**Dokumentasi syntax SQL**

Pembuatan Database : CREATE DATABASE rumah\_kita;

Pembuatan Tabel :

* Pengguna

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);

* Kamar

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);

* Booking

rumah\_kita=# CREATE TABLE booking (

rumah\_kita(# id\_boking 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);

**Source Code Pemrograman dengan Python**

from kivy.app import App

from kivy.uix.boxlayout import BoxLayout

from kivy.uix.button import Button

from kivy.uix.label import Label

from kivy.uix.textinput import TextInput

from kivy.uix.scrollview import ScrollView

from kivy.uix.gridlayout import GridLayout

from kivy.uix.popup import Popup

from sqlalchemy import create\_engine, Column, Integer, String, Numeric, Date, TIMESTAMP, ForeignKey, func

from sqlalchemy.ext.declarative import declarative\_base

from sqlalchemy.orm import sessionmaker, relationship

import xlsxwriter

import psycopg2

import os

from datetime import date

Base = declarative\_base()

# Database connection

def connect\_db():

    return psycopg2.connect(

        dbname="rumah\_kita",

        user="postgres",

        password="admin",

        host="localhost",

        port="5432"

    )

# Define models

class Pengguna(Base):

    \_\_tablename\_\_ = 'pengguna'

    id\_pengguna = Column(Integer, primary\_key=True, autoincrement=True)

    nama = Column(String(100), nullable=False)

    email = Column(String(100), unique=True, nullable=False)

    password = Column(String(100))

    created\_at = Column(TIMESTAMP, default=func.current\_timestamp())

class Kamar(Base):

    \_\_tablename\_\_ = 'kamar'

    id\_kamar = Column(Integer, primary\_key=True, autoincrement=True)

    nama\_kamar = Column(String(100), nullable=False)

    harga\_perbulan = Column(Numeric(10, 2), nullable=False)

    created\_at = Column(TIMESTAMP, default=func.current\_timestamp())

class Booking(Base):

    \_\_tablename\_\_ = 'booking'

    id\_booking = Column(Integer, primary\_key=True, autoincrement=True)

    id\_pengguna = Column(Integer, ForeignKey('pengguna.id\_pengguna', ondelete='CASCADE'), nullable=False)

    id\_kamar = Column(Integer, ForeignKey('kamar.id\_kamar', ondelete='CASCADE'), nullable=False)

    tanggal\_mulai\_sewa = Column(Date, nullable=False)

# Database setup

connection = connect\_db()

database\_url = 'postgresql+psycopg2://postgres:admin@localhost:5432/rumah\_kita'

engine = create\_engine(database\_url)

Base.metadata.create\_all(engine)

Session = sessionmaker(bind=engine)

session = Session()

# Kivy App

class RumahKitaApp(App):

    def build(self):

        layout = BoxLayout(orientation='vertical')

        # Input fields untuk Pengguna

        self.nama\_input = TextInput(hint\_text='Nama', size\_hint=(1, 0.1))

        self.email\_input = TextInput(hint\_text='Email', size\_hint=(1, 0.1))

        self.password\_input = TextInput(hint\_text='Password', size\_hint=(1, 0.1), password=True)

        add\_pengguna\_button = Button(text='Tambah Pengguna', size\_hint=(1, 0.1))

        add\_pengguna\_button.bind(on\_press=self.add\_pengguna)

        layout.add\_widget(self.nama\_input)

        layout.add\_widget(self.email\_input)

        layout.add\_widget(self.password\_input)

        layout.add\_widget(add\_pengguna\_button)

        # Input fields untuk Kamar

        self.nama\_kamar\_input = TextInput(hint\_text='Nama Kamar', size\_hint=(1, 0.1))

        self.harga\_perbulan\_input = TextInput(hint\_text='Harga Perbulan', size\_hint=(1, 0.1))

        add\_kamar\_button = Button(text='Tambah Kamar', size\_hint=(1, 0.1))

        add\_kamar\_button.bind(on\_press=self.add\_kamar)

        layout.add\_widget(self.nama\_kamar\_input)

        layout.add\_widget(self.harga\_perbulan\_input)

        layout.add\_widget(add\_kamar\_button)

        # Input fields untuk Booking

        self.id\_pengguna\_input = TextInput(hint\_text='ID Pengguna', size\_hint=(1, 0.1))

        self.id\_kamar\_input = TextInput(hint\_text='ID Kamar', size\_hint=(1, 0.1))

        add\_booking\_button = Button(text='Tambah Booking', size\_hint=(1, 0.1))

        add\_booking\_button.bind(on\_press=self.add\_booking)

        layout.add\_widget(self.id\_pengguna\_input)

        layout.add\_widget(self.id\_kamar\_input)

        layout.add\_widget(add\_booking\_button)

        # Export button

        export\_button = Button(text='Export ke Excel', size\_hint=(1, 0.1))

        export\_button.bind(on\_press=self.export\_to\_excel)

        layout.add\_widget(export\_button)

        # View data button

        view\_data\_button = Button(text='Lihat Semua Data', size\_hint=(1, 0.1))

        view\_data\_button.bind(on\_press=self.view\_data)

        layout.add\_widget(view\_data\_button)

        # Delete data button

        delete\_data\_button = Button(text='Hapus Data', size\_hint=(1, 0.1))

        delete\_data\_button.bind(on\_press=self.delete\_data)

        layout.add\_widget(delete\_data\_button)

        # Update data button

        update\_data\_button = Button(text='Update Data', size\_hint=(1, 0.1))

        update\_data\_button.bind(on\_press=self.update\_data)

        layout.add\_widget(update\_data\_button)

        return layout

    def add\_pengguna(self, instance):

        nama = self.nama\_input.text

        email = self.email\_input.text

        password = self.password\_input.text

        pengguna = Pengguna(nama=nama, email=email, password=password)

        session.add(pengguna)

        session.commit()

        self.nama\_input.text = ''

        self.email\_input.text = ''

        self.password\_input.text = ''

        popup = Popup(title='Sukses', content=Label(text='Pengguna berhasil ditambahkan'), size\_hint=(0.8, 0.4))

        popup.open()

    def add\_kamar(self, instance):

        nama\_kamar = self.nama\_kamar\_input.text

        harga\_perbulan = self.harga\_perbulan\_input.text

        kamar = Kamar(nama\_kamar=nama\_kamar, harga\_perbulan=harga\_perbulan)

        session.add(kamar)

        session.commit()

        self.nama\_kamar\_input.text = ''

        self.harga\_perbulan\_input.text = ''

        popup = Popup(title='Sukses', content=Label(text='Kamar berhasil ditambahkan'), size\_hint=(0.8, 0.4))

        popup.open()

    def add\_booking(self, instance):

        id\_pengguna = self.id\_pengguna\_input.text

        id\_kamar = self.id\_kamar\_input.text

        tanggal\_mulai\_sewa = date.today()

        booking = Booking(id\_pengguna=id\_pengguna, id\_kamar=id\_kamar, tanggal\_mulai\_sewa=tanggal\_mulai\_sewa)

        session.add(booking)

        session.commit()

        self.id\_pengguna\_input.text = ''

        self.id\_kamar\_input.text = ''

        popup = Popup(title='Sukses', content=Label(text='Booking berhasil ditambahkan'), size\_hint=(0.8, 0.4))

        popup.open()

    def export\_to\_excel(self, instance):

        file\_path = os.path.join(os.getcwd(), 'rumah\_kita.xlsx')

        with xlsxwriter.Workbook(file\_path) as workbook:

            pengguna\_sheet = workbook.add\_worksheet('Pengguna')

            kamar\_sheet = workbook.add\_worksheet('Kamar')

            booking\_sheet = workbook.add\_worksheet('Booking')

            # Export Pengguna

            pengguna\_sheet.write(0, 0, 'ID')

            pengguna\_sheet.write(0, 1, 'Nama')

            pengguna\_sheet.write(0, 2, 'Email')

            pengguna\_sheet.write(0, 3, 'Password')

            pengguna\_sheet.write(0, 4, 'Created At')

            for i, pengguna in enumerate(session.query(Pengguna).all(), start=1):

                pengguna\_sheet.write(i, 0, pengguna.id\_pengguna)

                pengguna\_sheet.write(i, 1, pengguna.nama)

                pengguna\_sheet.write(i, 2, pengguna.email)

                pengguna\_sheet.write(i, 3, pengguna.password)

                pengguna\_sheet.write(i, 4, str(pengguna.created\_at))

            # Export Kamar

            kamar\_sheet.write(0, 0, 'ID')

            kamar\_sheet.write(0, 1, 'Nama Kamar')

            kamar\_sheet.write(0, 2, 'Harga Perbulan')

            kamar\_sheet.write(0, 3, 'Created At')

            for i, kamar in enumerate(session.query(Kamar).all(), start=1):

                kamar\_sheet.write(i, 0, kamar.id\_kamar)

                kamar\_sheet.write(i, 1, kamar.nama\_kamar)

                kamar\_sheet.write(i, 2, float(kamar.harga\_perbulan))

                kamar\_sheet.write(i, 3, str(kamar.created\_at))

            # Export Booking

            booking\_sheet.write(0, 0, 'ID Booking')

            booking\_sheet.write(0, 1, 'ID Pengguna')

            booking\_sheet.write(0, 2, 'ID Kamar')

            booking\_sheet.write(0, 3, 'Tanggal Mulai Sewa')

            for i, booking in enumerate(session.query(Booking).all(), start=1):

                booking\_sheet.write(i, 0, booking.id\_booking)

                booking\_sheet.write(i, 1, booking.id\_pengguna)

                booking\_sheet.write(i, 2, booking.id\_kamar)

                booking\_sheet.write(i, 3, str(booking.tanggal\_mulai\_sewa))

        popup = Popup(title='Sukses',

                      content=Label(text=f'Data di ekspor ke {file\_path}'),

                      size\_hint=(0.8, 0.4))

        popup.open()

    def view\_data(self, instance):

        layout = GridLayout(cols=1, padding=10, spacing=10, size\_hint\_y=None)

        layout.bind(minimum\_height=layout.setter('height'))

        # Pengguna data

        pengguna\_label = Label(text='Pengguna Data', size\_hint\_y=None, height=40)

        layout.add\_widget(pengguna\_label)

        pengguna\_data = GridLayout(cols=5, size\_hint\_y=None)

        pengguna\_data.bind(minimum\_height=pengguna\_data.setter('height'))

        pengguna\_data.add\_widget(Label(text='ID'))

        pengguna\_data.add\_widget(Label(text='Nama'))

        pengguna\_data.add\_widget(Label(text='Email'))

        pengguna\_data.add\_widget(Label(text='Password'))

        pengguna\_data.add\_widget(Label(text='Created At'))

        for pengguna in session.query(Pengguna).all():

            pengguna\_data.add\_widget(Label(text=str(pengguna.id\_pengguna)))

            pengguna\_data.add\_widget(Label(text=pengguna.nama))

            pengguna\_data.add\_widget(Label(text=pengguna.email))

            pengguna\_data.add\_widget(Label(text=pengguna.password))

            pengguna\_data.add\_widget(Label(text=str(pengguna.created\_at)))

        layout.add\_widget(pengguna\_data)

        # Kamar data

        kamar\_label = Label(text='Kamar Data', size\_hint\_y=None, height=40)

        layout.add\_widget(kamar\_label)

        kamar\_data = GridLayout(cols=4, size\_hint\_y=None)

        kamar\_data.bind(minimum\_height=kamar\_data.setter('height'))

        kamar\_data.add\_widget(Label(text='ID'))

        kamar\_data.add\_widget(Label(text='Nama Kamar'))

        kamar\_data.add\_widget(Label(text='Harga Perbulan'))

        kamar\_data.add\_widget(Label(text='Created At'))

        for kamar in session.query(Kamar).all():

            kamar\_data.add\_widget(Label(text=str(kamar.id\_kamar)))

            kamar\_data.add\_widget(Label(text=kamar.nama\_kamar))

            kamar\_data.add\_widget(Label(text=str(kamar.harga\_perbulan)))

            kamar\_data.add\_widget(Label(text=str(kamar.created\_at)))

        layout.add\_widget(kamar\_data)

        # Booking data

        booking\_label = Label(text='Booking Data', size\_hint\_y=None, height=40)

        layout.add\_widget(booking\_label)

        booking\_data = GridLayout(cols=4, size\_hint\_y=None)

        booking\_data.bind(minimum\_height=booking\_data.setter('height'))

        booking\_data.add\_widget(Label(text='ID Booking'))

        booking\_data.add\_widget(Label(text='ID Pengguna'))

        booking\_data.add\_widget(Label(text='ID Kamar'))

        booking\_data.add\_widget(Label(text='Tanggal Mulai Sewa'))

        for booking in session.query(Booking).all():

            booking\_data.add\_widget(Label(text=str(booking.id\_booking)))

            booking\_data.add\_widget(Label(text=str(booking.id\_pengguna)))

            booking\_data.add\_widget(Label(text=str(booking.id\_kamar)))

            booking\_data.add\_widget(Label(text=str(booking.tanggal\_mulai\_sewa)))

        layout.add\_widget(booking\_data)

        scroll\_view = ScrollView(size\_hint=(1, None), size=(self.root.width, self.root.height))

        scroll\_view.add\_widget(layout)

        popup = Popup(title='Lihat Data', content=scroll\_view, size\_hint=(0.9, 0.9))

        popup.open()

    # delete data

    def delete\_data(self, instance):

        layout = BoxLayout(orientation='vertical', padding=10, spacing=10)

        self.delete\_id\_input = TextInput(hint\_text='Masukan ID yang akan di hapus', size\_hint=(1, 0.1))

        self.delete\_table\_input = TextInput(hint\_text='Tabel (pengguna/kamar/booking)', size\_hint=(1, 0.1))

        delete\_button = Button(text='Hapus', size\_hint=(1, 0.1))

        delete\_button.bind(on\_press=self.delete\_record)

        layout.add\_widget(self.delete\_id\_input)

        layout.add\_widget(self.delete\_table\_input)

        layout.add\_widget(delete\_button)

        popup = Popup(title='Hapus Data', content=layout, size\_hint=(0.8, 0.4))

        popup.open()

    def delete\_record(self, instance):

        record\_id = self.delete\_id\_input.text

        table\_name = self.delete\_table\_input.text

        if table\_name == 'pengguna':

            record = session.query(Pengguna).filter\_by(id\_pengguna=record\_id).first()

        elif table\_name == 'kamar':

            record = session.query(Kamar).filter\_by(id\_kamar=record\_id).first()

        elif table\_name == 'booking':

            record = session.query(Booking).filter\_by(id\_booking=record\_id).first()

        else:

            popup = Popup(title='Error', content=Label(text='Nama tabel tidak valid'), size\_hint=(0.8, 0.4))

            popup.open()

            return

        if record:

            session.delete(record)

            session.commit()

            popup = Popup(title='Sukses', content=Label(text='Berhasil dihapus'), size\_hint=(0.8, 0.4))

        else:

            popup = Popup(title='Error', content=Label(text='data tidak ditemukan'), size\_hint=(0.8, 0.4))

        popup.open()

    # update data

    def update\_data(self, instance):

        layout = BoxLayout(orientation='vertical', padding=10, spacing=10)

        self.update\_id\_input = TextInput(hint\_text='Masukan ID yang akan di update', size\_hint=(1, 0.1))

        self.update\_table\_input = TextInput(hint\_text='Tabel (pengguna/kamar/booking)', size\_hint=(1, 0.1))

        self.update\_field\_input = TextInput(hint\_text='masukan nama kolom yang akan di update', size\_hint=(1, 0.1))

        self.update\_value\_input = TextInput(hint\_text='masukan nama baru untuk kolom yang akan di update', size\_hint=(1, 0.1))

        update\_button = Button(text='Update', size\_hint=(1, 0.1))

        update\_button.bind(on\_press=self.update\_record)

        layout.add\_widget(self.update\_id\_input)

        layout.add\_widget(self.update\_table\_input)

        layout.add\_widget(self.update\_field\_input)

        layout.add\_widget(self.update\_value\_input)

        layout.add\_widget(update\_button)

        popup = Popup(title='Update Data', content=layout, size\_hint=(0.8, 0.4))

        popup.open()

    def update\_record(self, instance):

        record\_id = self.update\_id\_input.text

        table\_name = self.update\_table\_input.text

        field\_name = self.update\_field\_input.text

        new\_value = self.update\_value\_input.text

        if table\_name == 'pengguna':

            record = session.query(Pengguna).filter\_by(id\_pengguna=record\_id).first()

        elif table\_name == 'kamar':

            record = session.query(Kamar).filter\_by(id\_kamar=record\_id).first()

        elif table\_name == 'booking':

            record = session.query(Booking).filter\_by(id\_booking=record\_id).first()

        else:

            popup = Popup(title='Error', content=Label(text='Nama tabel tidak valid'), size\_hint=(0.8, 0.4))

            popup.open()

            return

        if record:

            setattr(record, field\_name, new\_value)

            session.commit()

            popup = Popup(title='Sukses', content=Label(text='Update Berhasil'), size\_hint=(0.8, 0.4))

        else:

            popup = Popup(title='Error', content=Label(text='Data tidak ditemukan'), size\_hint=(0.8, 0.4))

        popup.open()

if \_\_name\_\_ == '\_\_main\_\_':

    RumahKitaApp().run()