BAB 2
TINJAUAN PUSTAKA

2.1 Peraturan Pada Persimpangan

Pada penelitian ini, pelanggaran yang akan dideteksi adalah pelanggaran pada zebra cross, saat pengendara mobil ataupun motor berhenti pada area zebra cross, maka pengendara tersebut akan difoto dan diberikan himbauan. Begitupun juga dengan pelanggaran pada jalur kiri, ketika pengendara berhenti di jalur kiri, maka pengendara tersebut akan difoto dan diberikan himbauan. Peraturan ini mengacu kepada UU No 22 Tahun 2009, tentang Lalu Lintas dan Angkutan Jalan. Zebra cross merupakan tempat penyebrangan dan hak bagi pejalan kaki, tertera pada pasal 45 ayat 1 (c). Sedangkan untuk jalur kiri tertera pada pasal 112 ayat 3 tentang Belokan atau Simpangan yang berisi bahwa Pengemudi dilarang langsung belok kiri , kecuali ditentukkan lain oleh rambu lalu lintas atau alat pemberi isyarat lalu lintas. [2]



Gambar 1. 5 Isyarat Untuk Langsung Belok Kiri (Jalan Riau Perempatan Merdeka)

2.2 Computer Vision

Computer vision adalah otomatisasi dalam pengambilan informasi dari sebuah citra. Model 3D, Posisi Kamera, Deteksi dan Rekognisi Objek adalah jenis jenis yang informasi nya bisa diolah dengan Computer Vision. Computer Vision

tidak hanya berisi tentang rekognisi dan deteksi objek pada citra, namun di dalamnya berisi ada juga programming, matematika dan modelling dicampur menjadi satu. [3]

Boyle dan Thomas (1988) menyatakan bahwa Computer Vision menghadirkan *low level processing* yang dijadikan sebagai algoritma pengolahan citra yang dapat disebut purely, mereka juga menyatakan bahwa Computer Vision lebih dari hanya sekedar *image recognition*. [4]

2.3 Background Substraction

Background substraction adalah algoritma yang biasanya digunakan pada pembangunan sebuah aplikasi untuk mendeteksi, *tracking* dan menghitung jumlah objek yang ada pada citra. Contohnya menghitung jumlah orang yang masuk dan keluar sebuah mall, algoritma ini pada akhirnya bisa membedakan mana objek (manusia, kendaraan) yang bergerak dan mana latar belakang yang statis. [5]

Semisal dua citra dinyatakan sebagai matriks f(x,y) dan h(x,y). Operasi background substraction akan dinyatakan dalam matriks g(x,y) [6]. Yang dihasilkan dari perhitungan berikut,

$$g(x,y) = f(x,y) - h(x,y)$$

Akan terjadi normalisasi ke skala *grey* level semula jika hasil pengurangan tersebut bernilai negatif. [6]

1. Tambahkan grey level dikurangi 1 ke setiap nilai piksel hasil lalu dibagi dua. [7]

Berikut adalah contoh hasil background subtraction.



Gambar 2. 1 Citra asli(1), citra background(2), hasil subtraksi(3)

Untuk mendapatkan hasil background subtraction yang terbaik, karena deteksi yang dilakukan tidak tergantung dengan warna dan supaya tidak terpengaruh

dengan cahaya, maka langkah yang dilakukkan adalah dengan mengubah citra menjadi grayscale, lalu untuk mendapatkan citra yang mempunyai sedikit noise, maka ditambahkan efek blur pada citra dengan hasil gambar yang lebih halus.

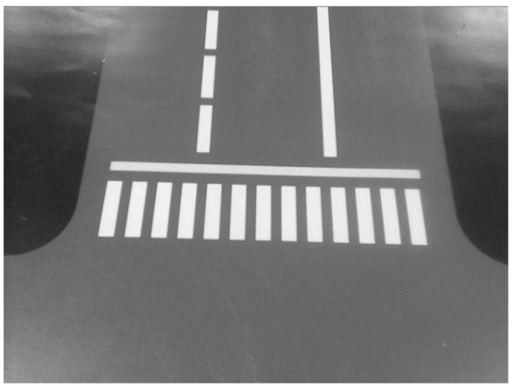
Terakhir, subtraksi gambar background yang statis dengan frame dari footage yang ditangkap oleh kamera. Hasil nya tidak akan sempurna, namun dengan langkah – langkah sederhana tersebut dapat dihasilkan citra yang cukup untuk mendeteksi objek.

Berikut adalah rumus yang dipakai untuk mengubah citra menjadi grayscale.

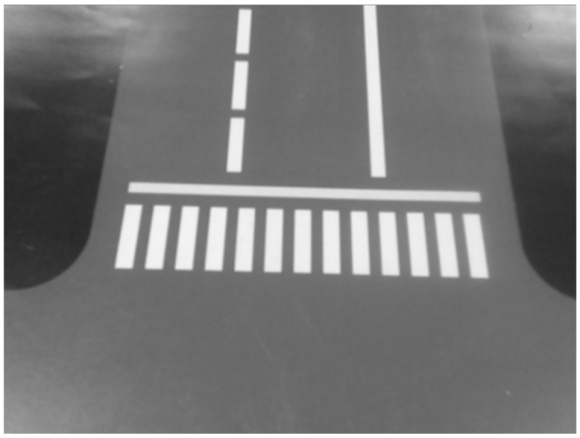
$$\text{RGB[A] to Gray:} Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$



Gambar 2. 2 Sebelum ditambahkan efek grayscale



Gambar 2. 3 Sesudah ditambahkan efek grayscale



Gambar 2. 4 Hasil Smoothing

2.4 Thresholding

Threshold adalah batas, di mana kita menetapkan nilai, apakah di atas atau di bawah. Metode ini digunakan biasanya untuk pemisahan objek dengan selain objek, biasanya menghasilkan dua buah warna yaitu hitam dan putih. Batasan biasanya berupa tipe data float antara 0 sampai 1. 0 adalah hitam dan 1 adalah putih. Jika data citra melebihi batas yang ada, maka akan dimasukan menjadi golongan 1 (berwarna putih) jika dibawah atau kurang dari batas yang ada maka akan dibuat masuk ke golongan 0 (berwarna hitam). [5] Menurut Gonzales dan Woods, 2002) Mengatur nilai intensitas semua piksel yang lebih besar dari nilai threshold T sebagai latar depan dan yang lebih kecil dari nilai threshold T sebagai latar belakang untuk partisi citra, adalah langkah yang digunakkan untuk melakukkan thresholding. Sebelum melakukkan thresholding, perlu dilakukkan grayscaling terlebih dahulu. Berikut adalah langkah – langkah thresholding yang bisa bekerja pada citra yang memiliki noise. [7]

1. Menentukan *initial* dari *Threshold* random atau dengan metode tertentu.
2. Citra dibedakan menjadi dua jenis, citra objek (G1) dan citra background (G2).

Berikut adalah persamaan yang didapat

$$G1 = \{f(x,y) : f(x,y) > T \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

$$G2 = \{f(x,y) : f(x,y) <= T \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Keterangan :

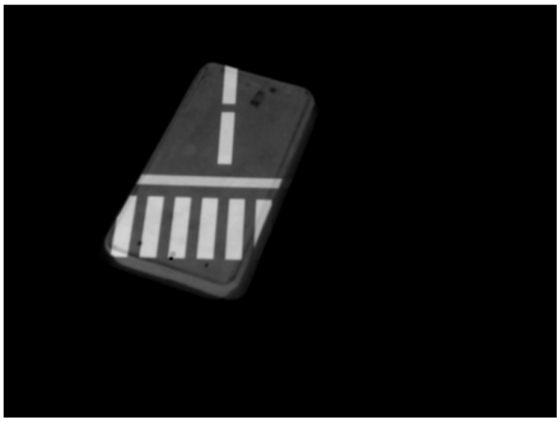
- f(m,n) adalah nilai piksel antara baris(y) dan kolom(x)

3. Setelah mendapatkan nilai – nlai antara baris dan kolom g1 dan g2 hitung nilai rata – rata masing masing citra nya.
4. Dari nilai rata – rata tersebut didapatkan nilai threshold baru dengan persamaan.

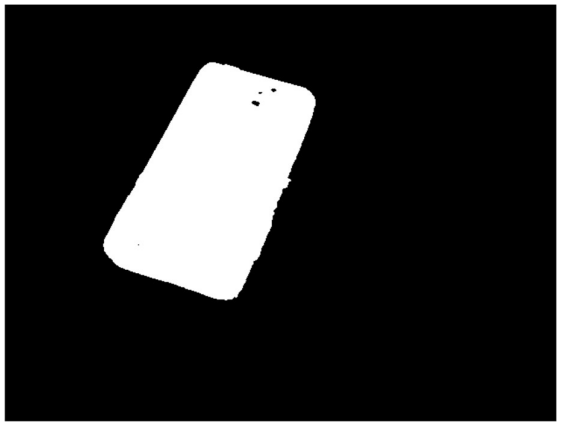
$$T = ½ (z1 + z2)……..(3)$$

5. Perhitungan dikatakan selesai ketika nilai threshold yang baru sama dengan nilai sebelumnya. Oleh karena itu langkah 2 sampai 4 harus diulang dengan nilai T berbeda.

Berikut adalah gambar asli yang diambil dari video dan hasil thresholdingnya.



Gambar 2. 5 Hasil Subtraksi yang akan di threshold



Gambar 2. 6 Hasil dari Thresholding gambar 2.5

2.5 Contour Detection

Contour adalah garis batas terluar dari objek yang mempunyai intensitas warna yang sama dan terdeteksi oleh computer. Dalam penelitian yang melibatkan

computer vision, contour adalah hal yang paling sering digunakan sebagai alat untuk menganalisa sebuah bentuk, [8] juga berguna untuk memperbaiki detail gambar yang kabur [9] Dapat kita lihat hasil dari thresholding pada gambar 6, menunjukan bahwa tepi dari piksel objek yang terdeteksi pada frame masih belum optimal. Hal ini akan mempengaruhi proses deteksi blob. Untuk mengoptimalkan piksel yang terdeteksi pada hasil thresholding maka digunakan deteksi contour, agar piksel menjadi mendekati bentuk objek yang ada dalam frame. Berikut adalah langkah – langkah dan contour yang ditemukkan dari hasil *thresholding* dengan *contour detection*.



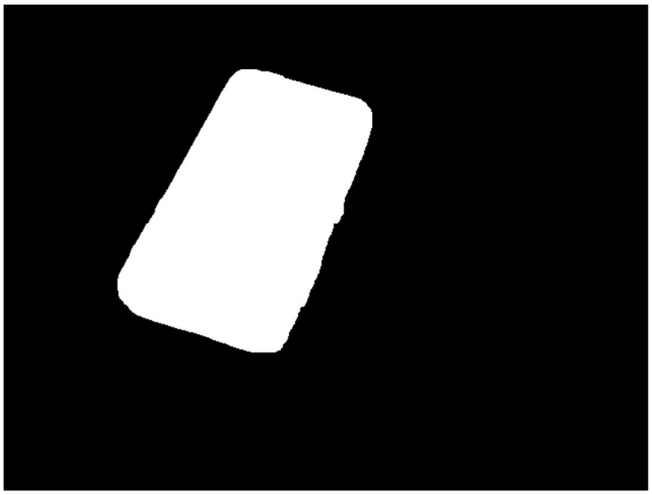
Gambar 2. 7 Langkah – langkah Deteksi Contour

Contour dibagi menjadi dua jenis, yaitu kontur eksternal dan kontur internal, piksel bagian kontur eksternal terletak di bagian luar objek sedangkan kontur internal ada di dalam objek. [10]

0	0	0	0	0	0
0	1	1	1	1	0
0	1	1	1	1	0
0	1	1	1	1	0
0	0	0	0	0	0

0	0	0	0	0	0
0	1	1	1	1	0
0	1	1	1	1	0
0	1	1	1	1	0
0	0	0	0	0	0

Gambar 2. 8 Kontur Eksternal (Kiri) Kontur Internal (Kanan)



Gambar 2. 9 Hasil deteksi contour dari thresholding

2.6 Convex Hull

Convex hull adalah tepi terluar dari objek yang telah terdeteksi oleh computer dan dihubungkan dengan garis. [11] Pada gambar 7 dapat terlihat hasil dari deteksi contur masih kurang optimal. *Convex Hull* mengambil titik piksel terluar yang terdeteksi dan menghubungkannya dengan garis.

Dengan kata lain, *Convex Hull* adalah sebuah polygon terkecil yang mengelilingi sebuah objek, polygon dianggap seperti objek elastis yang bisa melingkupi sebuah objek yang akan dideteksi. [10]

1	x	x
1	0	x
1	x	x

1	1	1
x	0	x
x	x	x

x	x	1
x	0	1
x	x	1

x	x	x
x	0	x
1	1	1

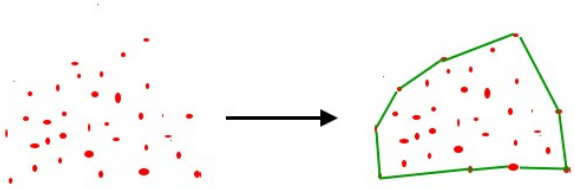
Gambar 2. 10 Empat elemen pembentuk convex Hull (kiri - kanan) B1, B2, B3, B4

Untuk melakukkan perhitungan *convex hull,* diberikan Xo1 = A, dimana A adalah citra nya. Berikut adalah perhitungan yang akan diproses.

$$X_k^i = thm\left(X_{k-1}, B^i\right) \cup A \; i = 1, 2, 3, 4 \; dan \; k \; 1, 2, 3$$

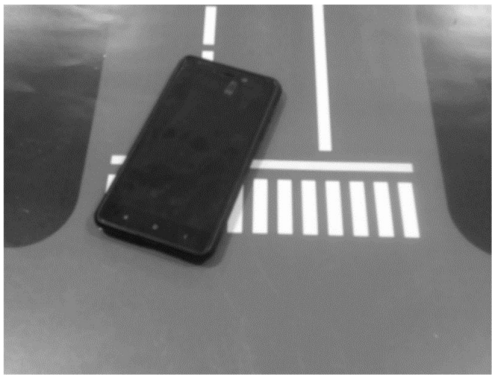
Konvergensi berhasil ketika

$$X_k^i = X_{k-1}^i$$



Gambar 2. 11 Ilustrasi *Convex Hull*

Ruang yang dikelilingi garis yang terhubung dapat diberikan warna, contoh putih. Berikut adalah langkah – langkah dan gambar contoh hasil dari *Convex Hull*.



Gambar 2. 12 Gambar asli



Gambar 2. 13 Langkah – langkah Convex Hull



Gambar 2. 14 Hasil *Convex Hull*

Piksel putih yang ada menjadi sebuah blob yang cukup untuk merepresentasikan objek yang ada di frame.

2.7 Blob Detection

Blob detection adalah langkah – langkah yang dilakukkan pada *background substraction* yaitu, *Thresholding*, *Contour Detection* dan berakhir pada *Convex Hull*. Dalam langkah convex hull dijelaskan bahwa hasilnya adalah sebuah blob yang artinya kumpulan pixel yang saling berhubungan dalam sebuah gambar yang mempunyai kesamaan properti [12]. Untuk menunjukan bahwa blob berhasil terdeteksi adalah dengan membuat kotak merah disetiap blobnya. Dapat dilihat pada gambar 3.11 .



Gambar 2. 15 Langkah – langkah blob detection