

Pemodelan Deteksi Dan Identifikasi Multiple Barcode

M Taufiq Maulana Fahmi¹ · Daniel Swanjaya²

Teknik Informatika, Fakultas Teknik, Universitas Nusantara PGRI Kediri

E-mail: ¹fahmf.official@gmail.com ²daniel@unpkediri.ac.id,

Abstrak – Membludaknya pasar online selama pandemi ini, membuat beberapa jasa pengiriman mengalami overload pengiriman. Ini mengakibatkan terlambatnya paket datang kepada customer. Dalam pengiriman, perpindahan paket akan di scan sesuai dengan barcode yang menempel pada paket. Langkah tersebut akan sangat lama bila di lakukan satu persatu, maka jika langkah tersebut bisa dilakukan secara bersamaan akan sangat menghemat waktu dalam pengiriman. Langkah tersebut bisa dilakukan menggunakan sebuah library pada python yang bernama ZBar. Nantinya library ini akan bisa memindai banyak barcode dalam satu langkah scan. Sehingga akan menghemat waktu yang dibutuhkan saat pengiriman.

Kata Kunci - Barcode, ZBar, Multiple Scan.

1. PENDAHULUAN

Internet merupakan produk teknologi yang banyak dimanfaatkan oleh masyarakat saat pandemi terjadi. Termasuk dalam sektor ekonomi juga sangat berpengaruh, dimulai dari banyaknya pengusaha, dan pelaku UMKM yang mulai memindahkan toko penjualan mereka secara online, hal ini dilakukan karena adanya aturan yang tidak memperbolehkan orang – orang berinteraksi secara langsung. Sebagian besar Usaha Mikro, Kecil, dan Menengah (UMKM) mengaku telah menggunakan perangkat yang terhubung internet untuk menjalankan usahanya. Dalam survei *Katadata Insight Center* (KIC), sebanyak 42,7% responden memanfaatkan PC/laptop dengan internet untuk menjajakan dagangannya. Sementara 84% responden sebagai pelaku UMKM menggunakan ponsel pintarnya untuk menjalankan usahanya dengan bantuan internet [1].

Kemajuan teknologi komunikasi disatu sisi memberikan dampak positif seperti memudahkan manusia untuk berinteraksi antara satu dengan yang lain, memudahkan manusia untuk menjalankan aktivitasnya, dan memudahkan manusia untuk memperoleh informasi yang dibutuhkan. Era industri 4.0 memberikan efek pada perkembangan teknologi dalam berbagai bidang di Indonesia. Cara pandang pada bisnis telah berubah semenjak perusahaan-perusahaan digital menguasai bisnis digital. Momen itu juga mempengaruhi banyak orang untuk mendirikan usaha dengan mengadopsi teknologi digital. Dengan begitu, mereka tetap bisa bersaing dalam persaingan global. Tren pertumbuhan UMKM yang positif turut berimbas pada pertumbuhan ekonomi Indonesia. Menurut data kementerian perindustrian Republik Indonesia dalam 5 tahun

terakhir, UMKM telah memberikan kontribusi produk domestic bruto sebesar 60.34%. [2].

Berbelanja online akan mengakibatkan melonjaknya jumlah pesanan pada setiap penjual online, dan secara langsung akan banyak sekali paket di setiap layanan jasa pengiriman, hal ini akan semakin bertambah saat setiap marketplace atau pasar online mengadakan promosi. Promosi penjualan juga digunakan untuk mengurangi biaya penyimpanan inventaris penjual dengan mengarahkan konsumen ke persediaan barang [3]

Tak dapat dimungkiri, jasa pengiriman online menjadi salah satu faktor meningkatkan gairah jual-beli usaha mikro, kecil, dan menengah (UMKM). Kehadiran layanan tersebut, memungkinkan pelaku UMKM menjangkau pembeli lebih jauh, dan lebih banyak. Jasa pengiriman online yang ditawarkan saat ini sangat banyak dan beragam. Pelaku usaha dapat dengan mudah memilih jasa pengiriman mana yang cocok dan sesuai kebutuhan. Jasa pengiriman barang juga telah banyak melakukan inovasi-inovasi mengikuti perkembangan zaman. Memanfaatkan teknologi sebagai penunjang fasilitas yang mereka tawarkan. Memang, harus diakui dengan pesatnya kemajuan teknologi, lanskap jasa pengiriman kian berubah lebih mudah dan serba online. Bukan pada prosesnya, tapi pada cara memanfaatkannya. Sebab, proses pengiriman barang tentu tidak mengalami perbedaan. Saat ini, pelaku usaha dapat dengan mudah memantau barang kirimannya. Jasa pengiriman telah menyediakan layanan yang dapat memantau keberadaan barang yang dikirim dengan sistem *tracking real time*. Di mana, dengan layanan tersebut, pelaku usaha juga memperoleh estimasi waktu tiba paket di alamat tujuan [4].

Promosi bulanan seperti 12.12, 7.7 yang di selenggarakan setiap bulan dan tanggal yang sama. Pada hari tersebut kemungkinan melonjaknya jumlah paket di setiap layanan jasa pengiriman adalah pasti, dan biasanya paket akan mengalami keterlambatan. Hal ini biasanya disebabkan pemrosesan paket atau scanning saat tiba di setiap drop point layanan jasa pengiriman dilakukan satu persatu dengan jumlah paket yang sedemikian banyaknya [4].

Penelitian ini bertujuan untuk membuat model yang memungkinkan penghematan waktu saat proses scanning karena proses scan yang dilakukan saat ini masih satu persatu, dapat dilakukan secara bersamaan. Penelitian menggunakan library Zbar untuk proses pembacaan barcode dan bahasa Python untuk mengolah data.

2. METODE PENELITIAN

Pada penelitian ini, pendekatan yang digunakan adalah pendekatan *kuantitatif* dan *kualitatif*, karena dalam penelitian ini didasarkan pada fenomena dalam lingkup bisnis online dan mengukur keakuratan hasil dari penelitian yang menggunakan angka sebagai tolok ukur keberhasilan penelitian.

Dalam penelitian ini lebih membahas tentang pengoptimalisasian penggunaan *library* yang telah ada, dan penerapannya ke dalam bahasa pemrograman *python*. *Library* yang ada dalam bahasan adalah *library* yang memang digunakan khusus untuk pemrosesan *barcode* atau *qr code*. Namun dalam bahasan ini akan lebih condong ke pembahasan barcode, karena sebagian besar jasa pengiriman masih menggunakan barcode dalam mengkodekan nomor resi.

2.1 Barcode

Barcode adalah sebuah bentuk *artificial identifier*. Barcode merupakan sebuah kode mesin yang dapat dibaca. Barcode terdiri dari sebuah bentuk bar dan spasi (hitam dan putih) dalam rasio yang didefinisikan yang merepresentasikan karakter *alphanumeric*. Barcode pertama kali muncul dan diperkenalkan oleh dua orang mahasiswa *Drexel Institute of Technology Bernard Silver* dan *Norman Joseph Woodland* pada tahun 1948. Mereka mempatenkan inovasi barcode tersebut pada tahun 1949 dan permohonan tersebut baru dikabulkan pada tahun 1952. Namun baru pada tahun 1996, penemuan mereka mulai digunakan dalam dunia komersial. Barcode pada dasarnya adalah susunan garis vertikal hitam dan putih dengan ketebalan yang

berbeda, sangat sederhana namun sangat berguna, dengan kegunaannya untuk menyimpan data-data spesifik misalnya seperti kode produksi, tanggal kadaluwarsa, serta nomor identitas lingkaran konsentris atau tersembunyi dalam sebuah gambar. [5]

2.2 Python

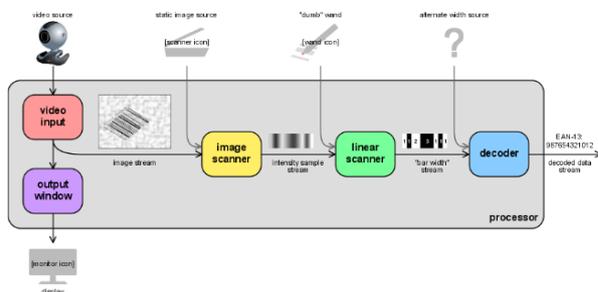
Python merupakan bahasa pemrograman tingkat tinggi yang sangat fleksibel. Python dikembangkan dan dibuat lebih lanjut oleh Guido Van Rossum, beliau adalah seorang programmer yang berasal dari Belanda. Dibuat dan dikembangkan pada tahun 1990 di kota Amsterdam. Lalu ditahun 1995, Python dikembangkan lagi untuk meningkatkan kompatibilitas dari program. Python sangat cocok digunakan untuk pembuatan program yang membutuhkan waktu singkat, karena bahasa pemrograman Python dilengkapi dengan modul dan paket yang mempermudah dalam pembuatan software. Python sendiri merupakan bahasa pemrograman yang tidak menggunakan compiler. Proses pengembangan bahasa python dikembangkan secara berkesinambungan oleh programmer, penguji dan pengguna yang kebanyakan bukan ahli IT di seluruh dunia hingga akhirnya menjadi seperti sekarang. Bahasa ini merupakan bahasa pemrograman yang tidak menggunakan compiler. Dengan sifat open-source yang dimilikinya, bahasa ini dapat digunakan untuk melakukan berbagai hal dari mulai mengembangkan web, mengembangkan video game, membangun GUI desktop, maupun mengembangkan perangkat lunak. Beberapa fitur yang dimiliki Python adalah :

- Memiliki kepastakaan yang luas, dalam distribusi Python telah disediakan modul-modul siap pakai untuk berbagai keperluan.
- Memiliki tata bahasa yang jernih dan mudah dipelajari
- Memiliki aturan *layout* kode sumber yang memudahkan pengecekan, pembacaan kembali dan penulisan ulang kode sumber
- Berorientasi obyek
- Memiliki sistem pengelolaan memori otomatis (garbage collection, seperti java).
- Modular, mudah dikembangkan dengan menciptakan modul-modul baru, modul-modul tersebut dapat dibangun dengan bahasa Python maupun C/C++.

2.3 Zbar

ZBar adalah rangkaian perangkat lunak sumber terbuka untuk membaca kode batang dari berbagai

sumber, seperti aliran video, file gambar, dan sensor intensitas mentah. Ini mendukung banyak simbologi populer (jenis kode batang) termasuk *EAN-13 / UPC-A, UPC-E, EAN-8, Kode 128, Kode 39, Interleaved 2 dari 5 dan Kode QR*.



Gambar 1 Algoritma Pustaka Zbar

Library ZBar menggunakan pendekatan yang lebih mendekati yang digunakan oleh pemindai "tongkat" dan "laser": kode batang linier (1D) dirancang untuk diterjemahkan oleh sensor cahaya sederhana yang melewati area terang dan gelap dari sebuah simbol. Memanfaatkan ini, implementasi ZBar membuat pemindaian linier melewati gambar, memperlakukan setiap piksel sebagai sampel dari satu sensor cahaya. Data dipindai, diterjemahkan dan dikumpulkan dengan cepat.[6]

Desain umum untuk "Image Scanner" kode batang adalah menerapkan teknik pemrosesan gambar digital ke gambar yang berisi kode batang: detail persisnya bervariasi, tetapi ini biasanya melibatkan beberapa langkah filter untuk membersihkan *noise*, mempertajam dan meningkatkan kontras, deteksi tepi, dan bentuk analisis untuk menentukan lokasi simbol dan orientasi, dll. Akhirnya data diekstraksi dari gambar murni ini. Semua tahapan pemrosesan ini memerlukan siklus CPU dan sumber daya memori dan seringkali sensitif terhadap berbagai konfigurasi "parameter filter" yang sulit dipahami dan disiapkan oleh pengguna akhir. [6].

Gambar 1 adalah langkah – langkah algoritma Pustaka ZBar Bekerja saat melakukan Operasi Pemindaian.

a. Input gambar

Abstraksi perangkat video yang menghasilkan aliran gambar untuk dipindai. Rilis saat ini memiliki antarmuka ke *video4linux* (versi 1 dan 2). Dukungan sedang berlangsung untuk *VfW (Windows)*.

b. Image scanner

Membuat pemindaian melewati gambar dua dimensi untuk menghasilkan aliran linier dari sampel intensitas. Gambar masukan mungkin berasal dari modul masukan video, atau sumber gambar eksternal (seperti keluaran file gambar dengan pemindai alas datar atau kamera digital). Modul ini juga menggabungkan heuristik konsistensi antar-frame opsional yang diterapkan pada aliran video

c. Linier Scanner

Memindai aliran sampel intensitas abstrak untuk menghasilkan aliran "lebar batang". Sampel intensitas dapat berupa nilai piksel dari pemindai gambar internal, nilai piksel dari pemindai gambar eksternal alternatif, atau bahkan sampel sensor mentah dari tongkat "tanpa dekoder" atau sensor laser. Batang-batang tersebut dideteksi dan diukur dengan menerapkan beberapa pemrosesan sinyal 1D yang sangat dasar ke aliran sampel masukan.

d. Decoder

Decoder mencari aliran lebar batang untuk pola yang dapat dikenali dan menghasilkan aliran data simbol yang sepenuhnya diterjemahkan. Rilis saat ini mengimplementasikan *decoding* untuk simbologi *EAN-13, UPC-A, UPC-E, EAN-8, Code 128 dan Code 39*. Dukungan direncanakan pada akhirnya untuk *add-on PDF-417 dan EAN / UPC*.

e. Output data

Abstraksi sederhana dari jendela keluaran tampilan yang dapat menampilkan gambar yang dipindai kepada pengguna dan menerima masukan sebagai tanggapan. Untuk memaksimalkan fleksibilitas, jendela dapat dibuka dan dimiliki oleh perpustakaan, atau dilampirkan ke jendela yang dikelola aplikasi yang disematkan dalam GUI. Rilis saat ini mendukung antarmuka X11 dasar (*XVideo dan XImage*). Dukungan sedang berlangsung untuk *VfW (Windows)*.

2.4 Rancangan dan Alur penelitian

Dalam melakukan penelitian ini, rancangan program percobaan menggunakan bahasa pemrograman python, langkah ini dapat diuraikan yaitu :

- a. Penyiapan library dan ekstensi yang dibutuhkan.

Dalam penelitian, langkah langkah yang akan di lakukan menggunakan text editor *Visual Studio Code*. Dan sebelum itu, Mendownload dan menginstal library dan ekstensi yang di butuhkan dalam pemrograman *python* dalam Visual Studio Code seperti *cv2*, *ZBar* dan pendukung lainnya

- b. Membuat file *api.py* yang di fungsikan untuk local server.

Dalam komposisi file, trdapat beberapa bagian penting dalam menampung gambar yang nantinya akan diproses dalam file pengolah barcode. File *local server* ini bernama *api.py*, dalam file ini terdapat beberapa bagian yaitu *library import*, yang berisi import – import library yang dibutuhkan. Membuat *route* dengan metode *GET*, dan *set up host*, dan *port*

Tabel 1 Tabel Skenario

No	Skenario	Keterangan
1	Skenario 1	Kesamaan ukuran (besar) dan kejelasan gambar barcode.
2	Skenario 2	Kesamaan ukuran (kecil) dan kejelasan gambar barcode
3	Skenario 3	Ketidak samaan ukuran (besar dan kecil) dan kejelasan barcode.
4	Skenario 4	Kesamaan ukuran (besar) dan ke tidak jelasan gambar barcode (jelas dan buram).
5	Skenario 5	Kesamaan ukuran (kecil) dan ke tidak jelasan gambar barcode (jelas dan buram).
6	Skenario 6	Ketidak samaan ukuran (besar dan kecil) dan ke tidak jelasan gambar barcode (jelas dan buram).

- c. Membuat file *barcode.py* yang berfungsi sebagai pengolah gambar

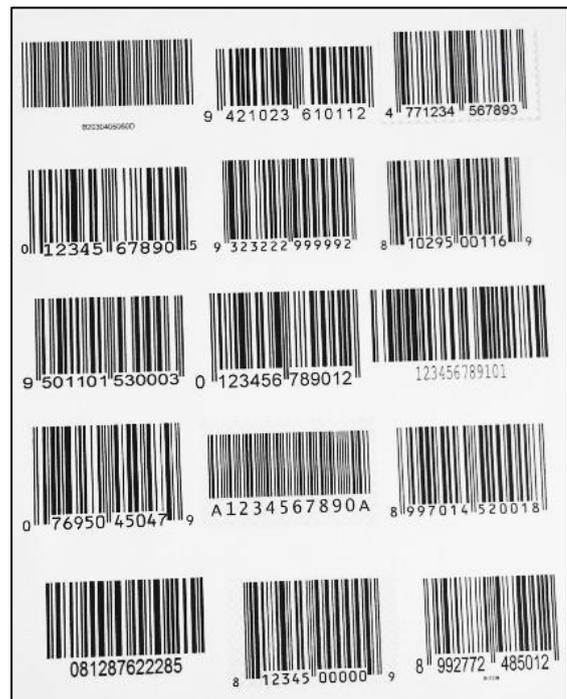
File ini mempunyai bagian – bagian sendiri juga library and package import. Fungsi *ZBar* akan di maksimalkan dalam file ini. File ini menerima gambar yang dikirimkan oleh file *api.py* lalu di teruskan ke bagian pemroses. Pada proses penguraian gambar, program akan membuat rectangle area sesuai jumlah barcode yang terdeteksi lalu menguraikan kode serta jenis barcode nya. Setelah terurai, program akan mengirimkan print out hasil pada file *api.py* dan menampilkannya

3. HASIL DAN PEMBAHASAN

Ketika rancangan telah selesai, alur penelitian ini diuraikan dalam langkah – langkah berikut :

3.1 Penyiapan dataset berupa lembaran barcode

Persiapan ini dengan membuat cetakan beberapa barcode yang akan di uji. Dalam pengujian ini menggunakan 15 barcode yang di cetak berdasarkan beberapa kemungkinan yang akan terjadi dalam langkah pemindaian seperti yang digambarkan dalam table skenario (tabel 1).



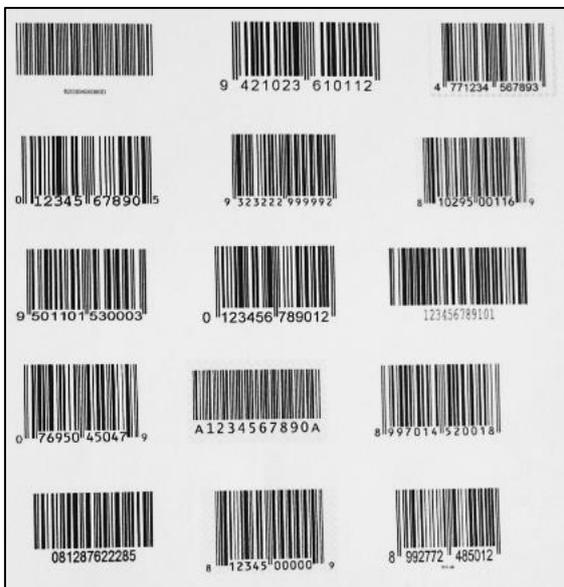
Gambar 1 susunan barcode skenario 1

Tabel 2 Kesamaan ukuran (besar) dan kejelasan gambar barcode.

No	Data Set	Hasil Scan
1	B2030405060D	B2030405060D
2	9421023610112	9421023610112
3	4771234567893	4771234567893
4	012345678905	012345678905
5	9323222999992	9323222999992
6	810295001169	810295001169
7	9501101530003	9501101530003
8	0123456789012	0123456789012
9	123456789101	123456789101
10	076950450479	076950450479
11	A1234567890A	A1234567890A
12	8997014520018	8997014520018
13	081287622285	081287622285
14	812345000009	812345000009
15	8992772485012	8992772485012

3.2 Percobaan dilakukan sesuai beberapa skenario percobaan

Percobaan atau ujicoba pemindaian barcode dilakukan dengan menggunakan 15 dataset yang dibuat variasi scenarionya. Pemindaian ini menggunakan kamera ponsel pintar, dan di ambil dalam jarak 15 sampai dengan 40 cm. percobaan pemindaian di lakukan 3 sampai dengan 5 kali setiap skenarionya, hal ini dilakukan untuk memastikan hasil pemindaian benar – benar konsisten dan tepat. Hasil pemindaian di tulis berdasarkan nilai yang sering keluar langkah pemindaian. Hasil dapat dilihat seperti pada tabel 2, hasil Ujicoba berdsarkan skenario.



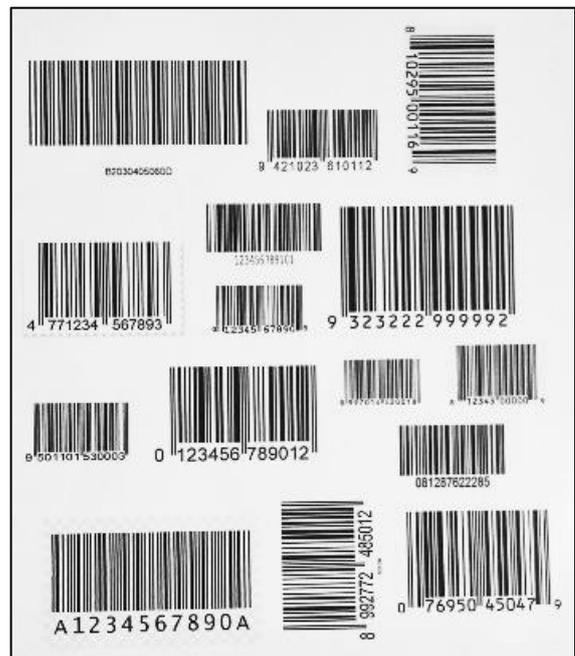
Gambar 2 susunan barcode skenario 2

Tabel 3.3 Kesamaan ukuran (kecil) dan kejelasan gambar barcode.

No	Data Set	Hasil Scan
1	B2030405060D	B2030405060D
2	9421023610112	9421023610112
3	4771234567893	4771234567893
4	012345678905	012345678905
5	012345678905	012345678905
6	9323222999992	9323222999992
7	810295001169	810295001169
8	9501101530003	9501101530003
9	0123456789012	0123456789012
10	123456789101	123456789101
11	076950450479	076950450479
12	A1234567890A	
13	8997014520018	8997014520018
14	081287622285	081287622285
15	812345000009	812345000009
16	8992772485012	8992772485012

Pada tabel 2 hasil yang di dapat sesuai dengan apa yang di diharapkan. Hasil nya benar semua. Hal ini dikarenakan barcode yang di cetak, sangat mendukung seperti ukuran (besar) dan kejelasan gambar barcode.

Pada tabel 3 terdapat 1 barcode yang tidak terdeteksi setelah beberapa percobaan. Hal ini di sebabkan karena ukuran barcode terlihat kurang jelas saat diubah ukuran barcodenya dari besar menjadi kecil.

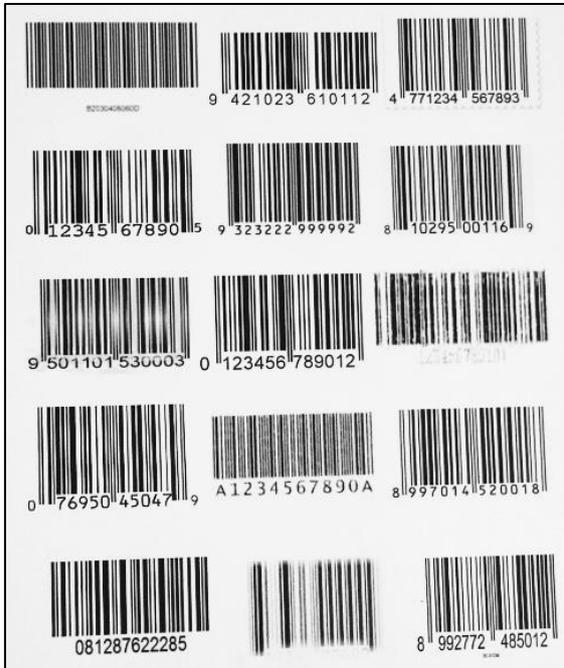


Gambar 3 susunan barcode skenario 3

Tabel 4 Ketidak samaan ukuran (besar dan kecil) dan kejelasan barcode.

No	Data Set	Hasil Scan
1	B2030405060D	B2030405060D
2	9421023610112	9421023610112
3	4771234567893	4771234567893
4	012345678905	012345678905
5	012345678905	012345678905
6	9323222999992	9323222999992
7	810295001169	810295001169
8	9501101530003	9501101530003
9	0123456789012	0123456789012
10	123456789101	123456789101
11	076950450479	076950450479
12	A1234567890A	A1234567890A
13	8997014520018	8997014520018
14	081287622285	081287622285
15	812345000009	812345000009
16	8992772485012	8992772485012

Pada tabel 4 menunjukkan pemindaian barcode bekerja dengan baik meskipun ukuran barcode berbeda – beda. Hal ini di sebabkan karena beberapa barcode yang dirubah ukurannya dari besar ke kecil tidak mempengaruhi kejelasan setiap gambar barcode



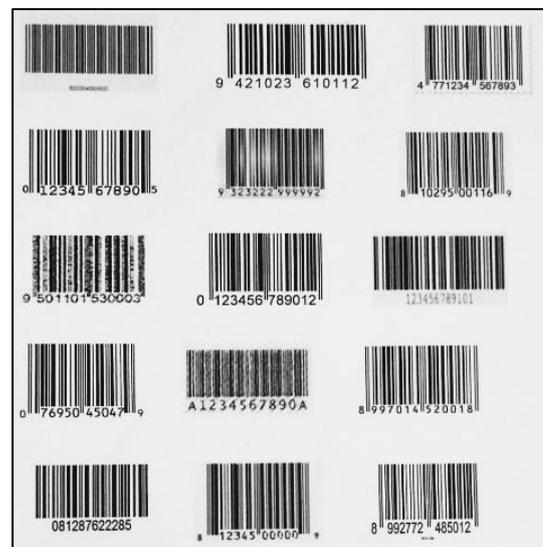
Gambar 4 susunan skenario 4

Tabel 5 Kesamaan ukuran (besar) dan ketidakjelasan gambar barcode (jelas dan buram).

No	Data Set	Hasil Scan
1	B2030405060D	
2	9421023610112	9421023610112
3	4771234567893	4771234567893
4	012345678905	012345678905
5	9323222999992	9323222999992
6	810295001169	810295001169
7	9501101530003	9501101530003
8	0123456789012	0123456789012
9	123456789101	123456789101
10	076950450479	076950450479
11	A1234567890A	A1234567890A
12	8997014520018	8997014520018
13	081287622285	081287622285
14	812345000009	812345000009
15	8992772485012	8992772485012

Pada tabel 5 ini tetap dapat memindai banyak barcode dengan baik meskipun terdapat gambar barcode yang diburamkan. Namun tetap terdapat 1 gambar barcode yang tidak dapat dipindai setelah beberapa percobaan. Hal ini disebabkan karena gambar barcode berubah strukturnya saat

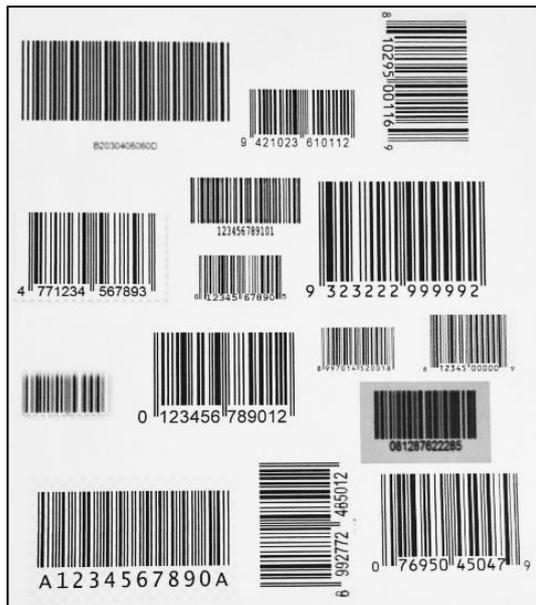
di buramkan, sehingga pemindai tidak dapat membacanya sama sekali.



Gambar 3.2.5 susunan barcode skenario 5

Tabel 3.6 Kesamaan ukuran (kecil) dan ke tidak jelasan gambar barcode (jelas dan buram).

No	Data Set	Hasil Scan
1	B2030405060D	
2	9421023610112	9421023610112
3	4771234567893	4771234567893
4	012345678905	012345678905
5	9323222999992	9323222999992
6	810295001169	810295001169
7	9501101530003	
8	0123456789012	0123456789012
9	123456789101	123456789101
10	076950450479	076950450479
11	A1234567890A	
12	8997014520018	8997014520018
13	081287622285	081287622285
14	812345000009	
15	8992772485012	8992772485012



Gambar 3.2.6 susunan barcode skenario 6

Hasil pada tabel 6 ini terdapat beberapa barcode yang tidak terpindai dengan baik. Karena ukuran yang kecil dan beberapa barcode yang buram terlihat mendominasi

Tabel 7 Ketidak samaan ukuran (besar dan kecil) dan ke tidak jelasan gambar barcode (jelas dan buram).

No	Data Set	Hasil Scan
1	B2030405060D	B2030405060D
2	9421023610112	9421023610112
3	4771234567893	4771234567893
4	012345678905	012345678905
5	9323222999992	9323222999992
6	810295001169	810295001169
7	9501101530003	
8	0123456789012	0123456789012
9	123456789101	123456789101
10	076950450479	076950450479
11	A1234567890A	A1234567890A
12	8997014520018	
13	081287622285	
14	812345000009	
15	8992772485012	8992772485012

Pada table 7 terdapat 4 barcode yang tidak terpindai dengan baik. Hal ini di sebabkan karena acaknya ukuran, posisi, dan keburaman barcode.

3.3 Evaluasi dan Penilaian hasil

Tabel 8 Rangkuman Hasil Ujicoba

No	Skenario	Benar	Salah	Erorr	Akurasi
1	Skenario 1	15	0	0	100

2	Skenario 2	14	1	6,6	93,3
3	Skenario 3	15	0	0	100
4	Skenario 4	14	1	6,6	93,3
5	Skenario 5	11	4	26,6	73,3
6	Skenario 6	11	4	26,6	73,3
Rata-Rata Erorr					11.06
Rata-Rata Akurasi					88,8

Nilai error di dapat dari perhitungan :

$$(\text{salah} / \text{jumlah data}) \times 100.$$

Nilai akurasi didapar dari hasil perhitungan :

$$(\text{benar} / \text{jumlah data}) \times 100$$

4. SIMPULAN

Berdasarkan beberapa hasil yang diperoleh dalam penelitian ini, maka dapat di ambil kesimpulan yaitu :

1. Library Zbar sangat praktis digunakan dalam implementasi python untuk program barcode scanner. Kemampuan dalam membaca banyak barcode sekaligus membuatnya sangat rekomendasi digunakan untuk mengembangkan sebuah sistem multi scanner
2. Dalam kondisi buram pun, library ini mampu memindai barcode dengan percobaan 3 sampai dengan 5 kali
3. 3. Skenario yang dibuat didasarkan dalam kondisi terbaik dan kondisi paling buruk dengan menggunakan 15 barcode yang sama dalam setiap skenarionya. Dan dalam percobaan 6 skenario ini menghasilkan nilai akurasi sebesar 88.8% dengan nilai error rata – rata 11%.
4. Jika skenario penelitian dilibatkan dalam kondisi terbaik saja, yang di maksud adalah skenario hanya melibatkan gambar – gambar yang jelas, dan data – data barcode yang beragam maka hasil akurasi bisa mencapai 99%.

5. SARAN

Agar dalam penelitian yang akan datang dapat lebih baik, maka beberapa saran yang dapat di lakukam yaitu :

1. Skenario dapat di tambah lebih banyak kondisi. Agar mengetahui seberapa akurat program dalam memindai barcode.

2. Penelitian yang akan datang dapat di kembangkan dalam pemindaian *QRCode* yang mana *ZBar* ini sudah mendukung untuk pemindaian *QRCode*
3. Menambah jumlah barcode dalam skenario. Hal ini di lakukan untuk mengetahui seberapa banyak lagi barcode yang dapat di pindai oleh *ZBar*.
4. Membedakan data barcode dalam setiap skenario. Langkah ini dilakukan untuk mengetahui seberapa banyak *ZBar* dalam memindai setiap barcode yang berbeda kondisi.

UNTUK IDENTIFIKASI PRODUK KEMASAN BERDASARKAN LABEL KEMASANNYA. 2.

- [6] Brown, J. (2010). *ZBar Barcode Reader*. Retrieved from *ZBar*: <http://zbar.sourceforge.net>

DAFTAR PUSTAKA

- [1] Alyusi, Shiefti Dyah. 2016. *Media Sosial: Interaksi, Identitas dan Modal Sosial*.
- [2] Astuti, Ana Puji, Anike Nurmalita RPS. *Teknologi komunikasi dan Perilaku Remaja*. *Jurnal Analisa Sosiologi*. April 2014, 3(1): 91-111
- [3] Ong, B.S., *Determinants of purchase intentions and stock-piling tendency of bonus packs*. *American Business Review*, 1999. 17(1): p. 57
- [4] Blattberg, R.C. and S.A. Neslin, *Sales promotion: Concepts, methods, and strategies*. 1990: Prentice Hall Englewood Cliffs, NJ
- [5] Nana Ramadijanti, S. M. (2010). *IMPLEMENTASI PENGOLAHAN CITRA*