

Bab III

Metode Penelitian

3.1 Jenis Penelitian

Penelitian "Analisis Pengaruh Cahaya Terhadap Akurasi Deteksi Tepi Pada Prototipe Parkir Pintar Berbasis Web" dikategorikan sebagai penelitian R&D (*Research and Development*) karena hasil penelitian yang dilakukan membutuhkan perhitungan matematis untuk memberikan hasil evaluasi yang akurat serta program yang dibuat dapat diaplikasikan secara nyata.

Untuk mencapai keberhasilan dari penelitian ini, tentunya terdapat langkah-langkah atau metode yang harus dilakukan untuk membuat program yang dapat menghasilkan suatu sistem parkir cerdas yang dapat digunakan dengan mudah dan efisien oleh pengguna. Langkah atau metode tersebut yaitu, melakukan kajian pustaka berdasarkan referensi yang relevan dan terbaru sehingga informasi yang didapat valid. Lalu, melakukan pengumpulan data untuk dijadikan contoh pengujian. Setelah itu, lakukan perancangan program dan web dan diakhiri dengan evaluasi jika diperlukan.

3.2 Variabel dan Definisi Operasional

a. Variabel

Variabel adalah faktor yang diuji atau diukur dalam penelitian. Contoh variabel dalam penelitian ini adalah akurasi

b. Definisi Operasional

Definisi operasional adalah definisi yang menjelaskan bagaimana variabel yang diukur dalam penelitian. Definisi operasional dalam penelitian ini meliputi cara mengukur akurasi pada sistem pendeteksi parkir mobil dengan menggunakan tingkat keberhasilan dalam mendeteksi keberadaan kendaraan di area parkir.

3.3 Variabel Bebas (Independen)

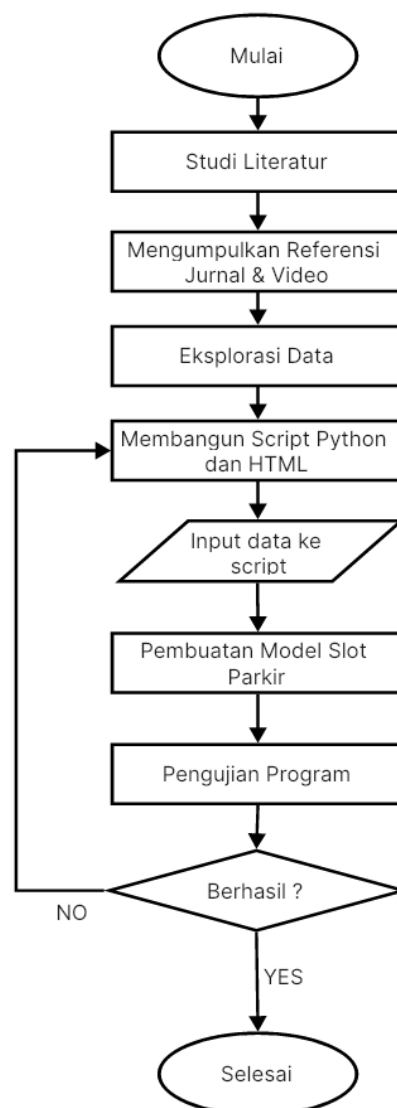
Implementasi teknologi *OpenCV* dan deteksi tepi dalam sistem pendeteksi parkir mobil serta cahaya yang memengaruhi akurasi dari deteksi tepi.

3.4 Variabel Terikat (Dependen)

- a. Sistem deteksi lahan parkir
- b. Parameter yang terdapat dalam sistem
- c. Akurasi yang di uji saat sistem sedang berjalan

3.5 Alur Penelitian

Untuk mendukung jalannya penelitian maka penulis merancang alur penelitian agar penelitian dilakukan dengan teratur dan sistematis agar penelitian mencapai hasil yang baik. Alur penelitian di rancang kedalam bentuk flowchart pada gambar 3.1.



Gambar 3. 1. Flowchart Alur Penelitian

3.5.1 Studi Literatur

Studi literatur merupakan proses awal dari sebuah penelitian yang mengharuskan penulis untuk mencari sumber data yang kredibel dan dapat dipertanggungjawabkan keasliannya. Penulis mencari sumber data melalui jurnal

Adapun sumber data yang dicari berupa jurnal dan video digunakan untuk melakukan simulasi berasal contoh video dari youtube yang didownload dan isinya sebuah rekaman parkir mobil yang nantinya akan diolah menggunakan aplikasi pemrograman supaya dapat dijadikan sebuah contoh apabila Parkir Pintar ini menggunakan sebuah video rekaman.

3.5.2 Eksplorasi Data

Explorasi data yang dilakukan adalah penggunaan video yang telah dicari sebelumnya untuk dijadikan sebuah data dan diolah melalui software pemrograman sehingga video tersebut dapat dimanipulasi oleh Kecerdasan Buatan. Selain itu mengambil sampel gambar dari sebuah video tersebut untuk dijadikan acuan dalam video yang telah di program.

3.5.3 Modelling

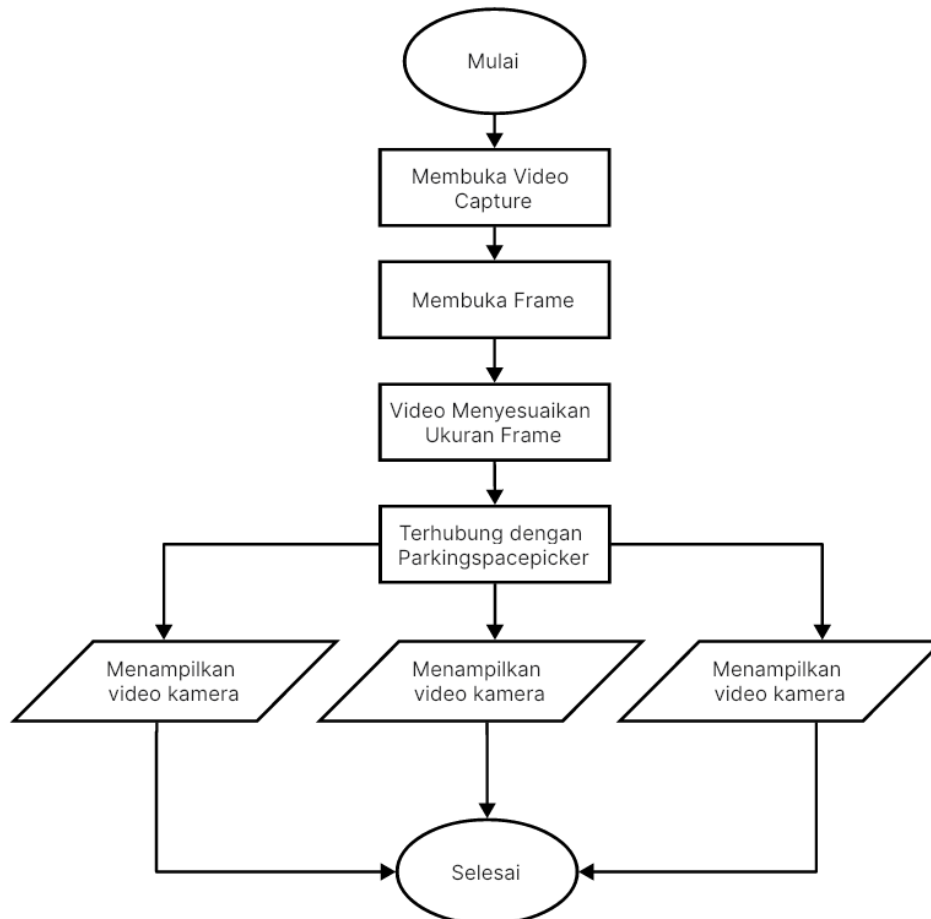
Dalam sistem modeling menggunakan *Image Thresholding* dan *Image blur* yang dimana *image thresholding* itu adalah suatu metode yang digunakan untuk memisahkan antara objek dan backgroundnya. Lalu, *image blur* adalah metode yang digunakan untuk mengurangi noise pada gambar, sehingga data yang didapat bisa lebih akurat.

3.6 Perancangan Program *Python* dan Web

Dalam membuat program Parkir Pintar dengan deteksi tepi ini membutuhkan bahasa pemrograman *Python* karena akan memudahkan dalam penginstalan library yang dibutuhkan seperti *Numpy*, *cv2*, *cvzone*, dan *pickle*. Pada perancangan program sampai benar-benar berfungsi, kita harus membuat dua file yang berbeda dengan dua sintaks yang berbeda namun dapat terhubung. Dua file ini diberi nama *main.py* dan *parkingspacepicker.py*. Berikut penjelasan mengenai dua file tersebut:

3.6.1 *main.py*

main.py merupakan file yang bertindak sebagai pengatur utama dalam program Parkir Pintar. Ini berisi sintaks *Python* yang mengendalikan alur kerja keseluruhan, termasuk tampilan video kamera yang mengintegrasikan deteksi slot parkir secara *realtime*.



Gambar 3.2. Flowchart main.py

Dalam konteks pengembangan program Parkir Pintar, di mana tujuan utamanya adalah untuk menciptakan solusi efisien dalam mengelola dan memantau tempat parkir, *main.py* menjadi inti dari sistem tersebut. Dalam file *main.py*, Anda akan menemukan implementasi kode yang menghubungkan berbagai komponen sistem, termasuk interaksi dengan kamera dan deteksi slot parkir. Di dalam *main.py*, Anda akan menemukan beberapa langkah penting:

a. Inisialisasi

Pada bagian awal *main.py*, Anda mungkin akan menemukan kode yang menginisialisasi berbagai komponen yang diperlukan, seperti mengakses kamera, memuat model deteksi, dan menghubungkan dengan file *parkingspacepicker.py*.

b. Loop Pemrosesan

Setelah inisialisasi, *main.py* mungkin akan memulai loop pemrosesan yang terus berjalan selama program dijalankan. Dalam *loop* ini, program akan terus memperbarui tampilan video kamera, mendeteksi perubahan dalam slot parkir, dan mengambil tindakan yang sesuai.

c. Integrasi dengan *parkingspacepicker.py*

Bagian penting dari *main.py* adalah integrasi dengan *parkingspacepicker.py*. Kode dalam *main.py* akan berinteraksi dengan fungsi-fungsi atau kelas-kelas yang didefinisikan dalam *parkingspacepicker.py*. Ini memungkinkan informasi tentang status slot parkir yang diperoleh dari *parkingspacepicker.py* untuk diintegrasikan ke dalam tampilan video kamera.

d. Deteksi dan Tampilan *Realtime*

Main.py akan memanggil video dari kamera ponsel yang terhubung melalui aplikasi droidcam dan menggabungkan hasil pengolahan gambar untuk deteksi objek pada slot parkir yang telah ditempatkan dalam *parkingspacepicker.py*. Hasil deteksi slot parkir yang dikembalikan oleh *parkingspacepicker.py* akan digunakan untuk menandai status setiap slot (tersedia atau terisi) dalam tampilan video secara *realtime*.

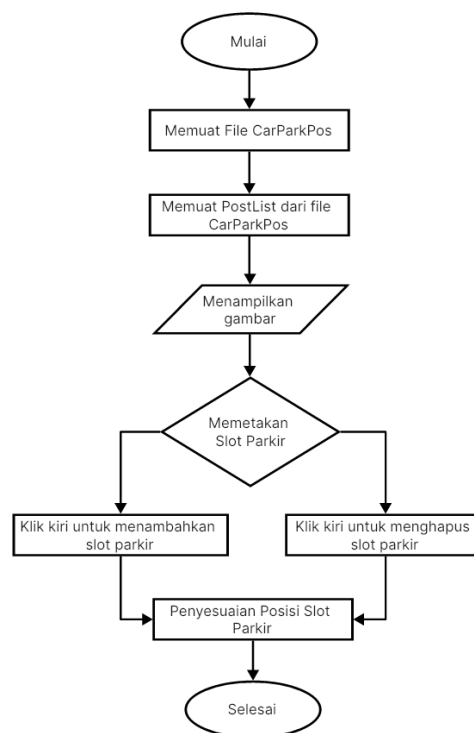
Dalam pengembangan *main.py*, penting untuk memastikan bahwa semua komponen terintegrasi dengan baik dan berfungsi sesuai harapan. Kode perlu diuji secara menyeluruh untuk memastikan deteksi slot parkir bekerja dengan benar, tampilan video aktual, dan respons sistem responsif terhadap perubahan dalam slot parkir.

Ketika Anda menciptakan koneksi yang mulus antara *main.py* dan *parkingspacepicker.py*, Anda akan mencapai tujuan utama dari program Parkir Pintar, yaitu menghasilkan tampilan video kamera yang dapat mendeteksi dan

menampilkan slot parkir secara *realtime*, memberikan informasi yang sangat berharga bagi pengguna sistem.

3.6.2 Parkingspacepicker.py

Parkingspacepicker.py merupakan komponen penting yang berperan sebagai pendukung utama bagi fungsionalitas *main.py* dalam sistem Parkir Pintar. File ini berisi kode-kode yang melibatkan pemrosesan gambar dan analisis visual yang diperlukan untuk melakukan pemetakan dan deteksi slot parkir pada lahan parkir yang ingin digunakan. Gambar 3.3 menampilkan bagaimana alur penggunaan file *parkingspacepicker.py*.



Gambar 3.3. Flowchart *Parkingspacepicker.py*

Dalam *Parkingspacepicker.py*, Anda akan menemukan langkah-langkah kunci berikut:

a. Membaca Gambar Lahan Parkir

Di bagian awal *Parkingspacepicker.py*, kode akan mengambil gambar dalam format *Portable Network graphics* (PNG) atau *Joint Photographic Group* (JPG) yang mewakili lahan parkir yang ingin digunakan sebagai input. Gambar ini akan digunakan sebagai dasar untuk melakukan analisis dan pemetakan slot parkir.

b. Pengolahan Gambar

Setelah gambar lahan parkir dibaca, *parkingspacepicker.py* mungkin akan melibatkan berbagai teknik pengolahan gambar. Ini mungkin termasuk deteksi tepi, segmentasi warna, atau algoritma pemrosesan lainnya yang diperlukan untuk mengidentifikasi garis-garis batas slot parkir.

c. Deteksi dan Pemetakan Slot Parkir

Menggunakan teknik-teknik pengolahan gambar yang sesuai, *Parkingspacepicker.py* akan mencoba mengidentifikasi garis batas yang memisahkan setiap slot parkir. Informasi ini kemudian akan digunakan untuk membuat peta atau representasi data yang merepresentasikan lokasi, jumlah, dan ukuran masing-masing slot parkir.

d. Penyimpanan Informasi Slot Parkir

Setelah analisis selesai, informasi tentang slot parkir yang berhasil dipetakan akan disimpan dalam file *CarParkPos*. File ini akan berisi data mengenai koordinat atau ukuran relatif dari setiap slot parkir, yang nantinya akan diakses oleh *main.py* untuk menampilkan status slot parkir pada tampilan video kamera secara *realtime*.

e. Integrasi dengan *main.py*

Parkingspacepicker.py akan berinteraksi dengan *main.py* melalui antarmuka yang telah didefinisikan. Kode dalam *main.py* akan menggunakan fungsi-fungsi atau kelas-kelas yang disediakan oleh *parkingspacepicker.py* untuk mendapatkan informasi tentang pemetakan slot parkir dan mengintegrasikannya ke dalam tampilan video kamera.

Dengan adanya *parkingspacepicker.py*, *main.py* dapat mengambil informasi yang akurat tentang lokasi dan status slot parkir dari pemetakan yang dilakukan oleh *parkingspacepicker.py*. Ini memungkinkan *main.py* untuk memberikan tampilan video kamera yang mencerminkan kondisi aktual dari lahan parkir, sehingga pengguna dapat melihat dengan jelas mana slot parkir yang tersedia dan mana yang sudah terisi.

3.6.3 Index.html

Index.html adalah halaman utama dari situs web ini. Halaman ini berfungsi sebagai titik masuk utama untuk menjalankan program *main.py* dan menyediakan beberapa fitur yang dapat digunakan oleh pengguna. Berikut beberapa fitur yang terdapat pada halaman index.html:

a. Header

Bagian kiri header halaman ini menampilkan logo yang menarik dan dapat diklik oleh pengguna. Ketika logo ini diklik, pengguna akan langsung kembali ke halaman utama, memberikan navigasi yang lancar dan cepat.

Pada bagian kanan, terdapat tautan menu yang mencakup "Home" dan "About". Tautan ini memungkinkan pengguna untuk dengan mudah beralih antara halaman-halaman penting. Tautan "Home" membawa pengguna kembali ke halaman utama, sementara tautan "About" mengarahkan pengguna ke halaman yang memberikan informasi lebih rinci tentang tujuan dan konten situs web ini.

b. Fungsi Open Camera

Salah satu fitur utama yang ditawarkan oleh halaman ini adalah kemampuan untuk membuka kamera. Dengan terhubung ke program *main.py*, pengguna memiliki akses ke kamera yang terhubung. Fitur ini memiliki berbagai potensi penggunaan, mulai dari konferensi video hingga pemantauan visual secara *realtime*. Pengguna dapat dengan mudah mengaktifkan kamera dengan mengklik tombol yang sesuai, memberikan kemudahan dan efisiensi.

Keseluruhan desain halaman ini bertujuan untuk memberikan pengalaman yang ramah pengguna dan menarik. Logo yang diklik, tautan menu untuk navigasi cepat, serta kemampuan untuk mengakses kamera dengan mudah, semuanya dirancang untuk memastikan pengguna merasa terhubung dengan konten dan fungsionalitas situs web ini secara intuitif.

3.6.4 About.html

Tampilan dari halaman about ini persis dengan halaman index.html dengan Bagian kiri header halaman ini menampilkan logo yang menarik dan dapat diklik oleh pengguna. Ketika logo ini diklik, pengguna akan langsung kembali ke halaman utama, memberikan navigasi yang lancar dan cepat

Pada bagian kanan, terdapat tautan menu yang mencakup "*Home*" dan "*About*". Tautan ini memungkinkan pengguna untuk dengan mudah beralih antara halaman-halaman penting. Tautan "*Home*" membawa pengguna kembali ke halaman utama, sementara tautan "*About*" mengarahkan pengguna ke halaman yang memberikan informasi lebih rinci tentang tujuan dan konten situs web ini.

3.7 Evaluasi Kinerja Sistem

Analisis data yang dijalankan dalam kerangka penelitian ini menyoroti pendekatan dengan pengujian yang dilakukan secara manual. Keputusan ini diambil mengingat kompleksitas kondisi data yang tidak memungkinkan penerapan pendekatan pemodelan otomatis secara efektif. Metode manual ini melibatkan intervensi manusia dalam proses pengolahan data, memungkinkan adaptasi yang lebih baik terhadap variasi yang ada dalam data masukan. Dalam mengukur akurasi hasil analisis, digunakan rumus berikut sebagai acuan utama:

$$Akurasi = \frac{Jumlah\ Berhasil}{Jumlah\ Uji\ Coba} \times 100\% \quad (1)$$

Setelah proses pengujian akurasi selesai dilakukan, langkah selanjutnya adalah melakukan evaluasi untuk mengukur sejumlah parameter kritis, termasuk *presisi*, *recall*, *F1 Score*, dan akurasi secara menyeluruh. Evaluasi ini memberikan pandangan lebih komprehensif tentang kinerja sistem dalam mengenali dan mengklasifikasikan status tempat parkir. Dalam evaluasi ini, nilai-nilai berikut akan dihitung dan dianalisis:

a. True Positive (TP)

Hal ini merujuk pada kasus di mana sistem mengenali tempat parkir sebagai kosong dan memang sebenarnya kosong.

b. True Negative (TN)

Situasi ini terjadi ketika sistem bahwa tempat parkir kosong dan sebenarnya ada mobil yang menempati.

c. False Positive (FP)

Situasi ini terjadi ketika sistem mengenali nya dengan tempat parkir yang sudah terisi, tetapi yang sebenarnya tempat parkir tersebut adalah tempat yang kosong.

d. False Negative (FN)

Hal ini terjadi saat sistem mengenalinya sebagai tempat yang terisi dan keadaan sebenarnya adalah tempat parkir tersebut terisi.

Melalui perhitungan dan analisis nilai-nilai ini, kita dapat mengukur *presisi* (kemampuan sistem dalam mengenali kondisi sebenarnya dari tempat parkir), *recall* (kemampuan sistem dalam mengidentifikasi tempat parkir yang sebenarnya), serta menghitung *F1 Score* yang merupakan harmonisasi antara *presisi* dan *recall*. Akurasi secara keseluruhan juga dievaluasi untuk memberikan gambaran tentang sejauh mana sistem konsisten dalam memberikan respons yang benar terhadap kondisi tempat parkir. Evaluasi ini memberikan pandangan yang lebih mendalam tentang performa sistem dan membantu mengidentifikasi area yang memerlukan perbaikan dalam pengembangan lebih lanjut.

Untuk rumus yang digunakan untuk mencari *Presisi*, *Recall*, *F1 Score* dan akurasi antara lain:

a. *Presisi*

Presisi adalah sebuah perbandingan prediksi dari benar positif kepada seluruh hasil prediksi positif. Pada bagian *presisi* ini akan ditemukan sebuah hasil dari berapa persen pada satu tempat parkir yang kosong dari keseluruhan tempat parkir yang diprediksi kosong. Berikut ini merupakan rumus untuk menghitung *presisi*:

$$Presisi = \frac{TP}{TP + FP} \quad (2)$$

b. *Recall*

Recall adalah sebuah perbandingan prediksi dari *True Positive* (TP) kepada data yang benar positif. Pada bagian *presisi* ini akan ditemukan sebuah hasil dari berapa persen pada satu tempat parkir yang kosong dari keseluruhan slot tempat parkir. Berikut ini merupakan rumus untuk menghitung *recall*:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

c. *F1 Score*

F1 Score dibutuhkan untuk menghitung sebuah perbandingan rata-rata antara *presisi* dengan *recall* yang sebelumnya sudah dihitung. Berikut ini merupakan rumus untuk menghitung *F1 Score*:

$$F1\ Score = 2 \times \frac{Recall \times Presisi}{Recall + Presisi} \quad (4)$$