

BAB I

PENDAHULUAN

A. Latar Belakang

Umumnya, penghitungan tagihan listrik dilakukan secara manual oleh petugas yang melakukan pengecekan dan pencatatan meteran listrik di berbagai lokasi pelanggan. Mereka melakukan pemeriksaan dan mencatat angka meteran listrik secara langsung. Namun, metode manual ini memiliki risiko kesalahan manusia yang dapat menyebabkan ketidakakuratan dalam penghitungan tagihan dan meningkatkan potensi terjadinya kecurangan.

Petugas yang bertanggung jawab mencatat angka meteran listrik dapat dengan sengaja memanipulasi angka yang tercatat untuk keuntungan pribadi. Mereka dapat menambahkan atau mengurangi angka pada meteran dengan tujuan untuk mengurangi atau meningkatkan tagihan pelanggan. Begitu pula kesalahan manusia dalam mencatat angka meteran listrik bisa terjadi. Misalnya, petugas dapat salah menulis angka, mengalami kesalahan pengetikan, atau mengalami kekeliruan saat mencatat angka meteran. Hal ini dapat menghasilkan ketidakakuratan dalam penghitungan tagihan.

Petugas yang melakukan penghitungan tagihan secara manual mungkin tidak sepenuhnya teliti dalam mencatat angka meteran. Mereka dapat melewati atau mengabaikan beberapa digit atau membaca angka dengan tidak jelas, yang berpotensi menyebabkan kesalahan dalam perhitungan tagihan. Selain itu, petugas

yang bertugas mencatat angka meteran listrik mungkin tidak melakukan pencatatan secara teratur atau tepat waktu. Hal ini dapat menyebabkan angka meteran yang tercatat tidak mencerminkan konsumsi energi yang sebenarnya dalam periode tertentu, sehingga menghasilkan tagihan yang tidak akurat.

Semua contoh di atas menunjukkan risiko kesalahan manusia dan potensi kecurangan dalam penghitungan tagihan listrik secara manual. Inilah mengapa otomatisasi menggunakan teknologi seperti OCR menjadi penting untuk meningkatkan akurasi, efisiensi, dan transparansi dalam proses tersebut.

Dengan menggunakan teknologi *Optical Character Recognition* (OCR), dapat dilakukan otomatisasi dalam pengenalan karakter pada meteran listrik. OCR adalah teknologi yang memungkinkan komputer untuk membaca dan mengenali karakter yang tercetak atau ditulis tangan dalam bentuk gambar atau teks.

Implementasi OCR dalam penghitungan tagihan listrik dapat memberikan beberapa manfaat. Pertama, dengan otomatisasi ini, proses penghitungan tagihan dapat dilakukan dengan lebih cepat dan efisien, mengurangi waktu dan biaya yang dibutuhkan dalam proses tersebut. Kedua, penggunaan OCR dapat mengurangi tingkat kesalahan manusia dalam mencatat angka meteran listrik, sehingga meningkatkan akurasi tagihan yang diterima oleh pelanggan. Ketiga, dengan adanya sistem otomatis ini, dapat tercipta transparansi yang lebih tinggi dalam proses penghitungan tagihan, mengurangi peluang adanya manipulasi atau kecurangan.

Dengan demikian, implementasi OCR dalam penghitungan tagihan listrik dapat memberikan manfaat signifikan dalam efisiensi operasional, akurasi, dan transparansi dalam penyediaan layanan listrik kepada pelanggan.

B. Rumusan Masalah

Berdasarkan uraian dari latar belakang diatas, peneliti dapat mengambil rumusan masalah yang akan dilakukan yaitu

1. Bagaimana mengimplementasikan teknologi *Optical Character Recognition* (OCR) dalam proses penghitungan tagihan listrik?
2. Apa manfaat yang dapat diperoleh dengan mengotomatiskan penghitungan tagihan listrik menggunakan OCR?
3. Apakah implementasi OCR dapat meningkatkan efisiensi dan akurasi dalam penghitungan tagihan listrik?
4. Bagaimana tingkat transparansi dalam proses penghitungan tagihan listrik dapat ditingkatkan dengan menggunakan OCR?
5. Bagaimana cara mengatasi potensi kesalahan manusia dan kecurangan dalam penghitungan tagihan listrik secara manual?

C. Tujuan Penelitian

1. Menerapkan teknologi *Optical Character Recognition* (OCR) dalam penghitungan tagihan listrik untuk mengotomatiskan proses tersebut.
2. Meningkatkan efisiensi operasional dalam penghitungan tagihan listrik dengan mengurangi waktu dan tenaga yang dibutuhkan melalui penggunaan OCR.
3. Meningkatkan akurasi penghitungan tagihan listrik dengan mengeliminasi kesalahan manusia yang mungkin terjadi dalam proses manual.
4. Meningkatkan transparansi dalam proses penghitungan tagihan listrik melalui penggunaan OCR untuk mengurangi potensi manipulasi atau kecurangan.
5. Mengatasi masalah kesalahan manusia dan potensi kecurangan dalam penghitungan tagihan listrik secara manual melalui implementasi OCR.

D. Batasan Masalah

Batasan masalah dalam hal ini adalah penelitian ini akan difokuskan pada penerapan teknologi OCR sebagai metode otomatisasi dalam pengenalan karakter pada meteran listrik untuk penghitungan tagihan listrik. Penelitian ini tidak akan secara rinci membahas keterbatasan teknologi OCR, seperti pengenalan karakter yang tidak sempurna pada kondisi tertentu, atau masalah dengan jenis font atau penulisan khusus.

E. Manfaat Penelitian

Manfaat dari aplikasi penghitungan tagihan listrik dengan metode OCR yaitu,

a. Manfaat bagi Penulis

Penulis akan mendapatkan pemahaman yang lebih mendalam tentang teknologi *Optical Character Recognition* (OCR) dan implementasinya dalam konteks penghitungan tagihan listrik. Mereka akan mempelajari prinsip kerja OCR, algoritma pengenalan karakter, serta aspek teknis dan praktis dalam mengimplementasikan OCR.

b. Manfaat bagi Pengguna

Dengan menggunakan teknik OCR, pengguna akan mendapatkan pengalaman yang lebih baik dalam hal penghitungan tagihan listrik, termasuk akurasi yang lebih tinggi, transparansi, efisiensi, dan layanan pelanggan yang ditingkatkan.

c. Manfaat bagi Universitas

Penelitian ini dapat memperkaya kurikulum pendidikan dengan mengintegrasikan topik-topik terkait OCR, otomasi, dan pengelolaan energi dalam program studi yang relevan. Dan juga meningkatkan kualitas dan keterampilan sumber daya manusia yang dihasilkan oleh universitas.

BAB II

TINJAUAN PUSTAKA

A. Kajian Hasil Penelitian

Kajian penelitian terdahulu ini memberikan pemahaman tentang penggunaan OCR dalam konteks penghitungan tagihan listrik. Penelitian-penelitian tersebut menunjukkan manfaat implementasi OCR, termasuk peningkatan akurasi, efisiensi, dan transparansi dalam proses penghitungan tagihan listrik. Implementasi OCR tidak terbatas pada penghitungan tagihan listrik, melainkan dapat diterapkan dalam berbagai konteks dan bidang untuk mengenali dan mengolah teks dalam berbagai bentuk data.

Penelitian sebelumnya dilakukan oleh Ali Firdaus dengan judul “Implementasi *Optical Character Recognition* (OCR) Pada Masa Pandemi Covid-19”[1]. Penelitian ini bertujuan untuk memproses dan memindai dokumen secara digital. Terutama dalam situasi pandemi seperti saat ini, di mana dokumen fisik sering kali harus diolah ulang menggunakan aplikasi pengolahan kata. Program OCR yang dibuat berhasil digunakan untuk memindai atau memindai dokumen fisik jika dokumen tersebut hilang atau rusak, dan dapat menghemat waktu dalam pembuatan, pengolahan, dan pengetikan dokumen.

Penelitian lain juga dilakukan oleh Aldi Setiawan dengan judul “Implementasi *Optical Character Recognition* (OCR) pada Mesin Penerjemah Bahasa Indonesia ke Bahasa Inggris”[2]. Hasil pengujian *black box* menunjukkan bahwa aplikasi tidak dapat digunakan dalam posisi landscape, namun dapat berjalan dengan baik

dalam pengenalan kata dan tanda baca ketika menggunakan inputan satu atau dua kata tanpa huruf kapital dan tanda baca. Pada pengujian kalimat, aplikasi dapat menangkap kata-kata dengan baik dalam satu kalimat, tetapi tidak dapat menerjemahkan kata dengan huruf kapital dan tanda baca.

Kristina Apriyanti juga melakukan penelitian dengan judul “Implementasi *Optical Character Recognition* Berbasis *Backpropagation* untuk *Text to Speech* Perangkat Android”[3]. Dalam penelitian ini, dilakukan pengembangan sistem yang menggunakan pemrosesan citra digital untuk mengenali dan mengekstraksi kata-kata dari gambar. Hasil pengolahan citra ini kemudian digunakan sebagai input untuk aplikasi teks ke suara, sehingga pengguna dapat mendengar kata-kata yang diinginkan tanpa harus mengetikkannya secara manual. Penelitian ini menunjukkan kemajuan dalam pengembangan aplikasi teks ke suara dengan memanfaatkan pemrosesan citra digital sebagai sistem input kata. Hipotesis penelitian ini dapat memberikan kontribusi dalam mempermudah pengguna dalam menggunakan aplikasi teks ke suara melalui metode input yang lebih efisien dan intuitif.

Berikut adalah tabel perbedaan penelitian, penting untuk membandingkan dan menganalisis perbedaan antara penelitian yang akan dilakukan dengan penelitian-penelitian terdahulu yang telah disebutkan sebelumnya. Berikut adalah perbedaan-perbedaan tersebut yang dilampirkan kedalam table *State Of The Art*:

Tabel 2. 1. State Of TheArt Penelitian

No	Judul Penelitian dan Nama Peneliti	Tahun Penelitian	Objek dan Permasalahan	Perbedaan
1	Judul : Implementasi <i>Optical Character Recognition</i> (OCR) Pada Masa Pandemi Covid-19 Nama Peneliti: Ali Firdaus	2021	Objek : Dokumen Permasalahan : Pemindaian dokumen fisik menggunakan teknologi OCR	Pada penelitian yang diusulkan berfokus pada implementasi OCR untuk otomasi penghitungan tagihan listrik, sementara abstrak yang disediakan berfokus pada penggunaan OCR untuk memindai dan mengolah dokumen fisik secara digital.
2	Judul: Implementasi <i>Optical Character Recognition</i> (OCR) pada Mesin Penerjemah Bahasa Indonesia ke Bahasa Inggris Nama Peneliti : Aldi Setiawan	2017	Objek : Kalimat dalam bentuk lembar teks sebagai materi pengujian. Permasalahan : Memanfaatkan OCR untuk memudahkan pengguna dalam memasukkan kata-kata yang akan diterjemahkan	Penelitian yang dibuat berfokus pada pengembangan OCR pada mesin penerjemah bahasa Indonesia ke bahasa Inggris dengan menggunakan platform Android. Sedangkan penelitian yang diusulkan berfokus pada implementasi OCR untuk otomasi penghitungan tagihan listrik.

Lanjutan Tabel 2. 1.

No	Judul Penelitian dan Nama Peneliti	Tahun Penelitian	Objek dan Permasalahan	Perbedaan
	<i>Backpropagation</i> untuk <i>Text to Speech</i> Perangkat Android Nama Peneliti : Kristina Apriyanti	2021	Permasalahan : Menggunakan pemrosesan citra digital untuk mengenali dan mengekstraksi kata-kata dari gambar	Penelitian OCR yang diusulkan untuk menghitung tagihan listrik berfokus pada otomatisasi penghitungan tagihan listrik menggunakan OCR. Penelitian ini bertujuan untuk menggantikan proses manual dalam penghitungan tagihan listrik dengan menggunakan teknologi OCR. Sementara itu, penelitian ini fokus pada pengembangan sistem input kata untuk aplikasi teks ke suara menggunakan pemrosesan citra digital.
4	Judul: Implementasi Algoritma Yolo Dan <i>Tesseract Ocr</i> Pada Sistem Deteksi Plat Nomor Otomatis[4] Nama Peneliti : Imam Husni	2022	Objek : Plat Kendaraan Permasalahan : Penelitian ini mengembangkan sistem deteksi otomatis plat nomor kendaraan berbasis YOLO. Dataset yang digunakan adalah	Penelitian ini berkaitan dengan deteksi dan pengenalan plat nomor kendaraan. Sedangkan penelitian OCR menghitung tagihan listrik berkaitan dengan pemindaian dan pengolahan

Lanjutan Tabel 2. 1.

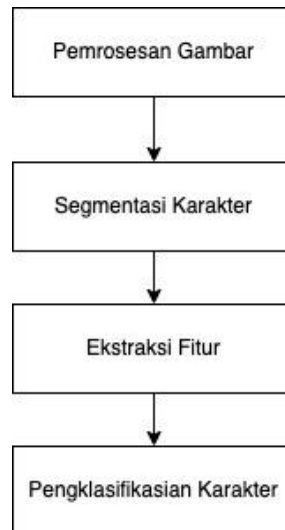
No	Judul Penelitian dan Nama Peneliti	Tahun Penelitian	Objek dan Permasalahan	Perbedaan
			model YOLOv3 yang telah dilatih sebanyak 700 data. Selanjutnya, dilakukan proses ekstraksi teks plat nomor menggunakan perpustakaan <i>Optical Character Recognition</i> (OCR) Tesseract, dan hasilnya akan disimpan dalam database.	dokumen fisik untuk penghitungan tagihan listrik.

B. Optical Character Recognition (OCR)

Optical Character Recognition (OCR) adalah suatu teknologi yang digunakan untuk mengenali dan mengubah teks yang tercetak atau ditulis tangan menjadi bentuk yang dapat dikenali oleh komputer. OCR memanfaatkan algoritma dan teknik pengolahan citra untuk mengidentifikasi karakter-karakter dalam gambar atau dokumen dan mengonversinya menjadi teks yang dapat diproses lebih lanjut.

Teknologi OCR digunakan secara luas dalam berbagai industri, termasuk layanan keuangan, medis, pemerintahan, dll. Contoh aplikasi dari OCR antara lain pemrosesan formulir kredit di bank, pencarian data pasien di rumah sakit, serta pengolahan dokumen hukum di kantor pemerintah.

Secara umum, proses pengenalan karakter yang dilakukan oleh OCR disajikan pada diagram di bawah ini:



Gambar 2. 1. Proses Pengenalan Karakter OCR

1. Pemrosesan Gambar: Gambar atau dokumen yang akan dikenali harus diproses terlebih dahulu. Langkah ini meliputi pemotongan gambar, pemulusan, normalisasi, peningkatan kontras, dan penghapusan noise untuk memperoleh citra yang lebih jelas dan optimal.
2. Segmentasi Karakter: Setelah gambar diproses, langkah selanjutnya adalah memisahkan karakter-karakter individu dalam gambar. Segmentasi karakter dilakukan dengan memanfaatkan teknik pemisahan objek berdasarkan perbedaan intensitas, bentuk, atau tekstur.

Proses selanjutnya adalah normalisasi, yang di dalamnya masih terdapat 2 proses lagi yaitu:

- a. *Scaling*, adalah fungsi untuk mengubah ukuran suatu gambar dimana *scaling* merupakan istilah yang cenderung untuk memperbesar gambar, dan *shrink* untuk memperkecil gambar.

- b. *Thinning*, adalah operasi morfologi yang digunakan untuk menghapus piksel *foreground* yang terpilih dari gambar biner, biasanya digunakan untuk proses mencari tulang dari sebuah objek.
- 3. Ekstraksi Fitur: Setelah karakter tersegmentasi, fitur-fitur kunci dari setiap karakter diekstraksi. Fitur-fitur ini dapat mencakup bentuk, ukuran, orientasi, dan pola piksel karakter. Fitur-fitur ini digunakan untuk membedakan dan mengklasifikasikan karakter.
- 4. Pengklasifikasian Karakter: Langkah terakhir adalah pengklasifikasian karakter. Dalam langkah ini, karakter-karakter yang telah tersegmentasi dan fitur-fiturnya dievaluasi menggunakan metode klasifikasi seperti jaringan saraf tiruan, metode pembelajaran mesin, atau pendekatan pengenalan pola lainnya. Hasilnya adalah teks yang dihasilkan dari karakter-karakter yang dikenali.

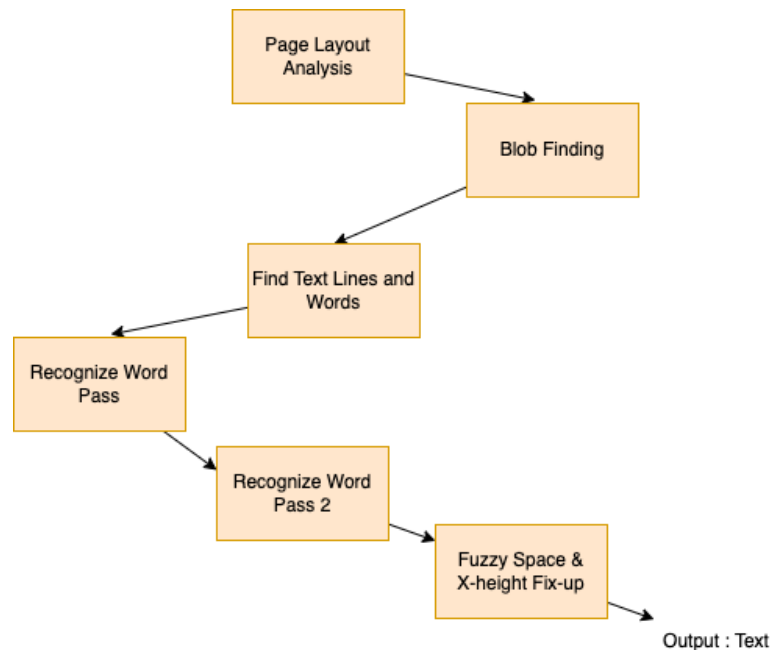
Secara keseluruhan, OCR menggunakan teknik pengolahan citra, analisis karakter, dan pengenalan pola untuk mengenali karakter-karakter dalam gambar atau dokumen dan mengonversinya menjadi teks yang dapat diproses lebih lanjut.

C. Tesseract OCR

Tesseract adalah *engine opensource* OCR yang awalnya dikembangkan HP (*Hewlett-Packard*) antara tahun 1984 dan 1994. *Tesseract* dimulai dari sebuah proyek penelitian PhD di HP Laboratorium, Bristol oleh Ray Smith. Setelah penelitian bersama antara HP Labs Bristol dan Scanner HP divisi di Colorado, *tesseract* secara signifikan memimpin dalam akurasi atas mesin komersial tetapi tidak menjadi produk.

Tahap perkembangan berikutnya kembali di HP Lab Bristol sebagai investigasi OCR untuk kompresi. Pada akhir proyek ini, pada tahun 1994, pengembangan berhenti sepenuhnya. Mesin ini dikirim ke UNLV (*University Nevada Las Vegas*) pada tahun 1995 menjalani tes akurasi tahunan OCR.

Tesseract OCR mengasumsikan input yang diterima berupa sebuah *binary image*. Pertama, analisis dilakukan pada komponen terhubung/*Connected Component* (CC) untuk menemukan di mana *outline* komponen disimpan. Pada tahap ini *outlines* dikumpulkan bersama menjadi *blob*. *Blob* disusun menjadi baris teks, sedangkan garis dan *region* dianalisis untuk *pitch* tetap dan teks *proporsional*. Baris teks dipecah menjadi kata-kata berbeda menurut jenis spasi karakter. Teks dengan *pitch* tetap dibagi menjadi sel-sel karakter. Teks proporsional dipecah menjadi kata-kata dengan menggunakan ruang pasti dan ruang *fuzzy*. Pengenalan kata pada *image* dilakukan pada dua tahap proses yang disebut *pass-two*.



Gambar 2. 2. Arsitektur Tesseract OCR

Pada *pass* pertama dilakukan untuk mengenali masing-masing kata pada gilirannya. Kata-kata yang sukses pada *pass* pertama yaitu kata-kata yang terdapat di kamus dan tidak ambigu kemudian diteruskan ke *adaptive classifier* sebagai datapelatihan. Begitu *adaptive classifier* memiliki sampel yang cukup, *adaptive classifier* ini dapat memberikan hasil klasifikasi bahkan pada *pass* pertama. Proses *pass* kedua dilakukan untuk mengenali kata-kata yang mungkin saja kurang dikenali atau terlewatkan pada *pass* pertama, pada tahap ini *adaptive classifier* telah memperoleh informasi lebih dari *pass* pertama. Tahap terakhir menyelesaikan ruang *fuzzy* dan memeriksa hipotesis alternatif pada ketinggian-x untuk mencari teks dengan *smallcap*.

Tesseract dirancang untuk mengenali teks putih di atas latar hitam dan teks hitam di atas latar putih. Hal ini menyebabkan rancangan mengarah pada analisis

komponen terhubung/*connected component* (CC) dan operasi pada *outline* komponen. Langkah pertama setelah analisis CC ialah menemukan *blob* pada *region* teks. Sebuah *blob* merupakan unit putatif yang dapat diklasifikasikan, yang mana bisa satu atau lebih komponen-komponen yang saling tumpang tindih secara horizontal.

Menurut Smith ada beberapa langkah yang dilakukan oleh *tesseract* untuk pengenalan karakter adalah sebagai berikut :

a. Pencarian Teks Line dan Kata

Algoritma *line finding* dirancang supaya halaman yang miring dapat dikenali tanpa harus *de-skew* (proses untuk mengubah halaman yang miring menjadi tegak lurus) sehingga tidak menurunkan kualitas gambar. Kunci bagian proses ini adalah *blob filtering* dan *line construction*.

Filtered blob lebih cenderung cocok dengan model *non-overlapping, parallel*, tetapi berupa garis-garis miring (*sloping line*). Pemrosesan *blob* oleh koordinat x memungkinkan untuk menetapkan *blob* ke sebuah baris teks yang unik. Sementara penelusuran kemiringan di seluruh halaman, dengan banyak mengurangi bahaya penugasan ke baris teks yang salah dengan adanya kemiringan (*skew*). Setelah *blob* tersaring ditetapkan ke garis, sebuah median terkecil dari kotak-kotak yang cocok digunakan untuk memperkirakan *baseline*, dan *blob* yang sudah difilter dengan baik dipasang kembali ke garis yang sesuai.

Langkah terakhir dari proses pembuatan garis (*line creation*) adalah menggabungkan *blob* yang *overlapping*, menempatkan *diacritical marks*

dengan dasar yang tepat, dan menghubungkan bagian-bagian dari beberapa karakter yang rusak secara benar.

b. *Baseline Fitting*

Setelah baris teks telah ditemukan, garis pangkal (*baseline*) dicocokkan secara lebih tepat menggunakan *quadratic spline*. Hal ini merupakan salah satu kelebihan sistem OCR dan memungkinkan *tesseract* untuk menangani halaman dengan garis pangkal (*baseline*) yang miring.

Baseline dicocokkan oleh partisi *blob* menjadi beberapa kelompok dengan sebuah perpindahan kontinu yang cukup layak untuk garis pangkal lurus yang asli. *Quadratic spline* dicocokkan ke partisi yang paling padat (diasumsikan sebagai *baseline*) dengan kuadrat terkecil. *Quadratic spline* memiliki keuntungan bahwa perhitungan ini cukup stabil tetapi merugikan jika muncul diskontinuitas ketika beberapa segmen *spline* diperlukan. Dalam hal ini, *cubic spline* bekerja lebih baik[6].



Gambar 2. 3. Contoh halaman dengan *baseline* miring

c. *Perkiraan Ketinggian X Pada Teks*

Setelah menemukan baris teks dan menyusun blok *blob* menjadi baris-baris, *Tesseract* mengestimasi ketinggian-x untuk setiap baris teks. Pertama, algoritma estimasi ketinggian-x menentukan batas-batas maksimum dan minimum dari ketinggianx yang dapat diterima berdasarkan ukuran garis inisial yang dihitung untuk blok. Kemudian, setiap baris secara terpisah,

ketinggian *bounding box blob* terjadi pada garis dikuantisasi dan dikumpulkan menjadi sebuah histogram. Dari histogram ini, algoritma pencarian ketinggian-x mencari ketinggian dua *mode* yang paling sering terjadi yang cukup jauh terpisah untuk menjadi ketinggian-x dan ketinggian-*ascender*. Untuk mengantisipasi *noise*, algoritma memastikan *mode* ketinggian yang diambil menjadi ketinggian-x dan ketinggian-*ascender* memiliki jumlah yang cukup atau kejadiankejadian relatif terhadap jumlah keseluruhan *blob* pada baris.

d. *Chopping* atau Pemotongan Karakter

Tesseract menguji garis teks (*text line*) untuk menentukan apakah mereka merupakan *fixed pitch*. Bila ditemukan *fixed pitch text*, *tesseract* memotong kata-kata menjadi karakter-karakter[6].

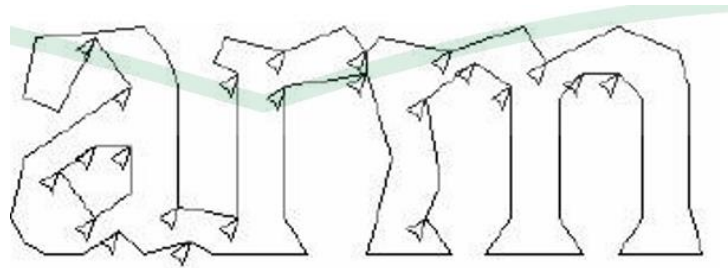


Gambar 2. 4. Pemotongan Karakter

e. Pemisahan Karakter Terhubung

Apabila hasil dari pengenalan kata tidak memuaskan, *tesseract* berusaha untuk memperbaiki hasil dengan memisahkan *blob* dengan keyakinan terburuk dari pengklasifikasian (*classifier*) karakter. Kandidat untuk titik-titik pemisahan ditemukan dari simpul cekung dari pendekatan poligonal *outline* dan mungkin saja terdapat titik cekung berlawanan lainnya atau segmen garis. Ini akan

menghabiskan sampai 3 pasang titik pemotongan untuk memisahkan karakter yang terhubung dari set ASCII[6].

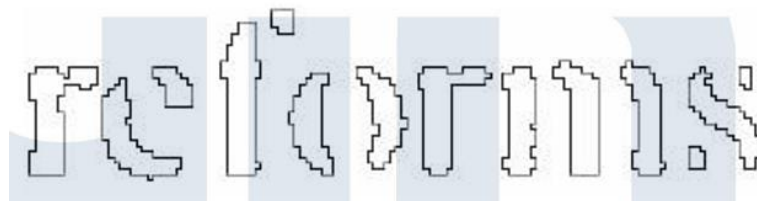


Gambar 2. 5. Kandidat Titik Potong

Gambar 2.5 di atas menunjukkan satu set calon titik potong (*chop points*) dengan tanda panah dan potongan terpilih sebagai sebuah garis melintasi kerangka dimana huruf 'r' bersentuhan dengan 'm'.

f. Asosiasi Karakter Patah

Ketika potongan yang potensial tidak ada lagi, ketika kata tersebut masih belum cukup baik, hal ini diberikan kepada *associator*. *Associator* membuat pencarian A* (*best first search*) dari segmentasi grafik yang mungkin kombinasi dari *blob* yang dipotong secara maksimal ke dalam kandidat karakter. Ketika A* segmentation digunakan untuk diimplementasikan pertama kali pada tahun 1989, akurasi *tesseract* terhadap karakter yang rusak cukup baik yang menjadikan *tesseract* mesin komersial pada saat itu.



Gambar 2. 6. Sebuah kata rusak bisa dikenali

D. *Preprocessing*

Preprocessing (pra-pemrosesan) dalam konteks aplikasi OCR (*Optical Character Recognition*) untuk perhitungan tagihan listrik otomatis adalah serangkaian langkah yang dilakukan pada gambar atau citra teks sebelum teks tersebut diubah menjadi teks yang dapat dikenali oleh komputer. *Preprocessing* bertujuan untuk meningkatkan kualitas citra teks dan mempersiapkannya untuk langkah-langkah pengenalan karakter.

Berikut adalah beberapa tahapan *preprocessing* yang umumnya digunakan dalam aplikasi OCR untuk perhitungan tagihan listrik otomatis:

1. Konversi ke *Grayscale*: Citra berwarna dikonversi menjadi citra grayscale untuk mengurangi kompleksitas pemrosesan dan meningkatkan fokus pada informasi teks. Rumus umum untuk mengubah citra warna menjadi grayscale adalah:

$$\text{Gray} = 0.299 * R + 0.587 * G + 0.114 * B, \text{ di mana } R, G, \text{ dan } B$$

adalah komponen warna merah, hijau, dan biru dari piksel.

2. Penghapusan *Noise*: *Noise* (titik-titik kecil atau garis-garis yang tidak diinginkan pada gambar) dihilangkan menggunakan teknik seperti filter median atau filter Gaussian untuk mempertajam tepi karakter.

Rumus dari penghapusan *noise*:

- a. Filter Median: **output_pixel = median(neighborhood_pixels)**

- b. Filter Gaussian: **output_pixel = sum(neighborhood_pixels * gaussian_weights)**

Salah satu jenis noise yang mengganggu adalah *gaussian noise*. *Gaussian noise* disebut juga sebagai *noise* elektronik karena muncul di amplifier atau detektor. *Gaussian Noise* disebabkan oleh sumber-sumber alam seperti getaran termal atom dan sifat diskrit dari radiasi benda hangat[8].



a. Citra Original dengan Noise



b. Hasil Konvolusi

Gambar 2. 7. Efek Filter Gaussian

Gaussian noise umumnya mengganggu nilai abu-abu dalam gambar digital. Itulah sebabnya *gaussian noise* dasarnya dirancang dan karakteristik oleh PDF atau menormalkan histogram dengan nilai abu-abu. Rumus untuk menghasilkan Gaussian noise pada titik atau sampel tertentu dalam deret waktu diskrit adalah:

$$N(x) = \mu + \sigma \cdot Z \quad \text{Equation 1}$$

Dimana:

- a. $N(x)$ adalah nilai kebisingan Gaussian pada titik x .
- b. μ adalah nilai mean (rata-rata) dari kebisingan. Biasanya, nilainya adalah nol untuk Gaussian noise.
- c. σ adalah sigma, yang mengontrol sebaran atau amplitudo kebisingan.

- d. Z adalah variabel acak yang diambil dari distribusi Gaussian standar dengan nilai mean 0 dan simpangan baku 1.
3. Pengaturan Kontras dan Kecerahan: Menyesuaikan kontras dan kecerahan citra dapat membantu karakter menjadi lebih terlihat jelas.
4. Penghalusan (*Smoothing*): Menggunakan teknik seperti filter *Gaussian* atau *filter mean* untuk menghaluskan citra, mengurangi *noise* yang tersisa, dan menghubungkan komponen karakter yang terpisah. Rumus
 - a. Filter Mean (rata-rata) : $\text{output_pixel} = \text{mean}(\text{neighborhood_pixels})$
 - b. Filter Gaussian: Sama seperti pada penghapusan noise.
5. Deteksi Tepi: Mendeteksi tepi karakter menggunakan operator seperti Sobel atau Canny untuk memperjelas batas karakter.
6. Binarisasi: Citra *grayscale* dikonversi menjadi citra biner di mana piksel-piksel yang mewakili latar belakang dijadikan hitam dan piksel-piksel yang mewakili karakter dijadikan putih. Binarisasi dapat dilakukan menggunakan metode seperti Otsu atau adaptive thresholding.
7. Segmentasi Karakter: Citra biner dipecah menjadi komponen-komponen yang mewakili karakter-karakter individual. Ini bisa melibatkan teknik-teknik seperti connected component analysis atau contour tracing.
8. Pembersihan: Menghapus komponen-komponen yang sangat kecil atau tidak mewakili karakter (misalnya, *noise* sisa setelah tahapan-tahapan sebelumnya).

E. Image Processing

Image Processing atau pengolahan citra adalah cabang ilmu komputer yang berkaitan dengan manipulasi dan analisis gambar digital. Tujuannya adalah untuk memperoleh informasi yang berguna dari gambar, mengubah gambar menjadi bentuk yang lebih baik atau lebih mudah diinterpretasikan, atau melakukan tugas khusus seperti deteksi objek, pengenalan pola, restorasi gambar, atau segmentasi.

Proses *Image Processing* dapat dibagi menjadi beberapa tahap sebagai berikut:

1. *Preprocessing*: Tahap awal dalam proses ini melibatkan *preprocessing* gambar seperti penghapusan noise (*filtering*), peningkatan kontras serta penstabilan warna
2. Segmentasi: Tahapan selanjutnya yaitu segmentasi dimana objek di dalam gambar dipisahkan satu sama lain menjadi bagian-bagian terpisah sehingga mempermudah analisis lanjutan.
3. *Feature Extraction* : Pada tahap *feature extraction* dilakukan ekstraksi informasi penting dari setiap objek pada gambar misalnya lokalisasi garis tepi, objek geometri, dan tekstur permukaan
4. *Recognition/Classification*: Pada tahap akhir, image processing digunakan untuk melakukan klasifikasi atau identifikasi terhadap obyek yang ada pada sebuah citra dengan menggunakan algoritma machine learning.

Aplikasi *Image Processing* sangat luas seperti deteksi wajah, sistem keamanan digital, pengenalan karakter tulisan tangan, *optical character recognition* (OCR), diagnosis medis dan masih banyak lagi. Dalam perkembangannya, image processing memiliki peranan penting karena sebagian besar aplikasinya sudah mendukung *Artificial Intelligence* (AI) sehingga dapat memberikan prediksi hasil analisis secara otomatis tanpa adanya campur tangan manusia.

1. Pengenalan Teks (Text Recognition)

Pengenalan teks atau *Text Recognition* adalah teknologi yang memungkinkan mesin untuk membaca dan mengenali teks dalam bentuk digital dari gambar, dokumen cetak, atau video. Teknologi pengenalan teks bekerja dengan menggunakan algoritma *machine learning* seperti jaringan saraf tiruan (*neural network*) dan *Deep Learning*.

Proses pengenalan teks dimulai dengan pemrosesan citra untuk melakukan segmentasi karakter pada sebuah dokumen. Kemudian dilakukan ekstraksi fitur pada setiap karakter yang tersegmentasi sehingga informasi penting dari karakter dapat diambil dan dianalisis lebih lanjut oleh sistem. Setelah itu, pola-pola karakter ini dibandingkan dengan database huruf dan angka yang sudah disimpan sebelumnya agar bisa diidentifikasi.

Terdapat beberapa pendekatan yang digunakan dalam pengenalan teks, antara lain:

- a. Metode *Template Matching*: Metode ini melibatkan perbandingan karakter-karakter yang terdapat dalam gambar dengan sekumpulan template

karakter yang telah ada sebelumnya. Pengenalan dilakukan dengan mencocokkan karakter pada gambar dengan template yang paling mirip.

- b. Metode Berbasis Aturan: Metode ini melibatkan penggunaan aturan-aturan yang telah ditentukan sebelumnya untuk mengenali karakter-karakter dalam gambar. Aturan-aturan ini dapat mencakup informasi tentang bentuk, ukuran, dan pola karakter.
- c. Metode Berbasis Pembelajaran Mesin: Metode ini menggunakan algoritma pembelajaran mesin untuk melatih model pengenalan teks menggunakan dataset gambar dan teks yang telah dilabeli sebelumnya. Model ini kemudian dapat digunakan untuk mengenali teks dalam gambar baru.
- d. Metode Berbasis Jaringan Saraf Tiruan: Metode ini melibatkan penggunaan jaringan saraf tiruan (*neural network*) untuk mengenali karakter-karakter dalam gambar. Jaringan saraf tiruan dilatih menggunakan dataset gambar dan teks untuk mengenali pola dan fitur-fitur karakter dalam gambar.

Aplikasi dari teknologi pengenalan teks sangat beragam mulai dari pencarian data pasien medis hingga proses otomatisasi pengecekan tiket pesawat secara online. Dalam industri keuangan, teknologi ini digunakan untuk memproses formulir kredit bank serta dalam bidang pemerintahan digunakan sebagai salah satu metode identifikasi wajah dalam e-KTP.

2. Metode Perhitungan Tagihan

Metode penghitungan tagihan adalah pendekatan atau prosedur yang digunakan untuk menghitung jumlah tagihan yang harus dibayar oleh pelanggan berdasarkan konsumsi atau penggunaan suatu layanan atau produk tertentu[11]. Metode penghitungan tagihan dapat bervariasi tergantung pada jenis layanan atau produk yang diberikan, seperti listrik, air, telepon, internet, atau layanan langganan lainnya.

Contoh metode penghitungan tagihan untuk listrik adalah sebagai berikut:

1. Metode Pembacaan Meteran: Metode ini melibatkan petugas yang melakukan pembacaan meteran listrik secara periodik di lokasi pelanggan. Pembacaan meteran dilakukan dengan mencatat angka meteran awal dan angka meteran akhir pada periode tertentu. Selisih antara kedua angka meteran ini digunakan untuk menghitung konsumsi listrik pelanggan. Tagihan kemudian dihitung berdasarkan tarif listrik yang berlaku.
2. Metode Estimasi: Jika pembacaan meteran tidak dapat dilakukan secara periodik, metode estimasi digunakan untuk menghitung konsumsi listrik. Metode ini melibatkan perkiraan berdasarkan data konsumsi sebelumnya atau faktor-faktor lain seperti ukuran rumah, jumlah penghuni, atau pola penggunaan listrik pelanggan. Estimasi ini digunakan untuk menentukan jumlah tagihan yang harus dibayar.
3. Metode *Smart Meter*: Metode ini melibatkan penggunaan smart meter yang secara otomatis mengukur dan mencatat konsumsi listrik pelanggan. Data konsumsi yang tercatat oleh smart meter dikirimkan secara langsung ke

penyedia layanan listrik. Dengan menggunakan smart meter, tagihan dapat dihitung secara akurat berdasarkan data yang tercatat oleh meter tersebut.

4. Metode Tarif Blok: Metode ini melibatkan penggunaan tarif listrik berdasarkan blok penggunaan. Setiap blok memiliki tarif listrik yang berbeda. Konsumsi listrik pelanggan dihitung berdasarkan jumlah blok yang terpakai dan tagihan dihitung sesuai dengan tarif masing-masing blok.
5. Metode *Prepaid*: Metode ini melibatkan penggunaan sistem pembayaran sebelumnya. Pelanggan membeli kredit listrik sebelumnya dan menggunakannya sesuai dengan kebutuhan. Jumlah tagihan dihitung berdasarkan jumlah kredit yang digunakan.

Dalam penagihan listrik, yang dihitung adalah pemakaian energi listrik oleh pelanggan dalam satu bulan. Setiap rumah atau gedung memiliki meteran listrik sebagai alat pengukur yang akan mencatat jumlah *kilowatt hour* (kWh) setiap kali ada penggunaan listrik.

KWH adalah satuan pemakaian energi elektrik pada arus AC dan DC, biasanya disebut dengan satuan kWh. Satu kWh berarti daya 1 kilowatt dipakai selama satu jam dan biaya tagihan tergantung pada harga per kWh dari PLN. Besarnya tagihan juga dapat ditentukan oleh tarif dasar yang diberikan oleh PLN sesuai dengan jenis golongan tarif pelanggan seperti rumah tangga atau industri. Biaya tambahan seperti PPN (Pajak Pertambahan Nilai), PPI (Pemeliharaan Penyulang Jaringan), serta beban maksimal akan ditambahkan ke dalam total tagihan akhir.



Gambar 2. 8. Meteran Listrik dengan Daya 900 VA

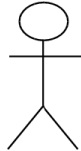
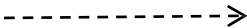
Berikut adalah contoh perhitungan tagihan listrik untuk pelanggan rumah tangga dengan tarif dasar 900 VA:

1. Pengukuran pemakaian listrik pada bulan Januari sebesar 300 kWh.
2. Harga per kWh untuk golongan tarif ini adalah Rp1.467,28.
3. Biaya energi yang harus dibayar = Jumlah kWh x harga per kWh = 300 x Rp1.467,28 = Rp440.184
4. Selain biaya energi, akan ditambahkan juga biaya beban maksimal (BM) dan Pajak Pertambahan Nilai (PPN). Untuk golongan tarif 900 VA:
 - BM: Rp7.500/kVA/bulan.
 - PPN: 10% dari total tagihan.

5. Dalam hal ini, beban maksimum adalah 0,9 kVA karena menggunakan tarif dasar 900 VA. Maka biaya beban maksimal yang harus dibayar = BM x jumlah kVA = $RP7.500 \times 0,9 = RP6.750$
6. $PPN = 10\%(\text{biaya energi} + \text{biaya beban maksimum}) = (10/100)(Rp440184 + Rp6750) = (10/100) * Rp446934 = RP44.693$
7. Total Tagihan Listrik = Biaya Energi + Biaya Beban Maksimal + Pajak Pertambahan Nilai (PPN) = $RP440184 + RP6750 + RP44.693 = RP493627$
Jadi, total tagihan listrik pelanggan di bulan Januari sebesar Rp493627.

F. Use Case Diagram

Tabel 2. 2 Simbol *Use Case Diagram*

GAMBAR	NAMA	KETERANGAN
	Actor	Menspesifikasikan himpunan peran yang pengguna mainkan Ketika berinteraksi dengan use case.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (independent).
	Generalization	Hubungan dimana objek anak (descendent) berbagai perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor)

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, bukan “bagaimana”. Sebuah *use case* mempresentasikan sebuah interaksi antara aktor dengan sistem.

G. Use Case Diagram

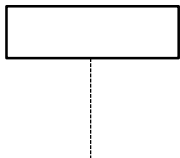
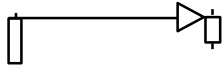
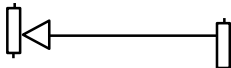
Tabel 2. 3 Simbol *Activity Diagram*

GAMBAR	NAMA	KETERANGAN
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
	<i>Initial Node</i>	Bagaimana objek dibentuk atau Diawali
	<i>Activity Final Node</i>	Bagaimana objek dibentuk dan Dihancurkan
	<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

Activity diagram menggambarkan berbagai aliran aktivitas dalam sistem yang sedang dirancang. *Activity* diagram dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi, berikut table simbol *Activity*.

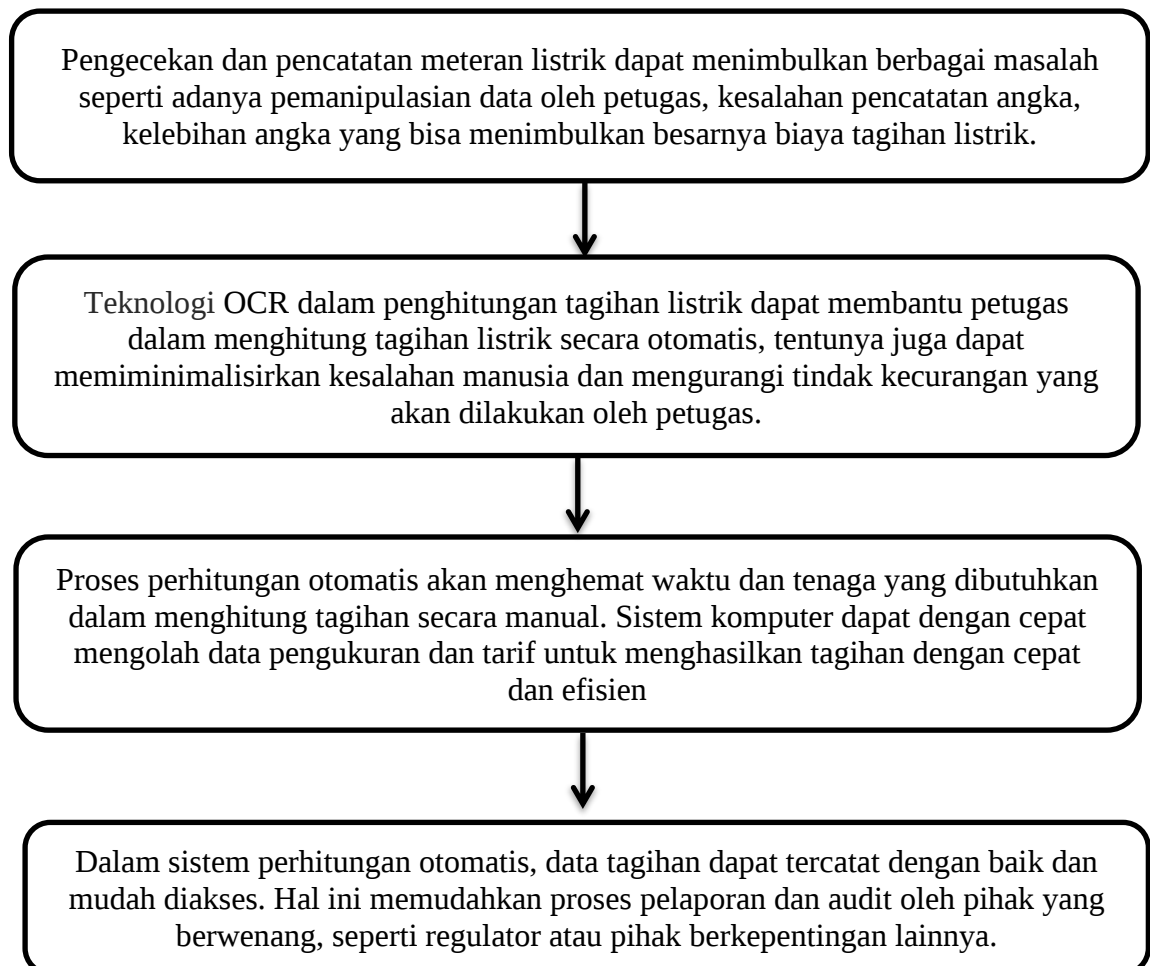
H. Sequence Diagram

Tabel 2. 4 Simbol *Sequence Diagram*

GAMBAR	NAMA	KETERANGAN
	LifeLine	Objek entity, antarmuka yang saling berinteraksi
	Message	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
	Message	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Sequence Diagram menggambarkan interaksi antar objek didalam dan disekitar sistem (termasuk pengguna, display, dan sebagainya) berupa pesan yang digambarkan terhadap waktu. Berikut tabel *Symbol Sequence Diagram*;

I. Kerangka Pikir



BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang dilakukan merupakan penelitian eksperimental, yaitu penelitian yang pengumpulan datanya melalui pencatatan secara langsung dari hasil percobaan yang dilakukan.

B. Lokasi dan Waktu Penelitian

Tempat penelitian dilakukan di tempat yang dapat mencakup area-area di mana PLN memiliki infrastruktur listrik, pelanggan listrik yang signifikan, atau daerah yang menghadapi tantangan khusus dalam perhitungan tagihan listrik di Kota Parepare. Waktu yang dibutuhkan dalam pelaksanaan penelitian ini adalah $\pm$ 3 bulan.

C. Alat dan Bahan Penelitian

Untuk melakukan proses penelitian dalam pembuatan aplikasi, maka diperlukan perangkat keras dan perangkat lunak guna mendukung kegiatan penelitian tersebut. Berikut ini merupakan penjelasan dari *hardware* dan *software* yang digunakan dalam pembuatan aplikasi penghitungan tagihan listrik ini.

1. Perangkat keras yang digunakan selama proses penelitian yaitu :

a. Laptop Acer Aspire A514-52K dengan spesifikasi:

1) *Processor : Intel (R) Core(TM) i3-8130U CPU @ 2.20GHz 2.21 GHz*

2) *RAM : 8 GB*

- 3) *SSD : 1 TB*
- 4) *Sistem Operasi : Windows 11*
- b. Smartphone Vivo Y21 dengan spesifikasinya :
 - 1) *Processor : Mediatek Helio P35*
 - 2) *RAM : 4 GB*
 - 3) *Baterai : 5000 mAh*
 - 4) *Layar : 6.51-inch 1600x720 HD+*
 - 5) *Kamera Depan : 8 mp f/2.0*
 - 6) *Kamera Belakang : f/2.2 (13MP) + f/2.4 (2MP)mp f/2.2*
 - 7) *Android : Android 12*
- c. *Meteran Listrik sebagai dataset*
- 2. *Software yang digunakan dalam pembuatan aplikasi yaitu :*
 - a. *Beberapa Library Open source (Tesseract OCR dan OpenCV)*
 - b. *Bahasa Pemrograman Python*
 - c. *Platform Visual Studio Code*
 - d. *MySQL*

D. Teknik Pengumpulan Data

Teknik pengumpulan data adalah proses untuk mengumpulkan informasi atau data yang relevan dengan penelitian atau studi yang sedang dilakukan. Dalam konteks penelitian tentang implementasi OCR untuk penghitungan tagihan listrik, beberapa teknik pengumpulan data yang dapat digunakan adalah sebagai berikut:

1. *Observasi: Teknik observasi melibatkan pengamatan langsung terhadap proses penghitungan tagihan listrik yang sedang berlangsung. Peneliti dapat*

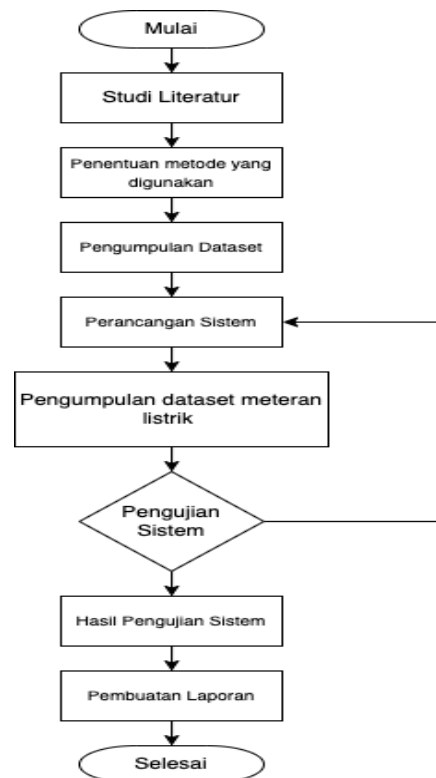
mengamati petugas yang melakukan penghitungan tagihan secara manual atau mengamati sistem yang digunakan dalam proses tersebut. Observasi ini dapat memberikan wawasan tentang proses yang ada dan masalah yang mungkin timbul.

2. Wawancara: Teknik wawancara melibatkan interaksi langsung antara peneliti dan pihak yang terlibat dalam penghitungan tagihan listrik, seperti petugas PLN atau manajer. Wawancara ini dapat dilakukan untuk mendapatkan pemahaman yang lebih dalam tentang proses penghitungan tagihan, kesulitan yang dihadapi, dan perspektif pengguna terkait implementasi OCR.
3. Studi Dokumen: Studi dokumen melibatkan analisis terhadap dokumen-dokumen yang relevan dengan penghitungan tagihan listrik, seperti pedoman penghitungan, data tagihan, atau laporan sebelumnya. Dokumen-dokumen ini dapat memberikan informasi tentang metode yang digunakan saat ini, kelemahan yang ada, atau informasi penting lainnya yang relevan dengan implementasi OCR.
4. Uji Coba dan Eksperimen: Uji coba dan eksperimen dapat dilakukan untuk menguji efektivitas dan keakuratan implementasi OCR dalam penghitungan tagihan listrik. Peneliti dapat mengumpulkan data dengan melakukan pengujian pada sampel tagihan listrik menggunakan OCR dan membandingkannya dengan hasil perhitungan manual untuk mengevaluasi keberhasilan dan efisiensi implementasi OCR.
5. Analisis Data Historis: Data historis yang ada dalam sistem atau basis data PLN dapat dianalisis untuk memahami pola dan tren dalam penghitungan tagihan

listrik. Analisis ini dapat memberikan wawasan tentang tingkat kesalahan manusia yang mungkin terjadi, variabilitas dalam perhitungan tagihan, dan area-area di mana implementasi OCR dapat memberikan manfaat yang signifikan.

E. Tahap Penelitian

Tahap-tahap penelitian yang dimaksud dalam penelitian ini sesuai pada gambar 3.1 berikut:



Gambar 3. 1. Tahapan Penelitian

1. Tahapan Studi Literatur

Tahapan studi literatur merupakan tahapan pencarian informasi dan referensi- referensi penelitian sebelumnya yang terkait dengan informasi system OCR dalam berbagai permasalahan

2. Tahapan Analisis dan Desain Sistem

Pada tahap analisis dilakukan perumusan masalah, batasan-batasan masalah dan bagaimana menyelesaikan masalah tersebut. Permasalahan yang diangkat pada penelitian ini adalah bagaimana system ini mampu mendeteksi angka meteran listrik berdasarkan karakteristik yang telah ditentukan pada ekstraksi fitur.

3. Tahapan Implementasi.

Tahapan implementasi adalah tahapan penerapan tahapan desain system dengan harapan tahapan ini menghasilkan keluaran yang sesuai dengan tahap analisis.

4. Tahapan Uji Coba dan Analisis Hasil

Tahapan uji coba dan analisa hasil merupakan tahapan dimana sistem yang telah diimplementasikan sesuai dengan desain sistem akan diuji coba pada tahapan ini. Uji coba dilakukam sesuai dengan skenario uji coba yang telah dirancang pada tahap desain sistem. Berdasarkan hasil uji coba tersebut, sistem akan dianalisis untuk mengetahui performa dari sistem yang dibangun.

5. Tahapan Penulisan Laporan

Tahapan ini merupakan tahapan penulisan laporan dari keseluruhan tahapan yang dilakukan pada penelitian ini. Setiap tahapan akan didokumentasikan secara tertulis sebagai bukti dari hasil penelitian yang telah dilakukan.

F. Metode Pengujian

Dalam penelitian tentang implementasi OCR untuk penghitungan tagihan listrik, beberapa metode pengujian yang dapat digunakan adalah pengujian Fungsional (Functional Testing). Metode ini dilakukan untuk memverifikasi apakah aplikasi OCR yang dikembangkan berfungsi sesuai dengan kebutuhan dan spesifikasi yang telah ditetapkan. Pengujian fungsional mencakup pengujian berbagai fitur dan fungsionalitas OCR, seperti kemampuan mendeteksi teks, mengenali karakter, dan melakukan penghitungan tagihan dengan akurasi yang tinggi.

G. Jenis Data

Jenis data yang dikumpulkan dalam penelitian ini adalah sebagai berikut:

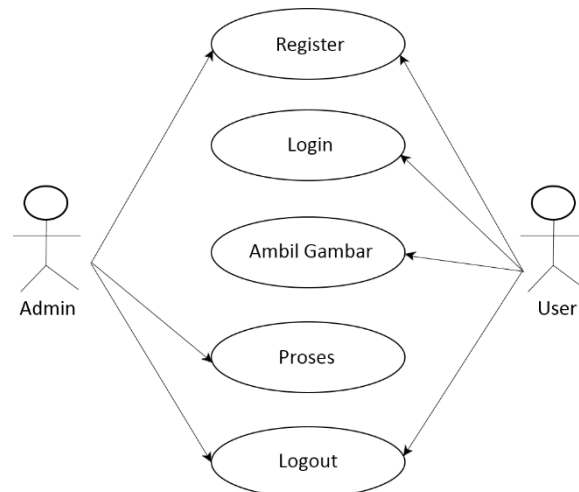
1. Data Tagihan Listrik: Data tagihan listrik merupakan data utama yang akan digunakan dalam pengujian. Ini termasuk informasi tentang penggunaan listrik, tarif listrik, biaya yang harus dibayar, dan detail lainnya terkait tagihan. Data ini dapat diperoleh dari sistem tagihan listrik yang ada atau dapat disimulasikan dalam pengujian.
2. Data Gambar Teks: Data gambar teks digunakan untuk menguji kemampuan OCR dalam mengenali dan mengubah teks yang terdapat dalam gambar menjadi teks yang dapat dibaca oleh komputer. Data ini bisa berupa gambar meteran listrik atau dokumen tagihan yang akan diolah oleh OCR.
3. Data Referensi: Data referensi digunakan untuk memvalidasi hasil pengenalan teks oleh OCR. Data ini berisi teks yang benar atau data tagihan listrik yang

telah dihitung secara manual sebelumnya. Data referensi ini dapat digunakan untuk membandingkan hasil OCR dan mengukur tingkat keakuratan OCR dalam penghitungan tagihan.

4. Data Uji Coba: Data uji coba digunakan untuk menguji kinerja dan akurasi OCR pada berbagai situasi dan kondisi. Data ini dapat berupa berbagai jenis gambar teks dengan variasi font, ukuran, orientasi, dan tingkat pencahayaan yang berbeda. Data uji coba ini membantu dalam mengevaluasi kehandalan OCR dalam mengenali dan mengolah berbagai jenis data.
5. Data Konfigurasi OCR: Data konfigurasi OCR mencakup pengaturan dan parameter yang digunakan dalam proses pengenalan teks oleh OCR. Ini termasuk pengaturan tingkat kecerahan, kontras, ukuran karakter, metode pengenalan, dan lain sebagainya. Data konfigurasi ini dapat diubah dan disesuaikan untuk mengoptimalkan performa OCR dalam penghitungan tagihan listrik.

H. Desain Sistem

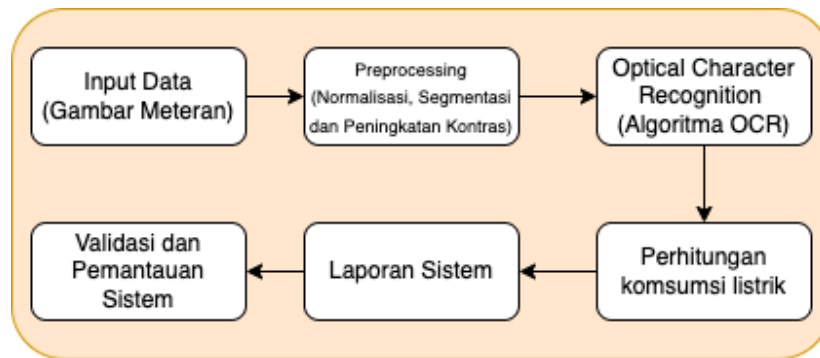
1. Diagram Usecase



Gambar 3. 2. *Use Case Diagram*

Dalam diagram Use Case pada gambar 3.2, pengguna memiliki akses ke aplikasi OCR untuk melakukan penghitungan tagihan listrik. Desain Interface dan admin dapat melihat hasil dari proses dari perhitungan tagihan listrik.

2. Rancangan Sistem



Gambar 3. 3. Rancangan Sistem Aplikasi

Penjelasan :

1. Input Data Gambar:

- a) Pengguna memasukkan gambar tagihan listrik melalui antarmuka pengguna.
- b) Sistem mengizinkan pengguna untuk mengunggah gambar dari perangkat mereka atau menggunakan kamera untuk mengambil gambar.

2. *Preprocessing*:

- a) Gambar yang diterima akan diproses untuk memastikan kualitasnya optimal untuk proses OCR.
- b) Proses preprocessing dapat mencakup penyesuaian kecerahan, kontras, dan penghapusan noise untuk meningkatkan kualitas gambar.
- c) Gambar yang sudah diproses akan dipecah menjadi bagian-bagian yang berisi teks.
- d) Teknik segmentasi digunakan untuk mengidentifikasi dan memisahkan area teks dari gambar.

3. *Optical Character Recognition (OCR)*:

- a) Setiap bagian teks yang dihasilkan akan diproses menggunakan algoritma OCR.
- b) OCR akan mengenali karakter dalam teks dan mengubahnya menjadi teks yang dapat dibaca oleh komputer.
- c) Setelah OCR selesai, sistem akan mengambil informasi yang relevan dari hasil pengenalan teks.
- d) Informasi yang diekstraksi dapat mencakup nomor meteran, tanggal pembacaan, jenis tarif, dan informasi lain yang diperlukan untuk penghitungan tagihan.

4. Laporan Sistem

- a) Sistem akan menggunakan informasi yang diekstraksi untuk melakukan perhitungan tagihan listrik.
- b) Perhitungan dapat mencakup penggunaan tarif listrik, faktor pemakaian, dan komponen biaya lainnya yang relevan.
- c) Hasil penghitungan tagihan akan ditampilkan kepada pengguna melalui antarmuka pengguna.
- d) Pengguna dapat melihat jumlah tagihan yang dihasilkan berdasarkan informasi yang diproses oleh sistem.

5. Validasi Sistem:

- a) Memeriksa kebenaran perhitungan tagihan yang dilakukan oleh sistem.
- b) Membandingkan hasil perhitungan dengan perhitungan manual atau referensi tagihan lainnya untuk memastikan keakuratan dan konsistensi.

BAB IV

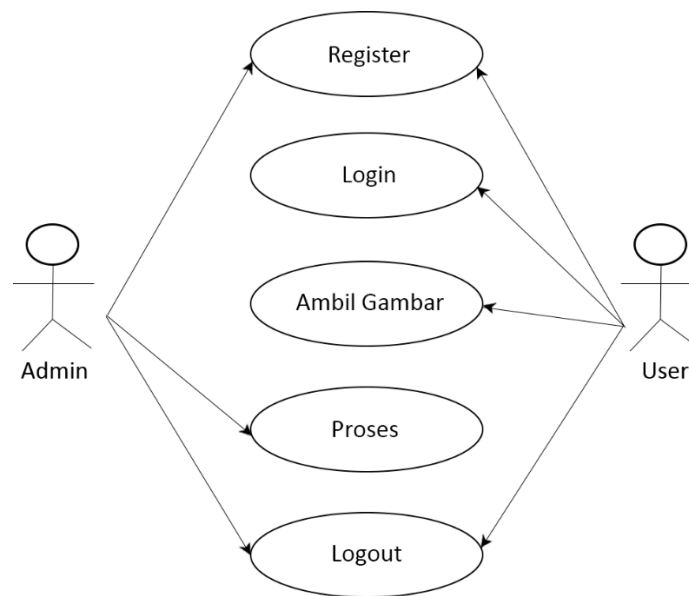
HASIL DAN PEMBAHASAN

A. Analisis Aliran Data Dengan UML

Proses pengembangan selanjutnya adalah proses desain pengembangan perangkat lunak. Desain sistem perangkat lunak yang dibuat nantinya akan dijadikan sebagai acuan pengembang dalam penulisan kode. Desain sistem perangkat lunak ini harus sesuai karakteristik perangkat lunak yang akan dibuat serta mampu mempermudah dan memperjelas pengembang dalam proses pembuatan perangkat lunak. Penelitian ini menggunakan desain sistem model *Unified Model Language* (UML) dikarenakan model ini paling sesuai digunakan Untuk mengembangkan sistem berinteraksi.

1. Use Case Diagram

Use case menggambarkan bagaimana seorang pengguna berinteraksi dengan sistem dengan cara menentukan langkahlangkah yang diperlukan untuk mencapai suatu tujuan tertentu. Variasi-variasi dalam urutan langkah yang terjadi antara aktor dan sistem merupakan sebuah skenario. Diagram *use case* memberikan suatu gambaran besar tentang fungsionalitas yang diharapkan dari sebuah sistem dilihat dari sudut pandang pengguna.



Gambar 4. 1 Use Case Diagram Sistem yang diusulkan

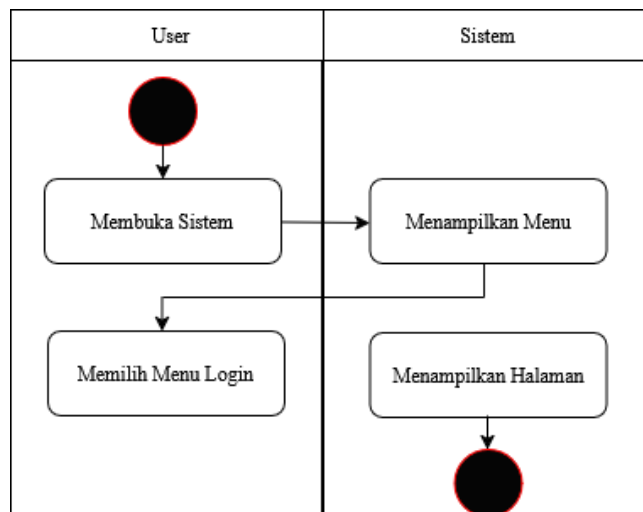
Tabel 4. 1. Pengujian Langsung dengan Kondisi Gambar Jelas

Nama Use Case	Deksripsi Use Case
Register	Merupakan tampilan pendaftar pada pengguna sebelum melakukan login
Login	Merupakan bagian menu yang menampilkan halaman login untuk masuk kedalam sistem, Sistem login dapat dilakukan oleh admin dan user
Ambil Gambar	Merupakan bagian menu untuk pensorotan meteran listrik kemudian digunakan untuk menghitung besarnya tagihan pelanggan sesuai dengan penggunaan kwh berdasarkan tarif watt pelanggan
Proses	Merupakan bagian menu untuk memproses hasil dari pengambilan gambar meteran listrik
Logout	Merupakan bagian keluar dari sistem

2. Activity Diagram

Activity Diagram ini menjelaskan tentang aktivitas-aktivitas yang terjadi dalam sebuah aliran proses pada Implementasi Optical Character Recognition (OCR) Untuk Otomasi Penghitungan Tagihan listrik.

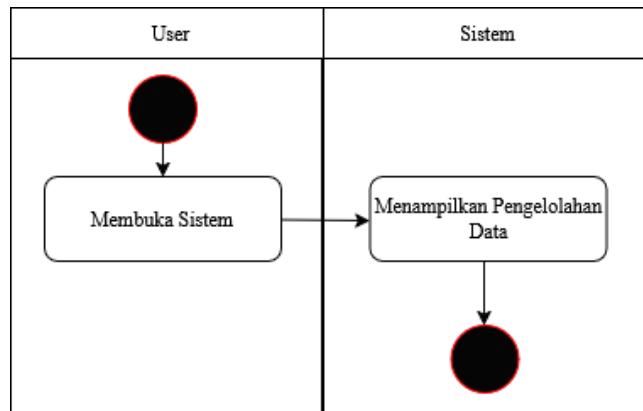
a. Activity diagram Login Admin



Gambar 4.2 Activity Diagram Login Admin

Gambar diatas menjelaskan alur pada menu Login yang dimulai dari membuka sistem dengan tujuan menampilkan menu dan memilih menu login untuk menampilkan halaman login.

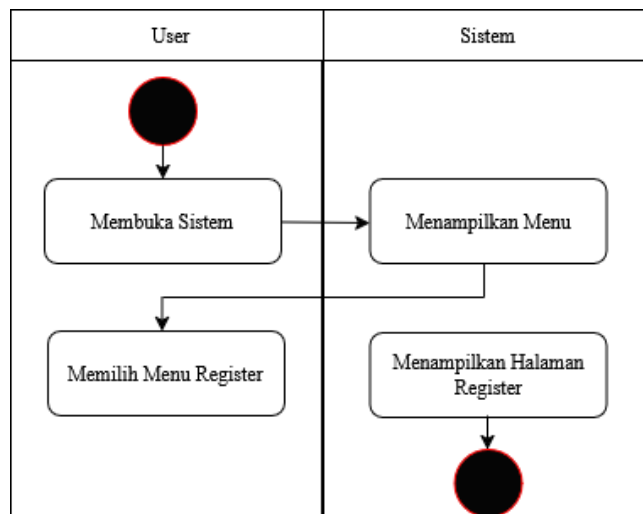
b. *Activity diagram* Pengelolah Data Pengguna



Gambar 4.3 *Activity Diagram* Login Admin

Gambar diatas menjelaskan alur pada menu Kelola Data yang dimulai dari membuka sistem dengan tujuan menampilkan data pengelola pengguna.

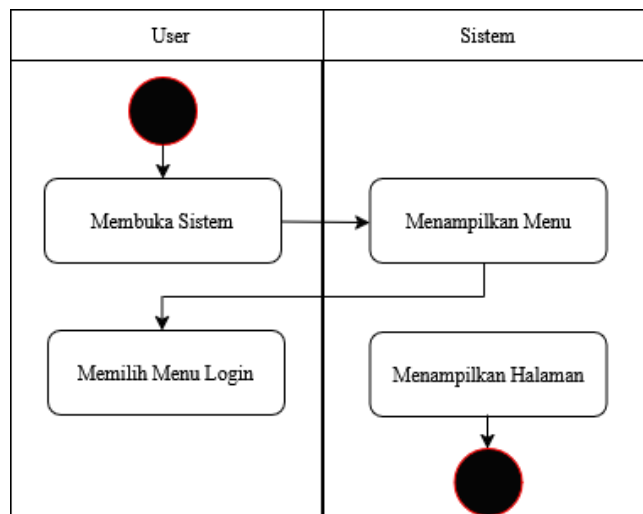
c. *Activity diagram* Register User



Gambar 4.4 *Activity Diagram* Register User

Gambar diatas menjelaskan alur pada menu Register User yang dimulai dari membuka sistem dengan tujuan menampilkan menu dan memilih menu Register untuk menampilkan halaman Register.

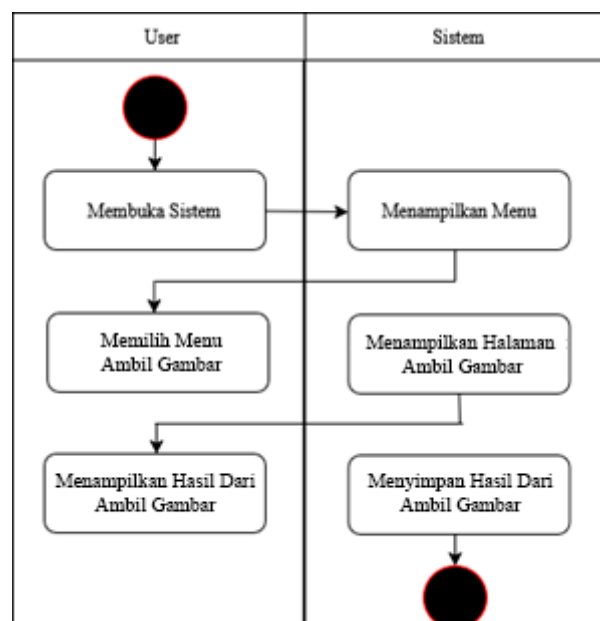
d. *Activity diagram Login User*



Gambar 4.5 *Activity Diagram Login User*

Gambar diatas menjelaskan alur pada menu Login yang dimulai dari membuka sistem dengan tujuan menampilkan menu dan memilih menu login untuk menampilkan halaman login.

e. *Activity diagram Ambil Gambar*



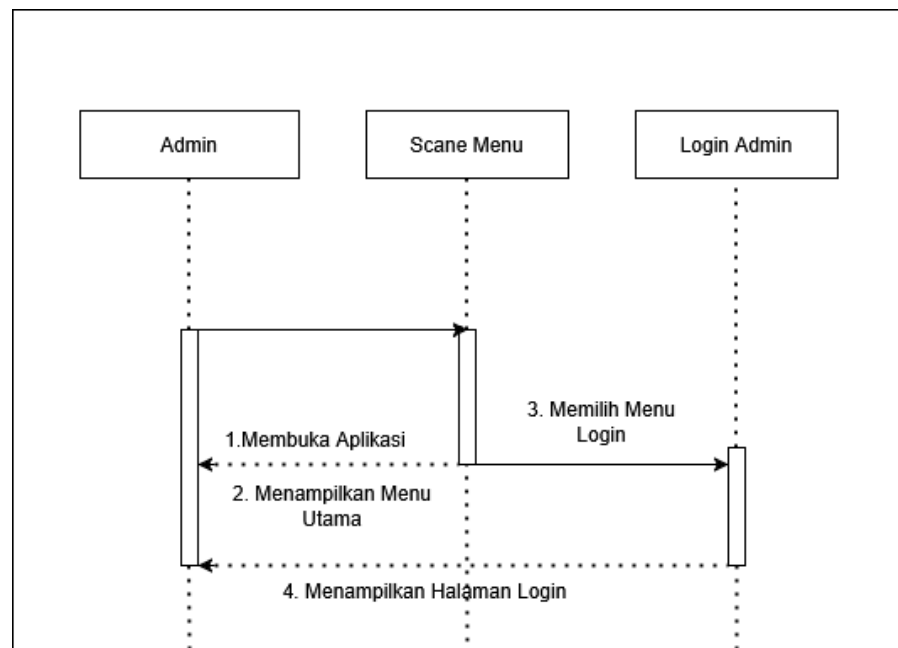
Gambar 4.6 *Activity Diagram Ambil Gambar*

Gambar diatas menjelaskan alur pada menu Ambil Gambar yang dimulai dari membuka sistem dengan tujuan menampilkan menu, kemudian menampilkan halaman pengambilan gambar meteran watt sebelum dilakukan penghitungan dengan tarif Watt pengguna

3. Sequence Diagram

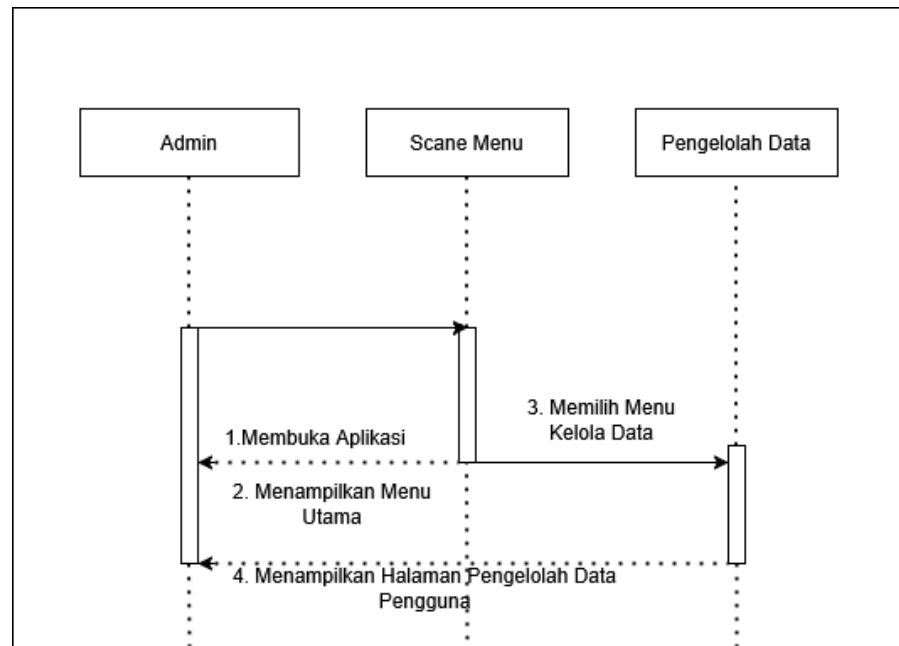
Sequence Diagram merupakan salah satu diagram *Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan *message* (pesan) apa yang dikirim dan kapan pelaksanaannya.

a. Sequence diagram Login Admin



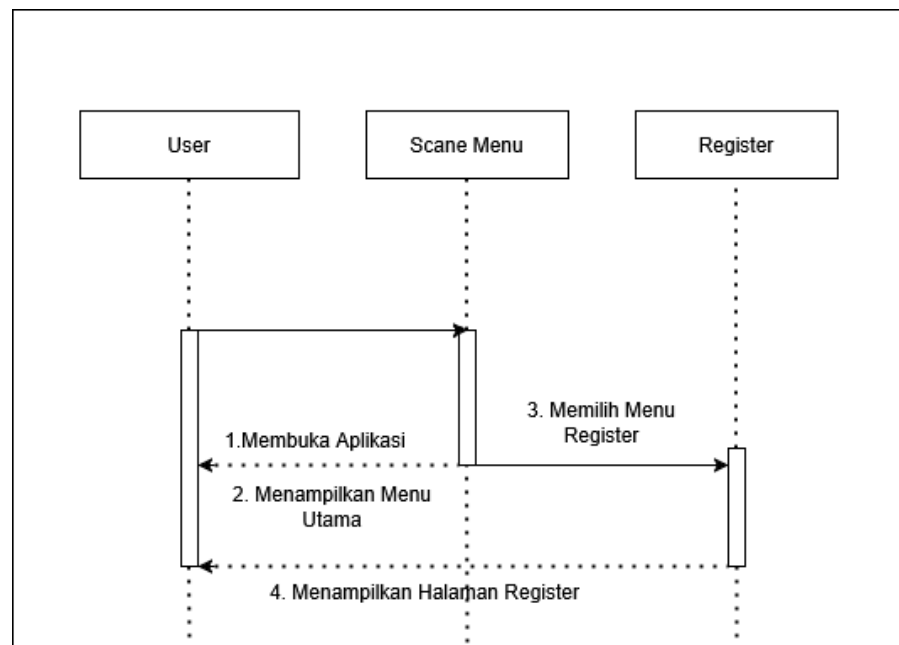
Gambar 4. 7 Sequence diagram Login Admin

b. *Sequence diagram* Pengelolah Data Pengguna



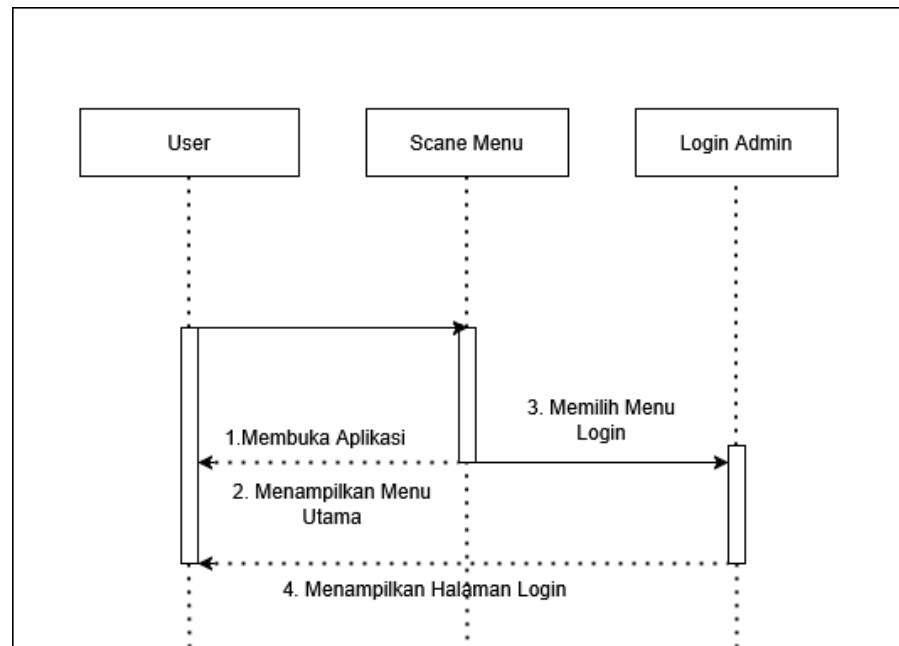
Gambar 4. 8 *Sequence diagram* Pengelolah Data Pengguna

c. *Sequence diagram* Register User



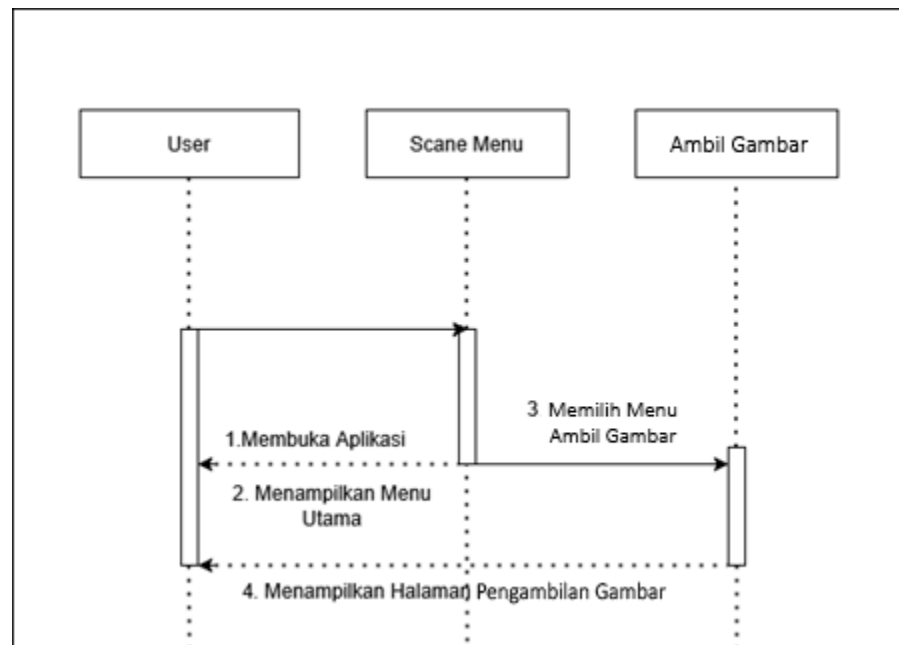
Gambar 4. 9 *Sequence diagram* register User

d. *Sequence diagram Login User*



Gambar 4. 10 *Sequence diagram Login User*

e. *Sequence diagram Ambil Gambar*



Gambar 4. 11 *Sequence diagram Ambil Gambar*

B. DESKRIPSI DATA

Data yang digunakan dalam penelitian ini berupa gambar tagihan listrik rumah tangga yang diambil dari beberapa sumber simulasi, seperti hasil pemindaian dan fotografi. Berikut adalah karakteristik gambar yang diuji:

1. **Format Gambar:** Gambar disimpan dalam format JPG dan PNG, yang kompatibel dengan sebagian besar perangkat OCR seperti Tesseract.
2. **Kondisi Gambar:**
 - a. Tagihan Simulasi: Gambar tagihan yang disusun berdasarkan format standar perusahaan listrik, mencakup informasi seperti:
 - 1) Nomor pelanggan.
 - 2) Pemakaian listrik (kWh).
 - 3) Tarif dasar listrik (TDL).
 - 4) Total tagihan.
 - b. Kondisi Beragam: Gambar diuji dengan berbagai skenario, seperti pencahayaan rendah, keberadaan noise, dan sudut kemiringan hingga 15 derajat untuk mensimulasikan penggunaan dalam kondisi nyata.
3. **Jumlah Gambar:** Untuk pengujian, digunakan 10 gambar meteran listrik dengan berbagai variasi untuk mengevaluasi performa sistem dalam kondisi berbeda.

C. PROSES IMPLEMENTASI OCR

Pada bagian ini, dijelaskan langkah-langkah implementasi *Optical Character Recognition* (OCR) menggunakan *Tesseract* untuk otomatisasi penghitungan tagihan listrik. Proses implementasi melibatkan beberapa tahapan, mulai dari persiapan data hingga integrasi hasil OCR ke dalam aplikasi. Berikut adalah langkah-langkahnya:

1. Instalasi dan Konfigurasi Tesseract

Langkah pertama dalam implementasi OCR adalah instalasi Tesseract OCR pada sistem. Berikut prosesnya:

a. Instalasi Tesseract

Tesseract OCR adalah sebuah engine *open-source* yang kuat untuk *Optical Character Recognition* (OCR). OCR adalah teknologi yang memungkinkan komputer untuk membaca teks dari gambar. Tesseract sangat populer karena akurasi yang tinggi dan kemampuannya untuk mengenali berbagai jenis font dan bahasa.

- 1) Tesseract diinstal pada sistem operasi yang digunakan menggunakan Homebrew dengan perintah:

```
(base)Data-D@Skripsi/AplikasiOCR ~ % brew install tesseract

==> Auto-updating Homebrew...
Adjust how often this is run with HOMEBREW_AUTO_UPDATE_SECS or disable with
HOMEBREW_NO_AUTO_UPDATE. Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man b
rew`).
==> Auto-updated Homebrew!
Updated 3 taps (caskroom/cask, homebrew/core and homebrew/cask).
==> New Formulae
acronym          libheinz          planus
cargo-dist       monocle           ruby@3.3
cargo-flamegraph neverest          sfml@2
git-ignore       odbc2parquet     static-web-server
go-parquet-tools oranda           swctl
humanlog         parallel-disk-usage the-way
libesedb         periphery        wcstools
libformfactor    pistache
==> New Casks
alcove           font-fairfax-hd   nexonplug
alcove           font-kreative-square nexonplug
ayugram          font-kreative-square tm-error-logger
ayugram          ghostty           tm-error-logger
```

Gambar 4. 12. Proses Instalasi Teserract

Di gambar diatas perangkat Tesseract diinstal menggunakan Homebrew yang dimana kemudian Homebrew melakukan membaruan secara otomatis di beberapa komponen dan daftar formula atau paket perangkat lunak yang tersedia.

- 2) Untuk menambahkan dukungan bahasa lain, gunakan perintah dibawah untuk menambahkan Bahasa Indonesia.

```
((base)Data-D@Skripsi/AplikasiOCR ~ % brew install tesseract-ind
Warning: No available formula with the name "tesseract-ind". Did you mean tesser
act or tesseract-lang?
==> Searching for similarly named formulae and casks...
==> Formulae
tesseract ✓                tesseract-lang ✓

To install tesseract ✓, run:
  brew install tesseract ✓
```

Gambar 4. 13. Proses Instalasi Penambahan Bahasa Indonesia ke OCR

Dengan menggunakan Bahasa Indonesia untuk menjalankan perangkat Tesseract akan memudahkan programer dalam mengoperasikan Tesseract kedepannya.

- 3) Untuk memastikan Tesseract sudah terinstal dengan benar, buka terminal dan jalankan perintah.

```

((base)Data-D@Skripsi/AplikasiOCR ~ % tesseract --version
tesseract 5.5.0
  leptonica-1.85.0
  libgif 5.2.2 : libjpeg 8d (libjpeg-turbo 3.1.0) : libpng 1.6.44 : libtiff 4.7.
0 : zlib 1.2.11 : libwebp 1.5.0 : libopenjp2 2.5.3
Found AVX2
Found AVX
Found FMA
Found SSE4.1
Found libarchive 3.7.7 zlib/1.2.11 liblzma/5.6.3 bz2lib/1.0.8 liblz4/1.10.0 lib
zstd/1.5.6
Found libcurl/7.79.1 SecureTransport (LibreSSL/3.3.6) zlib/1.2.11 nghttp2/1.45.
1

```

Gambar 4. 14. Proses Verifikasi Penginstalan Teserract

Dengan memanggil perintah diatas, programmer dapat mengetahui apakah Tesseract sudah terinstal dengan benar dan cocok sebelum melangkah ke tahap selanjutnya.

- 4) Install Teserract pada Bahasa PHP dengan menjalankan perintah berikut: *composer require thiagoalessio/tesseract-ocr-for-php*

b. *Preprocessing* Gambar

Agar Tesseract dapat membaca teks dengan lebih baik, gambar diproses terlebih dahulu menggunakan tools ImageMagick. Tahapan preprocessing meliputi:

1) Konversi ke *Grayscale*

Gambar diubah ke *grayscale* untuk mengurangi gangguan warna dan fokus pada teks.

```
// Memuat gambar menggunakan OpenCV
$srcImage = cv\imread($uploadFile, cv\IMREAD_COLOR);

if ($srcImage == null) {
    http_response_code(500);
    echo json_encode(['error' => 'Failed to load the image.']);
    exit;
}

$grayImage = cv\cvtColor($srcImage, cv\COLOR_BGR2GRAY);
```

Gambar 4. 15. Proses Konversi Gambar ke *Grayscale*

Dengan diubahnya gambar menjadi *greyscale* akan meringankan perangkat OCR mendeteksi karakter pada sebuah gambar dan membuat pendeteksian karakter lebih akurat.

2) Penghapusan *Noise*

Noise pada gambar dihapus menggunakan filter GaussianBlur untuk meningkatkan kontras antara teks dan latar belakang.

```
// Penghapusan noise menggunakan GaussianBlur
$denoisedImage = cv\GaussianBlur($grayImage, new cv\Size(5, 5), 0);

// Simpan hasil praproses sebagai file baru
$processedFile = $uploadDir . uniqid() . '_processed.png';
cv\imwrite($processedFile, $denoisedImage);
```

Gambar 4. 16. Proses Penghapusan Noise Pada Gambar

Dilakukannya penghapusan *noise* pada pengambilan gambar OCR juga akan memudahkan perangkat OCR untuk mendeteksi karakter dan lebih mudah untuk dikenali, sehingga diharapkan meningkatkan kualitas hasil pemindaian karakter.

3) Hanya Membaca Angka yang Terdeteksi oleh OCR.

Teks hasil OCR diambil, angka dipisahkan dengan regex (yaitu mencari atau mencocokkan karakter berdasarkan pola tertentu) dan dikonversi ke integer untuk menunjukkan jumlah kWh.

```

if ($data['IsForOcrProcessing'] == 'false') {
    // Hanya angka yang dibaca
    $parsedText = '';
    foreach ($data['ParsedResults'] as $result) {
        $parsedText .= $result['ParsedText'];
    }

    // Mengambil hanya angka yang ada di hasil OCR
    $onlyNumbers = preg_replace('/[^0-9]/', '', $parsedText);
    $kWh = intval($onlyNumbers); // Konversi ke integer untuk kWh

    echo json_encode(['kWh' => $kWh]);
} else {
    http_response_code(500);
    echo json_encode(['error' => 'OCR processing error: ' . $data['ErrorMessage']]);
}

```

Gambar 4. 17. Proses Pembacaan Hanya Angka di OCR

Pembacaan hanya angka untuk pengambilan gambar meteran listrik menggunakan OCR dilakukan supaya OCR akan memindai karakter yang ada didalam meteran listrik saja dan tidak mendeteksi karakter-karakter lain yang tidak relevan dengan angka jumlah kWh

D. PROSES PENDAFTARAN DAN LOGIN AKUN USER

Pada bagian ini, dijelaskan mengenai mekanisme pendaftaran dan login akun pengguna yang diterapkan dalam sistem. Proses pendaftaran dan login ini penting untuk memastikan bahwa hanya pengguna yang terdaftar yang dapat mengakses dan memanfaatkan fitur-fitur yang ada pada aplikasi, seperti penghitungan tagihan listrik dan pengelolaan data pelanggan.

1. Proses Pendaftaran Akun User

Proses pendaftaran akun dimulai ketika pengguna mengakses halaman pendaftaran melalui tampilan antarmuka aplikasi. Pengguna diminta untuk mengisi beberapa data penting, seperti nama lengkap, alamat email, nomor pelanggan, penggunaan Kwh terakhir, daya Watt pengguna dan kata sandi yang akan digunakan untuk mengakses akun.

Setelah pengguna mengisi formulir pendaftaran, sistem akan memeriksa kevalidan data yang dimasukkan. Sebagai contoh, sistem memverifikasi format alamat email yang valid dan memastikan kata sandi memiliki panjang yang cukup serta mengandung kombinasi karakter yang aman. Jika data yang dimasukkan valid, pengguna akan diminta untuk menyetujui syarat dan ketentuan penggunaan aplikasi.

Setelah persetujuan diterima, data pendaftaran pengguna disimpan dalam basis data sistem. Sistem kemudian mengirimkan email konfirmasi ke alamat email yang terdaftar untuk memastikan keabsahan dan keaktifan akun pengguna. Pengguna diminta untuk mengklik tautan konfirmasi yang ada

dalam email untuk mengaktifkan akun mereka. Setelah akun terverifikasi, pengguna dapat melanjutkan ke proses login.

2. Proses Login Akun User

Setelah pengguna terdaftar, mereka dapat masuk ke dalam aplikasi dengan menggunakan alamat email atau nomor akun dan kata sandi yang telah didaftarkan. Proses login dimulai dengan pengguna memasukkan kredensial mereka pada halaman login.

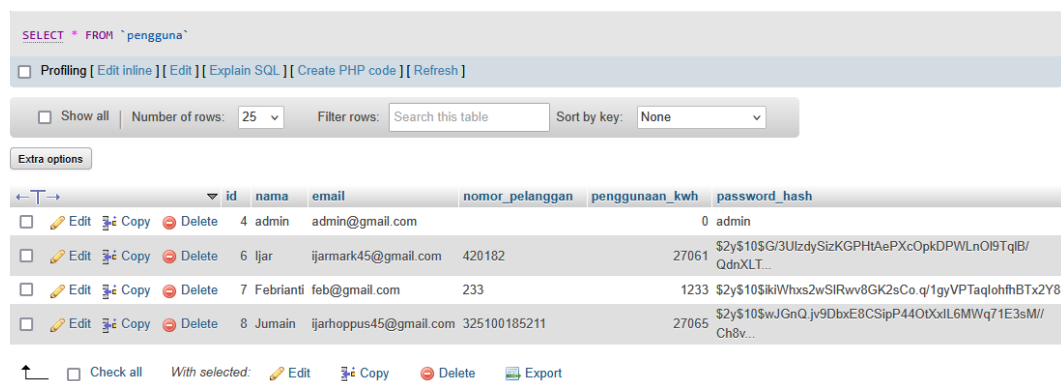
Sistem kemudian memverifikasi apakah alamat email dan kata sandi yang dimasukkan sesuai dengan data yang terdaftar dalam basis data. Jika kredensial valid, pengguna akan diarahkan ke halaman utama aplikasi, di mana mereka dapat memanfaatkan fitur-fitur yang telah disediakan, seperti penghitungan tagihan listrik menggunakan OCR.

3. Keamanan dan Privasi

Keamanan dan privasi data pengguna menjadi hal yang sangat penting dalam aplikasi ini. Oleh karena itu, untuk melindungi informasi sensitif seperti kata sandi, sistem menggunakan enkripsi yang kuat sehingga kata sandi akun pengguna tidak dapat dilihat oleh siapa pun, termasuk oleh admin sistem. Setiap kali pengguna login, aplikasi akan menerapkan sesi yang aman untuk memastikan bahwa hanya pengguna yang sah yang dapat mengakses akun mereka.

4. Data Sistem

Data hasil dari pendaftaran yang dilakukan oleh pengguna akan disimpan ke database MySQL dan jika ada perubahan yang dilakukan oleh user terkait pencatatan tagihan meteran listrik, data hasil sebelumnya akan digantikan oleh data baru yang sudah diinput melalui pedeteksi OCR.



SELECT * FROM `pengguna`

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	id	nama	email	nomor_pelanggan	penggunaan_kwh	password_hash
☐ Edit Copy Delete	4	admin	admin@gmail.com		0	admin
☐ Edit Copy Delete	6	Ijar	ijarmark45@gmail.com	420182	27061	\$2y\$10\$G/3UtzdySizKGPHtAePXcOpkDPWLnOI9TqIB/QdnXLT...
☐ Edit Copy Delete	7	Febrianti	feb@gmail.com	233	1233	\$2y\$10\$kiWfhs2wSIRwv8GK2sCo.q/1gyVPTaqlohfBTx2Y8..
☐ Edit Copy Delete	8	Jumain	ijarhoppus45@gmail.com	325100185211	27065	\$2y\$10\$wJGnQ.jv9DbxE8CSipP44OIXtL6MWq71E3sM//Ch8v...

[↑](#) ☐ Check all | With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

Gambar 4. 18. Database MySQL

E. PROSES PENGENALAN TEKS DAN PENGHITUNGAN TAGIHAN

Dalam penelitian ini, peneliti menggunakan teknologi *Optical Character Recognition* (OCR) untuk membantu membaca dan mengolah teks pada dokumen tagihan listrik. Dengan pendekatan ini, sistem dirancang agar mampu mengenali data penting dari dokumen secara otomatis, seperti nomor pelanggan dan jumlah pemakaian, kemudian menghitung total tagihan dengan akurat berdasarkan informasi yang tersedia.

1. Mekanisme Pengenalan Teks

Sistem ini dirancang dengan prinsip kemudahan dan efisiensi. Mekanisme pengenalan teks yang diterapkan berjalan melalui beberapa tahapan sebagai berikut:

a. Input Dokumen Digital.

Sistem menerima input berupa dokumen tagihan listrik yang telah tersedia dalam format digital seperti PDF atau hasil pemindaian (scan). Dokumen ini tidak perlu diunggah ke server, melainkan langsung diproses pada perangkat pengguna. Hal ini memberikan nilai tambah berupa privasi dan keamanan data pelanggan

b. Proses OCR

Teknologi OCR bekerja dengan membaca karakter pada dokumen digital menggunakan algoritma pendeteksian teks. Dalam hal ini, Tesseract membaca dokumen secara baris demi baris untuk memastikan tidak ada informasi yang terlewat, termasuk elemen-elemen penting seperti nomor pelanggan, pemakaian kWh, dan total tagihan.

c. Ekstraksi Informasi Kunci

Setelah proses OCR selesai, data yang diperoleh hanya berupa angka-angka yang mewakili informasi penting, seperti pemakaian kWh. Tesseract tidak dapat membaca huruf atau karakter lain, hanya angka yang dapat dikenali dan diproses. Sistem kemudian menggunakan logika parsing untuk mengidentifikasi dan memisahkan informasi yang relevan, seperti:

- 1) Pemakaian kWh: Data angka yang menunjukkan jumlah konsumsi listrik dalam satuan kilowatt-jam.
- 2) Total Tagihan: Angka yang menunjukkan jumlah total tagihan, yang dihitung berdasarkan jumlah pemakaian kWh sekarang dikurang dengan pemakaian kWh sebelumnya .

Dengan demikian, meskipun hanya angka yang terdeteksi, sistem dapat secara efektif menghitung dan menyajikan informasi yang dibutuhkan untuk tagihan listrik.

F. PENGHITUNGAN TAGIHAN LISTRIK

Pada bagian ini dijelaskan mengenai proses perhitungan tagihan listrik yang dilakukan oleh sistem berdasarkan data yang diperoleh melalui proses OCR dan data yang diinputkan oleh pengguna pada saat pendaftaran akun. Sistem ini dirancang untuk menghitung tagihan listrik secara otomatis dengan memanfaatkan informasi pemakaian kWh yang didapat dari hasil pemindaian dokumen tagihan listrik dan data pemakaian kWh sebelumnya yang telah disimpan dalam database pengguna.

1. Proses Penghitungan Tagihan Listrik

Proses perhitungan tagihan dimulai dengan memperoleh dua data utama: **pemakaian kWh sekarang** yang diperoleh melalui OCR pada dokumen tagihan listrik, dan **pemakaian kWh sebelumnya** yang diinputkan oleh pengguna pada saat pendaftaran akun mereka. Data pemakaian kWh sebelumnya ini disimpan dalam basis data pengguna untuk digunakan dalam perhitungan tagihan berikutnya.

Langkah pertama dalam perhitungan tagihan adalah menghitung selisih antara pemakaian kWh sekarang dengan pemakaian kWh sebelumnya. Hal ini dilakukan untuk mengetahui total konsumsi listrik yang digunakan dalam periode tertentu. Selisih ini akan menjadi dasar untuk menghitung total biaya yang harus dibayar oleh pengguna. Berikut adalah rumus perhitungan dasar:

$$\text{Total KWH} = \text{Pemakaian KWH Sekarang} - \text{Pemakaian KWH Sebelumnya} \dots\dots\dots (1)$$

Setelah mengetahui total konsumsi listrik dalam satu periode, sistem kemudian menghitung tagihan berdasarkan daya listrik yang terdaftar pada akun pengguna. Tarif per kWh yang dikenakan berbeda-beda tergantung pada daya listrik yang digunakan, dengan perhitungan sebagai berikut:

- a. **450 Watt (Non-subsidi)** : Dibebankan sebesar **352 Rupiah per 1 kWh.**
- b. **900 Watt (Subsidi)** : Dibebankan sebesar **605 Rupiah per 1 kWh.**
- c. **900 Watt (Non-subsidi)** : Dibebankan sebesar **1.352 Rupiah per 1 kWh.**
- d. **1300 Watt dan 2200 Watt** : Dibebankan sebesar **1.445 Rupiah per 1 kWh.**

Dengan mengetahui tarif per kWh berdasarkan daya yang digunakan, sistem kemudian mengalikan jumlah kWh yang digunakan dengan tarif yang berlaku untuk daya listrik pengguna. Rumus perhitungan tagihan menjadi:

$$\text{Tagihan Listrik} = \text{Total kWh} \times \text{Tarif Per kWh} \dots\dots\dots (2)$$

Setelah perhitungan tagihan selesai, sistem akan menampilkan total tagihan yang harus dibayar kepada pengguna. Tagihan ini akan mencakup biaya konsumsi listrik berdasarkan perhitungan di atas. Pengguna dapat melihat rincian biaya, termasuk total penggunaan kWh dan tarif yang dikenakan.

a. Contoh Perhitungan

Sebagai contoh, jika pengguna dengan daya listrik 900 Watt non-subsidi memiliki pemakaian kWh sekarang sebesar 200 kWh, dan pemakaian kWh sebelumnya sebesar 150 kWh, maka selisih pemakaian adalah 50 kWh.

Dengan tarif 1.352 Rupiah per kWh untuk 900 Watt non-subsidi, perhitungan tagihan menjadi:

$$\textbf{Tagihan Listrik} = 50 \text{ kWh} \times \textbf{Rp. 1352} = \textbf{Rp. 67.600}$$

.....(3)

Tagihan yang harus dibayar oleh pengguna adalah 67.600 Rupiah..

G. TAMPILAN APLIKASI

1. Halaman Pendaftaran Akun User

Halaman ini dirancang untuk memudahkan pengguna baru dalam mendaftar ke dalam aplikasi. Pengguna diminta untuk mengisi informasi penting seperti nama lengkap, email, nomor pelanggan, penggunaan kWh sebelumnya, daya Watt pengguna dan kata sandi.

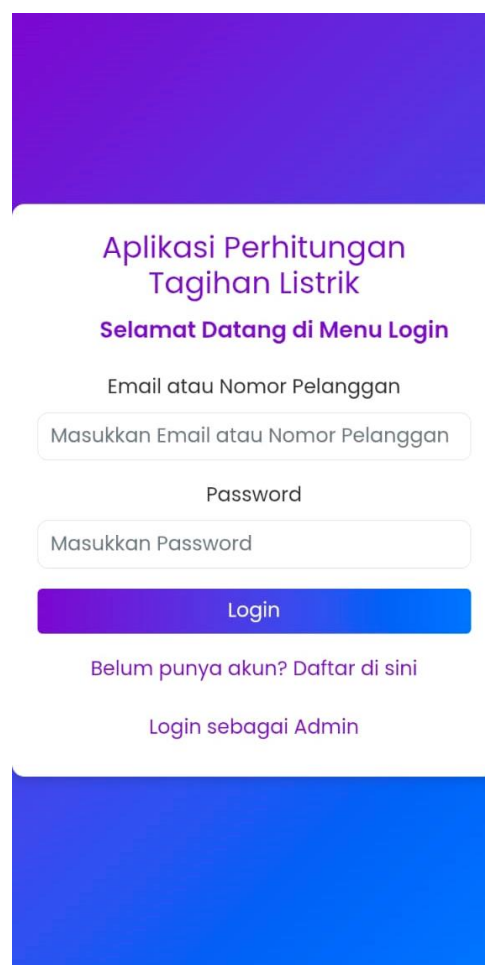
The image displays two versions of the user registration page for the 'Aplikasi Perhitungan Tagihan Listrik'. The left version shows the complete form with the following fields: 'Nama' (Masukkan nama lengkap), 'Email' (Masukkan email), 'Nomor Pelanggan' (Masukkan nomor pelanggan), 'Penggunaan KWH Terakhir' (Masukkan penggunaan KWH terakhir), 'Daya Pengguna' (Pilih daya pengguna), and 'Password' (Masukkan password). A pink 'Register' button and a link 'Sudah punya akun? Login di sini.' are at the bottom. The right version shows the same form but with a modal menu open for the 'Daya Pengguna' field, listing five options: 'Pilih daya pengguna' (selected), '450 Watt', '900 Watt (Subsidi)', '900 Watt (Non Subsidi)', '1300 Watt', and '2200 Watt'. The 'Register' button in this version is dark red.

Gambar 4. 19. Halaman Registrasi Akun User

Formulir ini dilengkapi dengan validasi untuk memastikan data yang dimasukkan sudah sesuai dan lengkap.

2. Halaman Login User

Untuk masuk ke aplikasi, pengguna harus memasukkan nomor pelanggan atau email dan kata sandi yang telah didaftarkan sebelumnya. Halaman login ini dirancang sesederhana mungkin, dengan kolom input yang jelas dan tombol login yang mudah diakses.



The screenshot shows a login interface for an application titled 'Aplikasi Perhitungan Tagihan Listrik'. The page has a purple header and a blue footer. The main content area is white and contains the following elements:

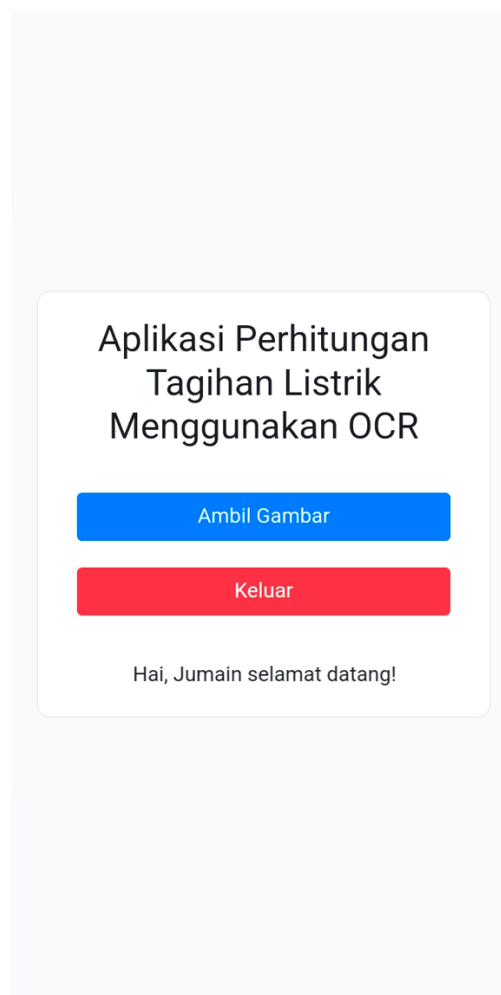
- Title: 'Aplikasi Perhitungan Tagihan Listrik' in purple.
- Greeting: 'Selamat Datang di Menu Login' in purple.
- Input Fields:
 - Label: 'Email atau Nomor Pelanggan'.
 - Input box: 'Masukkan Email atau Nomor Pelanggan'.
 - Label: 'Password'.
 - Input box: 'Masukkan Password'.
- Login Button: A blue button labeled 'Login'.
- Registration Link: 'Belum punya akun? Daftar di sini' in purple.
- Admin Login Link: 'Login sebagai Admin' in purple.

Gambar 4. 20. Halaman Login User

Keamanan kata sandi dijaga dengan enkripsi, sehingga kata sandi pengguna tidak dapat dilihat oleh siapa pun, termasuk admin.

3. Halaman Menu Utama

Halaman utama aplikasi menampilkan dua tombol utama: "Tes Realtime", dan "Logout". Halaman ini dirancang dengan judul aplikasi yang jelas dan tombol yang cukup besar untuk memudahkan interaksi pengguna.



Gambar 4.21. Halaman Menu Utama

- a. **Ambil Gambar** mengaktifkan kamera untuk memindai dokumen tagihan listrik secara langsung.
- b. **Keluar** memungkinkan pengguna untuk keluar dari aplikasi.

4. Halaman Ambil Gambar

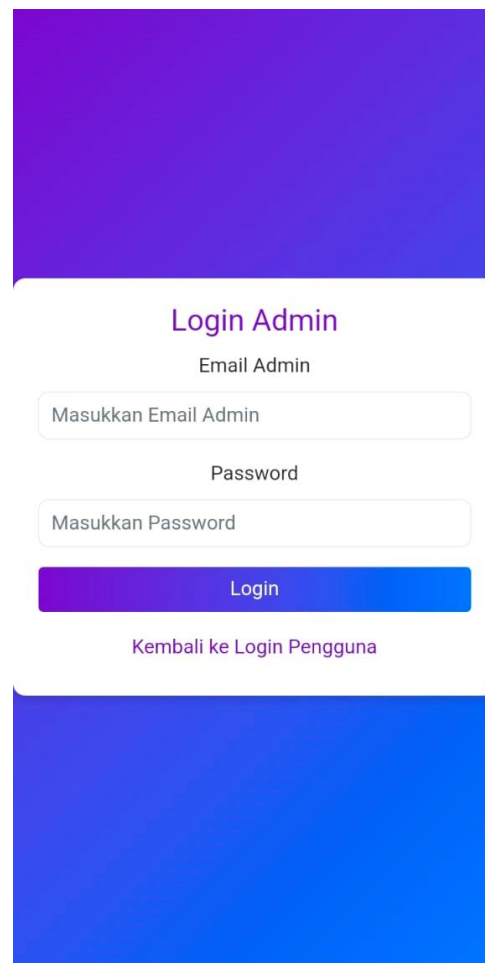
Halaman ini memungkinkan pengguna untuk melihat jumlah penggunaan daya listrik yang mereka gunakan di rumah dengan menghitung penggunaan kWh pengguna pada bulan lalu dan jumlah meteran sekarang.



Gambar 4.22. Halaman Ambil Gambar

5. Halaman Login Admin

Admin dapat masuk ke aplikasi menggunakan email dan kata sandi khusus yang telah ditentukan. Halaman ini juga dilengkapi dengan kolom input yang jelas dan tombol login yang mudah digunakan.



Login Admin

Email Admin

Masukkan Email Admin

Password

Masukkan Password

Login

[Kembali ke Login Pengguna](#)

Gambar 4.23. Halaman Login Admin

Setelah berhasil login, admin dapat melihat daftar pengguna yang terdaftar beserta informasi penting seperti nama, email, nomor pelanggan, penggunaan KWh, dan kata sandi yang telah dienkripsi.

6. Halaman Pengelolaan Data Pengguna

Pada halaman ini, admin dapat melihat dan mengelola data pengguna. Admin dapat melakukan perubahan data pengguna seperti mengubah penggunaan KWh atau menghapus akun pengguna jika diperlukan.

Selamat Datang Admin

Daftar Pengguna

Nama	Email	Nomor Pelanggan	Penggunaan KWH	Daya (Watt)	Aksi
Febrianti	feb@gmail.com	233	1233	1300	Edit Hapus
Jumain	ijarhoppus45@gmail.com	325100185211	27065	450	Edit Hapus
rifki	rifki@gmail.com	233	2200	900 Non Subsidi	Edit Hapus
loli	loli@gmail.com	14422	1222	1300	Edit Hapus

[Logout](#)

Gambar 4.24. Halaman Dashboard Admin

Semua perubahan yang dilakukan oleh admin akan langsung diperbarui pada basis data.

H. SKENARIO PENGUJIAN LANGSUNG

Pada tahap pengujian ini, aplikasi diuji dengan dua skenario yang berbeda untuk menguji kemampuan sistem dalam mengenali penggunaan KWh dari gambar tagihan listrik yang diberikan. Pengujian ini bertujuan untuk memastikan bahwa aplikasi dapat memberikan hasil yang akurat meskipun kondisi gambar berbeda, baik gambar yang jelas maupun gambar yang kabur atau blur. Berikut adalah dua skenario pengujian yang dilakukan:

1. Pengujian Terhadap Gambar yang Jelas

Pada skenario pertama, gambar tagihan listrik yang digunakan memiliki kualitas tinggi dan jelas. Gambar tersebut memiliki kontras yang baik, dengan teks yang mudah terbaca oleh sistem OCR. Pencahayaan yang ideal dan pemindaian yang tajam memastikan bahwa informasi yang tertera pada dokumen dapat dengan mudah diekstraksi tanpa kesalahan. Pengujian ini bertujuan untuk mengukur seberapa akurat aplikasi dapat mengenali dan mengekstrak informasi dari gambar dengan kualitas terbaik. Diharapkan, gambar dengan kualitas tinggi akan memberikan hasil yang optimal, dimana sistem dapat mendeteksi dan membaca angka penggunaan KWh dengan tepat, serta menghitung tagihan listrik dengan akurat.

Tabel 4. 2. Pengujian Langsung dengan Kondisi Gambar Jelas

Gambar	Angka	Keterangan
	25931	OCR Berhasil Mendeteksi Angka dengan Benar yaitu “25931”

Langkah Pengujian:

1. Pengguna mengarahkan kamera ke gambar tagihan listrik yang jelas dari galeri perangkat.
2. Sistem mengaplikasikan teknologi OCR untuk membaca teks dalam gambar tersebut.
3. Proses OCR berhasil mengidentifikasi data penggunaan KWh yang tercetak dalam dokumen dan menyimpannya.
4. Hasil yang diperoleh akan menunjukkan angka penggunaan KWh yang akurat, dan sistem akan menghitung tagihan listrik berdasarkan tarif yang telah ditentukan.

2. Pengujian Terhadap Pengujian Terhadap Gambar yang Kurang Jelas / Blur

Pada skenario kedua, gambar yang digunakan memiliki kualitas rendah dan kabur atau blur. Gambar ini mungkin disebabkan oleh pengambilan gambar yang tidak stabil, pencahayaan yang buruk, atau kualitas pemindaian yang rendah. Pengujian ini bertujuan untuk menguji sejauh mana sistem dapat menangani kondisi gambar yang kurang ideal dan tetap memberikan hasil yang akurat.

Tabel 4. 3. Pengujian Langsung dengan Kondisi Gambar Kurang Jelas / Blur

Gambar	Angka	Keterangan
	17057	OCR Gagal Mendeteksi Angka dengan Benar yaitu “17072”

Meskipun gambar blur, sistem harus tetap mencoba untuk membaca dan mengenali angka penggunaan KWh. Hasil penghitungan tagihan listrik bisa sedikit berbeda dari yang sebenarnya, tergantung pada akurasi OCR dalam mengenali data dari gambar yang kabur.

1. Perbandingan Hasil Skenario Pengujian

Setelah kedua skenario pengujian dilakukan, perbandingan antara gambar yang jelas dan gambar yang blur memberikan wawasan tentang kemampuan sistem OCR dalam menghadapi berbagai kondisi gambar. Pada gambar yang jelas, hasil OCR umumnya lebih akurat, dengan sedikit atau tanpa kesalahan dalam pengenalan angka penggunaan KWh. Sebaliknya, pada gambar yang blur, hasil OCR dapat mengalami kesalahan pengenalan, yang dapat memengaruhi perhitungan tagihan listrik.

3. Pengujian Akurasi Keberhasilan Dan Durasi Deteksi Pengambilan Gambar OCR Mendeteksi Angka Pada Meteran Listrik

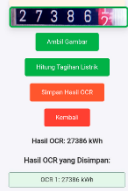
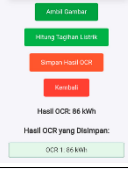
Untuk menganalisis performa OCR dalam mengenali 5 digit awal pada meteran listrik, telah dilaksanakan pengambilan gambar sebanyak 10 kali dengan jarak sorot kamera belakang antara angka meteran listrik yang relatif sama di 10 pengambilan gambar yang dilakukan yaitu antara **5-15 centimeter** dan di arahkan ke area yang mencakup angka meteran listriknya saja. Pengujian dilakukan dengan memakai dua *device* HP Android dengan spesifikasi kamera belakang berbeda yaitu:

1. Vivo Y21: $f/2.2$ (13MP) + $f/2.4$ (2MP) mp $f/2.2$

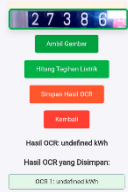
2. Redmi 12C: 16 MP, $f/1.8$, (wide), 1/2.76", PDAF

Pendeteksian dilakukan dengan metode pengujian yang sistematis dan dibawah ini evaluasi akurasi dengan menghitung rata-rata akurasi keberhasilan dan durasi pengambilan gambar melalui OCR.

Tabel 4. 4. Hasil Pengujian Keberhasilan dan Durasi Pengambilan Gambar OCR Mendeteksi Angka Meteran Listrik Menggunakan *Device* HP Vivo Y21

Meteran Listrik	Pengambilan Gambar OCR	Hasil OCR	Angka Sebenarnya	Jumlah Angka Benar	Keberhasilan	Durasi Pendeteksian Angka
		27386	27386	5	Ya	3 Detik
		27386	27386	5	Ya	4 Detik
		86	27386	2	Tidak	4 Detik
		27386	27386	5	Ya	5 Detik
		Undefined kWh	27386	0	Tidak	6 Detik

Lanjutan Tabel 4.4

Meteran Listrik	Pengambilan Gambar OCR	Hasil OCR	Angka Sebenarnya	Jumlah Angka Benar	Keberhasilan	Durasi Pendeteksian Angka
		Undefined kWh	27386	0	Tidak	5 Detik
		27386	27386	0	Ya	4 Detik
		3652	27386	2	Tidak	6 Detik
		Undefined kWh	27386	0	Tidak	5 Detik
		27386	27386	5	Ya	5 Detik

Tabel 4. 5. Hasil Pengujian Keberhasilan dan Durasi Pengambilan Gambar OCR Mendeteksi Angka Meteran Listrik Menggunakan *Device* HP Redmi 12C

Meteran Listrik	Pengambilan Gambar OCR	Hasil OCR	Angka Sebenarnya	Jumlah Angka Benar	Keberhasilan	Durasi Pendeteksian Angka
		27385	27385	5	Ya	3 Detik
		27385	27385	5	Ya	3 Detik
		27385	27385	5	Ya	4 Detik
		Undefined kWh	27385	0	Tidak	5 Detik
		27385	27385	5	Ya	4 Detik
		Undefined kWh	27385	0	Tidak	5 Detik
		27385	27385	5	Ya	4 Detik

Lanjutan Tabel 4.5

Meteran Listrik	Pengambilan Gambar OCR	Hasil OCR	Angka Sebenarnya	Jumlah Angka Benar	Keberhasilan	Durasi Pendeteksian Angka
		Undefined kWh	27385	0	Tidak	6 Detik
		385	27385	3	Tidak	4 Detik
		27385	27385	5	Ya	3 Detik

Parameter yang diukur berdasarkan hasil rata-rata keberhasilan akurasi, dan juga rata-rata waktu deteksi pengambilan gambar OCR pada 5 angka pertama meteran listrik, rumus untuk mengetahui akurasi keberhasilan sebagai berikut:

$$Keberhasilan = \begin{cases} \text{"Ya"} & \text{Jika 5 angka benar} \\ \text{"Tidak"} & \text{Jika 4 angka atau kurang benar} \end{cases}$$

Sedangkan untuk menghitung rata-rata keberhasilan:

$$Keberhasilan = \frac{\sum \text{Total Keberhasilan "Ya"}}{\text{Jumlah pengambilan gambar}} \times 100\%$$

Jadi dengan rumus diatas, dapat diketahui tingkat akurasi keberhasilan pengambilan gambar OCR pada angka meteran listrik, kemudian hasil akurasi dari setiap 10 pengambilan gambar OCR pada angka meteran listrik masing-masing di kedua *device* HP android dapat dihitung rata-rata akurasi keberhasilannya dengan rumus berikut:

- a. Hasil dari rata-rata keberhasilan 10 pengambilan gambar OCR pada meteran listrik di HP Vivo Y21

$$\text{Rata - rata keberhasilan} = \frac{5}{10} \times 100\% = 50\%$$

- b. Hasil dari rata-rata keberhasilan 10 pengambil gambar pada meteran listrik di HP Redmi 12C:

$$\text{Rata - rata keberhasilan} = \frac{6}{10} \times 100\% = 60\%$$

Dan dapat diketahui, akurasi keberhasilan OCR setiap 10 pengambilan gambaran meteran listrik berdasarkan dari hasil rata-rata keberhasilannya dikedua hp mendapatkan hasil yang berbeda yaitu di HP Vivo Y21 meraih 50% akurasi keberhasilan sedangkan HP Redmi 12C mendapatkan akurasi keberhasilan sebesar 60%. Dapat diketahui juga berapa lama durasi pengambilan gambar OCR pada angka meteran listrik dengan rumus sebagai berikut

$$\text{Rata - rata durasi} = \frac{\sum \text{Durasi deteksi per pengambilan gambar}}{\text{Jumlah pengambilan gambar}}$$

Dengan adanya juga rumus untuk mengetahui hasil rata-rata durasi pengambilan gambar OCR mendeteksi angka pada meteran listrik disetiap 10 pengambilan gambar dikedua *device* HP Android, maka hasil dari rata-rata durasinya adalah:

- a. Hasil dari rata-rata keberhasilan 10 pengambilan gambar OCR pada meteran listrik di HP Vivo Y21

$$Rata - rata \ durasi = \frac{3 + 4 + 4 + 5 + 6 + 5 + 4 + 6 + 5 + 5}{10} = 4,7 \text{ detik}$$

- b. Hasil dari rata-rata keberhasilan 10 pengambilan gambar OCR pada meteran listrik di HP Redmi 12C

$$Rata - rata \ durasi = \frac{3 + 3 + 4 + 5 + 4 + 5 + 4 + 6 + 4 + 3}{10} = 4,1 \text{ detik}$$

Jadi rata-rata durasi berapa lama pengambilan gambar OCR mendeteksi angka pada meteran listrik di kedua *device* HP Android yaitu pada HP Vivo Y12 mendapatkan rata-rata lama dengan durasi 4,7 sedangkan HP Redmi 12C mendapatkan rata-rata durasi yaitu 4,1 detik

Adapun faktor yang mempengaruhi seberapa lamanya durasi OCR dan juga mengapa OCR tidak dapat mendeteksi angka pada pengambilan gambar meteran listrik yang mengakibatkan OCR mengeluarkan data *undefined* kWh yaitu bisa seperti kualitas gambar yang buruk yang dapat mencakup gambar yang buram, terlalu gelap, memiliki resolusi yang rendah dan juga adanya noise pada gambar mempengaruhi besar akurasi OCR untuk mendeteksi karakter angka pada meteran listrik.

4. Pengujian Menggunakan HP Berbeda Terkait *Pixel* Kamera

Pengujian ini ditujukan untuk menganalisis pengaruh resolusi kamera di HP berbasis Android yang berbeda terhadap akurasi OCR dalam mendeteksi angka pada meteran listrik. Pengambilan gambar dilakukan dengan dua jenis HP yang tentunya memiliki spesifikasi kamera belakang yang berbeda yaitu:

1. Vivo Y21: $f/2.2$ (13MP) + $f/2.4$ (2MP) mp $f/2.2$
2. Redmi 12C: 16 MP, $f/1.8$, (wide), $1/2.76''$, PDAF

Parameter yang diukur dari pengujian dengan menggunakan dua HP Android yang berbeda tentunya ditarik berdasarkan akurasi, waktu durasi deteksi dan juga kualitas gambar dalam pengambilan gambar OCR untuk mendeteksi angka pada meteran listrik.

Tabel 4. 6. Pengujian Menggunakan HP Yang Berbeda Terkait *Pixel* Kamera

No	Hasil (Resolusi Kamera)	Pengambilan Gambar	Rata-rata akurasi keberhasilan	Waktu Deteksi Rata-rata	Kualitas Gambar (Subjektif)
1.	Vivo Y21 ($f/2.2$ (13MP) + $f/2.4$ (2MP) mp $f/2.2$)		50%	4,7 Detik	Cukup

Lanjutan Tabel 4.6

No	Hasil (Resolusi Kamera)	Pengambilan Gambar	Rata-rata akurasi keberhasilan	Waktu Deteksi Rata-rata	Kualitas Gambar (Subjektif)
2.	Redmi 12C (16 MP, <i>f</i> /1.8, (wide), 1/2.76", PDAF)		60%	4,1 Detik	Baik

Dengan menggunakan pengujian akurasi keberhasilan dan lama durasi pengambilan gambar OCR pada angka meteran listrik sebanyak 10 kali di kedua HP yang berbeda, dapat disimpulkan bahwa HP Redmi 12C dengan spesifikasi kamera yang lebih baik yaitu 16 *megapixel* menghasilkan akurasi yang tertinggi 60% dibandingkan Vivo Y21 yang mempunyai spesifikasi kamera 13 *megapixel* dengan akurasi 50%, sedangkan untuk durasi berapa lamanya pengambilan gambar OCR pada angka meteran listrik, HP Redmi 12C lebih cepat dengan durasi 4,1 detik yang berselisih 0,6 detik unggul jika dibandingkan dengan HP Vivo Y21 yang durasi deteksinya yaitu 4,7 detik.

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, telah dibuat aplikasi penghitungan tagihan listrik menggunakan deteksi *Optical Character Recognition* menggunakan *Tesseract OCR* berbasis *webview* yang mencakup dua peran, yaitu pengguna dan admin. Pengguna dapat menginput data sesuai dengan informasi tagihan listrik mereka dan kemudian melakukan perhitungan tagihan berdasarkan data yang diinput dengan angka pada meteran listrik, sedangkan admin dapat mengakses data pengguna jika dilakukan perubahan terhadap data tersebut, dengan hasil pengujian:

1. Dengan dilakukannya 10 pengujian pengambilan gambar OCR pada angka kWh meteran listrik untuk mengetahui akurasi keberhasilan disetiap penggunaan dua *device* HP Android dengan spesifikasi kamera belakang yang berbeda yaitu 13 *megapixel* dan 16 *megapixel*, dapat disimpulkan bahwa *device* dengan spesifikasi kamera 16 *megapixel* menghasilkan rata-rata akurasi keberhasilan sebesar 60% dibandingkan spesifikasi kamera belakang 13 *megapixel* yang rata-rata keberhasilan akurasinya sebesar 50% saja.
2. Di 10 pengujian pengambilan gambar OCR pada angka kWh meteran listrik antara dua *device* yang berbeda itu juga, dapat disimpulkan *device* dengan spesifikasi kamera 16 *megapixel* memiliki rata-rata waktu deteksi yaitu 4,1

detik yang berselisih 0,6 detik lebih unggul dibandingkan dengan *device* kamera dengan spesifikasi 13 *megapixel* yang rata-rata waktu deteksinya yaitu 4,7 detik.

3. Pengujian dengan skenario gambar yang blur ditambah lagi dengan beberapa faktor yang dapat mempengaruhi hasil OCR seperti adanya noise dan pencahayaan yang kurang memadai menunjukkan bahwa kualitas gambar sangat memengaruhi akurasi OCR, yang mengindikasikan bahwa pengambilan gambar yang baik menjadi faktor penting dalam keberhasilan aplikasi.
4. Sistem ini mampu menghitung tagihan listrik secara otomatis berdasarkan tarif penggunaan Watt pengguna dengan mengacu pada data penggunaan KWh yang diperoleh dari OCR dan data awal yang dimasukkan oleh pengguna saat pendaftaran.
5. Enkripsi kata sandi memastikan keamanan data pengguna menggunakan algoritma *hash* MD5 untuk memastikan kerahasiaan dan integritas informasi.

Dengan pencapaian ini, aplikasi menunjukkan potensi besar dalam meningkatkan efisiensi dan penanganan penghitungan tagihan listrik secara otomatis, sekaligus memberikan pengalaman pengguna yang lebih baik dalam mengelola data listrik mereka.

B. Saran

Meskipun aplikasi telah mencapai hasil yang memuaskan, beberapa saran berikut diusulkan untuk pengembangan lebih lanjut:

1. Peningkatan Akurasi OCR

Penggunaan model pelatihan tambahan atau algoritma OCR yang lebih canggih atau muthakhir dari *Tesseract* yang dapat membantu meningkatkan akurasi dalam pengambilan angka pada meteran listrik seperti *Google Cloud Vision API* yang berbasis cloud yang didukung oleh teknologi *machine learning Google* atau *Microsoft Azure Cognitive Services (Computer Vision)* yang menawarkan kemampuan OCR yang terintegrasi dengan layanan cloud *Azure* yang terus ditingkatkan dengan teknologi *machine learning* terbaru dari *Microsoft*.

2. Optimalisasi UI/UX

Tampilan antarmuka pengguna dapat disempurnakan lebih lanjut untuk memberikan pengalaman yang lebih intuitif, termasuk panduan pengambilan gambar agar kualitasnya sesuai dengan kebutuhan sistem OCR.

3. Integrasi dengan Sistem Pembayaran

Aplikasi dapat dikembangkan lebih lanjut untuk mengintegrasikan fitur pembayaran tagihan listrik secara langsung, memberikan nilai tambah bagi pengguna dengan adanya detail lebih lanjut seperti data pengguna yang diinput pada dilakukannya pendaftaran yang dimana pengguna dapat mengunduh data tersebut dalam bentuk struk pembayaran pengguna.

Dengan implementasi saran-saran tersebut, diharapkan aplikasi ini dapat terus berkembang dan memberikan manfaat yang lebih luas, baik bagi pengguna individu maupun institusi terkait.

DAFTAR PUSTAKA

- A. Firdaus, M. S. Kurnia, T. Shafera, dan W. I. Firdaus, "Implementasi Optical Character Recognition (OCR) Pada Masa Pandemi Covid-19," vol. 13, no. 2, 2021.
- A. Setiawan dan H. Sujaini, "Implementasi Optical Character Recognition (OCR) pada Mesin Penerjemah Bahasa Indonesia ke Bahasa Inggris," vol. 5, no. 2, 2017.
- I. H. Al Amin dan A. Aprilino, "IMPLEMENTASI ALGORITMA YOLO DAN TESSERACT OCR PADA SISTEM DETEKSI PLAT NOMOR OTOMATIS," *J. Teknoinfo*, vol. 16, no. 1, hlm. 54, Jan 2022, doi: 10.33365/jti.v16i1.1522.
- K. Apriyanti dan T. Wahyu Widodo, "Implementasi Optical Character Recognition Berbasis Backpropagation untuk Text to Speech Perangkat Android," *IJEIS Indones. J. Electron. Instrum. Syst.*, vol. 6, no. 1, hlm. 13, Apr 2016, doi: 10.22146/ijeis.10767.
- R. Smith, "An Overview of the Tesseract OCR Engine," dalam *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2*, Curitiba, Parana, Brazil: IEEE, Sep 2007, hlm. 629–633. doi: 10.1109/ICDAR.2007.4376991.
- T. H. W. Pramesti, "PENGENALAN KARAKTER TEKS MENGGUNAKAN METODE NEURAL NETWORK BACKPROPAGATION".
- verihubs, "Memahami Apa Itu OCR dan Bagaimana Cara Kerjanya," *Verihubs*, 25 Juli 2022. <https://verihubs.com/blog/memahami-apa-itu-ocr/> (diakses 22 Juni 2023).
- Y. Andreas, K. Gunadi, dan A. N. Purbowo, "Implementasi Tesseract OCR untuk Pembuatan Aplikasi Pengenalan Nota pada Android".
- Z. AFIFA, "IMPLEMENTASI METODE GAUSSIAN FILTER UNTUK PENGHAPUSAN NOISE PADA CITRA MENGGUNAKAN GPU," <http://etheses.uin-malang.ac.id>.
- "Image Processing: Pengertian dan Langkah-langkahnya," 21 September 2022. <https://www.trivusi.web.id/2022/09/image-processing.html> (diakses 22 Juni 2023).
- "Investasi dan Miliki Bisnis Menguntungkan Mulai dari 1 Jutaan di LandX." <https://landx.id/> (diakses 23 Juni 2023).

“Cara Menghitung Pemakaian Listrik Bulanan.” <https://berita.99.co/cara-menghitung-pemakaian-listrik-prabayar/> (diakses 23 Juni 2023).