# BAB 3

# ANALISIS DAN PERANCANGAN

## 3.1 Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya untuk mengidentifikasikan masalah-masalah, dan hambatan-hambatan sehingga dapat diusulkan kebutuhan-kebutuhan untuk perbaikannya. Melakukan analisis terhadap sistem yang sedang berjalan bertujuan sebagai dasar perancangan atau perbaikan sistem yang lama. Dan dari hasil analisis tersebut dapat diketahui kelemahan dan kekurangan sistem yang lama dan dapat dirancang atau diperbaiki menjadi sebuah sistem yang lebih efektif dan efisien.

### 3.1.1 Analisis Masalah

Berdasarkan data yang diperoleh dari hasil penelitian dan observasi langsung terhadap pihak Dinas Perhubungan Kota Cikampek, didapati beberapa masalah yang masih dihadapi Dinas Perhubungan Kota Cikampek dalam melakukan pengumpulan data jumlah kendaraan yang melintas, yaitu :

1. Cara penghitungan jumlah kendaraan yang melintas masih menggunakan cara yang manual yaitu dengan menugaskan beberapa orang petugas di lapangan, untuk menghitung setiap kendaraan yang lewat.
2. Pemborosan waktu dalam pembuatan laporan dikarenakan penghitungan jumlah kendaraan dengan menggunakan cara manual akan memakan waktu yang cukup lama dalam pembuatan laporan.
3. Kemungkinan terjadinya kesalahan perhitungan jumlah kendaraan cukup tinggi.
4. Keterlambatan petugas dalam mengalihkan kendaraan ke jalur alternatif.

### 3.1.2 Analisis Prosedur Yang Sedang Berjalan

Dari hasil observasi dan wawancara yang telah dilakukan kepada pihak Dinas Pehubungan Kota Cikampek didapati sistem penghitungan jumlah kendaraan yang dilakukan oleh pihak Dinas Perhubungan Kota Cikampek masih menggunakan cara manual yaitu :

1. Pihak Dinas Perhubungan menugaskan beberapa orang petugas di lapangan ( pos penelitian )
2. Petugas tersebut mencatat dan menghitung setiap jumlah kendaraan yang melintas pada jalan tersebut.

3. Kemudian dilakukan proses penjumlahan dengan perolehan data dari petugas yang lain.
4. Hasil penjumlahan tadi di rekap ulang sebagai data pegangan petugas.
5. Data hasil penjumlahan dan perakapan tadi dilaporkan ke kantor untuk dimasukan ke dalam buku besar.

### 3.1.3 Solusi Yang Ditawarkan

Untuk mengurangi permasalahan-permasalahan yang ada di Dinas Perhubungan Kota Cikampek, maka ditawarkan sebuah solusi yaitu dengan Membangun suatu perangkat lunak yang dapat melakukan penghitungan jumlah kendaraan dengan menggunakan kamera sebagai pengganti fungsi mata manusia sebagai pengawas.

Adapun keuntungan dari solusi yang ditawarkan oleh penulis adalah sebagai berikut:

1. Aplikasi perangkat lunak yang akan dibangun tersebut diharapkan dapat mengatasi masalah penghitungan jumlah kendaraan yang masih dilakukan dengan cara manual.
2. Dinas Perhubungan Kota Cikampek dapat mengantisipasi terjadinya penumpukan kendaraan.

3. Dinas Perhubungan Kota Cikampek dapat mengambil keputusan dalam pengalihan jalur kendaraan ke jalur alternatif tanpa menunggu terjadinya penumpukan kendaraan terlebih dahulu.
4. Memberikan kemudahan kepada pegawai Dinas Perhubungan Kota Cikampek dalam melakukan penghitungan jumlah kendaraan.

### 3.1.4 Analisis Metode Untuk Penyelesaian Masalah

Tahapan analisis masalah metode untuk penyelesaian masalah adalah tahapan yang digunakan untuk mengetahui proses-proses yang digunakan dalam metode yang penulis gunakan dalam pembangunan aplikasi.

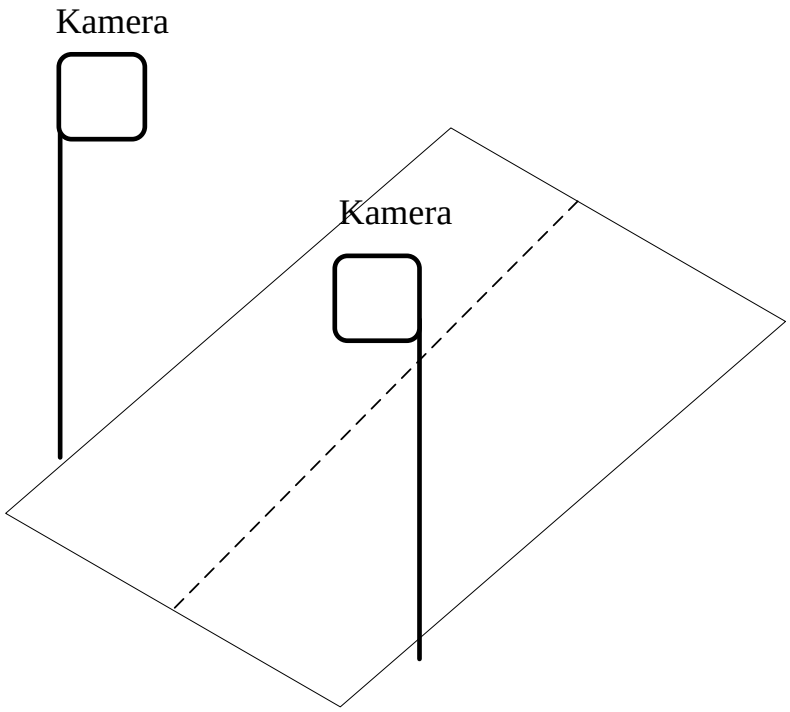
Metode utama yang digunakan dalam pembangunan perangkat lunak ini adalah metode *Normalized Sum Squared Differences* (NSSD), metode ini digunakan sebagai penghitung jumlah kendaraan yang melintas. Sebelum melakukan proses penghitungan terlebih dahulu dilakukan proses pengolahan citra terhadap data video yang menjadi bahan *input*an untuk perangkat lunak ini. Adapun tahapan dari metode *Normalized Sum Squared Differences* (NSSD) sendiri adalah sebagai berikut:

1. *Capture* Kamera
2. *Setting Detection Window*

3. *Grayscaling*
4. *Filtering*
5. *Tresholding*
6. Penghitungan kendaraan

#### 3.1.4.1 Proses Capture Kamera

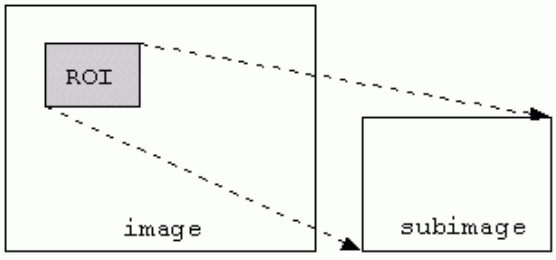
Proses *capture* ini merupakan proses yang sangat penting, karena melalui proses inilah data *input* akan didapatkan untuk dilakukan proses penghitungan. Pada system *realtime* proses peng-*capture*-an sendiri akan terus dilakukan walaupun jalan dalam keadaan kosong. Kecepatan kamera untuk proses peng-*capture*-an yang digunkan adalah 25fps. Adapun posisi untuk penempatan kamera yang digunakan dalam penelitian ini adalah berada di tepi sisi ruas jalan dengan ketinggian yang telah di tentukan dan kemiringan sekitar 75 derajat, untuk lebih jelasnya dapat dilihat dari gambar berikut :




**Gambar 3.3.1 Posisi Penempatan Kamera**

### 3.1.4.2 Proses Setting Detection Window

Setelah proses *capture* kamera proses selanjutnya adalah proses *setting Detection Window* (ROI), *Detection Window* (ROI) adalah daerah persegi panjang pada gambar untuk memproses gambar lebih lanjut.



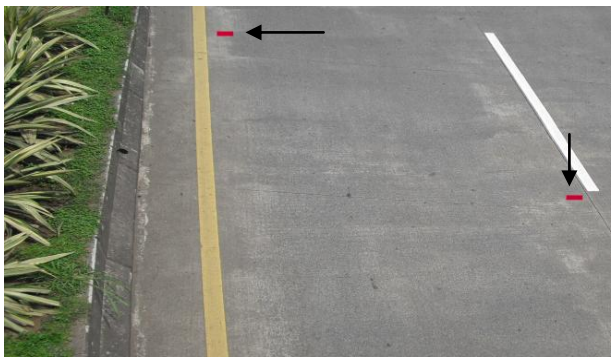

**Gambar 3.3.2 *Detection Window* (ROI)**

Dalam pembuatan *detection window* (ROI) memiliki 2 tahapan proses yang harus dilakukan untuk dapat menggambarkan posisi *detection window* (ROI) yang akan digunakan pada *frame* dan *background*. Adapun tahapan dari proses pembuatan *detection window* (ROI) adalah sebagai berikut :

1. Menentukan titik-titik sudut

Menentukan titik sudut adalah proses awal yang akan digunakan untuk pembuatan Detection window (ROI) itu sendiri. Cara untuk menentukan koordinat titik pikselnya sendiri adalah dengan mengambil posisi nilai default yang disediakan oleh tools developer yang digunakan yang kemudian di geser untuk mencari posisi yang pas.

Contoh penetuan titik adalah sebagai berikut :

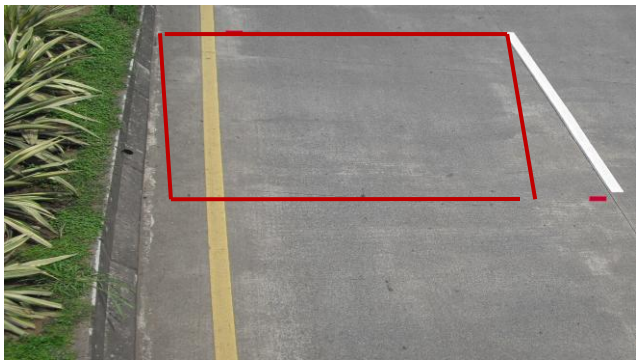


**Gambar 3.3.3 Penentuan Titik Untuk Pembuatan Detection Window (ROI)**

2. Membuat gambar persegi panjang

Setelah melakukan proses menetukan titik, proses selanjutnya dalam pembuatan *detection window* (ROI) adalah proses pembutan gambar persegi panjang dengan dua titik yang telah ditentukan tersebut.

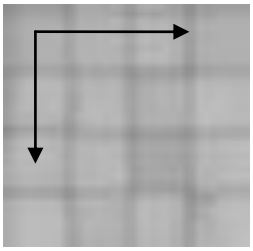
Contoh pembuatan persegi panjang dapat dilihat pada gambar berikut :



**Gambar 3.3.4 Membuat Gambar Persegi Panjang Untuk Pembuatan *Detection Window* (ROI)**

3. Penghitungan jumlah piksel dalam *Detection Window* (ROI)

Sebenarnya proses ini tidak termasuk dalam proses pembuatan *detection window* (ROI), akan tetapi karena dalam penggunaan rumus perhitungan *Normalized Sum Squared Differences* (NSSD) perlu diketahui jumlah piksel yang ada dalam luasan *detection window* (ROI) maka diperlukan proses ini.



**Gambar 3.3.5 Jumlah Piksel Dalam D*etection Window* (ROI)**

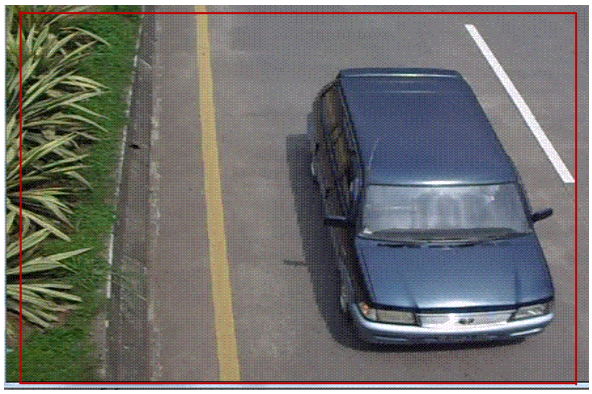
Dalam penelitian ini hanya terdapat satu *detection window* yang menjadi daerah pembanding untuk dilakukan proses penghitungan perbedaan jumlah piksel, luasan dari *detection window* disesuaikan dengan luasan image yang tercapture oleh kamera.



**Gambar 3.3.6 Contoh Penggunaan *Detection Window (ROI)* Pada Penelitian**

Selain membuat *detection window* (ROI) pada *background*, *detection window* (ROI) juga digunakan pada tiap *frame* video. Posisi

*detection window* (ROI) pada *background* dan *frame* haruslah sama karena di posisi inilah akan dilakukan proses penghitungan *Normalized Sum Squared Differences* (NSSD).



**Gambar 3.3.7 Contoh Penempatan *Detection Window (ROI)* Pada *Frame***

### 3.1.4.3 Grayscaling

Proses selanjutnya setelah proses *setting detection window* adalah proses *Grayscaling*, proses *Grayscaling* ialah mengubah citra berwarna menjadi citra abu-abu guna mempermudah proses perhitungan

Berikut adalah hasil dari proses *grayscaling* untuk *background image* :



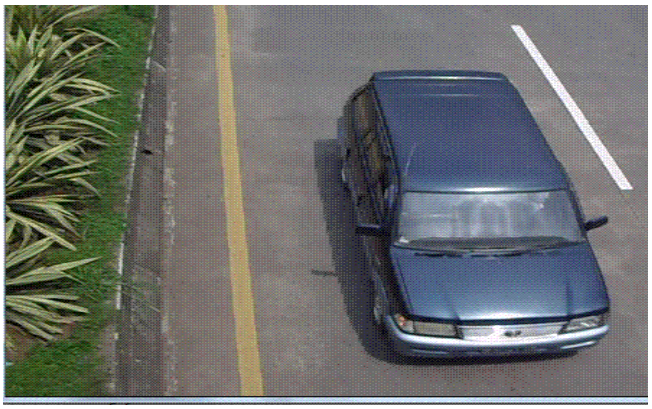
**Gambar 3.3.8 *Background Image* Dalam RGB**

Contoh *image background* yang telah di *grayscale* adalah sebagai berikut:



**Gambar 3.3.9 *Background Image Grayscale***

Berikut adalah hasil dari proses *grayscaling* untuk *image frame* :



**Gambar 3.3.10 *Frame Image* Dalam RGB**

Contoh *image frame* yang telah di *grayscale* adalah sebagai berikut:



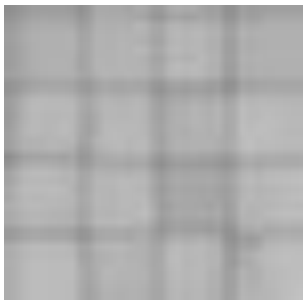
**Gambar 3.3.11 *Frame image Grayscale***

### 3.1.4.4 Perhitungan NSSD

Setelah melakukan proses *setting detection window* dan mengubah citra menjadi citra *gray*, langkah selanjutnya menghitung nilai NSSD yang terdapat pada setiap *frame* yang telah di *grayscale*. Perhitungan

*Normalized Sum Squared Differences* (NSSD) sendiri dilakukan di dalam *detection window* (ROI). Dalam perhitungan *Normalized Sum Squared Differences* (NSSD) dibutuhkan nilai intensitas piksel dari *detection window* (ROI) pada *background* dan *detection window* (ROI) pada *frame*. Intensitas piksel merupakan nilai derajat keabuan dari tiap piksel yang berada dalam *detection window* (ROI).

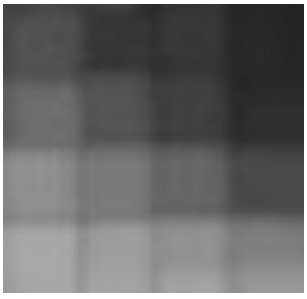
Rumus yang digunakan dalam perhitungan *Normalized Sum Squared Differences* (NSSD) adalah persamaan 2.9.



**gambar 3.3.13 Intensitas Piksel Pada ROI *Background***

| | | | |
|---|---|---|---|
| 0.4 | 0.2 | 0.2 | 0.3 |
| 0.3 | 0.1 | 0.2 | 0.1 |
| 0.2 | 0.1 | 0.4 | 0.1 |
| 0.1 | 0.2 | 0.1 | 0.2 |

**gambar 3.3.12 Nilai Intensitas Piksel Pada ROI *Background***



**gambar 3.3.15 Intensitas Piksel pada ROI *Frame***

| | | | |
|---|---|---|---|
| 0.7 | 0.9 | 0.8 | 1.0 |
| 0.6 | 0.7 | 0.8 | 1.0 |
| 0.4 | 0.5 | 0.6 | 0.8 |
| 0.1 | 0.2 | 0.2 | 0.5 |

**gambar 3.3.14 Nilai Intensitas Piksel Pada ROI *Frame***

Setelah didapatkan nilai intensitas pada *frame* dan *background* yang masih berbentuk matriks baris X kolom, kemudian nilai intensitas pada *frame* dan *background* dikonversi kedalam bentuk matriks deret. Sehingga hasil dari nilai di atas akan nampak seperti berikut:

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.7 | 0.9 | 0.8 | 1.0 | 0.6 | 0.7 | 0.8 | 1.0 | 0.4 | 0.5 | 0.6 | 0.8 | 0.1 | 0.2 | 0.2 | 0.5 |

**Gambar 3.16 Nilai Intensitas *frame* dalam matriks deret**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.4 | 0.2 | 0.2 | 0.3 | 0.3 | 0.1 | 0.2 | 0.1 | 0.2 | 0.1 | 0.4 | 0.1 | 0.1 | 0.2 | 0.1 | 0.2 |

**Gambar 3.17 Nilai Intensitas *background* dalam matriks deret**

Setelah nilai intensitas piksel pada *background* dan *frame* dirubah kedalam bentuk matriks deret, kemudian langkah selanjutnya adalah mengurangkan nilai intesitas piksel yang ada pada *frame* dengan nilai intensitas piksel yang ada pada *background.*

Untuk lebih jelasnya dapat dilihat pada contoh analisis dibawah ini:

$$\text{NSSD} = \frac{(0.7-0.4)^2 + (0.9-0.2)^2 + (0.8-0.2)^2 + \ldots + (0.2-0.1)^2 + (0.5-0.2)^2}{16}$$

$$= \frac{3.88}{16}$$

$$= 0.2425$$

Hasil dari perhitungan *Normalized Sum Squared Differences* (NSSD) sendiri digunakan untuk menunjukan ada atau tidaknya kendaraan yang lewat. Jika nilai *Normalized Sum Squared Differences* (NSSD) lebih besar dari nilai threshold yang telah ditentukan maka itu menunjukan keberadaan sebuah kendaraan namun jika nilai *Normalized Sum Squared Differences* (NSSD) lebih kecil dari nilai threshold yang telah ditentukan maka itu menunjukan tidak ada kendaraan yang melintas.

Algoritma dari Metode *Normalized Sum Squared Differences* (NSSD) adalah sebagai berikut:

1. *Capture Background*
2. Input *detection window* pada *background*
3. C*apture frame*
4. Nilai intensitas *piksel* pada *detection window frame* dikurangi nilai intensitas piksel pada *detection window background*.
5. Nilai hasil intensitas piksel *frame* yang telah dikurangi nilai intensitas *background* akan di ambil nilai rata-ratanya setiap 15 *frame*.
6. Jika nilai rata-rata yang diperoleh melebihi dari nilai *treshold* maka itu nilai tersebut mewakili keberadaan sebuah kendaraan. Jika nilai

rata-rata yang didapat kurang dari nilai *treshold* maka kembali ke proses *capture frame*.

### 3.1.4.5 Filtering

Setelah mendapatkan nilai NSSD dari setiap *frame* maka untuk mengurangi nilai yang mencolok di perlukan proses *Filtering*. Proses *filtering* adalah proses untuk mengurangi perbedaan nilai data yang mencolok. Teknik *filtering* yang digunakan dalam metode *Normalized Sum Squared Differences* (NSSD) adalah teknik *low Pass Filter* (persamaan). *Low Pass Filter* menggunakan cara merata-rata setiap data *Normalized Sum Squared Differences* (NSSD) yang diperoleh.

Dalam kasus ini proses *filtering* diterapkan terhadap setiap 15 data hasil perhitungan *Normalized Sum Squared Differences* (NSSD).

Contoh proses *filtering* terhadap 15 data hasil perhitungan *Normalized Sum Squared Differences* (NSSD) :

$$\text{Filtering} = \frac{NSSD1+NSSD2+NSSD3+NSSD4+\ldots+NSSD15}{15}$$

$$= \frac{67.16+70.42+68.02+84+24+\ldots\ldots..+74.20}{15}$$

$$= \frac{1721.56}{15}$$

$$= 1652.31$$

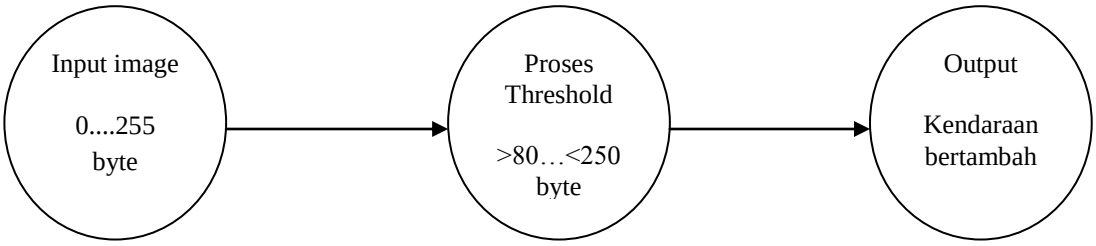
Setelah dilakukan proses untuk ke-15 data nilai NSSD, maka didapati nilai NSSD rata-rata untuk ke-15 frame adalah 1652.31. nilai rata-rata yang telah didapat tersebut akan dibandingkan dengan nilai *threshold* untuk mengetahui apakah dalam 15 frame tersebut ada kendaraan atau tidak.

### 3.1.4.6 Threshold

Tahpan selanjutnya setelah proses filtering adalah prosses membandingkan nilai nssd yang telah di *filtering* dengan nilai *treshold*. Nilai *threshold* sendiri telah ditentukan berdasarkan data perhitungan nilai *Normalized Sum Squared Differences* (NSSD) terendah. Dalam *threshold* dibutuhkan sebuah nilai *variabel* bantu yaitu Nilai *Base*. Nilai *base* sendiri akan mencegah perhitungan ganda terhadap satu kendaraan jika ada kendaraan yang masuk dalam beberapa *frame*.

Proses *threshold* sendir terjadi didalam system, cara kerja dari proses *threshold* sendiri adalah dengan cara melihat perbedaan intensitas warna dari suatu *image*. *Input* dari proses *tresholding* adalah *grayscale* image atau *color image* sedangkan *output* dari proses *tresholding* adalah *binary image* dimana piksel hitam untuk mewakili background dan piksel putih mewakili *foreground*.

Ilustrasi dari proses *threshold* yang ada pada metode ini dapat dilihat dari gambar berikut:




**Gambar 3.18 Ilustrasi Proses *Treshold* Dalam Sistem**

### 3.1.4.7 Proses Penghitungan Kendaraan

Proses ini merupakan terusan setelah proses t*reshold*, di dlam proses ini Jumlah kendaraan akan bertambah jika satu kendaraan itu telah melewati detection window untuk dihitung nilai NSSD dan nilai rata-rata NSSD-nya (proses filtering) lebih besar dari Nilai Treshold dan nilai base masih 0.

## 3.1.5 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan *non fungsional* merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan

untuk sistem yang akan dibangun sampai dengan sistem tersebut dapat di implementasikan.

Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

### 3.1.5.1 Analisis Pengguna

Analisis pengguna dilakukan untuk mengetahui siapa yang akan menggunakan aplikasi perangkat lunak penghitungan jumlah kendaraan dengan menggunakan kamera.

Berdasarkan penelitian yang telah dilakuan, pelaku sistem manual di Dinas Perhubungan Kota Cikampek di tujukan pada tabel 3.1 .

**Tabel 3.3-1 Pelaku sistem manual**

| Pengguna | Tugas | Keterangan |
|---|---|---|
| Kabag Lapangan DisHub Cikampek | Memeriksa laporan hasil laporan penghitungan jumlah kendaraan. | Penanggung jawab operasional lapangan Dinas Perhubungan Kota Cikampek dan memeriksa seluruh laporan. |
| Pegawai DisHub Div Lapangan | Menghitung, mencatat dan membuat laporan dari hasil pengamatan jumlah kendaraan di lapangan | Melakukan penghitungan dan pencatatan serta membuat laporan. |

Sementara untuk penerapan system yang baru hanya ada satu pengguna saja yaitu pegawai dishub divisi lapangan yang bertindak sebagai user peragkat lunak penghitung jumlah kendaraan menggunakan kamera. Sementara Kabag sendiri hanya bertugas menerima laporan hasil perhitungan yang telah dilakukan oleh pegawai dishub divisi lapangan.

**Tabel 3.3-2 Pelaku Sistem Yang Baru**

| Pengguna | Tugas | Keterangan |
|---|---|---|
| Pegawai DisHub Divisi Lapangan | Mengawasi dan mengoperasikan sistem perangkat lunak penghitung jumlah kendaraan | Bertindak sebagai pegawai dinas perhubungan divisi lapangan yang bertugas untuk mengoperasikan perangkat lunak penghitung jumlah kendaraan menggunakan kamera. |

### 3.1.5.2 Analisis Kebutuhan Perangkat Keras

Perangkat keras untuk implementasi program perangkat lunak sangatlah mendukung agar menghasilkan suatu informasi yang sesuai dengan yang diharapkan.

Berdasarkan hasil observasi dan penelitian langsung terhadap pihak Dinas Perhubungan Kota Cikampek, didapati perangkat keras yang digunakan untuk menghitung jumlah kendaraan yang melintas adalah alat penghitung manual.

Adapun perangkat keras yang dibutuhkan untuk mendukung penerapan perangkat lunak ini adalah sebagai berikut :

1. Kamera
   a. Handycam Sony DCR-TRV 250E.
   b. Kamera Digital Casio
   c. *Webcam* Logitech 5MP
   d. Kabel

2. Komputer
   a. Mikroprosessor : *Intel Pentium IV 4.0 GHz*
   b. Memori : 1 GB DDR2
   c. Media penyimpanan : *Harddisk* 160 GB
   d. Kartu Grafik (VGA) : 256 MB
   e. Monitor : 14 inch

### 3.1.5.3 Analisis Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak merupakan faktor yang harus dipenuhi untuk merancang sebuah perangkat lunak sehingga sesuai dengan maksud dan tujuan perangkat lunak tersebut dibuat.

Berdasarkan hasil observasi dan penelitian langsung terhadap pihak Dinas Perhubungan Kota Cikampek, didapati tidak ada perangkat lunak yang digunakan untuk menghitung jumlah kendaraan yang melintas.

Untuk penerapan perangkat lunak yang akan dibangun diusulkan ada beberapa perangkat lunak pendukung sebagai berikut :

1. Sistem Operasi

Sistem Operasi yang digunakan adalah *Windows XP SP2*. Dipilih karena sistem operasi ini *pegawai dinas perhubungan divisi lapangan friendly* dengan aplikasi apapun,

*2. Database Management Sistem*

Database yang digunakan untuk menyimpan data-data dalam penelitian dan pembangunan perangkat lunak ini adalah *Microsoft Office Access*.

3. *C#*

Merupakan *tools development* yang digunakan dalam proses *coding* perangkat lunak penghitung jumlah kendaraan menggunakan kamera ini.

4. AForge

*Library* tambahan yang berfungsi merekayasa *image.*

### 3.1.5.4 Analisis *Library*

*Library* merupakan tools tambahan yang ditanam didalam sistem. Penggunaan *library* sendiri bertujuan untuk mempermudah dalam pembangunan suatu sistem kerana di dalam librari biasanya sudah di sediakan fungsi fungsi tertentu.

Adapun *library* yang digunakan dalam pembangunan aplikasi ini adalah:

**Tabel 3-3 Analisis *Library***

| Nama Proses | Library Yang Digunakan | Keterangan |
|---|---|---|
| Membaca Driver Kamera | Tiger Video | Membaca driver kamera yang akan digunakan untuk melakukan perekaman. |
| Grayscaling | Aforge | *Library* tambahan yang berfungsi untuk melakukan pengolahan citra di dalam setiap *frame*, termasuk proses *grayscaling*. |
| | Aforge.imaging | |
| Filtering | Aforge | *Library* tambahan yang berfungsi untuk melakukan pengolahan citra di dalam setiap *frame*, termasuk proses |
| | Aforge.imaging | |

| | | |
|---|---|---|
| | | *filtering*. |
| Nama Proses | Library Yang Digunakan | Keterangan |
| Tresholding | Aforge | *Library* tambahan yang berfungsi untuk melakukan proses *tresholding*. |
| | Aforge.math | |

### 3.1.5.5Batasan Sistem

Batasan sistem ditujukan untuk mebatasi permasalahan yang dapat ditangani oleh aplikasi yang telah buat.

Adapun batasan system dari aplikasi perangkat lunak penghitung jumlah kendaraan menggunakan kamera ini antara lain:

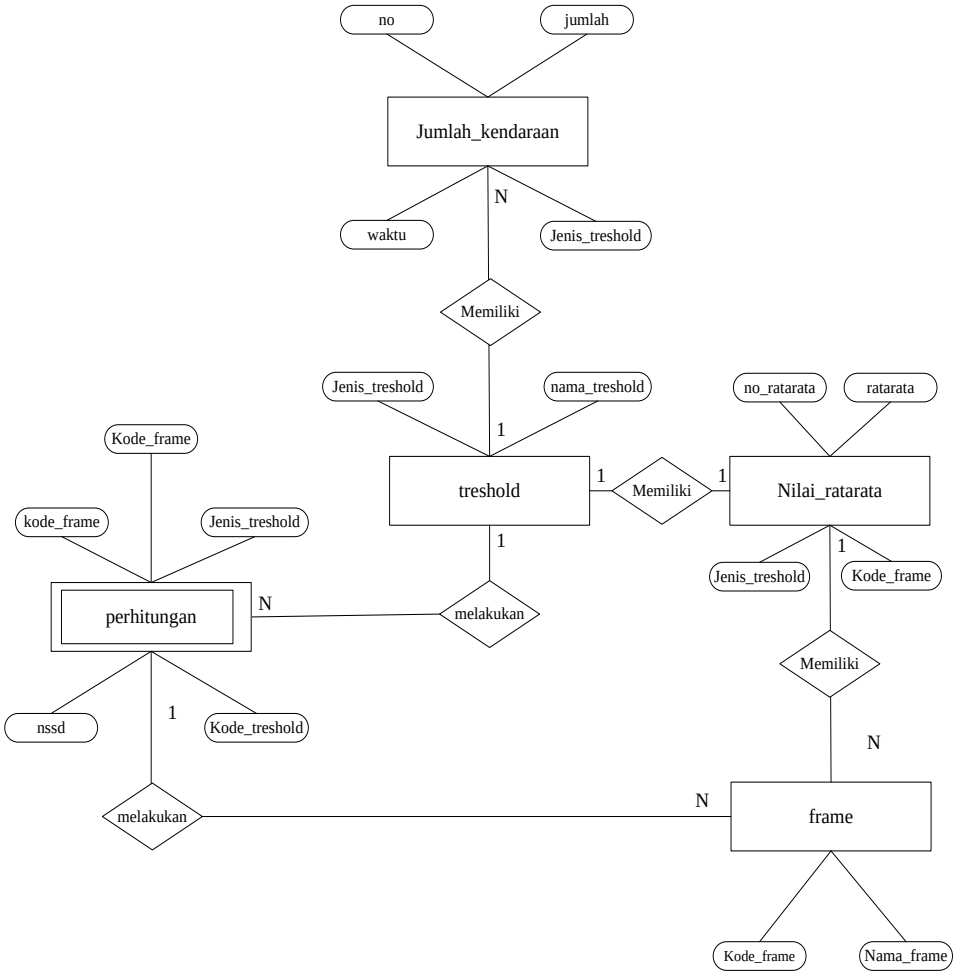
1. Satu kamera digunakan hanya untuk mengamati satu lajur.
2. Terjadi *delay* dari kamera ke aplikasi sekitar 2-3 detik.
3. Kondisi cahaya mempengaruhi hasil perhitungan.

### 3.1.6 Analisis Basis Data

ERD (Entity Relationship Diagram)

Untuk memodelkan data dan menggambarkan hubungan antara data yang ada, digunakan alat bantu yaitu diagram E-R. Usulan untuk

perancangan diagram E-R yaitu dapat membedakan dengan atribut lainnya sehingga tabel tersebut dapat dijadikan referensi untuk tabel lainnya. Adapun diagram E-R tentang proses akses informasi ini di tujukan pada gambar:




**Gambar 3.3.19 Entity Relationship Diagram Perangkat Lunak Penghitung Jumlah Kendaraan Menggunakan Kamera**

Adapun keterangan gambar diatas adalah sebagai berikut :

**Tabel 3.3.4 Keterangan ERD Perangkat Lunak Penghitung Jumlah Kendaraan Menggunakan Kamera**

| No | Entitas | Atribut | Primary Key |
|---|---|---|---|
| 1 | *Frame* | Kode_*frame*, nama_*frame*, | Kode_*frame* |
| 2 | Perhitungan | Kode_*frame*, jenis_treshold, nssd, | - |
| 3 | Treshold | Kode_treshold, jenis_treshold | Kode_treshold |
| 4 | Rata-rata | No_ratarata, ratarata, jenis_treshold | No_ratarata |
| 5 | Jumlah_kendaraan | No, jumlah, waktu, jenis_treshold | No |

### 3.1.7 Anlasis Kebutuhan Fungsional

### 3.1.7.1 Diagram konteks

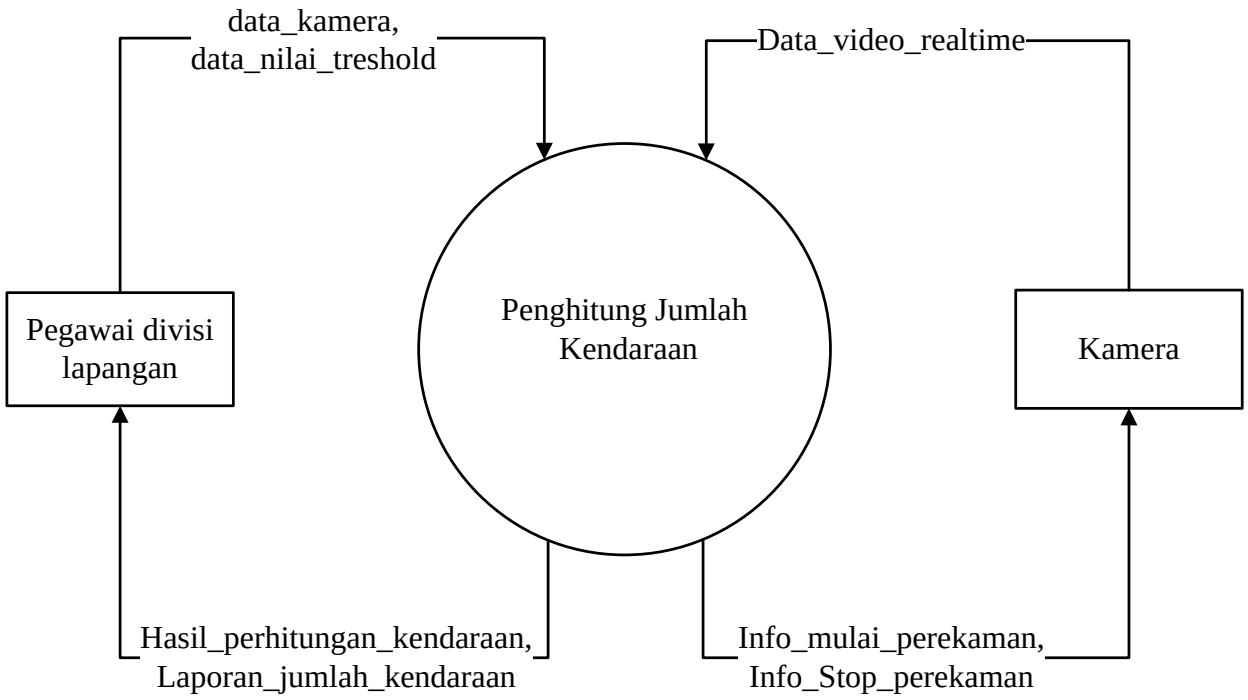
Analisis fungsional dilakukan untuk menghasilkan spesifikasi kebutuhan fungsional. Spesifikasi kebutuhan fungsional adalah spesifikasi

yang rinci tentang hal-hal yang akan dilakukan pada saat implementasi sistem.

1. Diagram Konteks
2. DFD (*Data Flow Diagram*)
3. Spesifikasi Proses
4. Kamus Data

Sistem mempunyai keterkaitan dengan jumlah entitas, baik itu keterkaitan dengan entitas luar sistem maupun keterkaitan dengan entitas dalam sistem. Hubungan antara entitas dengan sistem digambarkan dengan diagram *konteks*.

Gambar diagram konteks pada sistem perangkat lunak penghitung jumlah kendaraan menggunakan kamera dapat dilihat pada gambar berikut:




**Gambar 3.3.20 Diagram Konteks Perangkat Lunak Penghitung Jumlah Kendaraan Menggunakan Kamera**

Keterangan gambar :

Pegawai divisi lapangan : Entitas luar yang menggunakan sistem aplikasi perangkat lunak penghitung jumlah kendaraan menggunakan kamera.

Kamera : Entitas luar sebagai media *input* video.

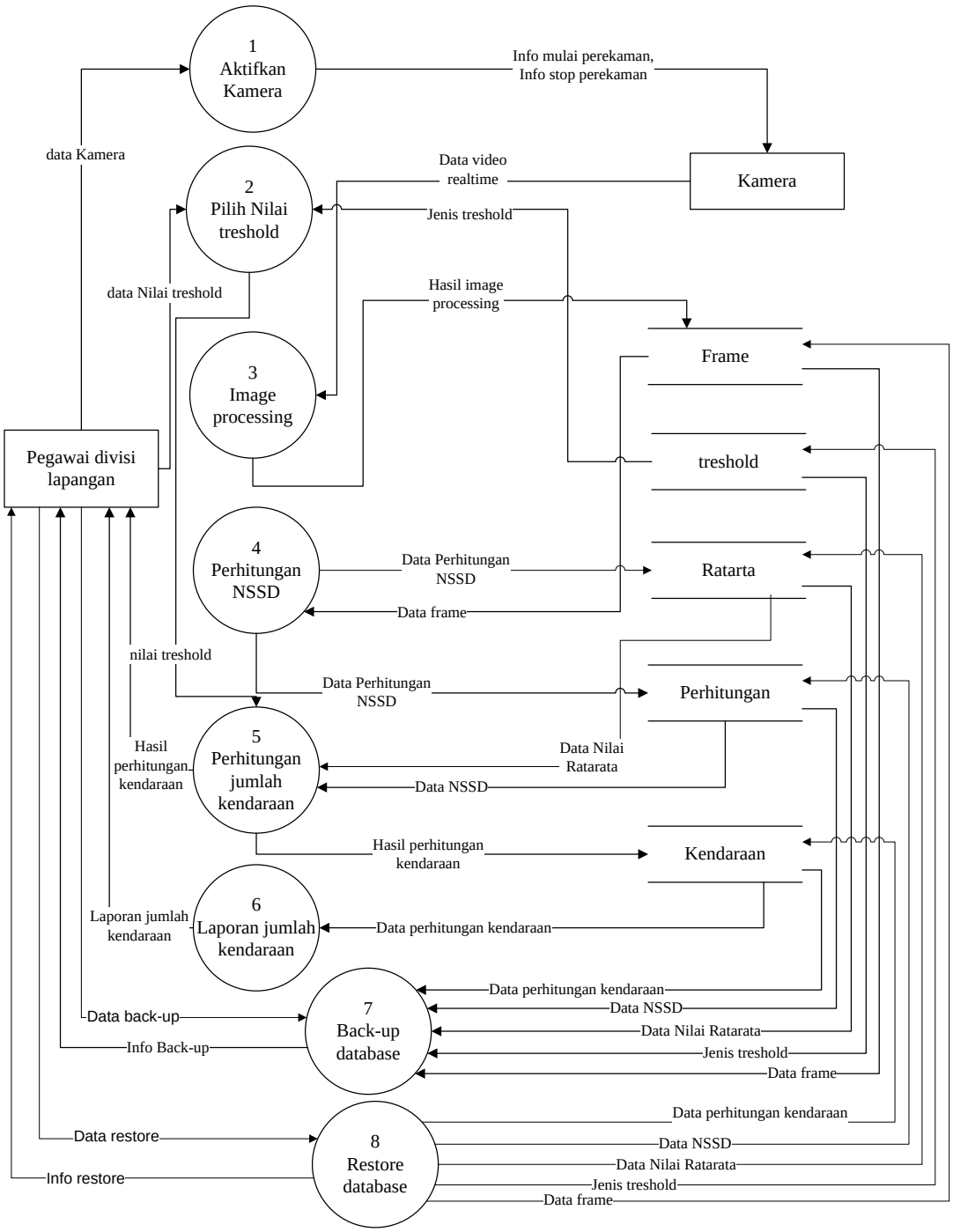
### 3.1.7.2 Data Flow Diagram (DFD)

*Data Flow Diagram* (DFD) adalah sebuah teknik yang menggambarkan aliran data dan transformasi yang digunakan sebagai perjalanan data dari masukan menuju keluaran. *Data Flow Diagram* DFD dibuat jika pada *Diagram Konteks* masih terdapat proses yang mesti dijelaskan lebih rinci.

#### 3.1.7.2.1 DFD Level 1 Penghitung Jumlah Kendaraan

Pada *level* ini, proses tunggal dari diagram konteks di pecah menjadi enam proses yang lebih terperinci, yaitu proses pilih data proses atur kamera, proses *Setting* Kamera, proses *image processing*, proses Setting *Detection Windows*, proses perhitungan NSSD, proses Perhitungan Jumlah Kendaraan dan proses Laporan. Pada level ini terdapat 5 penyimpanan data yaitu *table frame*, *table* perhitungan, *table* ratarata, *table roi*, dan *table* kendaraan untuk menyimpan data jumlah kendaraan yang melintas.

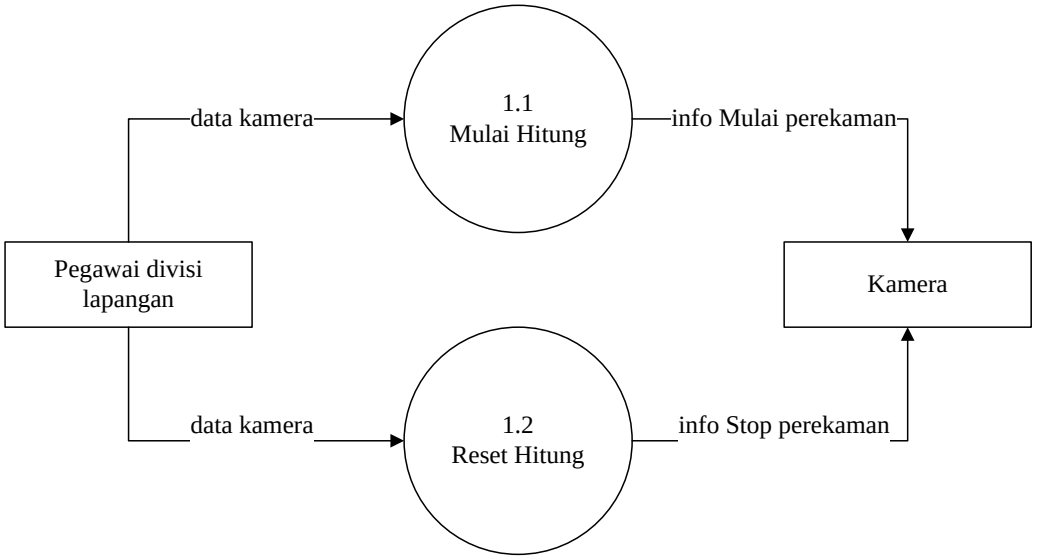
Adapun DFD level 1 dari aplikasi perangkat lunak penghitung jumlah kendaraan menggunakan kamera dapat dilahat pada gambar :




**Gambar 3.3.21 DFD level 1 Perangkat Lunak Penghitung Jumlah Kendaraan Menggunakan Kamera**

### 3.1.7.2.2 DFD Level 2 proses 2

DFD level 2 adalah turunan level dari DFD level 1, dimana masih terdapat proses yang harus di turunkan lagi agar lebih terperinci. pada proses *setting* di DFD Level 1 proses 2 terdapat dua proses, untuk itu di DFD level 2 ini akan dijelaskan secara lebih terperinci dari proses tersebut. Rincian proses dan penjelasan DFD level 2 sebagai berikut:




**Gambar 3.3.22 DFD Level 2 Proses 2 Perangkat Lunak Penghitung Jumlah Kendaraan Menggunakan Kamera**

## 3.1.8 Spesifikasi Proses

Sepsifikasi proses digunakan untuk mendeskripsikan proses yang terjadi pada level yang paling dasar dalam DFD. Model ini berfungsi

mendeskripsikan apa yang dilakukan ketika masukan ditransformasi menjadi keluaran. Metode yang digunakan dapat berupa Bahasa Indonesia atau Bahasa Inggris yang terstruktur.

Bentuk penyajian spesifikasi proses untuk sistem penghitung jumlah kendaraan menggunakan kamera adalah sebagai berikut :

**Tabel 3.3-5 Sfesipikasi Proses Perangkat Lunak Penghitung Jumlah Kendaraan Menggunakan Kamera**

| No | Proses | Keterangan |
|---|---|---|
| 1 | Nomor Proses | 1 |
| | Nama Proses | Aktifkan Kamera |
| | Deskripsi | Mengatur kamera sebagai media input untuk proses penghitungan. |
| | Input | Kamera |
| | Output | Video Realtime |

| No | Proses | Keterangan |
|---|---|---|
| | Proses | Video real time di proses pada poses *image processing* |
| | Logika Proses | 1. Pegawai Dinas Perhubungan divisi lapangan memiih kamera yang akan digunakan sebagai media input<br>2. Pegawai Dinas Perhubungan divisi lapangan mengaktifkan kamera yang telah dipilih.<br>3. Setelah kamera aktif maka Pegawai Dinas Perhubungan divisi lapangan menetukan posisi kamera untuk pengambilan *background.* |
| 2 | Nomor Proses | 2 |
| | Nama Proses | Pilih Nilai treshold |
| | Deskripsi | Memiih nilai treshold |
| | Input | Nilai treshold yang akan digunakan yang telah disediakan oleh sitem |
| | Output | Jenis treshold |

| No | Proses | Keterangan |
|---|---|---|
| | Proses | Tabel treshold akan memberikan data jenis treshold sesuai dengan nilai treshold yang dipilih oleh pegawai dinas perhubungan divisi lapangan. |
| | Logika Proses | 1. Pegawai dinas perhubungan divisi lapangan memilih nilai treshold yang akan digunakan yang telah disediakan oleh sitem.<br>2. nilai treshold ini sebagai pembanding untuk memperoleh hasil perhitungan |
| 3 | Nomor Proses | 3 |
| | Nama Proses | Image Processing |
| | Deskripsi | Pengolahan citra terhadap *frame-frame* video |
| | Input | Video Realtime |
| | Output | Video realtime yang telah di olah dengan *image processing* didalam system yang akan disimpan pada tabel *frame* |
| | Proses | Video real time yang di dapat pada proses 1 di olah menggunakan pengolahan citra |

| No | Proses | Keterangan |
|---|---|---|
| | Logika Proses | 1. proses image processing untuk setiap *frame* dari video *realtime* yang dilakukan didalam system.<br>*2.* Proses pengolahan citra ini meliputi proses *Grayscaling* dan proses *Filtering.* |
| 4 | Nomor Proses | 4 |
| | Nama Proses | Perhitungan nssd |
| | Deskripsi | Perhitungan nilai nssd |
| | Input | data *frame* |
| | Output | Nilai NSSD |
| | Proses | Data nilai nssd dari tiap frame dibandingkan dengan nilai threshold yang telah dipilih oleh pegawai Dinas Perhubungan Divisi Lapangan. |

| No | Proses | Keterangan |
|---|---|---|
| | Logika Proses | 1. Input data *frame*<br>2. Data nilai dari tiap *frame* yang telah dikurangi dengan nilai *background* akan diambil nilai rata-ratanya sebelum dilakukan proses perhitungan dengan menggunakan rumus pada metode nssd. |
| 5 | Nomor Proses | 5 |
| | Nama Proses | Perhitungan jumlah kendaraan |
| | Deskripsi | Menghitung jumlah kendaraan yang melintas berdasarkan nilai nssd yang telah diolah. |
| | Input | Data nssd, data nilai ratarata, jenis treshold |
| | Output | Hasil perhitungan kendaraan |
| | Proses | Data nilai nssd yang telah di ambil nilai rata-ratanya akan dibandingkan dengan nilai threshold guna mendapatkan nilai yang menunjukan keberadaan sebuah kendaraan. |

| No | Proses | Keterangan |
|---|---|---|
| | Logika Proses | 1. Data nssd yang telah didapat akan dibandingkan dengan nilai threshold yang telah dipilih oleh pegawai Dinas Perhubungan divisi lapangan.<br>2. Hanya nilai nssd yang melebihi nilai threshold yang telah dipilih pegawai Dinas Perhubungan divisi lapangan yang dapat menyatakan keberadaan sebuah kendaraan.<br>3. Hasil perhitungan kendaraan ditampilkan di form pada interface dan disimpan pada tabel kendaraan |
| 6 | Nomor Proses | 6 |
| | Nama Proses | Laporan |
| | Deskripsi | Pembuatan laporan jumlah kendaraan yang melintas |
| | Input | Data perhitungan jumlah kendaraan |
| | Output | Laporan jumlah kendaraan |

| No | Proses | Keterangan |
|---|---|---|
| | Proses | Data jumlah kendaraan yang ada pada tabel kendaraan dipanggil untuk diproses manjadi sebuah laporan yang ditampilkan kepada pegawai dinas perhubungan divisi lapangan berupa laporan jumlah kendaraan |
| | Logika Proses | 1. Nilai nssd yang telah dibandingkan dengan nilai threshold akan menunjukan keberadaan sebuah kendaraan.<br>2. Semakin banyak nilai nssd yang melebihi maka semakin banyak pula kendaraan yang terhitung. |
| 7 | Nomor Proses | 1.1 |
| | Nama Proses | Mulai Rekam |
| | Deskripsi | Mengaktifkan kamera untuk memulai perekaman |
| | Input | Info mulai perekaman |
| | Output | Video realtime |

| No | Proses | Keterangan |
|---|---|---|
| | Proses | Pegawai dinas perhubungan divisi lapangan meng-klik tombol yang ada pada tampilan untuk mengaktifkan kamera dan memulai perekaman |
| | Logika Proses | 1.Pilih sub menu aktifkan kamera untuk memilih kamera yang akan digunkan untuk melakukan perekaman.<br>2.Klik tombol Mulai hitung untuk memulai proses penghitungan. |
| 8 | Nomor Proses | 1.2 |
| | Nama Proses | Stop Rekam |
| | Deskripsi | Stop kamera untuk menghentikan proses perekaman |
| | Input | Info berhenti perekaman |
| | Output | Kamera berhenti malakukan perekaman. |
| | Proses | Pegawai dinas perhubungan divisi lapangan memerintahkan terhadap sistem untuk meengnon-aktifkan kamera dan menghentikan perekaman |

| | | |
|---|---|---|
| | Logika Proses | 1. Klik tombol reset perhitungan untuk menghentikan proses perhitungan. |
| 9 | No. Proses | 6.1 |
| | Nama Proses | Seleksi Tanggal |
| | Deskripsi | Menyeleksi tanggal yang akan ditampilkan |
| | Input | Info tanggal |
| | Output | Data tanggal |
| **No** | **Proses** | **Keterangan** |
| | Proses | Pegawai Dinas Perhubungan divisi lapangan mengi*input*kan tanggal yang akan dijadikan laporan. |
| | Logika Proses | 1.*input* tanggal yang dicari.<br>2.*input* bulan yang dicari.<br>3.data jumlah kendaraan tanggal yang diseleksi. |

| | | |
|---|---|---|
| 10 | Nomor Proses | 6.2 |
| | Nama Proses | Cetak Laporan |
| | Deskripsi | Mencetak laporan berdasarkan tanggal yang telah seleksi. |
| | Input | *printout* laporan |
| | Output | Laporan jumlah kendaraan |
| | Proses | Mencetak laporan jumlah kendaraan yang melintas berdasarkan tanggal yang telah di seleksi. |
| | Logika Proses | 1.*input* tanggal yang diseleksi<br>2.cetak laporan berdasarkan yang telah diseleksi.<br>3. laporan jumlah kendaraan. |

### 3.1.9 Kamus Data

Kamus data merupakan suatu daftar terorganisasi tentang komposisi elemen data, aliran data, dan data store yang digunakan dalam DFD. Pengisian kamus data dilakukan setiap saat selama proses

pengembangan berlangsung, ketika diketahui adanya data item atau saat diperlukan penambahan data item kedalam sistem.

Bentuk penyajian kamus data untuk sistem penghitung jumlah kendaraan menggunakan kamera adalah sebagai berikut :

**Tabel 3.3-6 Kamus Data Perangkat Lunak Penghitung Jumlah Kendaraan Menggunakan Kamera**

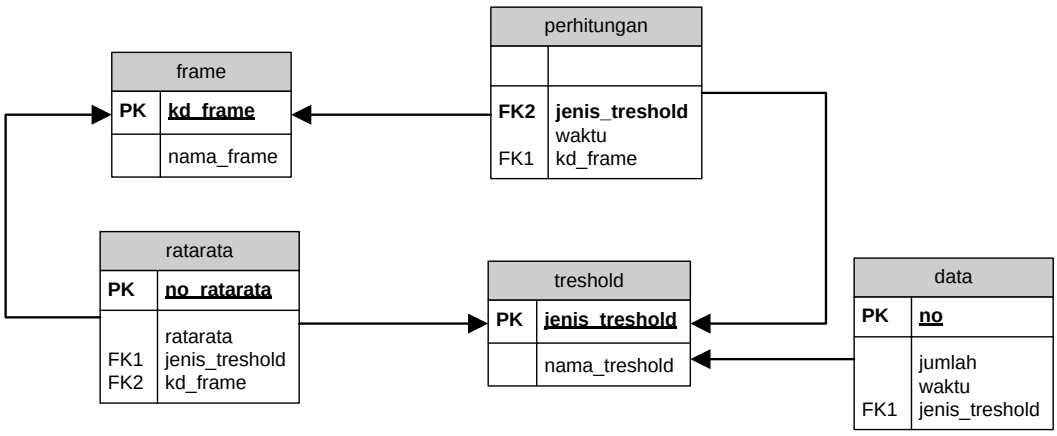
| No | Nama | Hasil image processing |
|---|---|---|
| 1 | Alur Data | Proses 3 image processing |
| | Struktur Data | Kode_*frame* = [0 . . . 9] |
| | | Nama_*frame* = [a . . . z] |
| | | Data hasil perhitungan nssd |
| 2 | Alur Data | Proses 5 perhitungan nssd |
| | Struktur Data | Nilai nssd = [0 . . . 9 ] |
| | | Kode_treshold = [0 . . . 9 ] |

| No | Nama | Hasil image processing |
|---|---|---|
| 3 | Alur Data | Hasil perhitungan kendaraan |
| | Alur Data | Proses 6 Perhitungan Kendaraan |
| | Struktur Data | No = [ 0 . . . 9] |
| | | Jumlah = [0 . . . 9] |
| | | waktu = [ 0 . . . 9 \|-/ : ] |

## 3.2 Perancangan Sistem

### 3.2.1 Skema Relasi

Proses relasi antar file merupakan gabungan antar file yang mempunyai kunci utama yang sama, sehingga file-file tersebut menjadi satu kesatuan yang dihubungkan oleh field kunci. Pada proses ini elemen-elemen data dikelompokkan menjadi suatu file data base beserta entitas dan hubungannya. Skema relasi yang digunakan dapat dilihat pada gambar berikut :




**Gambar 3.3.23 Skema Relasi Perangkat Lunak Penghitung Jumlah Kendaraan Menggunakan Kamera**

### 3.2.2 Struktur Tabel

Tabel-tabel yang terdapat dalam basis data yang digunakan dalam sistem ini adalah sebagai berikut :

**Tabel 3.3-7 Struktur Tabel *Frame* Perangkat Lunak Penghitung Jumlah Kendaraan Menggunakan Kamera**

| No | Nama Field | Tipe | Size | Kunci |
|---|---|---|---|---|
| 1 | Kd_*frame* | Integer | - | *Primary Key* |
| 2 | Nama_*frame* | Car | 20 | - |

**Tabel 3.3-8 Struktur Tabel Perhitungan Perangkat Lunak Penghitung Jumlah Kendaraan Menggunakan Kamera**

| No | Nama Field | Tipe | Size | Kunci |
|---|---|---|---|---|
| 1 | Jenis_treshold | Integer | - | *foreign key references Treshold (Jenis_treshold)* |
| 2 | Kd_frame | Integr | - | *foreign key references Frame (Kd_frame)* |
| 3 | nssd | Integer | - | - |

**Tabel 3.3-9 Struktur Tabel treshold Perangkat Lunak Penghitung Jumlah Kendaraan Menggunakan Kamera**

| No | Nama Field | Tipe | Size | Kunci |
|---|---|---|---|---|
| 1 | Jenis_treshold | Integer | - | *Primary key* |
| 2 | Nama_treshold | Car | 20 | - |

**Tabel 3.3-10 Struktur Tabel Ratarata Perangkat Lunak Penghitung Jumlah Kendaraan Menggunakan Kamera**

| No | Nama Field | Tipe | Size | Kunci |
|---|---|---|---|---|
| 1 | No_ratarata | Integer | - | *Primary key* |
| 2 | ratarata | Car | 20 | - |
| 3 | jenis_treshold | Integer | - | *Foreign key references treshold (Jenis_treshold)* |

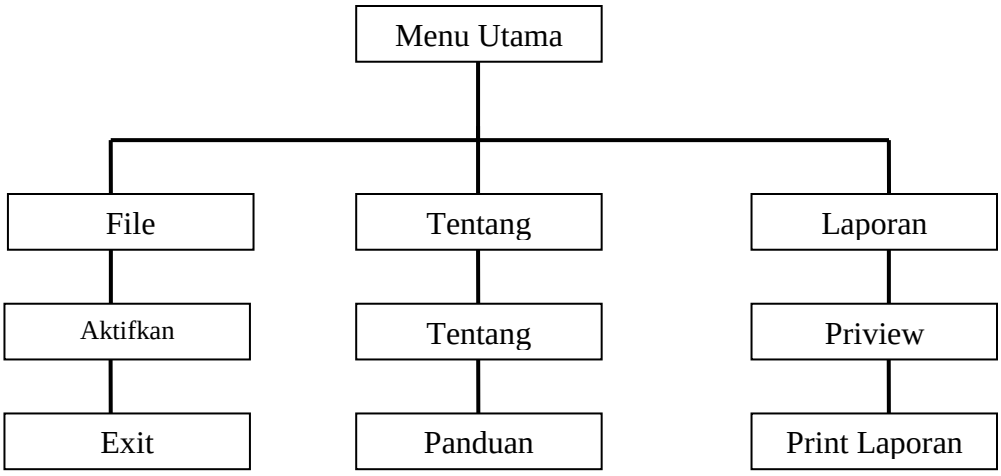
**Tabel 3.3-11 Struktur Tabel Kendaraan Perangkat Lunak Penghitung Jumlah Kendaraan Menggunakan Kamera**

| No | Nama Field | Tipe | Size | Kunci |
|---|---|---|---|---|
| 1 | No | Integer | - | *Primary key* |
| 2 | jumlah | Integer | - | - |
| 3 | waktu | datetime | - | - |
| 4 | Jenis_treshold | Integer | - | *Foreign key references treshold (Jenis_treshold)* |

### 3.2.3 Perancangan Arsitektur

#### 3.2.3.1 Perancangan Struktur Menu

Struktur menu dari aplikasi perangkat lunak penghitung jumlah kendaraan menggunakan kamera dapat dilihat pada bagan beikut.



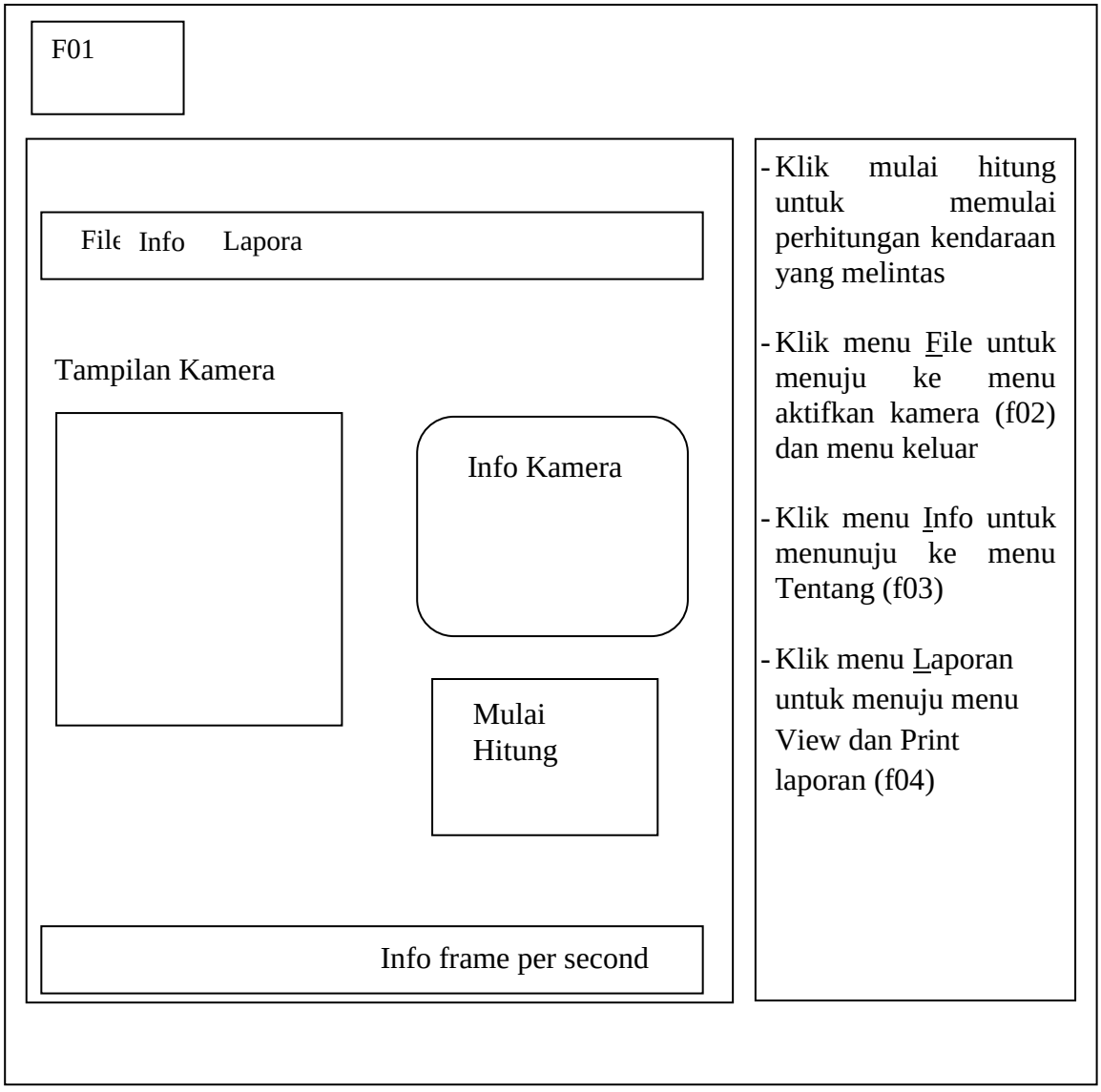

**Gambar 3.3.24 Perancangan Struktur Menu**

#### 3.2.3.2Perancangan Antar Muka

Perancangan antar muka atau *interface* adalah suatu tampilan dari suatu program aplikasi sebagai media untuk berkomunikasi untuk melakukan interaksi dengan *user*.

Berikut adalah tampilan antar muka yang dirancang pada sistem perangkat lunak penghitung jumlah kendaraan menggunakan kamera *:*

**1. Tampilan Form Utama Aplikasi**




**Gambar 3.3.25 Interface tampilan Perangkat Lunak Penghitung Jumlah Kendaraan Menggunakan Kamera**

**2. Deskripsi Objek Form Utama Aplikasi**

Berikut adalah deskripsi dari form utama aplikasi:

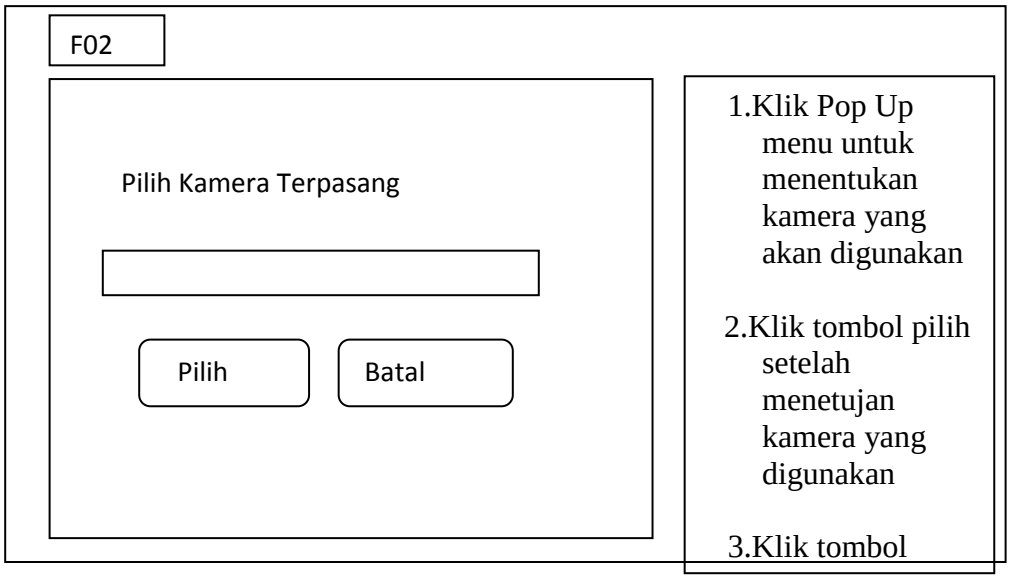
**3.3-12 Tabel Deskripsi Objek Form Utama**

| Objek | Jenis | Keterangan |
|---|---|---|
| File | Tool Bar | Menu untuk menuju F02 dan Keluar |
| Info | Tool Bar | Menu untuk menuju F03 |
| Laporan | Tool Bar | Menu untuk menuju F04 |
| Mulai Hitung | Button | Tombol untuk memulai penghitungan jumlah kendaraan yang melintas |
| Tampilan Kamera | Video | Tampilan video dari kamera secara real time |
| Info Kamera | Teks | Menunjukan waktu serta jumlah kendaraan yang terhitung oleh kamera |
| Info Freme Per Second | Teks | Menunjukan jumlah frame pada kamera |

### 3.2.3.3 Perancangan Antar Muka Form Aktifkan Kamera

Form ini diperuntukan bagi user (pegawai divisi lapangan dinas perhubungan kota cikampek) untuk memilih kamera yang akan di gunakan.

**1. Tampilan Form Aktifkan Kamera**




**Gambar 3.3.26 Gambar Form Aktifkan Kamera**

## 2. Deskripsi Objek Form Aktifkan Kamera

Berikut adalah deskripsi dari form aktifkan kamera:

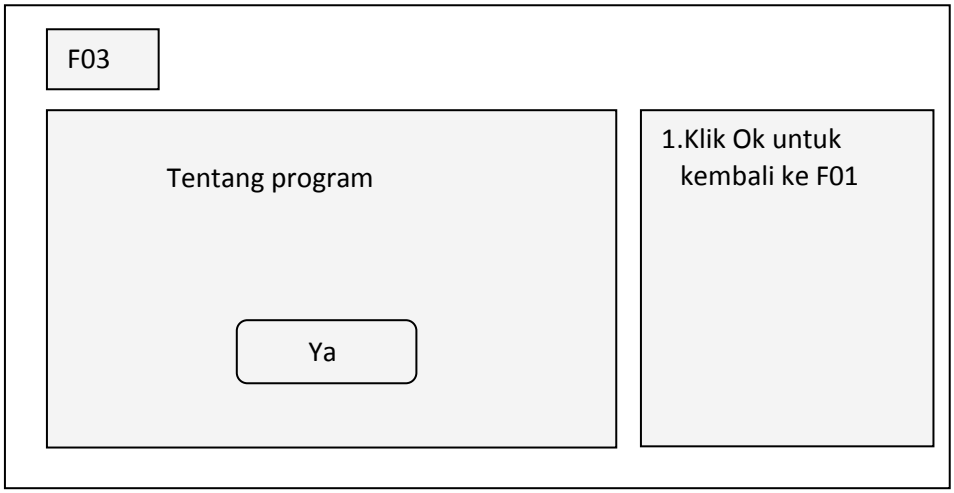
**3.3-13 Tabel Deskripsi Objek form Aktifkan Kamera**

| Objek | Jenis | Keterangan |
|---|---|---|
| Pilih Kamera Terpasang | Scrol menu | Memilih kamera yang telah terhubung dan yang akan digunakan |
| Pilih | Button | Memilih kamera yang akan di gunakan |
| Batal | Button | Kembali ke menu F01 |

### 3.2.3.4 Perancangan Tampilan Antar Muka Form Info

Form ini berisi tentang perangkat lumak penghitung jumlah kendraan dan biodata penulis.

**1. Tampilan Antar Muka Form Info**




**3.27 Gambar Form Info**

## 2. Deskripsi Objek form Info

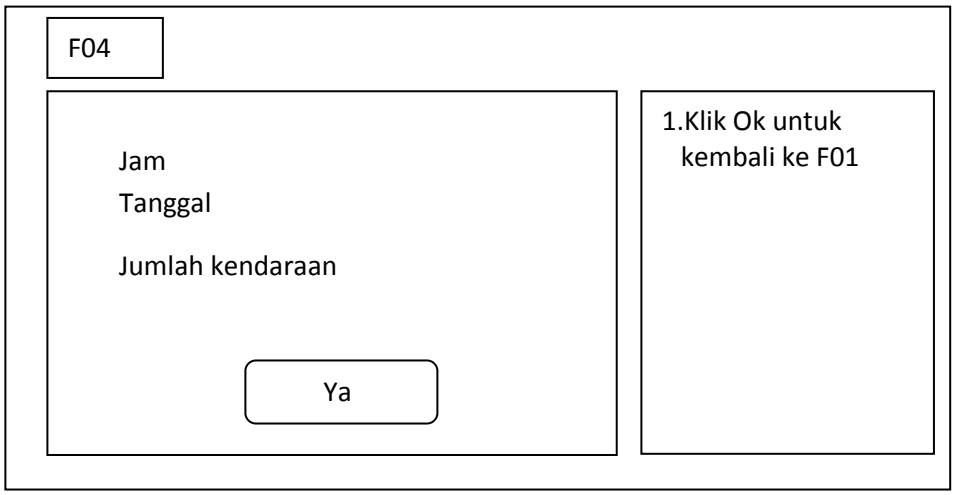
**3.3-14.Tabel Deskripsi Objek form Info**

| Objek | Jenis | Keterangan |
|---|---|---|
| Tentang Program | Memo | Keterangan tentang program dan penulis |
| OK | Button | Kembali ke menu F01 |

### 3.2.3.5 Perancangan Antar Muka Form Laporan

Form ini berisikan laporan dari jumlah kendaraan yang terhitung oleh program penghitung jumlah kendaraan menggunakan kamera.

## 1. Tampilan Antar Muka Form Laporan




**Gambar 3.3.28 Gambar Form Laporan**

## 2. Deskripsi Objek Form Laporan

Deskripsi dari gambar perancangan antar muka laporan diatas dijelaskan pada tabel berikut :

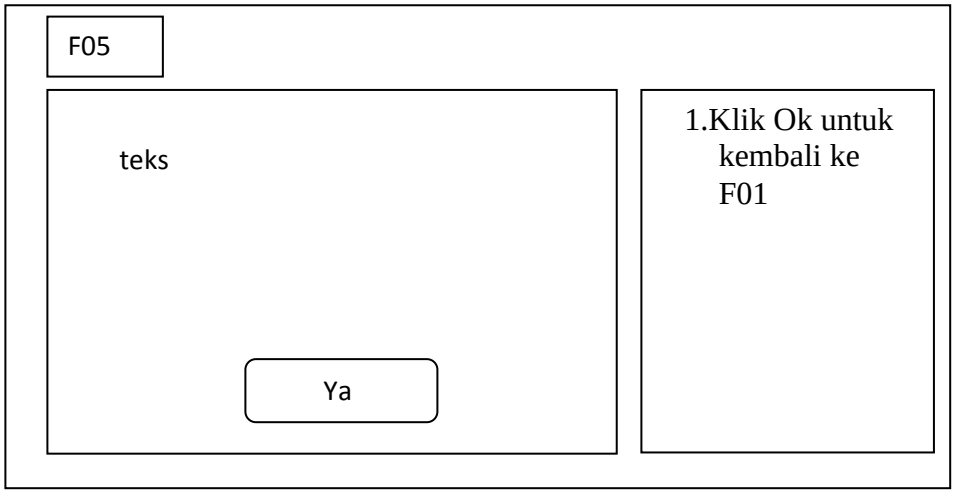
**Tabel 3.3-15 Keterangan Gambar Perancangan Antar Muka Laporan**

| Objek | Jenis | Keterangan |
|---|---|---|
| Jam | Time | Menunjukan waktu pada saat pembuatan laporan |
| Tanggal | Date | Menunjukan tanggal pada saat pembuatan laporan. |
| Jumlah Kendaraan | Number | Menunjukan jumlah kendaraan yang terdeteksi pada saat pembuatan laporan |

### 3.2.3.6 Perancangan Antar Muka Form Panduan

Form ini berisikan panduan mengenai penggunaan dari perangkat lunak penghitung jumlah kendaraan menggunakan kamera.

**1. Tampilan Antar Muka Panduan**




**Gambar 3.3.29 tampilan Form Panduan**

1. **Deskripsi Objek Form Panduan**

   Deskripsi dari gambar perancangan antar muka panduan diatas dijelaskan pada tabel berikut :

**Tabel 3.15 Keterangan Gambar Perancangan Antar Muka Laporan**

| Objek | Jenis | Keterangan |
|---|---|---|
| Teks | Teks | Panduan mengenai penggunaan program penghitung jumlah kndaraan menggunakan kamera |

### 3.2.4 Perancangan Pesan

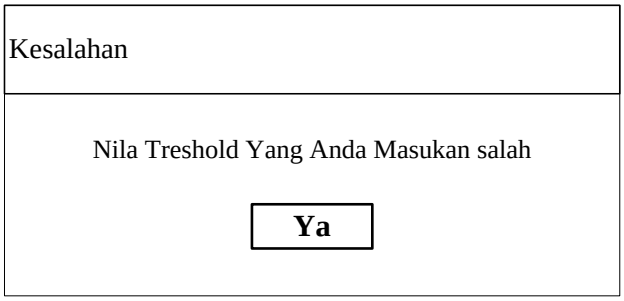
Perancangan pesan diperuntukan untuk memberikan peringatan kepada *user* jika *user* malakukan kesalahan dalam melakukan *input* data.

Berikut adalah tampilan pesan yang di rancang pada sistem perangkat lunak penghitung jumlah kendaraan menggunakan kamera:

1. Perancanagan Pesan Salah Nilai Treshold

   Pesan ini akan muncul apabila *user* belum memasukan nilai threshold pada sistem perangkat lunak penghitung jumlah kendaraan menggunakan kamera.

   Berikut adalah tampilan dari perancangan pesan salah nilai *threshold*:

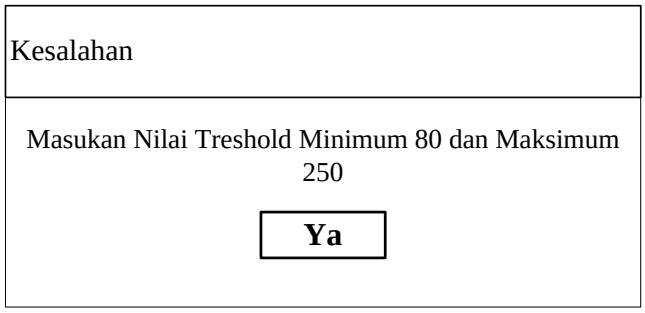



**Gambar 3.3.30 Perancangan Pesan Nilai Treshold**

2. Perancangan Pesan Nilai Treshold yang di tentukan

   Pesan ini akan muncul apabila *user* melakukan kesalahan dalam meng*input*kan nilai *threshold* atau melebihi nilai *treshold* yang telah ditentukan pada sistem perangkat lunak penghitung jumlah kendaraan menggunakan kamera.

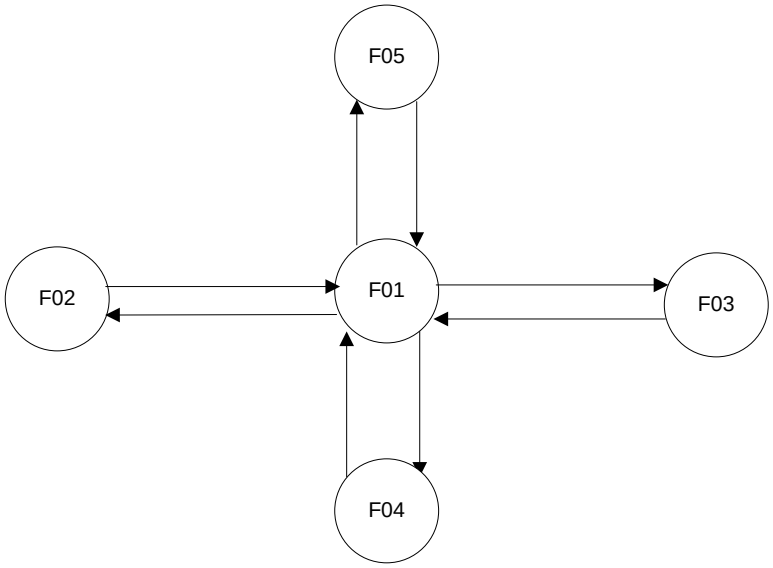
   Berikut adalah tampilan dari perancangan pesan nilai *threshold* yang ditentukan:




**Gambar 3.3.31 Perancangan Pesan Nilai Treshold Yang Ditentukan**

### 3.2.5 Jaringan Semantik

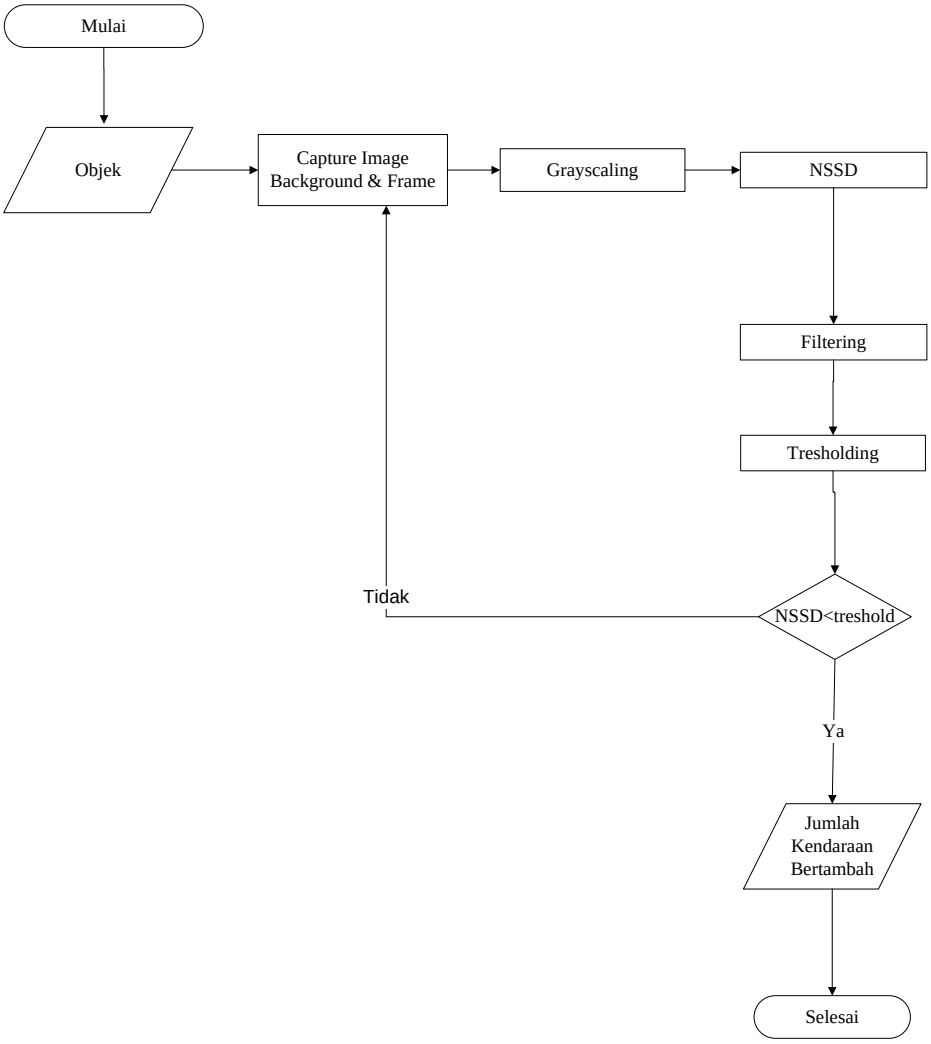
Jaringan semantik merupakan gambaran pengetahuan grafis yang menunjukkan hubungan antar berbagai objek. Jaringan semantik terdiri dari lingkaran-lingkaran yang menunjukkan objek dan informasi tentang objek-objek tersebut. Jaringan semantik dari program perangkat lunak penghitung jumlah kendaraan menggunakan kamera akan dijelaskan pada gambar berikut :




**Gambar 3.3.32 jaringan Simantik Perangkat Lunak Penghitung Jumlah Kendaraan menggunakan Kamera**

### 3.2.6 Perancangan Prosedural

Berikut adalah flowchart dari sistem penghitung jumlah kendaraan menggunakan kamera.




**Gambar 3.3.33 Flow Chart Penghitung Jumlah Kendaraan Menggunakan Kamera**