Tutorial CRUD Produk dengan Kivy dan Firebase

Daftar Isi

1. Instalasi Dependencies

Install packages yang diperlukan menggunakan pip:

```
pip install kivy
pip install pyrebase4
```

2. Struktur Project

Buat folder project dan buat file-file berikut:

```
project_folder/
    ├── main.py
    ├── views.py
    ├── database.py
    ├── config.py
    ├── product.kv
    └── .gitignore
```

3. Implementasi Kode

3.1 Konfigurasi Firebase (config.py)

```python
def get_firebase_config():
    config = {
        "apiKey": "your-api-key",
        "authDomain": "your-app-id.firebaseapp.com",
        "databaseURL": "https://your-app-id.firebaseio.com",
        "storageBucket": "your-app-id.appspot.com",
    }
    return config
```

3.2 Database Handler (database.py)

```
import pyrebase
from config import get_firebase_config

class Database:
    config = get_firebase_config()
    firebase = pyrebase.initialize_app(config)
    db = firebase.database()

    @staticmethod
    def get_all_products():
        try:
            products = Database.db.child("products").get()
            if products.each():
                return [(product.key(), product.val()) for product in
products.each()]
            return []
        except Exception as e:
            print(f"Error getting products: {e}")
            return []

    @staticmethod
    def add_product(product_data):
        try:
            return Database.db.child("products").push(product_data)
        except Exception as e:
            print(f"Error adding product: {e}")
            raise e

    @staticmethod
    def update_product(product_id, product_data):
        try:
            return
Database.db.child("products").child(product_id).update(product_data)
        except Exception as e:
            print(f"Error updating product: {e}")
            raise e

    @staticmethod
    def delete_product(product_id):
        try:
            return
Database.db.child("products").child(product_id).remove()
        except Exception as e:
            print(f"Error deleting product: {e}")
            raise e
```

3.3 Views (views.py)

```
from kivy.uix.screenmanager import Screen
from kivy.properties import ObjectProperty, StringProperty
from kivy.uix.popup import Popup
from kivy.uix.label import Label
from kivy.uix.button import Button
from kivy.uix.boxlayout import BoxLayout
from database import Database

class ProductItem(BoxLayout):
    def __init__(self, product_id, product_data, delete_callback,
edit_callback, **kwargs):
        super().__init__(**kwargs)
        self.orientation = 'horizontal'
        self.size_hint_y = None
        self.height = 120
        self.padding = 5
        self.spacing = 10

        # Save product id and data
        self.product_id = product_id
        self.product_data = product_data

        # Create info layout
        info_layout = BoxLayout(orientation='vertical', size_hint_x=0.7)

        # Product info label
        name = product_data.get('nama', 'No Name')
        price = product_data.get('harga', 0)
        stock = product_data.get('stok', 0)

        info_label = Label(
            text=f"Nama: {name}\nHarga: Rp {price:,.0f}\nStok: {stock}",
            size_hint_y=None,
            height=100,
            halign='left',
            valign='middle'
        )
        info_label.bind(size=info_label.setter('text_size'))
        info_layout.add_widget(info_label)

        # Button layout
        button_layout = BoxLayout(orientation='vertical', size_hint_x=0.3,
spacing=5)

        # Edit button
        edit_btn = Button(
            text='Edit',
            size_hint_y=0.5,
            background_color=(0.3, 0.5, 0.9, 1)
        )
        edit_btn.bind(on_press=lambda x: edit_callback(product_id,
product_data))

        # Delete button
        delete_btn = Button(
            text='Hapus',
            size_hint_y=0.5,
            background_color=(0.9, 0.3, 0.3, 1)
        )
        delete_btn.bind(on_press=lambda x: delete_callback(product_id))

        button_layout.add_widget(edit_btn)
        button_layout.add_widget(delete_btn)

        # Add layouts to main layout
        self.add_widget(info_layout)
        self.add_widget(button_layout)

class ProductList(Screen):
    container = ObjectProperty(None)

    def on_enter(self):
        self.load_products()

    def load_products(self):
        self.container.clear_widgets()
        products = Database.get_all_products()

        if products:
            for product_id, product_data in products:
                product_item = ProductItem(
                    product_id,
                    product_data,
                    self.delete_product,
                    self.edit_product
                )
                self.container.add_widget(product_item)
        else:
            self.container.add_widget(
                Label(
                    text="Tidak ada produk tersedia",
                    size_hint_y=None,
                    height=100
                )
            )

    def show_add_product(self):
        self.manager.current = 'add_product'

    def edit_product(self, product_id, product_data):
        edit_screen = self.manager.get_screen('edit_product')
        edit_screen.set_product(product_id, product_data)
        self.manager.current = 'edit_product'

    def delete_product(self, product_id):
        confirm_popup = Popup(
            title='Konfirmasi',
            size_hint=(None, None),
            size=(300, 200)
        )
        content = BoxLayout(orientation='vertical', padding=10,
spacing=10)
        content.add_widget(Label(text='Apakah Anda yakin ingin\nmenghapus
produk ini?'))

        buttons = BoxLayout(size_hint_y=None, height=40, spacing=10)

        # Cancel button
        cancel_btn = Button(text='Batal')
        cancel_btn.bind(on_press=confirm_popup.dismiss)

        # Confirm button
        def confirm_delete(instance):
            try:
                Database.delete_product(product_id)
                self.load_products()  # Reload the list
                confirm_popup.dismiss()
                self.show_popup('Sukses', 'Produk berhasil dihapus!')
            except Exception as e:
                confirm_popup.dismiss()
                self.show_popup('Error', f'Gagal menghapus produk:
{str(e)}')

        confirm_btn = Button(text='Hapus', background_color=(0.9, 0.3,
0.3, 1))
        confirm_btn.bind(on_press=confirm_delete)

        buttons.add_widget(cancel_btn)
        buttons.add_widget(confirm_btn)

        content.add_widget(buttons)
        confirm_popup.content = content
        confirm_popup.open()

    def show_popup(self, title, content):
        popup = Popup(
            title=title,
            content=Label(text=content),
            size_hint=(None, None),
            size=(400, 200)
        )
        popup.open()

class AddProduct(Screen):
    name_input = ObjectProperty(None)
    price_input = ObjectProperty(None)
    stock_input = ObjectProperty(None)

    def add_product(self):
        nama = self.name_input.text.strip()
        harga = self.price_input.text.strip()
        stok = self.stock_input.text.strip()
        if nama and harga and stok:
            try:
                # Konversi input ke format yang sesuai
                harga_float = float(harga)
                stok_int = int(stok)

                # Create product data
                product_data = {
                    'nama': nama,
                    'harga': harga_float,
                    'stok': stok_int
                }

                # Push to Firebase
                Database.add_product(product_data)

                # Clear inputs
                self.name_input.text = ''
                self.price_input.text = ''
                self.stock_input.text = ''

                # Show success popup
                self.show_popup('Sukses', 'Produk berhasil ditambahkan!')

                # Return to product list
                self.manager.current = 'product_list'
            except ValueError:
                self.show_popup('Error', 'Harga dan stok harus berupa
angka!')
            except Exception as e:
                self.show_popup('Error', f'Terjadi kesalahan: {str(e)}')
        else:
            self.show_popup('Error', 'Semua field harus diisi!')

    def show_popup(self, title, content):
        popup = Popup(
            title=title,
            content=Label(text=content),
            size_hint=(None, None),
            size=(400, 200)
        )
        popup.open()

    def cancel(self):
        self.manager.current = 'product_list'

class EditProduct(Screen):
    name_input = ObjectProperty(None)
    price_input = ObjectProperty(None)
    stock_input = ObjectProperty(None)
    product_id = StringProperty(None)

    def on_enter(self):
        if hasattr(self, 'product_data'):
            self.name_input.text = str(self.product_data.get('nama', ''))
            self.price_input.text = str(self.product_data.get('harga',
''))
            self.stock_input.text = str(self.product_data.get('stok', ''))

    def set_product(self, product_id, product_data):
        self.product_id = product_id
        self.product_data = product_data

    def update_product(self):
        nama = self.name_input.text.strip()
        harga = self.price_input.text.strip()
        stok = self.stock_input.text.strip()

        if nama and harga and stok:
            try:
                product_data = {
                    'nama': nama,
                    'harga': float(harga),
                    'stok': int(stok)
                }

                Database.update_product(self.product_id, product_data)
                self.show_popup('Sukses', 'Produk berhasil diupdate!')
                self.manager.current = 'product_list'
            except ValueError:
                self.show_popup('Error', 'Harga dan stok harus berupa
angka!')
            except Exception as e:
                self.show_popup('Error', f'Terjadi kesalahan: {str(e)}')
        else:
            self.show_popup('Error', 'Semua field harus diisi!')

    def show_popup(self, title, content):
        popup = Popup(
            title=title,
            content=Label(text=content),
            size_hint=(None, None),
            size=(400, 200)
        )
        popup.open()

    def cancel(self):
        self.manager.current = 'product_list'
```

3.4 KV File (product.kv)

```
<ProductList>:
    container: container
    BoxLayout:
        orientation: 'vertical'
        padding: 10
        spacing: 10

        Label:
            text: 'Daftar Produk'
            size_hint_y: None
            height: 50
            font_size: 24

        ScrollView:
            GridLayout:
                id: container
                cols: 1
                spacing: 10
                size_hint_y: None
                height: self.minimum_height

        Button:
            text: 'Tambah Produk'
            size_hint_y: None
            height: 50
            on_press: root.show_add_product()

<AddProduct>:
    name_input: name_input
    price_input: price_input
    stock_input: stock_input

    BoxLayout:
        orientation: 'vertical'
        padding: 10
        spacing: 10

        Label:
            text: 'Tambah Produk Baru'
            size_hint_y: None
            height: 50
            font_size: 24

        TextInput:
            id: name_input
            hint_text: 'Nama Produk'
            size_hint_y: None
            height: 40
            multiline: False

        TextInput:
            id: price_input
            hint_text: 'Harga'
            size_hint_y: None
            height: 40
            multiline: False
            input_filter: 'float'
        TextInput:
            id: stock_input
            hint_text: 'Stok'
            size_hint_y: None
            height: 40
            multiline: False
            input_filter: 'int'

        BoxLayout:
            size_hint_y: None
            height: 50
            spacing: 10

            Button:
                text: 'Batal'
                on_press: root.cancel()

            Button:
                text: 'Tambah'
                on_press: root.add_product()

<EditProduct>:
    name_input: name_input
    price_input: price_input
    stock_input: stock_input

    BoxLayout:
        orientation: 'vertical'
        padding: 10
        spacing: 10

        Label:
            text: 'Edit Produk'
            size_hint_y: None
            height: 50
            font_size: 24

        TextInput:
            id: name_input
            hint_text: 'Nama Produk'
            size_hint_y: None
            height: 40
            multiline: False

        TextInput:
            id: price_input
            hint_text: 'Harga'
            size_hint_y: None
            height: 40
            multiline: False
            input_filter: 'float'

        TextInput:
            id: stock_input
            hint_text: 'Stok'
            size_hint_y: None
            height: 40
            multiline: False
            input_filter: 'int'

        BoxLayout:
            size_hint_y: None
            height: 50
            spacing: 10

            Button:
                text: 'Batal'
                on_press: root.cancel()

            Button:
                text: 'Update'
                on
```