

I.1 Teori Pendukung

Pada subbab ini akan dibahas mengenai teori yang terkandung dalam tugas akhir ini.

I.1.1 *Haar-Like Feature*

Haar like Feature merupakan metode yang lazim digunakan dalam pendeteksian obyek. Nama Haar sendiri mengacu pada Haar Wavelet, sebuah fungsi matematika yang berbentuk kotak dan memiliki prinsip seperti pada fungsi Fourier. (Purwanto, Dirgantoro, & Jati, 2015). Haar-like features merupakan rectangular features (fungsi persegi), yang memberikan indikasi secara spesifik pada sebuah gambar atau image. Prinsip Haar-like features adalah mengenali obyek berdasarkan nilai sederhana dari fitur tetapi bukan merupakan nilai piksel dari image obyek tersebut. Metode ini memiliki kelebihan yaitu komputasinya sangat cepat, karena hanya bergantung pada jumlah piksel dalam persegi bukan setiap nilai piksel dari sebuah image (Viola, Paul, & Jones, 2011). Deteksi obyek kendaraan pada penelitian ini merupakan pemodifikasian sistem Haar-like features dari deteksi wajah yang pertama kali dilakukan oleh Viola dan Jones kemudian dikembangkan oleh Lienhart (Viola, Paul, & Jones, 2001) (Lienhart, Rainer, & Maydt, 2002). Metode yang diusulkan Viola dan Jones menggabungkan empat kunci utama untuk mendeteksi sebuah obyek (Viola, Paul, & Jones, 2001):

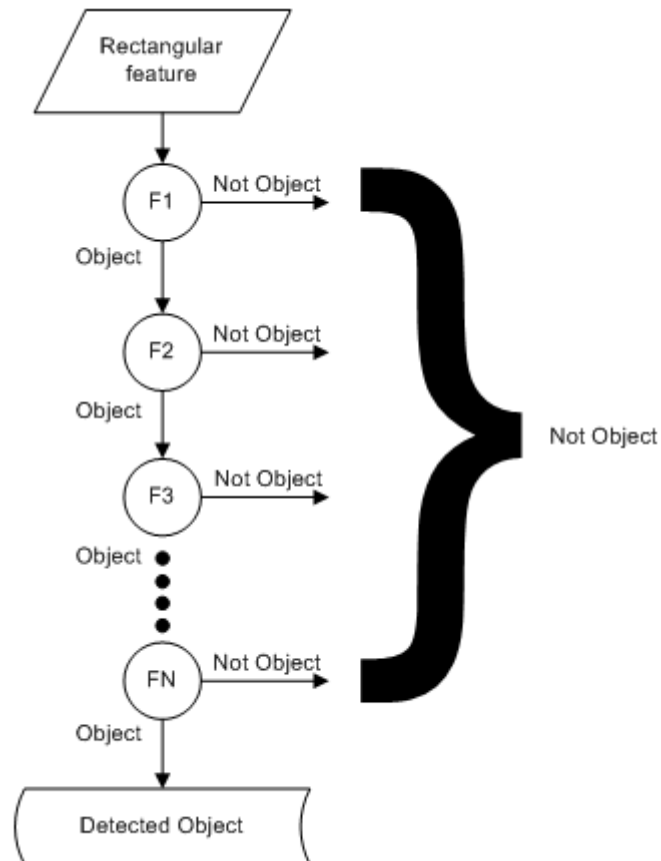
1. Fitur persegi sederhana, disebut fitur Haar
2. Integral image untuk pendeteksian fitur dengan cepat
3. Metoda AdaBoost machine-learning
4. Cascade classifier untuk mengkombinasikan banyak fitur

Haar like feature memproses citra dalam sebuah kotak persegi dengan ukuran tertentu misalnya 24 x 24 pixel seperti ditunjukkan pada gambar 1 sebagai berikut:



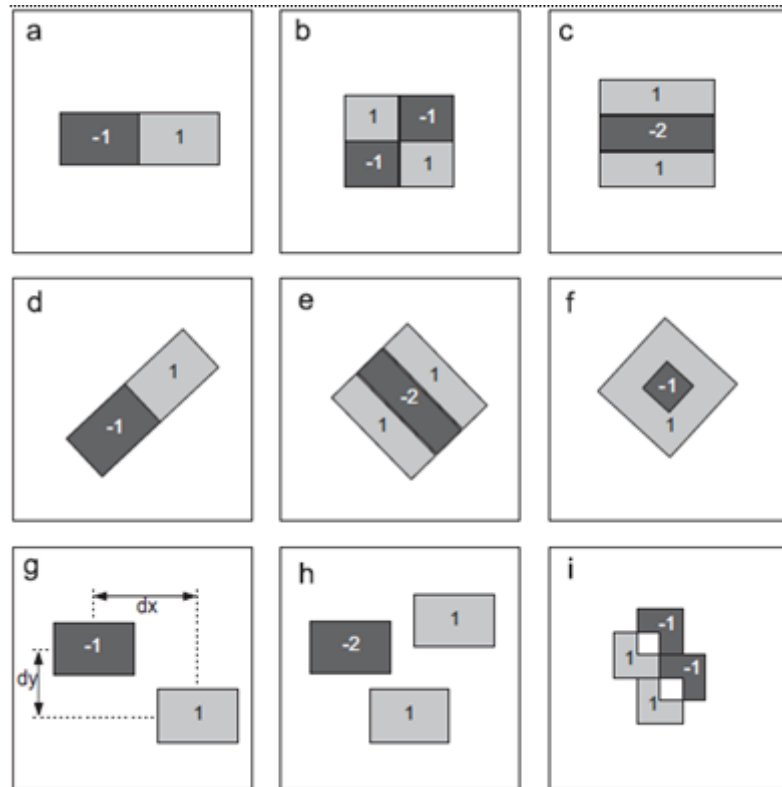
Gambar II.1. Rectangular Feature Haar Cascade

Di dalam kotak inilah proses filtering obyek dilakukan untuk diketahui apakah ada atau tidak obyek yang akan dideteksi. Proses filterisasi ini dilakukan secara bertingkat yang menyebabkan metode ini nantinya disebut sebagai Haar Cascade Classifier seperti ditunjukkan pada skema filter di gambar 2 sebagai berikut:



Gambar II.2. Skema Pendeteksi Obyek

Hasil deteksi *Haar-like Feature* kurang akurat jika hanya menggunakan satu fungsi saja. Semakin tinggi tingkatan filter pendeteksian maka semakin tepat pula sebuah obyek dideteksi akan tetapi akan semakin lama proses pendeteksian. Pemrosesan *Haar-like feature* yang banyak tersebut diatur dalam *classifier cascade*. *Haar-Wavelet* (Wave = Gelombang) merupakan gelombang persegi (interval gelap dan interval terang) yang kemudian dibandingkan nilai rata-rata *pixel* keduanya. Apabila perbandingan nilai rata-rata intensitas tersebut berada di atas *threshold* (ambang batas), maka dikatakan memenuhi syarat fitur *Haar*. Untuk gambar bergerak seperti video, proses ini dilakukan secara diskrit dengan mencuplik video pada frame rate tertentu. Macam-macam variasi *Haar-like feature* ditunjukkan pada gambar 3 sebagai berikut (Pavani, Delgado, & Frangia, 2010):



Gambar II.3 Berbagai variasi persegi Haar dengan Bobot Standar Persegi Panjang

Keterangan gambar 3 (Pavani, Delgadoa, & Frangia, 2010):

a, b : Fitur Haar yang diusulkan Papageogiou dkk.

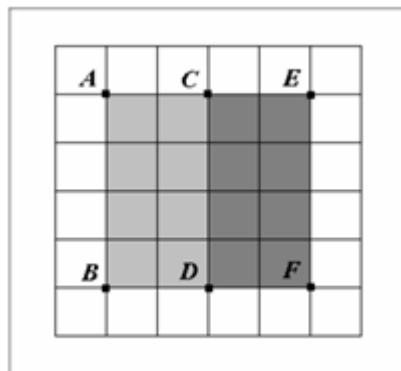
c : Fitur Haar yang diusulkan Viola dan James

d, e, f : Variasi fitur Haar yang diusulkan Leinhardt

g, h : Penguraian Liental terhadap Fitur Haar-like

i : Fitur Haar-like Viola dan James untuk menangkap struktur diagonal dalam penampilan obyek

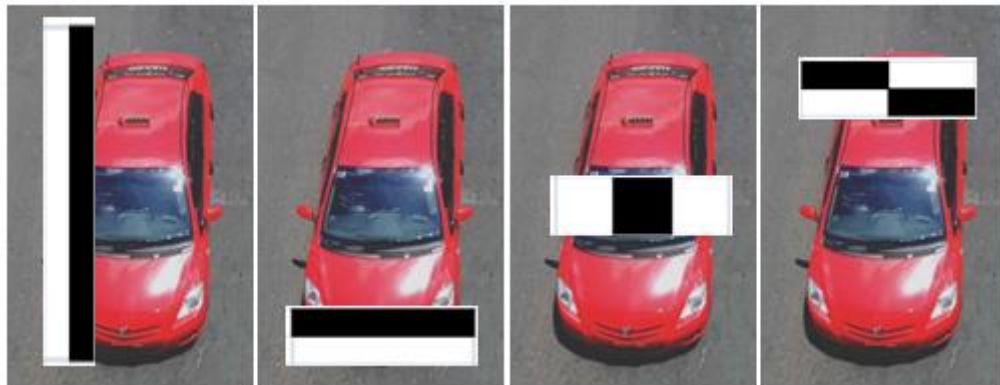
Perhitungan nilai fitur Gambar 3 di atas ditunjukkan pada Gambar 4 dan formula (1) sebagai berikut:



Gambar II.4. Fitur Persegi Haar-like

$$\text{NilaiFitur}(ABFE) = \text{JumlahNilaiPixel}(ABDC) - \text{JumlahNilaiPixel}(CDFE) \dots\dots (1)$$

Apabila NilaiFitur(ABFE) memiliki nilai di atas *threshold* maka dikatakan memenuhi syarat. Seperti dijelaskan pada gambar 2, apabila sebuah fitur dikatakan tidak memenuhi syarat, maka area ABFE tidak terdapat obyek yang dideteksi dan area persegiunya berpindah lokasi akan tetapi jika persegi ABFE memenuhi fitur, maka aturan fitur berikutnya dilakukan. Jika semua syarat fitur dipenuhi dikatakan pada persegi ABFE dikatakan terdapat obyek.
Proses pendeteksian mobil menggunakan *Haar-like feature* ditunjukkan pada Gambar 5 sebagai berikut:



Gambar II.5. Pendeteksian Kendaraan dengan Haar-like Feature

Integral image

Sebuah citra digital memiliki komponen nilai RGB (kombinasi dari warna merah, hijau dan biru). Dari nilai RGB tersebut dapat diketahui nilai grayscale (derajat keabu-abuan) yang dihitung menggunakan formula (2) sebagai berikut:

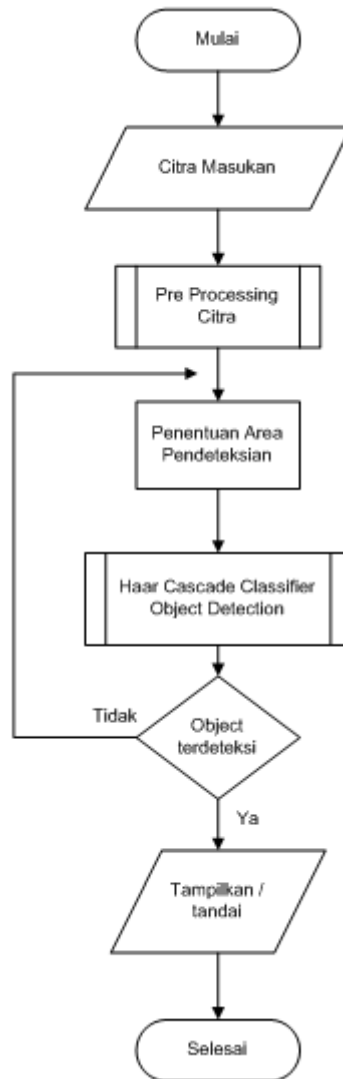
$$\text{Grayscale_pixel} = 0.2989R + 0.5870G + 0.1140B \dots\dots\dots (2)$$

Sebagai contoh sebuah *pixel* memiliki kombinasi warna R=100, B=100 dan G=100 maka nilai grayscale menurut formula 2 di atas sama dengan 99.99. Sebuah citra yang dirubah menjadi grayscale ditunjukkan pada gambar 6 sebagai berikut:



Gambar II.6. Perbedaan Citra Asli dengan Grayscale

Proses pendeteksian obyek ditunjukkan pada diagram alir di Gambar 7 sebagai berikut:



Gambar II.7. Pendeteksian Obyek dengan Haar Cascade Clasifier

Pertama, ditentukan terlebih dahulu area yang akan dideteksi apakah terdapat obyek atau tidak. Proses berikutnya adalah melakukan pendeteksian obyek menggunakan Haar Cascade Clasifier dengan langkah-langkah yang akan dijelaskan sebagai berikut. Citra integral adalah sebuah citra yang nilai tiap pixel-nya merupakan penjumlahan nilai pixel atas dan kirinya. Sebagai contoh pada gambar 5 (c) sebuah daerah persegi yang akan di-scan menggunakan persegi gelap terang memiliki nilai sebagai berikut:

2	3	1	3	6	5
3	1	2	5	4	4
1	2	3	4	5	4
4	4	5	6	7	3
5	4	3	6	7	4
1	2	3	4	5	6

Gambar II.8 Citra Masukan

Pada citra masukan yang diberi persegi pada gambar 5 (c) terlihat pada gambar 9 sebagai berikut:

2	3	1	3	6	5
3	1	2	5	4	4
1	2	3	4	5	4
4	4	5	6	7	3
5	4	3	6	7	4
1	2	3	4	5	6

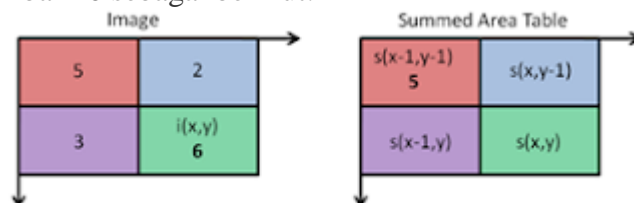
Gambar II.9 Persegi Haar Like Pada Citra masukan

Proses perhitungan nilai gelap dan nilai terang di formula 1 menghasilkan nilai Fitur Haar sama dengan 22. Untuk menghitung Fitur nilai Haar menggunakan Summed Area Table atau yang dikenal sebagai Integral image, pertama dibentuk matriks nilai integral image.

Proses integral image matriks citra ditentukan menggunakan formula (3) sebagai berikut:

$$s(x,y)=i(x,y)+s(x-1,y)+s(x,y-1)-s(x-1,y-1) \quad (3)$$

Dimana x-1 dan y-1 adalah pixel tetangga dari pixel yang akan dihitung seperti terlihat pada gambar 10 sebagai berikut:



Gambar II.10 Pixel Tetangga pada Proses Integral image

Menggunakan formula 3 di atas, *integral image* dari citra masukan di Gambar 6 terlihat pada gambar 11 sebagai berikut:

2	5	6	9	15	20
5	9	12	20	30	39
6	12	18	30	45	58
10	20	31	49	71	87
15	29	43	67	96	116
16	32	49	77	111	137

Gambar II.11 Matriks Integral Image dari Citra Masukan

Nilai fitur Haar dari area matriks di gambar 11 dihitung menggunakan formula (4) sebagai berikut:

$$i(x',y')=s(A)+s(D)+s(B)-s(C) \quad (4)$$

Sehingga:

$$\text{fitur nilai Haar} = (32+2-16-5)-(49+5-32-6)+(77+6-49-9) = 22$$

Hasil ini sesuai dengan perhitungan secara manual menggunakan formula 1. Nilai 22 tersebut kemudian dibandingkan dengan *threshold* yang sudah ditentukan sebagai pendeteksian obyek. Apabila nilai fitur Haar lebih tinggi daripada *threshold*, maka dapat dikatakan pada area tersebut memenuhi filter Haar. Sesuai *flowchart* pada gambar 2, proses ini akan dilanjutkan untuk menguji kembali area tersebut dengan filter Haar yang lain dan apabila seluruh filter Haar terpenuhi maka dikatakan pada area tersebut terdapat obyek yang diamati.

1.1.2 Regresi Non Linear

Regresi non linear adalah suatu metode untuk mendapatkan model *non linear* yang menyatakan hubungan *variabel dependen* dan *variabel independen*. *Regresi non linear* dapat mengestimasi model hubungan *variabel dependen* dan *independen* dalam bentuk *non linear* dengan keakuratan yang lebih baik daripada *regresi linear*, karena dalam mengestimasi model dipakai *iterasi algoritma*. *Regresi non linear* menghasilkan grafik dengan garis melengkung. Pada *regresi non linear* terdapat *regresi kuadratik* dan *regresi kubik* [10].

a. RUMUS Regresi Non Linear Kuadratik

$$Y = a + bX + cX^2 \quad \text{..... (1)}$$

$$Y = \beta_0 + \beta_1X + \beta_2X^2 + \varepsilon \quad \text{.....(2)}$$

atau

$$\mu_{Y|x} = \beta_0 + \beta_1X + \beta_2X^2 \quad \text{.....(3)}$$

Pada rumus (1), (2) dan (3) merupakan rumus umum yang dimiliki regresi *non linear* dengan berbeda variabel.

b. Cari β dengan eliminasi

$$n\beta_0 + \beta_1 \sum_{i=1}^n xi + \beta_2 \sum_{i=1}^n xi^2 = \sum_{i=1}^n y \quad \text{.....(4)}$$

$$\beta_1 \sum_{i=1}^n xi + \beta_2 \sum_{i=1}^n xi^2 + \beta_3 \sum_{i=1}^n xi^3 = \sum_{i=1}^n xiy \quad \text{.....(5)}$$

$$\beta_1 \sum_{i=1}^n xi^2 + \beta_2 \sum_{i=1}^n xi^3 + \beta_3 \sum_{i=1}^n xi^4 = \sum_{i=1}^n xi^2 y \quad \text{.....(6)}$$

Pada rumus (4), (5) dan (6) merupakan rumus turunan dari rumus awal untuk mencari nilai akhir dengan metode eliminasi.

c. R^2

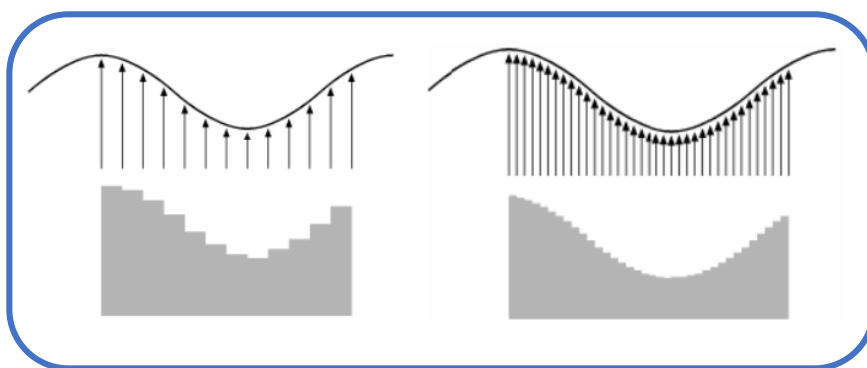
Koefisien determinasi \rightarrow mengetahui garis regresi apakah mendekati dengan garis non linier (0-1). $KD = R^2 \times 100\%$

I.1.3 ADC (Analog To Digital Converter)

ADC mengubah input analog menjadi kode – kode digital. ADC banyak digunakan sebagai pengatur proses sistem industri, konversi analog to digital dan rangkaian pengukuran/pengujian. ADC memiliki 2 karakter prinsip, yaitu

a) Kecepatan sampling

Kecepatan sampling suatu ADC menyatakan seberapa sering sinyal analog dikonversikan ke bentuk sinyal digital pada selang waktu tertentu. Kecepatan sampling biasanya dinyatakan dalam sample per second (SPS).



Gambar II. 12 Kecepatan sampling rendah dan tinggi

b) Resolusi

Resolusi ADC menentukan ketelitian nilai hasil konversi ADC. Ketelitian pada 8 bit dan 12 bit tergantung error terhadap tegangan per stepnya sesuai dengan

teorema Nyquist yaitu $\frac{1}{2}$ dari tegangan per step yang dihasilkan, teorema nyquist memberikan penjelasan bahwa frekuensi cuplik minimal harus 2 (dua) kali lipat frekuensi maksimum yang dikandung sinyal yang bersangkutan.. Sebagai contoh: ADC 8 bit akan memiliki output 8 bit data digital, ini berarti sinyal input dapat dinyatakan dalam 255 (2^8-1) nilai diskrit. ADC 12 bit memiliki 12 bit output data digital, ini berarti sinyal input dapat dinyatakan dalam 4096 nilai diskrit. Dari contoh diatas ADC 12 bit akan memberikan ketelitian nilai hasil konversi yang jauh lebih baik daripada ADC 8 bit.

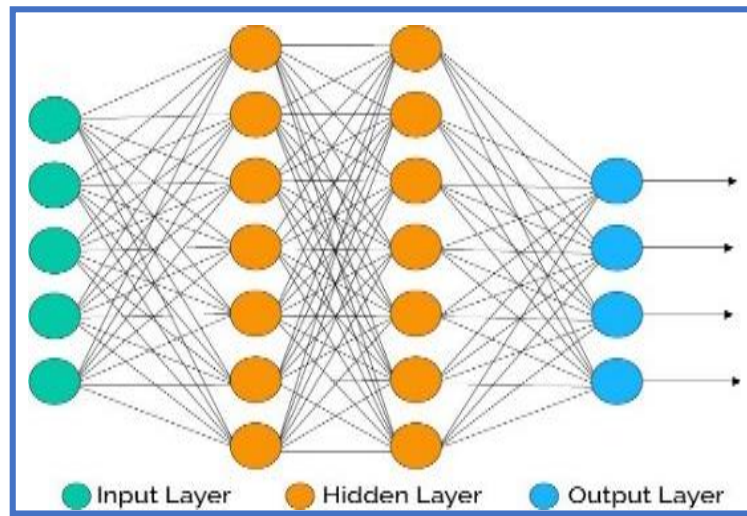
I.1.4 Prinsip Neural Network

Neural Network merupakan kategori ilmu Soft Computing. Neural Network sebenarnya mengadopsi dari kemampuan otak manusia yang mampu memberikan stimulasi/rangsangan, melakukan proses, dan memberikan output. Output diperoleh dari variasi stimulasi dan proses yang terjadi di dalam otak manusia. Kemampuan manusia dalam memproses informasi merupakan hasil kompleksitas proses di dalam otak. Misalnya, yang terjadi pada anak-anak, mereka mampu belajar untuk melakukan pengenalan meskipun mereka tidak mengetahui algoritma apa yang digunakan. Kekuatan komputasi yang luar biasa dari otak manusia ini merupakan sebuah keunggulan di dalam kajian ilmu pengetahuan.

Fungsi dari Neural Network diantaranya adalah:

- Pengklasifikasian pola
- Memetakan pola yang didapat dari input ke dalam pola baru pada output
- Penyimpan pola yang akan dipanggil kembali
- Memetakan pola-pola yang sejenis
- Pengoptimasi permasalahan

- Prediksi



Gambar II. 13 Neural Network