**PENERAPAN LIBARARY TESSERACT OPTICAL CHARACTER RECOGNITION DALAM FITUR REIMBURSEMENT MENGGUNAKAN METODE KANBAN**

**(STUDI KASUS PT. Kayana Nusa Teknologi)**

SKRIPSI

**Logo

Description automatically generated**

**OLEH:**

**ABDUL HAKIM**

**171021400238**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PAMULANG**

**PAMULANG**

**2023**

**PENERAPAN LIBARARY TESSERACT OPTICAL CHARACTER RECOGNITION DALAM FITUR REIMBURSEMENT MENGGUNAKAN METODE KANBAN**

**(STUDI KASUS PT. Kayana Nusa Teknologi)**

SKRIPSI

Diajukan Untuk Melengkapi Salah Satu Syarat Memperoleh Gelar Sarjana Komputer

**Logo

Description automatically generated**

**OLEH:**

**ABDUL HAKIM**

**171021400238**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PAMULANG**

**PAMULANG**

**2023**

# LEMBAR PERNYATAAN

Yang bertanda tangan di bawah ini

Nim : 171021400238

Nama : ABDUL HAKIM

Program Studi : TEKNIK INFORMATIKA

Fakultas : ILMU KOMPUTER

Jenjang Pendidikan : STRATA 1

Menyatakan bahwa skripsi yang saya buat dengan judul:

**PENERAPAN LIBARARY TESSERACT OPTICAL CHARACTER RECOGNITION DALAM FITUR REIMBURSEMENT** **MENGGUNAKAN METODE KANBAN (STUDI KASUS PT. Kayana Nusa Teknologi)**

1. Merupakan hasil karya tulis ilmiah sendiri, bukan merupakan karya yang pernah diajukan untuk memperoleh gelar akademik oleh pihak lain, dan bukan merupakan hasil plagiat.
2. Saya ijinkan untuk dikelola oleh Universitas Pamulang sesuai dengan norma hukum dan etika yang berlaku.

Pernyataan ini saya buat dengan penuh tanggung jawab dan saya bersedia menerima konsekuensi apapun sesuai aturan yang berlaku apabila di kemudian hari pernyataan ini tidak benar.

Pamulang, 10 Januari 2023

Abdul Hakim

# LEMBAR PERSETUJUAN

Nim : 171021400238

Nama : ABDUL HAKIM

Program Studi : TEKNIK INFORMATIKA

Fakultas : ILMU KOMPUTER

Jenjang Pendidikan : STRATA 1

Judul Skripsi : PENERAPAN LIBARARY TESSERACT OPTICAL CHARACTER RECOGNITION DALAM FITUR REIMBURSEMENT MENGGUNAKAN METODE KANBAN (STUDI KASUS PT. Kayana Nusa Teknologi)

Skripsi ini telah diperiksa dan di setjui oleh pembimbing untuk persyaratan sidang skripsi.

Pamulang, 10 Januari 2023

Pembimbing

Samsoni, S.Kom., M.Kom

NIDN : 0431127505

Mengetahui,

Ketua Program Studi Teknik Informatika

Achmad Udin Zailani, S.Kom., M.Kom.

NIDN: 0429058303

# LEMBAR PENGESAHAN

Nim : 171021400238

Nama : ABDUL HAKIM

Program Studi : TEKNIK INFORMATIKA

Fakultas : ILMU KOMPUTER

Jenjang Pendidikan : STRATA 1

Judul Skripsi : PENERAPAN LIBARARY TESSERACT OPTICAL CHARACTER RECOGNITION DALAM FITUR REIMBURSEMENT MENGGUNAKAN METODE KANBAN (STUDI KASUS PT. Kayana Nusa Teknologi)

Skripsi ini telah dipertahankan di hadapan dewan penguji ujian skripsi fakultas Teknik, program studi Teknik Informatika dan dinyatakan LULUS.

Pamulang, 10 Januari 2023

|  |  |
| --- | --- |
| Penguji I | Penguji II |
| NIDN: | NIDN: |

Pembimbing

Samsoni, S.Kom., M.Kom

NIDN : 0431127505

Mengetahui,

Ketua Program Studi Teknik Informatika

Achmad Udin Zailani, S.Kom., M.Kom.

NIDN: 0429058303

# KATA PENGANTAR

Puji syukur peneliti panjatkan kehadirat Allah Yang Maha Kuasa atas segala rahmat dan karunia-Nya, peneliti dapat menyelesailan laporan skripsi yang merupakan salah satu persyaratan untuk menyelesaikan program studi strata satu (S1) pada program studi Teknik Informatika di Universitas Pamulang yang berjudul Dengan segala keterbatasan, penulis menyadari pula bahwa skripsi dengan judul “PENERAPAN LIBARARY TESSERACT OPTICAL CHARACTER RECOGNITION DALAM FITUR REIMBURSEMENT MENGGUNAKAN METODE KANBAN (STUDI KASUS PT. Kayana Nusa Teknologi)”Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Hal ini disebabkan karena keterbatasan pengetahuan yang penulis miliki. Meskipun demikian penulis berusaha semaksimal mungkin agar dalam pembuatan skripsi ini dapat terselesaikan sebaik-baiknya sehingga dapat bermanfaat bagi pembaca dan khususnya rekan mahasiswa/mahasiswi Universitas Pamulang.

Bersama ini pula dengan segala kerendahan hati, peneliti menghaturkan banyak terima kasih kepada:

1. Allah Subhanahu wa ta'ala yang telah memberikan begitu banyak nikmat dan karunia diantaranya imam islam serta sehat dan umur panjang sehingga penulis dapat menyelesaikan penulisan skripsi ini.
2. Bapak Drs. (H.C). H. Darsono selaku Ketua Yayasan Sasmita Jaya.
3. Bapak Dr. H. E. Nurzaman, AM., M.M., M.Si selaku Rektor Universitas Pamulang.
4. Bapak Syaiful Bakhri, ST., M.Eng. Sc., Ph.D selaku Dekan Fakultas Teknik Universitas Pamulang.
5. Bapak Achmad Udin Zailani, S.Kom., M.Kom., selaku ketua program studi Teknik Informatika di Universitas Pamulang.
6. Bapak Samsoni, S.Kom., M.Kom., selaku pembimbing skripsi pada program studi Teknik Informatika di Universitas Pamulang yang telah banyak memberikan saran dan ilmu dalam penyusunan skripsi ini.
7. Seluruh Dosen pengajar Strata satu (S1) pada program studi Teknik Informatika di Universitas Pamulang yang telah memberikan ilmunya.
8. Ibu, bapak, dan Adik tercinta yang tiada henti selalu memberikan doa dan dorongan kepada penulis baik secara moral maupun materil sehingga skripsi ini dapat diselesaikan dengan baik.
9. Pihak PT. Kayana Nusa Teknologi yang telah mengijinkan dan memberi kesempatan saya untuk melakukan riset perusahaan tersebut.
10. Teman-teman yang telah memberikan dukungan moral untuk terus menyelesaikan skripsi ini.

Akhir kata mudah-mudahan skripsi ini dapat bermanfaat khususnya bagi penulis dan juga pembaca. Semoga kita senantiasa diberikan keberkahan dan selalu dalam lindungan Allah SWT sehingga dapat terus mencari ilmu dan bermanfaat bagi orang lain.

Pamulang, 10 Januari 2023

Abdul Hakim

# ABSTRACT

PT. Kayana Nusa Teknologi is a technology development company. Currently, the company is still manually processing reimbursements, which is deemed to be less effective for a company. Innovation is needed to streamline this process and maximize the use of time to improve work processes within the company. In designing this application, the Kanban method is used, which provides a sequential or ordered approach to the software lifecycle, starting with analysis, design, coding, testing, and supporting stages. In addition, the use of Optical Character Recognition (OCR) technology is used, which can identify and convert text in image documents into text that can be read by computers. With the existence of this application, it is expected that the performance of employees will be more efficient, especially for Android users in the company.

**Keywords**: Application, Design, Employee, Optical Character Recognition, Android

# ABSTRAK

PT. Kayana Nusa Teknologi adalah sebuah perusahaan yang bergerak di bidang pengembangan teknologi. Saat ini perusahaan tersebut masih melakukan proses reimbursement secara manual, sehingga dirasa kurang efektif untuk suatu perusahaan. Diperlukan inovasi untuk memangkas proses tersebut dan memaksimalkan pemanfaatan waktu yang ada untuk meningkatkan proses kerja dalam perusahaan. Dalam perancangan aplikasi ini menggunakan metode Kanban yang menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut, dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung. Ditambah dengan penggunaan teknologi Optical Character Recognition (OCR) yang dapat mengidentifikasi dan mengkonversi teks dalam dokumen gambar menjadi teks yang dapat dibaca oleh komputer. Dengan adanya aplikasi ini diharapkan kinerja pegawai dapat lebih efisien khususnya untuk pengguna Android di perusahaan tersebut.

**Kata Kunci** : Aplikasi, Perancangan, Pegawai, Optical Character Recognition, Android

**DAFTAR ISI**

# DAFTAR GAMBAR

# DAFTAR TABEL

# BAB I

# PENDAHULUAN

## 1.1 Latar Belakang

Dalam dunia professional, kita bekerja tidak terlepas dengan penggunaan dokumen, seperti membuat dokumen, mencetak dokumen maupun menyimpan dokumen baik dalam bentuk digital maupun dalam bentuk fisik. Termasuk dalam dunia perkantoran, dokumen menjadi sangat penting dalam kegiatan operasional sehari-hari seperti pada dokumen *reimbursement*.

*Reimburse* disini adalah kompensasi yang dilakukan PT. Kayana Nusa Teknologi untuk pengeluaran yang dikeluarkan oleh karyawan dengan menggunakan uang pribadinya. Hal ini merupakan salah satu benefit yang menjadi hak utama karyawan. Dimana ketika ada suatu keperluan yang menyangkut kepentingan perusahaan dan harus menggunakan uang pribadi terlebih dahulu, kamu nantinya dapat mengajukan reimbursement.

Namun pada saat ini proses *reimbursement* pada PT. Kayana Nusa Teknologi dilakukan secara input struk manual dan dapat menghabiskan banyak waktu yang digunakan untuk melakukan satu alur proses *reimbursement* tersebut. Sehingga tidak terlalu efisien dalam operasional sistem yang ada.

Berdasarkan masalah tersebut, perlu adanya penerapan sistem dengan mengandalkan salah satu teknik dari Pengolahan Citra Digital yaitu *Optical Character Recognition* dan penerapannya harus di sesuaikan dengan kebutuhan saat ini dan masa yang akan datang. Dengan itu peneliti menggunakan judul **“PENERAPAN LIBARARY TESSERACT OPTICAL CHARACTER RECOGNITION DALAM FITUR REIMBURSEMENT MENGGUNAKAN METODE KANBAN (STUDI KASUS PT. Kayana Nusa Teknologi)”.** Sistem ini di bangun dengan menggunakan Bahasa pemrograman Javascript dan menggunakan kerangka untuk mengembangkan aplikasi yaitu *Strapi* serta menggunakan PostgreSQL sebagai DBMS (*Database Management System*).

## 1.2 Identifikasi Masalah

Berdasarkan latar belakang diatas, peneliti mengidentifikasi masalah yang ada sebagai berikut :

1. Penggunaan dokumen reimbursement pada PT. Kayana Nusa Teknologi saat ini masih menggunakan dokumen fisik sehingga dokumen dapat hilang dan rusak.
2. Karyawan PT. Kayana Nusa Teknologi saat ini melakukan input transaksi masih secara manual dan memakan waktu yang cukup lama dalam pembuatan dokumen rimbursement sehingga mengurangi efisiensi dalam operasional pada perusahaan tersebut.

## 1.3 Rumusan Masalah

Berdasarkan dari identifikasi masalah yang telah dikemukakan di atas, maka rumusan masalahnya sebgai berikut:

1. Bagaimana format dokumen *reimbursement* yang benar pada umumnya.
2. Bagaimana proses *reimbursement* yang sudah ada di PT. Kayana Nusa Teknologi.
3. Bagaimana cara memberikan inovasi dengan penerapan alat teknologi terhadap PT. Kayana Nusa Teknologi.

## 1.4 Batasan Penelitian

Batasan masalah pada penelitian ini adalah :

1. Pengguna android dengan minimal versi *Lolipop* 5.0 yang dapat menggunakan aplikasi ini.
2. Hasil identifikasi jumlah total belanja pada struk bergantung pada kualitas gambar yang di ambil*.*
3. Sistem aplikasi mencakup *user role* admin dan karyawan.
4. *Database design* menggunakan *Strapi* yang dimana sudah disediakan *database* *guideline* sendiri sesuai arahan *Strapi.*

## 1.5 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut :

1. Dapat membantu menyelesaikan situasi penggunaan dokumen *reimburse* fisik serta menjadikannya menjadi format digital dokumen tersebut.
2. Meningkatkan efisiensi waktu dalam penulisan manual dokumen reimbursement dengan menciptakan aplikasi khusus untuk proses tersebut.

## 1.6 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut :

1. Manfaat bagi Universitas Pamulang
2. Sebagai bahan pertimbangan dan panduan bagi mahasiswa yang akan menyusun skripsi dimasa yang akan datang dan menambah pengetahuan dibidang android.
3. Dapat menambah sumber referensi bagi peneliti yang akan melakukan penelitian tentang penerapan Optical Character Recognition.
4. Manfaat bagi penulis
5. Dengan mengimplementasikan disiplin ilmu yang di peroleh selama kuliah serta memenuhi syarat kelulusan mendapatkan gelar S-1 pada jurusan teknik informatika Universitas Pamulang.
6. Dapat menambah wawasan dan pengalaman dalam hal pembuatan aplikasi android yang bermanfaat.
7. Menambah wawasan tentang hal-hal baru yang selama ini belum diketahui oleh peneliti.
8. Manfaat bagi instansi
9. Dengan adanya penerapan *Optical Character Recognition* dalam fitur sistem *reimbursement* dapat mempermudah karyawan PT. Kayana Nusa Teknologi dalam mengajukan *reimburse* tanpa harus menulis jumlah total belanja dalam struk secara manual.
10. Mengurangi penggunaan dokumen fisik dalam proses *reimbursement.*
11. Memudahkan staff administrasi PT. Kayana Nusa Teknologi dalam mengolah data *reimbursement* tanpa harus melihat dokumen fisik yang memakan tempat dalam ruangan.

## 1.7 Metodologi Penelitian

Metode pengumpulan data untuk mengumpulkan data dari sample penelitian dilakukan dengan metode tertentu sesuai dengan tujuannya. Teknik- teknik yang digunakan dalam pengumpulan adalah sebagai berikut:

## 1.7.1 Metode Pengumpulan Data

1. Wawancara

Mengadakan tanya jawab dengan karyawan PT. Kayana Nusa Teknologi untuk memperoleh gambaran, keterangan dan penjelasan untuk membantu bahan dalam penelitian dan penulisan skripsi.

1. Pengamatan (Observasi)

Pada teknis ini penelitian langsung ke lokasi PT. Kayana Nusa Teknologi untuk mengetahui dan mengamati bagaimana format *reimbursement* yang sudah berjalan.

1. Studi Pusaka

Pengumpulan data dan informasi dengan cara membaca buku, jurnal dan skripsi sejenis maupun website atau media elektronik lainnya yang menjadi referensi pembahasan dalam masalah ini.

## 1.7.2 Metode Pengembangan Sistem

Berdasarkan permasalahan di atas maka peneliti akan membuat sebuah aplikasi penerapan *Optical Character Recognition* berbasis android dengan menggunakan metode *Kanban. Kanban* adalah suatu alat untuk mengendalikan produksi yang digunakan dalam mengendalikan aliran-aliran material melalui sistem produksi dengan menggunakan kartu-kartu untuk memerintahkan suatu work center memindahkan dan menghasilkan material atau komponen tertentu.

## 1.7.3 Metode Pengujian

Black Box testing merupakan pengujian yang dilakukan hanya dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak.

## 1.8 Sistematika Penulisan

Untuk memahami lebih jelas tentang laporan ini, maka penulis menguraikan secara singkat mengenai struktur penulisan pada masing-masing bab dengan sistematika penulisan sebagai berikut:

**BAB I PENDAHULUAN**

Pada bab ini membahas tentang latar belakang, identifikasi masalah, rumusan masalah, batasan penelitian, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

**BAB II LANDASAN TEORI**

Dalam bab ini berisi tentang teori yang berupa pengertian dan definisi yang diambil dari kutipan buku ataupun jurnal yang berkaitan dengan penyusunan laporan skripsi.

**BAB III ANALISA DAN PERANCANGAN**

Bab ini berisi tentang perancangan system serta komponen- komponen pemodelan system yang digunakan.

**BAB IV IMPLEMENTASI DAN PENGUJIAN**

Dalam bab ini menjelaskan tentang implementasi dan pengujian dari aplikasi yang telah dibuat.

**BAB V PENUTUP**

Pada bab ini menguraikan tentang penutupan Tugas Akhir yang didalamnya terdapat kesimpulan dari pembahasan laporan penelitian dan saran yang diharapkan dapat bermanfaat untuk penulis, maupun pengembangan program aplikasi selanjutnya.

# BAB II

# TINJAUAN PUSTAKA

## Penelitian Terkait

Penelitian sebelumnya menjadi salah satu sumber referensi dalam penelitian ini untuk membahas topik tertentu. Beberapa jurnal yang relevan dengan topik tersebut antara lain sebagai berikut:

1. Referensi pertama

Judul : Penerapan Convolutional Neural Network (CNN) pada Pengenalan Aksara Lampung Berbasis Optical Character Recognition (OCR)

Penulis : (Agus Mulyanto & Erlina Susanti, 2021)

Provinsi Lampung memiliki bahasa dan aksara daerah yang disebut juga dengan Had Lampung atau KaGaNga yang merupakan aksara asli lampung. Melihat bagaimana pentingnya nilai akan eksistensi sebuah budaya dan pentingnya pelestarian aksara lampung maka dibutuhkan teknologi yang membantu dalam mengenalkan aksara lampung, salah satunya dengan teknologi *optical character recognition (*OCR*)* yang digunakan untuk merubah citra kedalam teks. Untuk mengenali pola citra Aksara Lampung dan klasifikasi model maka digunakan *Convolutional Neural Network* (CNN). CNN memiliki lapisan convolution yang terbentuk dari beberapa gabungan lapisan konvolusi, lapisan pooling dan lapisan fully connected. Pada peneilitian yang dilakukan dataset dikembangkan dengan pengumpulan hasil tulis tangan dari sampel responden yang telah ditentukan, kemudian dilakukan scanning gambar. Selanjutnya, dilakukan proses pelabelan dan disimpan dengan format yaitu TXT. Dari asitektur CNN yang dibangun berdasarkan hasil evaluasi menunjukan loss, accuracy menghasilkan nilai training accuracy mendapatkan nilai sebesar 0.57 dan precision mendapatkan nilai sebesar 0.87. Dari hasil nilai accuracy dan precision menunjukkan bahwa model training sudah baik.

1. Referensi kedua

Judul : PENERAPAN METODE EKSTRASI IMAGE KE TEXT DENGAN OPTICAL CHARACTER RECOGNITION UNTUK OTOMATISASI DATA KEPENDUDUKAN

Penulis : (M. Makmun Effendi & Ismasari Nawangsih, 2019)

Pemerintah sudah membuat suatu sistem informasi kependudukan menggunakan electronic Kartu Tanda Penduduk yang lebih dikenal e-KTP yang menggunakan teknologi digital dan diharapakan dengan adanya e-KTP dapat digunakan diberbagai keperluan. Menurut (Ashari,Ahmad., Sutanta,Edhy : 2012) pada e-KTP memiliki chip yang digunakan sebagai smart card dan memuat 4 sampai dengan 8 KB, yang memuat data NIK, nama, tempat dan tanggal lahir, jenis kelamin, agama, status perkawinan, golongan darah, alamat, pekerjaan, kewarganegaraan, foto, masa berlaku, tempat dan tanggal dikeluarkan, tandatangan, serta nama dan nomor induk pegawai pejabat yang menandatanganinya dan ada tanda tangan pemilik e- KTP pada bagian depan.

Dengan latar belakang tersebut maka peneliti akan melakukan penelitian dengan mengangkat judul “ Penerapan Metode Ekstraksi Image Ke Text dengan *Optical Character Recognition* Untuk Otomatisasi Data Kependudukan”. Dengan bantuan mesin scanner maka pengambilan e-KTP dapat dilakukan dan disimpan pada komputer, dan ketika hendak mengambil data yang ada dalam e-KTP pengambilan gambar dapat dilakukan Untuk mengambil data e-KTP agar dapat diinput dalam database diperlukan *Tesseract Optical Character Recognition* dan bantuan imagemagick mampu mengekstrak citra ataupun image dan keluarannya berupa text yang langsung dimasukan kolom yang sesuai dengan fungsi data. tersebut, dan sistemnya yang digunakan juga berbasis website sehingga mempermudah layanan administrasi.

1. Referensi ketiga

Judul : Aplikasi Pendeteksi Obat dan Makanan Menggunakan OCR (Optical Character Recognition)

Penulis : (Anisya Sonita & Khairunnisyah Khairunnisyah. 2018)

Makanan merupakan zat terpenting bagi makhluk hidup, baik itu manusia, hewan, maupun tumbuhan. Makanan bisa berupa bahan yang berasal dari tumbuhan ataupun hewan yang diolah oleh manusia sehingga dapat dimakan untuk mendapatkan energi dan bertahan hidup. Obat merupakan suatu benda olahan manusia untuk menyembuhkan sebuah penyakit, atau mencegah gejala - gejala tertentu. Obat bisa berasal dari tumbuh-tumbuhan alami (herbal), atau obat juga bisa berasal dari bahan-bahan tertentu yang dipadukan dan diolah oleh para ilmuwan, para ahli ataupun dokter, sehingga menghasilkan benda yang dapat dikonsumsi guna menyehatkan tubuh. OCR *(Optical Character Recognition)* atau sering disebut dengan Pengenalan Pola karakter yang merupakan salah satu cabang dari pengenalan pola. Salah satu metode pengenalan pola angka atau huruf. Metode String matching adalah suatu algoritma yang digunakan untuk memecahkan masalah pencocokan suatu teks terhadap suatu teks lain. Dari hasil analisa, perancangan, pengujian dan implementasi program yang telah dilakukan, Metode *String Matching*  ini dapat membantu masyarakat dalam mendeteksi obat dan makanan tersebut sudah terdaftar atau belum pada Badan Pengawasan obat dan makanan.

1. Referensi keempat

Judul : Penerapan Optical Character Recognition Pada Dokumen Surat Keputusan Menggunakan Algoritma Reduced Support Vector Machines

Penulis : (Naibaho & Lamhot Sudiarta, 2020)

*Optical Character Recognition (*OCR*)* adalah aplikasi yang menerjemahkan gambar karakter menjadi bentuk teks dengan cara menyesuaikan pola karakter . Surat Keputusan adalah surat yang berisi suatu keputusan yang dibuat oleh pimpinan suatu organisasi atau lembaga berkaitan dengan kebijakan organisasi atau lembaga tersebut. Penelitian ini akan membahas tentang OCR pada Surat Keputusan dengan menggunakan metode *Reduced Support Vector Machines (*RSVM*)* karena pada penelitian sebelumnya belum ditemukan penelitian yang membahas mengenai OCR pada dokumen Surat Keputusan dengan metode RSVM. Tujuan dari penelitian ini adalah untuk mengetahui tingkat akurasi dari ekstraksi informasi dengan menggunakan metode RSVM untuk dokumen Surat Keputusan. Pada penelitian ini terdapat beberapa tahapan yaitu preprocessing yang meliputi proses *resiza*, *grayscale*, *thresholding*, segmentasi dan *resize*. Citra SK yang digunakan adalah sebanyak 10 citra latih. Berdasarkan hasil pengujian didapatkan persentasi akurasi OCR tertinggi sebesar 96,20%. Hasil akurasi ini dipengaruhi oleh data latih yang digunakan pada proses pelatihan.

1. Referensi kelima

Judul : PENERAPAN OPTICAL CHARACTER RECOGNITION (OCR) UNTUK PENJADWALAN ACARA DENGAN GOOGLE CLOUD VISION API ALGORITMA CONVOLUTION NEURAL NETWORK (CNN)

Penulis : (S JULIANTO, 2020)

Teknologi *Optical Character Recognition* (OCR) adalah salah satu cabang pengenalan pola yang digunakan untuk konversi citra karakter baik berupa tulisan cetak maupun tulisan tangan menjadi karakter yang dapat dikenali oleh komputer. Pengenalan karakter oleh OCR pada suatu citra kemudian akan menghasilkan data digital yang dapat diolah lebih lanjut sesuai kebutuhan. OCR dapat dimanfaatkan untuk membuat suatu aplikasi yang mengotomatisasi penjadwalan acara pada poster dan surat undangan yang masuk melalui media sosial sehingga dapat membantu pengguna dalam mengurangi acara yang terlewati atau terlupakan. Aplikasi penjadwalan acara secara otomatis dibuat dengan menerapkan *Optical Character Recognition Google Cloud Vision algoritma Convolution Neural Network*, serta hasil pengolahan diproses dengan Regular Expression untuk mendeteksi atribut acara seperti tanggal, waktu, judul serta lokasi sehingga dapat dijadwalkan secara otomatis melalui aplikasi kalender pada smartphone android. Aplikasi penjadwalan ini memungkinkan melakukan pendeteksian atribut acara pada objek surat undangan dan poster. Hasil dari penelitian ini menunjukan bahwa penerapan *Google Cloud Vision API* untuk melakukan pengenalan karakter yang kemudian hasil pengenalan karakter tersebut akan diproses lebih lanjut menggunakan *Regular Expression* untuk mendapatkan informasi atribut acara berhasil diterapkan pada surat undangan dan poster. Dimana akurasi pendeteksian pada gambar surat undangan scan memiliki akurasi pendeteksian sebesar 96,00%, sedangkan untuk gambar surat undangan potret kamera memiliki akurasi sebesar 94,00 %. Pada hasil pengujian gambar poster scan memiliki akurasi pendeteksian sebesar 96,67%, sedangkan untuk gambar poster potret kamera memiliki akurasi sebesar 93,00%. Akurasi pendeteksian pada gambar hasil potret kamera memiliki akurasi lebih rendah dari hasil pendeteksian pada gambar hasil scan. Hal ini dikarenakan pada gambar yang diambil melalui potret kamera memungkinkan terdapat adanya noise tambahan saat dipotret sehingga mempengaruhi hasil pendeteksian.

## Landasan Teori

Adapun teori-teori yang berhubungan dengan judul yang dibahas dan penulis jadikan sebagai referensi dalam penulisan tugas akhir ini sebagai berikut :

## *Reimbursement*

Menurut Internal Revenue Service (2021:41)[[1]](https://www.irs.gov/pub/irs-pdf/p15.pdf) Reimbursement adalah suatu kompensasi yang dilakukan oleh pihak perusahaan untuk pengeluaran yang dikeluarkan dalam jumlah yang lebih oleh pihak karyawan yang menggunakan dana pribadinya.

## Sistem

Menurut Hutahaean (2015:2) Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran tertentu.

Menurut Tata Sutabri (2012:6) Pada dasarnya sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama- sama untuk mencapai tujuan tertentu.

## *Optical Character Recognition*

Menurut Jain et al (2013) *Optical Character Recognition* (OCR) dapat didefinisikan sebagai analisis elektronik dari sebuah citra dalam upaya mengidentifikasikan wilayah yang memiliki informasi tekstual dan ekstraksi atau pengenalan teks dari citra yang diberikan.

Menurut Lian (2009) Definisi *Optical Character Recognition* (OCR) secara sederhana adalah proses pengenalan teks hasil print, tulisan tangan, hasil mesin ketik dan sebagainya dari suatu berkas *hard copy* yang diubah ke berkas digital.

Menurut Eikvil (1993) Pengenalan optis dilakukan secara *off-line* setelah penulisan atau proses pencetakan selesai, berkebalikan dengan pengenalan online dimana komputer mengenali karakter saat digambarkan. Hasil pengenalan karakter dari tulisan tangan maupun cetak komputer tergantung pada kualitas dari input data tersebut.

## *Android*

Menurut Supriyade (2020) Android adalah sistem operasi yang dikeluarkan oleh *Google* khususnya untuk smartphone dan tablet. *Android* merupakan sebuah sistem operasi yang berbasis Linux untuk telepon selular, seperti telepon pintar dan komputer tablet. *Android* menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak. *Android* adalah sistem operasi yang bersifat terbuka (*open source*). Disebut *open source* karena sumber kode dari sistem operasi dapat dilihat, di unduh, dan dimodifikasi secara bebas.

Beberapa varian *Android* telah diluncurkan sejak pertama kali diperkenalkan pada tahun 2007, dimulai dengan *Android* 1.1 pada tahun 2009, kemudian disempurnakan dengan *Android* *Cupcake* 1.5, *Donut* 1.6, dan *Eclair* 2.0/2.1 pada tahun yang sama. Selanjutnya, versi Froyo 2.2 pada tahun 2010, *Gingerbread* 2.3 pada tahun yang sama, serta *Honeycomb* 3.0/3.1 pada tahun 2011 dirilis, khusus untuk tablet. Pada tahun 2011 juga, *Android* ICS 4.0, *Jelly Bean* 4.1 pada tahun berikutnya, dan *KitKat* 4.4 pada tahun 2013 dirilis. *Lollipop* 5.0.2 pada tahun 2014, *Marshmallow* 6.0 pada tahun 2015, *Nougat* 7.0 pada tahun 2016, *Oreo* 8.0 pada tahun 2017, serta *Pie* 9.0 pada tahun 2018 merupakan versi terbaru pada saat ketersediaan data ini, sementara *Android* 10 dan 11, serta *Android* 12 *Snow Cone* baru saja diperkenalkan pada Oktober 2021. Setiap versi membawa peningkatan fitur dan performa untuk memberikan pengalaman pengguna yang lebih baik.

## *Content Management System*

Menurut Ronaldo Prass (2018) CMS adalah sebuah sistem yang membantu pengguna untuk memudahkan mereka mengelola isi dari situs web. CMS biasanya terdiri dari tiga bagian utama: *frontend*, *backend*, dan *database*. *Frontend* adalah tampilan *website* yang dapat diakses oleh pengguna, *backend* adalah antarmuka pengguna yang hanya dapat diakses oleh admin atau pengguna tertentu untuk mengelola konten, sedangkan *database* menyimpan semua konten dan informasi lainnya yang diperlukan oleh aplikasi.

## Pengembangan Sistem

## Metode *Kanban*

Menurut David J. Anderson (2019), *Kanban* adalah sebuah sistem manajemen yang dirancang untuk meningkatkan kinerja dan kualitas dalam organisasi. *Kanban* memanfaatkan papan atau papan catatan (*board*) dengan kotak-kotak atau kolom-kolom untuk merepresentasikan langkah-langkah atau tahapan dalam proses kerja.

Metode *Kanban* dapat diaplikasikan dalam beberapa tahapan atau langkah, antara lain:

1. Visualisasi Proses: Menentukan dan memvisualisasikan langkah atau tahapan dalam proses kerja pada papan catatan atau *board* agar dapat dipahami oleh semua anggota tim.
2. Batas WIP (*Work in Progress*): Menetapkan batasan untuk jumlah tugas yang sedang dalam proses atau dalam tahap perbaikan agar anggota tim dapat fokus pada satu tugas atau pekerjaan dan mencegah terlalu banyak pekerjaan menumpuk.
3. Kelola Aliran Proses: Memastikan aliran pekerjaan berjalan secara efektif dengan melakukan pemantauan secara teratur dan melakukan perubahan atau penyesuaian jika terdapat gangguan atau hambatan dalam proses.
4. Membuat Aturan Proses (*Process Policies*): Menetapkan aturan atau kebijakan yang jelas dan mudah dipahami oleh semua anggota tim untuk melakukan tugas atau memecahkan masalah.
5. Menjalankan Pembaruan Berkelanjutan (*Continual Improvement*): Menerapkan prinsip pembaruan berkelanjutan sebagai siklus evaluasi dan perbaikan. Tim harus melakukan evaluasi terhadap proses kerja secara berkala dan melakukan perubahan dan perbaikan jika diperlukan untuk meningkatkan kinerja dan efisiensi tim.

Dalam menggunakan metode Kanban, sangat penting untuk memahami setiap tahapannya dan menerapkannya secara konsisten agar dapat memaksimalkan manfaatnya. Setiap anggota tim harus memahami prinsip-prinsip Kanban dan mematuhi aturan atau kebijakan yang telah ditetapkan untuk mencapai tujuan yang diharapkan. Dengan melalui setiap tahapan Kanban, tim dapat meningkatkan kinerja mereka dan mencapai hasil yang lebih baik.

## *Unified Modeling Language* (UML)

Untuk mengembangkan teknologi perangkat lunak, diperlukan sebuah alat untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang awam dapat mengerti pemodelan perangkat lunak. Maka dari itu dikembangkanlah unified modelling language (UML). Penggunaan UML ini hanya berfungsi untuk melakukan pemodelan (Maulana, 2014).

Menurut Rosa & Shalahuddin (2015:137), “UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung”. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.

Maka dari itu, *unified modelling language* (UML) dapat diartikan sebagai sebuah bahasa standar atau alat yang digunakan untuk memodelkan pembangunan perangkat lunak menggunakan symbol-simbol tertentu. Kategori unified modelling language (UML) yang digunakan dalam perancangan sistem ini terdiri dari *activity*, *use case*, *class* dan *sequence* *diagram*.

## Diagram-Diagram UML (*Unified Modeling Language*)

Berikut ini adalah diagram UML menurut (Rosa, dan Shalahuddin, 2015) yaitu :

1. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk melakukan (behavior) sistem informasi yang akan dibuat. Use case digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

### Tabel 2. 1 Simbol *Use Case Diagram*

|  |  |  |
| --- | --- | --- |
| **Nama** | **Gambar** | **Keterangan** |
| Use Case |  | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling tertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama *Use Case*. |
| Aktor |  | Orang, proses yang berinteraksi dengan sistem informasi yang dibuat diluar sistem informaasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang: biasanya dinyatakan menggunakan kata benda diawal frase nama aktor |
| Asosiasi |  | Komunikasi antara aktor dan *use case* yang berpartisifasi pada *use case* atau bisa saja *use case* mempunya interaksi dengan aktor |
| Ekstensi  /*Extend* |  | Relasi use case tambahan kesebuah use case dinamakan use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu |
| Generalisasi |  | Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. |

1. *Class Diagram*

*Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas- kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut atau operasi.

### Tabel 2.2 Simbol Class Diagram

|  |  |  |
| --- | --- | --- |
| **Nama** | **Gambar** | **Keterangan** |
| Kelas | Table  Description automatically generated | Kelas pada struktur sistem. |
| *Interface* |  | Sama dengan konsep interface dalam pemrograman berorientasi objek. |
| Asosiasi |  | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*. |

1. *Sequence Diagram*

*Sequence diagam* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek.

### Tabel 2.3 Simbol Sequence Diagram

|  |  |  |
| --- | --- | --- |
| **Nama** | **Gambar** | **Keterangan** |
| Aktor |  | Orang yang berinteraksi dengan sistem. |
| *Entity* | Shape, circle  Description automatically generated | Menggambarkan hubungan kegiatan yang akan dilakukan. |
| *Boundary* | Shape, circle  Description automatically generated | Menggambarkan sebuah gambaran dari *foem*. |
| *Control* | Shape, circle  Description automatically generated | Menggambarkan penghubung antar *Boundary* dengan tabel. |

1. *Activity Diagram*

*Activity diagram* menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Perlu diperhatikan bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

### Tabel 2. 4 Tabel Simbol *Activity* *Diagram*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nama** | | **Gambar** | | **Keterangan** |
| Status awal | |  | | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal |
| Aktivitas | | *A picture containing shape  Description automatically generated* | | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja |
| Percabangan/  *Decision* | |  | | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| Penggabungan/ *Join* | |  | | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu |
| Status akhir | |  | | Status akhir yang dilakukan sistem, sebuah diagramaktivitas memiliki sebuah status akhir. |
| *Swimlane* | *A picture containing text, shoji, tiled  Description automatically generated* | | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. | |

## Pengujian

*Black Box* testing adalah tahap yang digunakan untuk menguji kelancaran program yang telah dibuat. Pengujian ini penting dilakukan agar tidak terjadi kesalahan alur program yang telah dibuat.

Menurut Rosa dan Salahuddin (2015:275) “*Black Box* testing yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program”.

Menurut Rizky (2011:264) “*Black Box* testing adalah tipe testing yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya”.

Sedangkan menurut Mustaqbal (2015:34) “*Black Box* Testing befokus pada spesifikasi fungsional dari perangkat lunak, kumpulan kondisi input dan melakukan pengetesan pada fungsional program”.

## *Entity Relationship Diagram* (ERD)

Menurut A.S dan Shalahudin (2018:50) menyatakan bahwa “*Entity Relationship Diagram* (ERD) dikembangkan berdasar teori himpunan dalam bidang matematik, ERD digunakan untuk pemodelan basis data relasional, ERD 18 digununakan untuk permodelan basis data relasional”. Berikut adalah simbolsimbol yang digunakan pada ERD dengan Notasi Chen:

### Tabel 2. 5 Simbol ERD

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Deskripsi** |
| nama\_entitas | Entitas / *entity* | Entitas merupakan data inti yang akan disimpan, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer. |
| nama\_atribut | Atribut | Field atau kolom data yang disimpan dalam suatu entitas. |
| nama\_atribut | Atribut  Kunci Primer | Field atau kolom data yang disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* yang diinginkan. |
| relasi | Relasi | Relasi yang menghubungkan antar entitas. biasanya diawali dengan kata kerja. |
|  | Asosiasi /  *association* | Penghubung antara relasi dan entitas di mana kedua nya memiliki ikatan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas 1 ke N atau sering disebut dengan one to many. |

## *Language Record Record Structure* (LRS)

Menurut Hasugian dan shidiq (2012:608) memberikan batasan bahwa LRS adalah “sebuah model sistem yang digambarkan dengan sebuah diagram ERD akan mengikuti pola atau aturan permodelan tertentu dalam kaitannya dengan konversi ke LRS”.

Menurut Lestari (2013: 12) “*Logical* *Record* *Structure* dibentuk dengan nomor tipe record , beberapa tipe record digambarkan oleh kotak empat persegi panjang dan dengan nama yang unik”.

Dari pemahaman para ahli Sehingga dapat disimpulkan LRS adalah “model system berbentuk gambar dengan sebuah diagram ERD akan mengikuti pola dan dibentuk dengan nomor tipe record dan beberapa tipe record digambarkan oleh kotak empat persegi panjang”.

## Sistem Pendukung

Dalam pembuatan tugas akhir ini penulis menggunakan beberapa software pendukung guna menunjang keberhasilan tugas akhir ini. Adapun software pedukung yang di gunakan adalah sebagai berikut :

## *Javascript*

Bahasa pemrograman yang bersifat client side yang permrosesanya dilakukan oleh client sering digunakan pada web browser untuk menciptakan halaman web yang menarik.

Menurut Kadir dan Triwahyuni (2013:325) “*JavaScript* adalah bahasa pemrograman yang biasa diletakkan bersama kode HTML untuk menentukan suatu tindakan”.

Menurut Sibero (2013:150) “*Javascript* adalah bahasa skrip (*Scripting* *language*), yaitu kumpulan intruksi perintah yang digunakan untuk mengendalikan beberapa bagian dari sistem operasi”.

## PostgreSQL

Menurut (Obe and Hsu 2016) dalam buku *PostgreSQL*: *Up and Running* adalah sebuah sistem basis data yang disebarluaskan secara bebas menurut Perjanjian lisensi BSD. Piranti lunak ini merupakan salah satu basis data yang paling banyak digunakan saat ini, selain *MySQL* dan Oracle. *PostgreSQL* menyediakan fitur yang berguna untuk replikasi basis data. Fitur-fitur yang disediakan *PostgreSQL* antara lain DB *Mirror*, *PGPool*, *Slony*, *PGCluster*, dan lain-lain.

Berikut ada tiga tools utama dalam pada *PostgreSQL* yang mendukung penggunaan *PostgreSQL* bagi developer:

1. *PSQL*

PSQL adalah sebuah tampilan pada command-line yang berisi perintah tertulis (*query*) pada *PostgreSQL*.

1. *PgAdmin*

PgAdmin adalah aplikasi grafis yang digunakan sebagai alat untuk menggunakan *PostgreSQL*. *PgAdmin* merupakan aplikasi desktop dan dapat terhubung ke berbagai server *PostgreSQL* walaupun dengan sistem operasi yang berbeda-beda.

## Visual Studio Code

*Visual Studio Code* adalah editor kode sumber yang dikembangkan oleh *Microsoft* untuk *Windows*, *Linux* dan *MacOS* . *Visual Studio Code* termasuk dukungan untuk *debugging* , kontrol *Git* yang tertanam dan *GitHub*, penyorotan sintaksis, penyelesaian kode cerdas, *snippet*, dan *refactoring* kode.

Ini sangat dapat disesuaikan, memungkinkan pengguna untuk mengubah tema, pintasan keyboard, preferensi, dan menginstal ekstensi yang menambah fungsionalitas tambahan. Kode sumber adalah sumber bebas dan terbuka dan dirilis di bawah Lisensi MIT yang permisif. Binari yang dikompilasi adalah *freeware* dan gratis untuk penggunaan pribadi atau komersial.

Menurut Bree, R. T., & Gallagher, G. (2016). *Visual Studio Code* didasarkan pada *Electron* , sebuah kerangka kerja yang digunakan untuk menggunakan aplikasi Node.js untuk desktop yang berjalan pada mesin tata letak *Blink*. Meskipun menggunakan kerangka *Electron*, perangkat lunak tidak menggunakan Atom dan sebagai gantinya

## React Native

MenurutEisenman, B. (2015) *React native* adalah *framework* dari *javascript* yang digunakan untuk membuat aplikasi mobile pada platform IOS dan android. *React native* ini dibuat berdasarkan *React*, tetapi tidak mengacu kepada *browser*, melainkan ke platform *mobile*. seperti *reactjs, react native* ditulis dari perpaduan javascript dan *XMLesque* yang disebut JSX. *React native* ini yang menjembatani *native* *rendering* APIs pada *Objective-C*(untuk IOS) dan Java (untuk Android) sehingga dapat digunakan pada IOS dan Android sekaligus. *React native* juga bisa digunakan untuk mengakses fungsi dari platform contohnya seperti kamera dan GPS.

## Node.JS

*Node.js* merupakan salah satu platform pengembang yang dapat digunakan untuk membuat aplikasi berbasis *Cloud*. *Node.js* dikembangkan dari *engine* *JavaScript* yang dibuat oleh Google untuk browser *Chrome* serta beberapa pustaka lainnya. Node.js menggunakan JavaScript sebagai bahasa pemrograman dan *event-driven*, *non-blocking* I/O (*asynchronous*) model yang membuatnya ringan dan efisien. *Node.js* memiliki fitur *built-in* *HTTP* server *library* yang menjadikannya mampu menjadi sebuah web server tanpa bantuan software lainnya seperti *Apache* dan *Nginx*.

Menurut Tilkov, S., & Vinoski, S. (2010). Pada dasarnya, *Node.js* adalah sebuah runtime *environment* dan script library. Sebuah runtime environment adalah sebuah software yang berfungsi untuk mengeksekusi, menjalankan dan mengimplementasikan fungsi-fungsi serta cara kerja inti dari suatu bahasa pemrograman. Sedangkan *script* *library* adalah kumpulan, kompilasi atau bank data berisi skrip/kode-kode pemrograman.

*Node.js* dibangun menggunakan *JavaScript* dan C++, terdapat arsitektur serta fungsi dari *Google V8* di dalamnya yang berfungsi sebagai compiler ditulis dalam C++ dan *library* bekerja untuk menangani operasi *asynchronous* I/O dan main *event loop*.

## Strapi

*Strapi* merupakan sebuah Framework Manajemen Konten (CMS) *Open Source* yang dikembangkan untuk membantu para *developer* membangun API secara lebih mudah. *Framework* ini dikembangkan dengan *Node.js* yang terbukti kehandalannya dan memiliki beberapa fitur antara lain admin panel yang dapat disesuaikan sesuai dengan kebutuhan, sistem plugin yang memungkinkan developer menambahkan fitur-server API sesuai dengan kebutuhan, serta *frontend agnostic* yang memungkinkan *Strapi* dapat digunakan dengan berbagai *framework* seperti *React*, *Vue*, *Angular*, dan sebagainya. Selain itu, Strapi juga memiliki sistem keamanan standar seperti CSRF, CORS, P3P, Xframe, XSS, dan sebagainya yang telah diimplementasikan secara internal di dalamnya.

# BAB III

# ANALISA PERANCANGAN SISTEM

## Analisa Sistem

Merupakan kegiatan mengidentifikasi masalah, mengevaluasi, membuat model serta membuat spesifikasi sistem dengan tujuan untuk merancang sistem baru atau memperbaiki kekurangan dari sistem yang telah ada. Analisa sistem secara keseluruhan sangat perlu dalam penelitian ini untuk dapat mengetauhi kelemahan dan kekurangan dari sistem tersebut, baik dari cara kerja sistem maupun pihak pelaksananya dan segala sesuatu yang terlibat dalam sistem tersebut, untuk pembuatan sistem informasi baru harus lebih terprogram dan terimplementasi kedalam database. Analisis sistem merupakan tahapan untuk melakukan penguraian dari suatu sistem informasi yang akan dibangun ke dalam bagian komponen-komponen dengan maksud untuk mengidentifikasi dan mengevaluasi kekurangan-kekurangan yang ada, serta kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Analisa sistem akan sangat membantu dalam mengetahui informasi-informasi tentang sistem yang sedang berjalan, sehingga dalam analisa sistem diharapkan dapat diketahui sejauh mana kebutuhan yang telah ditangani oleh sistem yang sedang berjalan dan bagaimana agar kebutuhan-kebutuhan yang belum bisa terpenuhi dapat diberikan solusinya dan diterapkan dalam tahap perancangan sistem informasi gereja ini. Tahap analisis dilakukan sebelum tahap desain sistem. Tahap ini merupakan tahap yang kritis karena kesalahan yang ada dalam tahap ini akan menyebabkan kesalahan-kesalahan pada tahap selanjutnya.

## Analisa Sistem Berjalan

Berdasarkan observasi yang dilakukan pada PT. Kayana Nusa Teknologi akan dijabarkan mekanisme menguraikan proses-proses atau aktivitas-aktivitas kerja sistem *reimbursement* yang sedang berjalan sebelumnya, maka berikut aktifitas sistem yang sudah berjalan di PT. Kayana Nusa:

1. Para pegawai PT. Kayana Nusa Teknologi melampirkan struk asli kedalam kertas laporan dan menuliskan apa saja struk yang akan di *reimburse.*
2. Staff admin PT. Kayana Nusa Teknologi melakukan validasi terhadap laporan yang dikirimkan oleh karyawan.
3. Staff admin PT. Kayana Nusa Teknologi melakukan aksi setuju atau menolak laporan *reimburse* tersebut dengan alasan yang valid.

## Analisa Sistem Usulan

Berdasarkan kebutuhan pengguna maka dibuatlah analisa sistem usulan untuk mengatasi dan membenahi masalah yang sudah ada sebelumnya. Berikut adalah penjabaran analisa sistem usulan yang akan dibuat:

1. Pada sistem usulan akan dibangun sistem berbasis android guna membantu PT. Kayana Nusa Teknologi dalam efisiensi waktu dalam melakukan proses *reimbursement*
2. Pada proses *reimbursement* dilakukan secara terkomputerisasi menggunakan database PostgreSQL sehingga dengan begitu menjadikan proses *reimbursement* lebih terkendali dan juga cepat.
3. Dalam proses reimbursement tidak perlu lagi dibuat secara tertulis karena proses pembuatan laporan *reimbursement* sudah dibuat secara otomatis dalam aplikasi.
4. Dalam proses reimbursement juga terdapat opsi otomatis mendeteksi total harga sehigga menjadikan proses validasi form laporan reimbursement lebih cepat.

## Analisa Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang berisi tentang proses-proses yang akan dilakukan oleh sistem. Aplikasi Reimbursement PT. Kayana Nusa Teknologi memiliki beberapa fitur yang akan ditampilkan, di antaranya adalah sebagai berikut:

Superadmin:

1. Melakukan *login* pada *dashboard Strapi.*
2. Melakukan *input* data karyawan.
3. Memberi akses *role.*

Admin:

1. Melakukan login pada aplikasi.
2. Melakukan validasi terhadap pengajuan *reimbursement* karyawan.
3. Melakukan aksi setuju atau menolak terhadap pengajuan reimbursement dengan alasan yang valid.

Karyawan:

1. Melakukan *login* pada aplikasi.
2. Melakukan *input* form pengajuan reimbursement pada aplikasi.
3. Melihat aksi yang dilakukan staff terhadap pengajuan *reimbursement*.

## Analisa Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional adalah jenis kebutuhan yang terkait dengan perangkat keras dan lunak yang digunakan dalam membangun sebuah sistem. Kebutuhan non-fungsional menitik beratkan pada sifat khas yang dimiliki oleh sistem yang akan dirancang. Beberapa jenis kebutuhan non-fungsional antara lain sebagai berikut:

1. Kebutuhan Perangkat Keras (*Hardware*)

Berikut adalah spesifikasi yang digunakan untuk membangun sistem aplikasi reimbursement PT. Kayana Nusa Teknologi :

* Processor : Apple M1 Pro.
* Hardisk : 500 GB.
* RAM : 16 GB.
* VGA : Built-in Liquid Retina XDR Display.
* OS : macOS Monterey.

1. Kebutuhan Perangkat Lunak (Software)

Berikut adalah perangkat lunak yang digunakan untuk membangun sistem ini sebagai berikut :

* Visual Studio Code.
* PostgreSQL.
* Strapi.
* Google Chrome sebagai *browser.*
* Android SDK.

## Perancangan UML (*Unified Modeling Language*)

## *Use Case Diagram*

Use case diagram digunakan untuk menggambarkan interaksi antar pengguna sistem dengan sistem, yaitu penjelasan aktor-aktor yang melakukan suatu prosedur dalam sistem serta menjelaskan tanggapan-tanggapan sistem terhadap action yang dilakukan oleh aktor. Adapun penggambaran use case diagram adalah sebagai berikut.

Diagram

Description automatically generated

### Gambar 3.1 *Use Case Diagram*

## *Activity Diagram*

Pada activity diagram, aktivitas-aktivitas yang terjadi atau alur kerja pada use case selanjutnya digambarkan secara grafis dibuat dengan activity diagram. Dari use case diagram yang telah dibuat di atas, Berikut adalah activity diagram yang dibuat:

1. ***Actitivity Diagram Login***

Diagram

Description automatically generated

### Gambar 3.2 *Activity Diagram Login*

Aktifitas dimulai dengan aplikasi menampilkan form login. Kemudian memasukkan username dan password yang sudah terdaftar. Jika berhasil (valid) admin akan diarahkan ke halaman *dashboard*. Sedangkan jika tidak valid, maka admin akan diarahkan kembali ke halaman form login dan sistem menampilkan pesan *error*.

1. ***Actitivity* Pengajuan *Reimbursement***

Diagram

Description automatically generated

### Gambar 3.3 *Activity Diagram* Pengajuan *Reimbursement*

Aktifitas dimulai dengan aplikasi menampilkan form *login*. Kemudian memasukkan *username* dan *password* yang sudah terdaftar. Jika berhasil (valid) admin akan diarahkan ke halaman dashboard. Sedangkan jika tidak valid, maka karyawan akan diarahkan kembali ke halaman form *login* dan sistem menampilkan pesan eror. Setelahnya melakukan isi data *reimbursement* sesuai dengan data yang dibutuhkan. Sedangkan jika tidak valid maka karyawan akan mendapatkan pesan *error* dan jika benar maka karyawan dapat melihat riwayat *reimbursement* yang diajukannya.

1. ***Actitivity* Melihat Detail *Reimbursement***

*Diagram

Description automatically generated*

### Gambar 3.4 *Activity Diagram* Detail *Reimbursement*

Melihat detail *reimbursement* dapat dilakukan setelah karyawan melakukan pengajuan *reimbursement* pada aplikasi dan dapat memilih detail pengajuan berdasarkan tipe kategori yang tersedia

1. ***Activity* Admin *Approval***

### Diagram Description automatically generated

### Gambar 3.5 Activity Admin Approval

Aktifitas dilakukan ketika admin sudah login menggunakan kredensial yang di milikinya. Melihat daftar pengajuan yang membutuhkan persetujuan admin bisa melakukan aksi ketika memilih salah satu dari pengajuan yang ditujukan padanya diantaranya setuju atau menolak. Aksi tersebut dilakukan dengan menambahkan alasan agar lebih konkrit dalam penyampaiannya.

## *Class Diagram*

Class diagram berguna sebagai penggambaran dan pendeskripsian dari class, atribut, dan objek. Class diagram dapat dibuat dengan bagan secara jelas dan terperinci dengan cara memperhatikan kode spesifik apa saja yang dibutuhkan oleh suatu program sehingga mampu mengimplementasikannya ke struktur yang digambarkan.

# Diagram Description automatically generated

### Gambar 3.6 Class Diagram

## *Sequence Diagram*

Sequence diagram menggambarkan interaksi user dengan sistem ketika masuk atau mengakses ke dalam sistem. Dengan menggunakan sequence diagram maka interaksi antara objek yang dibutuhkan untuk menjalankan use case dalam interaksi objek sebagai berikut:

1. ***Sequence* *Diagram* *Login***

Diagram

Description automatically generated

### Gambar 3.7 Sequence Diagram Login

Aktifitas dimulai dengan aplikasi menampilkan form login. Kemudian memasukkan username dan password yang sudah terdaftar. Jika berhasil (valid) admin akan diarahkan ke halaman *dashboard*. Sedangkan jika tidak valid, maka admin akan diarahkan kembali ke halaman form login dan sistem menampilkan pesan *error*.

1. ***Sequence Diagram* Pengajuan *Reimbursement***

**Diagram

Description automatically generated**

### Gambar 3.8 Sequence Diagram Pengajuan Reimbursement

Aktifitas dimulai dengan aplikasi menampilkan form *login*. Kemudian memasukkan *username* dan *password* yang sudah terdaftar. Jika berhasil (valid) admin akan diarahkan ke halaman dashboard. Sedangkan jika tidak valid, maka karyawan akan diarahkan kembali ke halaman form *login* dan sistem menampilkan pesan eror. Setelahnya melakukan isi data *reimbursement* sesuai dengan data yang dibutuhkan. Sedangkan jika tidak valid maka karyawan akan mendapatkan pesan *error* dan jika benar maka karyawan dapat melihat riwayat *reimbursement* yang diajukannya.

1. ***Sequence Diagram* Melihat Detail *Reimbursement***

Diagram

Description automatically generated

### Gambar 3.9 Sequence Diagram Melihat Detail Reimbursement

Melihat detail *reimbursement* dapat dilakukan setelah karyawan melakukan pengajuan *reimbursement* pada aplikasi dan dapat memilih detail pengajuan berdasarkan tipe kategori yang tersedia

1. **Sequence Diagram Admin Approval**

### Diagram Description automatically generatedGambar 3.9 Sequence Diagram Melihat Detail Reimbursement

Aktifitas dilakukan ketika admin sudah login menggunakan kredensial yang di milikinya. Melihat daftar pengajuan yang membutuhkan persetujuan admin bisa melakukan aksi ketika memilih salah satu dari pengajuan yang ditujukan padanya diantaranya setuju atau menolak. Aksi tersebut dilakukan dengan menambahkan alasan agar lebih konkrit dalam penyampaiannya.

* 1. **Perancangan Basis Data**

Perancangan basis data bertujuan untuk memberikan penjelasan secara detail tentang masing-masing informasi yang ada dalam basis data yang digunakan dalam sistem aplikasi *reimbursement* PT. Kayana Nusa Teknologi.

## Entity Relationship Diagram (ERD)

Pada Entity Relationship Diagram (ERD), hubungan antara kunci relasi dengan file yang direlasikan merupakan kunci utama dari masing-masing file. ERD terdiri dari entitas-entitas yang saling berhubungan dalam pembuatan sistem aplikasi *reimbursement* PT. Kayana Nusa Teknologi.

Entitas user memiliki hubungan dengan entitas pengajuan, file, user, persetujuan di mana relasi ini berperan sangat penting dalam proses pengelolaan data *reimbursement* PT. Kayana Nusa Teknologi. Pada entitas setiap database memiliki atribut primary-key agar bisa dilakukan relasi antar *database*.

## *Logical Record Structure* (LRS)

LRS merupakan hasil dari pemodelan yang terbentuk dari relasi-relasi antar entitas yang telah dijelaskan sebelumnya. Berikut merupakan Logical Record Structure (LRS) yang ada pada sistem aplikasi *reimbursement* PT. Kayana Nusa Teknologi.

Graphical user interface, application, Teams

Description automatically generated

### Gambar 3.10 Logical Record Structure

## Spesifikasi Basis Data

Spesifikasi basis data bertujuan untuk memberikan penjelasan secara detail tentang masing-masing tabel yang ada dalam basis data yang digunakan dalam sistem aplikasi *reimbursement* PT. Kayana Nusa Teknologi. Berikut adalah tabel-tabel yang akan digunakan:

1. Tabel *Company*

Tabel ini berisi informasi mengenai perusahaan yang terkait dengan pengguna (user) pada sistem. Tabel ini memiliki *field id\_company* sebagai *primary-key* dan *field id\_user* sebagai *foreign-key* yang merujuk pada *field id\_user* pada tabel *User*. Setiap perusahaan hanya dapat terhubung dengan satu pengguna saja.

### Tabel 3.1 *Company*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NO** | **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |  |
| 1 | *id\_company* | *int* | - | *Primary Key* |  |
| 2 | *id\_user* | *int* | - | Foreign Key merujuk pada tabel *User* |  |
| 3 | *name* | *varchar* | 100 | Nama perusahaan |  |

1. Tabel *Role*

Tabel Role ini adalah tabel *default* yang *Strapi* punya. sistem *role-based user* merupakan salah satu fitur Strapi yang digunakan pada sistem aplikasi *reimbursement* PT. Kayana Nusa Teknologi.

Tabel Role merupakan tabel yang berisi data peran atau posisi yang dimiliki oleh pengguna pada sistem. Tabel ini memiliki enam kolom, yaitu *id, name, code, description, created\_at, updated\_at, created\_by\_id, dan updated\_by\_id*. Kolom id merupakan kolom kunci utama atau *primary key* pada tabel ini. Kolom name berisi nama peran atau posisi, kolom *code* berisi kode peran atau posisi, dan kolom *description* berisi deskripsi tentang peran atau posisi. Sedangkan kolom *created*\_at dan *updated\_at* berisi tanggal dan waktu pembuatan dan pembaruan data, dan kolom *created\_by\_id* dan *updated\_by\_id* merupakan kolom kunci asing atau *foreign key* yang berelasi dengan tabel *User*.

### Tabel 3.2 *Company*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NO** | **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| 1 | *id* | *int* | - | *Primary Key* |
| 2 | *name* | *varchar* | 100 | Nama *role* |
| 3 | *code* | *varchar* | 100 | Kode *role* |
| 4 | *description* | *varchar* | 100 | Deskripsi *role* |
| 5 | *created\_at* | *date* | - | Tanggal dibuat |
| 6 | *updated\_at* | *date* | - | Tanggal di*update* |
| 7 | *created\_by\_id* | *int* | - | ID pengguna yang membuat role |
| 8 | *updated\_by\_id* | *int* | - | ID pengguna yang mengupdate |

1. Tabel *User*

Tabel ini berisi informasi mengenai pengguna (user) pada sistem. Tabel ini memiliki field *id\_user* sebagai *primary-key* dan beberapa *field* sebagai *foreign-key* yang merujuk pada *field* pada tabel lain. Tabel ini juga memiliki beberapa field seperti *username, full\_name, password, resetPasswordToken, confirmationToken, confirmed, position, dan tokenNotification.*

### Tabel 3.3 *User*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NO** | **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| 1 | *id\_user* | *int* | - | *Primary Key* |
| 2 | *id\_role* | *int* | - | *Foreign Key* merujuk pada tabel *Role* |
| 3 | *id\_company* | *int* | - | *Foreign Key* merujuk pada tabel *Company* |
| 4 | *id\_approval* | *int* | - | *Foreign Key* merujuk pada tabel *Approval* |
| 5 | *id\_reimburse* | *int* | - | *Foreign Key* merujuk pada tabel *Reimburse* |
| 6 | *username* | *varchar* | 100 | Username pengguna |
| 7 | *full\_name* | *varchar* | 100 | Nama lengkap pengguna |
| 8 | *password* | *varchar* | 100 | Password pengguna |
| 9 | *resetPasswordToken* | *varchar* | 100 | Token untuk me*reset* *password* |
| 10 | *confirmationToken* | *varchar* | 100 | Token untuk konfirmasi akun |
| 11 | *confirmed* | *boolean* | FALSE | Status konfirmasi akun |
| 12 | *position* | *varchar* | 100 | Posisi pengguna |
| 13 | *tokenNotification* | *varchar* | 100 | Token notifikasi pengguna |

1. Tabel *Approval*

Tabel ini berisi informasi mengenai persetujuan yang dilakukan pada sistem. Tabel ini memiliki *field id\_approval* sebagai *primary-key* dan beberapa field sebagai *foreign-key* yang merujuk pada *field* pada tabel lain. Tabel ini juga memiliki *field* *status* dan *reason*.

### Tabel 4.4 *Approval*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NO** | **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| 1 | *id\_approval* | *int* | - | *Primary Key* |
| 2 | *id\_user* | *int* | - | *Foreign Key* merujuk pada tabel *User* |
| 3 | *id\_reimburse* | *int* | - | *Foreign Key* merujuk pada tabel *Reimburse* |
| 4 | *status* | *varchar* | 100 | Status persetujuan |
| 5 | *reason* | *varchar* | 100 | Alasan persetujuan atau penolakan |

1. Tabel *Reimburse*

Tabel ini berisi informasi mengenai pengajuan reimburse pada sistem. Tabel ini memiliki *field* *id*\_*reimburse* sebagai *primary-key* dan beberapa field sebagai *foreign-key* yang merujuk pada field pada tabel lain. Tabel ini juga memiliki *field type, description, amount, reimburseName, dan attachment*.

### Tabel 4.5 *Reimburse*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NO** | **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| 1 | *id\_reimburse* | *int* | - | *Primary Key* |
| 2 | *id\_user* | *int* | - | *Foreign Key* merujuk pada tabel *User* |
| 3 | *id\_approval* | *int* | - | *Foreign Key* merujuk pada tabel *Approval* |
| 4 | *type* | *varchar* | 100 | Tipe pengajuan *reimburse* |
| 5 | *description* | *varchar* | - | Deskripsi *Reimbursement* |

# BAB IV

# IMPLEMENTASI DAN PENGUJIAN

## Implementasi

Pada tahap implementasi, dilakukan penerapan sistem agar aplikasi *reimbursement* PT. Kayana Nusa Teknologi dapat beroperasi. Untuk memastikan operasinya berjalan secara maksimal, diperlukan perangkat keras dan perangkat lunak.

## Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak merupakan beberapa perangkat lunak yang digunakan dalam perancangan sistem. Agar dapat menjalankan sistem yang telah diusulkan secara optimal, maka dibutuhkan *Software* (perangkat lunak) sebagai berikut ini:

1. *macOS* sebagai sistem operasi.
2. *Node.JS*
3. *Google Chrome* sebagai *browser.*
4. *PostgreSQL*.

## Spesifikasi Perangkat Keras

Spesifikasi perangkat keras merupakan tahapan memaparkan beberapa perangkat keras yang digunakan dalam merancang sistem ini. Adapun kebutuhan hardware (perangkat keras) yang diperlukan dengan spesifikasi sebagai berikut:

a. *Processor Apple* M1.

b. RAM 16 GB.

c. HDD 500 GB.

d. VGA *Built-in Liquid Retina* XDR *Display*.

e. *Mouse*, *keyboard*.

## Implementasi Antarmuka Pengguna (*User Interface*)

Dalam proses pengembangan antarmuka pengguna, fokus diberikan pada elemen antarmuka dan objek yang dilihat atau digunakan oleh pengguna. Berikut adalah implementasi antarmuka pengguna dari aplikasi *reimbursement* PT. Kayana Nusa Teknologi.

1. *Form* *Login* *Superadmin* pada *Strapi*.

*Form* login merupakan halaman untuk user untuk masuk ke dalam *dashboard* admin. Pada halaman login, user harus memasukkan username dan password yang telah didaftarkan oleh admin untuk dapat mengakses dashboard. Jika tidak terdaftar maka user tidak dapat mengakses dashboard tersebut. Berikut merupakan halaman login.

A screenshot of a computer

Description automatically generated

### Gambar 4.1 Tampilan Halaman *Login Dashboard* Admin

1. Tampilan *Dashboard Strapi*

Pada gambar 4.2 terlihat tampilan *dashboard* setelah *superadmin* berhasil masuk ke dalam dashboard. *Superadmin* memiliki kemampuan untuk mengelola database, *input* data karyawan, serta dapat memonitor jalannya aplikasi yang telah berlangsung. Ini adalah halaman dasbor administrator.

A screenshot of a computer

Description automatically generated

### Gambar 4.2 Tampilan Halaman *Login Dashboard* Admin

1. Tampilan *Input* Data Karyawan

Pada gambar 4.3 menunjukkan tampilan yang digunakan untuk menambahkan data karyawan pada aplikasi *reimbursement* PT. Kayana Nusa Teknologi dengan menulis identitas yang dibutuhkan pada halaman *input* data karyawan. Ini adalah halaman untuk melakukan input data karyawan.

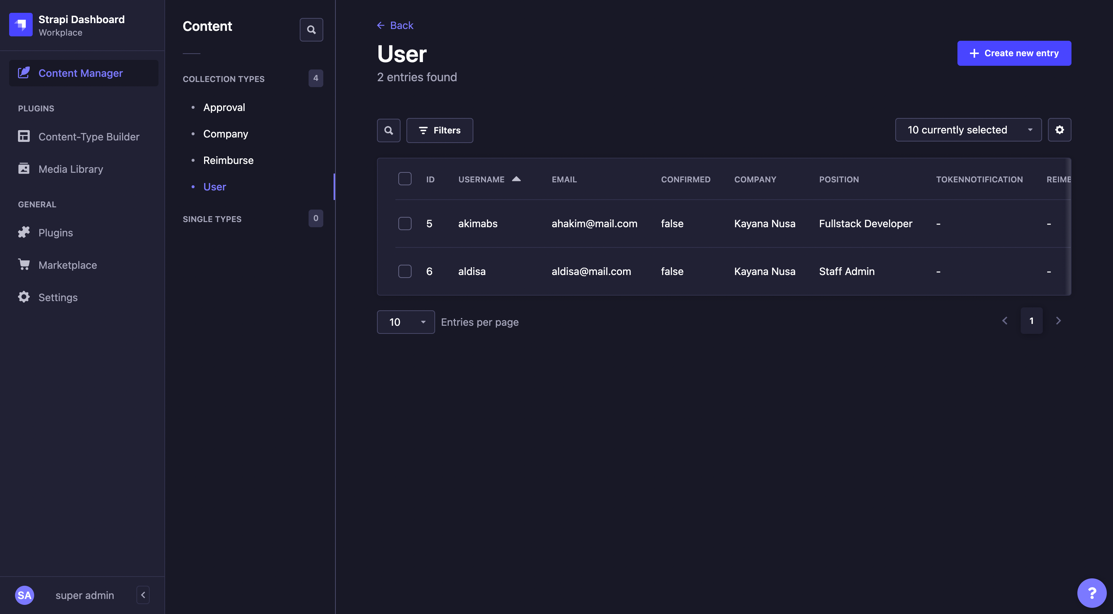
A screenshot of a computer

Description automatically generated with medium confidence

### Gambar 4.3 Tampilan Halaman *Input* Data Karyawan

1. Tampilan Data Karyawan

Pada gambar 4.4 menampilkan daftar karyawan yang telah ditambahkan oleh superadmin dan siap untuk di berikan kepada karyawan PT. Kayana Nusa Teknologi untuk bisa melakukan *login* pada aplikasi *reimbursement*. Ini adalah halaman daftar data karyawan.



### Gambar 4.4 Tampilan Halaman Data Karyawan

1. Tampilan Penyaringan Perusahaan

Pada gambar 4.5 menampilkan halaman penyaringan nama perusahaan pada aplikasi yang berfungsi untuk memastikan bahwa hanya karyawan PT. Kayana Nusa Teknologi yang memiliki hak akses ke halaman daftar data karyawan.

Graphical user interface, application, Teams

Description automatically generated

### Gambar 4.5 Tampilan Halaman Penyaringan Perusahaan

1. Tampilan *Login* Karyawan dan Admin

Pada gambar 4.6 terdapat halaman masuk pada aplikasi yang dapat diakses oleh karyawan PT. Kayana Nusa setelah melewati tahap penyaringan nama perusahaan. Dengan masuk pada halaman ini, karyawan dapat mengakses aplikasi *reimbursement* dan melakukan pengajuan klaim sesuai dengan kebijakan perusahaan.

Graphical user interface, application

Description automatically generated

### Gambar 4.6 Tampilan Halaman Login Aplikasi

1. Tampilan Beranda Karyawan

Pada gambar 4.7 menampilkan tampilan beranda aplikasi *reimbursement* dengan role karyawan, yang memberikan informasi tentang status pengajuan klaim dan tindakan selanjutnya yang dapat dilakukan oleh karyawan.

Graphical user interface, application

Description automatically generated

Gambar 4.7 Tampilan Halaman Beranda Aplikasi Karyawan

1. Tampilan *Input* Reimbursement Karyawan

Pada gambar 4.8 menampilkan tampilan form pengajuan reimbursement karyawan, yang memungkinkan karyawan untuk mengisi informasi terkait klaim pengeluaran yang akan diajukan.

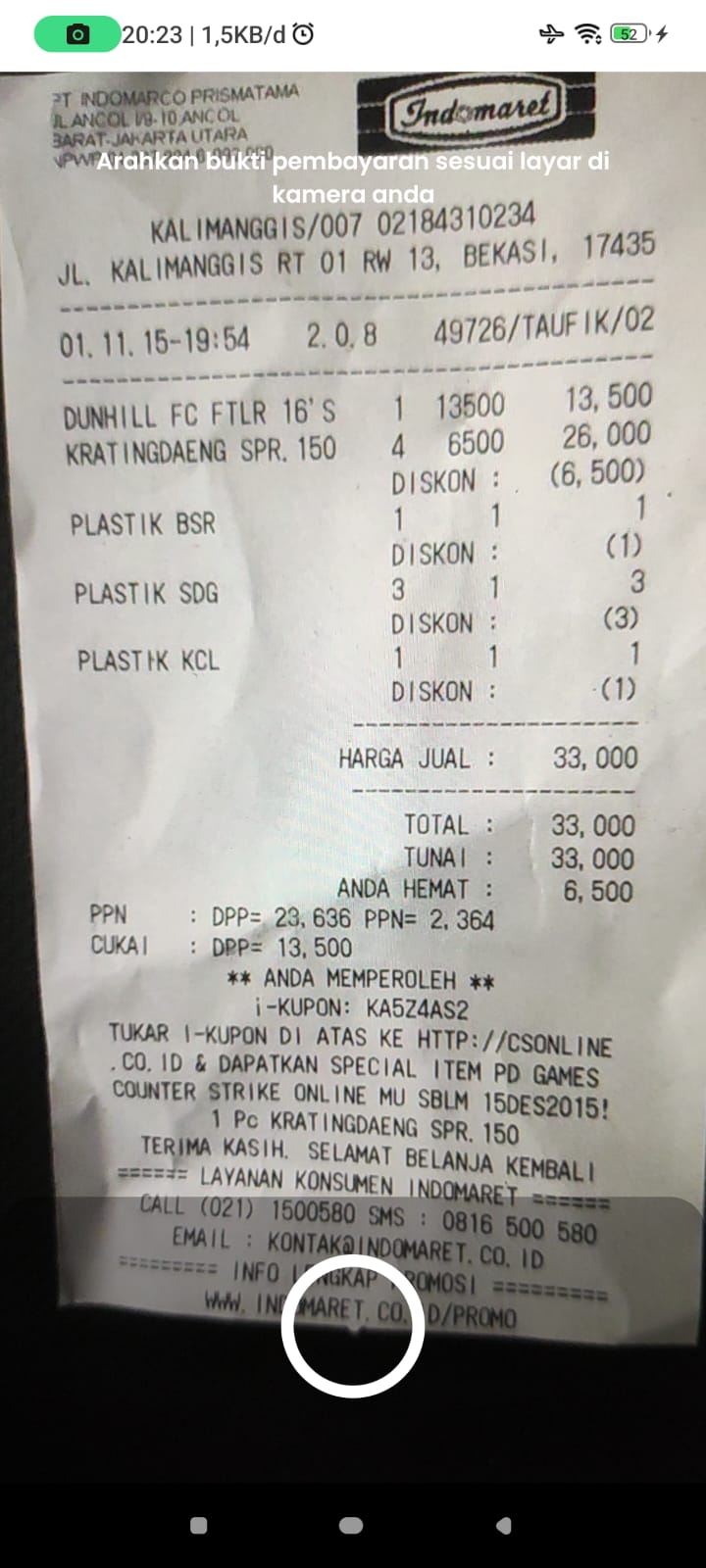
Graphical user interface, text, application

Description automatically generated

Gambar 4.8 Tampilan Halaman Input *Reimbursement* Karyawan

1. Tampilan Ambil Foto Struk

Pada gambar 4.9 menampilkan tampilan untuk mengambil foto struk, yang nantinya akan dimasukkan ke dalam form pengajuan *reimbursement* karyawan.



Gambar 4.9 Tampilan Halaman Ambil Foto Struk

1. Tampilan *Optical Character Recognition* (OCR) Berhasil Dilakukan

Pada gambar 4.10 menampilkan tampilan *Optical Character Recognition* (OCR) berhasil dilakukan, yang menunjukkan bahwa aplikasi berhasil membaca total harga dari gambar struk yang diambil gambarnya oleh karyawan.

Graphical user interface, text, application, chat or text message

Description automatically generated

### Gambar 4.10 Tampilan Kegunaan *Optical Character Recognition* (OCR)

1. Tampilan Pengajuan *Reimbursement* Karyawan di Aplikasi

Pada gambar 4.11 menampilkan tampilan data pengajuan reimbursement yang sudah masuk dan masih menunggu aksi dari staff admin.

Graphical user interface, text, application

Description automatically generated

### Gambar 4.11 Tampilan Pengajuan *Reimbursement* Karyawan di Aplikasi

1. Tampilan Detail Pengajuan Reimbursement

Pada gambar 4.12 menampilkan tampilan detail pengajuan reimburse karyawan, yang memberikan informasi lebih rinci tentang klaim pengeluaran yang diajukan.

### Graphical user interface, text, application Description automatically generated

### Gambar 4.12 Tampilan Detail Pengajuan *Reimbursement*

1. Tampilan Detail Foto Struk pada Pengajuan *Reimbursement*

Pada gambar 4.13 menampilkan tampilan foto detail struk yang diinputkan oleh karyawan pada form pengajuan *reimbursement*.

Text

Description automatically generated

Gambar 4.13 Tampilan Detail Foto Struk pada Pengajuan *Reimbursement*

1. Tampilan Data Pengajuan Berdasarkan Filter

Pada gambar 4.14 menampilkan tampilan data pengajuan berdasarkan filter yang dipilih yang memudahkan staff admin untuk memilah pengajuan-pengajuan yang masuk.

### A picture containing background pattern Description automatically generated

### Gambar 4.14 Tampilan Data Pengajuan Berdasarkan Filter

1. Tampilan Beranda dengan Pengajuan yang Ditolak oleh Staff Admin

Pada gambar 4.15 menampilkan tampilan pengajuan yang ditolak oleh staff admin yang menunjukkan bahwa klaim pengeluaran tersebut tidak akan mendapatkan penggantian.

Graphical user interface, text, application

Description automatically generated

### Gambar 4.15 Tampilan Beranda dengan Pengajuan yang Ditolak

1. Tampilan Detail Pengajuan yang Ditolak Beserta Alasannya

Pada gambar 4.16 menampilkan tampilan pengajuan yang disetujui oleh staff admin yang menunjukkan bahwa klaim pengeluaran tersebut akan mendapatkan penggantian.

### Graphical user interface, text, application, email Description automatically generated

### Gambar 4.16 Tampilan Detail Pengajuan yang Ditolak Beserta Alasannya

1. Tampilan Beranda dengan Pengajuan yang Disetujui Oleh Staff Admin

### Pada gambar 4.17 menampilkan tampilan pengajuan yang disetujui oleh staff admin yang menunjukkan bahwa klaim pengeluaran tersebut akan mendapatkan penggantian.

Graphical user interface, application

Description automatically generated

### Gambar 4.17 Tampilan Beranda dengan Pengajuan yang Disetujui

1. Tampilan Detail Pengajuan yang Disetujui Beserta Alasannya

Pada gambar 4.18 menampilkan tampilan detail pengajuan yang disetujui oleh staff admin yang memberikan informasi lebih rinci tentang klaim pengeluaran yang akan diberikan penggantian.

Graphical user interface, text, application

Description automatically generated

### Gambar 4.18 Tampilan Detail Pengajuan yang Disetujui Beserta Alasannya

1. Tampilan Beranda Staff Admin

Pada gambar 4.19 menampilkan tampilan beranda admin ketika ada pengajuan yang membutuhkan persetujuan, yang memberikan informasi tentang pengajuan yang masuk dan tindakan selanjutnya yang dapat dilakukan oleh staff admin.

Graphical user interface, text, application, chat or text message

Description automatically generated

Gambar 4.19 Tampilan Halaman Beranda Aplikasi Staff Admin

1. Tampilan Detail Reimburse Karyawan yang Membutuhkan Persetujuan

Gambar 4.20 menampilkan tampilan detail reimburse karyawan yang membutuhkan persetujuan, yang memberikan informasi lebih rinci tentang klaim pengeluaran yang diajukan oleh karyawan dan menunggu persetujuan dari staff admin.

Graphical user interface, text, application

Description automatically generated

### Gambar 4.20 Tampilan Detail Reimburse yang Membutuhkan Persetujuan

1. Tampilan untuk Memberikan Alasan Karena Reimburse Ditolak oleh Admin.

Pada gambar 4.21 menampilkan tampilan untuk memberikan alasan karena *reimburse* ditolak oleh admin, yang memungkinkan admin untuk memberikan penjelasan terkait penolakan yang diberikan kepada karyawan.

Graphical user interface, text, application

Description automatically generated

### Gambar 4.21 Tampilan untuk Memberikan Alasan Karena Reimburse Ditolak oleh Admin.

1. Tampilan untuk Memberikan Alasan Karena Reimburse Disetujui oleh Admin.

Pada gambar 4.22 menampilkan tampilan untuk memberikan alasan karena *reimburse* disetujui oleh admin, yang memungkinkan admin untuk memberikan penjelasan terkait persetujuan yang diberikan kepada karyawan.

Graphical user interface, application

Description automatically generated

## Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah elemen penting dalam pembangunan atau desain sistem. Tujuannya adalah memastikan bahwa sistem yang dibangun memiliki kualitas yang baik, yaitu mampu menggambarkan pokok-pokok dari analisis, desain, dan kode sistem tersebut. Metode pengujian *black box* digunakan dalam pengujian aplikasi *reimbursement* PT. Kayana Nusa ini.

Metode pengujian black box melibatkan pengamatan terhadap hasil eksekusi melalui data uji dan pemeriksaan fungsional perangkat lunak. Metode ini mengevaluasi hanya dari tampilan luar (*interface*) tanpa mengetahui apa yang terjadi di dalam proses detailnya. Berikut adalah pengujian yang dilakukan pada aplikasi *reimbursement* PT. Kayana Nusa Teknologi ini.

### Tabel 4.1 Pengujian Perangkat Lunak

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Deskripsi pengujian** | **Hasil yang Diharapkan** | **Hasil Pengamatan** | **Kesimpulan** |
| 1 | Form *Login* Dengan Strapi | Penguji akan berhasil *login* dan langsung diarahkan ke halaman *dashboard* | Penguji berhasil (valid) login dan langsung diarahkan ke halaman dashboard | Ok/sesuai |
| 2 | Tampilan *Dashboard Strapi* | Pada halaman *dashboard* akan terlihat beberapa menu seperti Karyawan, Perusahaan, dll | Pada halaman dashboard terdapat beberapa menu seperti Karyawan, Perusahaan, dll | Ok/sesuai |
| 3 | Tampilan Input Data Karyawan | Pada halaman ini, penguji dapat memasukkan data karyawan seperti nama, *email*, dan posisi | Pada halaman ini, terdapat *form input* untuk memasukkan data karyawan seperti nama, *email*, dan posisi | Ok/sesuai |
| 4 | Tampilan Data Karyawan | Pada halaman ini, terdapat daftar karyawan yang telah ditambahkan oleh *superadmin* | Pada halaman ini, terdapat daftar karyawan yang telah ditambahkan oleh *superadmin* dan siap untuk diberikan kepada karyawan PT. Kayana Nusa untuk bisa melakukan login pada aplikasi *reimbursement* | Ok/sesuai |
| 5 | Tampilan *Screening* Nama Perusahaan pada Aplikasi | Pada halaman ini, penguji dapat memasukan nama perusahaan pada aplikasi | Pada halaman ini, terdapat tampilan *screening* nama perusahaan pada aplikasi yang bisa di input sesuai | Ok/sesuai |
| 6 | Tampilan Login pada Aplikasi | Pada halaman ini, penguji dapat memasukkan *username* dan *password* untuk melakukan login pada aplikasi | Pada halaman ini, terdapat form input untuk memasukkan *username* dan *password* untuk melakukan login pada aplikasi | Ok/sesuai |
| 7 | Tampilan Beranda Aplikasi Reimbursement dengan *Role* Karyawan | Pada halaman ini, penguji dapat melihat daftar pengajuan reimbursement yang telah dibuat | Pada halaman ini, terdapat daftar pengajuan reimbursement yang telah dibuat | Ok/sesuai |
| 8 | Tampilan Form Reimbursement Karyawan | Pada halaman ini, penguji dapat mengisi *form* untuk membuat pengajuan *reimbursement* | Pada halaman ini, terdapat *form input* untuk membuat pengajuan *reimbursement* | Ok/sesuai |
| 9 | Tampilan Ambil Foto Struk | Pada halaman ini, penguji dapat mengambil foto struk sebagai bukti pembelian | Pada halaman ini, terdapat fitur untuk mengambil foto struk sebagai bukti pembelian | Ok/sesuai |
| 10 | Tampilan *Optical Character Recognition* Berhasil Dilakukan | Pada halaman ini, penguji dapat melihat bahwa sistem dapat membaca total harga dari gambar struk yang telah diambil | Pada halaman ini, terdapat tampilan bahwa sistem berhasil membaca total harga dari gambar struk yang telah diambil | Ok/sesuai |
| 11 | Tampilan Data Pengajuan *Reimbursement* yang Masih Status Menunggu Aksi Staff Admin. | Penguji dapat melihat daftar pengajuan reimbursement yang masih menunggu aksi dari staff admin. | Penguji dapat melihat daftar pengajuan reimbursement yang masih menunggu aksi dari staff admin. | Ok/sesuai |
| 12 | Tampilan Detail Pengajuan *Reimburse* Karyawan. | Penguji dapat melihat detail pengajuan *reimbursement* yang telah diajukan oleh karyawan dan status dari setiap pengajuannya. | Penguji dapat melihat detail pengajuan *reimbursement* yang telah diajukan oleh karyawan dan status dari setiap pengajuannya. | Ok/sesuai |
| 13 | Tampilan Foto Detail Struk yang di Input Karyawan. | Penguji dapat melihat foto struk yang telah diambil oleh karyawan dan telah diunggah ke dalam aplikasi. | Penguji dapat melihat foto struk yang telah diambil oleh karyawan dan telah diunggah ke dalam aplikasi. | Ok/sesuai |
| 14 | Tampilan Data Pengajuan Berdasarkan Filter yang di Pilih. | Penguji dapat memilih filter tertentu untuk melihat daftar pengajuan *reimbursement* yang telah disetujui atau ditolak. | Penguji dapat memilih filter tertentu untuk melihat daftar pengajuan *reimbursement* yang telah disetujui atau ditolak. |  |
| 15 | Filter Data Pengajuan Reimburse | Penguji akan memilih filter data pengajuan reimburse berdasarkan kategori tertentu | Penguji berhasil memilih filter dan melihat data pengajuan yang sesuai dengan filter yang dipilih. | Ok/sesuai |
| 16 | Menampilkan Pengajuan yang Ditolak | Penguji akan melihat daftar pengajuan *reimburse* yang telah ditolak oleh staf admin. | Penguji melihat daftar pengajuan yang ditolak oleh staf admin. | Ok/sesuai |
| 17 | Menampilkan Detail Pengajuan yang Ditolak | Penguji akan melihat detail pengajuan *reimburse* yang telah ditolak oleh staff admin beserta alasan penolakan. | Penguji melihat detail pengajuan yang ditolak oleh staff admin beserta alasan penolakan. | Ok/sesuai |
| 18 | Menampilkan Pengajuan yang Disetujui | Penguji akan melihat daftar pengajuan reimburse yang telah disetujui oleh staff admin. | Penguji melihat daftar pengajuan yang disetujui oleh staff admin. | Ok/sesuai |
| 19 | Menampilkan Detail Pengajuan yang Disetujui | Penguji akan melihat detail pengajuan reimburse yang telah disetujui oleh staff admin beserta informasi persetujuan. | Penguji melihat detail pengajuan yang disetujui oleh staff admin beserta informasi persetujuan. | Ok/sesuai |
| 20 | Menampilkan Beranda Admin dengan Pengajuan Baru | Penguji akan melihat halaman beranda admin ketika ada pengajuan *reimburse* baru yang perlu diproses persetujuan. | Penguji melihat halaman beranda admin ketika ada pengajuan *reimburse* baru yang perlu diproses persetujuan. | Ok/sesuai |
| 21 | Menampilkan Detail Pengajuan yang Perlu Disetujui | Penguji akan melihat detail pengajuan *reimburse* yang perlu disetujui oleh staff admin beserta informasi pengajuan dan pemohon. | Penguji melihat detail pengajuan *reimburse* yang perlu disetujui oleh staff admin beserta informasi pengajuan dan pemohon. | Ok/sesuai |
| 22 | Memberikan Alasan karena *Reimburse* Disetujui | Penguji akan memberikan alasan kenapa pengajuan *reimburse* disetujui oleh staf admin pada *form* yang tersedia. | Penguji berhasil memberikan alasan kenapa pengajuan *reimburse* disetujui oleh staf admin pada *form* yang tersedia. | Ok/sesuai |
| 23 | Memberikan Alasan karena *Reimburse* Ditolak | Penguji akan memberikan alasan kenapa pengajuan *reimburse* ditolak oleh staff admin pada form yang tersedia. | Penguji berhasil memberikan alasan kenapa pengajuan *reimburse* ditolak oleh staff admin pada form yang tersedia | Ok/sesuai |

# BAB V

# PENUTUP

## Kesimpulan

Penerapan *library Tesseract Optical Character Recognition* dalam fitur *reimbursement* berbasis *android* ini dirancang menggunakan metode *Kanban*. Di mana pada tahap visualisasi proses dan menetapkan batasan pekerjaan yang dilakukan lalu mengelola aliran proses serta membuat aturan proses dan dengan dilakukannya pembaruan berkelanjutan pada setiap prosesnya sangatlah efisien terhadap suatu pengembangan sistem. Perancangan sistem menggunakan diagram-diagram *Unified Modeling Language* (UML) dan untuk pengimplementasian dari rancangan tersebut penulis menggunakan bahasa pemrograman *Javascript*, dan database *PostgreSQL* sebagai media atau tempat penyimpanan data dan menggunakan Strapi sebagai *Content Management System*. Kesimpulan dari perancangan sistem informasi gereja ini adalah sebagai berikut:

1. Dengan dirancangnya aplikasi *reimburse* sesuai dengan format pengajuan yang umum digunakan dan di persingkat dengan menggunakan aplikasi android serta bantuan teknologi OCR. Hal ini akan mempercepat proses pengajuan reimburse dan menghemat waktu bagi para pengguna.
2. Dengan diterapkannya aplikasi *reimburse* dapat memberikan inovasi dalam bentuk penerapan proses reimbursement yang tersistemasi sehingga data yang dihasilkan lebih valid dan terhindar dari manipulasi. Dengan cara ini, dapat memastikan keamanan data dan mencegah terjadinya kecurangan dalam proses pengajuan *reimburse*.
3. Dengan dibangunnya aplikasi *reimburse* dapat meningkatkan efisiensi dan kecepatan proses pengajuan reimburse, serta mengurangi penggunaan kertas dan memperkuat validitas data yang dihasilkan.

## Saran

Adapun saran yang dapat disampaikan terhadap sistem aplikasi android *reimbusement* PT. Kayana Nusa ini adalah sebagai berikut:

1. Pihak PT. Kayana Nusa dapat lebih lagi mengembangkan sistem yang telah dibangun baik dari segi tampilan maupun penambahan fitur pada aplikasi.
2. Dapat menambahkan fitur fitur terbaru yang ada pada PT. Kayana Nusa sehingga menambah fungsionalitas dan menjadi lebih maju kedepan nya.
3. Melakukan pemeliharaan dan pembaruan aplikasi tersebut agar terus *up to* *date* sesuai dengan kebutuhan.

# DAFTAR PUSTAKA

Effendi, M. M., Permana, A. Y., & Nawangsih, I. (2019). Penerapan Metode Ekstrasi Image Ke Text Dengan Optical Character Recognition Untuk Otomatisasi Data Kependudukan. Simposium Nasional Ilmiah & Call for Paper Unindra (Simponi), 1(1).

Mulyanto, A., Susanti, E., Rossi, F., Wajiran, W., & Borman, R. I. (2021). Penerapan Convolutional Neural Network (CNN) pada Pengenalan Aksara Lampung Berbasis Optical Character Recognition (OCR). JEPIN (Jurnal Edukasi Dan Penelitian Informatika), 7(1), 52-57.

Shalahuddin, M., & Rosa. (2018). Rekayasa Perangkat Lunak Terstruktur. Bandung: Informatika.

Sonita, A., & Khairunnisyah, K. (2018). Aplikasi pendeteksi obat dan makanan menggunakan ocr (optical character recognition). Jurnal Informatika Upgris, 4(1).

Naibaho, L. S. (2020). Penerapan Optical Character Recognition Pada Dokumen Surat Keputusan Menggunakan Algoritma Reduced Support Vector Machines (Doctoral dissertation, Universitas Komputer Indonesia).

JULIANTO, S. (2022). PENERAPAN OPTICAL CHARACTER RECOGNITION (OCR) UNTUK PENJADWALAN ACARA DENGAN GOOGLE CLOUD VISION API ALGORITMA CONVOLUTION NEURAL NETWORK (CNN) (Doctoral dissertation, UPN" Veteran" Yogyakarta).

Yaro, I. K., & Adeiza, M. O. (2021). Impact of Taxation on Economic Growth and Development in Nigeria: A Review. IOSR Journal of Humanities and Social Science, 26(6), 41-45.

Matei, O., Pop, P. C., & Vălean, H. (2013). Optical character recognition in real environments using neural networks and k-nearest neighbor. Applied intelligence, 39, 739-748.

Supriyade, S., Listiyoko, L., & Fahrudin, A. (2020). Sistem Pendeteksi Ketinggian Air Menggunakan Internet of Things Berbasis Android Untuk Memberikan Informasi Data Ketinggian Air Melalui Notifikasi Email. Jurnal Komputer dan Informatika, 15(1), 260-273.

Anderson, D. J., Concas, G., Lunesu, M. I., Marchesi, M., & Zhang, H. (2012). A comparative study of Scrum and Kanban approaches on a real case study using simulation. In Agile Processes in Software Engineering and Extreme Programming: 13th International Conference, XP 2012, Malmö, Sweden, May 21-25, 2012. Proceedings 13 (pp. 123-137). Springer Berlin Heidelberg.

Anderson, D. J., & Carmichael, A. (2016). Essential kanban condensed. Blue Hole Press.

Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2015). Pengujian aplikasi menggunakan black box testing boundary value analysis (studi kasus: Aplikasi prediksi kelulusan smnptn). Jurnal Ilmiah Teknologi Infomasi Terapan, 1(3).

Obe, R. O., & Hsu, L. S. (2017). PostgreSQL: up and running: a practical guide to the advanced open source database. " O'Reilly Media, Inc.".

Nguyen, A. (2022). Building an E-commerce Website Using Next Js, Mantine, and Strapi.

Eisenman, B. (2015). Learning react native: Building native mobile apps with JavaScript. " O'Reilly Media, Inc.".

Tilkov, S., & Vinoski, S. (2010). Node. js: Using JavaScript to build high-performance network programs. IEEE Internet Computing, 14(6), 80-83.

Bree, R. T., & Gallagher, G. (2016). Using Microsoft Excel to code and thematically analyse qualitative data: a simple, cost-effective approach. All Ireland Journal of Higher Education, 8(2).

# LAMPIRAN

A couple of men using laptops

Description automatically generated with low confidenceA person holding a phone

Description automatically generated with medium confidence

