

# ***Pattern Recognition* untuk Deteksi Posisi pada AGV Berbasis Raspberry Pi**

## ***(Pattern Recognition for AGV's Position Detection Based on Raspberry Pi)***

Florentinus Budi Setiawan<sup>1</sup>, Franciska Amalia Kurnianingsih<sup>2</sup>, Slamet Riyadi<sup>3</sup>, Leonardus Heru Pratomo<sup>4</sup>

**Abstract**—The development of technology in automation and robotics is overgrowing because of its high-efficiency level in terms of labor and time. In the warehousing system, one of the robots used is Automated Guided Vehicle (AGV). AGV is a transportation device in the form of a robot that can be controlled automatically, which functions as a carrier of goods using a navigation system to move in a predetermined direction. One of the existing AGV navigation systems is by following a line pattern on the floor. The system is inefficient because, gradually, the line pattern will wear out and can not be detected again due to the friction force of the AGV wheels itself. Therefore, it is necessary to develop an AGV navigation system to minimize these obstacles. This pattern recognition system uses a pattern placed on the building ceiling and camera as a sensor facing upwards, so that AGV can freely detect patterns. Then, the detected pattern was processed through a programmed Raspberry Pi 4 Model B. The test results show that this system can detect the position and successfully displays the coordinate point  $(x, y)$  of the AGV and will continue to run at any time until the program is changed as ordered.

**Intisari**—Perkembangan teknologi di bidang otomatisasi dan robotika berkembang sangat pesat, karena memiliki tingkat efisiensi tinggi dari segi tenaga dan waktu. Pada sistem pergudangan, salah satu robot yang digunakan adalah *Automated Guided Vehicle* (AGV). AGV adalah alat transportasi berupa robot yang dikendalikan secara otomatis, yang berfungsi sebagai pengangkut barang, dengan menggunakan sistem navigasi agar bergerak ke arah yang telah ditentukan. Salah satu sistem navigasi AGV yang telah ada ialah dengan mengikuti pola garis pada lantai. Sistem tersebut kurang efisien karena lambat laun pola garis tersebut akan pudar dan tidak dapat terdeteksi kembali akibat gaya gesek dari roda AGV itu sendiri. Oleh karena itu, sangat diperlukan pengembangan sistem navigasi AGV untuk meminimalkan hambatan tersebut. Sistem *pattern recognition* ini menggunakan pola yang diletakkan pada langit-langit bangunan dan kamera sebagai sensor yang menghadap ke atas sehingga AGV mampu dengan leluasa mendeteksi pola. Kemudian, pola yang sudah terdeteksi diolah melalui perangkat komputer berupa Raspberry Pi 4 Model B yang telah diprogram. Hasil pengujian menunjukkan bahwa sistem ini mampu mendeteksi posisi dan berhasil menampilkan titik koordinat  $(x,y)$  dari AGV serta akan tetap berjalan sampai kapan pun hingga program diubah sesuai yang diperintahkan.

**Kata Kunci**—Otomatisasi, Robotika, AGV, *Pattern Recognition*, Kamera, Titik Koordinat.

### I. PENDAHULUAN

Pada awalnya kamera hanya berfungsi dan digunakan untuk menangkap gambar atau video saja. Namun, seiring dengan berkembangnya teknologi, kamera dapat dialihfungsikan sebagai sensor yang mampu mendeteksi objek benda mati atau benda hidup yang diinginkan. Nantinya gambar yang sudah dideteksi akan diolah melalui perangkat komputer yang telah diprogram untuk tujuan tertentu. Teknologi tersebut dinamakan *computer vision*. Dengan mengunggulkan kamera yang terkomputasi, teknologi ini dapat menggantikan fungsi banyak sensor yang digunakan di kehidupan sehari-hari, seperti sensor warna, sensor jarak, sensor gerak, dan sebagainya [1].

*Computer vision* pertama kali diciptakan pada awal 1970 untuk persepsi visual yang meniru kecerdasan manusia dan diaplikasikan pada robot, sehingga robot memiliki kemampuan melihat secara visual seperti manusia dengan tujuan penalaran dan perencanaan tingkat tinggi [1]. Salah satu contoh aplikasi dari *computer vision* adalah *pattern recognition* (pengenalan pola).

*Pattern recognition* merupakan pengelompokan simbolik secara otomatis oleh komputer yang bertujuan mengenali suatu objek atau pola, seperti manusia yang mampu mengenali objek karena otak manusia dapat mengenali dan membedakan suatu objek dengan objek lainnya. Kemampuan sistem visual manusia inilah yang dicoba ditiru oleh komputer [2]. Komputer menerima masukan berupa foto objek yang akan dikenali, lalu memprosesnya dan memberikan keluaran berupa keterangan nama objek tersebut [3]. Kecerdasan tersebut yang digunakan untuk mengembangkan sistem navigasi pada AGV *line-follower* yang masih memiliki hambatan, yaitu hilangnya jalur navigasi akibat gaya gesek dari roda robot, sehingga dinilai kurang efisien. Untuk menangkis hambatan tersebut, digunakanlah kamera sebagai sensor yang akan mendeteksi pola dan objek sebagai jalur navigasi.

Pada makalah ini, *pattern recognition* pada AGV menggunakan beberapa komponen atau *hardware* yang sudah beredar luas di pasaran dunia, yang mampu mendukung pemrograman dari *pattern recognition* itu sendiri. Komponen utama yang digunakan yaitu Raspberry Pi 4 Model B, yang berfungsi sebagai komputer pengolah data, yang memiliki dimensi seukuran dengan kartu kredit, tetapi memiliki kinerja mesin yang mumpuni untuk pemrograman yang berhubungan dengan visual atau gambar. Oleh karena itu, dapat dikatakan bahwa Raspberry Pi memiliki kinerja terbaik dalam *embedded system*. Selain memiliki *port General Purpose Input/Output* (GPIO), Raspberry Pi juga memiliki *micro SD card slot* yang digunakan sebagai tempat penyimpanan sistem operasi,

<sup>1,2,3,4</sup> Program Studi Teknik Elektro, Fakultas Teknik Universitas Katolik Soegijapranata, Jl. Pawiyatan Luhur Sel.IV No.1 Semarang 50234 INDONESIA (telp: 024-8441555; <sup>1</sup>f.budi.s@unika.ac.id, <sup>2</sup>franciskaamalia@gmail.com, <sup>3,4</sup>tu.elektro@unika.ac.id)

Bluetooth wireless 5.0, display port, dua buah port USB 2.0, dua buah port USB 3.0, ethernet port yang digunakan untuk menghubungkan Raspberry Pi dengan jaringan, serta port kamera. Kemudian, kamera yang digunakan adalah Kamera Raspberry Pi V2 yang memang diciptakan khusus untuk Raspberry Pi, sehingga lebih mudah dalam pengaplikasiannya [4]. Kamera ini memiliki resolusi 8 MP dan memiliki ukuran yang kecil, sekitar  $2 \times 2$  cm, sehingga mudah terpasang di mana saja [5].

## II. METODOLOGI

### A. Automated Guided Vehicle (AGV)

AGV adalah alat transportasi berupa robot yang dikendalikan secara otomatis. AGV berfungsi sebagai robot pengangkut barang yang menggunakan sistem navigasi dengan cara mengikuti sebuah petunjuk atau garis pada lantai atau menggunakan pantulan laser agar bergerak ke arah yang telah ditentukan. AGV memiliki beberapa keunggulan, di antaranya adalah sebagai berikut [6].

- Memiliki bobot robot yang ringan.
- Mampu mengangkut barang dengan kapasitas besar.
- Mekanisme yang sederhana.
- Merupakan robot transportasi yang aman dan fleksibel.
- Memiliki efisiensi dan otomatisasi yang tinggi.

Sistem AGV praktis umumnya bergantung pada arsitektur kontrol terpusat, yaitu satu unit pusat melakukan berbagai tugas kompleks, seperti penjadwalan misi, perencanaan jalur, dan koordinasi gerak. Unit pusat nantinya berkomunikasi dengan setiap sistem dalam AGV, mengamati posisi AGV, menghitung rencana pergerakan, dan mentransmisikan kontrol tindakan. AGV memiliki beberapa sistem utama, yaitu sistem kontrol, sistem *mobile*, sistem navigasi, sistem komunikasi, dan sistem daya. Makalah ini fokus pada sistem komunikasi.

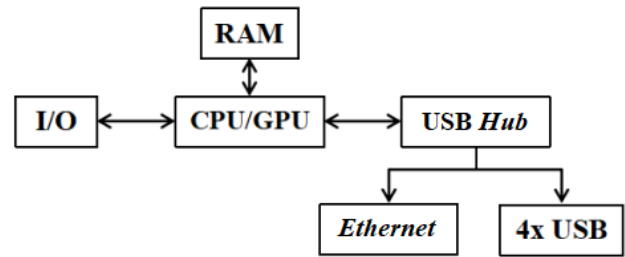
### B. Raspberry Pi 4 Model B

Raspberry Pi merupakan modul komputer berukuran kecil yang memiliki masukan dan keluaran seperti pada board mikrokontroler lainnya. Raspberry Pi hanya sebesar kartu kredit, tetapi dapat berfungsi untuk menjalankan program-program berat yang digunakan pada perkantoran, permainan komputer, dan juga dapat digunakan sebagai pemutar media dengan resolusi tinggi. Pada makalah ini digunakan Raspberry Pi 4 Model B. Gbr. 1 merupakan diagram blok Raspberry Pi.

Raspberry Pi 4 Model B merupakan generasi pertama dari generasi terbaru komputer Raspberry Pi yang mendukung lebih banyak RAM dan dengan peningkatan kinerja CPU, GPU, dan I/O yang signifikan dalam bentuk yang faktor daya dan biayanya seperti Raspberry Pi 3B + generasi sebelumnya. Raspberry Pi 4B tersedia dengan 1, 2, dan 4 GB SDRAM LPDDR4 [7].

### C. Modul Kamera Raspberry Pi

Modul kamera Raspberry Pi adalah sebuah kamera pintar yang merupakan produk kamera resmi yang diciptakan agar kompatibel (cocok) dengan modul Raspberry Pi 4 Model B. Modul kamera yang kompatibel langsung dengan perangkat



Gbr. 1 Diagram blok Raspberry Pi.

TABEL I  
PERBANDINGAN SPESIFIKASI DUA VERSI MODUL KAMERA

	Modul Kamera Versi 1	Modul Kamera Versi 2
Ukuran	25 × 24 × 9 mm	25 × 24 × 9 mm
Berat	3 g	3 g
Resolusi	5 MP	8 MP
Mode video	1.080p30, 720p60 dan 640 × 480p60/90	1.080p30, 720p60 dan 640 × 480p60/90
Sensor	Omnivision OV5647	Sony IMX219
Area sensor gambar	3,76 × 2,74 mm	3,68 × 2,76 mm
Ukuran piksel	1,4 μm × 1,4 μm	1,12 μm × 1,12 μm

komputer akan lebih mudah diakses dan diprogram karena telah sesuai dari segi spesifikasi dan fungsinya. Modul kamera ini memiliki fungsi untuk menangkap gambar statis yang akan digunakan sebagai *input* data untuk algoritme pemrosesan gambar [8]. Kamera tersebut memiliki dua versi, seperti disajikan pada Tabel I.

Di samping dua versi tersebut, Raspberry Pi 4 Model B juga memiliki dua tipe kamera berdasarkan gambar yang dihasilkan, yaitu tipe *Infra Red* (IR) dan *NO Infra Red* (NOIR). Tipe IR memiliki hasil gambar yang lebih baik dan cerah ketika siang hari atau keadaan terang, sedangkan tipe NOIR memiliki hasil yang lebih cerah dan terang ketika malam hari atau keadaan gelap.

### D. Computer Vision

*Computer vision* dapat diartikan sebagai teknologi yang mempelajari cara komputer melihat dan mengenali objek yang sedang diamati. Sebagai teknologi baru, *computer vision* menerapkan teori dan model untuk membangun sistem *computer vision* itu sendiri. Data pada gambar dapat memproses berbagai bentuk, seperti video, urutan gambar, ataupun dari data multidimensi dalam *scanner* medis.

Setelah kamera menangkap gambar sebagai *input* data, selanjutnya data digunakan untuk algoritme pemrosesan gambar. Algoritme analisis gambar tersebut akan menghasilkan informasi berupa pola dan menyimpan urutan pola. Untuk mengotomatisasi tugas-tugas ini, dasar pendekatan *computer vision* dimodifikasi dan diterapkan ke umpan kamera *real-time*. Beberapa fungsi yang disediakan oleh *Open Source Computer Vision* (OpenCV) berisi lebih dari 500 algoritme yang dioptimalkan untuk analisis dan manipulasi gambar atau video. OpenCV mendukung beberapa bahasa, yaitu C, C++, Python,

Java, dan MATLAB. OpenCV juga dapat dijalankan pada sistem operasi Windows, Linux, Android, dan macOS, sehingga dapat dirancang dengan bantuan pustaka OpenCV dan dioptimalkan untuk beroperasi secara efektif pada *platform* Raspberry Pi [9]. *Computer vision* dapat dikatakan sebagai kombinasi antara *image processing* (pengolahan citra) dan *pattern recognition*.

*Image processing* berhubungan dengan proses transformasi citra (*image*). Proses ini bertujuan untuk mengekstrak atau mengubah informasi ataupun pesan yang disampaikan oleh gambar yang dimaksud.

*Computer vision* diterapkan pada beberapa hal, di antaranya pada bidang medis (penggunaan alat-alat pendeteksi tumor/*x-ray*), pada bidang keamanan (penggunaan sistem *face recognition* pada saat melakukan absensi di perusahaan), pada bidang industri (penggunaan deteksi pada produk cacat), dan pada bidang otomotif (penggunaan sistem parkir pada kendaraan tanpa sopir).

**E. Pattern Recognition**

*Pattern recognition* merupakan suatu tindakan pengambilan data mentah berdasarkan klasifikasi data, yang akan membandingkan dengan satu ciri atau lainnya yang dimiliki oleh suatu pola. Keluaran dari proses ini berupa informasi atau suatu deskripsi pada citra. Proses pada bidang *pattern recognition* berhubungan dengan proses indentifikasi objek pada citra atau interpretasi citra. Klasifikasi ditujukan pada pengakuan pola yang digunakan untuk membedakan antara pola satu dengan lainnya, yang akan diwakilkan oleh pola huruf [10]. Contoh pola dan cirinya ditunjukkan pada Tabel II.

Dengan menggunakan ciri pola yang khas, akan tercipta daya pembeda yang tinggi, sehingga pada setiap polanya akan diperoleh keakuratan yang tinggi pula. *Pattern recognition* ini digunakan untuk mengetahui posisi suatu benda yang akan dituju oleh AGV. AGV nantinya akan secara mandiri merencanakan jalur dengan tujuan lokasi yang diberikan dan mengoordinasi gerakannya berdasarkan komunikasi dengan perbedaan pola. Pada setiap eksekusi, posisi AGV akan selalu diprogram ulang untuk mencapai target posisi selanjutnya.

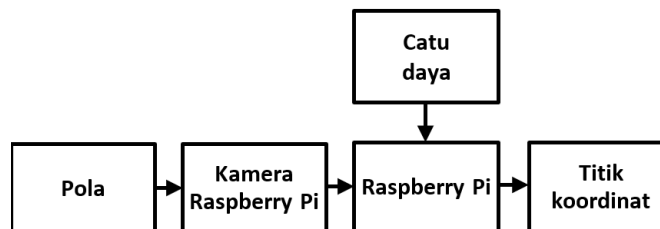
**F. OpenCV**

OpenCV, yang merupakan kepanjangan dari *Open Source Computer Vision Library*, berisikan pustaka fungsi yang telah ditentukan sebelumnya untuk membantu dalam pemrosesan gambar. OpenCV ini bersifat gratis. OpenCV memiliki beberapa bahasa pemrograman, antara lain C++, Python, dan Java. OpenCV juga mendukung sistem operasi Windows, Linux, macOS, iOS, dan Android. OpenCV dirancang untuk efisiensi komputasi dan dengan fokus yang kuat pada aplikasi *real-time*. Ditulis dalam bahasa C/C++ yang telah dioptimalkan, *library* akan dapat mengambil keuntungan pengolahan *multi-core* [11].

Karena bersifat *open source*, OpenCV dipilih sebagai *platform* untuk menguji proyek dengan lebih mudah. Dengan menggunakan OpenCV *library*, makalah ini telah mengimplementasikan mekanisme gambar pemrosesan seperti konversi karakter atau pola, erosi, dan pelebaran.

TABEL II  
CONTOH POLA DAN CIRINYA

Pola	Ciri - ciri
Suara	Frekuensi, nada, intonasi, amplitudo, dll.
Sidik Jari	Jumlah garis, lengkungan, dll.
Tulisan Tangan	Kerumitan, panjang, tekanan, dll.
Huruf	Tebal, titik, tinggi, lengkungan garis, sudut, dll.



Gbr. 2 Diagram blok sistem *pattern recognition*.

**G. Optical Character Recognition (OCR)**

*Optical Character Recognition* (OCR) merupakan *software* yang memiliki fungsi untuk mengubah gambar cetak berpola menjadi bentuk teks digital yang dapat dibaca dan diubah lebih lanjut. Terdapat dua jenis OCR, yaitu OCR *offline* dan OCR *online*. OCR *offline* adalah pengenalan karakter setelah pencetakan dokumen selesai, sedangkan OCR *online* adalah pengenalan karakter pada saat digambar atau dicetak.

Tesseract adalah *software* OCR gratis, yang awalnya dikembangkan oleh Hewlett dan Packard (HP), lalu dirilis di bawah lisensi Apache Versi 2.0 dan pengembangannya telah disponsori oleh Google sejak 2006 [12]. Tesseract juga merupakan mesin OCR mentah, tanpa tata letak analisis dokumen dan dapat mengonversi gambar atau video atau karakter teks secara *real-time*. Tesseract dapat dilatih untuk bekerja pada bahasa lain. Tesseract cocok digunakan sebagai *back end* dan mampu melaksanakan tugas OCR yang rumit, termasuk analisis tata letak yang menggunakan *front end* seperti OCR. Tidak ada format Tesseract yang tidak disertakan GUI dan sebaliknya dijalankan dari antarmuka baris perintah.

Pada makalah ini, digunakan sistem OCR Tesseract. Sistem OCR ini gratis dan mudah untuk diintegrasikan menggunakan bahasa Python. Selain itu, sistem ini memiliki tingkat akurasi yang baik. Dengan demikian, sistem dapat mencakup semua tindakan yang dilakukan.

III. METODOLOGI

Metode yang dilakukan pada makalah ini adalah sebagai berikut.

**A. Diagram Blok dari Pattern Recognition**

Gbr. 2 memperlihatkan diagram blok dari sistem *pattern recognition*. Diagram blok tersebut merupakan cara kerja *pattern recognition* untuk deteksi posisi pada AGV dengan keluaran berupa data digital. Alat ini dilengkapi dengan masukan sensor berupa perangkat kamera Raspberry Pi yang akan membaca beberapa pola (huruf abjad) yang telah

ditentukan. Bagian kendali menggunakan Raspberry Pi 4 Model B dengan RAM 4 GB dan *SD card* 64 GB. Raspberry Pi mendapatkan sumber daya dari satu berupa *accu* yang telah dikonversi menggunakan rangkaian *step-down chopper*.

Pola (huruf abjad) yang ditangkap oleh kamera diolah oleh sistem Raspberry Pi untuk menunjukkan atau mendeteksi posisi terkini AGV, sehingga Raspberry Pi akan memberi keluaran berupa titik koordinat  $(x,y)$  yang selanjutnya akan dikirimkan untuk menggerakkan motor sesuai perintah yang telah ditentukan.

Kemudian kamera akan mendeteksi objek (patung) yang digunakan sebagai tolok ukur dalam perpindahan dari AGV. Dengan adanya beberapa patung yang digunakan, dapat dihasilkan perpindahan yang lebih presisi menuju titik koordinat  $(x',y')$  selanjutnya.

### B. Momen Invariant dan Momen Zernike

Dalam *pattern recognition* atau objek, momen *invariant* dapat digunakan sebagai metode yang mengekstraksi sebuah pola atau objek. Metode ini menggunakan fungsi dua dimensi diskret, yaitu  $f(x,y)$  dalam bentuk digital dari momen biasa, yaitu  $(p + q)$ . Makalah ini mengadopsi momen *invariant* seperti pada (1).

$$m_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x,y). \quad (1)$$

Dari (1),  $(\bar{x} = \sum_x x, \bar{y} = \sum_y y)$  melambangkan pusat massa dari objek. Asal gambar digital  $f(x,y)$  pertama kali diasumsikan dalam menerjemahkan ke pusat massa dari suatu objek untuk mencapai terjemahan *invariant*.

Serupa dengan momen biasa, momen Zernike  $(m,n)$  digunakan untuk gambar digital yang ditentukan seperti pada (2) berikut.

$$A_{nm} = \frac{n+1}{\pi} \sum_x \sum_y f(x,y) V_{nm}(x,y), x^2 + y^2 \leq 1. \quad (2)$$

Dari (2),  $V_{nm}(x,y) = V_{nm}(\rho, \theta) = R_{nm}(\rho) e^{jm}$ ,  $|m| \leq n$ , dan  $n - |m|$  merupakan bilangan genap. Lalu  $R_{nm}(\rho)$  dapat didefinisikan seperti pada (3).

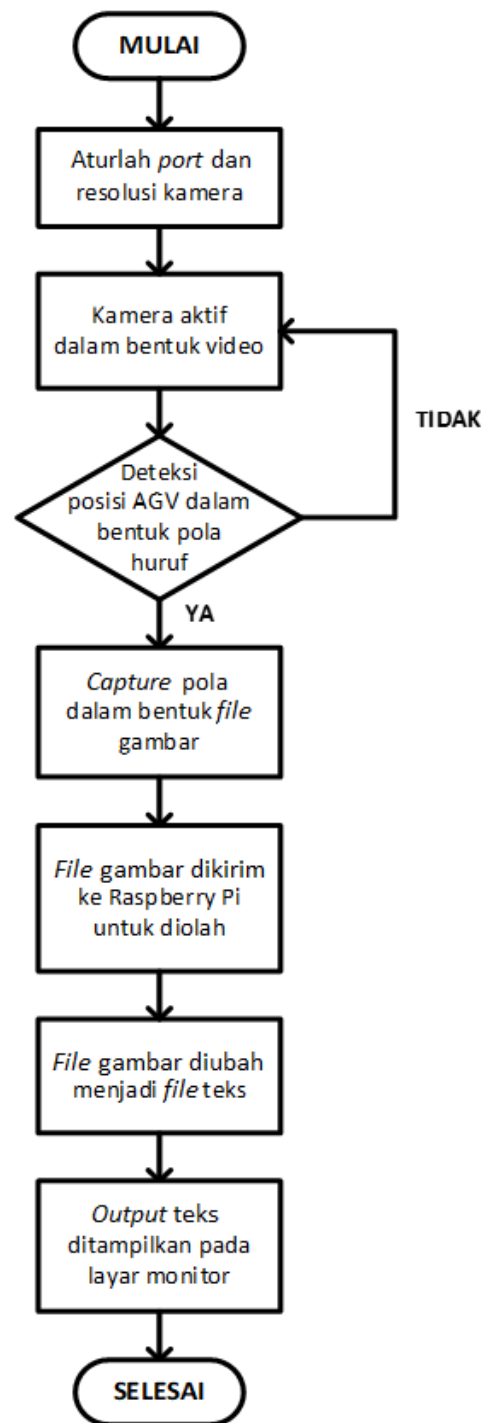
$$R_{nm}(\rho) = \sum_{s=0}^{\frac{n-|m|}{2}} (-1)^s \frac{(n-s)!}{s! \left(\frac{n+m-s}{2}\right)! \left(\frac{n-m-s}{2}\right)!} \rho^{n-2s}. \quad (3)$$

Untuk fungsi bayangan nyata  $f(x,y)$ , hanya suku  $V_{nm}(\rho, \theta)$  yang merupakan bilangan kompleks seperti pada (4).

$$A_{nm} = \frac{n+1}{\pi} \sum_{x_i} \sum_{y_j} f(x_i, y_j) [VR_{nm}(x,y) + jVI_{nm}(x,y)] \quad (4)$$

$$x_i^2 + y_j^2 \leq 1$$

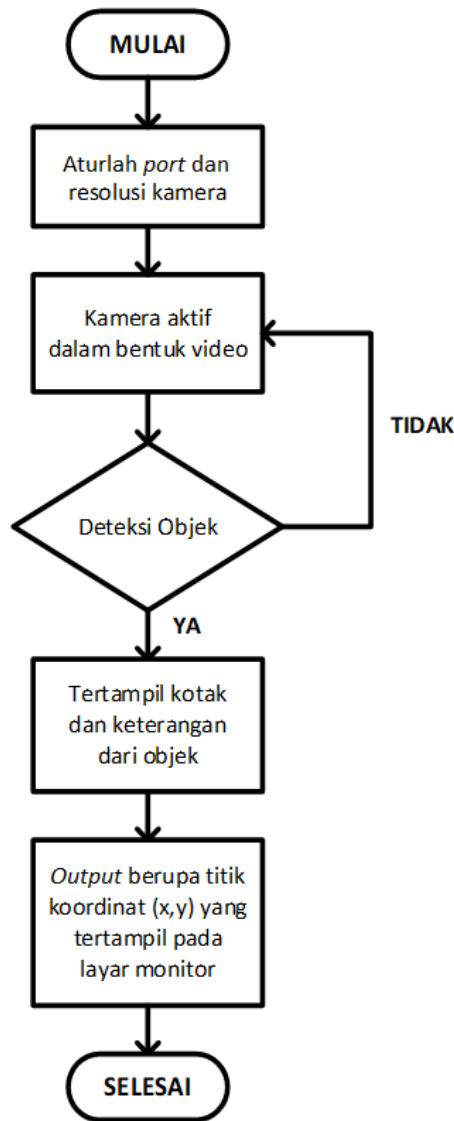
dengan  $VR$  dan  $VI$  merupakan fungsi nyata yang mempresentasikan komponen real dan imajiner dari fungsi dasar  $V_{nm}(\rho, \theta)$ , yang diubah menjadi bidang  $(x,y)$ .



Gbr. 3 Diagram alir proses sistem *pattern recognition*.

### C. Proses Pattern Recognition

Gbr. 3 memperlihatkan diagram alir yang menggambarkan rancangan proses sistem *pattern recognition*. Mula-mula kamera aktif dalam bentuk video, yang kemudian mulai mendeteksi pola (huruf abjad) untuk mengetahui posisi AGV terkini. Jika sudah terdeteksi, kamera akan menangkap (*capture*) pola yang berupa *file* gambar. Selanjutnya, *file* gambar tersebut dikirim ke Raspberry Pi untuk diubah menjadi *file* teks. Lalu



Gbr. 4 Diagram alir proses sistem *object detection*.

didapatkan keluaran berupa *file* teks yang ditampilkan pada layar monitor. Begitu selanjutnya hingga program dihentikan.

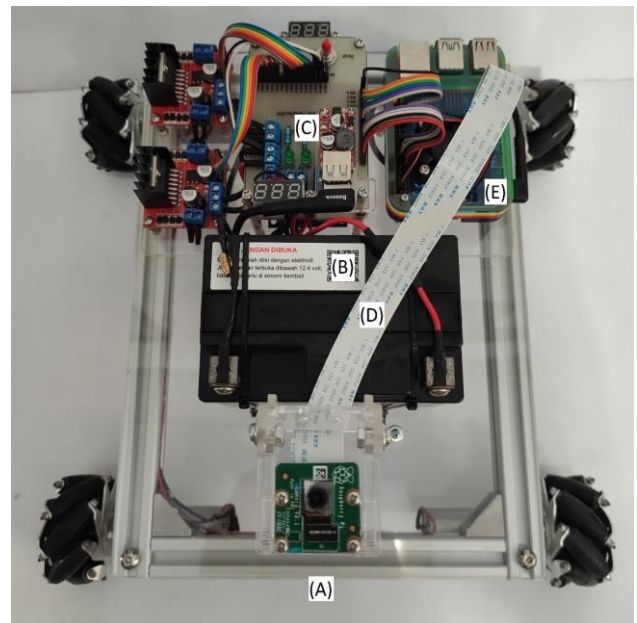
**D. Proses Object Detection**

Gbr. 4 menunjukkan diagram alir rancangan proses sistem *object detection*. Sama halnya dengan sistem *pattern recognition*, setelah kamera aktif dalam bentuk video, kamera mulai mendeteksi pola objek. Objek ini digunakan sebagai tolok ukur dalam perpindahan dari AGV. Jika objek sudah terdeteksi, akan tertampil kotak beserta keterangan yang disusun pada bagian kanan atas dari kotak, sesuai dengan nama objek tersebut. Kemudian didapatkan keluaran berupa titik koordinat (x,y) yang ditampilkan pada layar monitor.

**IV. HASIL DAN PEMBAHASAN**

**A. Hasil Alat**

Alat diperlihatkan pada Gbr. 5. Bagian (A) merupakan modul kamera Raspberry Pi, bagian (B) merupakan *accu*,



Gbr. 5 Hasil alat.



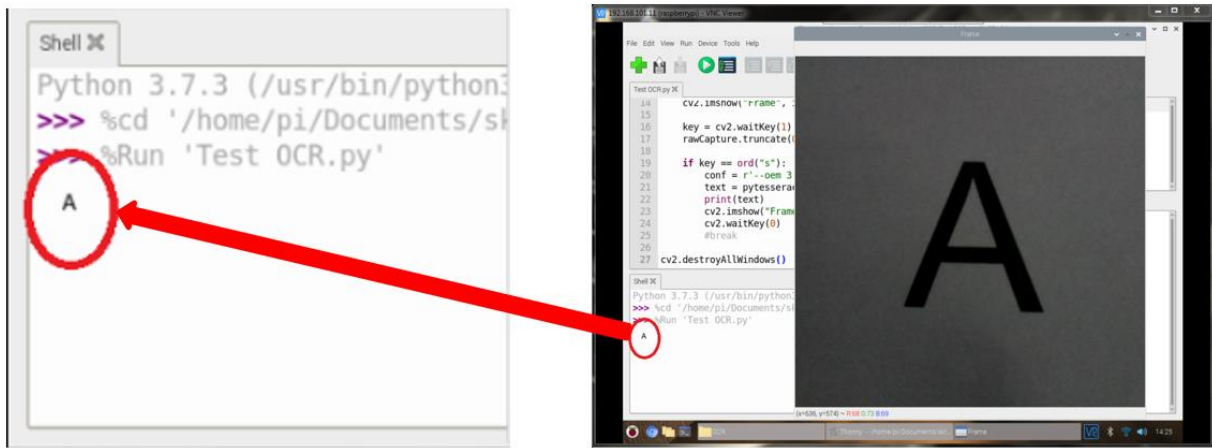
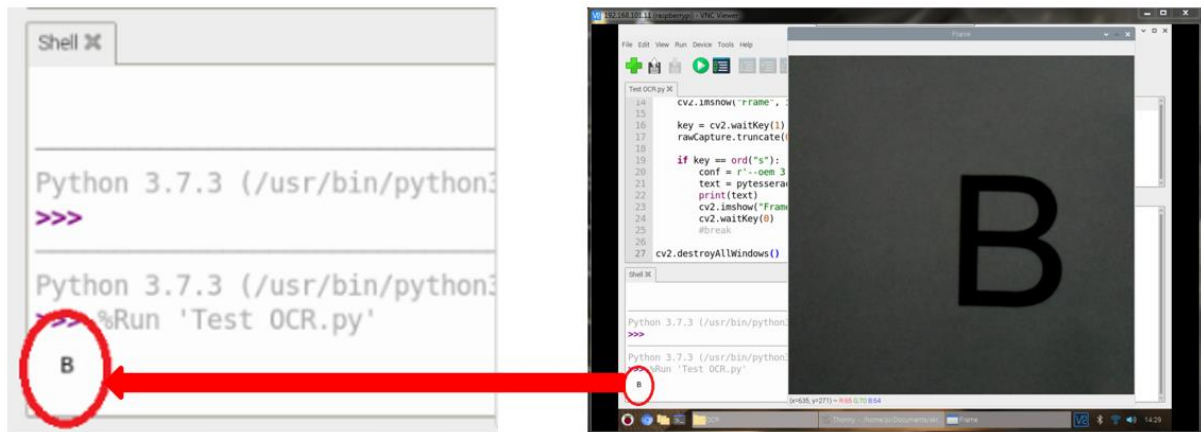
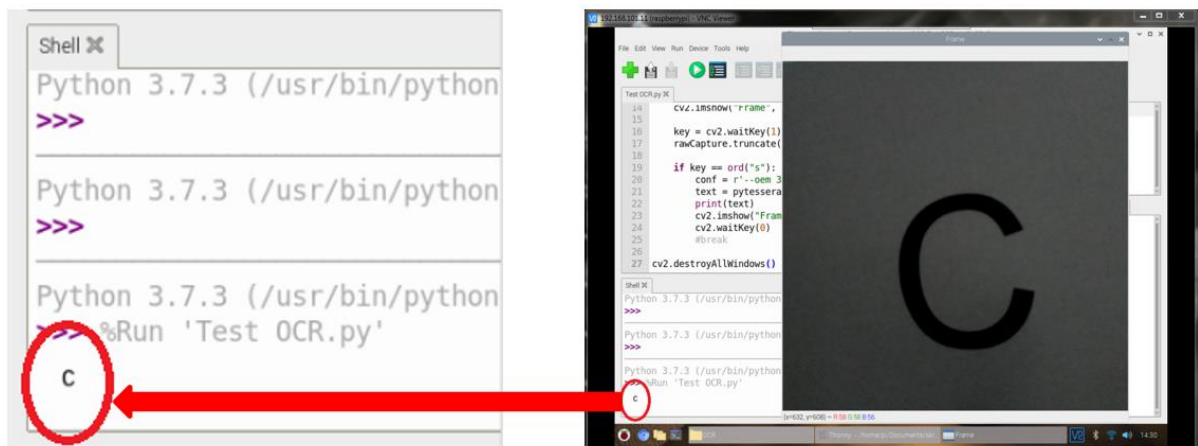
Gbr. 6 Objek deteksi, (a) pola huruf, (b) patung.

bagian (C) merupakan rangkaian pembagi tegangan, bagian (D) merupakan kabel konektor 15 pin I/O, dan bagian (E) merupakan modul Raspberry Pi.

Pengujian ini menggunakan pola huruf yang tercetak dengan warna hitam pada kertas HVS untuk *pattern recognition*, seperti ditunjukkan pada Gbr. 6(a). Selain itu, digunakan juga objek patung, seperti pada Gbr. 6(b).

**B. Hasil Pengujian Sistem OCR Capture**

Pada program OCR Capture dilakukan pengujian saat menangkap gambar pola, yaitu menggunakan huruf “A”, “B”, “C”, “D”, dan “P”. Pola “A”, “B”, “C”, dan “D” digunakan sebagai titik-titik tujuan dalam penyimpanan barang oleh AGV, sedangkan pola “P” digunakan sebagai titik posisi parkir pada robot AGV jika tidak digunakan atau saat pertama kali akan dijalankan.

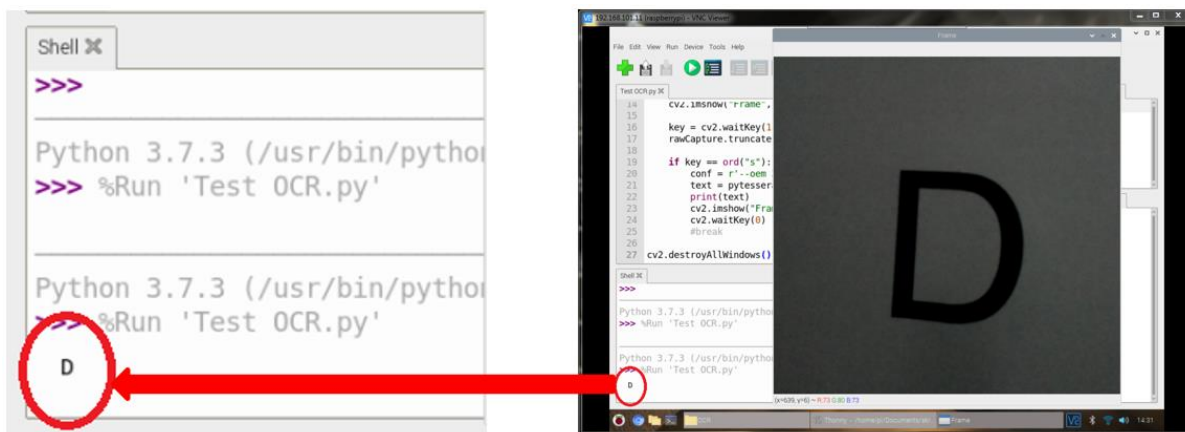
Gbr. 7 Hasil uji olah gambar menjadi teks pada *capture* pola “A”.Gbr. 8 Hasil uji olah gambar menjadi teks pada *capture* pola “B”.Gbr. 9 Hasil uji olah gambar menjadi teks pada *capture* pola “C”.

Pengujian program OCR Capture ini dapat dilakukan dengan cara menangkap gambar pola dari kamera. Kemudian, *file* gambar tersebut diolah dan ditampilkan menjadi *file* teks pada layar monitor. Keluaran berupa *file* teks inilah yang digunakan sebagai titik koordinat ( $x,y$ ) dari AGV.

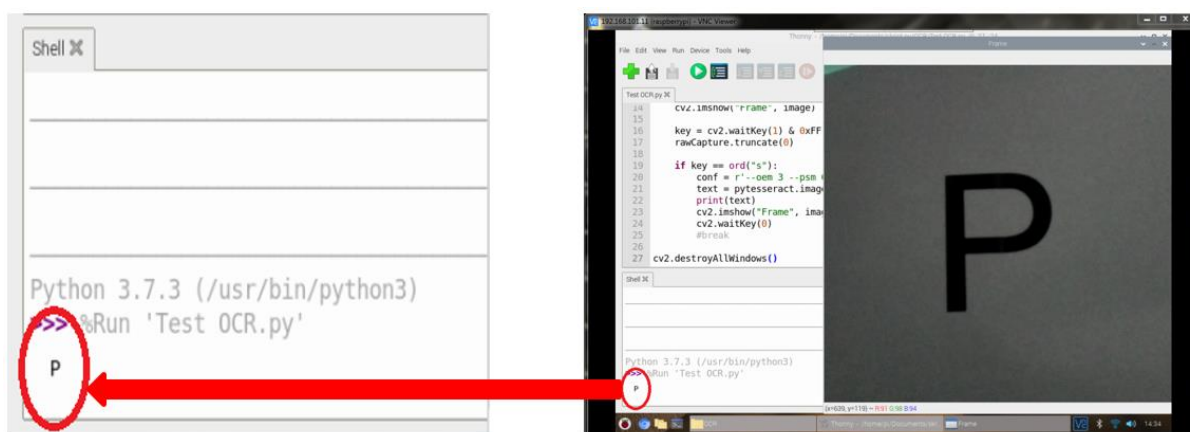
Hasil uji pemrosesan dalam mengubah *file* gambar menjadi *file* teks yang ditampilkan pada layar monitor ditunjukkan pada Gbr. 7 sampai Gbr. 11.

### C. Hasil Pengujian Sistem Object Detection

Pada program *object detection* dilakukan pengujian ketepatan program dalam mendeteksi objek. Objek yang digunakan adalah patung. Sebelumnya, data masukan berupa beberapa *file* gambar patung tersebut disimpan dalam satu *folder* dan diberikan nama atau keterangan “Patung”.



Gbr. 10 Hasil uji olah gambar menjadi teks pada *capture* pola “D”.



Gbr. 11 Hasil uji olah gambar menjadi teks pada *capture* pola “P”.



Gbr. 12 Hasil pengujian *object detection* pada objek patung terdeteksi.

Hasil uji deteksi objek adalah menampilkan kotak dengan keterangan di atasnya berupa tulisan “Patung” dari nama objek. Jika objek terdeteksi, tampilan seperti pada Gbr. 12 akan muncul di layar monitor. Jika objek tidak terdeteksi, tampilan seperti seperti Gbr. 13 muncul di layar monitor.

Sistem *object detection* ini juga mampu menampilkan hasil titik koordinat (173,64) dari objek “Patung” tersebut, yang terlihat pada bingkai “Shell”. Titik koordinat objek tersebut ditunjukkan pada titik *x* dan *y* paling bawah atau yang muncul paling akhir, seperti yang terlingkar pada Gbr. 14.

### V. KESIMPULAN

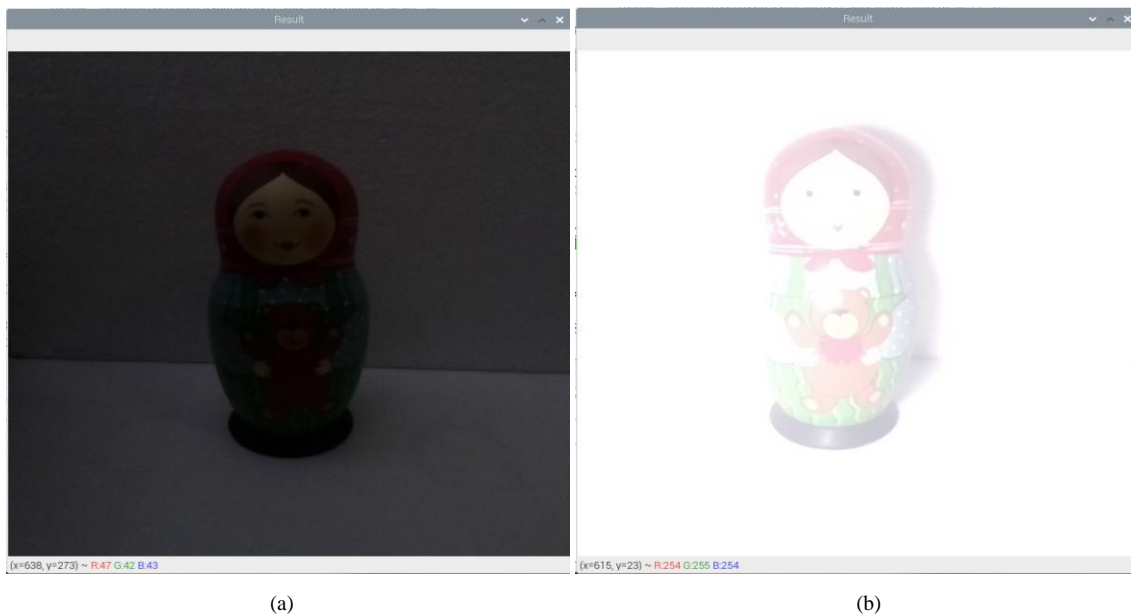
Program OCR PyTesseract berfungsi secara baik dan mampu menangkap pola “A”, “B”, “C”, “D”, dan “P” serta menghasilkan ketepatan olahan *picture-to-string*. Program kemudian menampilkan hasil olahan tersebut pada layar monitor. Program *object detection* juga berfungsi dengan sangat baik. Dalam proses pengenalan objek berbentuk *file* gambar hingga proses pendeteksian, program *object detection* mampu menghasilkan kotak deteksi dan keterangan nama objek, yaitu “Patung”, dengan tepat. Program juga telah berhasil menampilkan titik koordinat secara langsung (*real-time*) dari objek “Patung”, yaitu titik koordinat ( $x = 173, y = 64$ ) pada saat pendeteksian berlangsung.

### UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada pembimbing penelitian dan Program Studi Teknik Elektro Universitas Katolik Soegijapranata atas segala dukungan yang diberikan.

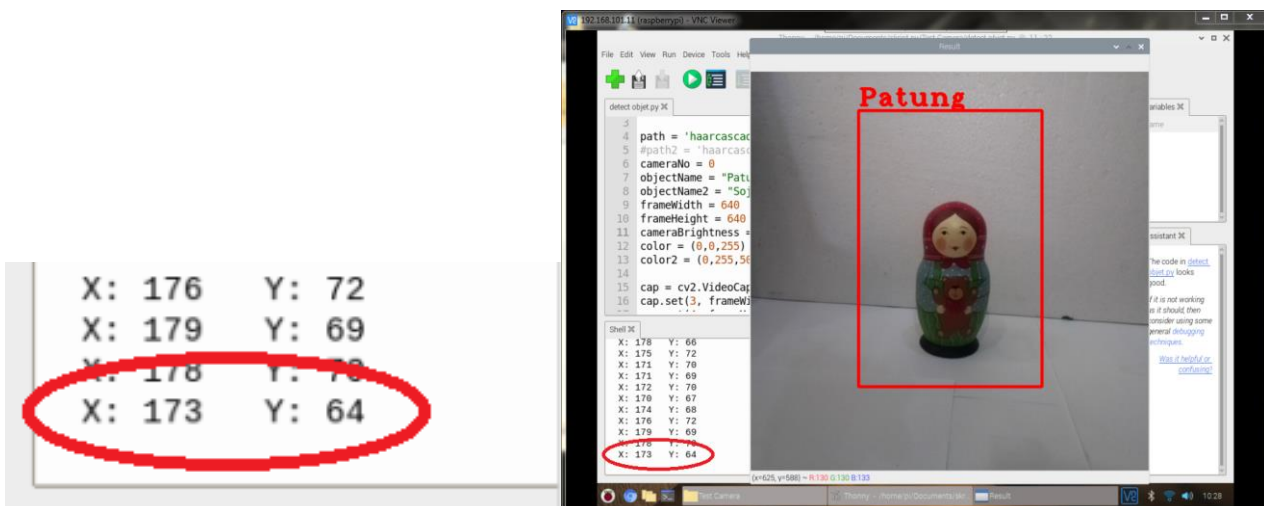
## REFERENSI

- [1] R. Szeliski, *Computer Vision : Algorithms and Applications*, London, UK: Springer-Verlag London, 2010.
- [2] M. Rafika, A. Rakhman, dan J. Endri, "Rancang dan Implementasi Pattern Recognition pada Garis Telapak Tangan untuk Akses Keamanan Pintu," *Proc. SNATIF Ke-4*, 2017, hal. 413–418.
- [3] K. Umam dan B.S. Negara, "Deteksi Obyek Manusia pada Basis Data Video Menggunakan Metode Background Subtraction dan Operasi Morfologi," *Jurnal CoreIT*, Vol. 2, No. 2, hal. 31–40, 2016.
- [4] (2020) "Datasheet Raspberry Pi 3 Model B Technical Specification," [Online], <http://www.farnell.com/datasheets/2027912.pdf>, tanggal akses: 16-Jun-2020.
- [5] A. Eames, *The Official Raspberry Pi Projects Book*, London, UK: Liz Upton, 2015.
- [6] X. Zhou, T. Chen, dan Y. Zhang, "Research on Intelligent AGV Control System," *Proc. 2018 Chinese Autom. Congr. CAC 2018*, 2019, hal. 58–61.
- [7] (2020) "Datasheet Raspberry Pi 4 Model B", [Online] <http://www.farnell.com/datasheets/2819352.pdf>, tanggal akses: 16-Jun-2020.
- [8] G. Arva dan T. Fryza, "Embedded Video Processing on Raspberry Pi", *Int. Conf. Radioelektronika*, 2017, hal. 4–7.
- [9] (2020) "The Open CV Tutorials Point," [Online] <https://www.tutorialspoint.com/opencv/opencvutorial.pdf>, tanggal akses: 16-Jun-2020.
- [10] M. Yang dan N. Ahuja, "Extraction and Classification of Visual Motion Patterns for Hand Gesture Recognition," *Proc. Computer Society Conference on Computer Vision and Pattern Recognition*, 1998, hal. 1–6.
- [11] Gaohe Li, Xinhao Li, dan Bo Xu, "Numerical Simulation Technology Study on Automatic Translation of Foreign Language Images Based on Tesseract-ORC," *Int. Conf. on Robots & Intelligent System (ICRIS)*, 2019, hal. 86–89.
- [12] G. Marne, R. Futane, B. Kolekar, D. Lakhadive, dan K. Marathe, "Identification of Optimal Optical Character Recognition (OCR) Engine for Proposed System," *Int. Conf. on Computing Communication Control and Automation (ICCUBEA)*, 2018, hal. 1–4.



(a)

(b)

Gbr. 13 Hasil pengujian *object detection* pada objek patung tidak terdeteksi, (a) intensitas cahaya terlalu rendah, (b) intensitas cahaya terlalu tinggi.Gbr. 14 Hasil pengujian titik koordinat dari objek berupa  $x$  dan  $y$ .