

PROTOTYPE SISTEM PENGATUR ISYARAT LALU-LINTAS TERKOORDINASI UNTUK SIMPANG EMPAT GONDONANAN DAN BINTARAN

Freddy Kurniawan¹, Anton Setiawan Honggowibowo²

¹Prodi Teknik Elektro, STT Adisutjipto

email: freddykurniawan@stta.ac.id

²Prodi Teknik Informatika, STT Adisutjipto

email: anton_s_h@yahoo.com

Abstract

Traffic jams in the city of Yogyakarta has resulted in losses of millions rupiahs every hour. One of the factors causing of traffic congestion especially at adjacent intersection such as at Gondomanan and Bintaran intersections is no coordination between the traffic controllers. One method to solve this problem, both traffic controllers must be coordinately controlled to make vehicles unacross all traffic controllers with minimum stop and delay. The intent of coordinating is to provide uninterrupted flow of traffic along streets or highways.

This paper presents a prototype of a coordinated traffic control systems. The system consists of two traffic controllers, one acting as master controller and the other acting as slave controller. Each controller controls the traffic signal based on its schedule, while in addition slave controller also controls the traffic signal based on timing from master. Slave controller could add or subtract green interval of any phase to synchronous its timing to the master timing.

The coordinated traffic control program takes places in ATmega128A AVR microcontroller systems. The program only uses less than 10% of microprocessor resources, so it can be widely developed without changing the processor.

Key words: coordinated traffic control systems, master controller, slave controller.

Intisari

Salah satu kelemahan alat pengatur isyarat lalu-lintas (APILL) di persimpangan Gondomanan dan Bintaran adalah tidak adanya koordinasi di antara keduanya. Hal ini telah mengakibatkan terjadinya antrian panjang kendaraan dan kemacetan lalu-lintas. Salah satu solusinya adalah digunakannya sistem pengatur isyarat lalu-lintas terkoordinasi. Tujuan metode ini adalah agar sebagian besar kendaraan di jalur utama dapat melewati kedua persimpangan tersebut tanpa mendapat isyarat merah.

Prototipe sistem pengatur isyarat lalu-lintas terkoordinasi ini terdiri dari dua APILL, yaitu APILL master dan slave. APILL master mengatur lalu-lintas sesuai jadwal, sedangkan APILL slave mengatur lalu-lintas sesuai jadwal dan mengikuti pewaktuan dari master. APILL slave dapat menambah atau mengurangi waktu hijau beberapa fase agar pewaktuan kedua APILL menjadi sinkron.

Program pengatur isyarat lalu-lintas terkoordinasi ini dijalankan di mikrokontroler AVR ATmega128A. Program hanya menggunakan tidak lebih dari 10 % sumber daya prosesor, sehingga dapat dikembangkan dengan leluasa tanpa mengganti prosesor yang digunakan.

Kata kunci: sistem pengatur isyarat lalu-lintas terkoordinasi, APILL master, APILL slave

PENDAHULUAN

Kemacetan di Kota Yogyakarta telah mengakibatkan kerugian jutaan rupiah setiap jamnya (Basuki dan Siswandi, 2008). Salah satu penyebab kemacetan adalah tidak optimalnya pewaktuan beberapa alat pengatur isyarat lalu lintas (APILL). Hal ini terjadi terutama untuk persimpangan yang berdekatan (Primantari, 2010), misalnya di simpang empat Gondomanan dan Bintaran.

Tidak terkoordinasinya pewaktuan isyarat lalu-lintas di kedua simpang empat tersebut

menyebabkan tidak pastinya waktu tempuh kendaraan pada ruas jalan tersebut. Sebagian besar kendaraan yang datang dari arah barat simpang empat Bintaran berasal dari simpang empat Gondomanan. Kadang kendaraan-kendaraan tersebut mendapatkan lampu hijau di simpang empat Bintaran, namun di waktu lain mendapatkan lampu merah. Jika sebagian besar kendaraan tersebut mendapatkan lampu merah setibanya di simpang empat Bintaran, maka terjadi antrian

panjang kendaraan. Dari sinilah kemacetan di ruas jalan tersebut terjadi.

Salah satu solusi atas hal tersebut adalah digunakannya APILL terkoordinasi. Suatu APILL dapat disinkronkan dengan kepadatan lalu-lintas setempat atau dikoordinasi oleh APILL lain (Kurniawan dan Adiprasetya, 2007). Selain dapat menurunkan kemungkinan terjadinya kemacetan, beberapa APILL terkoordinasi dapat menjadikan kendaraan dapat berjalan dengan halus (*smooth*) pada suatu ruas jalan. Bagi pengemudi, hal ini dapat mengurangi banyaknya perhentian, tundaan, dan waktu tempuh kendaraan (Federal Highway Administration, 2008).

Tujuan Penelitian

Penelitian ini bertujuan untuk membuat prototipe sistem pengatur isyarat lalu-lintas terkoordinasi yang dapat diimplementasikan di simpang empat Gondomanan dan Bintaran. Prototipe diharapkan dapat diimplementasikan di kedua simpang empat sehingga sebagian besar kendaraan:

- dari simpang empat Gondomanan akan mendapat isyarat hijau saat tiba di simpang empat Bintaran, dan
- dari simpang empat Bintaran akan mendapat isyarat hijau saat tiba di simpang empat Gondomanan.

Urgensi Penelitian

Keberadaan sistem ini untuk memperlancar lalu-lintas kendaraan dapat lebih optimal didukung oleh ATCS (*Area Traffic Control Systems*) yang telah diterapkan di simpang empat Gondomanan (Hanggara, 2012).

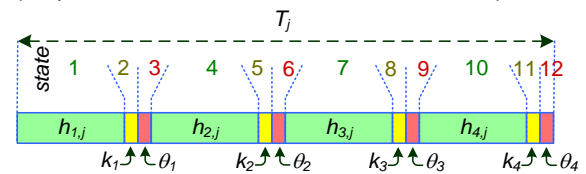
Lebih jauh lagi, sistem ini dapat dikembangkan menjadi sistem pengatur isyarat lalu-lintas terkoordinasi adaptif (Kurniawan dan Adiprasetya, 2007) dengan menambah algoritma penghitung kepadatan kendaraan berdasar gambar yang ditangkap kamera CCTV (Rachmadi dkk, 2012). Data kepadatan kendaraan juga dapat menjadi salah satu masukan bagi sistem ITS (*Intelligent Transportation System*) yang saat ini sedang dikembangkan di beberapa kota besar di Indonesia (Wrestti, 2012).

Tinjauan Pustaka

Pengaturan isyarat lalu-lintas didasarkan pada urutan *state* (keadaan). Dalam satu siklus, terdapat dua belas *state* pada APILL yang mengatur sebuah simpang empat.

Saat mulai aktif, sebuah APILL akan menjalankan pengaturan pada siklus pertama. Nomor siklus dinyatakan dengan *j*. Urutan *state* pada siklus ke-*j* mengikuti diagram pada

Gambar 1. Warna isyarat fase untuk setiap *state* dalam satu siklus mengikuti Tabel 1 (Dephub, 1996; Dresner dan Stone 2007).



Gambar 1 Urutan *state* dalam sebuah siklus
Keterangan :

- i* = nomor fase
- $h_{i,j}$ = durasi hijau fase *i* siklus ke-*j*
- k_i = durasi kuning fase *i*
- θ_i = waktu pengosongan fase *i*

Tabel 1. Warna isyarat pada setiap fase

| State | Warna isyarat | | | |
|-------|---------------|--------|--------|--------|
| | Fase 1 | Fase 2 | Fase 3 | Fase 4 |
| 1 | hijau | merah | merah | merah |
| 2 | kuning | merah | merah | merah |
| 3 | merah | merah | merah | merah |
| 4 | merah | hijau | merah | merah |
| 5 | merah | kuning | merah | merah |
| 6 | merah | merah | merah | merah |
| 7 | merah | merah | hijau | merah |
| 8 | merah | merah | kuning | merah |
| 9 | merah | merah | merah | merah |
| 10 | merah | merah | merah | hijau |
| 11 | merah | merah | merah | kuning |
| 12 | merah | merah | merah | merah |

Pada sebuah APILL terjadwal, durasi isyarat hijau setiap fase dapat berubah sesuai jadwal. Namun biasanya durasi isyarat kuning dan waktu pengosongan bernilai tetap (Dephub, 1996). Periode siklus (T_j) pengaturan isyarat lalu-lintas mengikuti Persamaan (1).

$$T_j = \sum_{i=1}^4 (h_{i,j} + k_i + \theta_i) \quad (1)$$

Pada setiap *state*, prosesor menghitung mundur waktu isyarat setiap fase. Hitungan tersebut dilaksanakan satu kali per detik.

Proses koordinasi dilakukan dengan menggeser offset pewaktuan suatu APILL terhadap APILL lain. Terdapat tiga metode utama untuk penggeseran offset agar tercapai kondisi terkoordinasi, yaitu: *dwelling*, penambahan dan/atau pengurangan durasi hijau terhadap arah terkoordinasi, dan penambahan dan/atau pengurangan durasi hijau terhadap semua arah (Federal Highway Administration, 2008).

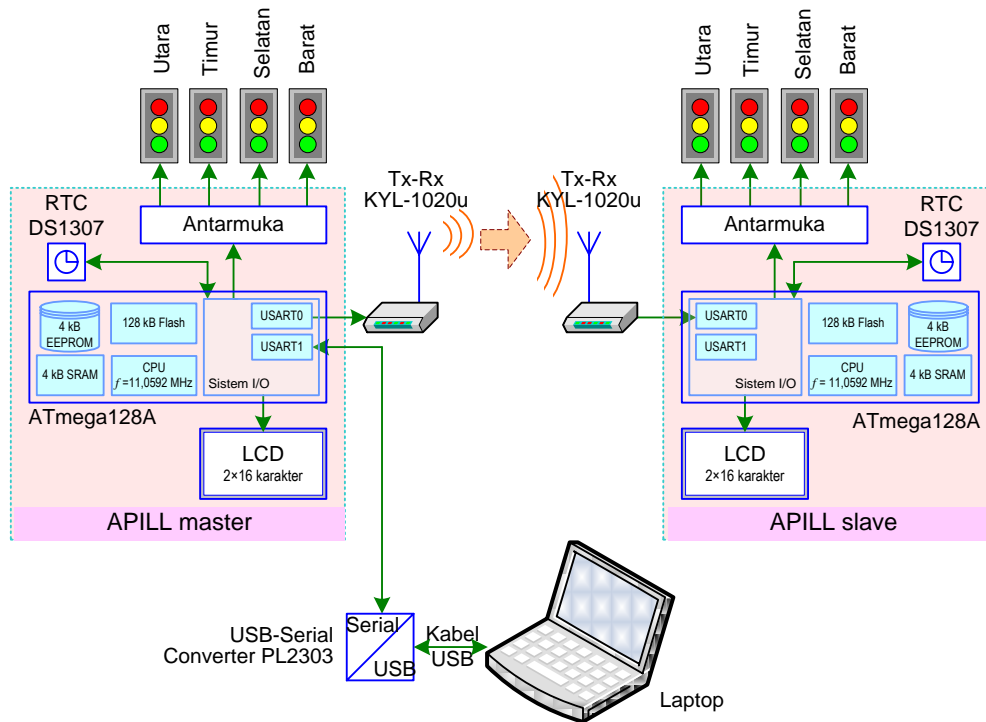
Prototipe Sistem

Prototipe sistem terdiri dari sebuah APILL master yang mewakili APILL di simpang empat Gondomanan dan sebuah

APILL slave yang mewakili APILL di simpang empat Bintaran. Diagram blok sistem dapat dilihat pada Gambar 2.

Pemilihan APILL di Gondomanan sebagai master dengan pertimbangan sebagai berikut.

- Kepadatan di simpang empat Gondomanan lebih tinggi dari pada di Bintaran.
- APILL di Gondomanan telah dilengkapi dengan ATCS.



Gambar 2 Diagram blok prototipe sistem pengatur isyarat lalu-lintas terkoordinasi

Pengaturan isyarat lalu-lintas pada konfigurasi ini menggunakan pedoman sebagai berikut.

- Master bertindak sebagai APILL berdiri sendiri, sehingga hanya mengatur isyarat lalu-lintas sesuai jadwal.
- Slave bertindak sebagai APILL terkoordinasi, sehingga mengatur isyarat lalu-lintas sesuai jadwal dan terkoordinasi dengan APILL master.

Pengaturan Isyarat Lalu-lintas

Dalam melakukan pengaturan isyarat lalu-lintas, mikrokontroler membaca data waktu dari RTC (*Real Time Clock*) DS1307 secara periodis. Berdasar data waktu ini, sistem menentukan durasi setiap isyarat sesuai jadwal yang tersimpan di EEPROM mikrokontroler.

APILL master dan slave mengatur nyala isyarat lampu dari empat arah kedatangan. Hitungan waktu setiap nyala lampu ditampilkan pada sebuah penampil LCD.

Pada setiap akhir *state* ke-12, master mengirimkan data koordinasi ke slave menggunakan protokol komunikasi USART (*Universal Synchronous and Asynchronous*

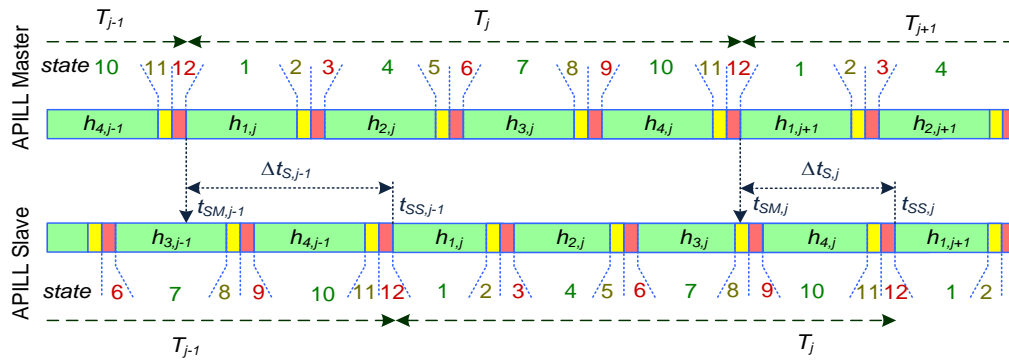
serial Receiver and Transmitter). Data keluaran USART diumpungkan ke modul Tx-Rx KYL-1020u untuk dikirim secara nirkabel menggunakan modulasi FSK (*Frequency Shift Keying*). Frekuensi keluaran modul ini adalah 433 MHz dengan daya keluaran 500 mW.

Mekanisme Koordinasi

Pada penelitian ini dipilih metode koordinasi dengan penambahan dan pengurangan durasi hijau untuk semua arah pada APILL slave. Nilai total perubahan durasi hijau fase 1 hingga 4 ditetapkan harus kurang dari 25 % dari nilai periode siklus. Jika dari hasil perhitungan ternyata perubahan durasi hijau total harus sama dengan atau melebihi 25 % dari nilai periode siklus, maka koordinasi akan dilaksanakan dalam waktu lebih dari satu siklus.

APILL master dan slave mengatur isyarat lalu-lintas sesuai jadwal. Pembacaan jadwal di basis data EEPROM mikrokontroler dilakukan pada setiap akhir *state* ke-11. Pada akhir *state* ke-12, APILL master mengirim data terkoordinasi ke APILL slave. Data tersebut diterima di slave melalui USART0. Waktu slave menerima data koordinasi dari

master pada siklus ke- j diberi notasi $t_{SM,j}$. Hal ini dapat dilihat pada Gambar 3.



Gambar 3. Mekanisme sinkronisasi antara master dan slave

Proses koordinasi menghasilkan nilai perubahan durasi hijau bagi siklus berikutnya. Nilai tersebut disimpan dalam variabel Δh_{ij+1} . Nilai Δh_{ij+1} direset pada akhir setiap siklus. Algoritma koordinasi dikerjakan oleh APILL slave pada akhir *state* ke-12. Algoritma koordinasi untuk siklus ke- j adalah sebagai berikut.

1. Baca data waktu dari RTC dan catat sebagai waktu slave mengerjakan koordinasi: $t_{SS,j}$.
2. Hitung delta koordinasi, $\Delta t_{S,j} = t_{SM,j} - t_{SS,j}$
3. Hitung galat koordinasi, $\xi_j = \omega_{MS} - \Delta t_{S,j}$ dengan ω_{MS} adalah waktu tempuh kendaraan dari APILL master ke slave
4. Cek validitas galat koordinasi, jika $\xi_j > T_j$ maka koordinasi dibatalkan dan kembali ke program utama.
5. Jika $\xi_j > 0$, maka kerjakan langkah ke-10.
6. Jika $\xi_j < -\frac{T_j}{4}$, maka $\xi_j = -\frac{T_j}{4}$
7. Kompensasikan ξ_j ke nilai durasi hijau pada siklus berikutnya mengikuti Persamaan 2.

$$\Delta h_{ij+1} = \frac{h_{i,j}}{\sum_{i=1}^4 h_{i,j}} \times \xi_j \quad (2)$$

dengan :

Δh_{ij+1} = nilai perubahan durasi hijau fase i siklus ke- $j+1$

h_{ij} = durasi hijau fase i siklus ke- j

8. Nalkan nilai ξ_j

Semua variabel yang digunakan pada sistem ini bertipe integer 8 bit atau 16 bit. Pengkompensasian nilai ξ_j menggunakan Persamaan (2) kemungkinan tidak akan menjadikan nilai ξ_j menjadi nol. Jika nilai ξ_j belum nol, maka nilai tersebut ke semua nilai Δh_{ij+1} agar bisa "menghabiskan" nilai ξ_j . Nilai tersebut didistribusikan ke setiap arah dimulai dari arah yang memiliki h_{ij+1} tertinggi dengan algoritma sebagai berikut.

- a. Tentukan arah dengan durasi hijau terlama, $i_{h-maks} = \text{maks}(h_{ij+1})$.
- b. Tambah satu nilai ξ_j dan kurangi satu nilai h_{ij+1} dengan $i=i_{h-maks}$
- c. Jika $\xi_j < 0$, maka hapus i_{h-maks} , dan kembali ke langkah 8.a.
9. Kembali ke program utama.
10. Jika $\xi_j = 0$, maka kembali ke program utama.
11. Jika $\xi_j > \frac{T_j}{4}$, maka $\xi_j = \frac{T_j}{4}$
12. Kompensasikan ξ_j ke semua nilai durasi hijau pada siklus berikutnya mengikuti Persamaan (2).
13. Nalkan nilai ξ_j sebagaimana langkah ke-8 dengan algoritma sebagai berikut.
 - a. Tentukan arah dengan durasi hijau terlama, $i_{h-maks} = \text{maks}(h_{ij+1})$.
 - b. Kurangi satu nilai ξ_j dan tambah satu nilai h_{ij+1} dengan $i=i_{h-maks}$
 - c. Jika $\xi_j > 0$, maka hapus i_{h-maks} , dan kembali ke langkah 13.a.
14. Kembali ke program utama.

Urutan Arah Fase

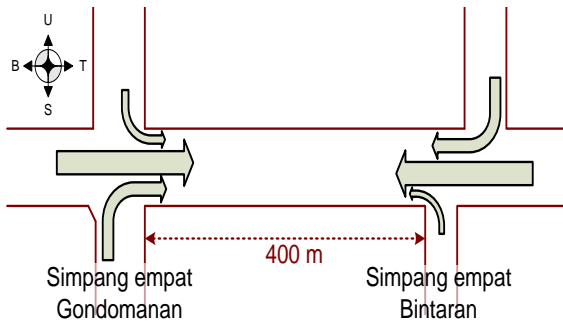
Urutan arah fase APILL master mengikuti arah jarum jam. Fase pertama untuk arah utara. Hal ini untuk menjaga keamanan kendaraan dan memperkecil konflik pergerakan kendaraan di persimpangan tersebut. Sementara itu, agar koordinasi dapat terwujud, urutan fase APILL slave tidak lagi mengikuti arah jarum jam. Urutan fase kedua APILL mengikuti Tabel 2.

Tabel 2 Arah fase APILL master dan slave

| Fase | Arah | |
|------|--------------|-------------|
| | APILL Master | APILL Slave |
| 1 | utara | selatan |
| 2 | timur | utara |
| 3 | selatan | timur |
| 4 | barat | barat |

Target Trayektori Kendaraan

Sebagian besar kendaraan yang keluar ke arah timur dari simpang empat Gondomanan masuk dari sisi barat persimpangan tersebut. Dan sebagian besar kendaraan yang keluar ke arah barat dari simpang empat Bintaran masuk dari sisi timur persimpangan tersebut. Hal ini di diperlihatkan pada Gambar 4.



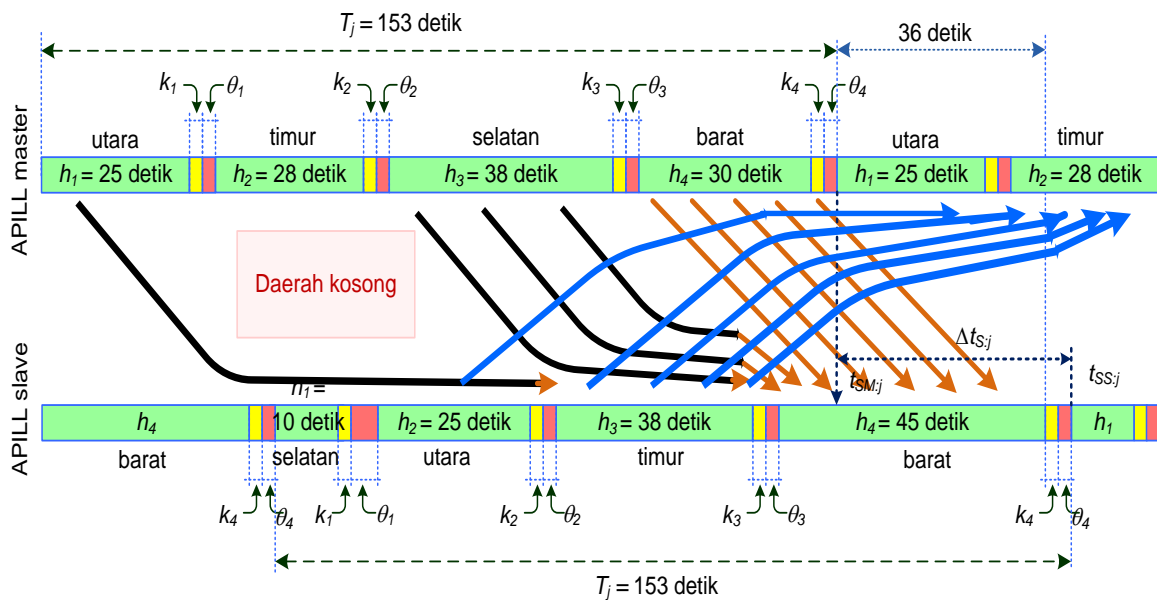
Gambar 4 Arus lalu-lintas di simpang empat Gondomanan dan Bintaran

Jarak antara simpang empat Gondomanan dan Bintaran adalah 400 meter. Dengan kecepatan rata-rata 40 km/jam, kendaraan dapat menempuh jarak tersebut dalam waktu tempuh $\omega_{MS} = 36$ detik. Waktu

tersebut cukup realistis ditempuh pada kondisi jam-jam puncak. Pada jam-jam agak longgar, kendaraan dapat menggunakan kecepatan hingga 60 km/jam, sehingga waktu tempuh hanya 26 detik.

Agar sebagian besar kendaraan dari APILL master langsung mendapat isyarat hijau setibanya di APILL slave, durasi hijau arah barat atau h_4 APILL slave harus diatur agar mempunyai offset ω_{MS} terhadap durasi hijau arah barat atau h_4 APILL master. Dengan kata lain, h_4 APILL slave harus tertinggal ω_{MS} terhadap h_4 APILL master.

Sementara itu, agar sebagian besar kendaraan dari APILL slave langsung mendapat isyarat hijau setibanya di APILL master, durasi hijau untuk arah timur atau h_2 APILL slave harus diatur agar mempunyai offset $-\omega_{MS}$ terhadap durasi hijau untuk arah timur atau h_2 APILL master. Dengan kata lain, h_2 APILL slave harus mendahului ω_{MS} terhadap h_2 APILL master. Atas dasar pertimbangan tersebut pewaktuan kedua APILL dibuat mengikuti diagram ruang waktu pada Gambar 5.



Gambar 5 Perkiraan trayektori optimal pada jam-jam puncak

Pada diagram tersebut, durasi hijau sisi barat persimpangan Bintaran untuk siklus ke- j mempunyai offset sebesar $\Delta t_{S,j}$. Bagi APILL Bintaran, parameter $\Delta t_{S,j}$ bernilai selisih waktu antara diterimanya data koordinasi dari master ($\Delta t_{SM,j}$) dengan waktu akan dieksekusinya algoritma koordinasi di slave ($\Delta t_{SS,j}$). Nilai tersebut harus sama dengan

waktu tempuh kendaraan dari master ke slave. Dengan kata lain $\Delta t_{S,j} = \omega_{MS}$. Pada jam-jam sibuk $\Delta t_{S,j}$ dapat diberi nilai 36 detik.

Sesuai dengan diagram trayektori tersebut, dengan digunakannya sistem pengatur isyarat lalu-lintas terkoordinasi ini, diharapkan kendaraan dari sisi barat APILL master langsung mendapat isyarat hijau

setibanya di sisi barat APILL slave. Sementara itu, kendaraan dari sisi timur APILL slave akan menanti paling lama sekitar 40 detik untuk mendapatkan isyarat hijau setibanya di sisi timur APILL master.

Optimasi Program

Sistem ini diharapkan dapat dikembangkan lebih lanjut tanpa banyak penggantian perangkat keras. Oleh karena itu program pengatur isyarat lalu-lintas termasuk algoritma koordinasi diusahakan:

- menggunakan sumber daya prosesor seminim mungkin, dan
- berjalan dengan kecepatan tinggi.

Untuk mencapai tujuan tersebut, dua langkah optimasi program telah dilakukan, yaitu:

- seluruh variabel menggunakan tipe integer 8 dan 16 bit, dan
- seluruh operasi matematika menggunakan tipe operasi integer.

HASIL DAN PEMBAHASAN

Data Hasil Survei

Berdasar hasil survei, dalam satu hari diperlukan sepuluh slot waktu pengaturan. Jadwal pengaturan APILL master dan slave pada hari kerja dapat mengikuti

Tabel 3. Tabel tersebut berisi data durasi isyarat hijau kedua APILL beserta waktu digunakannya. Data pada tabel tersebut disimpan di EEPROM masing-masing mikrokontroler.

Durasi kuning semua arah untuk kedua APILL dibuat sama, yaitu 3 detik. Atas pertimbangan letak geometris, nilai waktu pengosongan, yaitu semua isyarat berwarna merah, untuk semua fase adalah 5 detik, kecuali fase 4 APILL slave yaitu 8 detik.

Tabel 3 Jadwal pengaturan APILL master dan slave pada hari kerja

| slot waktu | Jam | APILL master | | | | APILL slave | | | | T |
|------------|-------|--------------|-------|-------|-------|-------------|-------|-------|-------|-----|
| | | h_1 | h_2 | h_3 | h_4 | h_1 | h_2 | h_3 | h_4 | |
| 1 | 04:30 | 8 | 8 | 10 | 8 | 0 | 0 | 0 | 0 | 0 |
| | 05:00 | 8 | 8 | 10 | 8 | 6 | 8 | 8 | 9 | 66 |
| 2 | 06:00 | 8 | 15 | 20 | 15 | 8 | 10 | 18 | 19 | 90 |
| 3 | 06:30 | 15 | 25 | 40 | 30 | 12 | 15 | 35 | 45 | 142 |
| 4 | 07:10 | 17 | 25 | 40 | 30 | 12 | 17 | 35 | 45 | 144 |
| 5 | 08:30 | 20 | 28 | 40 | 30 | 10 | 25 | 35 | 45 | 150 |
| 6 | 10:00 | 25 | 28 | 38 | 30 | 10 | 25 | 38 | 45 | 153 |
| 7 | 15:30 | 30 | 28 | 30 | 30 | 10 | 25 | 35 | 45 | 150 |
| 8 | 18:00 | 25 | 25 | 30 | 25 | 10 | 21 | 31 | 40 | 137 |
| 9 | 21:30 | 15 | 15 | 15 | 15 | 8 | 15 | 16 | 18 | 92 |
| 10 | 23:00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Setiap APILL mengatur lalu-lintas berdasar daftar slot waktu yang disimpan di basis data. Adanya pergantian slot pada sebuah APILL akan mengakibatkan terjadinya perubahan durasi hijau.

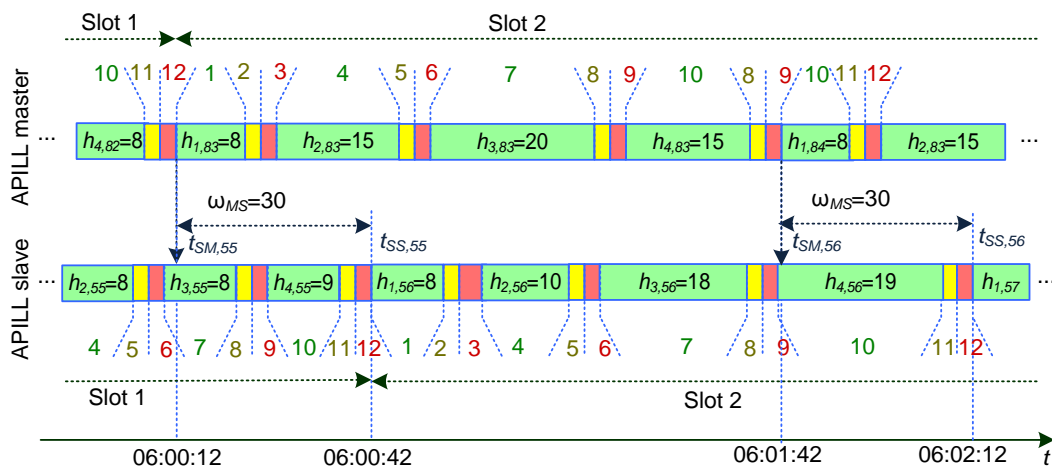
Agar tetap terkoordinasi, slot dengan nomor sama harus mempunyai waktu mulai dan waktu akhir sama, dan periode siklus yang sama. Kecuali untuk slot pertama.

Slot 1 APILL master dimulai pada pukul 04:30. Arah 1, 2, 3, dan 4 mendapat durasi hijau 8 detik, 8 detik, 10 detik, dan 8 detik. Sementara itu, slot 1 APILL slave dimulai pada pukul 05:00. Arah 1, 2, 3, dan 4 mendapat durasi hijau 6 detik, 8 detik, 8 detik, dan 9 detik. Sesuai Persamaan 1, periode siklus $T_j = 66$ detik.

Koordinasi pada Saat Pergantian Slot

Pada

Tabel 3 terlihat bahwa pada slot 2, APILL master dan slave mempunyai periode siklus 90 detik. Diagram pewaktuan durasi hijau APILL master dan slave pada saat pergantian dari slot 1 ke slot 2 dapat dilihat pada Gambar 6.



Gambar 6 Proses koordinasi pergantian dari slot 1 ke slot 2

Siklus terakhir pada slot 1 APILL master, yaitu siklus ke-82 berakhir pukul 06:00:12. APILL master segera menggunakan slot 2. Sesuai data pada

Tabel 3, siklus ke-83 menggunakan nilai periode siklus 90 detik. Siklus ini berakhir pukul 06:01:42. Di akhir siklus ini APILL master mengirimkan data koordinasi ke APILL slave. APILL slave menerima sebagai $t_{SM,56}$.

Slot 2 APILL slave dimulai pukul 06:00:42. Sebagaimana terlihat pada Gambar 6, siklus ke-56 di APILL slave ini berakhir pukul 06:02:12. Saat itu telah terjadi perbedaan 30 detik dari waktu diterimanya data koordinasi dari APILL master, sehingga ξ_j tetap bernilai nol. Dengan demikian perubahan slot waktu di APILL master dan slave tidak akan menyebabkan timbulnya galat koordinasi.

Koordinasi dalam Waktu Satu Siklus

Salah satu kasus yang mengharuskan algoritma koordinasi bekerja adalah pada saat APILL master dinonaktifkan, kemudian diaktifkan kembali. Ini dapat terjadi misalnya karena aliran listrik di APILL master terputus atau APILL master sedang dalam proses perbaikan.

Kejadian ini disimulasikan terjadi setelah pada pukul 10 pagi. Sesuai Tabel 2, APILL slave menjalankan pengaturan lalu-lintas

menggunakan slot waktu ke-6. APILL slave sedang menjalankan siklus ke-187. Siklus ini dimulai pukul 10:12:44 dan berakhir pukul 10:15:17. Periode siklus $T_j = 153$ detik.

APILL master diaktifkan pukul 10:12:04. Siklus pertama APILL master menggunakan slot waktu ke-6. Durasi hijau fase 1, 2, 3, dan 4 adalah 25 detik, 28 detik, 38 detik, dan 30 detik. Periode siklus APILL master juga sama dengan APILL slave yaitu 153 detik. Sementara itu, APILL slave sedang menjalankan siklus ke-187. Tabel 4 berikut merupakan data hasil rekaman perubahan durasi hijau selama proses koordinasi ini.

Dari hasil monitoring, didapat bahwa pada saat itu terdapat galat koordinasi $\xi_j = -10$ detik. Ini berarti pewaktuan di APILL slave terlambat 10 detik. Dari langkah ke-7 dan 8 algoritma koordinasi, dihasilkan perubahan durasi hijau siklus ke-188 untuk fase 1, 2, 3, dan 4 adalah 0, -2, -4, dan -4 detik. Hal ini menyebabkan durasi hijau siklus ke-188 untuk fase 1, 2, 3, dan 4 adalah 10, 23, 34, dan 41 detik. Perubahan ini hanya berlaku pada siklus ke-188. Pada akhir siklus ini, galat koordinasi telah menjadi nol. Dengan kata lain, APILL master dan slave telah terkoordinasi kembali.

Tabel 4 Durasi hijau saat terjadi koordinasi dalam waktu satu siklus

| APILL master | | | | | | APILL slave | | | | | | | | | | |
|--------------|-------|-------|-------|-------|-------|-------------|-------|-------|-------|-------|-------|---------|------------------|------------------|------------------|------------------|
| j | h_1 | h_2 | h_3 | h_4 | T_j | j | h_1 | h_2 | h_3 | h_4 | T_j | ξ_j | $\Delta h_{1;j}$ | $\Delta h_{2;j}$ | $\Delta h_{3;j}$ | $\Delta h_{4;j}$ |
| - | - | - | - | - | - | 187 | 10 | 25 | 38 | 45 | 153 | 0 | 0 | 0 | 0 | 0 |
| 1 | 25 | 28 | 38 | 30 | 153 | 188 | 10 | 23 | 34 | 41 | 143 | -10 | 0 | -2 | -4 | -4 |
| 2 | 25 | 28 | 38 | 30 | 153 | 189 | 10 | 25 | 38 | 45 | 153 | 0 | 0 | 0 | 0 | 0 |

Keterangan : j = nomor siklus
 T_j = periode siklus ke- j
 h_i = durasi hijau fase i

ξ_j = delta koordinasi siklus ke- j
 Δh_{ij} = delta durasi hijau fase i siklus ke- j

Koordinasi dalam Dua Siklus

Koordinasi dapat membutuhkan waktu lebih dari satu siklus. Hal ini dapat terjadi jika perubahan durasi hijau yang diperlukan melebihi dari seperempat periode siklus.

Kejadian ini disimulasikan dengan mengaktifkan APILL master pada pukul 10:11:34. Perubahan durasi hijau APILL master dan slave dapat dilihat pada Tabel 5.

Tabel 5 Durasi hijau saat terjadi koordinasi dalam waktu dua siklus

| APILL master | | | | | | APILL slave | | | | | | | | | | |
|--------------|-------|-------|-------|-------|-------|-------------|-------|-------|-------|-------|-------|---------|------------------|------------------|------------------|------------------|
| j | h_1 | h_2 | h_3 | h_4 | T_j | j | h_1 | h_2 | h_3 | h_4 | T_j | ξ_j | $\Delta h_{1;j}$ | $\Delta h_{2;j}$ | $\Delta h_{3;j}$ | $\Delta h_{4;j}$ |
| - | - | - | - | - | - | 187 | 10 | 25 | 38 | 45 | 153 | 0 | 0 | 0 | 0 | 0 |
| 1 | 25 | 28 | 38 | 30 | 153 | 188 | 8 | 17 | 25 | 30 | 143 | -40 | -2 | -6 | -9 | -11 |
| 2 | 25 | 28 | 38 | 30 | 153 | 189 | 10 | 25 | 38 | 45 | 153 | -2 | 0 | 0 | -1 | -1 |

Dari Tabel 5 terlihat bahwa pada akhir siklus ke-187 terdapat galat koordinasi ξ_j sebesar -40 detik. Ini berarti pewaktuan di

APILL slave terlambat 40 detik. Dikarenakan nilai tersebut kurang dari $-\frac{T_j}{4}$, maka pada

langkah ke-6 algoritma koordinasi, nilai ξ_j dijadikan $-\frac{T_j}{4}$ atau sebesar -38 detik. Pada langkah ke-7 dan 8, dihasilkan perubahan durasi hijau siklus ke-188 untuk fase 1, 2, 3, dan 4 adalah -2, -6, -9, dan -11 detik. Hal ini menyebabkan durasi hijau siklus ke-188 adalah 8, 17, 25, dan 30 detik.

Pada akhir siklus ke-188, ternyata terdapat galat koordinasi ξ_j sebesar -2 detik. Nilai galat tersebut dikompensasikan ke durasi hijau sehingga durasi hijau siklus ke-189 untuk fase 1, 2, 3, dan 4 adalah 10, 25, 37, dan 44 detik.

Penggunaan Sumber Daya Prosesor

Penggunaan sumber daya prosesor dapat dilihat pada Tabel 6. Hasil kompilasi program menunjukkan bahwa kode bahasa mesin program APILL master dan slave baru menggunakan ruang memory flash tidak lebih dari 7 %. Semua variabel juga baru menggunakan tidak lebih dari 2 % kapasitas SRAM. Sementara itu basis data yang memuat jadwal pengaturan waktu APILL baru menempati ruang tidak lebih dari 7 % kapasitas EEPROM.

Tabel 6 Penggunaan memory APILL

| Jenis memory | Kapasitas memory (Byte) | Memory digunakan | | | |
|--------------|-------------------------|------------------|-----|--------|-----|
| | | Master | | Slave | |
| | | (Byte) | (%) | (Byte) | (%) |
| Flash | 131072 | 6702 | 5,1 | 9092 | 6,9 |
| SRAM | 4351 | 67 | 1,5 | 77 | 1,8 |
| EEPROM | 4096 | 248 | 6,1 | 252 | 6,2 |

Dengan melihat masih kecilnya penggunaan sumber daya prosesor kedua APILL, maka program pengaturan isyarat lalu-lintas ini masih dapat dikembangkan lebih lanjut secara leluasa dengan menggunakan sisa memori yang tersedia tanpa ada penggantian prosesor yang digunakan.

Beban Komputasi Prosesor

Program di master dan slave telah dilengkapi dengan pembangkit clock. Sebuah isyarat clock dikeluarkan dari kaki PORTE.2 mikrokontroler ATmega128A. Waktu isyarat tinggi clock mewakili saat CPU sibuk. Siklus kerja (*duty cycle*, η) isyarat clock tersebut mewakili siklus kerja CPU, yaitu persentase waktu sibuk CPU dalam suatu waktu yang tersedia yaitu satu detik.

Algoritma pengaturan isyarat lalu-lintas dijalankan setiap satu detik setelah ada sisi naik isyarat clock dari keluaran RTC. Kesibukan CPU hanya terjadi setelah terjadi sisi naik clock ini. Dari hasil pengamatan menggunakan osiloskop, kesibukan APILL

master dan slave secara umum hanya berlangsung sekitar 4 milidetik sebagaimana Gambar 7. Siklus kerja prosesor adalah 0,4 %.



Gambar 7 Isyarat clock keluaran prosesor kedua APILL

Pada setiap akhir siklus, sistem membaca data waktu dari RTC dan membaca data jadwal dari EEPROM. Kedua aktifitas ini menaikkan waktu eksekusi prosesor di APILL master dan slave menjadi 4,8 milidetik. Pada setiap akhir siklus, APILL slave mengeksekusi algoritma koordinasi. Aktifitas ini menaikkan waktu eksekusi menjadi 7,8 milidetik. Dengan demikian siklus kerja tertinggi prosesor adalah 0,78 %.

Dengan melihat rendahnya siklus kerja prosesor kedua APILL, maka sistem pengatur isyarat lalu-lintas ini masih dapat dikembangkan dengan menambah beberapa fungsi lain.

Sistem ini dapat pula dikembangkan lebih lanjut dengan menambah beberapa APILL terkoordinasi lain sebagai slave yang dapat diterapkan di beberapa persimpangan lain, misalnya di simpang empat Pasar Sentul dan Kantor Pos Yogyakarta. Sistem master – slave ini pun dapat diaplikasikan di beberapa persimpangan lain dengan mengubah pola beserta jadwal pengaturan isyarat lalu-lintas.

KESIMPULAN

Dari hasil dan pembahasan yang telah dipaparkan dapat diambil beberapa kesimpulan sebagai berikut.

1. Sistem Pengatur Isyarat Lalu-lintas Terkoordinasi yang mengatur sesuai jadwal dapat dibentuk menggunakan mikrokontroler ATmega128A.
2. Mekanisme koordinasi antara APILL master dan slave dapat dibentuk dengan menambah atau mengurangi durasi hijau pada APILL slave.
3. Proses koordinasi dapat dilakukan dalam satu sampai dua siklus.
4. Program pengaturan isyarat lalu-lintas ini cukup menggunakan variabel bertipe integer 8 dan 16 bit, dan menggunakan operasi matematika integer.

5. Program baru menggunakan tidak lebih dari 10 % sumber daya prosesor sehingga masih dapat dikembangkan secara leluasa tanpa adanya penggantian prosesor yang digunakan.

DAFTAR PUSTAKA

- Basuki, Imam, Siswandi. 2008. Biaya Kemacetan Ruas Jalan Kota Yogyakarta, *Jurnal Teknik Sipil*. 9(1): 71 – 80.
- Dephub. 1996. *Pedoman Teknis Pengaturan Lalu-lintas di Persimpangan Berdiri Sendiri dengan Alat Pemberi Isyarat Lalu-lintas*. Departemen Perhubungan. Direktur Jenderal Perhubungan Darat.
- Dresner, K., Stone, P. 2007. Sharing the Road: Autonomous Vehicles Meet Human Drivers. *The Twentieth International Joint Conference on Artificial Intelligence (IJCAI 07)*. Hyderabad, India. Hal. 1263-1268.
- Federal Highway Administration. 2008. *Traffic Signal Timing Manual*. Federal Highway Administration, U.S. Department of Transportation.
- Hanggara, Yoga. 2012. Urai Macet, Tambah ATCS. <http://dishub-diy.net/perhubungan/urai-macet-tambah-atcs.html>. Diakses 22 Februari 2013.
- Kurniawan, Freddy, Adiprasetya, Rahmad Al Hasibi. 2007. Konsep Pengatur Lalu-lintas Sinkron Adaptif Kepadatan untuk Solusi Minimalisasi Waktu Waktu Tunggu Kendaraan. *Jurnal Ilmiah Semesta Teknik*. 10(2): 126 – 135.
- Primantari, Luky FA. 2010. Koordinasi Pengaturan Lampu Lalu Lintas (Studi kasus: Ruas Jalan Prof. Dr. Sorharso - Adi Sucipto - A.Yani - Adi Soemarmo). *Jurnal Teknik Sipil dan Arsitektur*. 8(12): 13 – 30.
- Rachmadi, M. Febrian, F. Al Afif, M. Anwar Ma'sum, M. Fajar, and A. Wibowo. 2012. Beagleboard Embedded System For Adaptive Traffic Light Control System With Camera Sensor. *Journal of Computer Science and Information*. 5(2): 63 – 71.
- Wresti, M Clara. 2012. Kompas.com: Wamenhub Resmikan ATCS di Yogyakarta, edisi 20 Desember. <http://regional.kompas.com/read/2012/12/20/11155457/Wamenhub.Resmikan.ATCS.di.Yogyakarta>. Diakses 11 Maret 2013.