

BAB II TINJAUAN PUSTAKA

Dalam usaha penulis untuk menjelaskan pemahaman terhadap konsep dan perkembangan dalam bidang yang relevan dengan penelitian ini serta memberikan pandangan yang lebih luas dan komprehensif, hal tersebut disampaikan melalui bab tinjauan pustaka.

2.1 Pencapaian Terdahulu

Penelitian ini perlu merujuk pada studi-studi sebelumnya yang memiliki relevansi untuk memperkuat dasar teoritisnya dan mengidentifikasi sumbangan literatur dalam domain terkait. Penelitian terdahulu juga membantu memberikan pandangan yang lebih luas dan komprehensif bagi peneliti. Tabel 1.1 mencantumkan pencapaian terdahulu yang memiliki relevansi dengan penelitian ini.

Tabel 1. 1 *Pencapaian Peneliti Terdahulu*

No	Peneliti (Tahun)	Judul	Hasil
1	Zaki Ahmad Fauzi (2023)	Deteksi Identitas KWH Meter Digital Satu Fasa Menggunakan Artificial Neural Network dan Template Matching	Penelitian ini berhasil mengembangkan teknologi deteksi identitas kWh meter digital satu fasa dengan menggunakan Artificial Neural Network dan Template Matching. Program yang dikembangkan mampu mendeteksi merek kWh meter dengan tingkat akurasi 100%. Hasil ini menunjukkan bahwa deteksi menggunakan perangkat yang berbeda memberikan tingkat akurasi yang bervariasi, dengan handphone iPhone Xr memiliki performa terbaik
2	Vera Firmansyah, Vebi Nadhira, Lutfia Silvi, Tesya Apnita Dewi (2020)	Iot Sistem Monitoring Meter Kwh Digital Menggunakan Sensor Ldr Dan Codeigniter Api Service	Penelitian ini mengembangkan prototipe alat monitoring penggunaan energi listrik pada meter kWh digital rumah tangga. Pengguna dapat memantau konsumsi listrik secara real-time, mengatasi kelemahan sistem pembayaran yang hanya mengetahui pemakaian saat pembayaran. Prototipe ini menggunakan sensor Light Dependent Resistor (LDR) untuk mendeteksi kedipan lampu LED pada meter kWh, dikontrol mikrokontroler, dan terhubung ke server melalui API service dengan framework

			CodeIgniter. Hasil pengujian menunjukkan nilai kesalahan pembacaan sebesar 0%, dengan waktu pengiriman data rata-rata 36,23 detik melalui koneksi wifi
3	Elly Mufida, Mochammad Iqbal Adriansyah, Nur Muhammad Ihsan, Rian Septian Anwar (2021)	Perancangan Alat Pendeteksi KWH Meter Berbasis Arduino Uno R3 dan ESP8266	Penelitian ini bertujuan untuk memberikan informasi kepada pengguna mengenai penggunaan energi listrik secara real-time dan meningkatkan kesadaran terhadap pengelolaan daya listrik. Hasil pengukuran ditampilkan pada LCD dan dikirimkan ke aplikasi berbasis web melalui modul ESP8266. Selain itu, alat ini dilengkapi dengan buzzer yang akan memberikan peringatan apabila konsumsi daya listrik melebihi batas maksimal yang telah ditentukan oleh pengguna. Dengan fitur ini, pengguna dapat menetapkan nilai batas penggunaan listrik maksimal dan menerima peringatan jika terjadi pelanggaran batas tersebut, sehingga memungkinkan pengendalian konsumsi energi listrik sesuai kebutuhan.
4	Adrien Petralia, Philippe harpentier, Paul Boniol, Themis Palpanas (2023)	<i>Appliance Detection Using Very Low- Frequency Smart Meter Time Series</i>	Penelitian ini mengevaluasi dan membandingkan klasifikator time series terbaru untuk mendeteksi perangkat rumah tangga pada data meteran pintar dengan frekuensi pengukuran rendah (30 menit). Meskipun klasifikator time series telah banyak diterapkan, belum ada penelitian yang membandingkannya untuk masalah ini. Hasil penelitian menunjukkan bahwa beberapa klasifikator berbasis pembelajaran mendalam memberikan akurasi yang menjanjikan meskipun menggunakan data dengan frekuensi rendah, dan dapat diskalakan untuk koleksi data besar. Namun, penelitian ini juga menunjukkan perlunya perbaikan lebih lanjut untuk meningkatkan akurasi deteksi. Penelitian ini dipublikasikan dalam ACM e-Energy 2023.
5	Dimas Setyawan Ramadhansyah, Arrie Kurniawardhani (2021)	Penelitian Deteksi Pelat Nomor Kendaraan: Kajian Pustaka	Penelitian ini berhasil menganalisis sepuluh literatur terdahulu mengenai metode deteksi pelat nomor kendaraan. Hasilnya menunjukkan bahwa metode deep learning, khususnya YOLO dan CNN, lebih unggul dibandingkan metode image processing seperti Connected Component, SVM, Connected Component Labelling, Transformasi Hough, dan Transformasi Hit

or Miss. Dari literatur yang diteliti, YOLO dan CNN mencapai akurasi dan mean average precision di atas 90%, sementara metode image processing kurang memuaskan dalam real-time processing dan akurasi. Meskipun ada kemajuan, belum ada metode yang berhasil digunakan secara efektif untuk deteksi pelat nomor kendaraan di seluruh negara, disebabkan oleh perbedaan karakteristik pelat nomor kendaraan di setiap negara.

2.2 Tinjauan Teoritis

Penelitian ini didasari oleh beberapa teori dan konsep yang relevan dengan permasalahan yang akan dikaji. Tinjauan teoritis bertujuan untuk memberikan landasan pemahaman yang kuat bagi peneliti dalam menganalisis masalah dan menarik kesimpulan.

2.2.1 Kecerdasan Buatan

Kecerdasan Buatan (*Artificial Intelligence/AI*) merupakan sebuah cabang dalam ilmu komputer yang memusatkan perhatian pada pengembangan mesin atau sistem komputer yang mampu menjalankan tugas-tugas yang umumnya memerlukan kecerdasan manusia. Tujuan utamanya adalah menciptakan mesin yang mampu belajar, berpikir, dan bertindak seperti manusia. AI meliputi berbagai teknik dan metode, seperti pemrograman logika, pemrosesan bahasa alami, dan pengenalan pola. Dalam konteks pemrograman logika, sistem AI menggunakan aturan logika untuk mengambil keputusan berdasarkan informasi yang diberikan.

a) *Machine Learning dan Deep Learning*

Algoritma dan model dalam machine learning dirancang untuk mengidentifikasi pola dalam data serta menghasilkan prediksi atau keputusan berdasarkan pola tersebut. Selain itu, machine learning juga dimanfaatkan dalam pengembangan model yang digunakan untuk menganalisis data dalam bentuk gambar maupun video (Fathurohman, 2021).

Seiring dengan kemajuan dalam bidang *machine learning*, muncul sub-bidang yang dikenal sebagai *Deep Learning*. *Deep learning* adalah cabang dari *machine learning* yang menggunakan jaringan saraf tiruan (artificial neural networks) yang kompleks dan dalam (dikenal sebagai "*deep*") untuk mengekstraksi representasi yang mendasari dari data (Raup, Ridwan, Khoeriyah, Supiana, & Zaqiah, 2022). *Deep learning* memungkinkan komputer untuk belajar secara hierarkis dan otomatis dari data yang besar dan kompleks, seperti gambar, suara, teks, dan lainnya, dengan bantuan jaringan saraf tiruan (Syafa'ah & Lestandy, 2021). Konsep ini berusaha mensimulasikan cara kerja otak manusia dengan menggunakan banyak lapisan *neuron* tiruan yang terhubung dalam jaringan untuk menganalisis dan menginterpretasi data.

b) *Computer Vision*

Computer vision semakin meningkat belakangan ini karena aplikasinya yang luas di berbagai sektor seperti pengawasan cerdas, bidang kesehatan, olahraga, robotika, mobil *self-driving*, dan lain sebagainya. Tugas-tugas visual, seperti pengenalan gambar, pelokalan, dan deteksi, menjadi bagian integral dari pemanfaatan teknologi ini yang telah menunjukkan hasil yang memuaskan dalam kinerja sistem visual. *Computer vision* kini telah diimplementasikan dalam berbagai aspek kehidupan, seperti deteksi pemakaian masker, aplikasi pemindai rumus matematika, hingga estimasi tinggi badan menggunakan kamera. Perkembangan ini menunjukkan bahwa teknologi ini semakin mudah diakses dan diterapkan dalam berbagai bidang. Teknologi *computer vision* bekerja berdasarkan prinsip di mana sistem komputer menganalisis gambar digital 2D untuk mengekstrak berbagai fitur, seperti piksel, luas area, dan perimeter, dengan bantuan algoritma pemrosesan gambar (Dr. Arnita, S.Si., M.Si, Marpaung, F., S.Si., M.Si, Aulia, F., Suryani, N., S.Kom., & Nabila, R. C., .Kom. 2022)..

c) Pengolahan Citra

Secara umum, citra atau gambar merupakan representasi dua dimensi dari cahaya. Dari perspektif matematis, citra dapat dipandang sebagai fungsi kontinu dari intensitas cahaya pada suatu bidang dua dimensi. Ketika sebuah objek

terkena sumber cahaya, sebagian cahaya tersebut dipantulkan dan membentuk citra. Saat ini, penggunaan citra digital telah menjadi hal yang umum, sering kali disebut sebagai gambar, dan banyak perangkat elektronik dirancang khusus untuk menghasilkan citra digital, seperti scanner dan kamera digital. Kualitas citra yang dihasilkan oleh perangkat-perangkat ini dapat bervariasi, karena sering kali dilakukan penyesuaian sesuai dengan kebutuhan pengguna (Iryanto, 2014).

2.2.2 Deteksi Angka (*Number Detection*)

Tujuan utama dari deteksi angka pada citra adalah untuk mengidentifikasi dan menandai lokasi angka pada kWh meter dalam sebuah citra, yang sering kali diwakili oleh persegi panjang yang mengelilingi area digit kWh meter yang terdeteksi. Proses deteksi angka melibatkan pengenalan pola dan struktur yang khas dari citra kWh meter oleh algoritma komputer untuk mengenali dan membedakan digit yang dimaksud dari objek atau latar belakang lainnya.

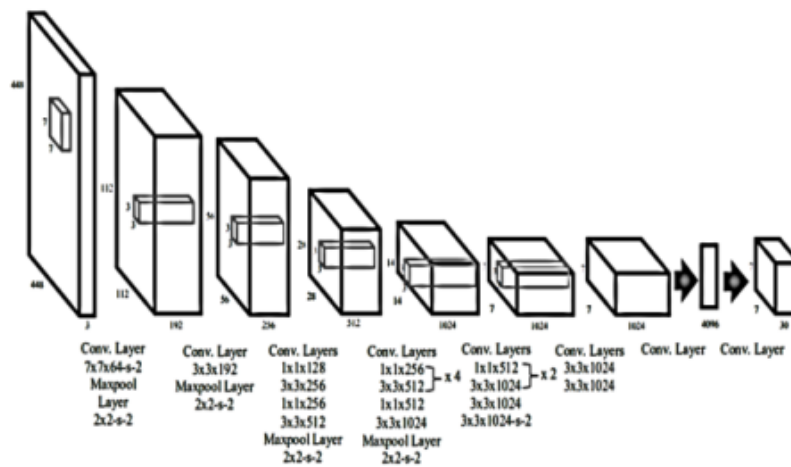
2.2.3 YOLO

Algoritma ini mampu mengidentifikasi dan mengenali objek dalam satu proses pemindaian, sehingga lebih cepat dibandingkan metode deteksi lainnya. Algoritma ini dirancang untuk mendeteksi objek secara langsung dalam gambar atau video dengan kecepatan tinggi (Rahma, Syaputra, Mirza, & Purnamasari, 2021).

Proses operasi YOLO melibatkan pemetaan citra *input* ke dalam beberapa jaringan sel. Langkah awalnya melibatkan pembagian citra menjadi bagian-bagian sel jaringan, dengan jumlah sel jaringan yang dapat dihitung dengan membagi resolusi gambar menggunakan langkah (*stride*) yang telah ditentukan sebelumnya. Tujuan dari langkah ini adalah untuk menghasilkan kotak dengan dimensi 8x8, 16x16, dan 32x32. Dalam praktiknya, gambar biasanya memiliki resolusi 640x640, sehingga langkah yang digunakan adalah 80x80 untuk objek kecil, 40x40 untuk objek sedang, dan 20x20 untuk objek yang lebih besar. YOLO membagi citra *input* menjadi *grid-grid* berukuran $S \times S$ (Agustina & Sukron, 2022), di mana setiap grid bertanggung jawab untuk memprediksi

keberadaan objek, kotak pembatas, dan tingkat keyakinan (*confidence*). *Confidence* digunakan untuk memperkirakan apakah objek ada dalam grid tersebut.

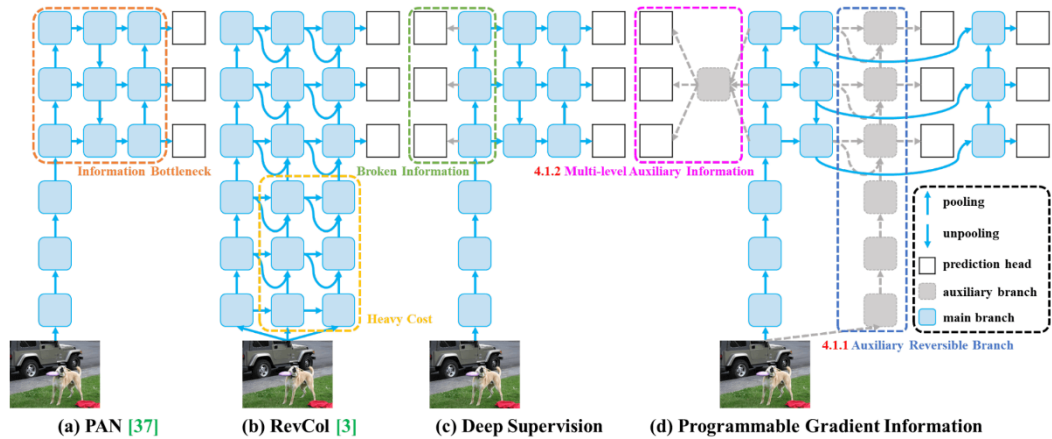
Salah satu keunggulan utama dari YOLO adalah kecepatannya. Dengan menghindari tahap-tahap terpisah, YOLO mampu memberikan hasil deteksi objek secara *real-time* dengan kecepatan yang sangat tinggi, menjadikannya cocok untuk aplikasi seperti deteksi objek dalam video bergerak atau aplikasi deteksi objek dalam waktu nyata.



Gambar 2.1 Arsitektur YOLO

Gambar 2.1 menampilkan detail dari arsitektur utama yang digunakan oleh algoritma YOLO dalam proses deteksi objek. Tiap kotak pembatas memiliki atribut seperti (x , y , w , h , dan *confidence*), di mana x dan y mewakili posisi kotak pembatas, sementara w dan h merepresentasikan dimensi tinggi dan lebar kotak pembatas. Confidence merujuk pada skor IoU yang dihasilkan dari perbandingan antara kotak prediksi dengan kotak referensi sebenarnya (*ground-truth box*). Di setiap grid, dilakukan perhitungan probabilitas prediksi kelas objek.

Algoritma YOLO memiliki arsitektur yang disesuaikan dengan setiap versinya. Perubahan dalam proses pendeteksian objek terjadi setiap kali versi algoritma tersebut ditingkatkan. Arsitektur YOLOv9 terdiri dari beberapa tahap utama yang terlibat dalam pendeteksian objek, yaitu proses *backbone*, *neck*, dan *head*. Rincian dari arsitektur YOLOv9 dapat dilihat pada gambar yang disajikan di bawah ini.



Gambar 2. 2 Arsitektur YOLOv9

a. *Path Aggregation Network* (PAN) [37]

Ini adalah jenis rangkaian konvolutif yang mengumpulkan fitur dari semua skala, dengan menggunakan operasi penghimpunan untuk mengambil informasi dari semua skala dan mengirimkannya ke layer berikutnya.

b. *Reversible Columns* (RevCol) [3]

RevCol adalah jenis rangkaian yang memanfaatkan kolom yang dapat dibalik untuk mengurangi biaya informasi yang hilang saat proses pembelajaran. Ini memungkinkan rangkaian untuk "membalik" aliran data dan mengambil keputusan ulang pada tahap sebelumnya, mengurangi kehilangan informasi.

c. *Deep Supervision*

Hal ini merujuk pada teknik pengawasan yang umum digunakan di dalam rangkaian sampel, yaitu dengan menambahkan cabang pengawasan tambahan pada layer tertentu. Cabang pengawasan ini akan membuat prediksi langsung dan membandingkannya dengan nilai target, dapat membantu mengurangi masalah gradient vanishing dan mewujudkan training yang lebih efisien.

d. *Programmable Gradient Information* (PGI)

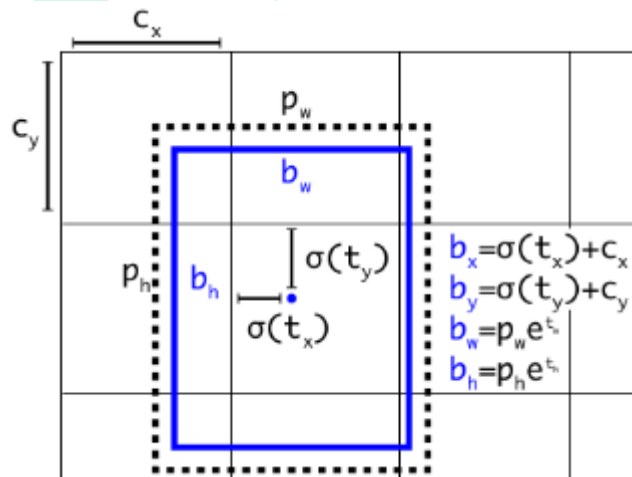
PGI adalah arsitektur yang diusulkan dalam gambar ini, yang terdiri dari tiga komponen utama: (1) cabang utama, yang digunakan untuk inferensi, (2) cabang balik terbatas, yang digunakan untuk menghasilkan gradien yang terpercaya dan diperbarui ke cabang utama, dan (3) informasi bantuan *multi-level*, yang digunakan untuk mengontrol pembelajaran

cabang utama dan mengambil informasi semantik dari *level-level* yang berbeda (Chien-Yao Wang, I-Hau Yeh, Hong-Yuan Mark Liao. 2024)

e. *Head*

Head adalah bagian terakhir dari arsitektur YOLOv9 yang bertanggung jawab untuk melakukan deteksi objek. *Head* ini mengambil fitur-fitur yang telah diproses melalui empat tahap yang sudah dijelaskan diatas, dan menggunakan teknik pemrosesan di level piksel untuk secara efisien menentukan kotak pembatas objek dan kelasnya.

Head juga menghasilkan prediksi berupa koordinat *bounding box* dan probabilitas kelas untuk setiap objek dalam gambar. Hal ini dilakukan dengan menggunakan lapisan-lapisan konvolusi dan aktivasi yang dirancang khusus untuk mendeteksi objek dengan presisi tinggi.



Gambar 2. 3 Bounding Box

Gambar 2.3 menggambarkan konsep *bounding box* yang digunakan untuk melakukan deteksi objek pada input gambar atau video. Pusat dari bounding box adalah titik tengahnya, sedangkan lebarnya adalah jarak antara titik kiri atas dan titik kanan bawah bounding box. Tingginya adalah jarak antara titik atas dan titik bawah dari bounding box. Probabilitas keberadaan objek merupakan probabilitas bahwa objek sebenarnya terdapat di dalam *bounding box* tersebut.

2.24 EasyOCR

EasyOCR adalah paket *Python open-source* yang memfasilitasi proses *Optical Character Recognition (OCR)* dengan pendekatan berbasis *deep learning*. Dibangun di atas *PyTorch* sebagai *backend*, *EasyOCR* memanfaatkan model-model canggih untuk mendeteksi dan mengenali teks dalam gambar. Proses deteksi teks dimulai dengan tahap *preprocessing*, seperti konversi gambar menjadi *grayscale* untuk mempermudah ekstraksi informasi tekstual. Selanjutnya, algoritma *Character Region Awareness for Text Detection (CRAFT)* digunakan untuk mendeteksi wilayah-wilayah yang mengandung teks. Dalam tahap pengenalan karakter, *EasyOCR* menerapkan model *Convolutional Recurrent Neural Networks (CRNN)*, yang terdiri dari tiga komponen utama: ekstraksi fitur menggunakan *ResNet*, pelabelan urutan menggunakan *Long Short-Term Memory (LSTM)*, dan dekoding menggunakan *Connectionist Temporal Classification (CTC)* untuk menghasilkan output yang akurat (Baek, J., Lee, Y., & Kim, S, 2019).

Model *CRNN* yang diterapkan pada *EasyOCR* memiliki kemampuan untuk mengenali teks dalam berbagai kondisi dan latar belakang gambar, yang menjadikannya sangat berguna dalam berbagai aplikasi seperti pemrosesan dokumen otomatis dan analisis gambar berbasis teks. Dengan menggabungkan teknik-teknik *deep learning* canggih, *EasyOCR* menawarkan solusi yang efisien dan akurat dalam mengatasi tantangan *OCR*, terutama pada teks yang kompleks dan tidak terstruktur. Implementasi *CRAFT* dalam tahap deteksi teks memungkinkan model untuk lebih sensitif terhadap berbagai bentuk dan orientasi teks, sementara komponen *CRNN* mendukung pengenalan urutan karakter yang lebih fleksibel dan efektif (Baek, J., Lee, Y., & Kim, S, 2019).

2.2.5 Sistem Deteksi Angka

Sistem deteksi angka merupakan suatu sistem komputer atau perangkat lunak yang dirancang khusus untuk mengidentifikasi dan mengenali angka-angka dalam gambar atau video. Tujuannya adalah untuk mengenali lokasi dan ciri-ciri angka pada kWh meter dalam sebuah citra. Beberapa alat yang digunakan oleh peneliti dalam merancang sistem deteksi angka antara lain:

a) Python

Python adalah bahasa pemrograman tingkat tinggi yang serbaguna dan mudah dipahami. Dikembangkan pertama kali pada akhir 1980-an oleh Guido van Rossum, Python dirancang dengan fokus pada keterbacaan kode dan sintaks yang jelas. Hal ini membuatnya menjadi bahasa yang cocok untuk pemula dan juga *programmer* berpengalaman. Python digunakan secara luas dalam berbagai bidang pengembangan perangkat lunak, termasuk pengembangan web, analisis data, kecerdasan buatan, pemrosesan bahasa alami, dan masih banyak lagi. Keunggulan Python antara lain adalah sintaks yang sederhana, dukungan terhadap paradigma pemrograman yang beragam, serta tersedianya berbagai pustaka dan *framework* yang memperluas fungsionalitasnya. Python juga dikenal dengan komunitas yang aktif dan ramah, serta dokumentasi yang kaya, membuatnya menjadi pilihan utama bagi banyak pengembang perangkat lunak.

b) OpenCV

OpenCV (*Open Source Computer Vision*) adalah perpustakaan perangkat lunak open-source yang digunakan untuk pengolahan citra dan analisis komputer vision. Tujuannya adalah untuk memberikan alat dan fungsi yang memungkinkan pengembang perangkat lunak untuk mengembangkan aplikasi yang melibatkan pengolahan gambar, deteksi objek, pengenalan pola, dan analisis video.

c) Visual Studio Code

Visual Studio Code (VS Code) adalah sebuah Lingkungan Pengembangan Terpadu (*Integrated Development Environment/IDE*) yang dirancang khusus untuk pengembangan aplikasi dengan berbagai bahasa pemrograman, termasuk Python. Dikembangkan oleh Microsoft, VS Code menyediakan berbagai fitur penting bagi pengembang, seperti code completion, debugging, dan integrasi dengan sistem kontrol versi.

Fitur-fitur ini sangat membantu pengembang dalam menulis, menguji, dan memperbaiki kode. Antarmuka pengguna VS Code dirancang dengan baik dan ramah pengguna, memudahkan pengembang dalam mengakses fitur-fitur yang

dibutuhkan. Selain itu, VS Code memiliki kemampuan untuk memantau dan memperbaiki kode secara otomatis, membantu pengembang untuk memastikan bahwa kode yang ditulis adalah kode yang baik.

2.2.6 Confusion matrix

Confusion matrix adalah metode evaluasi yang mengukur kinerja model klasifikasi dengan membandingkan hasil prediksi dengan data sebenarnya. Matriks ini menunjukkan jumlah prediksi yang benar maupun salah serta jenis kesalahan yang terjadi. Confusion matrix sangat bermanfaat dalam mengevaluasi efektivitas dan kelemahan model klasifikasi serta mendukung pengambilan keputusan yang lebih baik dalam pengembangan model (Rahayu, Prianto, & Novia, 2021).

2.2.6.1 Akurasi

Akurasi merupakan sebuah pendekatan evaluasi yang didasarkan pada seberapa baik nilai prediksi sesuai dengan nilai aktual. Dengan menganalisis jumlah data yang terklasifikasi dengan benar, kita dapat menentukan estimasi tingkat keakuratan dari prediksi tersebut.

$$\text{Akurasi} = \frac{\text{True Positive} + \text{True Noegative}}{\text{True Positivie} + \text{True Negative} + \text{False Positive} + \text{False Negative}}$$

2.2.6.2 Presisi

Presisi adalah suatu metode evaluasi yang membandingkan jumlah informasi yang relevan yang berhasil ditemukan oleh suatu sistem dengan total informasi yang diperoleh oleh sistem, baik yang relevan maupun yang tidak relevan. Persamaan untuk menghitung presisi dapat dinyatakan sebagai berikut:

$$\text{Presisi} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

2.2.6.3 Recall

Recall adalah suatu metode pengujian yang membandingkan jumlah informasi yang relevan yang berhasil ditemukan oleh sistem dengan total jumlah informasi relevan yang ada dalam kumpulan informasi, termasuk yang

ditemukan maupun yang tidak ditemukan oleh sistem. Persamaan untuk menghitung recall dapat dijabarkan sebagai berikut:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

2.2.7 Meteran Listrik Analog



Gambar 2. 4 kWh meter analog

Meteran listrik yang dimaksud memiliki kemampuan untuk menampilkan *counter digit* yang merekam jumlah energi listrik yang digunakan pada setiap harinya. Jenis meteran ini umumnya digunakan oleh pelanggan yang menerapkan metode pembayaran pascabayar. Pada prosesnya, perusahaan penyedia listrik (PLN) akan mengukur dan mencatat total energi listrik yang dikonsumsi oleh pelanggan sebagai dasar untuk menentukan besaran biaya yang dikenakan. Setelahnya, pelanggan akan dikenai biaya sesuai dengan jumlah energi yang tercatat dalam meteran (Arum Rifda, 2022).

2.2.8 White Box

Metode ini bertujuan untuk menguji sistematisasi aplikasi dengan mempertimbangkan struktur kode program yang telah direncanakan sebelumnya. Hal ini dilakukan untuk memperoleh sejumlah besar skenario pengujian aplikasi yang dapat dijalankan, serta untuk mengidentifikasi kesalahan dan cacat dalam implementasi kode program (Londjo, 2021). Secara

esensial, pengujian ini difokuskan pada evaluasi logika program dalam mendeteksi kesalahan.

2.2.9 Black Box

Metode ini digunakan untuk menguji aplikasi secara keseluruhan tanpa perlu mengakses rincian kode sumber, dengan tujuan memastikan bahwa setiap fungsi bekerja dengan baik sesuai dengan perencanaan yang telah ditetapkan sebelumnya (Ningrum, Suherman, Aryanti, Prasetya, & Saifudin, 2019). Berbeda dengan pendekatan *white box*, metode ini digunakan untuk mengevaluasi fungsionalitas penuh dari sistem yang menerapkan deteksi angka pada kWh meter, dengan memanfaatkan algoritma YOLOv9 untuk mendeteksi citra kWh meter dan EasyOCR untuk mengekstraksi angka yang terdapat pada kWh meter tersebut.