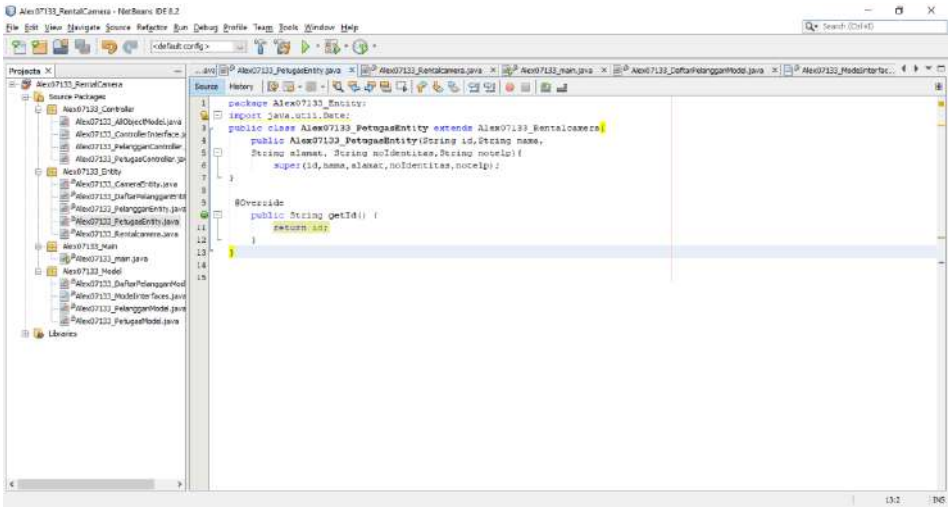
Langkah Ketiga

Membuat class Alex07133_PetugasEntity pada package Entity

Source Code

```
package Entity;
import java.util.Date;
public class Alex07133_PetugasEntity extends
Alex07133_RentalCamera{
    public Alex07133_PetugasEntity(String id,String
nama,
    String alamat, String noIdentitas,String notelp){
    super(id,nama,alamat,noIdentitas,notelp);
}
 @Override
    public String getId() {
        return id;
    }
}
```

Screenshot



Tugas Praktikum

Analisa

Terdapat inheritance yang diterapkan pada class ini, dimana class induknya adalah Alex07133_RentalCamera. Ditandai dengan keyword extends sebelum nama class induk disebutkan, keyword extens harus ditambahkan pada definisi class yang menjadi subclass. Selain itu terdapat keyword super untuk memanggil contructor dari class induk. Access modifier dari class induk akan diwariskan pada subclass ketika inheritance dilakukan.

Terdapat enkapsulasi yang diterapkan pada class ini, yaitu public. Public mempunyai sifat dapat diakses oleh semua class.

Terdapat polymorphism yang diterapkan pada class ini yaitu method overriding, method overriding digunakan untuk pendefinisian ulang method yang sama pada subclass, yang ditandai dengan keyword @Override.

Tugas Praktikum

Langkah Keempat

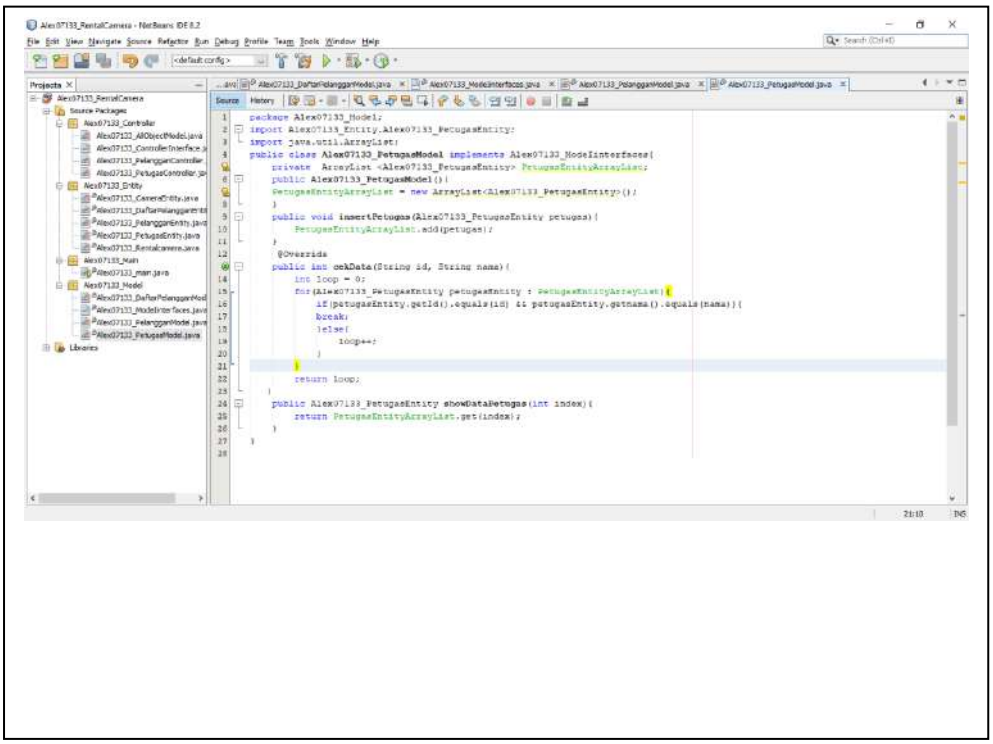
Membuat class Alex07133_PetugasModel pada package Model

Source Code

```
package Alex07133_Model;
import Alex07133_Entity.Alex07133_PetugasEntity;
import java.util.ArrayList;
public class Alex07133_PetugasModel implements
Alex07133_ModeIinterfaces{
    private  ArrayList <Alex07133_PetugasEntity>
PetugasEntityArrayList;
    public Alex07133_PetugasModel(){
    PetugasEntityArrayList = new
ArrayList<Alex07133_PetugasEntity>();
    }
    public void insertPetugas(Alex07133_PetugasEntity
petugas){
        PetugasEntityArrayList.add(petugas);
    }
     @Override
    public int cekData(String id, String nama){
        int loop = 0;
        for(Alex07133_PetugasEntity petugasEntity :
PetugasEntityArrayList){
            if(petugasEntity.getId().equals(id) &&
petugasEntity.getnama().equals(nama)){
            break;
            }else{
                loop++;
            }
        }
        return loop;
   }
    public Alex07133_PetugasEntity
showDataPetugas(int index){
        return PetugasEntityArrayList.get(index);
    }
}
```

Tugas Praktikum

Screenshot



Analisa

Terdapat implementasi yang diterapkan pada class, class ini mengimplementasi dari class Alex07133_ModeIInterface. Class melakukan implementasi agar class bisa mengadopsi behavior-behavior dari class Alex07133ModeIInterface.
Terdapat enkapsulasi yang diterapkan pada class ini, yaitu private dan public. Private memiliki sifat bahwa properti dan method hanya bisa diakses oleh class itu sendiri, sedangkan public dapat diakses oleh semua class.
Terdapat polymorphism yang diterapkan pada class ini yaitu method overriding, method overriding digunakan untuk pendefinisian ulang method yang sama pada subclass, yang ditandai dengan keyword @Override.

Langkah Kelima

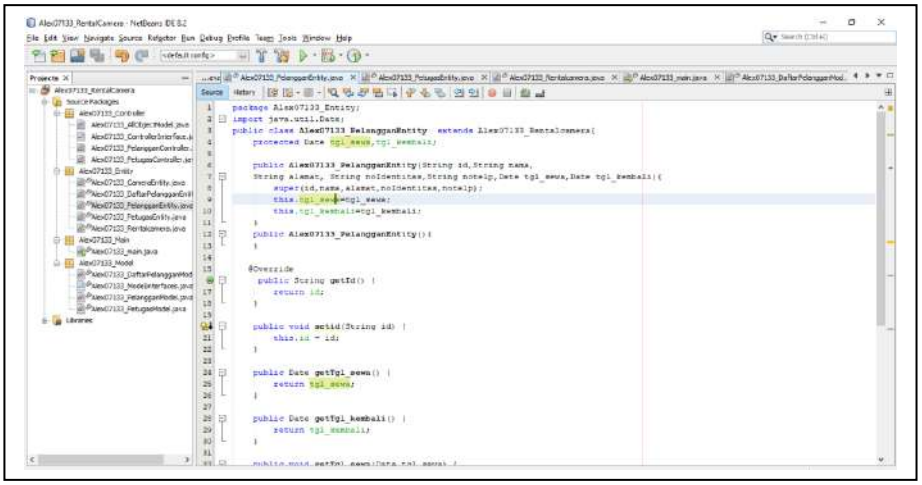
Membuat class Alex07133_PelangganEntity pada package Entity

Source Code

```
package Alex07133_Entity;
import java.util.Date;
public class Alex07133_PelangganEntity  extends
Alex07133_Rentalcamera{
    protected Date tgl_sewa,tgl_kembali;

    public Alex07133_PelangganEntity(String id,String
nama,
    String alamat, String noIdentitas,String
notelp,Date tgl_sewa,Date tgl_kembali){
        super(id,nama,alamat,noIdentitas,notelp);
        this.tgl_sewa=tgl_sewa;
        this.tgl_kembali=tgl_kembali;
    }
    public Alex07133_PelangganEntity(){
    }

   @Override
     public String getId() {
        return id;
    }

    public void setid(String id) {
        this.id = id;
    }

    public Date getTgl_sewa() {
        return tgl_sewa;
    }

    public Date getTgl_kembali() {
        return tgl_kembali;
    }

    public void setTgl_sewa(Date tgl_sewa) {
        this.tgl_sewa = tgl_sewa;
```

Tugas Praktikum

Screenshot



Analisa

Terdapat inheritance yang diterapkan pada class ini, dimana class induknya adalah Alex07133RentalCamera. Ditandai dengan keyword extends sebelum nama class induk disebutkan, keyword extens harus ditambahkan pada definisi class yang menjadi subclass. Selain itu terdapat keyword super untuk memanggil contructor dari class induk. Access modifier dari class induk akan diwariskan pada subclass ketika inheritance dilakukan.

Terdapat enkapsulasi yang diterapkan pada class ini, yaitu public dan protected. Public mempunyai sifat dapat diakses oleh semua class. Sedangkan protected, properti dan method bisa diakses oleh class itu sendiri dan subclass

Terdapat polymorphism yang diterapkan pada class ini yaitu method overriding, method overriding digunakan untuk pendefinisian ulang method yang sama pada subclass, yang ditandai dengan keyword @Override.

Tugas Praktikum

Langkah Keenam

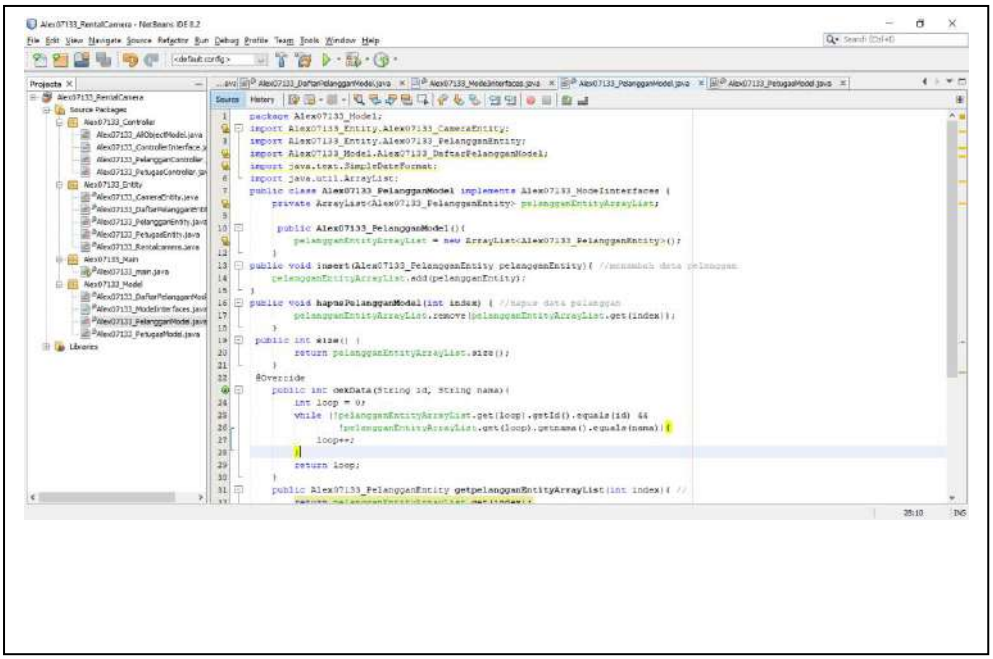
Membuat class Alex07133_PelangganModel pada package Model

Source Code

```
package Alex07133_Model;
import Alex07133_Entity.Alex07133_CameraEntity;
import Alex07133_Entity.Alex07133_PelangganEntity;
import
Alex07133_Model.Alex07133_DaftarPelangganModel;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
public class Alex07133_PelangganModel implements
Alex07133_ModeIinterfaces {
    private ArrayList<Alex07133_PelangganEntity>
pelangganEntityArrayList;

     public Alex07133_PelangganModel(){
        pelangganEntityArrayList = new
ArrayList<Alex07133_PelangganEntity>();
    }
public void insert(Alex07133_PelangganEntity
pelangganEntity){ //menambah data pelanggan
    pelangganEntityArrayList.add(pelangganEntity);
}
public void hapusPelangganModel(int index) { //hapus
data pelanggan

pelangganEntityArrayList.remove(pelangganEntityArrayL
ist.get(index));
    }
 public int size() {
        return pelangganEntityArrayList.size();
    }
 @Override
    public int cekData(String id, String nama){
        int loop = 0;
        while
(!pelangganEntityArrayList.get(loop).getId().equals(i
d) &&

!pelangganEntityArrayList.get(loop).getnama().equals(
nama)){
            loop++;
        }
```

Tugas Praktikum

Screenshot



Analisa

Terdapat implementasi yang diterapkan pada class, class ini mengimplementasi dari class Alex07133_ModeIinterface. Class melakukan implementasi agar class bisa mengadopsi behavior-behavior dari class Alex07133_ModeIinterface.
Terdapat enkapsulasi yang diterapkan pada class ini, yaitu private dan public. Private memiliki sifat bahwa properti dan method hanya bisa diakses oleh class itu sendiri, sedangkan public dapat diakses oleh semua class.
Terdapat polymorphism yang diterapkan pada class ini yaitu method overriding, method overriding digunakan untuk pendefinisian ulang method yang sama pada subclass, yang ditandai dengan keyword @Override.

Tugas Praktikum

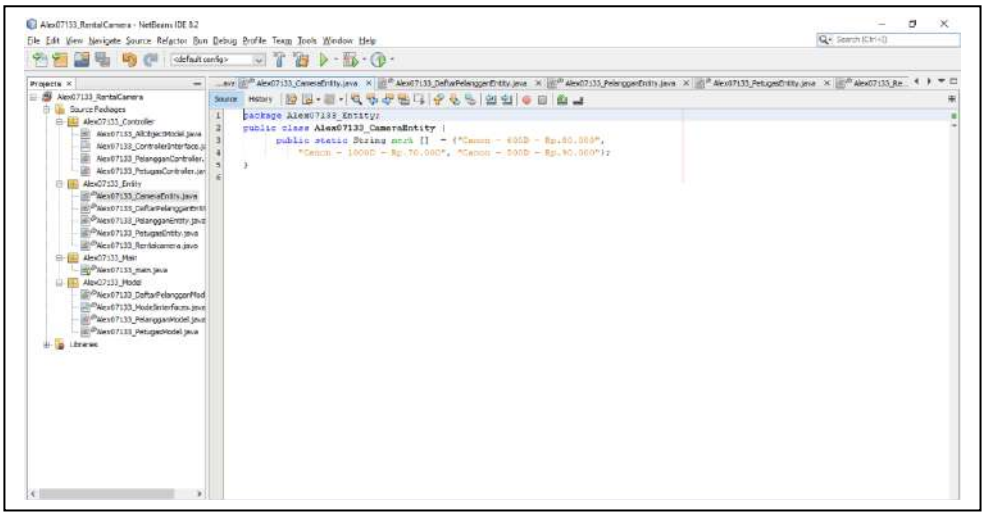
Langkah Ketujuh

Membuat class Alex07133_CameraEntity pada package Entity

Source Code

```
package Entity;
public class Alex07133_CameraEntity {
      public static String merk []  = {"Canon - 600D
- Rp.80.000", "Canon - 1000D - Rp.70.000", "Canon -
500D - Rp.90.000"}; //atribut
}
```

Screenshot



Analisa

Class terdapat data array yang berisi daftar pilihan camera, bertipe data String.
Class tidak menerapkan inheritance dan polymorphism.
Terdapat enkapsulasi yang diterapkan pada class ini, yaitu public. Public memiliki sifat bahwa properti dan method bisa diakses oleh semua class.

Tugas Praktikum

Langkah Kedelapan

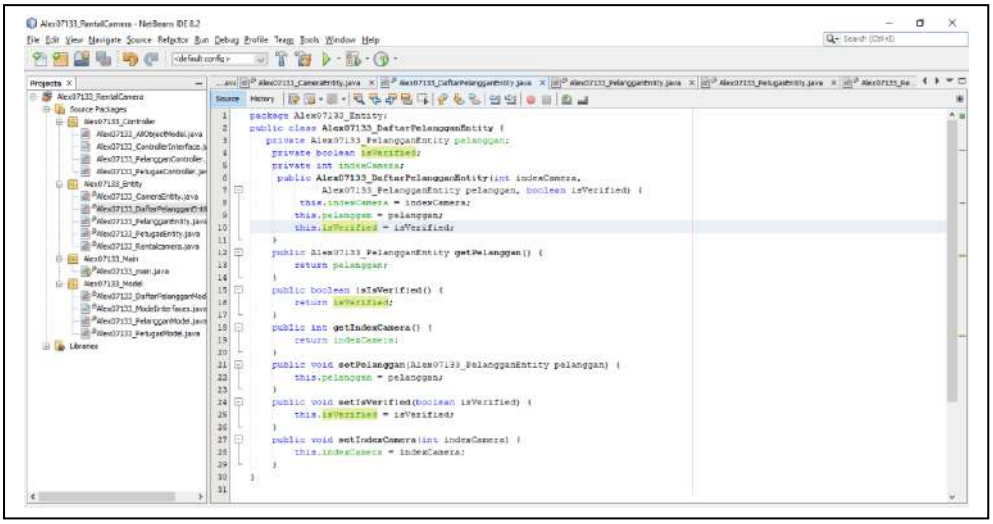
Membuat class Alex07133_DaftarPelangganEntity pada package Entity

Source Code

```
package Alex07133_Entity;
public class Alex07133_DaftarPelangganEntity {
   private Alex07133_PelangganEntity pelanggan;
    private boolean isVerified;
    private int indexCamera;
     public Alex07133_DaftarPelangganEntity(int
indexCamera,
             Alex07133_PelangganEntity pelanggan,
boolean isVerified) {
         this.indexCamera = indexCamera;
        this.pelanggan = pelanggan;
        this.isVerified = isVerified;
    }
    public Alex07133_PelangganEntity getPelanggan() {
        return pelanggan;
    }
    public boolean isIsVerified() {
        return isVerified;
    }
    public int getIndexCamera() {
        return indexCamera;
    }
    public void
setPelanggan(Alex07133_PelangganEntity pelanggan) {
        this.pelanggan = pelanggan;
    }
    public void setIsVerified(boolean isVerified) {
        this.isVerified = isVerified;
    }
    public void setIndexCamera(int indexCamera) {
        this.indexCamera = indexCamera;
    }
}
```

Tugas Praktikum

Screenshot



Analisa

Terdapat enkapsulasi yang diterapkan pada class ini, yaitu private dan public. Private memiliki sifat bahwa properti dan method hanya bisa diakses oleh class itu sendiri, sedangkan public dapat diakses oleh semua class.
Terdapat variable untuk menampung data pelanggan, serta isVerified untuk mengonfirmasi status pelanggan nantinya, dan juga indexCamera untuk menginput data pilihan camera.

Langkah Kesembilan

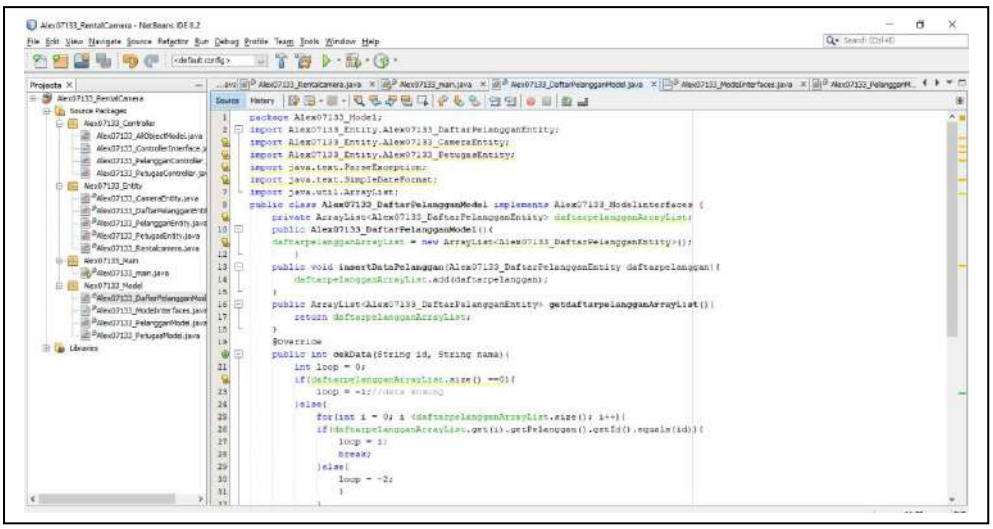
Membuat class Alex07133_DaftarPelangganModel pada package Model

Source Code

```
package Alex07133_Model;
import
Alex07133_Entity.Alex07133_DaftarPelangganEntity;
import Alex07133_Entity.Alex07133_CameraEntity;
import Alex07133_Entity.Alex07133_PetugasEntity;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
public class Alex07133_DaftarPelangganModel
implements Alex07133_ModeIinterfaces {
    private
ArrayList<Alex07133_DaftarPelangganEntity>
daftarpelangganArrayList;
    public Alex07133_DaftarPelangganModel(){
    daftarpelangganArrayList = new
ArrayList<Alex07133_DaftarPelangganEntity>();
        }
    public void
insertDataPelanggan(Alex07133_DaftarPelangganEntity
daftarpelanggan){

daftarpelangganArrayList.add(daftarpelanggan);
    }
    public ArrayList<Alex07133_DaftarPelangganEntity>
getdaftarpelangganArrayList(){
        return daftarpelangganArrayList;
    }
    @Override
    public int cekData(String id, String nama){
        int loop = 0;
        if(daftarpelangganArrayList.size() ==0){
            loop = -1;//data kosong
        }else{
            for(int i = 0; i
<daftarpelangganArrayList.size(); i++){

if(daftarpelangganArrayList.get(i).getPelanggan().get
Id().equals(id)){
                loop = i;
                break;
```

Tugas Praktikum

Screenshot



Analisa

Terdapat implementasi yang diterapkan pada class, class ini mengimplementasi dari class Alex07133_ModeIinterface. Class melakukan implementasi agar class bisa mengadopsi behavior-behavior dari class Alex07133_ModeIinterface.
Terdapat enkapsulasi yang diterapkan pada class ini, yaitu private dan public. Private memiliki sifat bahwa properti dan method hanya bisa diakses oleh class itu sendiri, sedangkan public dapat diakses oleh semua class.
Terdapat polymorphism yang diterapkan pada class ini yaitu method overriding, method overriding digunakan untuk pendefinisian ulang method yang sama pada subclass, yang ditandai dengan keyword @Override.

Tugas Praktikum

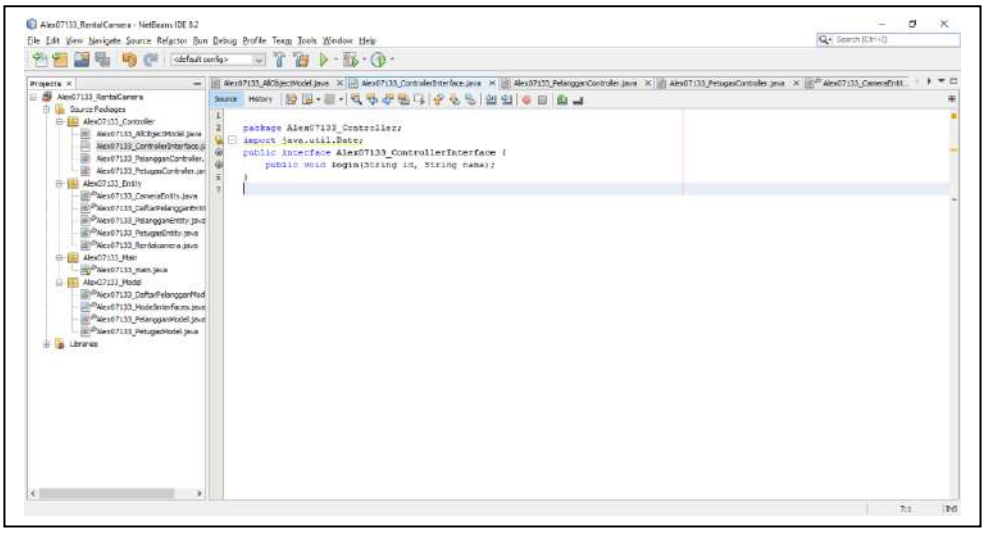
Langkah Kesepuluh

Membuat class Alex07133_ControllerInterface pada package Controller

Source Code

```
package Controller;

import java.util.Date;
public interface Alex07133_ControllerInterface {
    public void login(String id, String nama);
}
```

Screenshot



Analisa

Kelas interface hanya terdiri dari method-method kosong (abstract, tanpa body method). Interface biasanya lebih digunakan untuk mengelompokkan behavior-behavior menjadi grup behavior. Kelas interface menggambarkan kemampuan, bukan entitas/konsep objek.

Tugas Praktikum

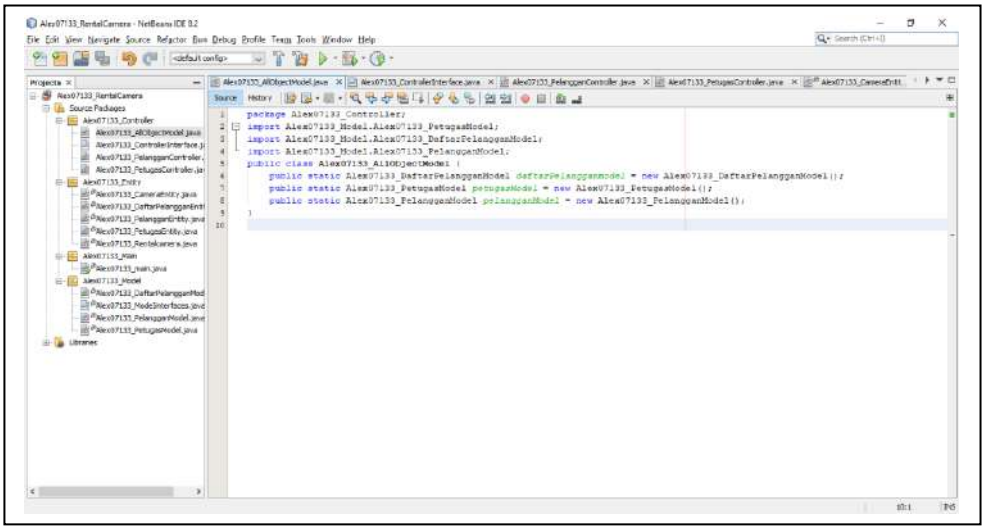
Langkah Kesebelas

Membuat class Alex07133_AllObjectModel pada package Controller

Source Code

```
package Alex07133_Controller;
import Alex07133_Model.Alex07133_PetugasModel;
import
Alex07133_Model.Alex07133_DaftarPelangganModel;
import Alex07133_Model.Alex07133_PelangganModel;
public class Alex07133_AllObjectModel {
    public static Alex07133_DaftarPelangganModel
daftarPelangganmodel = new
Alex07133_DaftarPelangganModel();
    public static Alex07133_PetugasModel petugasModel
= new Alex07133_PetugasModel();
    public static Alex07133_PelangganModel
pelangganModel = new Alex07133_PelangganModel();
}
```

Screenshot



Analisa

Kelas yang diciptakan untuk menjadi penghubung antara class-class yang berada pada package Model dan main program

Tugas Praktikum

Langkah Kedua belas

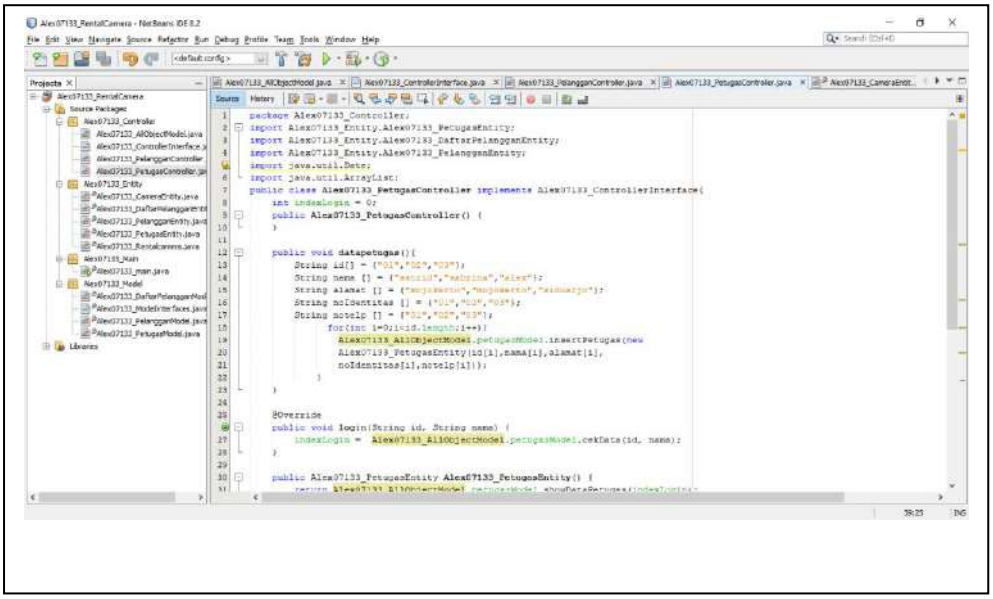
Membuat class Alex07133_PetugasController pada package Controller

Source Code

```
package Alex07133_Controller;
import Alex07133_Entity.Alex07133_PetugasEntity;
import
Alex07133_Entity.Alex07133_DaftarPelangganEntity;
import Alex07133_Entity.Alex07133_PelangganEntity;
import java.util.Date;
import java.util.ArrayList;
public class Alex07133_PetugasController implements
Alex07133_ControllerInterface{
    int indexLogin = 0;
    public Alex07133_PetugasController() {
    }

    public void datapetugas(){
        String id[] = {"01","02","03"};
        String nama [] = {"astrid","sabrina","alex"};
        String alamat [] =
{"mojokerto","mojokerto","sidoarjo"};
        String noIdentitas [] = {"01","02","03"};
        String notelp [] = {"01","02","03"};
              for(int i=0;i<id.length;i++){

Alex07133_AllObjectModel.petugasModel.insertPetugas(n
ew

Alex07133_PetugasEntity(id[i],nama[i],alamat[i],
                noIdentitas[i],notelp[i]));
            }
    }

    @Override
    public void login(String id, String nama) {
        indexLogin =
Alex07133_AllObjectModel.petugasModel.cekData(id,
nama);
    }
```

Tugas Praktikum

Screenshot



Analisa

Terdapat implementasi yang diterapkan pada class, class ini mengimplementasi dari class Alex07133_ControllerInterface. Class melakukan implementasi agar class bisa mengadopsi behavior-behavior dari class Alex07133_ControllerInterface.

Terdapat polymorphism yang diterapkan pada class ini yaitu method overriding, method overriding digunakan untuk pendefinisian ulang method yang sama pada subclass, yang ditandai dengan keyword @Override.

Tugas Praktikum

Langkah Ketiga belas

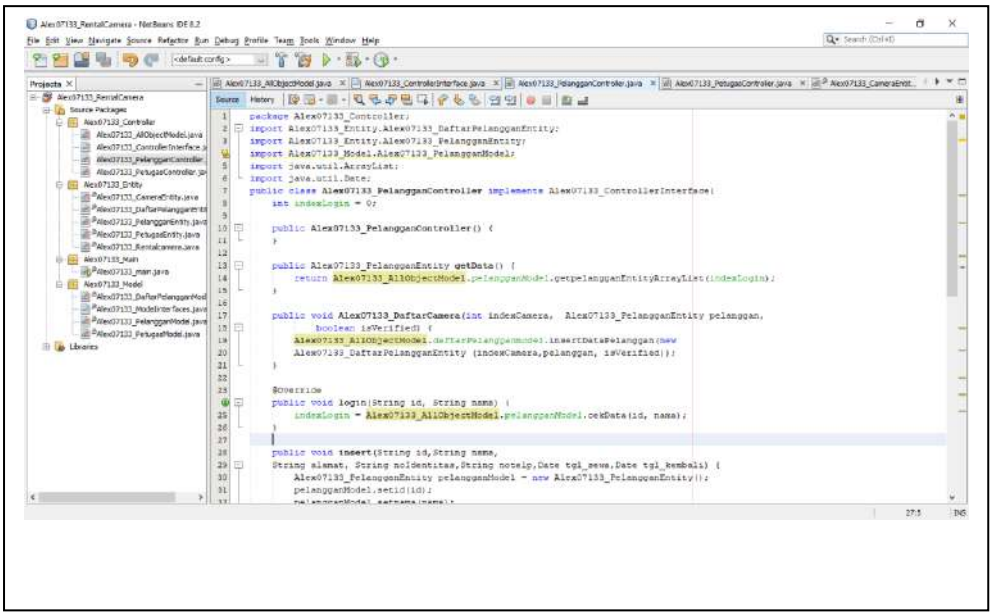
Membuat class Alex07133_PelangganController pada package Controller

Source Code

```
package Alex07133_Controller;
import
Alex07133_Entity.Alex07133_DaftarPelangganEntity;
import Alex07133_Entity.Alex07133_PelangganEntity;
import Alex07133_Model.Alex07133_PelangganModel;
import java.util.ArrayList;
import java.util.Date;
public class Alex07133_PelangganController implements
Alex07133_ControllerInterface{
    int indexLogin = 0;

    public Alex07133_PelangganController() {
    }

    public Alex07133_PelangganEntity getData() {
        return
Alex07133_AllObjectModel.pelangganModel.getpelangganE
ntityArrayList(indexLogin);
    }

    public void Alex07133_DaftarCamera(int
indexCamera,  Alex07133_PelangganEntity pelanggan,
            boolean isVerified) {

Alex07133_AllObjectModel.daftarPelangganmodel.insertD
ataPelanggan(new
        Alex07133_DaftarPelangganEntity
(indexCamera,pelanggan, isVerified));
    }

    @Override
    public void login(String id, String nama) {
        indexLogin =
Alex07133_AllObjectModel.pelangganModel.cekData(id,
nama);
    }

    public void insert(String id,String nama,
    String alamat, String noIdentitas,String
notelp,Date tgl_sewa,Date tgl_kembali) {
```

Tugas Praktikum

Screenshot



Analisa

Terdapat implementasi yang diterapkan pada class, class ini mengimplementasi dari class Alex07133_ControllerInterface. Class melakukan implementasi agar class bisa mengadopsi behavior-behavior dari class Alex07133_ControllerInterface.

Terdapat polymorphism yang diterapkan pada class ini yaitu method overriding, method overriding digunakan untuk pendefinisian ulang method yang sama pada subclass, yang ditandai dengan keyword @Override.

Tugas Praktikum

Langkah Keempat belas

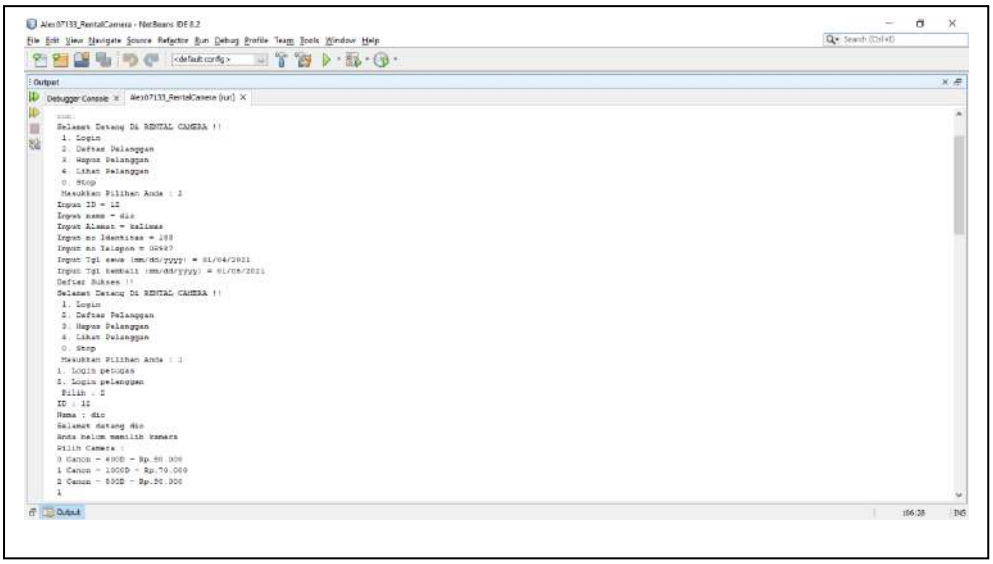
Mengisi code-code pada main class

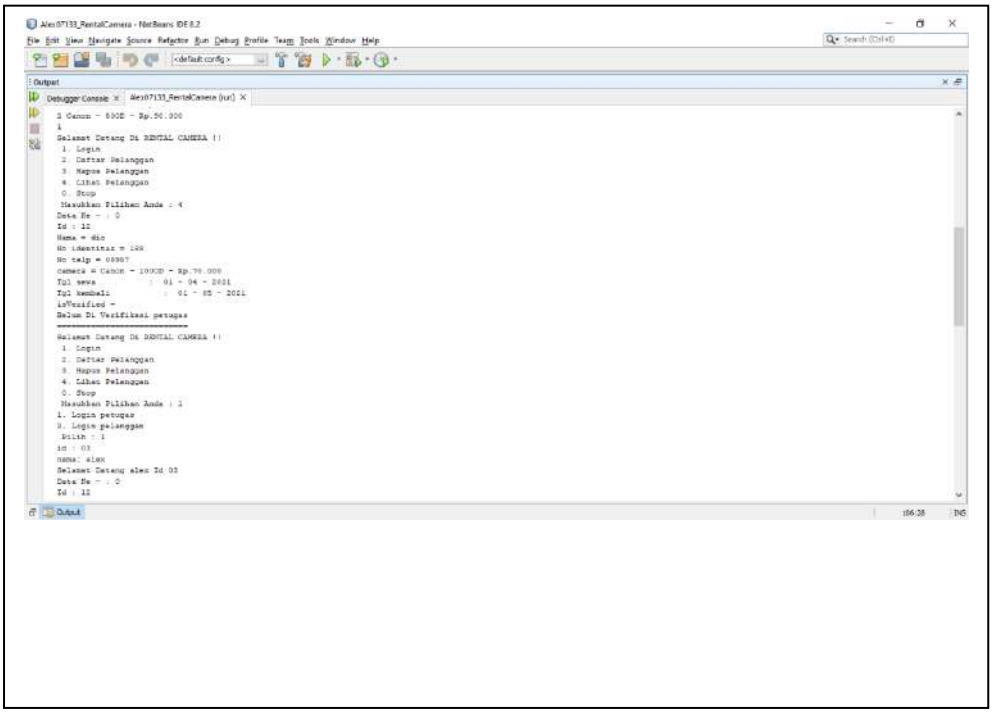
Source Code

```
package Alex07133_Main;
import Alex07133_Entity.Alex07133_CameraEntity;
import
Alex07133_Entity.Alex07133_DaftarPelangganEntity;
import
Alex07133_Model.Alex07133_DaftarPelangganModel;
import
Alex07133_Controller.Alex07133_PetugasController;
import Alex07133_Controller.Alex07133_AllObjectModel;
import
Alex07133_Controller.Alex07133_PelangganController;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;

public class Alex07133_main {

    private static Alex07133_PetugasController
petugasModel = new Alex07133_PetugasController();
    private static Alex07133_PelangganController
pelangganModel = new Alex07133_PelangganController();
    private static Scanner input = new Scanner
(System.in);

    public static void main(String[] args){
        Alex07133_main data = new Alex07133_main();
        data.viewMenu();
    }

    void viewMenu(){
        int loop = 0;
        int pilih = 0;
        int pilLogin = 0;
        petugasModel.datapetugas();
        do{
           System.out.print("Selamat Datang Di RENTAL
CAMERA !!" +
                            "\n 1. Login" +
                            "\n 2. Daftar Pelanggan"
+
```

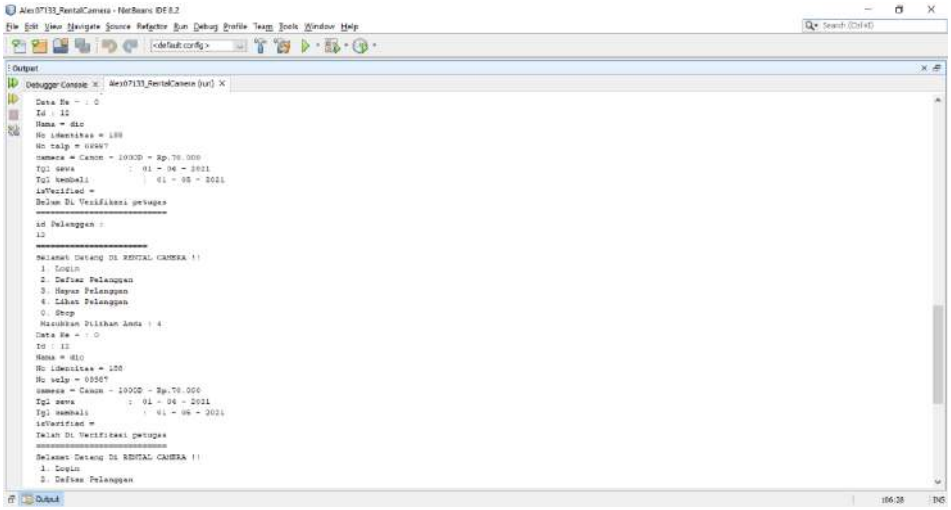
Tugas Praktikum

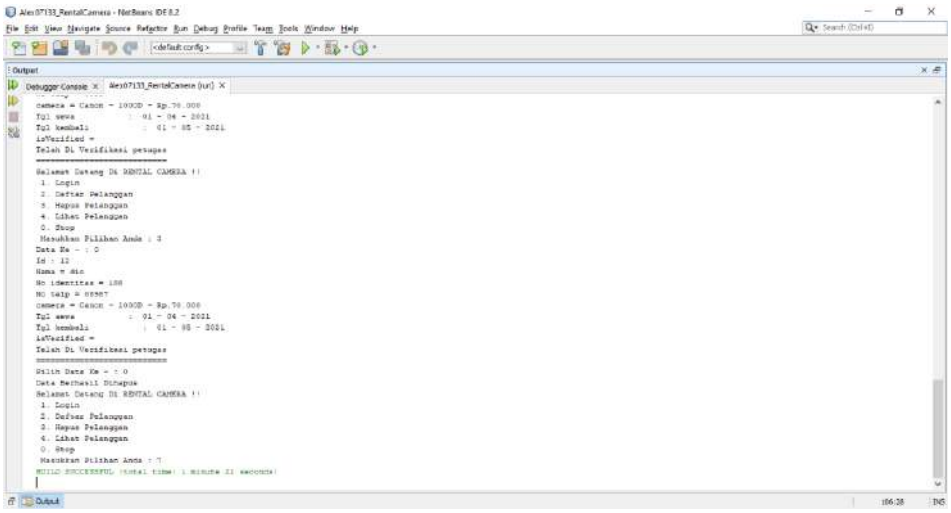
Screenshot





Tugas Praktikum





Tugas Praktikum

Analisa

Pada main program, semua inputan terdapat di dalamnya lalu akan diteruskan ke Controller untuk memberi perintah ke Model, kemudian dari Model akan dikembalikan Controller untuk dikembalikan ke main program lagi agar menampilkan output program. Semua code pada main program tidak memanggil method di Model, tetapi method di Controller. Pada main program juga tidak terdapat ArrayList karena ArrayList berada di Model.

Terdapat enkapsulasi yang diterapkan pada class ini, yaitu private. Private memiliki sifat bahwa properti dan method hanya bisa diakses oleh class itu sendiri.

Terdapat penggunaan Exception Handling pada method register(), dimana Exception Handling bertujuan agar apabila terdapat kesalahan pengisian inputan program tidak akan langsung terhenti, melainkan akan menampilkan pesan error dan akan otomatis kembali ke tampilan pilihan menu, lalu bisa melakukan pengisian data ulang.

Laporan Sementara

Soal Praktikum

Terapkan Controller

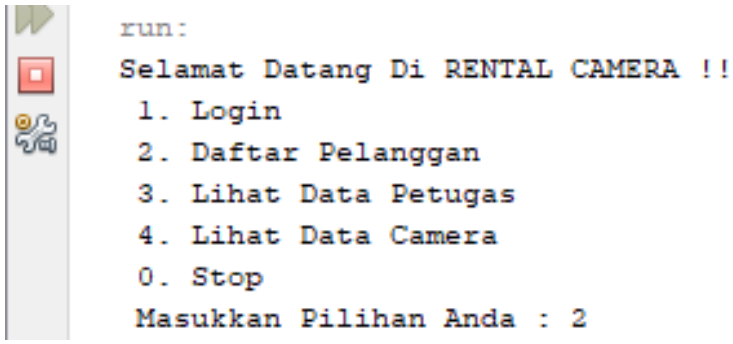
Langkah Pertama

Membuat Controller

Source Code

```
package Alex07133_Controller;
import
Alex07133_Model.Alex07133_PetugasModel;
import
Alex07133_Model.Alex07133_DaftarPelangganMode
l;
import
Alex07133_Model.Alex07133_PelangganModel;
public class Alex07133_AllObjectModel {
    public static
Alex07133_DaftarPelangganModel
daftarPelangganmodel = new
```

Screenshot



Analisa

Disini saya membuat package baru yang bernama controller yang digunakan untuk menghubungkan semua model interfaces nya

Laporan Sementara

Langkah Kedua

Menerapkan exception handling

Source Code

```
/*
 * To change this license header, choose
License Headers in Project Properties.
 * To change this template file, choose Tools
| Templates
 * and open the template in the editor.
 */
package Alex07133_Controller;
import
Alex07133_Entity.Alex07133_PetugasEntity;
import
Alex07133_Entity.Alex07133_DaftarPelangganEnt
itv:
```

Screenshot

```
Selamat Datang Di RENTAL CAMERA !!
 1. Login
 2. Daftar Pelanggan
 3. Lihat Data Petugas
 4. Lihat Data Camera
 0. Stop
 Masukkan Pilihan Anda : 1
1. Login petugas
2. Login pelanggan
 Pilih : 1
id : 1
nama: aw
NPM atau Password Anda Salah !!!
```

Analisa

Dengan exception mengatasi permasalahan jika user salah menginputkan nilai atau data

Laporan Sementara

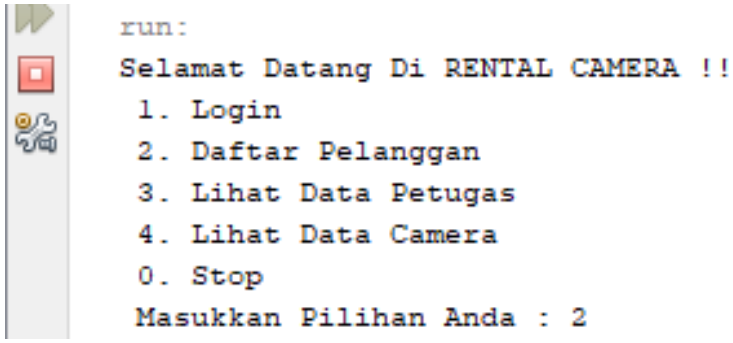
Langkah Ketiga

Pada Model dan Controller (Menggunkan Parameter) semua inputan ada di class main

Source Code

```
package Alex07133_Main;
import
Alex07133_Entity.Alex07133_CameraEntity;
import
Alex07133_Entity.Alex07133_PelangganEntity;
import
Alex07133_Controller.Alex07133_PetugasControl
ler;
import
Alex07133_Controller.Alex07133_AllObjectModel
;
import
Alex07133_Controller.Alex07133_PelangganContr
```

Screenshot



Analisa

Dengan ini semua inputannya berada pada main yang terhubung dengan controller

PRAKTIKUM
MODUL 4

LABORATORIUM REKAYASA PERANGKAT LUNAK
FAKULTAS TEKNIK ELEKTRO DAN
TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI ADHI TAMA
SURABAYA
2020

LAPORAN PRAKTIKUM

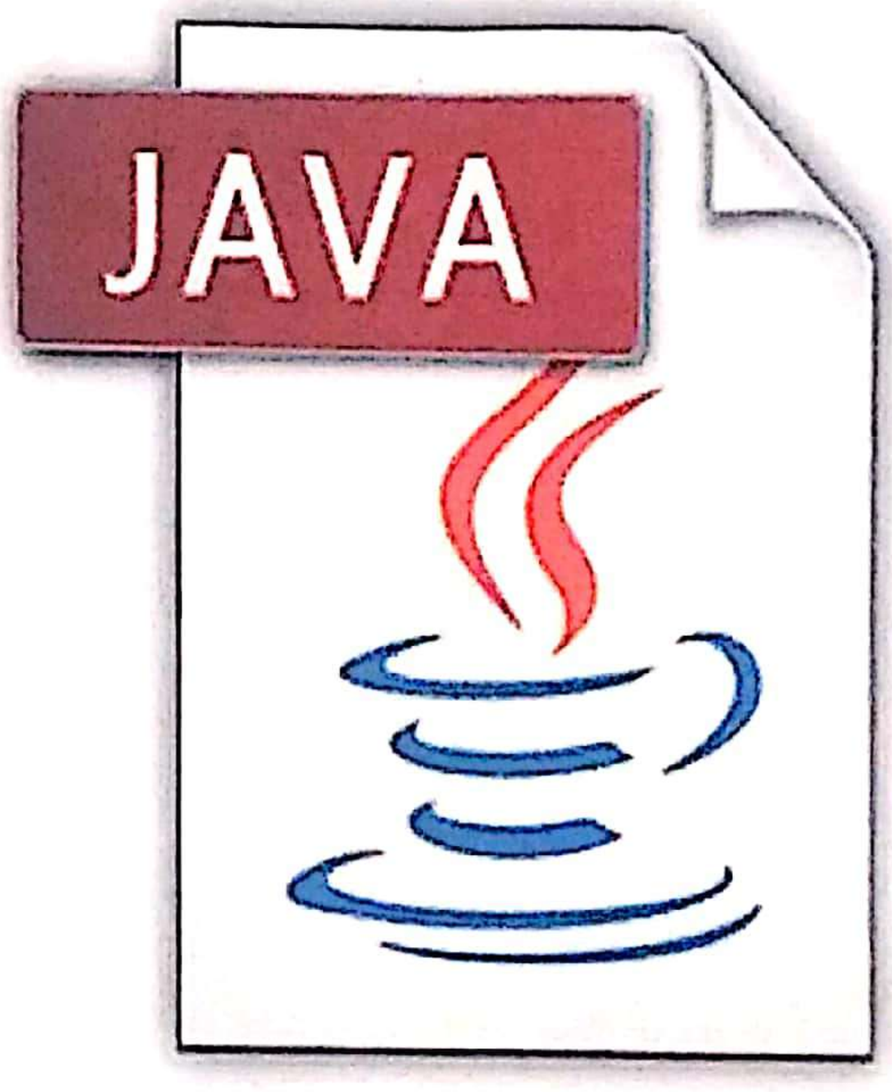
Java

PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
PERIODE VI - Tahun 2020/2021

NAMA	:	Alex Gilang S.
NPM	:	06.2019.1.07133
MODUL	:	4 (empat)

Dipindai dengan CamScanner

LAPORAN PENDAHULUAN



PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
PERIODE VI - Tahun 2020/2021

NAMA : Alex Gilang S.
NPM : 06.2019.1.07133
MODUL : 4 (empat)

Dipindai dengan CamScanner

Tugas Pertanyaan

MODUL IV

TAMPILAN DAN PENANGANAN

Kerjakan langkah-langkah berikut dan buktikan dengan menggunakan screenshoot pada saat dijalankan pada laporan Tugas pendahuluan.

Aturan pengerjaan : setiap nama class harus didahului NPM dengan format NPMNamaClass : 06123GUI

Pada bab ini kita akan melanjutkan sistem informasi pada modul 3, dimana kita akan menambahkan GUI pada Sistem Informasi Pendaftaran praktikum.

Langkah-langkah pengerjaan sebagai berikut:

1. Berikan tambahan code pada class DaftarprakModel yang berisikan method Arraylist<DaftarprakEntity> yang digunakan untuk meretum daftarprak Arraylist. Sebagai berikut:

```
Class Daftarprak Model
1.  public Arraylist < Daftar prak Entity>
2.  alldataprak ()&
3.  return
4.     daftarprak Arraylist ;
5.  }
```

2. Berikan tambahan pada class praktikan Controller pada package Controller dengan membuat method bernama defaultTableModel daftarprak () sebagai berikut:

```
Class PraktikanController
1.  public DefaultTableModel daftar prak ()&
2.     Default TableModel dtmdaftarprak = new
```


Dipindai dengan CamScanner

Tugas Pertanyaan

```
Default Table Model ();
3.  Object [] kolom = {"NPM", "Nama", "Password",
    "No Telp", "Tgl Lahir", "praktikum", "Verified",
    "kelas"};
4.  dtmdaftarprak.setColumnIdentifiers (kolom);
5.  int size = AllObjectModel.daftarprakModel.alldata
    prak().size();
6.  string verif = null;
7.  for (int i=0; i < size; i++){
8.  if (AllObjectModel.daftarprakModel.alldataprak
    ().get(i).isVerified()==false){
9.  verif = "no";
10. }else{
11.  verif = "yes";
12. }
13.  Object [] data = new Object [8];
14. data [0] = AllobjectModel.daftarprakModel.all
    dataprak().get(i).getPratikan().getNpm();
15. data [1] = AllobjectModel.daftarprakModel.all
    dataprak().get(i).getPratikan().getNama();
16. data [2] = AllobjectModel.daftarprakModel.all
    dataprak().get(i).getPratikan().getPassword();
17. data [3] = AllobjectModel.daftarprakModel.all
    dataprak.get(i).getPratikan().getNo_telp();
18. data [4] new SimpleDateFormat ("dd-MM-yyyy")
    .format (AllObjectModel.daftarprakModel.alldata
    prak().get(i).getPratikan().getTglLahir());
19. data [5] = PraktikumEntity.nama [AllobjectModel.
    daftarprakModel.alldataprak().get(i).getIndex
    prak()];
20. data [6] = verif;
21. data [7] = AllobjectModel.daftarprakModel.all
```


Dipindai dengan CamScanner

Tugas Pertanyaan

```
        dataprak().get(1).getPratikan().getkelas();
22.     dtmdaftar prak .addRow (data);
23.     }
24.     return dtmdaftarprak;
25.     }
26.     }
```

3. Terapkan code dibawah ini untuk membuat suatu tampilan dengan menggunakan GUI.

- Buatlah class baru pada package View dengan nama class allobjctrl

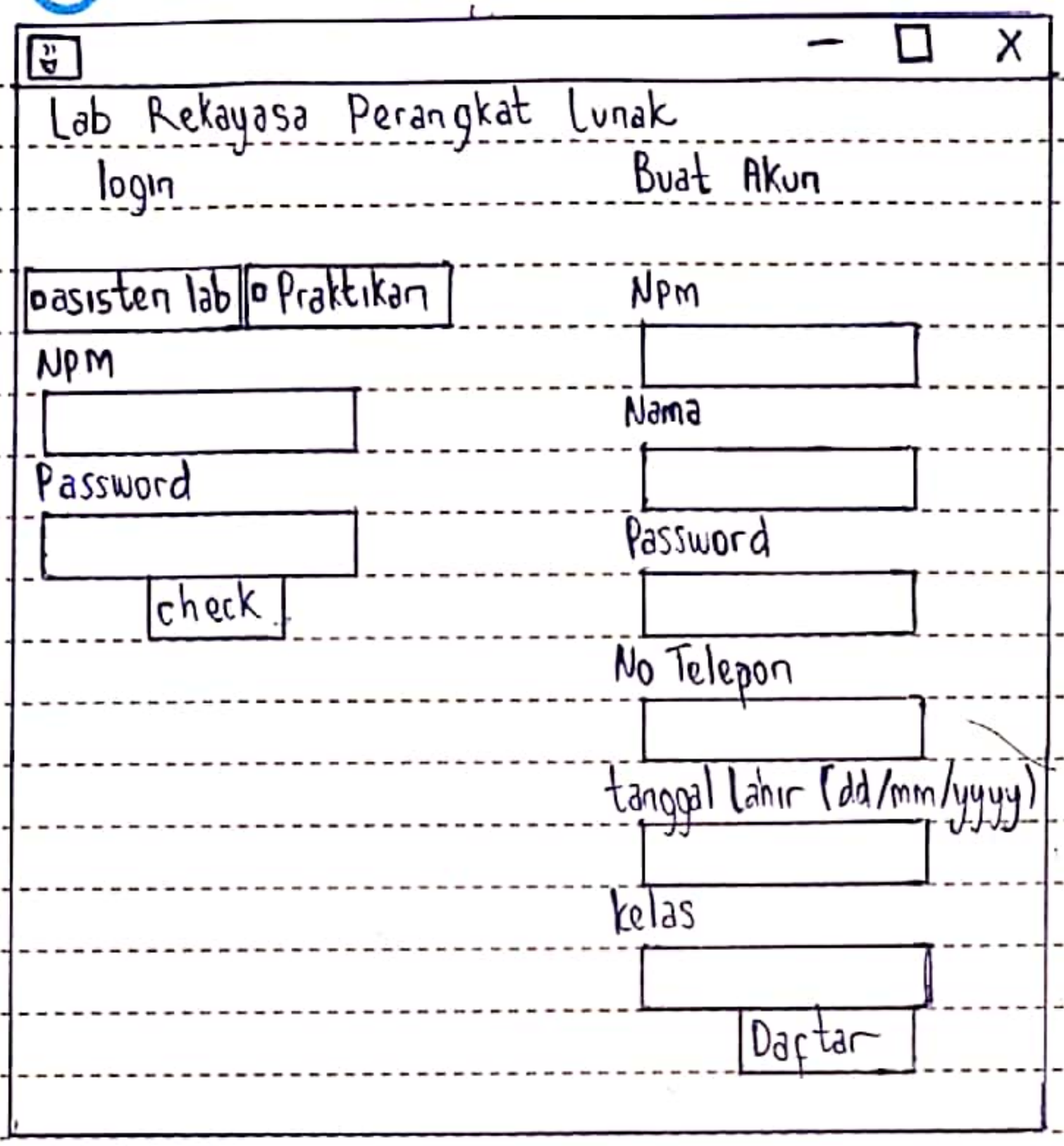
Class allobjctrl

```
1.    Package view;
2.    import Controller.AdminController;
3.    import Controller.PraktikanController;
4.    public class allobjctrl {
5.    public static AdminController admin = new
      AdminController();
6.    public static PraktikanController praktikan =
      new PraktikanController();
7.    }
```

- Buatlah tampilan seperti dibawah ini untuk membuat tampilan utama pada sistem informasi pendaftaran praktikum.

Dipindai dengan CamScanner

Tugas Pertanyaan



Dengan membuat class baru pada package view dengan nama class GUI seperti berikut:

Class GUI

```
1. Package view;
2. import java.awt.Color;
3. import java.awt.Font;
4. import java.awt.event.ActionEvent;
5. import java.awt.event.ActionListener;
6. import java.util.Date
7. import javax.swing.*;
8. public class GUI {
9.     JFrame LogReg = new JFrame();
```


Dipindai dengan CamScanner

Tugas Pertanyaan

```
JLabel login, daftar, top;
JRadioButton radioaslab, radio praktikan;
JTextfield textnpmlogin, textnama, textnotelp,
texttgllahir
, textnpmdaftar, textkelas;
JLabel labelnpmlogin, labelnama, labelpasswordlogin,
labelnotelp, labeltgllahir
, labelnpm daftar, labelpasswordddaftar, kelaslabel;
JButton check, reg;
JPasswordField passwordlogin, password daftar;
private boolean ceklogin = false;
public GUI (){
  logReg.setSize (700, 630);
  logReg.setLayout (null);
  logReg.getContentPane() setBackground (color.
PINK);
top = new Jlabel ("Lab Rekayasa Perangkat
lunak ");
top.setBounds (50, 10, 600, 50);
top.setFont (new Font ("Times New Roman", Font.
BOLD, 40));
logReg.add (top);
login = new Jlabel ("login");
login.setBounds (100, 50, 100, 100);
login.setFont (new Font ("Times New Roman", Font.
BOLD, 30));
logReg.add (login);
radioaslab = new JRadioButton ("asisten lab");
radioaslab.setBounds (30, 150, 100, 30);
radioaslab.setBackground (color.CYAN);
logReg.add (radioaslab);
radio praktikan = new JRadioButton ("Praktikan");
```


Dipindai dengan CamScanner

Tugas Pertanyaan

```
36.     radiopraktikan.setBounds (30,150 ,100,30);
37.     radiopraktikan.setBackground (color.CYAN);
38.     logReg.add (radiopraktikan)
39.     labelnpmlogin = new Jlabel ("NPM");
40.     labelnpmlogin.setBounds (30,180,30,30);
        logReg.add (labelnpmlogin);
41.     textnpmlogin = new JTextfield ();
42.     textnpmlogin.setBounds (30,210 ,200,30);
43.     logReg.add (textnpmlogin);
44.     labelpasswordlogin = new Jlabel ("Password");
45.     labelpasswordlogin.setBounds (30,240 ,100,30);
46.     logReg.add (labelpasswordlogin);
47.     passwordlogin = new JpasswordField ();
48.     passwordlogin.setBounds (30,270,200,30);
49.     logReg.add (passwordlogin);
50.     check = new JButton ("check")
51.     check.setBounds (90, 320, 100, 40);
52.     check.setBackground (color.GREEN);
53.     logReg.add (check);
54.      daftar = new Jlabel ("Buat Akun");
55.     daftar.setFont (new Font ("Times New Roman",
        Font.BOLD ,30));
56.     daftar.setBounds (440,50,150 ,100);
57.     logReg.add (daftar);
58.     labelnpmdaftar = new Jlabel ("NPM");
59.     labelnpmdaftar.setBounds (400,150,30,30);
60.     logReg.add (labelnpmdaftar);
61.     textnpmdaftar = new JTextfield ();
62.     textnpmdaftar.setBounds (400 180 ,200,30);
63.     logReg.add (textnpmdaftar)
64.     labelnama = new Jlabel ("Nama");
65.     labelnama.setBound (400 210 ,100,30);
```


Dipindai dengan CamScanner

Tugas Pertanyaan

```
66.     LogReg .add (labelnama);
67.     textnama = new JTextField ();
68.     textnama.setBounds (400,290,200,30);
69.     LogReg.add (textnama);
70.     labelpasworddaftar = new Jlabel ("Password");
71.     labelpasworddaftar.setBounds (400,270,100,30);
72.     LogReg.add (labelpasworddaftar);
73:     passwordaftar = new JPasswordField ();
74.     passwordaftar.setBounds (400,300,200,30);
75.     LogReg.add (passwordaftar);
76..    labelnotelp= new Jlabel ("No Telepon");
77.     labelnotelp.setBounds (400,330,100,30);
78.     LogReg.add (labelnotelp);
79.     textnotelp= new JTextField ();
80.     textnotelp.setBounds (400,360,200,30);
81.     LogReg.add (textnotelp);
82.     labeltgllahir = new Jlabel ("tanggal lahir (dd/mm
        /yyyy)");
83.     labeltgllahir.setBounds (400,390,200,30);
84.     LogReg.add (labeltgllahir);
85.     text tgllahir = new JTextField ();
86.     text tgllahir.setBounds (400,420,200,30);
87.     LogReg.add (texttgllahir);
88.     Kelaslabel= new Jlabel ("Kelas");
89.     Kelaslabel.setBounds (400,450,200,30);
90.     LogReg.add (kelaslabel);
91.     textkelas = new JTextField();
92.     textkelas.setBounds (400,480,200,30);
93.     LogReg.add (textkelas);
94.     reg=new JButton ("Daftar");
95.     reg.setBound (460,520,100,40);
96.     reg.setBackground (color.GREEN);
```


Dipindai dengan CamScanner

Tugas Pertanyaan

```
97.     logReg.add (reg);
98.     logReg.setDefaultCloseOperation (JFrame.
        EXIT_ON_CLOSE);
99.     logReg.setVisible (true);
100.    logReg.setLocationRelativeTo (null);
101.    radioaslab.addActionListener (new ActionListener())
102.    {
103.    @Override
104.    public void actionPerformed (ActionEvent ae){
105.    if (radioaslab.isSelected ()){
106     radiopraktikan.setSelected (false);
107.    ceklogin = true;
108.    }
109.    }
110.    });
111.    radiopraktikan.addActionListener (new actionlisten
        er ()
112.    {
113.    @Override
114.    public void actionPerformed (ActionEvent ae){
115.    if (radiopraktikan.isSelected ()){
116.        radioaslab.setSelected (false);
117.    ceklogin = false;
118.    }
119.    }
120.    });
121.    check.addActionListener (new ActionListener
        ()
122.    {
123.    @override
124.    public void actionPerformed (ActionEvent ae){
125.    if (ceklogin == true){
```


Dipindai dengan CamScanner

Tugas Pertanyaan

```
126.    try{
127.        allobjctrl.admin.dataAslab();
128.        allobjctrl.adminlogin(textnpmlogin.getText(),
        passwordlogin.getText());
129.        String nama = allobjctrl.admin.aslabentity().
        getNama();
130.        JOptionPane.showMessageDialog(null, "Selamat
        datang " + nama, "information", JOptionPane.INF
        ORMATION_MESSAGE);
131.    AslabGUI asisten = new AslabGUI();
132.    LogReg.dispose();
133.    }catch (Exception exception){
134.    JOptionPane.showMessageDialog(null, "NPM atau
        password salah", "information", JOptionPane.
        INFORMATION_MESSAGE);
135.    Kosong();
136.    }
137.    }else{
138.    try{
139.    allobjctrl.praktikan.login(textnpmlogin.get
        Text(), passwordlogin.getText());
140.    String nama = allobjctrl.praktikan.praktikan
        Entity().getNama();
141.    JOptionPane.showMessageDialog(null, "Selamat
        Datang " + nama, "Information", JOptionPane.INFO
        RMATION_MESSAGE);
142.    PraktikanGUI prak = new PraktikanGUI();
143.    LogReg.dispose();
144.    }catch (Exception exception){
145.    JOptionPane.showMessageDialog(null, "NPM
        atau password salah", "information", JOption
        Pane.INFORMATION_MESSAGE
```


Dipindai dengan CamScanner

Tugas Pertanyaan

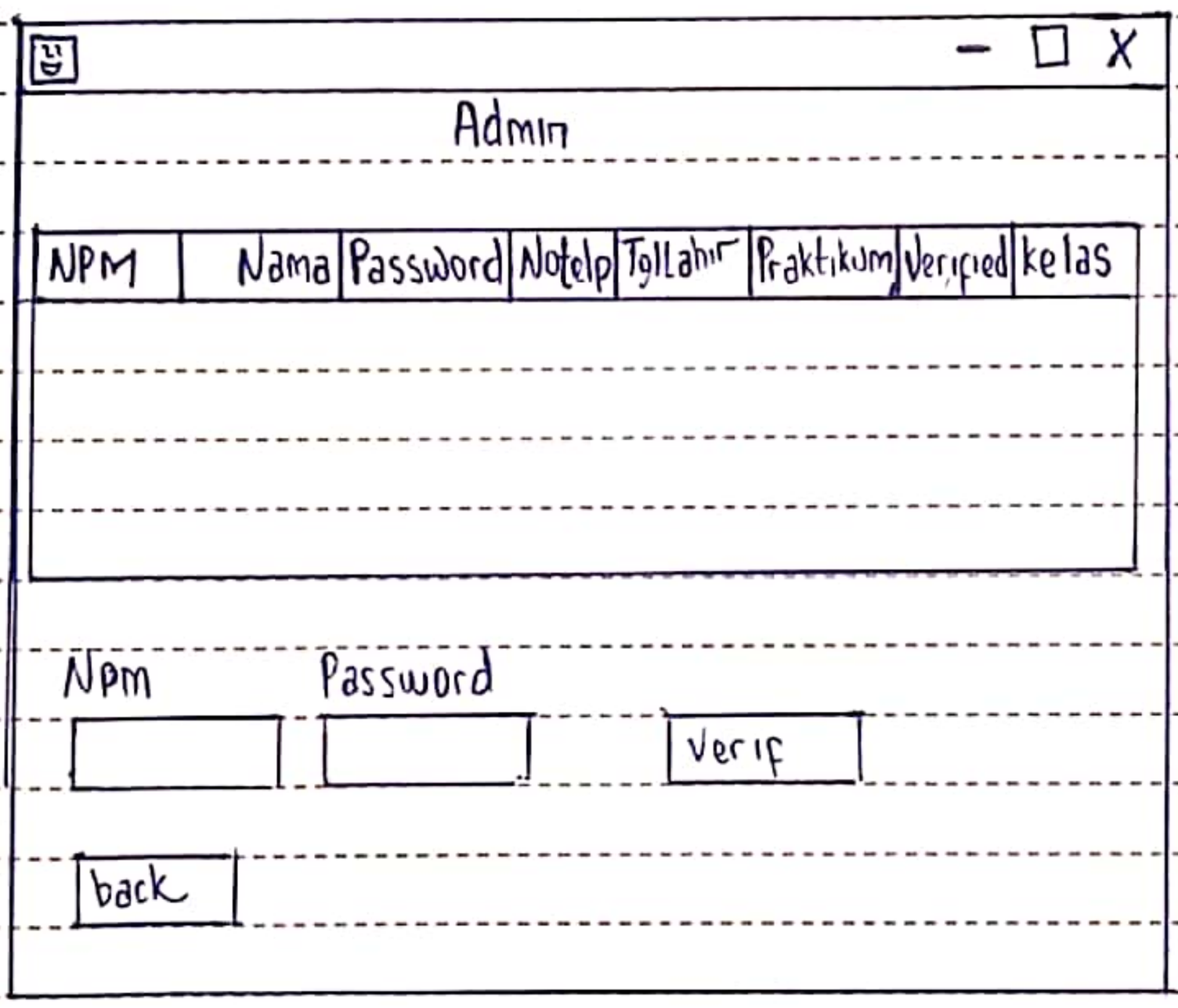
```
146.        kosong ();
147.    }
148.    }
149.    }
150.    });
151.    reg.addActionListener (new ActionListener()
152.    {
153.    @Override
154.    public void actionPerformed (ActionEvent ae)
        {
155.    try{
156.    String npm = textnpmdaftar.getText ();
157.    String pass = passworddaftar.getText ();
158.    String notelp = textnotelp.getText ();
159.    Date tgllahir = new Date (texttgllahir.getTe
        xt ());
160.    String kelas = textkelas.getText ();
161.    allobjctrl.Praktikan.insert (npm,pass,nama,no
        telp,tgllahir,kelas);
162.    JOptionPane.showMessageDialog (null, "Registrasi
        Sukses", "Information", JOptionPane.INFORMATION
        _MESSAGE);
163.    }
164.    }
165.    });
166.    }
167.    void kosong (){
168.    textnpmdaftar.setText (null);
169.    passworddaftar.setText (null);
170.    text nama.setText (null);
171.    textnotelp.setText (null);
172.    texttgllahir.setText (null);
```


Dipindai dengan CamScanner

Tugas Pertanyaan

```
173.        textnpmlogin.setText(null);
174.        passwordlogin.setText(null);
175.        textkelas.setText(null);
176.    }
177.    }
```

- Buat tampilan untuk class AslabGUI seperti di bawah ini sesuai dengan source code dibawah :



Class AslabGUI

```
1. package view;
2. import Controller.AllObjectModel;
3. import java.awt.event.ActionEvent;
4. import java.awt.event.ActionListener;
5. import javax.swing.*;
6. import java.awt.color;
7. import javax.swing.table.DefaultTableModel;
```


Dipindai dengan CamScanner

Tugas Pertanyaan

```
import Controller.praktikanController;
import Java.awt.font;
import Java.awt.event.MouseAdapter;
import Java.awt.event.MouseEvent;
public class AslabGui {
private static PraktikanController praktikan =
new PraktikanController();
JFrame Aslab = new JFrame();
JTable tabelpraktikan = new JTable();
JScrollPane scrollaslab = new JscrollPane(tabe
lpraktikan);
JButton back, verif;
Jlabel admin, npmlabel, passlabel;
JTextField npmtext, passtext;
public AslabGui() {
Aslab.setsize(650,500);
Aslab.setlayout(null);
Aslab.getContentPane().setBackground(color
.PINK);
admin = new Jlabel("Admin");
admin.setBounds(250,30,200,30);
admin.setFont("Times New Roman", Font.
BOLD,40));
Aslab.add(admin);
Scrollaslab.setBounds(30,100,570,200);
tabelpraktikan.setModel(praktikan.daftarprak
());
aslab.add(scrollaslab);
npmlabel = new JLabel("NPM");
npmlabel.setBounds(50,320,100,30);
Aslab.add(npmlabel);
npmtext = new JTextfield();
```


Dipindai dengan CamScanner

Tugas Pertanyaan

```
35.     npm.text .setBounds (200,350,100,30);
36.     Aslab.add (npmtext);
37.     passlabel =new Jlabel ("password");
38.     passlabel. SetBounds (200,320,100,30);
39.     Aslab. add (passlabel);
40.     passtext = new JTextField ();
41.     passtext .setBounds (200,350 ,100,30);
42.      Aslab.add (passtext);
43.     verif= new JButton ("Verif");
44.     verif.setBounds (350,350,100,30);
45.     verif. setBackground (color. GREEN);
46.    Aslab.add (verif);
47.     back=new JButton ("back");
48.     back. setBackground (color. RED);
49.     Aslab.add (back);
50.     Aslab.setLocationRelativeTo (null);
51.     Aslab. setVisible (true);
52.      Aslab. set Default Close Operation (JFrame .EXIT_
        ON_CLOSE);
53.     back .add ActionListener (new ActionListener()
54.     {
55.     @Override
56.       public void actionPerformed (ActionEvent ae){
57.     Aslab. dispose ();
58.      GUI men = new GUI ();
59.      }
60.    });
61.     verif.addActionListener (new Action listener ()
62.     {
63.     @Override
64.     public void actionPerformed (ActionEvent ae){
65.      String npm =npmtext. getText ();
```


Dipindai dengan CamScanner

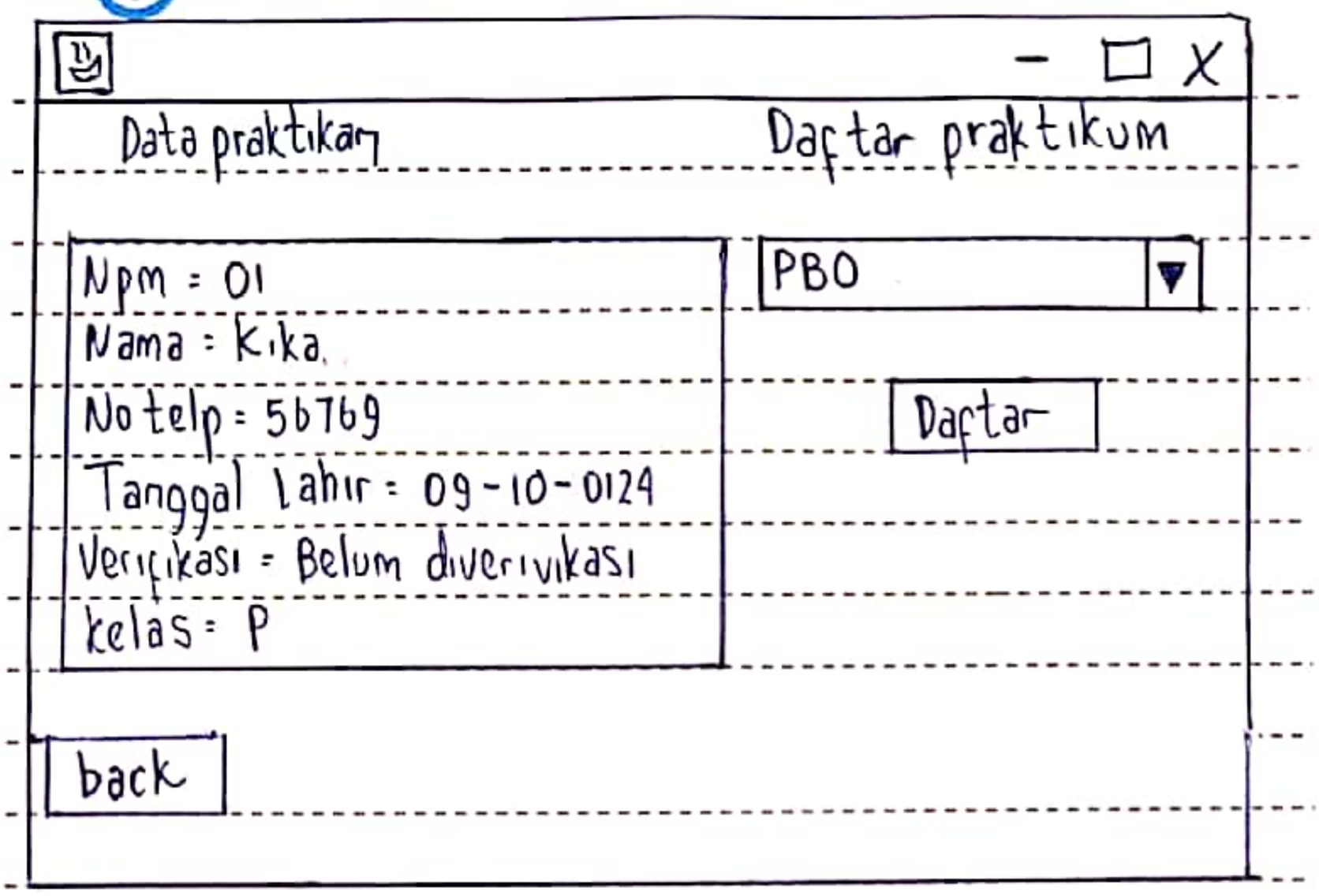
Tugas Pertanyaan

```
66.     String pass = passtext.getText();
67.     int index = AllObjectModel.daftarprakModel.
        cekdata(ripm,pass);
68.     allobjctrl.admin.updateIsVerified(index,allobj
        ctrl.Praktikan.showDaftarprak(index).getIndex
        Prak(),allobjctrl.Praktikan.showDaftarprak
        (index).getPraktikan());
69.     tabelpraktikan.setModel(praktikan.daftarprak
        ());
70.     }
71.     });
72.     tabelpraktikan.addMouseListener(new Mouse
        Adapter(){
73.     @Override
74.     public void mouseClicked(MouseEvent e){
75.     int i = tabelpraktikan.getSelectedRow();
76.     npmtext.setText(allobjctrl.Praktikan.daftar
        prak().getValueAt(i,0).toString());
77.     passtext.setText(allobjctrl.Praktikan.daftar
        prak().getValueAt(i,1).toString());
78.     }
79.     });
80.     }
81.     }
```

- Buat tampilan untuk pendaftaran praktikan seperti dibawah ini, dengan membuat class baru dengan nama PraktikanGUI dengan menggunakan source code dibawah :

Dipindai dengan CamScanner

Tugas Pertanyaan



Class PraktikanGUI

```
1.  public class PraktikanGUI {
2.      JFrame Praktikan = new JFrame();
3.      JButton back, daftarbtn;
4.      JTextArea area = new JTextArea();
5.      JLabel datadiri, daftarpraktikum;
6.      JcomboBox pilprak = new JcomboBox(Praktikum
        Entity.nama);
7.      int cek = allobjctrl.Praktikan.cekDaftarprak(allobj
        ctrl.Praktikan.praktikanEntity().getNpm());
8.      public PraktikanGUI(){
9.      praktikan.setSize(720, 600);
10.     praktikan.setLayout(null);
11.     Praktikan.getContentPane().setBackground(
        color.PINK);
12.     datadiri = new JLabel("Data Praktikan");
13.     datadiri.setFont(new Font("Times New Roman",
        Font.BOLD, 30));
14.     datadiri.setBounds(90,30,200,30);
```


Dipindai dengan CamScanner

Tugas Pertanyaan

```
15.     Praktikan.add (datadiri);
16.     area.setBounds (30,90,400,300);
17.     Praktikan.add (area)
18.     daftarpraktikum = new JLabel ("Daftar Praktikum");
19.     daftarpraktikum.setBounds (450,30,250,30);
20.     daftarpraktikum.setFont (new Font ("Times New Roman", Font.BOLD,30));
21.     Praktikan.add (daftarpraktikum);
22.     pilprak.setBounds (450,90,230,30);
23.     Praktikan.add (pilprak);
24.     daftarbtn = new JButton ("Daftar");
25.     daftarbtn.setBounds (500,200,100,30);
26.     daftarbtn.setBackground (color.GREEN);
27.     Praktikan.add (daftarbtn);
28.     back = new JButton ("back");
29.     back.setBounds (30,500,100,30);
30.     back.setBackground (color.red);
31.     Praktikan.add(back);
32.     Praktikan.setVisible (true);
33.     Praktikan.setLocationRelativeTo (null);
34.     Praktikan.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
35.     if (cek==-1){
36.     JOptionPane.showMessageDialog (null, "Anda Belum Daftar praktikum", "information", JOptionPane.INFORMATION_MESSAGE);
37.     }else{
38.     area.setText (datapraktikan ());
39.     }
40.     daftarbtn.addActionListener (new ActionListener ()
```


Dipindai dengan CamScanner

Tugas Pertanyaan

```
41.     {
42.     @Override
43.       public void actionPerformed (ActionEvent ae){
44.       int indexprak = pilprak.getSelectedIndex();
45.     allobjctrl.Praktikan.daftarPraktikum (indexprak,
        allobjctrl.Praktikan.getData(), false);
46.     area.setText (datapraktikan());
47.     }
48.     });
49.   back.addActionListener (new ActionListener()
50.   {
51.   @Override
52.   public void actionPerformed (ActionEvent ae){
53.     Praktikan.dispose();
54.     Gui men = new Gui();
55.     }
56.     });
57.   }
58.     String datapraktikan(){
59.     int cek = allobjctrl.Praktikan.cekDaftarprak
        (allobjctrl.praktikan.praktikanEntity().getNpm
        ());
60.     String cekverif;
61.     if (allobjctrl.Praktikan.ShowDaftarprak (cek).
        isVerified() == false){
62.       cekverif = "belum diverifikasi";
63.     } else{
64.       cekverif = "sudah diverifikasi";
65.     }
66.     String text = "Npm = " + allobjctrl.Praktikan.
        showDaftarprak (cek).getPratikan().getNpm()+
        "\n"
```


Dipindai dengan CamScanner

Tugas Pertanyaan

```
67.     + "Nama = " + allobjctrl.Praktikan.ShowDaftarprak
        (cek).getPratikan().getNama() + "\n"
68.     + "No telp = " + allobjctrl.Praktikan.showDaftarprak
        (cek).getPratikan().getNo_Telp() + "\n"
69.     + "Tanggal Lahir = " + new SimpleDateFormat("dd-
        MM-yyyy").format(allobjctrl.Praktikan.show
        Daftarprak(cek).getPratikan().getTglLahir()) +
        "\n"
70.     + "verifikasi = " + cekverif + "\n"
71.     + "kelas = " + allobjctrl.Praktikan.showDaftarprak
        (cek).getPratikan().getkelas() + "\n"
72.     + "praktikum = " + PraktikumEntity.nama[allobjctrl.
        Praktikan.ShowDaftarprak(cek).getIndexPrak()];
73.     return text
74.     }
75. }
```

- Selanjutnya pada class main.java akan berbeda dari modul sebelumnya. Tulis kode baru ini pada Main.java sebagai berikut:

Class Main.java

```
1. package Project;
2. import view.GUI;
3. public class Main {
4.     public static void main(String[]args) {
5.         GUI gui = new GUI();
6.     }
7. }
```

4. Jalankan dan periksa kembali programnya.

Dipindai dengan CamScanner

Tugas Praktikum

Soal Praktikum

- *Terapkan GUI sesuai dengan project masing-masing.*
- *Terapkan CRUD di GUI project kalian.*
- *Terapkan JOptionPane ke GUI kalian.*

Langkah Pertama

Membuat package view untuk mengatur tampilan gui lalu menambahkan beberapa class salah satunya class Alex07133_PetugasGUI

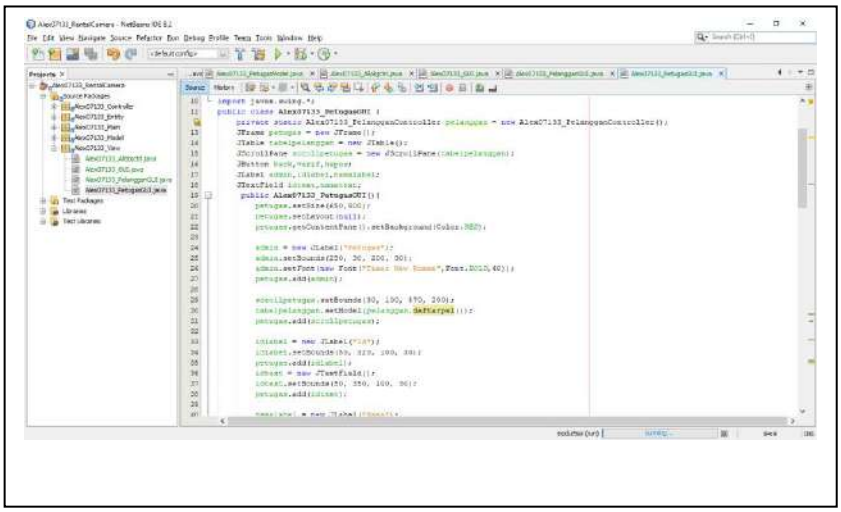
Source Code

```
package Alex07133_View;
import Alex07133_Controller.Alex07133_AllObjectModel;
import
Alex07133_Controller.Alex07133_PelangganController;
import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.*;
public class Alex07133_PetugasGUI {
    private static Alex07133_PelangganController
pelanggan = new Alex07133_PelangganController();
    JFrame petugas = new JFrame();
    JTable tabelpelanggan = new JTable();
    JScrollPane scrollpetugas = new
JScrollPane(tabelpelanggan);
    JButton back,verif,hapus;
    JLabel admin,idlabel,namalabel;
    JTextField idtext,namatext;
     public Alex07133_PetugasGUI(){
        petugas.setSize(650,500);
        petugas.setLayout(null);

petugas.getContentPane().setBackground(Color.RED);

        admin = new JLabel("Petugas");
        admin.setBounds(250, 30, 200, 30);
        admin.setFont(new Font("Timer New
Roman",Font.BOLD,40));
```

```
scrollpetugas.setBounds(30, 100, 570, 200);
        tabelpelanggan.setModel(pelanggan.daftarpel());
        petugas.add(scrollpetugas);

        idlabel = new JLabel("Id");
        idlabel.setBounds(50, 320, 100, 30);
        petugas.add(idlabel);
        idtext = new JTextField();
        idtext.setBounds(50, 350, 100, 30);
        petugas.add(idtext);

        namalabel = new JLabel("Nama");
        namalabel.setBounds(200, 320, 100, 30);
        petugas.add(namalabel);
        namatext = new JTextField();
        namatext.setBounds(200, 350, 100, 30);
        petugas.add(namatext);

        verif = new JButton("verif");
        verif.setBounds(350, 350, 100, 30);
        verif.setBackground(Color.GRAY);
        petugas.add(verif);

        back = new JButton("back");
        back.setBounds(50, 400, 100, 30);
        back.setBackground(Color.GRAY);
        petugas.add(back);

        hapus = new JButton("Hapus");
        hapus.setBounds(350, 410, 100, 30);
        hapus.setBackground(Color.GRAY);
        petugas.add(hapus);

        petugas.setLocationRelativeTo(null);
        petugas.setVisible(true);

petugas.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        back.addActionListener(new ActionListener(){
            @Override
            public void actionPerformed(ActionEvent
ae){
                petugas.dispose();
                Alex07133_GUI men = new
Alex07133_GUI();//nama objek men
            }
        });
```

Tugas Praktikum

Screenshot



Analisa

Pada tahap Pembuatan class Alex07133_PetugasGUI ini berfungsi untuk mengsahilkan tampilan pada saat kita menjalankan program dan masuk ke dalam menu bagian admin

Tugas Praktikum

Langkah Kedua

Membuat class Alex07133_PelangganGUI untuk memberikan tampilan GUI di dalam menu pelanggan

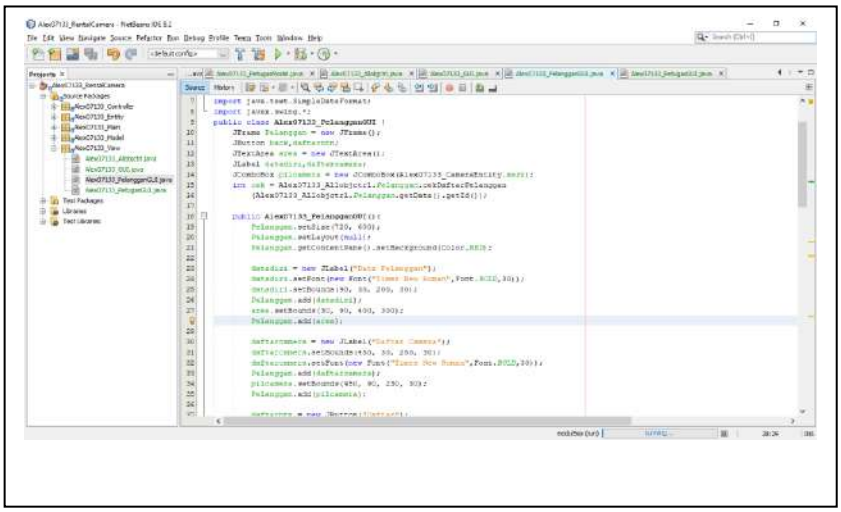
Source Code

```
package Alex07133_View;
import Alex07133_Entity.Alex07133_CameraEntity;
import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.text.SimpleDateFormat;
import javax.swing.*;
public class Alex07133_PelangganGUI {
    JFrame Pelanggan = new JFrame();
    JButton back,daftarbtn;
    JTextArea area = new JTextArea();
    JLabel datadiri,daftarcamera;
    JComboBox pilcamera = new
JComboBox(Alex07133_CameraEntity.merk);
    int cek =
Alex07133_Allobjctrl.Pelanggan.cekDaftarPelanggan

(Alex07133_Allobjctrl.Pelanggan.getData().getId());

    public Alex07133_PelangganGUI(){
        Pelanggan.setSize(720, 600);
        Pelanggan.setLayout(null);

Pelanggan.getContentPane().setBackground(Color.RED);

        datadiri = new JLabel("Data Pelanggan");
        datadiri.setFont(new Font("Timer New
Roman",Font.BOLD,30));
        datadiri.setBounds(90, 30, 200, 30);
        Pelanggan.add(datadiri);
        area.setBounds(30, 90, 400, 300);
        Pelanggan.add(area);

        daftarcamera = new JLabel("Daftar Camera");
        daftarcamera.setBounds(450, 30, 250, 30);
        daftarcamera.setFont(new Font("Times New
Roman",Font.BOLD,30));
        Pelanggan.add(daftarcamera);
        pilcamera.setBounds(450, 90, 230, 30);
```

```
        Pelanggan.add(pilcamera);

        daftarbtn = new JButton("Daftar");
        daftarbtn.setBounds(500, 200, 100, 30);
        daftarbtn.setBackground(Color.GRAY);
        Pelanggan.add(daftarbtn);

        back = new JButton("Back");
        back.setBounds(30, 500, 100, 30);
        back.setBackground(Color.GRAY);
        Pelanggan.add(back);

        Pelanggan.setVisible(true);
        Pelanggan.setLocationRelativeTo(null);

Pelanggan.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE
);
        if(cek==-1){ //cek data, apabila kita belum
daftar praktkum
            JOptionPane.showMessageDialog(null, "Anda
Belum Daftar Camera", "Information",
JOptionPane.INFORMATION_MESSAGE);
        }else if(cek==-2){
            JOptionPane.showMessageDialog(null,"Anda
Belum Daftar Camera", "Information",
JOptionPane.INFORMATION_MESSAGE);
        }else{
            area.setText(datapelanggan());
        }
        daftarbtn.addActionListener(new
ActionListener(){
        @Override
        public void actionPerformed(ActionEvent ae){
            int indexcamera =
pilcamera.getSelectedIndex();

Alex07133_Allobjctrl.Pelanggan.Alex07133_DaftarCamera

(indexcamera,Alex07133_Allobjctrl.Pelanggan.getData(),
false);
            area.setText(datapelanggan());
        }
        });

        back.addActionListener(new ActionListener(){
            @Override
```

Tugas Praktikum

Screenshot



Analisa

Padal class Alex07133_PelangganGUI disni kita membuat tampilan gui pada menu ketika kita masuk menu masuk sebagai praktikan

Tugas Praktikum

Langkah Ketiga

Membuat class Alex07133_GUI untuk memberikan tampilan GUI di menu awal

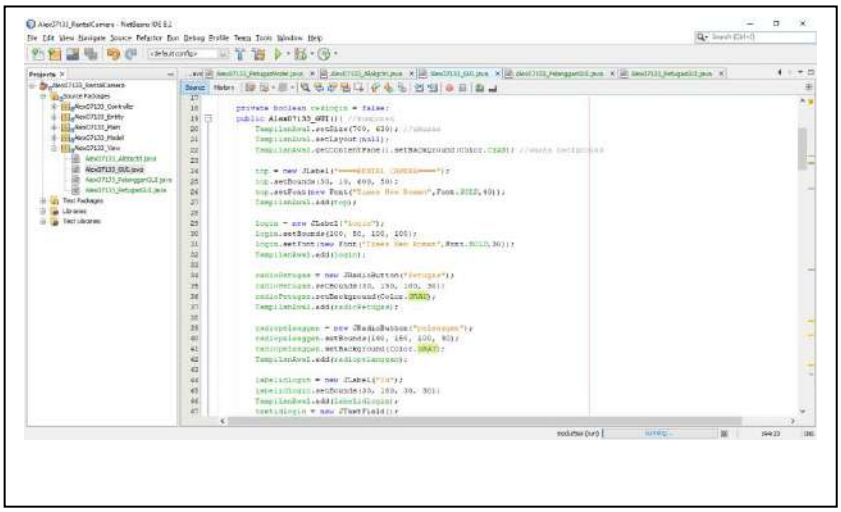
Source Code

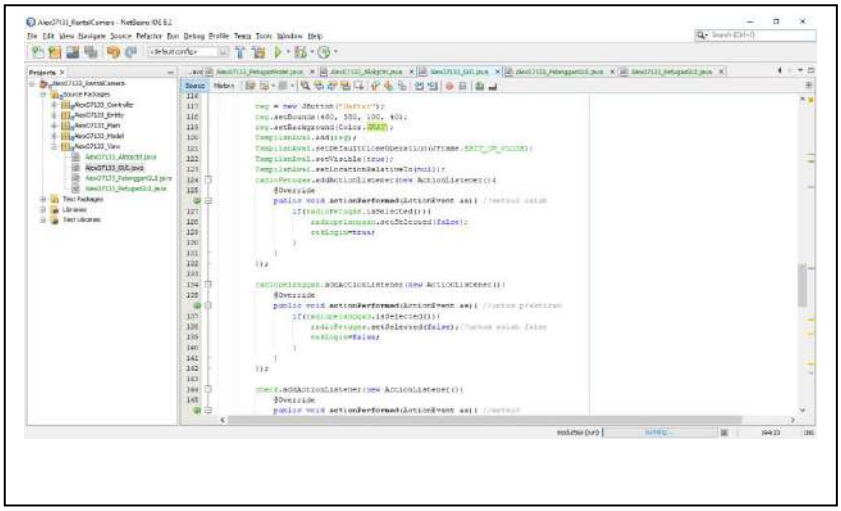
```
 package Alex07133_View;
import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Date;
import javax.swing.*;
public class Alex07133_GUI {
    JFrame TampilanAwal = new JFrame();//tampilan awal
    JLabel login,daftar,top;
    JRadioButton radioPetugas,radiopelanggan; //tombol
pilihan
    JTextField
textidlogin,textnamalogin,textalamat,textnoidentitas,te
xtnotelp,texttglsewa,

texttglkembali,textiddaftar,textnamadaftar;
    JLabel
labelidlogin,labelnamalogin,labelalamat,labelnoidentita
s,labelnotelp,labeltglsewa,

labeltglkembali,labeliddaftar,labelnamadaftar;
    JButton check,reg; //tombol

    private boolean ceklogin = false;
    public Alex07133_GUI(){ //komponen
        TampilanAwal.setSize(700, 630); //ukuran
        TampilanAwal.setLayout(null);

TampilanAwal.getContentPane().setBackground(Color.CYAN)
; //warna background

        top = new JLabel("====RENTAL CAMERA====");
        top.setBounds(50, 10, 600, 50);
        top.setFont(new Font("Times New
Roman",Font.BOLD,40));
        TampilanAwal.add(top);

        login = new JLabel("login");
        login.setBounds(100, 50, 100, 100);
        login.setFont(new Font("Times New
```

```
 TampilanAwal.add(top);

        login = new JLabel("login");
        login.setBounds(100, 50, 100, 100);
        login.setFont(new Font("Times New
Roman",Font.BOLD,30));
        TampilanAwal.add(login);

        radioPetugas = new JRadioButton("Petugas");
        radioPetugas.setBounds(30, 150, 100, 30);
        radioPetugas.setBackground(Color.GRAY);
        TampilanAwal.add(radioPetugas);

        radiopelanggan = new JRadioButton("pelanggan");
        radiopelanggan.setBounds(140, 150, 100, 30);
        radiopelanggan.setBackground(Color.GRAY);
        TampilanAwal.add(radiopelanggan);

        labelidlogin = new JLabel("Id");
        labelidlogin.setBounds(30, 180, 30, 30);
        TampilanAwal.add(labelidlogin);
        textidlogin = new JTextField();
        textidlogin.setBounds(30, 210, 200, 30);
        TampilanAwal.add(textidlogin);

        labelnamalogin = new JLabel("Nama");
        labelnamalogin.setBounds(30, 240, 100, 30);
        TampilanAwal.add(labelnamalogin);
        textnamalogin = new JTextField();
        textnamalogin.setBounds(30, 270, 200, 30);
        TampilanAwal.add(textnamalogin);

        check = new JButton("check");
        check.setBounds(90, 320, 100, 40);
        check.setBackground(Color.GRAY);
        TampilanAwal.add(check);

        daftar = new JLabel("Daftar");
        daftar.setFont(new Font("Times New
Roman",Font.BOLD,30));
        daftar.setBounds(440, 50, 150, 100);
        TampilanAwal.add(daftar);

        labeliddaftar = new JLabel("Id");
        labeliddaftar.setBounds(400, 150, 30, 30);
        TampilanAwal.add(labeliddaftar);
```

Tugas Praktikum

Screenshot





Analisa

Padal class Alex07133_GUI disni kita membuat tampilan gui pada menu awal ketika kita menjalankan program

Tugas Praktikum

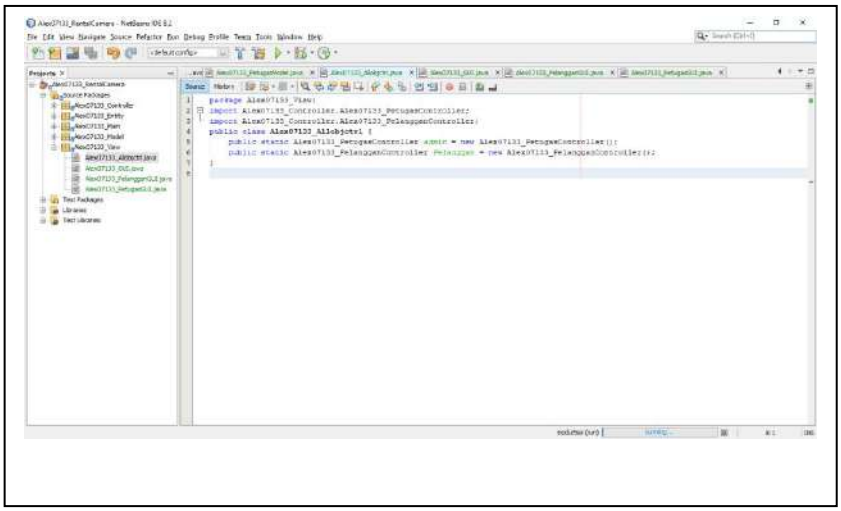
Langkah Keempat

Membuat class Alex07133_Allobjctrl untuk menggabungkan class GUI agar terhubung ke dalam Controller

Source Code

```
package Alex07133_View;
import
Alex07133_Controller.Alex07133_PetugasController;
import
Alex07133_Controller.Alex07133_PelangganController;
public class Alex07133_Allobjctrl {
    public static Alex07133_PetugasController admin =
new Alex07133_PetugasController();
    public static Alex07133_PelangganController
Pelanggan = new Alex07133_PelangganController();
}
```

Screenshot



Analisa

Padal class Alex07133_Allobjctrl disni kita membuat tampilan gui supaya dapat terhubung ke kontroller

Tugas Praktikum

Langkah Kelima

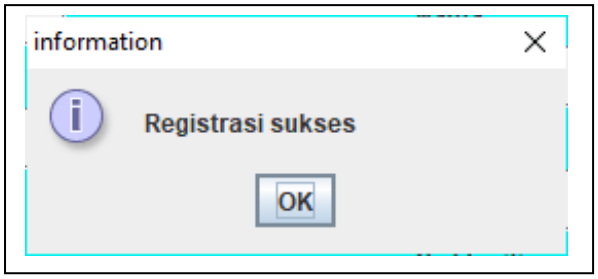
Menambahkan JOptionPane pada GUI

Source Code

```
@Override
            public void actionPerformed(ActionEvent
ae){ //method
                if(ceklogin==true){
                    try{

Alex07133_Allobjctrl.admin.datapetugas(); // login
untuk petugas

Alex07133_Allobjctrl.admin.login(textidlogin.getText(),
textnamalogin.getText());
                        String nama =
Alex07133_Allobjctrl.admin.Alex07133_PetugasEntity().ge
tnama(); //memasukkan id,nama, jika benar

JOptionPane.showMessageDialog(null, "Selamat datang
"+nama,"information", JOptionPane.INFORMATION_MESSAGE);
                        Alex07133_PetugasGUI petugas =
new Alex07133_PetugasGUI();
                        TampilanAwal.dispose();
                    }catch(Exception exception){ //jika
id/nama salah makan akan muncul pesan

JOptionPane.showMessageDialog(null, "id atau nama
salah", "information",
JOptionPane.INFORMATION_MESSAGE);
                    }
                        kosong();//method kosong, jika
datanya kosong
                }else{
                    try{ //login untuk pelanggan

Alex07133_Allobjctrl.Pelanggan.login(textidlogin.getTex
t(),textnamalogin.getText());
                       String nama =
Alex07133_Allobjctrl.Pelanggan.getData().getnama();
//memasukkan id,nama, jika benar

JOptionPane.showMessageDialog(null, "Selamat datang
"+nama,"information", JOptionPane.INFORMATION_MESSAGE);
                       Alex07133_PelangganGUI  pel =
```

Tugas Praktikum

Screenshot



Analisa

Joptionpane disni saya gunakan ketika berhasil mendaftarkan pelanggan dan juga masuk pelanggan pada saat berhasil registrasi

Tugas Praktikum

Langkah Keenam

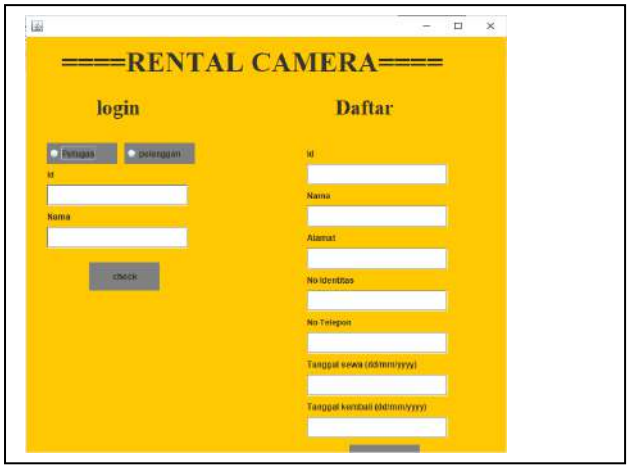
Menerapkan CRUD pada program GUI

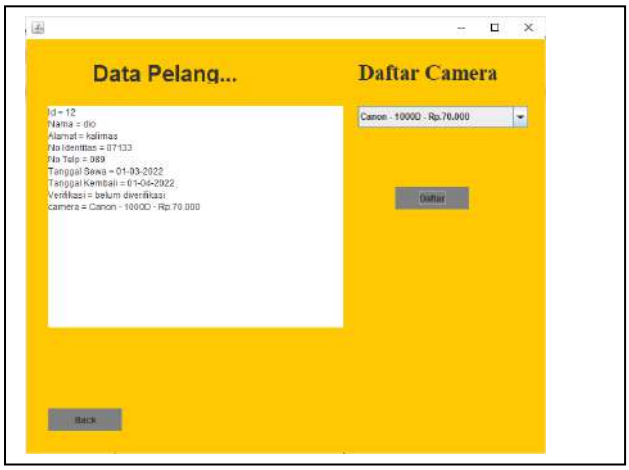
Source Code

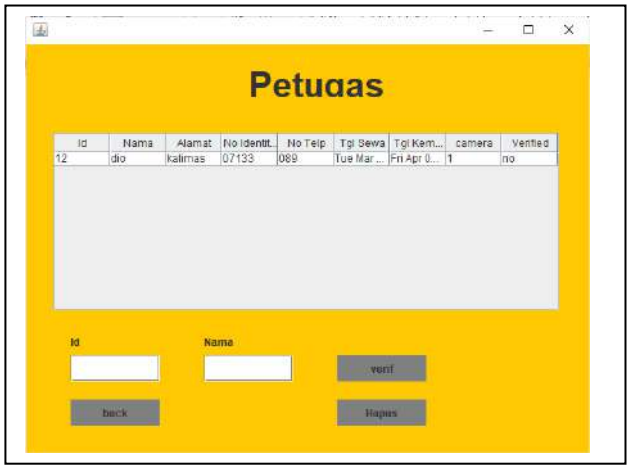
```
        verif.addActionListener(new ActionListener(){
            @Override
            public void actionPerformed(ActionEvent
ae){ //untuk verifikasi id, dan nama
                String id = idtext.getText();
                String nama = namatext.getText();
                int index =
Alex07133_AllObjectModel.daftarPelangganmodel.cekData(i
d, nama); //dicek

Alex07133_Allobjctrl.admin.updateIsVerified(index,
Alex07133_Allobjctrl.Pelanggan.showDaftarPelanggan(inde
x)

.getIndexCamera(),Alex07133_Allobjctrl.Pelanggan.showDa
ftarPelanggan(index).getPelanggan());

tabelpelanggan.setModel(pelanggan.daftarpel());
            }
        });

        hapus.addActionListener(new ActionListener(){
            @Override
            public void actionPerformed(ActionEvent
ae){ //untuk verifikasi id, dan nama
                String id = idtext.getText();
                int index =
Alex07133_AllObjectModel.daftarPelangganmodel.alldatape
langgan().size(); //dicek

Alex07133_Allobjctrl.Pelanggan.deleteDataPelanggan(inde
x);

tabelpelanggan.setModel(pelanggan.daftarpel());
            }
        });

        tabelpelanggan.addMouseListener(new
MouseAdapter(){
            @Override
```

Tugas Praktikum

Screenshot







Analisa

Proses CRUD dapat di akses melalui GUI dengan tampilan dan support untuk mouse klik (ActionListener)

Laporan Sementara

Soal Praktikum

- *Terapkan GUI sesuai dengan project masing-masing.*
- *Terapkan CRUD di GUI project kalian.*
- *Terapkan JOptionPane ke GUI kalian.*

Langkah Pertama

Membuat package view untuk mengatur tampilan gui lalu menambahkan beberapa class salah satunya class Alex07133_PetugasGUI

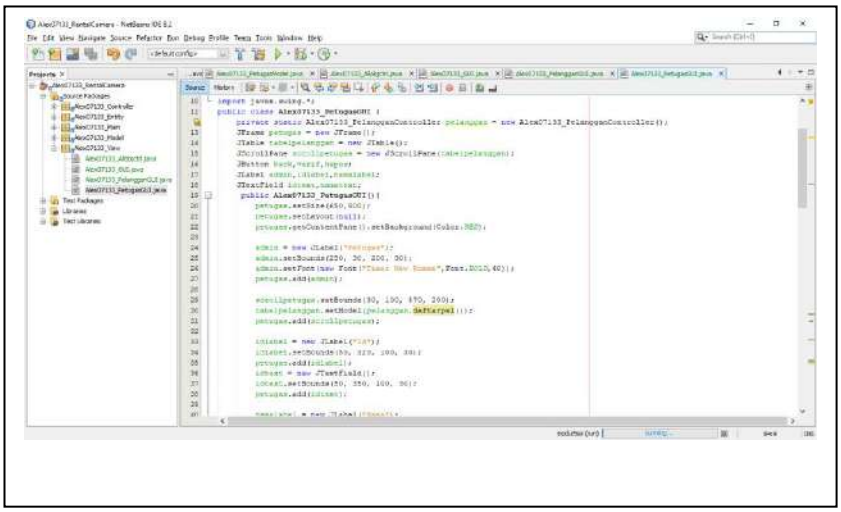
Source Code

```
package Alex07133_View;
import Alex07133_Controller.Alex07133_AllObjectModel;
import
Alex07133_Controller.Alex07133_PelangganController;
import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.*;
public class Alex07133_PetugasGUI {
    private static Alex07133_PelangganController
pelanggan = new Alex07133_PelangganController();
    JFrame petugas = new JFrame();
    JTable tabelpelanggan = new JTable();
    JScrollPane scrollpetugas = new
JScrollPane(tabelpelanggan);
    JButton back,verif,hapus;
    JLabel admin,idlabel,namalabel;
    JTextField idtext,namatext;
     public Alex07133_PetugasGUI(){
        petugas.setSize(650,500);
        petugas.setLayout(null);

petugas.getContentPane().setBackground(Color.RED);

        admin = new JLabel("Petugas");
        admin.setBounds(250, 30, 200, 30);
        admin.setFont(new Font("Timer New
Roman",Font.BOLD,40));
```

```
scrollpetugas.setBounds(30, 100, 570, 200);
        tabelpelanggan.setModel(pelanggan.daftarpel());
        petugas.add(scrollpetugas);

        idlabel = new JLabel("Id");
        idlabel.setBounds(50, 320, 100, 30);
        petugas.add(idlabel);
        idtext = new JTextField();
        idtext.setBounds(50, 350, 100, 30);
        petugas.add(idtext);

        namalabel = new JLabel("Nama");
        namalabel.setBounds(200, 320, 100, 30);
        petugas.add(namalabel);
        namatext = new JTextField();
        namatext.setBounds(200, 350, 100, 30);
        petugas.add(namatext);

        verif = new JButton("verif");
        verif.setBounds(350, 350, 100, 30);
        verif.setBackground(Color.GRAY);
        petugas.add(verif);

        back = new JButton("back");
        back.setBounds(50, 400, 100, 30);
        back.setBackground(Color.GRAY);
        petugas.add(back);

        hapus = new JButton("Hapus");
        hapus.setBounds(350, 410, 100, 30);
        hapus.setBackground(Color.GRAY);
        petugas.add(hapus);

        petugas.setLocationRelativeTo(null);
        petugas.setVisible(true);

petugas.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        back.addActionListener(new ActionListener(){
            @Override
            public void actionPerformed(ActionEvent
ae){
                petugas.dispose();
                Alex07133_GUI men = new
Alex07133_GUI();//nama objek men
            }
        });
```

Laporan Sementara

Screenshot



Analisa

Pada tahap Pembuatan class Alex07133_PetugasGUI ini berfungsi untuk mengsahilkan tampilan pada saat kita menjalankan program dan masuk ke dalam menu bagian admin

Langkah Kedua

Membuat class Alex07133_PelangganGUI untuk memberikan tampilan GUI di dalam menu pelanggan

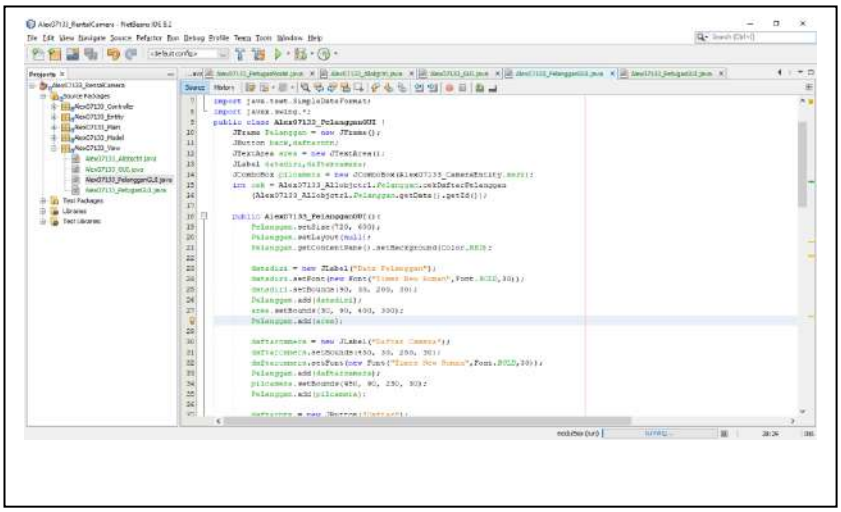
Source Code

```
package Alex07133_View;
import Alex07133_Entity.Alex07133_CameraEntity;
import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.text.SimpleDateFormat;
import javax.swing.*;
public class Alex07133_PelangganGUI {
    JFrame Pelanggan = new JFrame();
    JButton back,daftarbtn;
    JTextArea area = new JTextArea();
    JLabel datadiri,daftarcamera;
    JComboBox pilcamera = new
JComboBox(Alex07133_CameraEntity.merk);
    int cek =
Alex07133_Allobjctrl.Pelanggan.cekDaftarPelanggan

(Alex07133_Allobjctrl.Pelanggan.getData().getId());

    public Alex07133_PelangganGUI(){
        Pelanggan.setSize(720, 600);
        Pelanggan.setLayout(null);

Pelanggan.getContentPane().setBackground(Color.RED);

        datadiri = new JLabel("Data Pelanggan");
        datadiri.setFont(new Font("Timer New
Roman",Font.BOLD,30));
        datadiri.setBounds(90, 30, 200, 30);
        Pelanggan.add(datadiri);
        area.setBounds(30, 90, 400, 300);
        Pelanggan.add(area);

        daftarcamera = new JLabel("Daftar Camera");
        daftarcamera.setBounds(450, 30, 250, 30);
        daftarcamera.setFont(new Font("Times New
Roman",Font.BOLD,30));
        Pelanggan.add(daftarcamera);
        pilcamera.setBounds(450, 90, 230, 30);
```

```
        Pelanggan.add(pilcamera);

        daftarbtn = new JButton("Daftar");
        daftarbtn.setBounds(500, 200, 100, 30);
        daftarbtn.setBackground(Color.GRAY);
        Pelanggan.add(daftarbtn);

        back = new JButton("Back");
        back.setBounds(30, 500, 100, 30);
        back.setBackground(Color.GRAY);
        Pelanggan.add(back);

        Pelanggan.setVisible(true);
        Pelanggan.setLocationRelativeTo(null);

Pelanggan.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE
);
        if(cek==-1){ //cek data, apabila kita belum
daftar praktkum
            JOptionPane.showMessageDialog(null, "Anda
Belum Daftar Camera", "Information",
JOptionPane.INFORMATION_MESSAGE);
        }else if(cek==-2){
            JOptionPane.showMessageDialog(null,"Anda
Belum Daftar Camera", "Information",
JOptionPane.INFORMATION_MESSAGE);
        }else{
            area.setText(datapelanggan());
        }
        daftarbtn.addActionListener(new
ActionListener(){
        @Override
        public void actionPerformed(ActionEvent ae){
            int indexcamera =
pilcamera.getSelectedIndex();

Alex07133_Allobjctrl.Pelanggan.Alex07133_DaftarCamera

(indexcamera,Alex07133_Allobjctrl.Pelanggan.getData(),
false);
            area.setText(datapelanggan());
        }
        });

        back.addActionListener(new ActionListener(){
            @Override
```

Laporan Sementara

Screenshot



Analisa

Padal class Alex07133_PelangganGUI disni kita membuat tampilan gui pada menu ketika kita masuk menu masuk sebagai praktikan

Langkah Ketiga

Membuat class Alex07133_GUI untuk memberikan tampilan GUI di menu awal

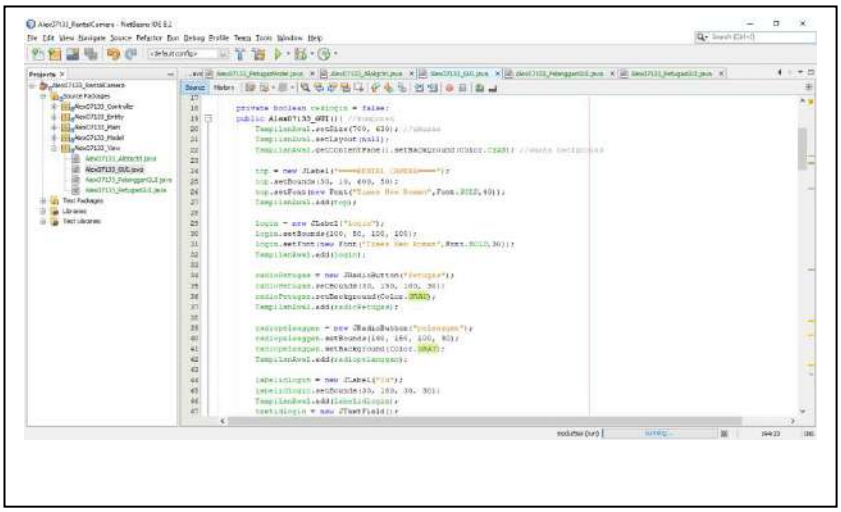
Source Code

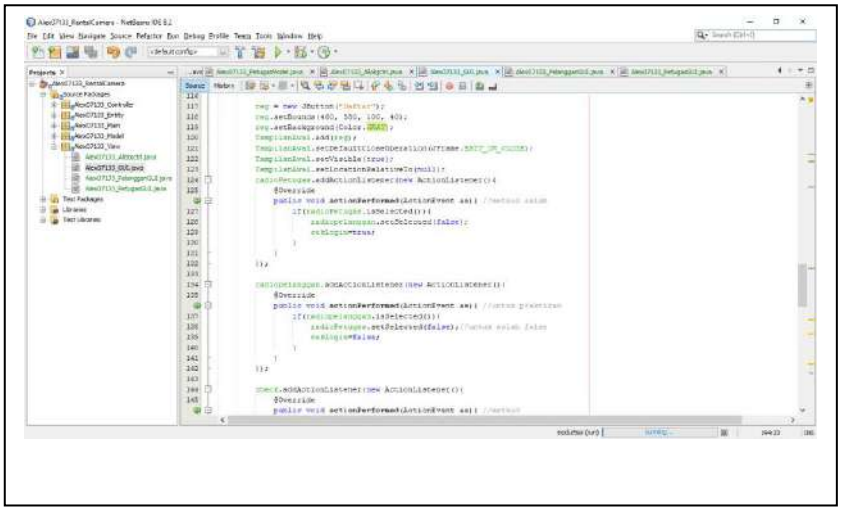
```
 package Alex07133_View;
import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Date;
import javax.swing.*;
public class Alex07133_GUI {
    JFrame TampilanAwal = new JFrame();//tampilan awal
    JLabel login,daftar,top;
    JRadioButton radioPetugas,radiopelanggan; //tombol
pilihan
    JTextField
textidlogin,textnamalogin,textalamat,textnoidentitas,te
xtnotelp,texttglsewa,

texttglkembali,textiddaftar,textnamadaftar;
    JLabel
labelidlogin,labelnamalogin,labelalamat,labelnoidentita
s,labelnotelp,labeltglsewa,

labeltglkembali,labeliddaftar,labelnamadaftar;
    JButton check,reg; //tombol

    private boolean ceklogin = false;
    public Alex07133_GUI(){ //komponen
        TampilanAwal.setSize(700, 630); //ukuran
        TampilanAwal.setLayout(null);

TampilanAwal.getContentPane().setBackground(Color.CYAN)
; //warna background

        top = new JLabel("====RENTAL CAMERA====");
        top.setBounds(50, 10, 600, 50);
        top.setFont(new Font("Times New
Roman",Font.BOLD,40));
        TampilanAwal.add(top);

        login = new JLabel("login");
        login.setBounds(100, 50, 100, 100);
        login.setFont(new Font("Times New
```

```
 TampilanAwal.add(top);

        login = new JLabel("login");
        login.setBounds(100, 50, 100, 100);
        login.setFont(new Font("Times New
Roman",Font.BOLD,30));
        TampilanAwal.add(login);

        radioPetugas = new JRadioButton("Petugas");
        radioPetugas.setBounds(30, 150, 100, 30);
        radioPetugas.setBackground(Color.GRAY);
        TampilanAwal.add(radioPetugas);

        radiopelanggan = new JRadioButton("pelanggan");
        radiopelanggan.setBounds(140, 150, 100, 30);
        radiopelanggan.setBackground(Color.GRAY);
        TampilanAwal.add(radiopelanggan);

        labelidlogin = new JLabel("Id");
        labelidlogin.setBounds(30, 180, 30, 30);
        TampilanAwal.add(labelidlogin);
        textidlogin = new JTextField();
        textidlogin.setBounds(30, 210, 200, 30);
        TampilanAwal.add(textidlogin);

        labelnamalogin = new JLabel("Nama");
        labelnamalogin.setBounds(30, 240, 100, 30);
        TampilanAwal.add(labelnamalogin);
        textnamalogin = new JTextField();
        textnamalogin.setBounds(30, 270, 200, 30);
        TampilanAwal.add(textnamalogin);

        check = new JButton("check");
        check.setBounds(90, 320, 100, 40);
        check.setBackground(Color.GRAY);
        TampilanAwal.add(check);

        daftar = new JLabel("Daftar");
        daftar.setFont(new Font("Times New
Roman",Font.BOLD,30));
        daftar.setBounds(440, 50, 150, 100);
        TampilanAwal.add(daftar);

        labeliddaftar = new JLabel("Id");
        labeliddaftar.setBounds(400, 150, 30, 30);
        TampilanAwal.add(labeliddaftar);
```

Laporan Sementara

Screenshot





Analisa

Padal class Alex07133_GUI disni kita membuat tampilan gui pada menu awal ketika kita menjalankan program

Laporan Sementara

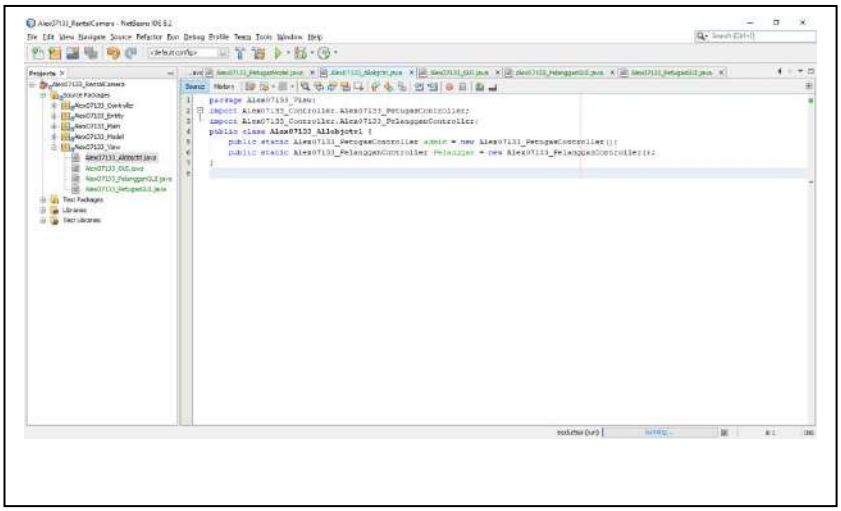
Langkah Keempat

Membuat class Alex07133_Allobjctrl untuk menggabungkan class GUI agar terhubung ke dalam Controller

Source Code

```
package Alex07133_View;
import
Alex07133_Controller.Alex07133_PetugasController;
import
Alex07133_Controller.Alex07133_PelangganController;
public class Alex07133_Allobjctrl {
    public static Alex07133_PetugasController admin =
new Alex07133_PetugasController();
    public static Alex07133_PelangganController
Pelanggan = new Alex07133_PelangganController();
}
```

Screenshot



Analisa

Padal class Alex07133_Allobjctrl disni kita membuat tampilan gui supaya dapat terhubung ke kontroller

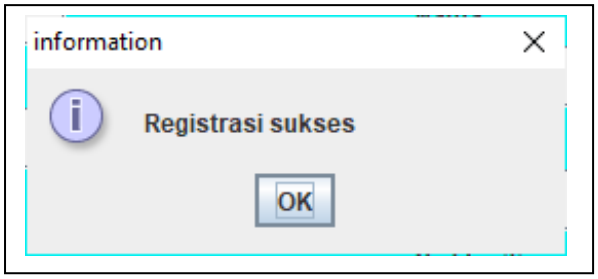
Langkah Kelima

Menambahkan JOptionPane pada GUI

Source Code

```
@Override
            public void actionPerformed(ActionEvent
ae){ //method
                if(ceklogin==true){
                    try{

Alex07133_Allobjctrl.admin.datapetugas(); // login
untuk petugas

Alex07133_Allobjctrl.admin.login(textidlogin.getText(),
textnamalogin.getText());
                        String nama =
Alex07133_Allobjctrl.admin.Alex07133_PetugasEntity().ge
tnama(); //memasukkan id,nama, jika benar

JOptionPane.showMessageDialog(null, "Selamat datang
"+nama,"information", JOptionPane.INFORMATION_MESSAGE);
                        Alex07133_PetugasGUI petugas =
new Alex07133_PetugasGUI();
                        TampilanAwal.dispose();
                    }catch(Exception exception){ //jika
id/nama salah makan akan muncul pesan

JOptionPane.showMessageDialog(null, "id atau nama
salah", "information",
JOptionPane.INFORMATION_MESSAGE);
                    }
                        kosong();//method kosong, jika
datanya kosong
                }else{
                    try{ //login untuk pelanggan

Alex07133_Allobjctrl.Pelanggan.login(textidlogin.getTex
t(),textnamalogin.getText());
                       String nama =
Alex07133_Allobjctrl.Pelanggan.getData().getnama();
//memasukkan id,nama, jika benar

JOptionPane.showMessageDialog(null, "Selamat datang
"+nama,"information", JOptionPane.INFORMATION_MESSAGE);
                       Alex07133_PelangganGUI  pel =
```

Screenshot



Analisa

Joptionpane disni saya gunakan ketika berhasil mendaftarkan pelanggan dan juga masuk pelanggan pada saat berhasil registrasi

Langkah Keenam

Menerapkan CRUD pada program GUI

Source Code

```
        verif.addActionListener(new ActionListener(){
            @Override
            public void actionPerformed(ActionEvent
ae){ //untuk verifikasi id, dan nama
                String id = idtext.getText();
                String nama = namatext.getText();
                int index =
Alex07133_AllObjectModel.daftarPelangganmodel.cekData(i
d, nama); //dicek

Alex07133_Allobjctrl.admin.updateIsVerified(index,
Alex07133_Allobjctrl.Pelanggan.showDaftarPelanggan(inde
x)

.getIndexCamera(),Alex07133_Allobjctrl.Pelanggan.showDa
ftarPelanggan(index).getPelanggan());

tabelpelanggan.setModel(pelanggan.daftarpel());
            }
        });

        hapus.addActionListener(new ActionListener(){
            @Override
            public void actionPerformed(ActionEvent
ae){ //untuk verifikasi id, dan nama
                String id = idtext.getText();
                int index =
Alex07133_AllObjectModel.daftarPelangganmodel.alldatape
langgan().size(); //dicek

Alex07133_Allobjctrl.Pelanggan.deleteDataPelanggan(inde
x);

tabelpelanggan.setModel(pelanggan.daftarpel());
            }
        });

        tabelpelanggan.addMouseListener(new
MouseAdapter(){
            @Override
```

Laporan Sementara

Screenshot

Analisa

Proses CRUD dapat di akses melalui GUI dengan tampilan dan support untuk mouse klik (ActionListener)