

LAPORAN PRAKTIKUM

Internet of Things

Virtual Drawing



Santo Martogi Simangunsong 11323017

Marshanda Kasih Simangunsong 11323020

Lasro P N Tamba 11323052

Anjelika A. Simamora 11323057

D-III Teknologi Informasi

INSTITUT TEKNOLOGI DEL

FAKULTAS VOKASI

Tema

Rancang Bangun Virtual Drawing

Deskripsi

Virtual Drawing merupakan sistem berbasis teknologi yang memungkinkan pengguna agar dapat melukis secara virtual dengan memanfaatkan pelacakan gerakan tangan menggunakan kamera. Sistem ini dirancang untuk mendeteksi gerakan tangan sebagai alat lukis digital, dilengkapi dengan berbagai pilihan warna, yang menghadirkan pengalaman melukis kreatif tanpa media fisik.

Masalah

Di era digital, menggambar manual menggunakan tangan sering dianggap kurang relevan karena meningkatnya penggunaan alat menggambar digital seperti aplikasi desain. Namun terdapat kebutuhan untuk mengintegrasikan proses menggambar secara teknologi digital sehingga gambar yang dihasilkan dapat disimpan ke dalam database secara real-time. Tantangan utama adalah bagaimana menciptakan sistem yang dapat merekam proses menggambar secara detail dan dapat disimpan ke dalam database secara efisien tanpa mengurangi kualitas gambar.

Tujuan

Dengan membuat virtual Drawing ini bertujuan untuk memberikan pengalaman baru yang inovatif bagi pengguna untuk melukis kreatif tanpa media fisik. Sistem ini dirancang untuk memanfaatkan teknologi pelacakan benda dengan warna cerah, selain itu sistem ini juga diharapkan dapat beroperasi dengan baik.

Rumusan Masalah:

1. Bagaimana cara mendeteksi dan merekam gerakan tangan untuk menghasilkan gambar digital secara detail?
2. Bagaimana cara menyimpan hasil gambar tersebut ke dalam database secara efisien?
3. Apa saja tantangan yang mungkin dihadapi dalam integrasi sistem ini, seperti kompatibilitas perangkat keras dan perangkat lunak?

Tools

- Visual Studio Code
- Kamera
- MySQL

Cara Menyimpan Data

Penyimpanan gambar digital dilakukan dengan memanfaatkan kamera sebagai sensor untuk mendeteksi dan merekam gerakan tangan secara real-time, sehingga setiap gerakan tangan yang tertangkan akan diolah menjadi gambar digital. Setelah gambar terbentuk, user dapat memilih opsi *save* untuk menyimpan gambar kedalam database. Ketika kita memilih opsi *save*, sistem akan mengolah data gambar yang telah dihasilkan dan akan mengonversinya langsung kedalam database. Proses ini memungkinkan gambar disimpan secara cepat dan dapat diakses kembali dengan mudah.

Metodologi

1. Perancang sistem

Sistem yang dirancang dengan pendekatan modular yang terdiri dari beberapa komponen, yaitu:

- **Kamera**, berfungsi sebagai sensor untuk mendeteksi gerakan tangan dan merekam gambar selama proses menggambar.
- **MediaPipe dan OpenCV**, memanfaatkan algoritma pemrosesan citra untuk mendeteksi gerakan tangan dan mengonversinya menjadi gambar digital.
- **Database**, seperti MySQL yang digunakan untuk menyimpan data hasil gambar.

2. Metode MediaPipe dan OpenCv

- Deteksi gerakan tangan dengan mediapipe dan opencv

```
import cv2
import numpy as np
import mediapipe as mp
from collections import deque
import requests
```

```
# Process hand landmarks
result = hands.process(framergb)
if result.multi_hand_landmarks:
    for hands_lms in result.multi_hand_landmarks:
        landmarks = [(int(lm.x * 640), int(lm.y * 480)) for lm in hands_lms.landmark]

        mpDraw.draw_landmarks(frame, hands_lms, mpHands.HAND_CONNECTIONS)

        fore_finger = landmarks[8]
        thumb = landmarks[4]
        cv2.circle(frame, fore_finger, 3, (0, 255, 0), -1)
```

Kode diatas bertanggung jawab untuk mendeteksi tangan dan melacak posisi jari.

- Pembuatan canvas dan pilihan warna

```
# Canvas setup
paintWindow = np.zeros((471, 636, 3)) + 255
buttons = [
    {"rect": (40, 1, 140, 65), "color": (0, 0, 0), "label": "CLEAR"},
    {"rect": (160, 1, 255, 65), "color": (255, 0, 0), "label": "BLUE"},
    {"rect": (275, 1, 370, 65), "color": (0, 255, 0), "label": "GREEN"},
    {"rect": (390, 1, 485, 65), "color": (0, 0, 255), "label": "RED"},
    {"rect": (505, 1, 600, 65), "color": (0, 255, 255), "label": "YELLOW"},
    {"rect": (615, 1, 700, 65), "color": (0, 0, 0), "label": "SAVE"},
]

for button in buttons:
    x1, y1, x2, y2 = button["rect"]
    paintWindow = cv2.rectangle(paintWindow, (x1, y1), (x2, y2), button["color"], 2)
    cv2.putText(
        paintWindow,
        button["label"],
        (x1 + 15, y1 + 40),
        cv2.FONT_HERSHEY_SIMPLEX,
        0.7,
        (0, 0, 0) if button["color"] != (0, 0, 0) else (255, 255, 255),
        2,
        cv2.LINE_AA,
    )

cv2.namedWindow("Paint", cv2.WINDOW_AUTOSIZE)
```

Kode diatas membuat tombol warna dan fitur menggambar.

- Penyimpanan data gambar

```
# Function to save image
def save_image():
    # Simpan gambar ke folder 'images'
    image_path = "images/painted_image.png"
    cv2.imwrite(image_path, paintWindow)
    print(f"Image saved as {image_path}")
    return image_path
```

Kode ini akan menyimpan gambar di folder lokal.

- Pengiriman data ke database

```
# Fungsi untuk mengirim data gambar ke server PHP
def send_image_to_server(image_path):
    url = 'http://localhost/drawing_virtual/fetch_data.php'
    data = {'image_path': image_path}
    try:
        response = requests.post(url, data=data)
        if response.status_code == 200:
            print("Image information saved to database.")
        else:
            print(f"Failed to save image information: {response.status_code}")
    except Exception as e:
        print(f"Error: {e}")
```

```

elif button["label"] == "SAVE":
    # Simpan gambar dan kirim informasi ke server
    image_path = save_image()
    send_image_to_server(image_path)
else:
    colorIndex = i - 1
    break

```

Kode diatas mengirim informasi gambar ke server PHP dan ketika tombol save ditekan, sistem akan menyimpan gambar dan mengirim data ke server. Proses penyimpanan ini dilakukan secara real-time yang dilakukan disisi server PHP saat menerima data dari python.

- Real-time feedback pada layar kamera

```

47
48 # Draw lines on the canvas and frame
49 points = [bpoints, gpoints, rpoints, ypoints]
50 for i in range(len(points)):
51     for j in range(len(points[i])):
52         for k in range(1, len(points[i][j])):
53             if points[i][j][k - 1] is None or points[i][j][k] is None:
54                 continue
55             cv2.line(frame, points[i][j][k - 1], points[i][j][k], colors[i], 2)
56             cv2.line(paintwindow, points[i][j][k - 1], points[i][j][k], colors[i], 2)
57
58 cv2.imshow("Output", frame)
59 cv2.imshow("Paint", paintwindow)
60
61 if cv2.waitKey(1) & 0xFF == ord("q"):
62     break
63

```

Kode diatas berfungsi untuk menggambar garis tangan di layar kamera.