

Optimasi Fase Durasi Hijau Lampu Lalu Lintas pada Persimpangan Jalan Padat Menggunakan Algoritma Pewarnaan Graf Welch-Powell

Naufal Hasbialhaq- 13525120

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung

E-mail: naufal.hasbialhaq18@gmail.com , 13525120@std.stei.itb.ac.id

Abstrak—Kemacetan lalu lintas di persimpangan jalan perkotaan merupakan masalah yang meresahkan dan mengakibatkan kerugian ekonomi dan menambah polusi udara. Makalah ini mengusulkan sebuah kerangka kerja untuk merancang sistem kendali lampu lalu lintas yang efisien dan aman pada persimpangan empat arah yang padat. Teori Graf digunakan untuk memodelkan pergerakan di persimpangan sebagai graf konflik tak berarah, simpul dari graf yang merepresentasikan masing-masing arus lalu lintas secara individu dan sisi pada graf menunjukkan konflik jalur spasial. Algoritma pewarnaan simpul Welch-Powell diterapkan untuk mengelompokkan arus-arus tersebut ke dalam jumlah fase bebas konflik yang minimum. Metode Webster diintegrasikan untuk menghitung Panjang siklus optimum secara dinamis dan mengalokasikan durasi lampu hijau berdasarkan data volume lalu lintas waktu nyata. Makalah ini menunjukkan bagaimana abstraksi matematika diskrit dapat direalisasikan secara efektif untuk menyelesaikan kendala infrastruktur fisik dalam perencanaan kota cerdas

Kata Kunci—Pewarnaan graf, Algoritma Welch-Powell, Metode Webster, optimasi lampu lalu lintas, bilangan kromatik

I. PENDAHULUAN

Pertumbuhan volume kendaraan bermotor di Kawasan perkotaan yang tidak diimbangi dengan perluasan jaringan infrastruktur jalan raya telah memicu masalah kemacetan lalu lintas yang meresahkan. Persimpangan jalan merupakan salah satu titik hambatan utama dalam sistem transportasi perkotaan. Pada titik-titik ini, berbagai arus kendaraan dari arah yang berbeda saling bertemu, berpotongan, dan berebut ruang jalan yang sama. Guna mengantisipasi risiko terjadinya kecelakaan dan mengatur pergerakan arus secara bergantian, dipasanglah sistem lampu lalu lintas.

Pengaturan lampu lalu lintas yang statis dan dirancang tanpa melalui analisis kuantitatif yang berbasis data volume arus riil, pengaturan durasi hijau dan merah yang tidak proporsional sering kali menyebabkan penumpukan kendaraan yang parah di satu lengan persimpangan, sementara lengan persimpangan lainnya mengalami kekosongan kendaraan saat lampu hijau menyala. Kondisi ini menurunkan tingkat pelayanan jalan dan memicu pemborosan bahan bakar serta menyebabkan polusi udara yang cukup masif

Penulisan karya ilmiah ini bertujuan untuk merancang sebuah solusi optimasi dua tingkat guna memecahkan masalah efisiensi lampu lalu lintas pada persimpangan jalan padat. Tingkat pertama berfokus pada optimasi struktur fase lampu lalu lintas menggunakan Teori Graf dan Algoritma Pewarnaan Graf Welch-Powell. Pendekatan ini digunakan untuk memodelkan hubungan konflik antar-arus kendaraan[4]. Tingkat kedua berfokus pada optimasi waktu menggunakan metode Webster [4] nyata untuk menghitung durasi detik lampu hijau yang paling optimal untuk setiap fase.

Dengan menggunakan dua metode tersebut, makalah ini tidak hanya menyajikan solusi teoritis yang elegan secara matematis, tetapi juga sebuah model rekayasa yang siap diimplementasikan untuk mendukung pengembangan sistem transportasi.

II. LANDASAN TEORI

A. Definisi Graf dan Terminologi Dasar

Secara formal, sebuah graf G didefinisikan sebagai pasangan set (V, E) , yang ditulis dengan notasi matematis[1]:

$$G=(V,E)$$

- V Adalah himpunan simpul yang tidak kosong dan merepresentasikan objek-objek diskrit dalam sistem[1].
- E Adalah himpunan sisi yang menghubungkan sepasang simpul dan merepresentasikan hubungan atau relasi biner antar objek tersebut[1].

Dua buah simpul dikatakan saling bertetangga jika terdapat sisi $e = (u,v) \in E$ yang menghubungkan keduanya. Sisi e tersebut dikatakan bersisian dengan simpul u dan v . Derajat dari suatu simpul v , yang disimbolkan dengan $d(v)$ Adalah jumlah total sisi yang berisian dengan simpul tersebut. Berdasarkan Lemma Jabat Tangan, jumlah derajat dari seluruh simpul di dalam graf selalu bernilai genap, yaitu tepat dua kali lipat dari total jumlah sisi graf[1]:

$$\sum d(v) = 2|E|.$$

B. Pewarnaan Simpul Graf dan Bilangan Kromatik

Pewarnaan simpul graf Adalah proses memberikan label (warna) kepada setiap simpul di dalam graf sedemikian rupa sehingga tidak ada dua simpul yang saling bertetangga memiliki warna yang sama[1]. Secara format, pewarnaan simpul pada graf $G=(V,E)$ menggunakan k warna adalah sebuah pemetaan

$$f : V \rightarrow \{1, 2, \dots, k\}$$

sedemikian rupa sehingga jika $(u, v) \in E$ maka $f(u) \neq f(v)$. Jumlah warna minimum yang dapat digunakan untuk mewarnai graf G disebut bilangan kromatik dari graf tersebut, yang dilambangkan dengan simbol $\chi(G)$ [1]. Masalah pencarian bilangan kromatik graf ini tergolong ke dalam kelas masalah yang tidak ada algoritma eksak waktu polinomial yang diketahui dapat menyelesaikan masalah ini untuk semua jenis graf berskala besar secara instan.

C. Algoritma Welch-Powell

Algoritma Welch-Powell merupakan salah satu algoritma heuristik yang sangat populer untuk menyelesaikan masalah pewarnaan graf dengan kompleksitas waktu kuadratik $O(v^2)$. Meskipun algoritma ini tidak selalu menjamin tercapainya bilangan kromatik yang paling minimal secara mutlak untuk seluruh variasi graf, ia terbukti memberikan batas atas (*upper bound*) jumlah warna yang sangat baik dan cepat untuk graf terapan di dunia nyata[4]

Langkah-langkah sistematis pelaksanaan Algoritma Welch-Powell adalah sebagai berikut[3] :

1. Urutkan seluruh simpul $v \in V$ dalam urutan menurun (*descending*) berdasarkan nilai derajatnya, sehingga diperoleh urutan:

$$d(v_1) \geq d(v_2) \geq \dots \geq d(v_n)$$

2. Gunakan warna pertama (misalnya warna 1) untuk mewarnai simpul pertama (v) di dalam daftar terurut.
3. Telusuri daftar simpul dari atas ke bawah. Berikan Warna 1 kepada setiap simpul yang tidak saling bertetangga dengan simpul-simpul yang telah diwarnai dengan Warna 1 sebelumnya.
4. Ulangi langkah kedua dan ketiga menggunakan warna berikutnya (Warna 2, Warna 3, dst.) untuk simpul-simpul yang masih belum memiliki warna, hingga seluruh simpul di dalam graf selesai diwarnai.

D. Metode Webster untuk Optimasi Waktu Lampu Lalu Lintas

Setelah struktur fase lampu lalu lintas berhasil dibentuk melalui pewarnaan graf, durasi waktu hijau optimum dihitung menggunakan Metode Webster[6]. Metode ini bertujuan meminimalkan total waktu tundaan kendaraan di persimpangan jalan[6].

Langkah-langkah dalam metode webster adalah sebagai berikut:

1. Hitung Rasio Arus Kritis (y_i) untuk setiap fase i yang terdiri dari kelompok arus kendaraan. Tentukan nilai rasio kritisnya, yaitu nilai maksimum dari perbandingan arus kendaraan riil di lapangan terhadap (q) arus jenis jalan (s):

$$y_i = \max\left(\frac{q}{s}\right) \text{ untuk seluruh fase } i$$

2. Hitung total rasio arus simpang (Y) : Jumlahkan seluruh rasio kritis dari setiap fase yang ada:

$$Y = \sum_{i=1}^n y_i$$

n menyatakan jumlah total fase lampu lalu lintas di persimpangan jalan. Nilai Y harus lebih kecil dari 1.0 agar persimpangan tidak mengalami kemacetan total.

3. Hitung total waktu hilang per siklus (L) : Waktu hilang dihitung berdasarkan *start-up* delay dan waktu *clearance* antarfase :

$$L = 2n + R$$

n adalah jumlah fase dan R adalah total waktu *all-red* di persimpangan

4. Hitung waktu siklus optimum (C_0) : Formulasikan waktu satu putaran penuh lampu lalu lintas yang paling efisien :

$$C_0 = \frac{1.5L + 5}{1 - Y}$$

5. Alokasikan waktu hijau efektif (G_i) : Distribusikan local waktu hijau efektif ($g_{tot} = C_0 - L$) secara proporsional ke masing-masing fase berdasarkan kontribusi rasio arusnya:

$$g_i = \frac{y_i}{Y} \times (C_0 - L)$$

III. METODOLOGI DAN PEMODELAN MASALAH

A. Deskripsi Kasus Persimpangan

Penelitian ini mengambil kasus sebuah simpang empat sebidang yang padat. Persimpangan ini menghubungkan empat ruas jalan utama, yaitu:

1. Lengan Utara (Jalan A)
2. Lengan Timur (Jalan B)
3. Lengan Selatan (Jalan C)
4. Lengan Barat (Jalan D)

Setiap pendekatan (*approach*) jalan memiliki lajur khusus yang membagi pergerakan kendaraan menjadi tiga arus utama, yaitu arus lurus, arus belok kanan, dan arus belok kiri.

B. Pemodelan Simpul Graf

Himpunan simpul V merepresentasikan total 12 arus pergerakan kendaraan yang ada di persimpangan jalan tersebut. Penamaan simpul disesuaikan dengan kode arah mata angin dan jenis pergerakannya:

- Lengan Utara (Jalan A) :
 - A_1 : Arus kendaraan dari utara bergerak lurus ke Selatan
 - A_2 : Arus kendaraan dari utara belok kanan ke barat
 - A_3 : Arus kendaraan dari utara belok kiri ke timur
- Lengan Timur (Jalan B) :
 - B_1 : Arus kendaraan dari timur bergerak lurus ke barat
 - B_2 : Arus kendaraan dari timur belok kanan ke utara
 - B_3 : Arus kendaraan dari timur belok kiri ke selatan
- Lengan Selatan (Jalan C) :
 - C_1 : Arus kendaraan dari selatan bergerak lurus ke utara
 - C_2 : Arus kendaraan dari selatan belok kanan ke timur
 - C_3 : Arus kendaraan dari Selatan belok kiri ke barat
- Lengan Barat (Jalan D) :
 - D_1 : Arus kendaraan dari barat bergerak lurus ke timur
 - D_2 : Arus kendaraan dari barat belok kanan ke selatan.
 - D_3 : Arus kendaraan dari barat belok kiri ke utara

C. Pemodelan Sisi Graf

Sebuah sisi tak-berarah $e = (u, v) \in E$ dibentuk untuk menghubungkan simpul u dan simpul v jika dan hanya jika jalur perlintasan fisik dari kedua arus tersebut saling memotong di dalam persimpangan, sehingga menimbulkan risiko tabrakan jika dijalankan secara bersamaan.

Berdasarkan geomteri persimpangan standar, gerakan belok kiri (A_3, B_3, C_3, D_3) diasumsikan gerakan bebas sehingga tidak memiliki titik konflik dengan arus lurus maupun kanan dari arah lain. Oleh karena itu, simpul-simpul belok kiri ini memiliki derajat $d(v) = 0$

Sebaliknya, arus gerakan lurus dan belok kanan saling memotong secara kompleks dirumuskan sebanyak 20 sisi konflik fisik di dalam persimpangan yang didefinisikan sebagai himpunan sisi E:

- (A_1, B_1) — Arus Utara Lurus memotong Timur Lurus.
- (A_1, D_1) — Arus Utara Lurus memotong Barat Lurus.
- (C_1, B_1) — Arus Selatan Lurus memotong Timur Lurus.
- (C_1, D_1) — Arus Selatan Lurus memotong Barat Lurus.

- (A_1, B_2) — Arus Utara Lurus memotong Timur Belok Kanan.
- (A_1, D_2) — Arus Utara Lurus memotong Barat Belok Kanan.
- (C_1, B_2) — Arus Selatan Lurus memotong Timur Belok Kanan.
- (C_1, D_2) — Arus Selatan Lurus memotong Barat Belok Kanan.
- (B_1, A_2) — Arus Timur Lurus memotong Utara Belok Kanan.
- (B_1, C_2) — Arus Timur Lurus memotong Selatan Belok Kanan.
- (D_1, A_2) — Arus Barat Lurus memotong Utara Belok Kanan.
- (D_1, C_2) — Arus Barat Lurus memotong Selatan Belok Kanan.
- (A_2, C_1) — Arus Utara Belok Kanan memotong Selatan Lurus.
- (C_2, A_1) — Arus Selatan Belok Kanan memotong Utara Lurus.
- (B_2, D_1) — Arus Timur Belok Kanan memotong Barat Lurus.
- (D_2, B_1) — Arus Barat Belok Kanan memotong Timur Lurus.
- (A_2, B_2) — Arus Utara Belok Kanan memotong Timur Belok Kanan.
- (A_2, D_2) — Arus Utara Belok Kanan memotong Barat Belok Kanan.
- (C_2, B_2) — Arus Selatan Belok Kanan memotong Timur Belok Kanan.
- (C_2, D_2) — Arus Selatan Belok Kanan memotong Barat Belok Kanan.

Visualisasi graf $G = (V, E)$ yang merepresentasikan struktur persimpangan ini disajikan pada bagian analisis di bawah

IV. PERHITUNGAN DAN ANALISIS EKSPERIMEN

A. Data Volume Arus kendaraan

Data arus kendaraan bukanlah data arus asli, melainkan data *dummy* dan Data kendaraan dalam bentuk standar satuan mobil penumpang per jam (smp/jam)

Arus jenuh (S) untuk lajur lurus 1800 smp/jam, sedangkan lajur belok belok kanan 1500 smp/jam. Berikut adalah ringkasan data:

No	Simpul	Volume (smp/jam)	Arus Jenuh (smp/jam)	Rasio Arus

1	A1	550	1800	0.306
2	A2	180	1500	0.120
3	A3	210	1500	0.140
4	B1	350	1800	0.194
5	B2	120	1500	0.080
6	B3	150	1500	0.100
7	C1	580	1800	0.322
8	C2	200	1500	0.133
9	C3	190	1500	0.127
10	D1	320	1800	0.178
11	D2	100	1500	0.067
12	D3	140	1500	0.093

B. Analisis Derajat Simpul Graf

Langkah pertama penerapan algoritma Welch-Powell adalah menentukan derajat dari masing-masing simpul. Berdasarkan pemodelan sisi konflik yang telah dijelaskan pada Bab III, menghitung tetangga (*adjacency*) dari setiap simpul:

- Simpul A_1 terhubung ke simpul:

$$B_1, D_1, B_2, D_2, C_2 \Rightarrow d(A_1) = 5$$

- Simpul C_1 terhubung ke simpul:

$$B_1, D_1, B_2, D_2, A_2 \Rightarrow d(C_1) = 5$$

- Simpul B_1 terhubung ke simpul:

$$A_1, C_1, A_2, C_2, D_2 \Rightarrow d(B_1) = 5$$

- Simpul D_1 terhubung ke simpul:

$$A_1, C_1, A_2, C_2, B_2 \Rightarrow d(D_1) = 5$$

- Simpul A_2 terhubung ke simpul:

$$B_1, D_1, C_1, B_2, D_2 \Rightarrow d(A_2) = 5$$

- Simpul C_2 terhubung ke simpul:

$$B_1, D_1, C_1, B_2, D_2 \Rightarrow d(C_2) = 5$$

- Simpul B_2 terhubung ke simpul:

$$A_1, C_1, D_1, A_2, C_2 \Rightarrow d(B_2) = 5$$

- Simpul D_2 terhubung ke simpul:

$$A_1, C_1, B_1, A_2, C_2 \Rightarrow d(D_2) = 5$$

- Simpul A_3, B_3, C_3, D_3 tidak memiliki tetangga

$$A_3 = B_3 = C_3 = D_3 = d(D_3) = 0$$

Urutan simpul berdasarkan derajat secara menurun (*descending*) adalah:

$$V_{sorted} = \{A_1, B_1, C_1, D_1, A_2, B_2, C_2, D_2, A_3, B_3, C_3, D_3\}$$

C. Eksekusi Proses Pewarnaan Algoritma Welch-Powell

memulai proses iterasi pewarnaan graf dengan memproses urutan V_{sorted} secara sekuensial:

[1] Iterasi 1 (Warna Kesatu - Merah):

[2] Simpul pertama A_1 adalah . Warnai A_1 dengan warna Merah.

[3] Simpul berikutnya, B_1 , bertetangga dengan A_1 . Lewati.

[4] Simpul berikutnya, C_1 , tidak bertetangga dengan A_1 . Warnai C_1 dengan warna Merah.

[5] Simpul berikutnya, D_1 , bertetangga dengan C_1 . Lewati.

[6] Simpul berikutnya, A_2 , bertetangga dengan C_1 . Lewati.

[7] Simpul berikutnya, B_2 , bertetangga dengan A_1 dan C_1 . Lewati.

[8] Simpul berikutnya, C_2 , bertetangga dengan A_1 . Lewati.

[9] Simpul berikutnya, D_2 , bertetangga dengan A_1 dan C_1 . Lewati.

[10] Simpul A_3, B_3, C_3, D_3 semuanya memiliki derajat 0 (tidak bertetangga dengan simpul mana pun). Warnai semuanya dengan warna Merah.

[11] Fase 1 (Warna Merah): $\{A_1, C_1, A_3, B_3, C_3, D_3\}$

- [12] Iterasi 2 (Warna Kedua - Biru):
- [13] Simpul tidak berwarna pertama di dalam daftar adalah B_1 . Warnai B_1 dengan warna Biru.
- [14] Simpul berikutnya, D_1 , tidak bertetangga dengan B_1 . Warnai D_1 dengan warna Biru.
- [15] Simpul berikutnya, A_2 , bertetangga dengan B_1 dan D_1 . Lewati.
- [16] Simpul berikutnya, B_2 , bertetangga dengan D_1 . Lewati.
- [17] Simpul berikutnya, C_2 , bertetangga dengan B_1 dan D_1 . Lewati.
- [18] Simpul berikutnya, D_2 , bertetangga dengan B_1 . Lewati.
- [19] Fase 2 (Warna Biru): $\{B_1, D_1\}$
- [20] Iterasi 3 (Warna Ketiga - Hijau):
- [21] Simpul tidak berwarna pertama di dalam daftar adalah A_2 . Warnai A_2 dengan warna Hijau.
- [22] Simpul berikutnya, B_2 , bertetangga dengan A_2 . Lewati.
- [23] Simpul berikutnya, C_2 , tidak bertetangga dengan C_2 . Warnai C_2 dengan warna Hijau.
- [24] Simpul berikutnya, D_2 , bertetangga dengan A_2 . Lewati.
- [25] Fase 3 (Warna Hijau): $\{A_2, C_2\}$
- [26] Iterasi 4 (Warna Keempat - Kuning):
- [27] Simpul tidak berwarna pertama di dalam daftar adalah B_2 . Warnai B_2 dengan warna Kuning
- [28] Simpul berikutnya, D_2 , tidak bertetangga dengan B_2 . Warnai D_2 dengan warna Kuning.
- [29] Fase 4 (Warna Kuning): $\{B_2, D_2\}$

Proses pewarnaan selesai karena seluruh simpul telah memperoleh warna. Berdasarkan hasil di atas, kita mendapatkan bilangan kromatik graf konflik sebesar:

$$\chi(G) = 4$$

Ini berarti persimpangan jalan ini membutuhkan tepat 4 fase lampu lalu lintas yang saling bergantian secara dinamis demi menjaga keamanan berkendara. Pengelompokan ini disajikan pada tabel berikut:

Fase	Warna Simpul	Anggota Arus Kendaraan	Representasi Gerakan Lalu Lintas
------	--------------	------------------------	----------------------------------

Fase 1	Merah	A1, C1, A3, B3, C3, D3	Lurus Utara-Selatan, seluruh belok kiri bebas
Fase 2	Biru	B1, D1	Lurus Timur-Barat
Fase 3	Hijau	A2, C2	Belok Kanan Utara dan Selatan
Fase 4	Kuning	B2, D2	Belok Kanan Timur dan Barat

D. Perhitungan Waktu Siklus Lampu Lalu Lintas

Dengan diketahuinya 4 fase lampu hasil optimasi algoritma Welch-Powell, selanjutnya melakukan optimasi waktu siklus menggunakan data volume kendaraan yang disajikan pada Bab IV Bagian A.

1. Penentuan Rasio Arus Kritis per Fase (y_i)

- Fase 1 (Arus lurus Utara B_2 dan Selatan C_1):
- Fase 2 (Arus lurus Timur B_1 dan Barat D_2):
 $y_1 = \max(y_{A1}, y_{C1}) = \max(0.306, 0.322) = 0.322$
- Fase 3 (Belok kanan Utara A_2 dan Selatan C_2):
 $y_2 = \max(y_{B1}, y_{D1}) = \max(0.194, 0.178) = 0.194$
- Fase 4 (Belok kanan Timur B_2 dan Barat D_2):
 $y_3 = \max(y_{A2}, y_{C2}) = \max(0.120, 0.133) = 0.133$

2. Perhitungan Jumlah Rasio Arus Kritis Simpang (Y)

$$y_4 = \max(y_{B2}, y_{D2}) = \max(0.080, 0.067) = 0.080$$

$$Y = \sum_{i=1}^4 y_i = 0.322 + 0.194 + 0.133 + 0.080 = 0.729$$

Karena nilai $Y = 0,739 < 1.0$, persimpangan berada dalam kondisi stabil dan dapat dioptimalkan tanpa risiko kemacetan eksponensial.

3. Perhitungan Total Waktu Hilang per Siklus (L)

Setiap fase memiliki waktu kuning (*yellow*) sebesar 3 detik dan waktu *all-red* pembersihan simpang sebesar 1 detik. Start-up delay kendaraan rata-rata adalah 2 detik per fase. Maka, total waktu hilang per siklus dihitung sebagai berikut:

$$L = 2n + R = 2(4) + (4 \times 1) = 8 + 4 = 12 \text{ detik}$$

4. Perhitungan Waktu Siklus Optimum (C_0)

Menggunakan persamaan Webster, diperoleh:

$$C_0 = \frac{1.5L + 5}{1 - Y} = \frac{1.5(12) + 5}{1 - 0.729} = \frac{18 + 5}{0.271} = \frac{23}{0.271} \approx 84.87 \text{ detik}$$

Waktu siklus optimum dibulatkan menjadi 85 detik guna mempermudah sinkronisasi waktu pada unit pengontrol mikrokontroler di lapangan.

5. Distribusi Waktu Hijau Efektif (g_i)

Total waktu hijau efektif bersih yang tersedia untuk didistribusikan adalah:

$$g_{\text{tot}} = C_0 - L = 85 - 12 = 73 \text{ detik}$$

Alokasi waktu hijau efektif untuk masing-masing fase adalah:

- Fase 1:

$$g_1 = \frac{y_1}{Y} \times g_{\text{tot}} = \frac{0.322}{0.729} \times 73 \approx 32.24 \text{ detik} \rightarrow \text{dibulatkan 32 detik}$$

- Fase 2:

$$g_2 = \frac{y_2}{Y} \times g_{\text{tot}} = \frac{0.194}{0.729} \times 73 \approx 19.42 \text{ detik} \rightarrow \text{dibulatkan 20 detik}$$

- Fase 3:

$$g_3 = \frac{y_3}{Y} \times g_{\text{tot}} = \frac{0.133}{0.729} \times 73 \approx 13.31 \text{ detik} \rightarrow \text{dibulatkan 13 detik}$$

- Fase 4:

$$g_4 = \frac{y_4}{Y} \times g_{\text{tot}} = \frac{0.080}{0.729} \times 73 \approx 8.01 \text{ detik} \rightarrow \text{dibulatkan 8 detik}$$

Total alokasi hijau efektif aktual:

$$32 + 20 + 13 + 8 = 73 \text{ detik.}$$

6. Hasil Akhir Desain Lampu Lalu Lintas Aktual

Waktu nyala lampu lalu lintas aktual yang ditampilkan kepada pengguna jalan dihitung berdasarkan waktu hijau efektif, kuning (3 detik), dan merah total di luar fase berjalan. Hasil kalkulasi menyeluruh sebagai berikut:

- Lengan Utara-Selatan (Fase 1)
 - Hijau : 32 detik
 - Kuning : 3 detik
 - Semua merah : 1 detik
 - Merah : 49 detik
 - Total 85 detik

- Lengan Timur-Barat (Fase 2)
 - Hijau : 20 detik
 - Kuning : 3 detik
 - Semua merah : 1 detik
 - Merah : 61 detik
 - Total 85 detik
- Lengan Utara-Selatan (Kanan) (Fase 3)
 - Hijau : 13 detik
 - Kuning : 3 detik
 - Semua merah : 1 detik
 - Merah : 68 detik
 - Total 85 detik
- Lengan Timur-Barat (Kanan) (Fase 4)
 - Hijau : 8 detik
 - Kuning : 3 detik
 - Semua merah : 1 detik
 - Merah : 73 detik
 - Total 85 detik

V. KESIMPULAN DAN SARAN

Makalah ini membuktikan teoretis bahwa integrasi antara Teori Graf dan teknik rekayasa transportasi sipil mampu menghadirkan sistem kendali lampu lalu lintas yang sangat optimal dan aman. Melalui implementasi Algoritma Pewarnaan Graf Welch-Powell, persimpangan jalan padat perkotaan yang memiliki 12 arus pergerakan kompleks berhasil disederhanakan menjadi model representasi 4 fase lampu lalu lintas yang bebas konflik $\chi(G) = 4$.

Saran untuk pengembangan selanjutnya adalah mengintegrasikan sistem sensor kamera berbasis *computer vision* untuk menghitung volume kendaraan secara otomatis dan dinamis sehingga data yang diberikan adalah data aktual dan didapatkan dengan mudah

DAFTAR PUSTAKA

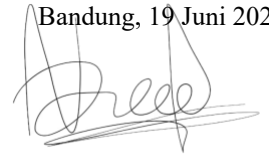
- [1]R. Munir, *Matematika Diskrit*, edisi ke-6. Bandung: Penerbit Informatika, 2016.
- [2]D. A. Sulistiani, C. Ais, dan A. Fanani, "Application of Graph Theory and Welch-Powell Method at Traffic Light Regulation," *JUTI: Jurnal Ilmiah Teknologi Informasi*, vol. 20, no. 2, pp. 54-60, Jul. 2022. Tersedia: <https://juti.if.its.ac.id/index.php/juti/article/view/1096>.
- [3]M. Soimah dan N. S. M. Mussafi, "Pewarnaan Simpul Dengan Algoritma Welch-Powell Pada Traffic Light di Yogyakarta," *Jurnal Ilmiah Matematika dan Pendidikan Matematika*, vol. 2, no. 2, pp. 112-120, Des. 2013.
- [4]I. Poernamasari, R. Tumilaar, dan C. E. J. C. Montolalu, "Optimasi Pengaturan Lampu Lalu Lintas dengan menggunakan Metode Webster (Studi Kasus Persimpangan Jalan Babe Palar)," *d'Cartesian: Jurnal Matematika dan Aplikasi*, vol. 8, no. 1, pp. 27-35, Mar. 2019. Tersedia: <https://ejournal.unsrat.ac.id/index.php/decartesian>
- [5]F. V. Webster, *Traffic Signal Settings*, Road Research Technical Paper No. 39. London: Her Majesty's Stationery Office, 1958.
- [6]M. R. Alfian, A. E. S. H. Maharani, S. T. Lestari, S. H. P. Ningrum, M. U. Romdhini, dan M. R. Alfian, "Max Plus Algebra for Urban Traffic Optimization in Mataram City, Indonesia," *Journal of Mechanics, Control and Mathematical Sciences*, vol. 15, no. 9, pp. 1-12, Sep. 2020. Tersedia: <https://doi.org/10.26782/jmcms.2025.10.00001>.

[7]A. Dewi, "Implementasi Pewarnaan Algoritma Welch-Powell Dengan Metode Webster Untuk Optimalisasi Durasi Lampu Lalu Lintas (Studi Kasus: Simpang Empat Bersinyal Kelurahan Semabung Baru Kota Pangkalpinang)," Skripsi, Universitas Bangka Belitung, 2025. Tersedia: <https://repository.ubb.ac.id/id/eprint/11324/>.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Juni 2025



Naufal Hasbialhaq dan 13525120