IMPLEMENTASI METODE *HIDDEN MARKOV MODEL* DAN *GABOR FILTER* UNTUK MENDETEKSI PELANGGARAN LALU LINTAS KENDARAAN

Kukuh Setiawan

Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
Email : kukusetiawan@email.unikom.ac.id

ABSTRAK

Jumlah pelanggaran lalu lintas kendaraan dari tahun ke tahun semakin meningkat. Hal itu diketahui dari data analisa dan evaluasi pelanggaran lalu lintas 2015, terdapat kenaikan jumlah pelanggaran lalu lintas yang dilakukan pengendara jika dibandingkan dengan periode di tahun 2014. Kemudian kaitannya dengan penelitian ini, yaitu mengenali jenis pelanggaran tersebut dengan cara mengimplementasikan metode *Hidden Markov Model* dan *Gabor Filter* dalam pengolahan citra berdasarkan masalah yang dipaparkan. Hal yang perlu diperhatikan dalam mengetahui pelanggaran tersebut yaitu mengklasifikasikan citra pelanggaran berdasarkan rambu di sekitar kendaraan. Dan dalam batasan masalah yang diamati penelitian fokus kepada rambu dilarang "Perboden" dan dilarang "balik arah". Sehingga pelanggaran itulah yang dijadikan penelitian menggunakan metode untuk dapat menghasilkan kesimpulan jenis pelanggaran. Ketika menjumpai rambu dengan pola seperti dilarang "Perboden" dan dilarang "balik arah", maka pelanggaran dapat dideteksi dari klasifikasi ekstraksi ciri pada saat melakukan tahapan *preprocessing* citra. Kemudian metode yang digunakan untuk mendeteksi citra pelanggaran ini yaitu *Hidden Markov Model* dalam proses pengklasifikasian citra dan *Gabor Filter* sebagai salah satu cara *preprocessing* citra. Dengan mengimplementasikan metode tersebut dimaksudkan akan mampu memenuhi kebutuhan mendeteksi pelanggaran yang dilakukan oleh kendaraan. Hasil dari penelitian ini berupa data klasifikasi *training* yang digunakan untuk mendeteksi data *testing*, dan menghasilkan output jenis pelanggaran yang dimaksudkan. Dan dari pengujian *K-Fold Cross Validation* menggunakan 82 data didapatkan nilai akurasi rata-rata 70,31% benar dalam klasifikasi dan 29.69% salah.

Kata kunci : *Gabor Filter*, *Hidden Markov Model*, *image processing*, klasifikasi citra, pelanggaran kendaraan.

1. PENDAHULUAN

Peraturan lalu lintas dibuat untuk kepentingan umum dengan tujuan supaya tercipta ketertiban dalam berkendara, karena dengan aturan yang dipatuhi maka akan tercipta suasana nyaman saat menggunakan fasilitas lalu lintas. Namun masih saja terdapat pihak yang melakukan pelanggaran, dan jumlah pelanggaran lalu lintas kendaraan dari tahun ke tahun semakin meningkat. Hal itu diketahui dari data analisa dan evaluasi pelanggaran lalu lintas 2015, terdapat kenaikan jumlah pelanggaran lalu lintas yang dilakukan pengendara jika dibandingkan dengan periode di tahun 2014.[1] Dan di Kota Bandung pada satu pekan saja terdapat rata-rata 1200 berkas perkara pelanggaran lalu lintas yang disidangkan.[2]

Dalam kaitannya tentang penelitian ini, yaitu mendeteksi pengolahan citra pelanggaran kendaraan yang akan dijadikan kajian untuk menentukan jenis pelanggaran yang terjadi berdasarkan masalah yang dipaparkan. Hal yang perlu diperhatikan dalam mengetahui pelanggaran yaitu obyek kendaraan dan simbol rambu di sekitarnya. Pelanggaran itulah yang akan dijadikan penelitian menggunakan metode untuk dapat menghasilkan kesimpulan jenis pelanggaran. Ketika menjumpai rambu dengan pola seperti dilarang "Perboden" atau dilarang "balik arah" maka pelanggaran dapat dideteksi dengan menggunakan pengenalan pola. Metode yang digunakan untuk mendeteksi citra pelanggaran ini yaitu Hidden Markov Model sebagai klasifikasi citra dan gabor filter sebagai salah satu *preprocessing* citra, dengan mengimplementasikan metode tersebut dimaksudkan agar mampu memenuhi kebutuhan mendeteksi pelanggaran yang dilakukan oleh kendaraan.

Mengenai *Hidden Markov Model* (HMM) merupakan metode yang dapat menentukan pola pelanggaran kendaraan berdasarkan probabilitas, maka dapat dianalisis bahwa kendaraan melakukan pelanggaran lalu lintas. Ekstraksi ciri atau *sequence* matriks citra hasil *preprocessing* yang terbentuk akan disimpan dalam database berupa nilai klasifikasi dan HMM bekerja menyembunyikan citra

lain yang tidak sesuai dalam database nilai dari deteksi tepi citra *training* dengan state yang ditentukan, sehingga akan mendeteksi pelanggaran yang dimaksud. [3]

Kemudian dijelaskan bahwa *Hidden Markov Model (HMM)* merupakan suatu model probabilitas yang menggambarkan hubungan statistik antara urutan observasi dan urutan state yang tidak diobservasi "*hidden*" atau disembunyikan.[4] Dalam HMM, keadaan tidak terlihat secara langsung, tetapi outputnya yang tergantung terhadap keadaan tersebut terlihat yaitu dalam kasus ini merupakan matriks dari hasil ekstraksi. Dan untuk membuat klasifikasi citra, perlu adanya tahapan *Preprocessing* terlebih dahulu yang memungkinkan me-*switch* kondisi dimana jika data citra asli ataupun *noise* tetap bisa dilakukan pengujian. Dan pada penelitian sebelumnya pendeteksian *noise* pada karakter seperti pada pengenalan pola tanda tangan HMM hanya memiliki rata-rata CER 43,35% yaitu dengan penjelasan semakin kecil CER (*Character Error Rate*) semakin baik akurasi yang dihasilkan, sementara dalam pengenalan huruf dan angka tulisan tangan mencapai akurasi 74%. Itu artinya pengenalan yang dihasilkan lebih baik dibandingkan dengan *preprocessing*.[4] Pada akhirnya untuk membantu dalam menyempurnakan kelemahan HMM tersebut maka perlu adanya metode lain yang mangkus dalam *preprocessing* citra tersebut sebelum melakukan tahapan pengklasifikasian citra.

Mengenai *Gabor Filter* merupakan metode dengan pengenalan obyek yang dapat didefinisikan sebagai proses penentuan identifikasi obyek. Metode ini dapat mendeteksi setiap titik-titik yang dihasilkan dengan akurasi tinggi[5][6] Citra sebelum masuk ke dalam basis data, citra akan lebih dahulu diolah menggunakan metode *preprocessing* dan salah satunya yaitu *Gabor FIlter*. Perhitungan konvolusi *Gabor Filter* akan menghasilkan nilai tertentu yang disebut dengan *gabor jet response* sampai mencapai *thresholding.*

Maka diperlukan kedua metode dengan fungsionalitas yang berbeda yaitu *gabor filter* untuk menyempurnakan tahapan *preprocessing* dan HMM untuk menghasilkan klasifikasi.

1.1 Gabor Filter

Sebagai filter digunakan *Gabor Filter kernel* 2D yang diperoleh dengan memodulasi gelombang *sinus* 2D pada frekuensi dan orientasi tertentu dengan *Gaussian envelope*. Persamaan dasar fungsi *Gabor Filter kernel* 2D ditunjukkan persamaan pada pada rumus berikut :

$$\begin{bmatrix} x_{\theta n} \\ y_{\theta n} \end{bmatrix} = \begin{bmatrix} \sin\theta_n & \cos\theta_n \\ -\cos\theta_n & \sin\theta_n \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \quad (1)$$

Kemudian σ_x dan σ_y adalah standar deviasi dari *Gaussian envelope* pada dimensi x dan y. λ dan θn adalah panjang gelombang dan orientasi dari gelombang sinus 2-D. Penyebaran dari *Gaussian envelope* didefinisikan dalam bentuk dari gelombang *sinus* λ. Rotasi dari $x - y$ sebesar sudut θ_n menghasilkan *Gabor filter* pada orientasi θ_n. Berikut merupakan rumus persamaan menghitung standar deviasi :

$$\begin{aligned} \sigma_x &= \sigma \\ \sigma_y &= \frac{\sigma}{\gamma} \\ \sigma^2{}_x &= \sigma^2 \\ \sigma^2{}_y &= \sigma/\gamma^2 \end{aligned} \quad (2)$$

Default parameter yang digunakan untuk menghitung *gabor filter* dapat diketahui pada dokumentasi *accord library*.[12]

Orientation θn = 0.6

Gaussian Variance Sigma = 2

Aspect Ratio Gamma = 0.3

Kernel Size fx=3

Panjang gelombang λ = 4

Adapun dalam menghitung hasil akhir nilai *gabor* dapat dilakukan pada persamaan rumus (3):

$$\begin{aligned} a &= 1 / (2 . PI . \sigma_x . \sigma_Y) \, Exp(-0.5 . (x^2{}_{\theta n}/\sigma^2{}_x + y^2{}_{\theta n}/\sigma^2{}_y)) \\ c &= Cos(2.0 . PI . fx . x_{\theta n} / 2) \\ Gabor(-1,-1) &= a . c \end{aligned} \quad (3)$$

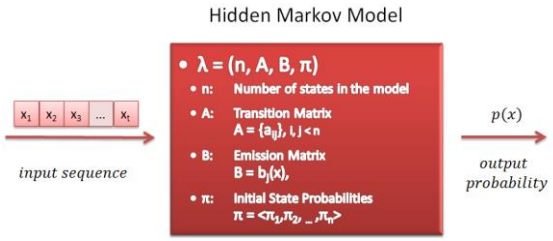
Jika semua *Gabor filter* dengan variasi panjang gelombang (λ) dan orientasi (θ_n) diterapkan pada satu titik tertentu *(x,y)*, maka didapatkan banyak respon *filter* untuk titik tersebut, misal: digunakan empat panjang gelombang (λ = 3, 5, 7, 10) dan delapan orientasi, maka akan dihasilkan tiga puluh dua respon *filter* untuk tiap titik citra yang dikonvolusikan dengan *filter* tersebut.

1.2 *Hidden Markov Model*

Model Markov Tersembunyi atau lebih dikenal sebagai *Hidden Markov Model* (HMM) adalah sebuah model statistik dari sebuah sistem yang diasumsikan sebuah Proses Markov dengan parameter yang tak diketahui, dan tantangannya

adalah menentukan parameter-parameter tersembunyi (*state*) dari parameter-parameter yang dapat diamati (*observer*). Parameter-parameter yang ditentukan kemudian dapat digunakan untuk analisis yang lebih jauh, misalnya untuk aplikasi pengenalan pola (*Pattern Recognition*). Sebuah HMM dapat dianggap sebagai sebuah *Bayesian Network* dinamis yang paling sederhana.

Pada model Markov umum (*Vanilla/Visible Markov Model*), *state*-nya langsung dapat diamati, oleh karena itu probabilitas transisi keadaan (*state)* menjadi satu-satunya parameter. Di dalam Model Markov yang tersembunyi, *state*-nya tidak dapat diamati secara langsung, akan tetapi yang dapat diamati adalah variabel-variabel yang terpengaruh oleh *state*. Setiap *state* memiliki distribusi probabilitas atas token-token output yang mungkin muncul. Oleh karena itu rangkaian token yang dihasilkan oleh HMM memberikan sebagian informasi tentang sekuens state-state.[4]



Gambar 1 HMM Model

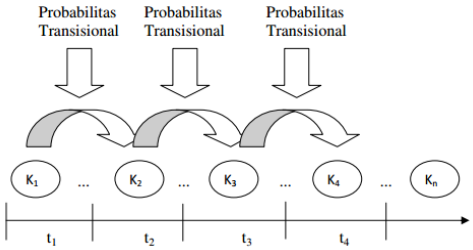
1.2.1 *Markov Chain*

Rantai Markov *(Markov Chains)* adalah merupakan suatu teknik matematika yang biasa digunakan untuk melakukan pemodelan (modeling) berbagai macam sistem dan proses bisnis. Teknik ini dapat digunakan untuk memperkirakan perubahan-perubahan di waktu yang akan datang dalam variabel-variabel dinamis atas dasar perubahan-perubahan dari variabel-variabel dinamis tersebut di waktu yang lalau. Teknik ini dapat digunakan juga untuk menganalisis kejadian-kejadian di waktu-waktu mendatang secara matematis.

Model Rantai Markov *(Markov Chains)* ditemukan oleh seorang ahli Rusia yang bernama Andrey Markov pada tahun 1906, yaitu: "*Untuk setiap waktu t, ketika kejadian adalah* K_t *dan seluruh kejadian sebelumnya adalah* $K_{t(j)}$... , $K_{t(j-n)}$ *yang terjadi dari proses yang diketahui, probabilitas seluruh kejadian yang akan datang Kt(j) hanya bergantung pada kejadian* $K_{t(j-1)}$*dan tidak bergantung pada kejadian-kejadian sebelumnya yaitu* $K_{t(j-2)}$, $K_{t(j-3)}$,..., $K_{t(j-n)}$*.*".

Rantai Markov *(Markov Chains)* apabia diberikan inputan keadaan dari keadaan saat ini, keadaan akan datang dapat diprediksi dan dapat di lepas dari keadaan di masa lampau. Artinya, deskripsi kondisi saat ini menangkap semua informasi yang mempengaruhi evolusi dari suatu sistem dimasa depan. Dengan kata lain, kondisi masa depan dituju dengan menggunakan prbabilitas bukan dengan determinitas.

Gambaran mengenai rantai Markov ini kemudian gerakan-gerakan dari beberapa variabel di masa yang akan datang bisa diprediksi berdasarkan gerakan-gerakan variabel tersebut pada masa lalu. K_{t4} dipengaruhi oleh kejadian K_{t3}, K_{t3} dipengaruhi oleh kejadian K_{t2} dan demikian seterusnya dimana perubahan ini terjadi karena peranan probabilitas transisi (transition probability). Kejadian K_{t2} misalnya, tidak akan mempengaruhi kejadian K_{t4}.



Gambar 2 Markov Chains

Rantai Markov *(Markov Chains)* bermanfaat untuk menghitung probabilitas suatu kejadian teramati yang secra umum dapat dirumuskan sebagai berikut:

$$P(\sigma_i) = P(\sigma_t|\sigma_{t-1}, \sigma_{t-2}, \sigma_{t-3}, \dots, \sigma_{t-n}) \quad (4)$$

σ_t adalah kondisi saat ini, dan σ_t adalah kondisi pada waktu tertentu yang berhubungan dengan σ_t. Sedangkan σ_{t-i} adalah kondisi sebelum σ_t. Kemudian dapat diasumsikan bahwa sebelah kanan persamaan bersifat invariant, yaitu, dihipotesiskan dalam keseluruahan sistem, transisi diantara keadaan tertentu tetap sama dalam hubungan probabilistiknya. Berdasarkan asumsi tersebut, dapat terbentuknya suatu stet keadaan probabilitas transisi diantara dua keadaan K_i dan t_j :

$$Probabilitas\ Transisi = P(\sigma_t = K_t|\sigma_{t-1} = t_j), 1 \le i, j \le N \quad (5)$$

1.2.2 Parameter HMM

HMM mempunyai parameter-parameter distribusi sebagai berikut :

a. Probabilitas Transisi (A)

Parameter A merupakan parameter dengan ukuran MxM dengan M adalah jumlah state yang ada, parameter transisi dapat dituliskan dalam bentuk matriks seperti berikut:

$$A_{ij} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1n} \\ A_{21} & A_{22} & A_{23} & \cdots & A_{2n} \\ A_{31} & A_{32} & A_{33} & \cdots & A_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & A_{n3} & \cdots & A_{nn} \end{bmatrix}$$

$$A = \{a_{ij}\}, a_{ij} = P_r(x_{t+1} = q_j | x_t = q_i), 1 \leq j, i \leq N \quad (6)$$

b. Probabilitas observasi (B)

Parameter B merupakan probabilitas observasi atau probabilitas state merupakan kemunculan suatu state deretan seluruh state yang ada. Parameter B didalam HMM dituliskan dalam bentuk matriks kolom dengan Mx1, dimana M merupakan jumlah seluruh state yang ada. Parameter B dapat dituliskan dalam bentuk matriks seperti berikut :

$$B = \begin{bmatrix} b1 \\ b2 \\ b3 \\ \vdots \\ bn \end{bmatrix}$$

$$B = \{b_i\}, b_i(k) = P_r(O_t = V_k | x_t = q_i\ t) \quad (7)$$

c. Distribusi keadaan awal (π)

Parameter phi (π), disebut sebagai parameter awal, merupakan probabilitas kemunculan suatu state di awal. Sama halnya dengan parameter B, parameter π juga dituliskan dalam bentuk matriks kolom dengan ukuran Mx1, dimana M adalah jumlah state-nya, parameter π dapat dituliskan dalam bentuk seperti berikut :

$$\pi = \begin{bmatrix} \pi 1 \\ \pi 2 \\ \pi 3 \\ \vdots \\ \pi n \end{bmatrix}$$

$$\pi = \{\pi_i\}, \pi_i = P_r(x_0 = q_i) \quad (8)$$

Sedangkan parameter tertentu HMM ada dua yaitu N dan M:

a. N, jumlah keadaan model. Dinotasikan himpunan terbatas untuk keadaan yang mungkin adalah $Q = \{q_1, \dots, q_N\}$ (9)

b. M, jumlah dari simbol observasi/keadaan, ukuran huruf diskret. Simbol observasi berhubungan dengan keluaran fisik dari sistem yang dimodelkan. Dinotasikan himpunan terbatas untuk observasi yang mungkin adalah

$$V = \{V_1, \dots, V_M\} \quad (10)$$

Parameter-parameter HMM ditaksir berdasarkan kriteria *maximum likelihood* (ML) dan algoritma *Baum-Welch* (EM = Expectation Modification).

1.2.3 Permasalahan HMM

Ada 3 masalah dasar HMM yang harus dipecahkan untuk model yang diterapkan di dunia nyata, yaitu:

a. Menghitung $P = (O|\lambda)$ bila diberikan urutan observasi $O = O_1, O_2, \dots, O_T$ dan λ = (A, B, π). (10)

Solusi:

Cara umum yang biasa digunakan adalah dengan memeriksa setiap kemungkinan urutan *N state* sepanjang *T* (banyaknya observasi). Hal ini tidak mungkin dilakukan karena perhitungannya kurang efisien. Ada prosedur lain yang lebih sederhana dan efisien adalah menggunakan prosedur *forward*.

- Prosedur Maju (*forward procedure*)

Asumsi variabel *forward* $\alpha_t(i)$ pada waktu ke-*t* dan *state* ke-*i* didefinisikan dengan. $\alpha_t(i) = P(O_1,O_2,\dots,O_T, q_t=i \mid \lambda)$. Fungsi peluang *forward* dapat diselesaikan untuk *N state* dan *T* simbol observasi secara induktif dengan langkah sebagai berikut:

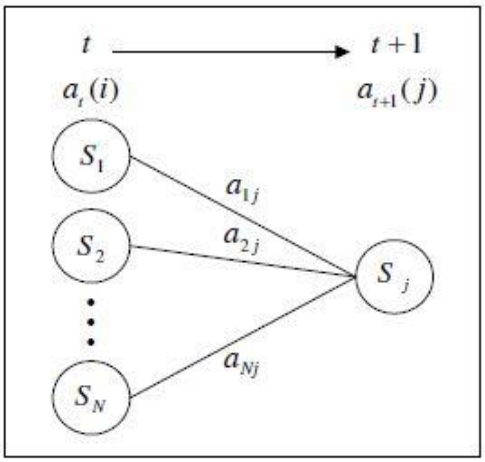
○ Inisialisasi, $\alpha_t(i) = \pi_i b_i(O_1)$, $1 \leq i \leq N$ (11)

○ Induksi, $\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)\alpha_{ij}\right] b_j(Oi+_1)$, $1 \leq t \leq$ T-1; 1 ≤ j ≤N (12)

○ Terminasi, $P(O|\lambda) = \sum_{i=1}^{N} \alpha\ T(i)$ (13)

Perhitungan peluang *forward* berdasarkan pola diagram *trellis*. Terdapat *N* titik tiap *time slot* pada pola, semua kemungkinan deretan *state* digabung sebanyak *N* titik tanpa memperdulikan panjang deretan observasi. Pada saat t=1, dihitung nilai $O_1(i)$, $1 \leq i \leq N$. Pada waktu t = 2,3,...,T hanya diperlukan perhitungan nilai $\alpha_t(j)$ dimana $1 \leq j \leq N$.

Tiap perhitungan membutuhkan nilai sebelumnya sebanyak *N* dari $\alpha_{t-1}(i)$ karena tiap *N* titik hanya dapat dihubungkan dengan *N* titik pada *time slot* sebelumnya.

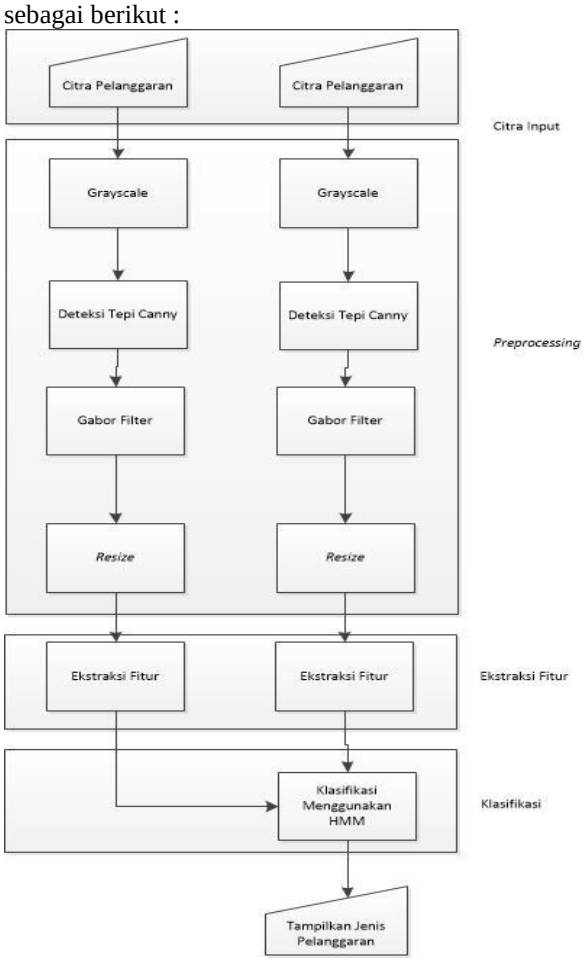


Gambar 3 Perhitungan Deret State

2. ISI PENELITIAN

2.1 Gambaran Umum Aplikasi

Tujuan utama dari tugas akhir ini adalah membuat aplikasi yang dapat melakukan pengenalan pelanggaran lalu lintas dilarang "Perboden" dan dilarang "balik arah" pada kendaraan. Proses diagram alir pemodelan aplikasi pada gambar 3.1 sebagai berikut :



Gambar 4 Flowchart Pemodelan Aplikasi

Pada alur pemodelan aplikasi dijelaskan terdapat 4 (empat) tahapan yaitu citra *input*, *preprocessing*, ekstraksi fitur dan klasifikasi. Dari masing-masing tahapan tersebut dapat dianalisis prosesnya lebih lanjut mulai dari data masukan berupa citra pelanggaran, kemudian masuk ke dalam *preprocessing* yaitu mengubah citra ke *grayscale*, menentukan deteksi tepi *canny*, *filtering* menggunakan *gabor filter* dan me-*resize* hasil dari gabor ke dalam 128x128. Setelah itu mencari ekstraksi ciri untuk dimasukan ke dalam klasifikasi menggunakan HMM.

2.2 Data Masukan

Data masukan yang akan diambil pada aplikasi ini merupakan citra yang diambil saat terjadi pelanggaran dilarang "Perboden" dan dilarang "balik arah". Sehingga jelas citra merupakan sebuah pelanggaran, dan dalam penelitian ini menentukan identifikasi kategori jenis pelanggaran yang terdapat pada citra.

Setiap proses pada proses pengolahan citra menggunakan nilai pixel dari citra untuk perhitungannya. Setiap pixel pada citra terdiri dari 3 unsur warna dasar yaitu Red (R), Green (G) dan Blue (B), yang sering disebut RGB, dan setiap unsur memiliki nilai antara 0-255 yang menjadi kode kedalaman warna dari masing-masing unsur. Sebagai contoh perhitungan pada proses-proses pengolahan citra, digunakan gambar "pelanggaran kendaraan" dapat dilakukan dengan ukuran inputan yang berbeda-beda. Citra pelanggaran dapat dilihat seperti pada gambar sebagai berikut :



Gambar 5 Gambar Pelanggaran Dilarang "balik arah"



Gambar 6 Pelanggaran Dilarang "Perboden"

Dari citra masukan yang ukurannya berbeda-beda karena dimensi setiap pengambilan citra dapat dilakukan dengan berbagai *tools* yang notabennya tidak memiliki ukuran pixel yang sama, maka perlu adanya *resize* pada *preprocessing* untuk menyamakan ukuran dimensi pixel citra. Pada penelitian ini resize akan dilakukan menjadi 128x128 pixel.

2.3 Pengujian Aplikasi

Pengujian aplikasi merupakan tahapan yang memiliki tujuan untuk menemukan kekurangan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan dan pengujian perangkat lunak ini menggunakan pengujian *black box*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak, dan juga pengujian terhadap performansi yang dilakukan untuk menghasil nilai pengujian terhadap klasifikasi.

Pengujian *black box* digunakan untuk menentukan pengujian yang dilakukan ini terbagi menjadi beberapa scenario. Seperti yang terlihat dibawah ini.

2.3.1 Skenario Pengujian Data

Berikut ini adalah scenario pengujian seperti terlihat pada tabel dibawah ini :

Skenario Pengujian Data

Kelas Uji	Butir Uji	Jenis Pengujian
Gambar	Menguji gambar masukan	*Blackbox*
Pengelolaan data *training*	Mengelola data dari citra latih	*Blackbox*

2.3.2 Skenario Fungsional

Skenario fungsional menjelaskan tentang pengujian terhadap fungsional aplikasi yang digunakan. Berikut merupakan hasil pengujian pada aplikasi tersebut :

Tabel Skenario Fungsional

Fungsional	Aktifitas yang diharapkan	Pengujian	Kesimpulan
Grayscale	Mengubah citra asli menjadi *grayscale*		[√] diterima [ ] tidak diterima
Deteksi Tepi *Canny*	Mengubah citra *grayscale* menjadi citra deteksi tepi *canny*.		[√] diterima [ ] tidak diterima
Gabor Filter	Mengubah citra deteksi tepi menjadi citra hasil *gabor filter*.		[√] diterima [ ] tidak diterima
Resize	Mengubah ukuran citra hasil *gabor filter* menjadi 128x128 pixel.		[√] diterima [ ] tidak diterima

2.3.3 Kasus dan Hasil Pengujian

Berdasarkan rencana pengujian yang telah tersusun, maka dapat dilakukan pengujian seperti yang dijelaskan pada tabel di bawah ini :

Kasus dan Hasil Uji

Data masukan	Aktifitas yang diharapkan	Pengamatan	Kesimpulan

Gambar	Terdeteksi "Perboden"	Aplikasi menampilkan jenis pelanggaran "Perboden" dari gambar masukan.	[√] diterima [ ] tidak diterima
	Terdeteksi "balik arah"	Aplikasi menampilkan jenis pelanggaran "balik arah" dari gambar masukan.	[√] diterima [ ] tidak diterima

2.4 Pengujian Performansi

Pengujian performansi digunakan untuk mengetahui akurasi yang dilakukan menggunakan metode *k-fold cross validation*, dengan nilai k sebanyak 10 *fold*. Bertujuan untuk menguji stabilitas akurasi jika diuji dengan data latih dan data uji yang berbeda. Penggunaan 10 *fold* ini dianjurkan karena merupakan jumlah *fold* terbaik untuk uji validitas. Pengujian dilakukan pada metode *Hidden Markov Model* (HMM). Untuk pegujian pada metode *Hidden Markov Model* (HMM) data yang digunakan sebanyak 82 data yang dibagi menjadi 2 *subset* dengan kategori pelanggaran "perboden" dan "balik arah", dan yang digunakan dalam pengujian performansi menggunakan iterasi sebanyak 50 kali. Skenario uji performasi akurasi dengan metode *fold cross validation* sebagai berikut :

a. Hasil Percobaan *Fold* 1

Assigned Class	Likelihood	Match
S	0,999900705020977	☐
S	0,99998621018447	☐
S	0,999999008132514	☐
S	0,999998340107268	☐
S	0,999956995233563	☐
S	0,9999981826972	☐
S	0,8294743178533	☐
P	0,948383153256537	☑
S	0,997309111024863	☐
S	0,999881150386368	☐
S	0,999991754716689	☐
S	0,999982706583785	☐
S	0,930941847374728	☐
S	0,998141541241458	☐
S	0,999999947473385	☐
S	0,999999996030738	☐
S	0,999999997396799	☐
S	0,999854151366954	☐
S	0,761191367573416	☐
P	0,996033790715403	☑
S	0,978296490589362	☐
P	0,727764947622747	☑

Hasil test yang tidak benar dalam hasil klasifikasi pada subset 1 yaitu 20. Sehingga nilai akurasi dalam percobaan *fold* 1 yaitu $\frac{2}{22} * 100\% = 9.09\%$.

b. Hasil Percobaan *Fold* 2

Assigned Class	Likelihood	Match
P	0,837644551917209	☑
P	0,830910270031823	☑
P	0,889592258925908	☑
P	0,999999999971863	☑
S	0,999097644025353	☐
P	0,813019440684304	☑
P	0,99994874586263 9	☑
P	0,99999999999261	☑
P	0,999999999988916	☑
P	0,999710927077499	☑
S	0,989999541174043	☐
P	0,999999910330307	☑
P	0,999999999367219	☑
P	0,987784454097558	☑
P	0,999997716911756	☑
P	0,999999999820318	☑
P	0,999999125993368	☑
P	0,99999999550954	☑
P	0,999911435220222	☑
P	0,999998478090411	☑
P	0,999999427704261	☑
P	0,999999684140043	☑

Hasil test yang tidak benar dalam hasil klasifikasi pada subset 2 yaitu 2. Sehingga nilai akurasi dalam percobaan *fold* 2 yaitu $\frac{20}{22} * 100\% = 90.9\ \%$.

c. Hasil Percobaan *Fold* 3

Assigned Class	Likelihood	Match
P	0,999999994247446	☑
P	0,999980847765273	☑
P	0,999999999986471	☑
P	0,99999999999784	☑
P	0,999995628832555	☑
P	0,999999996951999	☑
P	0,999999997486498	☑
P	0,999999962165646	☑
P	0,999999986028172	☑
P	0,999999999999943	☑
P	0,999999999995055	☑
P	0,999999999925308	☑
P	1	☑
P	1	☑
P	0,999999999995964	☑
P	1	☑
S	0,999999999999091	☑
S	0,99963751146435	☑
S	0,999992340193108	☑
S	0,99999866379481	☑
S	0,974570531262582	☑
S	0,999961914073503	☑

Hasil test yang tidak benar dalam hasil klasifikasi pada subset 3 yaitu 0. Sehingga nilai akurasi dalam percobaan *fold* 1 yaitu $\frac{22}{22} * 100\% = 100\ \%$.

d. Hasil Percobaan *Fold* 4

S	0,961108850710227	☑
S	0,999992649967748	☑
S	0,999679218928233	☑
S	0,96752973713067	☑
S	0,999987077591462	☑
S	0,999999137873177	☑
S	0,999402861017684	☑
S	0,999999971755756	☑
S	0,999998398119861	☑
S	0,886950927309348	☑
P	0,620852593272065	☐
S	0,57147815804708	☑
S	0,999148234832538	☑
S	0,999475568992577	☑
P	0,957059516996981	☐
P	0,712536826097485	☐

Hasil test yang tidak benar dalam hasil klasifikasi pada subset 4 yaitu 3. Sehingga nilai akurasi dalam percobaan *fold* 1 yaitu $\frac{13}{16} * 100\% = 81.25\%$.

Hasil Perhitungan Akurasi

Fold	Akurasi
Fold 1	50%
Fold 2	30%
Fold 3	40%
Fold 4	30%
Rata – rata akurasi	$\frac{9.09\% + 90.9\% + 100\% + 81.25\%}{4} = 70.31\%$

Dari pengujian *K-Fold Cross Validation* menggunakan 82 data didapatkan nilai akurasi rata-rata 70,31% benar dalam klasifikasi dan 29.69% salah. Faktor penyebab ketidak *valid*an dalam mengklasifikasikan menggunakan metode *Hidden Markov Model* (HMM) karena sangat bergantung pada ciri yang digunakan sebagai data latih.

3. KESIMPULAN

Kesimpulan dari penelitian yang berjudul : "Implementasi metode *Hidden Markov Model dan Gabor Filter Untuk Mendeteksi Pelanggaran Lalu Lintas Kendaraan*" adalah sebagai berikut :

1. Penelitian pada pengolahan citra pelanggaran kendaraan dapat digunakan untuk mendeteksi citra jenis pelanggaran kendaraan yaitu dilarang "Perboden" dan "balik arah".
2. Metode *Hidden Markov Model* dapat diterapkan dalam proses klasifikasi menentukan jenis pelanggaran "balik arah" dan "Perboden" yang dilakukan kendaraan pada pengolahan citra pelanggaran kendaraan dengan hasil akurasi rata-rata 70,31% benar dalam klasifikasi.

4. DAFTAR PUSTAKA

[1] Website Resmi Korps Lalu Lintas POLRI. 2015. http://lantas.polri.go.id.

[2] Website Resmi Pemerintahan Jawa Barat. 2015. http://www.jabarprov.go.id.

[3] D Sudian, Arman dan P Priambodo. "Aplikasi Pengenalan Wajah (*Face Recognition*) Menggunakan Metode Hidden Markov Model (HMM)". Teknik Elektro, UI.

[4] E Yuwitaning, B Hidayat dan N Andini. "Implementasi Metode Hidden Markov Model Untuk Deteksi Tulisan Tangan". Teknik Elektro, Universitas Telkom.

[5] H Kekre and V Bharadi. 2010. "*Gabor Filter Based Feature Vector for Dynamic Signature Recognition.*".

[6] D Murugan, S Arumugam, K Rajalakshmi dan Manish. 2010. "*Performance Evaluation of Face Recognition Using Gabor Filter, Log Gabor Filter and Disctere Wavelet Transform*".

[7] Sepritahara. 2012. "Aplikasi Pengenalan Wajah (*Face Recognition*) Menggunakan Metode Hidden Markov Model (HMM)". Skripsi. Jakarta: Fakultas Teknik, Universitas Indonesia.

[8] A Margono, I Gunawan dan R Lim. 2004. "Pelacakan dan Pengenalan Wajah Menggunakan Metode Embedded Hidden Markov Models".

[9] A Agung, Fazmah A Yulianto dan W Maharani. 2011. "Pengenalan Wajah Menggunakan Psedo-2D Hidden Markov Model".

[10] P Dymarski. 2011."*Hidden Markov Model, Theory and Aplications"*. India: InTech.

[11] A Khandual, G Baciu *and* N Rout. 2013. "*Colorimetric Preprocessing o*

f Digital Colour Image".

[12] *Gabor Filter Imaging Filter*. 2015. http://accord-framework.net/docs/html/T_Accord_Imaging_Filters_GaborFilter.htm.

[13] *Canny Edge Detector Class.* 2015. http://www.aforgenet.com/framework/docs/html/e08cae30-7a37-db9f-cede-05cf6521343f.htm

[14] P Mishra, R Chatterjee *and* V Mahapatra. 2010. "*Texture Gabor Filter Using Gabor Filter and Wavelets*".