

EKSPLORASI MEDIAPIPE UNTUK VIRTUAL PIANO INTERAKTIF BERBASIS DETEKSI GERAKAN TANGAN



Dibuat Oleh:

Annisa Suci Soleha	(231511005)
M. Ridho Firdaus	(231511022)
Naufal Asidiq	(231511023)

Kelas : 2A

JURUSAN TEKNIK INFORMATIKA

PRODI D3 TEKNIK KOMPUTER DAN INFORMATIKA

POLITEKNIK NEGERI BANDUNG

2025

1. Latar Belakang

Proyek ini bertujuan mengembangkan piano virtual interaktif menggunakan MediaPipe Hands untuk mendeteksi dan melacak gerakan tangan secara real-time. Dengan webcam sebagai input, sistem memetakan posisi jari ke tuts piano digital yang divisualisasikan menggunakan OpenCV. Antarmuka Streamlit memungkinkan pengguna mengatur parameter seperti jumlah tuts, rentang nada, dan sensitivitas deteksi. Tujuan utama adalah menciptakan alat edukasi atau hiburan berbasis computer vision yang memungkinkan pengguna berlatih piano tanpa alat fisik, sekaligus mengeksplorasi kemampuan MediaPipe dalam aplikasi interaktif.

2. Studi Literatur

MediaPipe adalah framework open-source dari Google untuk pemrosesan data visual dan audio real-time. Modul MediaPipe Hands menggunakan model deep learning berbasis Convolutional Neural Networks (CNN) untuk mendeteksi 21 landmark per tangan, termasuk ujung jari dan sendi, dengan akurasi tinggi (~95% pada dataset internal Google) dan efisiensi untuk perangkat low-end. Model ini dilatih pada dataset tangan sintetis dan nyata, memungkinkan deteksi robust dalam berbagai kondisi pencahayaan dan orientasi.

OpenCV digunakan untuk rendering tuts piano sebagai poligon dan deteksi interaksi jari dengan tuts melalui fungsi `pointPolygonTest`. Streamlit, sebagai alat pengembangan antarmuka web berbasis Python, memungkinkan visualisasi interaktif dan pengaturan parameter secara real-time. Untuk suara, kami menggunakan Pygame untuk memutar file .wav yang sesuai dengan not piano.

3. Alat dan Lingkungan

Bahasa Pemrograman:

- Python 3.9

Pustaka:

- mediapipe==0.10.21: Deteksi tangan.
- opencv-python==4.9.0.80: Rendering tuts dan pengolahan citra.
- numpy==1.26.0: Operasi numerik.
- streamlit==1.28.0: Antarmuka web.
- pygame==2.5.2: Pemutaran suara piano.

Lingkungan Pengembangan:

- Google Colab (untuk eksperimen awal).
- Visual Studio Code (untuk pengembangan utama).
- Sistem Operasi: Windows 10.

Perangkat Keras:

- Laptop dengan webcam 720p.
- Prosesor: Intel Core i5 (atau setara).
- RAM: 8 GB.

File Pendukung:

- Folder notes berisi file .wav untuk suara piano (contoh: C3.wav, Bb3.wav).

4. Referensi dan Dasar Model

Proyek ini menggunakan MediaPipe Hands sebagai inti deteksi gerakan. Model ini terdiri dari dua komponen:

1. Palm Detector: Mendeteksi keberadaan tangan dalam frame menggunakan BlazePalm, model CNN ringan.
2. Landmark Regressor: Memprediksi 21 koordinat 3D (x, y, z) per tangan, termasuk ujung jari (thumb, index, middle, ring, pinky).

Model ini dipilih karena:

- Efisiensi: Berjalan real-time pada perangkat standar tanpa GPU.
- Akurasi: Mampu mendeteksi gerakan cepat dengan latensi rendah (<50 ms per frame).
- Kemudahan Integrasi: API Python yang sederhana.

OpenCV digunakan untuk membuat tuts piano sebagai poligon (putih dan hitam) dan memetakan posisi jari ke tuts. Streamlit menyediakan antarmuka untuk mengatur parameter seperti jumlah oktaf (1–7), rentang nada (oktaf 1–7), dan sensitivitas ketukan (1–50 piksel). Pygame memutar suara not berdasarkan deteksi ketukan.

5. Langkah-Langkah Implementasi

1. Setup Lingkungan:

- a. Install pustaka: `pip install mediapipe opencv-python numpy streamlit pygame`.
- b. Siapkan folder notes dengan file .wav dari freesound.org.

2. Inisialisasi MediaPipe Hands:

- a. Konfigurasi: `min_detection_confidence=0.7, min_tracking_confidence=0.7, max_num_hands=2`.
- b. Kode contoh:

```
mp_hands = mp.solutions.hands
hands = mp_hands.Hands(min_detection_confidence=0.7,
min_tracking_confidence=0.7, max_num_hands=2)
```

3. Konfigurasi Tuts Piano:

- Tuts putih (7 per oktaf) dan hitam (5 per oktaf) digambar sebagai poligon menggunakan OpenCV.
- Jumlah tuts dihitung berdasarkan parameter `num_octaves` dari Streamlit.
- Contoh kode rendering tuts:

```
img = cv2.fillPoly(img, piano.white, (255, 255, 255)) # Tuts putih
img = cv2.fillPoly(img, piano.black, (0, 0, 0))
# Tuts hitam
```

4. Deteksi Landmark dan Ketukan:

- Landmark jari (indeks 4, 8, 12 untuk jempol, telunjuk, tengah) dipetakan ke koordinat frame.
- Ketukan dideteksi jika jari bergerak ke bawah melebihi `tap_threshold` (default 20 piksel).
- Contoh kode deteksi:

```
dis = y[tip] - previous_y[tip]
if dis > tap_threshold:
    tapped_keys.append(tip)
```

5. Visualisasi Interaktif:

- Tuts berubah warna (putih jadi hijau, hitam jadi abu-abu) saat diketuk.
- Streamlit menampilkan dua view: Piano View (webcam) dan Top View (perspektif atas).
- FPS ditampilkan di layar untuk pantau performa.

6. Integrasi Suara:

- File .wav diputar menggunakan `pygame.mixer.Sound` saat tuts diketuk.
- Contoh kode:

```
sound = pygame.mixer.Sound(f"notes/{note}.wav")
channel = pygame.mixer.Channel(channel_id)
channel.play(sound)
```

7. Antarmuka Streamlit:

- Slider untuk jumlah oktaf, rentang nada, sensitivitas ketukan.
- Checkbox untuk konfigurasi posisi piano menggunakan dua tangan.
- Tombol “Stop Playing” untuk stop aplikasi.

8. Pengujian Real-Time:

- Sistem diuji dengan webcam 720p di ruangan terang yang cukup untuk mendeteksi jari.
- Variasi: Berbagai jumlah oktaf (1–7), posisi tangan, dan pencahayaan.

6. Tantangan dan Solusi

Tantangan	Solusi
-----------	--------

Landmark bergeser saat tangan bergerak cepat	Atur <code>min_detection_confidence</code> ke 0.7 dan <code>max_num_hands</code> ke 2
Delay frame saat proses berat	Kurangi frekuensi deteksi (proses setiap 2 frame) dan optimasi resolusi ke 800x600.
Mapping tuts tidak presisi	Kalibrasi ulang koordinat tuts berdasarkan posisi tangan dan gunakan <code>pointPolygonTest</code> untuk deteksi akurat.
Variasi pencahayaan memengaruhi hasil	Tambah preprocessing (peningkatan kontras) dan uji di lingkungan terang dengan latar kontras.
Tidak bisa memainkan 2 note sekaligus dalam satu waktu	Implementasikan multi-channel audio di Pygame untuk putar beberapa suara secara paralel. Contoh: <pre>pygame.mixer.set_num_channels(8)</pre>

7. Analisis Kinerja

- FPS (frame per second): 10–20 FPS pada laptop 720p
- Akurasi Deteksi: ~90% untuk satu tangan (evaluasi manual: 90/100 ketukan terdeteksi benar). Menurun ke ~80% untuk dua tangan karena occlusion.
- Stabilitas: Stabil pada pencahayaan cukup dan posisi tangan jelas.
- Respons sistem: Latensi deteksi 100–200 ms, cukup responsif untuk aplikasi edukasi.
- Streamlit UI: Update parameter (misalnya, jumlah oktaf) real-time.

8. Kesimpulan

Proyek ini berhasil mengimplementasikan piano virtual interaktif menggunakan MediaPipe Hands untuk deteksi tangan real-time, OpenCV untuk visualisasi tuts, dan Streamlit untuk antarmuka pengguna. Sistem mampu mendeteksi ketukan jari dengan akurasi tinggi (~90%)

dan memberikan feedback visual serta audio yang responsif. Tantangan seperti pencahayaan dan sinkronisasi not telah diatasi dengan preprocessing dan optimasi audio. Proyek ini menunjukkan potensi MediaPipe dalam aplikasi computer vision interaktif, terutama untuk edukasi musik.

9. Saran Pengembangan

- **Integrasi MIDI Output:** Gunakan library seperti `mido` untuk menghasilkan output MIDI, memungkinkan integrasi dengan software musik profesional.
- **Skala Otomatis:** Implementasikan deteksi resolusi webcam untuk menyesuaikan ukuran tuts secara otomatis menggunakan transformasi perspektif OpenCV.
- **Multi-Tangan dan Gesture:** Tambah deteksi gesture (misalnya, swipe untuk ganti oktaf) menggunakan MediaPipe Holistic untuk pose tubuh.
- **Deployment Mobile:** Konversi model MediaPipe ke TensorFlow Lite untuk aplikasi Android menggunakan Flutter, meningkatkan aksesibilitas.
- **Perekaman dan Playback:** Tambah fitur rekam gerakan tangan ke file JSON dan playback untuk analisis belajar pengguna.

10. Referensi

Google, "MediaPipe Solutions Guide," <https://ai.google.dev/edge/mediapipe/solutions/guide?hl=id>, diakses Juni 2025.

Zhang et al., "MediaPipe Hands: On-device Real-time Hand Tracking," arXiv:2006.10214, 2020.

OpenCV, "OpenCV Documentation," <https://docs.opencv.org>, diakses Juni 2025.

Streamlit, "Streamlit Docs," <https://docs.streamlit.io>, diakses Juni 2025.

Freesound, "Piano Sound Samples," <https://freesound.org>, diakses Juni 2025.