P-ISSN : 2622-1594
E-ISSN : 2685-449X

Information Extraction Pada Pesanan Pembelian Menggunakan RetinaNet dan Tesseract untuk Toko Maju

Eillen Marchellita Hartono[a], Yosua Setyawan Soekamto[b]
[a,b]*Departemen Sistem Informasi, Universitas Ciputra Surabaya*
E-mail: emarchellita@student.ciputra.ac.id, yosua.soekamto@ciputra.ac.id

***Abstrak*—Proses jual beli berubah mengikuti perkembangan zaman. Kini, proses transaksi dapat dilakukan dengan sistem pemesanan yang diikuti dengan dokumen pesanan pembelian. Melalui pesanan pembelian, pengusaha dapat memperoleh banyak informasi untuk analisis bisnis. Akan tetapi, banyak dari pengusaha retail masih belum menyimpan data tersebut secara terstruktur, sehingga sulit untuk melakukan analisis dan pelacakan. Meskipun banyak dari pesanan pembelian telah berbentuk digital seperti PDF, pencatatan terstruktur bersumber dari dokumen digital masih memerlukan upaya dalam waktu dan usaha, serta rawan kesalahan jika dilakukan secara manual oleh manusia. Penelitian ini bertujuan untuk membuat model *information extraction* dari pesanan pembelian berbentuk PDF. Alur kerja dari penelitian ini dimulai dengan pengumpulan data, data *pre-processing*, *information extraction*, evaluasi, dan penyimpanan kedalam *database*. Data yang digunakan pada penelitian ini adalah pesanan pembelian dari "Toko Maju" yang berbentuk PDF. Pesanan pembelian akan dirubah ke format JPEG, sebelum dilakukan proses pelabelan dan pembentukan *bounding boxes*. Proses *information extraction* meliputi proses *object detection* dan OCR. *Object detection* akan menggunakan model Keras RetinaNet. Setelah letak daerah ekstraksi ditemukan, maka akan dilakukan deteksi karakter atau OCR dengan menggunakan *library* Tesseract. Informasi hasil ekstraksi akan disimpan ke database MySQL. Model *information extraction* memperoleh nilai confidence sebesar 95.6% dan nilai accuracy sebesar 95.5%.**

***Kata Kunci*— *Information Extraction*, *Object Detection*, RetinaNet, Tesseract**

I. PENDAHULUAN

Proses jual beli telah mengalami perubahan mengikuti perkembangan zaman. Tidak hanya dengan transaksi di toko fisik secara langsung, namun kini, proses transaksi dapat dilakukan melalui sistem pemesanan. Transaksi berupa pemesanan ini biasa diikuti dengan dokumen yang menguraikan data barang pesanan, berupa nama barang, jumlah pesanan, harga, tanggal pemesanan, dan berbagai informasi lainnya terkait barang yang dipesan. Dokumen ini disebut sebagai pesanan pembelian atau umumnya dikenal sebagai *purchase order* (PO) [1]. Pesanan pembelian dibuat oleh pembeli kepada pengusaha sebagai kontrak perjanjian ketika ingin melakukan pemesanan.

Melalui pesanan pembelian yang masuk, pengusaha dapat menemukan banyak informasi yang bisa digunakan untuk analisis bisnis. Selain itu, pengusaha juga dapat menghindari kesalahan pada jumlah pemesanan barang, sarana pengingat, dan bukti pemesanan. Namun, banyak dari pengusaha retail masih belum menyimpan data tersebut secara terstruktur [2], sehingga sulit untuk melakukan analisis dan pelacakan.

Salah satu tipe dokumen pesanan pembelian yang kini populer akibat perkembangan teknologi adalah *Portable Document Format* (PDF) [3]. Secara garis besar, PDF memiliki 2 tipe konten yaitu konten teks hasil pindai (*scan*) dan konten teks hasil cetak digital. Meskipun telah berbentuk digital, pencatatan terstruktur bersumber dari dokumen berbentuk PDF memerlukan upaya lebih dalam waktu dan usaha, serta rawan kesalahan jika dilakukan secara manual oleh manusia. Permasalahan yang akan diangkat pada penelitian ini berdasar dari "Toko Maju", dimana pesanan pembelian yang diterima oleh toko tersebut berbentuk gambar yang disimpan dalam format PDF. Hal ini menyebabkan pemilik toko perlu mencatat informasi seperti harga barang ataupun tanggal jatuh tempo secara manual melalui *excel*. Pemilik juga perlu menghitung total pendapatan secara satu-persatu dari setiap pesanan pembelian yang diterima. Setiap bulan, "Toko Maju" mendapatkan rata-rata pesanan sebanyak 300 hingga 350 pesanan. Secara omset, Toko Maju juga dapat dikategorikan ke dalam Usaha Mikro Kecil Menengah (UMKM).

Berangkat dari permasalahan tersebut, penelitian ini bertujuan untuk membuat model *information extraction* dari pesanan pembelian berbentuk PDF. *Information extraction* merupakan sebuah metode pengambilan informasi dari dokumen. Informasi yang diambil beragam disesuaikan dengan kebutuhan atau tujuan model yang dibangun [4]. Pada penelitian ini model *information extraction* yang dibuat menggunakan ekstraksi karakter berupa gambar (*image*) menjadi teks (*text*), atau dikenal juga dengan sebutan *Optical Character Recognition* (OCR). Hasil ekstraksi dari model *information extraction* kemudian akan disimpan kedalam *database* dan diharapkan dapat membantu pemilik "Toko Maju" dalam membuat pencatatan terstruktur.

Naskah Masuk : 05 September 2023
Naskah Direvisi : 17 April 2024
Naskah Diterima : 24 November 2024

DOI : 10.37823/insight.v6i2.315

II. STUDI TERDAHULU

Penelitian perihal *information extraction* melalui dokumen PDF pernah dilakukan sebelumnya oleh peneliti terdahulu. Salah satunya adalah penelitian yang dilakukan oleh Soekamto pada tahun 2020 mengenai ekstraksi judul dan abstrak artikel ilmiah dengan pendekatan model *information extraction* berbasis *rule* [4]. Berdasarkan penelitian tersebut, didapatkan bahwa information extraction yang berupa *text* PDF dapat dilakukan dengan pendekatan *rule* secara baik. Rule diterapkan pada fitur-fitur artikel ilmiah sehingga dapat diperoleh bagian-bagian yang sesuai dengan tujuan penelitian, seperti bagian judul dan abstrak artikel ilmiah.

Penelitian lainnya adalah penelitian yang dilakukan oleh Diego Leon pada tahun 2021 mengenai *information extraction invoice* dengan *deep learning* [3]. Berdasarkan penelitian tersebut, didapatkan kesimpulan jika *information extraction* dari *invoice* berbentuk PDF dengan pendekatan *deep learning object detection* lebih cocok daripada pendekatan *full rule based*, dengan catatan perlu menyediakan lebih banyak data. Tahapan *information extraction* yang disarankan adalah melakukan deteksi objek, kemudian melakukan deteksi karakter atau OCR, dan terakhir adalah *information extraction* menggunakan *rule-based* atau *regular expression.*

Pendekatan tersebut serupa dengan yang dicetuskan oleh Tan pada penelitiannya dalam melakukan *information extraction* terhadap *invoice*. Dengan maraknya dokumen berbentuk digital, maka metode OCR sangat membantu dalam melakukan *information extraction* berupa teks [5]. Namun, melakukan OCR terhadap dokumen mentah akan memakan waktu, tenaga, dan rawan terhadap kesalahan. Oleh karena itu, memetakan daerah ekstraksi lebih disarankan untuk efisiensi.

Salah satu model *deep learning* untuk *object detection* adalah RetinaNet. Model arsitektur Keras RetinaNet dengan *backbone* ResNet50 telah digunakan oleh Farady pada tahun 2021 untuk mendeteksi gambar dengan beberapa kelas, yaitu *good*, *none*, dan *bad*. Penelitian tersebut menghasilkan rata-rata nilai *confidence* atau nilai keyakinan dari model dalam memprediksi sebuah label sebesar 81.31% [6]. Penggunaan arsitektur serupa juga pernah dilakukan untuk mendeteksi karakter bahasa Mandarin dari sebuah gambar dokumen [7]. Metode evaluasi yang digunakan adalah nilai mAP (*Mean Average Precision*) yang mengacu pada nilai akurasi antara hasil prediksi dengan *ground truth*. Penelitian tersebut menghasilkan nilai mAP sebesar 85%.

Penelitian lainnya perihal *information extraction* ialah oleh Stevan pada tahun 2020 [8], dimana dilakukan identifikasi *wine* berdasarkan label nomor seri pada botolnya. Pada penelitian tersebut, dilakukan proses OCR menggunakan *library* Tesseract dengan membandingkan 2 cara, yaitu gambar nomor seri yang telah dilakukan *pre-processing* dan tidak. Gambar yang tidak dilakukan *pre-processing* memiliki tingkat keberhasilan deteksi karakter sebesar 62%, sedangkan gambar yang dilakukan *pre-processing* memiliki tingkat keberhasilan sebesar 87.5%.

Penggunaan *regular expression* untuk *information extraction* telah dibuktikan oleh Sulaiman, dimana dilakukan pencarian fitur-fitur tertentu pada struktur linguistik bahasa Malaysia [9]. Dikatakan bahwa *regular expression* dapat secara efisien menemukan fitur yang dicari secara presisi. Namun, tetap diperlukan bantuan manusia untuk memvalidasi kebenaran dari data yang diproses.

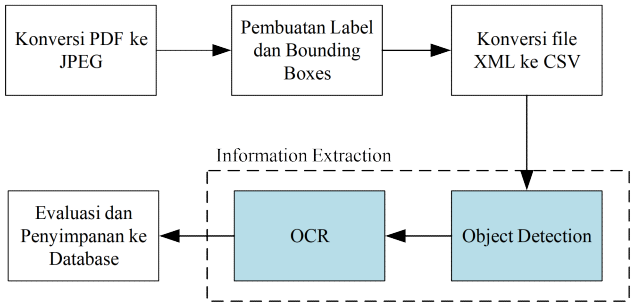
III. METODOLOGI

Pada penelitian ini akan dibangun model information extraction menggunakan pendekatan *deep learning* OCR dan dikombinasikan dengan *regular expression* untuk membersihkan dan menangkap teks. *Deep learning* OCR akan menggunakan model Keras RetinaNet dan Tesseract. Model ini dipilih karena memiliki banyak penelitian terkait yang mendukung penggunaan *deep learning* OCR ibandingkan *full rule based*.

Alur kerja penelitian dimulai dengan pengumpulan data, *data pre-processing*, *information extraction*, evaluasi, dan penyimpanan kedalam *database*. Data yang dipakai dalam penelitian ini merupakan data primer berupa dokumen pesanan pembelian berbentuk PDF yang diperoleh dari "Toko Maju". Jumlah data yang digunakan untuk proses *training* sejumlah 520 dokumen. Proses *data pre-processing* terdiri dari proses konversi dokumen PDF ke format JPEG, pembuatan label dan *bounding boxes*, serta konversi *file* XML ke CSV.

Information extraction dilakukan melalui dua proses, yaitu *object detection* untuk mendapatkan letak tiap label dan OCR untuk mendeteksi karakter dari setiap potongan label. Setelah seluruh karakter dideteksi, tahapan selanjutnya adalah melakukan evaluasi terhadap model yang telah dibangun. Tahapan terakhir adalah melakukan penyimpanan ke *database*. Alur kerja dapat dilihat pada Gambar 1.

Pada penelitian ini, hipotesis yang diajukan adalah bahwa model dapat mendeteksi letak *bounding boxes* dari setiap label dengan nilai *confidence* lebih dari atau sama dengan 80%.



Gambar 1. Alur Kerja Penelitian

A. Konversi PDF ke JPEG

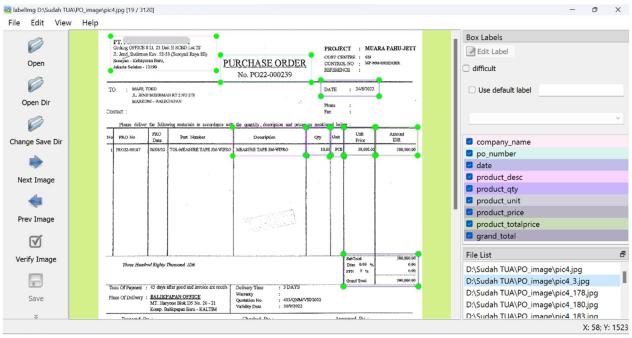
Seluruh dokumen pesanan pembelian yang diperoleh disimpan dalam format PDF. Sebelum dapat digunakan pada tahapan selanjutnya, dokumen tersebut perlu di konversi ke format JPEG. Proses konversi dilakukan dengan melakukan iterasi terhadap seluruh dokumen dalam satu folder dengan menggunakan *library* pdf2image. Hasil akhir dari proses konversi data adalah sebuah folder yang berisi 520 gambar pesanan pembelian.

B. Pembuatan Label dan Bounding Boxes

Untuk melakukan proses *training model object detection*,

 DOI : 10.37823/insight.v6i2.315

maka diperlukan *dataset* yang memuat letak dari daerah yang ingin diekstrak. Oleh karena itu, dengan menggunakan LabelImg, akan dibentuk 9 *bounding boxes* pada gambar pesanan pembelian, meliputi nama perusahaan, nomor pesanan pembelian, tanggal, nama barang, kuantitas, unit, harga satuan, subtotal harga, dan total harga keseluruhan. Setelah dilakukan pelabelan, akan dihasilkan sebuah *file* XML dengan nama serupa nama gambar, yang berisi koordinat dari setiap *bounding boxes* [10].



Guna menambah variasi pada dataset, dilakukan proses rotasi gambar dengan kemiringan sebesar 3, 178, 180, 183, dan 357 derajat. Data yang digunakan adalah 520 gambar yang telah dilabeli sebelumnya. Proses ini juga mencakup pembuatan label dan *bounding boxes* dari gambar yang dirotasi.

Hasil rotasi kemudian di *export* menjadi gambar dan *file* XML baru dengan total keseluruhan sejumlah 2600 data. Pada pembentukan koordinat untuk *bounding boxes*, diberikan batasan koordinat x dan y sesuai ukuran gambar yaitu 4134 × 5846. Hal ini dikarenakan ketika dirotasi, ada beberapa label memiliki koordinat minus ataupun melebihi ukuran gambar.

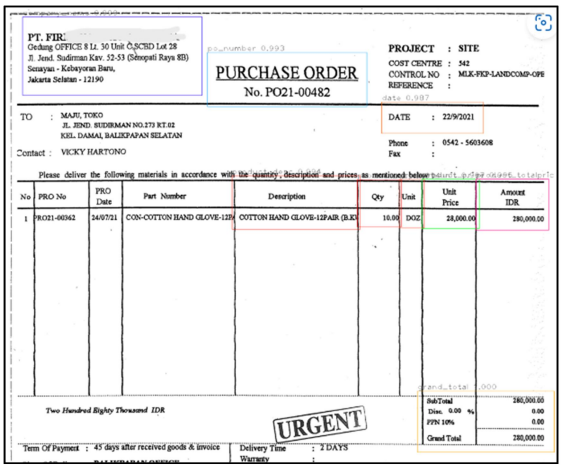
C. *Konversi File XML ke CSV*

Dataset yang diperlukan untuk proses *training* oleh RetinaNet adalah gambar dan koordinat letak dalam format CSV. Oleh karena itu, dilakukan proses konversi untuk merangkum seluruh letak koordinat dari *file* XML ke format CSV [10]. Dari setiap *file* XML akan diambil koordinat, berupa xmin, ymin, xmax, dan ymax dari setiap label. Hasil dari proses konversi *file* XML adalah sebuah *file* CSV sejumlah 28080 baris dan 6 kolom, berupa nama gambar, koordinat (xmin, ymin, xmax, ymax), dan kelas. Setiap kelas pada satu gambar didefinisikan sebagai satu baris baru.

D. *Object Detection*

Proses *training* diawali dengan *clone* github keras-retinanet dan melakukan instalasi untuk *dependencies* yang dibutuhkan, seperti tensorflow, numpy, opencv, dan lain-lain. Kemudian, untuk mempersingkat waktu training, dilakukan *transfer learning* dengan menggunakan *pre-trained* model ResNet50.

Dalam melakukan prediksi, dibuat beberapa fungsi untuk menggambar *bounding boxes* dan menampilkan koordinat label [6], [7], [10].



Gambar 3. Hasil Prediksi *Bounding Boxes*

Demi memastikan Tesseract mendeteksi karakter dengan baik, maka gambar dengan karakter yang terbalik perlu dirotasi. Proses rotasi berpatokan pada koordinat y minimum pada label nama perusahaan dengan grand total. Apabila koordinat y minimum pada label nama perusahaan lebih besar, maka gambar tersebut diperkirakan sebagai gambar yang terbalik dan akan dirotasi. Proses rotasi dilakukan dengan menggunakan *library* imgaug.

E. *Optical Character Recognition*

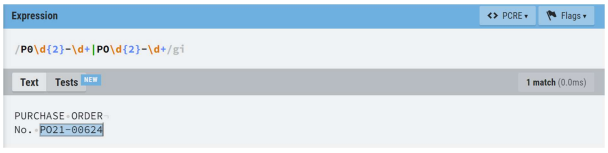
Proses deteksi karakter dilakukan dengan menggunakan *library* Tesseract dan dibantu dengan beberapa *library* lain seperti PIL, pandas, regex, dan os.

Setelah mengimpor seluruh *library*, tahapan selanjutnya adalah membaca gambar dan meng-*crop* gambar sejumlah 9 label, sesuai dengan ukuran *bounding boxes*. Setelah itu, karakter akan dideteksi dengan menggunakan perintah image_to_string. Proses ini akan dilakukan berulang pada setiap gambar. Gambar yang akan digunakan untuk prediksi adalah 60 gambar, dengan masing-masing 10 gambar untuk tiap derajat kemiringan.

```
| “SubTotal nsosesesesseeee""95'8°0100.00
Disc. 9.00 % 0.00
PPN 10% 0.00
Grand Total 225,000.00
```

Gambar 4. Hasil Deteksi Karakter

Namun, hasil deteksi karakter dari Tesseract masih belum bersih dan banyak mendeteksi hal diluar *value* yang diharapkan. Oleh karena itu, dilakukan pembersihan karakter dengan menggunakan *regular expression*.



Gambar 5. *Post-Processing* dengan Regex

F. *Evaluasi dan Penyimpanan ke Database*

Evaluasi dilakukan dengan 2 cara. Evaluasi nilai *confidence* dilakukan dengan menghitung nilai minimum, maksimum, dan rata-rata hasil prediksi *bounding boxes* dari model *object detection* yang telah dibuat. Nilai akurasi akan dihitung dari jumlah *value* terdeteksi pada tiap kolom dibagi

DOI : 10.37823/insight.v6i2.315

Iterasi	1		2		3		4		5		Rata-rata
Nama Kolom	Accuracy	Jumlah "None"	Accuracy	Jumlah "None"	Accuracy	Jumlah "None"	Accuracy	Jumlah "None"	Accuracy	Jumlah "None"	
Nama perusahaan	98.33	1	95	3	95	3	100	0	100	0	97.666
Tanggal	100	0	98.33	1	100	0	100	0	100	0	99.666
Nomor PO	98.33	1	96.67	2	98.33	1	98.33	1	98.33	1	97.998
Total Harga	100	0	100	0	100	0	100	0	100	0	100
Nama barang	85.91	21	85.91	21	83.24	30	88.69	19	90.97	14	86.944
Kuantitas	93.96	9	94.63	8	89.94	18	92.26	13	91.61	13	92.48
Unit	96.64	5	95.97	6	97.21	5	96.43	6	96.13	6	96.476
Harga	95.97	6	96.64	5	91.06	16	94.64	9	92.9	11	94.242
Subtotal	94.63	8	97.32	4	91.06	16	94.64	9	94.19	9	94.368

Gambar 6. Hasil Evaluasi Nilai Accuracy Pemilihan Data Random Sampling

dengan jumlah baris keseluruhan.

Penyimpanan ke *database* dibantu dengan *library* pymysql. Terdapat dua kali proses *insert*, yaitu *insert* ke tabel pemesanan dan *insert* ke tabel detail_pemesanan.

IV. PENGUJIAN

Pengujian dilakukan pada data baru yang sebelumnya belum pernah dilihat oleh model. Pengujian dilakukan dengan dua metode. Pada setiap metode, dilakukan pengujian terhadap 60 data yang terdiri dari 10 data untuk setiap derajat kemiringan (0, 3, 177, 180, 183, dan 357 derajat).

TABEL I
HASIL EVALUASI NILAI CONFIDENCE

Iterasi	Nilai Minimum	Nilai Maksimum	Nilai Rata-Rata
1	0.975	0.999	0.994
2	0.906	0.999	0.993
3	0.979	0.999	0.995
4	0.972	0.999	0.995
5	0.949	0.999	0.994

Metode pertama adalah dengan evaluasi nilai *confidence* dari proses *object detection*. Pada pengujian ini, dataset prediksi akan dikelompokkan berdasarkan nama kelompok dan derajat kemiringan. Kemudian, dihitung rata-rata nilai *confidence* dari setiap kelompok tersebut. Selanjutnya, dilakukan iterasi sebanyak 5 kali dengan *random sampling* untuk dihitung nilai minimum, maksimum, dan rata-ratanya. Hasil evaluasi nilai *confidence* pada model *object detection* dapat dilihat pada Tabel I.

Metode kedua adalah dengan evaluasi nilai *accuracy*. Nilai *accuracy* dihitung berdasarkan jumlah kata "None" dari tiap label dibagi dengan jumlah baris data tiap label. Kata "None" merupakan *value* yang dikembalikan apabila fungsi tidak berhasil menemukan karakter label yang diharapkan pada proses OCR. Proses evaluasi nilai *accuracy* dilakukan dengan 2 cara, yang pertama adalah dengan memilih 60 gambar secara spesifik yang seluruhnya berbeda dan yang kedua adalah dengan melakukan *random sampling* sebanyak 60 dari 600 gambar yang diulang sebanyak 5 iterasi. Proses ini dilakukan dengan dua cara untuk menghindari bias. Hasil evaluasi nilai *accuracy* dengan pemilihan secara spesifik dapat dilihat pada Tabel II.

TABEL II
HASIL EVALUASI NILAI ACCURACY PEMILIHAN DATA SPESIFIK

Nama Kolom	Accuracy	Jumlah "None"
Nama Perusahaan	98.33	1
Tanggal	98.33	1
Nomor PO	100	0
Total Harga	100	0
Nama Barang	96.90	4
Kuantitas	97.67	3
Unit	98.45	2
Harga	99.22	1
Subtotal	98.45	2

Hasil evaluasi nilai *accuracy* pada data prediksi yang dipilih secara *random sampling* ada pada Gambar 6.

V. PEMBAHASAN

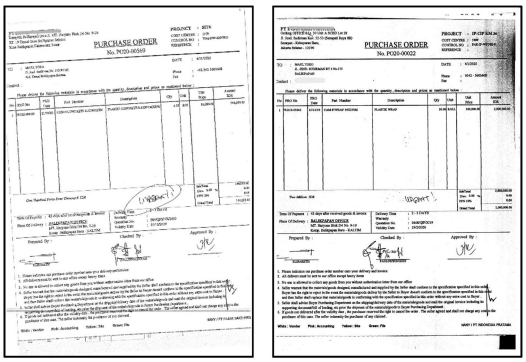
Proses pembangunan model *information extraction* diawali dengan pengumpulan dokumen pesanan pembelian. Dokumen pesanan pembelian yang diharapkan adalah dokumen dengan gambar yang jernih, tidak miring, dan tulisan terlihat jelas. Akan tetapi, tidak seluruh dokumen memiliki atribut tersebut.

Meskipun demikian, proses pelabelan tetap dilakukan baik terhadap dokumen yang sesuai harapan maupun tidak, untuk melatih model terhadap berbagai jenis dokumen. Pada proses pelabelan, ukuran dari *bounding boxes* dilebihkan sedikit dari ukuran gambar yang ingin diambil. Hal ini untuk memastikan seluruh bagian gambar masuk dalam *bounding boxes* dan tidak ada yang terpotong.

Rata-rata nilai minimum *confidence* dari model yang diperoleh dari penelitian ini adalah sebesar 95.6%. Nilai ini telah melebihi ekspektasi hipotesis yang ditentukan sebelumnya yaitu 80%. Pada Tabel I ditampilkan nilai *confidence* dari model *object detection*. Dari kelima iterasi, diperoleh nilai *confidence* minimum berkisar pada 90.6% - 97.9%. Nilai *confidence* maksimum berada pada angka 99.9%. Hal ini menunjukkan bahwa model telah dapat dengan baik melakukan prediksi letak *bounding boxes* dan label.

Pada awal percobaan dengan 300 dataset, nilai *confidence* yang dihasilkan tidak sebaik sekarang dengan dataset sejumlah 520×6. Hal ini sesuai dengan penelitian oleh Georgy yang menyebutkan bahwa semakin banyak dataset, akan meningkatkan keakuratan model dalam mendeteksi objek [11].

 DOI : 10.37823/insight.v6i2.315

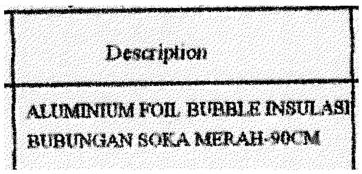


Gambar 7. Dokumen yang Tidak Sesuai Harapan

Pada proses OCR, hasil teks yang dideteksi akan dicek oleh fungsi. Fungsi akan melakukan pengecekan dengan menggunakan *pattern* regex untuk mengambil teks tertentu. Apabila tidak ditemukan teks yang sesuai dengan *pattern*, maka akan dikembalikan kata "None".

Tabel II berisi nilai *accuracy* dari data yang dipilih secara spesifik. Dari data tersebut, diperoleh rata-rata nilai *accuracy* sebesar 98.6% dengan jumlah teks tidak terdeteksi adalah 14 baris data pada 9 label. Pada Gambar 6 diperoleh nilai *accuracy* dari data *random sampling* sebanyak 5 kali iterasi. Nilai rata-rata *accuracy* yang diperoleh adalah 95.5% dengan jumlah rata-rata teks tidak terdeteksi adalah 60 baris data pada 9 label. Hasil *accuracy* dari data spesifik lebih bagus karena dipilih dokumen yang sesuai harapan dengan memastikan seluruh dokumen tidak ada yang kembar. Sedangkan pada cara kedua, data dipilih secara acak oleh sistem tanpa memperhatikan jenis dokumen.

Penelitian terdahulu menyebutkan bahwa Tesseract dapat bekerja lebih baik dengan data dalam *grayscale* daripada berwarna [12]. Meskipun begitu, Tesseract sangat sensitif dengan gambar yang memiliki *noise salt and pepper* [13]. Hal ini menyebabkan ada beberapa karakter yang tidak tertangkap secara sempurna dan bahkan tidak tertangkap sama sekali, seperti contoh pada Gambar 8.



Gambar 8. Gambar dengan Noise Salt & Pepper

Penyebab lain munculnya *value* "None" adalah hasil deteksi karakter oleh Tesseract yang tidak sesuai. Pada penelitian oleh Shannon, ditemukan informasi bahwa Tesseract kurang baik dalam mendeteksi kalimat dengan huruf kapital [14], yang memungkinkan Tesseract menangkap huruf "S" sebagai "g". Kasus tersebut mengakibatkan susahnya menentukan *pattern* untuk regex, sehingga terdapat beberapa kata yang tidak berhasil ditemukan fungsi.

Gambar 9. Contoh Potongan Label

Diketahui pula bahwa terdapat beberapa kata yang tidak berhasil dideteksi oleh Tesseract, seperti kata *Description* dari Gambar 9. Hal ini kemungkinan terjadi karena Tesseract kurang akurat dalam mendeteksi *font* dengan ukuran yang kecil [15], [16].

```
cessed. Paes
KUNCI L 17MM
KUNCI RING PAS
KUNCI SOCKET gET
```

Gambar 10. Hasil Deteksi Karakter

VI. KESIMPULAN

Model *information extraction* pesanan pembelian berbentuk PDF dengan menggunakan RetinaNet dan Tesseract yang memperoleh hasil yang memuaskan dengan nilai *confidence* sebesar 95.6% yang melebihi hipotesis sebesar 80%, dan nilai *accuracy* sebesar 95.5%.

Pada proses pelabelan untuk *object detection*, ukuran dari *bounding boxes* perlu dilebihkan sedikit dari ukuran gambar yang ingin diambil, dengan tujuan memastikan seluruh bagian gambar tidak ada yang terpotong. Pada proses OCR, Tesseract memiliki keterbatasan dalam mendeteksi teks dengan dokumen pada kualitas tertentu.

Saran yang diberikan untuk penelitian serupa kedepannya adalah untuk menyiapkan dataset sedikitnya 3.000 gambar, melakukan *image pre-processing* sebelum melakukan OCR, membuat model tersendiri untuk proses OCR, dan menggunakan dataset dengan *typography* yang umum.

DAFTAR PUSTAKA

[1] Munifah, "Pengertian Dan Fungsi Purchase Order (PO)," Jun. 20, 2022. http://komputerisasi-akuntansi-d3.stekom.ac.id/informasi/baca/Pengertian-Dan-Fungsi-Purchase-Order-PO/089e94bf8ffef5d8b5d0293f3c184677c556a7dd#:~:text=Purchase%20order%20(PO)%20adalah%20dokumen,ingin%20dibeli%20oleh%20pihak%20pembeli. (accessed Feb. 27, 2023).

[2] Risal and E. Kristiawati, "ANALISIS FAKTOR-FAKTOR YANG MEMPENGARUHI PENERAPAN PENCATATAN LAPORAN KEUANGAN PADA UMKM DI KOTA PONTIANAK," Equilibrium Jurnal Ekonomi-Manajemen-Akuntansi, Vol 16, No 2, 2020.

[3] D. Leon, "Extracting Information From PDF Invoices Using Deep Learning", Dissertation, 2021.

[4] Soekamto, Y. S. (2020). Ekstraksi Judul dan Abstrak Artikel Ilmiah Berbasis Rule. Journal of Information System,Graphics, Hospitality and Technology, 2(01), 9–13. https://doi.org/10.37823/insight.v2i01.69

[5] Q. M. Tan, Q. Cao, C. K. Seow, and P. C. Yau, "Information Extraction System for Cargo Invoices," *Res Sq*, 2023, doi: 10.21203.

[6] I. Farady, C. Y. Lin, A. Rojanasarit, K. Prompol, and F. Akhyar, "Mask Classification and Head Temperature Detection Combined with Deep Learning Networks," *2020 2nd International Conference on Broadband Communications, Wireless Sensors and Powering, BCWSP 2020*, pp. 74–78, 2020, doi: 10.1109/BCWSP50066.2020.9249454.

[7] G. S. Lin, J. C. Tu, and J. Y. Lin, "Keyword detection based on retinanet and transfer learning for personal information protection in document images," *Applied Sciences (Switzerland)*, vol. 11, no. 20, 2021, doi: 10.3390/app11209528.
[8] S. Cakic, T. Popovic, S. Sandi, S. Krco, and A. Gazivoda, "The Use of Tesseract OCR Number Recognition for Food Tracking and Tracing," *2020 24th International Conference on Information Technology, IT 2020*, no. February, 2020, doi: 10.1109/IT48810.2020.9070558.
[9] S. Sulaiman, R. A. Wahid, and F. Morsidi, "Feature extraction using regular expression in detecting proper noun for Malay news articles based on KNN algorithm," *Journal of Fundamental and Applied Sciences*, vol. 9, no. 5S, p. 210, Jan. 2018, doi: 10.4314/jfas.v9i5s.16.
[10] J. M. López-Correa, H. Moreno, A. Ribeiro, and D. Andújar, "Intelligent Weed Management Based on Object Detection Neural Networks in Tomato Crops," *Agronomy*, vol. 12, no. 12, 2022, doi: 10.3390/agronomy12122953.
[11] G. Dorrer, M. Koriukin, S. Yushkova, and L. Sviridova, "Vehicle detection in aerial images," in *IOP Conference Series: Earth and Environmental Science*, Institute of Physics Publishing, Aug. 2019. doi: 10.1088/1755-1315/315/2/022014.
[12] C. Patel, A. Patel, and D. Patel, "Optical Character Recognition by Open source OCR Tool Tesseract: A Case Study," *Int J Comput Appl*, vol. 55, no. 10, pp. 50–56, Oct. 2012, doi: 10.5120/8794-2784.
[13] T. Hegghammer, "OCR with Tesseract, Amazon Textract, and Google Document AI: a benchmarking experiment," *J Comput Soc Sci*, vol. 5, no. 1, pp. 861–882, May 2022, doi: 10.1007/s42001-021-00149-1.
[14] S. Heh, "Character and Image Recognition for Data Cataloging in Ecological Research," Academy and Industry Research Collaboration Center (AIRCC), Apr. 2018, pp. 65–76. doi: 10.5121/csit.2018.80606.
[15] R. G. De Luna, "A Tesseract-based Optical Character Recognition for a Text-to-Braille Code Conversion," vol. 10, no. 1, 2020.
[16] P. Chakraborty *et al.*, "Recognize Meaningful Words and Idioms from the Images Based on OCR Tesseract Engine and NLTK," in *Lecture Notes in Electrical Engineering*, Springer Science and Business Media Deutschland GmbH, 2022, pp. 297–310. doi: 10.1007/978-981-19-1520-8_23.

 DOI : 10.37823/insight.v6i2.315