

Pembuatan Aplikasi Validasi Document Tagihan Pembelian Barang Secara Digital Menggunakan OCR dengan tool tesseract pada System Portal Perusahaan

Pius Aldi Septio¹, Sri Yulianto Joko Prasetyo²

^{1,2}Teknik Informatika, Universitas Kristen Satya Wacana, Indonesia
e-mail: 672019046@student.uksw.edu¹, sri.yulianto@uksw.edu²

Abstract

This journal reviews the Making of a Digital Invoices Document Validation Application Using OCR with the tesseract tool on the Company Portal System. This study aims to implement Optical Character Recognition (OCR) using the tesseract tool in the company portal to make it easier to validate invoices. The method used is a prototype with the aim of building a system model where the system is based on user needs where the user does not provide input and output details. The results of the application that was made were tested using Black Box testing with the results that all application functions can be used properly.

Keywords: invoice, optical character recognition

Abstrak

Jurnal ini mengulas Pembuatan Aplikasi Validasi Document Tagihan Pembelian Barang Secara Digital Menggunakan OCR dengan tool tesseract pada System Portal Perusahaan. Penelitian ini bertujuan untuk melakukan penerapan Optical Character Recognition (OCR) menggunakan tool tesseract dalam portal perusahaan untuk mempermudah dalam melakukan validasi tagihan pembelian barang secara digital. Metode yang digunakan adalah prototype dengan tujuan membangun sebuah model system dimana system ini berdasarkan kebutuhan user yang dimana user tidak memberikan detail input dan output. Adapun hasil aplikasi yang di buat di test menggunakan Black Box testing dengan hasil semua fungsi aplikasi dapat digunakan dengan semestinya.

Kata kunci: invoice, optical character recognition

1. PENDAHULUAN

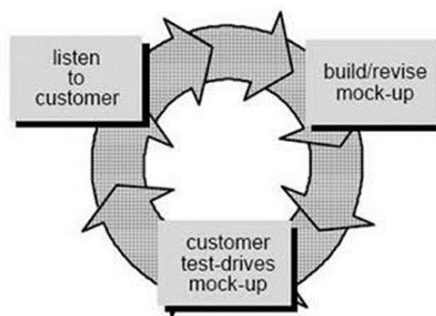
Dengan berkembangnya teknologi sekarang, banyak sekali digitalisasi yang dilakukan oleh perusahaan untuk mempermudah proses bisnis mereka, dapat dilihat dari perusahaan retailment yang banyak mengimplementasikan digitalisasi di badan perusahaan mereka untuk mempermudah bisnis mereka, salah satu digitalisasi yang dilakukan adalah digitalisasi invoice, dimana invoice penjualan dan pembelian vendor sekarang sudah berbentuk citra digital, Peralihan ini dilakukan untuk mempermudah proses bisnis mereka dimana bukti pembelian dan penjualan/ invoice barang dapat lebih mudah disimpan dan diproses jika dalam bentuk digital, dan juga mempermudah dalam menvalidasi suatu pembelian atau penjualan barang yang dilakukan customer hanya dengan menginput data transaksi diportal yang telah dibuat perusahaan itu sendiri. Walaupun sudah terdigitalisasi, tetap muncul masalah dimana banyaknya invoice yang diterima dari pembelian customer ataupun vendor sangat sulit untuk divalidasi dikarenakan berbagai jenis template invoice yang digunakan yang mempersulit proses validasi invoice tersebut, yang dimana kemudahan dalam validasi invoice sangatlah

penting untuk mempercepat proses suatu pembelian atau penjualan benda atau jasa agar dapat memastikan apakah suatu pembelian atau penjualan barang atau jasa tersebut sah dan dapat diterima pembeli ataupun penjual.

Solusi yang dapat diterapkan atas masalah tersebut adalah dengan penerapan OCR dalam membaca character pada invoice yang berfokus mencari no invoice dan total tagihan dari invoice tersebut yang disamakan dengan hasil inputan user dari portal dibuat perusahaan. Salah satu cara untuk mengenali character tersebut adalah dengan menggunakan *Optical Character Recognition* (OCR). *Character Recognition(OCR)* merupakan salah satu teknologi information retrieval yang berfungsi untuk memproses informasi yang didapatkan melalui Gambar ataupun text, Teknik automisasi ini muncul dengan tujuan mengatasi masalah menginput data digital dengan jumlah banyak, yang dimana masalah tersebut dapat diselesaikan dengan menggunakan OCR [1]. Tool yang digunakan untuk implementasi OCR dalam penelitian ini adalah tesseract, alasan pemilihan tesseract dikarenakan tesseract adalah salah satu open source OCR tool terbaik dalam implementasikan OCR dan termasuk salah satu tool OCR terpopuler yang banyak dipakai developer OCR. Dalam penelitian kali ini akan menggunakan metode prototype dalam perancangan system, setelah menemukan metode yang cocok dalam perancangan system dilanjutkan dengan pengumpulan data invoice dan selanjut akan diproses menggunakan Optical Character Recognition (OCR), hasil output berbentuk file text .txt. dilanjutkan dengan pengolahan data agar mendapatkan data terbaik berdasarkan hasil akurasi data tersebut.

2. METODOLOGI PENELITIAN

Metode perancangan system yang digunakan adalah system prototyping, prototype model adalah proses membangun sebuah model system dimana system tersebut akan dievaluasi, jika system itu tidak layak maka system tersebut akan diperbaiki dan apabila system tersebut dianggap layak maka system tersebut akan di produksi [3].



Gambar 1. bagan prototype model

Kelebihan dari metode ini adalah kemampuan membangun sesuai kebutuhan user yang memiliki requirement yang kurang jelas mengidentifikasi input, prose maupun output, metode prototype juga dapat membangun relasi yang dari pihak pengguna dan pengembang, sehingga pengguna mempunyai peran yang aktif

dalam proses pengembangan aplikasi, dapat diimplementasikan di setiap model perangkat lunak dan lebih menghemat waktu [5].

2.1. OCR

Optical Character Recognition (OCR) merupakan salah satu teknologi information retrieval yang berfungsi untuk memproses informasi yang didapatkan melalui Gambar ataupun text, Teknik automisasi ini muncul dengan tujuan mengatasi masalah menginput data digital dengan jumlah banyak, yang dimana masalah tersebut dapat diselesaikan dengan menggunakan OCR [1].

a) File Input

File input berupa file digital dengan format pdf.

b) Preprocessing merupakan suatu proses untuk menghilangkan bagian-bagian yang tidak diperlukan pada Gambar input, tahap preprocessing dapat dibagi menjadi 2 tahap yaitu grayscale image dan binary image, dimana grayscale adalah proses konversi Gambar yang semula hanya 3 lapis warna yaitu RGB menjadi Gambar gray yang hanya memiliki 1 lapis warna [6]. Perubahan citra RGB menjadi grayscale dapat dilakukan dengan menggunakan rumus:

$$S = \frac{\text{red} + \text{green} + \text{blue}}{3} \quad (1)$$

Dan tahap binary image. Binary image adalah tahap pengubahan Gambar menjadi warna hitam dan putih, yang dimana warna hitam akan mewakili angka 0 dan warna putih akan mewakili angka 1, proses pengubahan warna tersebut disebut thresholding [2], Yang dimana salah satu cara untuk memproses konversi ini adalah dengan melakukan proses thresholding, Proses thresholding adalah proses pengubahan Gambar menjadi Gambar binary dengan menentukan nilai pemisah (threshold value) [6]. Proses ini bertujuan untuk memisahkan daerah mana yang termasuk background dan foreground, hal ini akan mengubah citra grayscale menjadi binary image sehingga dapat memisahkan objek background yang ada.

$$g(x,y) = \begin{cases} 1, & \text{if } f(x,y) > T \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

c) Segmentasi

Segmentasi adalah proses memisahkan character image ke beberapa sub component, dimana terdapat 3 tahap segmentasi yaitu line segmentasi, word segmentasi, dan character segmentasi. Sebelum melakukan tahapan segmentasi dilakukan histogram projection method [7]. histogram projection method adalah metode menghitung jumlah foreground image yang ada didalam Gambar yang telah di proses di tahapan Binary Image [10]. Terdapat dua tahapan Histogram projection method yaitu Horizontal Histogram Method dan Vertikal Histogram Method.

d) Normalisasi

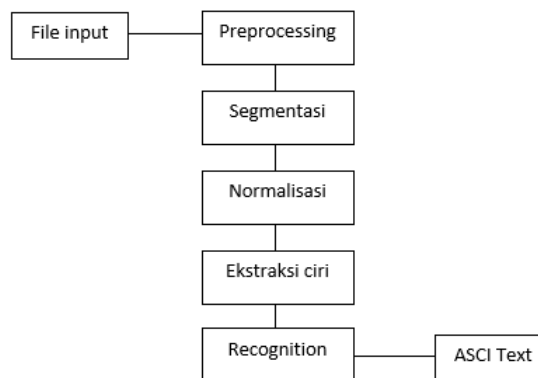
Normalisasi adalah proses perubahan dimensi dan ketebalan tiap karakter, normalisasi bertujuan untuk memastikan bahwa input data dapat menghasilkan informasi relevan dengan dimensi yang telah diubah tanpa menggunakan ukuran sebenarnya [8].

e) Ekstraksi ciri

Ekstraksi ciri adalah pengambilan ciri tertentu dari suatu character, setiap ciri tertentu digunakan untuk membedakan antar character dengan harapan dapat mendapatkan informasi yang relevan dari data yang di input, sehingga klasifikasi input data dapat dilakukan melalui ekstraksi ciri daripada mengenali data input secara keseluruhan [7].

f) Recognition

Recognition merupakan proses untuk mengenali karakter yang diamati dengan cara membandingkan ciri-ciri karakter yang diperoleh dari proses ekstraksi ciri dengan ciri-ciri karakter yang ada pada basis data, berikut proses OCR secara umum.



Gambar 2. Proses OCR secara umum

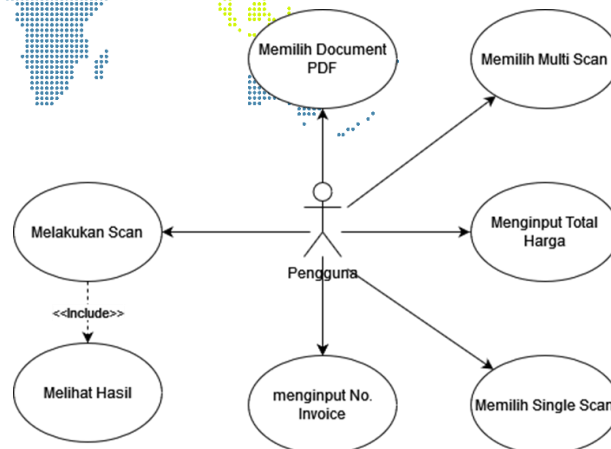
2.2. TESSERACT

Tesseract adalah open source Optical Character Recognition (OCR) engine yang tersedia dibawah Apache 2.0, tesseract dapat digunakan langsung ke program dengan menggunakan API untuk membaca text dari file print digital [2]. Tesseract awalnya dikembangkan sebagai perangkat lunak oleh Hewlett-Packard pada tahun 1985 - 1995, 10 tahun kemudian Hewlett Packard dan UNLV merilis Tesseract sebagai produk open source pada tahun 2005, dan sekarang Tesseract dikembangkan oleh google dan pernah menjadi salah satu engine OCR paling akurat pada tahun 2006 [4].

2.3. Perancangan System

a. Unified Modelling Language

System aplikasi dirancang menggunakan UML (Unified Modelling Language), Use Case Diagram digunakan untuk mendeskripsikan fungsional suatu system dan cara kerjanya, serta menjelaskan secara visual actor apa saja yang berperan dalam penggunaan system dan interaksinya.

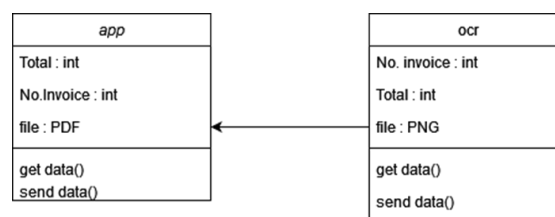


Gambar 3. Use Case Diagram

Pada aplikasi Validasi *Document Invoice* Pada Portal Perusahaan yang berperan sebagai actor adalah pengguna. Pengguna memilih fungsi memilih Document PDF, Memilih single scan atau multi scan, Menginput Total Harga, Menginput No. Invoice, melakukan scan dan melihat hasil. Fungsi memilih Document PDF untuk memilih Document PDF yang ingin di scan, fungsi memilih multi scan bertujuan untuk menampilkan menu multi scan, fungsi single scan bertujuan untuk menampilkan menu single scan, fungsi melakukan scan bertujuan untuk melihat hasil invoice yang telah di scan.

b. Class diagram

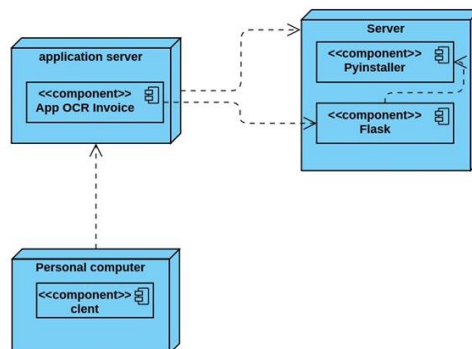
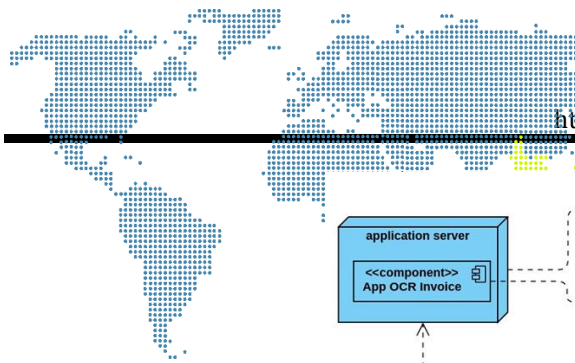
Class diagram mengGambarkan struktur objek, class, package, deskripsi objek dan relasi objek satu sama lain.



Gambar 4. Class diagram

c. Deployment diagram

Deployment diagram bertujuan memvisualisasikan hubungan antara *software* dan *hardware*, secara spesifik mengGambarkan bagaimana komponen perangkat lunak berhubungan dengan komponen perangkat keras.



Gambar 5. *Deployment diagram*

3. HASIL DAN PEMBAHASAN

3.1. Hasil dan implementasi Menu Home

Halaman menu home berfungsi sebagai user interface utama aplikasi, berdasarkan hasil rancangan dari prototype pertama, terdapat dashboard, dua tombol dan dua text box, tombol terdiri dari tombol submit dan tombol upload, text box terdiri dari text box untuk mengisi no. invoice dan Total, dan dashboard berisi menu ke page home dan Multi, hasil dapat dilihat pada Gambar 9.

Gambar 6. tampilan home aplikasi

Procedure 1 adalah kode program untuk melakukan pengecekan ekstensi file, penyimpanan file ke folder dan menampilkan hasil. Baris 1 – 3 bertujuan untuk mengambil data dari UI, Baris ke 4 bertujuan untuk memasukan semua path yang telah di buat ke dalam satu list atau array, dilanjutkan ke baris program 5 – 10 yang dimana baris tersebut bertujuan untuk mengecek setiap filename yang ada di list path dan menghapus semua file yang ada di list path tersebut, baris program 11 untuk mengambil nama file di file yang kita upload, baris 12 – 13 bertujuan untuk memisahkan nama file dan ekstensi file, baris 14 bertujuan untuk menggabung nama file dengan ekstensi png, baris 15 – 17 bertujuan untuk mengecek apakah ekstensi file yang kita upload berbentuk pdf yang dimana jika bukan akan menampilkan pesan error, baris program 18 bertujuan untuk menyimpan file di folder yang telah ditentukan, dan baris 19 – 20 bertujuan untuk



mengirim variable untuk di proses menggunakan Tesseract engine dan menampilkan hasil proses recognition.

```
Title : program pengecekan ekstensi file dan menyimpan file
Declaration
Var files : File
Var list_path : list
Var file, filename, extension, png,No, Total : String
Var folder_img, folder_pdf, folder_final : path_data
Implementation
// get data from ui
1. No = request.form['No']
2. Total = request.form['Total']
3. files = request.form['file']
//put all folder path in list
4. list_path = [folder_img, folder_pdf, folder_final]
//for each name of data in folder path, delete the data
5. for list_name in list_path:
6.     for file_name in os.listdir(list_name):
7.         file = list_name + file_name
8.         if os.path.isfile(file):
9.             print('Deleting file:', file)
10.            os.remove(file)
//get file name
11. filename = secure_filename(files.filename)
//separate the filename to filename and file extension
12. extension = os.path.splitext(filename)[1]
13. filename = os.path.splitext(filename)[0]
//combine filename and desire extension
14. png = filename+'.png'
//check the file extension
15. if extension not in app.config['UPLOAD_EXTENSIONS']:
16.     error = "file ekstensi salah (pdf)"
17.     return(error)
//save file
18. files.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
// send variable to proses ocr and show output
19. hasil = ocr.printall(No,Total,filename)
20. return(hasil,png)
```

Gambar 7. check extension file, save file pdf, and show output

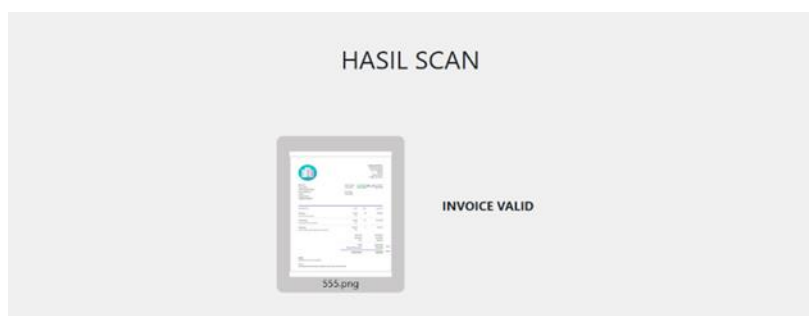
Procedure 2 adalah kode program untuk melakukan Recognition dengan tujuan mendapatkan hasil akhir recognition, pada line 1 – 11 adalah proses modifikasi Gambar untuk meningkatkan kualitas recognition dimana pada line 1 adalah proses gray dengan tujuan untuk menampilkan Gambar dalam bentuk hitam, putih ataupun gray, line 4 dan 5 berfungsi untuk mengurangi noise pada Gambar, pada line 5 - 10 berfungsi untuk menekankan foreground dan background pada img. Pada line 11 – 13 adalah proses recognition menggunakan engine tesseract, dimana line 11 adalah setingan untuk menggunakan semua engine yang ada dan melakukan proses recognition berdasarkan Gambar yang berbentuk invoice, line 12 adalah training data yang digunakan dimana training data yang digunakan adalah training data dalam Bahasa English/training data yang dikenal character ASCII, dan line 13 adalah proses recognition menggunakan tesseract engine. Pada line 14 – 17 adalah proses mengubah character yang tidak diinginkan menjadi character yang diperlukan dan hasil akhir setelah diubah akan berbentuk list, line 18 – 20 bertujuan untuk mencari character yang diinginkan dari input no invoice dan total, dan pada line 22 – 26 bertujuan untuk mencocokkan hasil character yang didapat dengan no dan total yang di input, dimana hasilnya akan sesuai dengan hasil dari kedua character yang didapatkan.



```
Title : program melakukan Optical Character Recognition
Declaration
Var No, Total, config, lang, inv, total : String
Var img : image
Var d, text : dict
Implementation
//preproses the image for a better recognition
1. gray = proses(img)
2. kernel = np.ones
3. img = cv2.dilate(img, kernel)
4. img = cv2.erode(img, kernel)
5. img = cv2.threshold(cv2.GaussianBlur(img)
6. img = cv2.threshold(cv2.bilateralFilter(img)
7. img = cv2.threshold(cv2.medianBlur(img)
8. img = cv2.adaptiveThreshold(cv2.GaussianBlur(img)
9. img = cv2.adaptiveThreshold(cv2.bilateralFilter(img)
10. img = cv2.adaptiveThreshold(cv2.medianBlur(img)
//set a configuration and language model, proses the recognition use tesseract engine
11. config='--oem 3 --psm 4'
12. lang='eng'
13. d = pytesseract.image_to_data(img, output_type=Output.DICT, config = config, lang = lang)
//convert default character for better recognition
14. keys = list(d.keys())
15. n_boxes = len(d['text'])
16. removetable = str.maketrans(',', '.', '@#%$')
17. text = [s.translate(removetable) for s in d['text']]
//searching the closes match of the text that we want to search
18. inv = difflib.get_close_matches(no, text,3,0.75)
19. print(inv)
20. ttl = difflib.get_close_matches(total, text,3,0.75)
21. print(ttl)
//show value valid or not valid if the condition are meet
22. If no != inv or total != not ttl:
23.     messege="INVOICE INVALID"
24. else:
25.     messege= "INVOICE VALID"
26. return(messege)
```

Gambar 8. perform OCR

Setelah semua di input maka akan menampilkan hasil akhir OCR dimana akan menampilkan hasil validasi dan bagian pada Gambar yang berhasil di scan dan mendapatkan hasil yang diinginkan, hasil akhir menu home dapat dilihat pada Gambar 9.

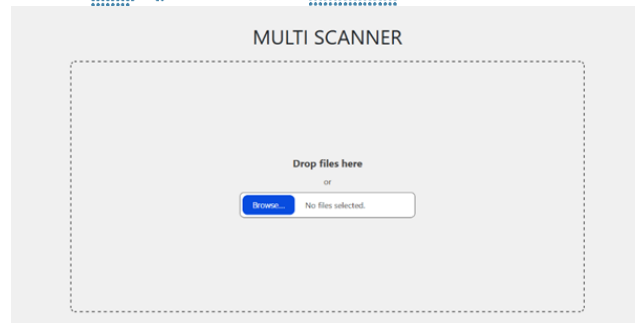


Gambar 9. hasil akhir OCR menu home

3.2. Hasil dan implementasi Menu Multi

Halaman menu multi berfungsi untuk melakukan input multi data, berdasarkan prototype pertama, halaman menu multi berisi dashboard untuk berpindah menu, box untuk mengupload data pdf dan setelah upload akan

mengenerate text box untuk menginput data No invoice dan Total, halaman menu multi dapat dilihat di Gambar 10.



Gambar 10. Tampilan menu multi

```
Title : program pengecekan extansi file,
        generate text box dan menyimpan file pada menu multi
Declaration
Var files : File
Var list_path, listfile : list
Var file, filename, png : String
Var folder_img, folder_pdf, folder_final : path_data
Implementation
1. files = request.files.getlist('file')
//put all folder path in list
2. list_path = [folder_img, folder_pdf, folder_final]
//for each name of data in folder path, delete the data
3. for list_name in list_path:
4.     for file_name in os.listdir(list_name):
5.         file = list_name + file_name
6.         if os.path.isfile(file):
7.             print('Deleting file:', file)
8.             os.remove(file)
// for each file in list file if extension true
// append file to list file, false return error
9. for file in files:
10.    file.save(os.path.join(app.config['UPLOAD_FOLDER'], file.filename))
11.    filename = secure_filename(file.filename)
12.    file_ext = os.path.splitext(filename)[1]
13.    if file_ext not in app.config['UPLOAD_EXTENSIONS']:
14.        error = "file extansi salah (pdf)"
15.        return(error)
16.    listfile.append(filename)
17. return(listfile)
```

Gambar 11. check extensi file dan generate text box menu multi

Procedure 3 adalah adalah kode untuk pengecekan extansi, save file dan generate text box, pada line 1 bertujuan untuk mengambil file request dari UI dalam bentuk list, dilanjutkan pada line 2 – 8 bertujuan untuk memasukan semu path yang diperlukan ke dalam list dan menghapus semua file yang terdapat di path tersebut, dan pada line 9 – 18 bertujuan untuk menyimpan file yang ada pada file list yang didapatkan dan melakukan pengecekan di setiap extansi file list yang ada, jika file tersebut memenuhi kriteria maka file tersebut akan dimasukan ke dalam list yang akan di return ke UI dan jika tidak memenuhi kriteria maka akan return error ke UI.

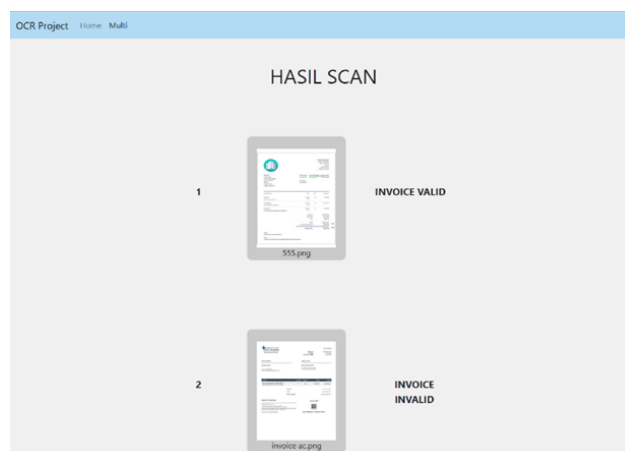


```
Title : program menampilkan output multi
Declaration
Var hasil, data : Dict
Var No, Total, Pdf, file_ext, png : String
Implementation
// for each folder in upload folder, get file name
4.   for i in range(1,len(os.listdir(app.config['UPLOAD_FOLDER']))+1):
// Get value form ui
5.     No = request.form['fname'+str(i)]
6.     Total = request.form['lname'+str(i)]
7.     Pdf = request.form['pdf'+str(i)]
// proses the img to ocr
8.     hasil = ocr.printall(No,Total,pdf)
//get the file name
9.     file_ext = os.path.splitext(pdf)[0]
// give new extension
10.    png = file_ext+'.png'
// put the final output in dict, and empty the dict to next value
11.    output["output"]=hasil
12.    output["nama"]=png
13.    data[i]=output
14.    hasil={}
// return the output
15.    return(data)
```

Gambar 12. menampilkan hasil output multi

Procedure 4 adalah code untuk menampilkan hasil untuk tampilan multi, pada line 1 bertujuan untuk mengambil value nama file yang telah di simpan, dilanjutkan di line 2 – 4 dimana mengambil value dari UI yang nama variable akan digabung dengan nama file yang telah disimpan, dilanjutkan dengan line 5 dan 6 yang bertujuan untuk melakukan proses OCR dengan memanggil fungsi OCR dan memisahkan nama file dari extension sehingga didapatkan nama file yang telah di proses di OCR, dan pada line 7 – 12 bertujuan untuk memberikan extansi baru yaitu png dan memasukan hasil akhir ke dictionary yang dictionary tersebut akan di return ke UI dan ditampilkan hasil akhirnya ke UI.

Setelah semua value di input akan mengGambarkan hasil akhir OCR dimana akan menampilkan semua hasil validasi dan bagian pada Gambar yang berhasil di scan dan mendapatkan hasil yang diinginkan, hasil akhir form multi dapat diliat di Gambar 13.



Gambar 13. hasil akhir OCR Multi

3.3. Pengujian system

Pengujian system dilakukan untuk mengetahui seberapa jauh suatu aplikasi dapat bekerja dan juga bertujuan untuk menemukan kesalahan yang terjadi dalam aplikasi tersebut. Pengujian aplikasi kali ini menggunakan blackbox testing. Blackbox Testing merupakan pengujian perangkat lunak yang bertujuan untuk menguji fungsionalitas suatu software tanpa memperhatikan detail software tersebut, pengujian hanya ingin mengetahui nilai keluar berdasarkan nilai masuk dari software tersebut[9]. Pengujian antar muka bertujuan untuk mengecek apakah setiap halaman dalam antarmuka dapat berfungsi dengan baik, dimana setiap halaman tersebut dapat saling berhubungan dengan halaman lain. Pengujian fungsi bertujuan untuk mengetahui apakah fungsi yang terdapat di menu utama dan menu multi dapat berjalan sesuai yang diharapkan.

Tabel 1. Pengujian system

Pengujian	Aksi	Hasil Yang Diharapkan	Hasil Yang Muncul	Status Pengujian
Halaman Home	Menginput Invoice dan Total	Menampilkan No invoice dan Total	Tampil No invoice dan Total	Valid
	Menginput Data PDF	Menampilkan nama file	Tampil nama file	Valid
	Menekan tombol input	Menampilkan hasil scanning OCR	Menampilkan hasil OCR	Valid
	Menekan tombol input dengan invoice dan total kosong	Menampilkan warning textbox invoice dan total kosong	Menampilkan warning textbox invoice dan total kosong	Valid
	Menekan tombol input dengan extensi file bukan PDF	Menampilkan warning Jenis file salah	Tampil warning jenis file salah	Valid
	Menekan hasil Gambar setelah OCR	Memperbesar Gambar hasil OCR	Tampil hasil Gambar yang diperbesar	Valid
	Menekan pilihan multi di dashbord	Menampilkan tampilan Multi	Tampil tampilan multi	Valid
	Menginput data PDF	Menampilkan nama file dan textbox No invoice dan Total	Tampil nama file dan textbox No invoice dan Total	Valid
	Menginput data file dengan extensi file bukan PDF	Menampilkan warning Jenis file salah	Tampil warning jenis file salah	Valid
	Menginput Invoice dan Total	Menampilkan No invoice dan Total	Tampil No invoice dan Total	Valid
	Menekan tombol input	Menampilkan hasil scanning multiple	Menampilkan hasil multiple scanning	Valid



Pengujian	Aksi	Hasil Yang Diharapkan	Hasil Yang Muncul	Status Pengujian
Halaman Multi		OCR	OCR	
	Menekan tombol input dengan invoice dan total kosong	Menampilkan warning textbox invoice dan total kosong	Menampilkan warning textbox invoice dan total kosong	Valid
	Menekan hasil Gambar setelah OCR	Memperbesar Gambar hasil OCR	Tampil hasil Gambar yang diperbesar	Valid
	Menekan home di dashbord	Menampilkan tampilan halaman utama	Tampil tampilan halaman utama	Valid

4. SIMPULAN

Dari penelitian ini dapat ditarik kesimpulan antara lain validasi document invoice dapat dilakukan dengan memanfaatkan teknologi OCR dalam membangun media membaca dan memvalidasi invoice dengan memasukan file PDF pada aplikasi OCR. Aplikasi OCR untuk memvalidasi document invoice dapat terwujud dengan memanfaatkan teknologi Tesseract dan berdasarkan uji coba blackbox diketahui bahwa aplikasi OCR untuk validasi document invoice dapat digunakan untuk memvalidasi document invoice.

DAFTAR PUSTAKA

- [1] K. A. Elmizan, D. A. Sunarna, B. Bangun and A. Dhanika, "mti.binus.ac.id," BINUS UNIVERSITY, 26 9 2018. [Online]. Available: <https://mti.binus.ac.id/2018/12/26/optical-character-recognition-ocr-menggunakan-tesseract-dan-penerapannya-pada-industri-digital-di-indonesia/>. [Accessed 24 11 2022].
- [2] F. Zelic and A. Sable, "nanonets.com," Nanonets, 2 2023. [Online]. Available: <https://nanonets.com/blog/author/anuj/>. [Accessed 3 11 2022].
- [3] V. M. M. Siregar, H. Sugara and G. A. Purba, "Aplikasi Pencatatan Laporan Penjualan," *TEKINKOM*, vol. 2, no. 1, p. 7, 2019.
- [4] A. Firdaus, M. S. Kurnia, T. Shafera and W. I. Firdaus, "Implementasi Optical Character Recognition (OCR) Pada Masa Pandemi Covid-19," *JUPITER*, vol. 13, no. 2, pp. 188 - 194, 2021.
- [5] F. Supandi, W. D. P, Y. A. S and M. Sudir, "Analisis Resiko Pada Pengembangan Perangkat Lunak Yang Menggunakan Metode Waterfall Dan Prototyping," *Prosiding Seminar Dinamika Informatika*, vol. 2, no. 1, 2018.
- [6] W. Satyawan, "Citizen Id Card Detection using Image Processing and Optical Character Recognition," *IOPscience*, vol. 1235, no. 1, 2018.
- [7] M. A. Awel and A. . I. Abidi, "REVIEW ON OPTICAL CHARACTER RECOGNITION," *International Research Journal of Engineering and Technology (IRJET)*, vol. 6, no. 6, pp. 3666 - 3669, 2019.
- [8] S. Hartanto, A. Sugiharto and S. N. Endah, "Optical Character Recognition Menggunakan Algoritma Template Matching Correlation," *Masyarakat Informatika*,



vol. 5, no. 9, 2014.

- [9] F. C. Ningrum, D. Suherman, S. Aryanti, H. A. Prasetya and A. Saifudin, "Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik Equivalence Partitions," *Jurnal Informatika Universitas Pamulang*, vol. 4, no. 4, pp. 125 - 130, 2019.
- [10] S. Reddy, "towardsdatascience.com," Medium, 25 march 2019. [Online]. Available: <https://towardsdatascience.com/segmentation-in-ocr-10de176cf373>. [Accessed 24 11 2022].