



## **LAPORAN PROJECT AKHIR**

### **SISTEM ADMINISTRASI KOST TAMARA**

MATAKULIAH:  
PRAKTIKUM PEMROGRAMAN VISUAL

Disusun Oleh:

#### **Kelompok C1 / 6 / 2015**

Gino Favian Leonardo 1515015101

Muhammad Hardianto 1515015103

Resty Primadana 1515015116

Muhammad Sarman Noorlah 1515015127

Asisten:

Nuzul Firjan  
1415015074

Rizky Ariesta  
1415015048

**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI DAN KOMUNIKASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS MULAWARMAN  
SAMARINDA  
2017**

## **KATA PENGANTAR**

Puji dan Syukur kita panjatkan kepada Allah SWT, Karena atas hidayah-Nyalah penulis dapat menyelesaikan tugas akhir mata kuliah Pratikum Pemrograman Visual dengan baik.

Banyak kesulitan yang kami hadapi ketika pembuatan konsep serta dalam pembuatan laporan akhir. Salah satu kesulitan yang kami hadapi mulai dari menentukan judul, konsep, dan penerapan fungsi-fungsi yang baik dan benar.

Kami menyadari bahwa penyusunan dan penyelesaian tugas laporan akhir ini tidak terselesaikan tanpa bantuan dari berbagai pihak. Oleh Karena itu, dengan kerendahan dan ketulusan hati kami mengucapkan terima kasih kepada :

1. Pak Edy Budiman MT selaku dosen mata kuliah Pemrograman Visual
2. Nuzul Firjan selaku lab pembimbing kelas praktikum C1
3. Rizky Ariesta selaku lab pembimbing kelas praktikum C1

Kami menyadari atas keterbatasan kemampuan yang kami miliki, sehingga dalam penyusunan laporan akhir ini masih banyak terdapat kesalahan didalamnya. Karena dari itu kami, saran dan kritikan yang sifatnya membangun sangat penulis harapkan dari pembaca. Dan akhir kita semoga penulisan laporan akhir ini dapat memberikan manfaat bagi kita semua.

Penulis

Kelompok 6

## **TAKARIR**

Daftar padanan kata bahasa asing dalam bahasa Indonesia yang digunakan adalah sebagai berikut :

<i>Database</i>	Basis data
<i>Managemen</i>	Mengatur
<i>Memonitor</i>	Memantau
<i>Textbox</i>	Kotak teks
<i>Event</i>	Aksi
<i>Layout</i>	Tata letak
<i>Report</i>	Laporan
<i>Library</i>	Pustaka
<i>North</i>	Utara
<i>West</i>	Barat
<i>East</i>	Timur
<i>South</i>	Selatan
<i>Center</i>	Tengah
<i>Project</i>	Proyek
<i>Login System</i>	Masuk sistem

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>ii</b>
<b>TAKARIR.....</b>	<b>iii</b>
<b>DAFTAR ISI.....</b>	<b>iv</b>
<b>DAFTAR GAMBAR.....</b>	<b>vi</b>
<b>DAFTAR TABEL .....</b>	<b>vii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Identifikasi Masalah .....	1
1.2 Batasan Masalah .....	1
1.3 Tujuan.....	2
1.4 Manfaat Aplikasi .....	2
<b>BAB II PERANCANGAN .....</b>	<b>3</b>
2.1 Landasan Teori .....	3
2.2 Konsep yang dipakai .....	3
2.2.1 Konsep Dasar Visual Java.....	3
2.2.2 <i>Layout Manager</i> dan <i>Desain Interface</i> .....	4
2.2.3 <i>Event Handling</i> Visual Java.....	4
2.2.4 Penggunaan Java Tabel .....	5
2.2.5 Koneksi <i>Database</i> JDBC Driver MySQL.....	5
2.2.6 Pembuatan Laporan.....	6
2.2.7 <i>Installer</i> dan <i>Login System</i> .....	6
2.2.8 <i>Splash Screen</i> .....	7
2.3 Analisa Aplikasi .....	7
2.3.1 Definisi Aplikasi .....	7

2.3.2 Pengguna Aplikasi .....	7
2.3.3 Data yang dibutuhkan .....	8
2.4 <i>Entity Relationship Diagram</i> (ERD) .....	9
<b>BAB III HASIL DAN PEMBAHASAN .....</b>	<b>12</b>
3.1 Alur Kerja Aplikasi .....	12
3.2 Implementasi .....	13
<b>BAB IV KESIMPULAN DAN SARAN.....</b>	<b>30</b>
4.1 Kesimpulan.....	30
4.2 Saran .....	30
<b>DAFTAR PUSTAKA .....</b>	<b>31</b>
<b>LAMPIRAN.....</b>	<b>32</b>

## DAFTAR GAMBAR

Gambar 2.1 ERD .....	9
Gambar 2.2 Tabel Fasilitas.....	9
Gambar 2.3 Tabel Kamar .....	10
Gambar 2.4 Tabel KamarFasilitas .....	10
Gambar 2.5 Tabel Penyewa.....	11
Gambar 2.6 Tabel Transaksi .....	11
Gambar 3.1 Alur Aplikasi .....	12
Gambar 3.2.1 <i>Splash Screen</i> .....	13
Gambar 3.2.2 Halaman <i>Login</i> .....	13
Gambar 3.2.3 Gagal <i>Login</i> .....	14
Gambar 3.2.4 Halaman Utama .....	14
Gambar 3.2.5 Halaman Data Penghuni .....	15
Gambar 3.2.6 Halaman Data Kamar .....	15
Gambar 3.2.7 Halaman Data Fasilitas.....	16
Gambar 3.2.8 Halaman Data Transaksi.....	16
Gambar 3.2.9 Kwitansi.....	17
Gambar 3.2.10 Halaman Laporan .....	17
Gambar 3.2.11 Halaman Tentang.....	18

## **DAFTAR TABEL**

Tabel 2.3 Tabel Pengguna Aplikasi .....	7
Tabel 2.4 Tabel Data yang Dibutuhkan.....	8

# **BAB I**

## **PENDAHULUAN**

### **1.1 Identifikasi Masalah**

Definisi kos hari ini sebenarnya sudah tidak melulu tepat. Banyak kos dibangun terpisah dari rumah induk, menjadi satu ragam bangun tersendiri. Kamar-kamar kos dibangun membentuk blok berbanjar. Rumah kos modern bahkan dibangun tidak hanya satu kamar dengan kamar mandi dan dapur komunal, tetapi telah dilengkapi kamar mandi (di) dalam (rumah), dapur, hingga ruang tamu dan ruang keluarga.

Banyaknya mahasiswa/mahasiswi yang menyewa kos karena berasal dari luar daerah tempat kuliah dia berada. Tentu ini perlu dicatat biodatanya. Para pemilik kos biasanya mencatat asal usul para penyewa kos dan disuruh mengisi biodata untuk proses registrasi.

Pencatatan yang manual tentu akan sangat merepotkan dan mempersulit para pemilik kos. Karenanya para pemilik memerlukan sebuah aplikasi yang mampu mempermudah karyawan untuk melakukan pencatatan setiap penyewa kos yang masuk.

### **1.2 Batasan Masalah**

Untuk mempermudah dalam proses administrasi kost, *system* diperlukan suatu pembatasan dari masalah-masalah yang ada. Sehingga dapat diketahui ruang lingkup dari system yang dirancang. Dalam pembuatan program ini, pembuat membatasi masalah sebagai berikut :

1. Harus mengisi data yang lengkap sesuai dengan apa yang disediakan oleh aplikasi ini.
2. Hanya dapat diakses oleh superadmin dan admin.
3. Harus menggunakan localhost.



### **1.3 Tujuan**

Tujuan diciptakan aplikasi ini yaitu :

1. Dapat menghasilkan Sistem Administrasi yang dapat membantu proses administrasi Kost Tamara.

### **1.4 Manfaat Aplikasi**

Adapun manfaat yang diambil dari aplikasi ini yaitu :

1. Mempermudah dan mempercepat proses administrasi pada Kost Tamara.
2. Mempercepat dalam proses mengolah data.

## **BAB II**

### **PERANCANGAN**

#### **2.1 Landasan Teori**

##### **a) Pengertian Kos**

Kos secara sederhana didefinisikan sebagai menempati satu ruang (kamar) rumah seseorang, dengan perjanjian membayar dalam jumlah tertentu sebagai kompensasi sewa dan fasilitas lain di dalamnya, seperti makan dan perabot yang dipakai.

##### **b) Administrasi**

Administasi adalah usaha dan kegiatan yang berkenaan dengan penyelenggaraan kebijaksanaan untuk mencapai tujuan. Pengertian administrasi juga dibedakan menjadi dua yaitu pengertian administrasi dalam arti sempit dan pengertian administrasi dalam arti luas. Administrasi dalam arti sempit adalah kegiatan yang meliputi dari catat-mencatat, surat-menyurat, pembukuan ringan, ketik-mengetik, agenda, dan sebagainya yang memiliki sifat teknis ketatausahaan. Sedangkan pengertian administrasi bersifat luas adalah seluruh proses kerja sama dari dua orang atau lebih dalam mencapai tujuan dengan pemanfaatan sarana dan prasarana tertentu secara berdaya guna dan berhasil guna

#### **2.2 Konsep yang Dipakai**

##### **2.2.1 Komponen Dasar Visual Java**

Objek-objek visual java dikenal dengan sebutan Java GUI (*Graphical User Interface*) yang disimpan dalam kelas `javax.swing`.

Objek -objek visual tersebut seperti :

- JLabel
- JButton
- JTextField

- JCheckBox
- JComboBox

### 2.2.2 Layout Manager dan Desain Interface

Pengaturan *layout* digunakan untuk mengatur posisi dari komponen visual penyusun program sesuai dengan desain *user interface*. Beberapa pilihan *layout* telah disediakan java, dimana keputusan untuk menggunakan jenis *layout* tertentu bergantung pada jenis aplikasi yang ingin dibuat serta tingkat kerapian yang diinginkan.

Jenis *layout manager* yang dapat digunakan antara lain :

- a. BorderLayout merupakan jenis *layout* yang bekerja dengan membagi *frame* menjadi lima bagian yaitu *NORTH*, *EAST*, *SOUTH*, *WEST*, dan *CENTER*.
- b. CardLayout adalah *layout manager* yang mampu menciptakan efek “tumpukan” pada komponen. Bisa menampung banyak komponen, namun hanya satu buah komponen yang akan ditampilkan dalam satu waktu.
- c. GridLayout adalah jenis *layout* yang bekerja berdasarkan baris dan kolom. Dengan *layout* ini kita dapat memberikan argument banyaknya baris dan kolom sesuai dengan kebutuhan.

### 2.2.3 Event Handling Visual Java

Event Handling adalah suatu metode untuk menangani sebuah *event*/aksi yang diberikan pengguna kepada suatu komponen GUI. *Event* adalah suatu peristiwa yang dipicu oleh pengguna pada suatu komponen, misalnya tombol ditekan. Dua paket yang biasa digunakan menangani kejadian adalah *java.awt.event* dan *java.swing.event*. *Event handling* dalam

menangani *event* terbagi menjadi 2 bagian yaitu *Event Source* dan *Event Listener* atau *Event Handler*.

#### 2.2.4 Penggunaan Java Tabel

JTable merupakan komponen *grid* yang terdiri dari baris dan kolom. JTable merupakan komponen yang digunakan untuk menampilkan data seperti pada *spreadsheet*.

JTable memiliki *mouse listener* untuk mendeteksi aksi mouse dan *KeyListener* untuk mendeteksi aksi *keyboard*. Dengan *class* JTable kita dapat menampilkan data, juga memperbolehkan user untuk mengedit data. Setiap cell menampilkan sebuah item data. Setiap *header* kolom merupakan kolom dari tabel. Setiap kolomnya memiliki tipe data yang sama.

#### 2.2.5 Koneksi Database JDBC Driver MySQL

JDBC dibutuhkan untuk menghubungkan bahasa pemrograman java dengan database. JDBC merupakan driver untuk mengakses database. Driver JDBC sendiri merupakan koleksi *class-class* Java yang dikumpulkan dalam satu atau beberapa file .jar. JDBC yang digunakan berbeda-beda untuk setiap database yang berbeda.

Ada beberapa langkah yang harus dilakukan untuk mengakses database menggunakan bahasa java:

- 1) Import packages terkait. Package yang biasanya di import yaitu `import java.sql.*`
- 2) Mendaftarkan driver JDBC tersebut.
- 3) Membuka koneksi. Caranya dengan menggunakan method `DriverManager.getConnection()`.
- 4) Eksekusi Query.
- 5) Ekstrak data hasil eksekusi Query.

### 2.2.6 Pembuatan Laporan

Pembuatan laporan di java dapat menggunakan iReport. IReport tersedia dalam bentuk aplikasi maupun iReport yang dapat di-integrasikan dengan Netbeans IDE. Artinya tidak perlu menggunakan 2 *tools* yang berbeda untuk membuat aplikasi yang lengkap dengan reportnya. Dengan Netbeans IDE + iReport plugin maka *tools* pengembangan aplikasi kita sudah lengkap.

### 2.2.7 Installer dan Login System

Untuk mendistribusikan aplikasi Java, proyek harus diubah menjadi jar (Java Archive). Jar adalah format kompresi untuk aplikasi Java. Dengan format jar, aplikasi java dapat dijalankan pada sistem operasi, dengan syarat sistem operasi tersebut telah terpasang JRE ( Java RunTime Environment).

Sistem login menyangkut 2 hal/konsep/proses yaitu Otentikasi & Otorisasi atau *Authentication & Authorization*. Kedua konsep ini perlu dipahami sebelum memulai membuat kode *login* sistem yang akan dijelaskan dalam modul ini.

Otentikasi atau *Authentication* adalah proses mengkonfirmasi identitas dari seseorang atau suatu benda. Otorisasi atau *Authorization* adalah proses memberi izin seseorang atau suatu benda untuk melakukan sesuatu.

Proses otentikasi & otorisasi menyangkut 4 komponen penting.

1. Orang atau benda yang akan melakukan sesuatu dan dikonfirmasi identitasnya.
2. Aksi yang dilakukan, misal membaca, menonton;
3. Benda yang dikenakan Aksi, misal tv, buku, kamar tidur;

#### 4. Peran atau *role* dari orang di dalam sistem.

##### 2.2.8 Splash Screen

*Splash Screen* adalah tampilan proses loading aplikasi yang muncul sebelum masuk ke menu utama. *Splash Screen* biasanya ditampilkan untuk menunggu proses load data yang cukup besar atau program yang berat. Lama *Splash Screen* biasanya tergantung waktu meload data pada program yang akan di jalankan.

### 2.3 Analisa Aplikasi

#### 2.3.1 Definisi Aplikasi

Aplikasi ini berguna untuk mempermudah pemilik Kost Tamara ini agar bisa menginputkan data penyewa, kamar, maupun fasilitas yang serta jumlah tagihan secara otomatis.

#### 2.3.2 Pengguna Aplikasi

Table 2.3 Tabel Pengguna Aplikasi

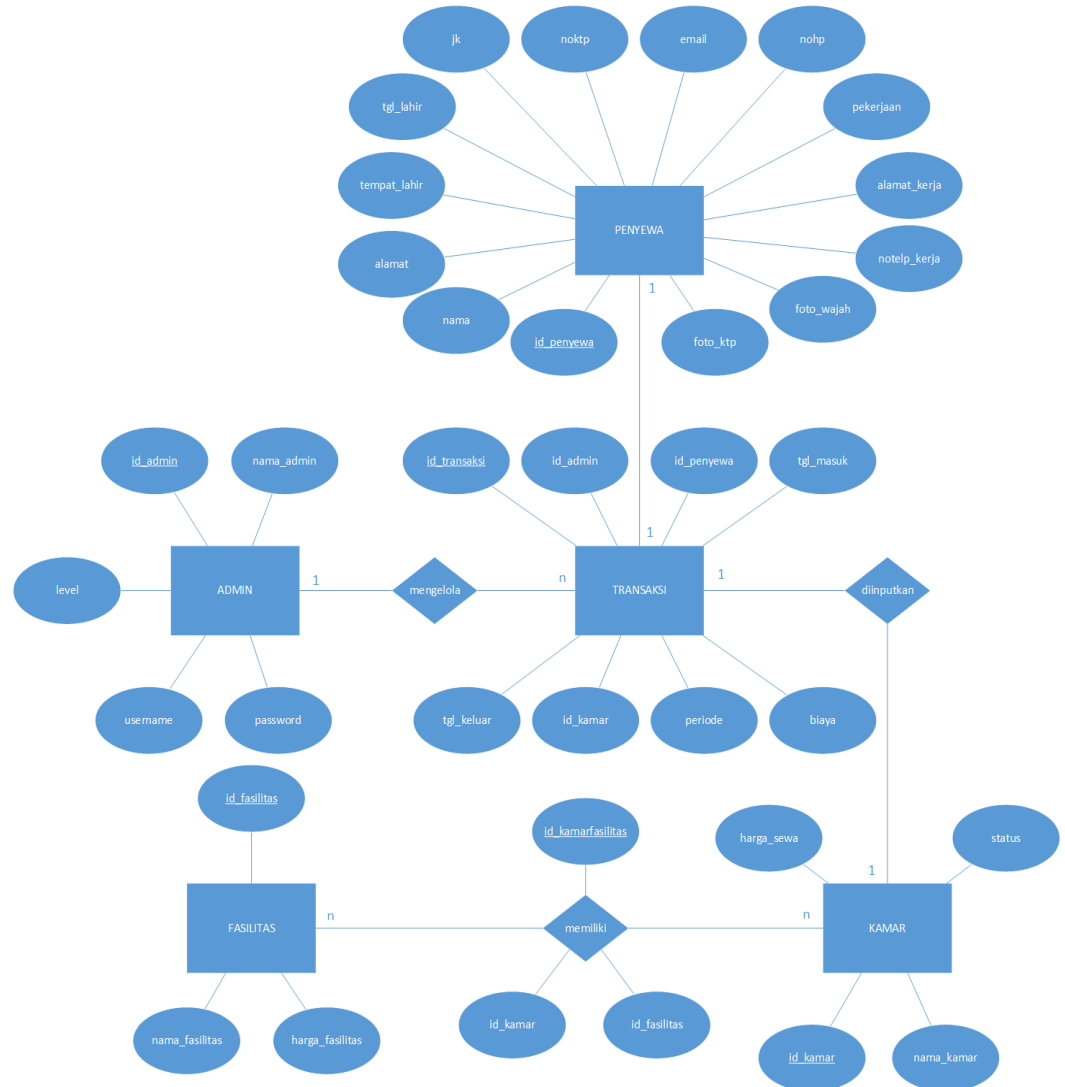
NO	PENGGUNA	HAK AKSES
1	Administrator	<ul style="list-style-type: none"><li>- Melakukan CRUD pada semua data</li><li>- Melihat data</li><li>- Mencetak kwitansi</li></ul>

### 2.3.3 Data yang dibutuhkan

Table 2.4 Tabel Data yang Dibutuhkan

NO	NAMA DATA	ATRIBUT
1	Fasilitas	Id_fasilitas, nama_fasilitas, harga_fasilitas
2	Kamar	Id_kamar, nama_kamar, harga_sewa, status
3	KamarFasilitas	Id_kamarfasilitas, id_kamar, id_fasilitas
4	Penyewa	Id_penyewa, nama, alamat, tempat_lahir, tgl_lahir, jk, noktp, email, nohp, pekerjaan, alamat_kerja, notelp_kerja, foto_wajah, foto_ktp
5	Transaksi	Id_transaksi, id_admin, id_penyewa, tgl_masuk, tgl_keluar, id_kamar, periode, biaya

## 2.4 Entity Relationship Diagram (ERD)



Gambar 2.1 ERD Aplikasi Sistem Administrasi Kost Tamara

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1	<b>id_fasilitas</b> 🔑	int(10)		No	None	AUTO_INCREMENT
<input type="checkbox"/>	2	<b>nama_fasilitas</b>	varchar(30)		No	None	
<input type="checkbox"/>	3	<b>harga_fasilitas</b>	int(20)		No	None	

Gambar 2.2 Tabel Fasilitas



#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/> 1	<b>id_kamar</b> 🔑	int(10)			No	None	AUTO_INCREMENT
<input type="checkbox"/> 2	<b>nama_kamar</b>	varchar(30)			No	None	
<input type="checkbox"/> 3	<b>harga_sewa</b>	int(20)			No	None	
<input type="checkbox"/> 4	<b>status</b>	enum('Kosong', 'Terisi')			No	None	


Gambar 2.3 Tabel Kamar

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/> 1	<b>id_kamarfasilitas</b> 🔑	int(10)			No	None	AUTO_INCREMENT
<input type="checkbox"/> 2	<b>id_kamar</b>	int(10)			No	None	
<input type="checkbox"/> 3	<b>id_fasilitas</b>	int(10)			No	None	

Gambar 2.4 Tabel KamarFasilitas

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/> 1	<b>id_penyewa</b> 🔑	int(10)			No	None	AUTO_INCREMENT
<input type="checkbox"/> 2	<b>nama</b>	varchar(30)			No	None	
<input type="checkbox"/> 3	<b>alamat</b>	varchar(30)			No	None	
<input type="checkbox"/> 4	<b>tempat_lahir</b>	varchar(15)			No	None	
<input type="checkbox"/> 5	<b>tgl_lahir</b>	date			No	None	
<input type="checkbox"/> 6	<b>jk</b>	varchar(15)			No	None	
<input type="checkbox"/> 7	<b>noktp</b>	varchar(20)			No	None	
<input type="checkbox"/> 8	<b>email</b>	varchar(40)			No	None	
<input type="checkbox"/> 9	<b>nohp</b>	varchar(30)			No	None	
<input type="checkbox"/> 10	<b>pekerjaan</b>	varchar(30)			No	None	
<input type="checkbox"/> 11	<b>alamat_kerja</b>	varchar(30)			No	None	
<input type="checkbox"/> 12	<b>notelp_kerja</b>	varchar(20)			No	None	
<input type="checkbox"/> 13	<b>foto_wajah</b>	varchar(250)			No	None	
<input type="checkbox"/> 14	<b>foto_ktp</b>	varchar(250)			No	None	

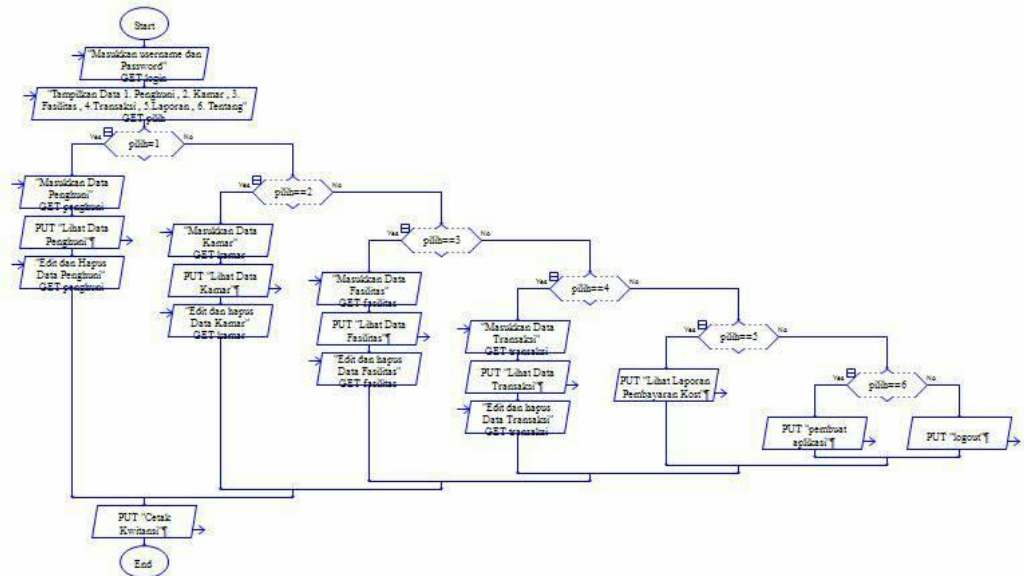
Gambar 2.5 Tabel Penyewa

	#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1	<b>id_transaksi</b> 	int(10)			No	None	AUTO_INCREMENT
<input type="checkbox"/>	2	<b>id_admin</b>	int(10)			No	None	
<input type="checkbox"/>	3	<b>id_penyewa</b>	int(10)			No	None	
<input type="checkbox"/>	4	<b>tgl_masuk</b>	date			No	None	
<input type="checkbox"/>	5	<b>tgl_keluar</b>	date			No	None	
<input type="checkbox"/>	6	<b>id_kamar</b>	int(10)			No	None	
<input type="checkbox"/>	7	<b>periode</b>	int(5)			No	None	
<input type="checkbox"/>	8	<b>biaya</b>	int(30)			No	None	

Gambar 2.6 Tabel Transaksi

### BAB III HASIL DAN PEMBAHASAN

#### 3.1 Alur Kerja Aplikasi



Gambar 3.1 Alur Aplikasi

Pada gambar diatas pertama-tama kita jalankan aplikasinya. Kemudian masukkan *username* dan *password*, apabila tidak cocok *username* dan *password* diulangi lagi sampe bisa berhasil masuk. Kemudian ada 6 pilihan yaitu ; Pilihan 1 merupakan data penghuni, disini mengisi apa yang ada di data penghuni setelah selesai maka akan kembali kemenu pilihan. Begitu juga pilihan 2, pilihan 3, pilihan 4. Dan ada juga ada menu cetak kwitansi pada pilihan 4. Apabila memilih pilihan 5 maka akan muncul pesan program masih tahap pengembangan kemudian klik ok maka akan kembali ke menu pilihan. Setelah itu pilihan 6 maka akan melihat halaman tentang yang berisi tentang aplikas dan tentang pengembangan, setelah itu apabila selesai memilih ke-6 pilihan tersebut kita bisa logout dari aplikasi dan login kembali. Apabila tidak kita bisa keluar dari aplikasi

### 3.2 Implementasi

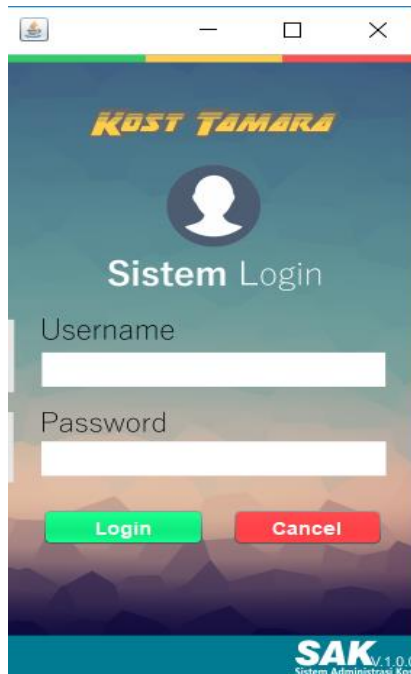
#### 1. Halaman *Splash Screen*



Gambar 3.2.1 Splash Screen

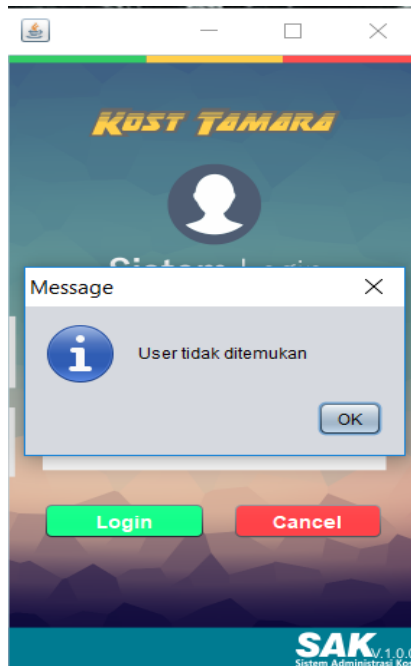
Pada gambar ini merupakan loading awal dalam menjalankan aplikasi ini.

#### 2. Halaman Login



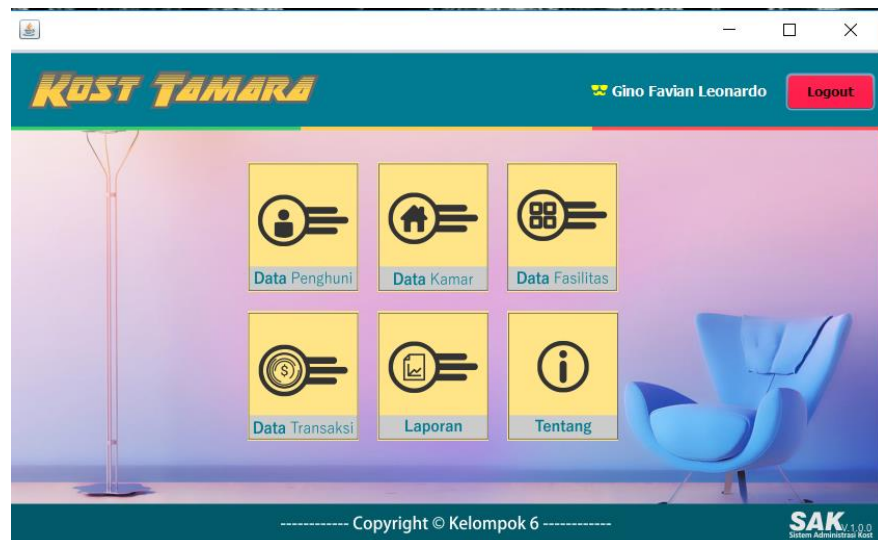
Gambar 3.2.2 Halaman Login

Pada halaman ini merupakan system login di dalam menjalankan aplikasi ini. Dan gambar ini dibawah ini merupakan gambar gagal login karena tidak ditemukan *user* tersebut.



Gambar 3.2.3 Gagal Login

### 3. Halaman Utama



Gambar 3.2.4 Halaman Utama

Pada halaman ini merupakan halaman utama dari aplikasi Sistem Administrasi yang didalam terdapat halaman Data Penghuni, Data Kamar, Data Fasilitas, Data Transaksi, Laporan dan Tentang.

#### 4. Halaman Data Penghuni

**Tambah Data**

Nama  
Alamat  
Trmpt. Lahir  
Tgl. Lahir  
Gender  
E-mail

No. KTP  
No. Telp  
Pekerjaan  
Almt. Pekerjaan  
No. Telp Pekerjaan

Pas Foto  
Foto KTP

Cari

Simpan Tambah

**Data Penghuni**

Cari Data Filter Berdasarkan Cari

ID	NAMA	ALAMAT	GENDER	NO. TELP	EMAIL	PAS FOTO	FOTO KTP
18	Arif Fahrizal	Jalan Pramuta	Pria	08994000442	arifcaem@gmail.com	pas1.jpg	KTP3.jpg
19	Fernando Eidapati	Jalan Damai Sentosa	Pria	08978857213	fnodoell@gmail.com	pas4.jpg	KTP4.jpg

Lihat Detail  
Hapus  
Edit

Copyright © Kelompok 6

SAK 1.0.0

Gambar 3.2.5 Halaman Data Penghuni

Pada halaman ini merupakan halaman untuk mengisi biodata tentang penghuni seperti yang ada digambar kemudian disimpan kedalam database. Apabila ingin lihat detail, hapus dan edit bisa mengklik tombol yang ada disebelah kanan table. Bahkan kita bisa mencari data sesuai filter yang ada.

#### 5. Halaman Data Kamar

**Tambah Data**

No. Kamar  
Nama Kamar  
Harga Kamar  
Status

**Data Kamar**

Cari Kamar Filter Berdasarkan Cari

ID	NAMA KAMAR	HARGA SEWA	STATUS KAMAR
1	Kamar 001	450000	Kosong
3	Kamar 002	500000	Kosong

EDIT  
HAPUS

SIMPAN TAMBAH Bersihkan

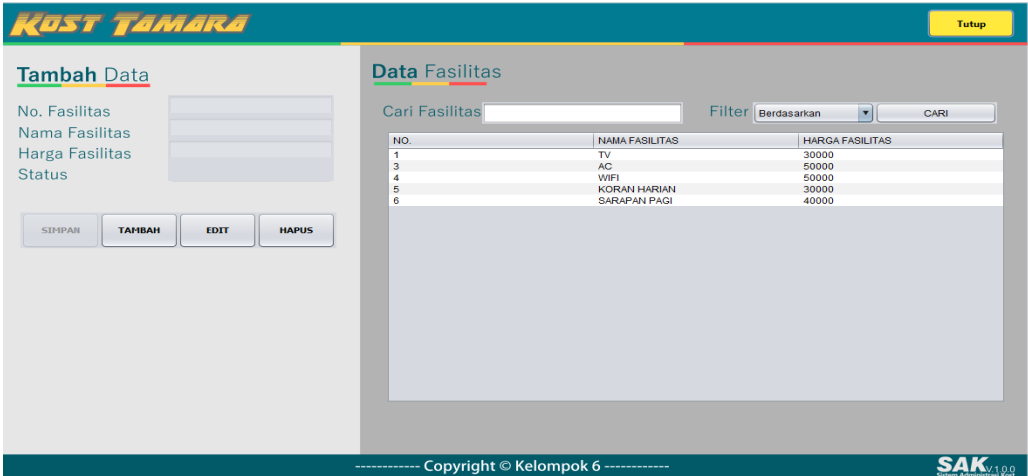
Copyright © Kelompok 6

SAK 1.0.0

Gambar 3.2.6 Halaman Data Kamar

Pada halaman ini merupakan halaman untuk mengisi data kamar seperti yang ada digambar kemudian disimpan kedalam database. Sama seperti halaman Data Penghuni apabila ingin, hapus dan edit bisa mengklik tombol yang ada disebelah kanan table. Bahkan kita bisa mencari data sesuai filter yang ada.

## 6. Halaman Data Fasilitas



**Kost Tamara** Tutup

Tambah Data

No. Fasilitas

Nama Fasilitas

Harga Fasilitas

Status

Data Fasilitas

Cari Fasilitas  Filter Berdasarkan

NO.	NAMA FASILITAS	HARGA FASILITAS
1	TV	30000
3	AC	50000
4	WIFI	50000
5	KORAN HARIAN	30000
6	SARAPAN PAGI	40000

----- Copyright © Kelompok 6 ----- SAK V1.0.0

Gambar 3.2.7 Halaman Data Fasilitas

Pada halaman ini merupakan halaman untuk mengisi data fasilitas seperti yang ada digambar kemudian disimpan kedalam *database*. Sama seperti halaman Data Penghuni apabila ingin, hapus dan edit bisa mengklik tombol yang ada. Bahkan kita bisa mencari data sesuai filter yang ada.

## 7. Halaman Data Transaksi

**KOST TAMARA**

**Tambah Pembayaran**

No. Pembayaran  
 Kode Penghuni  
 Nama Penghuni  
 No. KTP  
 Alamat Asal  
 Pilih Kamar  
 Nama Kamar  
 Harga Kamar  
 Periode  
 Total  
 Tgl. Masuk  
 Tgl. Keluar

**Data Pembayaran**

Cari Data  
 Filter  
 Berdasarkan  
 CARI  
 KWITANSI PEMBAYARAN

NO. PEMBAYARAN	NAMA PENYEWA	NAMA KAMAR	TOTAL PEMBAYARAN
1	Arif Fahrizal	Kamar 001	450000
2	Arif Fahrizal	Kamar 001	450000
3	Arif Fahrizal	Kamar 001	900000
5	Arif Fahrizal	Kamar 002	2000000
6	Fernando Eldapati	Kamar 002	1000000

Copyright © Kelompok 6

SAK V.1.0.0

Gambar 3.2.8 Halaman Data Transaksi

Pada halaman ini merupakan halaman untuk mengisi data transaksi seperti yang ada digambar kemudian disimpan kedalam database. Sama seperti halaman Data Penghuni apabila ingin, hapus dan edit bisa mengklik tombol yang ada. Bahkan kita bisa mencari data sesuai filter yang ada. Serta ada tombol tambahan untuk mencetak kwitansi pembayaran. Dan gambar dibawah ini merupakan gambar kwitansi.

**==== KWITANSI =====**

**Pembayaran Kost Tamara**

Nomor Pembayaran : 5  
 Sudah terima dari : Arif Fahrizal  
 Tanggal Masuk : 5/7/17 12:00 AM  
 Besarnya : Rp 2000000

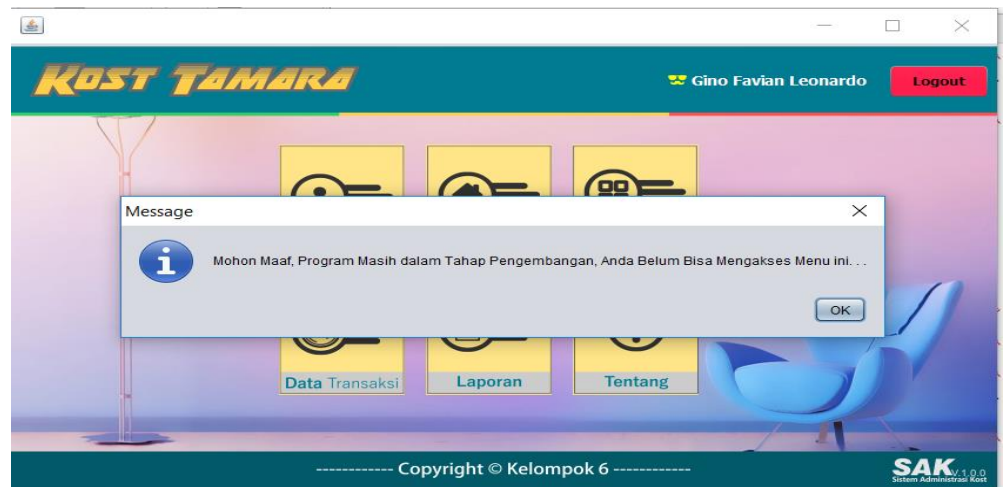
**KOST TAMARA**

Samarinda, Friday 12 May 2017  
 Yang Menyerahkan : Arif Fahrizal  
 Yang Menerima : Muhammad Sarman

Gambar 3.2.9 Kwitansi

## 8. Halaman Laporan





Gambar 3.2.10 Halaman Laporan

Pada halaman ini merupakan halaman melihat isi dari halaman laporan tetapi Program masih dalam tahap pengembangan jadi akan muncul notif seperti digambar diatas.

## 9. Halaman Tentang



Gambar 3.2.11 Halaman Tentang

Pada halaman ini merupakan halaman yang berisi tentang aplikasi dan tentang pengembang.

### 3.3 Source Code

```
package Class;

import java.awt.Toolkit;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import javax.swing.JOptionPane;

public class koneksi {
    private static Connection con;
    public static Connection getConnection(){
        try{
            con = DriverManager.getConnection(
                "jdbc:mysql://localhost:3306/db_sak","root","");
            Toolkit.getDefaultToolkit().beep();
        }
        catch(SQLException e){
            JOptionPane.showMessageDialog(null, "Koneksi Gagal"+e.getMessage());
        }
        return con;
    }
}
```

#### 3.3.1 Code Koneksi

```
package Class;

import java.awt.Toolkit;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import javax.swing.JOptionPane;

public class koneksi {

    private static Connection con;
```

```

public static Connection getConnection(){

    try{

        con = DriverManager.getConnection(

            "jdbc:mysql://localhost:3306/db_sak","root","");

        Toolkit.getDefaultToolkit().beep();

    }

    catch(SQLException e){

        JOptionPane.showMessageDialog(null, "Koneksi Gagal"+e.getMessage());

    }

    return con;

}

}

```

### 3.3.2 Code Login

```

package Forms;

import Class.koneksi;

import java.sql.*;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

```

```

public class frm_Login extends javax.swing.JFrame {

    private Object con;

    private ResultSet rss;

    private Statement stt;

    public static Object[] id_admin = new Object[3];

    /**
     * Creates new form LoginPanel
     */
    public frm_Login() {
        initComponents();
    }

    public static Object[] getform(){

        return id_admin;

    }

    private void ProsesLogin(String username, String password){

        try{

            Statement statement = (Statement)koneksi.getConnection().createStatement();

            ResultSet result=statement.executeQuery

            ("select * from admin where username='"+username+"'");

```

```

if(result.next()){

    if(password.equals(result.getString("password"))){

        id_admin[0] = result.getString("id_admin");

        id_admin[1] = result.getString("nama_admin");

        id_admin[2] = result.getString("level");

        String nama = id_admin[1].toString();

        String level =id_admin[2].toString();

OptionPane.showMessageDialog(rootPane, "Selamat datang "+nama+", Anda Login
Sebagai "+level);

        frm_Home awal = new frm_Home();

        frm_Home.lblUsername.setText(id_admin[1].toString());

        awal.setVisible(true);

        dispose();

    } else {

        JOptionPane.showMessageDialog(rootPane, "Password Salah");

        tfPassword.setText("");

        tfPassword.requestFocus();

    } } else {

OptionPane.showMessageDialog(rootPane, "User tidak ditemukan");

tfUsername.setText("");

tfPassword.setText("");

tfUsername.requestFocus();

    }
}

```

```

    } catch (SQLException ex) {

        Logger.getLogger(frm_Login.class.getName()).log(Level.SEVERE, null, ex);

        JOptionPane.showMessageDialog(rootPane, "Gagal Login!");

    }

}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */

@SuppressWarnings("unchecked")

    Generated Code

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

    // TODO add your handling code here:

}

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

    // TODO add your handling code here:

    System.exit(0);

}

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

    // TODO add your handling code here:

    ProsesLogin(tfUsername.getText(), tfPassword.getText());

}

```

```

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {

    /* Set the Nimbus look and feel */
        Look and feel setting code (optional)

//</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {

            new frm_Login().setVisible(true);

        }

    });

}

// Variables declaration - do not modify
private javax.swing.JButton btnCancel;
private javax.swing.JButton btnLogin;
private usu.widget.Panel panel1;
private javax.swing.JPasswordField tfPassword;
private javax.swing.JTextField tfUsername;

// End of variables declaration
}

```

### 3.3.3 Code Halaman Utama

```
package Forms;

import java.awt.Color;
import javax.swing.JOptionPane;

public class frm_Home extends javax.swing.JFrame {

    String username;

    /**
     * Creates new form HomePanel
     */
    public frm_Home() {
        initComponents();
        lblUsername.setText(username);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code
```



```

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

    // TODO add your handling code here:

    pnlDataPenghuni.setBackground(new Color(103,183,242));

}

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

    // TODO add your handling code here:

    pnlDataPenghuni.setBackground(new Color(255, 229, 136));

    new frm_Penghuni().setVisible(true);

}

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

    // TODO add your handling code here:

    pnlDataKamar.setBackground(new Color(103,183,242));

}

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

    // TODO add your handling code here:

    pnlDataKamar.setBackground(new Color(255, 229, 136));

    new frm_Kamar().setVisible(true);

}

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

    // TODO add your handling code here:

    pnlDataFasilitas.setBackground(new Color(103,183,242));

}

```

```

pnlDataFasilitas.setBackground(new Color(103,183,242));

    }

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

        // TODO add your handling code here:

        pnlDataFasilitas.setBackground(new Color(255, 229, 136));

        new frm_Fasilitas().setVisible(true);

    }

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

        // TODO add your handling code here:

        pnlDataTransaksi.setBackground(new Color(103,183,242));

    }

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

        // TODO add your handling code here:

        pnlDataTransaksi.setBackground(new Color(255, 229, 136));

        new frm_transaksi().setVisible(true);

    }

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

        // TODO add your handling code here:

        pnlLaporan.setBackground(new Color(103,183,242));

    }

```

```

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

    // TODO add your handling code here:

    pnlLaporan.setBackground(new Color(255, 229, 136));

    JOptionPane.showMessageDialog(rootPane, "Mohon Maaf, Program Masih dalam
    Tahap Pengembangan, Anda Belum Bisa Mengakses Menu ini. . .");

}

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

    // TODO add your handling code here:

    pnlTentang.setBackground(new Color(103,183,242));

}

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

    // TODO add your handling code here:

    pnlTentang.setBackground(new Color(255, 229, 136));

    new frm_about().setVisible(true);

}

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

    // TODO add your handling code here:

    frm_Login login = new frm_Login();

```

```

login.setVisible(true);

    dispose();

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {

    /* Set the Nimbus look and feel */

    Look and feel setting code (optional)

//</editor-fold>

    /* Create and display the form */

    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {

            new frm_Home().setVisible(true);

        }

    });

}

```

```
// Variables declaration - do not modify

private javax.swing.JButton btnLogout;

private org.jdesktop.layout.LayoutStyle layoutStyle1;

public static javax.swing.JLabel lblUsername;

private usu.widget.Panel panel2;

private usu.widget.Panel pnlDataFasilitas;

private usu.widget.Panel pnlDataKamar;

private usu.widget.Panel pnlDataPenghuni;

private usu.widget.Panel pnlDataTransaksi;

private usu.widget.Panel pnlLaporan;

private usu.widget.Panel pnlTentang;

// End of variables declaration

}
```

## **BAB IV**

### **KESIMPULAN DAN SARAN**

#### **4.1 Kesimpulan**

Aplikasi ini merupakan program yang bertujuan untuk mempermudah pemilik Kost Tamara ini. Dengan aplikasi ini data diri penyewa, dan jumlah tagihan yang sudah diinputkan bagi pemilik otomatis sudah terjumlah dan tidak perlu lagi repot-repot mencari data dan jumlah tagihan karena sudah tersimpan kedalam database aplikasi tersebut.

#### **4.2 Saran**

Program Sistem Administrasi Kost Tamara ini dapat dikembangkan dan disempurnakan lagi kedepannya agar benar-benar bisa dimanfaatkan dikalangan umum.

## DAFTAR PUSTAKA

- Edy (2014). Hal Depan Bab I edy pdf. Retrieved 10 May, 2017, from <http://eprints.umpo.ac.id/166/1/HAL%20DEPAN,%20BAB%20I%20edy.pdf>
- Hastiyanto, Febrie (2011). Kos, Kontrak, Sewa. Retrieved 10 May, 2017, from <https://rubrikbahasa.wordpress.com/2011/04/06/kos-kontrak-sewa/>
- Romadhon, Ahmad Saru (2010). Modul-pv1 pdf. Retrieved 10 May, 2017, from <https://kubaat.files.wordpress.com/2010/04/modul-pv1.pdf>

## LAMPIRAN

### Pertanyaan dan Jawaban

1. Event apa saja yang berada pada frm Home ?

Jawab:

Pada JFrame Home terdapat event mousepressed dan mousereleased

2. Apakah itu method dan method apa saja yang digunakan pada form penghuni ?

Jawab:

Method adalah perilaku apa saja yang bisa dilakukan suatu object, pada form penghuni terdapat method  
initTable, tambahdatapenghuni, ubahdatapenghuni, pencariandata, hapus data, tampildatapenghuni, kuncifield, bersihkanfield dan cek status.

3. Apakah yang dimaksud data[0] , data[1] .. ?

Jawab:

Untuk memanggil data data yang sesuai dengan database dan kolom pada table.

4. Apakah yang dimaksud Object[6] pada method tambahdata di frm penghuni?

Jawab:

Object[6] itu adalah array, jumlah nya ada 6 karean menyesuaikan kolom yang ada pada table.

5. Bagaimana query pengkondisian pada method pencarian data?

Jawab:

Dengan menambah where pada query lalu disesuaikan pada kondisi apa pencarian akan dilakukan misal ingin mencari nama makan akan disesuaikan pada kolom nama dst.

6. Layout apa saja yang digunakan pada aplikasi ini dan apa kegunaan layout tersebut?

Jawab:

Layout yang digunakan pada aplikasi ini adalah absolute layout yaitu layout yang mengatur posisi tiap komponen di layar berdasarkan posisi x dan y.