

## TRAFFIC COUNTING STUDI KASUS DI JALAN TEUKU NYAK ARIEF

Fauzan Adhima <sup>(1)</sup>, Fathiah <sup>(2)</sup>

Prodi Teknik Informatika, Fakultas Ilmu Komputer, Universitas U'Budiyah Indonesia  
Jl. Alue Naga, Tibang, Kec. Syiah Kuala, Banda Aceh, Indonesia  
Email: Fauzanadhima7@gmail.com<sup>(1)</sup>

### ABSTRAK

Banda Aceh merupakan salah satu kota wisata yang sebagian besar penduduknya bertumpu pada mode transportasi untuk keperluan sehari-hari dari mulai sepeda, sepeda motor, mobil dan angkutan umum. Pertumbuhan populasi penduduk kota Banda aceh meningkat dari tahun ke tahun menyebabkan ketidak seimbangan antara jumlah transportasi yang ada di jalan raya dengan kapasitas ruas jalan yang tersedia. Hal ini menyebabkan berbagai masalah lalu lintas Sehingga diperlukan kegiatan Monitoring arus lalu lintas pada jalan tertentu untuk mengetahui kelancaran arus lalu lintas. maka dari itu munculah ide untuk merancang sebuah sistem *traffic counting* yang dapat menghitung jumlah arus lalu lintas kendaraan Sebagai pedoman untuk dijadikan informasi arus lalu lintas. Sistem ini dibuat dengan menggunakan *Microsoft Visual Studio*, dan *Open Source Computer Vision*. Hasil dari pengujian sistem ini dengan menggunakan file video yang sudah direkam pada jalan Teuku Nyak Arief Banda Aceh. Program ini secara keseluruhan menunjukkan keberhasilan lebih dari 80%. Dengan sistem ini diharapkan dapat memudahkan untuk menghitung jumlah kendaraan pada jalan tersebut.

Kata Kunci : Jumlah Kendaraan, Arus Lalu Lintas, *Microsoft Visual Studio* dan *Open Source Computer Vision*

### 1. PENDAHULUAN

#### 1.1 Latar Belakang

Banda Aceh merupakan salah satu kota wisata yang sebagian besar penduduknya bertumpu pada mode transportasi untuk keperluan sehari-hari dari mulai sepeda, sepeda motor, mobil dan angkutan umum. Pertumbuhan populasi penduduk kota Banda aceh meningkat dari tahun ke tahun menyebabkan ketidak seimbangan antara jumlah transportasi yang ada di jalan raya dengan kapasitas ruas jalan yang tersedia. Hal ini menyebabkan berbagai masalah lalu lintas Sehingga diperlukan kegiatan Monitoring arus lalu lintas pada jalan tertentu untuk mengetahui kelancaran arus lalu lintas.

Monitoring arus lalu lintas pada jalan jalan di kota Banda Aceh dirasa sangat diperlukan, karena dapat memberikan informasi kelancaran arus lalu lintas sehingga dengan mudah dapat mengetahui jumlah kendaraan yang melewati jalan tersebut.

Berdasarkan permasalahan diatas maka dari itu munculah ide untuk merancang sebuah sistem *traffic counting* yang dapat memonitoring dan menghitung jumlah arus kendaraan yang melewati jalan tersebut. Dan untuk memudahkan dalam melakukan penelitian ini maka jalan yang akan dilakukan monitoring adalah pada jalan Teuku Nyak Arief Banda Aceh. Sehingga Untuk dapat menghitung jumlah arus kendaraan maka terlebih dulu merekam video pada jalan tersebut dengan menggunakan kamera sehingga hasil rekaman bisa

di counting oleh program yang akan kita buat dan ditampilkan pada layar monitor komputer.

#### 1.2 Rumusan Masalah

Berdasarkan latar belakang diatas maka identifikasi masalah dalam penelitian ini adalah:

1. Bagaimana sistem ini dapat mendeteksi objek pada video.
2. Bagaimana sistem ini dapat menghitung jumlah arus lalu lintas.
3. Bagaimana sistem ini dapat memudahkan dalam memberi informasi tentang jumlah kendaraan yang melewati arus lalu lintas tersebut.

#### 1.3 Tujuan Penelitian

Tujuan yang ingin diperoleh dari skripsi ini yaitu,

1. Mampu mengidentifikasi objek pada video.
2. Dapat menghitung kendaraan yang melewati jalan tersebut.
3. Monitoring jumlah Arus lalu lintas pada jalan tersebut.
4. Dengan sistem *traffic counting* memudahkan dalam mengawasi jumlah kendaraan yang melewati jalan tersebut.

#### 1.4 Manfaat Penelitian

Adapun Manfaat dari penelitian ini adalah sebagai berikut:

1. Memberikan kemudahan untuk mengetahui jumlah kendaraan yang melewati jalan tersebut.
2. Memberi kemudahan dalam mengetahui kelancaran Arus lalu lintas.

#### 1.5. Keaslian Penelitian

Berdasarkan hasil penelusuran literatur terdapat beberapa penelitian sejenis yang telah dilakukan, maka penulis membuat daftar literatur yang telah diteliti oleh peneliti sebelumnya seperti yang terlihat pada Tabel 1.1

Tabel 1.1

No	Peneliti	Judul	Metode Penelitian
1.	Annisa rachman (2010)	Analisa Simulasi Kepadatan Lalu Lintas Pada Persimpangan Traffic Dengan Metode Deteksi Tepi Canny	Pengembangan sistem dibangun menggunakan Matlab.
2.	Samuel Christian Susanto, Budi Setiawan dan Erdhi Widyarto (2011)	Deteksi kepadatan jalan dengan citra digital pada maket jalan simpang empat	Pengembangan sistem dibangun menggunakan Matlab.
3.	Adam vrileuis (2013)	Pemantau lalu lintas dengan sensor LDR berbasis Mikrokontroler aTmega 16	Pengembangan sistem dibangun menggunakan sensor LDR berbasis Mikrokontroler aTmega16

Berdasarkan tabel 1.1 dapat disimpulkan bahwa penelitian ini berbeda dengan penelitian terdahulu penelitian pertama dan kedua menggunakan *Software* Matlab. Pada penelitian ketiga membahas tentang simulasi dari sistem pemantau lalu lintas dengan sensor LDR, penelitian ini menggunakan sensor LDR berbasis Mikrokontroler aTmega 16.

## 2. TINJAUAN PUSTAKA

### 2.1 Arus Lalu Lintas

Arus lalu lintas terbentuk dari pergerakan individu pengendara dan pengendara yang

melakukan interaksi antara yang satu dengan yang lainnya pada satu ruas jalan dan lingkungannya. Arus lalu lintas pada suatu ruas jalan karakteristiknya akan bervariasi baik berdasarkan lokasi maupun waktunya. selain itu perilaku pengemudi ikut mempengaruhi terhadap perilaku arus lalu lintas. pengemudi pada suatu ruas jalan yang dirancang dengan kecepatan tertentu misalkan 80 km/jam dimungkinkan bahwa pengemudi akan mempunyai kecepatan yang bervariasi dari 30 km/jam sampai 120 km/jam. Dalam menggambarkan arus lalu lintas secara kuantitatif dalam rangka untuk mengerti tentang keberagaman karakteristiknya dan rentang kondisi perilakunya. parameter arus lalu lintas dapat di bedakan menjadi dua bagian utama yaitu parameter makroskopik arus lalu lintas secara umum dan parameter makroskopik yang menunjukkan tentang perilaku kendaraan individu dalam suatu arus lalu lintas yang terkait dengan antara yang satu dengan yang lainnya. suatu arus lalu lintas secara makroskopik dapat digambarkan tiga parameter utama, yaitu : volume dan arus, kecepatan dan kepadatan.

### 2.2 Kamera

Sebuah kamera terdiri dari pixel-pixel berupa *photodiode* yang bertugas menangkap energi cahaya (photon) yang dikonversikan kedalam bentuk energi listrik. Energi listrik yang berbentuk voltase dan amplitudo diperkuat kedalam level tertentu yang kemudian diproses menjadi bentuk digital oleh perangkat yang dinamakan *Analog to Digital Converter* (ADC).

*Pixel photodiode* sensitif terhadap cahaya dan mampu mengukur tingkat brightness dari cahaya itu sendiri. Karena *photodiode* adalah *device monokrom*, maka tidak mungkin mereka mengenali perbedaan dari setiap panjang gelombang cahaya yang diterima. Oleh karena itu, dibuatlah sebuah sistem penyaring warna berupa pattern-pattern mosaic yang disebut *dengan Color Filter Mosaic* (CFM). Saat ini terdapat banyak kamera digital profesional untuk para photographer.

Beberapa istilah penting yang umum ditemui pada kamera yaitu :

1. Megapixels dan resolusi adalah jumlah maksimum pixel yang bisa ditangkap sebuah sensor. 1 megapixels sama saja dengan seribu pixel, makin banyak pixel yang dimiliki dapat berarti pula makin lengkap detail sebuah gambar yang terekam pada sensor. Hal ini tentu berpengaruh pada saat kita melihat pada monitor, menyimpannya maupun mencetaknya. Untuk mencetak gambar pada ukuran 40x60 inchi diperlukan 12 megapixelss atau lebih agar mendapat hasil yang optimal.
2. Bits adalah bilangan berbasis 2 yang memiliki nilai antara angka 0 atau 1 yang berkorespondensi dengan *switch ON* atau *OFF*

yang menyatakan suatu nilai tertentu. Dalam 1 bit gambar, kita dapat mengatakan bahwa binary 0 menyatakan black atau gelap, atau binary 1 menyatakan white atau terang. Dalam 2 bit gambar, kita mendapatkan 4 (2) kemungkinan nilai, yaitu nilai yang mewakili warna (tone): 00 (black), 01(gray), 10 (gray) dan 11 (white). Dalam 8 bit gambar (1 byte), kita mendapatkan  $2^8 = 256$  kemungkinan nilai warna, yaitu antara range 00000000 (0) - 11111111 (255) . Gambar BMP sering disebut sebagai gambar 24 bit, sebab menyimpan informasi sebesar 8 bit pada setiap warna utama (R=256, G=256, B=256), sehingga sering disebut dengan istilah 16.7 million color (mendekati warna asli) yang didapat dari perkalian  $256 \times 256 \times 256$ .

3. *Burst rate* pada beberapa kamera memiliki kemampuan menangkap beberapa frame dalam waktu yang singkat, sangat berguna pada saat mengambil gambar yang membutuhkan kecepatan tinggi seperti pertandingan sepak bola. Dipengaruhi beberapa hal seperti kecepatan *processor*, *buffer*, dan *storage* bahkan lensa.
4. *Shutter lag* adalah jarak waktu jeda ketika pelepasan *shutter* dan selama sensor merekam gambar.
5. *Buffer* setelah sebuah sensor mengexpose gambar, gambar tersebut akan diproses oleh kamera dan disimpan kedalam *storage*. *Buffer* dalam kamera merupakan jenis memory yang secara sementara menyimpan gambar sebelum ditulis ke *storage*. Ini akan meningkatkan waktu antara ketika melakukan shot terutama ketika kita melakukan pemotretan dalam mode burst. Penggunaan *buffer* diperlukan karena kecepatan *storage* memory penyimpanan akhir yang lebih lambat tidak mampu mengimbangi kecepatan proses dari kamera. Saat ini hampir semua kamera digital memiliki *buffer* yang cukup besar yang memungkinkan kamera beroperasi dengan cepat seperti kamera film konvensional, artinya kamera dapat menyimpan gambar kedalam *storage* dalam sebuah background process. Lokasi *buffer* dalam kamera tidak selalu terspesifikasikan dengan jelas, tetapi pasti ada karena berpengaruh terhadap kecepatan saat melakukan pemotretan terutama pada mode burst continuous. Lokasi *buffer* sebuah kamera pada umumnya bekerja pada saat sebelum atau sesudah *image processing*.
6. *After Image Processing Buffer*  
Dengan metode ini gambar akan diproses terlebih dahulu baru kemudian masuk ke *buffer* sebelum disimpan kedalam *storage* dan konsekuensinya banyaknya jumlah gambar

yang disimpan dalam *buffer* dipengaruhi jenis format gambar yang dipilih.

7. *Before Image Processing Buffer*  
Dalam metode ini tidak ada pemrosesan gambar ketika gambar akan dimasukkan ke dalam *buffer*. Gambar yang telah diexpose oleh sensor akan langsung masuk ke *buffer* tanpa diproses terlebih dahulu, artinya pemilihan jenis format sebuah gambar tidak akan berpengaruh langsung terhadap kapasitas sebuah *buffer*.
8. *Smart Buffering*  
Mengkombinasikan kedua metoda diatas. Seperti halnya "*Before Image Processing Buffer*", gambar yang baru di expose akan langsung masuk ke *buffer* (1), kemudian gambar akan diproses oleh *image processor* (2) dan dikonversi ke dalam bentuk JPEG, TIFF atau RAW Perbedaannya, daripada menulis gambar yang telah diproses langsung ke *storage*, gambar hasil pemrosesan/ konversi ditulis kembali ke *buffer* (3). Dengan demikian *image processing* tidak akan terpengaruh oleh *bottleneck* yang ada terjadi didalam *storage card*. Lebih dari itu setelah gambar diproses oleh *image processor* (2), *space* yang dipakai dalam *buffer* akan langsung dikosongkan untuk hasil pemrosesan tahap 3, yang artinya akan memberi ruang lebih terutama untuk pemrosesan gambar yang dikonversi sebagai format JPEG. Kemudian setelah diproses di tahap 3, gambar baru akan ditulis di *storage*, dan tentunya akan dihapus dari *buffer*. Hal terpenting disini, pemrosesan gambar dilakukan paralel dengan proses penulisan gambar ke *storage*. Artinya pemrosesan sebuah gambar baru, dapat dilanjutkan ketika gambar sebelumnya sedang ditulis ke *storage*. Alhasil kita tidak perlu menunggu proses penulisan ke *storage* selesai untuk proses pengambilan gambar berikutnya, semuanya bekerja secara paralel. ( Askurifai Baksin, 2009 )

### 2.3 Open Computer Vision

*Computer Vision* adalah pencitraan komputer dimana aplikasi tidak melibatkan manusia dalam proses pengulangan visual. Dengan kata lain, gambar yang diperiksa dan di olah oleh komputer. Meskipun orang yang terlibat dalam pengembangan sistem aplikasi akhirnya membutuhkan komputer untuk mengambil informasi visual secara langsung. (Umbaugh, 1998).

*Computer vision* merupakan sebuah proses otomatis yang mengintegrasikan sejumlah besar proses persepsi visual, seperti pengolahan citra, klasifikasi citra, pengenalan citra dan akuisisi citra. *Computer vision* didefinisikan sebagai salah satu cabang ilmu pengetahuan yang mempelajari

bagaimana komputer dapat mengenali obyek yang diamati atau diobservasi. Cabang ilmu ini bersama kecerdasan buatan (*Artificial Intelligence*) akan mampu menghasilkan sistem kecerdasan visual (*Visual Intelligence System*) (Munir, 2004).

Proses-proses dalam *computer vision* dapat dibagi menjadi tiga aktivitas yaitu :

1. Memperoleh atau mengakuisisi citra digital.
2. Melakukan teknik komputasi untuk memproses atau memodifikasi data citra.
3. Menganalisis dan menginterpretasi citra dan menggunakan hasil pemrosesan untuk tujuan tertentu, misalnya memandu robot, mengontrol peralatan, memantau proses manufaktur, dan lain-lain.

#### 2.4 Microsoft Visual Studio

*Microsoft Visual Studio* adalah sebuah *integrated development environment* (IDE) yang digunakan untuk mengembangkan sebuah konsol dan graphical user interface yang meliputi *Windows Forms*, *website*, *web application*, dan *web service*, *desktop application*.

*Visual Studio* mendukung beberapa macam bahasa pemrograman built-in yang diantaranya adalah C/C++ (via Visual C++), VB.NET (via Visual Basic.NET), C# (via Visual C#), dan F# (Visual Studio 2010). Bahasa pemrograman lain seperti Phyton dan Ruby juga didukung dengan menginstall *language service* yang diinstal secara terpisah. *Visual Studio* juga mendukung XML/XSLT, HTML/XHTML, JavaScript dan CSS.

### 3. METODOLOGI PENELITIAN

#### 3.1 Metodologi Penelitian

Jenis penelitian yang digunakan penulis dalam mengerjakan skripsi ini adalah studi literatur dari berbagai sumber. Studi literatur dilakukan untuk mendapatkan metode yang tepat untuk merancang sistem yang akan dibuat. Penulisan skripsi ini juga menggunakan *Systems Development Life Cycle* (SDLC), yaitu melalui beberapa tahapan yaitu :

1. Perencanaan
2. Analisa
3. Rancangan
4. Implementasi

#### 3.2. Perencanaan

Penulis Terlebih dahulu melakukan perencanaan dengan melakukan Teknik pengumpulan data. Dan untuk mendapatkan informasi yang relevan, Penulis melakukan studi pustaka dengan mencari informasi maupun teori dari beberapa sumber referensi, jurnal dan artikel. Selain itu penulis juga menggunakan media *cyber* untuk mencari sumber tuntunan dalam proses pembuatan penelitian ini.

#### 3.3. Analisa

Setelah dilakukan perencanaan maka penulis akan melakukan Analisa dengan observasi peninjauan langsung di lapangan, hal ini dilakukan untuk memperoleh sejumlah informasi secara akurat. Dalam hal ini peneliti mencoba melakukan pengamatan langsung di jalan-jalan besar Kota Banda Aceh Sehingga Penulis mengambil kesimpulan dan akan melakukan pengambilan rekaman Video pada Jalan Teuku Nyak Arief.

#### 3.4. Alat dan bahan

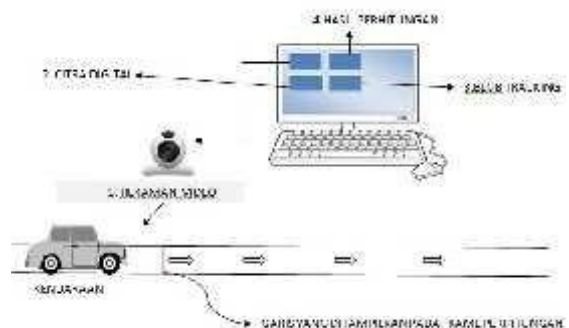
Peralatan yang digunakan dalam penelitian ini di spesifikasikan dalam *hardware* dan *Software*, yaitu sebagai berikut :

1. Perangkat keras atau *hardware* yang terdiri dari :
  - a. Perangkat komputer dengan spesifikasi *Lenovo Processor AMD A8-6410, Harddisk 500 GB, RAM 4.00 GB*. Perangkat ini nanti akan digunakan sebagai alat perancangan aplikasi ini.
  - b. Kamera, Perangkat ini nantinya akan digunakan sebagai alat untuk merekam jalan dalam pengujian aplikasi ini.
2. Perangkat lunak atau *Software*, yang terdiri dari :
  - a. Operating System : *Windows 10 Pro*
  - b. *Microsoft Visual Studio*.
  - c. *Open Source Computer Vision*.

#### 3.5. Perancangan Sistem Aplikasi

Perancangan system merupakan tahap awal dari perancangan perangkat lunak atau aplikasi. Perancangan ini dilakukan untuk mengetahui kondisi sistem secara umum.

Sistem *Traffic counting* yang akan dibuat dengan terlebih dahulu dilakuakan perekaman sebuah video arus lalu lintas pada Jalan Teuku Nyak Arief, Untuk dijadikan sebagai pedoman program Aplikasi sehingga memudahkan dalam pembuatan program untuk menghitung Arus lalu lintas sehingga Dapat di analogikan sebagai berikut :



Gambar 3.1. Sistem *Traffic counting*

Dari gambar diatas dapat disimpulkan bahwa hasil Rekaman video Arus Lalu Lintas Akan di putar

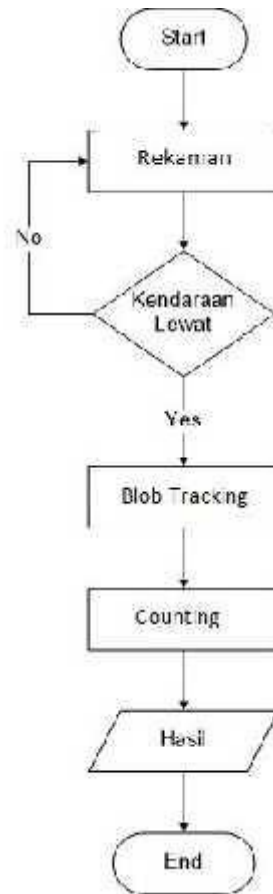
pada frame yang pertama dan selanjutnya akan diproses oleh frame citra digital untuk membedakan objek yang bergerak dengan gambar awal video dan blob tracking mudah dalam memisahkan tiap-tiap gumpalan yang terdapat dalam suatu gambar dan kemudian melabelinya Sehingga hasil perhitungan dengan Sistem *Traffic counting* akan bekerja dengan beberapa tahap seperti kendaraan yang melintasi garis yang sudah ditentukan dan Sistem *Traffic counting* akan melakukan perhitungan jumlah arus kendaraan yang sudah direkam pada jalan Teuku Nyak Arief

**3.6. Pemodelan Sistem Aplikasi**

Tahap pemodelan sistem aplikasi bertujuan untuk menspesifikan aspek solusi perancangan, untuk mempermudah dalam menterjemahkan proses alur sistem ke dalam bahasa pemrograman dan melakukan sebuah analisa desain, dan kerjanya suatu aplikasi yang kemudian dituangkan ke dalam logika. Sehingga dapat memudahkan penulis dalam membuat Aplikasi nantinya. Pada bagian ini dideskripsikan diagram seperti *flowchart*.

**3.6.1. Pemodelan Dengan Flowchat**

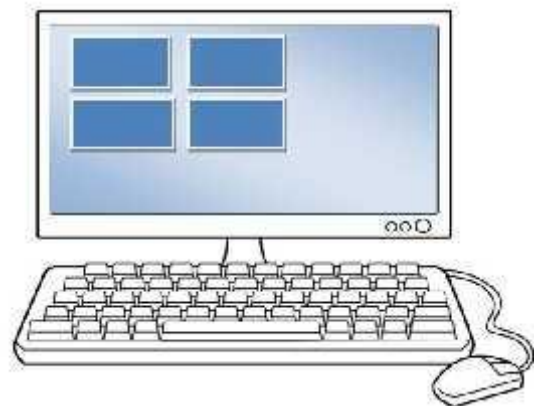
Gambaran proses alur kerja sistem aplikasi secara keseluruhan digambarkan dalam *flowchart*. Hal ini untuk menjelaskan langkah demi langkah jalannya system *traffic counting*. Langkah langkah tersebut dapat dilihat pada Gambar 3.2, *flowchart* sistem Traffic counting.



Gambar 3.2. *flowchart* sistem Traffic counting.

**3.7. Rancangan Interface Aplikasi**

Aplikasi yang akan dirancang ini mempunyai tampilan seperti dibawah ini



Gambar 3.3. Tampilan Aplikasi

Demikian kesimpulan inti pada Aplikasi terdapat input dan output yang di butuhkan sebagai berikut :

1. Rekaman awal yang ditampilkan.

Rekaman awal ditampilkan pada frame yang akan dibuat sehingga akan dijadikan pedoman untuk menentukan jalannya Aplikasi.

2. Citra Digital.  
Rekaman awal dari video akan diubah menjadi biner yang memuat intensitas piksel yang ada di dalamnya sehingga bias membedakan objek yang bergerak
3. Blob tracking.  
Untuk mempermudah dalam memisahkan tiap-tiap gumpalan yang terdapat dalam suatu gambar dan kemudian melabelinya sehingga antara gumpalan yang satu dengan gumpalan yang lainnya dapat dibedakan dengan identitas yang berbeda.
4. Hasil perhitungan arus lalu lintas  
Pada perancangan hasil perhitungan arus lalu lintas, *blob* hasil dari proses

*blobtracking* dibandingkan posisi titik pusatnya dengan garis hitung yang telah ditetapkan. Setiap titik pusat melewati garis yang telah ditetapkan maka akan menambah jumlah kendaraan yang berhasil dihitung.

### 3.8. Implementasi.

Beberapa ujicoba dilakukan untuk mendapatkan input dan output yang sesuai dengan yang diharapkan. Pengujian meliputi fitur-fitur yang terdapat dalam Aplikasi untuk memastikan bahwa fungsi-fungsi yang dirancang dapat berjalan sesuai rencana.

## 4. HASIL PENELITIAN DAN PEMBAHASAN

### 4.1 Komponen Dalam Implementasi Sistem

Agar sistem perancangan yang telah kita kerjakan dapat berjalan baik atau tidak, maka perlu kiranya dilakukan pengujian terhadap sistem yang telah kita kerjakan. Untuk itu dibutuhkan beberapa komponen utama mencakup perangkat keras (*hardware*), perangkat lunak (*software*), dan perangkat operator (*brainware*).

### 4.2. Implementasi Sistem

Implementasi sistem ini bertujuan untuk memastikan komponen-komponen dari sistem telah berfungsi sesuai dengan yang di harapkan. Uji coba perlu dilakukan untuk mencari kesalahan-kesalahan yang mungkin masih terjadi serta merupakan pengetesan dari sistem secara keseluruhan. Sistem telah di tes terlebih dahulu berdasarkan program yang telah di integrasikan,

### 4.3. Tampilan Utama

Pada halaman utama aplikasi Menampilkan 4 frame , Yaitu frame Input , frame Citra Digital, frame *Blob Tracking* dan frame Hasil Perhitungan yang saling berintegrasi dengan program untuk mengetahui jumlah kendaraan, Maka *content* yang berhubungan dengan aplikasi dapat berjalan.

Beberapa fitur yang terdapat pada halaman utama adalah sebagai berikut :



Gambar 4.1. Tampilan Utama

### 4.3.1. Frame Input

Dalam perancangan ini frame akan menampilkan file video awal untuk dijadikan pedoman frame Citra Digital, Berikut Gambar Frame Input :



Gambar 4.2. Frame Input

### 4.3.2. Frame Citra digital

Dalam perancangan ini frame dalam video akan ditampilkan secara berurutan sehingga membentuk suatu gambar bergerak. Sehingga proses grayscale, setting detection window dan menjumlahkan nilai pixel dilakukan pada masing-masing frame. Posisi *detection window* pada frame dan pada *background image* harus sama karena pada posisi inilah yang akan dilakukan perbandingan.

Dengan adanya *detection window* memudahkan proses perhitungan jumlah nilai pixel yang dilakukan oleh komputer. Bilamana tidak menggunakan *detection window*, komputer akan melakukan perhitungan pada frame secara frame video adalah sekumpulan image yang ditampilkan secara berurutan sehingga membentuk suatu gambar bergerak. Sehingga *proses grayscale*, *setting detection window* dan menjumlahkan nilai pixel dilakukan pada masing-masing frame. Posisi *detection window* pada frame dan pada *background image* harus sama karena pada posisi inilah yang akan dilakukan perbandingan.

Sehingga bisa memuat intensitas piksel yang ada di dalamnya. Sedangkan output dari *background subtraction* berupa gambar

*foreground* yang nantinya akan diproses dalam proses *blobtracking*, dan selisih dari jumlah nilai pixel pada background image dan pada frame. Jadi Selisih jumlah nilai pixel, memungkinkan untuk menghasilkan nilai negatif sehingga hasil selisih tersebut dikuadratkan untuk menghindari hasil negatif. Nilai negatif terjadi bilamana warna dari obyek pada frame lebih gelap daripada background image. Sebab warna gelap memiliki nilai pixel yang kecil.



Gambar 4.3. Citra Digital

#### 4.3.3. Frame Blob Tracking

Pada perancangan *Blob Tracking* untuk mempermudah dalam memisahkan tiap-tiap gumpalan yang terdapat dalam suatu gambar dan kemudian melabelinya sehingga antara gumpalan yang satu dengan gumpalan yang lainnya dapat dibedakan dengan identitas yang berbeda. Berikut Gambar Program Blob Tracking.



Gambar 4.4. Blob Tracking

#### 4.3.4. Hasil Perhitungan

Dalam proses penghitungan kendaraan ini dibagi menjadi beberapa bagian pengaturan metode penghitungan, pengaturan garis hitung dan proses penghitungan sehingga memudahkan dalam menghitung kendaraan.

Pada perancangan metode penghitungan kendaraan, *blob* hasil dari proses *blobtracking* dibandingkan posisi titik pusatnya dengan garis hitung yang telah ditetapkan. Setiap titik pusat melewati garis yang telah ditetapkan maka akan menambah jumlah kendaraan yang berhasil dihitung.



Gambar 4.8. Hasil Perhitungan

#### 4.3.5. Pengujian

Pengujian yang dilakukan terdiri dari beberapa bagian meliputi pengujian Citra Digital, *Blob Tracking* dan Pengujian hasil penghitungan kendaraan dengan menggunakan file rekaman video pada jalan Teuku Nyak Arief Banda Aceh.

Pengujian *Blob Tracking* dikatakan berhasil apabila setiap kendaraan yang melintas dideteksi oleh program sebagai objek. Kemungkinan lain yang akan terjadi jika objek yang terlalu berdekatan antar objek yang lain maka program akan mendeteksi menjadi satu buah objek dan juga dua objek yang terdeteksi sebagai satu *blob* disebabkan karena kendaraan yang terdeteksi melewati secara bersamaan.

Dalam pengujian ini dilakukan perbandingan perhitungan dengan menggunakan program *Traffic counting* Serta penghitungan secara manual. Hasil yang didapatkan dibandingkan untuk mendapatkan ketepatan program dalam menghitung jumlah kendaraan. Pengujian dilakukan dengan menggunakan satu buah file video yang sudah direkam pada Jalan Teuku Nyak Arief Banda Aceh, Hasil perhitungan dapat dilihat pada Tabel 4.1.

Tabel 4.1. Hasil Perhitungan

Manual	Program Traffic Counting	Tingkat Keberhasilan (%)
5	5	100 %
10	8	80 %
15	11	73 %
20	15	75 %
25	20	80 %
30	24	80 %
35	28	80 %
40	31	77%
Keberhasilan Program Traffic Counting		80 %

Hasil penghitungan dalam Pengujian Penghitung Kendaraan secara Manual Kendaraan yang diambil dari mulai 5 kendaraan sampai 40 kendaraan dengan selisih 5 kendaraan.

Hasil penghitungan dengan menggunakan



Pengujian *Program Traffic counting*, Kendaraan yang berhasil dihitung dengan beragam jumlah yang terdeteksi sehingga Dapat disimpulkan Keberhasilan *Program Traffic Counting* tersebut dalam menghitung adalah sebesar 80%.

## 5. KESIMPULAN DAN SARAN

### 5.1. Kesimpulan

Pada bab ini di paparkan kesimpulan dari pengerjaan tugas akhir ini berdasarkan hasil perancangan *Traffic Counting* studi kasus di Jalan Teuku Nyak Arief, yaitu:

1. Rekaman video yang digunakan menggunakan Resolusi video 320x180 dan Untuk melakukan proses *background subtraction*.
2. Kendaraan yang melewati dan terlalu berdekatan antar kendaraan yang lain maka program akan mendeteksi menjadi satu buah kendaraan.
3. Hasil yang didapatkan tidak Realtime Karena masih menggunakan file hasil rekaman video pada Jalan Teuku Nyak Arief.
4. Secara keseluruhan sistem bekerja cukup baik hal ini dibuktikan dengan tingkat keberhasilan 80% pada penghitungan .

### 5.2. Saran

Saran yang dapat di rekomendasikan oleh penulis dalam menyelesaikan tugas akhir ini adalah sebagai berikut:

1. Diharapkan menggunakan rekaman yang Resolusi Video yang lebih tinggi.
2. Diharapkan bisa dijalankan secara Realtime.
3. Diharapkan Kepada pengembang agar dapat dijalankan pada Mobile.

## 6. DAFTAR PUSAKA

Susanto Azhar, 2004. *Sistem Informasi Manajemen*, Bandung : Linggar jaya.

Christian Susanto, Budi Setiawan dan Erdhi Widyarto, 2011. *Deteksi kepadatan jalan dengan citra digital pada maket jalan simpang empat*, <http://core.ac.uk/download/files/476/12342453.pdf> diakses pada 21 Desember 2015.

Grundgeiger, 2002, *Programming Visual Basic.NET*

Kurniawan, Erick. 2011, *Cepat Mahir Visual Basic*, Yogyakarta : Andi.

Munir, 2004, *Pengolahan Citra Digital dengan Pendekatan Algoritmik*, Bandung: Informatika

Nixon dan Aguado, 2002, *Deteksi Tepi Unsur Pada Citra*, Universitas Sumatera Utara.

Rachman, Annisa, 2010, *Analisa Simulasi Kepadatan Lalu Lintas Pada Persimpangan Traffic Dengan Metode Deteksi Tepi Canny*, <http://eprints.upnjatim.ac.id/1848/1/file1.pdf> diakses pada 20 Desember 2015.

Syafi'i, Slamet imam, 2011. <http://slametux.blogdetik.com/2011/10/19/open-computer-vision-opency>. diakses pada 29 Desember 2015.

Tamin, Ofyar Z, 2000, *Perencanaan dan Pemodelan Transportasi*, ITB, Bandung

Umbaugh, 1998, *Computer vision and Image processing*.

Vrileuis, Adam, 2013. *Pemantau lalu lintas dengan sensor LDR berbasis Mikrokontroler ATmega16*, <http://download.portalgaruda.org/article.php?article=143261&val=3941&itle=Pemantau%20Lalu%20Lintas%20dengan%20Sensor%20LDR%20Berbasis%20Mikrokontroler%20ATmega16> diakses pada 22 Desember 2015.