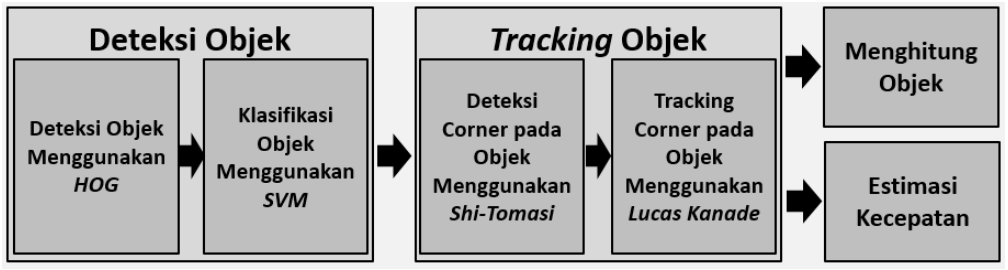
BAB II
TINJAUAN PUSTAKA

2.1 Deteksi Dan *Tracking* Multi Objek Berdasarkan Deteksi Ciri Pada *Intelligent Transportation*

Pada tugas akhir ini, penelitian yang berjudul deteksi dan *tracking* multi objek berdasarkan deteksi ciri pada *intelligent transportation*[2], digunakan sebagai deteksi dan *tracking* objek kendaraan.

Objek di deteksi melalui dua langkah, pada langkah pertama dibentuk sebuah detektor menggunakan *Histogram of Oriented Gradient* (HOG) yang berisi *feature vector*. Pada langkah kedua objek di klasifikasi menggunakan *feature vector* yang dibentuk oleh HOG menggunakan linear *Support Vector Machine* (SVM)[2].



Gambar 2.1 Diagram blok sistem deteksi dan *tracking* objek[2].

Pada tahap awal proses *tracking*, objek yang terdeteksi dilakukan proses pencarian *corner* menggunakan metode Shi-Tomashi *corner detector*, selanjutnya metode Lucas Kanade *tracker* digunakan untuk *tracking* terhadap *corner* yang telah dideteksi sebelumnya[2].

Sebuah *frame* video atau citra diambil dan dilakukan proses pergeseran *window* deteksi untuk mendeteksi sebuah objek. Ilustrasi pergeseran *window* dapat dilihat pada Gambar 2.*2*. Pada bagian citra yang terdapat *window* deteksi tersebut dilakukan ekstraksi *feature* menggunakan HOG dan *feature* tersebut dibandingkan dengan data latih menggunakan SVM. Jika objek terdeteksi maka bagian citra tersebut bernilai 1, jika tidak ada objek terdeteksi maka bernilai 0. Koordinat awal pada *window* yang terdapat objek ditambahkan ke dalam vektor

penyimpanan. Selanjutnya *window* deteksi digeser ke sebelah kanan kemudian ke bawah dengan jarak *pixel* tertentu dan dilakukan operasi yang sama[2].



Gambar 2.2 Ilustrasi pergeseran *window* pada proses deteksi objek.

Pada objek yang sama mungkin terdapat lebih dari satu titik koordinat deteksi, oleh karena itu dilakukan proses clustering pada titik – titik koordinat yang berdekatan, lalu ditentukan titik tengah dari objek hasil clustering ini yang akan dijadikan masukan untuk proses tracking objek[2].

Pada awalnya vektor *tracking* kosong, lalu sistem akan melakukan deteksi objek mobil menggunakan HOG. Titik *corner* yang baik dicari pada setiap objek mobil yang terdeteksi menggunakan metode Shi-Tomashi dalam jangkauan 15 *pixel* dari titik tengah objek. Langkah yang sama dilakukan juga untuk deteksi objek motor. Semua hasil titik *corner* yang baik tersebut ditambahkan kedalam vektor *tracking* untuk dilakukan proses *tracking*. Jika vektor *tracking* sudah terisi maka dilakukan proses *tracking*. Proses *tracking* ini menghitung perubahan titik *corner* pada satu *frame* dengan *frame* sebelumnya. Jika satu *corner* ada pada suatu *frame* dan ada pula pada *frame* sebelumnya, maka *corner* tersebut dianggap sebagai *corner* yang sama, dalam arti lain objek tersebut dianggap sebagai objek yang sama[2].

2.2 *Auto – Rectification Of User Photos*

Pada tugas akhir ini, penelitian yang berjudul *Auto – Rectification Of User Photos*[3], digunakan sebagai proses rektifikasi gambar untuk menghilangkan distorsi perspektif.

Salah satu kesalahan yang relatif umum ketika mengambil foto adalah membuat sebuah rotasi kamera yang tidak disengaja (ke segala arah) yang mengarah ke perspektif dan distorsi lainnya. Tujuan dari proyek ini adalah untuk secara otomatis memperbaiki distorsi ini dan mendekatkan gambar ke tampilan

fronto-paralel (tampilan fronto-paralel sesuai dengan orientasi kamera sehingga objek persegi panjang yang tegak pada kenyataan, misalnya pintu atau jendela, berbentuk segi empat juga pada gambar. Secara khusus, kami melakukan rektifikasi affine yang mengembalikan paralelisme garis yang sejajar dalam tampilan fronto-parallel. Homografi 2D yang sesuai dengan rektifikasi affine dihitung dari sepasang vanishing point. Komponen dalam matriks H tersebut berisi :

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_a & l_b & l_c \end{bmatrix}$$

Dimana *vanishing line* $\vec{l}_{12} = [l_a \quad l_b \quad l_c]^T$ didapatkan dari perkalian silang vektor $\vec{l}_{12} = \vec{v}p_1 \times \vec{v}p_2$. Misalkan $\vec{v} = [v_x \quad v_y \quad 0]^T$ adalah *vanishing point* setelah proses rektifikasi yang dekat ke sumbu Y gambar, $\vec{Y} = [0 \quad 1 \quad 0]^T$. Kemudian $\theta = \cos^{-1}\left(\frac{v_y}{\sqrt{v_x^2+v_y^2}}\right)$ adalah sudut dari rotasi, dan matriks rotasi adalah :

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Perkalian matriks dari matriks rektifikasi dan rotasi, $T = RH$, digunakan untuk melengkungkan gambar. Gambar yang telah dilengkungkan dipotong dan dikembalikan skala nya ke ukuran gambar awal[3].

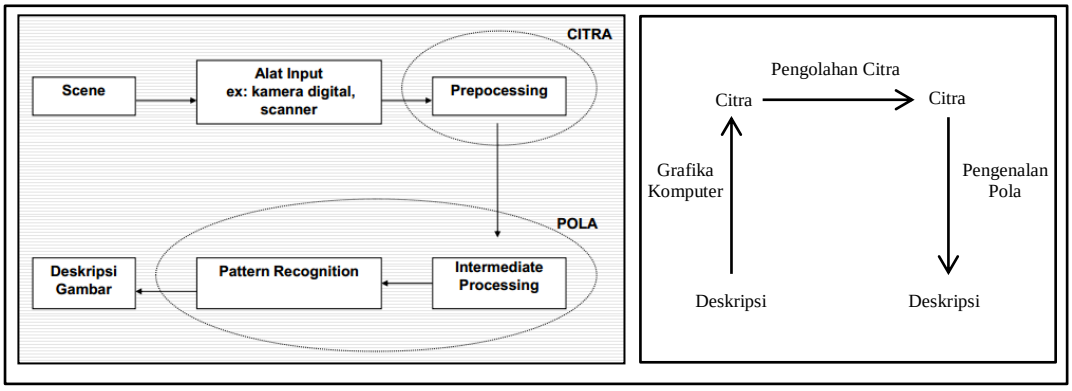
Untuk estimasi *vanishing point*, model terdiri dari sepasang edgelet yang dipilih secara acak (selanjutnya disebut sebagai model edgelets). Hipotesis *vanishing point* yang sesuai dengan model adalah titik perpotongan garis edgelet yang sesuai dengan dua model edgelets. Jadi, dengan model $M(E_{1,}\, E_2)$ yang terdiri dari edgelet $E_1(\overrightarrow{x_1}, \overrightarrow{d_1}, s_1)$ dan $E_2(\overrightarrow{x_2}, \overrightarrow{d_2}, s_2)$, hipotesis *vanishing point* adalah $\vec{V}_M = \vec{l}_{E1} \times \vec{l}_{E2}$ (di mana × menunjukkan perkalian silang vektor) adalah titik perpotongan garis edgelet $\vec{l}_{E1}$ dan $\vec{l}_{E1}$ (dalam matematika homogen, perkalian silang dari vektor homogen yang sesuai dengan garis menghasilkan titik perpotongan dalam koordinat homogen). Perhatikan bahwa $\vec{V}_M$ mungkin tidak

terbatas (ditunjukkan dengan nilai nol pada koordinat ketiga). perkalian silang selalu dapat dihitung-- kecuali pada kasus yang degenerasi, misalkan ketika garis edgelet untuk model edgelets bertepatan atau pemilih acak mengambil edgelet yang sama dua kali sebagai model edgelets. Begitu model baru (acak) dihasilkan, maka diperiksa apakah terjadi degenerasi, dan jika degenerasi, maka model itu ditolak dan menghasilkan yang lain. Untuk alasan performa, pilihan model edgelet tidak benar-benar acak. Sebagai gantinya, memilih edgelet model pertama secara acak dari 20 persentil teratas dari rangkaian edgelets (ingat bahwa array ini diurutkan berdasarkan kekuatan tepi) dan edgelet model kedua dari 50 persentil teratas. Ini secara efektif menyeimbangkan sistem ke tepi yang lebih kuat[3].

Secara matematis, edgelet direpresentasikan sebagai $E = \{\vec{x}, \vec{d}, s\}$. dimana $\vec{x}$ mewakili koordinat homogen untuk lokasi pixel tepi, $\vec{d}$ mewakili arah tepi pada koordinat homogen dan s adalah kekuatan tepi. Garis edgelet, $\vec{l}_E$ sesuai dengan edgelet E didefinisikan sebagai garis yang melewati $\vec{x}$ dan sejajar dengan $\vec{d}$ [3].

2.3 Pengolahan Citra

Computer Vision adalah proses otomatis yang mengintegrasikan sejumlah besar proses untuk persepsi visual, seperti akuisisi citra, pengolahan citra, klasifikasi, pengenalan (*recognition*), membuat keputusan. Di dalam bidang komputer, ada 3 bidang studi yang berkaitan dengan citra, namun tujuan ketiganya berbeda, yaitu Grafika Komputer (*Computer Graphics*), Pengolahan Citra (*Image Processing*), Pengenalan Pola (*Pattern Recognition / image interpretation*). (Dharmayanti, 2009)

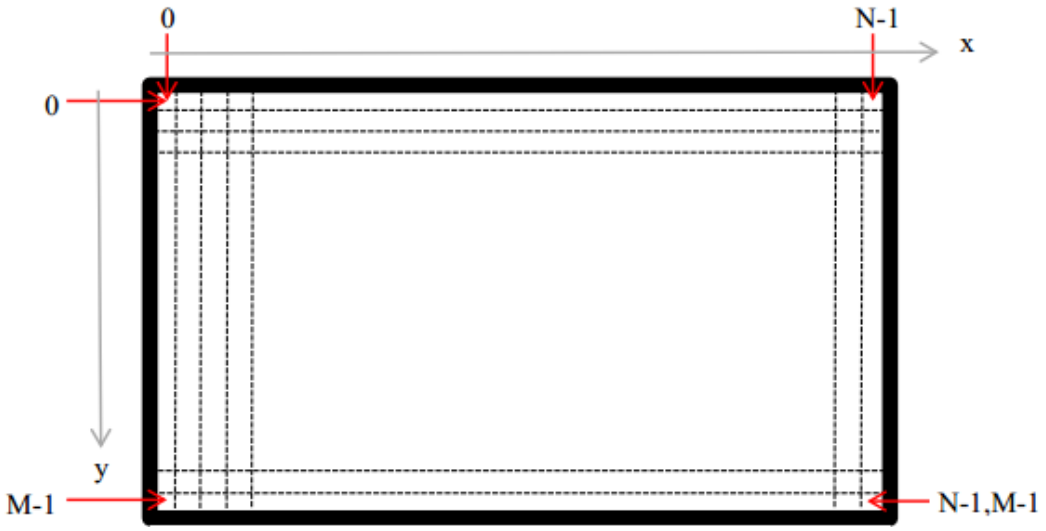


Gambar 2.3 Diagram *Computer Vision* (kiri), Pengolahan Citra (kanan)

Menurut Efford (2000), pengolahan citra adalah istilah umum untuk berbagai teknik yang keberadaannya untuk memanipulasi dan memodifikasi citra dengan berbagai cara. Foto adalah contoh gambar berdimensi dua yang bisa diolah dengan mudah. Setiap foto dalam bentuk citra digital (misalnya berasal dari kamera digital) dapat diolah melalui perangkat-lunak tertentu. Sebagai contoh, apabila hasil bidikan kamera terlihat agak gelap, citra dapat diolah agar menjadi lebih terang. Dimungkinkan pula untuk memisahkan foto orang dari latar belakangnya[4].

2.4 Citra Digital

Citra adalah suatu *matriks* dari suatu *pixel (picture elements)* yang diatur dalam suatu kolom dan baris. Terdapat dua jenis citra, yaitu citra analog dan citra digital. Citra analog adalah citra yang dihasilkan dari sistem optik yang menerima sinyal analog dan tidak dapat langsung diolah oleh komputer. Contoh sepeti citra yang tercetak di kertas atau kanvas. Citra digital adalah representasi gambar dua dimensi yang terdiri dari kumpulan titik-titik elemen gambar (*pixel*). Setiap *pixel* digambarkan sebagai satu kotak kecil. Setiap elemen gambar mempunyai koordinat posisi, sistem koordinat yang digunakan untuk menyatakan citra digital ditunjukkan pada Gambar 2.4[4] .

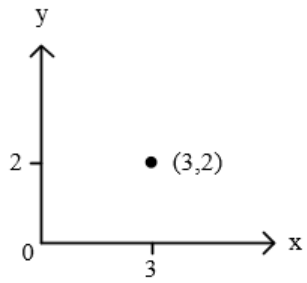


Gambar 2.4 Sistem koordinat citra berukuran M X N*[4]*

Dengan sistem koordinat yang mengikuti asas pemindaian pada layar TV standar, sebuah elemen gambar memiliki koordinat berupa sumbu x dan sumbu y, sumbu x menyatakan posisi kolom sedangkan sumbu y menyatakan posisi baris. Elemen gambar pada pojok kiri atas memiliki koordinat (0, 0) dan elemen gambar pada pojok kanan bawah memiliki koordinat (N-1, M-1)[4].

2.5 Titik

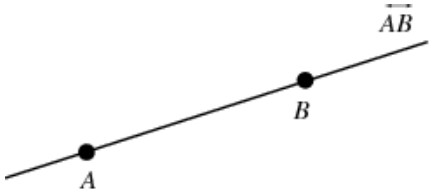
Titik adalah objek matematika 0-dimensi yang dapat ditentukan dalam ruang n-dimensi menggunakan n-tuple (x_1, x_2, ..., x_n) yang terdiri dari n koordinat. Dalam dimensi yang lebih dari atau sama dengan dua, titik kadang-kadang dianggap identik dengan vektor sehingga titik dalam ruang n-dimensi kadang-kadang disebut vektor-n. Struktur geometris dasar dari geometri dimensi tinggi (garis, bidang, ruang, dan ruang hipersi) semuanya terbangun dari jumlah titik tak terbatas yang diatur dengan cara tertentu[5]. Sebuah titik pada euclidean 2-ruang *(IR2)* ditunjukkan oleh sepasang bilangan real terurut, *(x, y)*[6].



Gambar 2.5 Titik pada IR2

2.6 Garis

Garis adalah figur satu dimensi lurus yang tidak memiliki ketebalan dan membentang tak terbatas ke dua arah. Ketika garis secara intrinsik merupakan objek berdimensi satu, garis itu mungkin tertanam dalam ruang dimensi yang lebih tinggi. Sebuah garis secara unik ditentukan oleh dua titik, dan garis yang melewati titik A dan B dilambangkan $\overleftrightarrow{AB}$. Demikian pula, panjang segmen garis yang terbatas yang berakhir pada titik-titik ini dapat dinotasikan dengan$\overline{AB}$ [5].



Gambar 2.6 Garis dan penamaannya

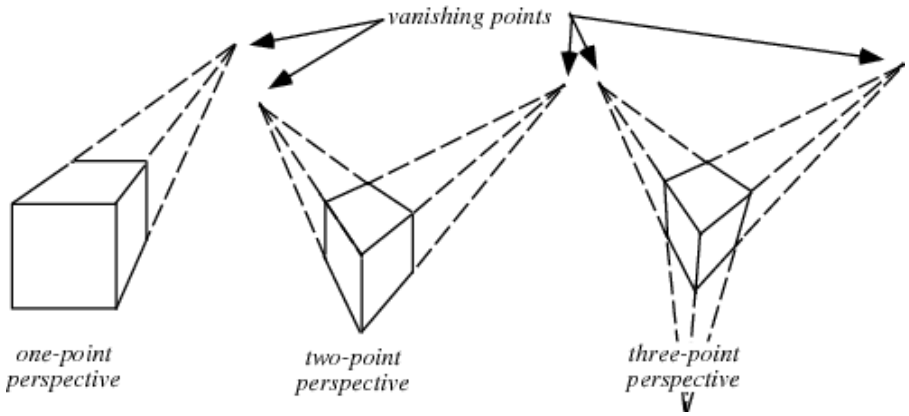
Garis pada bidang diwakili oleh persamaan seperti $ax+by+c=0$, pilihan yang berbeda dari a, b dan c yang menghasilkan garis yang berbeda. Sebuah titik $x=(x,y)^T$ terletak pada garis $l=(a,b,c)^T$ jika dan hanya jika $ax+by+c=0$. Ini dapat ditulis dalam bentuk perkalian dalam dari vektor yang merepresentasikan titik sebagai $(x,y,1)(a,b,c)^T=(x,y,1)l=0$; yaitu titik $(x,y)^T$ dalam *IR2* direpresentasikan sebagai vector-3 dengan menambahkan koordinat akhir dengan 1[6]. Sebuah vektor homogen representasi dari sebuah titik adalah dari bentuk $x=(x1,x2,x3)^T$, yang merepresentasikan titik $(x1/x3, x2/x3)^T$ di *IR2*.

Dengan dua garis $l = (a,b,c)^T$ dan $l' = (a',b',c')^T$, kami ingin menemukan perpotongan mereka. Tentukan vektor $x=l\times l'$, di mana × mewakili vektor atau perkalian silang. Dari tiga perkalian skalar identitas $l.(l\times l') = l'.(l\times l') = 0$, kita melihat bahwa $l^Tx = l'^Tx = 0$. Jadi, jika x dianggap mewakili suatu titik, maka x terletak pada kedua garis l dan l' dan karenanya adalah perpotongan dua garis[6].

2.7 *Vanishing Point*

Perspektif adalah seni dan matematika yang secara realistis menggambarkan benda tiga dimensi dalam bidang dua dimensi, kadang-kadang disebut sentris atau perspektif alami untuk membedakannya dari perspektif bicentrik. Studi proyeksi benda dalam bidang disebut *projective geometry*. Prinsip-prinsip gambar perspektif dijelaskan oleh arsitek Florentine F. Brunelleschi (1377-1446). Aturan-aturan ini dirangkum oleh Dixon (1991)[7]:

1. Ufuk tampak sebagai garis.
2. Garis lurus pada ruang tampak sebagai garis lurus pada gambar.
3. Set garis sejajar bertemu pada titik yang hilang (*vanishing point*).
4. Garis sejajar pada bidang gambar tampak sejajar dan karena itu tidak memiliki *vanishing point.*



Gambar 2.7 Perspektif dan *vanishing point*

Vanishing point adalah titik atau titik – titik yang merupakan perpanjangan dari garis – garis yang sejajar tampak berkumpul dalam gambar perspektif[8].

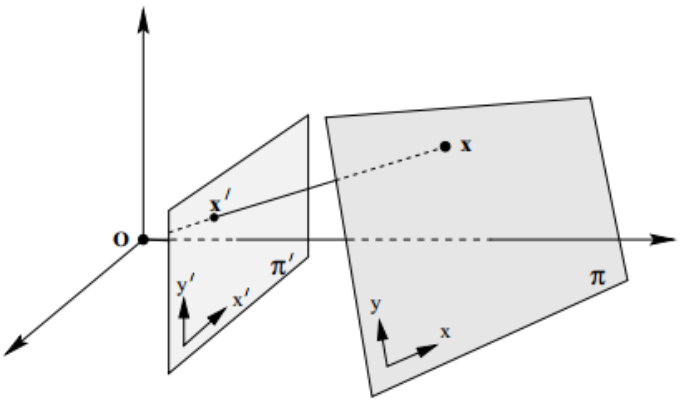
2.8 Proyeksi Geometri

Geometri adalah studi tentang sifat invarian di bawah kelompok transformasi. Dari sudut pandang ini, proyeksi geometri 2D adalah studi tentang sifat dari bidang proyeksi *IP2* yang invarian di bawah kelompok transformasi yang dikenal sebagai projektivitas. Sebuah keproyeksian adalah pemetaan yang dapat dibalik dari titik-titik di *IP2* (yaitu homogen vector-3) sampai titik-titik di *IP2* yang memetakan garis pada garis. Sebuah keproyeksian juga disebut kolineasi, transformasi proyektif atau homografi[6].

Transformasi proyektif planar adalah transformasi linier pada vektor 3 homogen yang ditunjukkan oleh matriks 3 × 3 non-tunggal:

$$\begin{pmatrix} x_1' \\ x_2' \\ x_3' \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

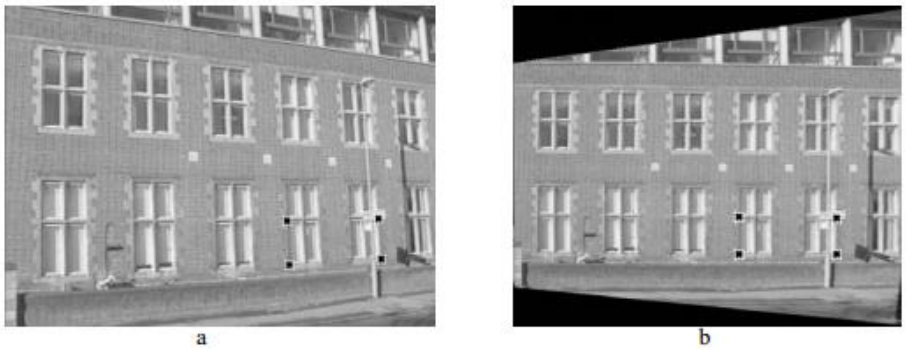
Lebih singkatnya adalah $x' = Hx$



Gambar 2.8 Proyeksi sentral memetakan titik dalam suatu bidang ke titik pada bidang lain. Proyeksi juga memetakan garis ke garis yang mungkin terlihat dengan mempertimbangkan sebuah bidang melalui pusat proyeksi yang berpotongan dengan dua bidang π dan π'. Karena garis dipetakan ke garis, proyeksi sentral adalah sebuah projektivitas dan mungkin diwakili oleh pemetaan linear dari koordinat homogen x' = Hx[6].

2.9 Rektifikasi Gambar

Sebuah bentuk terdistorsi dalam pencitraan perspektif. Misalnya, pada Gambar 2.9(a) jendela tidak berbentuk segi empat pada gambar, meskipun aslinya. Secara umum garis sejajar pada bidang nyata, tidak paralel pada gambar tapi malah menyatu ke *vanishing point*. Kita telah melihat bahwa gambar proyeksi sentral dari sebuah bidang (atau bagian dari bidang) berhubungan dengan bidang asli melalui transformasi proyektif, dan karenanya gambar tersebut merupakan distorsi proyektif dari yang asli. maka mungkin untuk "mengembalikan" transformasi proyektif ini dengan menghitung transformasi terbalik dan menerapkannya pada citra. Hasilnya akan menjadi citra baru yang disintesis dimana objek pada bidang diperlihatkan dengan bentuk geometrisnya yang benar. Perhatikan bahwa tanah dan bagian depan tidak berada pada bidang yang sama, transformasi proyektif harus diterapkan untuk rektifikasi bagian depan yang tidak sama dengan yang digunakan untuk tanah[6].

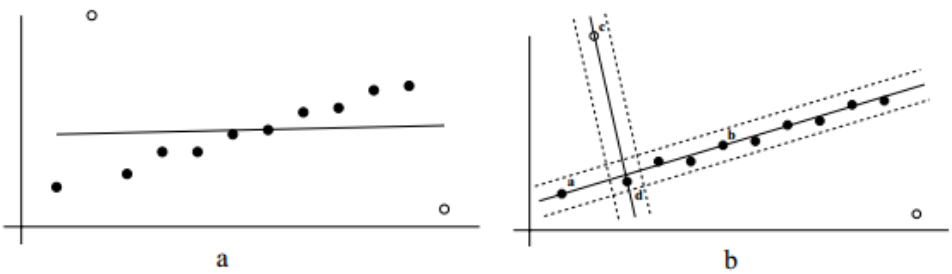


Gambar 2.9 (a) Citra asli dengan distorsi perspektif --garis jendela dengan jelas menyatu pada titik yang terbatas. (b) Tampilan ortogonal depan yang disintesis dari dinding depan.

Gambar 2.9(a) dinding dihubungkan melalui transformasi proyektif ke dinding dengan geometri sebenarnya. Transformasi terbalik dihitung dengan memetakan keempat sudut jendela yang dicitrakan ke sudut-sudut persegi panjang yang tepat. Empat titik korespondensi menentukan transformasi. Transformasi ini kemudian diterapkan ke keseluruhan gambar. Perhatikan bahwa bagian gambar tanah tunduk pada distorsi proyektif lebih lanjut. Ini juga dapat dihapus oleh transformasi proyektif[6].

2.10 *Random Sample Consensus* (RANSAC)

Algoritma *RANdom SAmple Consensus* (RANSAC) mampu mengatasi sebagian besar outlier. Idenya sangat sederhana: dua titik dipilih secara acak; titik-titik ini menentukan sebuah garis. Dukungan untuk garis ini diukur dengan jumlah titik yang berada dalam ambang batas jarak. Pemilihan acak ini diulang beberapa kali dan garis dengan dukungan paling banyak dianggap sesuai. Titik-titik di dalam ambang batas adalah inliers (dan merupakan konsensus yang ditetapkan secara naluriah). Intuisi adalah bahwa jika salah satu poinnya adalah outlier maka garis tidak akan banyak mendapat dukungan, lihat gambar Gambar 2.*10* (b) [6].



Gambar 2.10 Titik hitam adalah inliers, titik putih adalah outlier. (a) *leastsquares* (regresi ortogonal) yang sesuai dengan data titik sangat dipengaruhi oleh outlier. (b) Dalam algoritma RANSAC, dukungan untuk garis melalui pasangan titik yang dipilih secara acak, diukur dengan jumlah titik dalam ambang batas jarak untuk garis. Garis putus-putus menunjukkan jarak ambang batas. Untuk garis yang ditunjukkan, dukungannya adalah 10 untuk garis (a, b) (di mana kedua titik a dan b adalah inliers); dan 2 untuk garis (c, d) dimana titik c adalah outlier[6].

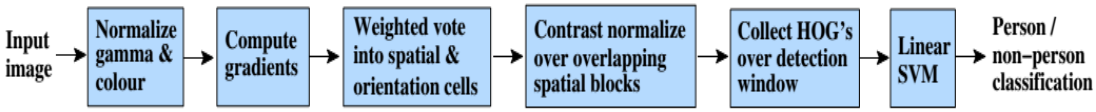
2.11 *Histogram of Oriented Gradient* (HOG)

Histogram Oriented Gradients (HOG) merupakan salah satu deskriptor ciri yang digunakan dalam pengolahan citra untuk keperluan deteksi objek, teknik yang digunakan dengan menghitung keadaan arah perubahan tingkat keabuan (gradien) pada bagian lokal dari sebuah citra. Metode ini hampir sama dengan metode *edge orientation histograms*, *scale-invariant feature transform* and *shape contexts*, tapi perbedaannya pada komputasi menggunakan teknik membagi citra ke dalam beberapa bagian (*dense grid*) dan menggunakan normalisasi *overlapping* pada kontras lokal untuk meningkatkan akurasi[2].

HOG pertama kali dikenalkan pada tahun 2005 oleh Navneet Dalal dan Bill Triggs peneliti dari *French National Institute for Research in Computer Science and Control* (INRIA). Penelitian tersebut fokus pada algoritma memecahkan masalah mendeteksi pejalan kaki untuk citra statis, selanjutnya mereka mengembangkan hasil penelitiannya untuk mendeteksi manusia pada citra dinamis (video dan kamera), yang kemudian di kembangkan lagi untuk mendeteksi binatang dan kendaraan pada citra statis[2].

Ide dasar dari metode ini adalah wujud dan bentuk objek lokal dapat dikarakteristikkan dengan mendistribusikan intensitas gradien lokal atau arah tepi, meskipun tanpa memiliki presisi yang tepat dari kesesuaian posisi gradien atau tepi. Secara teknis implementasi HOG dengan membagi citra ke dalam bagian-bagian yang kecil yang dinamakan sel (“*cell”*), pada setiap sel dihitung histogram lokal arah gradien atau tepi dari setiap *pixel* pada sel. Kombinasi dari histogram-histogram lokal akan membentuk representasi. Untuk pendeteksian lebih baik terhadap gangguan pencahayaan, bayangan dan lain-lain, dapat digunakan normalisasi kontras pada respon lokal. Normalisasi kontras dilakukan dengan

mengakumulasi beberapa histogram lokal yang berdekatan ke dalam bagian yang lebih besar dari sel yang dinamakan blok ("*block*"). Histogram-histogram tersebut dijadikan fitur vektor yang akan diklasifikasikan menggunakan linear *Support Vector Machine* untuk menghasilkan keluaran berupa objek atau bukan objek. Gambar 2.*11* merupakan proses ekstraksi fitur dan deteksi manusia menggunakan HOG[2].



Gambar 2.11 Proses ekstraksi fitur dan deteksi manusia menggunakan HOG dan linear SVM.

2.12 Arus lalu – lintas

Arus lalu - lintas (Q) untuk setiap gerakan dikonversi dari kendaraan per-jam menjadi satuan mobil penumpang (smp) per-jam dengan menggunakan ekivalen kendaraan penumpang (emp) untuk masing - masing pendekat terlindung dan terlawan[9] :

Tabel 2.1 Ekivalen kendaraan penumpang

Jenis Kendaraan	emp untuk tipe pendekat :	
	Terlindung	Terlawan
Kendaraan Ringan (LV)	1.0	1.0
Kendaraan Berat (HV)	1.3	1.3
Sepeda Motor (MC)	0.2	0.4

2.13 Derajat Kejenuhan

Derajat kejenuhan (DS) didefinisikan sebagai rasio arus terhadap kapasitas, digunakan sebagai faktor utama dalam penentuan tingkat kinerja simpang dan segmen jalan. Nilai DS menunjukan apakah segmen jalan tersebut mempunyai masalah kapasitas atau tidak. Derajat kejenuhan dihitung dengan menggunakan arus dan kapasitas dinyatakan dalam smp/jam[9].

$$DS = Q / C$$

Dimana :

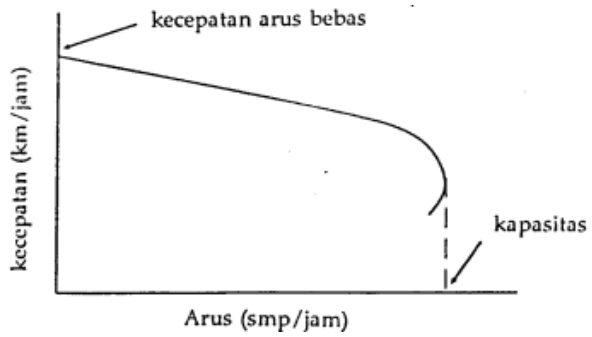
DS = derajat kejenuhan

Q = arus lalu – lintas (smp/jam)

C = kapasitas (smp/jam)

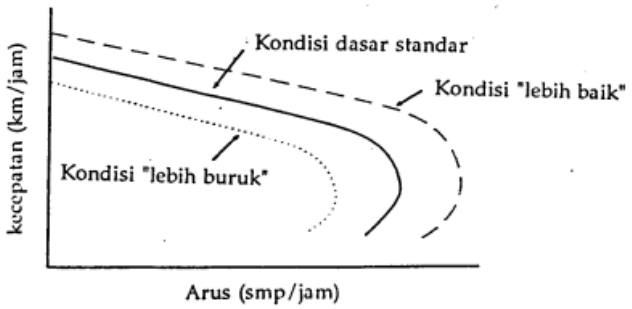
2.14 Kecepatan – Arus – Kerapatan

Prinsip dasar analisa kapasitas segmen jalan adalah kecepatan berkurang jika arus bertambah. Pengurangan kecepatan akibat penambahan arus adalah kecil pada arus rendah tetapi lebih besar pada arus yang lebih tinggi. Dekat kapasitas, pertambahan arus yang sedikit akan menghasilkan pengurangan kecepatan yang besar. Hal ini terlihat pada Gambar 2.*12,* Hubungan ini telah ditentukan secara kuantitatif untuk kondisi 'standar', untuk setiap tipe jalan. setiap kondisi standar mempunyai geometrik standar dan karakteristik lingkungan tertentu. Jika karakteristik jalan "lebih baik" dari kondisi standar (misalnya lebih lebar dari lebar jalur lalu-lintas normal), kapasitas menjadi lebih tinggi dan kurva bergesar ke sebelah kanan, dengan kecepatan lebih tinggi pada arus tertentu.



Gambar 2.12 Bentuk umum hubungan kecepatan – arus

Jika karakteristik jalan "lebih buruk" dari kondisi standar (misalnya hambatan samping tinggi) kurva bergeser ke kiri, kapasitas menjadi berkurang dan kecepatan pada arus tertentu lebih rendah seperti terlihat pada Gambar 2.*13*.



Gambar 2.13 Hubungan kecepatan – arus untuk kondisi standar dan bukan standar.

Untuk setiap tipe jalan, kurva standar untuk tipe jalan tersebut telah ditentukan berdasarkan data empiris. Analisa perilaku lalu-lintas kemudian dilakukan sebagai berikut :

1. Penentuan kecepatan arus bebas dan kapasitas untuk kondisi dasar yang ditentukan sebelumnya pada setiap tipe jalan.
2. Perhitungan kecepatan arus bebas dan kapasitas untuk kondisi jalan sesungguhnya dengan menggunakan tabel berisi faktor penyesuaian yang ditentukan secara empiris menurut perbedaan antara karakteristik dasar dan sesungguhnya dari geometrik, lalu-lintas dan lingkungan jalan yang diamati.
3. Penentuan kecepatan dari kurva umum kecepatan-arus untuk kecepatan arus bebas yang berbeda pada sumbu-y, dimana arus dinyatakan dengan derajat kejenuhan pada sumbu-x.

Model yang tepat dengan data kecepatan-arus empiris sering diperoleh dengan menggunakan model Rejim Tunggal:

$$V = FV\left[1 - \left(D/D_j\right)^{\ell - 1}\right]^{1/(1-m)}; \quad D_0/D_j = [(1-m)/(\ell - m)]^{1/(\ell - 1)}$$

dimana :

V = Kecepatan Tempuh

FV = Kecepatan arus bebas (km/jam)

D = Kerapatan (smp/km) (dihitung sehagai Q/V)

D_j = Kerapatan pada saat jalan mengaami kemacetan total(smp/km)

D_0 = Kerapatan pada kapasitas (smp/km)

ℓ,m = Konstanta