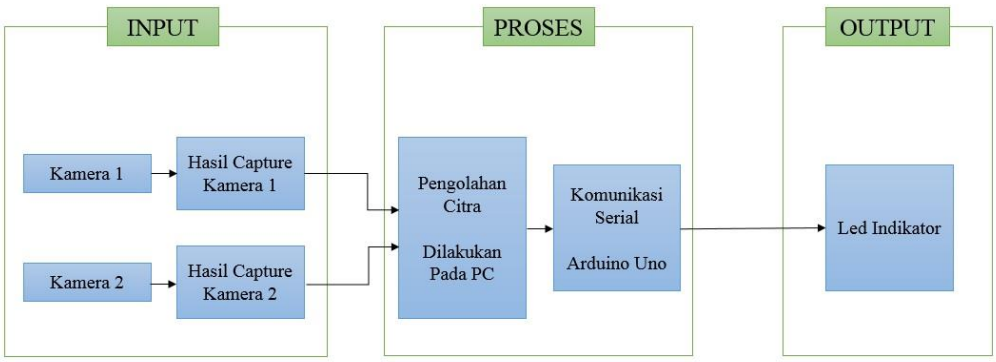
BAB III

PERANCANGAN ALAT DAN PEMBUATAN SOFTWARE

Dalam bab ini akan membahas mengenai perancangan dan realisasi sistem yang akan dibuat menggunakan MATLAB dan *hardware* yang digunakan merupakan miniatur tahap awal dari lahan parkir UNIKOM.

3.1 Perancangan Sistem

Perancangan dan pembuatan alat ini secara umum terdiri dari tiga bagian utama, yaitu masukan (*input),* proses (*process),* dan keluaran (*output).* Pada **Gambar 3.1** dibawah ini terlihat bahwa setiap bagian dari sistem ini memiliki fungsi yang berbeda, namun akan saling berkaitan dalam pengontrolan keluaranya (*output).*



Gambar 3.1 Blok Diagram Sistem

Pada gambar diatas memperlihatkan bagian-bagian perancangan sistem pengolahan citra untuk sistem informasi lahan parkir ini, yang meliputi beberapa bagian penting, yaitu :

- **Bagian Masukan (*input*)**

Masukan (*input*) terdiri dari dua buah masukan, yaitu kamera 1 dan kamera 2 yang menghasilkan *capture* 1 dan *capture* 2. Kedua masukan ini juga disebut sebagai masukan awal sebelum dilakukan proses pengolahan citra yang akan dilakukan dengan *software* MATLAB untuk mendeteksi jumlah kendaraan dari masing-masing lahan yang tersedia.

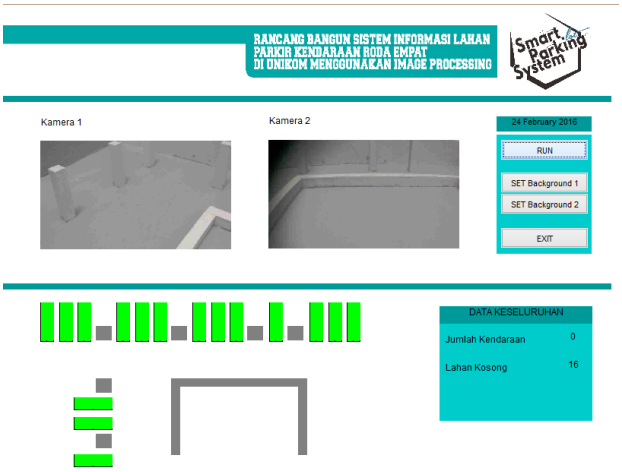
- **Bagian Proses**

  Pada bagian ini hasil *capture* dari kedua buah kamera diolah menggunakan MATLAB untuk mendeteksi jumlah kendaraan. Mendeteksi jumlah kendaraan dilakukan dengancara membandingkan background dengan hasil *capture* dan dilanjutkan ke proses segmentasi.

- **Bagian Keluaran (*Output*)**

  Bagian keluaran (*output*) ini merupakan hasil dari beberapa masukan (input) yang telah diproses pada bagian proses yang sudah dijelaskan sebelumnya. Keluaran yang dihasilkan berupa indikator led yang merupakan kondisi dari lahan parkir tersebut, untuk menandakan lokasi lahan parkir yang tersedia dan jumlah dari lahan parkir yang tersedia.

Dalam pengerjaan Tugas Akhir ini dibutuhkan suatu simulator dan sebuah miniatur lahan parkir yang telah disesuaikan dengan Basement 1 UNIKOM. Simulator yang digunakanadalah MATLAB R2012a seperti yang terlihat pada **Gambar 3.2** dibawah ini.



Gambar 3.2 Tampilan Simulasi di MATLAB

Miniatur atau *prototype* ini telah disesuaikan dengan Basement 1 Unikom dengan skala 1:50. Dalam *prototype* ini terdapat webcam USB sebagai alat untuk mengambil gambar yang berupa masukan (*input*).



Gambar 3.3 Tampilan *Prototype*

Hasil *capture* akan diproses menggunakan MATLAB untuk mendeteksi benda dan mengetahui jumlah kendaraan yang sedang berada pada lahan parkir yang tersedia, yang nantinya keluaranya berupa *led* sebagai indikator letak lahan parkir yang tersedia. *Led* indikator ini yang dihubungkan dengan menggunakan komunikasi serial ke Arduino Uno R3.

3.2 Perancangan *Software*

Hasil *capture* dari kamera 1 dan kamera 2 merupakan masukan awal untuk melanjutkan ke proses selanjutnya. Hasil dari *capture* ini adalah citra asli yang nantinya akan diproses untuk mendeteksi dan mengetahui berapa jumlah objek dari hasil *capture* tersebut. Untuk mengetahui jumlah kendaraan pada lahan parkir yang sudah disediakan dilakukan beberapa pemrosesan yaitu, konversi citra asli ke *grayscale*, konversi citra *grayscale* ke biner, pemisahan objek dengan latar menggunakan *background subtraction*, proses segmentasi, dan penentuan jumlah kendaraan.

3.2.1 Konversi Citra Asli ke *Grayscale*

Didalam tahap ini hasil *capture* yang merupakan citra asli akan dikonversikan ke dalam bentuk warna yang lain yaitu mode warna *grayscale*, yang bertujuan untuk menyederhanakan informasi warna yang terdapat pada ketiga layer matriks warna yaitu *Red*, *Green*, dan *Blue* (RGB) menjadi satu layer yang bernilai dari perkalian ketiga layer matriks warna tersebut dengan konstanta atau dengan cara melakukan rata-rata dari ketiga matriks.



Gambar 3.4a Citra Asli



Gambar 3.4b Citra *Grayscale*

Hasil *capture* yang berupa citra asli (**gambar 3.4a**) dirubah menjadi format *grayscale* (**gambar 3.4b**) dan mengalami perubahan nilai. Citra *grayscale* disimpan dalam format 8bit untuk setiap sample piksel, yang memungkinkan sebanyak 256 intensitas. Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing *red* (R), *green* (G) dan *blue* (B) menjadi citra grayscale dengan nilai X, maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai R, G dan B.



Gambar 3.5 Gambar *Visualisasi* 256 Aras Keabuan

3.2.2 Konversi Citra *Grayscale* ke Biner

Citra biner (*binary image*) adalah citra digital yang hanya memiliki 2 kemungkinan warna, yaitu hitam dan putih. Citra biner disebut juga dengan citra B&W (*Black&White*) atau citra monokrom. Hanya dibutuhkan 1 bit untuk mewakili nilai setiap piksel dari citra biner yaitu nilai 1 dan 0.



Gambar 3.6a Citra *Grayscale*



Gambar 3.6b Citra Biner

Perubahan format *grayscale* (**Gambar 3.6a**) menjadi format biner (**Gambar 3.6b**) mengalami perubahan nilai. Pembentukan citra biner memerlukan nilai batas keabuan yang akan digunakan sebagai nilai patokan. *Piksel* dengan

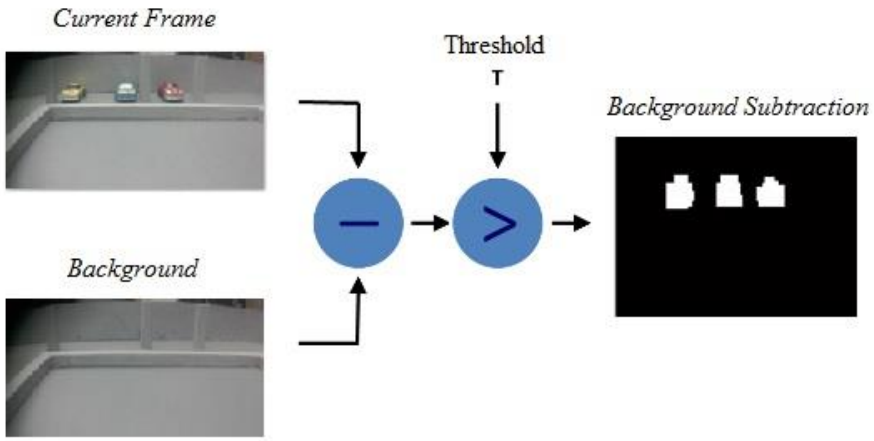
derajat keabuan lebih besar dari nilai batas akan diberi nilai 1 dan sebaliknya piksel dengan derajat keabuan lebih kecil dari nilai batas akan diberi nilai 0.



Gambar 3.7 Citra biner yang terdiri dari nilai 1 dan 0

3.2.3 Pemisahan Objek dengan Latar Menggunakan Metode *Background Subtraction*

Metode yang sering digunakan untuk segmentasi *moving region* secara real time adalah *background subtraction* yang merupakan *threshold* atau batas kesalahan antara gambar tanpa adanya objek yang bergerak (*background*) dengan gambar itu sendiri. *Background image* direpresentasikan sebagai *scene* atau adegan tanpa adanya objek yang bergerak. *Background image* harus selalu diperbaharui sehingga dapat beradapatasi dengan perubahan kondisi seperti perubahan pencahayaan. Perubahaan pencahayaan akan mempengaruhi proses pendeteksian objek. Pada *background subtraction*, gambar saat ini dibandingkan dengan gambar referensi untuk mendeteksi adanya perubahan pixel. Gambar referensi sebaiknya disesuaikan dengan kondisi pencahayaan dari suatu *scene* atau kejadian.



Gambar 3.8 Proses dari *background subtraction*

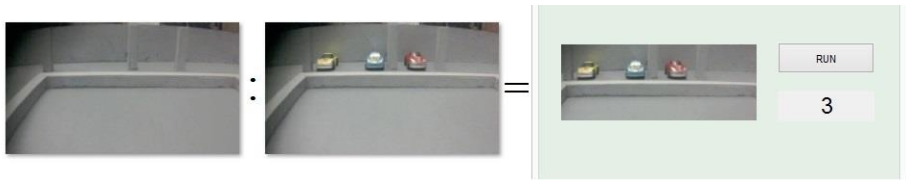
Dengan melakukan hal ini maka akan mempermudah untuk mendeteksi jumlah kendaraan. Hal ini dikarenakan nilai dari setiap piksel pada gambar yang sudah melalui proses ini sangatlah tegas, sehingga sulit untuk mendeteksi nilai bernilai 1 dan bernilai 0.

3.2.4 Segmentasi

Segmentasi merupakan sebuah proses untuk memisahkan sebuah objek dari *background* atau latar, sehingga objek tersebut dapat diproses untuk mendeteksi jumlah objek yang ada bersama *background* atau latar. Proses segmentasi dalam tugas akhir ini adalah hanya sebagai *window* atau *segment* untuk melihat kendaraan, dimana 1 *window* atau *segment* hanya untuk melihat 1 kendaraan. Berarti dalam 1 gambar pada kamera 1 terdapat 9 *window*, dan pada kamera 2 terdapat 7 window sehingga maksimum jumlah kendaraan yang terhitung ada sebanyak 16 kendaraan.

3.2.5 Pendeteksian Jumlah Kendaraan

Penentuan jumlah kendaraan ditentukan berdasarkan perbandingan *background* dengan hasil *capture* yang sudah ada objeknya dengan proses *image processing*.

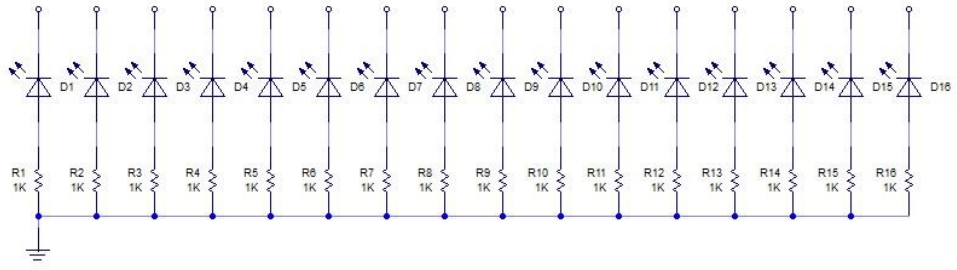


Gambar 3.9 *Background*, Citra Asli, dan Hasil Deteksi

Jumlah objek yang terapat pada hasil *capture* sebanyak 3 buah, setelah melakukan proses segmentasi dan *background subtraction* maka *background* dibandingkan dengan hasil *capture* maka jumlah yang terdeteksi sebanyak 3 buah.

3.3 Rangkaian Lampu Indikator Lahan Parkir

Led berwarna hijau digunakan sebagai indikator lahan parkir yang berjumlah sebanyak 16 buah yang mewakili 16 lahan parkir. Apabila *led* berwana hijau pada salah satu lahan parkir mati maka lahan tersebut sudah terisi oleh kendaraan, dan apabila sebaliknya maka artinya lahan parkir tersebut kosong.

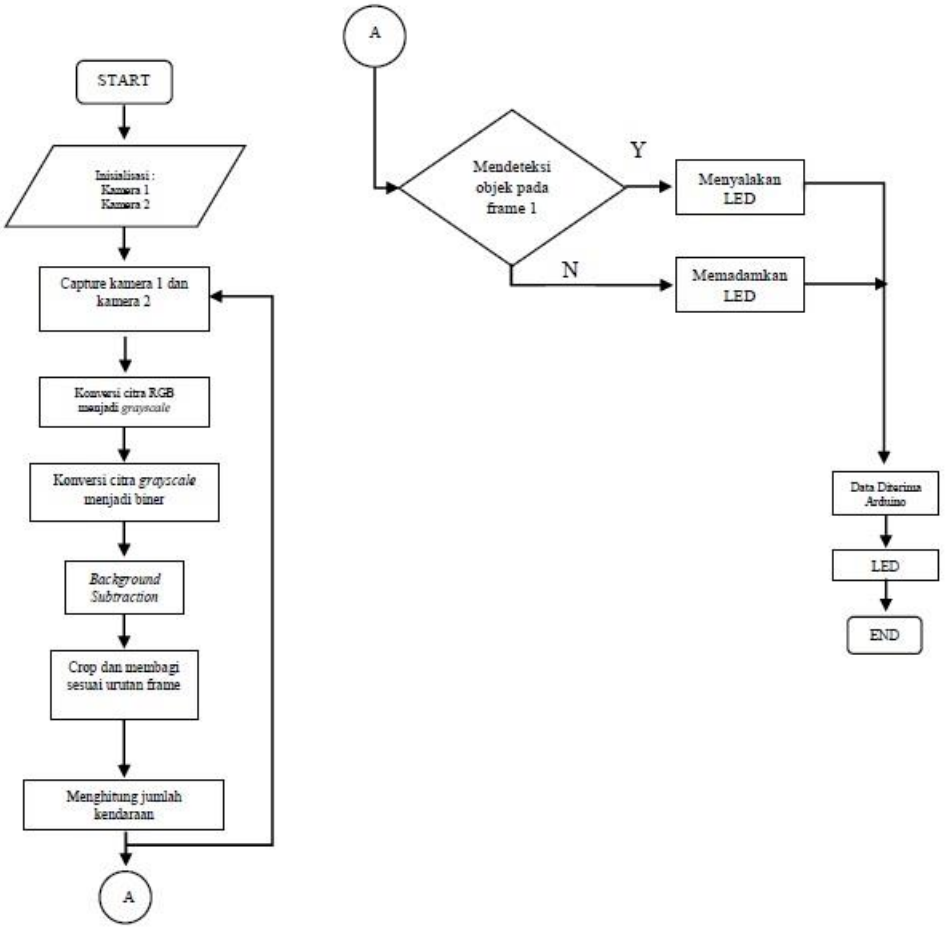


Gambar 3.10 Skematik Lampu Indikator

Masukan untuk *led* ini berupa data yang diterima oleh Arduino Uno dari komunikasi serial. Kondisi pada saat penyalaan *led* akan sesuai dengan *axes* yang ada pada pemrograman MATLAB.

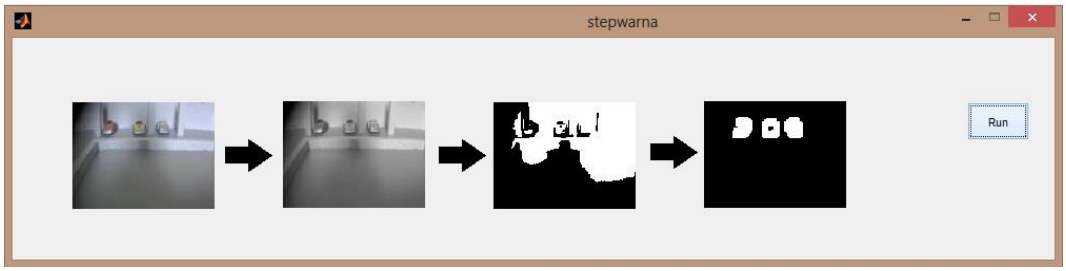
3.4 Perancangan Perangkat Lunak

Sistem informasi lahan parker ini menggunakan aplikasi MATLAB 2012a yang terhubung dengan komunikasi serial ke Arduino untuk menyalakan *output led* sebagai informasi lahan parker yang tersedia dan juga sebagai informasi lokasi lahan parker tersebut. *Flowchart* dari sistem ini dapat dilihat pada **Gambar 3.11** dibawah ini.



Gambar 3.11 *Flowchart* Rancang bangun Sistem Informasi Lahan Parkir

Pada saat program sudah berjalan yang pertama dilakukan adalah inisialisasi kedua buah kamera, lalu masing-masing kamera tersebut mengambil gambar (*capture*). hasil dari *capture* ini dirubah kedalam format *grayscale,* hal ini dilakukan agar mempermudah nantinya ketika dirubah kedalam format biner. Dengan format citra yang sudah dirubah kedalam format biner, dilakukan *Background Subtraction* untuk memisahkan benda pada latar. *Background Subtraction* membandingkan gambar saat ini dengan gambar referensi atau latarnya fungsinya untuk mendeteksi adanya berubahan pixel. Setelah itu dilakukan proses crop yang bertujuan untuk membagi citra menjadi *frame,* hal ini bertujuan untuk mempermudah penghitungan objek dan penentuan lokasi dimana lahan parkir yang tersedia dan tidak. Berikut ini adalah langkah-langkah dari proses pengolahan citra



Gambar 3.12 Langkah-langkah dari proses pengolahan citra

Setelah mendapatkan data jumlah kendaraan dan lokasi kendaraan tersebut maka akan dikirim melalui komunikasi serial yang telah terhubung ke Arduino untuk menyalakan *led* sebagai informasi akhirnya.