



---

Anggota 1: **Made Redy Wijaya (121140157)**

Tugas Ke: **Tugas Besar**

Anggota 2: **Farhan Apri Kesuma (121140179)**

Tanggal: **11-22-2024**

Anggota 3: **Carlos Piero Parhusip (121140193)**

Mata Kuliah: **Sistem/Teknologi Multimedia (IF4021)**

---

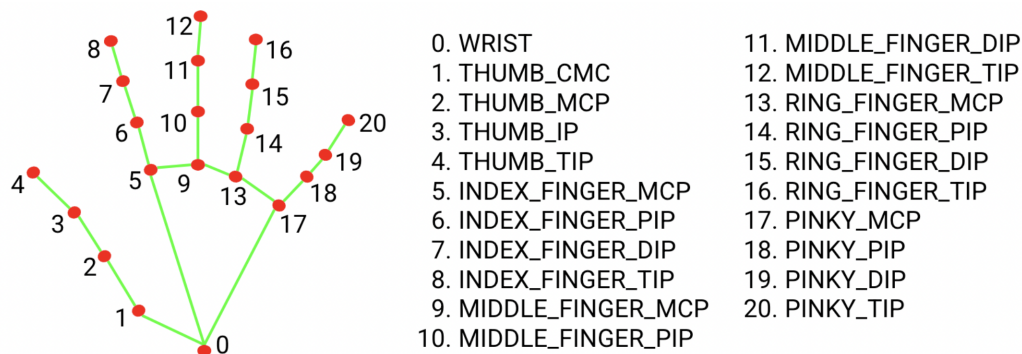
## Hand Sign Calculator Menggunakan MediaPipe dan OpenCV

### 1 Pendahuluan

Interaksi manusia-komputer (Human-Computer Interaction/HCI) terus berkembang, termasuk dalam teknologi berbasis gesture recognition yang memungkinkan perangkat lunak mengenali gerakan tangan sebagai input untuk menjalankan fungsi tertentu [1]. Teknologi ini telah banyak diterapkan, salah satunya pada filter Instagram, di mana gesture tangan digunakan untuk memicu efek tertentu secara interaktif. Dalam konteks ini, proyek bertujuan untuk mengembangkan kalkulator berbasis gesture tangan yang terinspirasi dari interaktivitas filter Instagram. Pengguna dapat menggunakan simbol tangan untuk memasukkan angka dan memilih operasi matematika seperti penjumlahan, pengurangan, perkalian, dan pembagian. Teknologi utama yang digunakan meliputi MediaPipe untuk mendeteksi tangan dan OpenCV untuk pemrosesan citra serta antarmuka pengguna. Penelitian ini berfokus pada bagaimana mendeteksi simbol angka berdasarkan pose tangan dan mengintegrasikan hasil deteksi gesture tangan dengan operasi matematika secara real-time [2]. Tujuan utama adalah menciptakan kalkulator interaktif yang memanfaatkan teknologi gesture recognition seperti pada filter Instagram dengan implementasi MediaPipe untuk deteksi tangan dan landmark secara akurat [3].

### 2 Alat dan Cara Kerja

Proyek ini menggunakan beberapa alat dan bahan utama untuk pengembangan sistem kalkulator berbasis gesture tangan. Bahasa pemrograman yang digunakan adalah **Python** dengan dukungan library **MediaPipe** untuk mendeteksi landmark tangan dan **OpenCV** untuk pemrosesan citra serta antarmuka video. Perangkat keras yang digunakan meliputi laptop dengan webcam bawaan atau kamera eksternal sebagai input visual. Gambar 1 merupakan nilai landmark dari tangan, untuk dokumentasi yang lebih lengkap bisa dilihat pada [tautan berikut ini](#).



Gambar 2.1: MediaPipe Hand Landmark

Dalam desain sistem, tahap pertama adalah deteksi tangan menggunakan MediaPipe untuk mengenali tangan dan landmarknya secara akurat. Selanjutnya, posisi landmark tangan dianalisis untuk mengidentifikasi simbol angka ('0' hingga '5') berdasarkan aturan tertentu. Sistem juga mencakup pilihan operasi matematika, di mana antarmuka menampilkan menu operasi seperti penjumlahan, pengurangan, perkalian, dan pembagian, yang dapat dipilih menggunakan posisi jari telunjuk. Hasil deteksi angka dari kedua tangan dan operasi yang dipilih diintegrasikan untuk melakukan perhitungan, dengan hasil akhirnya ditampilkan langsung di layar secara interaktif [4].

### 3 Penjelasan Kode Program

#### 3.1 Import Library

Library yang diperlukan dalam program ini diantaranya cv2, mediapipe, dan numpy.

```

1
2 import cv2
3 import mediapipe as mp
4 import numpy as np
5 import pygame
6

```

Kode 3.1: Library yang Digunakan

Penjelasan:

- cv2 (OpenCV): Digunakan untuk memproses citra/video secara real-time, termasuk menangkap gambar dari webcam dan menampilkan hasil.
- mediapipe: Library dari Google untuk deteksi landmark tangan yang mendukung pengenalan gesture.
- numpy: Library untuk manipulasi array, meskipun dalam kode ini tidak digunakan secara eksplisit.
- pygame: Dalam hal ini, pygame digunakan untuk merender efek suara dengan method mixer.Sound()

#### 3.2 Memuat File Audio

```

1
2 sound_click = pygame.mixer.Sound("sfx/click.mp3")
3 sound_success = pygame.mixer.Sound("sfx/succes.mp3")
4 sound_error = pygame.mixer.Sound("sfx/error.mp3")
5

```

Kode 3.2: Memuat File Audio

Penjelasan:

- *sound\_click*: Suara untuk klik operasi.
- *sound\_success*: Suara untuk hasil kalkulasi.
- *sound\_error*: Suara untuk kesalahan (misalnya, pembagian dengan nol).

### 3.3 Inisialisasi MediaPipe

```
1 mp_hands = mp.solutions.hands
2 hands = mp_hands.Hands(min_detection_confidence=0.7, min_tracking_confidence=0.5)
3 mp_drawing = mp.solutions.drawing_utils
4
5
```

Kode 3.3: Inisialisasi Mediapipe

- `mp.solutions.hands`: Komponen MediaPipe untuk deteksi tangan.
- `Hands`: Menginisialisasi model deteksi tangan dengan confidence untuk deteksi dan pelacakan (0.7 dan 0.5 secara default).
- `mp_drawing`: Komponen untuk menggambar landmark tangan pada frame yang diproses.

### 3.4 Variabel Global

```
1 left_hand_signs_detected = []
2 right_hand_signs_detected = []
3 active_operation = None
4 operation_processed = False
5 last_result = None
6 operations = {"+": "Tambah", "-": "Kurang", "*": "Kali", "/": "Bagi"}
7 selected = None
8 last_selected = None
9
10
```

Kode 3.4: Variabel Global

Penjelasan:

- *left\_hand\_signs\_detected* dan *right\_hand\_signs\_detected*: Menyimpan angka yang terdeteksi dari masing-masing tangan.
- *active\_operation*: Menyimpan operasi matematika yang dipilih (misalnya, "+", "-", "\*", "/").
- *operation\_processed*: Flag untuk memastikan operasi hanya diproses sekali.
- *last\_result*: Variabel untuk menyimpan hasil terakhir
- *operations*: Dictionary untuk mencocokkan simbol operasi dengan deskripsinya.
- *selected* dan *last\_selected*: Variabel untuk menyimpan operasi matematika yang dipilih supaya efek suara tidak dimainkan berkali-kali.

### 3.5 Variabel Kelola Waktu Jeda

```
1
2 last_sound_time = 0
3 click_delay = 1
4
```

Kode 3.5: Variabel Kelola Waktu Jeda

- *last\_sound\_time*: Waktu terakhir suara diputar.
- *click\_delay*: Waktu jeda antara suara klik.

### 3.6 Fungsi play\_sound\_with\_delay

```
1
2 global last_sound_time
3 current_time = time.time()
4 if current_time - last_sound_time >= delay:
5     sound.play()
6     last_sound_time = current_time
7
```

Kode 3.6: Fungsi memainkan suara dengan jeda

Penjelasan:

- Fungsi ini memainkan suara dengan jeda tertentu.
- *sound*: Objek suara yang akan diputar.
- *delay*: Waktu jeda antara pemutaran suara.
- Variabel global *last\_sound\_time* digunakan untuk menyimpan waktu terakhir suara diputar.
- Fungsi ini memeriksa apakah waktu sekarang sudah melewati waktu jeda sebelum memainkan suara.
- Jika sudah melewati waktu jeda, suara diputar dan *last\_sound\_time* diperbarui.

### 3.7 Fungsi detect\_hand\_sign

```
1
2 def detect_hand_sign(landmarks, handedness):
3
4     # Untuk tangan kanan (handedness == 'Right')
5     if handedness == "Right":
6         # Mengecek apakah tangan membentuk simbol 'nol'
7         if (landmarks[4].x > landmarks[3].x and landmarks[8].y > landmarks[6].y and landmarks
8             [12].y > landmarks[10].y and landmarks[16].y > landmarks[14].y and landmarks[20].y > landmarks
9             [18].y):
10             return '0'
11         # Mengecek apakah tangan membentuk simbol 'satu'
12         if (landmarks[4].x > landmarks[3].x and landmarks[8].y < landmarks[6].y and landmarks
13             [12].y > landmarks[10].y and landmarks[16].y > landmarks[14].y and landmarks[20].y > landmarks
14             [18].y):
15             return '1'
16         # Mengecek apakah tangan membentuk simbol 'dua'
17         if (landmarks[4].x > landmarks[3].x and landmarks[8].y < landmarks[6].y and landmarks
18             [12].y < landmarks[10].y and landmarks[16].y > landmarks[14].y and landmarks[20].y > landmarks
19             [18].y):
```

```

14         return '2'
15         # Mengecek apakah tangan membentuk simbol 'tiga'
16         if (landmarks[4].x > landmarks[3].x and landmarks[8].y < landmarks[6].y and landmarks
[12].y < landmarks[10].y and landmarks[16].y < landmarks[14].y and landmarks[20].y > landmarks
[18].y):
17             return '3'
18             # Mengecek apakah tangan membentuk simbol 'empat'
19             if (landmarks[4].x > landmarks[3].x and landmarks[8].y < landmarks[6].y and landmarks
[12].y < landmarks[10].y and landmarks[16].y < landmarks[14].y and landmarks[20].y < landmarks
[18].y):
20                 return '4'
21                 # Mengecek apakah tangan membentuk simbol 'lima'
22                 if (landmarks[4].x <= landmarks[3].x and landmarks[8].y < landmarks[6].y and landmarks
[12].y < landmarks[10].y and landmarks[16].y < landmarks[14].y and landmarks[20].y < landmarks
[18].y):
23                     return '5'
24
25         # Untuk tangan kiri (handedness == 'Left')
26         if handedness == "Left":
27             # Mengecek apakah tangan membentuk simbol 'nol'
28             if (landmarks[4].x <= landmarks[3].x and landmarks[8].y > landmarks[6].y and landmarks
[12].y > landmarks[10].y and landmarks[16].y > landmarks[14].y and landmarks[20].y > landmarks
[18].y):
29                 return '0'
30                 # Mengecek apakah tangan membentuk simbol 'satu'
31                 if (landmarks[4].x <= landmarks[3].x and landmarks[8].y < landmarks[6].y and landmarks
[12].y > landmarks[10].y and landmarks[16].y > landmarks[14].y and landmarks[20].y > landmarks
[18].y):
32                     return '1'
33                     # Mengecek apakah tangan membentuk simbol 'dua'
34                     if (landmarks[4].x <= landmarks[3].x and landmarks[8].y < landmarks[6].y and landmarks
[12].y < landmarks[10].y and landmarks[16].y > landmarks[14].y and landmarks[20].y > landmarks
[18].y):
35                         return '2'
36                         # Mengecek apakah tangan membentuk simbol 'tiga'
37                         if (landmarks[4].x <= landmarks[3].x and landmarks[8].y < landmarks[6].y and landmarks
[12].y < landmarks[10].y and landmarks[16].y < landmarks[14].y and landmarks[20].y > landmarks
[18].y):
38                             return '3'
39                             # Mengecek apakah tangan membentuk simbol 'empat'
40                             if (landmarks[4].x <= landmarks[3].x and landmarks[8].y < landmarks[6].y and landmarks
[12].y < landmarks[10].y and landmarks[16].y < landmarks[14].y and landmarks[20].y < landmarks
[18].y):
41                                 return '4'
42                                 # Mengecek apakah tangan membentuk simbol 'lima'
43                                 if (landmarks[4].x > landmarks[3].x and landmarks[8].y < landmarks[6].y and landmarks
[12].y < landmarks[10].y and landmarks[16].y < landmarks[14].y and landmarks[20].y < landmarks
[18].y):
44                                     return '5'
45
46         return None
47

```

Kode 3.7: Fungsi Mendeteksi Landmark Tangan

Penjelasan:

- Fungsi ini mendeteksi simbol angka ('0'-'5') berdasarkan posisi landmark tangan.
- Parameter:
- landmarks: Koordinat landmark tangan yang dihasilkan oleh MediaPipe.

- handedness: Menentukan apakah tangan adalah kiri atau kanan.
- Logika Deteksi:
- Setiap angka diwakili oleh posisi spesifik jari:
- Jari tertutup: Landmark y dari ruas atas jari lebih besar dari landmark y ruas bawah.
- Jari terbuka: Landmark y dari ruas atas lebih kecil dari landmark y ruas bawah.
- Fungsi mengembalikan angka ('0'-'5') jika simbol dikenali atau None jika tidak ada simbol yang valid.

### 3.8 Fungsi detect\_operation

```

1  def detect_operation(x, y):
2      global selected, last_selected
3      selected = False
4      for i, (symbol, label) in enumerate(operations.items()):
5          if 10 <= x <= 130 and 50 + i * 60 <= y <= 90 + i * 60:
6              if last_selected != symbol:
7                  play_sound_with_delay(sound_click, click_delay)
8                  last_selected = symbol
9      return symbol
10     last_selected = None
11     return None
12
13

```

Kode 3.8: Fungsi Mendeteksi Operasi Aktif

Penjelasan:

- Fungsi mendeteksi operasi matematika yang dipilih berdasarkan posisi jari telunjuk (koordinat x, y).
- Parameter x, y merupakan posisi jari telunjuk pada layar dalam piksel.
- Logika deteksinya jika jari telunjuk berada di dalam kotak menu operasi (dengan koordinat tetap di layar), simbol operasi dikembalikan

### 3.9 Fungsi calculate\_result

```

1  def calculate_result(left, right, operation):
2      global operation_processed, last_result
3      if operation == "*":
4          display_op = "X"
5      else:
6          display_op = operation
7
8      if operation_processed:
9          return last_result, display_op
10
11     left = int(left)
12     right = int(right)
13     result = None
14     if operation == "+":
15         result = left + right
16     elif operation == "-":
17

```

```

18     result = left - right
19     elif operation == "*" and right != 0:
20         result = left * right
21     elif operation == "/" and right != 0:
22         result = left / right
23     else:
24         result = "Error"
25         sound_error.play()
26     if result != "Error":
27         sound_success.play()
28
29     operation_processed = True
30     last_result = result
31     return result, display_op
32

```

Kode 3.9: Fungsi Perhitungan

Penjelasan:

- Fungsi menghitung hasil berdasarkan angka dari tangan kiri dan kanan serta operasi yang dipilih.
- Parameter left (angka dari tangan kiri), right (angka dari tangan kanan), dan operation (operasi matematika yang dipilih).
- Logika perhitungannya yaitu melakukan perhitungan sesuai dengan simbol operasi dan mengembalikan hasil operasi atau pesan "Error" jika operasi tidak valid (misalnya pembagian oleh nol).

### 3.10 Fungsi draw\_text\_with\_stroke

```

1     x, y = position
2     for dx, dy in [(-1, -1), (-1, 1), (1, -1), (1, 1)]:
3         cv2.putText(img, text, (x + dx, y + dy), font, scale, color_stroke, thickness_stroke)
4
5     cv2.putText(img, text, (x, y), font, scale, color_text, thickness_text)
6
7

```

Kode 3.10: Fungsi Menulis Teks dengan Stroke

Penjelasan:

- Fungsi ini digunakan untuk menulis teks dengan efek stroke pada gambar.
- Parameter:
- img: Gambar yang akan ditulis.
- text: Teks yang akan ditulis.
- position: Koordinat (x, y) untuk posisi teks.
- font: Font teks.
- scale: Skala teks.
- color\_stroke: Warna stroke teks.
- thickness\_stroke: Ketebalan stroke.
- color\_text: Warna teks.

- `thickness_text`: Ketebalan teks.
- Fungsi ini menulis teks dengan stroke menggunakan 4 arah (atas, bawah, kiri, kanan) untuk memberikan efek bayangan pada teks.
- Kemudian, teks utama ditulis di tengah dengan warna teks yang berbeda.
- Ini memberikan efek teks dengan stroke yang lebih menarik dan mudah dibaca.

### 3.11 Fungsi main

Membuka webcam untuk menangkap video real-time.

```
1 cap = cv2.VideoCapture(0)
2
3
```

Kode 3.11: Membuka Webcam

Membaca frame dari webcam dan memprosesnya secara terus-menerus hingga tombol keluar ditekan.

```
1 while cap.isOpened():
2     ret, frame = cap.read()
3     ...
4
5
```

Kode 3.12: Membaca Frame

Membalik frame secara horizontal untuk memberikan tampilan seperti cermin.

```
1 frame = cv2.flip(frame, 1)
2
3
```

Kode 3.13: Mirror Camera

Mengubah frame dari BGR ke RGB (MediaPipe memerlukan format RGB) dan menjalankan deteksi tangan dengan `hands.process(frame_rgb)`.

```
1 frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
2 results = hands.process(frame_rgb)
3
4
```

Kode 3.14: Konversi ke RGB

Landmark tangan digambar pada frame menggunakan MediaPipe Drawing Utilities

```
1 for idx, hand_landmarks in enumerate(results.multi_hand_landmarks):
2     mp_drawing.draw_landmarks(frame, hand_landmarks, mp_hands.HAND_CONNECTIONS)
3     ...
4
5
```

Kode 3.15: Menggambar Landmark

`detect_hand_sign` digunakan untuk mendeteksi angka pada tangan kiri dan kanan.

```
1 hand_sign = detect_hand_sign(hand_landmarks.landmark, handedness)
2
3
4
```

Kode 3.16: Proses Deteksi Nilai Tangan

Posisi jari telunjuk (landmark ke-8) digunakan untuk mendeteksi operasi yang dipilih dari menu.



```
1
2     selected_operation = detect_operation(x, y)
3
```

Kode 3.17: Pemilihan Operasi

Jika simbol tangan dan operasi terdeteksi, hasil perhitungan dihitung menggunakan `calculate_result`.

```
1
2     if active_operation and left_hand_signs_detected and right_hand_signs_detected:
3         result, display_op = calculate_result(left_hand_signs_detected[0],
4         right_hand_signs_detected[0], active_operation)
```

Kode 3.18: Proses Kalkulasi

Menampilkan hasil perhitungan, angka dari masing-masing tangan, dan operasi yang aktif pada layar

```
1
2     draw_text_with_stroke(frame, f"Hasil: {result}", (200, 100), ...)
3
```

Program keluar dari loop jika tombol "q" ditekan.

```
1
2     if cv2.waitKey(1) & 0xFF == ord("q"):
3         break
4
```

Kode 3.19: Escape Program

### 3.12 Eksekusi Program

Program akan memulai eksekusi dari fungsi `main`.

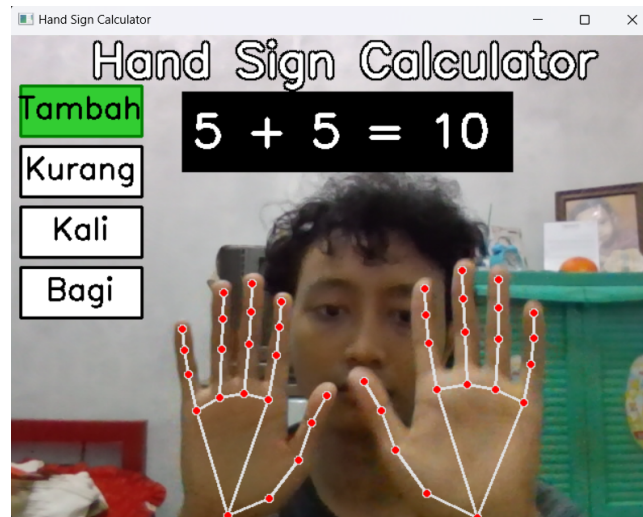
```
1
2     if __name__ == "__main__":
3         main()
4
```

Kode 3.20: Eksekusi Program

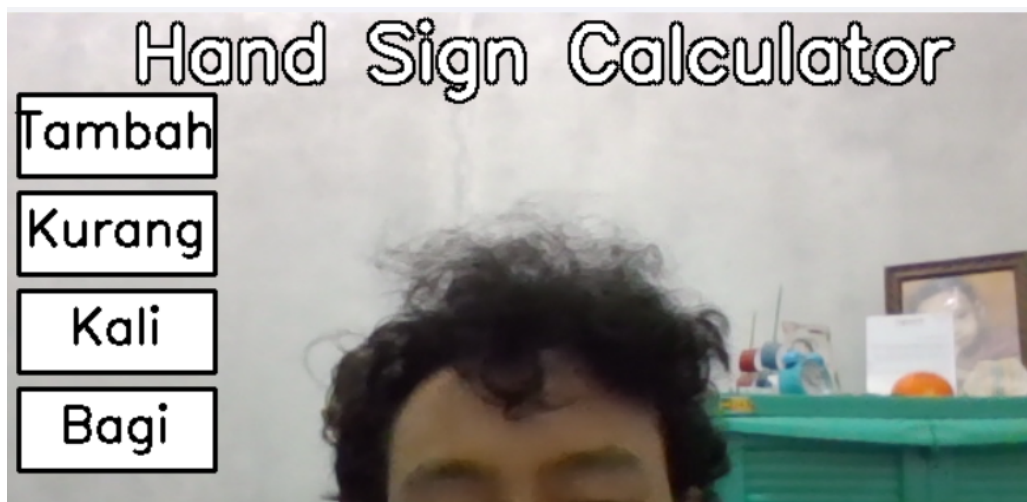
## 4 Hasil dan Pembahasan

### 4.1 Hasil

Program berhasil mendeteksi simbol angka ('0'-'5') dari pose tangan dan operasi matematika yang dipilih menggunakan jari telunjuk. Hasil perhitungan ditampilkan secara real-time di layar berdasarkan input dari gesture tangan. Program juga menyertakan efek suara untuk setiap interaksi, seperti memilih operasi matematika, hasil perhitungan, dan kesalahan (misalnya, pembagian dengan nol). Berikut adalah contoh hasil program saat digunakan:



Gambar 4.1: Hasil Run Program



Gambar 4.2: Antarmuka Menu Operasi

Terdapat menu operasi matematika yang dapat dipilih dengan menunjuk jari telunjuk ke simbol operasi yang diinginkan. Hasil perhitungan ditampilkan di layar berdasarkan input dari gesture tangan. Program juga menyertakan efek suara untuk meningkatkan interaktivitas dan memberikan umpan balik kepada pengguna. Ada 4 tombol operasi matematika yang dapat dipilih, yaitu penjumlahan, pengurangan, perkalian, dan pembagian. Tombol operasi matematika dapat dipilih dengan menunjuk jari telunjuk ke simbol operasi yang diinginkan. Hasil perhitungan ditampilkan di layar berdasarkan input dari gesture tangan. Program juga menyertakan efek suara untuk meningkatkan interaktivitas

dan memberikan umpan balik kepada pengguna.



Gambar 4.3: Deteksi Simbol Angka

Program dapat mendeteksi simbol angka ('0'-'5') berdasarkan pose tangan yang dibentuk. Setiap simbol angka diwakili oleh posisi jari tertentu yang membentuk simbol tersebut. Hasil perhitungan ditampilkan secara real-time di layar berdasarkan input dari gesture tangan. Program juga menyertakan efek suara untuk meningkatkan interaktivitas dan memberikan umpan balik kepada pengguna.

## 4.2 Pembahasan

Implementasi MediaPipe dan OpenCV memungkinkan program untuk mendeteksi landmark tangan dan mengenali simbol angka dengan akurat. Logika deteksi simbol angka didasarkan pada posisi jari tertentu yang membentuk simbol tertentu. Pemilihan operasi matematika menggunakan posisi jari telunjuk yang menunjuk ke menu operasi. Hasil perhitungan ditampilkan secara real-time di layar berdasarkan input dari gesture tangan. Program juga menyertakan efek suara untuk meningkatkan interaktivitas dan memberikan umpan balik kepada pengguna. Namun, program masih memiliki beberapa kekurangan, seperti sensitivitas terhadap pencahayaan dan latar belakang, serta keterbatasan dalam mendeteksi simbol angka yang kompleks. Untuk pengembangan selanjutnya, dapat dilakukan peningkatan dalam deteksi landmark tangan, pengenalan simbol angka yang lebih kompleks, dan integrasi dengan fitur lain seperti pengenalan gesture tambahan dan antarmuka pengguna yang lebih interaktif.

## 5 Cara Instalasi dan Penggunaan

### 5.1 Cara Instalasi

Clone repository ini.

```
1 git clone https://github.com/redywi/hand-sign-calculator.git
2 cd hand-sign-calculator
3
4
```

Kode 5.1: Clone Repository

Install dependencies.

```
1 pip install -r requirements.txt
2
3
```

Kode 5.2: Pemasangan Dependencies

### 5.2 Penggunaan

Jalankan program.

```
1 python main.py
2
3
```

Kode 5.3: Jalankan Program

Arahkan tangan ke kamera dan gunakan simbol berikut:

- Tangan kiri untuk angka pertama.
- Tangan kanan untuk angka kedua.
- Perlu diingat telapak tangan harus menghadap ke depan.
- Pilih operasi matematika dengan jari telunjuk.

Hasil perhitungan akan ditampilkan secara real-time di layar.

## References

- [1] S. Yadav and S. Jain, “Gesture recognition system for human-computer interaction using computer vision,” pp. 1–4, 2024.
- [2] M. L. Amit, A. C. Fajardo, and R. P. Medina, “Recognition of real-time hand gestures using mediapipe holistic model and lstm with mlp architecture,” pp. 292–295, 2022.
- [3] P. Padhi and M. Das, “Hand gesture recognition using densenet201-mediapipe hybrid modelling,” pp. 995–999, 2022.
- [4] Google, “Mediapipe face landmarker,” [https://ai.google.dev/edge/mediapipe/solutions/vision/face\\_landmarker?hl=id](https://ai.google.dev/edge/mediapipe/solutions/vision/face_landmarker?hl=id), accessed: 2024-12-14.