

## Bab 3

### ANALISIS DAN DESAIN

#### 3.1 Metode Analisis

##### 3.1.1 Metode Pengumpulan Data

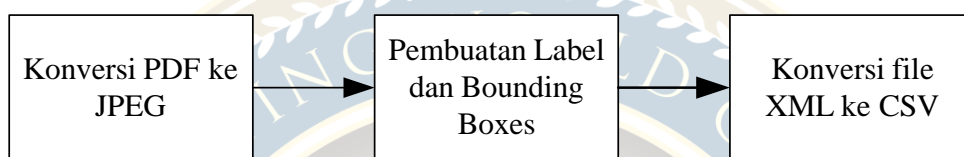
Data yang dipakai dalam tugas akhir ini merupakan data primer yang diperoleh dari “Toko Maju”. Toko Maju memperoleh *file* dokumen pesanan pembelian dari *client* yang terdiri dari beberapa perseroan terbatas (PT) dengan struktur *layout* serupa. Berdasarkan informasi dari pemilik toko, *file* yang dikirim oleh *client* merupakan hasil *scan* dari pesanan pembelian yang dicetak, sehingga terdapat gambar yang buram ataupun miring. *File* dokumen yang akan digunakan adalah sejumlah 520 untuk *training* dan 100 untuk *testing* model.

##### 3.1.2 Desain Instrumen

Proses pengumpulan data dimulai dengan melakukan kontak terhadap pemilik “Toko Maju”. Setelah dokumen pesanan pembelian diperoleh, dilakukan konversi dokumen dari format PDF ke JPEG dengan menggunakan *library* pdf2image.

Untuk melakukan proses *training model object detection*, maka diperlukan *dataset* yang memuat letak dari daerah yang ingin diekstrak. Oleh karena itu, dengan menggunakan LabelImg, akan dibentuk 9 *bounding boxes* pada gambar dokumen, yaitu nama perusahaan, nomor pesanan pembelian, tanggal, nama barang, kuantitas, unit, harga satuan, subtotal harga, dan total harga keseluruhan.

Hasil dari pelabelan gambar adalah *file* XML yang berisi koordinat *bounding boxes* beserta label kelas. *File* XML ini kemudian akan dikonversi menjadi format CSV dengan memuat nama dari dokumen gambar, koordinat (xmin, ymin, xmax, ymax), dan label kelas. Hasil akhir proses pengumpulan data adalah sebuah *file* CSV yang akan digunakan sebagai data *input* untuk proses *training*. Alur pengumpulan data dapat dilihat pada Gambar 3.1.



**Gambar 3.1 Alur Data Preparation**

Dari total 520 gambar, dilakukan *image processing* berupa *rotation* dengan beberapa sudut, sebesar 3, 177, 180, 183, dan 357 derajat. Total terdapat 3.120 gambar yang telah diberi *bounding boxes* dan siap diolah untuk tahap selanjutnya.

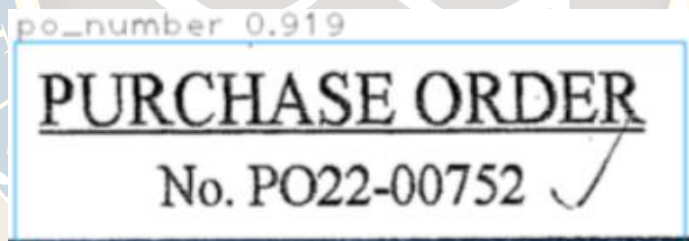
## **3.2 Persiapan Data**

### **3.2.1 Exploratory Data Analysis**

Dokumen pesanan pembelian memiliki banyak informasi penting yang dapat digunakan untuk kepentingan bisnis. Pada tugas akhir ini, peneliti memfokuskan penarikan informasi pada 9 kelas, yaitu nama perusahaan, nomor pesanan pembelian, tanggal pesanan pembelian, nama barang yang dipesan, kuantitas barang dipesan, unit barang dipesan, harga barang yang dipesan, subtotal harga barang dipesan, serta total harga keseluruhan dari barang yang dipesan.

Dataset yang digunakan untuk *training* model *object detection* merupakan dataset primer berupa *file* CSV sejumlah 9 baris  $\times$  520 gambar, yang memuat nama *file*, koordinat (*xmin*, *ymin*, *xmax*, *ymax*), dan label kelas. Proses *training* dengan model Keras RetinaNet dilakukan sebanyak 10 *epochs*, dengan *batch size* 25, dan *steps per-epoch* sebesar 120. Model yang dihasilkan dari proses *training* akan melakukan prediksi berupa letak *bounding boxes* beserta label kelas.

*Threshold* untuk nilai *Intersection over Union* (IoU) yang dipasang adalah 0.5 mengacu pada penelitian sebelumnya (Farady et al., 2020; Lin et al., 2021; López-Correa et al., 2022). Sedangkan, nilai *confidence* akan dipilih dari kumpulan nilai *confidence* terbesar dari setiap label untuk memastikan hanya terdapat 1 *bounding box* untuk 1 label (López-Correa et al., 2022). Contoh hasil dari proses prediksi oleh model dapat dilihat pada Gambar 3.2, dimana model menggambar letak prediksi dari nomor pesanan pembelian.



**Gambar 3. 2 Contoh Hasil Deteksi Letak**

Keseluruhan hasil prediksi kemudian disimpan kedalam sebuah *dataframe* yang memuat koordinat letak (*xmin*, *ymin*, *xmax*, *ymax*), label, serta nilai *confidence*, seperti pada Gambar 3.3.

	xmin	ymin	xmax	ymin	ymax	labels	score	id_label
0	211	118	1586	668	668	company_name	0.999678	0
1	1581	399	2770	852	852	po_number	0.997926	1
2	2870	878	3616	1109	1109	date	0.993312	2
3	1684	1361	2660	1762	1762	product_desc	0.997234	3
4	2629	1428	2967	1798	1798	product_qty	0.999905	4
5	2940	1443	3131	1808	1808	product_unit	0.999706	5
6	3112	1456	3554	1843	1843	product_price	0.999564	6
7	3518	1484	4080	1874	1874	product_totalprice	0.999867	7
8	2998	3043	3990	3527	3527	grand_total	0.999480	8

**Gambar 3. 3 Koordinat Hasil Prediksi**

### 3. 2. 2 Data Analysis

Proses OCR dilakukan dengan menggunakan *library* Tesseract pada setiap potongan label yang telah disebutkan pada sub-bab 3.1.2. sesuai dengan koordinat letak serta label kelas yang telah dideteksi. Pada tahapan OCR, dilakukan deteksi karakter dari gambar yang akan menghasilkan sekumpulan teks. Hasil deteksi karakter dapat dilihat pada Gambar 3.4.

Description
CABLE TIE NILON HITAM 400X4.5Mh
CABLE TIE NILON HITAM 550X5MM-
CABLE NYHY 3X2.5MM?-JUMBO
CABLE NYHY 4X4MM?-ETERNA

**Gambar 3. 4 Hasil Deteksi Karakter**

Setelah dilakukan deteksi karakter, akan dilakukan *post-processing* pada teks yang dihasilkan sebelum dimasukkan kedalam *database*. Proses ini meliputi beberapa perlakuan, seperti *case folding*, *punctuation removal*, dan *number*

*handling*. Kemudian, akan dilakukan tambahan metode *rule-based*, yang disesuaikan dengan kasus yang terjadi pada teks hasil deteksi karakter dengan kebutuhan. Meskipun demikian, metode *rule-based* bukan menjadi metode utama dalam *information extraction*, melainkan metode pendukung untuk memperbaiki hasil deteksi karakter.

Proses terakhir ialah memasukkan informasi yang telah ditemukan kedalam *database* dengan menggunakan *library pymysql*. *Database* yang digunakan adalah MySQL dengan *hostname* localhost. Hasil *information extraction* akan ditampilkan dengan bentuk tabel pada sebuah *webpage* yang menggunakan *framework* Laravel.

Evaluasi model akan menggunakan matriks *accuracy*, dimana evaluasi akan didasarkan pada berhasil atau tidaknya model dalam mengekstrak suatu label sebagaimana mestinya. Semakin tinggi nilai *accuracy*, menunjukkan semakin baik performa dari model dalam melakukan *information extraction*.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)}$$

Keterangan:

TP + TN = Jumlah label yang diprediksi benar

TP + FP + FN + TN = Jumlah keseluruhan label yang di prediksi