



UNIVERSITAS INDONESIA



**ANALISIS PENGGUNAAN *SCALE INVARIANT FEATURE*  
*TRANSFORM* SEBAGAI METODE EKSTRAKSI FITUR PADA  
PENGENALAN JENIS KENDARAAN**

**SKRIPSI**

**ROMMY RAKHMAN ARIEF**

**0606074325**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
DESEMBER 2010**



UNIVERSITAS INDONESIA

**ANALISIS PENGGUNAAN *SCALE INVARIANT FEATURE*  
*TRANSFORM* SEBAGAI METODE EKSTRAKSI FITUR PADA  
PENGENALAN JENIS KENDARAAN**

**SKRIPSI**

Diajukan sebagai salah satu syarat untuk memperoleh gelar  
Sarjana Teknik

**ROMMY RAKHMAN ARIEF**

**0606074325**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
DESEMBER 2010**

## HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar**

**Nama : Rommy Rakhman Arief**

**NPM : 0606074325**

**Tanda Tangan :**

**Tanggal : 17 Desember 2010**

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama : Rommy Rakhman Arief  
NPM : 0606074325  
Program Studi : Teknik Elektro  
Judul Skripsi : ANALISIS PENGGUNAAN *SCALE INVARIANT*  
*FEATURE TRANSFORM* SEBAGAI METODE  
EKSTRAKSI FITUR PADA PENGENALAN JENIS  
KENDARAAN

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia

### DEWAN PENGUJI

Pembimbing : Prof. Dr. Ir. Dadang Gunawan, M.Eng.  (.....)

Penguji : Dr. Abdul Halim, M.Eng.  (.....)

Penguji : Filbert Hilman Juwono, S.T., M.T.  (.....)

Ditetapkan di : Depok

Tanggal : 30 Desember 2010

## UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Program Studi Teknik Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Ir. Dadang Gunawan, M.Eng, selaku pembimbing yang membantu memberikan arahan dan nasihat sehingga saya dapat menyelesaikan skripsi ini;
2. Para peneliti sebelum ini yang telah memberikan sumber bacaan dan referensi bagi saya;
3. Teman – teman Departemen Teknik Elektro khususnya angkatan 2006 yang selalu memberikan bantuan dan dukungan kepada saya;
4. Kedua orang tua dan keluarga saya yang selalu memberikan dukungannya kepada saya;
5. Dan seluruh Sivitas Akademik Departemen Teknik Elektro Universitas Indonesia yang tidak dapat saya sebutkan satu persatu.

Akhir kata, semoga Tuhan YME berkenan membalas kebaikan semua pihak yang telah membantu. Semoga skripsi ini bermanfaat bagi perkembangan ilmu pengetahuan.

Depok, Desember 2010

Rommy Rakhman Arief

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI**  
**TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

---

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan dibawah ini:

Nama : Rommy Rakhman Arief  
NPM : 0606074325  
Program Studi : Teknik Elektro  
Departemen : Teknik Elektro  
Fakultas : Teknik  
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul:

**ANALISIS PENGGUNAAN *SCALE INVARIANT FEATURE TRANSFORM***  
**SEBAGAI METODE EKSTRAKSI FITUR PADA PENGENALAN JENIS**  
**KENDARAAN**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya

Dibuat di : Depok

Pada tanggal : 17 Desember 2010

Yang menyatakan

(.....)

## ABSTRAK

Nama : Rommy Rakhman Arief  
Program Studi : Teknik Elektro  
Judul : Analisis Penggunaan *Scale Invariant Feature Transform*  
Sebagai Metode Ekstraksi Fitur Pada Pengenalan Jenis  
Kendaraan

Ekstraksi fitur merupakan tahap penting dalam pengenalan objek. Fitur yang dihasilkan dalam tahapan ini akan sangat menentukan ke-akuratan suatu objek ketika dikenali. *Scale Invariant Feature Transform (SIFT)* merupakan suatu metode yang memberikan fitur-fitur khas dan invarian terhadap berbagai jenis perubahan pada citra. Pada skripsi ini akan dibahas tentang *SIFT* sebagai metode ekstraksi fitur pada pengenalan objek. Selain itu juga akan dilakukan perancangan simulasi untuk menguji kinerja dan performansi *SIFT* jika digunakan pada pengenalan jenis kendaraan. Performa dari *SIFT* ini akan diukur dari grafik *recall vs. 1-precision* yang dihasilkan dan lamanya waktu proses.

Kata kunci: *SIFT*, *keypoint*, kendaraan.

## ABSTRACT

Name : Rommy Rakhman Arief  
Study Program : Electrical Engineering  
Title : Analysis of Scale Invariant Feature Transform Usage  
As Feature Extraction Method For Vehicle Type  
Recognition

Feature Extraction is an important stage in object recognition. The features generated in this stage will determine the accuracy of object recognition. Scale Invariant Feature Transform (SIFT) is a method which generate a unique features and invariant against various changes in the image. This thesis will discuss about SIFT as a feature extraction method in object recognition. Also a simulation will be design to test the performance of SIFT if used in a vehicle type recognition. The performance of SIFT can be measured from the graph of recall vs. 1-precision and the processing time.

Keywords: SIFT, keypoint, vehicle.

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PERNYATAAN ORISINALITAS .....	ii
HALAMAN PENGESAHAN .....	iii
UCAPAN TERIMA KASIH .....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI .....	v
ABSTRAK .....	vi
<i>ABSTRACT</i> .....	vii
DAFTAR ISI .....	viii
DAFTAR GAMBAR .....	x
DAFTAR TABEL .....	xi
DAFTAR SINGKATAN .....	xii
DAFTAR LAMPIRAN .....	xiii
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah .....	3
1.3 Tujuan Penulisan.....	3
1.4 Batasan Masalah .....	4
1.5 Sistematika Penulisan .....	4
<b>BAB 2 DASAR TEORI.....</b>	<b>5</b>
2.1 Pengenalan Objek Pada Komputer .....	5
2.1.1 <i>Pre-processing</i> .....	5
2.1.1.1 Histogram Equalization .....	5
2.1.1.2 Spatial Low Pass Filter .....	6
2.1.1.3 Spatial High Pass Filter.....	7
2.1.1.4 Median Filter .....	7
2.1.2 Segmentasi Citra .....	8
2.1.3 Ekstraksi Fitur .....	9
2.2 <i>Scale Invariant Feature Transform</i> .....	9
2.2.1 Pencarian Nilai Ekstrim Pada Skala Ruang.....	10
2.2.2 Penentuan <i>Keypoint</i> .....	12
2.2.3 Penentuan Orientasi.....	13
2.2.4 Deskriptor Pada <i>Keypoint</i> .....	14
<b>BAB 3 PERANCANGAN SIMULASI SISTEM PENGENALAN JENIS KENDARAAN DENGAN METODE EKSTRAKSI FITUR <i>SIFT</i>....</b>	<b>16</b>
3.1 Perangkat Simulasi.....	16
3.2 Basis Data Citra .....	16
3.3 Diagram Alir Simulasi .....	17
3.3.1 Citra Masukan .....	17
3.3.2 Tahap <i>Pre-processing</i> .....	18
3.3.3 Deteksi <i>Region of Interest</i> .....	18
3.3.4 Pencocokan <i>Keypoint</i> .....	18
3.4 Parameter Analisis .....	18

<b>BAB 4 HASIL DAN ANALISIS SIMULASI.....</b>	<b>20</b>
4.1 Data Simulasi.....	20
4.1.1 Basis Data Citra .....	20
4.1.2 Citra Tes Yang Digunakan .....	20
4.2 Hasil Dan Analisis Simulasi.....	21
4.2.1 Proses Segmentasi Citra .....	21
4.2.2 Proses Deteksi Keypoint Menggunakan <i>SIFT</i> .....	22
4.2.3 Akurasi Pencocokan <i>Keypoint</i> .....	23
4.2.4 Akurasi Pengenalan Jenis Kendaraan .....	26
4.2.5 Waktu Proses .....	27
4.2.6 Nilai <i>Threshold</i> Optimum.....	27
4.3 Analisis Penggunaan <i>Scale Invariant Feature Transform</i> Pada Pengenalan Jenis Kendaraan .....	28
 <b>BAB 5 PENUTUP .....</b>	 <b>29</b>
5.1 Kesimpulan.....	29
5.2 Saran.....	30
 <b>DAFTAR REFERENSI .....</b>	 <b>31</b>

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Citra dan histogramnya .....	6
<b>Gambar 2.2</b> Hasil dari proses <i>histogram equalization</i> .....	6
<b>Gambar 2.3</b> Citra asli dan citra hasil low-pass filter .....	7
<b>Gambar 2.4</b> Citra asli dan citra hasil high-pass filter .....	7
<b>Gambar 2.5</b> Citra asli dan citra hasil median filter dengan wilayah 3x3 .....	8
<b>Gambar 2.6</b> Segmentasi citra menggunakan metode berbasis grafik .....	9
<b>Gambar 2.7</b> Diagram <i>Difference of Gaussians</i> .....	11
<b>Gambar 2.8</b> Ilustrasi pencarian maksimum atau minum lokal dari citra DoG ...	12
<b>Gambar 2.9</b> Deskriptor dari perhitungan besar gradien dan orientasi serta gambar lingkaran Gaussian dan gambar deksriptor <i>keypoint</i> .....	14
<b>Gambar 2.10</b> <i>Keypoint</i> yang diperoleh dari suatu citra menggunakan <i>SIFT</i> .....	15
<b>Gambar 3.1</b> Contoh citra <i>training</i> yang akan digunakan .....	16
<b>Gambar 3.2</b> Diagram alir proses simulasi .....	17
<b>Gambar 3.3</b> Deteksi <i>RoI</i> pada citra tes .....	18
<b>Gambar 4.1</b> Citra yang digunakan sebagai basis data .....	20
<b>Gambar 4.2</b> Contoh citra yang digunakan sebagai masukan .....	21
<b>Gambar 4.3</b> Citra masukan asli dan setelah dideteksi <i>RoI</i> .....	21
<b>Gambar 4.4</b> Kesalahan <i>matching keypoint</i> pada bagian citra yang tidak diperlukan .....	22
<b>Gambar 4.5</b> Grafik <i>recall vs. 1-precision</i> dari masing-masing jenis kendaraan..	25

## DAFTAR TABEL

<b>Tabel 4.1</b> Jumlah <i>keypoint</i> yang terdeteksi pada citra basis data.....	22
<b>Tabel 4.2</b> Rata-rata jumlah <i>keypoint</i> yang terdeteksi pada citra tes.....	23
<b>Tabel 4.3</b> Nilai recall dan 1-precision pada masing-masing jenis kendaraan untuk setiap nilai T .....	24
<b>Tabel 4.4</b> Jumlah jenis kendaraan yang dapat dikenali dan rata-rata waktu proses .....	26

## DAFTAR SINGKATAN

SIFT	<i>Scale Invariant Feature Transform</i>
RoI	<i>Region of Interest</i>
DoG	<i>Difference of Gaussian</i>



## DAFTAR LAMPIRAN

Lampiran 1. Citra Tes Yang Digunakan .....	32
Lampiran 2. Data Hasil Simulasi.....	37
Lampiran 3. Listing Kode Program .....	51



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Seiring dengan perkembangan teknologi saat ini, ilmu yang berkaitan dengan pemrosesan citra mulai banyak diaplikasikan untuk memudahkan kehidupan manusia. Pemrosesan citra bertujuan untuk mengembangkan algoritma yang menggunakan citra sebagai inputan dan menghasilkan suatu interpretasi simbolik yang mendeskripsikan objek yang ada, dalam bentuk apa dan hubungan spasial tiga dimensi antar objek yang ada [1]. Beberapa cabang ilmu yang berkaitan dengan pemrosesan citra ialah *pattern recognition* (pengenalan pola), *object recognition* (pengenalan objek) dan *shape recognition* (pengenalan bentuk). Beberapa contoh aplikasi yang sudah banyak dijumpai ialah sistem pengenalan wajah, OCR (*Optical Character Recognition*), pengenalan sidik jari, pengenalan buah, pengenalan nomor polisi pada kendaraan, dan lain-lain.

Pada saat ini penggunaan kendaraan bermotor menunjukkan perkembangan yang sangat tinggi. Disamping itu kebutuhan akan otomatisasi pada otentikasi dirasa semakin perlu. Selain untuk alasan keamanan tapi juga untuk memudahkan pekerjaan dan menghemat SDM yang digunakan. Pada pintu tol contohnya, sistem pengenalan kendaraan ini dapat digunakan untuk menentukan secara cepat mobil yang akan masuk termasuk kedalam golongan tertentu. Selain itu juga, sistem pengenalan pada kendaraan dapat digunakan pada tempat parkir yang dikhususkan untuk jenis kendaraan tertentu saja. Sejauh ini penelitian tentang sistem pengenalan kendaraan sudah banyak dilakukan baik di Indonesia maupun di mancanegara. Beberapa contoh penelitian yang telah dilakukan yang berkaitan dengan pengenalan kendaraan ialah:

- a. Pada tahun 2009, Desy Maulidiyanti Amin melakukan penelitian berjudul “Pengenalan Kendaraan Berbasis Pemodelan Obyek pada Video” sebagai skripsi di Institut Teknologi Sepuluh Nopember.

- b. Fransisco Jimmy Hartono melakukan penelitian berjudul “Perancangan dan Pembuatan Aplikasi Pengenalan Jenis Mobil Dengan Menggunakan Gabor Filter dan Jaringan Syaraf Tiruan” sebagai skripsi di Universitas Kristen Petra.
- c. Pada tahun 2001, Thiang, Andre Teguh Guntono dan Resmana Lim mempublikasikan jurnal penelitian berjudul “*Type of Vehicle Model Recognition Using Template Matching Method*” pada *Proceeding of the International Conference on Electrical, Electronics, Communication, and Information CECI' 2001*.

Pengenalan objek berarti memberikan klasifikasi terhadap objek, benda atau bentuk tertentu yang terdapat pada suatu citra atau gambar [1]. Pengenalan objek melibatkan tahap deteksi dan pengenalan. Termasuk didalam tahap tersebut yaitu *preprocessing*, proses segmentasi objek dan ekstraksi fitur. Ekstraksi fitur merupakan tahap yang penting dalam pengenalan objek. Tahap ini akan mengambil fitur-fitur penting yang mencirikan objek yang ingin dikenali pada suatu citra kemudian membandingkannya dengan basis data fitur dari objek-objek yang ada. Namun banyak dari metode ekstraksi fitur yang sangat rentan terhadap perbedaan antara objek yang ingin dikenali dengan objek yang terdapat pada basis data, seperti perbedaan sudut pandang, adanya translasi atau rotasi pada objek, perbedaan besar ukuran, perbedaan intensitas iluminasi cahaya, dan lain-lain.. Pengembangan dan penelitian akan metode ekstraksi fitur ini terus dilakukan demi mendapatkan suatu metode ekstraksi fitur yang dapat digunakan untuk skala yang lebih luas dan lebih dinamis.

Banyak penelitian dilakukan untuk mendapatkan metode ekstraksi fitur yang sesuai dengan harapan. Kemudian pada tahun 1999, David Lowe [3] pertama kali memperkenalkan suatu metode ekstraksi fitur yang disebut *Scale Invariant Feature Transform (SIFT)*. Pada [3] dan [4] dikatakan bahwa metode ini dapat memberikan fitur yang tidak dipengaruhi oleh perubahan ukuran objek, adanya translasi atau rotasi pada objek, serta sedikit terpengaruh terhadap perbedaan intensitas cahaya, dan perbedaan proyeksi tiga dimensi pada objek yang ingin dikenali. Setelah metode ini dikenal oleh Lowe pada tahun 1999,

penelitian untuk mengembangkan dan meningkatkan efisiensi dari metode ini terus dilakukan. Seperti yang dilakukan oleh Yan Ke & Rahul Sukthankar (2004) yang memadukan antara *Principal Component Analysis (PCA)* dan *SIFT* untuk mendapatkan akurasi pencocokan *keypoint* yang lebih baik.

Kemudian pada tahun 2009, A. Psyllos, C. N. Anagnostopoulos dan E. Kafayas mempublikasikan jurnal mereka [9] tentang pengenalan jenis kendaraan. Pada penelitiannya mereka menggunakan *SIFT* sebagai metode ekstraksi fitur untuk sistem pengenalan jenis kendaraan mereka. Pada penelitian tersebut, digunakan jenis mobil-mobil eropa yang sangat jarang ditemukan di Indonesia seperti Alfa Romeo, Citroen, Renault, Seat, dan lain-lain. Penulis merasa aplikasi sistem pengenalan kendaraan akan menjadi semakin dibutuhkan seiring dengan berkembangnya zaman. Oleh karena itu, penulis ingin mencoba melakukan suatu pengenalan jenis kendaraan dengan metode *SIFT* menggunakan jenis-jenis mobil Jepang yang sering dijumpai di Indonesia. Dengan kelebihan-kelebihan yang dimiliki oleh *SIFT*, penulis berharap akan mendapatkan hasil pengenalan jenis kendaraan yang baik.

## 1.2 Perumusan Masalah

*SIFT* sebagai metode ekstraksi fitur yang dikatakan dapat menghasilkan fitur-fitur yang invarian terhadap berbagai perubahan merupakan pilihan yang dianggap sesuai oleh penulis. Akan tetapi, dalam aplikasi suatu sistem pengenalan bukan hanya diperlukan ketepatan tapi juga kecepatan. Adapun perumusan masalah yang akan dibahas pada tulisan ini ialah:

- a. Bagaimana prinsip kerja *SIFT* dalam pengambilan fitur objek pada citra?
- b. Bagaimana performa dari *SIFT* pada aplikasi pengenalan jenis kendaraan?

## 1.3 Tujuan Penulisan

Penulisan ini bertujuan untuk mempelajari cara kerja *SIFT* sebagai metode ekstraksi fitur pada sistem pengenalan objek serta menguji sejauh mana efisiensi dan performansi dari metode ekstraksi fitur *SIFT* jika digunakan pada pengenalan jenis kendaraan dengan menggunakan program simulasi.

#### **1.4 Batasan Masalah**

Pada tulisan ini, masalah yang dibahas dibatasi hanya pada penelitian tentang efisiensi dan performansi dari metode ekstraksi fitur *SIFT* pada pengenalan jenis kendaraan.

#### **1.5 Sistematika Penulisan**

Sistematika pembahasan pada tulisan ini terdiri dari 5 bab sebagai berikut:

##### **BAB 1 PENDAHULUAN**

Bab ini merupakan gambaran permasalahan secara umum. Bab ini meliputi latar belakang penulisan, perumusan masalah pada tulisan, tujuan penulisan, pembatasan masalah pada tulisan dan sistematika penulisan.

##### **BAB 2 DASAR TEORI**

Pada bab ini akan dipaparkan tinjauan literatur serta dasar teori yang digunakan pada penelitian ini. Termasuk didalamnya pembahasan mengenai citra, pengenalan objek dan *Scale Invariant Feature Transform (SIFT)* sebagai metode ekstraksi fitur yang digunakan.

##### **BAB 3 PERANCANGAN SIMULASI SISTEM PENGENALAN JENIS KENDARAAN DENGAN METODE EKSTRAKSI FITUR *SIFT***

Pada bab ini akan dijelaskan struktur dan perangkat simulasi, proses simulasi, algoritma yang digunakan serta parameter analisis yang akan diambil

##### **BAB 4 HASIL DAN ANALISIS SIMULASI**

Pada bab ini akan dipaparkan hasil dari simulasi yang telah dilakukan oleh penulis. Selanjutnya juga akan dijelaskan analisis penulis tentang data yang diperoleh dari hasil simulasi.

##### **BAB 5 PENUTUP**

Pada bab ini akan dipaparkan kesimpulan yang dapat diambil oleh penulis dari penelitian ini serta saran-saran untuk pengembangan kedepannya.

## BAB 2

### DASAR TEORI

#### 2.1 Pengenalan Objek Pada Komputer

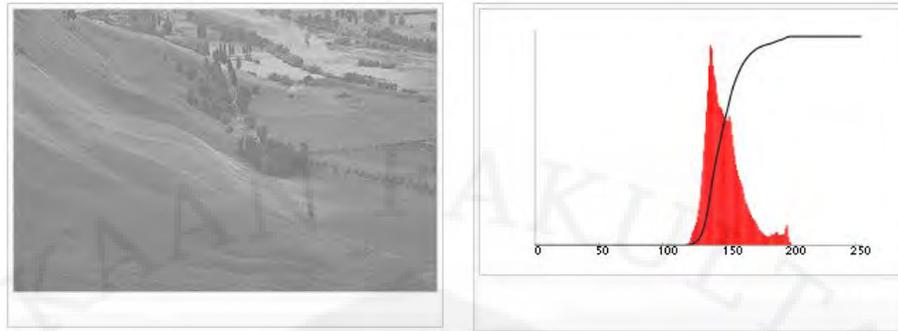
Pengenalan objek merupakan penelitian yang menggabungkan konsep citra digital, pengenalan pola, matematika, dan statistik [5]. Pengenalan objek berarti memberikan klasifikasi terhadap objek, benda atau bentuk tertentu yang terdapat pada suatu citra atau gambar [1]. Dalam prosesnya komputer akan mengambil elemen-elemen tertentu yang diperlukan untuk mengenali suatu objek yang terdapat pada citra yang ditangkapnya. Pengenalan objek umumnya terdiri dari deteksi dan pengenalan. Pada deteksi, komputer akan mencari dan mengidentifikasi komponen-komponen penting pada suatu citra untuk mengetahui ada atau tidaknya objek yang ingin dikenali pada citra tersebut. Komponen tersebut dapat berupa lokasi dari objek atau detail-detail lainnya yang kemudian akan diproses lebih lanjut. Pada pengenalan, objek yang telah dideteksi akan diambil fitur-fiturnya dan diklasifikasikan sesuai dengan aplikasi dari sistem tersebut. Pada sistem pengenalan objek terdapat beberapa tahapan penting yaitu: *pre-processing*, segmentasi objek dan ekstraksi fitur.

##### 2.1.1 Pre-processing

Pre-processing merupakan tahap awal yang dilakukan ketika menerima citra masukan. Umumnya tahap pre-processing akan menyeragamkan citra masukan menjadi *grayscale*. Selain untuk memperbaiki kualitas dari citra masukan, tahap *pre-processing* ini juga bertujuan untuk memanipulasi citra masukan sehingga sesuai dengan yang diinginkan. Tergantung dari masukan yang diinginkan, ada beberapa metode yang biasa digunakan pada tahap *pre-processing*.

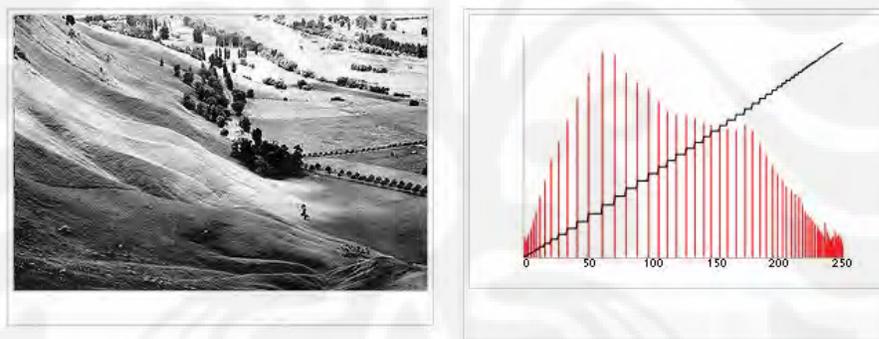
##### 2.1.1.1 Histogram Equalization

Histogram merupakan gambaran persebaran tingkat keabuan tertentu terhadap jumlah pixel pada suatu citra, seperti pada **Gambar 2.1**.



Gambar 2.1 Citra (kiri) dan histogramnya (kanan) [sumber: wikipedia.com]

Proses *histogram equalization* bertujuan untuk meratakan persebaran tingkat keabuan pada suatu citra *grayscale*. Pada **Gambar 2.2** terlihat bahwa citra yang diperoleh setelah melalui proses *histogram equalization* terlihat lebih baik daripada citra sebelumnya pada **Gambar 2.1**. Persebaran tingkat keabuan yang ditunjukkan oleh histogram pada **Gambar 2.2** juga terlihat lebih merata.



Gambar 2.2 Hasil dari (Gambar 2.1) setelah proses *histogram equalization*

[sumber: wikipedia.org]

*Histogram equalization* umumnya digunakan jika citra masukan terlalu gelap atau terlalu cerah.

#### 2.1.1.2 *Spatial Low-Pass Filter*

*Spatial low-pass filter* merupakan suatu proses untuk mereduksi frekuensi perubahan nilai pixel suatu citra. *Spatial low-pass filter* akan mempertahankan komponen dengan frekuensi rendah (perubahan nilai pixel yang bertahap) dan menghilangkan komponen frekuensi tinggi (perubahan nilai pixel secara

mendadak). Filter jenis ini akan menimbulkan efek halus (*smooth*) pada citra hasil keluaran. Contoh hasil dari *spatial low-pass filter* ditunjukkan pada **Gambar 2.3**.



Gambar 2.3 Citra asli (kiri) dan citra hasil low-pass filter (kanan)

#### 2.1.1.3 *Spatial High-Pass Filter*

Kebalikan dari *low-pass filter*, *high-pass filter* akan meningkatkan frekuensi perubahan pixel suatu citra. Filter ini akan mempertahankan komponen frekuensi tinggi dan menghilangkan komponen frekuensi rendah. Filter ini akan menimbulkan efek hasil citra yang lebih tajam dan pemisahan tepi yang lebih jelas. Hasil dari *spatial high-pass filter* ditunjukkan pada **Gambar 2.4**.



Gambar 2.4 Citra asli (kiri) dan citra hasil high-pass filter (kanan)

#### 2.1.1.4 *Median Filter*

*Median filter* termasuk kedalam jenis *low-pass filter*. Hanya saja, pada *median filter* tidak dilakukan operasi konvolusi. Median filter akan mengambil ukuran wilayah tertentu pada citra (3x3, 5x5, 7x7, dan lain-lain), kemudian menghitung nilai median dari pixel di wilayah tersebut. Kemudian pixel yang

berada di tengah akan diganti dengan nilai median yang telah diperoleh. Filter ini sangat efektif dalam menghilangkan derau (*noise*). Pada **Gambar 2.5**, terlihat bahwa *noise* pada citra asli hilang ketika dilakukan proses *median filter*.



Gambar 2.5 Citra asli (kiri) dan citra hasil median filter dengan wilayah 3x3 (kanan) [sumber: wikipedia.org]

### 2.1.2 Segmentasi Citra

Segmentasi citra merupakan proses untuk membagi suatu citra menjadi beberapa daerah (*region*) dimana setiap daerah memiliki kemiripan atribut [6]. Kemiripan atribut yang dimaksud ialah apakah pixel-pixel tersebut memiliki intensitas keabuan yang sama, membentuk suatu tekstur, menyusun bentuk tertentu, berada pada baris yang sama, dan lain-lain [5]. Pada sistem pengenalan objek, proses segmentasi ini bertujuan untuk memisahkan objek yang ingin dianalisa dengan keseluruhan citra. Terdapat banyak jenis metode yang digunakan untuk segmentasi citra, beberapa diantaranya ialah: segmentasi berbasis *cluster*, segmentasi berbasis histogram, deteksi tepi, segmentasi berbasis grafik, dan lain-lain. Pada **Gambar 2.6** ditunjukkan contoh dari proses segmentasi citra yang menggunakan metode berbasis grafik.



Gambar 2.6 Segmentasi citra menggunakan metode berbasis grafik [sumber: <http://people.cs.uchicago.edu/~pff/segment/>]

### 2.1.3 Ekstraksi Fitur

Ekstraksi fitur merupakan proses terpenting dalam pengenalan objek. Fitur adalah karakteristik atau hal-hal khusus yang mencirikan satu objek. Ekstraksi fitur berarti mengambil dan menyimpan karakteristik-karakteristik khusus dari suatu objek pada citra. Fitur-fitur inilah yang kemudian akan digunakan sebagai pembandingan untuk mengenali objek tertentu pada suatu citra. Karakteristik fitur yang baik sebisa mungkin memenuhi persyaratan sebagai berikut [6]:

- a. Dapat membedakan suatu objek dengan yang lainnya
- b. Kompleksitas komputasi yang tidak terlalu rumit
- c. Tidak terikat (invarian) terhadap transformasi
- d. Jumlahnya sedikit

## 2.2 Scale Invariant Feature Transform

Pada tahun 1999, David G. Lowe seorang peneliti dari University of British Columbia memperkenalkan suatu metode baru dalam ekstraksi fitur dari suatu citra. Metode ekstraksi fitur ini disebut sebagai *Scale-Invariant Feature Transform (SIFT)*. Dengan menggunakan *SIFT* ini, suatu citra akan di ubah menjadi vektor fitur lokal yang kemudian akan digunakan sebagai pendekatan dalam mendeteksi objek yang dimaksud.

Sebagai metode ekstraksi fitur pada pengenalan objek, *SIFT* ini memiliki kelebihan-kelebihan sebagai berikut:

- a. Hasil ekstraksi fitur bersifat invarian terhadap ukuran, translasi dan rotasi dua dimensi.
- b. Hasil ekstraksi fitur bersifat invarian sebagian terhadap perubahan iluminasi dan perubahan sudut pandang tiga dimensi
- c. Mampu meng-ekstrak banyak *keypoint* dari citra yang tipikal
- d. Hasil ekstraksi fitur benar-benar mencirikan secara khusus (*distinctive*)

Dengan kelebihan-kelebihan tersebut, penggunaan metode *SIFT* banyak dikembangkan untuk aplikasi pengenalan objek.

Secara garis besar, algoritma yang digunakan pada metode *SIFT* terdiri dari empat tahap [4], yaitu:

- a. Mencari Nilai Ekstrim Pada Skala Ruang
- b. Menentukan *Keypoint*
- c. Penentuan Orientasi
- d. Deskriptor *Keypoint*

Setelah melalui tahapan tersebut maka akan diperoleh fitur-fitur lokal yang digunakan sebagai *descriptor* (penciri) dari suatu objek untuk diolah lebih lanjut.

### 2.2.1 Pencarian Nilai Ekstrim pada Skala Ruang

Pencarian nilai ekstrim pada skala ruang merupakan tahap awal dalam penentuan *keypoint* dari suatu citra. Dengan menggunakan fungsi Gaussian, citra pada skala ruang dapat didefinisikan sebagai fungsi  $L(x,y,\sigma)$ , yang diperoleh dari hasil konvolusi skala-variabel Gaussian,  $G(x,y,\sigma)$ , dengan citra masukan  $I(x,y)$ , sehingga diperoleh [4]:

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \dots\dots\dots (2.1)$$

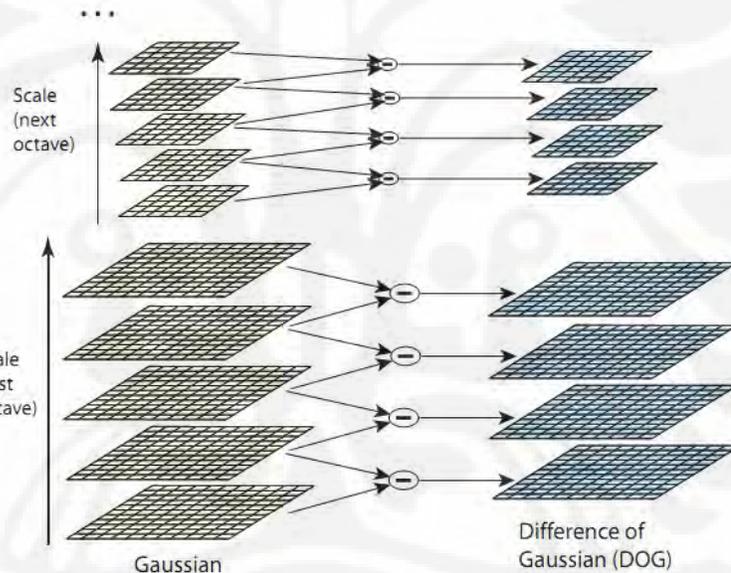
dimana  $*$  adalah operasi konvolusi antara  $x$  dan  $y$  dan  $G(x,y,\sigma)$  adalah skala-variabel Gaussian [4]:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}} \dots\dots\dots(2.2)$$

Citra hasil *Difference-of-Gaussian*,  $D(x, y, \sigma)$ , diperoleh dengan melakukan operasi konvolusi pada citra masukan dengan filter *Difference-of-Gaussian*, maka [4]:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \dots\dots\dots(2.3) \end{aligned}$$

Dari persamaan (2.3) terlihat bahwa citra hasil *Difference-of-Gaussian* sebenarnya merupakan selisih antara citra hasil pengkaburan Gaussian dengan nilai skala  $k$  yang berbeda. Proses ini diilustrasikan pada **Gambar 2.7**

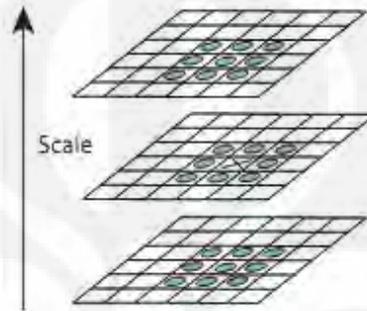


Gambar 2.7 Diagram *Difference-of-Gaussian* [4]

Citra hasil konvolusi kemudian dikelompokkan berdasarkan *octave* (satu *octave* setara dengan penggandaan besarnya nilai  $\sigma$ ), nilai  $k$  ditetapkan di awal sehingga diperoleh jumlah citra kabur yang sama pada setiap *octave* serta diperoleh citra hasil *DoG* yang sama untuk setiap *octave*.

Setelah diperoleh citra *DoG* pada setiap *octave*, maka langkah selanjutnya ialah mencari kandidat *keypoint*. Kandidat *keypoint* dideteksi sebagai titik

maksimum lokal atau titik minimum lokal dari citra hasil DoG. Untuk mencari nilai maksimum atau minimum lokal maka masing-masing piksel pada citra hasil DoG akan di bandingkan dengan 8 piksel disekitarnya yang berada pada skala yang sama dan dengan 9 piksel yang bersesuaian dengannya pada skala berbeda (perhatikan **Gambar 2.8**). Jika piksel tersebut merupakan maksimum atau minimum lokal, maka piksel tersebut akan dijadikan sebagai kandidat *keypoint*.



Gambar 2.8 Ilustrasi pencarian maksimum atau minimum lokal dari citra DoG [4]

### 2.2.2 Penentuan *Keypoint*

Setelah kandidat *keypoint* ditemukan melalui tahapan pada 2.2.1, maka langkah selanjutnya ialah untuk mengambil detail dari kandidat *keypoint* tersebut. Detail yang diambil merupakan lokasi, skala dan rasio kelengkungan inti dari kandidat *keypoint*. Pada tahap ini akan terjadi pengurangan jumlah kandidat *keypoint*. Dimana setiap kandidat *keypoint* yang dianggap sangat rentan terhadap gangguan (noise) akan dihilangkan, yaitu kandidat *keypoint* yang memiliki nilai kontras yang rendah dan kandidat *keypoint* yang kurang jelas dan terletak di sepanjang tepi.

Untuk menghilangkan *keypoint* dengan nilai kontras yang rendah, maka digunakan langkah-langkah berikut:

- Dengan menggunakan deret Taylor pada fungsi skala ruang  $D(x, y, \sigma)$  yang kemudian digeser sehingga titik asal berada pada titik sampel, sehingga [4]:

$$D(\mathbf{x}) = D + \frac{\partial D}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x} \dots\dots\dots(2.4)$$

Dimana  $D$  dan penurunannya dilihat dari titik sampel, dan  $x = (x, y, \sigma)^T$  adalah offset dari titik ini

- Lokasi dari nilai ekstrim,  $x$ , diambil dengan menurunkan persamaan 2.4 terhadap  $x$ , sehingga [4]:

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \dots\dots\dots(2.5)$$

- Fungsi nilai ekstrim, ( ), diperoleh dengan men-substitusikan persamaan 2.5 kepada persamaan 2.4 sehingga menghasilkan persamaan berikut [4]:

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \dots\dots\dots(2.6)$$

Pada SIFT ini, semua nilai ekstrim, ( ), yang bernilai kurang dari 0,03 akan dihilangkan.

Untuk menghilangkan *keypoint* yang kurang jelas pada tepi maka digunakan persamaan berikut [4]:

$$\frac{\text{Tr}(\mathbf{H})^2}{\text{Det}(\mathbf{H})} < \frac{(r+1)^2}{r} \dots\dots\dots(2.7)$$

dimana  $H$  merupakan matrix Hessian  $2 \times 2$  dan  $r$  merupakan ambang batas dari kecekungan inti yang diperbolehkan. Pada SIFT ini, semua *keypoint* yang memiliki nilai  $r$  lebih besar dari 10 akan dihilangkan.

### 2.2.3 Penentuan Orientasi

Pada tahap ini, masing-masing *keypoint* yang diperoleh akan diberikan suatu orientasi yang tetap berdasarkan sifat-sifat lokal pada citra. Dengan adanya proses ini maka *keypoint* yang diperoleh dapat direpresentasikan relatif terhadap orientasi ini sehingga *keypoint* yang dihasilkan tidak terpengaruh terhadap adanya rotasi pada citra. Untuk menentukan orientasi dari masing-masing *keypoint* maka dilakukan perhitungan terhadap besarnya gradien dan sudut arah orientasi. Adapun perhitungan terhadap besar nilai gradien,  $m(x,y)$ , dan arah orientasi,  $\theta(x,y)$ , dilakukan menggunakan persamaan berikut [4]:

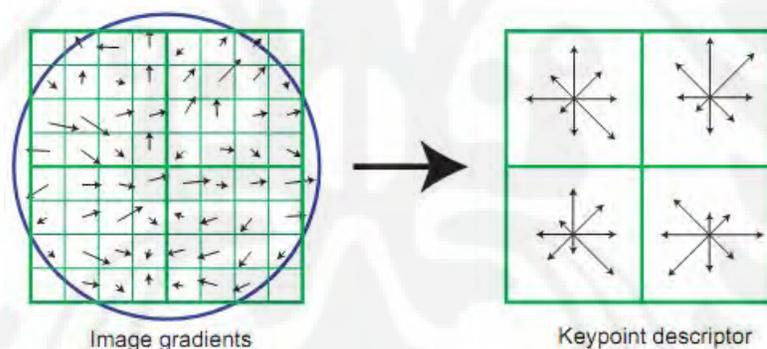
$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \quad (2.8)$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1)) / (L(x+1,y) - L(x-1,y))) \quad (2.9)$$

### 2.2.4 Deskriptor *Keypoint*

Pada proses ini, masing-masing *keypoint* yang telah diorientasikan akan diberikan pencirian khusus (deskriptor). Proses ini bertujuan untuk mendapatkan *keypoint* yang invarian terhadap perubahan intensitas cahaya atau perubahan sudut pandang tiga dimensi.

Deskriptor akan diukur sebagai suatu histogram orientasi pada wilayah piksel dengan ukuran 4x4. Nilai orientasi diperoleh dari citra Gaussian yang memiliki skala terdekat dengan skala *keypoint* yang akan dihitung. Agar *keypoint* yang diperoleh invarian terhadap orientasi, maka koordinat dari deskriptor dan gradien orientasi akan di rotasi relatif terhadap orientasi dari *keypoint*. Kemudian fungsi pembebanan Gaussian, dengan besar nilai  $\sigma$  satu setengah kali dari besar jendela deskriptor, akan digunakan sebagai pembeban pada setiap besaran nilai dari titik sampel. Proses ini ditunjukkan pada lingkaran yang terdapat pada **Gambar 2.9** sebelah kiri.



Gambar 2.9 Deskriptor dari perhitungan besar gradien dan orientasi serta gambar lingkaran Gaussian (kiri) dan gambar deskriptor *keypoint* (kanan) [4]

Deskriptor *keypoint* pada Gambar 2.9 menunjukkan adanya 8 arah pada masing-masing histogram orientasi dengan panjang masing-masing anak panah sesuai dengan besar nilai dari histogram asal. Selanjutnya deskriptor *keypoint* yang telah diperoleh akan dinormalisasi untuk mengatasi pengaruh perubahan cahaya.

Setelah melalui tahapan-tahapan tersebut, maka pada hasil akhir akan diperoleh suatu citra dengan *keypoint* yang invarian terhadap berbagai macam perubahan, seperti yang ditunjukkan pada gambar 2.10



Gambar 2.10 *Keypoint* yang diperoleh dari suatu citra menggunakan *SIFT* [4]

*Keypoint* ini yang kemudian menjadi fitur-fitur lokal pada suatu citra dan akan di cocokkan dengan *keypoint-keypoint* yang terdapat pada citra lain untuk menyesuaikan dengan objek yang tersedia pada data set training.

## BAB 3

### PERANCANGAN SIMULASI SISTEM PENGENALAN JENIS KENDARAAN DENGAN METODE EKSTRAKSI FITUR *SIFT*

#### 3.1 Perangkat Simulasi

Simulasi akan dirancang dengan menggunakan *software* MATLAB versi 7.9.0.529 (R2009b) dengan menggunakan *image processing toolboxes* dan *SIFT toolboxes* yang dibuat oleh David G. Lowe [10] dengan beberapa tambahan dari penulis.

Perangkat yang digunakan untuk menjalankan simulasi ialah komputer dengan prosesor Intel Core 2 Duo T5750 @2.00 GHz (2 CPUs) dengan memori 2 GB RAM.

#### 3.2 Basis Data Citra

Sebagai citra *training* digunakan bagian depan dari beberapa jenis kendaraan. Dimana bagian depan dari kendaraan ini yang akan menjadi fitur khas yang akan dicocokkan dengan citra tes.



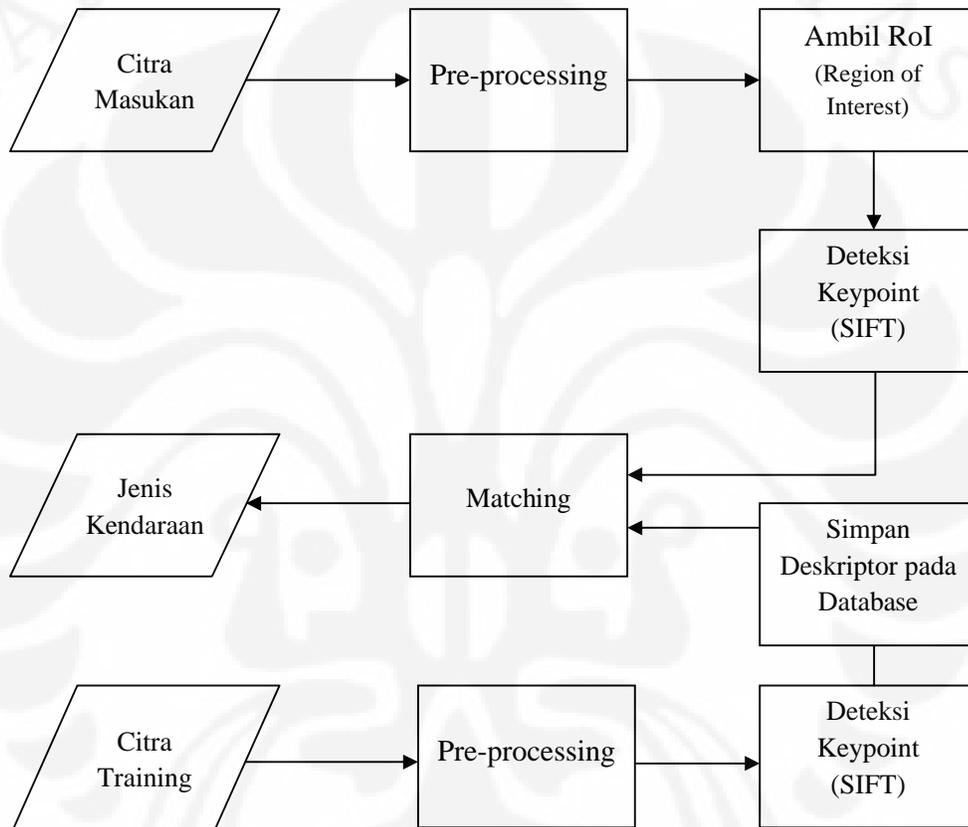
Gambar 3.1 Contoh citra *training* yang akan digunakan

Citra akan diambil menggunakan kamera digital Nikon Coolpix L21 dengan resolusi 640x480. Set data citra *training* akan terdiri dari 5 citra dari jenis kendaraan yang berbeda. Citra diambil dengan jarak sekitar 2 meter dari kendaraan dan tinggi sekitar 1,5 meter dari tanah. Citra *training* akan diambil dari kendaraan-kendaraan yang berada di lingkungan kampus UI. Citra training akan diproses secara manual untuk mengambil bagian depannya saja.

Sebagai basis data citra, akan diambil gambar dari berbagai jenis kendaraan yang berbeda dan disimpan sebagai citra training untuk dicocokkan dengan citra tes nantinya.

### 3.3 Diagram Alir Simulasi

Berikut merupakan diagram alir perancangan simulasi yang akan dibuat:



Gambar 3.2 Diagram Alir Proses Simulasi

#### 3.3.1 Citra Masukan

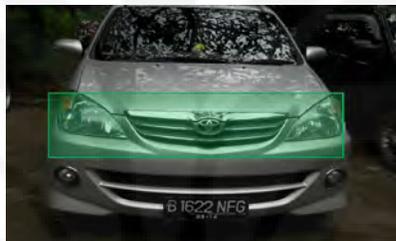
Data citra tes terdiri dari 10 jenis kendaraan yang sama dengan masing-masing jenis kendaraan. Citra tes akan diambil menggunakan kamera digital Nikon Coolpix L21 dengan resolusi 640x480. Citra diambil dengan jarak sekitar 2 meter dari kendaraan dan tinggi sekitar 1,5 meter dari tanah. Citra tes akan diambil dari kendaraan-kendaraan yang berada di lingkungan kampus UI.

### 3.3.2 Tahap *Pre-processing*

Tahap *pre-processing* akan dilakukan pada citra sebelum dilakukan proses pendeteksian *RoI* dan *keypoint*. Pada tahap ini citra akan diubah dari RGB menjadi *Grayscale*.

### 3.3.3 Deteksi *Region of Interest (RoI)*

Setelah melalui tahap *pre-processing* kemudian akan dilakukan pendeteksian *Region of Interest (RoI)* pada citra. *RoI* yang diambil adalah bagian depan kendaraan yang merupakan fitur khas dan fokus pada pencocokan *keypoint*.



Gambar 3.3 Deteksi *RoI* pada citra tes

### 3.3.4 Pencocokan *Keypoint (Matching)*

Pada tahap ini akan dilakukan pencocokan antara deskriptor dari citra tes dengan semua deskriptor dari basis data citra training. Selanjutnya jenis kendaraan akan dikenali melalui banyaknya jumlah deskriptor yang cocok antara citra tes dengan citra training. Proses matching ini menggunakan metode *Nearest Neighbour* dimana *keypoint* tersebut akan dianggap cocok jika jarak nilai deskriptor  $D_A$  (citra tes) dan  $D_B$  (citra *training*) lebih kecil dari nilai batas (*threshold*)  $T$ .

## 3.4 Parameter Analisis

Pada penelitian ini akan digunakan *recall* dan *1-precision* sebagai parameter analisis seperti yang dipaparkan pada [9]. *Recall* merupakan perbandingan antara *correct matches* dengan jumlah total *correspondences* seperti yang ditunjukkan pada persamaan (3.1) [9], sedangkan *1-precision* adalah

perbandingan antara jumlah *false matches* dengan total jumlah *keypoint* yang cocok seperti ditunjukkan pada persamaan (3.2) [9].

$$\text{recall} = \frac{\text{jumlah correct matches}}{\text{total correspondences}} \dots\dots\dots(3.1)$$

$$1\text{-precision} = \frac{\text{jumlah false matches}}{\text{total matches}} \dots\dots\dots(3.2)$$

Jumlah *correct matches* merupakan jumlah *keypoint* yang cocok dan berada pada lokasi yang sama, sedangkan jumlah *false matches* merupakan jumlah *keypoint* yang cocok namun berada pada lokasi yang berbeda. Total *correspondences* merupakan jumlah *keypoint* yang berada pada citra *training*, sedangkan total *matches* merupakan jumlah total dari *keypoint* yang terdeteksi cocok pada kedua citra.

Pada simulasi ini, fitur (*keypoint*) yang ada pada wilayah A dan B akan dibandingkan satu sama lain dengan menggunakan metode *Nearest Neighbour*. Kemudian *keypoint* tersebut akan dianggap cocok jika jarak nilai deskriptor  $D_A$  dan  $D_B$  lebih kecil dari nilai batas (*threshold*)  $T$ . Untuk mendapatkan perbandingan grafik *recall vs 1-precision* maka pada simulasi ini nilai  $T$  akan divariasikan. Nilai  $T$  akan divariasikan mulai dari 0.4 – 0.7 dengan skala kenaikan 0,05. Selain itu, akan dihitung juga persentase akurasi dari pengenalan jenis kendaraan serta waktu proses yang diperlukan untuk melakukan keseluruhan proses.

## BAB 4

### HASIL DAN ANALISIS SIMULASI

#### 4.1 Data Simulasi

##### 4.1.1 Basis Data Citra

Pada simulasi ini digunakan basis data berupa citra dari 5 jenis mobil yang berbeda. Jenis mobil yang digunakan pada simulasi ini ialah: Toyota Avanza, Honda Jazz, Daihatsu Terios, Isuzu Panther dan Toyota Kijang. Citra yang telah diambil, kemudian di-*crop* secara manual untuk mendapatkan bagian depan dari kendaraan yang kemudian digunakan sebagai fitur khusus yang akan dibandingkan. Citra yang digunakan sebagai basis data dapat dilihat pada **Gambar 4.1**.



Gambar 4.1 Citra yang digunakan sebagai basis data

##### 4.1.2 Citra Tes Yang Digunakan

Seperti yang telah dijelaskan pada bagian 3.3.1, citra tes yang digunakan pada penelitian ini diambil dari sekitar lingkungan kampus Universitas Indonesia. Hal ini berarti masing-masing citra diambil dari lokasi yang berbeda-beda. Setiap citra diambil pada resolusi 640 x 480 pixel. Selain itu citra juga tidak diambil pada saat yang sama, sehingga sangat besar kemungkinan terjadinya perbedaan intensitas cahaya pada citra satu dan citra lainnya.

Pada penelitian ini digunakan sepuluh citra tes untuk masing-masing jenis kendaraan yang akan diuji, sehingga terdapat total lima puluh citra masukan. Terdapat beberapa kendaraan yang diambil dari sudut pandang yang sedikit berbeda dan warna yang berbeda dari citra pada basis data. Beberapa contoh citra

yang digunakan sebagai masukan dapat dilihat pada **Gambar 4.2** di bawah ini, sedangkan citra masukan yang selengkapnya dapat dilihat pada **Lampiran 1**.



Gambar 4.2 Contoh citra yang digunakan sebagai masukan

## 4.2 Hasil Dan Analisis Simulasi

### 4.2.1 Proses Segmentasi Citra

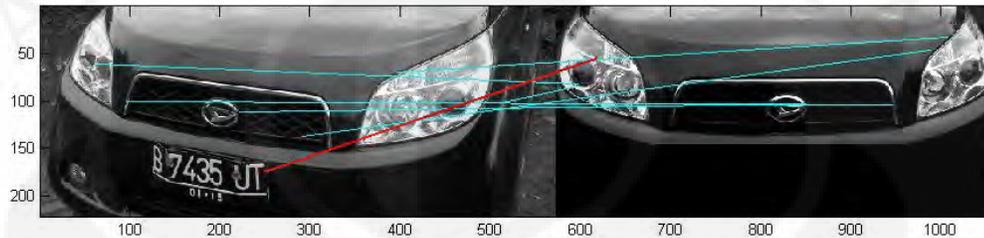
Pada simulasi ini digunakan proses segmentasi citra menggunakan *Region of Interest (RoI)*. Proses ini hanya mengambil bagian depan dari kendaraan saja untuk diambil *keypoint* melalui proses selanjutnya. Untuk proses deteksi *RoI* ini, digunakan teknik *crop* sederhana, dimana setiap citra masukan akan di *crop* atau dilewatkan pada bagian tertentu saja sesuai dengan wilayah yang telah ditetapkan. Kekurangan dari metode ini ialah, sulit diperoleh hasil segmentasi yang benar-benar hanya mendapatkan bagian depan kendaraan saja.



Gambar 4.3 Citra masukan asli (kiri) dan setelah dideteksi *RoI* (kanan)

Pada **Gambar 4.3** dapat dilihat bahwa setelah melalui proses deteksi *RoI* masih terdapat bagian yang tidak diinginkan, yaitu bagian nomor polisi, jika dibandingkan dengan citra basis data pada **Gambar 4.1**. Bagian yang tidak diinginkan ini akan ikut serta pada proses selanjutnya yaitu proses deteksi

*keypoint* dan proses *matching keypoint*. Dengan demikian, adanya bagian citra yang tidak diperlukan ini akan menambah waktu proses ketika melakukan *deteksi keypoint* dan *matching keypoint*. Hal ini dikarenakan proses akan tetap menghitung keseluruhan bagian yang terdapat pada citra baik yang diperlukan maupun tidak diperlukan. Adanya bagian yang tidak diperlukan juga dapat menurunkan akurasi pencocokan *keypoint* pada proses *matching keypoint*.



Gambar 4.4 Kesalahan *matching keypoint* pada bagian citra yang tidak diperlukan

Hal ini dikarenakan ada kemungkinan bagian tersebut memiliki kesamaan dengan *keypoint* yang terdapat pada basis data seperti contoh yang ditunjukkan pada **Gambar 4.4**.

#### 4.2.2 Proses Deteksi Keypoint Menggunakan *SIFT*

Pada proses ini, penggunaan algoritma *SIFT* berhasil mendeteksi *keypoint* pada masing-masing citra dengan baik. Jumlah *keypoint* yang terdeteksi dapat dilihat pada **Tabel 4.1** dan **Tabel 4.2**.

Tabel 4.1 Jumlah *keypoint* yang terdeteksi pada citra basis data

No	Jenis Mobil	Jumlah <i>keypoint</i> yang terdeteksi
1	Toyota Avanza	265
2	Honda Jazz	194
3	Daihatsu Terios	263
4	Isuzu Panther	237
5	Toyota Kijang	320

Tabel 4.2 Rata-rata jumlah *keypoint* yang terdeteksi pada citra tes

No	Jenis Mobil	Rata-rata jumlah <i>keypoint</i> yang terdeteksi (dengan pembulatan)
1	Toyota Avanza	438
2	Honda Jazz	331
3	Daihatsu Terios	598
4	Isuzu Panther	434
5	Toyota Kijang	696

Pada **Tabel 4.2** terlihat bahwa jumlah *keypoint* yang terdeteksi pada citra tes jauh lebih banyak dari pada jumlah *keypoint* yang terdeteksi pada citra basis data seperti yang ditunjukkan pada **Tabel 4.1**. Hal ini kemungkinan dikarenakan akibat segmentasi citra yang kurang sempurna seperti yang telah dijelaskan pada bagian 4.2.1.

#### 4.2.3 Akurasi Pencocokan *Keypoint*

Seperti yang telah dijelaskan pada bagian 3.4, untuk mengetahui akurasi dan efisiensi pencocokan *keypoint* yang dilakukan oleh algoritma SIFT akan digunakan variabel *recall* dan *1-precision* [9]. Seperti ditunjukkan pada persamaan (3.1), nilai *recall* akan menunjukkan seberapa mirip suatu citra dengan citra pada basis data, sedangkan *1-precision* akan menunjukkan akurasi dari proses pencocokan *keypoint*. Semakin tinggi nilai *recall* berarti kemungkinan citra tersebut sangat mendekati dengan citra tes, sedangkan semakin tinggi nilai *1-precision* menunjukkan kesesuaian *keypoint* yang kurang presisi.

Pada **Lampiran 2** dicantumkan data lengkap hasil dari simulasi yang telah dilakukan. Selama proses pengenalan terdapat tiga hasil pengenalan yang ditemukan oleh penulis, yaitu: jenis kendaraan yang berhasil dikenali dengan benar, jenis kendaraan yang salah dikenali dan jenis kendaraan yang tidak dikenali. Jenis kendaraan yang dikenali dengan benar (dinyatakan dengan kata 'ya'), berarti telah terjadi kecocokan (*matching*) *keypoint* antara citra tes dengan citra *training* dan jenis kendaraan pada citra tes sama dengan jenis kendaraan yang dikenali pada citra *training*. Untuk jenis kendaraan yang salah dikenali (dinyatakan dengan kata 'salah'), berarti telah terjadi kecocokan *keypoint* antara

citra tes dengan citra *training*, namun jenis kendaraan yang dikenali tidak sesuai antara citra tes dan citra *training*. Pada jenis kendaraan yang tidak dikenali (dinyatakan dengan kata ‘tidak’), berarti tidak terjadi kecocokan *keypoint* antara citra tes dengan seluruh citra *training* yang terdapat pada basis data. Pada penelitian ini, keseluruhan jenis pengenalan akan diperhitungkan sebagai akurasi dari pengenalan jenis kendaraan, namun hanya citra yang berhasil dikenali dengan benar saja yang datanya dimasukkan kedalam perhitungan *recall* dan *1-precision*.

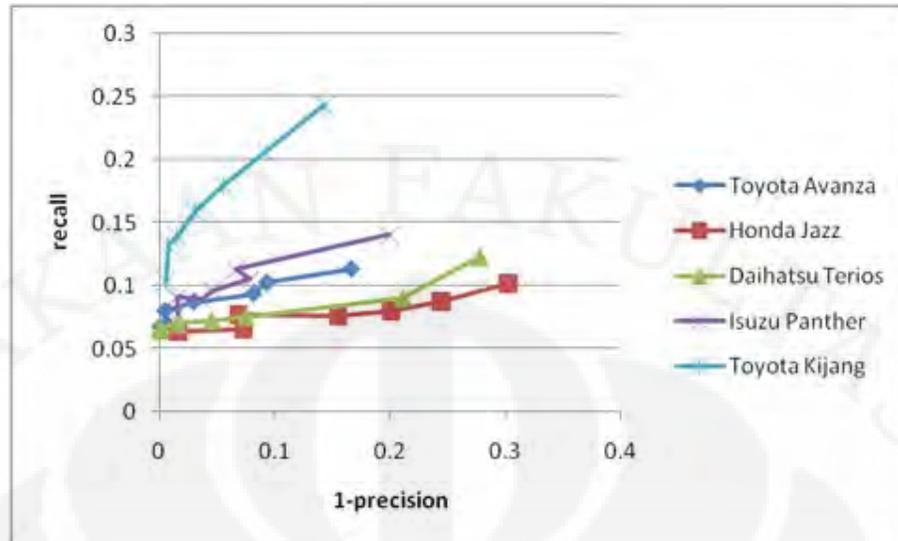
Pada **Tabel 4.3** diperlihatkan rata-rata nilai *recall* dan nilai *1-precision* yang diperoleh untuk masing-masing jenis kendaraan. Plot grafik dari *recall* vs. *1-precision* pada masing-masing jenis kendaraan dapat dilihat pada **Gambar 4.5**. Grafik pada **Gambar 4.5** menunjukkan akurasi pencocokan *keypoint* untuk masing-masing jenis kendaraan.

Tabel 4.3 Nilai recall dan 1-precision pada masing-masing jenis kendaraan untuk setiap nilai T

No	Threshold (T)	Toyota Avanza		Honda Jazz	
		1-precision	recall	1-precision	recall
1	0.5	0	0.0671	0.016	0.0634
2	0.55	0.0053	0.0701	0.0729	0.0654
3	0.6	0.0046	0.08	0.0687	0.0768
4	0.65	0.0294	0.0871	0.1551	0.0757
5	0.7	0.0814	0.0935	0.201	0.0798
6	0.75	0.0933	0.1026	0.2444	0.0876
7	0.8	0.1666	0.1132	0.3028	0.102

Tabel 4.3 Nilai recall dan 1-precision pada masing-masing jenis kendaraan untuk setiap nilai T

Daihatsu Terios		Isuzu Panther		Toyota Kijang	
1-precision	recall	1-precision	recall	1-precision	recall
0	0.0646	0.0159	0.078	0.0059	0.104
0	0.0669	0.0151	0.0913	0.0077	0.1325
0.0159	0.0703	0.0365	0.089	0.0154	0.1393
0.045	0.0726	0.0466	0.0962	0.0304	0.159
0.0744	0.0756	0.0787	0.105	0.0557	0.18
0.2112	0.0908	0.0665	0.1126	0.0921	0.2062
0.2777	0.1231	0.1997	0.1405	0.143	0.2434



Gambar 4.5 Grafik *recall* vs. *1-precision* dari masing-masing jenis kendaraan

Dari nilai yang ditunjukkan pada **Tabel 4.3** dan grafik pada **Gambar 4.5**, dapat dilihat bahwa nilai dari *recall* dan *1-precision* akan berbanding lurus dengan kenaikan nilai *Threshold* ( $T$ ). Hal ini terlihat pada **Gambar 4.5** dimana nilai plot *recall* akan semakin ke atas dan nilai plot *1-precision* semakin ke kanan seiring dengan kenaikan nilai  $T$ . Hal ini disebabkan karena semakin besar nilai  $T$  maka jarak antara dua deskriptor yang ditoleransi juga akan semakin besar sehingga sangat besar kemungkinan dua deskriptor dianggap cocok (*match*) oleh sistem meski kedua deskriptor tersebut belum tentu berada pada lokasi yang sama. Hal ini berpengaruh pada akurasi dari pencocokan *keypoint* itu sendiri. Meskipun semakin banyak *match* yang terjadi, tapi nilai akurasi dari kecocokan *keypoint* semakin berkurang dikarenakan semakin bertambah pula nilai *false matches* yang dihasilkan. Akan tetapi semakin kecil nilai  $T$  maka akurasi *matching keypoint* akan semakin baik, terlihat dari grafik yang menunjukkan nilai *1-precision* sama dengan nol sedangkan nilai *recall* tidak. Hal ini menunjukkan bahwa semua *keypoint* yang *match* benar-benar sesuai antara citra tes dan citra pada basis data.

Dari grafik pada **Gambar 4.5** terlihat bahwa akurasi pencocokan *keypoint* yang paling baik diperoleh untuk jenis kendaraan Toyota Kijang, sedangkan akurasi pencocokan *keypoint* yang paling buruk diperoleh untuk jenis kendaraan Honda Jazz. Hal ini menunjukkan hasil pengenalan pada jenis kendaraan Toyota

Kijang lebih baik dan stabil, sebaliknya hasil pengenalan pada jenis kendaraan Honda Jazz masih kurang terpercaya karena akurasi kecocokan antara *keypoint* yang rendah. Ada beberapa hal yang dapat penulis analisis dari hasil perolehan ini yaitu:

- a. Adanya refleksi dari lingkungan sekitar yang dihasilkan oleh kendaraan, hal ini kemungkinan menyebabkan perbedaan nilai kontras pada titik yang sama seperti yang dijelaskan pada bagian 2.2.2. Citra tes juga diambil dari lokasi yang berbeda-beda sehingga refleksi yang dihasilkan juga berbeda.
- b. Perbedaan intensitas dan pantulan cahaya matahari pada kendaraan ketika citra diambil.
- c. Perbedaan seri kendaraan untuk jenis yang sama menimbulkan perbedaan disain bagian depan dari kendaraan.
- d. Hasil perhitungan *correct matches* dan *false matches* dilakukan secara manual, sehingga kemungkinan didapatkan data yang kurang presisi

#### 4.2.4 Akurasi Pengenalan Jenis Kendaraan

Pada **Tabel 4.4** ditunjukkan jumlah kendaraan yang dapat dikenali serta jumlah kendaraan yang salah dikenali dan tidak dapat dikenali pada simulasi yang di rancang oleh penulis untuk masing-masing nilai T.

Tabel 4.4 Jumlah jenis kendaraan yang dapat dikenali dan rata-rata waktu proses

No	Threshold (T)	Benar	Salah	Tidak Dikenali	Persen Akurasi (%)	Rata-rata Waktu Proses (s)
1	0.5	18	1	31	36.00	1.201
2	0.55	20	1	29	40.00	1.203
3	0.6	30	7	13	60.00	1.203
4	0.65	33	9	8	66.00	1.2054
5	0.7	38	12	0	<b>76.00</b>	1.224
6	0.75	36	14	0	72.00	1.22
7	0.8	38	12	0	<b>76.00</b>	1.23

Dari **Tabel 4.4** dapat dilihat bahwa tingkat akurasi maksimum yang dapat dicapai oleh simulasi ini ialah 76% yaitu pada nilai *threshold* 0.7 dan 0.8. Dengan nilai akurasi tersebut, menurut penulis simulasi yang dirancang pada penelitian ini dapat dikatakan baik namun masih kurang efektif jika ingin digunakan sebagai aplikasi sehari-hari.

#### 4.2.5 Waktu Proses

Pada bagian 3.1 telah dijelaskan perangkat yang digunakan untuk menjalankan simulasi ini. **Tabel 4.4** menunjukkan rata-rata waktu proses yang diperlukan untuk melalui seluruh tahapan simulasi jika menggunakan perangkat yang dijelaskan pada bagian 3.1. Dari **Tabel 4.4** dapat dilihat bahwa kenaikan nilai T mempengaruhi lama waktu proses yang diperlukan meski tidak terlalu signifikan. Untuk proses rata-rata membutuhkan waktu 1,2 detik. Dapat dilihat bahwa rentang waktu tersebut dapat dikatakan cepat untuk suatu proses pengenalan. Akan tetapi, tentunya untuk jumlah basis data yang lebih banyak akan membutuhkan waktu proses yang mungkin lebih lama. Selain itu untuk perangkat simulasi yang lebih maju, mungkin akan menghasilkan waktu proses yang lebih singkat.

#### 4.2.6 Nilai *Threshold* Optimum

Pada **Tabel 4.4** diperlihatkan bagaimana akurasi yang dihasilkan serta waktu proses yang diperlukan untuk masing-masing nilai T. Dari tabel tersebut terlihat bahwa untuk akurasi, nilai *threshold* 0,7 dan 0,8 memiliki akurasi pengenalan yang sama yaitu 76%. Akan tetapi, jika melihat dari waktu proses yang dibutuhkan maka nilai *threshold* 0,7 lebih baik daripada nilai *threshold* 0,8. Selain itu, dari pembahasan pada bagian 4.2.3 juga disebutkan bahwa besarnya nilai *threshold* akan mempengaruhi akurasi kecocokan *keypoint*. Maka dapat diambil kesimpulan bahwa untuk simulasi ini, nilai *threshold* yang baik digunakan untuk mendapat hasil optimal ialah 0,7.

### 4.3 Analisis Penggunaan *Scale Invariant Feature Transform (SIFT)* Pada Pengenalan Jenis Kendaraan

Dari pembahasan yang telah dijabarkan pada bagian 4.2 dapat dilihat bahwa *SIFT* dapat bekerja dengan baik pada pengenalan jenis kendaraan. Akan tetapi perlu diperhatikan bahwa perbedaan keadaan lingkungan dan sudut pandang pengambilan citra dapat mempengaruhi proses pencocokan *keypoint*. Pencocokan *keypoint* yang dihasilkan oleh *SIFT* ini terlihat akan sangat baik jika digunakan pada citra yang sejenis. Sejenis maksudnya ialah objek yang berada pada citra basis data dan objek pada citra yang diuji merupakan satu objek yang sama meskipun terdapat perbedaan sudut pandang dan intensitas cahaya. Hal ini menunjukkan bahwa performa *SIFT* pada pengenalan kendaraan tetap sama seperti yang diungkapkan pada [4]. Dari sifatnya ini, dapat dinyatakan bahwa penggunaan *SIFT* sangat cocok jika digunakan pada aplikasi pengenalan jenis kendaraan yang ditujukan untuk fungsi keamanan, yang memerlukan pengenalan presisi untuk jenis kendaraan tertentu. Meski demikian tidak tertutup pula kemungkinan penggunaan *SIFT* untuk aplikasi pengenalan jenis kendaraan yang lebih umum, seperti pengenalan ditempat parkir atau pintu tol. Akan tetapi tentunya perlu dilakukan penelitian lebih lanjut untuk proses-proses yang perlu dilakukan agar ketepatan pengenalan berada diatas angka 90%.

Dari hasil simulasi yang telah dilakukan, maka metode *SIFT* yang mengambil fitur-fitur lokal sebagai *keypoint* dapat memberikan hasil pengenalan yang terpercaya. Hal ini dikarenakan fitur lokal yang dihasilkan invarian terhadap berbagai jenis perubahan. Akan tetapi, masalah yang dihadapi ialah ketika melakukan pencocokan antar masing-masing *keypoint* yang telah terdeteksi. Mungkin perlu dilakukan proses lebih lanjut atau metode lain selain *Nearest Neighbour* untuk melakukan pencocokan pada *keypoint*.

## BAB 5

### PENUTUP

#### 5.1 Kesimpulan

Dari hasil simulasi dan analisis, dapat disimpulkan:

- a. Metode *Scale Invariant Feature Transform* dapat digunakan pada pengenalan jenis kendaraan.
- b. Hanya dengan mengambil bagian depan sebagai fitur khas dari suatu kendaraan, pengenalan jenis kendaraan dapat dilakukan dengan baik.
- c. *Keypoint* yang dihasilkan melalui proses SIFT memiliki sifat yang sesuai dengan yang dijelaskan pada [4].
- d. SIFT bekerja optimal dan menunjukkan performa yang baik untuk jenis objek yang sama meski terdapat perbedaan pada citra.
- e. Refleksi yang ditimbulkan oleh kendaraan, serta perbedaan intensitas cahaya akan berpengaruh terhadap hasil pengenalan dan pencocokan *keypoint*.
- f. Besarnya nilai *threshold* akan berpengaruh terhadap akurasi dari pencocokan *keypoint* dan mempengaruhi hasil pengenalan secara keseluruhan.
- g. Untuk melihat akurasi dari pencocokan *keypoint* dapat digunakan variabel *recall* dan *1-precision* seperti yang diteliti oleh [9].
- h. Kemampuan pengenalan jenis kendaraan pada simulasi ini optimal pada nilai *threshold* 0,7 yaitu sebesar 76% akurat.
- i. Pada simulasi dengan menggunakan SIFT ini, waktu proses yang diperlukan untuk melakukan proses pengenalan relatif cepat yaitu berkisar pada nilai 1,2 detik.

## 5.2 Saran

Berikut beberapa saran dari penulis terkait dengan penelitian yang telah dilakukan:

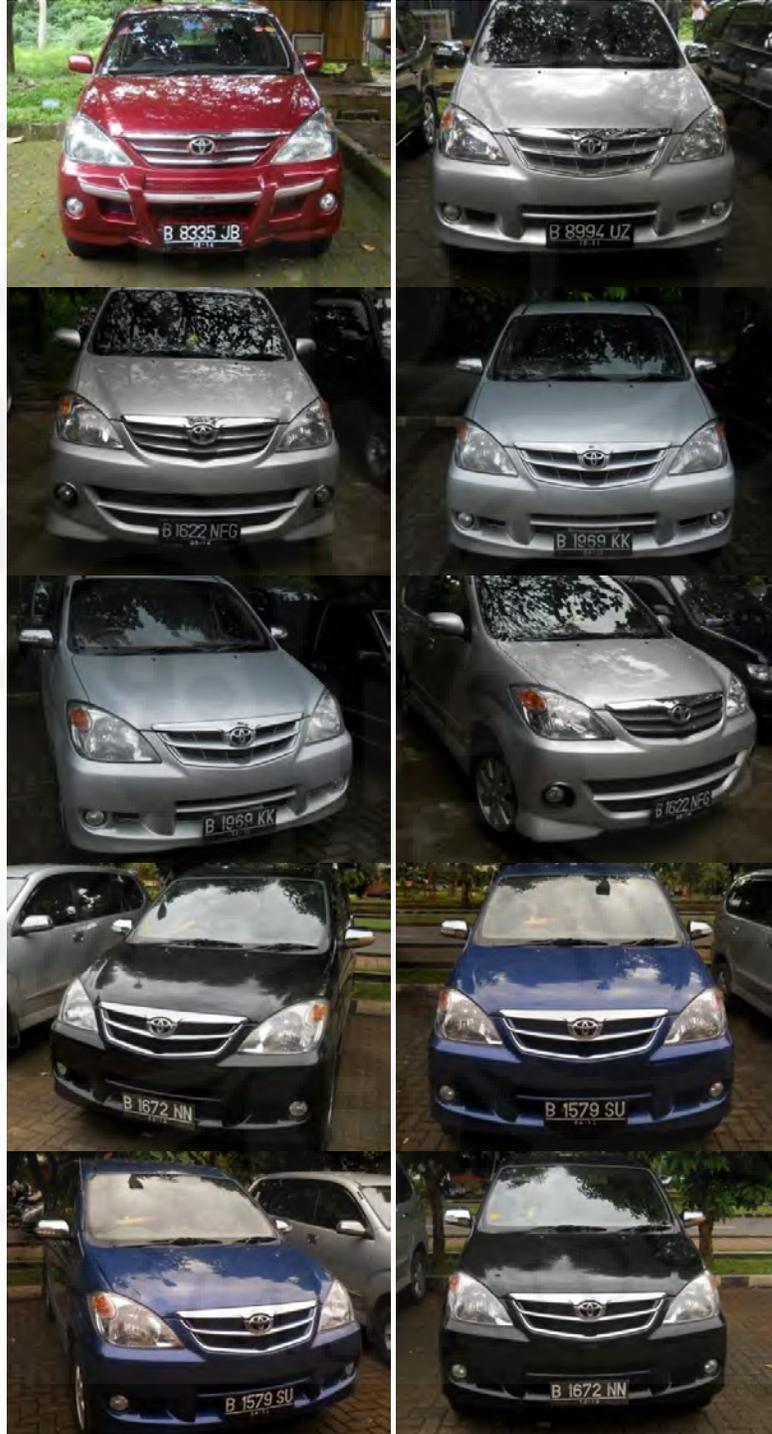
- a. Metode SIFT sebaiknya dilakukan untuk aplikasi pengenalan jenis kendaraan yang menitikberatkan pada fungsi keamanan.
- b. Sebaiknya dilakukan proses-proses tambahan agar pendeteksian dan pencocokan keypoint dapat berjalan lebih baik.
- c. Digunakan proses segmentasi citra yang lebih baik dalam menentukan dan mengambil *RoI* yang ingin digunakan.

## DAFTAR REFERENSI

- [1] Amit, Yali. (2002). *2D object recognition and detection: models, algorithms and networks*. Cambridge: MIT Press.
- [2] Nixon, Mark S., & Aguado, Alberto S. (2002). *Feature extraction and image processing*. Oxford: Newnes.
- [3] Lowe, David G. (1999, September). *Object recognition from local scale-invariant features*. International Conference on Computer Vision, Corfu.
- [4] Lowe, David G. (2004). *Distinctive image features from scale-invariant keypoint*. International Journal of Computer Vision.
- [5] Philips, Dwayne. (2000). *Image processing in C* (Electronic 2nd ed.). Kansas: Author.
- [6] Darma, Putra. (2010). *Pengolahan citra digital*. Yogyakarta: Penerbit Andi.
- [7] Gonzalez, Rafael C., Woods, Richard E., & Eddins, Steven L. (1999). *Digital image processing using MATLAB*. Upper Saddle River, NJ: Prentice Hall.
- [8] Thiang, Guntoro, Andre T., & Lim, Resmana. (2001, Maret). *Type of vehicle recognition using template matching method*. International Conference on Electrical, Electronics, Communication, and Information (CECI), Jakarta.
- [9] Psyllos, A., Anagnostopoulos, C. N., & Kayafas, E. (2009, September) *SIFT-based measurement for vehicle model recognition*. IMEKO XIX World Congress Fundamental and Applied Metrology, Lisbon. 2103 – 2108.
- [10] <http://www.cs.ubc.ca/~lowe/keypoints/>

## Lampiran 1: Citra Tes Yang Digunakan

Jenis Kendaraan Toyota Avanza



## Jenis Kendaraan Honda Jazz



## Jenis Kendaraan Toyota Kijang



## Jenis Kendaraan Isuzu Panther



## Jenis Kendaraan Daihatsu Terios



## Lampiran 2: Data Hasil Simulasi

Jenis Kendaraan Toyota Avanza

T = 0.5

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	tidak	0	0	0	265	1.505	0	0
2	tidak	0	0	0	265	1.274	0	0
3	tidak	0	0	0	265	0.951	0	0
4	ya	164	0	164	265	1.131	0	0.618868
5	ya	11	0	11	265	1.12	0	0.041509
6	tidak	0	0	0	265	1.019	0	0
7	tidak	0	0	0	265	1.177	0	0
8	tidak	0	0	0	265	1.099	0	0
9	tidak	0	0	0	265	1.057	0	0
10	ya	3	0	3	265	1.143	0	0.011321
rata-rata		17.8	0	17.8	265	<b>1.1476</b>	<b>0</b>	<b>0.06717</b>

T = 0.55

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	tidak	0	0	0	265	1.417	0	0
2	tidak	0	0	0	265	1.282	0	0
3	tidak	0	0	0	265	0.946	0	0
4	ya	170	0	170	265	1.034	0	0.641509
5	ya	11	1	12	265	1.172	0.083333	0.041509
6	tidak	0	0	0	265	1.15	0	0
7	tidak	0	0	0	265	1.206	0	0
8	tidak	0	0	0	265	1.176	0	0
9	tidak	0	0	0	265	1.176	0	0
10	ya	5	0	5	265	1.157	0	0.018868
rata-rata		18.6	0.1	18.7	265	<b>1.1716</b>	<b>0.005348</b>	<b>0.070189</b>

T = 0.6

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	tidak	0	0	0	265	1.409	0	0
2	salah	0	0	0	265	1.331	0	0
3	tidak	0	0	0	265	0.942	0	0
4	ya	174	0	174	265	1.036	0	0.656604

5	ya	19	1	20	265	1.053	0.05	0.071698
6	tidak	0	0	0	265	1.015	0	0
7	ya	5	0	5	265	1.125	0	0.018868
8	ya	8	0	8	265	1.129	0	0.030189
9	salah	0	0	0	265	1.102	0	0
10	ya	6	0	6	265	1.142	0	0.022642
rata-rata		21.2	0.1	21.3	265	<b>1.1284</b>	<b>0.004695</b>	<b>0.08</b>

T = 0.65

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	salah	0	0	0	265	1.46	0	0
2	salah	0	0	0	265	1.321	0	0
3	tidak	0	0	0	265	0.939	0	0
4	ya	179	0	179	265	1.026	0	0.675472
5	ya	24	2	26	265	1.066	0.076923	0.090566
6	tidak	0	0	0	265	1.013	0	0
7	ya	6	2	8	265	1.111	0.25	0.022642
8	ya	10	1	11	265	1.126	0.090909	0.037736
9	tidak	0	0	0	265	1.1	0	0
10	ya	12	2	14	265	1.141	0.142857	0.045283
rata-rata		23.1	0.7	23.8	265	<b>1.1303</b>	<b>0.029412</b>	<b>0.08717</b>

T = 0.7

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	salah	0	0	0	265	1.447	0	0
2	salah	0	0	0	265	1.441	0	0
3	ya	1	3	4	265	0.993	0.75	0.003774
4	ya	183	0	183	265	1.022	0	0.690566
5	ya	26	3	29	265	1.064	0.103448	0.098113
6	ya	1	6	7	265	1.044	0.857143	0.003774
7	ya	7	2	9	265	1.114	0.222222	0.026415
8	ya	14	3	17	265	1.124	0.176471	0.05283
9	ya	5	3	8	265	1.102	0.375	0.018868
10	ya	11	2	13	265	1.084	0.153846	0.041509
rata-rata		24.8	2.2	27	265	<b>1.1435</b>	<b>0.081481</b>	<b>0.093585</b>

T = 0.75

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	salah	0	0	0	265	1.456	0	0
2	salah	0	0	0	265	1.33	0	0
3	salah	0	0	0	265	0.986	0	0
4	ya	184	0	184	265	1.022	0	0.69434
5	ya	35	3	38	265	1.056	0.078947	0.132075
6	salah	0	0	0	265	1.049	0	0
7	ya	12	6	18	265	1.117	0.333333	0.045283
8	ya	20	8	28	265	1.125	0.285714	0.075472
9	ya	8	7	15	265	1.096	0.466667	0.030189
10	ya	13	4	17	265	1.155	0.235294	0.049057
rata-rata		27.2	2.8	30	265	<b>1.1392</b>	<b>0.093333</b>	<b>0.102642</b>

T = 0.8

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	8	19	27	265	1.456	0.703704	0.030189
2	salah	0	0	0	265	1.412	0	0
3	salah	0	0	0	265	0.983	0	0
4	ya	189	0	189	265	1.025	0	0.713208
5	ya	42	5	47	265	1.061	0.106383	0.158491
6	salah	0	0	0	265	1.048	0	0
7	ya	13	11	24	265	1.208	0.458333	0.049057
8	ya	39	11	50	265	1.132	0.22	0.14717
9	ya	9	14	23	265	1.184	0.608696	0.033962
10	salah	0	0	0	265	1.144	0	0
rata-rata		30	6	36	265	<b>1.1653</b>	<b>0.166667</b>	<b>0.113208</b>

Jenis Kendaraan Honda Jazz

T = 0.5

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	122	0	122	194	1.017	0	0.628866
2	tidak	0	0	0	194	0.966	0	0
3	tidak	0	0	0	194	0.985	0	0
4	ya	1	2	3	194	0.948	0.666667	0.005155
5	tidak	0	0	0	194	0.98	0	0

6	tidak	0	0	0	194	0.874	0	0
7	tidak	0	0	0	194	1.053	0	0
8	tidak	0	0	0	194	0.988	0	0
9	tidak	0	0	0	194	1.339	0	0
10	tidak	0	0	0	194	1.058	0	0
rata-rata		12.3	0.2	12.5	194	<b>1.0208</b>	<b>0.016</b>	<b>0.063402</b>

T = 0.55

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	126	0	126	194	1.049	0	0.649485
2	tidak	0	0	0	194	0.915	0	0
3	tidak	0	0	0	194	0.968	0	0
4	ya	1	4	5	194	0.95	0.8	0.005155
5	tidak	0	0	0	194	0.96	0	0
6	tidak	0	0	0	194	0.88	0	0
7	ya	0	6	6	194	1.078	1	0
8	tidak	0	0	0	194	0.91	0	0
9	tidak	0	0	0	194	1.304	0	0
10	tidak	0	0	0	194	1.056	0	0
rata-rata		12.7	1	13.7	194	<b>1.007</b>	<b>0.072993</b>	<b>0.065464</b>

T = 0.6

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	129	1	130	194	1.062	0.007692	0.664948
2	ya	5	0	5	194	1.006	0	0.025773
3	ya	2	1	3	194	1.015	0.333333	0.010309
4	ya	2	7	9	194	0.942	0.777778	0.010309
5	ya	3	1	4	194	1.012	0.25	0.015464
6	tidak	0	0	0	194	0.886	0	0
7	ya	6	0	6	194	1.075	0	0.030928
8	tidak	0	0	0	194	0.928	0	0
9	tidak	0	0	0	194	1.303	0	0
10	ya	2	1	3	194	1.111	0.333333	0.010309
rata-rata		14.9	1.1	16	194	<b>1.034</b>	<b>0.06875</b>	<b>0.076804</b>

T = 0.65

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	129	1	130	194	0.977	0.007692	0.664948

2	ya	6	0	6	194	1.066	0	0.030928
3	ya	2	1	3	194	1.008	0.333333	0.010309
4	ya	2	7	9	194	0.947	0.777778	0.010309
5	ya	4	2	6	194	1.022	0.333333	0.020619
6	ya	1	2	3	194	1.024	0.666667	0.005155
7	ya	0	7	7	194	1.083	1	0
8	tidak	0	0	0	194	0.904	0	0
9	ya	1	3	4	194	1.353	0.75	0.005155
10	ya	2	4	6	194	1.099	0.666667	0.010309
rata-rata		14.7	2.7	17.4	194	<b>1.0483</b>	<b>0.155172</b>	<b>0.075773</b>

T = 0.7

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	130	1	131	194	1.044	0.007634	0.670103
2	ya	8	0	8	194	0.999	0	0.041237
3	ya	2	1	3	194	1.008	0.333333	0.010309
4	ya	3	9	12	194	0.952	0.75	0.015464
5	ya	5	6	11	194	1.131	0.545455	0.025773
6	ya	1	4	5	194	0.933	0.8	0.005155
7	ya	0	10	10	194	1.091	1	0
8	ya	3	1	4	194	0.966	0.25	0.015464
9	salah	0	0	0	194	1.339	0	0
10	ya	3	7	10	194	1.103	0.7	0.015464
rata-rata		15.5	3.9	19.4	194	<b>1.0566</b>	<b>0.201031</b>	<b>0.079897</b>

T = 0.75

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	130	1	131	194	0.972	0.007634	0.670103
2	ya	12	1	13	194	0.964	0.076923	0.061856
3	ya	4	2	6	194	1.008	0.333333	0.020619
4	ya	5	15	20	194	0.939	0.75	0.025773
5	ya	6	8	14	194	1.119	0.571429	0.030928
6	ya	1	6	7	194	0.917	0.857143	0.005155
7	salah	0	0	0	194	1.181	0	0
8	ya	4	4	8	194	0.957	0.5	0.020619
9	ya	2	10	12	194	1.359	0.833333	0.010309
10	ya	6	8	14	194	1.099	0.571429	0.030928
rata-rata		17	5.5	22.5	194	<b>1.0515</b>	<b>0.244444</b>	<b>0.087629</b>

T = 0.8

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	131	2	133	194	0.965	0.015038	0.675258
2	ya	19	6	25	194	0.982	0.24	0.097938
3	ya	5	8	13	194	1.031	0.615385	0.025773
4	ya	7	18	25	194	1.026	0.72	0.036082
5	ya	10	12	22	194	1.026	0.545455	0.051546
6	salah	0	0	0	194	0.962	0	0
7	salah	0	0	0	194	1.092	0	0
8	ya	5	9	14	194	0.965	0.642857	0.025773
9	ya	11	21	32	194	1.362	0.65625	0.056701
10	ya	10	10	20	194	1.112	0.5	0.051546
rata-rata		19.8	8.6	28.4	194	<b>1.0523</b>	<b>0.302817</b>	<b>0.102062</b>

## Jenis Kendaraan Daihatsu Terios

T = 0.5

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	167	0	167	263	1.096	0	0.634981
2	tidak	0	0	0	263	1.199	0	0
3	tidak	0	0	0	263	1.243	0	0
4	ya	3	0	3	263	1.565	0	0.011407
5	tidak	0	0	0	263	1.485	0	0
6	tidak	0	0	0	263	1.108	0	0
7	tidak	0	0	0	263	1.143	0	0
8	tidak	0	0	0	263	1.194	0	0
9	salah	0	0	0	263	1.534	0	0
10	tidak	0	0	0	263	1.284	0	0
rata-rata		17	0	17	263	<b>1.2851</b>	<b>0</b>	<b>0.064639</b>

T = 0.55

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	170	0	170	263	1.11	0	0.646388
2	tidak	0	0	0	263	1.306	0	0
3	tidak	0	0	0	263	1.245	0	0
4	ya	6	0	6	263	1.549	0	0.022814
5	tidak	0	0	0	263	1.461	0	0
6	tidak	0	0	0	263	1.104	0	0

7	tidak	0	0	0	263	1.221	0	0
8	tidak	0	0	0	263	1.16	0	0
9	salah	0	0	0	263	1.515	0	0
10	tidak	0	0	0	263	1.382	0	0
rata-rata		17.6	0	17.6	263	<b>1.3053</b>	<b>0</b>	<b>0.06692</b>

T = 0.6

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	174	0	174	263	1.096	0	0.661597
2	tidak	0	0	0	263	1.203	0	0
3	salah	0	0	0	263	1.403	0	0
4	ya	7	0	7	263	1.545	0	0.026616
5	ya	4	3	7	263	1.516	0.428571	0.015209
6	tidak	0	0	0	263	1.099	0	0
7	tidak	0	0	0	263	1.135	0	0
8	tidak	0	0	0	263	1.173	0	0
9	salah	0	0	0	263	1.524	0	0
10	salah	0	0	0	263	1.333	0	0
rata-rata		18.5	0.3	18.8	263	<b>1.3027</b>	<b>0.015957</b>	<b>0.070342</b>

T = 0.65

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	176	0	176	263	1.106	0	0.669202
2	tidak	0	0	0	263	1.207	0	0
3	ya	2	2	4	263	1.302	0.5	0.007605
4	ya	9	1	10	263	1.551	0.1	0.034221
5	ya	4	6	10	263	1.507	0.6	0.015209
6	salah	0	0	0	263	1.148	0	0
7	tidak	0	0	0	263	1.14	0	0
8	tidak	0	0	0	263	1.171	0	0
9	salah	0	0	0	263	1.528	0	0
10	salah	0	0	0	263	1.336	0	0
rata-rata		19.1	0.9	20	263	<b>1.2996</b>	<b>0.045</b>	<b>0.072624</b>

T = 0.7

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	178	0	178	263	1.115	0	0.676806
2	ya	4	3	7	263	1.251	0.428571	0.015209

3	ya	3	4	7	263	1.293	0.571429	0.011407
4	ya	13	4	17	263	1.558	0.235294	0.04943
5	salah	0	0	0	263	1.5	0	0
6	salah	0	0	0	263	1.139	0	0
7	ya	1	5	6	263	1.291	0.833333	0.003802
8	salah	0	0	0	263	1.238	0	0
9	salah	0	0	0	263	1.537	0	0
10	salah	0	0	0	263	1.347	0	0
rata-rata		19.9	1.6	21.5	263	<b>1.3269</b>	<b>0.074419</b>	<b>0.075665</b>

T = 0.75

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	179	0	179	263	1.169	0	0.680608
2	salah	0	0	0	263	1.248	0	0
3	ya	6	17	23	263	1.3	0.73913	0.022814
4	ya	24	9	33	263	1.552	0.272727	0.091255
5	ya	7	15	22	263	1.51	0.681818	0.026616
6	salah	0	0	0	263	1.174	0	0
7	ya	6	6	12	263	1.502	0.5	0.022814
8	salah	0	0	0	263	1.22	0	0
9	ya	17	17	34	263	1.527	0.5	0.064639
10	salah	0	0	0	263	1.338	0	0
rata-rata		23.9	6.4	30.3	263	<b>1.354</b>	<b>0.211221</b>	<b>0.090875</b>

T = 0.8

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	183	1	184	263	1.117	0.005435	0.695817
2	salah	0	0	0	263	1.24	0	0
3	ya	12	28	40	263	1.308	0.7	0.045627
4	ya	40	17	57	263	1.546	0.298246	0.152091
5	ya	17	23	40	263	1.537	0.575	0.064639
6	salah	0	0	0	263	1.165	0	0
7	ya	8	9	17	263	1.27	0.529412	0.030418
8	salah	0	0	0	263	1.233	0	0
9	ya	33	30	63	263	1.53	0.47619	0.125475
10	ya	31		31	263	1.448	0	0.117871
rata-rata		32.4	12	43.2	263	<b>1.3394</b>	<b>0.277778</b>	<b>0.123194</b>

## Jenis Kendaraan Isuzu Panther

T = 0.5

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	1	3	4	237	1.351	0.75	0.004219
2	tidak	0	0	0	237	0.926	0	0
3	tidak	0	0	0	237	1.197	0	0
4	ya	171	0	171	237	1.17	0	146.1538
5	ya	7	0	7	237	1.18	0	0.029536
6	ya	6	0	6	237	1.193	0	0.025316
7	tidak	0	0	0	237	0.963	0	0
8	tidak	0	0	0	237	1.345	0	0
9	tidak	0	0	0	237	1.222	0	0
10	tidak	0	0	0	237	0.765	0	0
rata-rata		18.5	0.3	18.8	237	<b>1.1312</b>	<b>0.015957</b>	<b>0.078059</b>

T = 0.55

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	3	3	6	237	1.259	0.5	0.012658
2	tidak	0	0	0	237	0.937	0	0
3	tidak	0	0	0	237	1.203	0	0
4	ya	177	0	177	237	1.159	0	0.746835
5	ya	8	0	8	237	1.176	0	0.033755
6	ya	7	0	7	237	1.182	0	0.029536
7	tidak	0	0	0	237	0.857	0	0
8	tidak	0	0	0	237	1.252	0	0
9	tidak	0	0	0	237	1.197	0	0
10	tidak	0	0	0	237	0.848	0	0
rata-rata		19.5	0.3	19.8	237	<b>1.107</b>	<b>0.015152</b>	<b>0.091369</b>

T = 0.6

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	4	7	11	237	1.304	0.636364	0.016878
2	tidak	0	0	0	237	0.924	0	0
3	tidak	0	0	0	237	1.203	0	0
4	ya	179	0	179	237	1.138	0	0.755274
5	ya	16	0	16	237	1.177	0	0.067511
6	ya	10	0	10	237	1.187	0	0.042194

7	ya	2	1	3	237	0.907	0.333333	0.008439
8	salah	0	0	0	237	1.323	0	0
9	tidak	0	0	0	237	1.183	0	0
10	salah	0	0	0	237	0.819	0	0
rata-rata		21.1	0.8	21.9	237	<b>1.1165</b>	<b>0.03653</b>	<b>0.08903</b>

T = 0.65

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	6	10	16	237	1.282	0.625	0.025316
2	tidak	0	0	0	237	0.92	0	0
3	salah	0	0	0	237	1.25	0	0
4	ya	183	0	183	237	1.137	0	0.772152
5	ya	18	0	18	237	1.197	0	0.075949
6	ya	16	0	16	237	1.2	0	0.067511
7	salah	0	0	0	237	0.909	0	0
8	salah	0	0	0	237	1.3	0	0
9	salah	0	0	0	237	1.231	0	0
10	ya	5		5	237	0.839	0	0.021097
rata-rata		22.8	1.111111	23.8	237	<b>1.1265</b>	<b>0.046685</b>	<b>0.096203</b>

T = 0.7

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	8	13	21	237	1.296	0.619048	0.033755
2	salah	0	0	0	237	1.062	0	0
3	salah	0	0	0	237	1.242	0	0
4	ya	185	0	185	237	1.141	0	0.780591
5	ya	24	3	27	237	1.319	0.111111	0.101266
6	ya	22	0	22	237	1.176	0	0.092827
7	ya	3	3	6	237	0.923	0.5	0.012658
8	salah	0	0	0	237	1.31	0	0
9	salah	0	0	0	237	1.238	0	0
10	ya	7		7	237	0.895	0	0.029536
rata-rata		24.9	2.111111	26.8	237	<b>1.1602</b>	<b>0.078773</b>	<b>0.105063</b>

T = 0.75

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	salah	0	0	0	237	1.32	0	0
2	salah	0	0	0	237	0.978	0	0

3	salah	0	0	0	237	1.256	0	0
4	ya	190	1	191	237	1.128	0.005236	0.801688
5	ya	27	8	35	237	1.2	0.228571	0.113924
6	ya	30	2	32	237	1.202	0.0625	0.126582
7	salah	0	0	0	237	0.896	0	0
8	ya	8	6	14	237	1.313	0.428571	0.033755
9	salah	0	0	0	237	1.241	0	0
10	ya	12		12	237	0.815	0	0.050633
rata-rata		26.7	1.888889	28.4	237	<b>1.1349</b>	<b>0.06651</b>	<b>0.112658</b>

T = 0.8

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	salah	0	0	0	237	1.309	0	0
2	salah	0	0	0	237	0.989	0	0
3	ya	8	18	26	237	1.34	0.692308	0.033755
4	ya	192	2	194	237	1.136	0.010309	0.810127
5	ya	36	14	50	237	1.209	0.28	0.151899
6	ya	38	9	47	237	1.198	0.191489	0.160338
7	salah	0	0	0	237	0.9	0	0
8	ya	14	16	30	237	1.294	0.533333	0.059072
9	ya	20	14	34	237	1.24	0.411765	0.084388
10	ya	25		25	237	0.829	0	0.105485
rata-rata		33.3	8.111111	40.6	237	<b>1.1444</b>	<b>0.199781</b>	<b>0.140506</b>

Jenis Kendaraan Toyota Kijang

T = 0.5

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	tidak	0	0	0	320	1.606	0	0
2	ya	7	0	7	320	1.843	0	0.021875
3	ya	262	0	262	320	1.315	0	0.81875
4	ya	17	1	18	320	1.185	0.055556	0.053125
5	ya	8	1	9	320	1.411	0.111111	0.025
6	tidak	0	0	0	320	1.24	0	0
7	tidak	0	0	0	320	1.225	0	0
8	ya	16	0	16	320	1.204	0	0.05
9	ya	12	0	12	320	1.691	0	0.0375
10	ya	11	0	11	320	1.49	0	0.034375
rata-rata		33.3	0.2	33.5	320	<b>1.421</b>	<b>0.00597</b>	<b>0.104063</b>

T = 0.55

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	tidak	0	0	0	320	1.686	0	0
2	ya	11	0	11	320	1.686	0	0.034375
3	ya	265	0	265	320	1.407	0	0.828125
4	ya	27	1	28	320	1.179	0.035714	0.084375
5	ya	14	1	15	320	1.406	0.066667	0.04375
6	ya	2	1	3	320	1.281	0.333333	0.00625
7	tidak	0	0	0	320	1.233	0	0
8	ya	28	0	28	320	1.217	0	0.0875
9	ya	19	0	19	320	1.685	0	0.059375
10	ya	16	0	16	320	1.479	0	0.05
rata-rata		38.2	0.3	38.5	320	<b>1.4259</b>	<b>0.007792</b>	<b>0.132583</b>

T = 0.6

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	4	2	6	320	1.85	0.333333	0.0125
2	ya	17	0	17	320	1.668	0	0.053125
3	ya	271	0	271	320	1.32	0	0.846875
4	ya	38	1	39	320	1.176	0.025641	0.11875
5	ya	22	1	23	320	1.4	0.043478	0.06875
6	ya	7	1	8	320	1.281	0.125	0.021875
7	ya	3	1	4	320	1.279	0.25	0.009375
8	ya	38	0	38	320	1.208	0	0.11875
9	ya	23	1	24	320	1.687	0.041667	0.071875
10	ya	23	0	23	320	1.482	0	0.071875
rata-rata		44.6	0.7	45.3	320	<b>1.4351</b>	<b>0.015453</b>	<b>0.139375</b>

T = 0.65

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	6	3	9	320	1.708	0.333333	0.01875
2	ya	21	3	24	320	1.687	0.125	0.065625
3	ya	277	0	277	320	1.318	0	0.865625
4	ya	48	3	51	320	1.17	0.058824	0.15
5	ya	26	2	28	320	1.401	0.071429	0.08125
6	ya	11	1	12	320	1.287	0.083333	0.034375
7	ya	5	1	6	320	1.27	0.166667	0.015625

8	ya	53	1	54	320	1.219	0.018519	0.165625
9	ya	29	2	31	320	1.677	0.064516	0.090625
10	ya	33	0	33	320	1.486	0	0.103125
rata-rata		50.9	1.6	52.5	320	<b>1.4223</b>	<b>0.030476</b>	<b>0.159063</b>

T = 0.7

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	9	9	18	320	1.699	0.5	0.028125
2	ya	25	6	31	320	1.696	0.193548	0.078125
3	ya	280	0	280	320	1.329	0	0.875
4	ya	60	5	65	320	1.214	0.076923	0.1875
5	ya	36	2	38	320	1.396	0.052632	0.1125
6	ya	16	3	19	320	1.376	0.157895	0.05
7	ya	6	6	12	320	1.281	0.5	0.01875
8	ya	65	1	66	320	1.223	0.015152	0.203125
9	ya	37	2	39	320	1.675	0.051282	0.115625
10	ya	42	0	42	320	1.475	0	0.13125
rata-rata		57.6	3.4	61	320	<b>1.4364</b>	<b>0.055738</b>	<b>0.18</b>

T = 0.75

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	13	13	26	320	1.724	0.5	0.040625
2	ya	28	12	40	320	1.688	0.3	0.0875
3	ya	283	4	287	320	1.323	0.013937	0.884375
4	ya	79	5	84	320	1.184	0.059524	0.246875
5	ya	45	4	49	320	1.402	0.081633	0.140625
6	ya	24	5	29	320	1.31	0.172414	0.075
7	ya	14	8	22	320	1.285	0.363636	0.04375
8	ya	76	4	80	320	1.223	0.05	0.2375
9	ya	46	9	55	320	1.68	0.163636	0.14375
10	ya	52	3	55	320	1.494	0.054545	0.1625
rata-rata		66	6.7	72.7	320	<b>1.4313</b>	<b>0.09216</b>	<b>0.20625</b>

T = 0.8

No	Dikenali?	correct matches	false matches	total matches	correspondences	waktu proses (detik)	1-precision	recall
1	ya	30	20	50	320	1.832	0.4	0.09375
2	ya	40	23	63	320	1.676	0.365079	0.125
3	ya	286	6	292	320	1.314	0.020548	0.89375

4	ya	94	12	106	320	1.184	0.113208	0.29375
5	ya	56	10	66	320	1.392	0.151515	0.175
6	ya	39	12	51	320	1.293	0.235294	0.121875
7	ya	27	12	39	320	1.365	0.307692	0.084375
8	ya	92	7	99	320	1.245	0.070707	0.2875
9	ya	54	20	74	320	1.712	0.27027	0.16875
10	ya	61	8	69	320	1.487	0.115942	0.190625
rata-rata		77.9	13	90.9	320	<b>1.45</b>	<b>0.143014</b>	<b>0.243438</b>



### Lampiran 3: Listing Kode Program

Berikut kode program pada file *sift\_train.m* yang berfungsi untuk mendeteksi deskriptor pada citra training yang digunakan sebagai basis data

```
% [image, descriptors, locs] = sift(imageFile)
%
% This function reads an image and returns its SIFT keypoints.
% Input parameters:
%   imageFile: the file name for the image.
%
% Returned:
%   image: the image array in double format
%   descriptors: a K-by-128 matrix, where each row gives an
invariant
%   descriptor for one of the K keypoints. The descriptor
is a vector
%   of 128 values normalized to unit length.
%   locs: K-by-4 matrix, in which each row has the 4 values for
a
%   keypoint location (row, column, scale, orientation).
The
%   orientation is in the range [-PI, PI] radians.
%
% Credits: Thanks for initial version of this program to D. Alvaro
% and J.J. Guerrero, Universidad de Zaragoza (modified by D. Lowe)

function [image, descriptors, locs] = sift_train(imageFile)

% Load image
image = imread(imageFile);

% If you have the Image Processing Toolbox, you can uncomment the
following
% lines to allow input of color images, which will be converted
to grayscale.
if isrgb(image)
    image = rgb2gray(image);
end
[rows, cols] = size(image);

% Convert into PGM imagefile, readable by "keypoints" executable
f = fopen('tmp.pgm', 'w');
if f == -1
    error('Could not create file tmp.pgm.');
```

```
end
fprintf(f, 'P5\n%d\n%d\n255\n', cols, rows);
fwrite(f, image, 'uint8');
fclose(f);

% Call keypoints executable
if isunix
    command = '!./sift ';
else
    command = '!siftWin32 ';
```

```

end
command = [command ' <tmp.pgm >tmp.key'];
eval(command);

% Open tmp.key and check its header
g = fopen('tmp.key', 'r');
if g == -1
    error('Could not open file tmp.key.');
```

\* PERPUSTAKAAN FAKULTAS TEKNIK \*  
UNIVERSITAS INDONESIA

```

end
[header, count] = fscanf(g, '%d %d', [1 2]);
if count ~= 2
    error('Invalid keypoint file beginning.');
```

```

end
num = header(1);
len = header(2);
if len ~= 128
    error('Keypoint descriptor length invalid (should be 128).');
```

```

end

% Creates the two output matrices (use known size for efficiency)
locs = double(zeros(num, 4));
descriptors = double(zeros(num, 128));

% Parse tmp.key
for i = 1:num
    [vector, count] = fscanf(g, '%f %f %f %f', [1 4]); %row col
    scale ori
    if count ~= 4
        error('Invalid keypoint file format');
```

```

    end
    locs(i, :) = vector(1, :);

    [descrip, count] = fscanf(g, '%d', [1 len]);
    if (count ~= 128)
        error('Invalid keypoint file value.');
```

```

    end
    % Normalize each input vector to unit length
    descrip = descrip / sqrt(sum(descrip.^2));
    descriptors(i, :) = descrip(1, :);
end
fclose(g);

```

Berikut kode program pada file *skripsi\_train.m* yang berfungsi untuk mengambil deskriptor citra training dan menyimpannya pada file *train.mat*

```

clear all;
clc;

x = 5;

for counter=1:x
    switch counter
        case 1
            [imtr1,destr1,loctr1] = sift_train('tr1.jpg');
        case 2
            [imtr2,destr2,loctr2] = sift_train('tr2.jpg');
```

```

        case 3
            [imtr3,destr3,loctr3] = sift_train('tr3.jpg');
        case 4
            [imtr4,destr4,loctr4] = sift_train('tr4.jpg');
        case 5
            [imtr5,destr5,loctr5] = sift_train('tr5.jpg');
        otherwise
            display('training image tidak ditemukan');
        end
    end
end

save train.mat;

```

Berikut kode program pada file *sift.m* yang berfungsi untuk mendeteksi *RoI* dan mengambil deskriptor dari citra tes.

```

function [image, descriptors, locs] = sift(imageFile)

% Load image
image = imread(imageFile);
image = imcrop(image,[37.5 183.5 571 220]); %Deteksi RoI

if isrgb(image)
    image = rgb2gray(image);
end
[rows, cols] = size(image);

% Convert into PGM imagefile, readable by "keypoints" executable
f = fopen('tmp.pgm', 'w');
if f == -1
    error('Could not create file tmp.pgm.');
```

```

end
fprintf(f, 'P5\n%d\n%d\n255\n', cols, rows);
fwrite(f, image, 'uint8');
fclose(f);

% Call keypoints executable
if isunix
    command = '!./sift ';
else
    command = '!siftWin32 ';
end
command = [command ' <tmp.pgm >tmp.key'];
eval(command);

% Open tmp.key and check its header
g = fopen('tmp.key', 'r');
if g == -1
    error('Could not open file tmp.key.');
```

```

end
[header, count] = fscanf(g, '%d %d', [1 2]);
if count ~= 2
    error('Invalid keypoint file beginning.');
```

```

end
num = header(1);
len = header(2);

```

```

if len ~= 128
    error('Keypoint descriptor length invalid (should be 128).');
end

% Creates the two output matrices (use known size for efficiency)
locs = double(zeros(num, 4));
descriptors = double(zeros(num, 128));

% Parse tmp.key
for i = 1:num
    [vector, count] = fscanf(g, '%f %f %f %f', [1 4]); %row col
    scale ori
    if count ~= 4
        error('Invalid keypoint file format');
    end
    locs(i, :) = vector(1, :);

    [descrip, count] = fscanf(g, '%d', [1 len]);
    if (count ~= 128)
        error('Invalid keypoint file value. ');
    end
    % Normalize each input vector to unit length
    descrip = descrip / sqrt(sum(descrip.^2));
    descriptors(i, :) = descrip(1, :);
end
fclose(g);

```

Berikut kode program pada file *skripsi.m* yang berfungsi untuk memanggil file *sift.m* dan melakukan pencocokan antara deskriptor citra tes dan citra trainig.

```

function skripsi(imagel, T)
tic;
load('train.mat');

x = 5; %jumlah train image pada database
num2 = 0;
y=0;

[im1, des1, loc1] = sift(imagel);

distRatio = T;

for counter=1:x
    switch counter
        case 1
            des2=destr1;
        case 2
            des2=destr2;
        case 3
            des2=destr3;
        case 4
            des2=destr4;
        case 5
            des2=destr5;
        otherwise
            display('training image tidak ditemukan');
    end
end

```

```

end

des2t = des2';
for i = 1 : size(des1,1)
    dotprods = des1(i,:) * des2t;
    [vals,indx] = sort(acos(dotprods));

    if (vals(1) < distRatio * vals(2))
        match(i) = indx(1);
    else
        match(i) = 0;
    end
end

num = sum(match > 0);

if num>num2
    num2 = num;
    y = counter;
    for i = 1 : size(des1,1)
        match2(i) = match(i);
    end
end
end

if num2 > 2
    switch y
        case 1
            im2=imtr1;loc2=loctr1;
        case 2
            im2=imtr2;loc2=loctr2;
        case 3
            im2=imtr3;loc2=loctr3;
        case 4
            im2=imtr4;loc2=loctr4;
        case 5
            im2=imtr5;loc2=loctr5;
        otherwise
            display('training image tidak ditemukan');
    end

    % Create a new image showing the two images side by side.
    im3 = appendimages(im1,im2);

    % Show a figure with lines joining the accepted matches.
    figure('Position', [100 100 size(im3,2) size(im3,1)]);
    colormap('gray');
    imagesc(im3);
    hold on;
    cols1 = size(im1,2);
    for i = 1: size(des1,1)
        if (match2(i) > 0)
            line([loc1(i,2) loc2(match2(i),2)+cols1], ...
                [loc1(i,1) loc2(match2(i),1)], 'Color', 'c');
        end
    end
    hold off;

```

```
fprintf('\n Found %d matches.\n\n', num2);

if num2>2
    switch y
        case 1
            display('Jenis Mobil: Honda Jazz');
        case 2
            display('Jenis Mobil: Daihatsu Terios');
        case 3
            display('Jenis Mobil: Toyota Avanza');
        case 4
            display('Jenis Mobil: Isuzu Panther');
        case 5
            display('Jenis Mobil: Toyota Kijang');
        otherwise
            display('tidak ada gambar yang sesuai');
        end
    else
        display('tidak ada gambar yang sesuai');
    end
    fprintf('\nWaktu Proses: %.3f detik\n\n',toc);
else
    display('tidak ada gambar yang sesuai');
    fprintf('\nWaktu Proses: %.3f detik\n\n',toc);
end
end
```