

CHALLENGE UAS PEMDAS

Nama : Daanya Kirana Arifa

NIM : 25031554272

Kelas : 2025G

Challenge : Merubah tampilan

```
from kivy.app import App
from kivy.uix.boxlayout import BoxLayout
from kivy.uix.floatlayout import FloatLayout
from kivy.uix.gridlayout import GridLayout
from kivy.uix.label import Label
from kivy.uix.button import Button
from kivy.uix.image import Image
from kivy.core.window import Window
from kivy.graphics import Color, Rectangle, RoundedRectangle, Ellipse, Line,
Canvas
from kivy.animation import Animation
from kivy.clock import Clock
from kivy.properties import NumericProperty, StringProperty, ListProperty
import random

class AnimatedBackground(FloatLayout):
    """Background dengan animasi partikel"""
    def __init__(self, **kwargs):
        super().__init__(**kwargs)
        with self.canvas.before:
            # Gradient background
            Color(0.10, 0.18, 0.25, 1) # Dark blue base
            self.bg_rect = Rectangle(pos=self.pos, size=self.size)

            Color(0.15, 0.27, 0.35, 0.8) # Medium blue layer
            self.bg_layer = Rectangle(
                pos=(self.pos[0], self.pos[1] + self.size[1]*0.6),
                size=(self.size[0], self.size[1]*0.4)
            )

        self.bind(pos=self.update_bg, size=self.update_bg)
        self.create_particles()

    def update_bg(self, *args):
        self.bg_rect.pos = self.pos
        self.bg_rect.size = self.size
        self.bg_layer.pos = (self.pos[0], self.pos[1] + self.size[1]*0.6)
        self.bg_layer.size = (self.size[0], self.size[1]*0.4)

    def create_particles(self):
        """Membuat partikel animasi"""
```

```

        for i in range(20):
            Clock.schedule_once(lambda dt, idx=i: self.add_particle(), i *
0.2)

    def add_particle(self):
        """Menambahkan partikel ke background"""
        particle = Particle()
        particle.pos = (random.randint(0, int(Window.width)),
                        random.randint(0, int(Window.height)))
        self.add_widget(particle)

class Particle(Label):
    """Partikel animasi untuk background"""
    def __init__(self, **kwargs):
        super().__init__(**kwargs)
        self.text = '●'
        self.font_size = random.randint(12, 24)
        self.color = (0.27, 0.55, 0.71, random.uniform(0.2, 0.5))
        self.size_hint = (None, None)
        self.size = (self.font_size, self.font_size)

        # Animasi partikel
        self.animate_particle()

    def animate_particle(self):
        anim = Animation(
            x=random.randint(0, int(Window.width)),
            y=random.randint(0, int(Window.height)),
            duration=random.uniform(8, 15)
        )
        anim += Animation(
            x=random.randint(0, int(Window.width)),
            y=random.randint(0, int(Window.height)),
            duration=random.uniform(8, 15)
        )
        anim.repeat = True
        anim.start(self)

class OptionButton(FloatLayout):
    """Tombol pilihan dengan logo dan teks"""
    def __init__(self, label_text, icon_color, **kwargs):
        super().__init__(**kwargs)
        self.size_hint = (None, None)
        self.size = (180, 220)

        # Background tombol
        with self.canvas.before:
            Color(0.20, 0.29, 0.37, 0.9)

```

```

        self.bg_rect = RoundedRectangle(
            pos=self.pos,
            size=self.size,
            radius=[20]
        )

        # Efek highlight
        Color(*icon_color, 0.3)
        self.highlight = RoundedRectangle(
            pos=(self.pos[0], self.pos[1] + self.size[1] - 50),
            size=(self.size[0], 50),
            radius=[20, 20, 0, 0]
        )

        # Container untuk logo
        logo_container = FloatLayout(
            size_hint=(1, 0.6),
            pos_hint={'x': 0, 'y': 0.4}
        )

        # Logo lingkaran
        with logo_container.canvas:
            Color(*icon_color)
            self.circle = Ellipse(
                pos=(self.size[0]*0.5 - 40, self.size[1]*0.6 - 40),
                size=(80, 80)
            )

        # Ikon di dalam logo
        icon_label = Label(--
            font_size=36,
            color=(1, 1, 1, 1),
            pos_hint={'center_x': 0.5, 'center_y': 0.5},
            size_hint=(None, None),
            size=(100, 100)
        )
        logo_container.add_widget(icon_label)

        # Label teks
        text_label = Label(
            text=label_text,
            font_size=18,
            bold=True,
            color=(0.9, 0.9, 0.9, 1),
            pos_hint={'center_x': 0.5, 'center_y': 0.2},
            size_hint_y=0.4
        )

```

```

self.add_widget(logo_container)
self.add_widget(text_label)

self.bind(pos=self.update_graphics, size=self.update_graphics)

# Animasi hover
self.original_size = self.size.copy()
self.original_y = self.pos[1]

def update_graphics(self, *args):
    self.bg_rect.pos = self.pos
    self.bg_rect.size = self.size
    self.highlight.pos = (self.pos[0], self.pos[1] + self.size[1] - 50)
    self.highlight.size = (self.size[0], 50)

    # Update posisi circle
    if hasattr(self, 'circle'):
        self.circle.pos = (self.pos[0] + self.size[0]*0.5 - 40,
                           self.pos[1] + self.size[1]*0.6 - 40)

def on_touch_down(self, touch):
    if self.collide_point(*touch.pos):
        anim = Animation(size=(self.size[0]*0.95, self.size[1]*0.95),
duration=0.1)
        anim += Animation(size=self.original_size, duration=0.1)
        anim.start(self)
        return True
    return super().on_touch_down(touch)

def on_touch_move(self, touch):
    if self.collide_point(*touch.pos):
        self.canvas.before.clear()
        with self.canvas.before:
            Color(0.25, 0.35, 0.45, 0.9)
            self.bg_rect = RoundedRectangle(
                pos=self.pos,
                size=self.size,
                radius=[20]
            )
        return super().on_touch_move(touch)

def on_touch_up(self, touch):
    if self.collide_point(*touch.pos):
        self.canvas.before.clear()
        with self.canvas.before:
            Color(0.20, 0.29, 0.37, 0.9)
            self.bg_rect = RoundedRectangle(
                pos=self.pos,

```

```

        size=self.size,
        radius=[20]
    )
    return super().on_touch_up(touch)

class TaskReminderScreen(FloatLayout):
    def __init__(self, **kwargs):
        super().__init__(**kwargs)
        self.setup_ui()

    def setup_ui(self):
        # Background animasi
        self.bg = AnimatedBackground()
        self.add_widget(self.bg)

        # Container utama
        main_container = FloatLayout(size_hint=(1, 1))

        # Header dengan logo kecil
        self.create_header(main_container)

        # Judul utama TASK REMINDER
        self.create_main_title(main_container)

        # Subtitle
        self.create_subtitle(main_container)

        # Pilihan opsi (Login, Sign Up, Exit)
        self.create_options(main_container)

        # Footer
        self.create_footer(main_container)

        self.add_widget(main_container)

    def create_header(self, container):
        """Membuat header dengan logo kecil"""
        header_layout = BoxLayout(
            orientation='horizontal',
            size_hint=(1, None),
            height=80,
            pos_hint={'top': 1},
            padding=[30, 10]
        )

        # Logo kecil di kiri
        logo_small = FloatLayout(
            size_hint=(None, None),

```

```

        size=(60, 60)
    )

    with logo_small.canvas:
        Color(0.27, 0.78, 0.78, 1)
        Ellipse(pos=logo_small.pos, size=logo_small.size)

        # Ikon checklist
        Color(1, 1, 1, 1)
        Line(
            points=[
                logo_small.pos[0] + 15, logo_small.pos[1] + 30,
                logo_small.pos[0] + 25, logo_small.pos[1] + 40,
                logo_small.pos[0] + 45, logo_small.pos[1] + 20
            ],
            width=2
        )

def create_main_title(self, container):
    """Membuat judul utama TASK REMINDER"""
    # Efek shadow untuk judul
    shadow_label = Label(
        text="TASK REMINDER",
        font_size=72,
        bold=True,
        color=(0, 0, 0, 0.3),
        pos_hint={'center_x': 0.5, 'center_y': 0.75},
        size_hint=(None, None),
        size=(600, 100)
    )
    shadow_label.x += 4
    shadow_label.y -= 4
    container.add_widget(shadow_label)

    # Judul utama dengan efek glowing
    main_title = Label(
        text="TASK REMINDER",
        font_size=72,
        bold=True,
        color=(0.27, 0.78, 0.78, 1),
        pos_hint={'center_x': 0.5, 'center_y': 0.75},
        size_hint=(None, None),
        size=(600, 100)
    )
    container.add_widget(main_title)

```

```

# Animasi judul
self.animate_title(main_title)

def animate_title(self, title):
    """Animasi untuk judul utama"""
    anim = Animation(color=(0.27, 0.78, 0.78, 1), duration=2) + \
        Animation(color=(0.35, 0.85, 0.85, 1), duration=2)
    anim.repeat = True
    anim.start(title)

def create_subtitle(self, container):
    """Membuat subtitle"""
    subtitle = Label(
        text="choose one option to continue :",
        font_size=24,
        color=(0.8, 0.8, 0.8, 0.9),
        pos_hint={'center_x': 0.5, 'center_y': 0.63},
        size_hint=(None, None),
        size=(400, 50)
    )
    container.add_widget(subtitle)

def create_options(self, container):
    """Membuat pilihan opsi"""
    options_container = GridLayout(
        cols=3,
        spacing=60,
        size_hint=(0.8, 0.4),
        pos_hint={'center_x': 0.65, 'center_y': 0.30}
    )

    # Opsi Login
    login_btn = OptionButton(
        label_text="LOGIN",
        icon_color=(0.27, 0.78, 0.78, 1) # Teal
    )

    # Opsi Sign Up
    signup_btn = OptionButton(
        label_text="SIGN UP",
        icon_color=(0.35, 0.65, 0.85, 1) # Blue
    )

    # Opsi Exit
    exit_btn = OptionButton(
        label_text="EXIT",
        icon_color=(0.85, 0.45, 0.45, 1) # Red
    )

```

```

    )

    options_container.add_widget(login_btn)
    options_container.add_widget(signup_btn)
    options_container.add_widget(exit_btn)

    container.add_widget(options_container)

def create_footer(self, container):
    """Membuat footer"""
    footer = Label(
        text="@ 2023 Task Reminder App | Manage your tasks efficiently",
        font_size=14,
        color=(0.6, 0.6, 0.6, 0.7),
        pos_hint={'center_x': 0.5, 'center_y': 0.05},
        size_hint=(None, None),
        size=(400, 30)
    )
    container.add_widget(footer)

class TaskReminderApp(App):
    def build(self):
        # Set window properties
        Window.clearcolor = (0.10, 0.18, 0.25, 1)
        Window.size = (1200, 800)
        Window.set_title('Task Reminder - Welcome')
        Window.minimum_width = 900
        Window.minimum_height = 600

        return TaskReminderScreen()

if __name__ == '__main__':
    TaskReminderApp().run()

```


HASIL PERUBAHAN :

