

PEMROGRAMAN BERBASIS OBJEK

MANAJEMEN SISTEM KOST



DISUSUN OLEH :

- Rizky Gunawan (15225007)
- Roy Jordy (15225005)
- Jeni Fatmawati (15225008)
- Elsa Mutia Sari (15225009)

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS BINA SARANA INFORMATIKA

2023

KATA PENGANTAR

Puji dan Syukur kita panjatkan kehadiran Allah SWT, yang telah melimpahkan Rahmat dan hidayat-Nya sehingga kami dapat menyelesaikan modul ini dengan judul “Program Sistem Manajemen Kost” menggunakan Java GUI NetBeans. Modul ini disusun sebagai bagian dari Upaya peningkatan pemahaman dan keterampilan dalam pengembangan aplikasi berbasis Java, khususnya dalam konteks sistem manajemen kost.

Sistem manajemen kost menjadi sangat penting dalam mengelola dan memonitoring kegiatan sehari-hari yang terjadi dalam lingkungan hunian berbagi, seperti kos. Dalam era digitalisasi ini, penggunaan teknologi informasi menjadi kunci utama dalam meningkatkan efisiensi dan efektivitas dalam pengelolaan kost. Oleh karena itu, kami berusaha menggali dan menyajikan informasi yang relevan serta Solusi implementatif menggunakan Java GUI dengan bantuan NetBeans sebagai alat pengembangan.

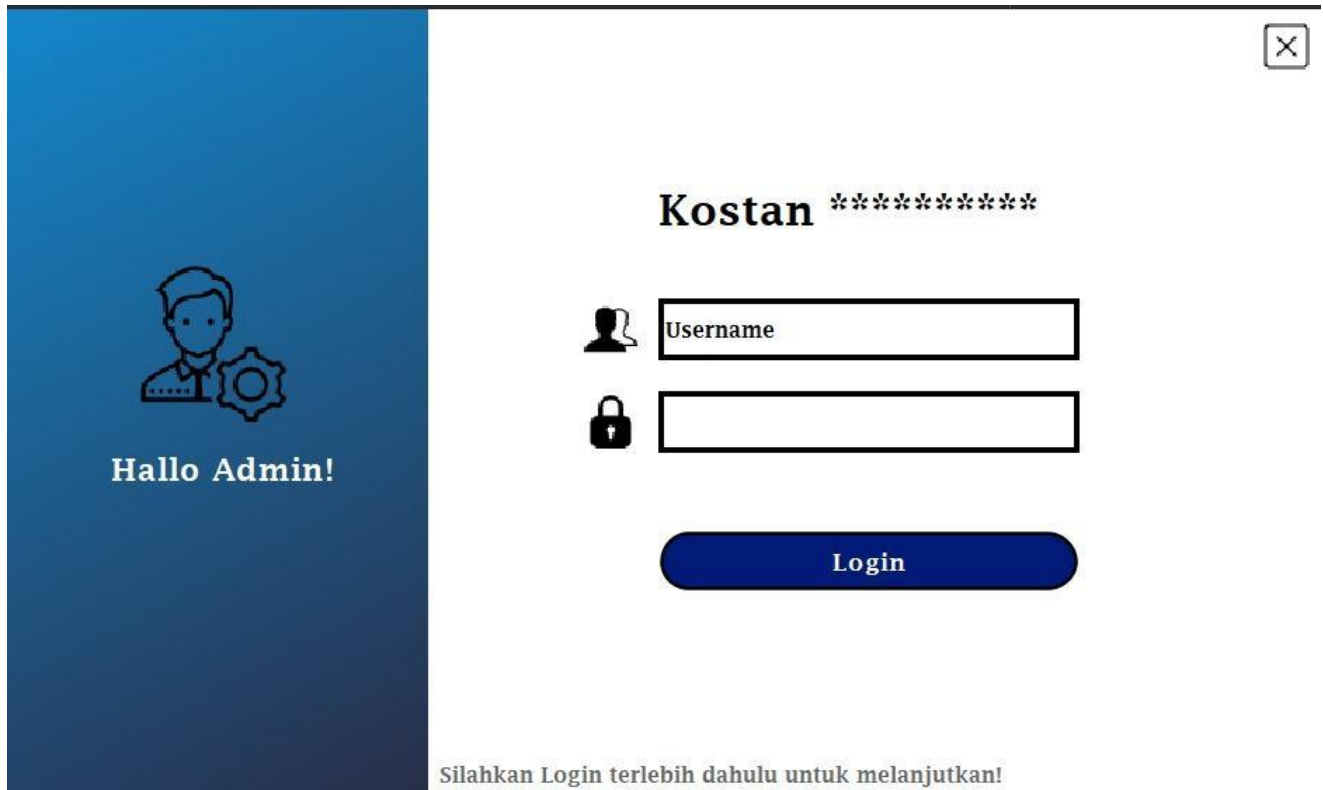
Modul ini dirancang untuk membantu pembaca memahami konsep dasar pembuatan antarmuka grafis(GUI) menggunakan Bahasa pemrograman Java dengan memanfaatkan lingkungan pengembangan NetBeans. Materi yang disajikan meliputi pembuatan form, pengelolaan tata letak, penggunaan komponen GUI, serta integrasi dengan basis data untuk mendukung fungsi manajemen kost.

Penulis menyadari bahwa penulisan modul ini masih jauh dari kesempurnaan. Oleh karena itu, kritik, saran, dan masukan dari pembaca sangat diharapkan demi perbaikan dan penyempurnaan materi ini di masa yang akan datang.

Akhir kata, penulis berharap modul ini dapat memberikan kontribusi positif bagi pembaca, khususnya bagi mereka yang tertarik dalam pengembangan aplikasi berbasis Java GUI untuk sistem manajemen kost. Semoga modul ini dapat menjadi panduan yang bermanfaat dalam memahami dan mengimplementasikan konsep konsep dasar tersebut.

Tampilan dan Cara penggunaan Aplikasi

- Melakukan Login dengan memasukkan Username dan Password yang sudah terdaftar atau menggunakan Username dan Password Default.



Kostan *****

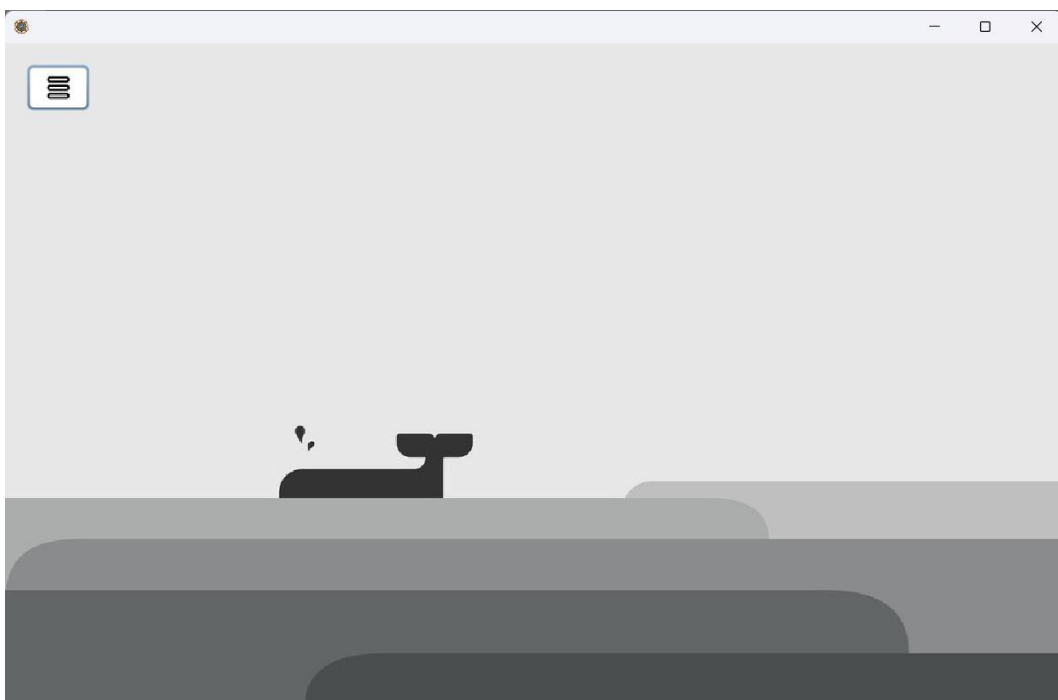
Hallo Admin!

Username

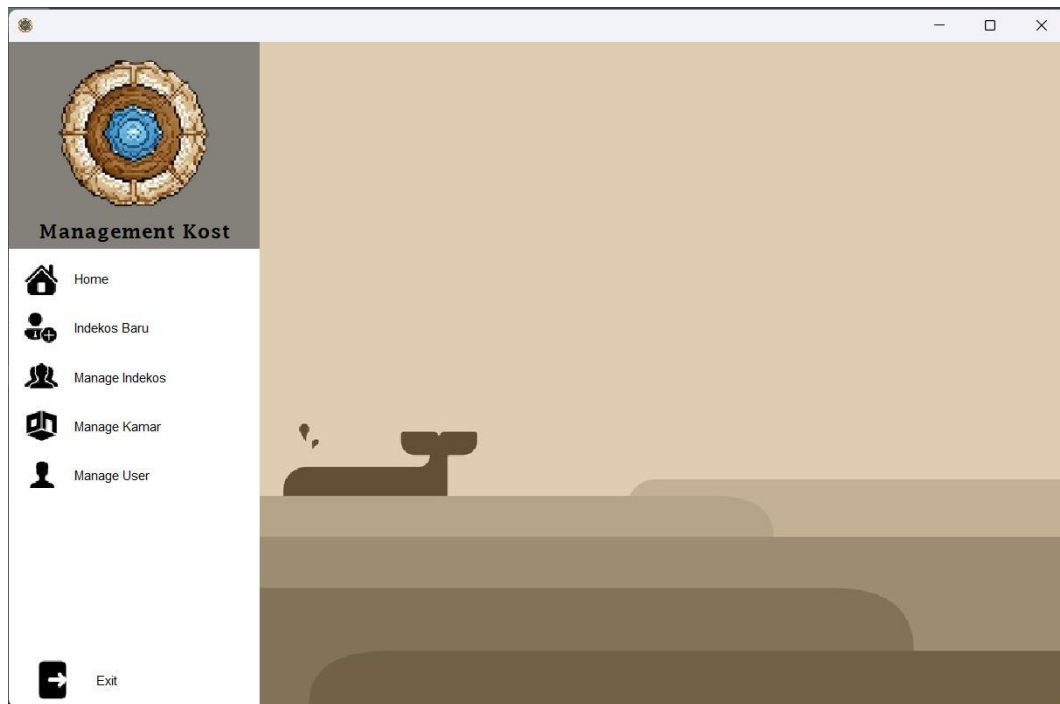
Login

Silahkan Login terlebih dahulu untuk melanjutkan!

- Setelah melakukan Login akan masuk ke Halaman Menu Utama.



- Setelah Masuk ke Menu Utama, ada Icon Menu di Pojok kanan atas untuk melihat sub Menu seperti Home, Indekos Baru, Manage Indekos, Manage Kamar dan Manage User.



- Menu Indekos Baru dapat Digunakan untuk menambahkan Penghuni kost Baru. Jika memilih Tambah maka data akan ditambahkan dan disimpan, dan jika memilih batal maka data tidak akan ditambahkan.

Masukkan Data Indekost

Nama	<input type="text"/>	Status	<input type="text" value="Menikah"/>
NIK	<input type="text"/>	Kesibukan	<input type="text" value="Bekerja"/>
No Hp	<input type="text"/>	Gender	<input type="radio"/> Laki-Laki <input type="radio"/> Perempuan
Tipe Kamar	<input type="text" value="Tipe A"/>	Tarif	<input type="text" value="500000"/> per Bulan
Durasi Sewa	<input type="text"/> Bulan		
Check-In	<input type="text" value="20-12-2023"/> <input type="button" value="..."/>	<input type="button" value="Ok"/>	

- Menu Manage Indekos merupakan menu untuk menampilkan data data para penghuni kost secara rinci, Pada menu ini juga terdapat fitur search untuk mencari data penghuni kost, serta fitur hapus untuk menghapus data yang ada pada tabel.

NIK	Nama	No Handphone	Kestibukan	Gender	Status	Tipe Kamar	Check In	Lama Sewa	Harga	Total Biaya
12345678901...	Jeni Fatmawati	87654543	Mahasiswa	Perempuan	Belum Menik...	Tipe B	24-1-2024	12	700000	8400000
61710502030...	Rizky Gunaw...	89631213121	Mahasiswa	Laki-Laki	Belum Menik...	Tipe C	10-1-2024	6	1000000	6000000

- Selanjutnya ada menu manage kamar yang berfungsi mengatur dan menambahkan data kamar.

Tipe Kamar	Harga	Kamar Tersedia
Tipe A	500000	10
Tipe B	700000	10
Tipe C	1000000	10
Tipe D	1300000	10
Tipe E	1500000	10

- Ada menu manage admin juga yang berfungsi mengatur dan menambahkan admin untuk melakukan login, disitu juga terdapat fitur atau tombol untuk melakukan refresh agar data yang baru diinputkan bisa muncul setelah di refresh.

Manajemen Data Admin

Nama Admin

Username

Password

Jabatan

Owner

Hapus

Refresh

Cari

Nama Admin	Username	Jabatan Admin
admin	Master Admin	Owner
rizky	rizky gunawan	Owner

Simpan

Bersihkan

Home

Syntax yang digunakan

- Menu Login

```
1 package MainForm;
2
3 import java.sql.*;
4 import javax.swing.*;
5 import MainForm.config.Koneksi;
6 import java.awt.Toolkit;
7 import java.awt.event.KeyEvent;
```

```
9 public class Login extends javax.swing.JFrame {
10
11     Connection conn = Koneksi.getKoneksi();
12     ResultSet rs = null;
13     PreparedStatement pst = null;
14
15     public Login() {
16         initComponents();
17         setLocationRelativeTo(c: null);
18         setIcon();
19     }
20
21     private void setIcon(){
22         setIconImage(image: Toolkit.getDefaultToolkit().getImage(url: getClass().getResource(name: "/Icon/144noBg.png")));
23     }
```

```
210 private void jLabel1MouseClicked(java.awt.event.MouseEvent evt) {
211     dispose();
212 }
213
214 private void tfUsernameFocusGained(java.awt.event.FocusEvent evt) {
215     // Agar tulisan username hilang saat di klik
216     String username = tfUsername.getText();
217     if (username.equals(anObject: "Username")){
218         tfUsername.setText(t: "");
219     }
220
221 private void tfUsernameFocusLost(java.awt.event.FocusEvent evt) {
222     String user = tfUsername.getText();
223     if(user.equals(anObject: "") || user.equals(anObject: "Username")){
224         tfUsername.setText(t: "Username");
225     }
226 }
```

```

229 private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {
230     String sql = "select username, password from sistemkost.login where username=? and password=?";
231     try{
232         pst = conn.prepareStatement(string:sql);
233         pst.setString(i: 1, string:tfUsername.getText());
234         pst.setString(i: 2, string:String.valueOf(data: tfPassword.getPassword()));
235         rs = pst.executeQuery();
236         if(rs.next()){
237             JOptionPane.showMessageDialog(parentComponent: null, message: "Login Berhasil");
238             Home home = new Home();
239             home.setVisible(b: true);
240             this.dispose();
241         }else{
242             JOptionPane.showMessageDialog(parentComponent: null, message: "Username atau Password Salah!");
243         }
244     }catch(Exception e){
245         JOptionPane.showMessageDialog(parentComponent: null, message: e.getMessage());
246         e.printStackTrace();
247     }
248 }
249
250 private void tfPasswordFocusGained(java.awt.event.FocusEvent evt) {
251     String pass = tfPassword.getText();
252     if (pass.equals(anObject: "Password")){
253         tfPassword.setText(t: "");
254     }
255 }

```

```

257 private void tfPasswordFocusLost(java.awt.event.FocusEvent evt) {
258     String user = tfUsername.getText();
259     if(user.equals(anObject: "")||user.equals(anObject: "Username")){
260         tfUsername.setText(t: "Username");
261     }
262 }
263
264 private void tfUsernameKeyPressed(java.awt.event.KeyEvent evt) {
265     if(evt.getKeyCode() == KeyEvent.VK_ENTER)
266         tfPassword.requestFocus();
267 }
268
269 private void tfPasswordKeyPressed(java.awt.event.KeyEvent evt) {
270     if(evt.getKeyCode() == KeyEvent.VK_ENTER)
271         btnLoginActionPerformed(evt: null);
272 }
273
274 public static void main(String args[]) {
275     /* Set the Nimbus look and feel */
276     Look and feel setting code (optional)
277
278     /* Create and display the form */
279     java.awt.EventQueue.invokeLater(new Runnable() {
280         public void run() {
281             new Login().setVisible(b: true);
282         }
283     });
284 }

```


- Menu Home

```

1  package MainForm;
2
3  import MainForm.SideBar.Header;
4  import java.awt.Color;
5  import java.awt.Toolkit;
6  import javax.swing.drawer.Drawer;
7  import javax.swing.drawer.DrawerController;
8  import javax.swing.drawer.DrawerItem;
9  import javax.swing.drawer.EventDrawer;
10 import javax.swing.ImageIcon;

```

```

12 public class Home extends javax.swing.JFrame {
13
14     private DrawerController drawer;
15
16     public Home() {
17         initComponents();
18         setIcon();
19         SideBar();
20     }
21
22     private void setIcon() {
23         setIconImage(image: Toolkit.getDefaultToolkit().getImage(url: getClass().getResource(name: "/Icon/144noBg.png"))));
24     }

```

```

26     private void SideBar() {
27         drawer = Drawer.newDrawer(this)
28             .header(new Header())
29             .space(height: 5)
30             .background(new Color(r: 204, g: 142, b: 53))
31             .backgroundTransparent(backgroundTransparent: 0.3f)
32             .drawerWidth(drawerWidth: 230)
33             .enableScroll(enableScroll: true)
34             .addChild(component: new DrawerItem(text: "Home").icon(new ImageIcon(location: getClass().getResource(name: "/Icon/Home.png"))).build())
35             .addChild(component: new DrawerItem(text: "Indekos Baru").icon(new ImageIcon(location: getClass().getResource(name: "/Icon/Add Administrator.png"))).build())
36             .addChild(component: new DrawerItem(text: "Manage Indekos").icon(new ImageIcon(location: getClass().getResource(name: "/Icon/People.png"))).build())
37             .addChild(component: new DrawerItem(text: "Manage Kamar").icon(new ImageIcon(location: getClass().getResource(name: "/Icon/Room.png"))).build())
38             .addChild(component: new DrawerItem(text: "Manage User").icon(new ImageIcon(location: getClass().getResource(name: "/Icon/User.png"))).build())
39             .addFooter(component: new DrawerItem(text: "Exit").icon(new ImageIcon(location: getClass().getResource(name: "/Icon/Logout.png"))).build())
40             .event(new EventDrawer() {
41
42                 @Override
43                 public void selected(int i, DrawerItem di) {
44                     PilihItem(drawerItem: di);
45                 }
46             })
47             .build();
48     }

```

```

49     private void PilihItem(DrawerItem drawerItem) {
50         String itemName = drawerItem.getText();
51         switch (itemName) {
52             case "Home":
53                 new Home().setVisible(b: true);
54                 this.dispose();
55                 break;
56             case "Indekos Baru" :
57                 new TambahDaftar().setVisible(b: true);
58                 this.dispose();
59                 break;
60             case "Manage Indekos":
61                 new Penyewa().setVisible(b: true);
62                 this.dispose();
63                 break;
64             case "Manage Kamar":
65                 new DataKamar().setVisible(b: true);
66                 this.dispose();
67                 break;
68             case "Manage User":
69                 new Admin().setVisible(b: true);
70                 this.dispose();
71                 break;
72             case "Exit":
73                 new Login().setVisible(b: true);
74                 this.dispose();
75                 break;
76             default:
77                 System.out.println(x: "Invalid selection");
78         }
79     }

```

```

108     private void btnMenuActionPerformed(java.awt.event.ActionEvent evt) {
109         if(drawer.isShow()){
110             drawer.hide();
111         } else {
112             drawer.show();
113         }
114     }
115
116     public static void main(String args[]) {
117         /* Set the Nimbus look and feel */
118         Look and feel setting code (optional)
119
120         /* Create and display the form */
121         java.awt.EventQueue.invokeLater(new Runnable() {
122             public void run() {
123                 new Home().setVisible(b: true);
124             }
125         });
126     }

```

- Indekos Baru atau Menambahkan data Indekos

```

1 package MainForm;
2
3 import java.sql.*;
4 import MainForm.config.Koneksi;
5 import com.raven.datechooser.SelectedDate;
6 import java.awt.HeadlessException;
7 import java.awt.Toolkit;
8 import java.awt.event.KeyEvent;
9 import javax.swing.JOptionPane;

```

```

11 public class TambahDaftar extends javax.swing.JFrame {
12
13     Connection conn = Koneksi.getKoneksi();
14     ResultSet rs = null;
15     PreparedStatement pst = null;
16
17     private static final int max_nik = 16;
18
19
20     public TambahDaftar() {
21         initComponents();
22         updateTable();
23         setLocationRelativeTo(c: null);
24         setIcon();
25         reset();
26     }
27
28     private void setIcon(){
29         setIconImage(image: Toolkit.getDefaultToolkit().getImage(url:getClass().getResource(name: "/Icon/144noBg.png")));
30     }
31
32     private void reset(){ //untuk mereset data saat sudah diinput
33         tfNama.setText(t: null);
34         tfNIK.setText(t: null);
35         tfNoHp.setText(t: null);
36     }

```

```

38     private void updateTable(){
39         String sql = "select * from sistemkost.data kamar";
40         try {
41             pst = conn.prepareStatement(string:sql);
42             rs = pst.executeQuery();
43             cmbTipe.removeAllItems();
44             while (rs.next()){
45                 String jenisKamar = rs.getString(string:"jenis");
46                 cmbTipe.addItem(item: jenisKamar);
47             }
48         } catch (Exception e) {
49             JOptionPane.showMessageDialog(parentComponent: null, message: e);
50         }
51     }
52

```

```

438     private void btnTambahActionPerformed(java.awt.event.ActionEvent evt) {
439         String sql = "INSERT INTO sistemkost.data indekos(nama, nik, hp, status, kesibukan, gender, tipe, harga, tanggal, durasi, total) VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
440         try{
441             SelectedDate selectedDate = date.getSelectedDate();
442             String dateString = selectedDate.getDay() + "-" + selectedDate.getMonth() + "-" + selectedDate.getYear();
443
444             int durasi = Integer.parseInt(t: tfSewa.getText());
445
446             long nik = Long.parseLong(t: tfNIK.getText());
447             long noHp = Long.parseLong(t: tfNoHp.getText());
448             long harga = Long.parseLong(t: tfTarif.getText());
449             String gender = rbGenderLk.isSelected() ? "Laki-Laki" : "Perempuan";
450
451             long totalByr = harga * durasi;

```

```

453     pst = conn.prepareStatement(string:sql);
454     pst.setString(i: 1, string:tfNama.getText());
455     pst.setLong(i: 2, 1: nik);
456     pst.setLong(i: 3, 1: noHp);
457     pst.setString(i: 4, (String) cmbStatus.getSelectedItemAt());
458     pst.setString(i: 5, (String) cmbKesibukan.getSelectedItemAt());
459     pst.setString(i: 6, (String) gender);
460     pst.setString(i: 7, (String) cmbTipe.getSelectedItemAt());
461     pst.setLong(i: 8, 1: harga);
462     pst.setString(i: 9, string:dateString);
463     pst.setInt(i: 10, i1: durasi);
464     pst.setLong(i: 11, 1: totalByr);
465
466     int jumlahBaris = pst.executeUpdate();
467
468     if(jumlahBaris > 0){
469         JOptionPane.showMessageDialog(parentComponent: null, message: "Data Tersimpan");
470         Penyewa indekos = new Penyewa();
471         indekos.setVisible(b: true);
472         this.dispose();
473     } else {
474         JOptionPane.showMessageDialog(parentComponent: null, message: "Data Gagal Disimpan");
475     }
476
477     }catch (SQLException | HeadlessException e) {
478         e.printStackTrace();
479         JOptionPane.showMessageDialog(parentComponent: null, "Terjadi Kesalahan Umum:" + e.getMessage());
480     }
481     reset();
482 }

```

```

485     reset();
486     Penyewa indekos = new Penyewa();
487     indekos.setVisible(b: true);
488     this.dispose();
489 }
490
491 private void cmbTipeActionPerformed(java.awt.event.ActionEvent evt) {
492     String sql = "SELECT harga FROM data kamar WHERE jenis=?";
493     try {
494         String selectedTipe = (String) cmbTipe.getSelectedItemAt();
495
496         if (selectedTipe != null) {
497             pst = conn.prepareStatement(string:sql);
498             pst.setString(i: 1, string:selectedTipe);
499             ResultSet rs = pst.executeQuery();
500             if(rs.next()) {
501                 tfTarif.setText(t: rs.getString(string:"harga"));
502             } else {
503                 tfTarif.setText(t: "");
504             }
505         } else {
506             tfTarif.setText(t: "");
507         }
508     } catch (Exception e){
509         JOptionPane.showMessageDialog(parentComponent: null, message: e.getMessage());
510     }
511 }
512

```

```

514 private void btnTampilTglActionPerformed(java.awt.event.ActionEvent evt) {
515     date.showPopup();
516 }
517 private void tfNamaKeyPressed(java.awt.event.KeyEvent evt) {
518     if(evt.getKeyCode() == KeyEvent.VK_ENTER)
519         tfNIK.requestFocus();
520 }
521 private void tfNIKKeyPressed(java.awt.event.KeyEvent evt) {
522     if(evt.getKeyCode() == KeyEvent.VK_ENTER)
523         tfNoHp.requestFocus();
524 }
525 private void tfNoHpKeyPressed(java.awt.event.KeyEvent evt) {
526     if(evt.getKeyCode() == KeyEvent.VK_ENTER)
527         cmbTipe.requestFocus();
528 }
529 private void cmbTipeKeyPressed(java.awt.event.KeyEvent evt) {
530     if(evt.getKeyCode() == KeyEvent.VK_ENTER)
531         tfSewa.requestFocus();
532 }
533 private void tfSewaKeyPressed(java.awt.event.KeyEvent evt) {
534     if(evt.getKeyCode() == KeyEvent.VK_ENTER)
535         tfTanggal.requestFocus();
536 }
537 private void tfTanggalKeyPressed(java.awt.event.KeyEvent evt) {
538     if(evt.getKeyCode() == KeyEvent.VK_ENTER)
539         cmbStatus.requestFocus();
540 }
541 private void cmbStatusKeyPressed(java.awt.event.KeyEvent evt) {
542     if(evt.getKeyCode() == KeyEvent.VK_ENTER)
543         cmbStatus.requestFocus();
544 }
545 }

```

- Manage Indekos atau Penyewa

```
1 package MainForm;  
2  
3 import MainForm.config.Koneksi;  
4 import java.awt.Toolkit;  
5 import javax.swing.JOptionPane;  
6 import javax.swing.table.DefaultTableModel;  
7 import java.sql.*;  
8 import javax.swing.RowFilter;  
9 import javax.swing.table.TableRowSorter;  
10 import table.TableCustom;
```

```
12 public class Penyewa extends javax.swing.JFrame {  
13  
14     Connection conn = Koneksi.getKoneksi();  
15     ResultSet rs = null;  
16     PreparedStatement pst = null;  
17  
18     public Penyewa() {  
19         initComponents();  
20         menampilkanTable();  
21         setLocationRelativeTo(c: null);  
22         setIcon();  
23  
24         CTable.apply(scroll:jScrollPane, type: TableCustom.TableType.DEFAULT);  
25     }  
26  
27     private void setIcon(){  
28         setIconImage(image: Toolkit.getDefaultToolkit().getImage(url:getClass().getResource(name: "/Icon/144noBg.png")));  
29     }  
30 }
```

```

31 public void menampilkanTable() {
32     DefaultTableModel tb = new DefaultTableModel();
33     tb.addColumn(columnName: "NIK");
34     tb.addColumn(columnName: "Nama");
35     tb.addColumn(columnName: "No Handphone");
36     tb.addColumn(columnName: "Kesibukan");
37     tb.addColumn(columnName: "Gender");
38     tb.addColumn(columnName: "Status");
39     tb.addColumn(columnName: "Tipe Kamar");
40     tb.addColumn(columnName: "Check In");
41     tb.addColumn(columnName: "Lama Sewa");
42     tb.addColumn(columnName: "Harga");
43     tb.addColumn(columnName: "Total Biaya");
44     daftarTable.setModel(dataModel: tb);
45     try{
46         String sql = "select * from data_indekos";
47         pst = conn.prepareStatement(string: sql);
48         rs = pst.executeQuery();
49         while (rs.next()){
50             tb.addRow(new Object[]{
51                 rs.getString(string: "nik"),
52                 rs.getString(string: "nama"),
53                 rs.getString(string: "hp"),
54                 rs.getString(string: "kesibukan"),
55                 rs.getString(string: "gender"),
56                 rs.getString(string: "status"),
57                 rs.getString(string: "tipe"),
58                 rs.getString(string: "tanggal"),
59                 rs.getString(string: "durasi"),
60                 rs.getString(string: "harga"),
61                 rs.getString(string: "total")
62             });
63         }
64     } catch (Exception e) {
65         JOptionPane.showMessageDialog(parentComponent: null, "Data Gagal Ditampilkan" + e.getMessage());
66     }
67 }

```

```

68 private void editSelectedRow() {
69     int selectedRow = daftarTable.getSelectedRow();
70     if (selectedRow != -1) {
71         int selectedColumn = daftarTable.getSelectedColumn();
72         if (selectedColumn != -1) {
73             String columnName = daftarTable.getColumnModel().getColumnName(selectedColumn);
74             String newValue = JOptionPane.showInputDialog(this, "Masukkan nilai baru untuk kolom " + columnName + ":");
75             // Perbarui nilai di tabel
76             daftarTable.setValueAt(newValue, selectedRow, selectedColumn);
77             // Perbarui data di database (ganti dengan metode sesuai kebutuhan)
78             updateDataInDatabase(
79                 daftarTable.getValueAt(selectedRow, 0).toString(), // NIK sebagai primary key
80                 columnName,
81                 newValue
82             );
83         } else {
84             JOptionPane.showMessageDialog(this, "Pilih kolom yang akan diubah");
85         }
86     } else {
87         JOptionPane.showMessageDialog(this, "Pilih data yang akan diubah");
88     }
89 }

```

```

89 private void updateDataInDatabase(String nik, String columnName, String newValue) {
90     try {
91         String updateQuery = "UPDATE data_indekos SET " + columnName + "=? WHERE nik=?";
92         pst = conn.prepareStatement(updateQuery);
93         pst.setString(1, newValue);
94         pst.setString(2, nik);
95         int rowsAffected = pst.executeUpdate();
96         if (rowsAffected > 0) {
97             JOptionPane.showMessageDialog(this, "Data berhasil diubah.", "Sukses", JOptionPane.INFORMATION_MESSAGE);
98         } else {
99             JOptionPane.showMessageDialog(this, "Tidak ada baris yang terpengaruh. Data mungkin tidak ditemukan.", "Peringatan", JOptionPane.WARNING_MESSAGE);
100         }
101     } catch (SQLException e) {
102         e.printStackTrace();
103         JOptionPane.showMessageDialog(this, "Error mengubah data: " + e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
104     }
105 }

```

```

114 private void performSearch() {
115     DefaultTableModel df = (DefaultTableModel) daftarTable.getModel();
116     TableRowSorter<DefaultTableModel> obj = new TableRowSorter<>(df);
117     daftarTable.setRowSorter(obj);
118     obj.setRowFilter(filter: RowFilter.regexFilter("(?i)" + tfCari.getText()));
119     // "(?i)" untuk konversi keyword ke huruf kecil
120     // "^" untuk memfilter agar hanya kata depan saja yang muncul
121 }

```

```

123 private void hapusDataDB(Object primaryKey) {
124     try {
125         String deleteQuery = "DELETE FROM data_indeks WHERE nik = ?";
126         pst = conn.prepareStatement(deleteQuery);
127         pst.setObject(1, primaryKey);
128         int kolomDipilih = pst.executeUpdate();
129
130         System.out.println("Kolom Dipilih" + kolomDipilih);
131         System.out.println("Primary Key Dihapus" + primaryKey);
132
133         if (kolomDipilih > 0) {
134             JOptionPane.showMessageDialog(parentComponent, this, message: "Data berhasil dihapus dari database.", title: "Sukses", messageType: JOptionPane.INFORMATION_MESSAGE);
135         } else {
136             JOptionPane.showMessageDialog(parentComponent, this, message: "Data tidak ditemukan dalam database.", title: "Peringatan", messageType: JOptionPane.WARNING_MESSAGE);
137         }
138     } catch (SQLException e) {
139         e.printStackTrace();
140         JOptionPane.showMessageDialog(parentComponent, this, message: "Error menghapus data dari database.", title: "Error", messageType: JOptionPane.ERROR_MESSAGE);
141     }
142 }

```

```

337 private void btnTambahActionPerformed(java.awt.event.ActionEvent evt) {
338     TambahDaftar tambah = new TambahDaftar();
339     tambah.setVisible(b: true);
340     this.dispose();
341 }
342
343 private void fButton3ActionPerformed(java.awt.event.ActionEvent evt) {
344     new Home().setVisible(b: true);
345     this.dispose();
346 }
347
348 private void daftarTableComponentAdded(java.awt.event.ContainerEvent evt) {
349     // TODO add your handling code here:
350 }
351
352 private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {
353     DefaultTableModel model = (DefaultTableModel) daftarTable.getModel();
354     if (daftarTable.getSelectedRow() == -1) {
355         if (daftarTable.getRowCount() == 0) {
356             JOptionPane.showMessageDialog(parentComponent: null, message: "Tidak Ada Data");
357         } else {
358             JOptionPane.showMessageDialog(parentComponent: null, message: "Pilih Data Yang Akan Dihapus");
359         }
360     } else {
361         Object primaryKey = daftarTable.getValueAt(row: daftarTable.getSelectedRow(), column: 0);
362         model.removeRow(row: daftarTable.getSelectedRow());
363         hapusDataDB(primaryKey);
364     }
365 }

```

```

367 private void tfCariMouseClicked(java.awt.event.MouseEvent evt) {
368     tfCari.setText(t: "");
369 }
370
371 private void tfCariKeyTyped(java.awt.event.KeyEvent evt) {
372     if (evt.getKeyChar() == '\n') { // agar saat menekan tombol enter langsung melakukan pencarian
373         performSearch();
374     }
375 }
376
377 private void btnCariActionPerformed(java.awt.event.ActionEvent evt) {
378     performSearch();
379 }
380
381 private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {
382     editSelectedRow();
383 }

```


- Manage Kamar

```
1 package MainForm;
2
3 import java.sql.*;
4 import MainForm.config.Koneksi;
5 import java.awt.HeadlessException;
6 import java.awt.Toolkit;
7 import java.awt.event.KeyEvent;
8 import javax.swing.JOptionPane;
9 import javax.swing.table.DefaultTableModel;
10 import table.TableCustom;
11
```

```
12 public class DataKamar extends javax.swing.JFrame {
13
14     Connection conn = Koneksi.getKoneksi();
15     ResultSet rs = null;
16     PreparedStatement pst = null;
17
18     public DataKamar() {
19         initComponents();
20         setLocationRelativeTo(c: null);
21         menampilkanTable();
22         setIcon();
23
24         CTable.apply(scroll: jScrollPane1, type: TableCustom.TableType.DEFAULT);
25     }
26
27     private void setIcon(){
28         setIconImage(image: Toolkit.getDefaultToolkit().getImage(url: getClass().getResource(name: "/Icon/144noBg.png")));
29     }
30
```

```
31     private void reset() { //untuk mereset data saat sudah diinput
32         tfTipeKamar.setText(t: null);
33         tfHarga.setText(t: null);
34         tfJumlahKamar.setText(t: null);
35     }
36
37     public void menampilkanTable() {
38         DefaultTableModel tb = new DefaultTableModel();
39         tb.addColumn(columnName: "Tipe Kamar");
40         tb.addColumn(columnName: "Harga");
41         tb.addColumn(columnName: "Kamar Tersedia");
42         jTable1.setModel(dataModel: tb);
43         try{
44             String sql = "select * from data_kamar";
45             pst = conn.prepareStatement(string: sql);
46             rs = pst.executeQuery();
47             while (rs.next()){
48                 tb.addRow(new Object[]{
49                     rs.getString(string: "jenis"),
50                     rs.getString(string: "harga"),
51                     rs.getString(string: "jumlah"),
52                 });
53             }
54         } catch (Exception e) {
55             JOptionPane.showMessageDialog(parentComponent: null, "Data Gagal Ditampilkan" + e.getMessage());
56         }
57     }
58
```

```

59 private void deleteFromDatabase(Object primaryKey) {
60     try {
61         String deleteQuery = "DELETE FROM data_kamar WHERE jenis = ?";
62         pst = conn.prepareStatement(string: deleteQuery);
63         pst.setObject(i: 1, o: primaryKey);
64         int kolomDipilih = pst.executeUpdate();
65
66         System.out.println("Kolom Dipilih" + kolomDipilih);
67         System.out.println("Primary Key Dihapus" + primaryKey);
68
69         if (kolomDipilih > 0) {
70             JOptionPane.showMessageDialog(parentComponent: this, message: "Data berhasil dihapus dari database.", title: "Sukses", messageType: JOptionPane.INFORMATION_MESSAGE);
71         } else {
72             JOptionPane.showMessageDialog(parentComponent: this, message: "Data tidak ditemukan dalam database.", title: "Peringatan", messageType: JOptionPane.WARNING_MESSAGE);
73         }
74     } catch (SQLException e) {
75         e.printStackTrace();
76         JOptionPane.showMessageDialog(parentComponent: this, message: "Error menghapus data dari database.", title: "Error", messageType: JOptionPane.ERROR_MESSAGE);
77     }
78 }
79

```

```

80 private void refreshData() {
81     DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
82     model.setRowCount(rowCount: 0); // Hapus semua baris saat ini dari tabel
83     // Panggil kembali metode untuk menampilkan data baru ke dalam tabel
84     menampilkanTable();
85 }

```

```

317 private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {
318     String sql = "INSERT INTO sistemkost.data_kamar(jenis, harga, jumlah) VALUES(?, ?, ?)";
319     try {
320         String tipe_kamar = tfTipeKamar.getText();
321         long harga_kamar = Long.parseLong(s: tfHarga.getText());
322         int jumlah_kamar = Integer.parseInt(s: tfJumlahKamar.getText());
323
324         pst = conn.prepareStatement(string: sql);
325         pst.setString(i: 1, string: tipe_kamar);
326         pst.setLong(i: 2, l: harga_kamar);
327         pst.setInt(i: 3, i: jumlah_kamar);
328
329         int jumlahBaris = pst.executeUpdate();
330
331         if (jumlahBaris > 0) {
332             JOptionPane.showMessageDialog(parentComponent: null, message: "Data Tersimpan");
333         } else {
334             JOptionPane.showMessageDialog(parentComponent: null, message: "Data Gagal Disimpan");
335         }
336     } catch (SQLException | HeadlessException e) {
337         e.printStackTrace();
338         JOptionPane.showMessageDialog(parentComponent: null, "Terjadi Kesalahan Umum:" + e.getMessage());
339     }
340     reset();
341 }

```

```

343 private void btnHomeActionPerformed(java.awt.event.ActionEvent evt) {
344     new Home().setVisible(b: true);
345     this.dispose();
346 }
347
348 private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {
349     DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
350     if (jTable1.getSelectedRow() == -1) {
351         if (jTable1.getRowCount() == 0) {
352             JOptionPane.showMessageDialog(parentComponent: null, message: "Tidak Ada Data");
353         } else {
354             JOptionPane.showMessageDialog(parentComponent: null, message: "Pilih Data Yang Akan Dihapus");
355         }
356     } else {
357         Object primaryKey = jTable1.getValueAt(row: jTable1.getSelectedRow(), column: 0);
358         model.removeRow(row: jTable1.getSelectedRow());
359         deleteFromDatabase(primaryKey);
360     }
361 }

```

- Data Admin

```
1 package MainForm;
2
3 import MainForm.config.Koneksi;
4 import java.awt.HeadlessException;
5 import java.awt.Toolkit;
6 import java.awt.event.KeyEvent;
7 import java.sql.SQLException;
8 import javax.swing.JOptionPane;
9 import java.sql.*;
10 import javax.swing.RowFilter;
11 import javax.swing.table.DefaultTableModel;
12 import javax.swing.table.TableRowSorter;
13 import table.TableCustom;
```

```
15 public class Admin extends javax.swing.JFrame {
16
17     Connection conn = Koneksi.getKoneksi();
18     ResultSet rs = null;
19     PreparedStatement pst = null;
20
21     public Admin() {
22         initComponents();
23         setLocationRelativeTo(c: null);
24         menampilkanTable();
25         setIcon();
26
27         CTable.apply(scroll:jScrollPane, type: TableCustom.TableType.DEFAULT);
28     }
```

```
30 private void setIcon(){
31     setIconImage(Image: Toolkit.getDefaultToolkit().getImage(uri.getClass().getResource(name: "/Icon/144noBg.png")));
32 }
33
34 public void menampilkanTable() {
35     DefaultTableModel tb = new DefaultTableModel();
36     tb.addColumn(columnName: "Nama Admin");
37     tb.addColumn(columnName: "Username");
38     tb.addColumn(columnName: "Jabatan Admin");
39     jTable1.setModel(dataModel:tb);
40     try{
41         String sql = "select * from login";
42         pst = conn.prepareStatement(string:sql);
43         rs = pst.executeQuery();
44         while (rs.next()){
45             tb.addRow(new Object[]{
46                 rs.getString(string:"username"),
47                 rs.getString(string:"nama"),
48                 rs.getString(string:"jabatan"),
49             });
50         }
51     }catch (Exception e) {
52         JOptionPane.showMessageDialog(parentComponent: null, "Data Gagal Ditampilkan" + e.getMessage());
53     }
54 }
55
56 private void reset(){ //untuk mereset data saat sudah diinput
57     tfNama.setText(t: null);
58     tfUsername.setText(t: null);
59     tfPassword.setText(t: null);
60 }
```

```

69 private void deleteFromDatabase(Object primaryKey) {
70     try {
71         String deleteQuery = "DELETE FROM login WHERE username = ?";
72         pst = conn.prepareStatement(deleteQuery);
73         pst.setObject(1, o.primaryKey);
74         int kolomDipilih = pst.executeUpdate();
75
76         if (kolomDipilih > 0) {
77             JOptionPane.showMessageDialog(parentComponent.this, message: "Data berhasil dihapus dari database.", title: "Sukses", messageType: JOptionPane.INFORMATION_MESSAGE);
78         } else {
79             JOptionPane.showMessageDialog(parentComponent.this, message: "Data tidak ditemukan dalam database.", title: "Peringatan", messageType: JOptionPane.WARNING_MESSAGE);
80         }
81     } catch (SQLException e) {
82         e.printStackTrace();
83         JOptionPane.showMessageDialog(parentComponent.this, message: "Error menghapus data dari database.", title: "Error", messageType: JOptionPane.ERROR_MESSAGE);
84     }
85 }

```

```

87 private void refreshData() {
88     DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
89     model.setRowCount(rowCount: 0); // Hapus semua baris saat ini dari tabel
90
91     // Panggil kembali metode untuk menampilkan data baru ke dalam tabel
92     menampilkanTable();
93 }

```

```

390 private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {
391     String sql = "INSERT INTO sistemkost.login(username, password, nama, jabatan) VALUES(?, ?, ?, ?)";
392     try {
393         pst = conn.prepareStatement(sql);
394         pst.setString(1, string: tfUsername.getText());
395         pst.setString(2, string: tfPassword.getText());
396         pst.setString(3, string: tfNama.getText());
397         pst.setString(4, (String) cmbJabatan.getSelectedItem());
398
399         int jumlahBaris = pst.executeUpdate();
400
401         if (jumlahBaris > 0) {
402             JOptionPane.showMessageDialog(parentComponent: null, message: "Data Tersimpan");
403         } else {
404             JOptionPane.showMessageDialog(parentComponent: null, message: "Data Gagal Disimpan");
405         }
406     } catch (SQLException | HeadlessException e) {
407         e.printStackTrace();
408         JOptionPane.showMessageDialog(parentComponent: null, "Terjadi Kesalahan Umum:" + e.getMessage());
409     }
410     reset();
411 }
412

```

```

415 private void btnBersihActionPerformed(java.awt.event.ActionEvent evt) {
416     tfNama.setText(t: "");
417     tfUsername.setText(t: "");
418     tfPassword.setText(t: "");
419 }
420
421 private void tfCariMouseClicked(java.awt.event.MouseEvent evt) {
422     tfCari.setText(t: "");
423 }
424
425 private void tfCariKeyTyped(java.awt.event.KeyEvent evt) {
426     if (evt.getKeyChar() == '\n') {
427         performSearch();
428     }
429 }
430
431 private void btnCariActionPerformed(java.awt.event.ActionEvent evt) {
432     performSearch();
433 }

```

```

435 private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {
436     DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
437     if(jTable1.getSelectedRow()==-1){
438         if(jTable1.getRowCount()==0){
439             JOptionPane.showMessageDialog(parentComponent: null, message: "Tidak Ada Data");
440         }else {
441             JOptionPane.showMessageDialog(parentComponent: null, message: "Pilih Data Yang Akan Dihapus");
442         }
443     } else {
444         Object primaryKey = jTable1.getValueAt(row:jTable1.getSelectedRow(), column:0);
445         model.removeRow(row: jTable1.getSelectedRow());
446         deleteFromDatabase(primaryKey);
447     }
448 }

```

```

450 private void btnHomeActionPerformed(java.awt.event.ActionEvent evt) {
451     new Home().setVisible(b: true);
452     this.dispose();
453 }
454
455 private void fButton1ActionPerformed(java.awt.event.ActionEvent evt) {
456     refreshData();
457 }
458
459 private void tfNamaKeyPressed(java.awt.event.KeyEvent evt) {
460     if(evt.getKeyCode() == KeyEvent.VK_ENTER)
461         tfUsername.requestFocus();
462 }
463
464 private void tfUsernameKeyPressed(java.awt.event.KeyEvent evt) {
465     if(evt.getKeyCode() == KeyEvent.VK_ENTER)
466         tfPassword.requestFocus();
467 }
468
469 private void tfPasswordKeyPressed(java.awt.event.KeyEvent evt) {
470     if(evt.getKeyCode() == KeyEvent.VK_ENTER)
471         btnSimpanActionPerformed(evt: null);
472 }
473
474 public static void main(String args[]) {
475     Look and feel setting code (optional)
476
477     java.awt.EventQueue.invokeLater(new Runnable() {
478         public void run() {
479             new Admin().setVisible(b: true);
480         }
481     });
482 }
483
484 }

```

Struktur Basis Database

- Rancangan Struktur Database

menggunakan 3 buah tabel yang terdiri dari data_indekos, data_kamar dan login
















Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> data_indekos	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> data_kamar	★ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> login	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
3 tables	Sum	9	InnoDB	utf8mb4_general_ci	64.0 KiB	0 B

- Table data_indekos berisikan data data para penghuni kost yang telah terdaftar, meliputi nama, nik, nomor hp, status, kesibukan, gender, tipe, harga kamar per bulan, tanggal check in, durasi sewa, total biaya.

			nama	nik	hp
<input type="checkbox"/>	Edit	Copy	Delete	Jeni Fatmawati	1234567890123456 87654543
<input type="checkbox"/>	Edit	Copy	Delete	Rizky Gunawan	6171050203040002 89631213121

status	kesibukan	gender	tipe	harga	tanggal	durasi	total
Belum Menikah	Mahasiswa	Perempuan	Tipe B	700000	24-1-2024	12	8400000
Belum Menikah	Mahasiswa	Laki-Laki	Tipe C	1000000	10-1-2024	6	6000000

- table data_kamar berisikan data data kamar seperti jenis kamar, harga perbulan, dan jumlah kamar yang tersedia.

				jenis	harga	jumlah
<input type="checkbox"/>	 Edit	 Copy	 Delete	Tipe A	500000	10
<input type="checkbox"/>	 Edit	 Copy	 Delete	Tipe B	700000	10
<input type="checkbox"/>	 Edit	 Copy	 Delete	Tipe C	1000000	10
<input type="checkbox"/>	 Edit	 Copy	 Delete	Tipe D	1300000	10
<input type="checkbox"/>	 Edit	 Copy	 Delete	Tipe E	1500000	10

- table login pada database berisikan data data user untuk melakukan login, yang berisikan nama lengkap, username dan juga password.

username	password	nama	jabatan
admin	admin	Master Admin	Owner
rizky	rizky	rizky gunawan	Owner

KESIMPULAN

Dalam pengembangan Sistem Manajemen Indekos menggunakan Java GUI di NetBeans, kami berhasil menciptakan sebuah aplikasi yang dapat meningkatkan efisiensi dan akurasi dalam manajemen indekos. Aplikasi ini tidak hanya memberikan kemudahan dalam pemantauan kamar dan penyewa, tetapi juga menyediakan berbagai fitur tambahan yang dapat mendukung keberlanjutan operasional indekos.

Melalui antarmuka grafis yang user-friendly, pengguna dapat dengan mudah mengelola informasi mengenai ketersediaan kamar, data penyewa. Fungsionalitas pencarian dan filter memberikan kecepatan akses ke data yang dibutuhkan, sementara laporan yang dihasilkan oleh sistem memberikan wawasan mendalam mengenai performa indekos.

Penerapan konsep Object-Oriented Programming (OOP) dalam pengembangan dengan Java memungkinkan modularitas yang baik, mempermudah pemeliharaan dan pengembangan sistem di masa mendatang. Selain itu, penggunaan Java GUI di NetBeans memberikan keunggulan dalam desain antarmuka yang menarik dan responsif.

Dalam konteks manajemen indekos, aplikasi ini dapat menjadi solusi efektif untuk meningkatkan produktivitas pemilik indekos dan memastikan pengalaman nyaman bagi penyewa. Integrasi teknologi ini diharapkan dapat membawa perubahan positif dalam efisiensi operasional dan pengelolaan data.

Dengan demikian, proyek ini mencerminkan dedikasi kami untuk menghadirkan solusi teknologi yang inovatif dan relevan dalam konteks manajemen indekos. Meskipun telah mencapai sejumlah pencapaian yang signifikan, pengembangan dan peningkatan lebih lanjut terus diperlukan untuk

memastikan keberlanjutan dan kesesuaian aplikasi ini dengan perkembangan kebutuhan manajemen indekos di masa mendatang.