



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**RANCANG BANGUN PENGHITUNG JUMLAH MANUSIA
DAN PENGAWASAN DALAM RUANGAN**

Sub Judul:

**SISTEM PENGHITUNG JUMLAH MANUSIA DALAM
RUANGAN MENGGUNAKAN ALGORITMA *BACKGROUND*
SUBSTRATION PADA *COMPUTER VISION***

**POLITEKNIK
NEGERI
JAKARTA**

Wahyuningrum

2203433014

PROGRAM STUDI INSTRUMENTASI KONTROL INDUSTRI

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI JAKARTA

2024



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**RANCANG BANGUN PENGHITUNG JUMLAH MANUSIA
DAN PENGAWASAN DALAM RUANGAN**

Sub Judul:

**SISTEM PENGHITUNG JUMLAH MANUSIA DALAM
RUANGAN MENGGUNAKAN ALGORITMA *BACKGROUND*
SUBSTRACTION PADA *COMPUTER VISION***

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar

Sarjana Terapan

**POLITEKNIK
NEGERI
JAKARTA**

Wahyuningrum

2203433014

PROGRAM STUDI INSTRUMENTASI KONTROL INDUSTRI

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI JAKARTA

2024

LEMBAR PERNYATAAN ORISINILITAS

Skripsi ini adalah hasil karya saya sendiri dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Wahyuningrum

NIM : 2203433014

Tanda Tangan :



Tanggal : 4 Februari 2024



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

LEMBAR PENGESAHAN

SKRIPSI

Skripsi Diajukan Oleh:

Nama : Wahyuningrum
NIM : 2203433014
Program Studi : Instrumentasi dan Kontrol Industri
Judul Tugas Akhir : Sistem Penghitung Jumlah Manusia Dalam Ruangan Menggunakan Algoritma *Background Substraction* Pada *Computer Vision*

Telah diuji oleh tim penguji dalam Sidang Skripsi pada 5 Februari 2024 dan dinyatakan **LULUS**.

Pembimbing I : Hariyanto, S.Pd., M.T.

NIP.199101282020121008

Depok, 12 Februari 2024

Disahkan oleh



Ketua Jurusan Teknik Elektro

Rika Novita Wardhani, S.T., M.T.

NIP. 197011142008122001



KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan skripsi ini. Penulisan Skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Terapan, Politeknik Negeri Jakarta, Jurusan Teknik Elektro, Program Studi Instrumentasi dan Kontrol Industri. Skripsi ini berjudul “Rancang Bangun Penghitung Jumlah Manusia dan Pengawasan Dalam Ruang” dengan sub judul “Sistem Penghitung Jumlah Manusia Dalam Ruang Menggunakan Algoritma *Background Substraction* Pada *Computer Vision*”. Disadari tanpa bantuan dan bimbingan dari berbagai pihak, penyusunan skripsi ini tidak akan selesai tepat pada waktunya. Oleh karena itu, diucapkan terima kasih kepada:

1. Rika Novita Wardhani, S.T., M.T, selaku Ketua Jurusan Teknik Elektro;
2. Sulis Setiowati, S.Pd., M.Eng, selaku Ketua Program Studi Instrumentasi dan Kontrol Industri;
3. Hariyanto, S.Pd., M.T. selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan penulis dalam penyusunan skripsi ini;
4. Orangtua yang telah memberikan bantuan dukungan material dan moral;
5. Difna Yasmika, selaku rekan satu tim dalam pelaksanaan skripsi ini yang telah mendukung, membantu, dan memotivasi dalam menyelesaikan laporan Skripsi;
6. Bella selaku teman baik yang selalu bersedia dimintakan pertolongan.

Akhir kata, semoga Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu dan semoga Tugas Akhir ini membawa manfaat bagi pengembangan ilmu pengetahuan. Kritik serta saran yang membangun sangat diharapkan untuk perbaikan dan penyempurnaannya.

Depok, 4 Februari 2024

Penulis



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR ISI

LEMBAR PERNYATAAN ORISINILITAS	iii
LEMBAR PENGESAHAN	iv
KATA PENGANTAR	v
Abstrak	vi
<i>Abstract</i>	vii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
BAB I	1
1. 1 Latar Belakang	1
1. 2 Rumusan Masalah	2
1. 3 Tujuan	2
1. 4 Batasan Masalah	3
1. 5 Luaran	3
BAB II	4
2. 1 <i>State of the Art</i>	4
2. 2 <i>Computer Vision</i>	5
2. 3 <i>OpenCV</i>	7
2. 4 Metode <i>Background Substraction</i>	8
2. 5 Bahasa Pemrograman <i>Python</i>	12
2. 6 <i>Raspberry Pi 4</i> model B	13
2. 7 <i>Webcam</i>	14
2. 8 Layar Monitor	15
BAB III	16
3. 1 Rancangan Alat	16
3. 1. 1 Deskripsi Alat	16
3. 1. 2 Cara Kerja Rancang Bangun Penghitung Jumlah Manusia Dan Pengawasan Dalam Ruang	16
3. 1. 3 Cara Kerja Sistem Penghitung Jumlah Manusia Dalam Ruang Menggunakan Algoritma <i>Background Substraction</i> Pada <i>Computer Vision</i>	18



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

3. 1. 4	Spesifikasi Alat	20
3. 1. 5	Spesifikasi Perangkat Lunak (<i>Software</i>)	21
3. 1. 6	Diagram Blok	21
3. 2	Realisasi Sistem	22
3. 2. 1	Instalasi Python dan <i>Library</i> pada Raspberry Pi 4	22
3. 2. 2	Program Inisialisasi Awal	23
3. 2. 3	Membuat Garis Pembatas Atas dan Bawah	23
3. 2. 4	Program Loop Deteksi Pergerakan Menggunakan Background Substraction	25
3. 2. 5	Program Contour Objek Terdeteksi	26
3. 2. 6	Program Penghitung Jumlah Manusia yang Masuk dan Keluar	27
BAB IV	29
4. 1	Pengujian Webcam Untuk Menghitung Manusia Masuk dan Keluar	29
4. 1. 1	Deskripsi Pengujian	29
4. 1. 2	Daftar Peralatan yang digunakan	29
4. 1. 3	Prosedur Pengujian	30
4. 1. 4	Data Hasil Pengujian	30
4. 1. 5	Analisa Perhitungan Jumlah Manusia Masuk dan Keluar Sebanyak 1 (Satu) Manusia	38
4. 1. 6	Analisa Perhitungan Jumlah Manusia Masuk dan Keluar Sebanyak 2 (Satu) Manusia yang Berjalan Berhadapan	39
4. 1. 7	Analisa Perhitungan Jumlah Manusia Masuk dan Keluar Sebanyak 2 (Satu) Manusia yang Berjalan Berdampingan	41
BAB V	43
5. 1	Kesimpulan	43
5. 2	Saran	44
DAFTAR PUSTAKA	45
LAMPIRAN	L-1



DAFTAR GAMBAR

Gambar 2. 1 Hasil pengolahan <i>Computer Vision</i>	6
Gambar 2. 2 Struktur OpenCV.....	7
Gambar 2. 3 Hasil Background Substraction.....	8
Gambar 2. 4 Penerapan RGB menjadi <i>Grayscale</i>	10
Gambar 2. 5 Penerapan Background Modeling.....	10
Gambar 2. 6 Penerapan Foreground Detection.....	11
Gambar 2. 7 Penerapan Thresholding.....	12
Gambar 2. 8 Logo Python.....	13
Gambar 2. 9 Raspberry Pi 4 Model B.....	14
Gambar 2. 10 Webcam.....	15
Gambar 2. 11 Layar Monitor.....	15
Gambar 3. 1 Flowchart Cara Kerja Alat Keseluruhan.....	18
Gambar 3. 2 Flowchart Cara Kerja Sistem Sub Bab.....	20
Gambar 3. 3 Blok Diagram Sistem.....	21
Gambar 3. 4 Program Inisialisasi.....	23
Gambar 3. 5 Program Pembagi layar.....	24
Gambar 3. 6 Tampilan Layar Monitor.....	24
Gambar 3. 7 Program Loop Background Substraction.....	25
Gambar 3. 8 Tampilan Deteksi Pergerakan Menggunakan Background Substraction.....	25
Gambar 3. 9 Program Contour.....	26
Gambar 3. 10 Tampilan saat Contour Objek Terdeteksi.....	26
Gambar 3. 11 Program Penghitung Jumlah Manusia yang Masuk dan Keluar..	27
Gambar 3. 12 Realisasi Alat.....	28
Gambar 3. 13 Tampilan Awal Layar Monitor.....	28
Gambar 4. 1 Hasil Deteksi Perhitungan Jumlah Manusia Masuk dan Keluar Sebanyak 1 Manusia.....	39
Gambar 4. 2 Tampilan Layar Monitor Perhitungan Masuk dan Keluar Sebanyak 1 Manusia.....	39
Gambar 4. 3 Hasil Perhitungan Jumlah Manusia Masuk dan Keluar Sebanyak 2 (Satu) Manusia yang Berjalan Berhadapan.....	41
Gambar 4. 4 Perhitungan Jumlah Manusia Masuk dan Keluar Sebanyak 2 (Satu) Manusia yang Berjalan Berdampingan.....	42
Gambar 4. 5 Tampilan Layar Monitor Perhitungan Jumlah Masuk dan Keluar sebanyak 2 Manusia.....	42
Gambar L 1. Tampilan Alat.....	L-2
Gambar L 2 Tampilan Layar Monitor.....	L-2

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



DAFTAR TABEL

Tabel 2. 1 Penelitian terdahulu oleh Khairul dan Benny	4
Tabel 2. 2 Penelitian terdahulu oleh Monica, Aghus dan Achmad.....	4
Tabel 2. 3 Penelitian terdahulu oleh Afifudin, Gede dan Wahyu	5
Tabel 2. 4 Jurnal terdahulu oleh Ari dan Sularso.....	5
Tabel 2. 5 Spesifikasi Raspberry Pi 4 Model B	13
Tabel 2. 6 Spesifikasi Webcam.....	14
Tabel 2. 7 Spesifikasi Layar Monitor.....	15
Tabel 4. 1 Daftar Peralatan.....	29
Tabel 4. 2 Pengujian Perhitungan Masuk Ruang sebanyak 1 Manusia.....	31
Tabel 4. 3 Pengujian Perhitungan Keluar Ruang sebanyak 1 Manusia.....	32
Tabel 4. 4 Pengujian Perhitungan Masuk Ruang Sebanyak 1 Manusia oleh 2 Manusia Berjalan Secara Berhadapan.....	33
Tabel 4. 5 Pengujian Perhitungan Keluar Ruang Sebanyak 1 Manusia oleh 2 Manusia Berjalan Secara Berhadapan.....	34
Tabel 4. 6 Pengujian Perhitungan Masuk sebanyak 2 Manusia.....	36
Tabel 4. 7 Pengujian Perhitungan Keluar sebanyak 2 Manusia.....	37

**POLITEKNIK
NEGERI
JAKARTA**

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



SISTEM PENGHITUNG JUMLAH MANUSIA DALAM RUANGAN MENGUNAKAN ALGORITMA BACKGROUND SUBTRACTION PADA COMPUTER VISION

Abstrak

Dilatarbelakangi oleh permasalahan kecurangan dalam absensi kelas serta menerapkan perkembangan teknologi dalam bidang computer vision, maka dibuatlah sistem penghitung jumlah manusia masuk dan keluar serta deteksi wajah. Pendeteksian objek manusia mengadaptasi cara penglihatan manusia yang saat ini dikenal dengan teknologi Computer Vision. Penelitian ini bertujuan untuk dapat menghitung jumlah manusia yang berjalan masuk dan keluar dalam ruangan. Dengan menerapkan algoritma Background Substraction untuk mengekstraksi citra gambar. Model Background Substraction terdiri dari 4, yaitu: Preprocessing pada Background Substraction mencakup serangkaian langkah untuk meningkatkan kualitas citra dan membantu dalam deteksi objek yang bergerak; Modeling melibatkan penciptaan dan pemeliharaan model latar belakang yang mewakili elemen-elemen statis dari suatu adegan; Foreground Detection pada dasarnya dihitung dengan mengambil perbedaan intensitas citra dengan background gambar; dan Thresholding adalah metode menetapkan suatu nilai, kemudian mengambil nilai lebih dari atau kurang dari nilai yang ditetapkan. Sistem akan menghitung objek manusia yang masuk dan keluar ruangan dengan memantau koordinat dari objek manusia tersebut. Sistem dibuat menggunakan bahasa pemrograman python dengan library OpenCV. Input diperoleh dari sebuah webcam dan mikrokomputer menggunakan Raspberry Pi 4 Model B sebagai pemroses data dari webcam. Hasil output penghitung jumlah manusia ditampilkan pada layar monitor dan tersimpan di file berformat txt. Sedangkan hasil output deteksi wajah ditampilkan di layar monitor dan tersimpan dalam format jpg yang tersimpan didalam sebuah folder penyimpanan. Tingkat akurasi dipengaruhi oleh ketinggian tiang penyangga Webcam, kecepatan berjalan manusia, pencahayaan ruangan, kemiringan peletakan sudut Webcam, dan banyaknya manusia yang melintas. Hasil model percobaan pengujian memberikan akurasi berhasil sebesar 93,75% pada pengujian perhitungan masuk dan keluar sebanyak 1 manusia, 81,25% pada pengujian perhitungan masuk dan keluar 2 manusia yang berjalan secara berhadapan.

Kata Kunci: Raspberry Pi 4 Model B, Webcam, OpenCV, Background Substraction, Computer Vision

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



COUNTING THE NUMBER OF PEOPLE USING BACKGROUND SUBTRACTION IN COMPUTER VISION

Abstract

Problem of cheating in class attendance and implementing technological developments in the field of computer vision, a system for counting the number of people entering and leaving as well as face detection was created. Human object detection adapts human vision, which is currently known as Computer Vision technology. This research aims to be able to count the number of people walking in and out of a room. By applying the Background Subtraction algorithm to extract image images. The Background Subtraction model consists of 4, namely: Preprocessing in Background Subtraction includes a series of steps to improve image quality and assist in the detection of moving objects; Modeling involves creating and maintaining background models that represent the static elements of a scene; Foreground Detection is basically calculated by taking the difference in image intensity with the image background; and Thresholding is a method of setting a value, then taking a value that is more than or less than the set value. The system will count human objects entering and leaving the room by monitoring the coordinates of the human objects. The system was created using the Python programming language with the OpenCV library. Input is obtained from a webcam and a microcomputer using a Raspberry Pi 4 Model B as a data processor from the webcam. The output results of the human count are displayed on the monitor screen and saved in a TXT format file. Meanwhile, the facial detection output results are displayed on the monitor screen and saved in jpg format which is stored in a storage folder. The level of accuracy is influenced by the height of the Webcam support pole, human walking speed, room lighting, the slope of the Webcam angle, and the number of people passing by. The results of the experimental testing model provide a successful accuracy of 93,75% in the entry and exit calculation test of 1 human, 81,25% in the entry and exit calculation test of 2 humans walking face to face.

Keywords: *Raspberry Pi 4 Model B, Webcam, OpenCV, Background Subtraction, Computer Vision*

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB I

PENDAHULUAN

1.1 Latar Belakang

Penghitung jumlah manusia dalam ruangan kelas merupakan sistem yang berguna bagi institusi dalam mengelola kehadiran mahasiswa dengan lebih efisien dan efektif. Absensi dalam kelas menjadi indikator penting bagi peserta didik, dosen, dan sistem pendidikan secara keseluruhan. Kehadiran dalam kelas sering menjadi faktor dalam penilaian akademik dan evaluasi kinerja mahasiswa. Kecurangan dalam absensi kelas adalah praktik yang melibatkan manipulasi atau pelanggaran terhadap proses pencatatan kehadiran mahasiswa di dalam kelas, terlebih jika proses pencatatan masih dilakukan secara manual. Kecurangan dalam absensi kelas sering terjadi dalam bentuk penandatanganan untuk teman yang tidak hadir atau rekaman kehadiran palsu. Hal ini mengakibatkan data absensi yang tidak akurat dan dapat memberikan keuntungan yang tidak pantas bagi mahasiswa yang melakukan kecurangan. Dalam konteks keberlanjutan, pemantauan jumlah manusia dalam suatu ruangan dapat membantu dalam mengelola penggunaan energi misalnya, pencahayaan, pengaturan suhu, dan sistem pendingin udara dapat diatur secara otomatis berdasarkan kehadiran jumlah manusia untuk menghemat energi.

Dengan menggunakan sistem penghitung jumlah manusia, kecurangan absensi dapat dicegah. Sistem ini dapat secara otomatis mendeteksi kehadiran mahasiswa saat masuk dan keluar dari kelas. Dengan demikian, mahasiswa tidak dapat memalsukan kehadiran mereka atau melakukan kecurangan lainnya terkait absensi kelas. Secara umum cara kerja dari deteksi dan penghitung objek manusia ini mengadaptasi dari cara pengamatan apa yang ada pada dunia nyata melalui indra pengelihat. Dengan kata lain, pendeteksian objek manusia dengan menggunakan indra penglihatan manusia (*Human Vision*) yang saat ini dikenal dengan teknologi *Computer Vision* (Solichin & Harjoko, 2013). *OpenCV* pada *Python* merupakan sebuah pustaka perangkat lunak untuk pengolahan citra dinamis.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Background Substraction adalah teknik dalam pemrosesan citra dan komputer visi yang digunakan untuk mengidentifikasi objek atau pergerakan dalam suatu scene dengan cara menghapus latar belakang dari citra tangkapan *Webcam* secara realtime. Sistem penghitung jumlah manusia dan deteksi wajah dalam suatu ruangan dapat memproses dan mendeteksi manusia yang bergerak masuk dan keluar pada jalur yang sama. Digunakan metode *Background Substraction* dengan menggunakan *library OpenCV* yang akan ditangkap oleh *Webcam*. Metode *Background Substraction* ini akan memisahkan objek dengan citra latar belakang dengan melakukan pengurangan citra sederhana pada area RGB (*Red, Green, Blue*) dan *grayscale*. Metode *Background Substraction* tidak akan mendeteksi objek yang ukurannya lebih kecil dari manusia seperti hewan, pohon, kursi, dll karena dilakukan pengaturan luas ukuran objek. Proses *Background Substraction* bekerja dengan mengurangi *frame* terbaru dengan *frame* sebelumnya. Kemudian hasil pengurangan tersebut di *threshold* untuk mengidentifikasi objek *foreground*. Untuk pemrosesan data digunakan mikrokomputer *Raspberry Pi 4* model B. Hasil output perhitungan jumlah masuk dan keluar manusia akan ditampilkan pada layar monitor dan tersimpan disebuah file berformat *.txt* dan deteksi wajah akan tersimpan dalam sebuah folder.

1.2 Rumusan Masalah

Perumusan masalah dapat diuraikan sebagai berikut:

- 1) Bagaimana membangun Sistem Penghitung Jumlah Manusia Menggunakan Algoritma *Background Substraction*?
- 2) Bagaimana menghitung jumlah objek manusia yang terdeteksi memasuki ruangan atau keluar ruangan?

1.3 Tujuan

Berdasarkan latar belakang yang dijelaskan dan sesuai tujuan dari penulisan ini, maka pokok permasalahan yang akan dibahas berupa:

- 1) Terbangunya sistem penghitung jumlah manusia menggunakan algoritma *Background Substraction*.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- 2) Mengetahui cara menghitung jumlah manusia yang terdeteksi memasuki ruang atau keluar ruang.

1.4 Batasan Masalah

Dengan luasnya cakupan pembahasan, maka perlu dibatasi agar dapat fokus pada pembahasan yang ada. Ruang lingkup pembahasan adalah sebagai berikut:

- 1) Perangkat lunak yang digunakan adalah *Python, OpenCV*.
- 2) Perangkat keras yang digunakan adalah *Raspberry Pi 4* model B dengan kapasitas RAM 4 GB dan memori eksternal 32 GB.
- 3) Pengujian hanya dilakukan dengan ketinggian tiang tripod penyangga *Webcam* setinggi kurang lebih 270 cm.
- 4) Algoritma yang digunakan adalah hanya algoritma *Background Substraction*.
- 5) Sistem ini dapat menampilkan perhitungan jumlah manusia yang masuk dan keluar ruang menggunakan kamera *Webcam* dengan hasil output ditampilkan pada layar monitor.
- 6) Tidak mengklasifikasikan jenis kelamin objek manusia, jenis objek, dll.
- 7) Pengujian hanya menggunakan *Webcam* dengan resolusi 1080P HD.
- 8) Pengujian dilakukan dengan keadaan kondisi ruangan yang tenang.
- 9) Pengujian hanya dilakukan dengan objek berjalan sebesar 30cm/s, tidak berlari.
- 10) Pengujian hanya dilakukan dengan sudut kemiringan *Webcam* sebesar 90° dan 40°.
- 11) Pengujian hanya dilakukan oleh maksimal 2 manusia yang melintas pada saat percobaan.

1.5 Luaran

- 1) Luaran Wajib

Sistem Penghitung Jumlah Manusia Dalam Ruangan Menggunakan Algoritma *Background Substraction* Pada *Computer Vision*.
- 2) Laporan Akhir
 - Luaran Tambahan
 - Publikasi Media Sosial

BAB V

KESIMPULAN

5.1 Kesimpulan

Berdasarkan pembuatan dan percobaan implementasi sistem penghitung jumlah manusia dalam ruangan diperoleh kesimpulan sebagai berikut:

1. Perancangan Sistem Penghitung Jumlah Manusia Dalam Ruangan Menggunakan Metode *Background Substraction* dengan tahapan *preprocessing*, *background modeling*, *foreground detection*, dan *thresholding*. Hasil proses *preprocessing* yaitu mengubah inputan yang memiliki citra awal RGB diubah menjadi citra *grayscale*. Hasil proses *background modeling* yaitu untuk menetapkan salah satu frame pada inputan menjadi sebuah *background*. Hasil proses *foreground detection* yang membedakan semua *frame* pada inputan dengan *background* model untuk menghasilkan perbedaan citra. Pada proses *thresholding* yaitu dimana hanya mengambil nilai matriks pada gambar citra yang lebih besar dari nilai yang sudah di tetapkan oleh peneliti untuk menghasilkan gambar citra biner yang baik.
2. Sistem menggunakan *Webcam* dan *Raspberry Pi 4* telah memberikan hasil yang cukup akurat. Pengaturan peletakan kemiringan sudut dan ketinggian tiang serta kecepatan berjalan menjadi titik yang perlu diperhatikan. Percobaan pengujian perhitungan lebih dari satu manusia belum menghasilkan hasil yang optimal.
3. Tingkat akurasi pengujian perhitungan masuk dan keluar sebanyak 1 manusia memberikan hasil sebesar 93,75 % dengan ketinggian tiang tripod sebesar 270cm dan kecepatan berjalan sebesar 30cm/s.
4. Tingkat akurasi pengujian perhitungan masuk dan keluar sebanyak 1 manusia oleh 2 manusia yang berjalan berhadapan memberikan hasil sebesar 81,25 % dengan ketinggian tiang tripod sebesar 270cm dan kecepatan berjalan sebesar 30cm/s.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

5. Tingkat akurasi pengujian perhitungan masuk dan keluar sebanyak 2 manusia yang berjalan berdampingan memberikan hasil sebesar 65% dengan sudut ketinggian tiang tripod sebesar 270cm dan kecepatan berjalan sebesar 30cm/s.
6. Tingkat peletakkan sudut kemiringan pada *Webcam* mempengaruhi akurasi perhitungan. Sudut kemiringan 40° lebih optimal dalam perhitungan jumlah manusia masuk dan keluar dalam ruangan.

5.2 Saran

Sistem penghitung manusia ini masih memiliki beberapa kekurangan dan keterbatasan. Oleh karena itu, terdapat beberapa hal yang perlu dipertimbangkan untuk pengembangan sistem ini agar lebih baik, yaitu sebagai berikut:

1. Sistem belum mampu mengklasifikasikan objek inputan yang melintas sebagai manusia atau benda lain. Diharapkan untuk penelitian selanjutnya dapat menerapkan algoritma untuk membedakan manusia atau benda lain agar penghitung jumlah manusia masuk dan keluar lebih akurat.
2. Disarankan penggunaan input tangkapan layar menggunakan kamera *Wifi* agar tangkapan layar dapat menjangkau lebih luas, yang nantinya dapat memperbaiki tingkat akurasi dalam perhitungan jumlah 2 manusia masuk dan keluar.
3. Diharapkan penelitian selanjutnya dapat menggunakan mikrokomputer dengan processor yang lebih baik, agar hasil pendeteksian manusia tidak terdapat delay yang akan mempengaruhi keakuratan perhitungan.



DAFTAR PUSTAKA

- Afifudin, M., Pratama, G. A., & J.S, W. S. (2022). Deteksi Ruang Parkir Menggunakan OpenCV.
- Andre, J. A. (2016). Sistem Security Webcam Dengan Menggunakan Microsoft Visual Basic. *STMIK Amik Riau*.
- Foundation, T. R. (2020). *The Raspberry Pi Foundation*. Retrieved from raspberrypi.org
- Hariyanto, M. S., Sofwan, A., & Hidayatno, A. (2018). Perancangan Sistem Penghitung Jumlah Kendaraan Pada Area Parkir dengan Metode Background Substraction Berbasis Internet Of Things.
- Jainita, U. (2022). *Puluhan penonton pingsan di acara "Berdendang Bergoyang"*. Retrieved from <https://www.antaraneews.com/>
- Kandir, N. (2022). *Mengenal OpenCV dan Python*. Retrieved from <https://norkandirblog.wordpress.com>
- Khairul Umam, B. S. (2016). Deteksi Obyek Manusia Pada Basis Data Video Menggunakan Metode Background Substraction Dan Operasi Morfologi. *Riau: informatika*.
- Outsiah, S. (2016). APLIKASI PEMBELAJARAN MATEMATIKA DASAR BANGUN DATAR MENGGUNAKAN PYTHON PADA PERANGKAT BERGERAK'.
- Ramadan, A. K., & Laksono, S. B. (2023). Rancang Bangun Aplikasi Deteksi Objek Untuk Menghitung Jumlah Pengunjung Restoran Berbasis Computer Vision. *IKRAITH-INFORMATIKA*.
- Solichin, A., & Harjoko, A. (2013). Metode Background Substraction untuk Deteksi Obyek Pejalan Kaki pada Lingkungan Statis. *SNATI*.
- Umam, K., & Negara, B. S. (2016). *Deteksi Obyek Manusia Pada Basis Data Video Menggunakan Metode Background Substraction dan Operasi Morfologi*.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



© Hak Cipta milik Politeknik Negeri Jakarta

LAMPIRAN 1

LAMPIRAN

DAFTAR RIWAYAT HIDUP PENULIS



Wahyuningrum

Lahir dan dibesarkan di Kota Depok sejak 14 Maret 2000. Sekolah dasar di SDN Kalibaru 3, Sekolah Menengah Pertama di SMPN 16 Depok dimana sewaktu masuk baru menginjak tahun ajaran ke-4 di sekolah tersebut. Sekolah Menengah Atas di SMAN 8 Depok pada Angkatan ke-5 di sekolah tersebut. Gelar Diploma Tiga diperoleh tahun 2021 dari Jurusan Teknik Elektro Politeknik Negeri Jakarta. Melanjutkan pendidikan Sarjana di Politeknik Negeri Jakarta dengan Prodi Instrumentasi dan Kontrol Industri.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



© Hak Cipta milik Politeknik Negeri Jakarta

LAMPIRAN 2

FOTO ALAT



Webcam Penghitung
Jumlah Manusia

Webcam Deteksi
Wajah

Gambar L 1. Tampilan Alat



Gambar L 2 Tampilan Layar Monitor

- Hak Cipta :**
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



LAMPIRAN 3

PENGHITUNG JUMLAH MANUSIA DAN PENGAWASAN DALAM RUANGAN

```

import numpy as np
import cv2
import Person
import time
import os
from tensorflow.keras.models import load_model

try:
    log = open('log.txt', "w")
except:
    print( "Tidak dapat membuka file log")

#Penghitung masuk dan keluar
cnt_up = 0
cnt_down = 0

#sumber video
cap = cv2.VideoCapture(2)

face_cascade =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
np.set_printoptions(suppress=True)

output_folder = "detected_faces"
if not os.path.exists(output_folder):
    os.makedirs(output_folder)

model = load_model("model1.h5")
labels = open("modelcnn-3.txt", "r").readlines()
print("[INFO] Model berhasil dimuat...")

camera = cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_COMPLEX
camera.set(cv2.CAP_PROP_FRAME_WIDTH, 480)
camera.set(cv2.CAP_PROP_FRAME_HEIGHT, 320)

for i in range(19):
    print( i, cap.get(i))

h = 480
w = 640
frameArea = h*w
areaTH = frameArea/6
print( 'Area Threshold', areaTH)

```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
#jalur masuk/keluar
line_up = int(2*(h/5))
line_down = int(3*(h/5))

up_limit = int(1*(h/5))
down_limit = int(4*(h/5))

print( "Red line y:",str(line_down))
print( "Blue line y:", str(line_up))
line_down_color = (255,0,0)
line_up_color = (0,0,255)
pt1 = [0, line_down];
pt2 = [w, line_down];
pts_L1 = np.array([pt1,pt2], np.int32)
pts_L1 = pts_L1.reshape((-1,1,2))
pt3 = [0, line_up];
pt4 = [w, line_up];
pts_L2 = np.array([pt3,pt4], np.int32)
pts_L2 = pts_L2.reshape((-1,1,2))

pt5 = [0, up_limit];
pt6 = [w, up_limit];
pts_L3 = np.array([pt5,pt6], np.int32)
pts_L3 = pts_L3.reshape((-1,1,2))
pt7 = [0, down_limit];
pt8 = [w, down_limit];
pts_L4 = np.array([pt7,pt8], np.int32)
pts_L4 = pts_L4.reshape((-1,1,2))

#Pengurang latar belakang
fgbg = cv2.createBackgroundSubtractorMOG2(detectShadows = True)

#Filter untuk morfologi
kernelOp = np.ones((3,3),np.uint8)
kernelOp2 = np.ones((5,5),np.uint8)
kernelCl = np.ones((11,11),np.uint8)

#Variables
font = cv2.FONT_HERSHEY_SIMPLEX
persons = []
max_page = 5
pid = 1

while cap.isOpened():
    ret, img = camera.read()
    ret1, frame = cap.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.1, 5)

    for (x, y, w, h) in faces:
        face_img = cv2.resize(img[y:y + h, x:x + w], (224, 224),
interpolation=cv2.INTER_AREA)
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

face_img = np.asarray(face_img, dtype=np.float32).reshape(1, 224,
224, 3)
face_img = (face_img / 127.5) - 1

result = model.predict(face_img)
idx = np.argmax(result)
print(idx)
class_name = labels[idx].strip()
confidence = result.max(axis=1) * 100

if confidence > 90:
    label_text = "%s (%.2f %%" % (class_name[2:], confidence)
    cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
    cv2.rectangle(img, (x, y - 40), (x + w, y), (0, 255, 0), -2)
    cv2.putText(img, label_text, (x, y - 10), font, 0.75, (255,
255, 255), 1, cv2.LINE_AA)

#         timestamp = time.strftime("%Y%m%d_%H.%M%p")
#         safe_label = label_text.replace(' ', '').replace(':',
#         '').replace('.', '').replace('-', '')
#         cv2.imwrite(os.path.join(output_folder,
f"{label_text}_{timestamp}.jpg"), img)
    else:
        label_text = "Unknow"
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
        cv2.rectangle(img, (x, y - 40), (x + w, y), (0, 255, 0), -2)
        cv2.putText(img, label_text, (x, y - 10), font, 0.75, (255,
255, 255), 1, cv2.LINE_AA)
    for i in persons:
        i.age_one()

#Terapkan pengurangan latar belakang
fgmask = fgbg.apply(frame)
fgmask2 = fgbg.apply(frame)

#Untuk menghilangkan bayangan(warna abu-abu)
try:
    ret,imBin= cv2.threshold(fgmask,130,255,cv2.THRESH_BINARY)
    ret,imBin2 = cv2.threshold(fgmask2,130,255,cv2.THRESH_BINARY)

    mask = cv2.morphologyEx(imBin, cv2.MORPH_OPEN, kernel10p)
    mask2 = cv2.morphologyEx(imBin2, cv2.MORPH_OPEN, kernel10p)

    mask = cv2.morphologyEx(mask , cv2.MORPH_CLOSE, kernel1C1)
    mask2 = cv2.morphologyEx(mask2, cv2.MORPH_CLOSE, kernel1C1)
except:
    print('EOF')
    print( 'Masuk:',cnt_up)
    print ( 'Keluar:',cnt_down)
    break
contours0, hierarchy =
cv2.findContours(mask2,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
for cnt in contours0:
    area = cv2.contourArea(cnt)

```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

if area > areaTH:

    M = cv2.moments(cnt)
    cx = int(M['m10']/M['m00'])
    cy = int(M['m01']/M['m00'])
    x,y,w,h = cv2.boundingRect(cnt)

    new = True
    if cy in range(up_limit,down_limit):
        for i in persons:
            if abs(x-i.getX()) <= w and abs(y-i.getY()) <= h:
                #objek tersebut dekat dengan objek yang telah
                terdeteksi sebelumnya
                new = False
                i.updateCoords(cx,cy) #memperbarui koordinat
                dalam objek dan mengatur ulang usia
                if i.going_UP(line_down,line_up) == True:
                    cnt_up += 1;
                    print( "ID:",i.getId(),'crossed going up
                    at',time.strftime("%c"))
                    log.write("ID: "+str(i.getId())+' crossed
                    going up at ' + time.strftime("%c") + '\n')
                elif i.going_DOWN(line_down,line_up) == True:
                    cnt_down += 1;
                    print( "ID:",i.getId(),'crossed going down
                    at',time.strftime("%c"))
                    log.write("ID: " + str(i.getId()) + ' crossed
                    going down at ' + time.strftime("%c") + '\n')
                    break
                if i.getState() == '1':
                    if i.getDir() == 'down' and i.getY() >
                    down_limit:
                        i.setDone()
                    elif i.getDir() == 'up' and i.getY() < up_limit:
                        i.setDone()
                    if i.timedOut():
                        #hapus i dari daftar orang
                        index = persons.index(i)
                        persons.pop(index)
                        del i #hapus i

                if new == True:
                    p = Person.MyPerson(pid,cx,cy, max_p_age)
                    persons.append(p)
                    pid += 1

            cv2.circle(frame,(cx,cy), 5, (0,0,255), -1)
            img = cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),2)
            #cv2.drawContours(frame, cnt, -1, (0,255,0), 3)

        for i in persons:
            cv2.putText(frame,
            str(i.getId()),(i.getX(),i.getY()),font,0.3,i.getRGB(),1,cv2.LINE_AA)

            str_up = 'Masuk: '+ str(cnt_up)
            str_down = 'Keluar: '+ str(cnt_down)

```



© Hak Cipta milik Politeknik Negeri Jakarta

```

frame =
cv2.polylines(frame,[pts_L1],False,line_down_color,thickness=2)
frame =
cv2.polylines(frame,[pts_L2],False,line_up_color,thickness=2)
frame =
cv2.polylines(frame,[pts_L3],False,(255,255,255),thickness=1)
frame =
cv2.polylines(frame,[pts_L4],False,(255,255,255),thickness=1)
cv2.putText(frame, str_up
,(10,40),font,0.5,(255,255,255),2,cv2.LINE_AA)
cv2.putText(frame, str_up ,(10,40),font,0.5,(0,0,255),1,cv2.LINE_AA)
cv2.putText(frame, str_down
,(10,90),font,0.5,(255,255,255),2,cv2.LINE_AA)
cv2.putText(frame, str_down
,(10,90),font,0.5,(255,0,0),1,cv2.LINE_AA)

cv2.imshow("Hasil", img)
cv2.imshow('Frame', frame)
k = cv2.waitKey(30) & 0xff
if k == 27:
    break

log.flush()
log.close()
camera.release()
cap.release()
cv2.destroyAllWindows()

```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

PROGRAM PERSON

```

from random import randint
#Modul random adalah modul bawaan Python yang menyediakan fungsi-fungsi
untuk menghasilkan angka-angka acak.
import time

class MyPerson:
    tracks = []
    def __init__(self, i, xi, yi, max_age):
        self.i = i #inisialisasi id
        self.x = xi #inisialisasi koordinat x
        self.y = yi #inisialisasi koordinat y
        self.tracks = []
        self.R = randint(0,255) #ambil angka acak 0 sampe 255
        self.G = randint(0,255)
        self.B = randint(0,255)
        self.done = False
        self.state = '0'
        self.age = 0
        self.max_age = max_age
        self.dir = None #inisialisasi arah
    def getRGB(self):
        return (self.R,self.G,self.B)
    def getTracks(self):
        return self.tracks
    def getId(self):
        return self.i
    def getState(self):
        return self.state
    def getDir(self): #mengembalikan arah pergerakan objek
        return self.dir
    def getX(self):
        return self.x
    def getY(self):
        return self.y
    def updateCoords(self, xn, yn): #update koordinat
        self.age = 0
        self.tracks.append([self.x,self.y]) #menambahkan koordinat
        sebelumnya ke dalam tracks
        self.x = xn #update koordinat baru
        self.y = yn
    def setDone(self):
        self.done = True #mengatur status objek menjadi selesai
    def timedOut(self):
        return self.done
    def going_UP(self,mid_start,mid_end): #periksa objek bergerak keatas
        if len(self.tracks) >= 2: #periksa jumlah objek
            if self.state == '0':
                if self.tracks[-1][1] < mid_end and self.tracks[-2][1] >=
mid_end: #periksa objek sedang melewati satu garis
                    state = '1' #update state objek
                    self.dir = 'up'

```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

        return True
    else:
        return False
    else:
        return False
def going_DOWN(self,mid_start,mid_end):
    if len(self.tracks) >= 2:
        if self.state == '0':
            if self.tracks[-1][1] > mid_start and self.tracks[-2][1]
<= mid_start:
                state = '1'
                self.dir = 'down'
                return True
            else:
                return False
        else:
            return False
def age_one(self):
    self.age += 1
    if self.age > self.max_age:
        self.done = True
    return True
class MultiPerson:
    def __init__(self, persons, xi, yi):
        self.persons = persons
        self.x = xi
        self.y = yi
        self.tracks = []
        self.R = randint(0,255)
        self.G = randint(0,255)
        self.B = randint(0,255)
        self.done = False

```