

# Optimasi Pengatur Siklus Waktu Lampu Lalu-Lintas Dinamik Menggunakan Metode Optimasi *Goal Programming* dengan *Java* dan *Opencv* untuk Meminimalisir Kesenjangan Kepadatan Lalu-Lintas (Studi Kasus: Persimpangan Jl. A. Yani – Jl. Jenggala – Jl. Gubernur Suryo, Sidoarjo)

Rizky Praditya, Wiwik Anggraeni, S.Si, M.Kom

Sistem Informasi, Fakultas Teknologi Informasi (FTIf), Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

*e-mail*: wiwik@its-sby.edu

**Abstrak**—Peningkatan jumlah kepadatan lalu-lintas sedang menjadi perhatian khusus saat ini. Salah satu faktor signifikan yang mempengaruhi kepadatan lalu-lintas adalah sistem pengaturan siklus waktu lampu lintas. Perbaikan dalam sistem pengaturan siklus waktu lampu lintas yang mampu mengatur waktu siklusnya secara dinamis dan optimal akan menjadi suatu hal yang cukup penting di masa depan. Hal ini akan sangat terasa kegunaannya pada saat kepadatan lalu-lintas di salah satu jalur persimpangan terjadi. Siklus waktu lampu lalu-lintas bisa disesuaikan secara otomatis sesuai densitas (kepadatan) kendaraan yang ada pada lajur-lajur jalan yang ada di sekitar persimpangan jalan. Oleh karena itu dalam tugas akhir ini kami menawarkan sebuah sistem yang mampu mengatur waktu siklus lampu lalu-lintas secara dinamis dan optimal sebagai salah satu jalur dalam meminimalisir kesenjangan kepadatan lalu-lintas.

Tujuan penelitian ini merujuk pada studi kasus persimpangan yang memiliki tiga buah jalur. Dalam hal optimasi kami menggunakan metode *Goal Programming* yang dapat menyelesaikan permasalahan dengan fungsi tujuan beragam dan banyak. Metode ini memodelkan fungsi tujuan, batasan tujuan, batasan fungsional, bobot, pinalti serta solusi optimal model sistem lalu-lintas persimpangan studi kasus. Kami menggunakan metode *Haar Feature Cascade Classifier* dalam mengidentifikasi objek kendaraan yang melewati persimpangan tersebut dan menggunakan *machine vision OpenCV* dalam memproses hasil *training cascade* tersebut. Serta menggunakan bahasa pemrograman *Java* dalam membuat sistem input dinamis dari *machine vision OpenCV*, memproses iterasi *simplex Goal Programming*, dan menyajikan informasi hasil solusi siklus waktu lampu lalu-lintas dan kepadatan kendaraan setiap jalur pada persimpangan studi kasus dalam sebuah *interface multithreading* yang di-*refresh* setiap detik.

Hasil optimasi dengan menggunakan model *Goal Programming* didapatkan durasi lampu hijau optimal dibandingkan dengan kondisi realita pada setiap jalur pada persimpangan sesuai kondisi lalu-lintas dengan memperhatikan reduksi kendaraan dan waktu tunggu serta batasan-batasan yang dimilikinya. Hasil optimasi lebih baik dibandingkan realita dengan melihat aspek Penurunan Redudansi kendaraan sebesar 45.95%, Percepatan Waktu Tunggu sebesar 39.43%, Penurunan Kendaraan Tersisa sebesar 29.88% dan Penurunan kendaraan Tersisa Terakhir sebesar 52.55%, namun lebih buruk pada aspek Peningkatan Kendaraan Melintas sebesar 41.50% dikarenakan oleh beberapa kondisi.

**Kata Kunci**— Optimasi, *Goal Programming*, Kemacetan, *Java*, Kendaraan, *OpenCV*, Pengatur Lampu Lalu-Lintas, *Haar Feature*, *Cascade Classifier*, *Machine Vision*

## I. PENDAHULUAN

PENGAWASAN dan pengendalian lalu-lintas sedang menjadi masalah utama di banyak Negara. Jumlah pengguna jalan dan kendaraan meningkat secara terus menerus di seluruh dunia. Sedangkan fasilitas jalan terbatas seperti jumlah jalan, lebar jalan, dan kapasitas jalan [1]. Oleh karena itu diperlukan peningkatan pengaturan sistem lalu-lintas yang baik supaya kondisi lalu-lintas tetap terjaga lancar dan jumlah kesenjangan kepadatan lalu-lintas dapat ditekan seminimal mungkin. Salah satu sarana dalam pengaturan lalu-lintas adalah lampu lalu-lintas yang berguna untuk mengatur aliran dan arah kendaraan-kendaraan yang sedang melintas di persimpangan jalan. Sebagai bagian vital dari sistem lalu-lintas, keberadaan lampu lalu-lintas tidak dapat dipungkiri. Pemberlakuan tiga warna (merah, kuning, hijau) pada lampu lalu-lintas juga telah menjadi standar umum internasional dan berlaku secara global [2].

Namun, pengoperasian lampu lalu-lintas bukanlah tanpa masalah. Siklus waktu lampu lalu-lintas (merah - kuning - hijau) saat ini kebanyakan masih diatur secara konstan dan manual. Misal, berapa detik lampu merah menyala atau lampu kuning maupun lampu hijau. Contoh masalah adalah lampu lalu-lintas tidak akan menyesuaikan lamanya *delay* dengan kepadatan kendaraan yang berubah-ubah sepanjang hari. Sehingga sekalipun arus lalu-lintas pada suatu lajur jalan sedang sepi (kepadatan rendah) lamanya *delay* waktu siklus tidak berbeda dengan lama *delay* disaat keadaan arus lalu-lintas pada lajur jalan tersebut sedang ramai (kepadatan tinggi). Padahal idealnya, pada lajur jalan yang kepadatan arus kendaraannya tinggi warna hijau harusnya memiliki *delay* yang lebih lama dibandingkan dengan lajur yang kepadatan kendaraannya rendah. Tentu saja hal ini sangat berguna untuk memberikan kesempatan lebih banyak kepada kendaraan-kendaraan yang melewati lampu lalu-lintas pada lajur yang kepadatan kendaraan lebih tinggi tersebut [2].

Dengan adanya suatu sistem pengaturan siklus waktu lampu lintas yang mampu mengatur waktu siklusnya secara otomatis akan menjadi suatu hal yang cukup penting di masa depan. Hal ini akan sangat terasa kegunaannya pada saat kepadatan lalu-lintas di persimpangan terjadi. Siklus waktu lampu lalu-lintas bisa disesuaikan secara otomatis sesuai densitas (kepadatan) kendaraan yang ada pada lajur-lajur jalan yang ada di sekitar persimpangan jalan [2].

Sebuah penelitian melakukan pengembangan algoritma optimasi pada pengatur lampu lalu-lintas statis menggunakan *Dynamic Programming* menghasilkan suatu algoritma yang memiliki waktu eksekusi 0.99 detik [3]. Sedangkan penelitian lain terkait Sistem Pengaturan Lampu Lalu-lintas secara Sentral dari Jarak Jauh menghasilkan suatu sistem yang mampu menjadikan pengatur lampu lalu-lintas menjadi pra-dinamik dengan bantuan seorang operator yang bertugas menginputkan arus dari kendaraan yang melintasi pada setiap jalur persimpangan tersebut [4].

Dalam mempertimbangkan berbagai hasil penelitian yang telah dilakukan, kami mencoba membuat sistem pengatur siklus waktu lalu-lintas yang optimal dan dinamis dengan metode optimasi *Goal Programming* dan inputan dinamis berupa visual yang akan dideteksi menggunakan metode *Haar Feature Cascade Classifier* pada *OpenCV*. Inputan visual digunakan karena sistem pendeteksi kepadatan lalu-lintas yang paling banyak dipakai saat ini adalah melalui CCTV dalam bentuk visual. Sedangkan *Goal Programming* dipilih karena mampu menyelesaikan permasalahan lalu-lintas yang memiliki batasan tujuan lebih dari satu serta waktu eksekusinya pada *Java* hanya 26 *milliseconds* [5], dalam artian memungkinkan untuk dijalankan di sistem nyata.

## II. TINJAUAN PUSTAKA

### A. Lalu-Lintas

PP No.43 Th. 1993 mendefinisikan bahwa pengelolaan lalu-lintas meliputi perencanaan, pengaturan, pengawasan, dan pengendalian lalu-lintas. Perencanaan lalu-lintas meliputi penetapan tingkat pelayanan yang diinginkan, inventarisasi dan evaluasi tingkat pelayanan, penetapan pemecahan masalah lalu-lintas, serta penyusunan rencana dan program pelaksanaannya pada komponen lalu-lintas terdiri dari manusia (pengguna jalan), kendaraan, dan jalan yang saling berkaitan satu sama lain yang saling mempengaruhi [6].

Persimpangan pada studi kasus tugas akhir ini memiliki empat jalur seperti yang ditunjukkan pada Gambar 2.1. Sebelah utara merupakan Jl. Jenggolo, sebelah selatan merupakan Jl. Gubernur Suryo serta sebelah timur merupakan Jl. Untung Suropati. Namun karena Jl. Untung Suropati hanya memiliki satu ruas jalan (bukan jalan umum) dan pengaturan siklus waktu lampu lalu-lintasnya sama seperti pada Jl. Gubernur Suryo, maka Jl. Untung Suropati sengaja tidak dimasukkan kedalam model studi kasus.



Gambar III Persimpangan Studi Kasus

### B. Goal Programming

*Goal Programming* merupakan metode optimasi hasil pengembangan dari *Linear Programming* yang mampu menyelesaikan masalah optimasi yang memiliki fungsi tujuan lebih dari satu dengan memberikan variabel deviasi pada tiap batasan tujuan yang ada [7].

Bentuk umum dari *Goal Programming* dengan faktor prioritas di dalam strukturnya adalah sebagai berikut [2]:

- Fungsi Tujuan

$$\min \sum_{i=1}^n P_x W_i^+ d_i^+ + P_y W_i^- d_i^-$$

- Batasan Tujuan

$$\sum_{i=1}^n a_{ij} X_j + d_i^- - d_i^+ = b_i$$

- Batasan Fungsional

$$\sum_{i=1}^n g_{kj} X_j \leq \text{atau} \geq \text{atau} = C_k$$

- Keterangan:

- $X_j$  : Variabel keputusan
- $b_i$  : Batasan Tujuan
- $a_{ij}$  : Koefisien fungsi batasan tujuan
- $d_i^-$  : Variabel deviasi yang mempresentasikan tingkat pencapaian dibawah target
- $d_i^+$  : Variabel deviasi yang mempresentasikan tingkat pencapaian diatas target
- $W_{i,y}^-$  : Bobot deviasi yang mempresentasikan tingkat pencapaian dibawah target
- $W_{i,y}^+$  : Bobot deviasi yang mempresentasikan tingkat pencapaian diatas target
- $g_{ij}$  : Koefisien fungsi batasan fungsional
- $C_k$  : Sumberdaya k yang tersedia sesuai batasan
- $P_{xy}$  : Faktor prioritas

### C. Java

*Java* terkenal dengan kelengkapan *library* / perpustakaan (kumpulan program program yang disertakan dalam pemrograman *java*) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas *Java* yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi [8].

Java memiliki beberapa *library* yang berisikan fungsi-fungsi dalam menyelesaikan masalah *Linear Programming*, salah satunya adalah *SPCSolver* yang menyediakan beragam fungsi *simplex* untuk menyelesaikan permasalahan *Linear Programming* dengan menggabungkan dua fungsi *library* lain *GLPKSolverPack* dan *LPSOLVE.SolverPack* [5].

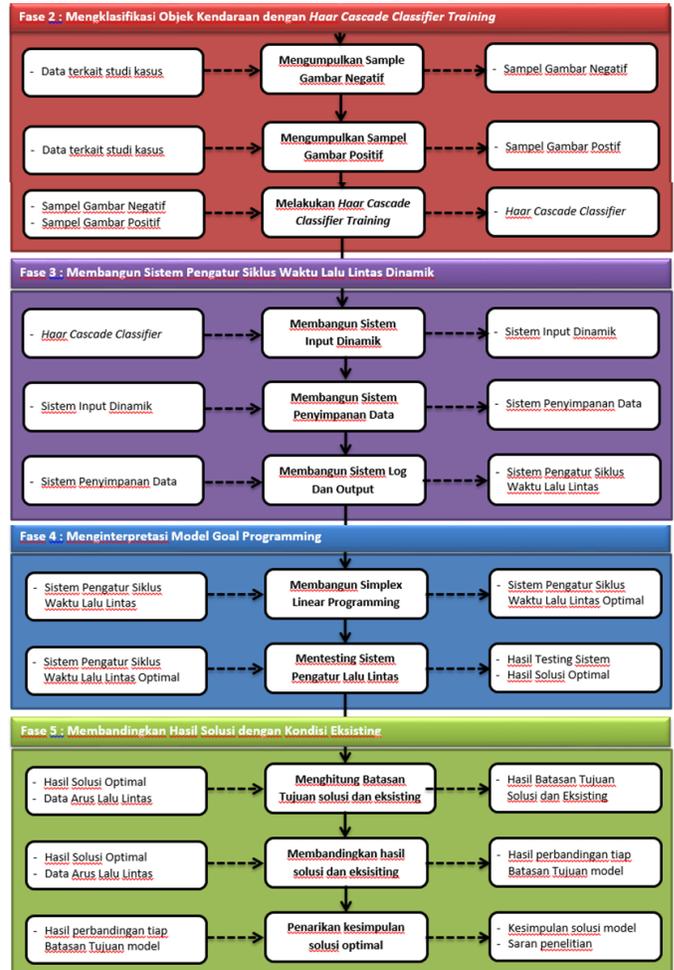
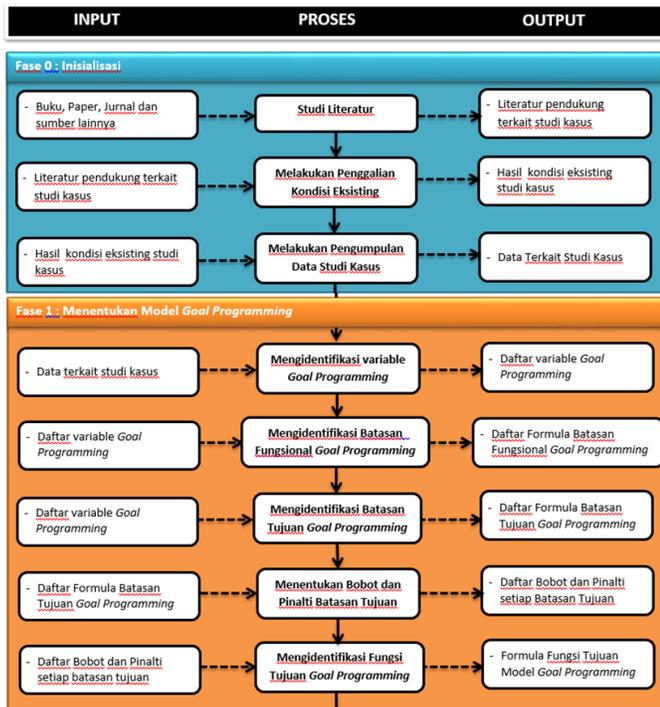
**D. OpenCV**

*OpenCV (Open Source Computer Vision Library)* merupakan sebuah *library machine learning* dan *computer vision open source* yang dibuat untuk menyediakan infrastruktur untuk aplikasi *computer vision*. *OpenCV Library* tersedia dalam berbagai bahasa pemrograman seperti C, C++, *Phyton* dan *Java* yang mulai dirilis pada *OpenCV 2.4.9*. Dengan berlisensi sebagai produk BSD, *OpenCV* mudah digunakan untuk bisnis dan untuk dikembangkan [9].

*OpenCV* memiliki lebih dari 2500 algoritma optimasi termasuk algoritma *Cascade Classifier*. Algoritma ini dapat digunakan untuk mendeteksi wajah, mengidentifikasi objek, mengklasifikasi pergerakan manusia, mengikuti pergerakan kamera, mencatat pergerakan objek bergerak, mengekstrak 3D model suatu objek, menemukan gambar serupa dalam database, mengikuti gerakan kelopak mata dan masih banyak lainnya. Salah satu metode pelatihan klasifikasi ini adalah *Haar Feature Training* [10].

Untuk melakukan proses *Haar Feature Cascade Classifier Training* diperlukan proses pengumpulan gambar sampel positif dan negatif. Sampel positif berisikan objek gambar yang ingin dideteksi sedangkan sampel negative berisikan objek gambar yang tidak ingin dideteksi [11].

**III. METODE PENELITIAN**



Gambar 2 Diagram Alir Penelitian

**A. Fase 0: Inisiasi**

Fase inisiasi terdiri dari Studi literatur, Penggalan Kondisi Eksisting serta Melakukan Pengumpulan Data Studi Kasus dilakukan untuk menggali, mengidentifikasi dan memahami permasalahan pada studi kasus secara lebih dalam, memilih variabel penelitian yang akan digunakan dalam pembuatan model riset operasi, serta metode yang akan digunakan dalam penelitian ini.

**B. Fase 1: Permodelan**

Fase permodelan terdiri dari Mengidentifikasi Variable *Goal Programming*, Mengidentifikasi Batasan Fungsional *Goal Programming*, Mengidentifikasi Batasan Tujuan *Goal Programming* serta Mengidentifikasi Fungsi Tujuan *Goal Programming* untuk menghasilkan model optimasi yang akan diterapkan pada studi kasus.

**C. Fase 2: Klasifikasi**

Fase klasifikasi terdiri dari Mengumpulkan Sampel Gambar Positif, Mengumpulkan Sampel Gambar Negatif serta Mentraining Sampel Gambar berisi Objek dengan metode *Haar Feature Cascade Classifier Training* untuk menghasilkan pengklasifikasi objek kendaraan sesuai studi kasus. .

**D. Fase 3: Konstruksi**

Fase konstruksi terdiri dari Membangun Sistem Input Dinamik, Membangun Sistem Repositori serta Membangun

Sistem *Log* dan *Output* dengan bahasa pemrograman *Java* untuk menghasilkan alat pemroses input dan visualisasi.

**E. Fase 4: Interpretasi**

Fase interpretasi terdiri dari Membangun *Simplex Linear Programming* dan *Men-testing* Sistem Pengaturan Siklus Lampu Lalu-Lintas Dinamik dengan aplikasi berbasis *Java* yang telah dikonstruksi pada fase 3.

**F. Fase 5: Analisa**

Fase analisa terdiri dari Menghitung Batasan Tujuan Model Solusi dan Eksisting, Perbandingan Hasil Solusi dan Eksisting serta Penarikan Kesimpulan Solusi Optimal.

**IV. PERMODELAN**

**A. Pembuatan Model Umum**

Di dalam membuat model, dibutuhkan data masukan yang didapat dari proses pengamatan pada persimpangan. Data masukan tersebut kemudian akan diformulasikan melalui beberapa tahap ke dalam bentuk *Goal Programming*. Model ini bisa berbeda apabila di implementasikan pada kondisi persimpangan selain studi kasus. Berikut adalah perancangan model untuk permasalahan dalam penelitian ini.

**Fungsi Tujuan**

- Meminimalkan redudansi jumlah kendaraan yang bisa melewati persimpangan jalur ke-1  $(f_1)$
- Meminimalkan redudansi jumlah kendaraan yang bisa melewati persimpangan jalur ke-2  $(f_2)$
- Meminimalkan redudansi jumlah kendaraan yang bisa melewati persimpangan jalur ke-3  $(f_3)$
- Meminimalkan waktu tunggu kendaraan pada jalur ke-1  $(f_4)$
- Meminimalkan waktu tunggu kendaraan pada jalur ke-2  $(f_5)$
- Meminimalkan waktu tunggu kendaraan pada jalur ke-3  $(f_6)$

**Variabel Keputusan**

- Lama siklus lampu lalu-lintas  $(Ct)$
- Lama nyala lampu hijau pada jalur ke-i dalam suatu siklus  $(tg_i)$
- Lama nyala lampu hijau pada jalur ke-i dalam suatu siklus  $(tr_i)$
- Jumlah kendaraan pada jalur ke-i dalam suatu siklus  $(v_i)$
- Kapasitas jalan pada jalur ke-i  $(rc_i)$
- Lama waktu tunggu maksimum kendaraan pada jalur ke-i dalam suatu siklus  $(wt_i)$
- Estimasi frekuensi jumlah kendaraan melewati lampu lalu-lintas pada jalur ke-i dalam satu detik  $(f_i)$
- Estimasi periode waktu yang dibutuhkan satu kendaraan untuk melewati lampu lalu-lintas pada jalur ke-i  $(p_i)$
- Jumlah kendaraan pada jalur ke-i periode-1 dalam suatu siklus  $(v'_i)$

**Batasan**

- Jumlah lampu hijau setiap jalur di persimpangan  $(Cg)$
- Lama waktu minimal lampu hijau jalur ke-i  $(Cm_i)$
- Kesetaraan lama waktu lampu merah jalur ke-i dengan jumlah lampu hijau di jalur lain  $(Cr_i)$

- Kesetaraan lama siklus waktu lampu lalu-lintas jalur ke-i dengan lama siklus waktu di jalur lain  $(Cs_i)$

**B. Pembobotan Model Goal Programming**

Setelah mendapatkan persamaan dalam bentuk *Linear Programming*, setiap batasan tujuan dan batasan fungsional disesuaikan ke dalam bentuk *weighted Goal Programming*. Fungsi tujuan pada model *Goal Programming* akan terbentuk dari kumpulan variabel deviasi batasan tujuan.

**Batasan Tujuan**

- Batasan Tujuan 1: Meminimalkan redudansi jumlah kendaraan yang bisa melewati persimpangan jalur ke-1  $\frac{tg_1}{p_1} + d_1^- - d_1^+ = v_1$   $(49)$

- Batasan Tujuan 2: Meminimalkan redudansi jumlah kendaraan yang bisa melewati persimpangan jalur ke-2  $\frac{tg_2}{p_2} + d_2^- - d_2^+ = v_2$   $(50)$

- Batasan Tujuan 3: Meminimalkan redudansi jumlah kendaraan yang bisa melewati persimpangan jalur ke-3  $\frac{tg_3}{p_3} + d_3^- - d_3^+ = v_3$   $(51)$

- Batasan Tujuan 4: Meminimalkan waktu tunggu kendaraan jalur ke-1  $\sum_{i=0}^n (v_i - v'_i)(tg_1 + tr_1) + d_4^- - d_4^+ = 0$   $(52)$

- Batasan Tujuan 5: Meminimalkan waktu tunggu kendaraan jalur ke-2  $\sum_{i=0}^n (v_i - v'_i)(tg_2 + tr_2) + d_5^- - d_5^+ = 0$   $(53)$

- Batasan Tujuan 6: Meminimalkan waktu tunggu kendaraan jalur ke-3  $\sum_{i=0}^n (v_i - v'_i)(tg_3 + tr_3) + d_6^- - d_6^+ = 0$   $(54)$

**Fungsi Tujuan**

Dengan mengidentifikasi setiap batasan tujuan yang ada serta bobot dan pinalti masing-masing, maka terakhir perumusan fungsi tujuan dari model *Goal Programming* sebagai berikut:

$$\min \left( \begin{matrix} P_1 d_1^+ + P_1 d_1^- + P_2 d_2^+ + P_2 d_2^- + P_3 d_3^+ \\ + P_3 d_3^- + P_4 d_4^- + P_5 d_5^- + P_6 d_6^- \end{matrix} \right) \quad (94)$$

**V. PERANCANGAN DAN IMPLEMENTASI**

**A. Pengumpulan Data**

Pengumpulan data merupakan hal mutlak yang harus dilakukan di dalam penelitian ini. Proses pengumpulan data pada Persimpangan Jl. Jendral Ahmad Yani - Jl. Jenggolo - Jl. Gubernur Suryo, Sidoarjo dilakukan dengan melakukan pengamatan dan perekaman lalu-lalang kendaraan pada ketiga jalur yang terlibat di dalam arus lalu-lintas.

Tabel 1  
Data Lampu Lalu-Lintas

Jalan	Durasi Lampu (detik)		
	Hijau	Merah	Kuning
Jl. Jendral A. Yani	47.6	46.3	2.23
Jl. Jenggolo	35.2	58.7	2.23

Jl. Gubernur Suryo	58.7	35.2	2.23
--------------------	------	------	------

Pada penelitian ini, objek penelitian dispesifikkan kedalam pengaturan siklus lampu lalu lintas guna meminimalisir kesenjangan kepadatan arus lalu-lintas. Tabel 5.1 menunjukkan data-data masukan yang diperlukan didalam membuat model optimasi *Goal Programming* yang selanjutnya akan diolah menggunakan aplikasi berbasis *Java*.

**B. Pengklasifikasi Objek Kendaraan**

Pengumpulan sampel gambar positif dilakukan dengan memilih gambar yang berisikan objek - objek kendaraan seperti motor dan mobil. Setelah dikumpulkan selanjutnya adalah melakukan pemotongan (crop) kordinat sesuai objek kendaraan tanpa *background*. Gambar 5.1 merupakan beberapa hasil pemotongan kordinat kendaraan pada masing-masing jalur.



Gambar 3 Hasil Sampel Positif

Pengumpulan sampel gambar negatif dilakukan dengan memilih gambar yang berisikan objek non-kendaraan seperti *background*. Tidak seperti gambar positif yang perlu dilakukan pemotongan kordinat, pengumpulan sampel negative tidak memerlukan pemotongan. Gambar 5.4, merupakan beberapa hasil pengumpulan gambar negatif, bahwa tidak hanya gambar *background* yang dimasukan kedalam sampel gambar negatif, namun gambar potongan objek kendaraan (tidak lengkap) ataupun yang bukan merupakan kendaraan pun masuk kedalam sampel gambar negatif.



Gambar 4 Hasil Sampel Negatif

Setelah didapatkan dua persyaratan dalam proses *training Haar Cascade Classifier* yakni sampel gambar positif dan sampel gambar negatif, maka selanjutnya melakukan proses *training Haar Cascade Classifier*. Proses *training* menggunakan fitur *Haar Training* pada *OpenCV* dengan bantuan kode *batch* pada *Windows* yang nantinya menghasilkan *file* pengklasifikasi objek kendaraan berformat *.xml*. Tabel 5.3 menunjukkan hasil dari proses *training Haar Cascade Classifier* pada masing-masing jalur.

**C. Penyelesaian Model dengan Java**

Setelah mendapatkan model untuk *simplex Linear Programming*, selanjutnya adalah memasukkan model tersebut kedalam program *Java library SPC Solver*.

**VI. HASIL DAN PEMBAHASAN**

Dengan melihat perbedaan hasil jumlah kendaraan yang tersisa, jumlah kendaraan menunggu dan waktu tunggu terdapat lima aspek perbedaan dengan masing-masing metriknya antara hasil optimasi dan realita pada Jl. Jendral Ahmad Yani Tabel 6-1., Jl. Jenggala Tabel 6-2, Jl. Gubernur Suryo Tabel 6-3 dan seluruh jalur Persimpangan Studi Kasus pada Tabel 6-4 yang selanjutnya dianalisis lebih lanjut.

Tabel 2  
Hasil Perbandingan pada Jl. Jendral Ahmad Yani

No.	Pembeda	Metriks	Solusi <i>Goal Programming</i>	Kondisi Eksisting
1	Redudansi Kendaraan	Total	393	333.693693
		Rerata	56.14285714	83.4234234
2	Waktu Tunggu	Total	11666.655	21575.8
		Rerata	1666.665	5393.95
3	Kendaraan Melintas	Total	393	336
		Rerata	56.14285714	84
4	Kendaraan Tersisa	Total	173	305
		Rerata	24.71428571	76.25
		Terakhir	72	129
5	Lampu Hijau	Rerata	32.4675	46.3

Hasil optimasi dengan menggunakan model *Goal Programming* pada Jl. Jendral Achmad Yani lebih baik dilihat dari aspek Redudansi Kedaraan yang lebih sedikit 32.70%, Waktu Tunggu yang lebih cepat 69.10%, Kendaraan Tersisa yang lebih sedikit 67.59% dan Kendaraan Tersisa Terakhir yang lebih sedikit 44.19%. Namun kurang baik apabila dilihat dari aspek Kendaraan Melintas yang lebih sedikit 33.16%, karena hasil optimasi mengikuti jumlah kendaraan yang ada pada jalur tersebut, dan saat itu jumlahnya tidak banyak sehigga durasi lampu hijau menjadi tidak lama dan kendaraan yang melintas sesuai dengan durasi tersebut.

Tabel 3  
Hasil Perbandingan pada Jl. Jenggala

No.	Pembeda	Metriks	Solusi <i>Goal Programming</i>	Kondisi Eksisting
1	Redudansi Kendaraan	Total	468	423.063063
		Rerata	58.5	105.765765
2	Waktu Tunggu	Total	14393.37	15907.7
		Rerata	1799.17125	3976.925
3	Kendaraan Melintas	Total	468	424
		Rerata	58.5	106
4	Kendaraan Tersisa	Total	197	54
		Rerata	24.625	13.5
		Terakhir	21	65
5	Lampu Hijau	Rerata	32.4675	58.7

Hasil optimasi dengan menggunakan model *Goal Programming* pada Jl. Jenggala lebih baik dilihat dari aspek Redudansi Kedaraan yang lebih sedikit 44.69%, Waktu Tunggu yang lebih cepat 54.76% dan Kendaraan Tersisa Terakhir 67.69%. Namun kurang baik apabila dilihat dari aspek Kendaraan Melintas yang lebih sedikit 44.81% dan Kendaraan Tersisa. Untuk Kendaraan Tersisa yang lebih banyak 82.41% karena hasil optimasi meminimalisir kesenjangan kepadatan

setiap jalur persimpangan, pada Jl. Jendral Achmad Yani diturunkan dan sebaliknya menaikkan jumlah kendaraan tersisa pada Jl. Jenggala ini. Sedangkan pada aspek Kendaraan Melintas karena hasil optimasi mengikuti jumlah kendaraan yang ada pada jalur tersebut, dan saat itu jumlahnya tidak banyak sehingga durasi lampu hijau menjadi tidak lama dan kendaraan yang melintas sesuai dengan durasi tersebut.

Tabel 4  
Hasil Perbandingan pada Jl. Gubernur Suryo

No.	Pembeda	Metriks	Solusi <i>Goal Programming</i>	Kondisi Eksisting
1	Redudansi Kendaraan	Total	98.03361345	169.027611
		Rerata	12.25420168	42.2569027
2	Waktu Tunggu	Total	1289.888	7673.6
		Rerata	161.236	1918.4
3	Kendaraan Melintas	Total	33	31
		Rerata	4.125	7.75
4	Kendaraan Tersisa	Total	1	0
		Rerata	0.125	0
		Terakhir	0	2
5	Lampu Hijau	Rerata	10.20775	35.2

Hasil optimasi dengan menggunakan model *Goal Programming* pada Jl. Gubernur Suryo lebih baik dilihat dari aspek Redudansi Kendaraan yang lebih sedikit 71% dan Waktu Tunggu yang lebih cepat 91.60%. Namun kurang baik apabila dilihat dari aspek Kendaraan Melintas yang lebih sedikit 46.77%, karena hasil optimasi mengikuti jumlah kendaraan yang ada pada jalur tersebut, dan saat itu jumlahnya tidak banyak sehingga durasi lampu hijau menjadi tidak lama dan kendaraan yang melintas sesuai dengan durasi tersebut. Serta memiliki hasil yang sama pada aspek Kendaraan Tersisa

Tabel 5  
Hasil Perbandingan Seluruh Jalur Persimpangan

No.	Pembeda	Metriks	Solusi <i>Goal Programming</i>	Kondisi Eksisting
1	Redudansi Kendaraan	Total	959.033613	925.784368
		Rerata	41.697114	77.148697
2	Waktu Tunggu	Total	27349.913	45157.1
		Rerata	26.14714436	43.17122371
3	Kendaraan Melintas	Total	953	850
		Rerata	41.43478261	70.83333333
4	Kendaraan Tersisa	Total	371	359
		Rerata	32.4675	46.3
		Terakhir	93	196
5	Iterasi	Siklus	8	4
		Periode	25	14

Hasil optimasi dengan menggunakan model *Goal Programming* pada keseluruhan jalur di persimpangan Jl. Jendral Achmad Yani – Jl. Jenggala – Jl. Gubernur Suryo, Sidoarjo lebih baik dilihat dari aspek Redudansi Kendaraan yang lebih sedikit 45.95%, Waktu Tunggu yang lebih cepat 39.43%, Kendaraan Tersisa yang lebih sedikit 41.50% dan Kendaraan Tersisa Terakhir yang lebih sedikit 29.88%. Namun kurang baik apabila dilihat dari aspek Kendaraan Melintas yang lebih banyak 52.55%, karena hasil optimasi mengikuti jumlah kendaraan yang ada pada jalur tersebut, dan saat itu jumlahnya tidak banyak sehingga durasi lampu hijau menjadi tidak lama dan kendaraan yang melintas sesuai dengan durasi tersebut.

## VII. KESIMPULAN/RINGKASAN

Berdasarkan proses penelitian yang telah dilakukan, berikut

ini merupakan kesimpulan yang dapat diambil:

1. Metode *Goal Programming* mampu menjadi metode penyelesaian untuk kasus multi tujuan, yang mana dalam penelitian ini terkait pengoptimalan kesenjangan kepadatan lalu-lintas persimpangan Jl. Jendral Achmad Yani – Jl. Jenggala – Jl. Gubernur Suryo, Sidoarjo
2. Implementasi model optimasi, dapat memenuhi keseluruhan batasan tujuan dan batasan fungsional.
3. Hasil optimasi dengan menggunakan model *Goal Programming* didapatkan durasi lampu hijau optimal dibandingkan dengan kondisi realita pada Jl. Jendral Ahmad Yani, Jl. Jenggala, Jl. Gubernur Suryo serta setiap jalur pada persimpangan sesuai kondisi lalu-lintas dengan memperhatikan redudansi kendaraan dan waktu tunggu serta batasan-batasan yang dimilikinya. Hasil optimasi lebih baik dibandingkan realita dengan melihat aspek Penurunan Redudansi kendaraan sebesar 45.95%, Percepatan Waktu Tunggu sebesar 39.43%, Penurunan Kendaraan Tersisa sebesar 29.88% dan Penurunan kendaraan Tersisa Terakhir sebesar 52.55%, namun lebih buruk pada aspek Peningkatan Kendaraan Melintas sebesar 41.50% dikarenakan oleh beberapa kondisi.

Berikut ini merupakan saran untuk penelitian selanjutnya:

1. Sebaiknya melibatkan kriteria-kriteria kualitatif (*intangible*) di dalam model permasalahan *goal programming*. Seperti sisi psikologis pengguna jalan saat terjadi kemacetan maupun sedang dalam kondisi kecepatan tinggi.
2. Penggalan variabel lain yang berpengaruh maupun tidak berpengaruh secara signifikan. Seperti kendaraan darurat Mobil Pemadam Kebakaran, Ambulan dan Polisi, sehingga apa bila *machine vision* mendeteksi ada salah satu kendaraan tersebut di salah satu jalur, lampu lalu-lintas menjadi hijau.
3. Menggunakan *Machine Vision* selain yang lebih ringan, karena pengembangan *library OpenCV* membutuhkan daya CPU yang besar atau menyiapkan CPU minimal 6M 3.4 GHz.

## DAFTAR PUSTAKA

- [1] A. Triana and M. Aria, Simulasi Pengaturan Lampu Lalu Lintas Multi Persimpangan menggunakan Logika Fuzzy, Universitas Komputer Indonesia, 2009.
- [2] R. Hartono, D. W. Saputra and T. J. Hutasoit, "Penerapan Algoritma Greedy pada Optimasi Pengaturan Lampu Lalu Lintas Sederhana," pp. 2-4, 2006.
- [3] S. Samra, A. El-Mahdy and Y. Wada, "A Linear-Time and Space Algorithm for Optimal Traffic Signal Durations at an Intersection," 2011.
- [4] Tjia May On, P. B. Mardjoko and N. Martanto, "Sistem Pengaturan Lampu Lalu Lintas Secara Sentral Dari Jarak Jauh," vol. 9, 2007.
- [5] H. Planatscher and M. Schober, "Java Linear Programming Interface," Java, [Online]. Available: <http://scpsolver.org/>. [Accessed 9 2 2016].
- [6] Presiden Republik Indonesia, *Peraturan Pemerintah Republik Indonesia Nomor 43 Tahun 1993 Tentang Prasarana Dan Lalu Lintas Jalan*.
- [7] B. W. Taylor III, Introduction to Management Science, Ninth Edition, Virginia: Prentice Hall, 2006.

- [8] N. Jussien, G. Rochart and X. Lorca, "Choco: an *Open Source Java Constraint Programming Library*," *HAL*, no. hal-00483090, 2008.
- [9] "About *OpenCV*," itseez, 2016. [Online]. Available: <http://opencv.org/about>. [Accessed 9 2 2016].
- [10] R. M. Alhaqqi, N. Ramadijanti and Setiawardhana, "Finger Tracking untuk Interaksi pada Virtual Keyboard," *JUTI*, 2011.
- [11] S. Soo, "Object detection using *Haar-cascade Classifier*," *Institute of Computer Science, University of Tartu*, 2009.