

**Pemrograman Berorientasi Objek Praktik**  
**“Pengembangan Sistem Informasi Berbasis Desktop dengan Python (Kivy), PostgreSQL,  
dan Integrasi Excel”**

Dosen Pengampu : Muhammad Henri Syuhada S.T.,M.T.



**Nama : Dery Wahyu Perdana**

**NPM : 5230411153**

**Kelas : Informatika – IV**

**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS SAINS DAN TEKNOLOGI**  
**UNIVERSITAS TEKNOLOGI YOGYAKARTA**

**2024**

## Pendahuluan

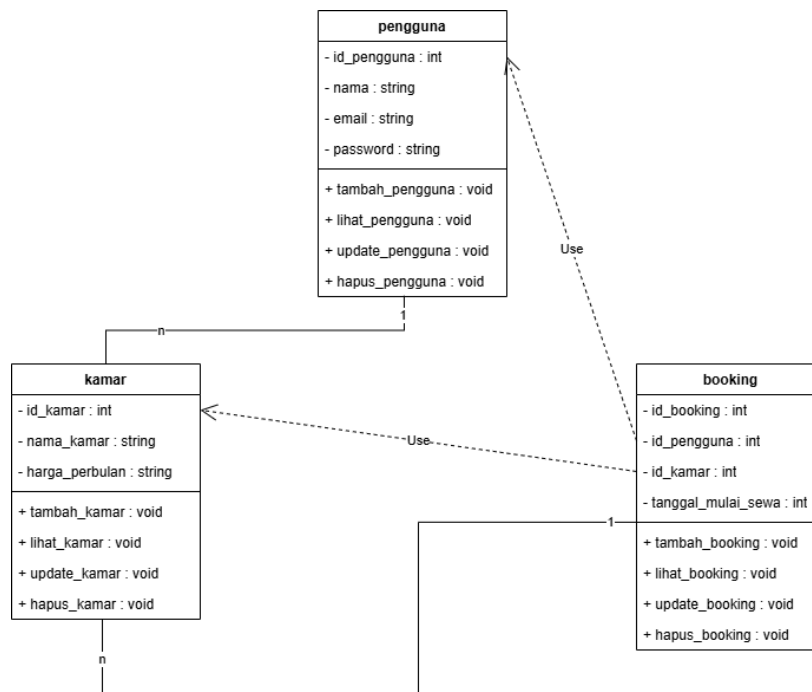
Aplikasi rumah\_kita adalah sistem manajemen rumah kost berbasis GUI yang dikembangkan menggunakan framework **Kivy** dan terintegrasi dengan database **PostgreSQL** melalui ORM **SQLAlchemy**. Aplikasi ini dirancang untuk membantu pengelola rumah kost dalam mengelola data pengguna, kamar, dan booking secara efisien. Pengguna dapat dengan mudah melakukan operasi seperti menambah, melihat, memperbarui, dan menghapus data melalui antarmuka pengguna yang sederhana dan intuitif.

Selain itu, aplikasi ini dilengkapi dengan fitur untuk mengekspor data ke format Excel menggunakan pustaka **xlsxwriter**, yang mempermudah pengelola dalam mendokumentasikan informasi atau membagikannya ke pihak lain. Pengelolaan data dilakukan dengan memanfaatkan model basis data relasional, yang memastikan data tersimpan dengan terstruktur dan terorganisir.

Namun, aplikasi ini masih memiliki ruang untuk pengembangan lebih lanjut, seperti meningkatkan validasi input untuk mencegah kesalahan data, serta menambahkan lapisan keamanan pada autentikasi pengguna untuk melindungi informasi sensitif. Secara keseluruhan, aplikasi ini memberikan solusi sederhana namun efektif untuk kebutuhan pengelolaan rumah kost.

## Perancangan Sistem

### a. UML (Class Diagram)



## b. Skema Database PostgreSQL

Nama Database : rumah\_kita

- Tabel pengguna :
  - o id\_pengguna : SERIAL PRIMARY KEY
  - o nama : VARCHAR (100) NOT NULL
  - o email : VARCHAR (100) UNIQUE NOT NULL
  - o password : VARCHAR (100)
  - o created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP)
  
- Tabel kamar :
  - o id\_kamar : SERIAL PRIMARY KEY
  - o nama\_kamar : VARCHAR (100) NOT NULL
  - o harga\_perbulan : NUMERIC (10, 2) NOT NULL
  - o created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP)
  
- Tabel booking :
  - o id\_booking : SERIAL PRIMARY KEY
  - o id\_pengguna : INT NOT NULL
  - o id\_kamar : INT NOT NULL
  - o tanggal\_mulai\_sewa : DATE NOT NULL
  - o FOREIGN KEY (id\_pengguna) REFERENCES pengguna(id\_pengguna) ON DELETE CASCADE
  - o FOREIGN KEY (id\_kamar) REFERENCES kamar(id\_kamar) ON DELETE CASCADE)

## Langkah Pertama

Langkah pertama yang harus kita lakukan adalah membuka aplikasi postgresQL yang sudah terinstall.

```
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]: postgres
Password for user postgres:

psql (17.2)
WARNING: Console code page (437) differs from Windows code page (1252)
8-bit characters might not work correctly. See psql reference
page "Notes for Windows users" for details.
Type "help" for help.

postgres=# |
```

Kemudian pada bagian server [localhost], database [postgres], port [5432] kita tekan tombol enter saja pada keyboard, pada bagian username [postgres] kita masukan **postgres** dan bagian password for user postgres kita masukan **admin**, kemudian tekan enter.

## Langkah Kedua

Kemudian kita membuat database nya, disini saya memilih tentang kos dengan nama rumah\_kita. Ketik perintah yaitu **CREATE DATABASE rumah\_kita;**

```
postgres=# \c rumah_kita
You are now connected to database "rumah_kita" as user "postgres".
```

Karna database nya sudah saya buat kita bisa langsung masuk kedalam file database nya dengan mengetik perintah **\c rumah\_kita** nanti kita akan otomatis masuk kedalam file database yang telah kita buat.

## Langkah Ketiga

Selanjutnya kita akan melakukan pembuatan tabel pada database. Disini saya akan membuat 3 tabel yaitu tabel pengguna, tabel kamar, dan tabel booking.

```
rumah_kita=# CREATE TABLE pengguna (
rumah_kita(# id_pengguna SERIAL PRIMARY KEY,
rumah_kita(# nama VARCHAR(100) NOT NULL,
rumah_kita(# email VARCHAR(100) UNIQUE NOT NULL,
rumah_kita(# password VARCHAR(100),
rumah_kita(# created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP);
CREATE TABLE
```

```
rumah_kita=# CREATE TABLE kamar (
rumah_kita(# id_kamar SERIAL PRIMARY KEY,
rumah_kita(# nama_kamar VARCHAR(100) NOT NULL,
rumah_kita(# harga_perbulan NUMERIC(10, 2) NOT NULL,
rumah_kita(# created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP);
CREATE TABLE
```

```
rumah_kita=# CREATE TABLE booking (
rumah_kita(# id_booking SERIAL PRIMARY KEY,
rumah_kita(# id_pengguna INT NOT NULL,
rumah_kita(# id_kamar INT NOT NULL,
rumah_kita(# tanggal_mulai_sewa DATE NOT NULL,
rumah_kita(# FOREIGN KEY (id_pengguna) REFERENCES pengguna(id_pengguna) ON DELETE CASCADE,
rumah_kita(# FOREIGN KEY (id_kamar) REFERENCES kamar(id_kamar) ON DELETE CASCADE);
CREATE TABLE
```

Setelah kita mengetikan kode diatas berarti kita sudah berhasil membuat ketiga tabel tersebut.

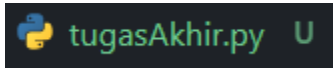
```
rumah_kita=# \dt
          List of relations
Schema | Name      | Type  | Owner
-----+-----+-----+-----
public | booking   | table | postgres
public | kamar     | table | postgres
public | pengguna  | table | postgres
(3 rows)
```

Sekarang kita melakukan pengecekan tabel dengan mengetikan perintah **\dt** kemudian hasil tabel yang telah kita buat akan muncul seperti gambar diatas.

## Pengembangan Sistem

### Langkah Pertama (Koneksi Database)

Langkat pertama yang perlu kita lakukan untuk melakukan pengembangan sistem adalah melakukan koneksi ke database postgres nya. Disini saya menggunakan aplikasi Vscode.



Buat file nya terlebih dahulu disini saya memberikan nama **tugasAkhir.py**.

```
1  from kivy.app import App
2  from kivy.uix.boxlayout import BoxLayout
3  from kivy.uix.button import Button
4  from kivy.uix.label import Label
5  from kivy.uix.textinput import TextInput
6  from kivy.uix.scrollview import ScrollView
7  from kivy.uix.gridlayout import GridLayout
8  from kivy.uix.popup import Popup
9  from sqlalchemy import create_engine, Column, Integer, String, Numeric, Date, TIMESTAMP, ForeignKey, func
10 from sqlalchemy.ext.declarative import declarative_base
11 from sqlalchemy.orm import sessionmaker, relationship
12 import xlswriter
13 import psycpg2
14 import os
15 from datetime import date
16
17 Base = declarative_base()
```

Ketik semua kode diatas karna nanti kita membutuhkan itu semua agar program ini bisa berjalan dengan sempurna.

```
19  # Database connection
20  def connect_db():
21      return psycpg2.connect(
22          dbname="rumah_kita",
23          user="postgres",
24          password="admin",
25          host="localhost",
26          port="5432"
27      )
```

Ketik kode diatas untuk membuat koneksi ke database PostgreSQL menggunakan library psycpg2 yang sudah kita inputkan pada kode sebelumnya.

### Langkah Kedua (Definisi Model)

Selanjutnya kita mendefinisikan model menggunakan SQLAlchemy untuk merepresentasikan tabel dalam database PostgreSQL.

```

30 class Pengguna(Base):
31     __tablename__ = 'pengguna'
32
33     id_pengguna = Column(Integer, primary_key=True, autoincrement=True)
34     nama = Column(String(100), nullable=False)
35     email = Column(String(100), unique=True, nullable=False)
36     password = Column(String(100))
37     created_at = Column(TIMESTAMP, default=func.current_timestamp())

```

Kode diatas bertujuan untuk merepresentasikan data pengguna pada tabel pengguna, dan juga memiliki beberpa kolom seperti id\_pengguna, nama, email, password, dan created\_at.

```

39 class Kamar(Base):
40     __tablename__ = 'kamar'
41
42     id_kamar = Column(Integer, primary_key=True, autoincrement=True)
43     nama_kamar = Column(String(100), nullable=False)
44     harga_perbulan = Column(Numeric(10, 2), nullable=False)
45     created_at = Column(TIMESTAMP, default=func.current_timestamp())

```

Kode diatas bertujuan untuk merepresentasikan data kamar yang tersedia untuk disewa dari tabel kamar, dan juga memiliki beberapa kolom seperti id\_kamar, nama\_kamar, harga\_perbulan, dan created\_at.

```

47 class Booking(Base):
48     __tablename__ = 'booking'
49
50     id_booking = Column(Integer, primary_key=True, autoincrement=True)
51     id_pengguna = Column(Integer, ForeignKey('pengguna.id_pengguna', ondelete='CASCADE'), nullable=False)
52     id_kamar = Column(Integer, ForeignKey('kamar.id_kamar', ondelete='CASCADE'), nullable=False)
53     tanggal_mulai_sewa = Column(Date, nullable=False)

```

Kode diatas bertujuan untuk merepresentasikan data pemesanan kamar dari tabel booking, dan juga memiliki beberapa kolom seperti id\_booking, id\_pengguna, id\_kamar, tanggal\_mulai\_sewa.

### Langkah Ketiga (struktur Aplikasi dengan Kivy)

```

63 # Kivy App
64 class RumahKitaApp(App):
65     def build(self):
66         layout = BoxLayout(orientation='vertical')
67
68         # Input fields untuk Pengguna
69         self.nama_input = TextInput(hint_text='Nama', size_hint=(1, 0.1))
70         self.email_input = TextInput(hint_text='Email', size_hint=(1, 0.1))
71         self.password_input = TextInput(hint_text='Password', size_hint=(1, 0.1), password=True)
72         add_pengguna_button = Button(text='Tambah Pengguna', size_hint=(1, 0.1))
73         add_pengguna_button.bind(on_press=self.add_pengguna)
74
75         layout.add_widget(self.nama_input)
76         layout.add_widget(self.email_input)
77         layout.add_widget(self.password_input)
78         layout.add_widget(add_pengguna_button)

```

```

80     # Input fields untuk Kamar
81     self.nama_kamar_input = TextInput(hint_text='Nama Kamar', size_hint=(1, 0.1))
82     self.harga_perbulan_input = TextInput(hint_text='Harga Perbulan', size_hint=(1, 0.1))
83     add_kamar_button = Button(text='Tambah Kamar', size_hint=(1, 0.1))
84     add_kamar_button.bind(on_press=self.add_kamar)
85
86     layout.add_widget(self.nama_kamar_input)
87     layout.add_widget(self.harga_perbulan_input)
88     layout.add_widget(add_kamar_button)

```

```

90     # Input fields untuk Booking
91     self.id_pengguna_input = TextInput(hint_text='ID Pengguna', size_hint=(1, 0.1))
92     self.id_kamar_input = TextInput(hint_text='ID Kamar', size_hint=(1, 0.1))
93     add_booking_button = Button(text='Tambah Booking', size_hint=(1, 0.1))
94     add_booking_button.bind(on_press=self.add_booking)
95
96     layout.add_widget(self.id_pengguna_input)
97     layout.add_widget(self.id_kamar_input)
98     layout.add_widget(add_booking_button)

```

```

100    # Export button
101    export_button = Button(text='Export ke Excel', size_hint=(1, 0.1))
102    export_button.bind(on_press=self.export_to_excel)
103    layout.add_widget(export_button)
104
105    # View data button
106    view_data_button = Button(text='Lihat Semua Data', size_hint=(1, 0.1))
107    view_data_button.bind(on_press=self.view_data)
108    layout.add_widget(view_data_button)

```

```

110    # Delete data button
111    delete_data_button = Button(text='Hapus Data', size_hint=(1, 0.1))
112    delete_data_button.bind(on_press=self.delete_data)
113    layout.add_widget(delete_data_button)
114
115    # Update data button
116    update_data_button = Button(text='Update Data', size_hint=(1, 0.1))
117    update_data_button.bind(on_press=self.update_data)
118    layout.add_widget(update_data_button)
119
120    return layout

```

Kode diatas bertujuan untuk membuat struktur aplikasinya. Kivy Components menggunakan BoxLayout untuk menampilkan input field dan tombol secara vertikal kemudian database berbasis SQLAlchemy, data dikelola melalui tabel pengguna, kamar, dan booking.

## Langkah Keempat (Fitur Tambah Data dan Ekspor ke Excel)

Selanjutnya kita membuat fitur untuk menambah pengguna, kamar, booking, dan ekspor ke excel.

```
122     def add_pengguna(self, instance):
123         nama = self.nama_input.text
124         email = self.email_input.text
125         password = self.password_input.text
126
127         pengguna = Pengguna(nama=nama, email=email, password=password)
128         session.add(pengguna)
129         session.commit()
130
131         self.nama_input.text = ''
132         self.email_input.text = ''
133         self.password_input.text = ''
134
135         popup = Popup(title='Sukses', content=Label(text='Pengguna berhasil ditambahkan'), size_hint=(0.8, 0.4))
136         popup.open()
```

Kode diatas bertujuan untuk menambahkan pengguna(add\_pengguna) yang mengambil input nama, email, dan password dari pengguna kemudian menyimpan data ke tabel pengguna, dan juga dapat menampilkan popup konfirmasi.

```
138     def add_kamar(self, instance):
139         nama_kamar = self.nama_kamar_input.text
140         harga_perbulan = self.harga_perbulan_input.text
141
142         kamar = Kamar(nama_kamar=nama_kamar, harga_perbulan=harga_perbulan)
143         session.add(kamar)
144         session.commit()
145
146         self.nama_kamar_input.text = ''
147         self.harga_perbulan_input.text = ''
148
149         popup = Popup(title='Sukses', content=Label(text='Kamar berhasil ditambahkan'), size_hint=(0.8, 0.4))
150         popup.open()
```

Kode diatas bertujuan untuk menambahkan kamar(add\_kamar) yang mengambil input nama\_kamar dan harga\_perbulan kemudian menyimpan data ke tabel kamar, dan juga dapat menampilkan popup konfirmasi.

```
152     def add_booking(self, instance):
153         id_pengguna = self.id_pengguna_input.text
154         id_kamar = self.id_kamar_input.text
155         tanggal_mulai_sewa = date.today()
156
157         booking = Booking(id_pengguna=id_pengguna, id_kamar=id_kamar, tanggal_mulai_sewa=tanggal_mulai_sewa)
158         session.add(booking)
159         session.commit()
160
161         self.id_pengguna_input.text = ''
162         self.id_kamar_input.text = ''
163
164         popup = Popup(title='Sukses', content=Label(text='Booking berhasil ditambahkan'), size_hint=(0.8, 0.4))
165         popup.open()
```

Kode diatas bertujuan untuk menambah booking(add\_booking) yang mengambil input ID pengguna, ID kamar, dan otomatis mengambil tanggal saat ini kemudian menyimpan data ke tabel booking, dan juga dapat menampilkan popup konfirmasi.



```

167     def export_to_excel(self, instance):
168         file_path = os.path.join(os.getcwd(), 'rumah_kita.xlsx')
169
170         with xlsxwriter.Workbook(file_path) as workbook:
171             pengguna_sheet = workbook.add_worksheet('Pengguna')
172             kamar_sheet = workbook.add_worksheet('Kamar')
173             booking_sheet = workbook.add_worksheet('Booking')

```

```

175         # Export Pengguna
176         pengguna_sheet.write(0, 0, 'ID')
177         pengguna_sheet.write(0, 1, 'Nama')
178         pengguna_sheet.write(0, 2, 'Email')
179         pengguna_sheet.write(0, 3, 'Password')
180         pengguna_sheet.write(0, 4, 'Created At')
181
182         for i, pengguna in enumerate(session.query(Pengguna).all(), start=1):
183             pengguna_sheet.write(i, 0, pengguna.id_pengguna)
184             pengguna_sheet.write(i, 1, pengguna.nama)
185             pengguna_sheet.write(i, 2, pengguna.email)
186             pengguna_sheet.write(i, 3, pengguna.password)
187             pengguna_sheet.write(i, 4, str(pengguna.created_at))

```

```

189         # Export Kamar
190         kamar_sheet.write(0, 0, 'ID')
191         kamar_sheet.write(0, 1, 'Nama Kamar')
192         kamar_sheet.write(0, 2, 'Harga Perbulan')
193         kamar_sheet.write(0, 3, 'Created At')
194
195         for i, kamar in enumerate(session.query(Kamar).all(), start=1):
196             kamar_sheet.write(i, 0, kamar.id_kamar)
197             kamar_sheet.write(i, 1, kamar.nama_kamar)
198             kamar_sheet.write(i, 2, float(kamar.harga_perbulan))
199             kamar_sheet.write(i, 3, str(kamar.created_at))

```

```

201         # Export Booking
202         booking_sheet.write(0, 0, 'ID Booking')
203         booking_sheet.write(0, 1, 'ID Pengguna')
204         booking_sheet.write(0, 2, 'ID Kamar')
205         booking_sheet.write(0, 3, 'Tanggal Mulai Sewa')
206
207         for i, booking in enumerate(session.query(Booking).all(), start=1):
208             booking_sheet.write(i, 0, booking.id_booking)
209             booking_sheet.write(i, 1, booking.id_pengguna)
210             booking_sheet.write(i, 2, booking.id_kamar)
211             booking_sheet.write(i, 3, str(booking.tanggal_mulai_sewa))
212
213         popup = Popup(title='Sukses',
214                       content=Label(text=f'Data di ekspor ke {file_path}'),
215                       size_hint=(0.8, 0.4))
216         popup.open()

```

Kode diatas bertujuan untuk melakukan export ke excel(`export_to_excel`) yang menyimpan data dari ketiga tabel ke file excel yaitu sheet pengguna, kamar dan booking menggunakan library `xlsxwriter`. Setiap sheet memiliki kolom sesuai dengan atribut tabel di database, dan juga dapat menampilkan popup dengan lokasi file yang diekspor.

## Langkah Kelima (Membuat Fitur Menampilkan Data)

Selanjutnya kita membuat tampilan antarmuka berbasis kivy yang menampilkan data dari tiga tabel database **Pengguna**, **kamar**, dan **Booking**.

```
218     def view_data(self, instance):
219         layout = GridLayout(cols=1, padding=10, spacing=10, size_hint_y=None)
220         layout.bind(minimum_height=layout.setter('height'))
```

Kode diatas bertujuan untuk membuat layout utama. Layout utama adalah **GridLayout** dengan satu kolom yang mengatur komponen secara vertikal. Layout ini bersifat dinamis, menyesuaikan tinggi kontennya.

```
222     # Pengguna data
223     pengguna_label = Label(text='Pengguna Data', size_hint_y=None, height=40)
224     layout.add_widget(pengguna_label)
225     pengguna_data = GridLayout(cols=5, size_hint_y=None)
226     pengguna_data.bind(minimum_height=pengguna_data.setter('height'))
227     pengguna_data.add_widget(Label(text='ID'))
228     pengguna_data.add_widget(Label(text='Nama'))
229     pengguna_data.add_widget(Label(text='Email'))
230     pengguna_data.add_widget(Label(text='Password'))
231     pengguna_data.add_widget(Label(text='Created At'))
232
233     for pengguna in session.query(Pengguna).all():
234         pengguna_data.add_widget(Label(text=str(pengguna.id_pengguna)))
235         pengguna_data.add_widget(Label(text=pengguna.nama))
236         pengguna_data.add_widget(Label(text=pengguna.email))
237         pengguna_data.add_widget(Label(text=pengguna.password))
238         pengguna_data.add_widget(Label(text=str(pengguna.created_at)))
239
240     layout.add_widget(pengguna_data)
```

Kode diatas bertujuan untuk menampilkan data pengguna yang membuat grid dengan 5 kolom untuk header (ID, Nama, Email, Password, dan Created At), kemudian mengambil semua data dari tabel **Pengguna** menggunakan SQLAlchemy dan menampilkan setiap kolom sebagai label pada grid.

```
242     # Kamar data
243     kamar_label = Label(text='Kamar Data', size_hint_y=None, height=40)
244     layout.add_widget(kamar_label)
245     kamar_data = GridLayout(cols=4, size_hint_y=None)
246     kamar_data.bind(minimum_height=kamar_data.setter('height'))
247     kamar_data.add_widget(Label(text='ID'))
248     kamar_data.add_widget(Label(text='Nama Kamar'))
249     kamar_data.add_widget(Label(text='Harga Perbulan'))
250     kamar_data.add_widget(Label(text='Created At'))
251
252     for kamar in session.query(Kamar).all():
253         kamar_data.add_widget(Label(text=str(kamar.id_kamar)))
254         kamar_data.add_widget(Label(text=kamar.nama_kamar))
255         kamar_data.add_widget(Label(text=str(kamar.harga_perbulan)))
256         kamar_data.add_widget(Label(text=str(kamar.created_at)))
257
258     layout.add_widget(kamar_data)
```

Kode diatas bertujuan untuk menampilkan data kamar yang membuat grid dengan 4 kolom untuk header (ID, Nama Kamar, Harga Perbulan, Created At), kemudian mengambil semua data dari tabel **Kamar** dan menambahkan ke grid.

```

260 # Booking data
261 booking_label = Label(text='Booking Data', size_hint_y=None, height=40)
262 layout.add_widget(booking_label)
263 booking_data = GridLayout(cols=4, size_hint_y=None)
264 booking_data.bind(minimum_height=booking_data.setter('height'))
265 booking_data.add_widget(Label(text='ID Booking'))
266 booking_data.add_widget(Label(text='ID Pengguna'))
267 booking_data.add_widget(Label(text='ID Kamar'))
268 booking_data.add_widget(Label(text='Tanggal Mulai Sewa'))
269
270 for booking in session.query(Booking).all():
271     booking_data.add_widget(Label(text=str(booking.id_booking)))
272     booking_data.add_widget(Label(text=str(booking.id_pengguna)))
273     booking_data.add_widget(Label(text=str(booking.id_kamar)))
274     booking_data.add_widget(Label(text=str(booking.tanggal_mulai_sewa)))
275
276 layout.add_widget(booking_data)

```

Kode diatas bertujuan untuk menampilkan data booking yang membuat grid dengan 4 kolom untuk header (ID Booking, ID Pengguna, ID Kamar, Tanggal Mulai Sewa), kemudian mengambil semua data dari tabel **Booking** dan menambahkan ke grid.

```

278 scroll_view = ScrollView(size_hint=(1, None), size=(self.root.width, self.root.height))
279 scroll_view.add_widget(layout)
280
281 popup = Popup(title='Lihat Data', content=scroll_view, size_hint=(0.9, 0.9))
282 popup.open()

```

Kode diatas bertujuan untuk menambahkan ScrollView dan menampilkan popup.

## Langkah Keenam (Membuat Fitur Hapus Data)

Selanjutnya kita membuat fitur hapus data dari database menggunakan antarmuka kivy.

```

284 # delete data
285 def delete_data(self, instance):
286     layout = BoxLayout(orientation='vertical', padding=10, spacing=10)
287
288     self.delete_id_input = TextInput(hint_text='Masukan ID yang akan di hapus', size_hint=(1, 0.1))
289     self.delete_table_input = TextInput(hint_text='Tabel (pengguna/kamar/booking)', size_hint=(1, 0.1))
290     delete_button = Button(text='Hapus', size_hint=(1, 0.1))
291     delete_button.bind(on_press=self.delete_record)
292
293     layout.add_widget(self.delete_id_input)
294     layout.add_widget(self.delete_table_input)
295     layout.add_widget(delete_button)
296
297     popup = Popup(title='Hapus Data', content=layout, size_hint=(0.8, 0.4))
298     popup.open()

```

Kode diatas bertujuan untuk membuat antarmuka berupa popup dengan dua kolom input satu untuk memasukkan ID data yang ingin dihapus dan satu lagi untuk menentukan nama tabel (opsi: pengguna, kamar, atau booking). Juga terdapat tombol "Hapus" yang terhubung dengan fungsi **delete\_record**.

```

300 def delete_record(self, instance):
301     record_id = self.delete_id_input.text
302     table_name = self.delete_table_input.text
303
304     if table_name == 'pengguna':
305         record = session.query(Pengguna).filter_by(id_pengguna=record_id).first()
306     elif table_name == 'kamar':
307         record = session.query(Kamar).filter_by(id_kamar=record_id).first()
308     elif table_name == 'booking':
309         record = session.query(Booking).filter_by(id_booking=record_id).first()
310     else:
311         popup = Popup(title='Error', content=Label(text='Nama tabel tidak valid'), size_hint=(0.8, 0.4))
312         popup.open()
313         return
314
315     if record:
316         session.delete(record)
317         session.commit()
318         popup = Popup(title='Sukses', content=Label(text='Berhasil dihapus'), size_hint=(0.8, 0.4))
319     else:
320         popup = Popup(title='Error', content=Label(text='data tidak ditemukan'), size_hint=(0.8, 0.4))
321
322     popup.open()

```

Kode diatas berfungsi untuk mengambil input pengguna, kamar, dan booking, memvalidasi nama tabel, dan mencari data berdasarkan ID menggunakan SQLAlchemy. Jika data ditemukan, data dihapus dari tabel yang sesuai, dan perubahan disimpan ke database. Umpan balik berupa popup diberikan untuk menampilkan status penghapusan (sukses atau gagal).

## Langkah ketujuh (Membuat Fitur Update Data)

Dan yang terakhir kita membuat fitur update data dari database menggunakan antarmuka kivy.

```

324 # update data
325 def update_data(self, instance):
326     layout = BoxLayout(orientation='vertical', padding=10, spacing=10)
327
328     self.update_id_input = TextInput(hint_text='Masukan ID yang akan di update', size_hint=(1, 0.1))
329     self.update_table_input = TextInput(hint_text='Tabel (pengguna/kamar/booking)', size_hint=(1, 0.1))
330     self.update_field_input = TextInput(hint_text='masukan nama kolom yang akan di update', size_hint=(1, 0.1))
331     self.update_value_input = TextInput(hint_text='masukan nama baru untuk kolom yang akan di update', size_hint=(1, 0.1))
332     update_button = Button(text='Update', size_hint=(1, 0.1))
333     update_button.bind(on_press=self.update_record)
334
335     layout.add_widget(self.update_id_input)
336     layout.add_widget(self.update_table_input)
337     layout.add_widget(self.update_field_input)
338     layout.add_widget(self.update_value_input)
339     layout.add_widget(update_button)
340
341     popup = Popup(title='Update Data', content=layout, size_hint=(0.8, 0.4))
342     popup.open()

```

```

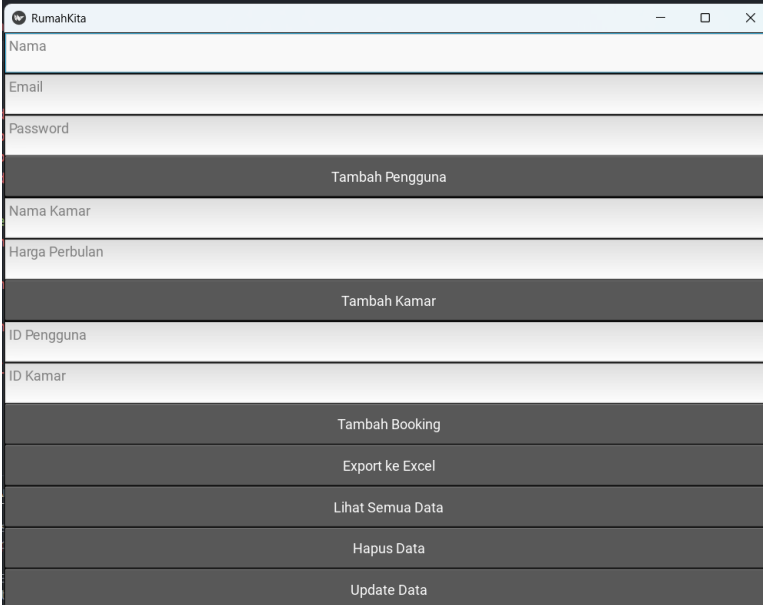
344 def update_record(self, instance):
345     record_id = self.update_id_input.text
346     table_name = self.update_table_input.text
347     field_name = self.update_field_input.text
348     new_value = self.update_value_input.text
349
350     if table_name == 'pengguna':
351         record = session.query(Pengguna).filter_by(id_pengguna=record_id).first()
352     elif table_name == 'kamar':
353         record = session.query(Kamar).filter_by(id_kamar=record_id).first()
354     elif table_name == 'booking':
355         record = session.query(Booking).filter_by(id_booking=record_id).first()
356     else:
357         popup = Popup(title='Error', content=Label(text='Nama tabel tidak valid'), size_hint=(0.8, 0.4))
358         popup.open()
359         return
360
361     if record:
362         setattr(record, field_name, new_value)
363         session.commit()
364         popup = Popup(title='Sukses', content=Label(text='Update Berhasil'), size_hint=(0.8, 0.4))
365     else:
366         popup = Popup(title='Error', content=Label(text='Data tidak ditemukan'), size_hint=(0.8, 0.4))
367
368     popup.open()

```

Kode diatas bertujuan untuk memperbarui data dalam aplikasi berbasis Kivy dengan mengakses dan mengubah data di dalam tabel database. Ketika dipanggil, fungsi `update_data` akan menampilkan form input untuk memasukkan ID data yang ingin diperbarui, nama tabel yang relevan (pengguna, kamar, atau booking), nama kolom yang ingin diubah, dan nilai baru untuk kolom tersebut. Setelah form diisi dan tombol "Update" ditekan, fungsi `update_record` akan mencari data di tabel yang dipilih berdasarkan ID, kemudian memperbarui nilai kolom yang diminta. Jika data ditemukan, perubahan disimpan ke database, dan popup akan menampilkan pesan sukses. Jika tabel atau data tidak valid, popup error akan ditampilkan.

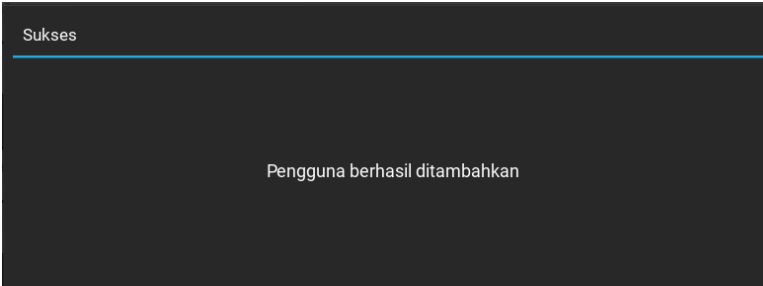
## Pengujian Aplikasi

### - Tampilan Awal



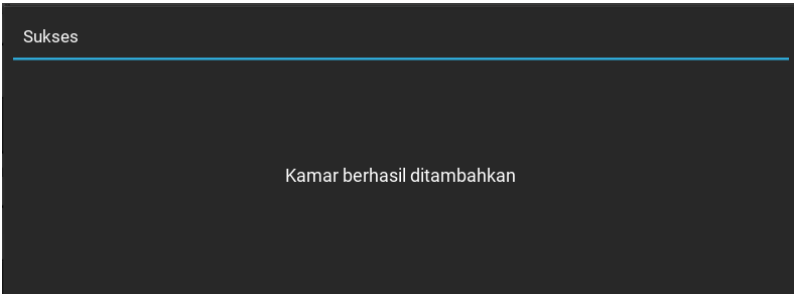
Nama
Email
Password
Tambah Pengguna
Nama Kamar
Harga Perbulan
Tambah Kamar
ID Pengguna
ID Kamar
Tambah Booking
Export ke Excel
Lihat Semua Data
Hapus Data
Update Data

### - Tambah Pengguna

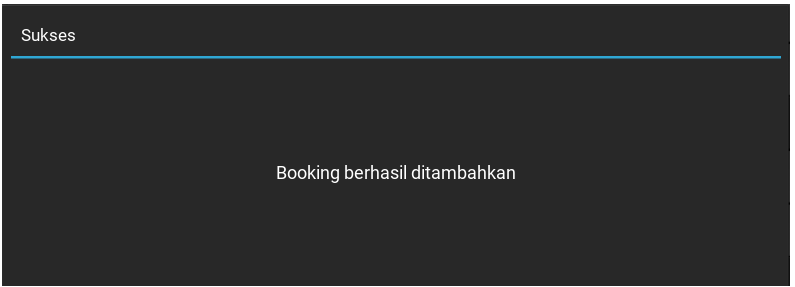


Sukses
Pengguna berhasil ditambahkan

- Tambah Kamar



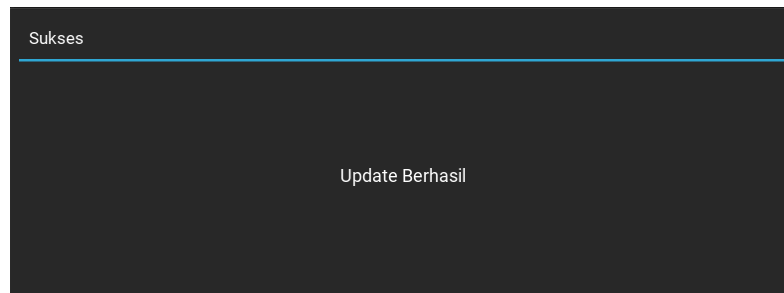
- Tambah Booking



- Lihat Semua Data

Lihat Data				
Pegguna Data				
ID	Nama	Email	Password	Created At
4	Dery Wahyu Perdana	dery@gmail.com	12345	2025-01-04 23:38:08.9320
Kamar Data				
ID	Nama Kamar	Harga Perbulan		Created At
5	Kamar Mawar	500000.00		2025-01-04 23:40:00.326632
Booking Data				
ID Booking	ID Pengguna	ID Kamar	Tanggal Mulai Sewa	
4	4	5	2025-01-04	

- Update Data Tabel Pengguna



- Lihat Hasil Setelah di Update

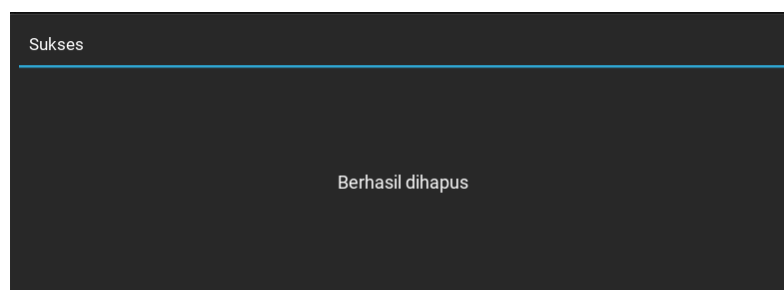
Lihat Data

Pengguna Data				
ID	Nama	Email	Password	Created At
4	Dery	dery@gmail.com	12345	2025-01-04 23:38:08.9320

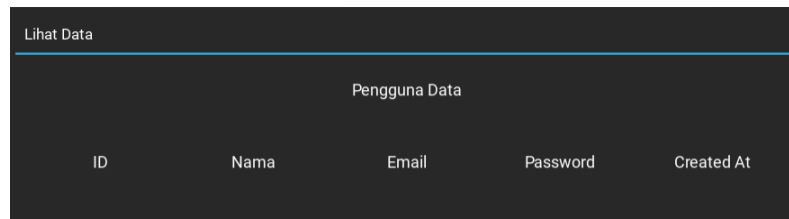
Kamar Data			
ID	Nama Kamar	Harga Perbulan	Created At
5	Kamar Mawar	500000.00	2025-01-04 23:40:00.326632

Booking Data			
ID Booking	ID Pengguna	ID Kamar	Tanggal Mulai Sewa
4	4	5	2025-01-04

- Hapus Data

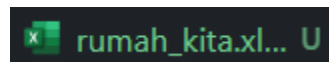
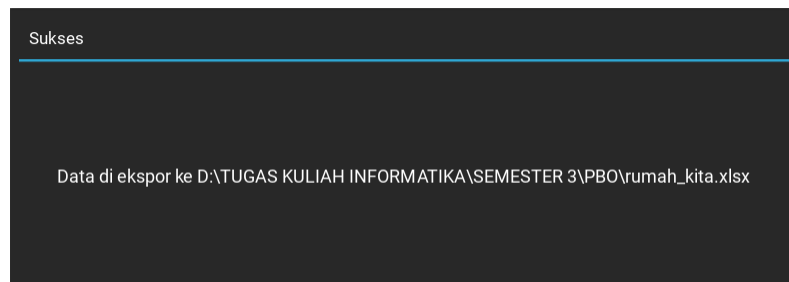


- Tampilan Setelah di Hapus



Pengguna Data				
ID	Nama	Email	Password	Created At

- Ekspor ke Excel



## Kesimpulan

Aplikasi yang dibuat adalah sistem manajemen rumah sewa yang menggunakan framework Kivy untuk antarmuka pengguna dan SQLAlchemy untuk interaksi dengan database PostgreSQL. Aplikasi ini memungkinkan pengguna untuk menambah, mengubah, melihat, dan menghapus data terkait pengguna, kamar, dan booking. Pengguna dapat menginputkan data baru untuk pengguna, kamar, dan booking, serta memperbarui data yang ada dengan memilih kolom yang ingin diubah. Fitur lain termasuk ekspor data ke file Excel untuk backup atau analisis lebih lanjut. Aplikasi ini memiliki tampilan grafis yang sederhana dengan popup untuk memberikan umpan balik kepada pengguna, serta menggunakan database PostgreSQL untuk penyimpanan data yang terstruktur.