

Implementasi *Optical Character Recognition* dan *Text Similarity* untuk Pelaporan Uang Muka Kerja

Meredita Susanty^{1*}, Megandi², Erwin Setiawan³

^{1,2,3} Program Studi Ilmu Komputer, Universitas Pertamina, Jl. Teuku Nyak Arief Simprug
Kebayoran Lama Jakarta Selatan 12220, Indonesia
e-mail: meredita.susanty@universitaspertamina.ac.id

*Corresponding author

Submitted Date: July 13th, 2022
Revised Date: October 9th, 2023

Reviewed Date: July 26th, 2022
Accepted Date: Desember 31th, 2024

Abstract

The request and accountability reporting of cash advances in companies are now widely supported by web-based applications. However, the process of verifying the completeness and consistency of expense receipts against financial reports is still performed manually by financial staff, which is time-consuming. This study aims to implement Optical Character Recognition (OCR) technology to extract text from payment receipts and match it with expenditure items in financial reports. This technology enables the automation of financial verification processes using a text similarity algorithm based on Levenshtein Distance. By implementing this system, the time required for financial report verification can be significantly reduced, allowing staff to focus on more strategic tasks. The system was tested by integrating the Veryfi OCR application with an API-based system to process payment receipts in various formats. The test results indicate that the system successfully identifies and matches transactions with a high accuracy rate. However, further development is needed to incorporate discount and tax calculations from payment receipts.

Keywords: Cash Advance, Application Programming Interface, Text Similarity, Optical Character Recognition

Abstrak

Permintaan dan pelaporan pertanggungjawaban uang muka kerja dalam perusahaan saat ini telah banyak dibantu oleh aplikasi berbasis web. Namun, proses pemeriksaan kelengkapan dan kesesuaian bukti pengeluaran terhadap laporan pertanggungjawaban masih dilakukan secara manual oleh staf keuangan, yang memerlukan waktu cukup lama. Penelitian ini bertujuan mengimplementasikan teknologi Optical Character Recognition (OCR) untuk membaca tulisan pada bukti pembayaran dan menyesuaikannya dengan item pengeluaran dalam laporan pertanggungjawaban. Teknologi ini memungkinkan otomatisasi proses verifikasi keuangan dengan memanfaatkan algoritma kesamaan teks (text similarity) berbasis Levenshtein Distance. Dengan sistem ini, waktu yang dibutuhkan untuk memeriksa laporan keuangan dapat dikurangi secara signifikan, sehingga staf dapat lebih fokus pada tugas yang lebih strategis. Pengujian dilakukan dengan mengintegrasikan aplikasi Veryfi OCR dan sistem berbasis API untuk memproses bukti pembayaran dalam berbagai format. Hasil pengujian menunjukkan bahwa sistem mampu mengidentifikasi dan mencocokkan transaksi dengan tingkat akurasi yang cukup tinggi. Namun, sistem masih perlu dikembangkan lebih lanjut untuk menangani perhitungan diskon dan pajak yang terdapat dalam bukti pembayaran.

Kata kunci: Uang Muka Kerja, *Application Programming Interface*, *Text Similarity*, *Optical Character Recognition*



1. Pendahuluan

Uang Muka Kerja (UMK) adalah sejumlah dana yang diberikan kepada karyawan sebagai dukungan operasional perusahaan dalam jangka waktu tertentu (Stefan, 2020). Untuk mendapatkan UMK, karyawan perlu mengajukan permohonan dan menunggu persetujuan dari atasan sebelum dana tersebut dicairkan. Setelah melakukan pekerjaan, karyawan yang telah menerima UMK harus mempertanggungjawabkan penggunaan dana dengan mengirimkan laporan yang menyertakan bukti pembelanjaan. Dalam proses penyelesaian, karyawan tersebut harus melaporkan jumlah dana yang diterima, jumlah dana yang digunakan untuk setiap item atau kegiatan tertentu, serta melampirkan bukti transaksi seperti struk (John et al., 2021).

Setelah karyawan melaporkan penggunaan UMK, tim keuangan atau sumber daya manusia akan melakukan pengecekan terhadap kecocokan penggunaan UMK dan kesesuaian bukti pengeluaran yang terdapat dalam laporan. Tindakan ini sangat penting dilakukan untuk mencegah terjadinya penyalahgunaan dana dan potensi kerugian finansial yang dapat menimpa perusahaan. Penggunaan dana UMK oleh karyawan akan diperiksa dengan membandingkan jumlah dana yang diajukan, jumlah dana yang diterima, jumlah dana yang digunakan, serta bukti transaksi yang menunjukkan penggunaan dana tersebut.

Umumnya perusahaan sudah menggunakan sistem untuk melakukan pengajuan dan pelaporan UMK, namun pemeriksaan bukti pembelanjaan dan laporan penggunaan dana untuk setiap item pengeluaran umumnya masih dilakukan secara manual oleh manusia. Misalkan, volume transaksi 100 laporan pengeluaran per bulan dengan jumlah bukti transaksi hingga 350 struk. Jika waktu rata-rata untuk memeriksa sebuah struk adalah 5 menit, maka total waktu yang dibutuhkan untuk memeriksa satu laporan pengeluaran bisa mencapai dua hari. Jumlah laporan pengeluaran yang tinggi dibandingkan jumlah staf dan waktu yang dibutuhkan untuk memeriksa satu persatu bukti transaksi menyebabkan seringnya keterlambatan pemrosesan pelaporan UMK pada perusahaan.

Melihat perkembangan teknologi *Optical Character Recognition* (OCR) (Du et al., 2020) (Kim et al., 2022) (Park & Lee, 2019) yang performanya semakin meningkat dan semakin luas digunakan diberbagai aplikasi seperti pembacaan plat nomor kendaraan (Hanif et al., 2023)(Cahyo,

2019)(Tirtana et al., 2021), struk atau nota belanja (Benedikt et al., 2020)(Kumar et al., 2020)(Kata et al., 2018) atau pembacaan dokumen admisi (Susanty & Nugroho, 2020).

Penelitian ini bertujuan menggunakan teknologi OCR untuk membaca struk atau nota kemudian mencocokkan data yang dibaca dengan laporan keuangan yang dibuat oleh karyawan. Dengan otomasi yang dilakukan, waktu yang dibutuhkan untuk memeriksa laporan keuangan dapat dikurangi sehingga staf perusahaan dapat melakukan pekerjaan lain yang lebih strategis.

2. Metodologi Penelitian

Rancang bangun aplikasi mobile dilakukan menggunakan mengikuti metodologi waterfall (Doolittle, 2020) (Kodmelwar et al., 2022) (Pressman & Maxim, 2014) (Sommerville, 2016) dalam pengembangan perangkat lunak. Tahapan pengembangan perangkat lunak yang dilakukan ditunjukkan pada Gambar 1, yang terdiri dari pengumpulan kebutuhan, desain, pengembangan, dan pengujian. Secara umum, aplikasi yang dikembangkan dirancang untuk pembacaan karakter dari gambar menggunakan OCR dan mengelompokkan data pada bukti transaksi menjadi tanggal transaksi, item transaksi, jumlah tiap item, harga tiap item, pajak, dan total transaksi kemudian mengirimkan hasil pembacaan menggunakan API ke sistem pengelolaan UMK untuk diproses lebih lanjut. Pada sistem pengelolaan UMK data item pembelanjaan akan dikelompokkan sesuai kategori dan dicocokkan dengan item yang dilaporkan pada laporan keuangan. Setiap item yang persentase kemiripan dengan item pada laporan keuangan lebih besar dari 80% diberi tanda sesuai sedangkan item yang kurang dari 80% akan diberi tanda tidak sesuai.



Gambar 1. Tahapan Penelitian

Tahapan pengumpulan kebutuhan dilakukan dengan melakukan observasi proses pelaporan UMK yang dilakukan oleh pegawai dan proses pemeriksaan UMK yang dilakukan oleh staf keuangan. Dari hasil observasi dilakukan analisis proses-proses yang dapat dibantu oleh sistem. Hasil pengumpulan kebutuhan diformalkan dalam bentuk *activity diagram* (Koç et al., 2021) dan daftar kebutuhan. Selanjutnya, berdasarkan daftar kebutuhan tersebut dilakukan desain yang

mencakup desain Application Programming Interface (API) (M et al., 2020) (Neumann et al., 2021) dan desain arsitektur perangkat lunak yang dinyatakan menggunakan *component* diagram (Koç et al., 2021).

Pada tahapan pengembangan, akan dibangun API untuk komunikasi antara aplikasi pelaporan UMK yang sudah ada saat ini dengan aplikasi yang mengolah kwitansi pembayaran menjadi teks. Aplikasi pengolahan kwitansi akan dikembangkan menggunakan bahasa pemrograman PHP. Pemeriksaan kemiripan antara teks yang dihasilkan dari pembacaan bukti pembayaran dengan item-item pengeluaran pada laporan pertanggung jawaban (LPJ) akan dihitung menggunakan algoritma Levenshtein seperti yang ditunjukkan pada persamaan 1 di mana $tail(x)$ adalah string dari semua kecuali karakter pertama dari x . Jika teks yang dibandingkan sama, nilai levenshtein *distance*-nya nol. Semakin besar nilai levenshtein *distance* menunjukkan kedua teks yang dibandingkan semakin berbeda. Karena nilai levenshtein *distance* bergantung pada panjangnya string yang dibandingkan, untuk menghitung persentasi kemiripan penelitian ini akan menggunakan persamaan 2. Terakhir pada tahapan pengujian akan dilakukan pengujian fungsional yang mencakup kemampuan aplikasi untuk bertukar data dengan aplikasi-aplikasi eksternal dan menguji kemampuan aplikasi untuk mencocokkan teks hasil pembacaan aplikasi OCR dengan item pengeluaran pada LPJ.

$$lev(a, b) = \begin{cases} |a|, & \text{jika } |b| = 0 \\ |b|, & \text{jika } |a| = 0 \\ lev(tail(a), tail(b)), & \text{jika } a[0] = b[0] \\ 1 + \min \begin{cases} lev(tail(a), b) \\ lev(a, tail(b)) \\ lev(tail(a), tail(b)) \end{cases}, & \text{lainnya} \end{cases} \dots\dots\dots(1)$$

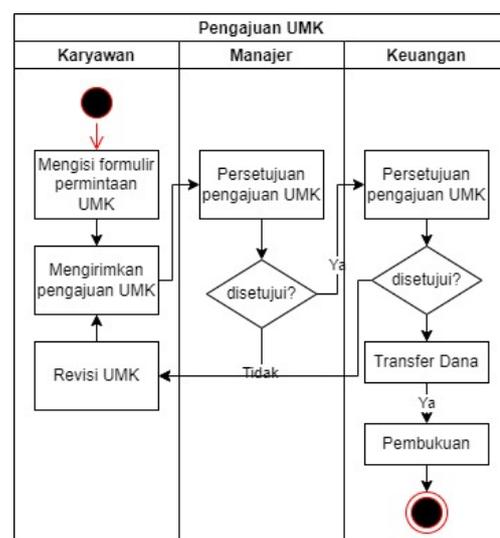
$$sim = \frac{\max(|a|, |b|) - lev(a, b)}{\max(|a|, |b|)} * 100\% \dots\dots\dots(2)$$

3. Hasil dan Pembahasan

Dari wawancara dengan pegawai dan pihak keuangan serta analisis dokumen prosedur operasional standar perusahaan, didapatkan alur proses pengajuan permintaan dan LPJ UMK yang digambarkan pada *activity diagram* pada Gambar 2 dan Gambar 3. Seluruh aktivitas pada proses pengajuan permintaan UMK dilakukan menggunakan aplikasi UMK daring yang berbasis web. Begitupula dengan sebagian besar aktivitas pelaporan LPJ UMK dilakukan pada aplikasi UMK

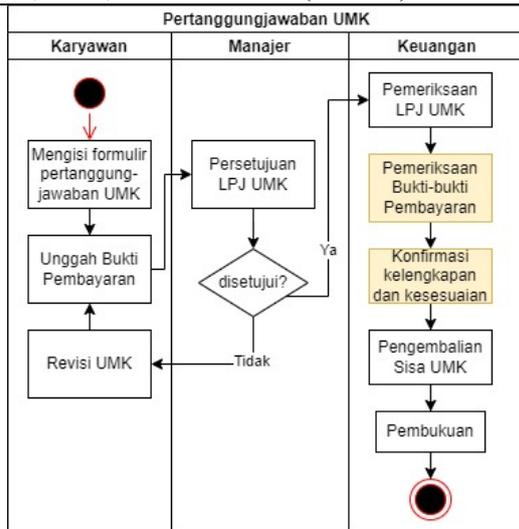
daring. Ada dua aktivitas yang dilakukan secara manual oleh staf keuangan, yakni pemeriksaan kelengkapan dan kesesuaian bukti pembayaran dengan item pengeluaran pada LPJ dan upaya tindak lanjut dari staf keuangan untuk meminta dokumen yang tidak lengkap atau mengkonfirmasi perbedaan nominal belanja.

Dalam melakukan proses pemeriksaan bukti pengeluaran, staf keuangan akan memeriksa hal-hal sebagai berikut; apakah bukti pengeluaran diunggah oleh karyawan, apakah nominal pengeluaran yang dituliskan pada laporan sesuai dengan bukti, apakah item pengeluaran berada pada kategori yang sesuai, apakah nominal pengeluaran sesuai dengan pengajuan (tidak melebihi batasan yang ditentukan perusahaan). Hanya pengeluaran yang memenuhi kriteria tersebut yang disetujui. Staf keuangan juga akan meminta bukti pembayaran yang kurang dan akan mengkonfirmasi item-item yang tidak disetujui kepada karyawan melalui email. Setelah mendapatkan konfirmasi dari karyawan yang bersangkutan, staf keuangan akan mengubah nominal untuk item-item yang tidak sesuai tersebut. Jika ada selisih jumlah antara UMK dan laporan, maka karyawan harus mengembalikan kelebihan UMK kepada perusahaan. Penelitian ini akan mencoba melakukan pemeriksaan kesesuaian bukti pengeluaran dengan item pengeluaran yang diisikan pada LPJ UMK yang diisi secara daring. Setiap item pengeluaran yang memiliki bukti pengeluaran dan jumlahnya sesuai akan ditandai dengan sesuai. Sebaliknya item pengeluaran yang tidak memenuhi salah satu kriteria yang ditentukan akan ditandai dengan tidak sesuai.



Gambar 2. Alur Proses Pengajuan UMK





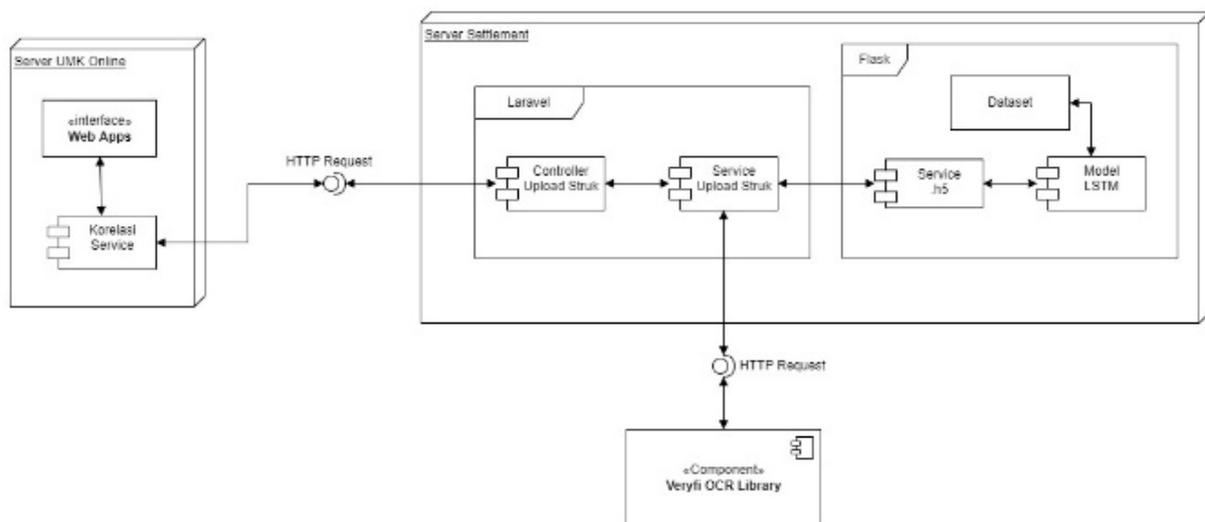
Gambar 3. Alur Proses Pelaporan LPJ UMK

Agar tidak mengganggu proses bisnis perusahaan yang berjalan menggunakan aplikasi UMK daring, dalam penelitian ini dikembangkan aplikasi terpisah untuk pemeriksaan bukti pengeluaran. *Requirement* untuk aplikasi pemeriksaan bukti pengeluaran dijelaskan pada Tabel 1.

Aplikasi pemeriksaan bukti pengeluaran akan berkomunikasi dengan aplikasi UMK daring yang sudah ada saat ini menggunakan API berbasis REST, komunikasi antara berbagai komponen yang terlibat ditunjukkan dalam *component diagram* pada Gambar 4. Aplikasi akan mengakses data

laporan pertanggung jawaban beserta bukti-bukti pembayaran dari aplikasi UMK daring. Setiap file bukti pembayaran akan dikirimkan satu per satu ke aplikasi OCR kemudian hasil pembacaan gambar bukti pembayaran akan diakses dalam format JSON.

Untuk membaca teks dari bukti pembayaran yang berupa teks, penelitian ini menggunakan aplikasi Verryfi¹ yang menerima gambar bukti pembayaran misalnya kwitansi, nota, *invoice* baik dalam bentuk tulisan tangan maupun teks yang dicetak seperti yang ditunjukkan pada Gambar 6. Aplikasi ini akan menghasilkan pembacaan teks per baris dalam bentuk JSON seperti pada Gambar 7. File JSON berisi semua informasi teks pada struk seperti nama toko, alamat dan nomor telepon toko, tanggal dan waktu transaksi, item transaksi, harga tiap item, total transaksi, pajak, diskon, total pembayaran, serta informasi lainnya. Pada penelitian ini, teks yang akan diproses hanyalah teks item. Untuk mengambil teks item pembelian dan harganya, dilakukan pemisahan dengan mengambil nilai *attribute* item. Apabila nilai *attribute* tersebut berbentuk array, pengambilan data dilakukan secara berulang untuk setiap elemen dalam array. Potongan kode program untuk melakukan tahapan pemilihan ini ditunjukkan pada Gambar 8 dan hasil pemilihan item ditunjukkan pada Gambar 9.



Gambar 4. Desain Arsitektur Perangkat Lunak

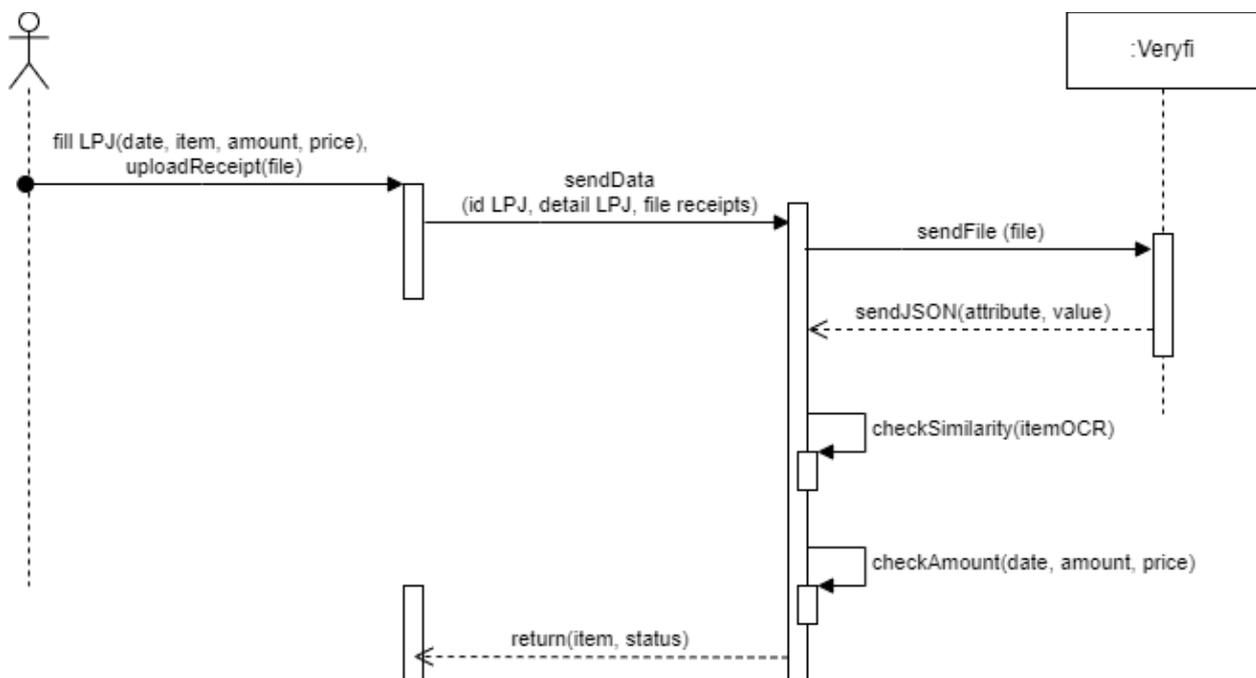
¹ <https://www.verryfi.com/receipt-ocr-api/>



Table 1. Daftar Kebutuhan Pengguna

No.	Requirement
1.	Aplikasi mampu mengambil data UMK dan item-item pengeluarannya dari aplikasi yang sudah ada saat ini.
2.	Aplikasi mampu mengakses dan mengambil file-file bukti pengeluaran untuk suatu UMK dalam format JPG, JPEG, PNG, PDF.
3.	Aplikasi mampu mengidentifikasi suatu file bukti pengeluaran merupakan bagian dari suatu LPJ UMK tertentu.
4.	Waktu maksimum yang dibutuhkan aplikasi untuk bertukar informasi dengan aplikasi eksternal adalah 1 menit untuk ukuran file hingga 100 MB dengan kecepatan internet 256 Kbps

5. Aplikasi mampu mengirimkan sebuah file bukti pembayaran dalam format JPG, JPEG, PNG ke aplikasi OCR eksternal.
6. Aplikasi mampu menerima hasil pembacaan OCR dari aplikasi eksternal dalam format JSON.
7. Aplikasi mampu menyimpan informasi dari format JSON dengan tepat.
8. Aplikasi mampu menentukan apakah sebuah item pengeluaran pada LPJ sesuai dengan sebuah item pembayaran pada teks hasil pembacaan bukti pembayaran.
9. Dua buah teks dinyatakan sama jika kemiripannya lebih dari 80%.



Gambar 5. Activity Diagram - Interaksi Antar Objek dalam Perangkat Lunak

Pada tahapan pengujian dilakukan pengujian terhadap semua kebutuhan yang ditentukan pada Tabel 1. Sebuah contoh resi pembayaran seperti yang ditunjukkan pada Gambar 6 diunggah ke aplikasi pelaporan UMK. Aplikasi ini berhasil mengirimkan data struk ke Veryfi dan menerima respon dari Veryfi dalam format file JSON seperti yang ditunjukkan pada Gambar 7.

Pengujian dilakukan menggunakan sembilan buah bukti pembayaran berbeda dengan format pdf, JPEG, dan PNG. Masing-masing bukti pembayaran terdiri dari tujuh hingga dua puluh item pembelanjaan dengan layout *portrait* dan *landscape*. Seluruh file berhasil dikirimkan ke aplikasi eksternal Veryfi dengan waktu pertukaran

data di bawah satu menit. Veryfi mampu membaca seluruh informasi dalam bukti pembayaran dengan benar, namun pada beberapa kwitansi yang memiliki kode barang, kode barang sering kali dianggap sebagai jumlah barang yang dibeli. Aplikasi juga dapat menerima dan membaca file JSON dari Veryfi dengan baik dan mampu memisahkan informasi tanggal transaksi, nama item pembelian, jumlah pembelian, harga dengan baik. Perhitungan kemiripan yang dilakukan sudah sesuai dan mampu mencocokkan item-item yang sesuai dengan tanggal, jumlah dan harga beli. Di sisi lain, jika dalam sebuah bukti pembayaran terdapat diskon atau pajak, aplikasi saat ini belum mampu mengolah data pajak dan diskon. Pada



kasus seperti ini, data laporan yang itemnya sesuai ditolak karena jumlah pembayarannya yang berbeda.

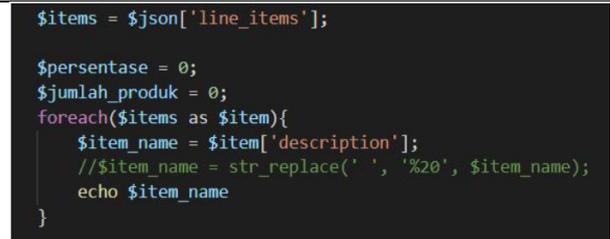
Hasil pemeriksaan manual terhadap item pembayaran yang ditandai tidak sesuai menunjukkan bahwa item-item yang kemiripannya di atas 65% memiliki kecenderungan sama dengan item yang dituliskan pada LPJ. 98% item yang kemiripannya 65-80% dan tanggal transaksinya sama merupakan item yang diterima.



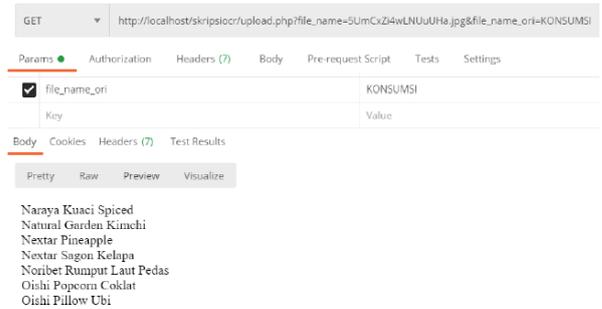
Gambar 6. Contoh Bukti Pembayaran



Gambar 7. Potongan File JSON Hasil Pembacaan Bukti Pembayaran



Gambar 8. Potongan Source Code untuk Seleksi Data



Gambar 9. Hasil Pemilihan Data dari File JSON

4. Kesimpulan

Pengembangan aplikasi pemeriksaan laporan pertanggung jawaban penggunaan uang muka kerja bertujuan menggantikan proses pemeriksaan yang umumnya dilakukan secara manual oleh staf keuangan yang cukup menyita waktu. Aplikasi yang dibangun memanfaatkan teknologi *optical character recognition* untuk membaca tulisan yang ada pada bukti pembayaran dan *text similarity* untuk mengukur kesesuaian setiap item yang dibaca dari bukti pembayaran dengan item pengeluaran yang ada pada laporan pertanggungjawaban. Teknologi *optical character recognition* menggunakan aplikasi eksternal bernama Veryfi sehingga dibutuhkan *application programming interface* untuk komunikasi antar aplikasi. Aplikasi yang dikembangkan mampu mengirimkan file bukti pembayaran dalam format gambar dan mendapatkan hasil pembacaan berbentuk teks dalam format file JSON kemudian mengukur kesamaan teks menggunakan algoritma *levenshtein distance*. Hasil pengujian menunjukkan seluruh kebutuhan pengguna berhasil dipenuhi namun performa pembacaan *optical character recognition* untuk item pembelian yang memiliki karakter angka masih perlu ditingkatkan.

5. Saran

Penambahan kapabilitas perhitungan diskon dan pajak perlu dirancang dan dikembangkan agar proses pemeriksaan LPJ UMK lebih sesuai dengan

keadaan sebenarnya. Selain itu, perlu dilakukan penyesuaian persentase kemiripan yang dianggap sesuai dan tidak sesuai berdasarkan hasil pemeriksaan manual yang dilakukan.

Referensi

- Benedikt, L., Joshi, C., Nolan, L., De Wolf, N., & Schouten, B. (2020). *Optical Character Recognition and Machine Learning Classification of Shopping Receipts*.
- Cahyo, N. D. W. I. (2019). Pengenalan Nomor Plat Kendaraan Dengan Metode Optical Character Recognition. *Ubiquitous: Computers and Its Applications Journal*, 2, 75–84. <https://doi.org/10.51804/ucaiaj.v2i1.75-84>
- Doolittle, J. (2020). Jeremy Miller on Waterfall Versus Agile. *IEEE Software*, 37(4), 107–109. <https://doi.org/10.1109/MS.2020.2987493>
- Du, Y., Li, C., Guo, R., Yin, X., Liu, W., Zhou, J., Bai, Y., Yu, Z., Yang, Y., Dang, Q., & Wang, H. (2020). *PP-OCR: A Practical Ultra Lightweight OCR System*. <http://arxiv.org/abs/2009.09941>
- Hanif, A. R., Nasrullah, E., & Setyawan, F. X. A. (2023). Deteksi Karakter Plat Nomor Kendaraan Dengan Menggunakan Metode Optical Character Recognition (Ocr). *Jurnal Informatika Dan Teknik Elektro Terapan*, 11(1), 109–117. <https://doi.org/10.23960/jitet.v11i1.2897>
- John, S., Mebawondu, J. O., Olajide, A. O., & Josephine, M. O. (2021). Design of Cash Advance Payment System in a Developing Country: A Case Study of First Bank of Nigeria Mortgages Limited. In S. Misra & B. Muhammad-Bello (Eds.), *Information and Communication Technology and Applications* (pp. 703–714). Springer International Publishing.
- Kata, S., Pabboju, S., Babu, V., & Medishetti, A. (2018). Snap and split: an android application for bill payment using tesseract OCR. *International Journal of Engineering & Technology*, 7(4.5), 634. <https://doi.org/10.14419/ijet.v7i4.5.21175>
- Kim, G., Hong, T., Yim, M., Nam, J., Park, J., Yim, J., Hwang, W., Yun, S., Han, D., & Park, S. (2022). OCR-Free Document Understanding Transformer. In S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, & T. Hassner (Eds.), *Computer Vision -- ECCV 2022* (pp. 498–517). Springer Nature Switzerland.
- Koç, H., Erdoğan, A. M., Barjakly, Y., & Peker, S. (2021). *UML Diagrams in Software Engineering Research: A Systematic Literature Review*. 13. <https://doi.org/10.3390/proceedings2021074013>
- Kodmelwar, M. K., Futane, P. R., Pawar, S. D., Lokhande, S. A., Dhanure, S. P., & Professor, A. (2022). A Comparative Study of Software Development Waterfall, Spiral and Agile Methodology. *Journal of Positive School Psychology*, 2022(3), 7013–7017. <http://journalppw.com>
- Kumar, V., Kaware, P., Singh, P., Sonkusare, R., & Kumar, S. (2020). Extraction of information from bill receipts using optical character recognition. *Proceedings - International Conference on Smart Electronics and Communication, ICOSEC 2020, Icosec*, 72–77. <https://doi.org/10.1109/ICOSEC49089.2020.9215246>
- Park, S., & Lee, H. (2019). *CORD: A Consolidated Receipt Dataset for Post-OCR Parsing. I*, 1–4.
- Pressman, R. S., & Maxim, B. (2014). *Software Engineering: A Practitioner's Approach 8th Edition*. McGraw-Hill Education.
- Sommerville, I. (2016). *Software Engineering Tenth Edition*. Pearson.
- Stefan, A. C. (2020). ACCOUNTING PRACTICES REGARDING THE MEANS OF OPERATION WITH CASH. *EUROPEAN JOURNAL OF ACCOUNTING, FINANCE & BUSINESS, XIII(XXIII)*.
- Susanty, M., & Nugroho, H. (2020). Optical Character Recognition Implementation for Admission System in Universitas Pertamina. *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, 11(1), 165–170. <https://doi.org/10.24176/simet.v11i1.3838>
- Tirtana, E., Gunadi, K., & Sugiarto, I. (2021). Penerapan Metode YOLO dan Tesseract-OCR untuk Pendataan Plat Nomor Kendaraan Bermotor Umum di Indonesia Menggunakan Raspberry Pi. *Jurnal Infra*, 9(2), 241–247. <https://publication.petra.ac.id/index.php/teknik-informatika/article/view/11454>

