

Deteksi Citra dan Posisi Barcode Menggunakan Metode Oriented Fast and Rotated Brief (ORB) dan Maximally Stable Extremal Regions (MSER)

Hafizhal Tilaksosno, Berlian Al Khindi, dan Fivitria Istiqomah
Departemen Teknik Elektro Otomasi, Institut Teknologi Sepuluh Nopember (ITS)
e-mail: berlian@its.ac.id

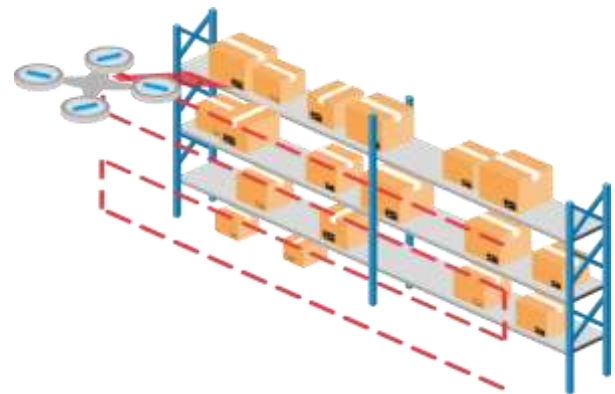
Abstrak—Warehouse yang digunakan untuk tempat menyimpan barang yang di kirim dari produsen ke konsumen. Dari kondisi yang ada di lapangan, antara barang yang ada di warehouse dan catatan inventaris sering terdapat kesalahan. Kesalahan yang kerap terjadi adalah salah penempatan barang yang disebabkan oleh kelalaian yang dilakukan oleh operator disana. Hal ini kerap menyebabkan kekeliruan antara inventaris dalam warehouse dan catatan inventaris yang dimiliki. Dalam pengecekan inventaris yang ada pada warehouse masih menggunakan operator dalam mengecek tiap palet yang ada pada rak setinggi kurang lebih 10 meter. Hal ini membutuhkan waktu rata rata 8 menit untuk melakukan pengecekan dalam satu rak yang terdiri dari 12 barcode. Sistem ini dirancang untuk membantu penyelesaian masalah tersebut dengan membuat drone yang dapat melakukan pengecekan secara otomatis dan memindai barcode menggunakan image processing beserta metode ORB dan MSER untuk mengetahui posisi barang tersebut. Dari hasil yang didapat sistem ini berhasil membaca rata rata 12 barcode dalam 4 menit. Dari yang semula dalam pengecekan manual 12 barcode membutuhkan waktu 8 menit, menghasilkan selisih 4 menit dalam pengecekan 12 barcode.

Kata Kunci—Warehouse, ORB, MSER, Barcode, Image Processing.

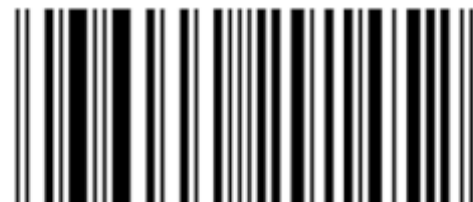
I. PENDAHULUAN

WAREHOUSE memiliki berbagai fungsi yang salah satunya merupakan tempat penyimpanan barang yang akan dikirim oleh produsen ke konsumen. Barang yang disimpan dalam warehouse ini terdapat berbagai macam jenis barang meliputi barang jadi atau setengah jadi, dokumen, dan hasil produksi. Barang yang disimpan dalam warehouse akan diberi label dan ditempatkan pada rak yang tersedia untuk mengelompokkan barang sesuai jenis dan jadwal pemberangkatannya.

Pada proses penempatan barang pada rak, terdapat beberapa proses yang akan dilalui di warehouse. Awal proses yang harus dilewati saat proses memasukan barang ke warehouse adalah pengelompokan barang yang sama untuk dijadikan dalam satu palet. Setelah penggabungan ini palet yang akan dimasukan ke dalam warehouse akan diberi label berupa barcode untuk tanda pengenalan barang tersebut. Kemudian barang akan diletakan dalam rak sesuai dengan pengelompokannya. Proses pengecekan ini dimulai dari pengecekan barcode pada tiap palet yang ada di rak. Lalu kemudian data yang diambil oleh scanner akan masuk ke dalam excel yang akan dicocokkan dengan catatan inventaris yang ada pada proses loading in barang. Hal ini dilakukan secara berkala guna mencocokkan data barang yang ada pada warehouse dan data yang ada pada catatan inventaris.



Gambar 1. Arah drone dalam pembacaan rak.

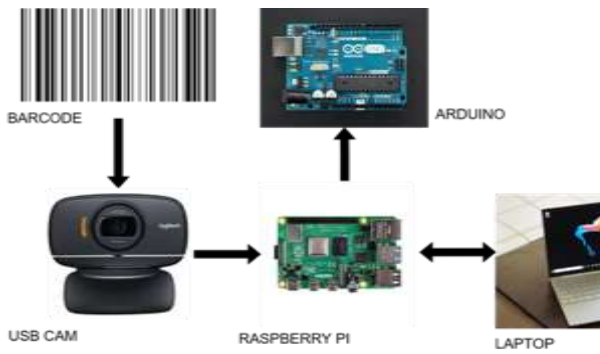


Gambar 2. Contoh barcode.

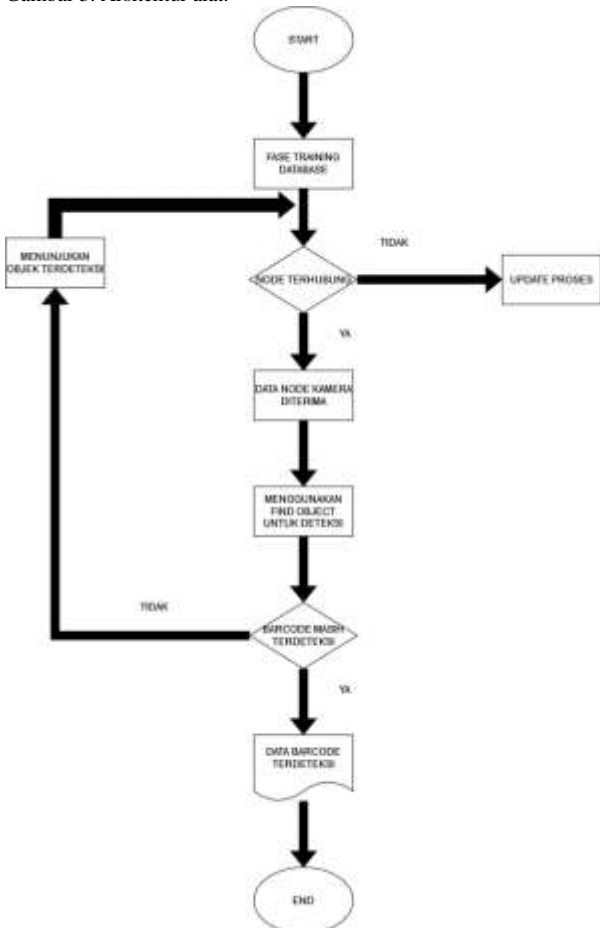
Dari kondisi pengecekan yang ada, terkadang ada kesalahan dalam perhitungan barang yang ada dengan catatan inventaris yang ada. Hal ini disebabkan kesalahan dalam penempatan palet ke dalam rak yang berbeda sehingga menciptakan kebingungan di dalam warehouse. Di dalam warehouse masih menggunakan pengecekan manual dengan operator yang akan berjalan dan menggunakan forklift dalam menggapai rak yang tinggi. Hal ini membutuhkan waktu sekitar 10 menit dalam melakukan pengecekan dalam satu rak yang memiliki rata rata 12 barcode barang didalamnya.

Dengan semakin berkembangnya zaman maka hal hal diatas dapat di bantu dengan menggunakan pengembangan agar mempermudah sistem manajemen inventaris serta mengurangi intervensi dari manusia atau dapat dikatakan ditambahkan proses otomasisasi, dalam hal ini proses otomasisasi yang akan digunakan adalah proses pengecekan barcode dalam rak yang akan memunculkan posisi dari barcode tersebut untuk memudahkan dalam pencocokan antara barang dalam rak dan dalam catatan inventaris. Metode deep learning digunakan untuk membantu proses image processing dalam menemukan gambar barcode yang ada pada rak. Drone juga akan membantu dalam proses pengecekan barcode pada rak yang ada sehingga operator tidak perlu menggunakan forklift dalam menggapai barang yang tinggi.

Selain proses otomasisasi ini, hasil dari pembacaan barcode dapat ditampilkan pada website untuk mencocokkan



Gambar 3. Arsitektur alat.



Gambar 4. Flowchart pembacaan barcode.

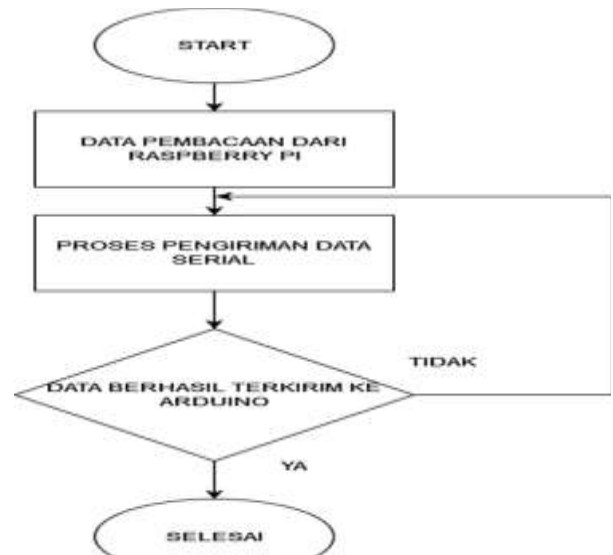
dengan catatan yang ada. Hal ini dapat mengurangi biaya pada forklift yang digunakan untuk melakukan pengecekan pada rak.

II. TEORI PENUNJANG

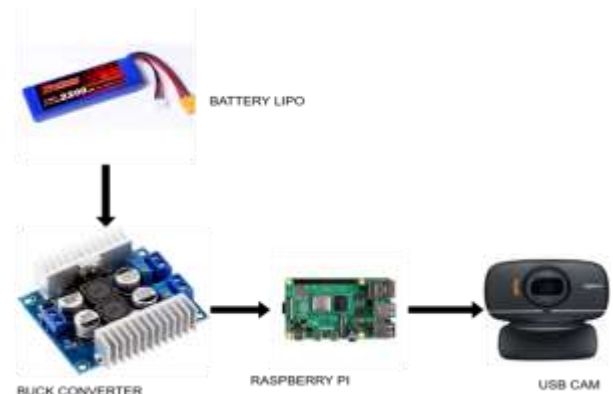
A. Penelitian Terdahulu

Penelitian terdahulu terdiri dari: Pada Pengerjaan proyek akhir ini, penulis mengambil beberapa referensi berupa penelitian yang telah dilakukan sebelumnya.

1. Jurnal berjudul *Inventory Management through Mini-Drones: Architecture and Proof-of-Concept Implemetation*. Jurnal ini ditulis oleh Davide Cristiani, Filippo Bottoneli, Angela Trotta, dan Marco Di Felice ini menjelaskan tentang kegunaan UAV dalam satu dekade terakhir. Persediaan gudang menggunakan UAV dianggap sebagai penelitian baru, diantara solusi industri yang relevan mengutip dari *Flytbase* [1].



Gambar 5. Flowchart pengiriman data.



Gambar 6. Perancangan alat proyek akhir.

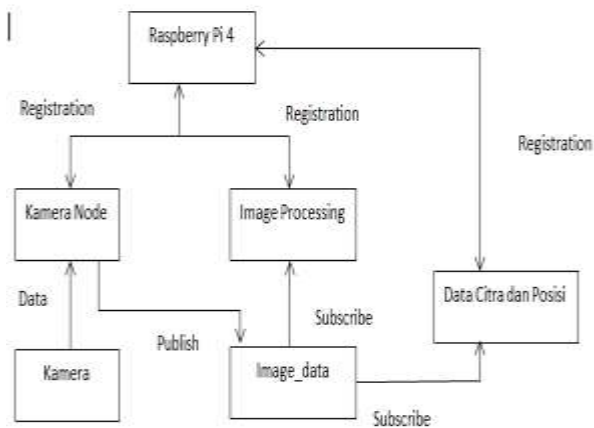
2. Jurnal berjudul *Applications of Drones in Warehouse Operations*. Jurnal yang ditulis oleh Lukas Wawrla, Omid Maghazei, dan Prof. Dr. Torbjorn Netland berisi penelitian sistem warehouse menggunakan *cycle counting* [2].
3. Jurnal berjudul *2D Barcode Detection using Images for Drone-assisted Inventory Management*. Jurnal yang ditulis Hyeon Cho, Dongyi Kim, Junho Park, Kyungshik Roh, dan Wonjun Hwang. Jurnal berisi deteksi 2 dimensi untuk menentukan letak barcode pada barang yang ada di gudang. Menggunakan kamera untuk membantu menentukan letak barcode yang ada pada barang didalam palet gudang. Jurnal ini menjelaskan tentang penentuan kandidat gambar yang akan diambil untuk memproses citra barcode tersebut [3].
4. Jurnal berjudul *Real-Time Barcode and Classification using Deep Learning*. Jurnal ini ditulis oleh Daniel Kold Hansen, Kamal Nasrollahi, Christoffer B. Rasmusen dan Thomas B. Moeslund. Jurnal ini berisi perbandingan antara 2 metode *deep learning* yang digunakan untuk membaca barcode [4].

B. Dasar Teori

Dasar teori terdiri antara lain:

1) Warehouse Management

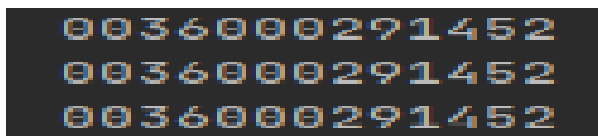
Sistem manajemen gudang ditunjukkan untuk menangani dan mengendalikan inventaris dalam gudang. Warehouse



Gambar 7. Blok diagram proses.



Gambar 8. Hasil pembacaan barcode.



Gambar 9. Hasil decode dari gambar 10.

Management berkaitan dengan pergerakan, pengambilan, pengepakan, dan penyimpanan stok di dalam gudang. Fungsi dasar dalam sistem manajemen pergudangan adalah:

1. Pemenuhan permintaan pelanggan lebih cepat
2. Penentuan lokasi dan ketersediaan stok
3. Pelaporan inventory dan reorder tepat waktu

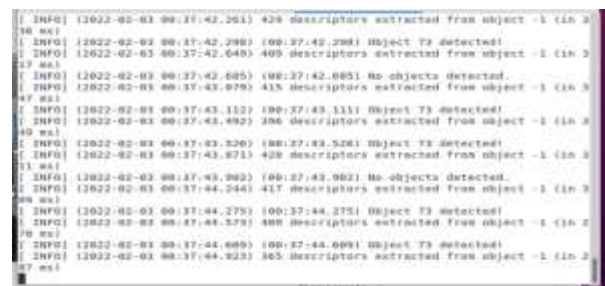
Sistem ini menangani pekerjaan utama dari sebuah gudang seperti manajemen penyimpanan, manajemen barang berbahaya, pemrosesan barang, keluar dan masuknya material, pengambilan stok, pemeriksaan dan pengisian ulang. Sistem penyimpanan barang yang baik diperlukan untuk membuat arus barang tertata rapi dan teratur serta dapat diperiksa sewaktu waktu.

Salah satu cara dalam sistem manajemen gudang adalah melakukan kembali pengecekan sebelum masuk ke gudang. Pengecekan ini dapat dilakukan dengan cara memberikan kode agar perusahaan dapat mengecek ada atau tidaknya barang yang dicatat dalam catatan stok barang di gudang. Pengecekan ini dapat membantu perusahaan dalam mengecek kondisi barang sebelum di letakan dalam gudang. Dalam sistem manajemen gudang juga terdapat penerapan tata letak yang optimal. Yang harus dilakukan pertama adalah mengetahui karakter dalam stok barang tersebut, kemudian perusahaan dapat mengatur layout warehouse yang dimiliki.

Pencatatan secara rutin dilakukan setiap ada barang masuk, barang dipindahkan, dan barang keluar. Hal ini dapat dilakukan secara berkala sesuai kebutuhan dalam melakukan manajemen stok barang di gudang. Stock Opname

Tabel 1.

Hasil pengujian menggunakan webcam				
Uji Ke.	Tulisan barcode	Hasil decode	Keterangan	Akurasi
1.	03600029145	03600029145	Terbaca	
2.	7396032301		Tidak	
3.	0592752312		Terbaca	
4.	1234567890128	1234567890128	Terbaca	
5.	72527273070	72527273070	Terbaca	
6.	811204012344	811204012344	Terbaca	
7.	00223393	00223393	Terbaca	
8.	013700206252	013700206252	Terbaca	
9.	9501101530003		Tidak	
10.	780863185779	780863185779	Terbaca	85%
11.	0123456	0123456	Terbaca	
12.	9781234567897	971234567897	Terbaca	
13.	9789791935159	978791935159	Terbaca	
14.	09501101530003	09501101530003	Terbaca	
15.	61076256932	61076256932	Terbaca	
16.	A23342453D	A23342353D	Terbaca	
17.	A1234567890A	A1234567890A	Terbaca	
18.	9770123456787	9770123456787	Terbaca	
19.	ABC123	ABC123	Terbaca	
20.	8991609119892	8991609119892	Terbaca	



Gambar 10. Tampilan deteksi barcode ROS.

merupakan proses berkala dalam melakukan perhitungan fisik barang yang ada kemudian hasil tersebut akan dibandingkan dengan jumlah yang ada pada catatan persediaan. Menggunakan opsi pelacakan bergerak akan memungkinkan untuk melihat dimana produk tersebut ditempatkan. Dalam pelacakan barcode, drone akan menyisir setiap rak yang ada pada warehouse. Hal ini dapat dilihat pada Gambar 1.

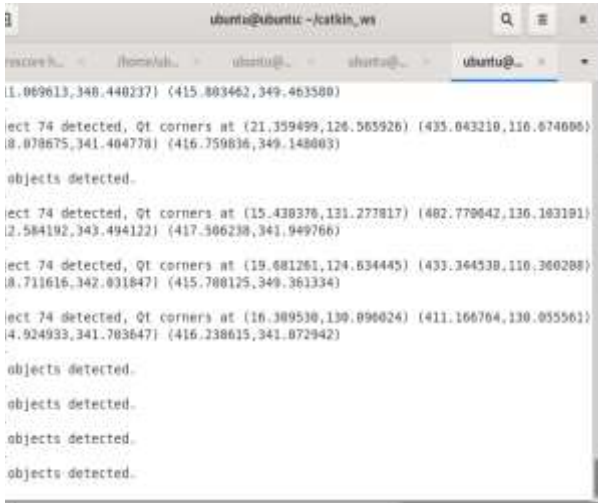
2) Barcode

Barcode adalah sebuah teknologi data optik yang memiliki garis atau strip yang dapat dibaca oleh mesin. Kode yang berbeda ini berfungsi membedakan satu jenis produk dengan produk lainnya. Kode bar dicetak menggunakan label barcode.

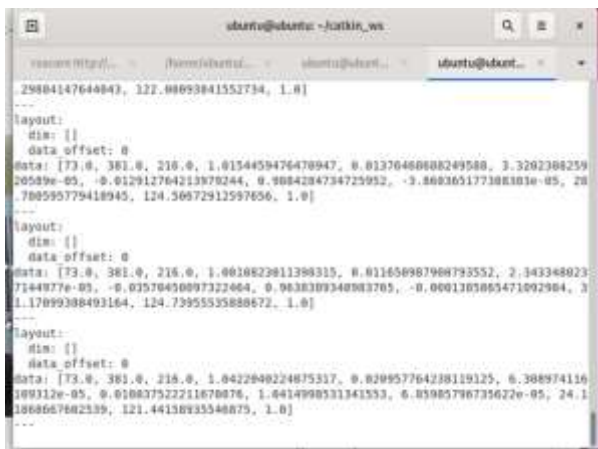
Sistem penyortiran ini pertama kali digunakan oleh Wallace Flint dalam karya ilmiahnya pada tahun 1930-an. Barcode digunakan untuk mendata barang yang masuk ke dalam gudang serta mengelompokkan barang sesuai dengan jenisnya. Contoh barcode sederhana dapat dilihat pada Gambar 2.

3) ROS

ROS (Robot Operating System) adalah suatu sistem yang dapat mengendalikan robot dari PC. Sistem dari ros itu sendiri terdiri dari node node yang saling terhubung dan berkomunikasi dengan cara publish dan subscribe. Dalam pembacaan barcode ini data string akan di publish dan node lain dapat mengkonsumsi data ini dengan mudah.



Gambar 11. Data *QT corners* dalam menentukan *rectangle*.



Gambar 12. Tampilan data posisi *barcode* pada kamera.

Kenapa kebanyakan orang memilih menggunakan ROS dibandingkan dengan aplikasi lain mengingat ROS ini dapat digunakan hanya didalam *Operating System* Ubuntu. Ini merupakan keunggulan dari fitur ROS, yaitu node. Node tidak harus berada pada sistem yang sama atau dapat digunakan dalam beberapa komputer. Contoh ketika ingin menangkap data dari sensor, komputer lain, atau bahkan dari Arduino maka komputer yang kita gunakan akan menggunakan fitur subscribe dan membuat sensor, komputer lain, atau Arduino tersebut sebagai node yang digunakan untuk menerbitkan data mereka sehingga dapat di subscribe oleh komputer yang kita gunakan.

4) *OpenCV*

OpenCV (*Open Source Computing Vision Library*) adalah sebuah *library* berlisensi *BSD open-source* yang mencakup beberapa ratus algoritma visi komputer. Terdapat banyak fitur yang ada pada *OpenCV* dalam memproses gambar. Salah satunya adalah *MSER* dan *ORB* yang dapat digunakan dalam pembacaan *barcode*.

5) *MSER*

Metode *MSER* telah diidentifikasi sebagai salah satu detektor wilayah terbaik karena ketahanannya terhadap perubahan sudut pandang, skala, dan pencahayaan, serta sensitif terhadap citra yang kabur. Deteksi ini merupakan dasar dari banyak tugas visi komputer yang penting. Contohnya pengenalan objek, pencocokan stereo, mosaik,

Tabel 2.

Hasil pengujian <i>barcode</i> menggunakan ROS		
Uji Ke	Jumlah Barcode	Presentase Keberhasilan
1.	12	41%
2.	12	50%
3.	12	67%



Gambar 13. Contoh penempatan *barcode* pada ruangan.

pelacakan objek, dan lain lain bergantung pada deteksi poin poin minat yang memiliki perbedaan, sangat invariant, dan property yang stabil. Struktur tersebut sering disebut sebagai *DR (Distinguishhed Region)* dan memberikan representasi pola pada suatu gambar. *MSER* melakukan performa terbaik dalam berbagai urutan tes. *MSER* menunjukkan satu set daerah yang didefinisikan oleh sifat ekstrim dari fungsi intensitasnya di wilayah dan di batas luarnya. Selain itu, *MSER* memiliki semua sifat yang dibutuhkan dari detektor stabil.

Berbeda dari deteksi sudut di bagian terakhir, *MSER* menggunakan konsep yang mirip *DAS* untuk mendapatkan daerah yang stabil secara lokal. Algoritma *DAS* dalam pengolahan gambar adalah terutama digunakan dalam segmentasi gambar dan fokus pada “tingkat air” gambar tingkat abu abu ketika daerah bergabung, tetapi daerah wilayah tidak stabil sementara *MSER* fokus pada daerah “permukaan air” ketika daerah daerah tersebut stabil [5].

6) *ORB*

Fitur biner lokal memiliki beberapa keunggulan dibandingkan histogram fitur berbasis gradient karena mereka bisa lebih cepat untuk dihitung, lebih kompak untuk disimpan, dan lebih efisien untuk membandingkan. Meskipun, cepat untuk menghitung jarak Hamming antara pasangan fitur biner. String bit deskriptor biner $BD = (bit1, \dots, bitM)$ diperoleh dengan banyak gambar perbandingan intensitas antara piksel pasangan (p_i, q_i) sekitar dari posisi *keypoint* setelah merapikan gambar *I*. Biar $bit_i = 1$ jika $I(p) - I(q) > 0$ dan 0 sebaliknya [6]. *ORB* meningkatkan invariansi rotasi dengan orientasi computing vektor berdasarkan metode intensitas sentroid yang didefinisikan sebagai berikut (1).

$$m_{n_1 n_2} = \sum_{xy} x^{n_1} y^{n_2} I(x, y) \tag{1}$$

III. PERANCANGAN DAN IMPLEMENTASI

Metode penelitian Proyek akhir yang dilakukan adalah dapat mengambil citra dan posisi dari *barcode* serta

memproses pembacaan *barcode*. Dengan cara mengambil gambar pada objek yaitu *barcode* yang pembacaanya diproses menggunakan Raspberry dan ditampilkan di *website*.

A. Perancangan Sistem

Sistem membacanya menggunakan kamera untuk mendeteksi *barcode*, objek yang terdeteksi pada kamera akan diproses citra *barcode* yang didapat agar dapat menentukan posisi dari *barcode* tersebut. Segmentasi merupakan pengambilan database untuk Raspberry (*controller* nya) untuk mengenali objek (*barcode*) dimana data yang diambil berupa data string pembacaan dan data posisi, yang nantinya akan dicocokkan ke dalam database *website*. Data posisi dari *barcode* akan di upload ke dalam *website*, hal ini akan mempermudah proses pengecekan inventaris dimana nantinya bisa di tampilkan pada monitor pc. Untuk pengolahannya menggunakan OpenCV. Selain itu proses tadi dapat di pantau dan di kontrol secara *realtime* menggunakan *remote dekstop* dimana nantinya Raspberry dan PC (user) saling terhubung melalui LAN yang ada dengan host IP yang sudah di setting. Hal ini dapat dilihat di dalam Gambar 3.

B. Flowchart Sistem

Pada Gambar 5, merupakan flowchart pada proses pada pemrosesan citra. Saat sistem dimulai akan mengambil citra dan data posisi barcode menggunakan kamera. Lalu citra dan posisi barcode akan diproses didalam Raspberry menggunakan OpenCV. Data citra dan posisi barcode akan diproses di dalam Raspberry yang akan mengirim data tersebut ke website. Setelah data terkirim, maka data akan ditampilkan ke website yang tersedia sebagai tampilan akhir dari proses. Ketika citra dan posisi barcode tidak terdeteksi, maka proses pengiriman ke Raspberry Pi akan dilewati sehingga dianggap tidak dapat membaca citra serta data posisi dari barcode. Hal ini dapat dilihat pada Gambar 4. Setelah data berhasil masuk ke Raspberry Pi, maka data tersebut akan dikirim ke Arduino untuk ditampilkan ke dalam website. Flowchart ini dapat dilihat pada Gambar 5.

C. Perancangan Alat Proyek Akhir

Dari alat yang digunakan dalam proyek akhir ini, drone yang akan digunakan untuk proyek akhir sebagai berikut:

1. Battery LiPo 16 V
2. Raspberry Pi 4
3. Usb Camera Logitech B525
4. Buck Converter

Dalam perancangan hardware yang digunakan dari komponen penting yaitu drone dan Raspberry Pi akan membentuk suatu kesatuan yang akan menjalankan proses tracking. Dalam perancangan elektrikal ini dapat dilihat pada Gambar 6.

Proses tracking ini menggunakan berbagai cara agar barcode dapat diambil citra dan data posisi. Tracking ini dirancang untuk membaca objek deteksi yaitu barcode. Pembacaan barcode dilakukan dengan menggunakan USB Camera yang menangkap gambar lalu diproses ke dalam Raspberry Pi untuk proses pengambilan data citra dan posisi barcode. Data barcode yang berupa posisi dibutuhkan untuk mengetahui posisi palet yang digunakan dalam manajemen inventaris. Hal ini dibutuhkan untuk menentukan posisi

barang pada rak yang diberi barcode tersebut. Diagram blok pada proses tracking dapat dilihat pada Gambar 7.

IV. PENGUJIAN DAN ANALISA

A. Pengujian Pembacaan Barcode

Dalam pengujian *barcode* ini menggunakan sampel barcode sederhana yang diambil dari internet untuk melihat apakah kamera bisa menangkap citra *barcode* dan *decode* dari *barcode* tersebut. Contoh tampilan pembacaan terdapat pada Gambar 8. Dari pengujian yang dilakukan, menggunakan sampel barcode dari internet secara acak. Dalam pembacaan kamera berhasil menangkap gambar barcode dan mengubah menjadi data decode. Hasil ini dapat dilihat pada Gambar 9.

Tujuan dari pengujian ini adalah untuk mengetahui apakah kamera dapat mengambil gambar kandidat barcode secara *realtime*. Hasil dari pengujian terhadap 20 barcode dapat dilihat pada Tabel 1.

B. Pengujian pada ROS

Dalam pengujian menggunakan ROS ini menggunakan sampel barcode yang akan diolah oleh ROS. Barcode yang digunakan dimasukan ke dalam database untuk mengklasifikasi objek yang akan dibaca. Setiap database barcode yang akan dibaca akan dimasukan secara manual ke dalam program pembacaan.. Hasil dari pengujian ini akan menunjukkan posisi barcode pada kamera yang digunakan. Hal ini dapat dilihat pada Gambar 10.

Pada tampilan ini memperlihatkan *frame* kamera dunia nyata dan *database* yang diambil secara manual dengan mengambil citra *barcode* lalu dimasukan ke dalam *database*. Dalam pengujian ini penulis menggunakan *barcode* dengan label nomor 18 pada *database*. Ketika *barcode* tersebut terbaca oleh kamera, maka terminal akan menampilkan tulisan "Object 18 detected!" seperti pada Gambar 11 diatas. Dalam hal ini menggunakan penggabungan antara metode MSER dan ORB. Gabungan kedua metode ini akan membentuk beberapa titik titik antara garis garis *barcode* yang akan dianggap sebagai deteksi antara kamera dan *database* dalam mendeteksi *barcode* tersebut. Setelah *barcode* terdeteksi maka dapat memunculkan data berupa *QT corners* yang menunjukkan antara *frame barcode* dan *frame* pada dunia nyata. Data *QT corners* ini berupa data angka yang menunjukkan posisi *rectangle* dalam *frame barcode*. Data angka ini dapat dilihat pada Gambar 11.

Dalam Gambar 13 menunjukkan terminal pada Ubuntu yang menunjukkan data angka dari *rectangle barcode* nomor 18. Jika dalam frame kamera tidak terdeteksi kecocokan antara gambar dunia nyata dengan gambar pada *database* maka akan menunjukkan tulisan "No objects detected.". Hal ini digunakan untuk menunjukkan apakah kamera pada saat proses *tracking* berhasil menangkap gambar *barcode* atau tidak. Ketika data *rectangle* berhasil ditangkap, maka data yang akan dikeluarkan selanjutnya adalah data letak posisi *barcode* pada kamera. Letak posisi ini dilakukan dengan menambahkan data yang tertera pada terminal Ubuntu ke dalam matriks yang telah disediakan dibawah ini.

$$H = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x1 \\ y1 \\ 1 \end{bmatrix} = H \begin{bmatrix} x2 \\ y2 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x2 \\ y2 \\ 1 \end{bmatrix} \quad (2)$$

Dalam matriks ini H merupakan matriks 3×3 dari gambar yang tersimpan dan (x_2, y_2) adalah perhitungan posisi dari objek pada *frame* kamera. Hal ini dapat dilihat pada Gambar 12.

Dalam 3 kali percobaan yang dilakukan, banyak terjadi kegagalan dalam pengambilan citra dikarenakan komputasi dan kamera yang kurang dalam melakukan *image processing*. Dalam hal ini dalam setiap percobaan mendapat hasil yang berbeda beda dalam presentase keberhasilan. Berikut adalah data yang disajikan dalam bentuk Tabel 2. Didapatkan bahwa jumlah kumulatif keberhasilan dari ke 3 percobaan yang dilakukan adalah 52,6%.

Dalam pengujian ini penulis mendapatkan bahwa ada faktor yang mempengaruhi dari pengambilan citra secara optimal, yaitu cahaya, posisi kamera, ukuran *barcode*, dan komputasi yang optimal. Ketika semua hal ini dapat dimaksimalkan maka proses pengambilan citra yang dilakukan dapat dengan baik dijalankan oleh PC. Dalam percobaan ini penulis menggunakan ruangan *indoor* agar menyerupai *warehouse* yang notabennya juga bersifat *indoor*. Pemetaan *barcode indoor* ini dapat dilihat pada Gambar 13.

V. KESIMPULAN

A. Kesimpulan

Setelah dilakukan pengambilan sampel deteksi barcode ini mendapat kesimpulan bahwa dalam rata rata deteksi per 12

barcode mendapatkan rata rata waktu 3 menit. Pengujian menggunakan ROS ini juga mendapatkan presentase 55,6% dalam 3 kali percobaan dengan sampel yang sama. Posisi yang diambil tidak dapat dibaca dikarenakan dalam kondisi indoor GPS tidak bisa terbaca yang mengakibatkan harus menambahkan komponen sensor VL6180X, V15310x, dan bumper untuk membentuk map dalam ROS untuk mendapatkan data posisi barcode.

DAFTAR PUSTAKA

- [1] D. Cristiani, F. Bottonelli, A. Trotta, and M. Di Felice, "Inventory Management through Mini-Drones: Architecture and Proof-of-Concept Implementation," in *21st IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2020*, 2020, pp. 317–322, doi: 10.1109/WoWMoM49955.2020.00060.
- [2] L. Wawrla, O. Maghazei, and P. D. T. Netland, "Applications of Drones in Warehouse Operations," ETH Zurich, 2019.
- [3] H. Cho, D. Kim, J. Park, K. Roh, and W. Hwang, "2D Barcode Detection using Images for Drone-assisted Inventory Management," in *2018 15th International Conference on Ubiquitous Robots, UR 2018*, 2018, no. June, pp. 461–465, doi: 10.1109/URAI.2018.8441834.
- [4] D. K. Hansen, K. Nasrollahi, C. B. Rasmussen, and T. B. Moeslund, "Real-Time Barcode Detection and Classification Using Deep Learning," in *Proceedings of the 9th International Joint Conference on Computational Intelligence (IJCCI 2017)*, 2017, pp. 321–327, doi: 10.5220/0006508203210327.
- [5] H. Guo and L. Chen, "Object recognition Based on maximally stable extremal region and scale-invariant feature transform," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 14, no. 2, pp. 622–629, 2016, doi: 10.12928/telkomnika.v14i2.2754.
- [6] M. Hamidia and N. Zenati-henda, "Object Recognition Based on ORB Descriptor for Markerless Augmented Reality," in *9eme Conference sur le Genie Electrique*, 2015, no. April, pp. 1–6.