

Identifikasi Jumlah dan Tingkat Aktivitas Orang Berbasis Pengolahan Citra Menggunakan Raspberry Pi

Daniel Krisrenanto, Muhammad Rivai, dan Fajar Budiman

Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111

e-mail: muhammad_rivai@ee.its.ac.id, fajarbudiman@ee.its.ac.id

Abstrak— *Air conditioner* (AC) merupakan suatu perangkat yang paling banyak digunakan sebagai pendingin ruangan. Walaupun AC saat ini sudah menggunakan teknologi hemat energi, namun apabila tidak digunakan dengan baik maka penggunaan AC hanya menjadi suatu pemborosan listrik saja. Untuk mengurangi pemborosan listrik pada AC, maka pengaturan suhu harus disesuaikan dengan kondisi ruangan. Beberapa faktor yang mempengaruhi kondisi ruangan antara lain adalah jumlah orang serta tingkat aktivitas dalam ruangan. Sebagai upaya penghematan konsumsi listrik pada AC, maka dalam penelitian ini dilakukan rancang bangun suatu sistem identifikasi jumlah orang dan tingkat aktivitas. Pada sistem ini digunakan kamera USB sebagai perangkat pengambil citra. Pada pengolahan citra menggunakan metode *Histogram of Oriented Gradient* (HOG) sebagai penghitung jumlah orang, sedangkan metode *background subtraction* digunakan sebagai penghitung tingkat aktivitas. Sistem ini diimplementasikan ke dalam Raspberry Pi 3 sebagai mikrokontroler. Hasil pengujian menunjukkan bahwa sistem dapat mendeteksi orang dari jarak 3 m sampai 9 m dan membedakan 2 orang dengan jarak antara 30 cm sampai 150 cm. Sistem juga dapat membedakan tingkat aktivitas sedikit, sedang dan tinggi.

Kata Kunci— *Background Subtraction; Histogram of Oriented Gradient; Jumlah orang; Tingkat aktivitas.*

I. PENDAHULUAN

PERMASALAHAN seperti perubahan iklim, pencemaran air, udara, dan tanah, krisis air, energi, dan sumber daya alam, serta berkurangnya lahan hijau saat ini menjadi isu lingkungan global yang merupakan masalah nyata. Institut Teknologi Sepuluh Nopember (ITS) berkomitmen untuk berperan aktif dalam pengembangan ilmu dan teknologi serta penerapan gaya hidup yang berwawasan lingkungan dan berbasis eco-campus [1].

Penggunaan listrik yang tidak efektif ini dikarenakan pemakaian perangkat elektronik secara tidak wajar. Salah satu perangkat elektronik yang sering digunakan secara tidak wajar adalah *air conditioner* (AC). Penggunaan AC yang tidak wajar terjadi antara lain, ketika AC yang dinyalakan walaupun tidak ada kegiatan di dalam ruangan, pengaturan suhu yang kurang tepat, dan AC dipasang pada ruangan yang tidak cocok dengan desain AC. AC akan bekerja terus bekerja semaksimal mungkin, walaupun efek pendinginan tidak dapat dirasakan. Hal ini hanya menjadi suatu tindakan pemborosan listrik saja.

Melihat kondisi seperti ini, dibutuhkan suatu sistem otomatis untuk mengatur AC. Sistem otomatis yang dirancang dituntut untuk fleksibel dan mementingkan kenyamanan pengguna

ruangan serta mampu meningkatkan penghematan listrik. Pada penelitian ini telah dirancang sebuah sistem yang dapat mengidentifikasi jumlah orang dan tingkat aktivitas. Sistem ini diharapkan dapat menghitung jumlah orang dan tingkat aktivitas, sehingga dapat digunakan sebagai acuan remote control AC dalam menentukan suhu ruangan. Sistem identifikasi ini dilengkapi dengan kamera sebagai perangkat pengambilan citra yang kemudian diproses untuk mendapatkan data jumlah orang dan tingkat aktivitas. Raspberry Pi 3 digunakan sebagai perangkat pemroses citra.

II. TEORI PENUNJANG

A. *Histogram of Oriented Gradient*

Pada desain penelitian ini digunakan metode *Histogram of Oriented Gradient* (HOG) sebagai metode *object detection*. Metode HOG merupakan metode *feature based* yang melihat tampilan lokal dan bentuk objek pada citra berupa intensitas distribusi gradien atau arah kontur [2]. Metode ini menggunakan *database* berupa histogram yang berisi kumpulan nilai gradien pada sebaran orientasi tertentu. Citra yang diperoleh kamera diubah menjadi *grayscale* kemudian dinormalisasi kecerahannya serta melalui proses koreksi gamma. Algoritma HOG dapat disederhanakan menjadi 3 tahapan. Tahapan tersebut yaitu: komputasi gradien, pengelompokan orientasi, dan pembuatan blok deskriptor [3].

1) *Komputasi Gradien*

Citra yang telah diubah menjadi *grayscale* diturunkan dengan mengkonvolusikan citra dengan matriks:

$$D_x = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \text{ dan } D_y = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}^T \quad (1)$$

Setelah mendapatkan turunan x dan y (I_x dan I_y), kemudian nilai besaran dan orientasi vektor tersebut dihitung:

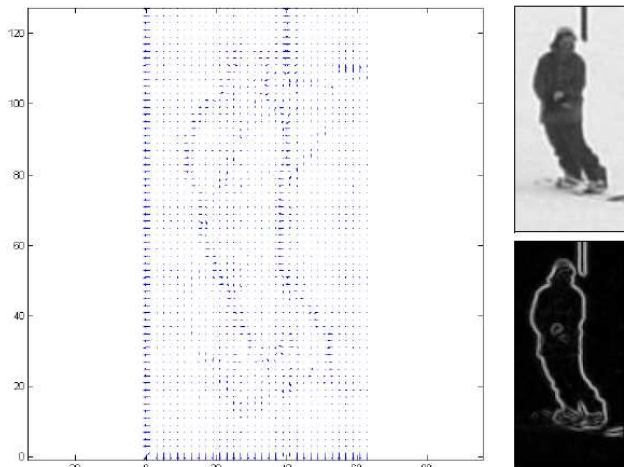
$$|G| = \sqrt{I_x^2 + I_y^2} \text{ dan } \theta = \tan^{-1} \frac{I_y}{I_x} \quad (2)$$

Perlu diperhatikan bahwa orientasi θ harus dalam rentang $[0^\circ, 360^\circ]$, karena dalam implementasi ini nilai orientasi harus bernilai positif. Hasil komputasi gradien pada citra dapat dilihat pada Gambar 1.

2) *Pengelompokan Orientasi*

Tahap berikutnya adalah pembuatan histogram pada tiap sel yang akan digunakan untuk membuat blok deskriptor. Citra dibagi menjadi sel-sel berukuran 8×8 piksel. Gradien piksel tiap sel dikelompokkan berdasarkan orientasi. Gradien piksel

tiap sel dibagi menjadi 9 kelompok dalam rentang 0° sampai 180° [2], [3]. Kemudian besaran gradien piksel akan dijumlahkan pada setiap kelompok.



Gambar 1. Visualisasi hasil komputasi besaran dan orientasi gradien [3]

3) Blok Deskriptor

Proses normalisasi untuk membuat blok deskriptor diawali dengan pengelompokan sel-sel menjadi blok. Setiap blok berukuran 2×2 sel dan setiap sel berukuran 8×8 piksel. Blok yang berdekatan menindih sebagian blok lain yang berdekatan [2].

Pada citra lokal terdapat kontras yang kompleks antara *foreground*, *background* dan pencahayaan, sehingga rentang nilai gradien sangat besar dan gradien lokal memerlukan standarisasi yang efektif. Terdapat berbagai macam metode standarisasi dimana prinsip dasarnya adalah dengan menggabungkan sel-sel menjadi ruangan yang lebih besar dan menstandarisasi setiap blok. Karena setiap blok merupakan hasil sampling data yang padat, maka akan terjadi banyak *hallmark* yang dihitung berulang kali di blok yang berbeda. Walaupun terdapat data yang sangat banyak, metode yang menggunakan blok yang saling menindih dan penghitungan fitur yang terstandarisasi dapat meningkatkan hasil klasifikasi [4].

Asumsikan bahwa $\mathbf{v}$ adalah vektor tanpa normalisasi, $\|\mathbf{v}\|_k$ adalah vektor hasil normalisasi dengan metode k -norm dimana $K = 1, 2$. ϵ adalah konstanta normalisasi yang kecil untuk menghindari pembagi nol. Terdapat 4 macam metode normalisasi:

Metode L2-Norm:

$$\mathbf{v} \rightarrow \frac{\mathbf{v}}{\sqrt{\|\mathbf{v}\|_2^2 + \epsilon^2}} \quad (3)$$

Metode L1-Norm:

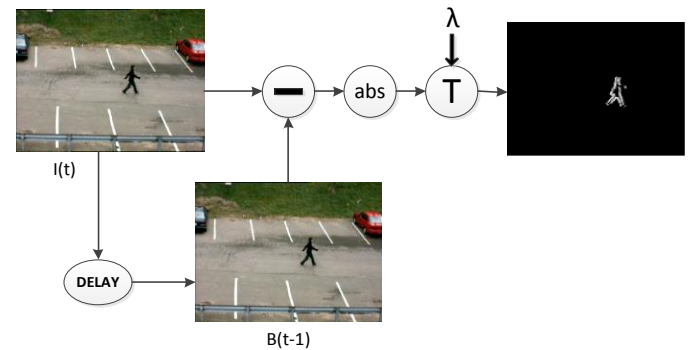
$$\mathbf{v} \rightarrow \frac{\mathbf{v}}{\sqrt{\|\mathbf{v}\|_1 + \epsilon}} \quad (4)$$

Metode L1-Sqrt:

$$\mathbf{v} \rightarrow \sqrt{\frac{\mathbf{v}}{\|\mathbf{v}\|_1 + \epsilon}} \quad (5)$$

Metode L2-Hys adalah metode L2-Norm yang dinormalisasi kembali ketika $\mathbf{v}$ bernilai maksimum 0.2 [4].

Deskriptor hasil normalisasi tersebut disebut dengan deskriptor HOG yang berupa vektor 1-D dengan ukuran sebesar jumlah blok pada citra $\times$ jumlah sel pada citra $\times 9$ [2].



Gambar 2. Proses background subtraction dengan 2 frame

Proses deskriptor HOG pada setiap blok yang ada pada citra kemudian digunakan sebagai acuan proses *object detection*.

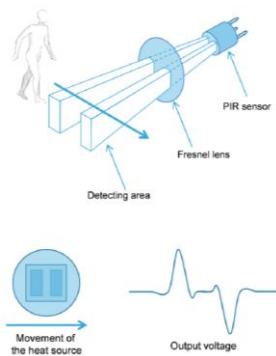
B. Background Subtraction

Tujuan utama dari *background subtraction* adalah untuk mendeteksi semua objek *foreground* dengan mengambil *frame* secara berurutan dari sebuah kamera tetap. Pendekatan paling sederhana adalah dengan mengurangi citra *frame* dengan citra *background* yang statis. Pendekatan ini memerlukan sebuah citra *background* yang statis, dimana citra tersebut dapat merepresentasikan latar belakang dari objek *foreground* suatu citra *frame*. Citra *background* tidak selalu tetap, tetapi mampu beradaptasi dengan berbagai perubahan, seperti kecerahan, getaran pada kamera, dan perubahan geometri *background*.

Selain menggunakan citra *background* statis, proses *background subtraction* dapat dilakukan dengan mengurangi *frame* terbaru dengan *frame* sebelumnya. Kemudian hasil pengurangan tersebut di *threshold* untuk mengidentifikasi objek *foreground*. Metode ini lebih mudah digunakan dan lebih adaptif. Metode ini memerlukan dua *frame* yang diambil secara berurutan dengan waktu *delay* tertentu. Objek yang diam akan dianggap sebagai *background*, sedangkan objek yang bergerak akan terdeteksi. Gambaran proses *background subtraction* dengan 2 *frame* dapat dilihat di Gambar 2.

C. Sensor Passive Infrared

Sensor *Passive Infrared* (PIR) adalah sensor elektronik yang mengukur cahaya inframerah yang diradiasikan oleh suatu objek yang melewati daerah deteksi sensor, ditunjukkan pada Gambar 3 Tujuan dari sensor PIR adalah mendeteksi gerakan dalam suatu daerah tertentu. Sensor ini memiliki 2 bagian yang sensitif terhadap inframerah. Karena pergerakan yang dideteksi, maka sinyal yang dihasilkan oleh sensor mencerminkan perbedaan nilai inframerah antara kedua bagian. Jika kedua bagian menangkap jumlah inframerah yang sama, maka sensor tidak mendeteksi apapun. Tetapi jika satu bagian membaca jumlah inframerah yang lebih atau kurang dari yang lain, maka sensor mendeteksi suatu gerakan dan tegangan output berubah.



Gambar 3. Prinsip kerja sensor PIR [5]

D. Open Source Computer Vision

Open Source Computer Vision (OpenCV) adalah sebuah API (*Application Programming Interface*) library yang sudah sangat familiar pada bidang pengolahan citra berbasis komputer. *Computer vision* sendiri adalah salah satu cabang dari bidang ilmu pengolahan citra (*Image Processing*) yang memungkinkan komputer dapat melihat seperti manusia. Dengan *computer vision* tersebut komputer dapat mengambil keputusan, mengambil tindakan, dan mengenali suatu objek tertentu. Beberapa implementasi dari *computer vision* adalah *face recognition*, *face/object detection*, *face/object tracking*, *road tracking*, *motion detection*, dan lain sebagainya.

OpenCV adalah *library open source* untuk *computer vision* yang dapat diimplementasikan menggunakan bahasa pemrograman C/C++, Python, Java dan Matlab/Octave. OpenCV didesain untuk aplikasi *real-time*, memiliki fungsi-fungsi akuisisi yang baik untuk citra dan video. OpenCV juga menyediakan *interface* ke *Integrated Performance Primitives* (IPP) Intel sehingga aplikasi *computer vision* dapat dioptimasi jika menggunakan prosesor Intel [6].

E. Thresholding

Proses segmentasi citra merupakan proses pemisahan citra menjadi beberapa daerah. Salah satu parameter yang digunakan sebagai dasar untuk proses segmentasi adalah intensitas. Setiap piksel pada tiap daerah memiliki intensitas yang berbeda-beda. Segmentasi yang menggunakan parameter intensitas tiap daerah disebut juga *thresholding*, yaitu pemisahan daerah intensitas tinggi (terang) dengan daerah intensitas rendah (gelap). Proses *thresholding* menciptakan citra biner dari citra *grayscale*.

F. Contour Finding

Kontur adalah sebuah kumpulan titik-titik yang merepresentasikan sebuah kurva dalam suatu citra. Titik-titik ini membentuk sebuah kurva yang membatasi daerah yang memiliki perbedaan intensitas yang signifikan. Beberapa model dalam pencarian kontur yang sering dipakai antara lain, pendekatan struktural dengan perluasan daerah, pendekatan stokastik, dan pendekatan variasional, yang berdasarkan berbagai macam strategi seperti *level set formulations*, meminimalisasi total variasi dari kuantitas tertentu atau fungsi Mumford-Shah, metode kontur aktif dan metode kontur aktif geodesik, snakes, transformasi wavelet, atau gradien bentuk. Pendekatan lain adalah berdasarkan analisis topologi

asimptotik yang terdiri dari pendeteksian tepi sebagai retakan [7].

G. Konvolusi Citra

Konvolusi mengoperasikan dua buah sinyal (dalam 1D) atau dua buah citra (dalam 2D). Satu sinyal (atau gambar) merupakan input dan yang lain disebut sebagai kernel yang berfungsi untuk memfilter input, dan menghasilkan citra output. Sebuah kernel adalah sebuah matriks kecil yang digunakan pada konvolusi. Kernel dengan ukuran dan nilai yang berbeda akan menghasilkan citra hasil konvolusi yang berbeda juga [8].

Konvolusi citra merupakan konvolusi dikrit pada dua dimensi. Definisi secara matematis ditunjukkan pada persamaan berikut:

$$I'(u,v) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(u+i, v+j) \cdot H(i,j) \quad (6)$$

Dimana:

$I'(u,v)$: output piksel di koordinat (u,v)

$I(u,v)$: input piksel di koordinat (u,v)

$H(i,j)$: filter kernel berukuran $i \times j$

Citra input dan output terdiri dari $\text{width} \times \text{height}$ piksel. Konvolusi citra menggunakan filter kernel dilakukan dengan mengalikan citra input dengan filter kernel secara element-wise. Piksel yang sedang dihitung, $I'(u,v)$, adalah jumlah semua piksel $I(u,v)$ pada citra input yang telah dikalikan dengan koefisien pada filter kernel H [9]. Gambar 4 menunjukkan proses konvolusi citra.

III. PERANCANGAN SISTEM

A. Perancangan Perangkat Lunak

Pada tahap perancangan *software* Raspberry Pi diprogram untuk mampu mengidentifikasi jumlah orang dan aktivitas. Selain itu, Raspberry Pi juga diprogram untuk dapat melakukan *halt* secara otomatis. Secara garis besar, perancangan *software* dapat dilihat pada Gambar 5. *Software* dirancang menggunakan bahasa pemrograman python.

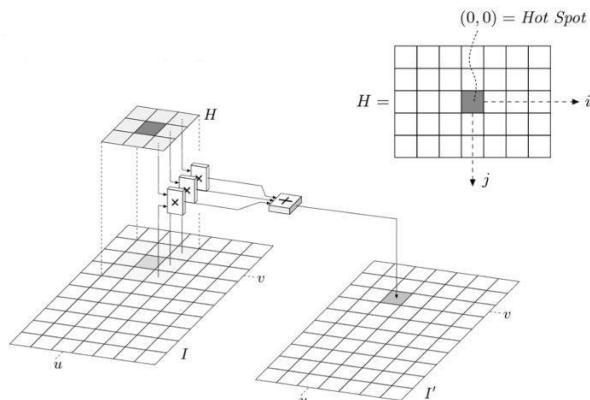
1) Identifikasi Jumlah Orang

Proses identifikasi jumlah orang sesuai dengan diawali dengan pengambilan citra dari kamera. Kemudian ukuran citra diperkecil untuk mengurangi waktu deteksi. Selanjutnya citra dideteksi menggunakan sintaks *detectMultiScale* dengan HOG deskriptor untuk menghitung jumlah orang pada citra yang diambil.

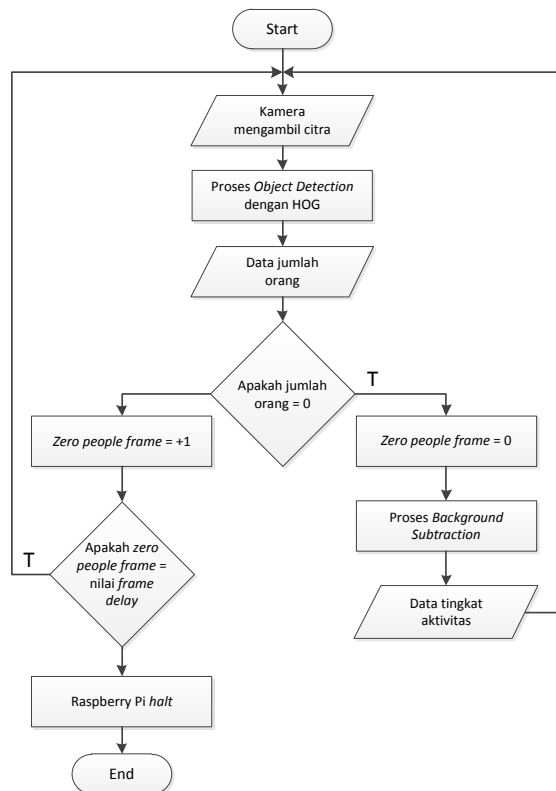
2) Identifikasi Tingkat Aktivitas

Proses identifikasi tingkat aktivitas menggunakan metode *background subtraction*. Proses diawali dengan kamera mengambil citra pertama. Setelah selang waktu tertentu kamera mengambil citra berikutnya. Kemudian ukuran citra diperkecil dan dilanjutkan dengan mengubah citra menjadi *grayscale*. Pencarian citra selisih menggunakan sintaks *cv2.absdiff*. Sintak *cv2.absdiff* melakukan diferensiasi absolut antara dua matriks citra untuk menghasilkan citra selisih. Citra selisih antara citra pertama dan citra berikutnya kemudian di *threshold* untuk mendapatkan citra biner.

Proses penghitungan aktivitas diawali dengan mencari kontur pada citra selisih yang telah di *threshold*. Setelah daerah-daerah kontur ditemukan, maka luasan daerah-daerah tersebut dihitung. Apabila luasan daerah tersebut di bawah nilai tertentu, maka pada daerah tersebut dinyatakan tidak ada gerakan atau aktivitas. Apabila luasan daerah tersebut di atas nilai tertentu, maka pada daerah tersebut dinyatakan terdapat gerakan atau aktivitas. Kemudian daerah-daerah yang terdapat aktivitas dihitung sebagai output identifikasi tingkat aktivitas.



Gambar 4. Visualisasi konvolusi citra

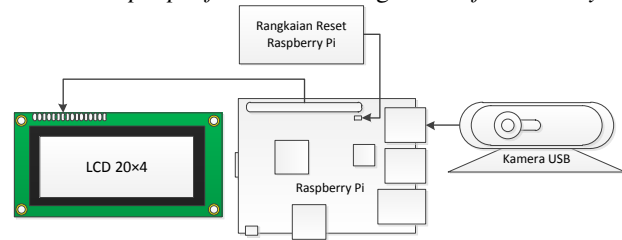


Gambar 5. Diagram alir software

3) Halt otomatis

Sistem dirancang untuk dapat melakukan *halt* otomatis. *Halt* otomatis digunakan untuk mematikan Raspberry Pi dan AC. Raspberry Pi dan AC dimatikan apabila kondisi ruangan tidak memerlukan pendinginan AC yaitu ketika tidak ada orang di dalam ruangan. Proses *halt* otomatis diawali dengan penghitungan *frame* citra dengan data jumlah orang sama

dengan nol. Nilai penghitungan ini disimpan pada variabel *zero people frame*. Penghitungan dilakukan terus sampai variabel *zero people frame* sama dengan nilai *frame delay*.



Gambar 6. Diagram hardware sistem

Frame delay adalah sebuah nilai yang harus dicapai sebelum Raspberry Pi dan AC di *halt*. Variabel ini diperlukan agar Raspberry Pi dan AC tidak langsung di *halt* apabila satu *frame* memiliki data jumlah orang sama dengan nol. Apabila variabel *zero people frame* sama dengan nilai *frame delay* maka Raspberry Pi dan AC di *halt*. Sekuen *halt* otomatis dapat dilihat pada Gambar 5.

B. Perancangan Perangkat Keras

1) Rangkaian Reset

Gambar 6 menunjukkan perancangan perangkat keras. Sistem dirancang untuk dapat menyala secara otomatis, karena sistem diharapkan mampu berjalan tanpa bantuan atau campur tangan orang. Sistem didesain untuk menyala kembali ketika kondisi ruangan memerlukan pendinginan AC yaitu ketika terdapat aktivitas orang di dalam ruangan. Aktivitas atau gerakan orang dapat dideteksi dengan sensor PIR.

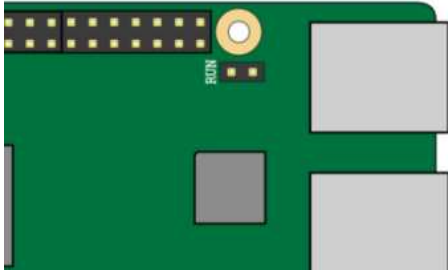
Penyalan sistem secara otomatis dilakukan dengan menyalakan kembali (*reboot*) Raspberry Pi yang dalam kondisi *halt*. Raspberry Pi dinyatakan dalam kondisi *halt* yaitu ketika Raspberry Pi tidak melakukan proses apapun walaupun masih menerima suplai daya. Raspberry Pi dapat di *reboot* dari kondisi *halt* dengan menghubungkannya pin RUN pada Raspberry Pi. Lokasi pin RUN Raspberry Pi 3 berdekatan dengan pin 39 GPIO dan dapat dilihat pada Gambar 7.

Raspberry Pi dirangkai dengan sebuah rangkaian reset. Rangkaian reset akan mendeteksi gerakan orang dengan sensor PIR, kemudian sensor PIR mengirim sinyal ke sebuah *relay* untuk menghubungkannya pin RUN Raspberry Pi. Skematik rangkaian reset dapat dilihat di Gambar 8.

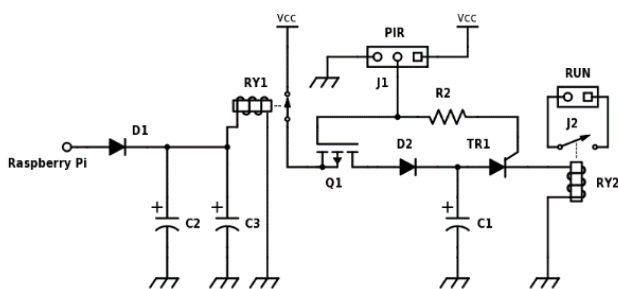
Relay RY1 digunakan sebagai pengaman Raspberry Pi. *Relay* RY1 yang Normally Closed akan menjadi *open* ketika mendapat sinyal HIGH dari Raspberry Pi. Sinyal HIGH ini menandakan bahwa Raspberry Pi tidak dalam kondisi *halt*. Pemasangan RY1 ini bertujuan untuk mencegah terjadinya hubung singkat pada pin RUN Raspberry Pi yang sedang berjalan.

Tipe MOSFET Q1 adalah P-Channel dengan $V_{GS(th)}$ bernilai negatif. Ketika output sensor PIR dalam keadaan LOW, maka V_{GS} MOSFET Q1 bernilai negatif, sehingga R_{DS} On dan arus mengalir mengisi kapasitor C1. Ketika sensor PIR mendeteksi gerakan orang, maka output sensor menjadi HIGH. Hal ini mengakibatkan V_{GS} MOSFET Q1 mendekati nol sehingga R_{DS} Off dan menghentikan proses pengisian kapasitor C1. Dalam waktu yang sama output PIR memberikan sinyal HIGH pada

SCR TR1 sehingga kapasitor C1 mengeluarkan muatannya ke *relay* RY2. Hal ini mengakibatkan perubahan pada *relay* RY2 yang semula Normally Opened menjadi *closed* dan menghubungsingkatkan pin RUN Raspberry Pi. Ketika seluruh muatan kapasitor C1 dikeluarkan ke *relay* RY2, SCR TR1 akan berhenti konduksi dan menghalangi arus menuju *relay* RY2.



Gambar 7. Pin RUN Raspberry Pi



Gambar 8. Skematik rangkaian reset

Tabel 1.
Pemasangan port LCD dan GPIO Raspberry Pi

Port LCD	GPIO Raspberry Pi
RS	GPIO 10
RW	GPIO 9
EN	GPIO 11
D4	GPIO 6
D5	GPIO 13
D6	GPIO 19
D7	GPIO 26



Gambar 9. Hasil pengujian penghitung orang

2) LCD Display

Sistem yang dirancang dilengkapi dengan LCD *display*. Tujuan dari pemasangan LCD display ini adalah sebagai media yang menampilkan informasi tentang keadaan sistem. Informasi yang ditampilkan pada LCD adalah jumlah orang, jumlah aktivitas, suhu AC sekarang, dan FPS. Pin-pin LCD dihubungkan ke port GPIO Raspberry Pi. Pin LCD dan GPIO Raspberry Pi dirangkai sesuai Tabel 1. Pin VSS dihubungkan dengan port GROUND, pin VDD dihubungkan dengan port

5V POWER, dan V0 dihubungkan ke pin 2 potensiometer. Potensiometer bernilai 5KΩ.

Tabel 2.

Hasil pengujian sesor PIR tanpa halangan	
Jarak dari sensor (m)	Keterangan
1	Terdeteksi
2	Terdeteksi
3	Terdeteksi
4	Terdeteksi
5	Terdeteksi
6	Terdeteksi
7	Terdeteksi
8	Terdeteksi

IV. HASIL PENGUJIAN DAN ANALISIS

A. Pengujian Sensor PIR

Pengujian ini dilakukan untuk mengetahui seberapa jauh jangkauan PIR dalam mendeteksi gerakan manusia. Pengujian dilakukan dengan tanpa memberi halangan pada sensor.

Dari Tabel 2 dapat disimpulkan bahwa sensor PIR mampu mendeteksi objek sampai 8 m.

B. Pengujian Detektor Orang

1) Pengujian Penghitung Orang

Pengujian dilakukan untuk mengetahui kemampuan detektor dalam mengidentifikasi orang. Pengujian dilakukan dengan menempatkan objek berupa orang pada jarak tertentu, ditunjukkan pada Gambar 9. Kamera diletakkan 122 cm dari tanah. Jumlah sampel yang diambil pada pengujian ini sebanyak 22 sampel. Jumlah sampel yang diambil terdapat total 66 orang dan terdeteksi sebanyak 41 orang.

2) Pengujian Jarak Deteksi Antara Kamera Dengan Orang

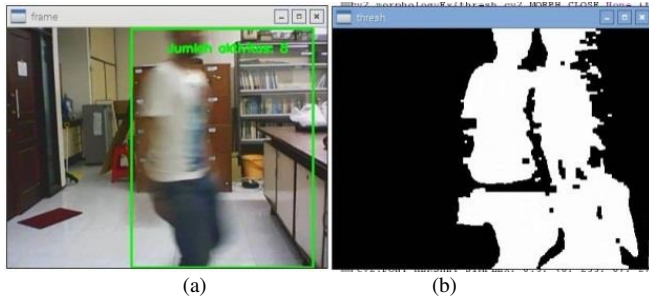
Pengujian dilakukan untuk mengetahui jarak minimum dan maksimum detektor dalam mengidentifikasi orang. Pengujian dilakukan dengan menempatkan satu objek orang pada jarak tertentu. Dari data Tabel 3 menunjukkan bahwa detektor mampu mendeteksi pada jarak antara 3 m sampai 9 m. Hal ini terjadi karena ukuran piksel objek terlalu besar apabila berjarak kurang dari 3 m. Dan ukuran piksel terlalu kecil apabila berjarak lebih dari 9 m.

Tabel 3.

Hasil pengujian jarak deteksi orang	
Jarak dari kamera (m)	Keterangan
1	Tidak Terdeteksi
2	Tidak Terdeteksi
3	Terdeteksi
4	Terdeteksi
5	Terdeteksi
6	Terdeteksi
7	Terdeteksi
8	Terdeteksi
9	Terdeteksi
10	Tidak Terdeteksi
11	Tidak Terdeteksi
12	Tidak Terdeteksi
13	Tidak Terdeteksi
14	Tidak Terdeteksi
15	Tidak Terdeteksi

Tabel 4.
Hasil pengujian dengan orang berdekatan

Jarak (cm)	Orang 1	Orang 2
150	√	√
120	√	√
90	√	√
60	√	√
30	√	√
0	√	x



Gambar 10. (a) Hasil pengujian detektor dan (b) citra biner

Tabel 5.
Rata-rata luasan hasil deteksi

Kategori Objek	Sedikit	Sedang	Tinggi
Pertama	4671,9	12752,9	19625,1
Kedua	5105,8	12950,9	18358
Ketiga	5575,2	12828,3	21563,3

3) Pengujian Dengan Orang yang Berdekatan

Pengujian dilakukan untuk mengetahui kemampuan detektor dalam mendeteksi objek yang berdekatan. Pengujian dilakukan dengan menggunakan 2 orang. Jarak antara objek dengan kamera ditetapkan pada satu nilai tertentu. Dari data Tabel 4 dapat dilihat bahwa detektor mampu membedakan 2 orang dari jarak 150 cm sampai 30 cm. Di bawah 30 cm detektor tidak dapat membedakan 2 orang yang saling berdekatan. Hal ini terjadi karena postur 2 orang tersebut berhimpitan, sehingga detektor hanya dapat mendeteksi 1 orang saja.

C. Pengujian Detektor Gerakan Orang

Pengujian dilakukan untuk mengetahui kemampuan detektor untuk mendeteksi gerakan orang. Pengujian dilakukan dengan mendeteksi beberapa gerakan, ditunjukkan pada Gambar 10. Pada kategori tingkat aktivitas sedikit, objek melakukan gerakan mengangkat tangan. Pada kategori sedang, objek berjalan bolak-balik di hadapan kamera. Pada kategori gerakan tinggi objek berlari bolak-balik di hadapan kamera. Pada setiap kategori diambil sebanyak 10 sampel. Objek yang bergerak ditangkap oleh kamera yang diletakkan setinggi 122 cm dari tanah. Pengujian dilakukan dengan 3 objek. Objek pertama memiliki tinggi badan 155 cm dan berat badan 57 kg. Objek kedua mempunyai tinggi badan 165 cm dan berat badan 60 kg. Objek ketiga memiliki tinggi badan 163 cm dan berat badan 66 kg. Data yang diambil merupakan luasan daerah putih pada citra biner yang dihasilkan oleh metode *background subtraction*.

Dari data Tabel 5 dapat dilihat bahwa semakin tinggi tingkat aktivitas, maka semakin besar pula rata-rata daerah luasan yang dihasilkan. Selain itu, terdapat perbedaan yang signifikan

pada rata-rata daerah luasan antarkategori. Sehingga dari hasil tersebut dapat disimpulkan bahwa detektor gerakan dapat membedakan tingkat aktivitas orang.

V. KESIMPULAN

Telah dirancang sebuah sistem identifikasi jumlah orang dan tingkat aktivitas yang berbasis pengolahan citra berbasis Raspberry Pi board. Sistem ini dirancang mampu menyala dan mati sesuai kondisi ruangan. Berdasarkan hasil pengujian dan analisis dapat disimpulkan bahwa sensor PIR dapat digunakan sebagai detektor gerakan orang sampai jarak 8 m. Metode *Histogram of Oriented Gradient* yang digunakan dalam sistem dapat diimplementasikan sebagai metode *object detection* untuk mendeteksi jumlah orang pada jarak 3 m sampai 9 m. Metode *background subtraction* dapat digunakan sebagai metode untuk membedakan tingkat aktivitas orang dengan melihat daerah luasan yang dihasilkan. Sebagai pengembangan disarankan untuk menggunakan *multithreading* yaitu proses parallel pada komputasi untuk mempercepat proses identifikasi. Selain itu penambahan kamera stereo untuk meningkatkan kemampuan sistem dalam mendeteksi jumlah orang..

DAFTAR PUSTAKA

- [1] Fadilla, Nurin. "Building Automation System Berbasis Mikrokontroler untuk Monitoring dan Kontrol Energi", Institut Teknologi Sepuluh Nopember, 2015.
- [2] Kachouane, M. "HOG Based Fast Human Detection", Université Saad Dahlab de Blida, 2012.
- [3] Yıldız, Cansin. "An Implementation on Histogram of Oriented Gradients for Human Detection", Bilkent University, 2012.
- [4] LiFei. "Pedestrian Detection Based On Histogram of Oriented Gradient in Intelligent Transportations System", Shenyang University of Technology, 2013.
- [5] Colliard-Piraud, Sylvain. "Signal conditioning for pyroelectric passive infrared (PIR) sensors", AN4368, STMicroelectronics, 2013.
- [6] Syafii, S. I. "Open Computer Vision (OpenCV)", 2012
- [7] Auroux, Didier. "Contour Detection and Completion for Inpainting and Segmentation Based on Topological Gradient and Fast Marching Algorithms", Université de Nice Sophia Antipolis, 2012.
- [8] Ludwig, Jamie. "Image Convolution", Portland State University, 2013.
- [9] Ström, Henrik. "A Parallel FPGA Implementation of Image Convolution", Department of Electrical Engineering, Linköping University, 2016.
- [10] Said, Yahia. et.al. "Human Detection Based On Integral Histograms of Oriented Gradients and SVM", Faculty of Sciences Monastir, 2012.