

## LAPORAN SEMENTARA PERTEMUAN KE 2

Nama : Yorrys Firdausi

Kelas : TKA-6D

NIM : 234308113

### A. Pendahuluan

Python termasuk bahasa pemrograman yang sangat direkomendasikan untuk pemula karena cara penulisannya sederhana dan mudah dipahami. Banyaknya komunitas dan sumber belajar juga membuat proses belajar jadi lebih mudah. Dibandingkan bahasa seperti C++ atau Java yang sering menuntut pemahaman teknis cukup rumit sejak awal, Python memberi kesempatan bagi pemula untuk fokus memahami logika dan konsep dasar pemrograman terlebih dahulu tanpa harus terbebani hal teknis yang kompleks (Belajarpython.com, 2025) (Maulana, 2024).

Menurut PYPI: *Popularity of Programming Language* (2026), Python menempati posisi pertama sebagai bahasa pemrograman paling populer. Hal ini menunjukkan bahwa Python banyak digunakan di berbagai bidang salah satunya adalah *machine learning*. Dengan adanya perkembangan teknologi, hal-hal yang semula dilakukan manusia, saat ini dapat dilakukan oleh mesin. Salah satunya adalah teknologi *computer vision* (*PYPL PopularitY of Programming Language Index*, 2026).

*Computer vision* adalah teknik yang digunakan komputer untuk memahami dan mengambil informasi penting dari gambar digital. Sistem ini bekerja dengan menerima gambar, lalu memprosesnya menggunakan pengolahan citra dan pengenalan pola. Selanjutnya, algoritma computer vision menganalisis gambar tersebut untuk mencapai tujuan atau tugas tertentu (Suradi & Rasyid, 2022). Dengan *computer vision*, mesin tidak hanya dapat mengidentifikasi suatu objek, namun juga dapat menganalisa suatu pergerakan objek tersebut.

Pada praktikum ini, library yang digunakan adalah mediapipe yang mendukung klasifikasi objek berbasis pengenalan pola (*landmark*). Mediapipe adalah framework yang dibuat untuk membantu pengembangan aplikasi berbasis kecerdasan buatan. Di dalamnya terdapat dukungan TensorFlow yang memungkinkan pemrosesan berjalan lebih cepat dengan memanfaatkan GPU (Graphics Processing Unit) maupun CPU pada perangkat (Zhang dkk., 2020).

MediaPipe Pose merupakan salah satu solusi dalam framework MediaPipe yang dirancang untuk mendeteksi dan melacak pose tubuh manusia secara *real-time* melalui gambar maupun video. Model yang digunakan, dikenal sebagai *BlazePose*, bekerja dengan mendeteksi tubuh terlebih dahulu kemudian mengidentifikasi titik-titik penting (*landmark*) secara detail pada bagian tubuh seperti kepala, bahu, siku, pergelangan tangan, pinggul, lutut, hingga pergelangan kaki. Secara keseluruhan, terdapat 33 landmark yang masing-masing memiliki informasi koordinat (x, y, z) serta tingkat visibilitas yang menunjukkan seberapa jelas titik tersebut terdeteksi. Keunggulan MediaPipe Pose terletak pada kemampuannya yang ringan namun tetap akurat sehingga dapat dijalankan secara efisien pada berbagai perangkat, baik menggunakan CPU maupun GPU, sehingga sangat cocok digunakan dalam aplikasi analisis gerakan, interaksi manusia-komputer, maupun sistem pembelajaran berbasis pengenalan pose tubuh (Remiro dkk., 2023) (Kim dkk., 2023).

#### B. Tujuan

1. Mengembangkan sistem deteksi tubuh manusia menggunakan MediaPipe dan OpenCV.
2. Mengidentifikasi suatu pose yang telah ditentukan.
3. Mendeteksi landmark tubuh untuk mengetahui titik-titik penting pada seluruh tubuh.
4. Menguji kinerja dan akurasi sistem deteksi pose tubuh yang dibuat.

#### C. Manfaat

1. Meningkatkan pemahaman tentang pengembangan sistem deteksi tubuh manusia.
2. Melatih kemampuan dalam mengenali dan menganalisis pose tertentu.

3. Memahami struktur dan titik-titik penting pada tubuh manusia.
4. Mengevaluasi kinerja dan akurasi sistem deteksi pose.

#### D. Kode Program Beserta Penjelasannya

##### 1. Deteksi Tubuh

```
import cv2
import mediapipe as mp
```

- Program diawali dengan mengimpor library OpenCV (cv2) dan MediaPipe. OpenCV digunakan untuk mengakses serta menampilkan video dari webcam, sedangkan MediaPipe digunakan untuk melakukan proses deteksi pose dan landmark tubuh manusia.

```
mpose = mp.solutions.pose #inisialisasi mediapipe pose
pose = mpose.Pose(static_image_mode=False,
model_complexity=1, smooth_landmarks=True,
enable_segmentation=False, smooth_segmentation=True,
min_detection_confidence=0.5, min_tracking_confidence=0.5)
```

- Bagian ini berfungsi untuk mengambil modul Pose dari MediaPipe dan menginisialisasi sistem deteksi pose.
- Parameter static\_image\_mode=False menunjukkan bahwa sistem bekerja dalam mode video (real-time).
- model\_complexity=1 digunakan untuk menyeimbangkan antara kecepatan pemrosesan dan tingkat akurasi.
- smooth\_landmarks=True mengaktifkan penghalusan titik landmark agar hasil deteksi lebih stabil.
- min\_detection\_confidence dan min\_tracking\_confidence diatur sebesar 0.5 untuk menentukan batas minimal tingkat kepercayaan sistem dalam mendeteksi dan melacak pose.

```
cap = cv2.VideoCapture(0) #video dari webcam 1
```

- Program mengaktifkan kamera utama (indeks 0) sebagai sumber input video yang akan diproses oleh sistem deteksi pose.

```

while True:
    success, img = cap.read() #pembacaan image
    imgrgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) #konversi
    warna dari bgr ke rgb
    hasil = pose.process(imgrgb) #ekstrasi dari image

```

- Program menggunakan perulangan untuk membaca frame dari webcam secara terus-menerus.
- Setiap frame yang diperoleh diubah format warnanya dari BGR ke RGB karena MediaPipe memproses citra dalam format RGB.
- Frame tersebut kemudian diproses menggunakan model Pose untuk mendeteksi keberadaan landmark tubuh.

```

if hasil.pose_landmarks :
    print("terdeteksi")
else:
    print("tidak terdeteksi")

```

- Sistem melakukan pengecekan apakah landmark tubuh berhasil dideteksi.
- Jika landmark terdeteksi, program menampilkan informasi “terdeteksi” pada terminal.
- Jika tidak ditemukan landmark, program menampilkan “tidak terdeteksi”.

```

cv2.imshow("Webcam", img)
cv2.waitKey(1)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

```

- Frame yang telah diproses ditampilkan pada jendela bernama “Webcam”.
- Program memberikan jeda singkat agar tampilan video berjalan dengan lancar.
- Program akan berhenti apabila tombol “q” ditekan.

```
cap.release() #tutup webcam dan jendela tampilan saat "q" ditekan  
cv2.destroyAllWindows()
```

- Setelah perulangan dihentikan, program menonaktifkan akses kamera dan menutup seluruh jendela tampilan untuk memastikan proses berhenti dengan baik.

## 2. Body Landmark

Beberapa baris kode yang sudah dijelaskan pada poin sebelumnya akan dilewati pada penjelasan kode program berikut.

```
pose = mpose.Pose()  
mdraw = mp.solutions.drawing_utils
```

- Sistem menginisialisasi kembali objek Pose dengan pengaturan standar (default).
- Program juga mengambil modul drawing\_utils dari MediaPipe yang berfungsi untuk menggambar titik landmark dan garis penghubung antar titik pada tubuh.

```
success, img = cap.read() #pembacaan image  
if not success:  
    break
```

- Program membaca frame dari webcam.
- Jika pembacaan frame gagal (success bernilai False), maka perulangan dihentikan untuk mencegah error akibat gambar yang tidak terbaca.

```
if hasil.pose_landmarks:  
    mdraw.draw_landmarks(img, hasil.pose_landmarks,  
    mpose.POSE_CONNECTIONS) #menggambar koneksi Landmark
```

- Program memeriksa apakah landmark tubuh berhasil terdeteksi.

- Jika terdeteksi, sistem akan menggambar titik-titik landmark beserta garis penghubungnya pada tubuh menggunakan POSE\_CONNECTIONS, sehingga struktur pose terlihat secara visual pada layar.

```
for id, lm in enumerate(hasil.pose_landmarks.landmark):
    print(id, lm.x, lm.y)
```

- Program melakukan perulangan untuk membaca seluruh landmark yang terdeteksi.
- id menunjukkan nomor indeks landmark (total 33 titik pada MediaPipe Pose).
- lm.x dan lm.y merupakan koordinat relatif (0–1) terhadap ukuran frame.
- Data ini ditampilkan pada terminal sebagai informasi posisi masing-masing titik tubuh.

```
cv2.imshow('webcam', img)
```

- Menampilkan frame yang sudah diberi gambar landmark dan garis koneksi, sehingga pengguna dapat melihat hasil deteksi pose secara langsung.

### 3. Deteksi Pose Angkat Tangan Kanan/Kiri

Beberapa baris kode yang sudah dijelaskan pada poin sebelumnya akan dilewati pada penjelasan kode program berikut.

```
mp_pose = mp.solutions.pose
pose = mp_pose.Pose()
mp_drawing = mp.solutions.drawing_utils
```

- Program mengambil modul Pose dari MediaPipe dan menginisialisasi sistem deteksi pose dengan pengaturan standar.
- Modul `drawing_utils` digunakan untuk menggambar titik landmark serta garis koneksi antar bagian tubuh pada frame video.

```
success, img = cap.read()
if not success:
    break
```

- Program membaca frame dari webcam secara terus-menerus.
- Jika frame gagal dibaca, perulangan dihentikan untuk mencegah terjadinya kesalahan pada proses berikutnya.

```
img = cv2.flip(img, 1)
```

- Frame dibalik secara horizontal (mirror effect) agar tampilan terlihat seperti bercermin.
- Hal ini membuat gerakan pengguna terasa lebih natural saat ditampilkan di layar.

```
imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
results = pose.process(imgRGB)
```

- Frame diubah dari format warna BGR ke RGB karena MediaPipe memproses gambar dalam format RGB.
- Gambar kemudian diproses untuk mendeteksi landmark tubuh.

```
if results.pose_landmarks:
    mp_drawing.draw_landmarks(img, results.pose_landmarks,
    mp_pose.POSE_CONNECTIONS)
```

- Program memeriksa apakah landmark tubuh terdeteksi.
- Jika terdeteksi, sistem menggambar titik-titik landmark beserta garis koneksi antar bagian tubuh pada frame.

- ```
landmarks = results.pose_landmarks.landmark
```
- Seluruh data landmark disimpan dalam variabel untuk memudahkan pengambilan koordinat titik tubuh tertentu.

```
# Ambil Landmark penting
left_shoulder =
landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value]
left_wrist = landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value]

right_shoulder =
landmarks[mp_pose.PoseLandmark.RIGHT_SHOULDER.value]
right_wrist =
landmarks[mp_pose.PoseLandmark.RIGHT_WRIST.value]
```

- Program mengambil landmark penting yaitu bahu dan pergelangan tangan kanan serta kiri.
- Landmark ini digunakan sebagai acuan untuk menentukan apakah tangan sedang terangkat atau tidak.

```
if left_wrist.y < left_shoulder.y:
    cv2.putText(img, "Tangan Kiri Terangkat", (30, 50),
                cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 255, 0),
                2)
```

- Sistem membandingkan koordinat vertikal (sumbu y) antara pergelangan tangan kiri dan bahu kiri.
- Jika nilai y pergelangan tangan lebih kecil dari bahu (artinya posisi lebih tinggi di layar), maka tangan kiri dianggap terangkat.

```
if right_wrist.y < right_shoulder.y:
    cv2.putText(img, "Tangan Kanan Terangkat", (30, 90),
                cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 255, 0),
                2)
```

- Prinsip yang sama diterapkan pada tangan kanan.
- Jika pergelangan tangan kanan berada lebih tinggi dari bahu kanan, maka tangan kanan dianggap terangkat.

`cv2.putText(...)`

- Jika kondisi tangan terangkat terpenuhi, program menampilkan teks pada layar sebagai indikator bahwa sistem berhasil mengenali pose tersebut.

`cv2.imshow("Deteksi Tangan Terangkat", img)`

- Menampilkan hasil deteksi pose beserta informasi teks pada jendela dengan judul sesuai program.

## E. Analisis

Secara keseluruhan, ketiga program menunjukkan tahapan pengembangan sistem deteksi pose tubuh menggunakan MediaPipe dan OpenCV secara bertahap. Program pertama berfokus pada proses dasar, yaitu membaca input dari webcam, memproses gambar menggunakan MediaPipe Pose, dan mengecek apakah landmark tubuh berhasil terdeteksi atau tidak. Pada tahap ini, sistem masih sebatas mendeteksi keberadaan pose tanpa menampilkan detail struktur tubuh. Program ini dapat dikatakan sebagai fondasi awal untuk memahami alur kerja deteksi pose secara real-time.

Program kedua merupakan pengembangan dari program pertama, di mana sistem tidak hanya mendeteksi keberadaan pose, tetapi juga menampilkan visualisasi landmark dan garis koneksi antar titik tubuh. Selain itu, koordinat setiap landmark diekstraksi dan ditampilkan pada terminal. Hal ini menunjukkan bahwa sistem sudah masuk ke tahap analisis data, karena posisi tiap titik tubuh dapat digunakan untuk kebutuhan lebih lanjut, seperti pengukuran sudut, analisis gerakan, atau klasifikasi pose tertentu. Pada tahap ini, pemahaman terhadap struktur 33 landmark pada MediaPipe Pose menjadi lebih jelas dan aplikatif.

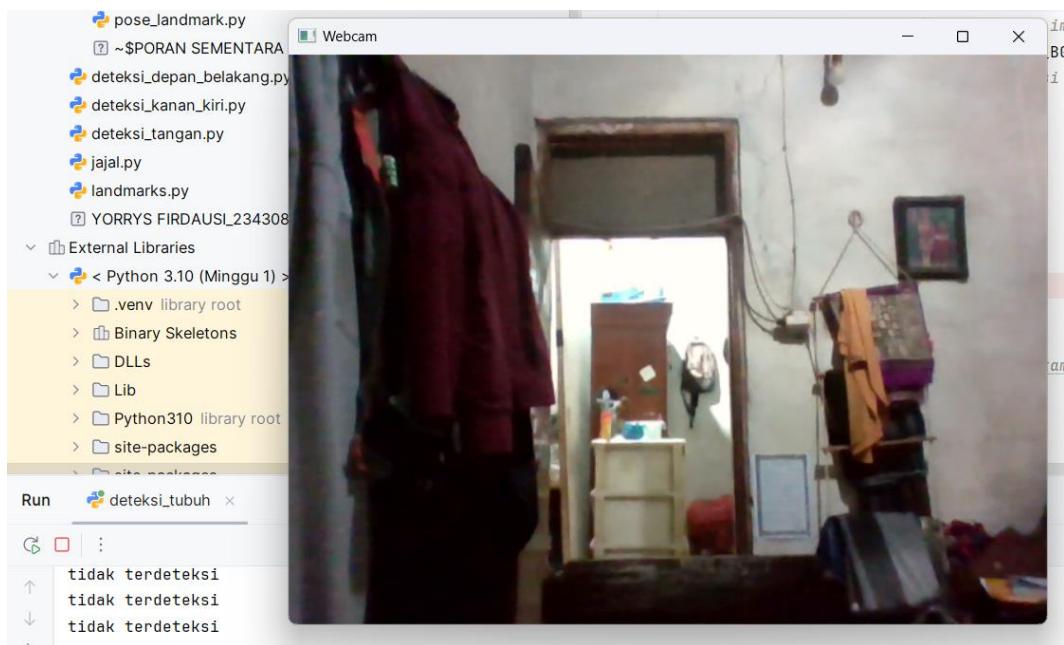
Program ketiga merupakan implementasi lanjutan yang lebih aplikatif, karena sistem sudah mampu mengenali pose tertentu, yaitu mendeteksi apakah tangan terangkat atau tidak. Logika yang digunakan cukup sederhana, yaitu membandingkan posisi vertikal (sumbu y) antara pergelangan tangan dan bahu. Jika pergelangan tangan berada lebih tinggi dari bahu, maka sistem menyimpulkan bahwa tangan terangkat. Penambahan fitur mirror (flip horizontal) juga membuat interaksi terasa lebih natural bagi pengguna. Secara keseluruhan, ketiga program tersebut menunjukkan perkembangan dari deteksi dasar, visualisasi landmark, hingga analisis pose sederhana berbasis koordinat.

Inti dari ketiga program tersebut adalah menunjukkan bagaimana data landmark yang dihasilkan oleh MediaPipe Pose dapat dimanfaatkan secara bertahap menjadi sebuah sistem yang lebih cerdas dan fungsional. Awalnya sistem hanya mendeteksi, kemudian memahami struktur tubuh, dan akhirnya mampu mengambil keputusan berdasarkan posisi titik-titik tertentu. Dengan kata lain, praktikum ini tidak hanya melatih penggunaan library, tetapi juga melatih cara berpikir logis dalam mengolah data hasil deteksi menjadi sebuah informasi atau aksi yang bermakna.

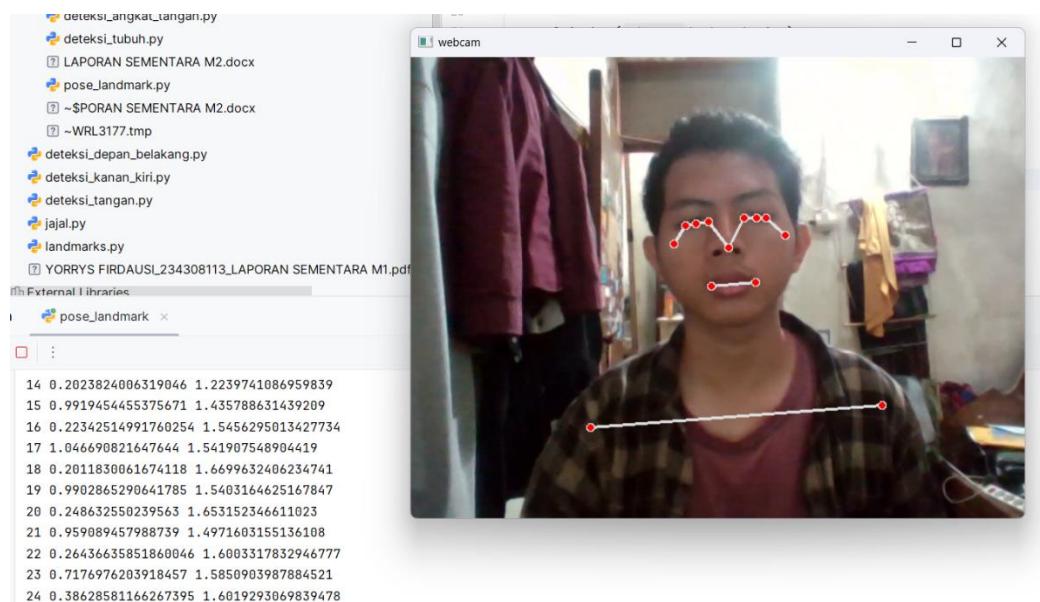
## F. Hasil



Gambar 1 Tubuh Terdeteksi



Gambar 2 Tubuh Tidak Terdeteksi



Gambar 3 Landmark pada Wajah



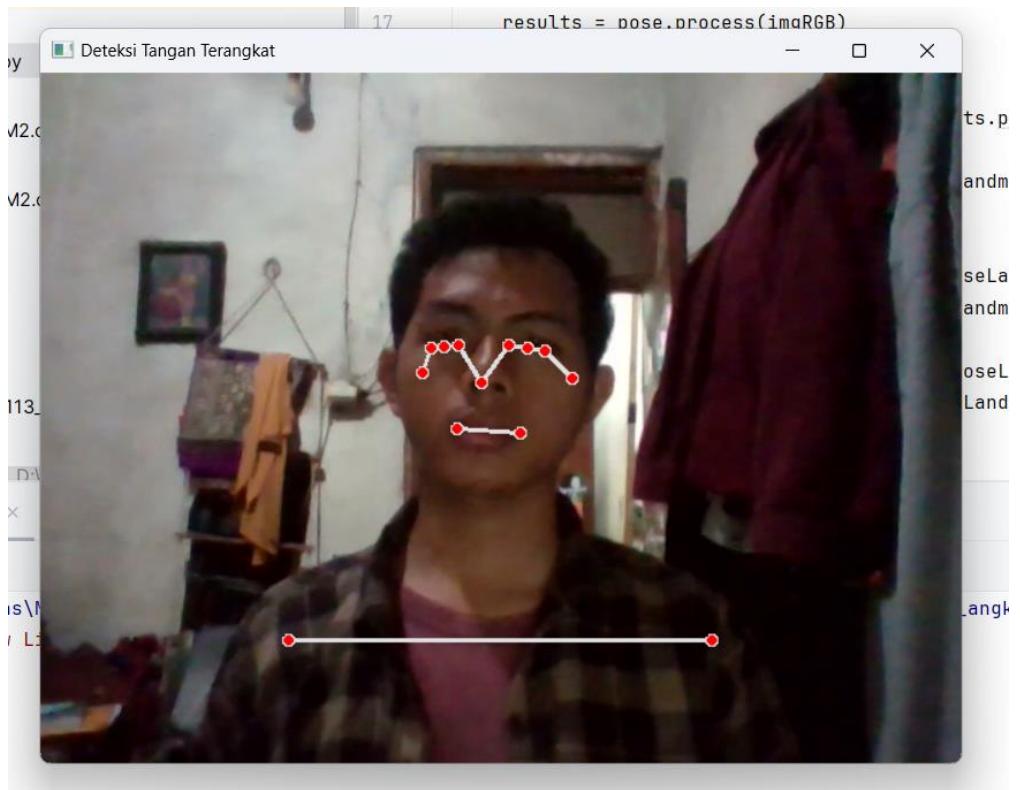
#### Gambar 4 Landmark pada Tubuh



Gambar 5 Deteksi Pose Tangan Kanan Terangkat



Gambar 6 Deteksi Pose Tangan Kiri Terangkat



Gambar 7 Pose Tidak Terdeteksi

## G. Kesimpulan

Setelah melakukan praktikum dan menganalisisnya, dapat diperoleh beberapa kesimpulan sebagai berikut :

1. Praktikum ini menunjukkan bahwa MediaPipe Pose dapat digunakan secara efektif untuk mendeteksi landmark tubuh manusia secara real-time melalui webcam dengan bantuan OpenCV sebagai pengolah citra.
2. Sistem mampu mengenali 33 titik landmark pada tubuh dan menampilkan struktur pose secara visual, sehingga memudahkan dalam memahami posisi serta pergerakan bagian tubuh.
3. Data koordinat landmark yang dihasilkan dapat dimanfaatkan untuk melakukan analisis pose sederhana, seperti mendeteksi kondisi tangan terangkat dengan membandingkan posisi pergelangan tangan dan bahu.
4. Proses pengembangan program dilakukan secara bertahap, mulai dari deteksi dasar, visualisasi landmark, hingga implementasi logika analisis pose, sehingga memberikan pemahaman yang sistematis tentang cara kerja computer vision berbasis pose detection.
5. Secara keseluruhan, praktikum ini tidak hanya melatih penggunaan library MediaPipe dan OpenCV, tetapi juga melatih kemampuan berpikir logis dalam mengolah data hasil deteksi menjadi informasi yang bermakna dan aplikatif.

## H. Referensi

- Belajarpython.com. (2025). *Tutorial Pemrograman Python Bahasa Indonesia | Dasar Sampai Advance*. Belajarpython.com. Belajarpython.com.  
<https://belajarpython.com/>
- Kim, J.-W., Choi, J.-Y., Ha, E.-J., & Choi, J.-H. (2023). Human Pose Estimation Using MediaPipe Pose and Optimization Method Based on a Humanoid Model. *Applied Sciences*, 13(4), 2700.  
<https://doi.org/10.3390/app13042700>
- Maulana, M. H. (2024). *Python Bahasa Pemograman Yang Ramah Bagi Pemula. 2.*
- PYPL PopularitY of Programming Language index. (2026).  
<https://pypl.github.io/PYPL.html>

Remiro, M. Á., Gil-Martín, M., & San-Segundo, R. (2023). Improving Hand Pose Recognition Using Localization and Zoom Normalizations over MediaPipe Landmarks. *ECSA 2023*, 69. <https://doi.org/10.3390/ecsa-10-16215>

Suradi, A. A. M., & Rasyid, M. F. (2022). *Sistem Perhitungan Jumlah Kendaraan Berbasis Computer Vision.* (1).

Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.-L., & Grundmann, M. (2020). *MediaPipe Hands: On-device Real-time Hand Tracking* (arXiv:2006.10214). arXiv. <https://doi.org/10.48550/arXiv.2006.10214>