

**PERANCANGAN APLIKASI KOMPUTER PUSAT
UNTUK MENGENDALIKAN LAMPU
LALU LINTAS DINAMIS SECARA WIRELESS**

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Faisal Amir

NIM: 125150300111016



**PROGRAM STUDI TEKNIK INFORMATIKA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016**

PENGESAHAN

PERANCANGAN APLIKASI KOMPUTER PUSAT UNTUK MENGENDALIKAN LAMPU
LALU LINTAS DINAMIS SECARA WIRELESS

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Faisal Amir

NIM: 125150300111016

Skripsi ini telah diuji dan dinyatakan lulus pada
7 April 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Gembong Edhi Setyawan, S.T, M.T.

NIK: 201208 761201 1 001

Barlian Henryranu Prasetio, S.T, M.T.

NIK: 201102 821024 1 001

Mengetahui

Ketua Program Studi Teknik Informatika

Drs. Marji, M.T.

NIP: 19670801 199203 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 7 April 2016

Faisal Amir

NIM: 125150300111016



KATA PENGANTAR

Puja dan puji syukur kami panjatkan kehadiran Allah SWT yang mana atas limpahan rahmat, taufik serta hidayah-Nya, skripsi ini dapat terselesaikan. Skripsi yang berjudul **“Perancangan Aplikasi Komputer Pusat untuk Mengendalikan Lampu Lalu Lintas Dinamis Secara Wireless”** ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer.

Penulis menyadari bahwa penyusunan skripsi ini terlepas dari bantuan berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih kepada pihak-pihak yang telah bersedia untuk memberikan bantuan demi kelancaran penyusunan skripsi ini diantaranya:

1. Bapak Gembong Edhi Setyawan, S.T, M.T dan Barlian Henryranu Prasetyo, S.T, M.T. selaku dosen pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. Bapak Drs. Marji, M.T. selaku Ketua Program Studi Teknik Informatika Program Teknologi Informasi dan Ilmu Komputer.
3. Ayahanda dan Ibunda serta seluruh keluarga besar yang saya cintai atas segala dukungan dan semangat serta tiada henti-hentinya memberikan do’a demi terselesaikannya skripsi ini.
4. Seluruh civitas akademik Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.
5. Rekan-rekan Sistem Komputer 2012 yang selalu memberikan motivasi dalam menyelesaikan skripsi ini.

Pada akhirnya, penulis menyadari bahwa skripsi ini masih belum sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun. Penulis berharap semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi.

Malang, 7 April 2016

Penulis

faisalamir9@gmail.com

ABSTRAK

Rambu lalu lintas merupakan faktor penting dalam jalan raya. Terdapat berbagai macam rambu lalu lintas salah satunya yaitu lampu lalu lintas. Lampu lalu lintas menurut Undang-undang nomor 22 tahun 2009 adalah lampu yang mengendalikan arus lalu lintas yang terpasang di persimpangan jalan, tempat penyeberangan, dan tempat arus lalu lintas lainnya. Saat ini mulai berkembang kendali lalu lintas secara terpusat. Terdapat lampu lalu lintas yang dapat dikendalikan secara terpusat sesuai dengan keadaan lalu lintas. Sistem ini disebut dengan lampu lalu lintas dinamis (*dynamic traffic light*). Dinamis disini berarti dapat merubah estimasi waktu maupun tampilan dari lampu lalu lintas. Tampilan lampu lalu lintas dinamis biasanya menggunakan *Light Emiting Diode (LED) dotmatrix*. Karena lampu lalu lintas saat ini beragam, tidak hanya statis di warna merah harus berhenti. Ada persimpangan yang menandakan arus kendaraan yang akan jalan lurus harus berhenti, namun apabila lalu lintas padat kendaraan lurus diperbolehkan jalan terus. Disinilah peran lampu lalu lintas dinamis digunakan. Seiring perkembangan sistem kendali terpusat dan perkembangan lampu lalu lintas dinamis maka penelitian ini membahas tentang lampu lalu lintas dinamis terpusat. Dalam sistem ini terdapat 2 metode pengiriman menggunakan komunikasi *wireless* NRF24L01. Yaitu pengiriman metode auto dan metode manual. Metode auto yaitu pengiriman hanya dengan mengirim data yang berisi simbol yang nantinya diolah pada node lampu lalu lintas. Dengan menggunakan metode ini memiliki tingkat akurasi lebih tinggi dan total waktu pengirimannya mencapai 18,9 ms dan 145,72 ms untuk 3 node. Sedangkan dengan menggunakan metode manual total waktu pengiriman data rata-rata 506,54 ms dan 4292,1 ms untuk 3 node. Tingkat akurasi pengiriman data dengan menggunakan metode pengiriman auto yaitu 100% untuk 2 node dan 3 node. Sedangkan dengan metode manual tingkat akurasinya 86% untuk 2 node dan 84% untuk 3 node. Ini disebabkan metode pengiriman manual panjang data yang dikirim lebih besar dibandingkan dengan menggunakan metode auto.

Kata kunci: lampu lalu lintas dinamis, komputer pusat, NRF24L01, *LED dot matrix*, komunikasi *wireless*.

ABSTRACT

Traffic signs are important factors in the highway. There are various kinds of traffic signs one of which is a traffic light. Traffic lights under the Act number 22 of 2009 are lights that control the flow of traffic that is attached at a crossroads, crossings, and the flow of other traffic. Now starting to grow traffic control centrally. There are traffic lights that can be controlled centrally in accordance with the traffic situation. This system is called dynamic traffic light. Dynamic here means can change the estimated time and the display of traffic lights. Display dynamic traffic lights typically use Emitting Light Diode dot-matrix. Due to the traffic lights at this time vary, not just static in red must stop. There are intersections that indicate the flow of vehicles going straight road should stop, but if the heavy traffic of vehicles allowed straight road continues. Here, the role of dynamic use of traffic lights. As the development of centralized control system and dynamic development of the traffic lights, this study discusses the dynamic centralized traffic light. In this system there are two methods of delivery using wireless communication NRF24L01. Namely the delivery method of auto and manual methods. Auto method of sending only to send the data it contains symbols that will be processed at the nodes of traffic lights. By using this method has a higher accuracy rate and total delivery time of 18.9 ms and 145.72 ms to 3 nodes. While using the manual method total data transmission time average of 506.54 ms and 4292.1 ms to 3 nodes. The level of accuracy of data transmission using the auto delivery method that is 100% for the second node and a third node. While the manual method's accuracy is 86% for two nodes and 84% for 3 nodes. This is due to the long manual method of sending data that is sent greater than using auto.

Keywords: dynamic traffic lights, central computer, NRF24L01, LED dot matrix, wireless communication.

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
<i>ABSTRACT</i>	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Manfaat.....	2
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Tinjauan Pustaka	4
2.2 Dasar Teori	5
2.2.1 Lampu Lalu Lintas.....	5
2.2.2 Komunikasi <i>Wireless</i>	5
2.2.3 <i>Dot Matrix</i>	6
2.2.4 Mikrokontroler.....	6
2.2.5 Modul NRF24L01.....	7
BAB 3 METODOLOGI	10
3.1 Metode Penelitian.....	10
3.2 Studi Literatur	11
3.3 Rekayasa Kebutuhan Sistem	11
3.3.1 Gambaran Umum Sistem.....	11
3.3.2 Kebutuhan Antarmuka Eksternal.....	11

3.3.3 Kebutuhan Fungsional	12
3.3.4 Kebutuhan Non-Fungsional	12
3.4 Perancangan Sistem.....	12
3.4.1 Perancangan <i>Node</i> Komputer Pusat.....	13
3.4.2 Perancangan <i>Node</i> Lampu Lalu Lintas Dinamis	13
3.4.3 Perancangan <i>Node Forwarder</i>	13
3.5 Implementasi Sistem.....	13
3.5.1 Implementasi <i>Node</i> Komputer Pusat.....	13
3.5.2 Implementasi <i>Node</i> Lampu Lalu Lintas Dinamis	13
3.5.3 Implementasi <i>Node Forwarder</i>	13
3.6 Pengujian dan Analisis.....	13
3.6.1 Pengujian Fungsionalitas <i>Hardware</i> Lampu Lalu Lintas Dinamis ...	13
3.6.2 Pengujian Fungsionalitas <i>Software</i>	14
3.6.3 Pengujian Fungsional <i>Database</i>	14
3.6.4 Pengujian Pengiriman Data menggunakan Komunikasi <i>Wireless</i> ..	14
3.7 Kesimpulan.....	15
BAB 4 REKAYASA KEBUTUHAN.....	16
4.1 Gambaran Umum Sistem.....	16
4.1.1 Tujuan	16
4.1.2 Kegunaan	16
4.1.3 Karakteristik Pengguna	16
4.1.4 Lingkungan Operasi.....	16
4.1.5 Batasan Perancangan dan Implementasi	17
4.1.6 Asumsi dan Ketergantungan.....	17
4.2 Kebutuhan Antarmuka Eksternal.....	17
4.2.1 Kebutuhan Antarmuka Pengguna.....	17
4.2.2 Kebutuhan Antarmuka Perangkat Keras.....	17
4.2.3 Kebutuhan Antarmuka Perangkat Lunak.....	18
4.2.4 Kebutuhan Antarmuka Komunikasi	18
4.3 Kebutuhan Fungsional	18
4.4 Kebutuhan Non-Fungsional	19
4.4.1 Kebutuhan Performansi.....	19

4.4.2 Kebutuhan Keamanan.....	19
4.4.3 Kebutuhan Keselamatan.....	20
BAB 5 PERANCANGAN DAN IMPLEMENTASI	21
5.1 Perancangan Sistem.....	21
5.1.2 Perancangan <i>node</i> Komputer Pusat	22
5.1.3 Perancangan <i>node</i> Lampu Lalu Lintas Dinamis.....	28
5.1.4 Perancangan <i>Node Forwarder</i>	33
5.2 Implementasi Sistem.....	33
5.2.1 Implementasi <i>Node</i> Komputer Pusat.....	34
5.2.2 Implementasi <i>node</i> Lampu Lalu Lintas Dinamis	40
5.2.3 Implementasi <i>Node Forwarder</i>	45
BAB 6 PENGUJIAN	46
6.1 Pengujian Fungsionalitas <i>Hardware</i> Lampu Lalu Lintas Dinamis.....	46
6.1.1 Hasil Pengujian.....	46
6.1.2 Analisa Pengujian.....	49
6.2 Pengujian Fungsionalitas <i>Software</i>	50
6.2.1 Hasil Pengujian.....	50
6.2.2 Analisa Pengujian	52
6.3 Pengujian Fungsional <i>Database</i>	53
6.3.1 Hasil Pengujian.....	53
6.3.2 Analisa Pengujian	54
6.4 Pengujian Pengiriman Data Menggunakan Komunikasi <i>Wireless</i>	54
6.4.1 Hasil Pengujian.....	54
6.4.2 Analisa Pengujian	62
BAB 7 PENUTUP	64
7.1 Kesimpulan.....	64
7.2 Saran	64
DAFTAR PUSTAKA.....	65
LAMPIRAN	66

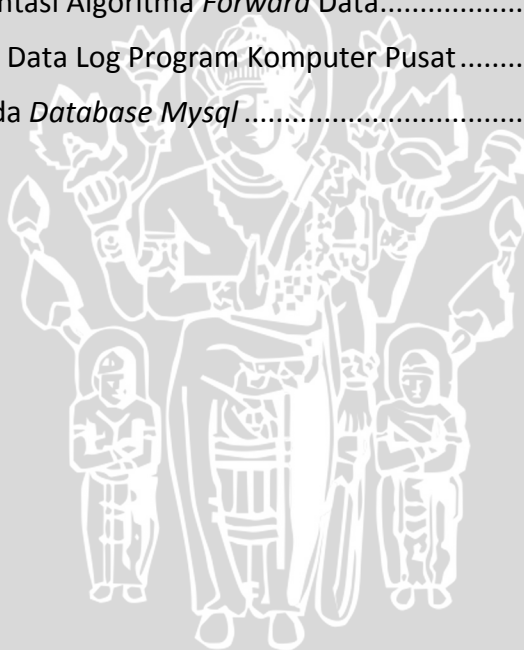
DAFTAR TABEL

Tabel 2.1 Spesifikasi NRF24L01	8
Tabel 5.1 Rangkaian NRF24L01 dan Arduino Nano	22
Tabel 5.2 Port I/O Lampu lalu lintas dinamis	29
Tabel 5.3 Port I/O <i>Shift Register</i>	30
Tabel 5.4 Port I/O Modul Komunikasi NRF24L01	30
Tabel 5.5 Spesifikasi <i>Dot matrix</i>	40
Tabel 5.6 Spesifikasi 74HC595	41
Tabel 5.7 Spesifikasi Arduino Mega 2560	41
Tabel 5.8 Spesifikasi NRF24L01	41
Tabel 6.1 Pengujian Nilai <i>Array</i>	46
Tabel 6.2 Pengujian <i>Switching</i>	48
Tabel 6.3 Pengujian Metode Auto	50
Tabel 6.4 Pengujian Metode Manual	50
Tabel 6.5 Pengujian Pengiriman 2 <i>Node</i> Auto	54
Tabel 6.6 Pengujian Pengiriman 3 <i>Node</i> Auto	55
Tabel 6.7 Pengujian Pengiriman 2 <i>Node</i> Manual	55
Tabel 6.8 Pengujian Pengiriman 3 <i>Node</i> Manual	59

DAFTAR GAMBAR

Gambar 2.1	Diagram Blok Penelitian Sebelumnya	4
Gambar 2.2	8x8 <i>Dot matrix display</i>	6
Gambar 2.3	Arduino Mega 2560 Board	7
Gambar 2.4	Antarmuka Sirkuit Modul NRF24L01	8
Gambar 3.1	Diagram Alir Metode Penelitian	10
Gambar 5.1	Gambaran Umum Sistem	21
Gambar 5.2	Diagram Blok Perancangan <i>Node</i> Komputer Pusat.....	22
Gambar 5.3	Skematik Rangkaian Komputer Pusat	22
Gambar 5.4	<i>Flowchart</i> Algoritma Pengiriman Data	23
Gambar 5.5	Format Data Metode Auto	24
Gambar 5.6	Format Data Metode Manual.....	25
Gambar 5.7	Use Case Perancangan Interface Program Lampu Lalu Lintas	26
Gambar 5.8	<i>Interface</i> Tampilan Utama Program Lampu Lalu Lintas	26
Gambar 5.9	<i>Interface</i> Metode Auto	27
Gambar 5.10	<i>Interface</i> Metode Manual	27
Gambar 5.11	Perancangan <i>Database</i> Log data.....	28
Gambar 5.12	Diagram Blok Perancangan <i>Node</i> Lampu Lalu Lintas.....	28
Gambar 5.13	Skematik Rangkaian Lampu Lalu Lintas Merah dan Kuning.....	29
Gambar 5.14	Skematik Rangkaian Lampu Lalu Lintas Hijau	29
Gambar 5.15	Skematik Rangkaian Modul komunikasi <i>node</i> lampu lalu lintas.....	30
Gambar 5.16	<i>Flowchart</i> Algoritma Penerimaan Data.....	31
Gambar 5.17	Algoritma Tampil Lampu	32
Gambar 5.18	Algoritma <i>Forward</i> Data.....	33
Gambar 5.19	Implementasi Perangkat Keras Komputer Pusat	34
Gambar 5.20	Arduino IDE	34
Gambar 5.21	<i>Library</i> RF24 Arduino.....	35
Gambar 5.22	Implementasi Algoritma Pengiriman Data.....	35
Gambar 5.23	Implementasi Header Data	36
Gambar 5.24	Implementasi Format Akhir Data	36
Gambar 5.25	Implementasi <i>Channel</i>	37

Gambar 5.26 Diagram Alir Program Komputer Pusat.....	38
Gambar 5.27 <i>Interface</i> Home Program Lampu Lalu Lintas.....	38
Gambar 5.28 <i>Interface</i> Metode Auto.....	39
Gambar 5.29 <i>Interface</i> Metode Manual	39
Gambar 5.30 <i>Interface</i> Log data	40
Gambar 5.31 Implementasi Perangkat Keras Modul Komunikasi	42
Gambar 5.32 Implementasi <i>Dot matrix</i>	42
Gambar 5.33 Implementasi Algoritma Penerimaan Data.....	43
Gambar 5.34 Implementasi Baca <i>EEPROM</i>	44
Gambar 5.35 Implementasi Tampil Lampu.....	44
Gambar 5.36 Implementasi Perangkat Keras <i>Forwarder</i>	45
Gambar 5.37 Implementasi Algoritma <i>Forward</i> Data.....	45
Gambar 6.1 Tampilan Data Log Program Komputer Pusat.....	53
Gambar 6.2 Tabel Pada <i>Database Mysql</i>	53



DAFTAR LAMPIRAN

A.1 Source code node komputer pusat	66
A.2 Source code node lampu lalu lintas	67



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Rambu lalu lintas dinilai penting bagi pengguna jalan. Rambu lalu lintas merupakan perlengkapan dari jalan yang dapat membantu para pengguna saat melintasi jalanan. Rambu lalu lintas memiliki berbagai bentuk, ada yang berupa simbol, huruf, angka, kalimat dan lain sebagainya. Rambu lalu lintas yang berupa simbol biasanya digunakan untuk menandakan keadaan dari jalan. Untuk rambu lalu lintas yang mengatur tentang tingkat kecepatan kendaraan biasanya menggunakan rambu lalu lintas berupa angka. Ada juga rambu lalu lintas yang biasa digunakan pada persimpangan yaitu lampu lalu lintas.

Menurut Undang-undang nomor 22 tahun 2009, lampu lalu lintas adalah lampu yang mengendalikan arus lalu lintas yang terpasang di persimpangan jalan, tempat penyeberangan, dan tempat arus lalu lintas lainnya. Penggunaan lampu lalu lintas sangat dibutuhkan untuk persimpangan, terutama pada kota-kota besar di Indonesia. Lampu lalu lintas ini akan menandakan kendaraan harus berhenti atau berjalan secara bergantian. Pengaturan lalu lintas di persimpangan jalan digunakan untuk mengatur kelompok kendaraan agar dapat bergerak secara teratur dan bergantian tidak saling mengganggu kendaraan dari kelompok arah yang lain. Namun saat ini terdapat beberapa pengecualian dalam pengaturan arus pada suatu titik persimpangan. Terdapat suatu titik persimpangan yang tidak memperbolehkan kendaraan untuk jalan lurus dalam satu lajur, namun apabila keadaan lalu lintas sedang padat maka kendaraan yang ingin lurus diperbolehkan langsung jalan. Disinilah peran lampu lalu lintas dinamis digunakan.

Dalam mengatur kelompok kendaraan yang ada di persimpangan maka perlu adanya pengaturan lampu lalu lintas pada persimpangan (Collotta, et al., 2014). Penjadwalan dilakukan untuk mengatasi suatu kondisi. Seperti kendaraan *ambulance* dan kendaraan pemadam kebakaran yang melintasi suatu persimpangan akan diberikan prioritas karena kondisi darurat. Penjadwalan pada lampu lalu lintas di persimpangan dapat diatasi dengan pengontrolan lampu lalu lintas terpusat. Tidak hanya terjadi penjadwalan lampu lalu lintas berdasarkan lama waktu lampu dalam lampu lalu lintas, pengontrolan juga dapat digunakan untuk memberikan rambu yang sesuai untuk keadaan lalu lintas di persimpangan. Terdapat beberapa metode dalam pengontrolan lampu lalu lintas terpusat saat ini.

Salah satu metode pengontrolan lampu lalu lintas yang saat ini mulai dikembangkan yaitu *Area Traffic Control System (ATCS)*. Menurut Direktorat Jenderal Perhubungan Darat Kementerian Perhubungan Republik Indonesia *ATCS* yaitu sistem pengendali lalu lintas terpusat yang digunakan untuk mengoptimalkan keadaan lalu lintas. *ATCS* juga dapat meminimalisasi jumlah dan beban petugas yang mengatur keadaan lalu lintas. *ATCS* menerapkan dari jenis lampu lalu lintas yang terkoordinasi. Lokasi *ATCS* atau *Central Room* memiliki

lokasi yang strategis untuk mengatur lalu lintas. *Node* yang dikontrol juga beragam sehingga infrastruktur yang digunakan yaitu *wireless*. Karena menggunakan komunikasi secara *wireless* lebih mudah untuk instalasinya disamping itu saat ini mulai banyak berkembang teknologi *wireless* yang lebih baik. Pemanfaatan ATCS tersebut dapat didukung penggunaan kendali lampu lalu lintas dinamis terpusat.

Terdapat beberapa algoritma dalam sistem pengontrolan lampu lalu lintas yang bisa digunakan. Contoh algoritma yang digunakan untuk mengendalikan lampu lalu lintas yang ada yaitu *Traffic System Communicatioin Algoritm (TSCA)* dan *Traffic Signal Time Manipulation Algoritm (TSTMA)*. Kedua algoritma ini dapat dapat mengestimasi lampu lalu lintas lebih adaptif dan efisien dengan lampu lalu lintas dinamis. Dari kedua algoritma ini dapat dikembangkan lagi agar lebih efektif dalam menangani kemacetan yang terjadi di persimpangan. (Yousef, et al., 2010). Dari algoritma yang ada dapat diambil poin-poin penting tentang keefektifan lampu lalu lintas untuk mengembangkan penelitian ini.

Dari kajian-kajian mengenai pentingnya lampu lalu lintas pada persimpangan jalan untuk menangani kemacetan yang ada, serta mulai dikembangkan pengontrolan lampu lalu lintas terpusat saat ini. Untuk itu ditawarkan suatu sistem untuk mengendalikan lampu lalu lintas dinamis secara *wireless*.

1.2 Rumusan Masalah

Berdasarkan latar belakang, dapat dirumuskan beberapa masalah sebagai berikut.

1. Bagaimana hasil perancangan komunikasi data yang terjadi antara komputer pusat dengan lampu lalu lintas dinamis?
2. Berapa lama waktu pengiriman data yang dilakukan secara *wireless* ?
3. Bagaimana akurasi pengiriman menggunakan *wireless* ?

1.3 Tujuan

Tujuan dari penelitian ini yaitu untuk mengatur jalannya komunikasi data antara komputer pusat dengan lampu lalu lintas dinamis secara *wireless*. Agar nantinya dapat mempermudah pemberian instruksi pada lampu lalu lintas dinamis.

1.4 Manfaat

Manfaat yang ingin diperoleh dari hasil penelitian ini dipaparkan sebagai berikut.

1. Mengurangi beban petugas yang mengatur lalu lintas untuk mengendalikan lampu lalu lintas pada persimpangan
2. Memberikan solusi alternatif untuk mengurai kemacetan pada persimpangan jalan yang menjadi titik rawan kemacetan

1.5 Batasan Masalah

Adapun pembatasan masalah pada penelitian ini, yaitu sebagai berikut.

1. Pembahasan difokuskan pada komunikasi yang terjadi antara komputer pusat dengan lampu lalu lintas dinamis
2. Media transfer data secara *wireless*
3. Mikrokontroler yang digunakan yaitu Arduino Mega untuk node pada lampu lalu lintas dinamisnya
4. Uji coba dilakukan dengan *prototype* lampu lalu lintas berdiameter kurang lebih 25cm
5. Tampilan lampu lalu lintas sesuai dengan program yang dibuat pada komputer pusat
6. Menggunakan NRF24L01 sebagai modul komunikasi datanya
7. Komunikasi data bekerja pada frekuensi 2,4GHz

1.6 Sistematika Pembahasan

Penulisan skematik pembahasan dan penyusunan laporan penelitian implementatif masuk dalam *development* dapat diuraikan sebagai berikut.

BAB I PENDAHULUAN

Menjelaskan tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan dari penelitian.

BAB II LANDASAN KEPUSTAKAAN

Pada bab ini akan menjelaskan tentang landasan teori yang terkait dengan penelitian. Pada bab ini juga dijelaskan tentang penelitian serupa yang pernah dilakukan.

BAB III METODE PENELITIAN

Membahas tentang langkah kerja yang dilakukan dalam penulisan diantaranya studi literatur, analisis kebutuhan sistem, implementasi dan pengujian

BAB IV REKAYASA KEBUTUHAN

Membahas tentang kebutuhan yang harus dipenuhi dalam proses penelitian yang akan dilakukan. Terdapat beberapa kebutuhan dalam penelitian seperti kebutuhan fungsional, kebutuhan non fungsional, kebutuhan user dan sebagainya.

BAB V PERANCANGAN DAN IMPLEMENTASI

Membahas tentang perancangan, analisis kebutuhan, pengujian dari komunikasi data komputer pusat dan lampu lalu lintas dinamis.

BAB VI PENGUJIAN DAN ANALISIS

Menampilkan hasil dari data yang didapat dari perancangan sistem komunikasi data yang terjadi antara komputer pusat dengan lampu lalu lintas dinamis. Kemudian menganalisis hasil apakah sesuai dengan yang diharapkan.

BAB VII PENUTUP

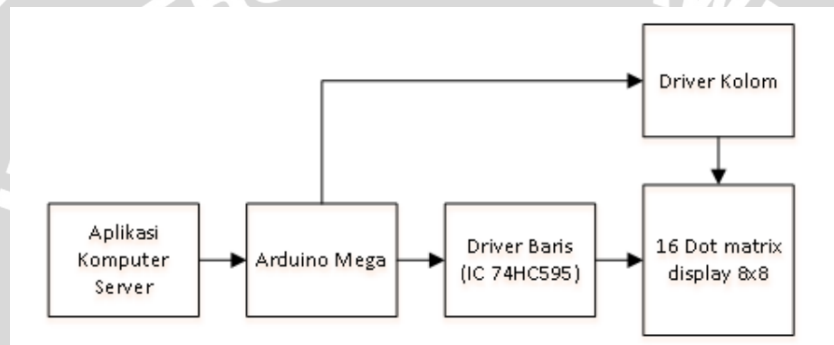
Bab ini menguraikan kesimpulan yang didapatkan dari perancangan sistem ini. Kemudian memberi saran dari hasil pengujian yang terjadi serta pengembangan dari sistem yang perlu ditingkatkan.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini akan menjelaskan tentang tinjauan pustaka yang berisi penelitian sebelumnya yang memiliki metode dan tujuan serta berisikan dasar teori yang mendukung dalam penelitian “*Perancangan Aplikasi Komputer Pusat untuk Mengendalikan Lampu Lalu Lintas Dinamis Secara Wireless*”.

2.1 Tinjauan Pustaka

Pada penelitian pengendalian lampu lalu lintas dinamis sebelumnya komunikasi data yang digunakan masih menggunakan komunikasi secara serial yang menggunakan kabel untuk proses komunikasinya. Bentuk analisis data yang digunakan dalam penelitian tersebut berdasarkan kecepatan data saat terjalin komunikasi.



Gambar 2.1 Diagram Blok Penelitian Sebelumnya

Sumber : Halim (2014)

Dalam diagram blok diatas dapat dilihat bahwa penelitian yang dilakukan masih belum memperhitungkan komunikasi yang dilakukan antara komputer pusat dengan lampu lalu lintas dinamisnya. Dalam penelitian tersebut fokus pada kecepatan data yang dihasilkan berdasarkan penggunaan dari *baudrate*. (Halim, 2014). Untuk itu dalam penelitian selanjutnya dikembangkan suatu sistem untuk mengendalikan lampu lalu lintas dinamis dengan media nirkabel.

Terdapat penelitian yang lain membahas tentang manajemen data lampu lalu lintas menggunakan *wireless*. Dalam penelitian ini menggunakan *Institute of Electrical and Electronics Engineers (IEEE) 802.15.4* untuk berkomunikasi antara *node*. Poin-poin yang diatur dalam penelitian ini yaitu mengatur tentang lama tidaknya nyala lampu hijau sesuai dengan keadaan lalu lintas. Pada penelitian ini membahas tentang efektifitas dan efisiensi dari penggunaan *dynamic traffic light*. Prinsip komunikasi data dalam penelitian tersebut menggunakan *wireless*. Komunikasi secara *wireless* saat ini dinilai sebagai terobosan terbaru yang lebih efektif karena disini juga mempertimbangkan efisiensi. (Collotta, et al., 2014).

2.2 Dasar Teori

2.2.1 Lampu Lalu Lintas

Lampu lalu lintas merupakan lampu yang digunakan untuk mengendalikan lalu lintas yang biasa terpasang pada perempatan jalan. Lampu lalu lintas memiliki kendali yang dapat ditentukan oleh petugas. Salah satu sistem kendali yang dapat digunakan untuk mengatur lampu lalu lintas yaitu sistem kendali program berdasarkan permintaan (*fully traffic actuated program*), sistem kendali ini adalah jenis kendali dimana baik urutan tahap maupun waktu yang dibutuhkan untuk tiap-tiap tahap secara otomatis dapat disesuaikan dengan permintaan dari lalu lintas tiap tahap (Dewan Standarisasi Nasional, 1992).

Berdasarkan Keputusan Menteri Perhubungan nomor KM 62 tahun 1993 dalam Lampu lalu lintas terdapat 3 jenis warna yaitu warna merah, kuning, dan hijau. Warna merah mengisyaratkan kendaraan harus berhenti sebelum batas berhenti dan apabila jalur lalu lintas tidak dilengkapi dengan batas berhenti, kendaraan harus berhenti sebelum alat pemberi isyarat lalu lintas. Warna kuning mengisyaratkan kendaraan yang belum sampai pada batas berhenti atau sebelum alat pemberi isyarat lalu lintas, bersiap untuk berhenti dan bagi kendaraan yang sudah sedemikian dekat dengan batas berhenti sehingga tidak dapat berhenti lagi dengan aman dapat berjalan. Sedangkan warna hijau mengisyaratkan kendaraan harus berjalan.

Tujuan utama adanya lampu lalu lintas yaitu untuk mengatur jalannya arus lalu lintas yang ada di persimpangan jalan. Agar pengguna tidak berebutan jalur di persimpangan dan kemacetan dapat dihindari.

2.2.2 Komunikasi *Wireless*

Komunikasi *wireless* berkembang pesat dalam sektor industri. Contoh perkembangan yang terlihat pesat yaitu dengan penggunaan telepon seluler saat ini. Dimana dengan komunikasi ini dapat menghubungkan banyak orang diseluruh dunia dengan cepat. Tidak hanya pada penggunaan telepon seluler, komunikasi *wireless* saat ini terdapat *Wireless Local Area Network (WLAN)* yang menggantikan *wired network* di beberapa sektor seperti rumah, sekolah, kampus dan lain lain. Terdapat banyak pengaplikasian komunikasi *wireless*, sebagai contoh saat ini pada *Wireless Sensor Network (WSN)*, kendali lalu lintas, perusahaan, dan *smart home* (Goldsmi, 2005).

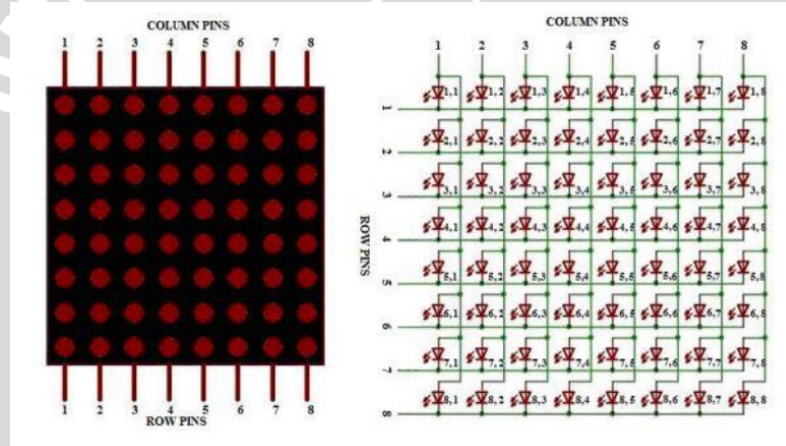
Terdapat 2 aspek yang menjadi permasalahan pada komunikasi *wireless*. Aspek-aspek tersebut dapat merubah secara signifikan pada proses komunikasi. Aspek yang pertama adalah fenomena *fading*. *Fading* yaitu gangguan pada sinyal yang menyebabkan kehilangan data pada saat pengiriman. Aspek yang kedua, tidak seperti komunikasi dengan kabel dimana masing-masing perangkat *transmitter-receiver* dianggap sebagai terisolasi link *point-to-point*, pengguna nirkabel berkomunikasi melalui udara dan ada gangguan yang signifikan antara mereka. Interferensi dapat terjadi antara pemancar komunikasi dengan

penerima, antara sinyal dari pemancar ke beberapa penerima, atau antara berbagai pengirim dan penerima (Tse, Viswaneth, dan Pramod, 2005).

2.2.3 Dot Matrix

Dot matrix adalah sejumlah *LED* yang disusun dalam kolom dan baris. *Dot matrix* memiliki berbagai macam ukuran. Untuk mengatur nyala lampu yang ada pada *dot matrix* tersebut kita harus mengetahui lokasi dari lampu yang ada. Lampu dapat dikontrol berdasarkan baris dan kolom yang digunakan dari lampu tersebut. Untuk itu perlu adanya program yang mengatur nyala lampu yang sesuai dengan kondisi yang diinginkan.

Karena *dot matriks display* ini tersusun secara matriks, maka diperlukan proses *scanning* setiap baris dan kolomnya. Proses *scanning* terdapat 2 metode yaitu *vertikal scanning* dan *horizontal scanning*. *Horizontal scanning* dilakukan dengan menyalakan *LED* secara bergantian yang dikontrol dari tiap kolomnya. *Vertikal scanning* dilakukan dengan menyalakan *LED* secara bergantian yang dikontrol melalui barisnya (Huang, 2013)



Gambar 2.2 8x8 Dot matrix display.

Sumber: Yedilkhan (2012)

Pada Gambar 2.3 menunjukkan bahwa terdapat 8x8 *Dot matrix display* dengan menggunakan 64 buah *LED* yang terdapat 8 pin kolom dan 8 pin baris. Pin kolom terhubung dengan anoda *LED*, sedangkan pin baris terhubung ke katoda *LED*. Jadi setiap *LED* dapat diakses secara individu dengan memberi *supply* ke setiap pin. Pada pin baris diberi *low signal* dan pada pin kolom diberi *high signal* untuk mengaktifkan *LED*.

2.2.4 Mikrokontroler

Mikrokontroler yang digunakan dalam penelitian ini yaitu Arduino Mega. Arduino Mega adalah *board* yang menggunakan mikrokontroler Atmega2560, *board* ini memiliki 54 digital I/O, 16 *input* analog, 16 MHz osilator kristal, koneksi USB dan tombol *reset*. Arduino Mega menggunakan koneksi USB sebagai penghubung ke komputer sekaligus sumber tegangan didapatkan dari komputer atau sumber tegangan bisa dari adaptor AC-DC 5volt agar dapat beroperasi.

Seperti ditunjukkan pada Gambar 2.3 yang merupakan bentuk fisik dari Arduino Mega 2560 (Arduino, 2016).



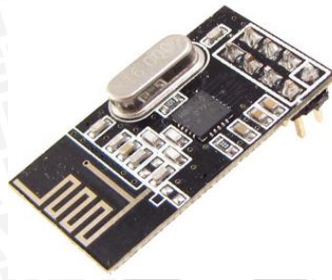
Gambar 2.3 Arduino Mega 2560 Board

Sumber: Arduino (2016)

Arduino Atmega2560 dapat beroperasi dengan daya eksternal 6 - 20 volt. Jika diberi tegangan kurang dari 7 volt maka pin 5volt akan menghasilkan tegangan kurang dari 5volt dan akan membuat Arduino tidak stabil. Jika sumber tegangan menggunakan lebih dari 12 volt, regulator tegangan akan mengalami panas dan akan merusak Arduino. Maka rentang sumber tegangan yang harus diberikan adalah 7 volt sampai 12 volt. Memori yang digunakan pada Arduino yaitu terdapat *flash memory* dengan ukuran 256KB berfungsi untuk menyimpan program. 8KB *Static Random Access Memory (SRAM)* dan 4KB *Electrically Erasable Programmable Read-Only Memory (EEPROM)* yang berfungsi sebagai *storage* ketika Arduino mati data yang disimpan pada *EEPROM* tidak hilang dan dapat dibaca kembali ketika Arduino dinyalakan. Untuk berkomunikasi dengan komputer Arduino menyediakan komunikasi serial *Universal Asynchronous Receiver Transmitter (UART)* 5 Volt. Sebuah *Atmega16U2* yang terdapat di Arduino digunakan sebagai komunikasi serial melalui USB dan muncul sebagai *COM Port* virtual pada *device* komputer (Arduino, 2015).

2.2.5 Modul NRF24L01

NRF24L01 merupakan sebuah modul komunikasi jarak jauh dengan *single-chip RF-transceiver* yang ditujukan untuk aplikasi pada gelombang 2.4 GHz *ISM band* (Nordic, 2007). Modul NRF24L01 menggunakan antar muka *Serial Peripheral Interface (SPI)* untuk komunikasi dengan tegangan kerja 3,3 VDC. *Output power* dan saluran komunikasi dapat dengan mudah dikonfigurasi dengan program. Konsumsi daya NRF24L01 rendah dan dengan berbagai mode daya rendah, NRF24L01 lebih hemat energi dan lebih nyaman. Antar muka modul NRF24L01 dapat dilihat pada Gambar 2.4 berikut.



Gambar 2.4 Antarmuka Sirkuit Modul NRF24L01

Sumber: ASA (2007)

Spesifikasi yang terdapat pada modul NRF24L01 yaitu sebagai berikut.

Tabel 2.1 Spesifikasi NRF24L01

Spesifikasi NRF24L01	
<i>Radio</i>	Worldwide 2.4GHz ISM band operation 126 RF channels Common RX and TX pins GFSK modulation 1 and 2Mbps air data rate 1MHz non-overlapping channel spacing at 1Mbps 2MHz non-overlapping channel spacing at 2Mbps
<i>Transmitter</i>	Programmable output power: 0, -6, -12 or -18dBm 11.3mA at 0dBm output power
<i>Receiver</i>	Integrated channel filters 12.3mA at 2Mbps -82dBm sensitivity at 2Mbps -85dBm sensitivity at 1Mbps Programmable LNA gain
<i>RF Synthesizer</i>	Fully integrated synthesizer No external loop filter, VCO varactor diode or resonator Accepts low cost ± 60 ppm 16MHz crystal
<i>Enhanced ShockBurst™</i>	1 to 32 bytes dynamic payload length Automatic packet handling Auto packet transaction handling 6 data pipe MultiCeiver™ for 1:6 star networks
<i>Power Management</i>	Integrated voltage regulator 1.9 to 3.6V supply range Idle modes with fast start-up times for advanced power management 22uA Standby-I mode, 900nA power down mode Max 1.5ms start-up from power down mode Max 130us start-up from standby-I mode
<i>Host Interface</i>	4-pin hardware SPI Max 8Mbps 3 separate 32 bytes TX and RX FIFOs

	5V tolerant inputs
Compact 20-pin 4x4mm QFN package	

Sumber : ASA (2007)

Modul NRF24L01 dapat di implementasikan dalam beberapa perangkat, antara lain :

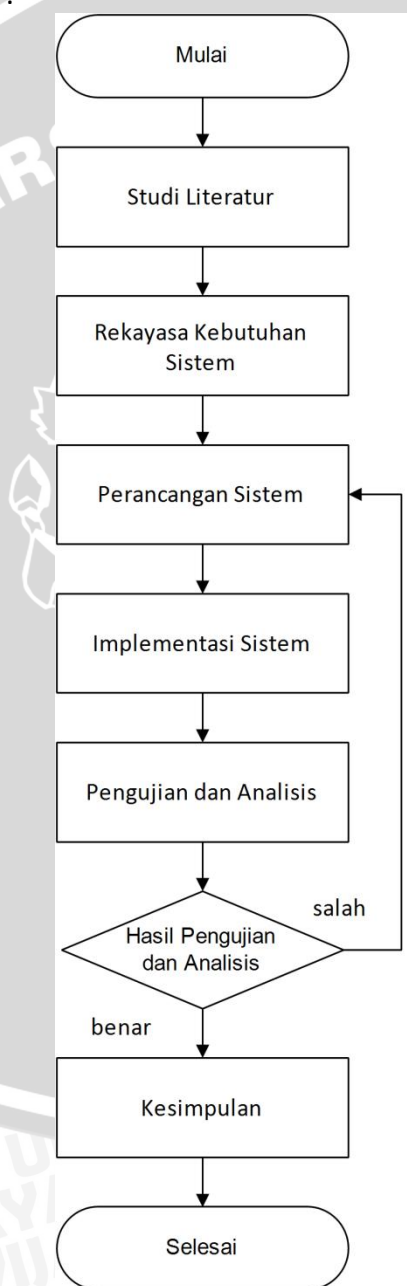
- Wireless PC Peripherals*
- Mouse, keyboard dan *remote control*
- 3-in-1 desktop bundles
- Advanced Media center remote controls
- VoIP headsets
- Pengontrol game
- Jam tangan olahraga dan *remote control*
- Remot kontrol *Radio Frequency (RF)* untuk pengguna elektronika
- Rumah dan automasi komersil
- Jaringan sensor daya rendah



BAB 3 METODOLOGI

3.1 Metode Penelitian

Berikut merupakan diagram alir yang menjelaskan tentang alur metode penelitian yang digunakan. Penelitian ini merupakan jenis penelitian implementatif *Development* atau penelitian yang mengembangkan produk jadi yang pernah ada. Tahapan-tahapan yang dilakukan mulai dari mencari literatur hingga menyelesaikan sistem. Diagram alir metode penelitian dapat dilihat pada Gambar 3.1 berikut ini :



Gambar 3.1 Diagram Alir Metode Penelitian

3.2 Studi Literatur

Dari beberapa referensi pendukung yang diperoleh dari berbagai sumber. Teori yang diambil dalam penelitian ini meliputi :

a. Teori mengenai lampu lalu lintas

Pada tahap ini dielaskan peran dari lampu lalu lintas dalam mengatur keadaan lalu lintas yang ada saat ini. Serta dicantumkan sumber referensi pendukung mengenai lalu lintas yang ada di Indonesia.

b. Teori mengenai komunikasi *wireless*

Pada tahap ini yaitu mencari beberapa referensi pendukung tentang penerapan komunikasi *wireless*, keuntungan menggunakannya, serta implementasi komunikasi yang sudah ada.

c. Teori penggunaan *dot matrix*

Tahap ini dilakukan kajian mengenai cara kerja dari *dot matrix*. Serta mencari referensi yang terkait dalam kontrol *dot matrix* dengan mikrokontroler.

d. Teori komunikasi data menggunakan modul NRF24L01.

Tahap ini mengkaji tentang cara kontrol modul NRF24L01 dengan mikrokontroler. Meliputi cara mengubah metode pengiriman data *transceiver* menjadi *receiver* atau sebaliknya.

3.3 Rekayasa Kebutuhan Sistem

Pada analisis kebutuhan menjelaskan tentang analisis beberapa kebutuhan yang diperlukan untuk menunjang penelitian ini.

3.3.1 Gambaran Umum Sistem

Menjelaskan tentang kegunaan sistem. Sistem dapat mengirim data dari *node* komputer pusat menuju ke *node* lampu lalu lintas dinamis secara *wireless* berdasarkan format data. Kegunaan dari sistem yaitu untuk mengendalikan lampu lalu lintas dinamis. Untuk lebih jelas mengenai gambaran umum sistem dapat dilihat pada bab selanjutnya.

3.3.2 Kebutuhan Antarmuka Eksternal

Dalam kebutuhan antarmuka eksternal meliputi antarmuka eksternal, antarmuka perangkat keras, antarmuka perangkat lunak, dan antarmuka komunikasi yang digunakan dalam sistem.

3.3.2.1 Antarmuka Pengguna

Merupakan kebutuhan yang menghubungkan adanya interaksi antara sistem dengan pengguna. Kebutuhan antarmuka pengguna dalam sistem ini yaitu *interface* dari program komputer pusat. Desain dari *interface* yang digunakan dalam komputer pusat dibuat dengan sederhana agar pengguna dapat mengoperasikan program komputer pusat dengan mudah.

3.3.2.2 Antarmuka Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini yaitu :

- 1 unit *Personal Computer (PC)* sebagai kendali program komputer pusat
- 3 perangkat Arduino
- 3 buah modul NRF24L01
- 1 buah *prototype* lampu lalu lintas

3.3.2.3 Antarmuka Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini yaitu :

- Aplikasi pada komputer pusat dengan menggunakan Delphi Embarcardero 2010 menggunakan bahasa pemrograman *Pascal*. Didukung dengan penggunaan komponen *comport* yang digunakan untuk menagmbil data serial yang ada pada perangkat yang digunakan.
- Arduino *Integrated Development Environment (IDE)* menggunakan bahasa pemrograman *C*. Didukung dengan menggunakan library RF24 untuk penggunaan modul komunikasi NRF24L01.

3.3.2.4 Antarmuka Komunikasi

Kebutuhan antarmuka komunikasi dalam sistem ini meliputi :

- Frekuensi yang digunakan
- Penggunaan *Channel*
- Penggunaan format data.

3.3.3 Kebutuhan Fungsional

Kebutuhan fungsional dari sistem komunikasi data antara komputer pusat dengan lampu lalu lintas dinamis ini adalah :

- Node* komputer pusat dapat mengirim data sesuai dengan format data yang ditentukan dalam sistem
- Node* lampu lalu lintas dapat menerima data dan menampilkan gambar sesuai dengan data

3.3.4 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional dari sistem komunikasi antara komputer pusat dan lampu lalu lintas dinamis yaitu :

- Kebutuhan Performasi
- Kebutuhan Keamanan
- Kebutuhan Keselamatan

3.4 Perancangan Sistem

Perancangan sistem dalam penelitian ini terjadi pada *node* komputer pusat dan *node* lampu lalu lintas.

3.4.1 Perancangan *Node* Komputer Pusat

Perancangan yang terjadi pada *node* komputer pusat terdapat perancangan perangkat keras yang digunakan serta penerapan algoritma dari sistem yang digunakan. Dalam *node* komputer pusat juga terdapat perancangan aplikasi komputer pusat yang digunakan untuk mengendalikan *node* lampu lalu lintas

3.4.2 Perancangan *Node* Lampu Lalu Lintas Dinamis

Perancangan sistem pada *node* lampu lalu lintas. Dalam *node* lampu lalu lintas terdapat perancangan perangkat keras dan perancangan algoritma yang digunakan dalam *node* lampu lalu lintas.

3.4.3 Perancangan *Node Forwarder*

Dalam perancangan sistem terdapat perancangan perangkat keras dan algoritma *forward* data dari *node* komputer pusat menuju *node* lampu lalu lintas dinamis.

3.5 Implementasi Sistem

3.5.1 Implementasi *Node* Komputer Pusat

Implementasi pada *node* komputer pusat sesuai dengan perancangan yang telah dilakukan sebelumnya. Yang meliputi perancangan perangkat keras dan perangkat lunak yang digunakan.

3.5.2 Implementasi *Node* Lampu Lalu Lintas Dinamis

Implementasi pada *node* lampu lalu lintas dinamis sesuai dengan perancangan yang telah dilakukan sebelumnya. Yang meliputi perancangan perangkat keras dan perangkat lunak yang digunakan.

3.5.3 Implementasi *Node Forwarder*

Implementasi pada *node forwarder* sesuai dengan perancangan yang dilakukan. Meliputi implementasi perangkat keras yang terdiri dari mikrokontroler dan modul komunikasi yang digunakan. Kemudian implementasi algoritma *forward* data yang dilakukan untuk meneruskan data dari *node* komputer pusat menuju *node* lampu lalu lintas dinamis.

3.6 Pengujian dan Analisis

Dalam pengujian sistem memiliki skenario pengujian sebagai berikut.

3.6.1 Pengujian Fungsionalitas *Hardware* Lampu Lalu Lintas Dinamis

3.6.1.1 Tujuan Pengujian

Tujuan dari pengujian fungsional *hardware* lampu lalu lintas yaitu untuk memastikan data yang diolah dengan hasil tampilan LED *dot matrix* sesuai.

Apabila tidak sesuai dapat mengakibatkan data yang diterima tidak dapat menampilkan lampu sesuai dengan data yang diterima.

3.6.1.2 Prosedur Pengujian

Pengujian dilakukan dengan memberikan nilai untuk menampilkan *LED dot matrix* yang digunakan. Setelah itu melakukan pengujian untuk switching ke *dot matrix* lainnya. Pengujian dilakukan dengan menghungkan mikrokontroler Arduino Mega2560 dengan *dot matrix* Merah, Kuning, dan Hijau serta dihubungkan pula dengan NR24L01.

3.6.2 Pengujian Fungsionalitas Software

3.6.2.1 Tujuan Pengujian

Tujuan dari pengujian software yaitu untuk memastikan hasil output data dari program yang ada pada komputer pusat yang masuk pada mikrokontroler Arduino yang digunakan sesuai. Apabila tidak sesuai dapat menyebabkan data gagal dikirim. Data diapat dikatakan sesuai apabila format data yang dikirim juga sesuai dengan ketentuan yang digunakan dalam perancangan sebelumnya.

3.6.2.2 Prosedur Pengujian

Pengujian dilakukan dengan menggunakan program yang ada pada komputer pusat. Kemudian mencoba untuk menampilkan isi data yang ada pada komputer pusat dan selanjutnya dicek di mikrokontroler Arduino. Apakah data yang ada pada program komputer pusat dengan data yang di Arduino sama.

3.6.3 Pengujian Fungsional Database

3.6.3.1 Tujuan Pengujian

Pengujian fungsional *database* bertujuan untuk mlihat log data yang ditampilkan pada program komputer pusat dengan log data yang tersimpan pada *database* menggunakan *Mysql*.

3.6.3.2 Prosedur Pengujian

Pengujian dilakukan dengan melihat tabel yang ada pada data log dari program kemudian membandingkan hasil data log dengan tabel yang ada pada *database Mysql*.

3.6.4 Pengujian Pengiriman Data menggunakan Komunikasi Wireless

3.6.4.1 Tujuan Pengujian

Tujuan pengujian pengiriman data yaitu untuk mengetahui total waktu pengiriman data yang dilakukan dari komputer pusat menuju ke *node* lampu lalu lintasnya. Pengujian juga melihat akurasi data yand dikirim. Dalam pengiriman dikatakan data terkirim apabila *node* lampu lalu lintas mengirimkan pesan bahwa data sudah diterima. Apabila tidak ada balasan data dari *node* lampu lalu lintas maka dinyatakan *timed out*. Kondisi *timed out* bukan berarti data tidak terkirim

ke *node* lampu lalu lintas. Kondisi ini masih memungkinkan data diterima oleh *node* lampu lalu lintas.

3.6.4.2 Prosedur Pengujian

Pengujian pengiriman data yang dilakukan antara *node* komputer pusat dengan *node* lampu lalu lintas dinamis dibagi menjadi 2 tahap. Tahap pertama yaitu pengiriman dengan menggunakan 2 *node* saja. Tahap selanjutnya yaitu pengujian menggunakan 3 *node*. Dimana terdapat 1 *node* yang menjembatani *node* komputer pusat dengan *node* lampu lalu lintas dinamisnya. Dalam pengujian dicatat waktu pengiriman apabila *node* lampu lalu lintas membalas pesan. Apabila tidak membalas maka dikatakan *timed out*. Namun bukan berarti data tidak terkirim ke *node* lampu lalu lintas.

3.7 Kesimpulan

Kesimpulan didapatkan dari hasil pengujian sistem apakah sesuai dengan rumusan masalah yang ada. Agar nantinya didapatkan hasil penelitian yang didapatkan dapat mengatasi permasalahan yang ada.



BAB 4 REKAYASA KEBUTUHAN

4.1 Gambaran Umum Sistem

Pada gambaran umum sistem terdapat tujuan, kegunaan, karakteristik pengguna, lingkungan operasi, batasan perancangan dan implementasi, serta terdapat juga asumsi dan ketergantungan dari sistem.

4.1.1 Tujuan

Tujuan dari sistem ini yaitu untuk mengatur jalannya komunikasi data antara komputer pusat dengan lampu lalu lintas dinamis secara nirkabel. Data yang diterima oleh *node* lampu lalu lintas harus sama dengan data yang dikirim oleh pengirim yang ada pada *node* komputer pusat. Komunikasi data dilakukan dengan menambahkan struktur data yang akan digunakan agar nantinya dapat dikenali oleh *node* tujuan.

Sistem ini juga bertujuan untuk mempermudah pengguna dalam mengendalikan lampu lalu lintas dinamis. Sistem ini menyediakan tampilan *Graphical User Interface (GUI)* untuk mempermudah pengguna mengoperasikan program pada komputer pusat. Keuntungan menggunakan sistem *GUI* yaitu pengguna akan mudah berinteraksi dengan program sehingga nantinya program dapat dioperasikan oleh semua kalangan. Dalam program ini juga dapat mengetahui total waktu untuk pengiriman data.

4.1.2 Kegunaan

Sistem ini berguna untuk mengontrol lampu lalu lintas dinamis melalui komputer pusat dengan mudah. Pengiriman data secara nirkabel akan mempermudah instalasi perangkat yang akan digunakan. Karena tidak perlu ada penataan kabel yang banyak, cukup dengan mengatur catu daya yang digunakan dalam rangkaian. Dari sisi program komputer pusat yang digunakan berguna mempermudah pengguna untuk mengakses data yang dibutuhkan dalam sistem.

4.1.3 Karakteristik Pengguna

Pengguna dalam sistem ini bertindak sebagai pemberi instruksi untuk menampilkan data yang sesuai pada lampu lalu lintas dinamis yang ada. pengguna juga dapat mengetahui catatan kejadian yang telah dilakukan yang telah disimpan dalam *database* sistem.

4.1.4 Lingkungan Operasi

Persyaratan yang dilakukan untuk mendukung jalannya sistem ini yaitu :

- a. Kedua *node* memiliki frekuensi dan *channel* yang sama
- b. Kedua *node* memiliki jarak radius *node* dapat bekerja
- c. Data yang dikirim harus berdasarkan program yang ada pada komputer pusat

4.1.5 Batasan Perancangan dan Implementasi

Batasan perancangan dan implementasi dari sistem ini dapat diringkas sebagai berikut :

- Lampu lalu lintas hanya mampu mengolah data yang diterima dari program komputer pusat
- Lampu lalu lintas menggunakan *prototype* yang memiliki ukuran yang tersedia
- Waktu kirim data yang dilakukan dimulai saat data masuk dari pengguna sampai dengan *node* lampu lalu lintas memberikan konfirmasi data

4.1.6 Asumsi dan Ketergantungan

Asumsi dan ketergantungan pada sistem ini meliputi :

- Node* lampu lalu lintas memiliki ketergantungan dengan program yang ada pada komputer pusat
- Program komputer pusat memiliki ketergantungan terhadap input data dari pengguna

4.2 Kebutuhan Antarmuka Eksternal

4.2.1 Kebutuhan Antarmuka Pengguna

Kebutuhan antarmuka pengguna dalam sistem ini yaitu :

- Sistem dapat menampilkan gambar pada lampu lalu lintas dinamis sesuai dengan data yang dimasukkan pada *interface* komputer pusat.
- Sistem dapat menyimpan log data keadian yang pernah dilakukan pada komputer pusat

4.2.2 Kebutuhan Antarmuka Perangkat Keras

Kebutuhan antarmuka perangkat keras yang digunakan dalam penelitian ini yaitu :

- Pada *node* komputer pusat
 - Personal Computer (PC)* dengan sistem operasi *Windows* untuk menjalankan program untuk kontrol lampu lalu lintas
 - Mikrokontroler untuk mengolah data yang masuk melalui program menggunakan Arduino Nano.
 - Modul komunikasi nirkabel menggunakan NRF24L01 untuk mengirim atau menerima data 1 buah
- Pada *node* lampu lalu lintas dinamis
 - Lampu lalu lintas dinamis menggunakan *dot matrix* untuk menampilkan gambar sesuai data yang diterima
 - Mikrokontroler untuk mengatur jalannya data pada *node* ini menggunakan Arduino Mega
 - Modul komunikasi nirkabel menggunakan NRF24L01 untuk mengirim atau menerima data 1 buah

c. Pada *node forwarder*

- Mikrokontroler untuk mengolah data yang masuk melalui program menggunakan Arduino Nano.
- Modul komunikasi nirkabel menggunakan NRF24L01 untuk mengirim atau menerima data 1 buah

4.2.3 Kebutuhan Antarmuka Perangkat Lunak

Kebutuhan antarmuka perangkat lunak yang digunakan dalam penelitian ini yaitu :

a. Pada *node* komputer pusat

- Delphi Embarcardero 2010 untuk mengola tampilan data dengan menggunakan bahasa pemrograman *Pascal*.
- Komponen *comport* pada Delphi untuk pembacaan data secara serial dari mikrokontroler yang digunakan
- *Database* untuk menyimpan *log* data sistem
- *Arduino RF24 Library* untuk mengatur modul komunikasi yang digunakan
- *Arduino IDE* untuk mengimpemntasikan bahasa pemrograman C/C++

b. Pada *node* lampu lalu lintas dinamis

- *Arduino RF24 Library* untuk mengatur modul komunikasi yang digunakan
- *Arduino IDE* untuk mengimpemntasikan bahasa pemrograman C/C++

c. Pada *node forwarder*

- *Arduino RF24 Library* untuk mengatur modul komunikasi yang digunakan
- *Arduino IDE* untuk mengimpemntasikan bahasa pemrograman C/C++

4.2.4 Kebutuhan Antarmuka Komunikasi

Kebutuhan antarmuka perangkat lunak yang digunakan dalam penelitian ini yaitu :

- Komunikasi menggunakan frekuensi 2,4 GHz
- *Channel* yang dapat digunakan yaitu antara 0 – 125
- Terdapat format data yang ditentukan antara *node* lampu lalu lintas dengan *node* komputer pusat agar dapat saling berkomunikasi dengan baik.

4.3 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang harus dipenuhi dalam sistem. Berikut beberapa kebutuhan fungsional dari sistem ini yang harus bekerja agar dapat bekerja dengan baik.

A. Node komputer pusat dapat mengirim data sesuai dengan format data yang ditentukan dalam sistem

a. Penjelasan

Fitur ini menjelaskan tentang *node* komputer pusat dapat mengolah data yang diterima dari pengguna melalui *interface* program pada komputer pusat untuk diteruskan ke mikrokontroler dan dikirim melalui modul komunikasi yang digunakan.

b. Stimulus/ respon sistem

Sistem akan menerima data yang masuk dari pengguna kemudian mengolah data pada mikrokontroler sebelum dikirim ke *node* lampu lalu lintas.

c. Kebutuhan fungsional

Fungsi ini yaitu mengolah data yang masuk dari program pada komputer pusat agar sesuai dengan format pengiriman data yang ditetapkan dalam sistem. Sehingga tidak semua *node* bisa melakukan kontrol terhadap *node* lampu lalu lintas.

B. Node lampu lalu lintas dapat menerima data dan menampilkan gambar sesuai dengan data

a. Penjelasan

Pada fungsi ini *node* lampu lalu lintas menerima data dari *node* komputer pusat. Format data yang diterima harus sesuai dengan format data yang digunakan. Apabila menerima data yang tidak sesuai dengan format data yang digunakan *node* lampu lalu lintas tidak akan memproses data yang diterima.

b. Stimulus/ respon sistem

Node lampu lalu lintas menerima data dengan format data yang ditetapkan kemudian kontroler akan menampilkan gambar pada *dot matrix* yang digunakan.

c. Kebutuhan fungsional

Fungsi ini mengharuskan *node* lampu lalu lintas untuk menerima data hanya yang memiliki format data yang sesuai dengan format data.

4.4 Kebutuhan Non-Fungsional

4.4.1 Kebutuhan Performansi

Sistem akan bekerja dengan optimal apabila tidak terdapat gangguan frekuensi dari perangkat lain. Serta tidak terjadi tabrakan kanal data yang digunakan. Untuk modul komunikasi yang digunakan bekerja pada frekuensi 2,4GHz dan dapat bekerja dengan kanal data antara 0 – 125 kanal.

4.4.2 Kebutuhan Keamanan

Keamanan dalam sistem ini terdapat pada kanal data yang digunakan dengan frekuensi 2,4GHz. Kanal yang digunakan usahakan tidak sama dengan kanal yang

digunakan oleh perangkat lain yang memiliki frekuensi yang sama. Keamanan juga terdapat pada format data yang digunakan untuk komunikasi antara *node* lampu lalu lintas dengan *node* komputer pusat. *Node* lampu lalu lintas tidak akan melakukan eksekusi data apabila format data yang diterimatidak sesuai dengan ormat data yang disepakati dalam sistem.

4.4.3 Kebutuhan Keselamatan

Kebutuhan keselamatan dalam sistem ini terdapat pada catu daya yang digunakan dalam tiap rangkaian dan keselamatan data yang digunakan. Rangkaian yang digunakan dalam sistem ini dapat bekerja optimal pada tegangan 5V. Kebutuhan keselamatan data dalam sistem ini yaitu dengan format data yang digunakan. Format data harus dijaga agar tidak dapat dikenali dari luar sistem.

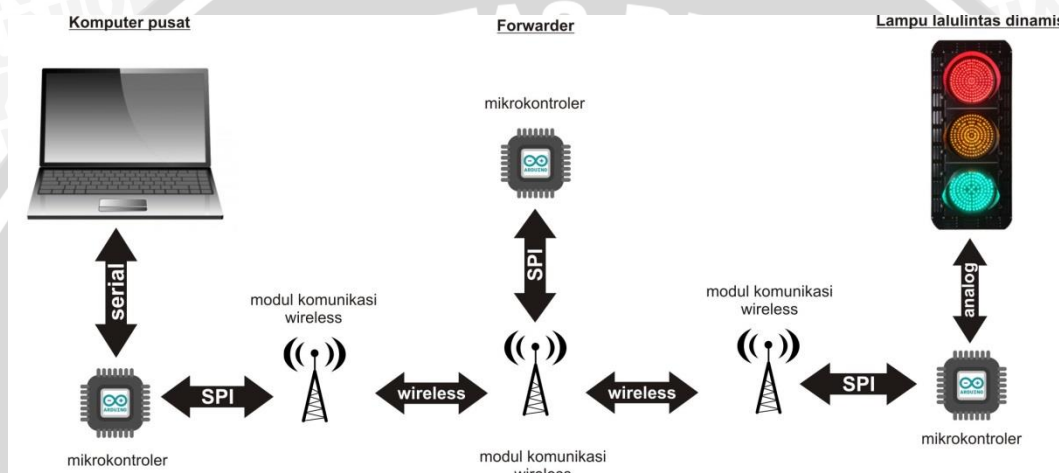


BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan membahas perancangan sistem kendali lampu lalu lintas dinamis, serta implementasi dari sistem ini.

5.1 Perancangan Sistem

Perancangan sistem meliputi perancangan pada *node* komputer pusat dan pada *node* lampu lalu lintas dinamis. Terdapat juga *node* sebagai *forwarder* penerus data yang dilakukan apabila data yang dikirim dari komputer pusat tidak dapat sampai di *node* lampu lalu lintas dinamis. Gambaran umum dari sistem dapat dilihat pada Gambar 5.1 berikut.



Gambar 5.1 Gambaran Umum Sistem

Berdasarkan Gambar 5.1 dari gambaran keseluruhan system dapat dijelaskan sebagai berikut.

1. *Node* komputer pusat
 - a. Terdapat *PC* untuk menjalankan program komputer pusat yang ada.
 - b. Mikrokontroler sebagai pengolah data dari *PC* dan pada modul komunikasi yang digunakan.
 - c. Modul komunikasi wireless digunakan untuk media pengiriman data secara *wireless*.
2. *Node forwarder*
 - a. Mikrokontroler sebagai pengolah data yang terjadi pada node ini
 - b. Modul komunikasi digunakan untuk menerima dan meneruskan data
3. *Node* lampu lalu lintas dinamis
 - a. Lampu lalu lintas dinamis untuk menampilkan data yang diterima.
 - b. Mikrokontroler untuk mengolah data yang masuk.
 - c. Modul komunikasi *wireless* untuk menerima data yang dikirim dari *node* sebelumnya.

5.1.2 Perancangan *node* Komputer Pusat

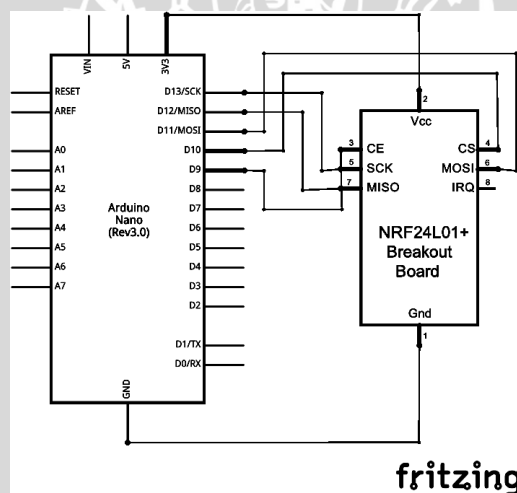


Gambar 5.2 Diagram Blok Perancangan *Node* Komputer Pusat

Pada Gambar 5.2 menjelaskan tentang diagram blok perancangan sistem yang dilakukan pada *node* komputer pusat. Dalam komputer pusat terdapat perancangan perangkat keras dan perancangan perangkat lunak yang meliputi algoritma program.

5.1.2.1 Perancangan Perangkat Keras

Perangkat keras yang digunakan dalam *node* komputer pusat yaitu mikrokontroler dan modul komunikasi. Mikrokontroler yang digunakan yaitu Arduino Nano dan modul komunikasi yang digunakan yaitu modul NRF24L01. Pada Gambar 5.3 dijelaskan rangkaian Arduino dengan NRF24L01.



Gambar 5.3 Skematik Rangkaian Komputer Pusat

Modul NRF24L01 bekerja pada tegangan 3,3V dan menggunakan port *Serial Pheripheral Interface (SPI)* pada Arduino. Berikut tabel komunikasi antara Arduino dan NRF24L01.

Tabel 5.1 Rangkaian NRF24L01 dan Arduino Nano

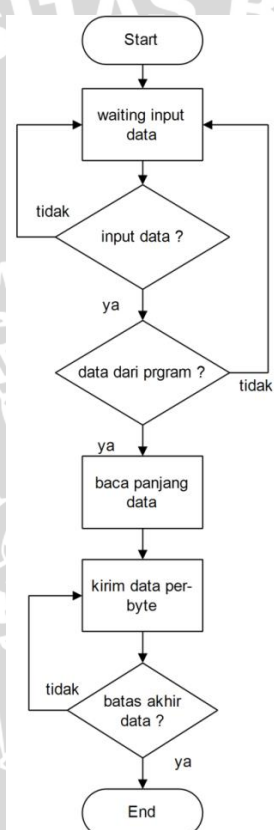
NRF24L01	Arduino Nano
GND	GND
VDD	3,3V
CE/SS	D9
CSN	D10

NRF24L01	Arduino Nano
SCK	D13
MOSI	D11
MISO	D12

5.1.2.2 Perancangan Perangkat Lunak

A. Perancangan Algoritma Pengiriman Data

Pengiriman data dilakukan oleh *node* komputer pusat. Proses pengiriman data terjadi apabila kondisi perangkat komunikasi radio dalam kondisi *stopListening*. Kondisi ini yaitu kondisi perangkat NRF24L01 yang sedang tidak menjalankan mode listening data atau menunggu data masuk. Dengan itu dapat terjadi pengiriman data. Algoritma pengiriman dapat dilihat pada Gambar 5.3.



Gambar 5.4 Flowchart Algoritma Pengiriman Data

Berdasarkan Gambar 5.4 *flowchart* algoritma pengiriman data diatas dijelaskan rincian dari tiap-tiap tahapnya sebagai berikut.

Waiting input data : Yaitu menunggu input data yang berasal dari program yang ada pada komputer pusat.

Input data : Menandakan apakah sudah terdapat input data benar dari program, jika masih belum ada data yang masuk maka akan kembali menunggu data.

- Data dari program : Memastikan apakah data yang masuk berasal dari program yang digunakan untuk memberikan instruksi pada sistem. Apabila bukan berasal dari program maka data tidak akan dijalankan
- Baca panjang data : Apabila data sudah dipastikan berasal dari program selanjutnya akan dibaca panjang data yang akan dikirim apakah sesuai dengan format data.
- Kirim data : Mengirim data menuju *node* lampu lalu lintas sampai akhir data.
- Batas akhir data : Memastikan apakah sudah mencapai batas akhir data yang telah diidentifikasi dari membaca panjang data sebelumnya. Apabila belum mencapai akhir data maka akan dikirim terus.

B. Perancangan Format Data

Format data yang digunakan dalam sistem ini terbagi menjadi 2 tipe yaitu format data metode pengiriman auto dan metode pengiriman manual. Dari kedua metode pengiriman data tersebut memiliki bentuk data yang berbeda. Metode auto memiliki struktur data lebih sedikit dibandingkan dengan metode manual. Hal ini memiliki tujuan agar dengan menggunakan metode auto pengiriman lebih cepat dibandingkan dengan menggunakan metode manual.

1. Pengiriman Metode Auto

Pengiriman menggunakan metode auto digunakan untuk menampilkan gambar rambu lalu lintas yang sering digunakan. Dengan menggunakan metode ini dimaksudkan sistem akan bekerja lebih cepat dibandingkan dengan menggunakan metode manual. Format data dengan menggunakan metode auto dapat dilihat pada Gambar 5.5 berikut.

Byte-1	Byte-2 & Byte-3	Byte-4
Header Data	Data Auto	Batas Akhir Data

Gambar 5.5 Format Data Metode Auto

Pada format data metode auto jumlah total data yang dikirim yaitu 4 byte data dimana 1 byte untuk header data, 2 byte untuk isi data, dan 1 byte untuk batas akhir data sekaligus untuk menandakan lampu yang digunakan. Batas akhir data berfungsi untuk menandakan akhir data sekaligus membedakan untuk lampu merah, kuning, atau hijau. Isi data yaitu simbol dalam *ASCII* yang menandakan data yang akan tampil pada lampu lalu lintas.

2. Pengiriman Metode Manual

Pengiriman dengan menggunakan metode manual digunakan untuk menampilkan data sesuai dengan keinginan pengguna. Bentuk input data yang diberikan berdasarkan alamat *LED* yang terpasang pada lampu lalu lintas. Format data menggunakan metode manual dapat dilihat pada Gambar 5.6 berikut.

Byte-1	Byte-2 sampai byte-129						Byte-130
Header Data	Data ke-1	Data ke-2	Data ke-3	Data ke-4	...	Data ke-128	Batas akhir Data

Gambar 5.6 Format Data Metode Manual

Panjang data menggunakan metode pengiriman manual lebih besar dibandingkan menggunakan metode auto. Data yang dikirim sebanyak jumlah bit yang ada pada lampu lalu lintas dinamis yang sama dengan jumlah *LED* yang ada pada *dot matrix* yaitu 1024 *LED* sehingga terdapat 1024 bit atau 128 byte. Header data 1 byte menandakan awal data. Batas akhir data 1 byte sekaligus menandakan untuk lampu merah, kuning, atau hijau. Keseluruhan total data menggunakan metode manual yaitu 130 byte data. Dengan menggunakan modul komunikasi NRF24L01 yang memiliki kapasitas maksimal *payload* data 8 byte. Maka dari itu pengiriman dilakukan dengan memilah data yang dikirim. Kemudian pada *node* lampu lalu lintas data akan digabung kembali.

C. Perancangan Komunikasi Data

Perancangan komunikasi dalam sistem ini meliputi penggunaan *channel* dan frekuensi.

1. Perancangan Channel Sistem

Penggunaan *channel* dalam sistem ini bertujuan untuk menghindari tabrakan data. Modul komunikasi yang digunakan mendukung 126 *channel* yang terdiri dari 0 – 125 *channel* yang dapat digunakan..

2. Perancangan Frekuensi Sistem

Penggunaan frekuensi dengan menggunakan metode komunikasi ini memiliki range frekuensi 2,400GHz sampai dengan 2.525GHz. Frekuensi yang digunakan dalam sistem berhubungan dengan *channel* yang digunakan. berikut rumus perhitungan frekuensi menggunakan modul komunikasi NRF24L01.

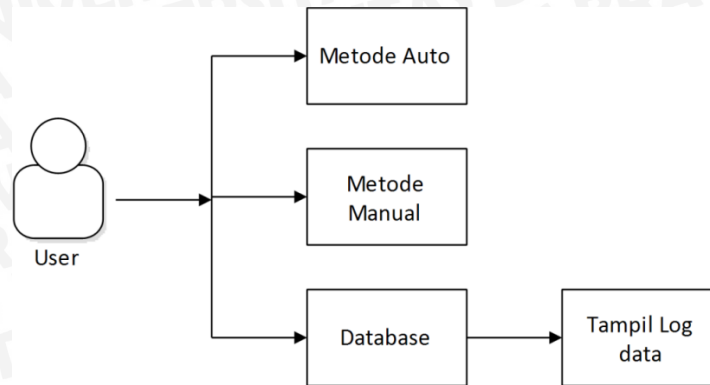
$$F_0 = 2400 + RF_CH \text{ [MHz]}$$

F_0 = Frekuensi yang digunakan

2400 = Range frekuensi awal

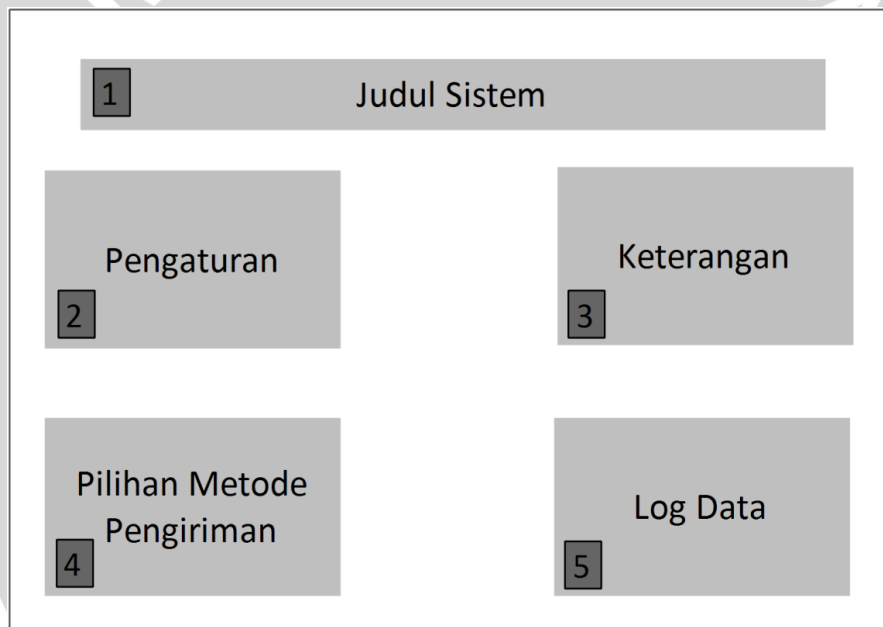
RF_CH = *Channel* yang digunakan

D. Perancangan Program Komputer Pusat



Gambar 5.7 Use Case Perancangan Interface Program Lampu Lalu Lintas

Pada Gambar 5.7 menjelaskan use case user untuk mengakses *interface* dari program lampu lalu lintas pada komputer pusat. Untuk perancangan antar muka dari sistem akan ditunjukkan pada Gambar 5.8.

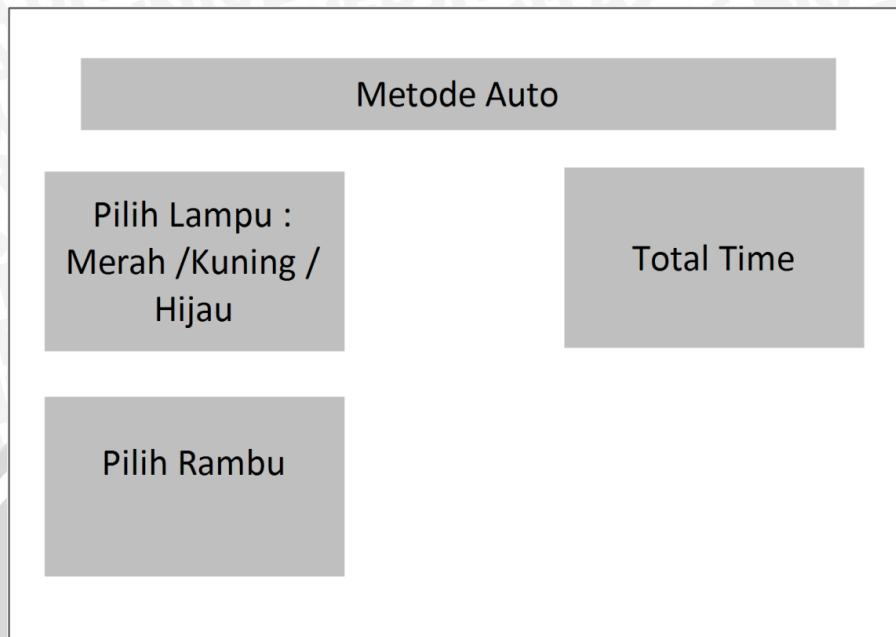


Gambar 5.8 Interface Tampilan Utama Program Lampu Lalu Lintas

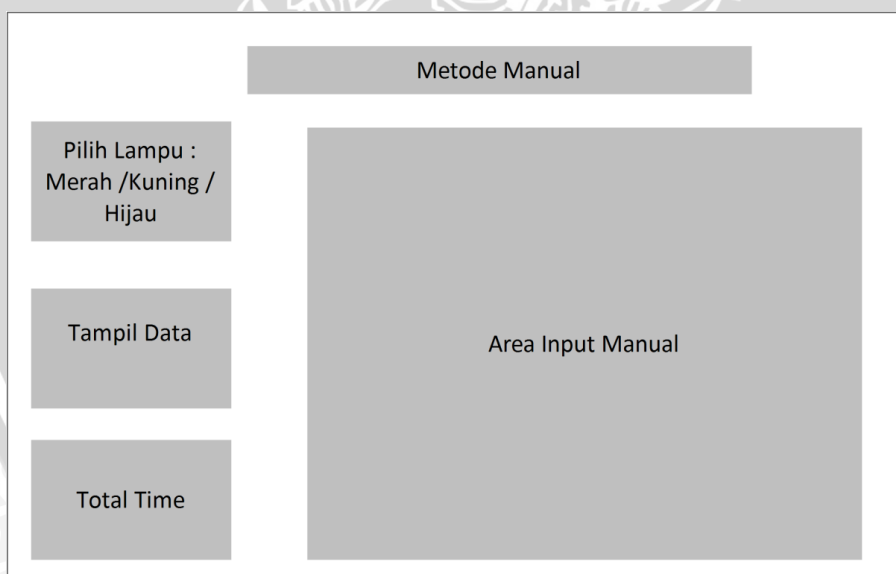
Seperti pada Gambar 5.8, berikut penjelasan bagian-bagian dari perancangan *interace* :

1. Judul Sistem terletak pada bagian paling atas *interface*
2. Pengaturan, digunakan untuk mengatur port yang akan digunakan serta untuk menghubungkan program kemikrokontroler
3. Keterangan, berisi keterangan dan cara menggunakan program
4. Pemilihan Metode Pengiriman, berisi pemilihan metode auto atau metode manual
5. Log Data, untuk menampilkan data log yang disimpan di *database*

Untuk perancangan *interface* dari metode auto dan manual dapat dilihat pada Gambar 5.9 dan 5.10 berikut.



Gambar 5.9 Interface Metode Auto



Gambar 5.10 Interface Metode Manual

E. Perancangan Database

Database pada sistem ini digunakan untuk menyimpan aktifitas pengiriman data yang pernah dilakukan *user*. Struktur rancangan desain relational diagram *database* yang digunakan pada sistem ini dijelaskan pada Gambar 5.11.

lalulintas	
id (PK)	int(11)
waktu	datetime
lampu	varchar(255)
delay	int(11)
data	varchar(1024)

Gambar 5.11 Perancangan Database Log data

Dari Gambar 5.10 berikut penjelasan dari perancangan *database* :

- Id (PK)** : Digunakan untuk menunjukkan no urut data juga digunakan sebagai *primary key* dengan *type integer* maksimum 11 digit
- Waktu** : Digunakan untuk mencatat kapan waktu eksekusi perintah terjadi pada sistem. *Type* tabel menggunakan *datetime*.
- Lampu** : Menandakan lampu mana yang sedang digunakan merah, kuning, atau hijau.
- Delay** : Digunakan untuk mencatat waktu pengiriman yang terjadi.
- Data** : Digunakan untuk menampilkan data yang dikirim.

5.1.3 Perancangan *node* Lampu Lalu Lintas Dinamis



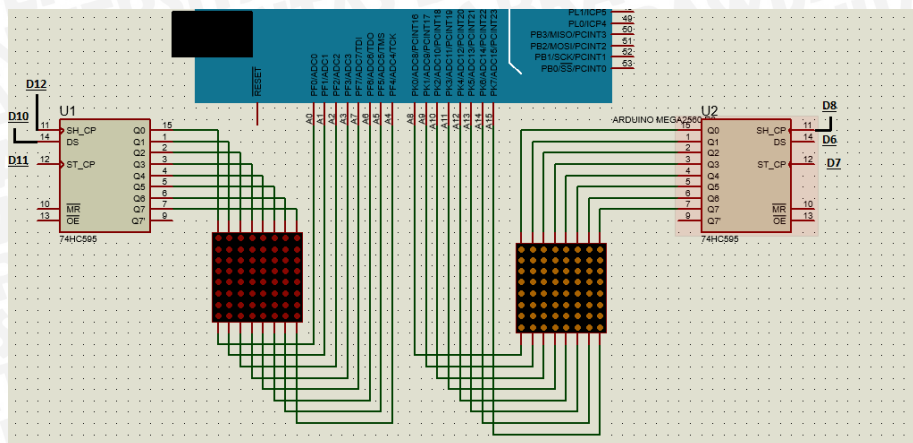
Gambar 5.12 Diagram Blok Perancangan Node Lampu Lalu Lintas

Pada Gambar 5.12 menjelaskan tentang diagram blok perancangan sistem yang dilakukan pada *node* lampu lalu lintas dinamis. Dalam *node* lampu lalu lintas dinamis terdapat perancangan perangkat keras dan perancangan perangkat lunak yang juga meliputi algoritma program.

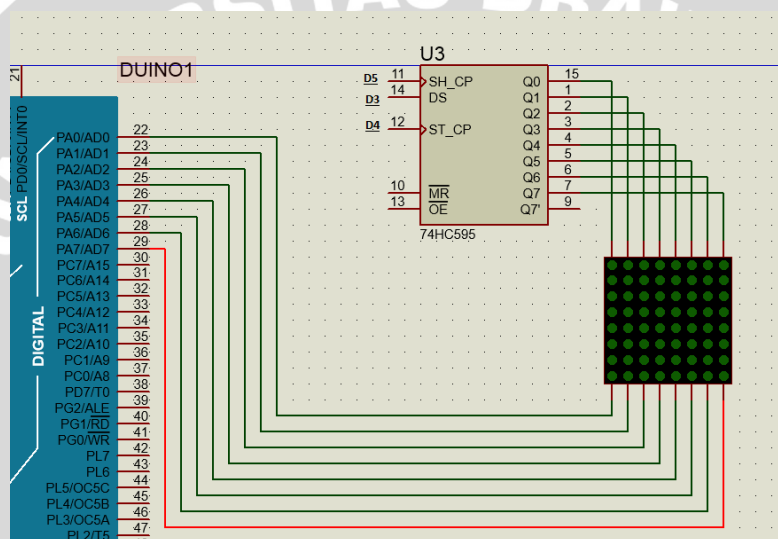
5.1.3.1 Perancangan Perangkat Keras

A. Perancangan Rangkaian *LED Dot Matrix*

Rangkaian *LED Dot Matrix* yang digunakan terdiri dari *LED Dot Matrix* dan *IC shift register*. Perancangan skematik rangkaian lampu lalu lintas dinamis dapat dilihat pada Gambar 5.13 dan Gambar 5.14 berikut.



Gambar 5.13 Skematik Rangkaian Lampu Lalu Lintas Merah dan Kuning



Gambar 5.14 Skematik Rangkaian Lampu Lalu Lintas Hijau

Terdapat 3 macam lampu yaitu merah, kuning, dan hijau. Dari ketiga lampu akan membentuk rangkaian lagi yang menjadi kesatuan 32x32 dot matrix. Arduino Mega 2560 dengan ditambahkan rangkaian shift register Integrated Circuit (IC) 74HC595. Peletakan port yang digunakan akan dijelaskan pada Tabel 5.2 dan Tabel 5.3.

Tabel 5.2 Port I/O Lampu lalu lintas dinamis

Dot matrix	Arduino Mega 2560
Dot matrix Merah kolom 1 – 8	PORTF yaitu A0, A1, A2, A3, A4, A5, A6, A7
Dot matrix Kuning kolom 1 – 8	PORTK yaitu A8, A9, A10, A11, A12, A13, A14, A15
Dot matrix Hijau kolom 1 – 8	PORTA yaitu D22, D23, D24, D25, D26, D27, D28, D29

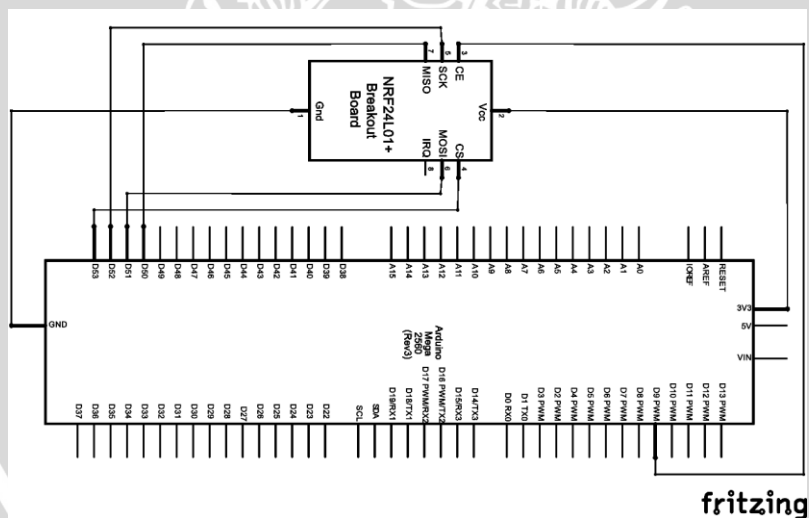
Untuk rangkaian dari *shift register* yang digunakan direpresentasikan dalam tabel 5.3 berikut.

Tabel 5.3 Port I/O Shift Register

Shift Register (74HC595)	Arduino Mega 2560
Dot matrix Merah :	
- SH-CP	D12
- ST-CP	D11
- DS	D10
Dot matrix Kuning :	
- SH-CP	D8
- ST-CP	D7
- DS	D6
Dot matrix Hijau :	
- SH-CP	D5
- ST-CP	D4
- DS	D3

B. Perancangan modul komunikasi NRF24L01

Perancangan rangkaian mikrokontroler dengan modul komunikasi NRF24L01 dapat dilihat pada Gambar 5.15 berikut.



Gambar 5.15 Skematik Rangkaian Modul komunikasi *node* lampu lalu lintas

Kombinasi pin antara Arduino Mega2560 dengan NRF24L01 dapat dilihat pada Tabel 5.4 berikut.

Tabel 5.4 Port I/O Modul Komunikasi NRF24L01

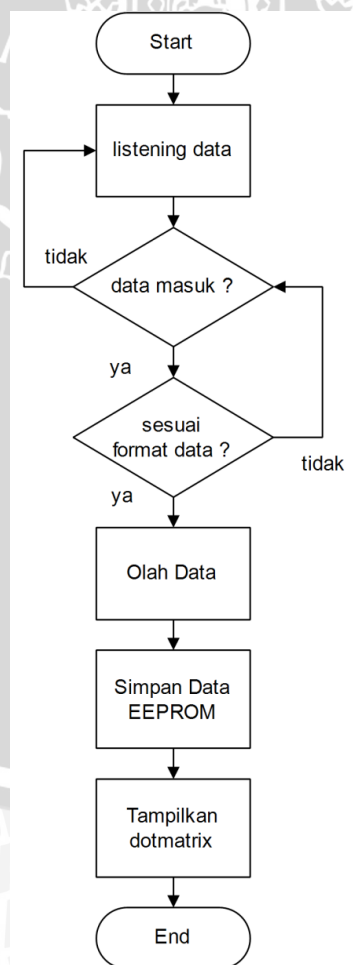
NRF24L01	Arduino Mega2560
(1) GND	GND
(2) VDD	3,3V

NRF24L01	Arduino Mega2560
(3) CE/SS	D9
(4) CSN	D53
(5) SCK	D52
(6) MOSI	D51
(7) MISO	D50

5.1.3.2 Perancangan Perangkat Lunak

A. Perancangan Algoritma Penerimaan Data

Selain terjadi pengiriman data pada sistem ini terjadi pula penerimaan data. Penerimaan data yang terjadi pada sistem ini sebagian besar dilakukan oleh *node* lampu lalu lintas. Proses penerimaan data dengan mengubah mode perangkat NRF24L01 menjadi mode *listening*. Mode ini yaitu mode selalu menerima data. Apabila data sesuai dengan format data maka data akan dijalankan. Flowcart dapat dilihat pada Gambar 5.16.



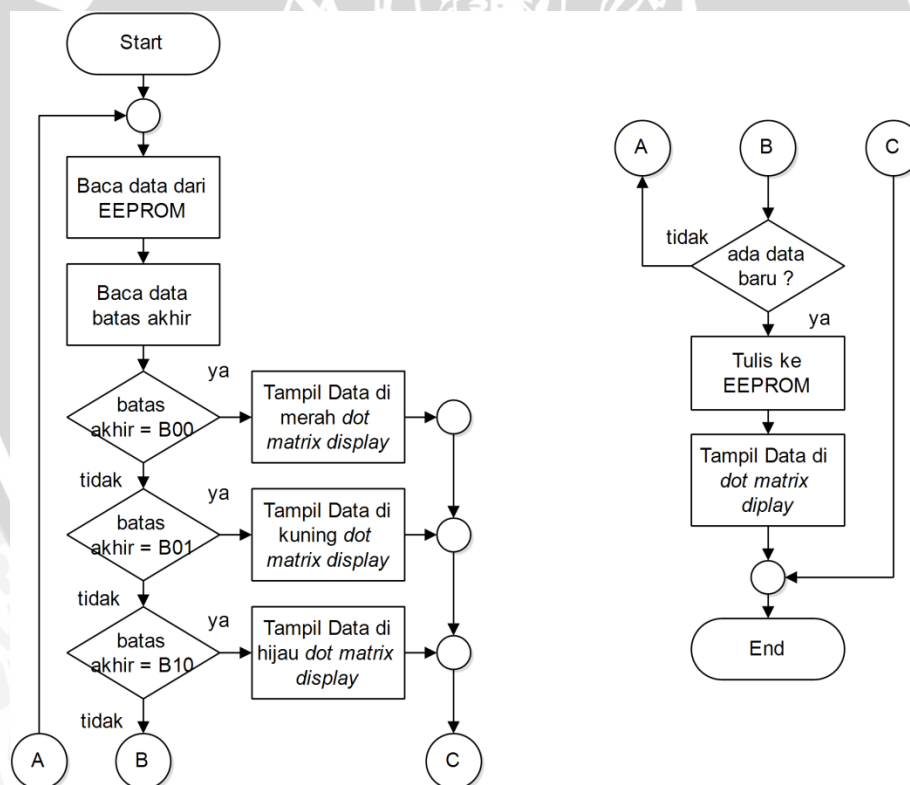
Gambar 5.16 Flowchart Algoritma Penerimaan Data

Berikut rincian penjelasan berdasarkan Gambar 5.15 flowchart penerimaan data.

- Listening data** : Yaitu menunggu data diterima dari *node* komputer pusat
- Data masuk** : Apabila ada data yang masuk melalui frekuensi dan *channel* yang sama
- Sesuai format data** : Memastikan apakah data yang masuk sesuai dengan format data. Apabila sesuai berarti data berasal dari *node* komputer pusat
- Olah data** : Olah data meliputi menggabungkan data yang terpisah, mengubah data menjadi *Integer*.
- Simpan data** : Yaitu menyimpan data pada *EEPROM* yang ada pada mikrokontroler yang digunakan
- Tampilkan data** : Menampilkan data yang diterima kedalam *dot matrix* yang ada pada *node* lampu lalu lintas dinamis.

B. Perancangan Algoritma Tampil Lampu

Algoritma untuk menampilkan lampu lalu lintas dinamis sesuai dengan data yang diterima dapat dilihat pada Gambar 5.17 berikut.



Gambar 5.17 Algoritma Tampil Lampu

Dari Gambar 5.17 terdapat langkah-langkah yang ada dalam proses untuk menampilkan lampu lalu lintas dinamis adalah sebagai berikut.

- Pertama kali membaca *EEPROM* dari mikrokontroler.

- Selanjutnya membaca batas akhir data yang tersimpan pada *EEPROM* apabila batas akhir merupakan bilangan biner 00 maka akan tampil merah, jika bilangan biner 01 maka akan tampil lampu kuning, dan jika bilangan biner 10 maka tampil lampu hijau.
- Selanjutnya membaca apakah ada data baru yang masuk, apabila ada maka akan disimpan ke *EEPROM* dan dibaca lagi seperti pada proses sebelumnya.

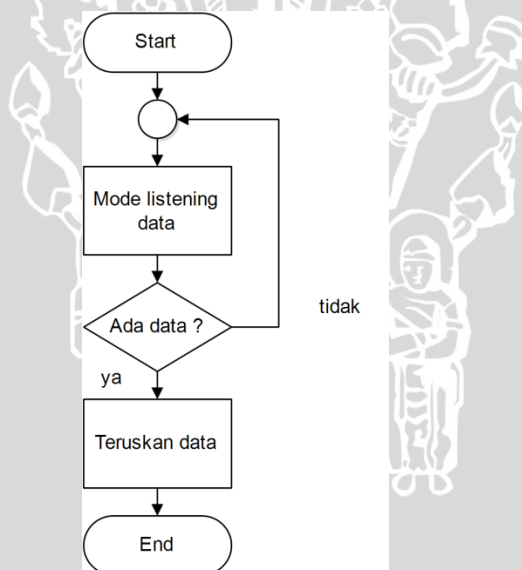
5.1.4 Perancangan *Node Forwarder*

5.1.4.1 Perancangan Perangkat Keras

Perancangan skematik yang digunakan dalam *node forwarder* memiliki skematik sama dengan yang digunakan pada *node* komputer pusat. Yaitu menggunakan Arduino nano dengan modul komunikasi NRF24L01 yang terhubung dengan menggunakan *SPI* pada Arduino. Catu daya yang digunakan menggunakan baterai atau dengan menggunakan adaptor DC dengan kapasitas tegangan 5 volt.

5.1.4.2 Perancangan Perangkat Lunak

Algoritma data yang digunakan untuk *node forwarder* digambarkan pada Gambar 5.18 berikut.



Gambar 5.18 Algoritma *Forward Data*

Dari Gambar 5.18 *node* sebagai *forwarder* pada mode listening data yaitu menunggu data masuk dari *node* komputer pusat, kemudian apabila ada data yang masuk maka data akan diteruskan menuju *node* lampu lalu lintas dinamis.

5.2 Implementasi Sistem

Implementasi sistem meliputi implementasi pada *node* komputer pusat, *node* sebagai *forwarder*, dan *node* lampu lalu lintas dinamis.

5.2.1 Implementasi *Node* Komputer Pusat

Meliputi implementasi perangkat keras yang digunakan pada *node* komputer pusat dan implementasi perangkat lunak yang digunakan.

5.2.1.1 Implementasi Perangkat Keras

Pada komputer pusat perangkat keras yang digunakan yaitu mikrokontroler Arduino Nano, modul komunikasi NRF24L01, dan *Personal Computer* (PC). Catu daya mikrokontroler yang digunakan dalam komputer pusat diambil dari catu daya serial *Universal Serial Bus* (USB) yang ada pada PC. Implementasi perangkat keras dapat dilihat pada Gambar 5.19 berikut.



Gambar 5.19 Implementasi Perangkat Keras Komputer Pusat

Pada *node* komputer pusat modul komunikasi menggunakan NRF24L01 yang terhubung dengan Arduino Nano menggunakan port *SPI*. Kemudian dari Arduino Nano ke komputer pusat terhubung dengan kabel *USB* serial.

5.2.1.2 Implementasi Perangkat Lunak

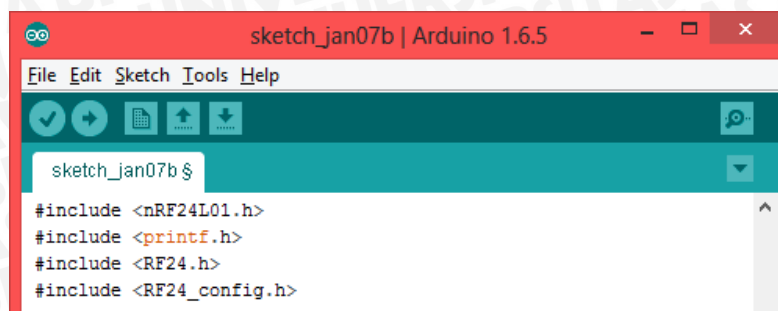
A. Implementasi Perangkat Lunak Arduino

Arduino *IDE* merupakan program berisi editor teks untuk kode, area pesan, konsol teks, *toolbar* dengan tombol untuk fungsi umum dan serangkaian menu.



Gambar 5.20 Arduino IDE

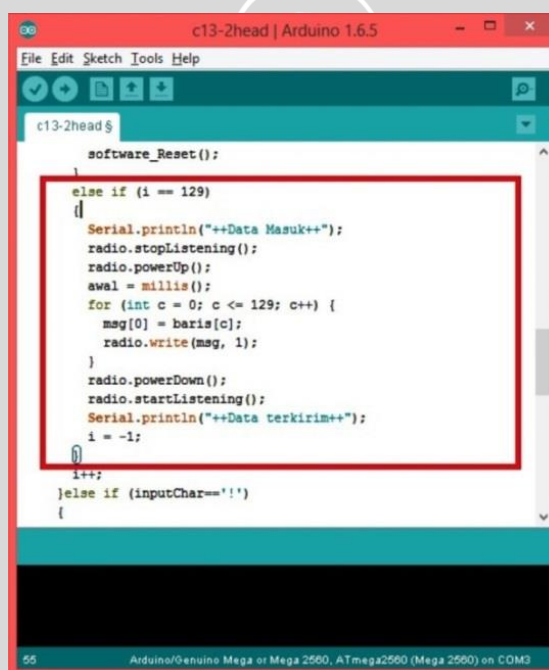
Library yang digunakan dalam sistem ini yaitu *RF24 Library*. *Library RF24* merupakan *library* untuk kendali modul komunikasi NR24L01. Dapat dilihat pada Gambar 5.21.



Gambar 5.21 Library RF24 Arduino

B. Implementasi Algoritma Pengiriman Data

Pengiriman data dalam sistem diimplementasikan dalam arduino yang mengenai pengiriman data dan penerimaan data.

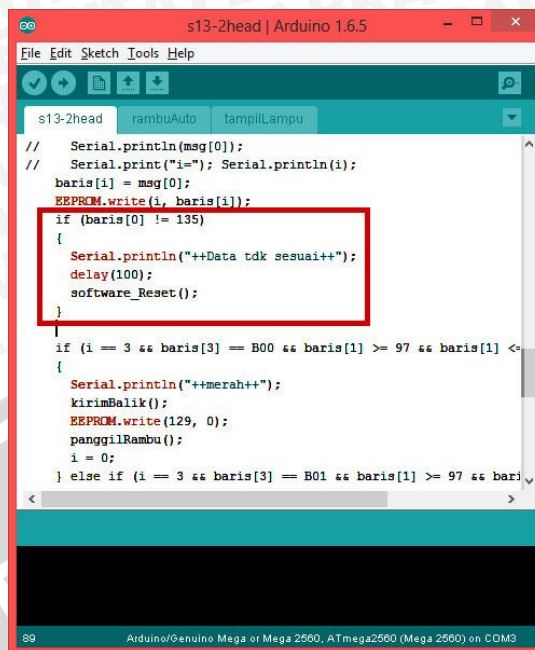


Gambar 5.22 Implementasi Algoritma Pengiriman Data

Pada implementasi pengiriman data dapat dilihat pada Gambar 5.22 yaitu menunggu input Serial Arduino. Kemudian data akan dimasukkan dalam *array* dengan panjang *array* 130 data. Setelah panjang data sesuai maka data akan dikirim

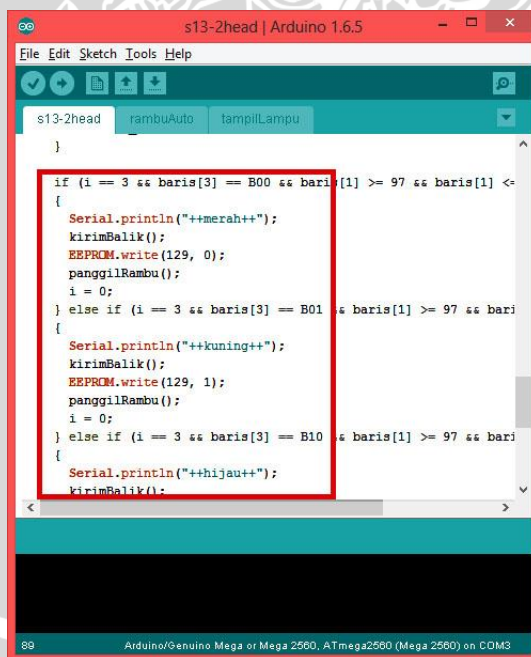
C. Implementasi Format Data pada Arduino

Format data yang ada pada sistem ini yang menjadi acuan yaitu header dan akhir data. Implementasi header dan akhir data dapat dilihat pada Gambar 5.23 dan Gambar 5.24.



Gambar 5.23 Implementasi Header Data

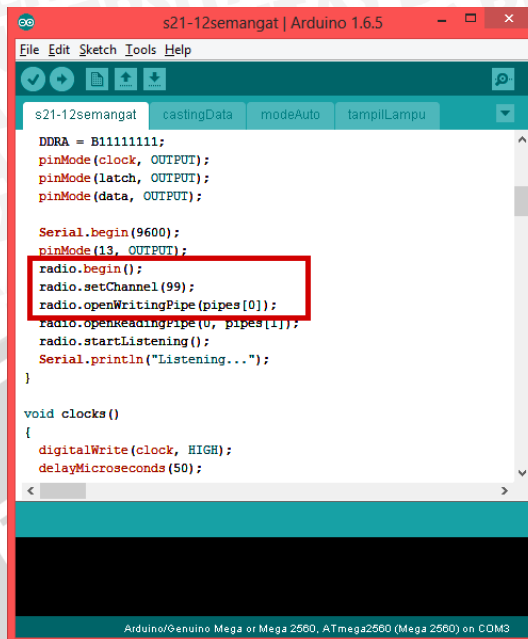
Gambar 5.23 menampilkan implementasi header data. Header data yaitu 135. Apabila header data tidak sesuai maka program akan reset dari awal.



Gambar 5.24 Implementasi Format Akhir Data

Dapat dilihat pada Gambar 5.24 batas akhir data menggunakan angka biner yang dimana apabila data *B00* akan menampilkan lampu merah, *B01* akan menampilkan lampu kuning, dan apabila *B10* akan menampilkan lampu hijau.

D. Implementasi Komunikasi Data pada Arduino



Gambar 5.25 Implementasi *Channel*

Pada Gambar 5.25 yaitu setting *channel* pada arduino. *Channel* yang digunakan pada yaitu 99. Berdasarkan channel tersebut dapat menghitung frekuensi yang digunakan.

$$F0 = 2400 + RF_CH \text{ [MHz]}$$

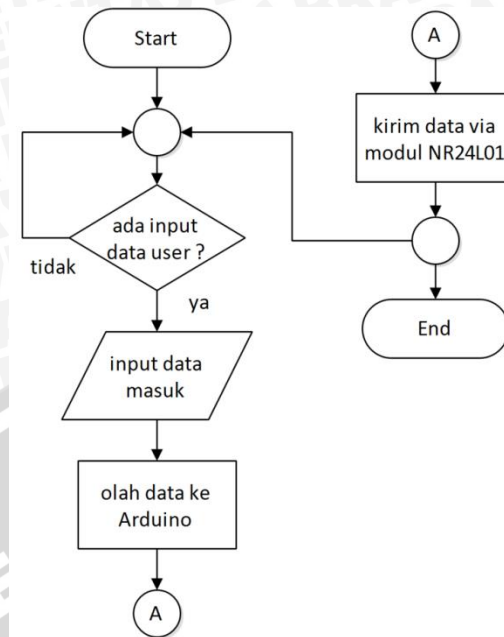
$$= 2400 + 99 \text{ [MHz]}$$

$$= 2499 \text{ [MHz]}$$

Frekuensi yang digunakan yaitu 2499 MHz atau sama dengan 2,499 GHz.

E. Implementasi Program Komputer Pusat

Perangkat yang digunakan untuk pembuatan aplikasi interface program komputer pusat yaitu *Delphi Embarcadero 2010*. *Delphi* merupakan program yang digunakan untuk membuat program baik dalam bentuk *console*, *desktop*, atau *mobile*. *Delphi* menggunakan bahasa pemrograman Pascal. Terdapat banyak komponen pendukung dalam *Delphi* yang dapat dimanfaatkan. Salah satu komponen yang dapat dimanfaatkan dalam pembuatan program lampu lalu lintas ini yaitu komponen *tcomport*. Komponen ini dapat membaca *device* yang terhubung dengannya melalui port yang digunakan. Jalannya program ini dapat dilihat pada diagram alir pada Gambar 5.26 berikut.



Gambar 5.26 Diagram Alir Program Komputer Pusat

Implementasi program pada komputer pusat menggunakan Delphi Embarcadero 2010 dengan komponen *tcomport*. Pada awal jalankan program user akan memilih port mana yang akan digunakan sebagai jalur pengiriman data. Setiap menu dalam program sudah ditampilkan dalam bentuk *GUI* dalam program ini. Untuk itu pengguna akan lebih mudah dalam pengoperasiannya.



Gambar 5.27 Interface Home Program Lampu Lalu Lintas

Pada Gambar 5.27 dapat dilihat terdapat beberapa menu yang memiliki fungsi masing-masing. Pengaturan digunakan untuk memilih port yang akan dijadikan jalur data. Pemilihan port yang digunakan yaitu alamat port Arduino yang digunakan. Untuk metode pengiriman terdapat 2 pilihan yaitu metode auto dan metode manual. Metode auto dapat dilihat pada Gambar 5.28. Dan metode manual pada Gambar 5.29.

Gambar 5.28 Interface Metode Auto

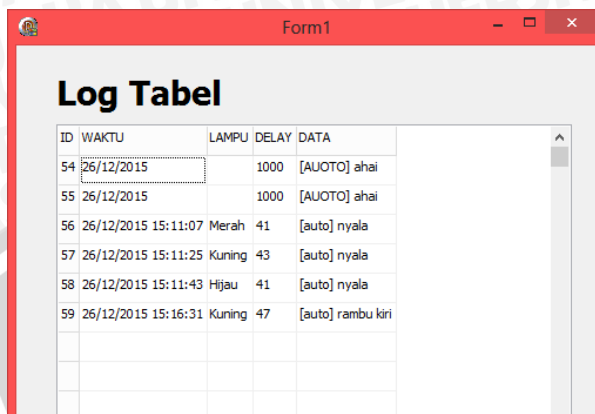
Pada Gambar 5.28 merupakan *interface* untuk program menggunakan metode auto. Langkah awal yaitu pilih lampu mana yang akan dikontrol dalam program, selanjutnya pilih rambu yang sudah tersedia pada *scrolbox*. Setelah itu kirim data. Nanti akan mendapatkan info total waktu pengiriman data yang dapat dilihat pada menu *Total Time*.

Gambar 5.29 Interface Metode Manual

Gambar 5.29 merupakan *interace* pengiriman dengan menggunakan metode manual langkah-langkah pengiriman hampir sama dengan menggunakan metode auto. Namun dalam metode ini user akan memilih satu persatu lampu mana saja yang akan diaktifkan. Kemudian pilih Tampilkan Data untuk melihat data yang akan dikirim menuju *node* lampu lalu lintas. Setelah itu pilih Kirim. Data akan dikirim menuju *node* lampu lalu lintas. Pengguna juga dapat melihat lama pengiriman data ada menu *Total Time*.

F. Implementasi Database

Setiap melakukan proses pengiriman data akan tersimpan log data pada *database* untuk mengetahui log data pengguna dapat melihat pada menu Data pilih Tampil Log. Dapat dilihat pada Gambar 5.30 berikut.



ID	WAKTU	LAMPU	DELAY	DATA
54	26/12/2015		1000	[AUOTO] ahai
55	26/12/2015		1000	[AUOTO] ahai
56	26/12/2015 15:11:07	Merah	41	[auto] nyala
57	26/12/2015 15:11:25	Kuning	43	[auto] nyala
58	26/12/2015 15:11:43	Hijau	41	[auto] nyala
59	26/12/2015 15:16:31	Kuning	47	[auto] rambu kiri

Gambar 5.30 Interface Log data

Pada *interface* log data akan menampilkan waktu pengiriman data yang pernah dilakukan, lampu yang dikontrol, delay pengiriman atau total waktu pengiriman, dan data yang dikirim. Semua log data disimpan pada *database* yang dapat dilihat sewaktu-waktu.

5.2.2 Implementasi *node* Lampu Lalu Lintas Dinamis

5.2.2.1 Spesifikasi perangkat keras

Sebelum melakukan implementasi perangkat berikut merupakan spesifikasi dari device yang digunakan dalam sistem ini. Dapat dilihat pada Tabel 5.5 sampai dengan Tabel 5.8.

- Dot matrix* 8x8

Tabel 5.5 Spesifikasi *Dot matrix*

Forward Current	25 mA
Power Dissipation	60 mW
Reverse Voltage	5 V
Peak Forward Current	150 mA
Operation Temperature	-40 to +80 °C
Storage Temperature	-40 to +80 °C
Lead Soldering Temperature	Max.260±5°C for 3 sec Max

b. Shift Register 74HC595

Tabel 5.6 Spesifikasi 74HC595

Supply Voltage	2 – 6 Volt
High-level input voltage	1.5 – 4.2 Volt
Low-level input voltage	0.5 – 1.8 Volt
Input transition rise/fall time	400 – 1000 ns
Operating free-air temperature	-40 – 85 °C

c. Mikrokontroler Arduino Mega 2560

Tabel 5.7 Spesifikasi Arduino Mega 2560

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

d. NRF24L01

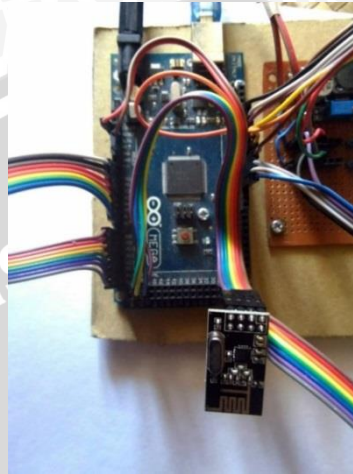
Tabel 5.8 Spesifikasi NRF24L01

Power supply	1.9 - 3.6V
Power Consumption :	
a. Idle Modes	900 nA
b. Transmit	8.0 – 11.3 mA
c. Receive	8.4 – 12.3 mA
IO port working voltage	0~3.3 volt
Transmitting rate	+7dB
Receiving sensitivity	-90dB
Transmission range	250m in open area
Dimension	15x29mm

5.2.2.2 Implementasi perangkat keras

Dalam implementasi perangkat keras yang ada pada lampu lalu lintas dinamis terdapat implementasi perangkat keras modul komunikasi yang digunakan dan implementasi *dot matrix*.

A. Implementasi Modul Komunikasi



Gambar 5.31 Implementasi Perangkat Keras Modul Komunikasi

Pada Gambar 5.31 merupakan implementasi perangkat yang digunakan dalam *node* komputer pusat. Modul komunikasi yang digunakan dalam *node* lampu lalu lintas ini menggunakan NRF24L01 port yang digunakan dalam mikrokontroler juga menggunakan port *SPI*. Namun lokasi port *SPI* yang digunakan dalam Arduino Mega2560 dengan Arduino Nano berbeda. Konfigurasi pin telah dijelaskan dalam perancangan sebelumnya.

B. Implementasi *Dot matrix*



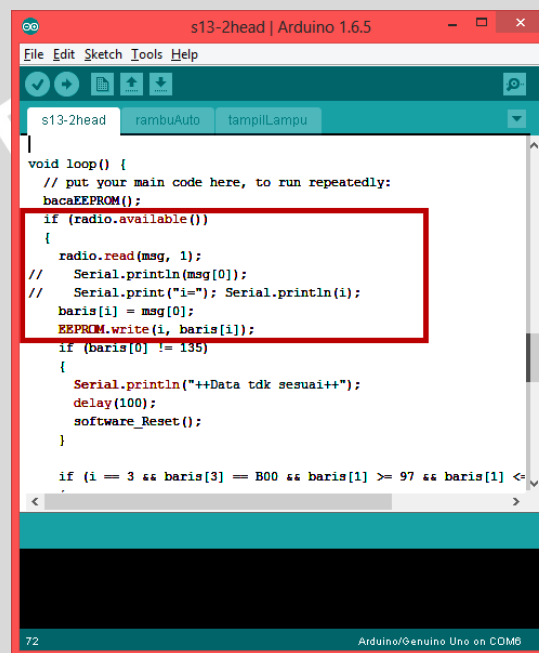
Gambar 5.32 Implementasi *Dot matrix*

Pada Gambar 5.32 merupakan implementasi aktuator dari *node* lampu lalu lintas dinamis. Implementasi *dot matrix* dalam sistem ini yaitu menggunakan 3 buah *dot matrix*. Yaitu *dot matrix* Merah, Kuning, dan Hijau. *Dot matrix* yang digunakan dalam sistem memiliki dimensi ± 25 cm setiap *dot matrix*. Terdapat pula konektor yang menghubungkan ketiga *dot matrix* ke mikrokontroler Arduino Mega2560.

5.2.2.3 Implementasi perangkat lunak

A. Implementasi Algoritma Penerimaan Data

Implementasi dari algoritma pengiriman data dapat dilihat pada Gambar 5.33 berikut.

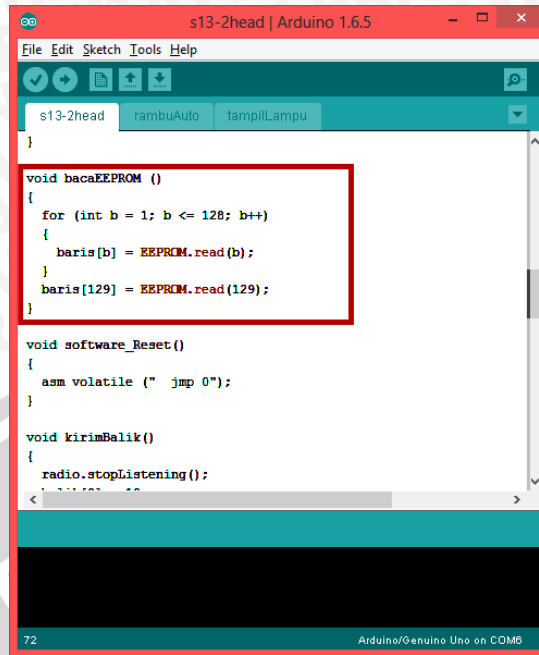


Gambar 5.33 Implementasi Algoritma Penerimaan Data

Dari Gambar 5.33 data diterima dengan memasukkan data pada *array*. *Array* yang digunakan memiliki jumlah data sesuai dengan jumlah data yang dikirim oleh *node* komputer pusat. Kemudian data juga disimpan pada *EEPROM*. Penyimpanan pada *EEPROM* berdasarkan alamat yang ditentukan.

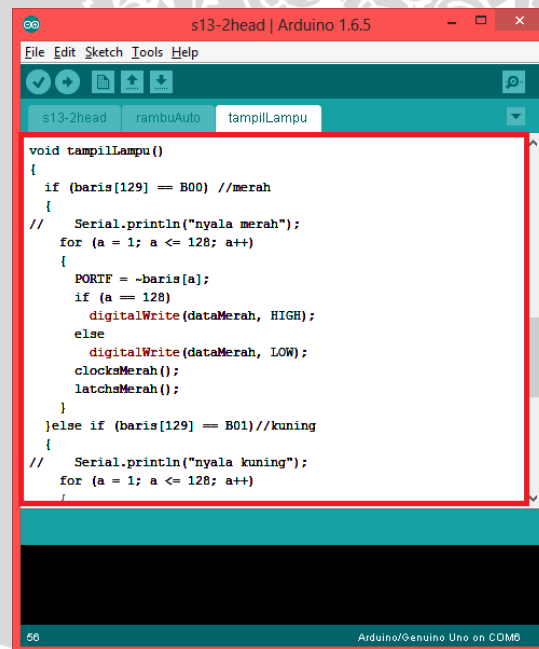
B. Implementasi Algoritma Tampil Lampu

Dalam implementasi algoritma tampil lampu yang pertama yaitu membaca data dari *EEPROM* yang ditunjukkan pada Gambar 5.33 berikut.



Gambar 5.34 Implementasi Baca *EEPROM*

Gambar 5.34 menjelaskan pembacaan data pada *EEPROM*. Pembacaan berdasarkan alamat penyimpanan data sebelumnya.



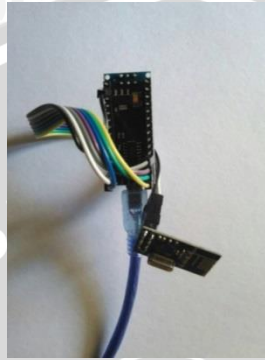
Gambar 5.35 Implementasi Tampil Lampu

Gambar 5.35 menampilkan algoritma untuk menampilkan lampu. Dimana terdapat kode yang tersimpan pada *array* ke 129 yang menentukan untuk lampu yang digunakan.

5.2.3 Implementasi *Node Forwarder*

5.2.3.1 Implementasi Perangkat Keras

Perangkat keras yang digunakan pada *node forwarder* yaitu menggunakan Arduino nano dan modul komunikasi *wireless* NRF24L01. Fungsi dari *node forwarder* disini untuk penguat sinyal dari data yang dikirim dari *node* komputer pusat menuju *node* lampu lalu lintas dinamis.

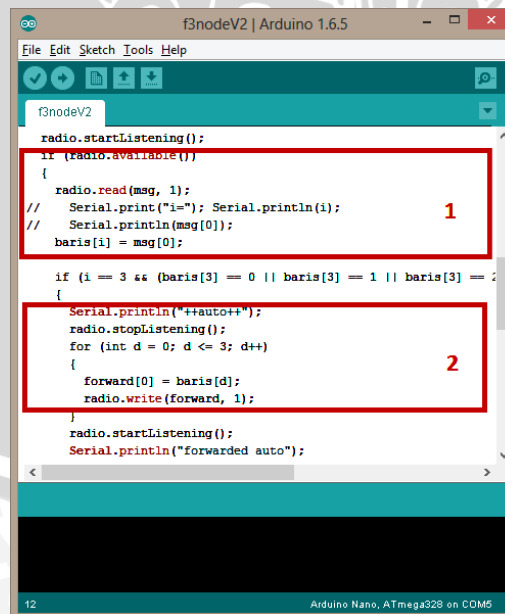


Gambar 5.36 Implementasi Perangkat Keras *Forwarder*

Gambar 5.36 merupakan implementasi perangkat keras pada *node forwarder*. Dimana Arduino nano dengan modul komunikasi NRF24L01 terhubung menggunakan *Serial Peripheral Interface (SPI)*.

5.2.3.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak *node forwarder* sesuai dengan algoritma meneruskan data yang telah dijelaskan pada sub bab sebelumnya.



Gambar 5.37 Implementasi Algoritma *Forward Data*

Pada Gambar 5.37 (1) merupakan implementasi algoritma *listening* data atau menunggu data masuk, (2) merupakan implementasi meneruskan data.

BAB 6 PENGUJIAN

Pengujian dan analisis dilakukan untuk menganalisis apakah sistem telah bekerja sesuai perancangan. Pengujian dilakukan per-blok kemudian secara keseluruhan. Pengujian setiap blok ini dilakukan untuk mempermudah analisa apabila sistem ini tidak bekerja sesuai perancangan. Pengujian yang perlu dilakukan adalah sebagai berikut:

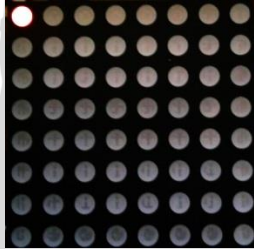
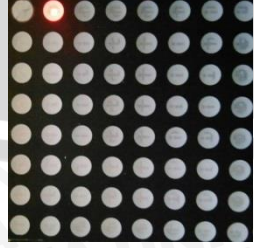
1. Pengujian fungsionalitas *hardware* lampu lalu lintas dinamis.
2. Pengujian fungsionalitas *software*.
3. Pengujian fungsional *database*.
4. Pengujian pengiriman data menggunakan komunikasi *wireless*.

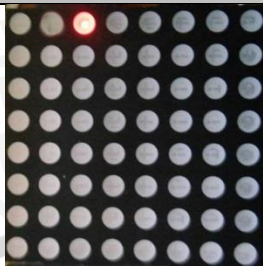
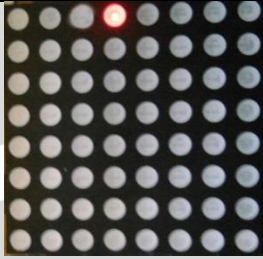
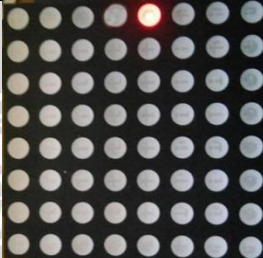
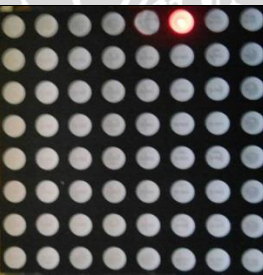
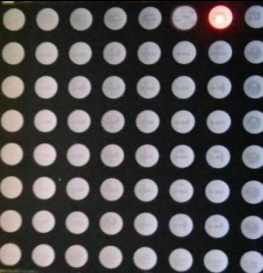
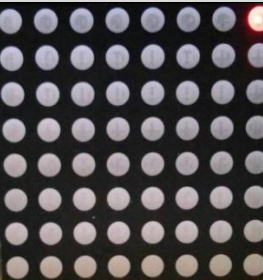
6.1 Pengujian Fungsionalitas *Hardware* Lampu Lalu Lintas Dinamis

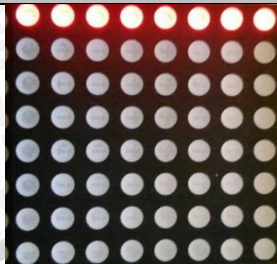
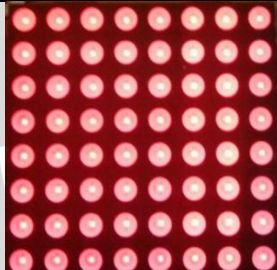
6.1.1 Hasil Pengujian

Pengujian dilakukan dengan nilai data yang masuk dengan hasil lampu yang tampil pada Tabel 6.1 berikut.

Tabel 6.1 Pengujian Nilai Array

No	Nilai Array	Hasil Keluaran	Keterangan
1	0,0,0,0,0,0,0,0		<i>LED Dot matrix</i> off semua
2	1,0,0,0,0,0,0,0		<i>LED Dot matrix</i> tampil pada baris ke-1 kolom ke-1
3	2,0,0,0,0,0,0,0		<i>LED Dot matrix</i> tampil pada baris ke-1 kolom ke-2

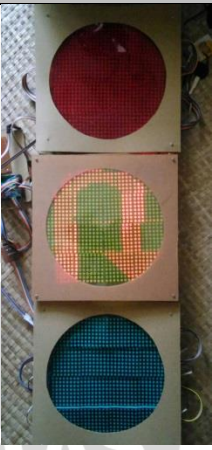
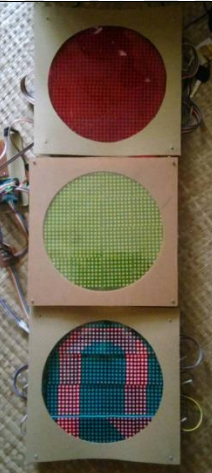
No	Nilai Array	Hasil Keluaran	Keterangan
4	4,0,0,0,0,0,0,0		LED Dot matrix tampil pada baris ke-1 kolom ke-3
5	8,0,0,0,0,0,0,0		LED Dot matrix tampil pada baris ke-1 kolom ke-4
6	16,0,0,0,0,0,0,0		LED Dot matrix tampil pada baris ke-1 kolom ke-5
7	32,0,0,0,0,0,0,0		LED Dot matrix tampil pada baris ke-1 kolom ke-6
8	64,0,0,0,0,0,0,0		LED Dot matrix tampil pada baris ke-1 kolom ke-7
9	128,0,0,0,0,0,0,0		LED Dot matrix tampil pada baris ke-1 kolom ke-8

No	Nilai Array	Hasil Keluaran	Keterangan
10	255,0,0,0,0,0,0		LED Dot matrix tampil pada semua kolom dibaris ke-1
11	255,255,255,255,255,255,255		Tampil di semua LED Dot matrix

Selanjutnya yaitu pengujian untuk *switching* dengan lampu dot-matrix merah, kuning dan hijau. Pengujian *switching dot matrix* dapat dilihat pada Tabel 6.2 berikut.

Tabel 6.2 Pengujian Switching

No	Switch	Kondisi Dotmatrix	Keterangan
1	Merah		Berhasil <i>switch</i> nyala merah saja

No	Switch	Kondisi Dotmatrix	Keterangan
2	Kuning		Berhasil <i>switch</i> nyala kuning saja
3	Hijau		Berhasil <i>switch</i> nyala hijau saja

6.1.2 Analisa Pengujian

Berdasarkan pengujian nilai *array* yang digunakan untuk menampilkan *dot matrix* pada Tabel 6.1 dapat disimpulkan bahwa untuk mengontrol sebuah *dot matrix* membutuhkan *array* sebanyak 8 nilai. Apabila terdapat 16 *dot matrix* berarti terdapat 128 nilai. Dalam sistem terdapat 3 buah *dot matrix* yang memiliki ukuran sama yaitu merah terdapat 16 *dot matrix*, kuning 16 *dot matrix*, dan hijau juga terdapat 16 *dot matrix*. Untuk itu dalam kontrol nyala lampu *switching* berdasarkan port yang digunakan sertiap *dot matrix*.

Untuk analisa mengenai proses *switching* yang dilakukan dari ketiga *dot matrix* yaitu berdasarkan port yang digunakan. Setiap *dot matrix* memiliki port yang digunakan untuk menampilkan data yang sesuai dengan hasil pengiriman data dari *node* komputer pusat.

Hasil dari pengujian yang dilakukan untuk fungsional *software* dibagi menjadi 2 macam yaitu pengujian dengan menggunakan metode auto dan menggunakan metode manual.

Tabel 6.3 Pengujian Metode Auto

No	Data Program Komputer Pusat	Data di Arduino	Akurasi 10x Percobaan	Keterangan
1	135!102!102!0!	[135,102,102,0]	100%	Nyala Merah, Putar Balik
2	135!102!102!1!	[135,102,102,1]	100%	Nyala Kuning, Putar Balik
3	135!102!102!2!	[135,102,102,2]	100%	Nyala Hijau, Putar Balik
4	135!101!101!0!	[135,101,101,0]	100%	Nyala Merah, Panah Kanan
5	135!103!103!2!	[135,103,103,2]	100%	Nyala Hijau, Orang menyebrang

Tabel 6.4 Pengujian Metode Manual

[illegible]

6.2.2 Analisa Pengujian

52

6.3.2 Analisa Pengujian

Analisis mengenai pengujian *database* yang dilakukan pada program komputer pusat menunjukkan bahwa log data yang ditampilkan pada program komputer pusat sama dengan tabel yang ada pada database *Mysql*. Data yang disimpan meliputi :

- Id** : merupakan id data urut yang tersimpan pada *database*.
- Waktu** : merupakan waktu penyimpanan data atau dapat juga dikatakan waktu program melakukan pengiriman data.
- Lampu** : merupakan lampu yang ditampilkan saat eksekusi data. Terdapat lampu merah, kuning, dan hijau
- Delay** : merupakan total delay yang terjadi saat mengirimkan data dari *node* komputer pusat ke *node* lampu lalu lintas. Dan *node* lampu lalu lintas mengkonfirmasi data suda diterima.
- Data** : merupakan data yang dikirim.

6.4 Pengujian Pengiriman Data Menggunakan Komunikasi *Wireless*

6.4.1 Hasil Pengujian

Pengujian menggunakan pengiriman 2 *node*, yaitu *node* lampu lalu lintas dan *node* komputer pusat, sedangkan untuk 3 *node*, yaitu *node* komputer pusat, *node* komputer pusat, dan *node* sebagai *Forwarder*. Pengujian dengan menggunakan metode auto dapat dilihat pada Tabel 6.5 berikut.

Tabel 6.5 Pengujian Pengiriman 2 Node Auto

No	Data Node Komputer Pusat	Data Node Lampu Lalulintas	Rata-rata Waktu 10x pengiriman (ms)	Akurasi 10x Pengiriman (%)	Keterangan
1	135!102!102!2!	[135,102,102,2]	19	100%	Hijau Rambu Putar Balik
2	135!102!102!0!	[135,102,102,2]	20,8	100%	Merah Rambu Putar Balik
3	135!102!102!0!	[135,102,102,2]	17,3	100%	Kuning Rambu Putar Balik
4	135!99!99!2!	[135,99,99,2]	18,9	100%	Hijau Rambu Panah Atas
5	135!101!101!2!	[135,101,101,2]	18,5	100%	Hijau Rambu Panah Kanan
		Rata-rata =	18,9	100%	

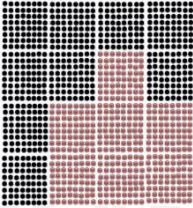
Untuk pengujian dengan menggunakan *node* tambahan sebagai forwarder dapat dilihat pada Tabel 6.6 berikut.

Tabel 6.6 Pengujian Pengiriman 3 Node Auto

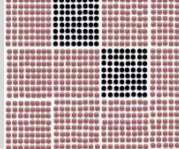
No	Data Node Komputer Pusat	Data Node Lampu Lalulintas	Rata-rata Waktu 10x pengiriman (ms)	Akurasi 10x Pengiriman (%)	Keterangan
1	135!102!102!2!	[135,102,102,2]	137,6	100%	Hijau Rambu Putar Balik
2	135!102!102!0!	[135,102,102,2]	146,5	100%	Merah Rambu Putar Balik
3	135!102!102!0!	[135,102,102,2]	145,7	100%	Kuning Rambu Putar Balik
4	135!99!99!2!	[135,99,99,2]	149,4	100%	Hijau Rambu Panah Atas
5	135!101!101!2!	[135,101,101,2]	149,4	100%	Hijau Rambu Panah Kanan
		Rata-rata =	145,72	100%	

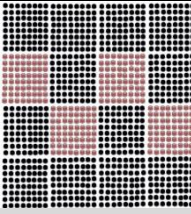
Untuk pengujian dengan menggunakan metode pengiriman manual dapat dilihat pada Tabel 6.7 berikut.

Tabel 6.7 Pengujian Pengiriman 2 *Node* Manual

No	Data Node Komputer Pusat	Data Node Lampu Lalulintas	Rata- rata Waktu 10x pengiri man (ms)	Akurasi 10x Pengiri man (%)	Keterangan
1	[135,0,0,0,0,0,0,0, ,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0, ,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0, ,0,0,0,255,255,25, 5,255,255,255,25, 5,255,0,0,0,0,0,	[135,0,0,0,0,0,0,0, ,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0, ,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0, ,0,0,0,255,255,25, 5,255,255,255,25, 5,255,0,0,0,0,0,	504,85	80%	

56

	55,255,255,255,2 55,255,255,255,2	55,255,255,255,2 55,255,255,255,2			
4	[135,255,255,255, 255,255,255,255, 255,255,255,255, 255,255,255,255, 255,255,255,255,	[135,255,255,255, 255,255,255,255, 255,255,255,255, 255,255,255,255, 255,255,255,255,	505,87	80%	

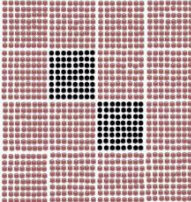
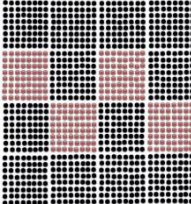
No	Data Node Komputer Pusat	Data Node Lampu Lalulintas	Rata- rata Waktu 10x pengiri man (ms)	Akurasi 10x Pengiri man (%)	Keterangan
	5,255,255,255,25 5,255,255,255,25 5,255,255,255,25 5,255,255,255,25 5,255,255,255,25 5,255,255,255,25 5,0,0,0,0,0,0,0,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,0	5,255,255,255,25 5,255,255,255,25 5,255,255,255,25 5,255,255,255,25 5,255,255,255,25 5,255,255,255,25 5,0,0,0,0,0,0,0,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,0			
5	[135,0,0,0,0,0,0,0 ,0,0,0,0,0,0,0,0,0 ,0,0,0,0,0,0,0,0,0 ,0,0,0,0,0,0,255,2 55,255,255,255,2 55,255,255,0,0,0, 0,0,0,0,0,255,255, 255,255,255,255, 255,255,0,0,0,0,0, 0,0,0,0,0,0,0,0,0 ,0,255,255,255,25 5,255,255,255,25 5,0,0,0,0,0,0,0,2 55,255,255,255,2 55,255,255,255,0, 0,0,0,0,0,0,0,0,0 ,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0 ,0,0,1]	[135,0,0,0,0,0,0,0 ,0,0,0,0,0,0,0,0,0 ,0,0,0,0,0,0,0,0,0 ,0,0,0,0,0,0,255,2 55,255,255,255,2 55,255,255,0,0,0, 0,0,0,0,0,255,255, 255,255,255,255, 255,255,0,0,0,0,0, 0,0,0,0,0,0,0,0,0 ,0,255,255,255,25 5,255,255,255,25 5,0,0,0,0,0,0,0,2 55,255,255,255,2 55,255,255,255,0, 0,0,0,0,0,0,0,0,0 ,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0 ,0,0,1]	506	80%	
		Rata-rata =	506,54	86%	

Dan untuk pengujian metode manual dengan menggunakan 3 *node* dapat dilihat pada Tabel 6.8 berikut.

Tabel 6.8 Pengujian Pengiriman 3 *Node* Manual

[illegible]

60

No	Data Node Komputer Pusat	Data Node Lampu Lalulintas	Rata- rata Waktu 10x pengiri man (ms)	Akurasi 10x Pengiri man (%)	Keterangan
4	[135,255,255,255, 255,255,255,255, 255,255,255,255, 255,255,255,255, 255,255,255,255, 255,255,255,255, 255,255,255,255, 255,255,255,255, 255,255,255,255, 255,255,255,255, 255,0,0,0,0,0,0,0, 0,255,255,255,25, 5,255,255,255,25, 5,255,255,255,25, 5,255,255,255,25, 5,255,255,255,25, 5,255,255,255,25, 5,255,255,255,25, 5,255,255,255,25, 5,255,255,255,25, 5,0,0,0,0,0,0,0,0,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,0]	[135,255,255,255, 255,255,255,255, 255,255,255,255, 255,255,255,255, 255,255,255,255, 255,255,255,255, 255,255,255,255, 255,255,255,255, 255,255,255,255, 255,255,255,255, 255,0,0,0,0,0,0,0, 0,255,255,255,25, 5,255,255,255,25, 5,255,255,255,25, 5,255,255,255,25, 5,255,255,255,25, 5,255,255,255,25, 5,255,255,255,25, 5,255,255,255,25, 5,255,255,255,25, 5,0,0,0,0,0,0,0,0,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,2 55,255,255,255,0]	4294,8	80%	
5	[135,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,255,2 55,255,255,255,2 55,255,255,0,0,0, 0,0,0,0,0,255,255,]	[135,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,255,2 55,255,255,255,2 55,255,255,0,0,0, 0,0,0,0,0,255,255,]	4292,1	80%	

Berdasarkan pengujian pengiriman data dapat dianalisis sebagai berikut.

- ### 6.4.2 Analisa Pengujian

62

menggunakan metode manual memiliki akurasi 86%. Hal ini disebabkan karena beberapa faktor berikut.

a. Faktor *fading*

Fading yaitu gangguan pada sinyal yang dapat menyebabkan gangguan daya sinyal pada modul komunikasi. Pengujian dilakukan didalam ruangan yang masih terdapat benda-benda yang dapat mengganggu aliran data.

b. Faktor *Multipath Progragation*

Merupakan gangguan sinyal yang disebabkan oleh terdapat beberapa frekuensi sinyal lain yang lebih besar dalam suatu area. Sehingga *receiver* tidak dapat menerima data secara normal. Gangguan frekuensi disebabkan oleh adanya benda-benda yang ada dalam area pengiriman data, karena setiap benda memiliki frekuensi yang berbeda. Gangguan *multipath propagation* dapat menyebabkan data tidak dapat diterima secara sebagian bahkan secara utuh, atau biasa disebut dengan *packet loss*.



BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan rumusan masalah sebelumnya maka dapat dipaparkan kesimpulan sebagai berikut.

1. Komunikasi data yang terjadi antara *node* komputer pusat dengan *node* lampu lalu lintas dinamis dibagi menjadi 2 metode pengiriman. Yang pertama yaitu dengan menggunakan metode auto. Metode pengiriman auto yaitu dengan mengirimkan tanda kepada node lampu lalu lintas dinamis, karena data yang ingin ditampilkan sudah disimpan pada *node* lampu lalu lintas. Panjang data pengiriman metode auto yaitu 4 byte. Selanjutnya yaitu metode pengiriman manual. Dimana pengiriman manual yaitu metode pengiriman data berdasarkan *dot matrix* yang ingin ditampilkan pada lampu lalu lintas dinamis. Panjang data pada metode pengiriman ini yaitu 130 byte data. Dari kedua metode pengiriman data tersebut terdapat *header* 1 byte dan batas akhir data 1 byte yang sekaligus digunakan sebagai tanda untuk menghidupkan nyala lampu merah, kuning, atau hijau.
2. Lama waktu pengiriman data dari pengujian sistem didapatkan untuk pengiriman dengan menggunakan 2 *node* metode auto yaitu 18,6 ms, sedangkan dengan menggunakan metode manual memiliki rata-rata waktu pengiriman 506,54 ms. Kemudian menggunakan 3 *node* hasil rata-rata waktu pengiriman metode auto yaitu 145,72, sedangkan dengan menggunakan metode manual waktu pengiriman mencapai 4292,1 ms.
3. Akurasi pengiriman menggunakan *wireless* dalam penelitian ini memiliki akurasi untuk pengiriman dengan menggunakan metode auto 2 *node* yang hanya mengirim data yaitu 100% dari pengujian yang dilakukan. Kemudian dengan menggunakan metode pengiriman secara manual memiliki tingkat akurasi hingga 86%. Sedangkan dengan menggunakan 3 *node* tingkat akurasi pengiriman metode auto masih 100 % sedangkan pengiriman metode manual memiliki tingkat akurasi menurun hingga 84%.

7.2 Saran

Berikut beberapa saran yang dapat dikembangkan dalam sistem ini.

1. Media komunikasi yang digunakan lebih dikembangkan agar dapat mengirim data secara optimal dan *reliable*.
2. Pemasangan perangkat komunikasi diharapkan dengan tepat. Pemasangan sebaiknya dilakukan pada tempat yang memiliki *fading* dan *multipath propagation* dari jaringan lain yang minim. Sehingga data yang dikirim lebih optimal.

DAFTAR PUSTAKA

- ASA, Nordic Semiconductor, *nRF24L01 Product Specification V2.0*. [Online] <http://www.nordicsemi.com/eng/nordic/download_resource/8041/1/62435711> [diakses tanggal 1 Februari 2016].
- Arduino, 2015. Arduino Mega 2560. [Online] Tersedia di: <<http://arduino.cc/en/Main/ArduinoBoardMega2560>> [Diakses 1 Februari 2016]
- Collotta, Mario, T. Giuffre, G. Pau, Gianfranco Scata. 2014. *Smart Traffic Light Junction Using Wireless Sensor Network*. Kore University of Enna Faculty of Engineering And Architecture Italy.
- Dewan Standarisasi Nasional – DSN, 1992. *SNI 04-2763-1992 Lampu pengatur lalu lintas*. Jakarta: ICS.
- Direktorat Jenderal Perhubungan Darat Kementerian Perhubungan Republik Indonesia, 2013. *ATCS di beberapa Provinsi dan Kota di Indonesia*. [Online] Tersedia di: <<http://hubdat.dephub.go.id/berita/1222-atcs-di-beberapa-provinsi-dan-kota-di-indonesia>> [Diakses 1 Februari 2016]
- Goldsmith, Andrea. 2005. *Wireless Communication*. Cambridge University Press.
- Halim, Alifian Nur A.. 2014. *Perancangan Aplikasi pada Komputer Pusat untuk Mengendalikan Tampilan Lampu Lalu Lintas Secara Dinamis*. Universitas Brawijaya.
- Huang, W.-F., 2013. *Three Approaches To Light an 8x8 Led Dot matrix Display*. National Kaohsiung University of Applied Sciences. IEEE.
- Keputusan Menteri Perhubungan nomor KM 62 Tahun 1993 tentang Alat Pemberi Isyarat Lalu lintas. Jakarta: Kementerian Perhubungan.
- Nordic Semiconductor. 2007. *nrf24L01 Single Chip 2.4Ghz Transceiver Product Specification*. Norwegian. (www.nordicsemi.no diakses tanggal 1 Februari 2016).
- Tse, David, Viswaneth, Pramod. 2005. *Fundamental of Wireless Communication*. University of California, Berkeley and University of Illinois, Urbana-Champaign. Cambridge University Press.
- Undang-undang Republik Indonesia nomor 22 tahun 2009 tentang Lalu lintas dan Angkutan Jalan. Jakarta: Kementerian Perhubungan.
- Yedilkhan, A., Rassim, S. & Andrey, B., 2012. *Design of an Information Display Based on Several LED Matrices and a Single Microcontroller*. Suleymen Demirel University.
- Yousef, M., Jamal N., Alkharaki, Ali M. Shatnawi. 2010. *Intellegent Traffic Light Flow Control System Using Wireless Sensor Network*. Department of Computer Engineering Jordan University of Science and Technology Irbid 22110, Jordan

LAMPIRAN

A.1 Source code node komputer pusat

```

1  #include <SPI.h>
2  #include <nRF24L01.h>
3  #include <RF24.h>
4  #include <RF24_config.h>
5
6  byte msg[1];
7  RF24 radio(9, 10);
8  const uint64_t pipes[2] = { 0xF0F0F0F0E1LL, 0xF0F0F0F0D2LL };
9  String theMessage;
10 byte baris[129];
11 byte barisAuto[1];
12 unsigned long awal, akhir;
13 int i = 0;
14 int iAuto = 0;
15 int iA1, iA2;
16 String sA1, sA2;
17
18 void setup(void) {
19   Serial.begin(115200);
20   radio.begin();
21   radio.setChannel(99);
22   radio.openWritingPipe(pipes[1]);
23   radio.openReadingPipe(1, pipes[0]);
24   radio.startListening();
25   Serial.println("Client beginning...");
26 }
27
28 void software_Reset()
29 {
30   asm volatile (" jmp 0");
31 }
32
33 void loop() {
34   if (Serial.available())
35   {
36     char inputChar = (char)Serial.read();
37     theMessage += inputChar;
38     if (inputChar == ',')
39     {
40       iA1 = theMessage.indexOf(',');
41       sA1 = theMessage.substring(0, iA1);
42       theMessage = "";
43       int a = sA1.toInt();
44       baris[i] = a;
45       if (baris[0] != 135 )
46       {
47         Serial.println(++data tdk sesuai++);
48         delay(100);
49         software_Reset();
50       }
51     } else if (i == 129)
52     {

```

```

53 Serial.println(++Data Masuk++);
54 radio.stopListening();
55 radio.powerUp();
56 awal = millis();
57 for (int c = 0; c <= 129; c++) {
58     msg[0] = baris[c];
59     radio.write(msg, 1);
60 }
61 radio.powerDown();
62 radio.startListening();
63 Serial.println(++Data terkirim++);
64 i = -1;
65 }
66 i++;
67 }else if (inputChar=='!')
68 {
69     Serial.println(theMessage);
70     iA2 = theMessage.indexOf(',');
71     sA2 = theMessage.substring(0, iA2);
72     int d = sA2.toInt();
73     barisAuto[iAuto]= d;
74     theMessage="";
75     Serial.println(d);
76     if (iAuto==3){
77         radio.stopListening();
78         radio.powerUp();
79         awal = millis();
80         for (int c = 0; c <= 3; c++) {
81             msg[0] = barisAuto[c];
82             radio.write(msg, 1);
83         }
84         radio.powerDown();
85         radio.startListening();
86         Serial.println(++Auto terkirim++);
87         iAuto=-1;
88     }
89     iAuto++;
90 }
91 }
92 if (radio.available())
93 {
94     radio.read(msg, 1);
95     akhir = millis() - awal;
96     Serial.print("waktu="); Serial.println(akhir);
97     Serial.print("T");Serial.print(akhir);Serial.println("*");
98     delay(300);
99     software_Reset();
100 }
101 }

```

A.2 Source code node lampu lalu lintas

```

1 #include <EEPROM.h>
2 #include <SPI.h>
3 #include "RF24.h"
4
5 RF24 radio(9, 53);

```

```

6  const uint64_t pipes[2] = { 0xF0F0F0F0E1LL, 0xF0F0F0F0D2LL };
7  byte msg[1], balik[1];
8  byte baris[129];
9  int i = 0;
10 int a, inRambu;
11
12 int clockMerah = 12;
13 int latchMerah = 11;
14 int dataMerah = 10;
15 int clockKuning = 8;
16 int latchKuning = 7;
17 int dataKuning = 6;
18 int clockHijau = 5;
19 int latchHijau = 4;
20 int dataHijau = 3;
21
22 void setup() {
23     DDRF = B11111111; //merah
24     DDRK = B11111111; //kuning
25     DDRA = B11111111; //hijau
26     pinMode(clockMerah, OUTPUT);
27     pinMode(latchMerah, OUTPUT);
28     pinMode(dataMerah, OUTPUT);
29     pinMode(clockKuning, OUTPUT);
30     pinMode(latchKuning, OUTPUT);
31     pinMode(dataKuning, OUTPUT);
32     pinMode(clockHijau, OUTPUT);
33     pinMode(latchHijau, OUTPUT);
34     pinMode(dataHijau, OUTPUT);
35     Serial.begin(115200);
36     radio.begin();
37     radio.setChannel(99);
38     radio.openWritingPipe(pipes[0]);
39     radio.openReadingPipe(0, pipes[1]);
40     radio.startListening();
41     Serial.println("Server beginning...");
42 }
43
44 void bacaEEPROM ()
45 {
46     for (int b = 1; b <= 128; b++)
47     {
48         baris[b] = EEPROM.read(b);
49     }
50     baris[129] = EEPROM.read(129);
51 }
52
53 void software_Reset()
54 {
55     asm volatile (" jmp 0");
56 }
57 void kirimBalik()
58 {
59     radio.stopListening();
60     balik[0] = 10;
61     radio.write(balik, 1);
62     radio.startListening();

```

```

63   Serial.println("++Kirim balik->>>");
64   }
65
66   void loop() {
67       // put your main code here, to run repeatedly:
68       bacaEEPROM();
69       if (radio.available())
70       {
71           radio.read(msg, 1);
72           baris[i] = msg[0];
73           EEPROM.write(i, baris[i]);
74           if (baris[0] != 135)
75           {
76               Serial.println("++Data tdk sesuai++");
77               delay(100);
78               software_Reset();
79           }
80
81           if (i == 3 && baris[3] == B00 && baris[1] >= 97 && baris[1] <=
82 104) //merah
83           {
84               Serial.println("++merah++");
85               kirimBalik();
86               EEPROM.write(129, 0);
87               panggilRambu();
88               i = 0;
89           } else if (i == 3 && baris[3] == B01 && baris[1] >= 97 &&
90 baris[1] <= 104) //kuning
91           {
92               Serial.println("++kuning++");
93               kirimBalik();
94               EEPROM.write(129, 1);
95               panggilRambu();
96               i = 0;
97           } else if (i == 3 && baris[3] == B10 && baris[1] >= 97 &&
98 baris[1] <= 104) //hijau
99           {
100               Serial.println("++hijau++");
101               kirimBalik();
102               EEPROM.write(129, 2);
103               panggilRambu();
104               i = 0;
105           }
106           else {
107               i++;
108           }
109       }
110       else if (i == 130) {
111           Serial.println("++Data Diterima++");
112           kirimBalik();
113           delay(300);
114           software_Reset();
115       }
116       else
117       {
118           tampilLampu();
119       }

```

```
}
}
```

Fungsi tambahan untuk rambu auto pada node lampu llintas

```
1 void panggilRambu()
2 {
3     if (baris[1] == 97)
4     {
5         rambu(97);
6     } else if (baris[1] == 98)
7     {
8         rambu(98);
9     } else if (baris[1] == 99)
10    {
11        rambu(99);
12    } else if (baris[1] == 100)
13    {
14        rambu(100);
15    } else if (baris[1] == 101)
16    {
17        rambu(101);
18    } else if (baris[1] == 102)
19    {
20        rambu(102);
21    } else if (baris[1] == 103)
22    {
23        rambu(103);
24    } else if (baris[1] == 104)
25    {
26        rambu(104);
27    }
28 }
29
30 void rambu(int inRambu)
31 {
32     int mati[129] = {135, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
33 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
34 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
35 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
36 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
37 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
38 0, 0, 0, 0};
39     int nyala[129] = {135, 255, 255, 255, 255, 255, 255, 255, 255,
40 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
41 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
42 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
43 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
44 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
45 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
46 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
47 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
48 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
49 255, 255, 255};
50     int rambuAtas[129] = {135, 0, 0, 0, 0, 0, 0, 0, 0, 0, 128, 192, 224,
51 240, 248, 252, 254, 255, 1, 3, 7, 15, 31, 63, 127, 255, 0, 0, 0, 0,
52 0, 0, 0, 0, 128, 192, 224, 240, 248, 252, 254, 127, 255, 255, 239,
```

```

53 231, 227, 225, 224, 224, 255, 255, 247, 231, 199, 135, 7, 7, 1, 3,
54 7, 15, 31, 63, 127, 254, 0, 0, 0, 0, 0, 0, 0, 0, 224, 224, 224,
55 224, 224, 224, 224, 224, 7, 7, 7, 7, 7, 7, 7, 7, 0, 0, 0, 0, 0,
56 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 224, 224, 224, 224, 224, 224, 224,
57 224, 7, 7, 7, 7, 7, 7, 7, 7, 0, 0, 0, 0, 0, 0, 0, 0};
58 int rambuKiri[129] = {135, 0, 0, 0, 0, 0, 0, 0, 0, 128, 192, 224,
59 240, 248, 252, 254, 127, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
60 0, 0, 128, 192, 224, 240, 248, 252, 254, 255, 63, 31, 15, 7, 3,
61 255, 255, 255, 0, 0, 0, 0, 0, 255, 255, 255, 0, 0, 0, 0, 255,
62 255, 255, 255, 254, 252, 248, 240, 224, 192, 128, 255, 255, 255, 3,
63 7, 15, 31, 63, 255, 255, 255, 0, 0, 0, 0, 0, 255, 255, 255, 0, 0,
64 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 127, 254, 252, 248, 240, 224, 192,
65 128, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
66 int rambuKanan[129] = {135, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
67 0, 0, 0, 0, 1, 3, 7, 15, 31, 63, 127, 254, 0, 0, 0, 0, 0, 0, 0,
68 0, 0, 0, 0, 0, 255, 255, 255, 0, 0, 0, 0, 255, 255, 255, 252,
69 248, 240, 224, 192, 255, 255, 255, 1, 3, 7, 15, 31, 63, 127, 255,
70 255, 255, 255, 0, 0, 0, 0, 0, 255, 255, 255, 0, 0, 0, 0, 255,
71 255, 255, 192, 224, 240, 248, 252, 255, 127, 63, 31, 15, 7, 3, 1,
72 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 254, 127, 63, 31,
73 15, 7, 3, 1, 0, 0, 0, 0, 0, 0, 0, 0};
74 int rambuPutar[129] = {135, 0, 0, 128, 192, 224, 240, 248, 248,
75 0, 254, 255, 255, 255, 15, 3, 1, 0, 31, 127, 255, 255, 252, 240,
76 224, 0, 0, 0, 0, 1, 3, 7, 7, 248, 248, 248, 248, 248, 248, 248,
77 248, 1, 1, 1, 1, 1, 1, 1, 1, 224, 224, 224, 224, 224, 224, 224,
78 224, 7, 7, 7, 7, 7, 7, 7, 7, 248, 248, 248, 248, 248, 248, 248,
79 248, 1, 1, 1, 1, 1, 1, 1, 1, 224, 224, 224, 224, 224, 224, 224,
80 224, 7, 7, 7, 7, 7, 7, 7, 7, 248, 248, 248, 248, 248, 248, 248,
81 248, 1, 1, 1, 1, 1, 1, 1, 1, 255, 254, 252, 248, 240, 224, 192,
82 128, 255, 127, 63, 31, 15, 7, 3, 1};
83 int rambuNyebrang[129] = {135, 0, 0, 0, 0, 0, 0, 0, 0, 224, 240,
84 240, 240, 240, 224, 252, 1, 3, 3, 3, 3, 3, 7, 31, 0, 0, 0, 0,
85 0, 0, 0, 0, 0, 0, 128, 192, 240, 254, 255, 124, 255, 255, 255, 255,
86 251, 249, 240, 224, 127, 255, 255, 255, 255, 63, 63, 63, 0, 1, 3,
87 15, 31, 31, 62, 62, 0, 0, 0, 0, 0, 0, 0, 128, 224, 240, 248, 252,
88 252, 126, 63, 31, 63, 63, 63, 127, 254, 252, 248, 240, 62, 60, 28,
89 12, 4, 1, 1, 3, 192, 224, 246, 255, 254, 124, 56, 16, 15, 7, 3, 1,
90 0, 0, 0, 0, 224, 192, 128, 0, 0, 0, 0, 0, 7, 15, 31, 63, 30, 15, 7,
91 2};
92 int rambuSepeda[129] = {135, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
93 4, 2, 1, 2, 0, 0, 0, 0, 0, 15, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
94 224, 24, 136, 132, 68, 4, 252, 6, 9, 87, 164, 72, 72, 136, 3, 5,
95 249, 25, 37, 70, 254, 5, 0, 0, 1, 6, 4, 8, 8, 8, 8, 24, 224, 0, 0,
96 0, 0, 0, 4, 6, 1, 0, 0, 0, 0, 0, 8, 24, 224, 0, 0, 0, 0, 0, 4, 6,
97 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
98 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
99
100 if (inRambu == 97)
101 {
102     for (int i = 1; i <= 128; i++)
103     {
104         EEPROM.write(i, mati[i]);
105     }
106     Serial.println("inRambu 1 mati");
107 } else if (inRambu == 98)
108 {
109     for (int i = 1; i <= 128; i++)

```

```

110     {
111         EEPROM.write(i, nyala[i]);
112     }
113     Serial.println("inRambu 1 nyala");
114 } else if (inRambu == 99)
115 {
116     for (int i = 1; i <= 128; i++)
117     {
118         EEPROM.write(i, rambuAtas[i]);
119     }
120     Serial.println("inRambu 1 rambuAtas");
121 } else if (inRambu == 100)
122 {
123     for (int i = 1; i <= 128; i++)
124     {
125         EEPROM.write(i, rambuKiri[i]);
126     }
127     Serial.println("inRambu 1 rambuKiri");
128 } else if (inRambu == 101)
129 {
130     for (int i = 1; i <= 128; i++)
131     {
132         EEPROM.write(i, rambuKanan[i]);
133     }
134     Serial.println("inRambu 1 rambuKanan");
135 } else if (inRambu == 102)
136 {
137     for (int i = 1; i <= 128; i++)
138     {
139         EEPROM.write(i, rambuPutar[i]);
140     }
141     Serial.println("inRambu 1 rambuPutar");
142 } else if (inRambu == 103)
143 {
144     for (int i = 1; i <= 128; i++)
145     {
146         EEPROM.write(i, rambuNyebrang[i]);
147     }
148     Serial.println("inRambu 1 rambuNyebrang");
149 } else if (inRambu == 104)
150 {
151     for (int i = 1; i <= 128; i++)
152     {
153         EEPROM.write(i, rambuSepeda[i]);
154     }
155     Serial.println("inRambu 1 rambuSepeda");
156 }
157 }

```

Fungsi tambahan untuk menampilkan pada lampu lalu lintas

```

1 void clocksMerah()
2 {
3     digitalWrite(clockMerah, HIGH);
4     delayMicroseconds(65);
5     digitalWrite(clockMerah, LOW);

```

```

6   }
7   void latchMerah()
8   {
9       digitalWrite(latchMerah, LOW);
10      delayMicroseconds(65);
11      digitalWrite(latchMerah, HIGH);
12  }
13  void clocksKuning()
14  {
15      digitalWrite(clockKuning, HIGH);
16      delayMicroseconds(65);
17      digitalWrite(clockKuning, LOW);
18  }
19  void latchKuning()
20  {
21      digitalWrite(latchKuning, LOW);
22      delayMicroseconds(65);
23      digitalWrite(latchKuning, HIGH);
24  }
25  void clocksHijau()
26  {
27      digitalWrite(clockHijau, HIGH);
28      delayMicroseconds(65);
29      digitalWrite(clockHijau, LOW);
30  }
31  void latchHijau()
32  {
33      digitalWrite(latchHijau, LOW);
34      delayMicroseconds(65);
35      digitalWrite(latchHijau, HIGH);
36  }
37  void tampilLampu()
38  {
39      if (baris[129] == B00) //merah
40      {
41          for (a = 1; a <= 128; a++)
42          {
43              PORTF = ~baris[a];
44              if (a == 128)
45                  digitalWrite(dataMerah, HIGH);
46              else
47                  digitalWrite(dataMerah, LOW);
48              clocksMerah();
49              latchMerah();
50          }
51      } else if (baris[129] == B01) //kuning
52      {
53          for (a = 1; a <= 128; a++)
54          {
55              PORTK = ~baris[a];
56              if (a == 128)
57                  digitalWrite(dataKuning, HIGH);
58              else
59                  digitalWrite(dataKuning, LOW);
60              clocksKuning();
61              latchKuning();
62          }

```

```
63 }else if (baris[129] == B10) //hijau
64 {
65     for (a = 1 ; a <= 128; a++)
66     {
67         PORTA = ~baris[a];
68         if (a == 128)
69             digitalWrite(dataHijau, HIGH);
70         else
71             digitalWrite(dataHijau, LOW);
72         clocksHijau();
73         latchsHijau();
74     }
75 }
76 }
```

UNIVERSITAS BRAWIJAYA

