

**IMPLEMENTASI *PROTOTYPE SYSTEM*
MONITORING AND CONTROLLING SMART HOME
MENGUNAKAN *EMBEDDED SYSTEM***

SKRIPSI

Laboratorium Sistem Komputer Dan Robotika

Untuk memenuhi sebagian persyaratan untuk mencapai gelar Sarjana Komputer



Disusun oleh :

HIMAWAN ADITYA PERDANA

NIM. 0910683051

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS BRAWIJAYA

PROGRAM TEKNOLOGI INFORMASI & ILMU KOMPUTER

PROGRAM STUDI TEKNIK INFORMATIKA

MALANG, 2014

LEMBAR PERSETUJUAN

Implementasi Prototype System Monitoring and Controlling Smart Home Menggunakan Embedded system

SKRIPSI

Untuk memenuhi sebagian persyaratan untuk mencapai gelar Sarjana Komputer



Disusun Oleh :

Himawan Aditya Perdana

0910683051

Skripsi ini telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Gembong Edhi Setyawan, S.T., M.T.
NIK. 761201 16 1 1 0373

Eko Setiawan, ST., M.Eng.
NIK. 870610 06 1 1 0256

LEMBAR PENGESAHAN

**IMPLEMENTASI PROTOTYPE SYSTEM MONITORING AND CONTROLLING
SMART HOME MENGGUNAKAN EMBEDDED SYSTEM**

Disusun oleh :

**HIMAWAN ADITYA PERDANA
NIM. 0910683051**

**Skripsi ini diuji dan dinyatakan lulus pada
Tanggal 4 Juli 2014**

Penguji I

Agung Setia Budi, S.T., M.T.
NIK

Penguji II

Wijaya Kurniawan, S.T., M.T.
NIK 820125 16 1 1 0481

Penguji III

Hurriyatul Fitriyah, S.T., M.Sc.
NIK.

**Mengetahui,
Ketua Program Studi Informatika/Ilmu Komputer**

Drs. Marji, M.T.
NIP. 19670801 199203 1 001

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas segala limpahan rahmat, hidayah dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Implementasi *Prototype System Monitoring and Controlling Smart Home Menggunakan Embedded system*”. Tugas akhir ini diajukan sebagai salah satu persyaratan kelulusan dalam menyelesaikan studi di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya, Malang.

Melalui kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih penulis yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan baik lahir maupun batin selama penulisan tugas akhir ini. Oleh karena itu, pada kesempatan ini penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Allah S.W.T atas semua karunia-Nya, rahmat-Nya serta anugrah-Nya hingga penulis dapat menyelesaikan skripsi dengan baik.
2. Kedua orang tua penulis, Ayahanda Sumadi dan Ibunda Lilik Puryanti yang selalu tidak lepas dari doa dan harapan untuk terselesaikannya skripsi ini dan terus memberikan dorongan moral, material dan kasih sayangnya tiada akhir.
3. Adik penulis Karina Noviyanti, dan Salsabila Khoirunisa Putri yang senantiasa memberi doa dan motivasi, serta keluarga yang selalu memberikan dukungan moral dan doa.
4. Bapak Drs. Marji, MT. dan Bapak Issa Arwani S.Kom, M.Sc. selaku Ketua Program dan Sekretaris Program Informatika dan Ilmu Komputer, serta segenap Bapak/Ibu Dosen dan seluruh Staff Karyawan PTIIK Universitas Brawijaya.
5. Bapak Gembong Edhi Setyawan ST., M.T. selaku Dosen Pembimbing I dan Bapak Eko Setiawan ST., M.Eng. selaku Dosen Pembimbing II yang telah memberikan bimbingan dan arahan untuk kesempurnaan penulisan skripsi ini.

6. Mas Didit selaku laboran Laboratorium Siskombot yang selalu bersedia meminjamkan lab dan alat untuk melaksanakan percobaan dan pengujian.
7. Teman-teman TPL angkatan 2009: Yusuf Oktofani S.Kom., Yuri Citra Pratama S.Kom., Dhimas Sawung Pamungkas S.Kom., Aрга Suwastika S.Kom., Iqbal Luthfillah S.Kom, Hafidz Rozaq S.Kom., Nabila Mahastika Priadana S.Kom., Darmawan Lahru Riatma S.Kom., Hanas Subakti S.Kom., Alan Nur Abdan S.Kom., Gerry Wisudawan S.Kom., dan teman-teman lainnya yang tidak bisa disebutkan satu persatu. Terimakasih telah memberikan semangat, dukungan, dan masukan selama pengerjaan skripsi.
8. Teman-teman Kos Mbetek: Zunan Ahsan Algony S.T., Lauma Yudha S.T., Galih Satya S.H., Doni Sukmawan S.E., Gilang S.H., Lauma Dwi Sistyawan S.E., Adi Wahyu Purwanto S.Ikom, Erwin Tidar Pratama, Yana Adi Prakosa, Yanuar Riski Hanani, Dicky Arinta, Rizki Fauzan Amin.
9. Teman-teman Teknik Informatika PTIIK Universitas Brawijaya angkatan 2008, 2010, 2011 atas segala bantuannya selama menempuh studi di Program Teknologi Informasi dan Ilmu Universitas Brawijaya.
10. Dan semua pihak yang tidak dapat disebutkan satu persatu yang telah mendukung dan membantu dalam penyelesaian skripsi ini. Penulis berharap, semoga skripsi ini dapat memberikan manfaat bagi pembaca serta dapat menjadi bahan acuan untuk penelitian lebih lanjut.

Malang, 11 Juni 2014

Penulis

Himawan Aditya Perdana. 2014. Implementasi *Prototype System Monitoring and Controlling Smart Home* Menggunakan *Embedded system*. Skripsi Program Studi Teknik Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya, Malang. Pembimbing: Gembong Edhi Setyawan ST., M.T. dan Eko Setiawan ST., M.Eng.

ABSTRAK

Tingginya aktifitas yang dilakukan seseorang terkadang membuat lupa pada kondisi rumah seperti menyalakan atau mematikan lampu, mematikan televisi ketika meninggalkan rumah. Dibutuhkan sebuah sistem kendali secara otomatis yang bisa dilakukan dimanapun dan kapanpun. Pada Penelitian ini dilakukan perancangan dan implementasi prototype sebuah system smart home dengan memanfaatkan embedded system dan sebuah *website* sebagai tempat monitoring dan controlling. Monitoring dan controlling yang dilakukan adalah melihat kondisi apakah peralatan elektronik dalam kondisi menyala atau mati.

Pada sistem monitoring dan controlling ini menggunakan 3 buah xbee yang dapat saling berkomunikasi untuk mengirimkan data perintah yang berasal dari *website*. Masing – masing xbee diletakkan pada beberapa titik yang saling berjauhan. Dari hasil beberapa pengujian didapatkan hasil bahwa kemampuan xbee berkomunikasi didalam ruangan dengan halangan cukup baik, jarak yang dapat dijangkau dalam pengiriman data antara xbee saat berada di rumah yaitu 20 meter. Penambahan 1 buah node diantara 2 buah node yang berjauhan dapat membantu komunikasi data yang putus. Waktu periode yang baik untuk pengiriman data pada sistem monitoring dan controlling smart home ini adalah 2 detik.

Kata kunci : *Smart home, Embedded system, Xbee*

Himawan Aditya Perdana. 2014. Implementasi Prototype System Monitoring and Controlling Smart Home Menggunakan Embedded system. Undergraduate Thesis of Informatic Study Program, Information Technology and Computer Science Program, Brawijaya University, Malang. Advisor: Gembong Edhi Setyawan ST., M.T. and Eko Setiawan ST., M.Eng.

ABSTRACT

The high activities of a person sometimes make them forget the condition of their home such as turning on/off the lights or turn off the television when leaving the house. To solve this matter, it takes an automatic control system that can take control anywhere and anytime. This research was carried out on the design and implementation of a smart home system prototype that can be done any where and any time by using embedded systems and a website as a monitoring and controlling tool. Monitoring and controlling was done to look at the condition of electronic equipments whether they are switched-on or off.

The proposed monitoring and controlling system is using 3 xbee that can communicate with each other to transmit data from website orders. Each xbee was placed on some far points. The result of experiment showed that xbee could communicate well on 20 meters indoor distance. The addition of 1 node between 2 nodes can help to relay the break up data communication. The best time period for delivery is 2 second.

Key word : Smart home, Embedded system, xbee

DAFTAR ISI

KATA PENGANTAR	i
ABSTRAK	iii
ABTRACT	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR	viii
DAFTAR TABEL.....	x
DAFTAR LAMPIRAN.....	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Kajian pustaka.....	5
2.2 <i>Embedded System</i>	6
2.3 <i>Website</i>	7
2.4 <i>Web Server</i>	8
2.5 PHP	9
2.6 Arduino Uno	10
2.7 Xbee	10
2.9 Processing	13
BAB III METODOLOGI PENELITIAN DAN PERANCANGAN.....	15
3.1 Metodologi Penelitian	15
3.1.1 Studi literatur.....	16
3.1.2 Analisis Kebutuhan	16
3.1.3 Perancangan	16

3.1.4 Implementasi	16
3.1.5 Pengujian dan Analisis.....	17
3.1.6 Kesimpulan dan Saran	17
3.2 Perancangan	18
3.2.1 Perancangan Perangkat Keras	18
3.2.2 Perancangan Perangkat Lunak	21
3.2.2.1 Perancangan Program Arduino	21
3.2.2.2 Perancangan Website.....	24
3.2.2.3 Perancangan program processing.....	26
3.2.2.4 Perancangan Protokol	27
BAB IV IMPLEMENTASI	29
4.1 Batasan Impelementasi	29
4.2 Implementasi Perangkat keras.....	30
4.3 Konfigurasi Xbee	34
4.4 Implementasi Program Arduino	39
4.4.1 Implementasi Program Arduino Coordinator	39
4.4.2 Implementasi Program Arduino Router satu	40
4.4.3 Implementasi Program Arduino Router dua	42
4.5 Implementasi Program Processing.....	42
4.6 Implementasi Program web.....	43
BAB V PENGUJIAN DAN ANALISIS.....	46
5.1 Pengujian Range area.....	46
5.2 Pengujian Komunikasi data.....	48
5.3 Pengujian Pengiriman Data.....	50
5.3.1 Pengujian Pengiriman Langsung	50
5.3.2 Pengujian Pengiriman Tidak Langsung	53
5.3.3 Pengujian Nilai Periode Ideal.....	56
5.4 Pengujian sistem secara keseluruhan	60
BAB VI PENUTUP	64
6.1 Kesimpulan	64
6.2 Saran.....	64



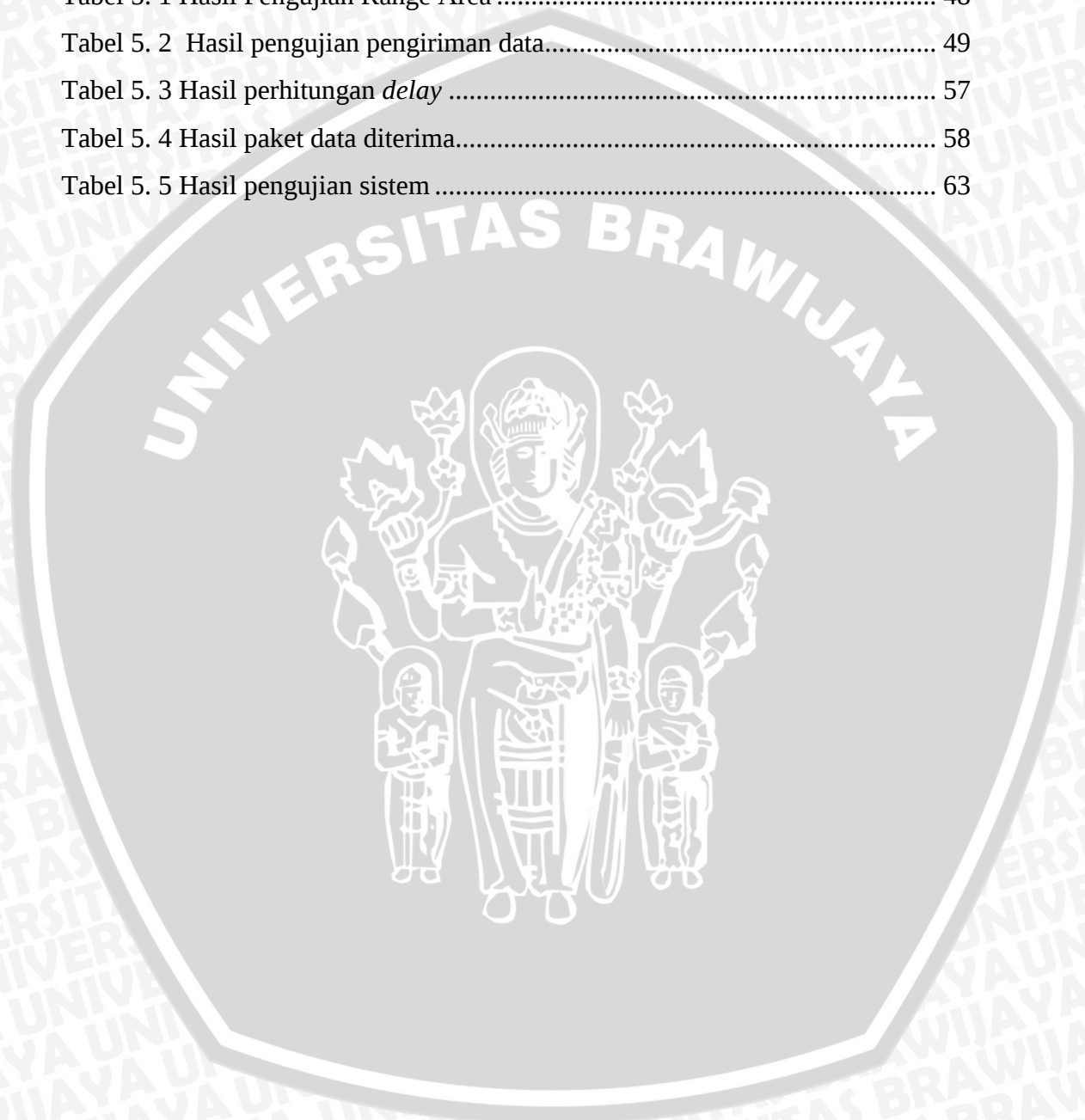
DAFTAR GAMBAR

Gambar 2. 1 Susunan Standar Embedded System	7
Gambar 2. 2 Struktur php.....	9
Gambar 2. 3 Arduino Uno.....	10
Gambar 2. 4 Komunikasi Serial Xbee.....	11
Gambar 2. 5 Xbee shield.....	12
Gambar 2. 6 Tampilan Processing	14
Gambar 3. 1 Diagram alir metodologi penelitian.....	15
Gambar 3. 2 Perancangan arduino koordinator.....	19
Gambar 3. 3 Perancangan arduino router satu dan arduino router dua.....	19
Gambar 3. 4 Skema topology	20
Gambar 3. 5 Flowchart perancangan sistem pada koordinator	22
Gambar 3. 6 Diagram Perancangan Sistem pada router satu	23
Gambar 3. 7 Flowchart sistem pada arduino router dua	24
Gambar 3. 8 Flowchart perancangan <i>website</i>	25
Gambar 3. 9 Tampilan web	26
Gambar 3. 10 Diagram alir processing	27
Gambar 3. 11 Rancangan Protokol awal.....	28
Gambar 3. 12 Rancangan Protokol akhir	28
Gambar 4. 1 Diagram Bab Implementasi.....	29
Gambar 4. 2 Perangkat coordinator	30
Gambar 4. 3 Pemasangan Xbee pro,xbee shield dan arduino uno	31
Gambar 4. 4 Perangkat Router	32
Gambar 4. 5 Hasil implementasi.....	33
Gambar 4. 6 Hasil Implementasi.....	33
Gambar 4. 7 Konfigurasi dalam mode reset.....	34
Gambar 4. 8 Tampilan awal X-CTU.....	35
Gambar 4. 9 Tampilan Kotak dialog Test/Query sukses	36
Gambar 4. 10 Tampilan kotak dialog test/query gagal	36
Gambar 4. 11 Menu konfigurasi pada X-CTU.....	37
Gambar 4. 12 Setting konfigurasi Xbee selesai	37

Gambar 4. 13 Setting Pan id coordinator dan router.....	38
Gambar 4. 14 Potongan <i>Source code</i> arduino coordinator	39
Gambar 4. 15 Potongan <i>Source code</i> arduino coordinator	40
Gambar 4. 16 Potongan <i>Source code</i> Arduino coordinator	40
Gambar 4. 17 Potongan <i>source code</i> arduino router satu	40
Gambar 4. 18 Potongan <i>source code</i> arduino router satu	41
Gambar 4. 19 Potongan <i>Source code</i> arduino dua.	42
Gambar 4. 20 Potongan <i>Source code</i> Program processing	43
Gambar 4. 21 Potongan <i>Source code</i> Program Processing.....	43
Gambar 4. 22 Tampilan <i>Website</i>	44
Gambar 4. 23 Potongan <i>Source code website</i>	44
Gambar 5. 1 Denah lokasi pengujian range area.....	46
Gambar 5. 2 Tampilan Data Sinyal.....	47
Gambar 5. 3 Hasil pengiriman pada X-CTU	49
Gambar 5. 4 Denah lokasi pengujian pengiriman data	50
Gambar 5. 5 Tampilan <i>docklight</i>	51
Gambar 5. 6 Code pengujian <i>delay</i>	51
Gambar 5. 7 Tampilan pada sisi pengiriman	52
Gambar 5. 8 Tampilan pada sisi penerima.....	52
Gambar 5. 9 Hasil Pengiriman data pada titik K - B	53
Gambar 5. 10 Skema pengujian pengiriman tidak langsung.....	54
Gambar 5. 11 Hasil pengiriman dari titik K menuju ke titik B1 tanpa titik A.....	55
Gambar 5. 12 Hasil pengujian pengiriman tidak langsung.....	55
Gambar 5. 13 Hasil data akhir.....	57
Gambar 5. 14 Grafik hasil perhitungan <i>delay</i>	58
Gambar 5. 15 Tampilan menu utama	61
Gambar 5. 16 LED pada arduino router dua menyala	62
Gambar 5. 17 LED pada arduino router dua menyala	62

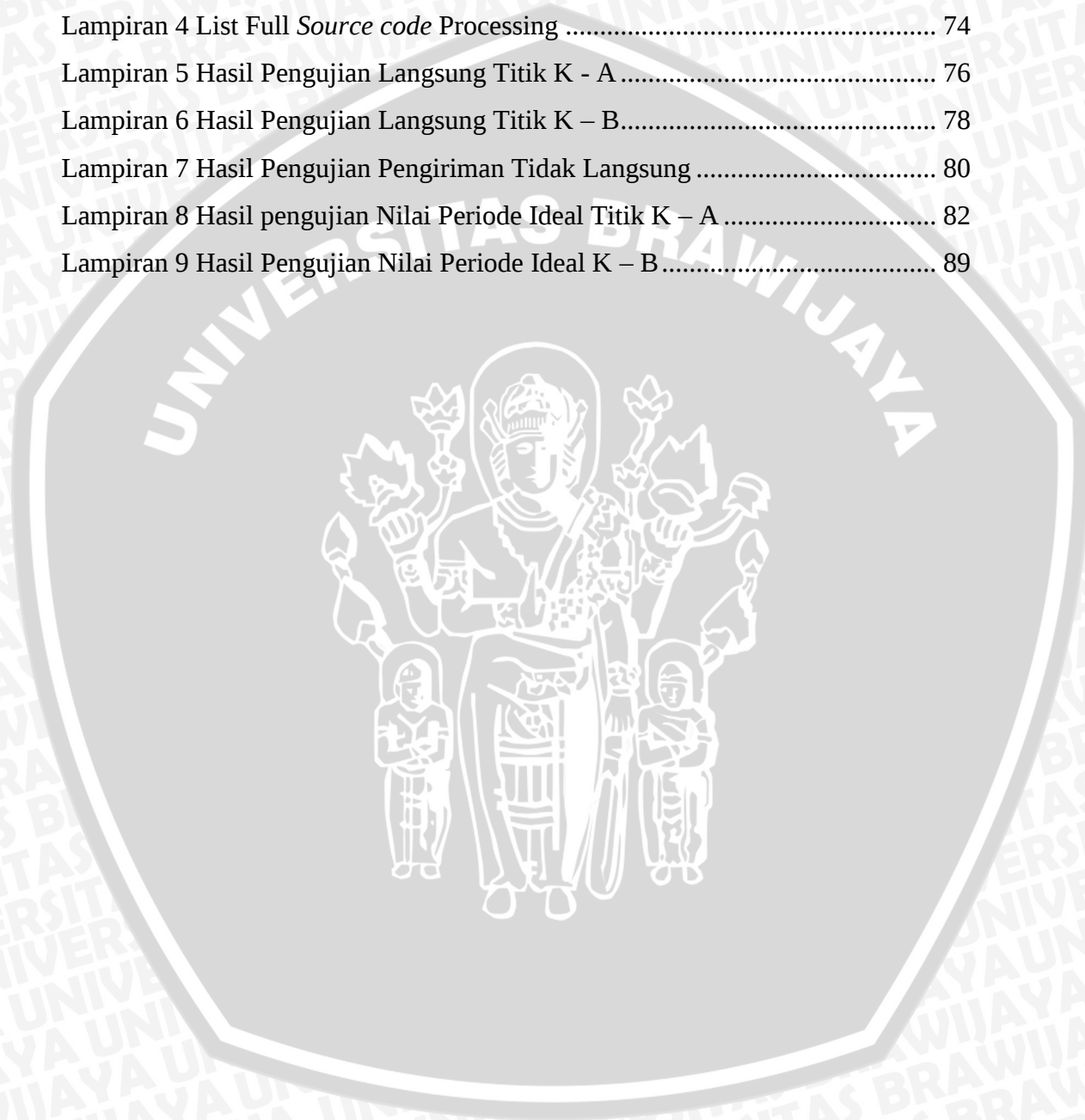
DAFTAR TABEL

Tabel 2. 1 Spesifikasi Xbee dan Xbee proo	11
Tabel 2. 2 Tabel koneksi pin arduino-xbee.....	13
Tabel 5. 1 Hasil Pengujian Range Area	48
Tabel 5. 2 Hasil pengujian pengiriman data.....	49
Tabel 5. 3 Hasil perhitungan <i>delay</i>	57
Tabel 5. 4 Hasil paket data diterima.....	58
Tabel 5. 5 Hasil pengujian sistem	63



DAFTAR LAMPIRAN

Lampiran 1 List Full Coding Arduino Coordinator	67
Lampiran 2 List Full <i>Source code</i> Arduino Router satu	70
Lampiran 3 List Full <i>Source code</i> Arduino Router dua	72
Lampiran 4 List Full <i>Source code</i> Processing	74
Lampiran 5 Hasil Pengujian Langsung Titik K - A	76
Lampiran 6 Hasil Pengujian Langsung Titik K – B.....	78
Lampiran 7 Hasil Pengujian Pengiriman Tidak Langsung	80
Lampiran 8 Hasil pengujian Nilai Periode Ideal Titik K – A	82
Lampiran 9 Hasil Pengujian Nilai Periode Ideal K – B	89



BAB I

PENDAHULUAN

1.1 Latar Belakang

Belakangan ini kehidupan manusia di jaman modern semakin padat, banyak aktifitas yang dilakukan di luar rumah seperti bekerja maupun melakukan perjalanan. Tingginya aktifitas ini terkadang membuat orang lupa dengan kondisi rumah seperti menyalakan atau mematikan lampu, mematikan televisi ketika meninggalkan rumah. Akibatnya menimbulkan dampak kerugian yang menyebabkan terjadi pembengkakan biaya tagihan listrik oleh sebab itu dibutuhkan sebuah sistem kendali secara otomatis yang bisa dilakukan dimanapun dan kapanpun.

Controlling merupakan pengaturan ataupun pengendalian terhadap satu atau beberapa besaran (*variabel,parameter*) sehingga benda pada suatu harga atau suatu rangkuman (*range*) tertentu [FZN - 2013]. *Monitoring* sendiri merupakan sebuah kegiatan ingin mengetahui apa saja yang ada didalamnya sehingga dapat diambil sebuah informasi. *Monitoring* ruangan yakni ingin mengetahui keadaan dalam suatu ruangan tersebut meliputi fasilitas ruangan di dalamnya, fasilitas didalamnya meliputi lampu, AC, pintu dan lainnya [NFD - 2011]. Dengan adanya sistem *monitoring dan controlling* ini merupakan sebuah solusi untuk mengatasi sistem manualisasi yang kurang efisien.

Monitoring dan controlling yang dilakukan adalah untuk mengetahui bagaimana kondisi peralatan elektronik di rumah ketika ditinggalkan. Dalam melakukan *Monitoring dan controlling* nantinya akan digunakan *emdedded system* yaitu arduino sebagai alat penghubung dan pengontrol antara lampu dirumah dengan server pusat.

Pada penelitian sebelumnya telah membahas tentang sistem *smart home* salah satunya adalah aplikasi *smart home* yang memanfaatkan RFID sebagai *controlling*, sistem ini bekerja dengan cara mendekatkan kartu RFID pada sensor yang terdapat pada penguncian pintu rumah [ALN - 2013]. Pada

penelitian lainnya sistem *monitoring* dan *controlling* dibuat dengan memanfaatkan aplikasi desktop berbasis *Delphi 7.0* [ALI -2010] yang masih memiliki kelemahan untuk *monitoring* dan *controlling* hanya pada laptop tertentu. Dengan adanya permasalahan tersebut maka penulis mengusulkan solusi dengan melakukan penelitian berjudul “**Implementasi Prototype System Monitoring and Controlling Smart Home Menggunakan Embedded system**” yang nantinya aktifitas *monitoring* dan *controlling* akan dilakukan melalui sebuah *website* yang terintegrasi dengan *arduino* sehingga dapat diakses di manapun dan kapanpun dengan memanfaatkan jaringan internet. Monitoring yang dilakukan disini hanya dapat melihat keadaan dari website. Selain itu juga ditambahkan satu buah node yang diletakan antara node pengirim dan node penerima yang terletak berjauhan. Penambahan node ini berfungsi sebagai perantara pengiriman data saat node tujuan pengiriman data berada diluar jangkauan. Sistem ini akan memberikan informasi yang ditampilkan lewat *web server* yang menunjukkan kondisi peralatan elektronik dirumah sedang menyala ataupun mati.

1.2 Rumusan Masalah

Berdasarkan Latar belakang diatas maka rumusan masalahnya adalah sebagai berikut :

1. Bagaimana merancang dan mengimplementasi sistem *monitoring dan controlling* smart home menggunakan *embedded system* ?
2. Bagaimana hasil kinerja sistem jika terdapat penambahan 1 node diantara 2 node yang letaknya saling berjauhan?
3. Berapa waktu periode pengiriman yang baik dalam sistem *monitoring dan controlling* smart home ini?

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini antara lain :

1. Implementasi hanya sebatas prototype.

2. Perangkat embbeded yang digunakan adalah Arduino uno
3. Perangkat pengganti lampu dan alat elektronik lainnya adalah *light emitting diode* (LED)
4. Penelitian menggunakan sistem operasi windows dan *software* arduino
5. Penelitian sistem ini tidak memperhatikan aspek keamanan jaringan

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah merancang dan membuat sistem prototype *monitoring* dan *controlling* pada rumah dengan memanfaatkan *embedded system* yang dapat di pantau dimanapun melalui web server .

1.5 Manfaat Penelitian

a. Bagi penulis :

1. Menerapkan ilmu yang telah didapatkan dari Teknik Informatika Universitas Brawijaya.
2. Mengerti dan memahami tentang perancangan, penerapan dan analisa tentang *monitoring* dan *controlling* menggunakan *embedded system*.

b. Bagi Pengguna :

1. Dengan adanya system *monitoring* dan *controlling* menggunakan *embedded system* diharapkan dapat mempermudah pengguna dalam mengetahui informasi tetang kondisi rumah
2. Mendapatkan informasi tentang implementasi prototype implementasi system *monitoring* dan *controlling* pada *embedded system*

1.6 Sistematika Penulisan

Untuk mencapai tujuan yang diharapkan, maka sistematika penulisan yang disusun dalam tugas akhir ini adalah sebagai berikut:

BAB I Pendahuluan

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II Kajian Pustaka dan Dasar Teori

Bab ini membahas tentang penelitian terdahulu dan teori-teori yang mendukung dalam. Implementasi *Prototype System Monitoring and Controlling Smart Home* Menggunakan *Embedded system*.

BAB III Metodologi Penelitian dan Perancangan

Bab ini membahas tentang metode yang digunakan dalam penulisan yang terdiri dari studi literatur, analisis kebutuhan dan Implementasi *Prototype System Monitoring and Controlling Smart Home* Menggunakan *Embedded system* serta urutan pengujian sistem.

BAB IV Implementasi

Bab ini membahas tentang hasil perancangan dari analisis kebutuhan. Bab ini juga membahas Implementasi *Prototype System Monitoring and Controlling Smart Home* Menggunakan *Embedded system*

BAB V Pengujian dan Analisis

Bab ini memuat tentang hasil pengujian dan analisis terhadap sistem yang telah diimplementasikan.

BAB VI Penutup

Bab ini memuat kesimpulan yang diperoleh dari pembuatan dan pengujian yang dikembangkan dalam skripsi ini serta saran – saran untuk pengembangan lebih lanjut.

BAB II

TINJAUAN PUSTAKA

Bab Tinjauan Pustaka membahas tentang kajian pustaka dan dasar teori untuk menunjang penulisan skripsi tentang Implementasi *Prototype System Monitoring and Controlling Smart Home Menggunakan Embedded System*. Pada penelitian ini dasar teori yang dibutuhkan untuk menunjang penulisan meliputi Arduino, Xbee, dan Web.

2.1 Kajian pustaka

Telah ada sebelumnya penelitian terdahulu mengenai smart home (rumah pintar) atau bisa disebut dengan otomatisasi perangkat yang berada di rumah. Pada penelitian tersebut dibahas bagaimana tentang perancangan aplikasi yang dapat mengontrol lampu dan *air conditioner* (AC) pada ruangan agar digunakan sesuai kebutuhan. Dalam dokumen tersebut penulis merancang aplikasi sistem kontroling menggunakan aplikasi desktop dan menggunakan *database* sebagai tempat penyimpanan data waktu [ALI- 2010]

Penelitian lainnya dibahas bagaimana *controlling* dan *monitoring* lampu pada ruangan melalui aplikasi desktop dan data waktu yang disimpan pada database. Monitoring yang dilakukan yaitu melihat apakah lampu sudah menyala sesuai dengan jadwal yang sudah ditetapkan. Mikrokontroler yang digunakan adalah AVR ATmega 8535 sebagai master yang terhubung dengan laptop dimana sudah terinstall aplikasi desktop namun masih memiliki kekurangan dimana aplikasi controlling dan monitoring hanya bisa dilakukan dari laptop yang sudah terinstall aplikasi desktop tersebut. [JLI - 2012].

Ada juga penelitian tentang *smart home* dengan memanfaatkan teknologi RFID sebagai alat controlling. Kartu RFID didekatkan pada sensor yang telah dirangkai sehingga dapat mengirimkan perintah kepada dari slave menuju master server. Pada penelitian ini juga memanfaatkan dua buah node sebagai master dan penerima [ALN - 2013]

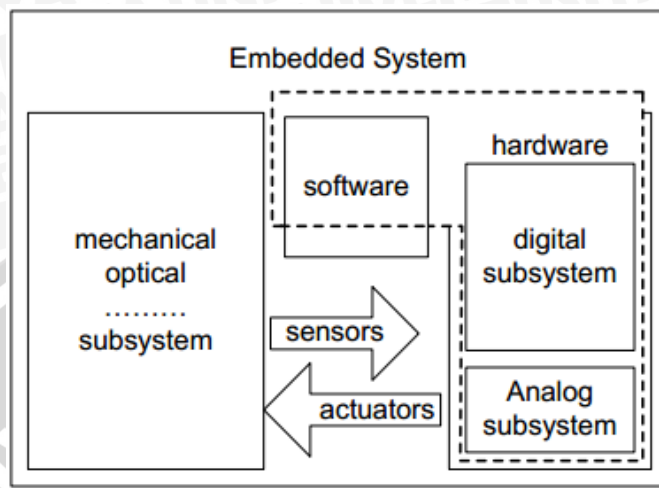
Pada Penelitian “Implementasi *Prototype System Monitoring and Controlling Smart Home* Menggunakan *Embedded System*”. Penulis akan melakukan kontroling dan monitoring menggunakan mikrokontroller Arduino dan aplikasi berbasis *website* dimana nantinya kontroling dapat dilakukan dengan mudah dengan mengakses *website* yang sudah. Untuk monitoring sendiri penggunaan nantinya akan dapat melihat kondisi peralatan elektronik melalui *website*. Selain itu juga sistem dirancang menggunakan tiga node berbeda yang terdiri dari arduino coordinaor yang bertindak sebagai pusat pengiriman perintah dari *webserver*. Arduino koordinator juga mampu mendeteksi arduino *router* baru yang telah dikonfigurasi sebelumnya sehingga dapat bergabung pada topology yang ada. Sedangkan pada arduino *router* satu dan arduino *router* dua bertindak sebagai tempat eksekusi perintah yang dikirim dari arduino *coordinator*. Pada arduino *router* satu juga dapat melakukan pengiriman perintah yang berasal dari arduino koorinator menuju ke arduino *router* dua. Fungsi ini berlaku ketika penempatan arduino koordinator dan arduino *router* dua berjauhan yang menyebabkan data gagal diterima sehingga diletakan arduino *router* satu yang berada di antara arduino koordinator dan arduino *router* dua untuk meneruskan perintah yang berasal dari arduino koordinator sehingga data dapat diterima oleh arduino *router* dua. Skema ini merupakan perbaikan dari penelitian sebelumnya, dimana pada penelitian tersebut hanya menggunakan dua buah node sehingga ketika kedua node terletak berjauhan tidak dapat saling berkomunikasi [ALN - 2013] .

2.2 *Embedded System*

Embedded System adalah sebuah sistem komputer yang dirancang dan dibangun untuk melakukan suatu tujuan atau fungsi tertentu. Sistem tertanam ini memiliki sebuah peran dalam mengatur dan menentukan perilaku sebuah sistem. Pada umumnya sistem tertanam cenderung didesain untuk perangkat tertentu sehingga memiliki sebuah kemampuan spesifik yang jelas, berbeda dengan operating pada PC yang termasuk dalam multi porpose. [ZHT - 2013] .

Sistem tertanam dirancang untuk suatu pekerjaan khusus yang dapat dikerjakan secara otomatis sehingga dapat dijadikan suatu sistem kontrol. Adanya sistem tertanam dapat membantu suatu sistem untuk bekerja secara otomatis dan

bebiaya murah dibanding dengan menggunakan komputer, karena sistem tertanam memiliki ukuran lebih kecil. [ABD - 2012].



Gambar 2. 1Susunan Standar Embedded System

Sumber:[ABD - 2012]

2.3 Website

Website atau situs juga dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data Gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*). Bersifat statis apabila isi informasi *website* tetap, jarang berubah, dan isi informasinya searah hanya dari pemilik *website*. Bersifat dinamis apabila isi informasi *website* selalu berubah-ubah, dan isi informasinya interaktif dua arah berasal dari pemilik serta pengguna *website*. Contoh *website* statis adalah berisi profil perusahaan, sedangkan *website* dinamis adalah seperti Friendster, Multiply, dll. Dalam sisi pengembangannya, *website* statis hanya bisa diupdate oleh pemiliknya saja, sedangkan *website* dinamis bisa diupdate oleh pengguna maupun pemilik [DEE - 2008]. Dalam pengaplikasiannya *website* membutuhkan sebuah tempat untuk menyimpan data atau halaman *website* itu sendiri. Tempat penyimpanan itu sendiri biasa disebut dengan *webserver*.

2.4 Web Server

Web server merupakan software yang memberikan layanan data yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan browser web dan mengirimkan kembali hasilnya dalam bentuk halaman - halaman web yang umumnya berbentuk dokumen HTML [WRL-12].

Macam - macam *Web Server* diantaranya:

1. Apache *Web Server* - The HTTP Web Server
2. Apache Tomcat
3. Microsoft windows Server 2003 Internet Information Services (IIS)
4. Lighttpd

Cara kerja *Web server* merupakan mesin dimana tempat aplikasi atau software beroperasi dalam medistribusikan web page ke user, tentu saja sesuai dengan permintaan user. Hubungan antara Web Server dan Browser Internet merupakan gabungan atau jaringan Komputer yg ada di seluruh dunia. Setelah terhubung secara fisik, *Protocol TCP/IP (networking protocol)* yg memungkinkan semua komputer dapat berkomunikasi satu dengan yg lainnya.

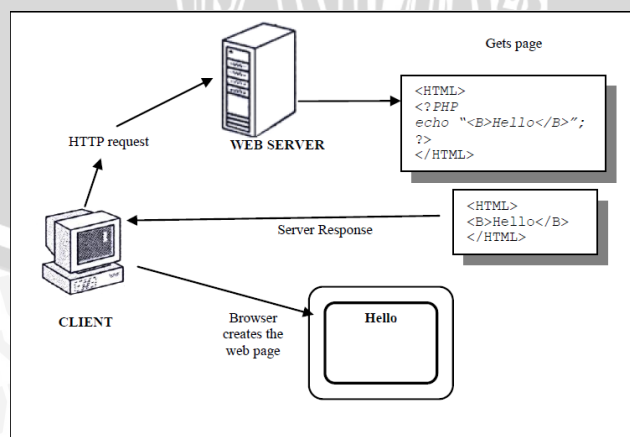
Pada saat browser meminta data web page ke server maka instruksi permintaan data oleh browser tersebut dikemas di dalam TCP yg merupakan *protocol* transport dan dikirim ke alamat yg dalam hal ini merupakan *protocol* berikutnya yaitu Hyper Text Transfer *Protocol* (HTTP). HTTP ini merupakan *protocol* yg digunakan dalam World Wide Web (WWW) antar komputer yg terhubung dalam jaringan di dunia ini. Data yg di passing dari browser ke Web server disebut sebagai HTTP request yg meminta web page dan kemudian web server akan mencari data HTML yg ada dan dikemas dalam TCP *protocol* dan di kirim kembali ke browser. Data yg dikirim dari server ke browser disebut sebagai HTTP response. Jika data yg diminta oleh browser tidak ditemukan oleh si Web server maka akan menimbulkan error yg sering dilihat di web page yaitu page yaitu Error : 404 Page Not Found.

2.5 PHP

PHP (PHP Hypertext Preprocessor) yang merupakan bahasa pemrograman berbasis web yang memiliki kemampuan untuk memproses data dinamis. PHP dikatakan sebagai sebuah server-side embedded script language artinya sintaks-sintaks dan perintah yang kita berikan akan sepenuhnya dijalankan oleh server tetapi disertakan pada halaman HTML biasa. Aplikasi-aplikasi yang dibangun oleh PHP pada umumnya akan memberikan hasil pada web browser, tetapi prosesnya secara keseluruhan dijalankan di server. [WKM-12]

Pada prinsipnya server akan bekerja apabila ada permintaan dari client. Dalam hal ini client menggunakan kode-kode PHP untuk mengirimkan permintaan ke server (dapat dilihat pada Gambar dibawah). Ketika menggunakan PHP sebagai server-side embedded script language maka server akan melakukan hal-hal sebagai berikut :

1. Membaca permintaan dari *client* / browser
2. Mencari halaman/page di server
3. Melakukan instruksi yang diberikan oleh PHP untuk melakukan modifikasi pada halaman/page.
4. Mengirim kembali halaman tersebut kepada client melalui internet atau intranet. [WPH - 2012]



Gambar 2. 2 Struktur php

Sumber : [WPH 2012]

2.6 Arduino Uno

Arduino merupakan sebuah *platform* yang bersifat *open source*. Di dalam arduino terdapat komponen utama yaitu sebuah chip mikrokontroler jenis AVR yang berasal dari perusahaan Atmel. Bahasa pemrograman yang pada arduino yaitu bahasa C yang memiliki sejumlah fungsi – fungsi yang mudah untuk dipelajari.[HSA - 2012].

Terdapat beberapa jenis arduino pada saat ini yaitu Arduino mega, Arduino Nano, Arduino ARM dan Arduino Uno. Arduino Uno merupakan papan mikrokontroler berbasis Atmega 328. Memiliki 14 pin digital input maupun output (dimana 6 diantaranya digunakan sebagai PWM output), 6 pin analog input, 16 MHZ *ceramic resonator*, *power jack* dan tombol reset. Arduino dapat dengan mudah disambungkan dengan pc, *battery* maupun adaptor listrik menggunakan kabel usb. Arduino uno berbeda dengan beberapa pendahulunya dimana tidak menggunakan FTDI USB to serial driver chip.



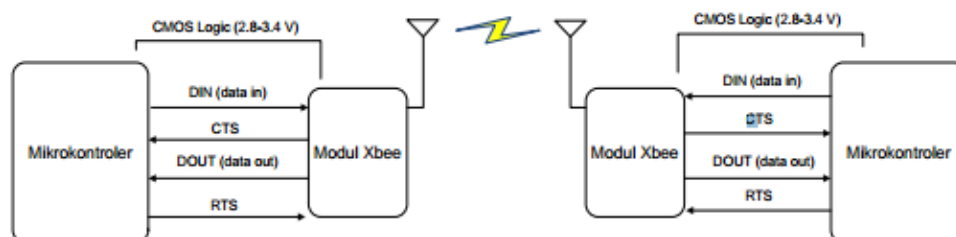
Gambar 2. 3 Arduino Uno

Sumber: [ARD 2013]

2.7 Xbee

Xbee merupakan sebuah modul yang memungkinkan arduino untuk berkomunikasi secara wireless menggunakan *protocol* zigbee yang beroperasi pada spesifikasi IEEE 802.15.4 radio fisik yang di munculkan oleh Digi International. Modul Xbee terdiri dari dua jenis yaitu xbee s2 dan xbee pro. Untuk xbee s2 memiliki jangkauan berkisar antara 20 – 30 meter untuk didalam ruangan dan

memiliki jangkauan 50 – 100 m pada luar ruangan, sedangkan untuk xbee pro memiliki jangkauan 70 – 90 m dalam ruangan dan 200 – 1500 m untuk di luar ruangan.



Gambar 2. 4 Komunikasi Serial Xbee

Sumber: [ASQ - 2012]

Xbee dapat bekerja dalam topology jaringan mesh, star dan tree. Untuk melakukan konfigurasi pada xbee digunakan software pendukung yaitu X-CTU. Agar dapat berkomunikasi antara satu xbee dengan yang lainnya perlu dilakukan konfigurasi beberapa parameter yang terdapat pada X-CTU seperti PAN ID (*Personal Area Network ID*) dan DL (*Destination Address Low*). Perbedaan karakteristik dan spesifikasi antara xbee dan xbee pro dapat dilihat pada Tabel 2.1

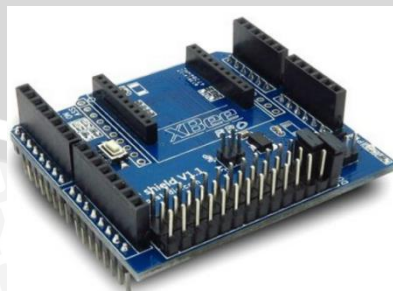
Tabel 2. 1 Spesifikasi Xbee dan Xbee pro



Specification	XBee	XBee-PRO
Performance		
Indoor/Urban Range	Up to 100 ft (30 m)	Up to 300 ft (90 m), up to 200 ft (60 m) International variant
Outdoor RF line-of-sight Range	Up to 300 ft (90 m)	Up to 1 mile (1600 m), up to 2500 ft (750 m) international variant
Transmit Power Output (software selectable)	1mW (0 dBm)	63mW (18dBm)* 10mW (10 dBm) for International variant
RF Data Rate	250,000 bps	250,000 bps
Serial Interface Data Rate (software selectable)	1200 bps - 250 kbps (non-standard baud rates also supported)	1200 bps - 250 kbps (non-standard baud rates also supported)
Receiver Sensitivity	-92 dBm (1% packet error rate)	-100 dBm (1% packet error rate)
Power Requirements		
Supply Voltage	2.8 – 3.4 V	2.8 – 3.4 V
Transmit Current (typical)	45mA (@ 3.3 V)	250mA (@3.3 V) (150mA for international variant) RPSMA module only: 340mA (@3.3 V) (180mA for international variant)
Idle / Receive Current (typical)	50mA (@ 3.3 V)	55mA (@ 3.3 V)
Power-down Current	< 10 μ A	< 10 μ A
General		
Operating Frequency	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960" x 1.297" (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip or U.FL Connector, RPSMA Connector	Integrated Whip, Chip or U.FL Connector, RPSMA Connector
Networking & Security		
Supported Network Topologies	Point-to-point, Point-to-multipoint & Peer-to-peer	
Number of Channels (software selectable)	16 Direct Sequence Channels	12 Direct Sequence Channels
Addressing Options	PAN ID, Channel and Addresses	PAN ID, Channel and Addresses

Sumber : [ZBB - 2012]

Pengkomunikasian Arduino dengan Xbee ada dua cara yaitu dengan melalui shield yang biasa disebut dengan Xbee shield. Xbee shield berupa papan yang memiliki kaki-kaki yang sama dengan Arduino yang berfungsi sebagai penghubung antara Xbee dengan Arduino. Shield ini dihubungkan langsung dengan papan Arduino atau USB.



Gambar 2. 5 Xbee shield

Sumber :[THR – 2013]

Cara yang kedua melalui kabel jumper. Pada arduino dan xbee dipasang kabel jumper pada pin yang tersedia. Kabel jumper digunakan jika xbee tidak terhubung langsung pada arduino seperti menggunakan xbee shield namun tetap dapat berfungsi sama seperti menggunakan shield. Pin yang disambungkan ditunjukkan pada Tabel 2.2

Tabel 2. 2Tabel koneksi pin arduino-xbee

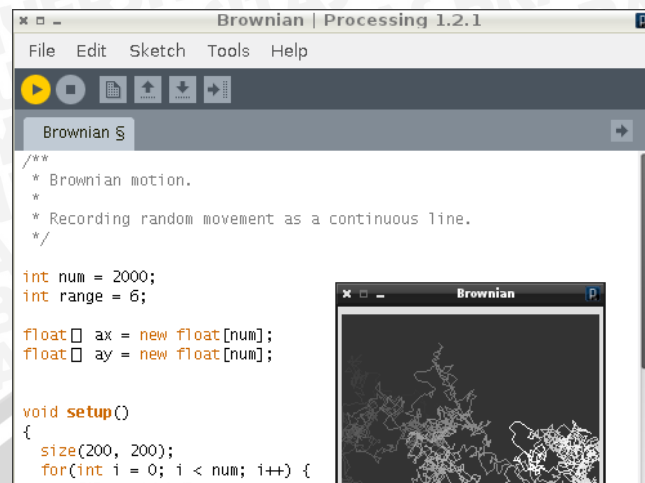
XBee	Arduino
VCC atau 3.3 V	3V3
TX atau DOUT	RX atau 0
RX atau DIN	TX atau 1
GND	GND

Sumber : [THR - 2013]

2.9 Processing

Processing merupakan sebuah bahasa pemrograman visual berbasis java yang bersifat *open source* dan mudah digunakan untuk pemula. Processing mulai diperkenalkan pada tahun 2001 oleh Casey Reas dan Benjamin Fry. Salah satu tujuan dari processing adalah memberikan sebuah alat untuk mereka yang bukan ahli di bidang programmer tetapi dapat mempelajari pemrograman visual dengan mudah.

Bahasa pemrograman yang digunakan oleh processing adalah java. Processing dapat digunakan sebagai software bahasa pembantu pada arduino seperti menstabilkan koneksi arduino dalam pembacaan atau penulisan pada data yang bersifat .txt dikarenakan processing support IDE. Tampilan processing mirip seperti tampilan pemrograman pada arduino seperti yang ditunjukkan pada Gambar 2.6



Gambar 2. 6 Tampilan Processing

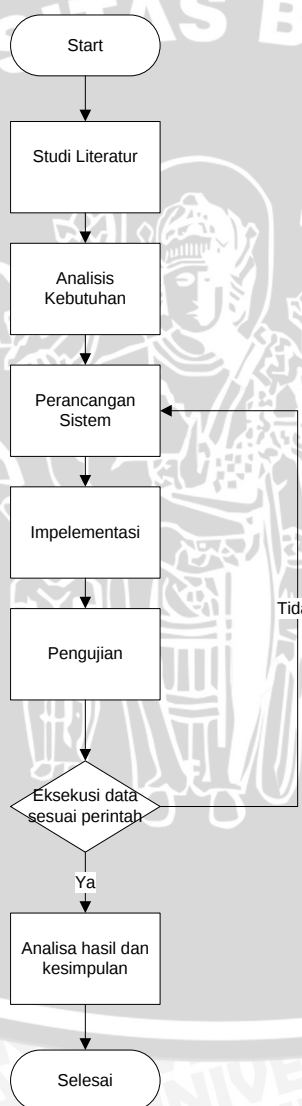


BAB III

METODOLOGI PENELITIAN DAN PERANCANGAN

3.1 Metodologi Penelitian

Pada bagian ini membahas tentang metodologi penelitian yang digunakan untuk skripsi ini dimana terdiri dari beberapa tahap yaitu studi literatur, analisa kebutuhan, perancangan sistem, implementasi, pengujian dan analisis dan pengambilan kesimpulan.



Gambar 3. 1 Diagram alir metodologi penelitian

3.1.1 Studi literatur

Studi literatur mempelajari mengenai penjelasan dasar teori yang digunakan untuk menunjang penulisan skripsi. Teori-teori pendukung tersebut diperoleh dari buku, artikel, jurnal, e-book dan dokumentasi proyek. Teori-teori pendukung tersebut meliputi tentang *Embedded system*, Arduino, Xbee, *Website*.

3.1.2 Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk mengetahui semua kebutuhan yang diperlukan oleh sistem yang akan dibangun. Analisis kebutuhan ini disesuaikan dengan kebutuhan yang peneliti gunakan, kebutuhan tersebut antara lain :

1. Pengiriman data / perintah dari *website* yang disimpan ke dalam notepad untuk diteruskan ke arduino master
2. Pencocokan perintah yang dikirim dari Arduino master ke Arduino penerima melalui xbee
3. Perintah yang diterima arduino penerima nantinya akan diolah yang kemudian masuk ke sistem sehingga terlihat perubahan pada objek yang di kontrol sesuai perintah master.
4. Peralatan yang digunakan antara lain Arduino, xbee, Laptop sebagai *webserver*. Alasan digunakan arduino dan xbee karena mudah dalam melakukan konfigurasi dan pemrograman.

3.1.3 Perancangan

Pada perancangan sistem penulis mulai merancang sebuah sistem yang dapat memenuhi semua kebutuhan fungsional aplikasi yang ada dalam tugas akhir. Teori - teori dan referensi yang didapat dijadikan satu dengan ilmu yang didapat serta diimplementasikan untuk merancang dan mengembangkan suatu sistem monitoring dan controlling smart home.

3.1.4 Implementasi

Implementasi sistem dilakukan dengan mengacu kepada perancangan sistem dimana terbagi jadi dua bagian yaitu implementasi perangkat keras dan perangkat lunak. Pada perangkat keras dilakukan penyusunan / konfigurasi sensor

node yang meliputi arduino, xbee shield, xbee, aktuator. Pada tahap perangkat lunak pada sisi sensor node meliputi pemrograman arduino menggunakan bahasa C. Pada sisi sink meliputi pembuatan *website* menggunakan bahasa html dan php.

3.1.5 Pengujian dan Analisis

Pengujian sistem pada penelitian ini dilakukan agar menunjukkan bahwa sistem telah mampu bekerja sesuai dengan kebutuhan yang telah ditetapkan. Pengujian yang dilakukan antara lain adalah:

1. pengujian range area
2. Pengujian komunikasi data
3. Pengujian pengiriman data
4. Pengujian sistem secara keseluruhan

Pengujian range area bertujuan untuk mengetahui bagaimana kualitas dari sinyal xbee dan kemampuan perangkat dalam berkomunikasi amupun menjangkau area yang akan digunakan.

Pengujian komunikasi data yaitu bertujuan untuk mengetahui apakah antara xbee koordinator dengan xbee router satu maupun router dua dapat saling berkomunikasi dengan baik atau tidak.

Pengujian pengiriman data yaitu bertujuan untuk mengetahui apakah data yang dikirim dari arduino koordinator dapat sampai pada arduino router satu dan router dua. Dan juga untuk mengetahui berapa bersaran nilai periode ideal untuk sekali pengiriman data.

Pengujian sistem secara keseluruhan bertujuan untuk mengetahui kinerja sistem secara keseluruhan mulai dari perintah pada perangkat lunak hingga pada hardware apakah sudah bekerja sesuai dengan perintah yang dikirimkan.

3.1.6 Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan dan implementasi dilakukan. Kesimpulan diambil untuk menjawab rumusan masalah yang sudah ditetapkan. Tahap akhir penulisan yaitu saran dimana bertujuan untuk

memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakan penulisan dan pengembangan pada penelitian selanjutnya

3.2 Perancangan

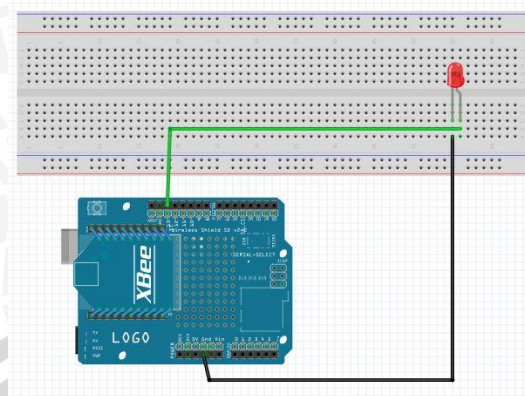
Perancangan dalam pembuatan sistem ini terbagi menjadi dua yaitu perancangan perangkat keras dan perangkat lunak, perangkat keras terdiri dari perancangan arduino, xbee shield, dan LED, sedangkan perancangan perangkat lunak terdiri dari pemrograman arduino dan web.

3.2.1 Perancangan Perangkat Keras

Perangkat keras yang dibutuhkan untuk Implementasi *Prototype System Monitoring and Controlling Smart Home* Menggunakan *Embedded system*, yaitu meliputi :

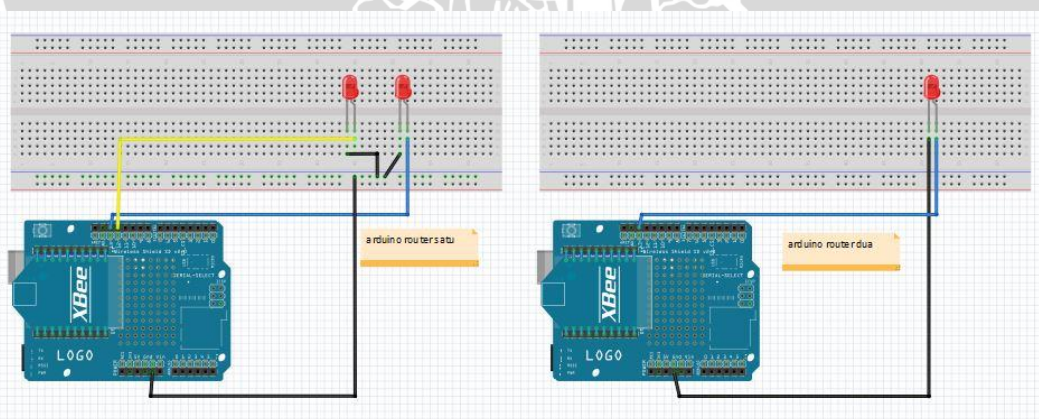
1. 3 buah Arduino Uno R3
2. 2 buah Xbee pro
3. Xbee
4. 3 buah Xbee shield
5. LED
6. Kabel

Semua kebutuhan perangkat keras nantinya akan dirangkai menjadi tiga buah rangkaian yang terdiri dari arduino koordinator, arduino router satu dan arduino router dua. Pemilihan arduino uno dan xbee dikarenakan alat yang mudah didapat serta mudah dalam pengkonfigurasi. Untuk perancangan pada perangkat arduino koordinator dijelaskan pada Gambar 3.2 .



Gambar 3. 2 Perancangan arduino koordinator

Pada Gambar 3.2 terdapat beberapa susunan rangkaian yaitu arduino uno R3 , xbee shield itead, xbee pro yang terangkai menjadi satu susunan. Untuk xbee itead shield menggunakan semua pin yang ada pada arduino uno. Sehingga untuk pemasangan LED maupun sensor cahaya tidak perlu melepas xbee shield. Kemudian LED yang terpasang pada breadboard disambungkan menggunakan kabel jumper pada pin 13 sebagai masukan LED dan *ground* (GND). Pada Gambar 3.3 merupakan perancangan rangkaian pada arduino router satu dan arduino router dua.

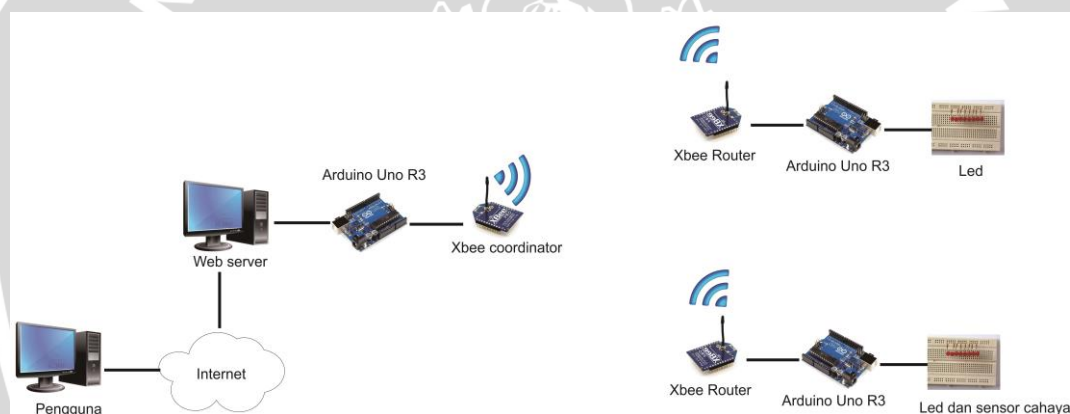


Gambar 3. 3 Perancangan arduino router satu dan arduino router dua

Pada Gambar 3.3 terdapat dua Gambar perancangan yaitu pada arduino router satu dan arduino router dua. Perbedaan mendasar dari kedua rangkaian

tersebut hanya jumlah LED yang digunakan, alur rangkaian kabel serta xbee yang digunakan adalah xbee seri biasa. Untuk rangkaian arduino sama seperti arduino koordinator yaitu arduino uno R3 , xbee shield itead, dan xbee yang terangkai menjadi satu susunan. Untuk xbee itead shield menggunakan semua pin yang ada pada arduino uno.

Pada arduino router satu pin 13 dan pin 12 digunakan sebagai masukan LED satu dan dua, sedangkan ground (GND) menggunakan satu pin saja. Untuk arduino router dua pin 13 digunakan untuk masukan data ke LED. Setelah dilakukan perancangan semua hardware yang ada kemudian dirangkai menjadi sebuah topology yang terdapat pada Gambar 3.4



Gambar 3. 4 Skema topology

Gambar 3.4 menunjukkan topology yang akan digunakan. Pertama pengguna akan mengakses system *monitoring* dan *controlling* pada web yang telah disediakan dengan memanfaatkan layanan internet, kemudian perintah yang telah dimasukan oleh pengguna akan disimpan dan di proses oleh webserver. Perintah yang sudah di proses oleh webserver akan diteruskan ke arduino koordinator melalui kabel serial. Di dalam arduino koordinator perintah akan dicek kemudian di kirim ke arduino router secara wireless dengan memanfaatkan xbee pada koordinator. Data yang dikirim diterima oleh xbee pada router yang kemudian di teruskan ke arduino router satu dan arduino router dua, di dalam arduino router satu dan dua dilakukan

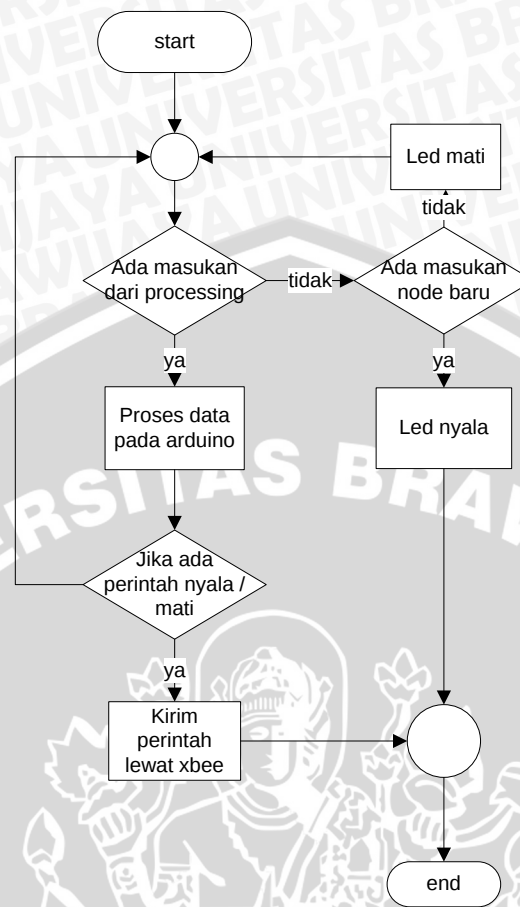
pencocokan perintah yang dikirim dengan perintah yang ada pada arduino router satu dan dua, apabila perintah sesuai maka lampu LED yang terpasang pada arduino router dan dua akan menyala sesuai dengan perintah yang dikirim dari webserver namun jika perintah tidak sesuai maka tidak akan diolah. Pada arduino dua jika perintah masuk ke dalam mode menyala maka sistem akan otomatis pindah menggunakan sensor cahaya namun tetap dapat dimatikan sesuai keinginan. Selain itu jika arduino koordinator tidak dapat menjangkau keberadaan arduino dua maka perintah dari arduino koordinator diteruskan oleh arduino router satu yang ditempatkan antara arduino koordinator dan arduino router dua.

3.2.2 Perancangan Perangkat Lunak

Pada perancangan program terbagi menjadi dua tahap yaitu perancangan program pada Arduino IDE dan perancangan tampilan *website*. Perancangan ini dilakukan untuk memenuhi kebutuhan fungsional sistem.

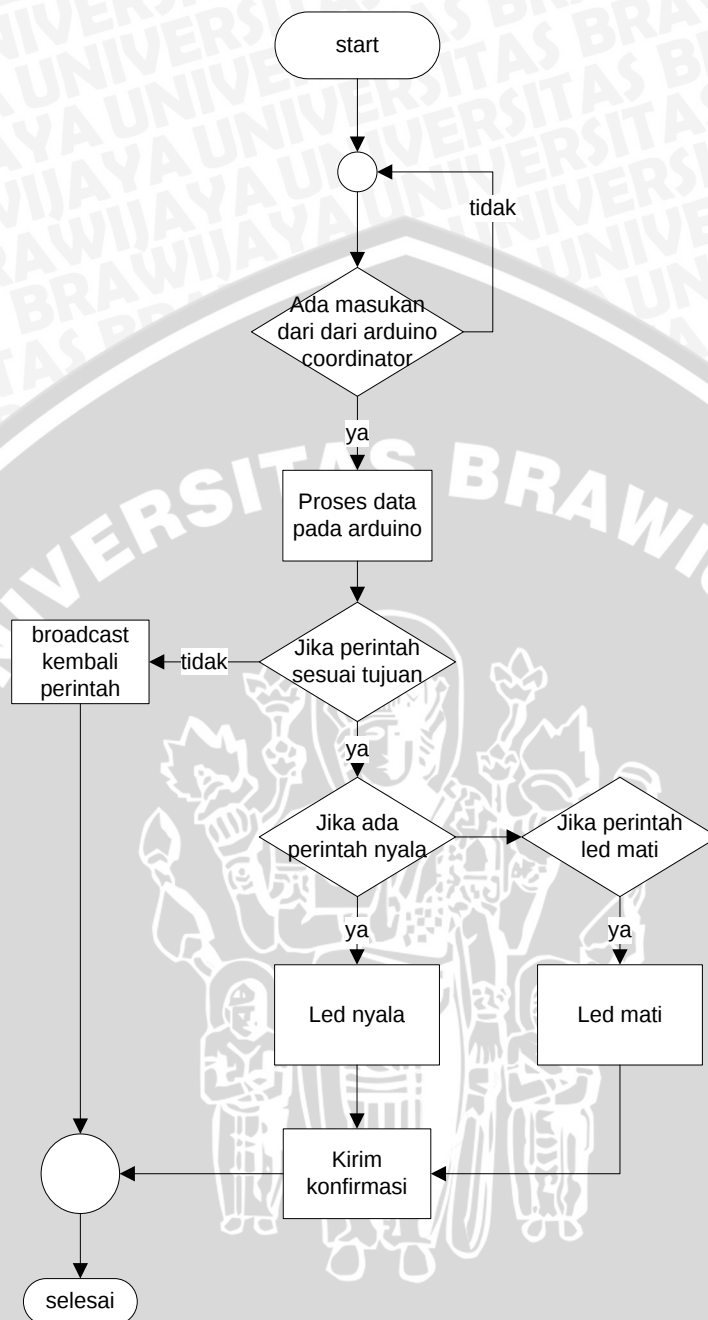
3.2.2.1 Perancangan Program Arduino

Perancangan pada program arduino menggunakan bahasa C dimana nantinya arduino akan terhubung dengan xbee dan pc yang bertindak sebagai web server pada sisi koordinator. Sementara pada sisi router terdapat dua buah arduino router akan terhubung dengan xbee serta LED dan sensor cahaya pada masing-masing arduino. Penjelasan pada sisi koordinator ditunjukkan pada Gambar 3.5



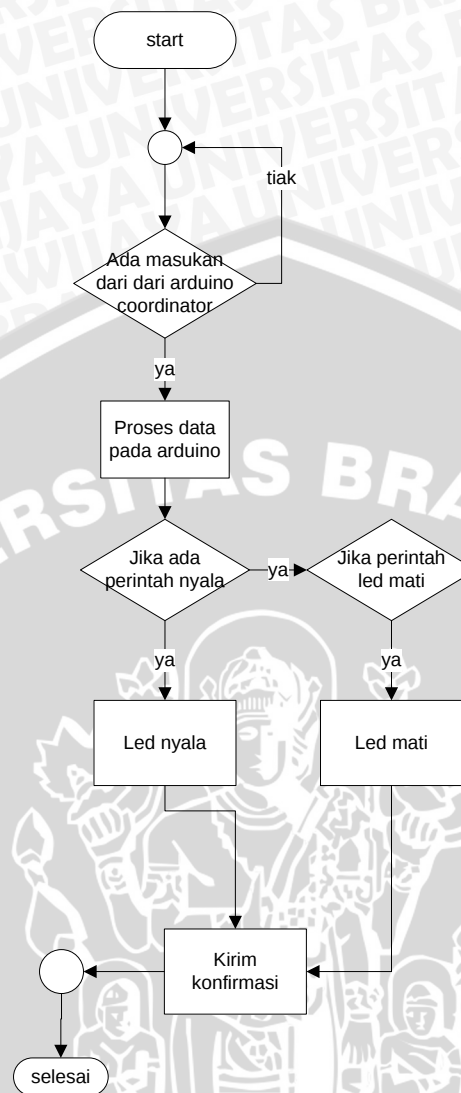
Gambar 3. 5 Flowchart perancangan sistem pada koordinator

Pada program sisi koordinator akan menerima apakah ada masukan dari web server untuk menyalakan LED atau tidak. Jika terdapat perintah maka yang dilakukan adalah melakukan pengecekan LED berapa yang akan dihidupkan. LED disini dianggap sebagai lampu yang akan dinyalakan atau dimatikan. Jika pengecekan sesuai dengan data perintah LED yang ada pada arduino koordinator maka akan dilakukan pengiriman data perintah kepada arduino router satu melalui xbee. Namun apabila tidak ada perintah atau perintah tidak sesuai dengan data yang ada maka tidak akan ada pengiriman data dan kembali ke proses pengecekan data. Pada arduino koordinator juga mampu mendeteksi jika ada arduino router baru yang sebelumnya sudah terkonfigurasi sesuai standart yang ada dan menerima perintah masukan maka LED sebagai indikator menyala yang menandakan bahwa ada arduino router baru jika tidak maka akan kembali ke proses pengecekan perintah.



Gambar 3. 6 Diagram Perancangan Sistesm pada router satu

Pada Gambar 3.6 menunjukkan bahwa arduino router satu akan mengecek apakah ada masukan dari arduino koordinator atau tidak . Jika ada perintah yang sesuai dengan data yang ada pada arduino router satu maka akan mengeksekusi apakah LED menyala atau mati. Namun jika perintah tidak sesuai dengan data yang ada maka akan diteruskan kepada arduino router selanjutnya.

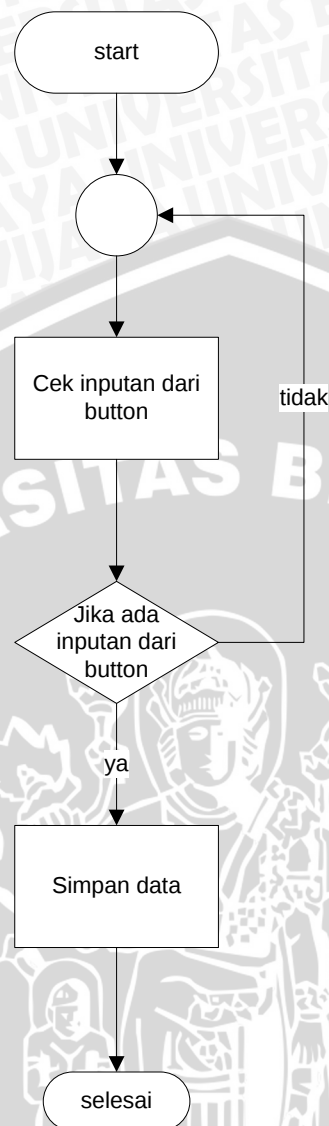


Gambar 3. 7 Flowchart sistem pada arduino router dua

Pada arduino router dua cara kerja hampir sama dengan arduino router satu. Jika ada perintah untuk enyalakan LED maka sistem akan mengeksekusi apakah LED akan menyala atau mati. *Flowchart* sistem ditunjukkan pada Gambar 3.7

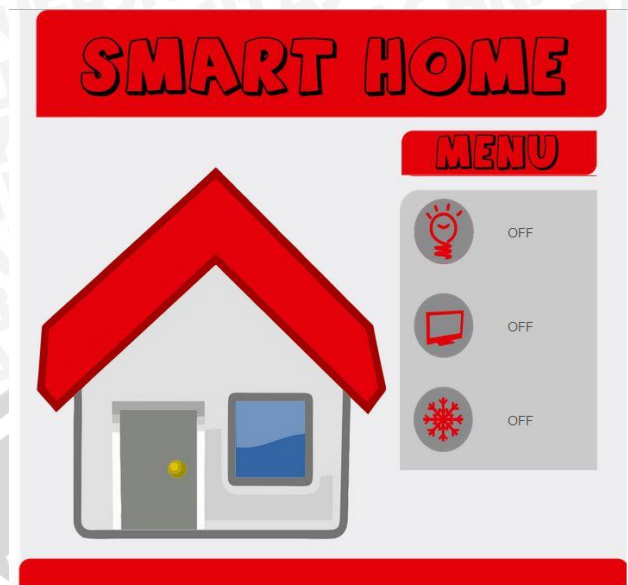
3.2.2.2 Perancangan Website

Perancangan *website* bertujuan sebagai pusat control dari sistem *monitoring* dan *controlling*. *Website* nantinya akan dipasang pada webserver yang berada pada pc. Tugas dari *website* adalah memberikan perintah kepada arduino koordinator apakah LED nyala atau mati dan juga sebagai *monitoring* kondisi LED yang ada apakah nyala atau mati. Perancangan *website* menggunakan bahasa pemrograman PHP dan HTML.



Gambar 3. 8 Flowchart perancangan website

Pada Gambar 3.8 ditunjukkan proses yang terjadi pada *website*. Langkah pertama yang dilakukan adalah pengecekan masukan yang berasal dari *button*, jika terdapat masukan dari *button* maka nilai masukan akan disimpan pada notepad yang nantinya akan dibaca oleh arduino namun jika tidak ada perintah dari *button* maka akan kembali ke pengecekan awal. Untuk tampilan *website* ditunjukkan pada Gambar 3.9

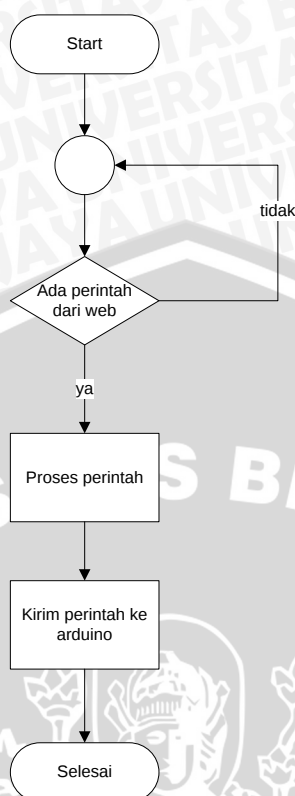


Gambar 3. 9 Tampilan web

Data yang digunakan *website* berupa perintah yang disimpan dalam file .txt kemudian file .txt akan dikonversi ke kode ASCII yang dikomunikasikan melalui kabel serial sehingga dapat dibaca oleh koordinator yang kemudian diteruskan ke router.

3.2.2.3 Perancangan program processing

Perancangan program processing menggunakan bahasa java sederhana. Program ini berfungsi sebagai pembaca data pada file.txt yang berisi perintah dari *website*, kemudian perintah yang ada diterjemahkan ke dalam ASCII code sehingga bisa dibaca arduino. Processing nantinya akan dipasang pada server yang terhubung dengan arduino koordinator. Pengiriman data dari *processing* ke arduino koordinator menggunakan kabel serial. Berikut flowchart pada *processing* ditunjukkan pada Gambar 3.10



Gambar 3. 10 Diagram alir processing

Pada perancangan program processing yang dilakukan pertama adalah membaca masukan yang berada pada file.txt. Perintah yang terbaca oleh processing akan di teruskan ke arduino koordinator yang terhubung melalui kabel serial. Jika tidak terdapat perintah maka kembali ke proses pengecekan perintah.

3.2.2.4 Perancangan Protokol

Untuk protokol yang digunakan diadopsi dari protokol pada penelitian sebelumnya (ALN - 2013) . Penggunaan protokol ini sendiri dikarenakan untuk pengiriman data antar xbee belum terdapat protokol yang sudah standarisasi untuk sistem smarthome terutama dilingkungan PTIIK sendiri. Protokol yang digunakan juga sedikit di modifikasi dari 13 data yang sudah dirancang di ubah menjadi 9 data saja. Perubahan ini dilakukan untuk menyesuaikan kebutuhan sistem. Gambar rancangan protokol dari penelitian sebelumnya ditunjukkan oleh Gambar 3.11

1	2	3	4	5	6	7	8	9	10	11	12	13
---	---	---	---	---	---	---	---	---	----	----	----	----

Header	separator	data	separator	data	separator	data	separator	data	separator	data	separator	Footer
--------	-----------	------	-----------	------	-----------	------	-----------	------	-----------	------	-----------	--------

Gambar 3. 11 Rancangan Protokol awal

Perubahan perancangan prtokol dilakukan agar proses kerja beban pada saat pengecekan tidak terlalu besar. 1 buah ukuran data pada frame protokol sebelumnya 1 byte sehingga ketika dijumlah dalam sekali melakukan pengiriman diperlukan 13 byte. Penyederhanaan dilakukan dengan mengurangi panjang frame menjadi 5 buah Sehingga diasumsikan tidak memakan memory yang terlalu besar. Frame yang digunakan seperti pada Gambar 3.12

1	2	3	4	5
Header	sperator	data	separator	footer

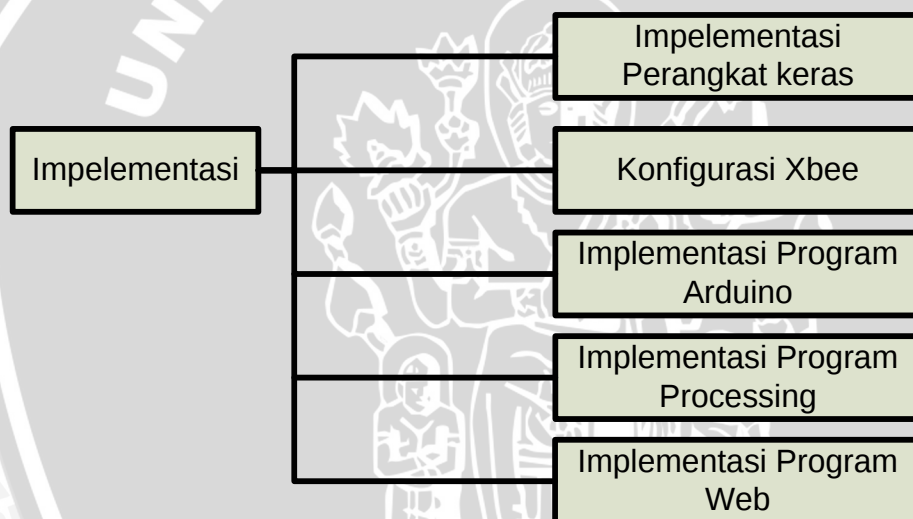
Gambar 3. 12 Rancangan Protokol akhir

Untuk data yang dikirim sendiri menggunakan konsep seperti bilangan biner dengan kombinasi 4 angka yaitu (0000 - 1111) . Tujuan dari penggunaan ini dilakukan agar pengalamatan pada masing – masing perintah lebih sederhana .

BAB IV

IMPLEMENTASI

Pada bab ini akan dibahas langkah – langkah yang dilakukan dalam pembuatan sistem Implementasi *Prototype System Monitoring and Controlling Smart Home* Menggunakan *Embedded System* sesuai dengan perancangan pada bab III. Pada implementasi ini, langkah – langkah yang akan dilakukan adalah implementasi perangkat keras, konfigurasi xbee, pembuatan program untuk arduino, serta pembuatan program pada php.



Gambar 4. 1 Diagram Bab Implementasi

4.1 Batasan Impelementasi

Beberapa batasan dalam implementasi prototype sistem controlling dan monitoring adalah sebagai berikut :

1. Peralatan elektronik seperti lampu dan televisi digantikan oleh LED.
2. Sistem Monitoring hanya sebatas mengetahui kondisi alat elektronik dalam keadaan on / off.
3. Perangkat arduino router baru yang akan di deteksi sudah di setting sesuai standart ketentuan yang ada.

4. Perangkat keras yang digunakan berupa papan arduino uno R3 sebagai tempat eksekusi data, Xbee sebagai alat pengiriman data antar arduino.
5. Tempat controlling dan monitoring berbasis web, menggunakan bahasa pemrograman php.
6. Pada penelitian ini tidak membahas aspek keamanan jaringan.

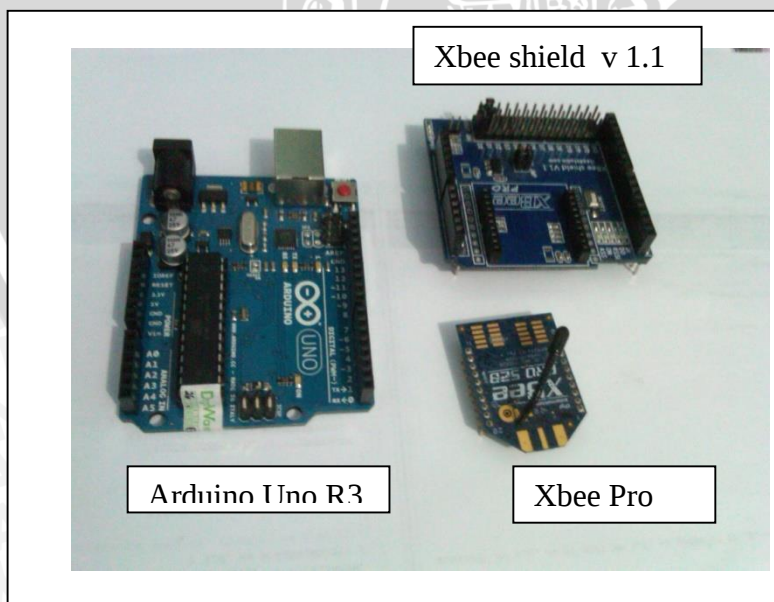
4.2 Implementasi Perangkat keras

Pada implementasi perangkat keras terbagi menjadi 2 buah bagian yaitu pada bagian koordinator yang bertindak sebagai master yang terhubung dengan laptop dan juga pengirim data dan pada router yang bertindak sebagai penerima dan pengolah data dari koordinator yang diteruskan ke LED.

1. Perangkat keras pada coordinator

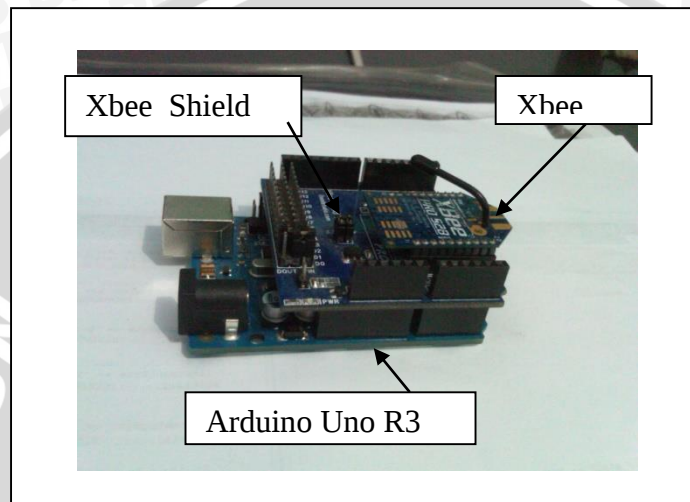
Langkah awal dalam pembangunan sistem pada sisi coordinator adalah mempersiapkan peralatan yang digunakan yaitu :

- a. Arduino Uno R3
- b. Xbee shield ITEAD V1.1
- c. Xbee pro



Gambar 4. 2 Perangkat coordinator

Pada Tahap selanjutnya yaitu pemasangan xbee pro ke xbee shield dengan cara ditumpuk dan disesuaikan dengan pin yang telah tersedia pada xbee shield, kemudian xbee shield ditumpuk pada arduino. Berikut Gambar yang menjelaskan bentuk tumpukan pada sisi coordinator.



Gambar 4. 3 Pemasangan Xbee pro,xbee shield dan arduino uno

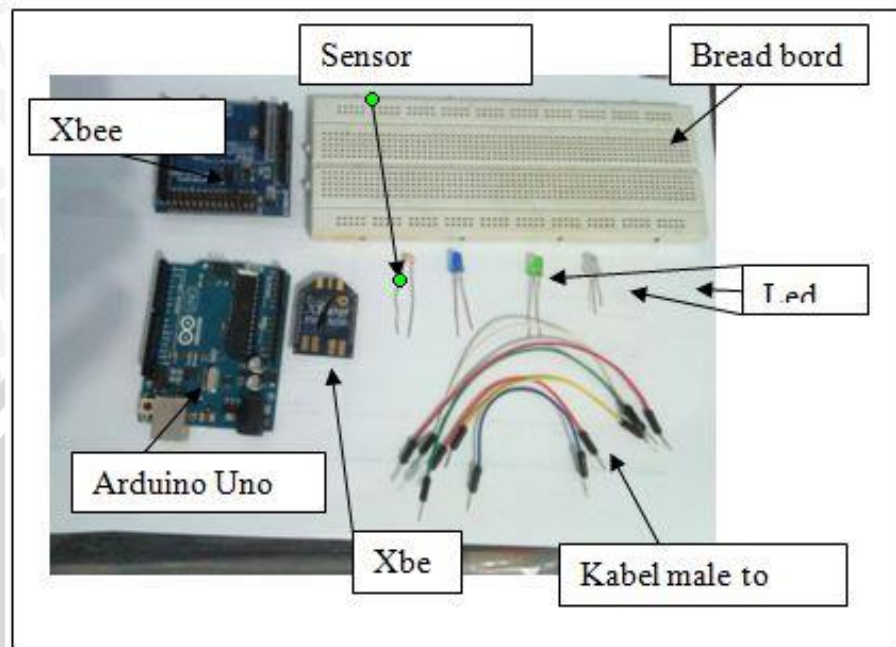
Secara keseluruhan pin yang terdapat pada arduino uno tersambung pada xbee shield. Sehingga memudahkan untuk pemasangan untuk perangkat tambahan seperti kabel, sensor tanpa harus kesulitan melepas xbee shield.

2. Perangkat keras pada arduino router

Arduino router yang digunakan berjumlah 2 buah yaitu arduino router A dan dan arduino router B. Langkah awal pada sisi router adalah mempersiapkan peralatan yang digunakan yaitu :

- a. 2 buah Arduino uno R3
- b. 2 buah Xbee Shield ITEAD V1.1
- c. Xbee pro
- d. Xbee s2
- e. Kabel male-male
- f. 3 buah LED

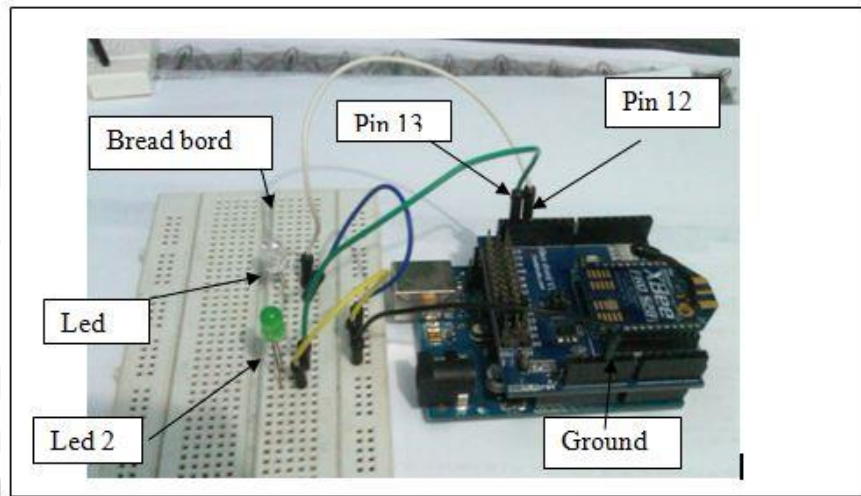
- g. 1 buah sensor cahaya
- h. 2 buah bread board



Gambar 4. 4 Perangkat Pouter

Pada tahap selanjutnya adalah dimulai dari arduino router satu .Arduino uno dipasangkan dengan xbee shield dan xbee pro dengan cara ditumpuk seperti pada konfigurasi sisi coordinator. Selanjutnya menghubungkan LED beserta rangkaian kabel pada papan bread board yang kemudian dilanjutkan dengan menghubungkan pada arduino router tahapan yg dilakukan yaitu :

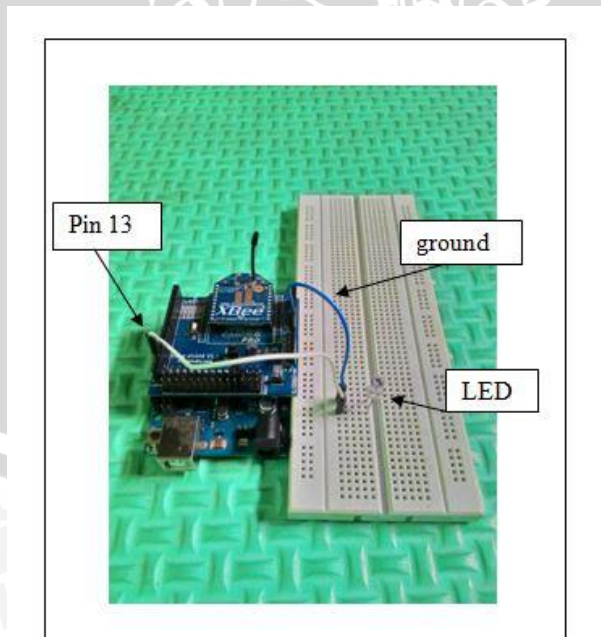
1. Masing-masing LED ditempatkan sejajar pada bread board kemudian.
2. Pin digital 13 pada arduino dihubungkan dengan pin anode(+) pada LED putih. Katode (-) terhubung dengan pin GND pada arduino.
3. Pin digital 12 pada arduino dihubungkan dengan pin anode(+) pada LED hijau .Katode (-)terhubung dengan pin GND pada arduino.



Gambar 4. 5 Hasil implementasi

Untuk implementasi pada rangkaian arduino router dua tahapan awal yang dilakukan sama dengan arduino router satu. Untuk rangkaian keseluruhan lebih sederhana dibandingkan arduino router satu. Tahapan yang dilakukan yaitu :

1. LED dan sensor cahaya ditempatkan pada bread board.
2. Pin analog 13 terhubung dengan anode(+) pada LED. Pin ini sebagai Katoda (-) akan terhubung dengan GND pada arduino.

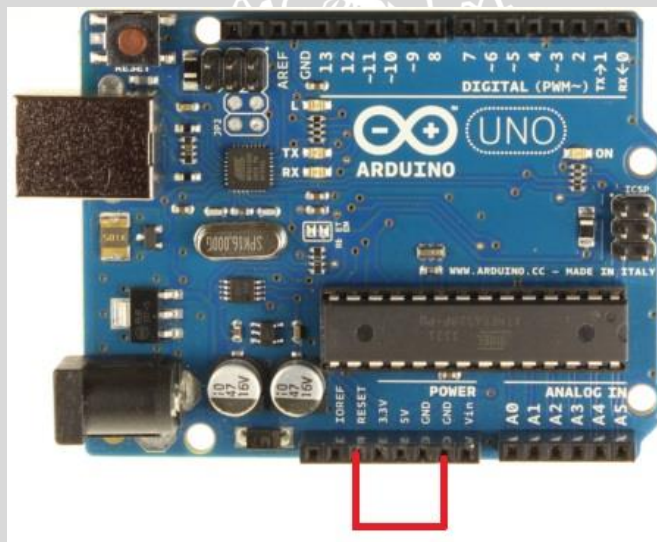


Gambar 4. 6 Hasil Implementasi

4.3 Konfigurasi Xbee

Implementasi komunikasi antar arduino meliputi komunikasi data antar tiga xbee. Xbee pada arduino pengirim atau disebut arduino coordinator mengirim data serial yang kemudian dilanjutkan melalui radio *frequency* dan dapat ditangkap oleh xbee pada arduino penerima atau disebut arduino router. Sebelum dapat digunakan xbee perlu konfigurasi terlebih dahulu agar dapat saling berkomunikasi.

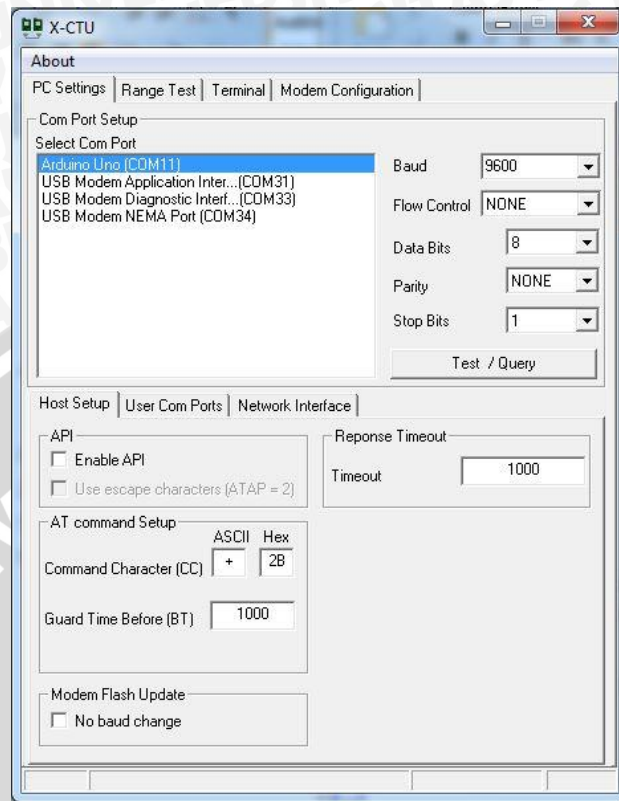
Konfigurasi tersebut dibantu dengan program komputer yaitu X-CTU. Pertama setting hardware pada xbee dan xbee shield yang terpasang pada arduino agar masuk ke dalam mode reset dengan cara menghubungkan pin ground pada pin reset dengan menggunakan kabel jumper. Seperti pada Gambar 4.7 berikut ini.



Gambar 4. 7 Konfigurasi dalam mode reset

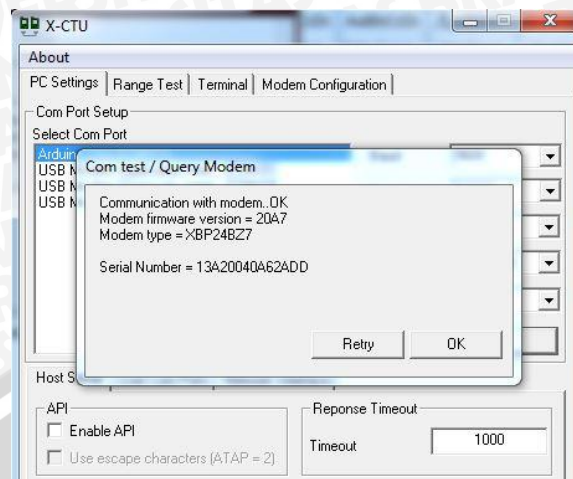
Setelah konfigurasi hardware tahap selanjutnya yaitu menyambungkan arduino pada pc melalui kabel serial kemudian membuka program X-CTU pada pc. Berikut langkah –langkah yang dilakukan :

1. Membuka program X-CTU

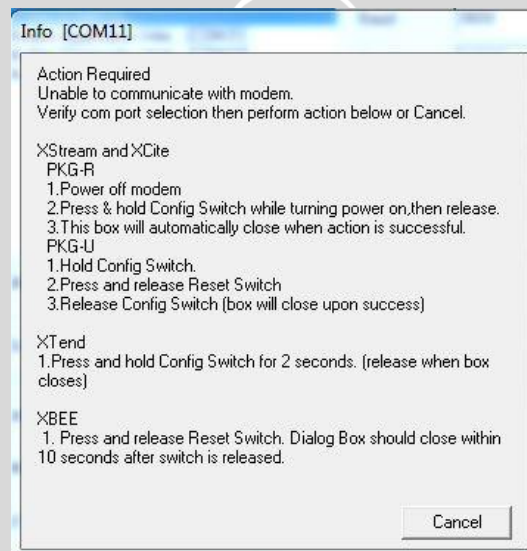


Gambar 4. 8 Tampilan awal X-CTU

2. Pada Tampilan awal pilih port yang digunakan, pada arduino coordinator menggunakan port COM. Kemudian setting *baudrate*, *flow control*, data bit, *parity*, *stop bits* sesuai kebutuhan. Dalam implementasi program ini digunakan settingan default seperti Gambar diatas.
3. Kemudian klik “Tes / Query ”untuk mengetahui apakah port sudah bisa digunakan atau belum. Jika Test yang dilakukan berhasil maka akan muncul kotak dialog yang menampilkan status dan model hardware xbee yang digunakan kemudian dialnjutkan dengan menekan tombol ok seperti yang ditunjukkan pada Gambar 4.9 namun jika test gagal maka akan muncul kotak dialog seperti pada Gambar 4.10



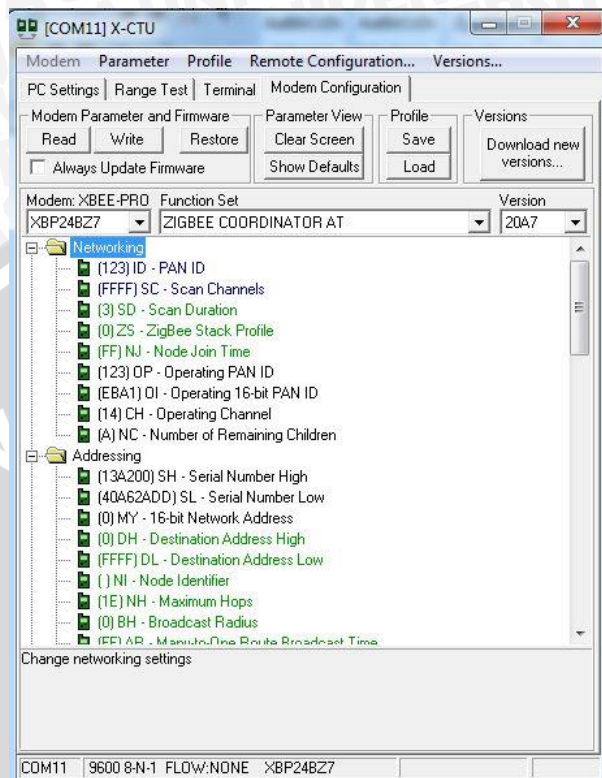
Gambar 4. 9 Tampilan Kotak dialog Test/Query sukses



Gambar 4. 10 Tampilan kotak dialog test/query gagal

4. Memilih sub menu Modem Configuration dan kemudian klik read pada bagian modem parameter and firmware, kemudian check list pada bagian *Always Update Firmware* dimana X-CTU akan menemukan model konfigurasi xbee yang digunakan.
5. Pada bagian *Tab Fuction Set* pilih *Zigbee Coordinator AT* yang berfungsi sebagai coordinator.

6. Setting Pan Id berupa karakter 0 – 1234. Masukkan Pan Id yang sama untuk semua device xbee yang digunakan agar dapat saling berkomunikasi.



Gambar 4. 11 Menu konfigurasi pada X-CTU

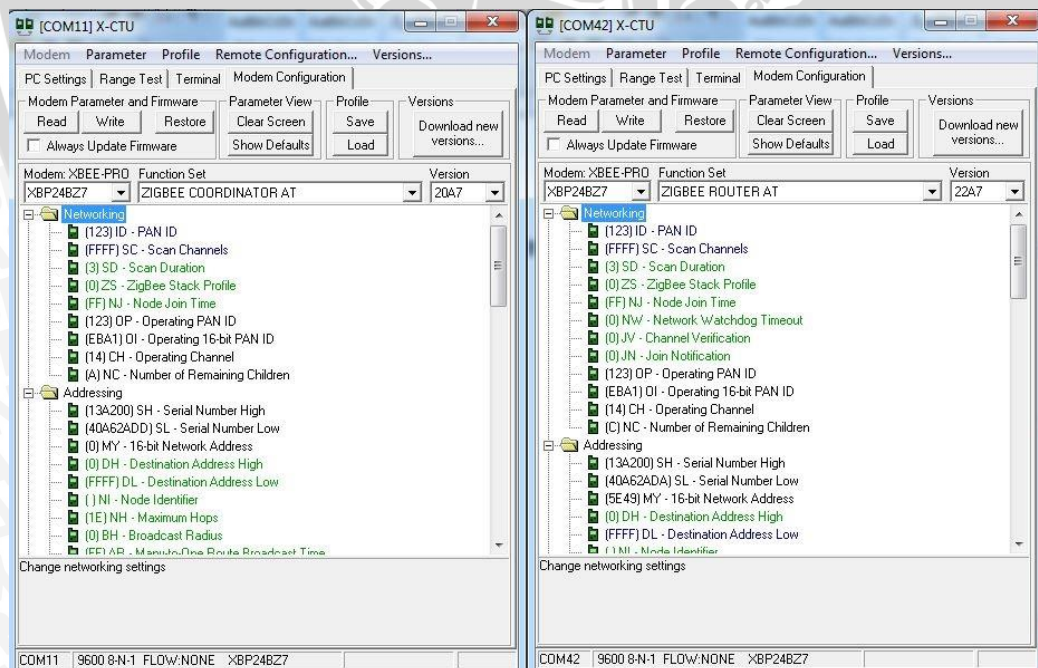
7. Setelah setting sesuai klik “write” dan tunggu sekitar 20 – 30 detik untuk konfigurasi pada xbee hingga muncul kotak dialog pada Gambar 4.12



Gambar 4. 12 Setting konfigurasi Xbee selesai

Pada sisi router setting yang dilakukan sama persis pada sisi coordinator hanya dibedakan pada parameternya saja. Langkah yang dilakukan adalah sebagai berikut :

1. Buka kembali program X-CTU
2. Pada tampilan awal pilih com port (Com 42 dan Com 16) yang digunakan dan setting default untuk *baudrate*, *flow control*, *bit*, *parity*, *stop bit*.
3. Kemudian klik “Tes / Query ” untuk mengetahui apakah sudah bisa digunakan atau belum jika sukses seperti pada Gambar 4.9 klik OK
4. Klik sub menu Modem Configuration dan kemudian klik read pada bagian modem parameter and firmware, kemudian check list pada bagian *Always Update Firmware* dimana X-CTU akan menemukan model konfigurasi Xbee yang digunakan
5. Pada bagian *Tab Fuction Set* pilih Zigbee Router AT yang berfungsi sebagai router.
6. Setting Pan Id yang digunakan sama dengan Pan Id pada coordinator . Jika berbeda 1 karakter saja maka tidak akan terjadi komunikasi antara coordinator dan router.



Gambar 4. 13 Seting Pan id coordinator dan router

7. Jika sudah selesai kemudian klik “write” kemudian tunggu 20-30 detik hingga konfigurasi selesai dan muncul kotak dialog seperti pada Gambar 4.12

4.4 Implementasi Program Arduino

Setelah melakukan tahapan konfigurasi perangkat keras dan konfigurasi xbee pada arduino, tahapan selanjutnya adalah membuat implementasi program pada arduino. Bahasa yang digunakan dalam pemrograman arduino adalah bahasa C. Implementasi program pada arduino terbagi menjadi tiga yaitu pada sisi coordinator, router satu dan router dua. Perbedaan antara router satu dan dua yaitu pada arduino router dua hanya menggunakan satu buah LED saja.

4.4.1 Implementasi Program Arduino Coordinator

Pada sisi arduino coordinator bertujuan untuk mengolah data yang diterima dari webserver yang kemudian dikirim melalui komunikasi Xbee menuju ke dua arduino router yang berbeda.

```

1. void loop() {
2.
3.     // jika ada komunikasi serial masuk
4.     if (Serial.available() > 0) {
5.         incomingByte = Serial.read();
6.
7.         //pencocokan perintah
8.         if (incomingByte == 'H') { // perintah nyala lampu 1
9.             mySerial.print("@,0001,#"); // data yang dikirim
melalui xbee
10.        }
11.
12.        if (incomingByte == 'L') { // perintah mati lampu 1
13.            mySerial.print("@,0000,#");
14.        }

```

Gambar 4. 14 Potongan Source code arduino coordinator

Pada Gambar 4.14 baris 3-5 merupakan pengecekan apakah ada komunikasi masuk yang berasal dari web atau tidak. Jika terdapat perintah maka akan di proses ke tahap selanjutnya yaitu pada baris 8 – 14. Pada baris ini dicocokkan data yang masuk dengan perintah yang ada, jika data yang masuk dan perintah sesuai maka arduino koordinator akan mengirimkan perintah selanjutnya menuju arduino router satu dan arduino router dua melalui xbee. Pada baris 8 – 14 merupakan pencocokan untuk LED satu pada arduino router satu, proses ini akan diulangi untuk pengiriman

data pada LED dua dan tiga yang berada pada arduino router satu dan dua, hanya dibedakan pada alamat pengiriman data saja.

```
1.   if(mySerial.available()>0){
2.       char dataku = (char)mySerial.read();
3.       data += dataku;
```

Gambar 4. 15 Potongan Source code arduino coordinator

Pada Gambar 4.15 merupakan pengecekan apakah ada kiriman data yang berasal dari arduino router baru. Jika terdeteksi ada pengiriman data maka akan lanjut ke proses selanjutnya .

```
1.   String perintah = dt[1];
2.   //Serial.println(dt[1]);
3.
4.   if(perintah == "1111"){
5.       digitalWrite(LEDpin, HIGH);
6.       delay(10000);
7.       digitalWrite(LEDpin, LOW);
8.   }
9.
10.  else {
11.      digitalWrite(LEDpin, LOW);
12.  }
13. }
```

Gambar 4. 16 Potongan Source code Arduino coordinator

Pada Gambar 4.16 merupakan proses terakhir dari pengecekan apakah ada arduino router baru yang masuk atau tidak. Jika ada maka akan otomatis menghidupkan LED yang berada pada arduino coordinator.

4.4.2 Implementasi Program Arduino Router satu

Pada implementasi router satu menerima data yang dikirimkan dari coordinator melalui xbee. Data yang diterima kemudian dicocokkan dengan perintah yang ada jika perintah sudah sesuai dengan data maka akan ada proses eksekusi sehingga LED bekerja sesuai perintah.

```
1.   void loop()
2.   {
3.       if(mySerial.available()>0){
4.           char dataku = (char)mySerial.read();
5.           data += dataku;
```

Gambar 4. 17 Potongan source code arduino router satu

Pada Gambar 4.17 merupakan perintah untuk memeriksa apakah ada masukan data dari arduino coordinator. Jika ada perintah masuk maka akan disimpan pada variabel data yang kemudian dilanjutkan kepada proses pengecekan dan eksekusi.

```
1.   String perintah = dt[1];
2.   //Serial.println(dt[1]);
3.
4.   if(perintah == "0001"){
5.       digitalWrite(LEDPin1, HIGH);
6.       //mySerial.print("@,1111,#");
7.       delay(1000);
8.   }
9.
10.  if(perintah == "0000"){
11.      digitalWrite(LEDPin1, LOW);
12.  }
13.
14.
15.  if(perintah == "0010"){
16.      digitalWrite(LEDPin, HIGH);
17.      Serial.print("nyala2");
18.  }
19.
20.  if(perintah == "0011"){
21.      digitalWrite(LEDPin, LOW);
22.      Serial.print("off");
23.  }
24.
25.  if(perintah == "0100"){
26.      //digitalWrite(LEDPin1, HIGH);
27.      mySerial.print("@,0100,#");
28.  }
29.
30.  if(perintah == "0101"){
31.      //digitalWrite(LEDPin1, LOW);
32.      mySerial.print("@,0101,#");
33.  }
34. }
```

Gambar 4. 18 Potongan source code arduino router satu

Pada Gambar 4.18 merupakan source code untuk proses eksekusi. Perintah yang masuk dari arduino coordinator akan dicocokkan pada masing-masing data yang ada. Jika sesuai maka LED akan menyala sesuai dengan perintah yang ada. Untuk baris 25 – 32 merupakan perintah untuk meneruskan pengiriman data yang berasal dari arduino coordinator menuju ke arduino router dua.

4.4.3 Implementasi Program Arduino Router dua

Pada implementasi router dua cara kerja program hampir sama dengan router satu yaitu menerima perintah yang dikirimkan dari arduino koordinator. Data kemudian dicocokkan, jika sesuai maka akan terjadi proses eksekusi LED sesuai dengan perintah

```
1.   if(perintah == "0100"){
2.       digitalWrite(LEDPin2, HIGH);
3.       mySerial.print("@,1111,#");
4.
5.   }
6.
7.
8.   if(perintah == "0101"){
9.       digitalWrite(LEDPin2, LOW);
10.      mySerial.print("@,1111,#");
11.  }
12.
13. }
```

Gambar 4. 19 Potongan Source code arduino dua.

Seperti pada penjelasan sebelumnya pada Gambar 4.19 merupakan potongan *source code* untuk arduino router dua. Cara kerja yang digunakan pada arduino router dua sama dengan arduino router satu. Pada baris 1 – 5 merupakan pencocokan perintah untuk menyalakan LED,

4.5 Implementasi Program Processing

Tujuan adanya implementasi program pada processing adalah membantu komunikasi data yang berasal dari web server agar dapat terbaca oleh arduino coordinator. Komunikasi data disini merupakan pembacaan data yang berada pada file .txt. Perintah yang berasal dari webserver akan tersimpan pada file .txt, kemudian isi file tersebut dibaca oleh processing dan dicocokkan pada perintah yang terdapat dalam processing. Jika perintah sesuai maka processing akan mengirimkan data selanjutnya berupa kode ascii menuju ke arduino coordinator melalui kabel serial.

```
1. void draw() {
2.
```

```

3. String onoroff[] =
loadStrings("http://localhost/ard/aku.txt");// membaca lokasi
file txt
4. String onoroff2[]
=loadStrings("http://localhost/ard/aku2.txt");
5. String onoroff3[] =
loadStrings("http://localhost/ard/aku3.txt");
6.

```

Gambar 4. 20 Potongan Source code Program processing

Pada Gambar 4.20 menunjukan ptongan soruce code untuk pembacaan lokasi penyimpanan file.txt. Isi data yang dibaca adalah perintah yang berasal dari *webserver*. Ketika pembacaan berhasil maka akan dilanjutkan dengan pada proses pencocokan perintah yang masuk dengan perintah yang sudah di konfigurasi.

```

1.   if (onoroff!=null)
2.   {
3.       if (onoroff[0].equals("1") == true) {
4.           println(" - ke 1");
5.           port.write('H');// mengirim data H untuk perintah
menyala
6.       }
7.       else {
8.
9.           println(" -matikan ke 1");
10.          port.write('L');// mengirim data L untuk perintah
mematikan
11.      }
12.  }

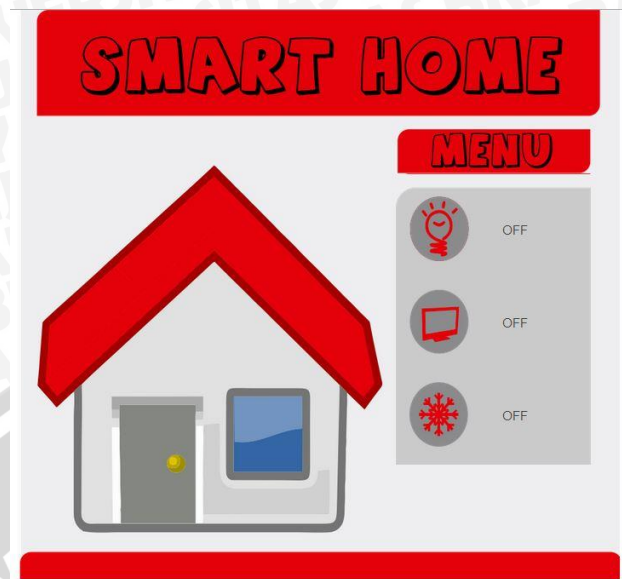
```

Gambar 4. 21 Potongan Source code Program Processing

Selanjutnya pada Gambar 4.21 merupakan perintah pencocokan data yang berasal dari file .txt. Pada baris 1 -10 perintah diproses jika perintah yang masuk berupa angka satu maka program akan mengirimkan perintah H (“menyala”) pada pada port dan selanjutnya dikirim melalui koneksi kabel serial menuju arduino coordinator. Namun jika perintah tidak sesuai maka data yang dikirim adalah perintah L (“mati”) pada port. Proses ini akan diulang untuk perintah LED ke dua dan ke tiga . yang dibedakan adalah pada printah yang dikirim.

4.6 Implementasi Program web

Untuk Implementasi *website* sesuai dengan apa yang telah disebutkan pada perancangan. Pada *website* ini memberikan informasi kepada pengguna bagaimana kondisi LED yang ada. Kondisi yang dimaksud adalah apakah LED berada dalam posisi mati atau menyala sesuai dengan perintah dan keadaan yang ada.



Gambar 4. 22 Tampilan Website

Pada Gambar 4.22 tampilan *website* tampak bahwa kondisi LED pada sisi sebelah kanan *website* yaitu kondisi off atau mati. Jika kita menekan *button* yang tersedia maka akan berubah menjadi on atau menyala dan secara otomatis mengirim perintah kepada arduino coordinator untuk mengolah data yang kemudian dilanjutkan untuk pengiriman kepada arduino router satu dan dua.

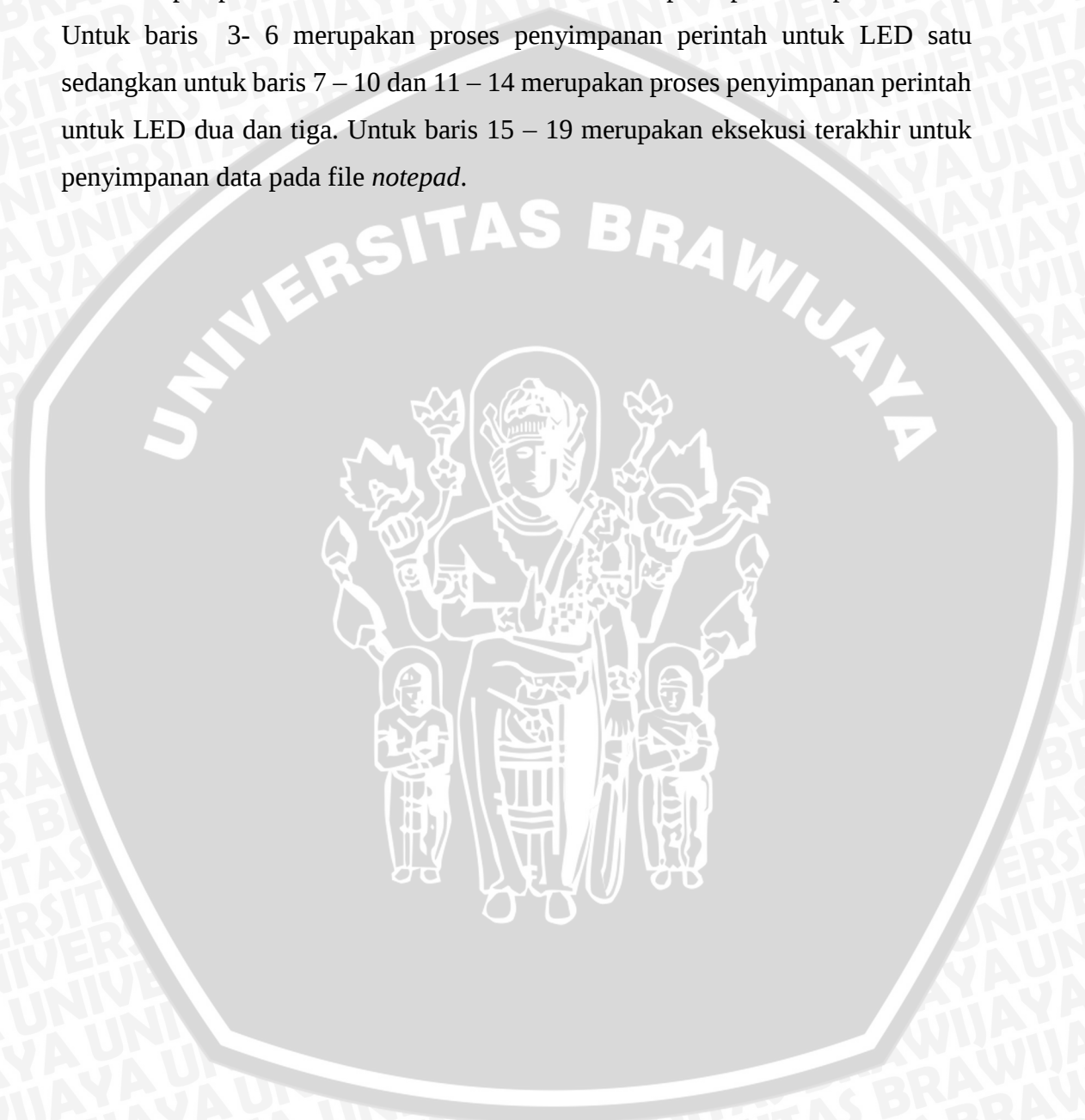
```

1. <?php
2. $sonoroff = $_POST["state"]; // dmenerima request dari
   website
3. if($sonoroff==1||$sonoroff==0)
4. {
5. $textfile = "aku.txt"; // menyimpan file di aku.txt
6. }
7. else if($sonoroff==2||$sonoroff==3)
8. {
9. $textfile = "aku2.txt"; // menyimpan file di aku2.txt
10. }
11. else if($sonoroff==4||$sonoroff==5)
12. {
13. $textfile = "aku3.txt"; // menyimpan file di aku3.txt
14. }
15. $fileLocation = "$textfile";
16. $fh = fopen($fileLocation, 'w+') ;// buka file txt yang
   ada dan replace data yg lama
17. $stringToWrite = "$sonoroff"; //
18. fwrite($fh, $stringToWrite); // menuliskan data pada
   file aku.txt
19. fclose($fh);

```

Gambar 4. 23 Potongan Source code website

Pada Gambar 4.23 merupakan potongan *source code* yang menunjukkan proses penyimpanan perintah dari *website* menuju file *aku.txt* pada baris 2 merupakan pembacaan apakah ada perintah yang masuk dari *website* atau tidak. Jika terdapat perintah masuk maka akan diteruskan pada perintah pencocokan. Untuk baris 3- 6 merupakan proses penyimpanan perintah untuk LED satu sedangkan untuk baris 7 – 10 dan 11 – 14 merupakan proses penyimpanan perintah untuk LED dua dan tiga. Untuk baris 15 – 19 merupakan eksekusi terakhir untuk penyimpanan data pada file *notepad*.



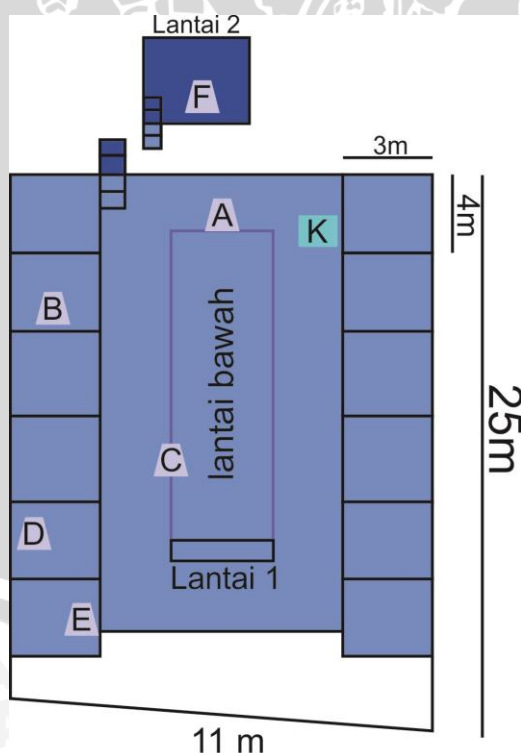
BAB V

PENGUJIAN DAN ANALISIS

Pada bab pengujian dan analisis ini dilakukan proses pengujian dan analisis terhadap keseluruhan sistem monitoring dan controlling yang telah dibuat. Pengujian dan analisis ini dilakukan untuk mengetahui apakah sistem yang dibuat sudah sesuai dan dapat berjalan dengan baik atau tidak. Pengujian terdiri dari beberapa proses yaitu pengujian range area, pengujian komunikasi data, pengiriman data dan pengujian sistem secara keseluruhan.

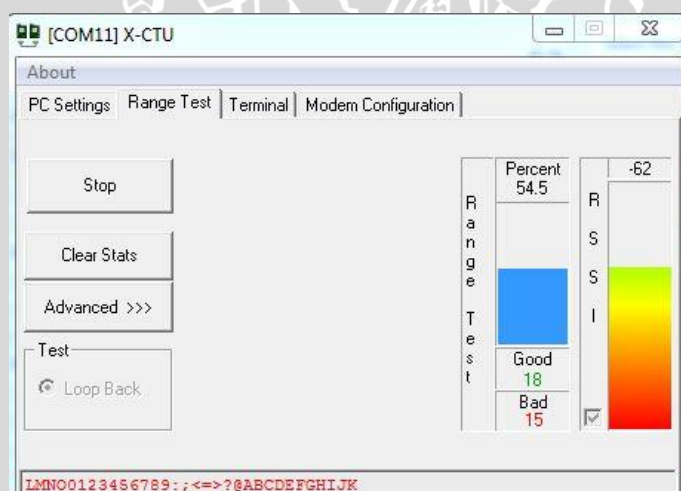
5.1 Pengujian Range area

Pengujian range area adalah pengujian yang bertujuan untuk mengetahui kemampuan jangkauan sinyal xbee sehingga masih dapat berkomunikasi. Pengujian dilakukan di sebuah rumah. Pada Gambar 5.1 menunjukkan denah lokasi untuk pengujian range area.



Gambar 5. 1 Denah lokasi pengujian range area

Proses yang dilakukan adalah meletakkan arduino koordinator yang telah terpasang xbee pada titik C yang berwarna biru. Kemudian meletakkan arduino router satu dan arduino router dua secara bergantian pada titik yang telah ditentukan yaitu pada titik A,B,C,D,E dan F yang berwarna silver. Pengujian ini dilakukan dengan bantuan software X-CTU. Pertama buka program X-CTU pada sisi arduino koordinator kemudian klik tab terminal pada software X-CTU. Pada terminal monitor mulai dengan memasukan perintah “+++” tanpa enter hingga muncul nilai keluaan “OK”, selanjutnya ketik “atnd” dan enter untuk melihat dan melaporkan node yang berada disekitaran arduino koordinator. Kembali masukan perintah “+++” pada X-CTU tanpa tanda enter hingga muncul nilai keluaran “OK”, kemudian ketik “atci12” dan tekan tombol enter, perintah ini berfungsi untuk mengirimkan data kepada arduino router satu maupun arduino router dua dan ketika data diterima maka secara otomatis akan dikirim kembali menuju arduino koordinator. Setelah memasukan komen kemudian klik Range test dan centang pada RSSI untuk memunculkan kekuatan sinyal seperti pada Gambar 5.2



Gambar 5. 2 Tampilan Data Sinyal

Setelah muncul tampilan sinyal seperti Gambar 5.4 kemudian dilakukan pencatatan nilai sinyal. Proses yang sama dilakukan untuk mengetahui sinyal pada titik selanjutnya serta pada range test arduino router dua. Hasil yang didapat pada pengujian range area ditunjukkan pada Tabel 5.1

Tabel 5. 1 Hasil Pengujian Range Area

No	Titik Router	Jarak (m)	Xbee Pro (arduino router satu) (-dbm)	Xbee S2 (Arduino router dua) (-dbm)
1	A	1	45	50
2	B	6	65	70
3	C	10	60	65
4	D	15	72	75
5	E	18	90	85
6	F	6	78	81

Analisis Pengujian Range Area

Pada pengujian ini telah didapat hasil pengujian seperti pada Tabel 5.1 Kualitas sinyal dianggap baik apabila nilai sinyal(-dbm) yang didapat <450ms [DHO - 2013]. Pada hasil Tabel terlihat bahwa pada titik E memiliki nilai sinyal paling besar untuk xbee pro maupun xbee s2 dikarenakan titik ini memiliki jarak paling jauh dari letak pusat pengiriman data pada arduino koordinator serta memiliki halangan (tembok) paling banyak dibandingkan titik lainnya.

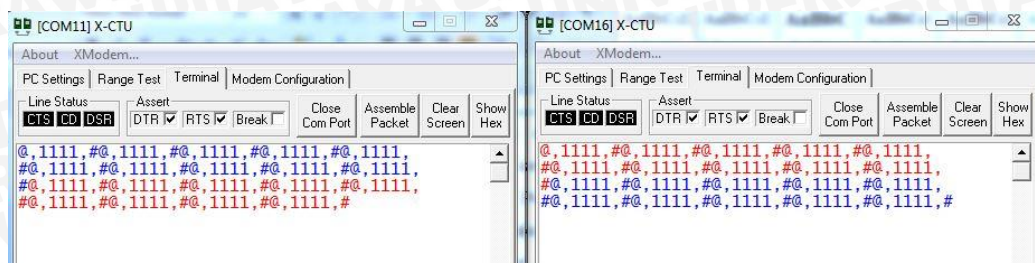
Pada titik ini juga terlihat bahwa nilai sinyal yang dimiliki xbee s2 memiliki nilai lebih kecil dibandingkan dengan xbee pro ini menunjukkan bahwa walaupun xbee s2 memiliki jangkauan yang lebih kecil dibandingkan xbee pro namun kemampuan menangkap sinyal sama baiknya dengan xbee pro. Untuk titik ideal jangkauan sinyal dengan halangan dan jarak yang jauh pada kedua xbee tersebut yaitu berada pada titik B dan D sementara titik terbaik tanpa halangan berada pada titik C.

5.2 Pengujian Komunikasi data

Pengujian pengiriman data adalah pengujian yang bertujuan untuk mengetahui apakah komunikasi antar xbee dapat berjalan dengan baik atau tidak dan untuk mengetahui apakah data yang dikirim dapat diterima dengan baik atau tidak. Pada pengujian ini prosedur yang dilakukan hampir sama dengan pengujian range area yaitu meletakkan arduino koordinator, arduino router satu dan arduino router dua pada titik – titik yang telah ditentukan sesuai dengan Gambar 5.1.

Perbedaan yang mendasar adalah pada perintah yang dimasukan. Pada tab terminal tidak dilakukan perintah “+++” ataupun “atcn” namun diketikkan karakter

pengiriman data . Karakter akan diketikan pada terminal X-CTU yang berada di arduino koordinator akan muncul pada arduino router satu dan arduino router dua begitu juga sebaliknya seperti yang ditunjukkan pada Gambar 5.3.



Gambar 5. 3 Hasil pengiriman pada X-CTU

Pengiriman karakter ini dilakukan sebanyak sepuluh kali pada tiap titik yang telah ditentukan. Hasil dari pengujian pengiriman data ditunjukkan pada Tabel 5.2

Tabel 5. 2 Hasil pengujian pengiriman data

No	Titik router	Data dikirim	Xbee Pro	Xbee
1	A	@,1111,#	Diterima	Diterima
2	B	@,1111,#	Diterima	Diterima
3	C	@,1111,#	Diterima, Delay	Diterima, Delay
4	D	@,1111,#	Diterima, Delay	Diterima, Delay
5	E	@,1111,#	Diterima, Delay	Diterima, Delay
6	F	@,1111,#	Diterima, Delay	Diterima, Delay

Analisis Pengujian Komunikasi Data

Pada pengujian pengiriman data didapatkan hasil bahwa sepuluh data yang dikirim dapat diterima dengan baik tanpa ada kehilangan satu karakter. Untuk titik A dan B data dapat diterima dengan lancar tanpa adanya kehilangan karakter atupun *delay* ini disebabkan jarak antara arduino koordinator dengan arduino router satu dan arduino router dua tidak terlalu jauh dan juga pada titik B hanya terdapat satu halangan (tembok) saja.

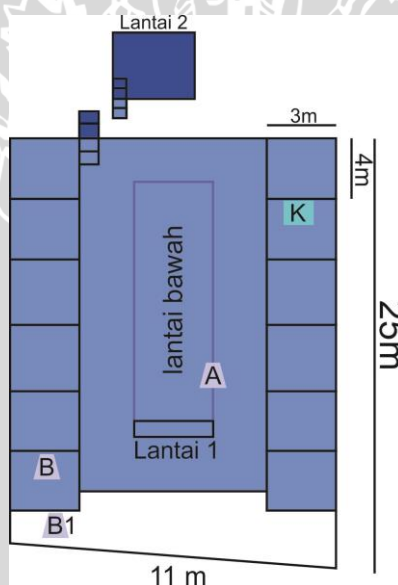
Sementara pada titik C, D, E, F data juga dapat diterima dengan utuh, namun terdapat *delay* ketika menerima data. Ini dikarenakan letak antara titik pengiriman

data dan penerimaan jauh selain itu juga pada titik D, E, F memiliki halangan(tembok) sehingga mempengaruhi sinyal untuk menyampaikan data.

5.3 Pengujian Pengiriman Data

Pengujian Pengiriman data adalah pengujian yang dilakukan untuk mengetahui apakah data dapat terkirim dengan baik dari arduino koordinator menuju ke arduino router dua dengan memanfaatkan arduino router satu dan pengiriman data dari arduino koordinator langsung menuju arduino router dua tanpa arduino router satu dengan nilai periode tertentu sehingga akan diketahui berapakah besaran nilai periode yang efektif untuk digunakan pada skema pengiriman data ini.

Pada pengujian pengiriman data ini dilakukan sedikit perubahan pada tempat pengiriman antara arduino koordinator dan kedua arduino router. Titik perubahan ditunjukkan pada Gambar 5.4



Gambar 5. 4 Denah lokasi pengujian pengiriman data

5.3.1 Pengujian Pengiriman Langsung

Pengujian pertama yang dilakukan yaitu Pengujian pengiriman Data langsung. Pada pengujian ini terdapat dua skema yang dilakukan yaitu pertama mengirimkan data dari arduino koordinator yang berada di titik K menuju ke arduino router satu yang berada pada titik A, kedua mengirimkan data dari titik K menuju ke arduino router dua yang berada di titik B maupun B1 dengan kondisi

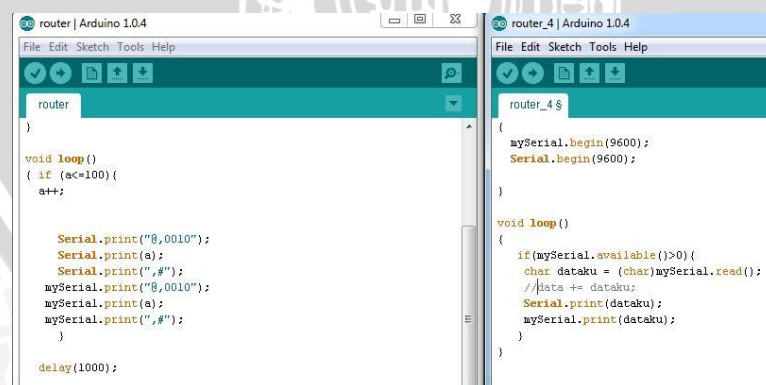
arduino router satu yang berada pada titik A mati. Langkah pertama yang dilakukan adalah menempatkan arduino koordinator pada titik K yang tampak pada Gambar 5.4. Kemudian menempatkan arduino router satu yang berada pada titik A, sedangkan pada arduino router dua yang berada pada titik B dalam kondisi mati.

Setelah menempatkan arduino koordinator dan arduino router pada titik yang telah ditentukan dilanjutkan dengan menghidupkan aplikasi *docklight* pada masing – masing titik. Aplikasi ini bertugas sebagai pencatat waktu data yang dikirim dan waktu data yang diterima pada masing – masing arduino.



Gambar 5. 5 Tampilan *docklight*

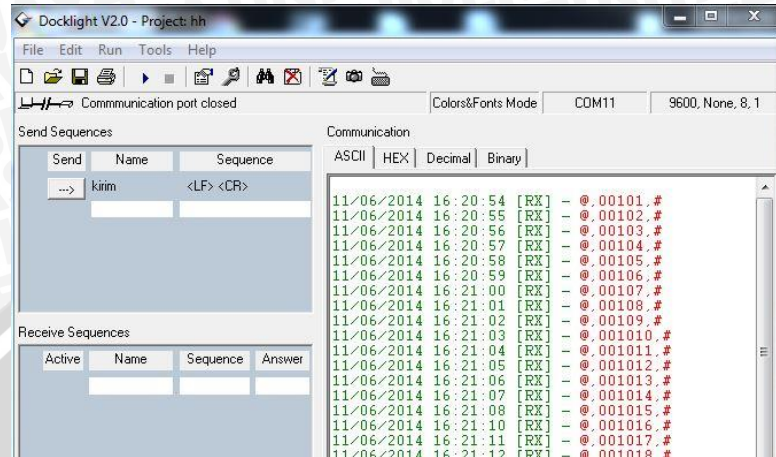
Selanjutnya mengunggah program yang sudah di buat pada IDE arduino di kedua titik secara bersamaan seperti pada Gambar 5.6. Sistem akan mengirim data yang sudah di konfigurasi sehingga terjadi proses pengiriman data dari titik K menuju ke titik A . Begitu juga saat dilakukan skema untuk pengiriman dari titik K menuju ke titik B ataupun B1



Gambar 5. 6 Code pengujian *delay*

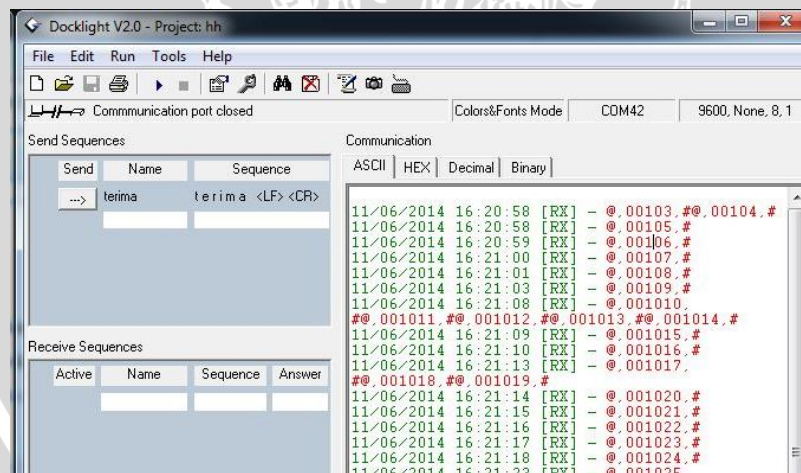
Pengiriman data yang dilakukan sebanyak 50 kali per waktu *delay*. Hasil yang tampil pada aplikasi *docklight* akan secara otomatis disimpan dalam format

.txt. Tampilan hasil pengiriman dan penerimaan data pada *docklight* ditunjukkan pada Gambar 5.7



Gambar 5. 7 Tampilan pada sisi pengiriman

Kemudian pada tampilan sisi arduino router satu hasil yang akan muncul adalah hasil komen perintah dari data yang masuk. Komentar ini muncul ketika data yang masuk sesuai dengan data yang sudah ada di arduino seperti pada Gambar 5.8



Gambar 5. 8 Tampilan pada sisi penerima

Setelah data masuk dan tersimpan di file.txt kemudian data diolah untuk mengetahui *delay*. Selanjutnya proses yang sama dilakukan yaitu mengirim data dari titik K menuju ke titik B setelah hasil didapat seperti pada Gambar 5.7 dan 5.8 dilanjutkan dengan pengiriman data dari titik K menuju ke titik B1. Hasil dari

pengujian ini seperti pada Gambar 5.9. Untuk hasil keseluruhan data hasil pengiriman terdapat pada lampiran.

Delay 1 detik K - B		
Dikirim	Diterima	Delay
09:22:52 - @,00101,#	Gagal	
09:22:53 - @,00102,#	Gagal	
09:22:54 - @,00103,#	Gagal	
09:22:55 - @,00104,#	Gagal	
09:22:56 - @,00105,#	09:23:52 - @,00105,#	96
09:22:57 - @,00106,#	09:23:53 - @,00106,#	96
09:22:58 - @,00107,#	09:23:54 - @,00107,#	96
09:22:59 - @,00108,#	Gagal	
09:23:00 - @,00109,#	Gagal	
09:23:01 - @,001010,#	Gagal	
09:23:02 - @,001011,#	09:24:00 - @,001011,#@,00101	98
09:23:03 - @,001012,#	Tumpuk	
09:23:04 - @,001013,#	Tumpuk	
09:23:05 - @,001014,#	09:24:05 - @,001014,#@,00101	100
09:23:06 - @,001015,#	Tumpuk	
09:23:07 - @,001016,#	Tumpuk	
09:23:08 - @,001017,#	Tumpuk	
09:23:09 - @,001018,#	Tumpuk	
09:23:10 - @,001019,#	09:24:05 - @,001019,#	95
09:23:11 - @,001020,#	09:24:06 - @,001020,#	95
09:23:12 - @,001021,#	09:24:07 - @,001021,#	95
09:23:13 - @,001022,#	09:24:08 - @,001022,#	95
09:23:14 - @,001023,#	09:24:10 - @,001023,#	96

Gambar 5. 9 Hasil Pengiriman data pada titik K - B

Analisis Pengujian Pengiriman Langsung

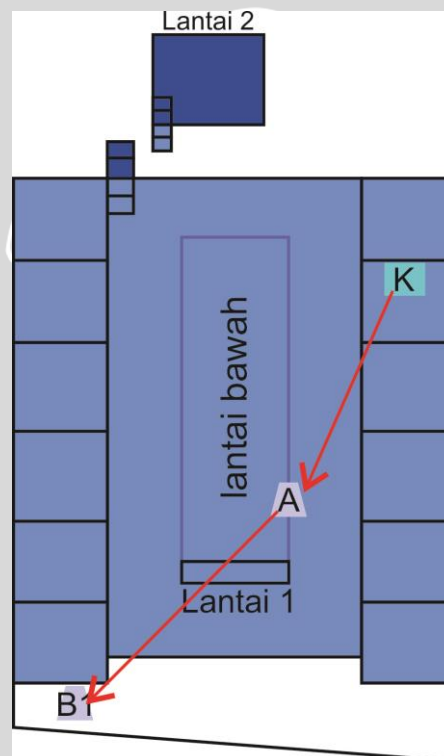
Dari hasil yang sudah didapat menunjukkan bahwa data dari titik K menuju ke titik A dapat diterima. Meskipun pada beberapa data gagal terkirim ataupun data yang masuk secara bersamaan dalam satu waktu. Ini disebabkan oleh waktu periode pengiriman yang singkat yaitu satu detik sehingga pengiriman data terjadi penumpukan pada waktu yang sama. Sementara untuk data yang terkirim dari titik K menuju ke titik B juga terdapat data yang menumpuk disaat yang bersamaan namun juga terdapat data yang gagal dikirim. Data yang gagal dikirim disebabkan jauhnya jarak antara titik K dan titik B sehingga menyebabkan adanya data yang tidak sampai ke tujuan. Selain itu adanya halangan (tembok) juga salah satu faktor yang membuat pengiriman data sedikit terhambat. Sementara untuk hasil pengiriman dari titik K menuju titik B1 tidak ada. Data yang dikirim dari titik K tidak diterima oleh titik B1 karena jarak dan halangan yang lebih banyak dari pada titik B sehingga menyebabkan data tidak dapat diterima. Sehingga jangkauan maksimal yang bisa ditempuh yaitu 20 meter.

5.3.2 Pengujian Pengiriman Tidak Langsung

Proses pengujian selanjutnya yaitu pengujian pengiriman data tidak

langsung. Pada pengujian ini proses yang dilakukan hampir sama dengan yang dilakukan dengan pengujian langsung.

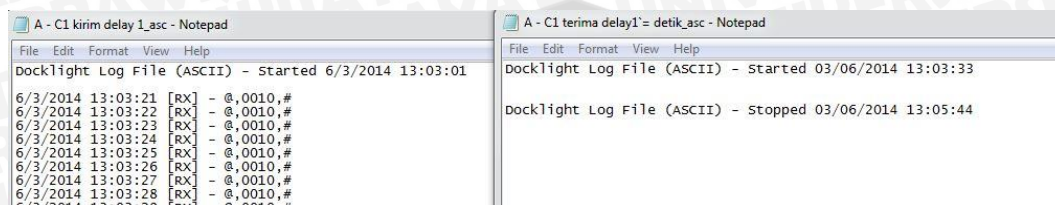
Perbedaan mendasar yaitu pada alur pengiriman data. Pengiriman data dari titik K menuju ke titik B1 harus melewati titik A terlebih dahulu kemudian dari titik A perintah yang masuk akan diteruskan ke titik B1. Perintah yang dikirim oleh arduino koordinator yang berada di titik K sebisa mungkin tidak dapat terbaca oleh arduino router dua yang berada pada titik B1. Namun pada titik B1 diberi perintah yang dapat membaca jika ada perintah terselip yang tidak melalui titik A sehingga dapat diketahui apakah perintah tersebut melalui titik A atau langsung. Skema titik pengujian ditunjukkan pada gambar 5.10



Gambar 5. 10 Skema pengujian pengiriman tidak langsung

Pengujian dimulai dengan melakukan pengiriman data yang berasal dari titik K. Untuk aplikasi memantau data yang dikirim dan data masuk, aplikasi *docklight* hanya dipasang pada titik K sebagai pengirim dan titik B1 sebagai penerima. Untuk menghidupkan arduino router satu pada titik A digunakan 1 unit powerbank. Arduino router yang digunakan pada titik B1 yaitu arduino router dua.

Percobaan dimulai dengan mengirim data dari titik K. Jumlah data yang dikirim sama dengan pengujian sebelumnya yaitu sebanyak lima puluh kali. Hasil yang didapat pada pengiriman dari titik K menuju ke titik B1 tanpa titik A terlihat bahwa data tidak masuk atau tidak sampai ke tujuan ini terlihat pada Gambar 5.11



Gambar 5. 11 Hasil pengiriman dari titik K menuju ke titik B1 tanpa titik A

Pada Gambar 5.11 terlihat bahwa data yang dikirim pada sisi kiri Gambar tidak terkirim, terlihat pada sebelah kanan Gambar yang hanya menampilkan informasi waktu pencatat saja. Kemudian saat pengiriman data menggunakan titik A sebagai perantara pengiriman dari titik K menuju ke titik B1. Data yang masuk akan tersimpan dalam bentuk file .txt yang kemudian diolah pada ms.excel seperti pada Gambar 5.12. Untuk hasil keseluruhan data hasil pengiriman terdapat pada lampiran 1.

	CB	CC	CD	CE	CF	CG	CH	CI
1	Delay 1 A B C1							
2	Dikirim				Diterima			delay
3	13:22:16 @,0010,#				13:22:25 nyala			9
4	13:22:17 @,0010,#				13:22:31 nyala			14
5	13:22:18 @,0010,#				13:22:35 nyala			17
6	13:22:19 @,0010,#				13:22:36 nyala			17
7	13:22:21 @,0010,#				13:22:41 nyala			20
8	13:22:21 @,0010,#				13:22:45 nyala			24
9	13:22:22 @,0010,#				13:22:47 nyala			25
10	13:22:23 @,0010,#				13:22:51 nyala			28
11	13:22:25 @,0010,#				13:22:52 nyala			27
12	13:22:26 @,0010,#				13:22:55 nyala			29
13	13:22:27 @,0010,#				13:23:01 nyala			74
14	13:22:28 @,0010,#				13:23:06 nyala			78

Gambar 5. 12 Hasil pengujian pengiriman tidak langsung

Analisis Pengujian Pengiriman Tidak Langsung

Untuk pengujian Pengiriman tidak langsung hasil yang didapat sesuai dengan yang di harapkan. Data yang dikirim dari titik K menuju ke titik B1 melalui

titik A dapat diterima dengan baik dan utuh. Meskipun terdapat jeda waktu *delay* ketika menerima data namun dapat berjalan dengan baik .

Ini ditunjukkan pada Gambar 5.11 ketika data dikirim dari titik K menuju ke titik B1 tanpa titik A tidak ada data yang masuk dikarenakan letak kedua titik yang jauh serta terdapat beberapa halangan tembok. Namun ketika diantara titik K dan titik B1 ditambahkan sebuah arduino router satu yaitu titik A untuk membantu komunikasi data yang tadinya tidak terkirim dapat diterima oleh titik B1 hasil ditunjukkan pada Gambar 5.12. Data yang dikirim dari titik K diterima terlebih dahulu oleh titik A kemudian titik A mengirim kembali menuju ke titik B1 sehingga titik B1 dapat menerima data yang dikirim dari titik K .

5.3.3 Pengujian Nilai Periode Ideal

Pengujian nilai periode ideal dilakukan untuk mencari nilai periode yang efektif untuk jeda setiap pengiriman data. Jeda nilai pengiriman yang dimaksud adalah ketika arduino koordinator mengirimkan data sebanyak lima puluh kali tiap satu kali pengiriman perintah di beri nilai periode antara satu detik, dua detik, tiga detik, empat detik dan lima detik. Pemilihan jeda waktu ini untuk mengetahui apakah dengan jeda waktu yang kecil sistem sudah dapat bekerja dengan baik atau tidak.

Pengujian ini dilakukan untuk mengetahui berapa nilai periode yang baik untuk proses pengiriman data dari arduino koordinator menuju arduino router satu dan arduino router dua. Skema yang dilakukan hampir sama dengan skema pengiriman langsung.

Langkah pertama yang dilakukan adalah meletakkan arduino pada titik yang telah ditentukan sama seperti dengan pengujian langsung yaitu pengiriman dari titik K menuju ke titik A, kemudian dari titik K menuju ke titik B dan titik B1. Denah pengujian ditunjukkan pada Gambar 5.1. Setelah meletakkan arduino koordinator dan arduino router pada titik yang ditentukan, Selanjutnya menentukan besaran nilai periode yang digunakan. Pertama buka IDE arduino kemudian merubah nilai besaran *delay* yang akan diuji, besaran nilai *delay* yang diuji yaitu satu detik , dua

detik, tiga detik, empat detik dan lima detik. Untuk pengujian pertama *delay* yang digunakan sebesar satu detik atau 1000ms.

Data yang dicatat pada aplikasi *docklight* kemudian disimpan pada file.txt yang kemudian diolah pada microsoft excel untuk mengetahui keutuhan data *delay* yang dihasilkan dari masing - masing nilai periode. Hasil perhitungan *delay* saat berada pada microsoft excel ditunjukkan pada Gambar 5.13. Untuk hasil keseluruhan data hasil pengiriman terdapat pada lampiran .

Dikirim				Diterima		delay (Detik)	
13:23:05	@,0010,#			13:25:12	nyala	207	
13:23:06	@,0010,#			13:25:16	nyala	210	
						5964	
						50	119,28

Gambar 5. 13 Potongan Hasil data akhir

Dari data yang masuk kemudian dilakukan perhitungan total *delay* pada masing – masing data yang sudah tersimpan. Dengan menggunakan persamaan :

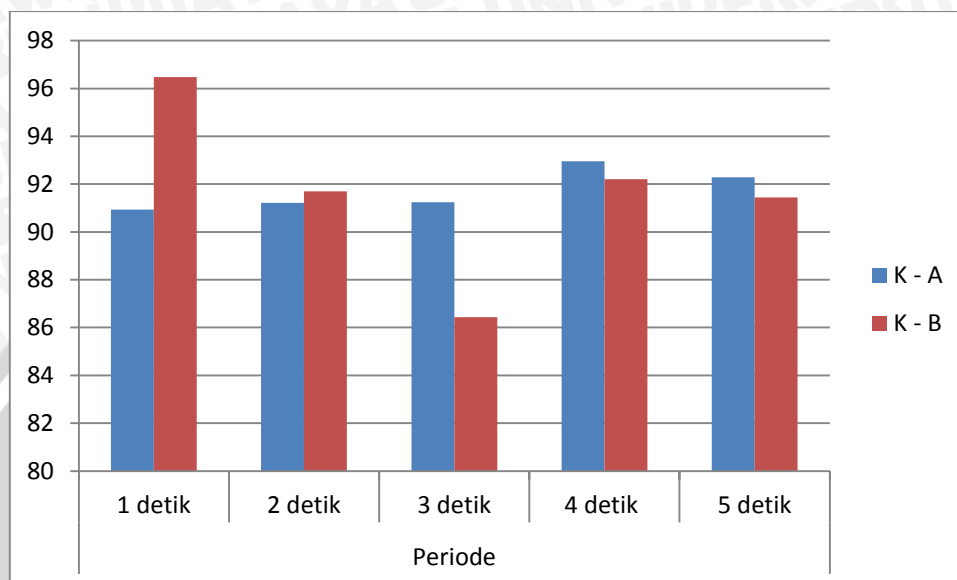
$$\text{Delay rata-rata} = \frac{\text{Total delay}}{\text{Total packet yang diterima}}$$

Perhitungan pertama yaitu mencari selisih jeda waktu antara data dikirim dan data di terima kemudian hasil pengurangan tersebut dibagi dengan jumlah total paket yang diterima sehingga mendapatkan hasil nilai periode rata – rata tiap pengiriman data. Hasil perhitungan *delay* terdapat pada Tabel 5.3.

Tabel 5. 3 Hasil perhitungan *delay*

Skema	Periode				
	1 detik	2 detik	3 detik	4 detik	5 detik
K - A	90,93	91,21	91,24	92,95	92,285
K - B	96,48	91,7	86,44	92,2	91,44
K - B1	0	0	0	0	0

Hasil yang didapat dimasukan dalam grafik pada gambar 5.14 agar mudah untuk melihat waktu periode yang cocok untuk pengiriman data yang diterapkan pada sistem keseluruhan ini.



Gambar 5. 14 Grafik hasil perhitungan *delay*.

Setelah menghitung *delay* untuk waktu setiap periode dilakukan pengecekan paket yang dapat diterima dan gagal perwaktu periode. Hasil yang didapat terlihat pada Tabel 5.4.

Tabel 5. 4 Hasil paket data diterima

Periode	Titik K - Titik A		Titik K - Titik B	
	Diterima	Gagal	Diterima	Gagal
1 Detik	30	20	25	25
2 Detik	46	4	48	2
3 Detik	49	1	29	21
4 Detik	49	1	35	15
5 Detik	49	1	45	5

Berdasarkan Gambar 5.14 dan Tabel 5.4 terlihat bahwa waktu periode yang baik untuk digunakan dalam sistem ini adalah dua detik. Ini terlihat dari jumlah data yang diterima pada masing - masing titik banyak ketimbang data yang gagal diterima.

Analisis Pengujian Nilai Periode Ideal

Dari beberapa skema pengujian nilai periode ideal dari titik K menuju titik A, B dan B1 memberikan hasil yang beragam. Ini terlihat pada Gambar 5.14 dimana waktu yang dihasilkan dari masing- masing titik pengiriman pengujian berbeda antara satu dengan yang lainnya.

Untuk Skema pengujian pertama yaitu pengiriman data dari titik K menuju titik A memberikan nilai *delay* yang besar pada setiap pengiriman data, dikarenakan jarak pada titik K menuju titik A dekat dan juga hanya terhalang oleh satu buah tembok saja. Untuk selisih *delay* pada periode dua detik hingga lima detik nilai *delay* yang dihasilkan pada masing- masing periode hampir sama , dikarenakan data yang masuk sukses diterima lebih ketimbang data yang gagal, sehingga hasil perhitungan jumlah nilai total lebih besar. Namun kekurangan pada titik ini yaitu ketika nilai periode pengiriman data bernilai 1000ms atau satu detik paket data yang diterima terkadang menumpuk dalam satu waktu yang sama . Untuk hasil keseluruhan data hasil pengiriman terdapat pada lampiran .

Untuk skema pengujian kedua yaitu dari titik K menuju ke titik B memiliki nilai *delay* pengiriman yang kecil dibanding skema lain. Ini disebabkan data yang dikirim banyak yang tidak sampai ke tujuan, sehingga nilai total *delay* yang dihasilkan tiap masing – masing periode hanya sedikit. Hambatan (tembok) yang lebih banyak membuat sinyal tidak bisa tersebar dengan baik menyebabkan paket data yang akan masuk sedikit ada yang gagal dikirim.

Untuk skema pengujian yang ketiga yaitu dari titik K menuju titik B1 tidak terjadi respon penerimaan. Data yang dikirim dari titik K tidak masuk ke titik B1 dikarena halangan tembok yang lebih banyak dari pada titik B. Selain itu jarak yang ditempuh lebih jauh dibandingkan dengan titik B sehingga data tidak sampai ke titik B1.

Dari semua skema pengujian data yang dilakukan terlihat bahwa waktu periode pengiriman yang cocok digunakan untuk sistem monitoring dan controlling ini yaitu dua detik. Ini terlihat dari keseluruhan skema yang dilakukan, nilai *delay* yang dihasilkan sama rata di masing- masing titik pengujian. Ini menunjukkan bahwa

paket yang dikirim dapat diterima dengan baik oleh kedua titik dibandingkan dengan paket yang gagal masuk hasil terlihat pada Tabel 5.4 dan 5.14.

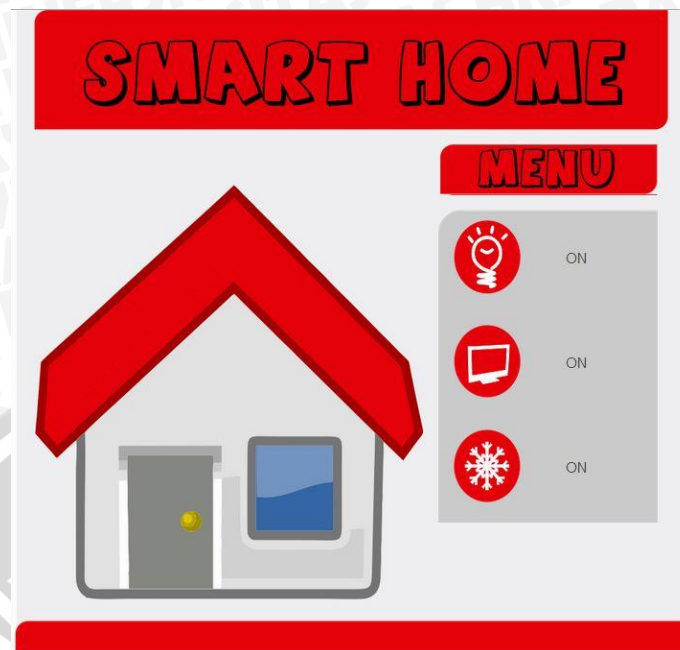
Data yang masuk ke masing- masing penerima lengkap walaupun ada beberapa data yang gagal diterima namun tidak sebanyak pada titik pengiriman yang lain dan juga tidak terjadi penumpukan data pada waktu yang sama, sehingga sistem dapat berjalan dengan baik. Disini juga terlihat bahwa waktu periode yang dibutuhkan untuk sistem ini tidak besar untuk mendapatkan hasil pengiriman yang baik .

5.4 Pengujian sistem secara keseluruhan

Pengujian sistem secara keseluruhan bertujuan untuk mengetahui apakah sistem yang dibangun sudah sesuai dengan perancangan dan implementasi atau tidak serta dapat berjalan dengan baik atau tidak.

Langkah pertama yang dilakukan yaitu mengkonfigurasi program pada masing – masing arduino. Selanjutnya menempatkan arduino koordinator dan laptop pada titik K sementara untuk arduino router satu dan arduino router dua pada titik A seperti Gambar 5.1. Untuk sumber daya arduino router satu dan dua menggunakan *powerbank* sementara arduino koordinator terhubung dengan laptop.

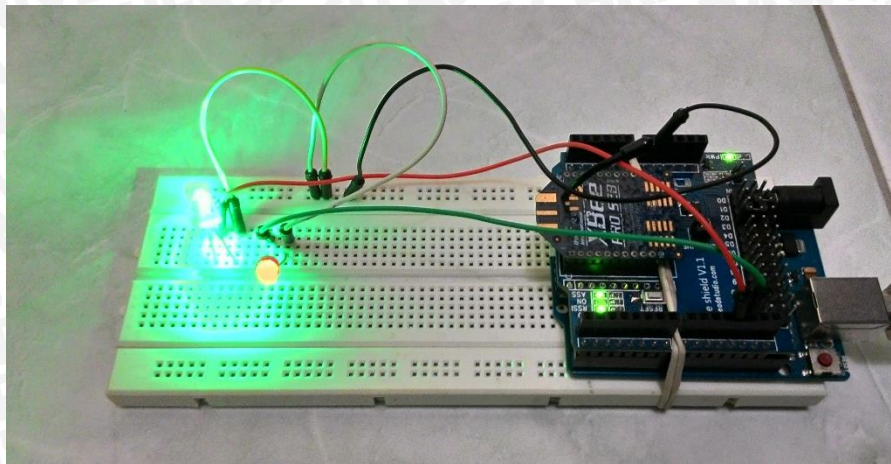
Setelah semua konfigurasi selesai selanjutnya penulis membuka *website* yang berada pada jaringan lokal untuk memberikan perintah LED menyala seperti pada Gambar 5.15



Gambar 5.15 Tampilan menu utama

Pada Gambar 5.15 terlihat bahwa *button* yang berisi nilai perintah untuk menyalakan LED telah diklik. Selanjutnya perintah yang ada akan diteruskan ke arduino koordinator melalui kabel serial. Pada arduino koordinator akan perintah yang masuk akan dicocokkan kembali dengan data perintah yang sudah tersedia. Apabila perintah sesuai maka arduino koordinator akan mengirim perintah melalui xbee menuju ke arduino router satu dan arduino router dua. Proses broadcast perintah ini akan berulang terus dengan *delay* pengiriman selama tiga detik tiap prosesnya. Nilai *delay* ini didapatkan dari hasil pengujian pengiriman data.

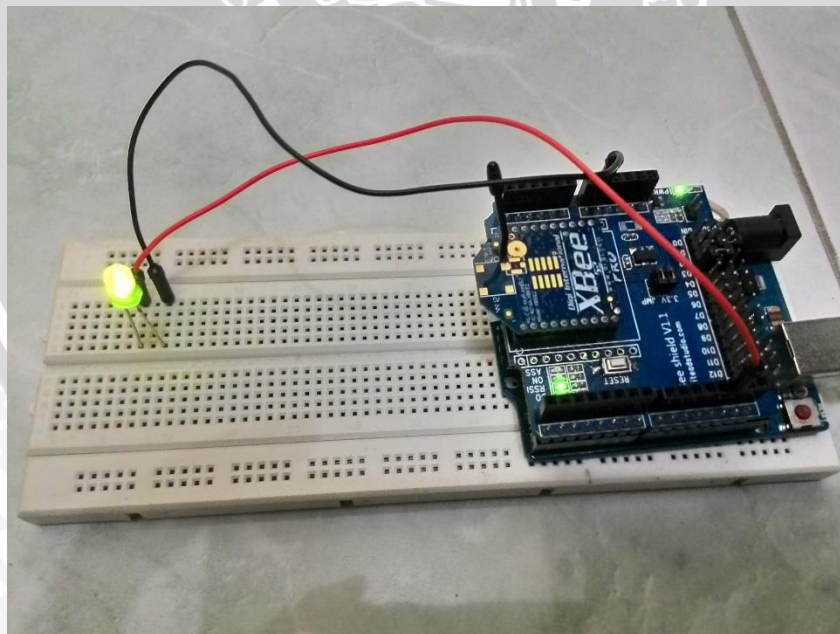
Perintah yang masuk di arduino router satu dan arduino router dua akan dicocokkan kembali pada data masing- masing arduino. Pada arduino router satu jika perintah sesuai dengan data yang tersedia maka LED akan menyala jika perintah yang masuk menyala .



Gambar 5. 16 LED pada arduino router satu menyala

Pada Gambar 5.15 terlihat bahwa ketika perintah yang masuk pada arduino router satu dan perintah sesuai dengan data yang ada maka LED akan menyala sesuai perintah yang masuk. Pada Gambar tersebut LED yang menyala yaitu LED satu dan LED dua.

Selanjutnya pada arduino router dua perintah yang masuk juga akan dicek. Jika sesuai dengan data yang ada maka LED akan menyala seperti Gambar 5.17



Gambar 5. 17 LED pada arduino router dua menyala

Setelah seluruh pengeujian sistem keseluruhan dilakukan. Hasil pengujian sistem keseluruhan terdapat pada Tabel 5.6

Tabel 5. 5 Hasil pengujian sistem

No	Perintah	Hasil
1	Perangkat arduino koordinator mampu menangkap perintah dari <i>website</i>	Sukses
2	Perangkat arduino koordinator dapat mengirim perintah ke perangkat arduino router satu dan dua	Sukses
3	Perangkat arduino router satu dan dua dapat menerima perintah dan mengeksekusi sesuai perintah	Sukses
4	Perangkat arduino router dua dapat menerima perintah manual ketika dalam mode otomatis sensor cahaya	Sukses

Analisis Pengujian Sistem secara Keseluruhan

Dari Tabel 5.5 terlihat bahwa sistem sudah berjalan sesuai dengan perancangan. Arduino koordinator mampu membaca perintah yang berasal dari *website* dan mengirim kembali menuju arduino router satu dan router dua.

Pada arduino router satu dapat merespon perintah sesuai dengan perancangan. Perintah yang masuk dicocokkan dengan data yang ada sehingga LED dapat menyala dan mati sesuai dengan perintah .

Pada arduino router dua mampu merespon data yang dikirim dari arduino koordinator dan mengeksekusi sesuai dengan perintah yang masuk.

BAB VI

PENUTUP

6.1 Kesimpulan

Setelah melakukan penelitian dan percobaan implementasi prototype smart home ini maka didapatkan kesimpulan bahwa :

1. Hasil Perancangan dan implementasi prototype sistem monitor dan controlling smart home berbasis embedded system sudah sesuai perancangan
2. Pada pengiriman langsung hasil pengiriman pengiriman dari titik k – A memiliki keberhasilan 60 % . Pada titik K – B memiliki keberhasilan 50 % dengan delay rata – rata sekitar 90 -96 detik.
3. Penambahan 1 buah node diantara 2 node pengiriman dan penerima data yang memiliki jarak berjauhan dapat membantu memperlancar komunikasi data yang putus.Hasil ini terlihat pada pengujian pengiriman data tidak langsung yang memiliki keberhasilan 100 %, Dengan delay rata – rata 1 menit.
4. Waktu periode yang baik untuk sistem prototype monitoring dan controlling smart home ini adalah 2 detik. Terlihat dari hasil pengujian nilai periode bahwa data yang dikirim dapat diterima utuh.

6.2 Saran

Saran yang dapat diberikan untuk pengembangan sistem ini lebih lanjut adalah :

1. Diharapkan pada penelitian selanjutnya ditambah penggunaan sensor yang lebih beragam.
2. Diharapkan pada sistem yang selanjutnya tidak hanya sebatas memantau kondisi hidup atau mati saja namu juga dapat memantau keamanan rumah.

DAFTAR PUSTAKA

- [ALI - 2010] Insan, Latifah,Aulia. 2010. Aplikasi Sistem Pengendali dan Manajemen Penjadwal Peralatan Listrik dalam Ruangan". Jurusan Teknik Elektro, Fakultas Teknik Universitas Diponegoro. Semarang
- [ALN - 2014] Nur Abdan, Alan. 2014 "Perancangan Dan Implementasi Protokol Komunikasi Uuntuk Home Automation". Program Studi Teknik Informatika, Program Teknologi Informasi Dan Ilmu Komputer Universitas Brawijaya. Malang
- [ARD - 2013] "Arduino", <http://www.arduino.cc>, diakses pada 20 november 2013
- [ART - 2013] "Pemrograman : Belajar Processing" <http://www.ART.com/> diakses 20 Oktober 2013
- [AZM - 2011] Muzakhim, Azam. 2011 "Telematri dan Telekontrol Antar Mikrokontroler Menggunakan Xbee Pro Wireless".Jurusan Teknik Elektro, Program Studi Teknik Telekomunikasi, Politeknik Negeri Malang
- [DHO - 2013] Herianto, Denny. 2013 "Embedded System Network Analyzer Pada Jaringan LAN". Program Studi Teknik Informatika, Program Teknologi Informasi Dan Ilmu Komputer Universitas Brawijaya. Malang
- [FZN - 2013] Nurahmadi, Fauzan. 2013. "Perancangan Sistem Control dan Monitoring Suhu Jarak Jauh Memanfaatkan Embeded System Berbasis Mikroprosesor W5100 dan AT 8535". Program Studi Sistem Informasi, Universitas Sari Mutiara. Medan
- [HSA - 2012] "Apa itu arduino " <http://hardi-santosa.blog.ugm.ac.id/2012/06/23/apa-itu-arduino/> diakses pada 20 november 2013
- [JLI - 2012] Ilham, Julian, Perancangan Sistem Pengendali Dan Penjadwal Lampu Ruangan Berdasarkan Database melalui Komunikasi Wireless Zigbee, Makalah Seminar Tugas Akhir, Semarang, 2012

- [KXP - 2013] "Konfigurasi XBee ", <http://www.Robosoccer.com>, diakses pada 13 November 2013
- [MVA - 2007] Varchola, Michal. 2007 "Zigbee Based Home Automation Wireless Sensor Network". Department of electronics and Multimedia Communication, Techncl University of Kosice
- [PRO - 2013] "Processing", <http://www.processing.com>, diakses pada 13 September 2013
- [QIS - 2012] Qisthi, Amelia Syarfina. 2012. "Perancangan Pendeteksi Banjir Jarak Jauh Menggunakan Sistem Komunikasi 802.15.4". Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro. Semarang
- [RIO - 2009] Peter Jon, Rio. 2009 "Rancang Bangun System Pemantau SUhu Ruangan Dengan Media Web Server Berbasis Mikrokontroler AT 89S51". Jurusan Teknik Elektro, Fakultas Teknik, Universitas Indonesia
- [SYT - 2012] "Sistem Tertanam", <http://goo.gl/ZaUDu7>, diakses pada 18 November 2013
- [WKM - 2012] Adelheid ,Andrea., Nst, Khairil. 2012. Buku Pintar Menguasai PHP & MySQL.Jakarta: Penerbit Mediakita
- [WKP - 2013] "Processing", <http://www.wikipedia.com>, diakses pada 20 Januari 2014
- [WPH - 2012] Peranginangin, Kasiman. 2006. Aplikasi Web dengan PHP dan MySql. Yogyakarta: Penerbit Andi Offset.
- [ZBB - 2012] XBee®/XBee-PRO® ZB RF Modules datasheet. Digi International Inc. 2010.
- [ZHT - 2012] "apa itu embedded system "<http://zahrotul-mfst10.web.unair.ac.id> diakses pada tanggal 10 february 2014

Lampiran 1 List Full Coding Arduino Coordinator

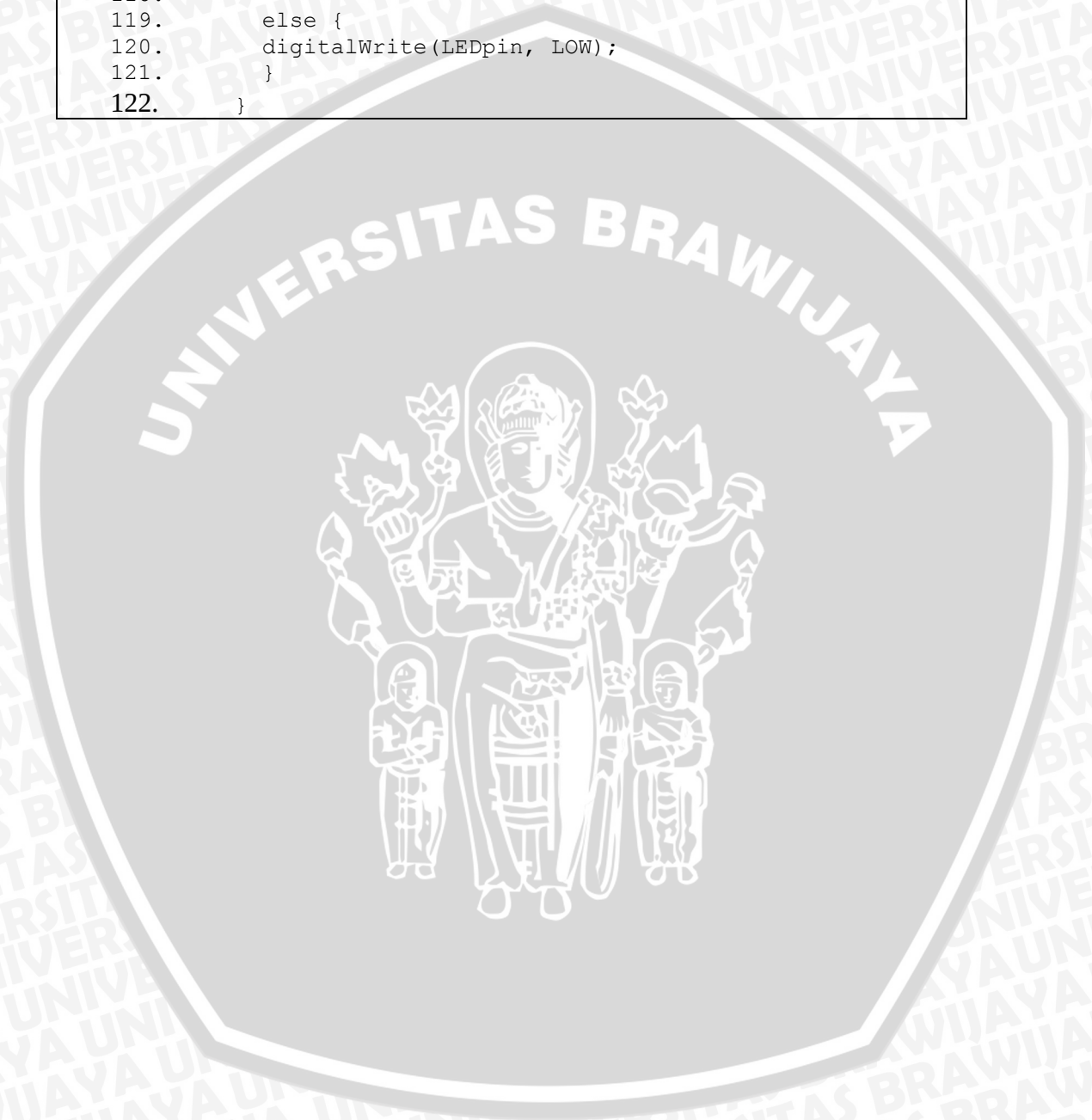
```
1. #include <SoftwareSerial.h>
2. SoftwareSerial mySerial(2,3);
3.
4.
5. //-----deklarasi variabel
6.
7. int perintah;
8. String data;
9. String dataku;
10.     boolean cek=false;
11.     boolean cek2=false;
12.     String dt[30];
13.     int i;
14.     int LEDpin = 13;
15.
16.     void setup() {
17.         // initialize serial communication:
18.         Serial.begin(9600);
19.         mySerial.begin(9600);
20.         pinMode(LEDpin,OUTPUT);
21.
22.     }
23.
24.     void loop() {
25.
26.         // aika ada komunikasi serial masuk
27.         if (Serial.available() > 0) {
28.             perintah = Serial.read();
29.
30.             //pencocokan perintah
31.             if (perintah == 'H') {
32.                 mySerial.print("@,0001,#"); // data yang
dikirim melalui xbee
33.             }
34.
35.             if (perintah == 'L') {
36.                 mySerial.print("@,0000,#");
37.             }
38.
39.             if (perintah == 'D') {
40.                 mySerial.print("@,0010,#");
41.             }
42.
43.             if (perintah == 'E') {
44.                 mySerial.print("@,0011,#");
45.             }
46.
47.             if (perintah == 'A') {
48.                 mySerial.print("@,0100,#");
49.             }
50.
51.             if (perintah == 'B') {
52.                 mySerial.print("@,0101,#");
53.             }
54.             //delay(1000);
```

```

55.     }
56.
57.
58.     if(mySerial.available()>0){
59.         char dataku = (char)mySerial.read();
60.         data += dataku;
61.
62.         //check header dan footer
63.         if (dataku == '@') {
64.             cek = true;
65.         }
66.
67.         if (dataku == '#') {
68.             cek2 = true;
69.         }
70.
71.     }
72.
73.
74.     //mulai cek data
75.     if((cek == true) && (cek2 == true)){
76.         //Serial.println(data);
77.         cekData();
78.         cek=false;
79.         cek2=false;
80.         data="";
81.         Serial.flush();
82.         mySerial.flush();
83.     }
84.
85.
86. }
87.
88. void cekData(){
89.     int a=0;
90.
91.     dt[a]="";
92.
93.     //proses cek data
94.     for(i=1;i<data.length();i++){
95.
96.         if ((data[i] == '#') || (data[i] == ','))
97.         {
98.             a++;
99.             dt[a]="";
100.        }
101.        else
102.        {
103.
104.            dt[a] = dt[a] + data[i];
105.        }
106.    }
107.
108.
109.
110.    String perintah = dt[1];
111.    //Serial.println(dt[1]);

```

```
112.  
113.     if(perintah == "1111"){  
114.         digitalWrite(LEDpin, HIGH);  
115.         delay(10000);  
116.         digitalWrite(LEDpin, LOW);  
117.     }  
118.  
119.     else {  
120.         digitalWrite(LEDpin, LOW);  
121.     }  
122. }
```



Lampiran 2 List Full Source code Arduino Router satu

```
1. #include <SoftwareSerial.h>
2. SoftwareSerial mySerial(2,3);
3.
4. int i;
5. String dt[50];
6. String data;
7. boolean cek=false;
8. boolean cek2=false;
9. int x;
10.     int LEDPin = 13;
11.     int LEDPin1 = 12;
12.
13.     //String inData;
14.     void setup(){
15.         Serial.begin(9600);
16.         mySerial.begin(9600);
17.         pinMode(LEDPin,OUTPUT);
18.         pinMode(LEDPin1,OUTPUT);
19.
20.
21.     }
22.     void loop()
23.     {
24.         if(mySerial.available()>0){
25.             char dataku = (char)mySerial.read();
26.             data += dataku;
27.
28.             if (dataku == '@') {
29.                 cek = true;
30.             }
31.
32.             if (dataku == '#') {
33.                 cek2 = true;
34.             }
35.         }
36.
37.         //mulai cek data
38.         if((cek == true) && (cek2 == true)){
39.             //Serial.println(data);
40.             cekData();
41.             cek=false;
42.             cek2=false;
43.             data="";
44.             //Serial.flush();
45.             mySerial.flush();
46.         }
47.     }
48.
49. }
50.
51.
52. //fungsi untuk cek data
53. void cekData(){
54.     int a=0;
55.
```

```
56.         dt[a]="";
57.
58.         //proses cek data
59.         for(i=1;i<data.length();i++){
60.
61.             if ((data[i] == '#') || (data[i] == ','))
62.             {
63.                 a++;
64.                 dt[a]="";
65.             }
66.             else
67.             {
68.
69.                 dt[a] = dt[a] + data[i];
70.             }
71.         }
72.
73.
74.
75.         String perintah = dt[1];
76.         //Serial.println(dt[1]);
77.
78.         if(perintah == "0001"){
79.             digitalWrite(LEDPin, HIGH);
80.             //mySerial.print("@,1111,#");
81.             delay(1000);
82.         }
83.
84.         if(perintah == "0000"){
85.             digitalWrite(LEDPin, LOW);
86.
87.         }
88.
89.         if(perintah == "0010"){
90.             digitalWrite(LEDPin1, HIGH);
91.             //Serial.print("nyala2");
92.         }
93.
94.         if(perintah == "0011"){
95.             digitalWrite(LEDPin1, LOW);
96.             //Serial.print("off");
97.         }
98.
99.         if(perintah == "0100"){
100.             //digitalWrite(LEDPin1, HIGH);
101.             mySerial.print("@,0100,#");
102.         }
103.
104.         if(perintah == "0101"){
105.             //digitalWrite(LEDPin1, LOW);
106.             mySerial.print("@,0101,#");
107.         }
108.         //delay(3000);
109.     }
110.
```

Lampiran 3 List Full Source code Arduino Router dua

```

1. #include <SoftwareSerial.h>
2. SoftwareSerial mySerial(2,3);
3.
4. int i;
5. String dt[50];
6. String data;
7. boolean cek=false;
8. boolean cek2=false;
9. int x;
10.     int LEDPin2 = 13;
11.     const int analogsensor = A5;
12.
13.
14.     //Deklarasi Variabel
15.     int analogValue = 0;
16.
17.     //String inData;
18.     void setup(){
19.         Serial.begin(9600);
20.         mySerial.begin(9600);
21.         pinMode (analogsensor, INPUT);
22.         pinMode(LEDPin2,OUTPUT);
23.
24.     }
25.
26.     void loop(){
27.         if(mySerial.available()>0){
28.             char dataku = (char)mySerial.read();
29.             data += dataku;
30.
31.             //check header dan footer
32.             if (dataku == '@') {
33.                 cek = true;
34.             }
35.
36.             if (dataku == '#') {
37.                 cek2 = true;
38.             }
39.
40.         }
41.
42.
43.         //mulai cek data
44.         if((cek == true) && (cek2 == true)){
45.             //Serial.println(data);
46.             cekData();
47.             cek=false;
48.             cek2=false;
49.             data="";
50.             Serial.flush();
51.             mySerial.flush();
52.         }
53.     }
54.     //fungsi untuk cek data
55.     void cekData(){

```

```
56.     int a=0;
57.
58.     dt[a]="";
59.
60.     //proses cek data
61.     for(i=1;i<data.length();i++){
62.
63.         if ((data[i] == '#' ) || (data[i] == ','))
64.         {
65.
66.             a++;
67.             dt[a]="";
68.         }
69.         else
70.         {
71.
72.             dt[a] = dt[a] + data[i];
73.         }
74.     }
75.     String perintah = dt[1];
76.     //Serial.println(dt[1]);
77.
78.     if(perintah == "0100"){
79.         digitalWrite(LEDPin2, HIGH);
80.         mySerial.print("@,1111,#");
81.
82.     }
83.
84.
85.     if(perintah == "0101"){
86.         digitalWrite(LEDPin2, LOW);
87.         mySerial.print("@,1111,#");
88.     }
89.
90. }
91.
92.
```

Lampiran 4 List Full Source code Processing

```

1.
2.
3. import processing.serial.*;
4. Serial port;
5.
6. void setup() {
7.
8.
9.   println(Serial.list());
10.   port = new Serial(this, Serial.list()[0], 9600);
    //untuk membuka port
11.
12. }
13. void draw() {
14.
15.   String onoroff[] =
    loadStrings("http://localhost/ard/aku.txt");// membaca
    lokasi file txt
16.   String onoroff2[]
    =loadStrings("http://localhost/ard/aku2.txt");
17.   String onoroff3[] =
    loadStrings("http://localhost/ard/aku3.txt");
18.
19.
20.   if (onoroff!=null)
21.   {
22.     if (onoroff[0].equals("1") == true) {
23.       println(" - ke 1");
24.       port.write('H'); // mengirim data H untuk
        perintah menyalakan
25.     }
26.     else {
27.
28.       println(" -matikan ke 1");
29.       port.write('L'); // mengirim data L untuk
        perintah mematikan
30.     }
31.   }
32.
33.   if (onoroff2!=null)
34.   {
35.     if (onoroff2[0].equals("2") == true) {
36.       println(" - ke 2");
37.       port.write('D');
38.     }
39.     else {
40.
41.       println(" - matikan ke 2");
42.       port.write('E');
43.     }
44.   }
45.
46.   if (onoroff3!=null)
47.   {
48.     if (onoroff3[0].equals("4") == true) {

```

```
49.         println(" - ke 3");
50.         port.write('A');
51.     }
52.     else {
53.
54.         println(" - matikan ke 3");
55.         port.write('B');
56.     }
57. }
58.     delay(1000); // set delay
59. }
```

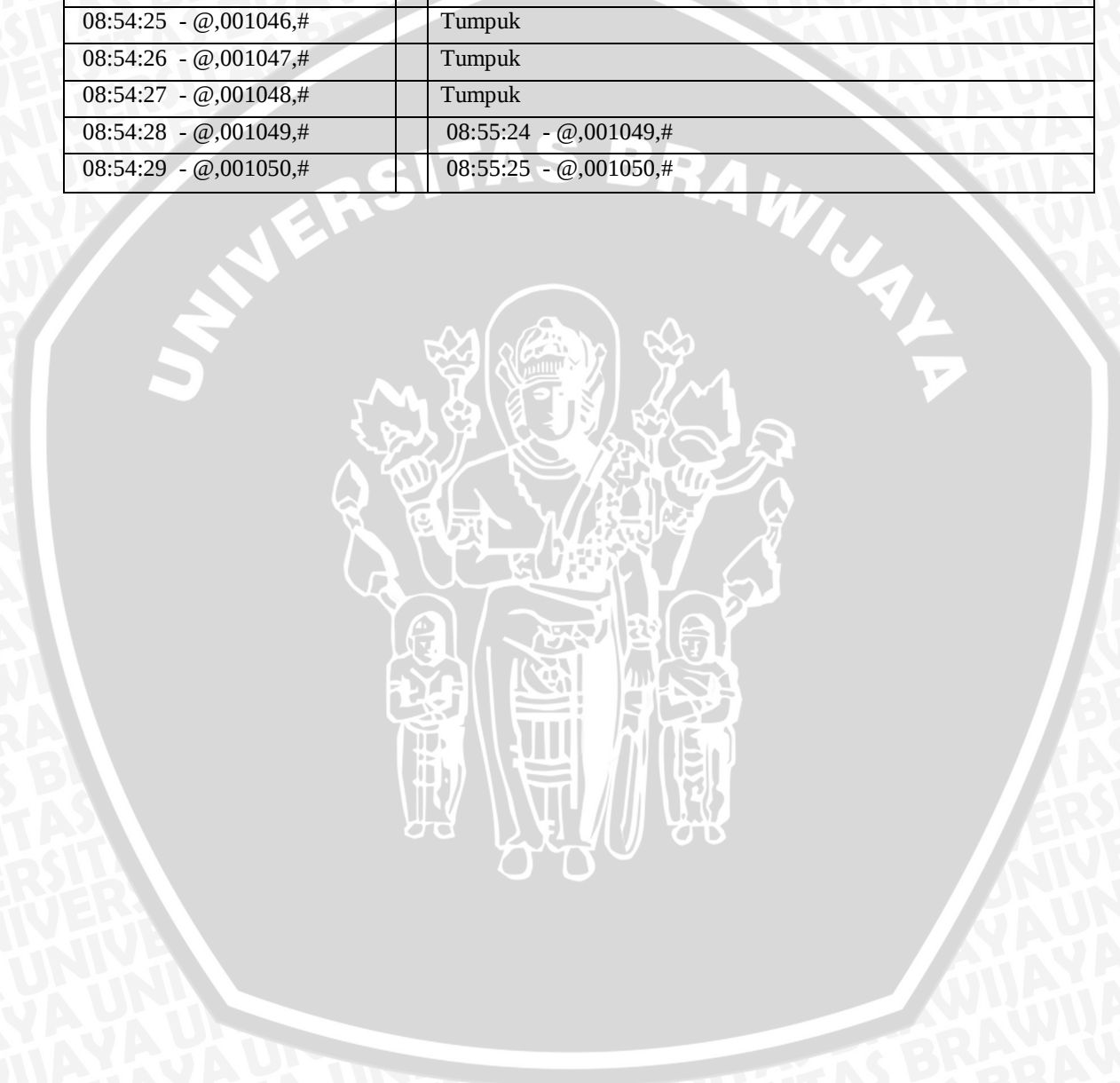
UNIVERSITAS BRAWIJAYA



Lampiran 5 Hasil Pengujian Langsung Titik K - A

K - A	
Dikirim	Diterima
08:53:40 - @,00101,#	08:54:38 - @,00101,#@,00102,#@,00103,#
08:53:41 - @,00102,#	Tumpuk
08:53:42 - @,00103,#	Tumpuk
08:53:43 - @,00104,#	08:54:38 - @,00104,#
08:53:44 - @,00105,#	08:54:39 - @,00105,#
08:53:45 - @,00106,#	08:54:40 - @,00106,#
08:53:46 - @,00107,#	08:54:41 - @,00107,#
08:53:47 - @,00108,#	08:54:43 - @,00108,#
08:53:48 - @,00109,#	08:54:44 - @,00109,#
08:53:49 - @,001010,#	08:54:48 - @,001010,#@,001011,#@,001012,#@,001013,#
08:53:50 - @,001011,#	Tumpuk
08:53:51 - @,001012,#	Tumpuk
08:53:52 - @,001013,#	Tumpuk
08:53:53 - @,001014,#	08:54:49 - @,001014,#
08:53:54 - @,001015,#	08:54:53 - @,001015,#@,001016,#@,001017,#@,001018,#
08:53:55 - @,001016,#	Tumpuk
08:53:56 - @,001017,#	Tumpuk
08:53:57 - @,001018,#	Tumpuk
08:53:58 - @,001019,#	08:54:53 - @,001019,#
08:53:59 - @,001020,#	08:54:54 - @,001020,#
08:54:00 - @,001021,#	08:54:55 - @,001021,#
08:54:01 - @,001022,#	08:54:56 - @,001022,#
08:54:02 - @,001023,#	08:54:58 - @,001023,#
08:54:03 - @,001024,#	08:54:59 - @,001024,#
08:54:04 - @,001025,#	08:55:03 - @,001025,#@,001026,#@,001027,#@,001028,#
08:54:05 - @,001026,#	Tumpuk
08:54:06 - @,001027,#	Tumpuk
08:54:07 - @,001028,#	Tumpuk
08:54:08 - @,001029,#	08:55:04 - @,001029,#
08:54:09 - @,001030,#	08:55:08 - @,001030,#@,001031,#@,001032,#@,001033,#
08:54:10 - @,001031,#	Tumpuk
08:54:11 - @,001032,#	Tumpuk
08:54:12 - @,001033,#	Tumpuk
08:54:13 - @,001034,#	08:55:08 - @,001034,#
08:54:14 - @,001035,#	08:55:09 - @,001035,#
08:54:15 - @,001036,#	08:55:10 - @,001036,#
08:54:16 - @,001037,#	08:55:11 - @,001037,#
08:54:17 - @,001038,#	08:55:13 - @,001038,#

08:54:18 - @,001039,#	08:55:14 - @,001039,#
08:54:19 - @,001040,#	08:55:18 - @,001040,#@,001041,#@,001042,#@,001043,#
08:54:20 - @,001041,#	Tumpuk
08:54:21 - @,001042,#	Tumpuk
08:54:22 - @,001043,#	Tumpuk
08:54:23 - @,001044,#	08:55:19 - @,001044,#
08:54:24 - @,001045,#	08:55:23 - @,001045,#@,001046,#@,001047,#@,001048,#
08:54:25 - @,001046,#	Tumpuk
08:54:26 - @,001047,#	Tumpuk
08:54:27 - @,001048,#	Tumpuk
08:54:28 - @,001049,#	08:55:24 - @,001049,#
08:54:29 - @,001050,#	08:55:25 - @,001050,#



Lampiran 6 Hasil Pengujian Langsung Titik K – B

K - B	
Dikirim	Diterima
09:22:52 - @,00101,#	Gagal
09:22:53 - @,00102,#	Gagal
09:22:54 - @,00103,#	Gagal
09:22:55 - @,00104,#	Gagal
09:22:56 - @,00105,#	09:23:52 - @,00105,#
09:22:57 - @,00106,#	09:23:53 - @,00106,#
09:22:58 - @,00107,#	09:23:54 - @,00107,#
09:22:59 - @,00108,#	Gagal
09:23:00 - @,00109,#	Gagal
09:23:01 - @,001010,#	Gagal
09:23:02 - @,001011,#	09:24:00 - @,001011,#@,001012,#@,001013,#
09:23:03 - @,001012,#	Tumpuk
09:23:04 - @,001013,#	Tumpuk
09:23:05 - @,001014,#	09:24:05 - @,001014,#@,001015,#@,001016,#@,001017,#@,001018,#
09:23:06 - @,001015,#	Tumpuk
09:23:07 - @,001016,#	Tumpuk
09:23:08 - @,001017,#	Tumpuk
09:23:09 - @,001018,#	Tumpuk
09:23:10 - @,001019,#	09:24:05 - @,001019,#
09:23:11 - @,001020,#	09:24:06 - @,001020,#
09:23:12 - @,001021,#	09:24:07 - @,001021,#
09:23:13 - @,001022,#	09:24:08 - @,001022,#
09:23:14 - @,001023,#	09:24:10 - @,001023,#
09:23:15 - @,001024,#	09:24:10 - @,001024,#
09:23:16 - @,001025,#	09:24:11 - @,001025,#
09:23:17 - @,001026,#	09:24:15 - @,001026,#@,001027,#@,001028,#
09:23:18 - @,001027,#	Tumpuk
09:23:19 - @,001028,#	Tumpuk
09:23:20 - @,001029,#	09:24:20 - @,001029,#@,001030,#@,001031,#@,001032,#@,001033,#
09:23:21 - @,001030,#	Tumpuk
09:23:22 - @,001031,#	Tumpuk
09:23:23 - @,001032,#	Tumpuk
09:23:24 - @,001033,#	Tumpuk
09:23:25 - @,001034,#	09:24:21 - @,001034,#
09:23:26 - @,001035,#	09:24:22 - @,001035,#
09:23:27 - @,001036,#	09:24:23 - @,001036,#
09:23:28 - @,001037,#	09:24:24 - @,001037,#

09:23:29 - @,001038,#		09:24:25 - @,001038,#
09:23:30 - @,001039,#		09:24:26 - @,001039,#
09:23:31 - @,001040,#		09:24:27 - @,001040,#
09:23:32 - @,001041,#		09:24:30 - @,001041,#@,001042,#@,001043,#
09:23:33 - @,001042,#		Tumpuk
09:23:34 - @,001043,#		Tumpuk
09:23:35 - @,001044,#		09:24:35 @,001044,#@,001045,#@,001046,#@,001047,#@,001048,#
09:23:36 - @,001045,#		Tumpuk
09:23:37 - @,001046,#		Tumpuk
09:23:38 - @,001047,#		Tumpuk
09:23:39 - @,001048,#		Tumpuk
09:23:40 - @,001049,#		09:24:36 - @,001049,#
09:23:41 - @,001050,#		09:24:37 - @,001050,#



Lampiran 7 Hasil Pengujian Pengiriman Tidak Langsung

K - A - B1	
Dikirim	Diterima
13:22:16 @,0010,#	13:22:25 nyala
13:22:17 @,0010,#	13:22:31 nyala
13:22:18 @,0010,#	13:22:35 nyala
13:22:19 @,0010,#	13:22:36 nyala
13:22:21 @,0010,#	13:22:41 nyala
13:22:21 @,0010,#	13:22:45 nyala
13:22:22 @,0010,#	13:22:47 nyala
13:22:23 @,0010,#	13:22:51 nyala
13:22:25 @,0010,#	13:22:52 nyala
13:22:26 @,0010,#	13:22:55 nyala
13:22:27 @,0010,#	13:23:01 nyala
13:22:28 @,0010,#	13:23:06 nyala
13:22:29 @,0010,#	13:23:07 nyala
13:22:30 @,0010,#	13:23:12 nyala
13:22:31 @,0010,#	13:23:17 nyala
13:22:32 @,0010,#	13:23:21 nyala
13:22:33 @,0010,#	13:23:22 nyala
13:22:34 @,0010,#	13:23:26 nyala
13:22:35 @,0010,#	13:23:31 nyala
13:22:36 @,0010,#	13:23:32 nyala
13:22:37 @,0010,#	13:23:37 nyala
13:22:38 @,0010,#	13:23:41 nyala
13:22:39 @,0010,#	13:23:46 nyala
13:22:40 @,0010,#	13:23:51 nyala
13:22:41 @,0010,#	13:23:51 nyala
13:22:42 @,0010,#	13:23:52 nyala
13:22:43 @,0010,#	13:23:56 nyala
13:22:44 @,0010,#	13:24:01 nyala
13:22:45 @,0010,#	13:24:02 nyala
13:22:46 @,0010,#	13:24:07 nyala
13:22:47 @,0010,#	13:24:07 nyala
13:22:48 @,0010,#	13:24:11 nyala
13:22:49 @,0010,#	13:24:17 nyala
13:22:50 @,0010,#	13:24:21 nyala
13:22:51 @,0010,#	13:24:21 nyala
13:22:52 @,0010,#	13:24:22 nyala
13:22:53 @,0010,#	13:24:27 nyala
13:22:54 @,0010,#	13:24:32 nyala

13:22:55 @,0010,#	13:24:36 nyala
13:22:56 @,0010,#	13:24:37 nyala
13:22:57 @,0010,#	13:24:41 nyala
13:22:58 @,0010,#	13:24:46 nyala
13:22:59 @,0010,#	13:24:52 nyala
13:23:00 @,0010,#	13:24:56 nyala
13:23:01 @,0010,#	13:25:01 nyala
13:23:02 @,0010,#	13:25:06 nyala
13:23:03 @,0010,#	13:25:07 nyala
13:23:04 @,0010,#	13:25:07 nyala
13:23:05 @,0010,#	13:25:12 nyala
13:23:06 @,0010,#	13:25:16 nyala



Lampiran 8 Hasil pengujian Nilai Periode Ideal Titik K – A

Data delay 1 detik	K A		
Dikirim	Diterima	Delay (Detik)	
08:53:40 - @,00101,#	08:54:38 - @,00101,#@,00102,#@,00103,#		98
08:53:41 - @,00102,#	Tumpuk	Tumpuk	
08:53:42 - @,00103,#	Tumpuk	Tumpuk	
08:53:43 - @,00104,#	08:54:38 - @,00104,#	95	
08:53:44 - @,00105,#	08:54:39 - @,00105,#	95	
08:53:45 - @,00106,#	08:54:40 - @,00106,#	95	
08:53:46 - @,00107,#	08:54:41 - @,00107,#	95	
08:53:47 - @,00108,#	08:54:43 - @,00108,#	96	
08:53:48 - @,00109,#	08:54:44 - @,00109,#	96	
08:53:49 - @,001010,#	08:54:48 - @,001010,#@,001011,#@,001012,#@,001013,#		99
08:53:50 - @,001011,#	Tumpuk	Tumpuk	
08:53:51 - @,001012,#	Tumpuk	Tumpuk	
08:53:52 - @,001013,#	Tumpuk	Tumpuk	
08:53:53 - @,001014,#	08:54:49 - @,001014,#	96	
08:53:54 - @,001015,#	08:54:53 - @,001015,#@,001016,#@,001017,#@,001018,#		99
08:53:55 - @,001016,#	Tumpuk	Tumpuk	
08:53:56 - @,001017,#	Tumpuk	Tumpuk	
08:53:57 - @,001018,#	Tumpuk	Tumpuk	
08:53:58 - @,001019,#	08:54:53 - @,001019,#	95	
08:53:59 - @,001020,#	08:54:54 - @,001020,#	95	
08:54:00 - @,001021,#	08:54:55 - @,001021,#	55	
08:54:01 - @,001022,#	08:54:56 - @,001022,#	55	
08:54:02 - @,001023,#	08:54:58 - @,001023,#	56	
08:54:03 - @,001024,#	08:54:59 - @,001024,#	56	
08:54:04 - @,001025,#	08:55:03 - @,001025,#@,001026,#@,001027,#@,001028,#		99
08:54:05 - @,001026,#	Tumpuk	Tumpuk	
08:54:06 - @,001027,#	Tumpuk	Tumpuk	
08:54:07 - @,001028,#	Tumpuk	Tumpuk	
08:54:08 - @,001029,#	08:55:04 - @,001029,#	96	
08:54:09 - @,001030,#	08:55:08 - @,001030,#@,001031,#@,001032,#@,001033,#		99
08:54:10 - @,001031,#	Tumpuk	Tumpuk	
08:54:11 - @,001032,#	Tumpuk	Tumpuk	
08:54:12 - @,001033,#	Tumpuk	Tumpuk	
08:54:13 - @,001034,#	08:55:08 - @,001034,#	95	
08:54:14 - @,001035,#	08:55:09 - @,001035,#	95	
08:54:15 - @,001036,#	08:55:10 - @,001036,#	95	
08:54:16 - @,001037,#	08:55:11 - @,001037,#	95	
08:54:17 - @,001038,#	08:55:13 - @,001038,#	96	

08:54:18 - @,001039,#	08:55:14 - @,001039,#	96		
08:54:19 - @,001040,#	08:55:18 @,001040,#@,001041,#@,001042,#@,001043,#		99	
08:54:20 - @,001041,#	Tumpuk	Tumpuk		
08:54:21 - @,001042,#	Tumpuk	Tumpuk		
08:54:22 - @,001043,#	Tumpuk	Tumpuk		
08:54:23 - @,001044,#	08:55:19 - @,001044,#	96		
08:54:24 - @,001045,#	08:55:23 - @,001045,#@,001046,#@,001047,#@,001048,#		99	
08:54:25 - @,001046,#	Tumpuk	Tumpuk		
08:54:26 - @,001047,#	Tumpuk	Tumpuk		
08:54:27 - @,001048,#	Tumpuk	Tumpuk		
08:54:28 - @,001049,#	08:55:24 - @,001049,#	96		
08:54:29 - @,001050,#	08:55:25 - @,001050,#	96		
		2728		
		50	54,56	

Data delay 2 detik K - A			
Dikirim	Diterima	Delay (Detik)	
08:57:45 - @,00101,#	Gagal	Gagal	
08:57:47 - @,00102,#	Gagal	Gagal	
08:57:49 - @,00103,#	Gagal	Gagal	
08:57:51 - @,00104,#	Gagal	Gagal	
08:57:53 - @,00105,#	08:58:49 - @,00105,#	96	
08:57:55 - @,00106,#	08:58:51 - @,00106,#	96	
08:57:57 - @,00107,#	08:58:53 - @,00107,#	96	
08:57:59 - @,00108,#	08:58:55 - @,00108,#	96	
08:58:01 - @,00109,#	08:58:57 - @,00109,#	56	
08:58:03 - @,001010,#	08:58:59 - @,001010,#	56	
08:58:05 - @,001011,#	08:59:01 - @,001011,#	96	
08:58:07 - @,001012,#	08:59:03 - @,001012,#	96	
08:58:09 - @,001013,#	08:59:05 - @,001013,#	96	
08:58:11 - @,001014,#	08:59:07 - @,001014,#	96	
08:58:13 - @,001015,#	08:59:09 - @,001015,#	96	
08:58:15 - @,001016,#	08:59:11 - @,001016,#	96	
08:58:17 - @,001017,#	08:59:13 - @,001017,#	96	
08:58:19 - @,001018,#	08:59:15 - @,001018,#	96	
08:58:21 - @,001019,#	08:59:17 - @,001019,#	96	
08:58:23 - @,001020,#	08:59:19 - @,001020,#	96	
08:58:25 - @,001021,#	08:59:21 - @,001021,#	96	
08:58:27 - @,001022,#	08:59:23 - @,001022,#	96	
08:58:29 - @,001023,#	08:59:25 - @,001023,#	96	
08:58:31 - @,001024,#	08:59:27 - @,001024,#	96	

08:58:33 - @,001025,#	08:59:29 - @,001025,#	96	
08:58:35 - @,001026,#	08:59:31 - @,001026,#	96	
08:58:37 - @,001027,#	08:59:33 - @,001027,#	96	
08:58:39 - @,001028,#	08:59:35 - @,001028,#	96	
08:58:41 - @,001029,#	08:59:37 - @,001029,#	96	
08:58:43 - @,001030,#	08:59:39 - @,001030,#	96	
08:58:46 - @,001031,#	08:59:41 - @,001031,#	95	
08:58:48 - @,001032,#	08:59:43 - @,001032,#	95	
08:58:50 - @,001033,#	08:59:45 - @,001033,#	95	
08:58:52 - @,001034,#	08:59:47 - @,001034,#	95	
08:58:54 - @,001035,#	08:59:49 - @,001035,#	95	
08:58:56 - @,001036,#	08:59:51 - @,001036,#	95	
08:58:58 - @,001037,#	08:59:53 - @,001037,#	95	
08:59:00 - @,001038,#	08:59:55 - @,001038,#	55	
08:59:02 - @,001039,#	08:59:57 - @,001039,#	55	
08:59:04 - @,001040,#	08:59:59 - @,001040,#	55	
08:59:06 - @,001041,#	09:00:01 - @,001041,#	95	
08:59:08 - @,001042,#	09:00:03 - @,001042,#	95	
08:59:10 - @,001043,#	09:00:05 - @,001043,#	95	
08:59:12 - @,001044,#	09:00:07 - @,001044,#	95	
08:59:14 - @,001045,#	09:00:09 - @,001045,#	95	
08:59:16 - @,001046,#	09:00:11 - @,001046,#	95	
08:59:18 - @,001047,#	09:00:13 - @,001047,#	95	
08:59:20 - @,001048,#	09:00:15 - @,001048,#	95	
08:59:22 - @,001049,#	09:00:17 - @,001049,#	95	
08:59:24 - @,001050,#	08:59:19 - @,001050,#	95	
		4196	
		46	91,21739

Data delay 3 detik K - A			
Dikirim	Diterima	Delay (Detik)	
09:00:37 - @,00101,#	Gagal	Gagal	
09:00:40 - @,00102,#	09:01:35 - @,00102,#	95	
09:00:43 - @,00103,#	09:01:38 - @,00103,#	95	
09:00:46 - @,00104,#	09:01:41 - @,00104,#	95	
09:00:49 - @,00105,#	09:01:44 - @,00105,#	95	
09:00:52 - @,00106,#	09:01:47 - @,00106,#	95	
09:00:55 - @,00107,#	09:01:50 - @,00107,#	95	
09:00:58 - @,00108,#	09:01:53 - @,00108,#	95	
09:01:01 - @,00109,#	09:01:56 - @,00109,#	55	
09:01:04 - @,001010,#	09:01:59 - @,001010,#	55	

09:01:07 - @,001011,#	09:02:02 - @,001011,#	95	
09:01:10 - @,001012,#	09:02:05 - @,001012,#	95	
09:01:13 - @,001013,#	09:02:08 - @,001013,#	95	
09:01:16 - @,001014,#	09:02:11 - @,001014,#	95	
09:01:19 - @,001015,#	09:02:14 - @,001015,#	95	
09:01:22 - @,001016,#	09:02:17 - @,001016,#	95	
09:01:25 - @,001017,#	09:02:20 - @,001017,#	95	
09:01:28 - @,001018,#	09:02:23 - @,001018,#	95	
09:01:31 - @,001019,#	09:02:26 - @,001019,#	95	
09:01:34 - @,001020,#	09:02:29 - @,001020,#	95	
09:01:37 - @,001021,#	09:02:32 - @,001021,#	95	
09:01:40 - @,001022,#	09:02:35 - @,001022,#	95	
09:01:43 - @,001023,#	09:02:38 - @,001023,#	95	
09:01:46 - @,001024,#	09:02:41 - @,001024,#	95	
09:01:49 - @,001025,#	09:02:44 - @,001025,#	95	
09:01:52 - @,001026,#	09:02:47 - @,001026,#	95	
09:01:55 - @,001027,#	09:02:50 - @,001027,#	95	
09:01:58 - @,001028,#	09:02:53 - @,001028,#	95	
09:02:01 - @,001029,#	09:02:56 - @,001029,#	55	
09:02:04 - @,001030,#	09:02:59 - @,001030,#	55	
09:02:07 - @,001031,#	09:03:02 - @,001031,#	95	
09:02:10 - @,001032,#	09:03:05 - @,001032,#	95	
09:02:13 - @,001033,#	09:03:08 - @,001033,#	95	
09:02:16 - @,001034,#	09:03:11 - @,001034,#	95	
09:02:19 - @,001035,#	09:03:15 - @,001035,#	96	
09:02:22 - @,001036,#	09:03:18 - @,001036,#	96	
09:02:25 - @,001037,#	09:03:21 - @,001037,#	96	
09:02:28 - @,001038,#	09:03:24 - @,001038,#	96	
09:02:31 - @,001039,#	09:03:27 - @,001039,#	96	
09:02:34 - @,001040,#	09:03:30 - @,001040,#	96	
09:02:37 - @,001041,#	09:03:33 - @,001041,#	96	
09:02:40 - @,001042,#	09:03:36 - @,001042,#	96	
09:02:43 - @,001043,#	09:03:39 - @,001043,#	96	
09:02:46 - @,001044,#	09:03:42 - @,001044,#	96	
09:02:49 - @,001045,#	09:03:45 - @,001045,#	96	
09:02:52 - @,001046,#	09:03:48 - @,001046,#	96	
09:02:55 - @,001047,#	09:03:51 - @,001047,#	96	
09:02:58 - @,001048,#	09:03:54 - @,001048,#	96	
09:03:01 - @,001049,#	09:03:57 - @,001049,#	56	
09:03:04 - @,001050,#	09:04:00 - @,001050,#	96	
		4471	
		49	91,2449

Delay 4 detik K - A			
Dikirim	Diterima	Delay (Detik)	
09:04:39 - @,00101,#	Gagal	Gagal	
09:04:43 - @,00102,#	09:05:38 - @,00102,#	95	
09:04:47 - @,00103,#	09:05:42 - @,00103,#	95	
09:04:51 - @,00104,#	09:05:46 - @,00104,#	95	
09:04:55 - @,00105,#	09:05:50 - @,00105,#	95	
09:04:59 - @,00106,#	09:05:54 - @,00106,#	95	
09:05:03 - @,00107,#	09:05:58 - @,00107,#	55	
09:05:07 - @,00108,#	09:06:02 - @,00108,#	95	
09:05:11 - @,00109,#	09:06:06 - @,00109,#	95	
09:05:15 - @,001010,#	09:06:10 - @,001010,#	95	
09:05:19 - @,001011,#	09:06:14 - @,001011,#	95	
09:05:23 - @,001012,#	09:06:18 - @,001012,#	95	
09:05:27 - @,001013,#	09:06:22 - @,001013,#	95	
09:05:31 - @,001014,#	09:06:26 - @,001014,#	95	
09:05:35 - @,001015,#	09:06:30 - @,001015,#	95	
09:05:39 - @,001016,#	09:06:34 - @,001016,#	95	
09:05:43 - @,001017,#	09:06:38 - @,001017,#	95	
09:05:47 - @,001018,#	09:06:42 - @,001018,#	95	
09:05:51 - @,001019,#	09:06:46 - @,001019,#	95	
09:05:55 - @,001020,#	09:06:50 - @,001020,#	95	
09:05:59 - @,001021,#	09:06:54 - @,001021,#	95	
09:06:03 - @,001022,#	09:06:58 - @,001022,#	55	
09:06:07 - @,001023,#	09:07:02 - @,001023,#	95	
09:06:11 - @,001024,#	09:07:06 - @,001024,#	95	
09:06:15 - @,001025,#	09:07:10 - @,001025,#	95	
09:06:19 - @,001026,#	09:07:14 - @,001026,#	95	
09:06:23 - @,001027,#	09:07:18 - @,001027,#	95	
09:06:27 - @,001028,#	09:07:22 - @,001028,#	95	
09:06:31 - @,001029,#	09:07:26 - @,001029,#	95	
09:06:35 - @,001030,#	09:07:30 - @,001030,#	95	
09:06:39 - @,001031,#	09:07:35 - @,001031,#	96	
09:06:43 - @,001032,#	09:07:39 - @,001032,#	96	
09:06:47 - @,001033,#	09:07:43 - @,001033,#	96	
09:06:51 - @,001034,#	09:07:47 - @,001034,#	96	
09:06:55 - @,001035,#	09:07:51 - @,001035,#	96	
09:06:59 - @,001036,#	09:07:55 - @,001036,#	96	
09:07:03 - @,001037,#	09:07:59 - @,001037,#	56	
09:07:07 - @,001038,#	09:08:03 - @,001038,#	96	

09:07:11 - @,001039,#	09:08:07 - @,001039,#	96	
09:07:15 - @,001040,#	09:08:11 - @,001040,#	96	
09:07:19 - @,001041,#	09:08:15 - @,001041,#	96	
09:07:23 - @,001042,#	09:08:19 - @,001042,#	96	
09:07:27 - @,001043,#	09:08:23 - @,001043,#	96	
09:07:31 - @,001044,#	09:08:27 - @,001044,#	96	
09:07:35 - @,001045,#	09:08:31 - @,001045,#	96	
09:07:39 - @,001046,#	09:08:35 - @,001046,#	96	
09:07:43 - @,001047,#	09:08:39 - @,001047,#	96	
09:07:47 - @,001048,#	09:08:43 - @,001048,#	96	
09:07:51 - @,001049,#	09:08:47 - @,001049,#	96	
09:07:55 - @,001050,#	09:08:51 - @,001050,#	96	
		4555	
		49	92,95918

Delay 5 detik K - A			
Dikirim	Diterima	Delay (Detik)	
09:09:02 - @,00101,#	Gagal	Gagal	
09:09:07 - @,00102,#	09:10:02 - @,00102,#	95	
09:09:12 - @,00103,#	09:10:07 - @,00103,#	95	
09:09:17 - @,00104,#	09:10:12 - @,00104,#	95	
09:09:22 - @,00105,#	09:10:17 - @,00105,#	95	
09:09:27 - @,00106,#	09:10:22 - @,00106,#	95	
09:09:32 - @,00107,#	09:10:27 - @,00107,#	95	
09:09:37 - @,00108,#	09:10:32 - @,00108,#	95	
09:09:42 - @,00109,#	09:10:37 - @,00109,#	95	
09:09:47 - @,001010,#	09:10:42 - @,001010,#	95	
09:09:52 - @,001011,#	09:10:47 - @,001011,#	95	
09:09:57 - @,001012,#	09:10:52 - @,001012,#	95	
09:10:02 - @,001013,#	09:10:57 - @,001013,#	55	
09:10:07 - @,001014,#	09:11:02 - @,001014,#	95	
09:10:12 - @,001015,#	09:11:07 - @,001015,#	95	
09:10:17 - @,001016,#	09:11:12 - @,001016,#	95	
09:10:22 - @,001017,#	09:11:17 - @,001017,#	95	
09:10:27 - @,001018,#	09:11:22 - @,001018,#	95	
09:10:32 - @,001019,#	09:11:27 - @,001019,#	95	
09:10:37 - @,001020,#	09:11:32 - @,001020,#	95	
09:10:42 - @,001021,#	09:11:37 - @,001021,#	95	
09:10:47 - @,001022,#	09:11:42 - @,001022,#	95	
09:10:52 - @,001023,#	09:11:47 - @,001023,#	95	
09:10:57 - @,001024,#	09:11:53 - @,001024,#	96	

09:11:02 - @,001025,#	09:11:58 - @,001025,#	56	
09:11:07 - @,001026,#	09:12:03 - @,001026,#	96	
09:11:12 - @,001027,#	09:12:08 - @,001027,#	96	
09:11:17 - @,001028,#	09:12:13 - @,001028,#	96	
09:11:22 - @,001029,#	09:12:18 - @,001029,#	96	
09:11:27 - @,001030,#	09:12:23 - @,001030,#	96	
09:11:32 - @,001031,#	09:12:28 - @,001031,#	96	
09:11:37 - @,001032,#	09:12:33 - @,001032,#	96	
09:11:42 - @,001033,#	09:12:38 - @,001033,#	96	
09:11:47 - @,001034,#	09:12:43 - @,001034,#	96	
09:11:52 - @,001035,#	09:12:48 - @,001035,#	96	
09:11:57 - @,001036,#	09:12:53 - @,001036,#	96	
09:12:02 - @,001037,#	09:12:58 - @,001037,#	56	
09:12:07 - @,001038,#	09:13:03 - @,001038,#	96	
09:12:12 - @,001039,#	09:13:08 - @,001039,#	96	
09:12:17 - @,001040,#	09:13:13 - @,001040,#	96	
09:12:22 - @,001041,#	09:13:18 - @,001041,#	96	
09:12:27 - @,001042,#	09:13:23 - @,001042,#	96	
09:12:32 - @,001043,#	09:13:28 - @,001043,#	96	
09:12:37 - @,001044,#	09:13:33 - @,001044,#	96	
09:12:42 - @,001045,#	09:13:38 - @,001045,#	96	
09:12:47 - @,001046,#	09:13:43 - @,001046,#	96	
09:12:52 - @,001047,#	09:13:48 - @,001047,#	96	
09:12:57 - @,001048,#	09:13:53 - @,001048,#	96	
09:13:02 - @,001049,#	09:13:58 - @,001049,#	56	
09:13:07 - @,001050,#	09:14:03 - @,001050,#	96	
		4522	
		49	92,28571

Lampiran 9 Hasil Pengujian Nilai Periode Ideal K – B

Delay 1 detik K - B			
Dikirim	Diterima	Delay (detik)	
09:22:52 - @,00101,#	Gagal	Gagal	
09:22:53 - @,00102,#	Gagal	Gagal	
09:22:54 - @,00103,#	Gagal	Gagal	
09:22:55 - @,00104,#	Gagal	Gagal	
09:22:56 - @,00105,#	09:23:52 - @,00105,#	96	
09:22:57 - @,00106,#	09:23:53 - @,00106,#	96	
09:22:58 - @,00107,#	09:23:54 - @,00107,#	96	
09:22:59 - @,00108,#	Gagal	Tumpuk	
09:23:00 - @,00109,#	Gagal	Tumpuk	
09:23:01 - @,001010,#	Gagal	Tumpuk	
09:23:02 - @,001011,#	09:24:00 - @,001011,#@,001012,#@,001013,#	98	
09:23:03 - @,001012,#	Tumpuk	Tumpuk	
09:23:04 - @,001013,#	Tumpuk	Tumpuk	
09:23:05 - @,001014,#	09:24:05 - @,001014,#@,001015,#@,001016,#@,001017,#@,001018,#	100	
09:23:06 - @,001015,#	Tumpuk	Tumpuk	
09:23:07 - @,001016,#	Tumpuk	Tumpuk	
09:23:08 - @,001017,#	Tumpuk	Tumpuk	
09:23:09 - @,001018,#	Tumpuk	Tumpuk	
09:23:10 - @,001019,#	09:24:05 - @,001019,#	95	
09:23:11 - @,001020,#	09:24:06 - @,001020,#	95	
09:23:12 - @,001021,#	09:24:07 - @,001021,#	95	
09:23:13 - @,001022,#	09:24:08 - @,001022,#	95	
09:23:14 - @,001023,#	09:24:10 - @,001023,#	96	
09:23:15 - @,001024,#	09:24:10 - @,001024,#	95	
09:23:16 - @,001025,#	09:24:11 - @,001025,#	95	

09:23:17 @,001026,#	-	09:24:15 - @,001026,#@,001027,#@,001028,#	98	
09:23:18 @,001027,#	-	Tumpuk	Tumpuk	
09:23:19 @,001028,#	-	Tumpuk	Tumpuk	
09:23:20 @,001029,#	-	09:24:20 @,001029,#@,001030,#@,001031,#@,001032,#@,001033,#	100	
09:23:21 @,001030,#	-	Tumpuk	Tumpuk	
09:23:22 @,001031,#	-	Tumpuk	Tumpuk	
09:23:23 @,001032,#	-	Tumpuk	Tumpuk	
09:23:24 @,001033,#	-	Tumpuk	Tumpuk	
09:23:25 @,001034,#	-	09:24:21 - @,001034,#	96	
09:23:26 @,001035,#	-	09:24:22 - @,001035,#	96	
09:23:27 @,001036,#	-	09:24:23 - @,001036,#	96	
09:23:28 @,001037,#	-	09:24:24 - @,001037,#	96	
09:23:29 @,001038,#	-	09:24:25 - @,001038,#	96	
09:23:30 @,001039,#	-	09:24:26 - @,001039,#	96	
09:23:31 @,001040,#	-	09:24:27 - @,001040,#	96	
09:23:32 @,001041,#	-	09:24:30 - @,001041,#@,001042,#@,001043,#	98	
09:23:33 @,001042,#	-	Tumpuk	Tumpuk	
09:23:34 @,001043,#	-	Tumpuk	Tumpuk	
09:23:35 @,001044,#	-	09:24:35 @,001044,#@,001045,#@,001046,#@,001047,#@,001048,#	100	
09:23:36 @,001045,#	-	Tumpuk	Tumpuk	
09:23:37 @,001046,#	-	Tumpuk	Tumpuk	
09:23:38 @,001047,#	-	Tumpuk	Tumpuk	
09:23:39 @,001048,#	-	Tumpuk	Tumpuk	
09:23:40 @,001049,#	-	09:24:36 - @,001049,#	96	
09:23:41 @,001050,#	-	09:24:37 - @,001050,#	96	
			2412	
			25	96,48

Delay 2 detik K - B			
Dikirim	Diterima	Delay (Detik)	
09:25:21 - @,00101,#	gagal	Gagal	
09:25:23 - @,00102,#	09:26:19 - @,00102,#	96	
09:25:25 - @,00103,#	gagal	Gagal	
09:25:27 - @,00104,#	09:26:23 - @,00104,#	96	
09:25:29 - @,00105,#	09:26:25 - @,00105,#	96	
09:25:31 - @,00106,#	09:26:27 - @,00106,#	96	
09:25:34 - @,00107,#	09:26:29 - @,00107,#	95	
09:25:36 - @,00108,#	09:26:31 - @,00108,#	95	
09:25:38 - @,00109,#	09:26:33 - @,00109,#	95	
09:25:40 - @,001010,#	09:26:35 - @,001010,#	95	
09:25:42 - @,001011,#	09:26:37 - @,001011,#	95	
09:25:44 - @,001012,#	09:26:39 - @,001012,#	95	
09:25:46 - @,001013,#	09:26:41 - @,001013,#	95	
09:25:48 - @,001014,#	09:26:43 - @,001014,#	95	
09:25:50 - @,001015,#	09:26:45 - @,001015,#	95	
09:25:52 - @,001016,#	09:26:47 - @,001016,#	95	
09:25:54 - @,001017,#	09:26:49 - @,001017,#	95	
09:25:56 - @,001018,#	09:26:51 - @,001018,#	95	
09:25:58 - @,001019,#	09:26:53 - @,001019,#	95	
09:26:00 - @,001020,#	09:26:55 - @,001020,#	55	
09:26:02 - @,001021,#	09:26:57 - @,001021,#	55	
09:26:04 - @,001022,#	09:26:59 - @,001022,#	55	
09:26:06 - @,001023,#	09:27:01 - @,001023,#	95	
09:26:08 - @,001024,#	09:27:03 - @,001024,#	95	
09:26:10 - @,001025,#	09:27:05 - @,001025,#	95	
09:26:12 - @,001026,#	09:27:07 - @,001026,#	95	
09:26:14 - @,001027,#	09:27:09 - @,001027,#	95	
09:26:16 - @,001028,#	09:27:11 - @,001028,#	95	
09:26:18 - @,001029,#	09:27:13 - @,001029,#	95	
09:26:20 - @,001030,#	09:27:15 - @,001030,#	95	
09:26:22 - @,001031,#	09:27:17 - @,001031,#	95	
09:26:24 - @,001032,#	09:27:19 - @,001032,#	95	
09:26:26 - @,001033,#	09:27:21 - @,001033,#	95	
09:26:28 - @,001034,#	09:27:23 - @,001034,#	95	
09:26:30 - @,001035,#	09:27:25 - @,001035,#	95	
09:26:32 - @,001036,#	09:27:27 - @,001036,#	95	
09:26:34 - @,001037,#	09:27:29 - @,001037,#	95	
09:26:36 - @,001038,#	09:27:31 - @,001038,#	95	
09:26:38 - @,001039,#	09:27:33 - @,001039,#	95	
09:26:40 - @,001040,#	09:27:35 - @,001040,#	95	

09:26:42 - @,001041,#	09:27:37 - @,001041,#	95	
09:26:44 - @,001042,#	09:27:39 - @,001042,#	95	
09:26:46 - @,001043,#	09:27:41 - @,001043,#	95	
09:26:48 - @,001044,#	09:27:43 - @,001044,#	95	
09:26:50 - @,001045,#	09:27:45 - @,001045,#	95	
09:26:52 - @,001046,#	09:27:47 - @,001046,#	95	
09:26:54 - @,001047,#	09:27:49 - @,001047,#	95	
09:26:56 - @,001048,#	09:27:51 - @,001048,#	95	
09:26:58 - @,001049,#	09:27:53 - @,001049,#	95	
09:27:00 - @,001050,#	09:27:55 - @,001050,#	55	
		4404	
		48	91,75

Delay 3 detik K - B			
Dikirim	Diterima	Delay (Detik)	
09:28:27 - @,00101,#	Gagal	Gagal	
09:28:30 - @,00102,#	Gagal	Gagal	
09:28:33 - @,00103,#	09:29:29 - @,00103,#	96	
09:28:36 - @,00104,#	09:29:32 - @,00104,#	96	
09:28:39 - @,00105,#	09:29:35 - @,00105,#	96	
09:28:42 - @,00106,#	09:29:38 - @,00106,#	96	
09:28:45 - @,00107,#	09:29:41 - @,00107,#	96	
09:28:48 - @,00108,#	09:29:44 - @,00108,#	96	
09:28:51 - @,00109,#	09:29:47 - @,00109,#	96	
09:28:54 - @,001010,#	09:29:50 - @,001010,#	96	
09:28:57 - @,001011,#	09:29:53 - @,001011,#	96	
09:29:00 - @,001012,#	09:29:56 - @,001012,#	56	
09:29:03 - @,001013,#	09:29:59 - @,001013,#	56	
09:29:06 - @,001014,#	09:30:02 - @,001014,#	96	
09:29:09 - @,001015,#	09:30:05 - @,001015,#	96	
09:29:12 - @,001016,#	09:30:08 - @,001016,#	96	
09:29:15 - @,001017,#	09:30:11 - @,001017,#	96	
09:29:18 - @,001018,#	Gagal	Gagal	
09:29:21 - @,001019,#	09:30:17 - @,001019,#	96	
09:29:24 - @,001020,#	09:30:20 - @,001020,#	96	
09:29:27 - @,001021,#	09:30:23 - @,001021,#	96	
09:29:30 - @,001022,#	09:30:26 - @,001022,#	96	
09:29:33 - @,001023,#	Gagal	Gagal	
09:29:37 - @,001024,#	Gagal	Gagal	
09:29:40 - @,001025,#	09:30:35 - @,001025,#	95	
09:29:43 - @,001026,#	Gagal	Gagal	

09:29:46 - @,001027,#	09:30:42 - @,001027,#	96	
09:29:49 - @,001028,#	09:30:44 - @,001028,#	95	
09:29:52 - @,001029,#	09:30:47 - @,001029,#	95	
09:29:55 - @,001030,#	Gagal	Gagal	
09:29:58 - @,001031,#	09:30:53 - @,001031,#	95	
09:30:01 - @,001032,#	Gagal	Gagal	
09:30:04 - @,001033,#	Gagal	Gagal	
09:30:07 - @,001034,#	Gagal	Gagal	
09:30:10 - @,001035,#	Gagal	Gagal	
09:30:13 - @,001036,#	Gagal	Gagal	
09:30:16 - @,001037,#	Gagal	Gagal	
09:30:19 - @,001038,#	09:31:15 - @,001038,#	96	
09:30:22 - @,001039,#	Gagal	Gagal	
09:30:25 - @,001040,#	Gagal	Gagal	
09:30:28 - @,001041,#	Gagal	Gagal	
09:30:31 - @,001042,#	Gagal	Gagal	
09:30:34 - @,001043,#	Gagal	Gagal	
09:30:37 - @,001044,#	Gagal	Gagal	
09:30:40 - @,001045,#	Gagal	Gagal	
09:30:43 - @,001046,#	Gagal	Gagal	
09:30:46 - @,001047,#	Gagal	Gagal	
09:30:49 - @,001048,#	09:31:45 - @,001048,#	96	
09:30:52 - @,001049,#	09:31:47 - @,001049,#	95	
09:30:55 - @,001050,#	Gagal		
		2507	
		29	86,44828

Delay 4 detik K - B			
Dikirim	Diterima	Delay (Detik)	
09:38:23 - @,00101,#	Gagal	Gagal	
09:38:27 - @,00102,#	09:39:23 - @,00102,#	96	
09:38:31 - @,00103,#	09:39:27 - @,00103,#	96	
09:38:35 - @,00104,#	09:39:30 - @,00104,#	95	
09:38:39 - @,00105,#	09:39:34 - @,00105,#	95	
09:38:43 - @,00106,#	09:39:38 - @,00106,#	95	
09:38:47 - @,00107,#	09:39:43 - @,00107,#	96	
09:38:51 - @,00108,#	09:39:46 - @,00108,#	95	
09:38:55 - @,00109,#	09:39:51 - @,00109,#	96	
09:38:59 - @,001010,#	09:39:54 - @,001010,#	95	
09:39:03 - @,001011,#	09:39:58 - @,001011,#	55	
09:39:07 - @,001012,#	09:40:02 - @,001012,#	95	

09:39:11 - @,001013,#	09:40:07 - @,001013,#	96	
09:39:15 - @,001014,#	09:40:10 - @,001014,#	95	
09:39:19 - @,001015,#	09:40:14 - @,001015,#	95	
09:39:23 - @,001016,#	09:40:18 - @,001016,#	95	
09:39:27 - @,001017,#	09:40:22 - @,001017,#	95	
09:39:31 - @,001018,#	09:40:26 - @,001018,#	95	
09:39:35 - @,001019,#	09:40:30 - @,001019,#	95	
09:39:39 - @,001020,#	09:40:34 - @,001020,#	95	
09:39:43 - @,001021,#	09:40:38 - @,001021,#	95	
09:39:47 - @,001022,#	09:40:43 - @,001022,#	96	
09:39:51 - @,001023,#	09:40:47 - @,001023,#	96	
09:39:55 - @,001024,#	09:40:51 - @,001024,#	96	
09:39:59 - @,001025,#	09:40:55 - @,001025,#	96	
09:40:03 - @,001026,#	09:40:59 - @,001026,#	56	
09:40:07 - @,001027,#	09:41:03 - @,001027,#	96	
09:40:11 - @,001028,#	09:41:07 - @,001028,#	96	
09:40:15 - @,001029,#	Gagal	Gagal	
09:40:19 - @,001030,#	Gagal	Gagal	
09:40:23 - @,001031,#	Gagal	Gagal	
09:40:27 - @,001032,#	Gagal	Gagal	
09:40:31 - @,001033,#	Gagal	Gagal	
09:40:35 - @,001034,#	Gagal	Gagal	
09:40:39 - @,001035,#	Gagal	Gagal	
09:40:43 - @,001036,#	Gagal	Gagal	
09:40:47 - @,001037,#	09:41:43 - @,001037,#	96	
09:40:51 - @,001038,#	09:41:47 - @,001038,#	96	
09:40:55 - @,001039,#	09:41:52 - @,001039,#	97	
09:40:59 - @,001040,#	Gagal	Gagal	
09:41:03 - @,001041,#	09:41:59 - @,001041,#	56	
09:41:07 - @,001042,#	09:42:03 - @,001042,#	96	
09:41:11 - @,001043,#	09:42:07 - @,001043,#	96	
09:41:15 - @,001044,#	09:42:11 - @,001044,#	96	
09:41:19 - @,001045,#	Gagal	Gagal	
09:41:23 - @,001046,#	09:42:20 - @,001046,#	97	
09:41:27 - @,001047,#	Gagal	Gagal	
09:41:31 - @,001048,#	Gagal	Gagal	
09:41:35 - @,001049,#	Gagal	Gagal	
09:41:39 - @,001050,#	Gagal	Gagal	
		3227	
		35	92,2

Delay 5 detik			
Dikirim	Diterima	Delay (Detik)	
09:44:11 - @,00101,#	Gagal	Gagal	
09:44:16 - @,00102,#	09:45:12 - @,00102,#	96	
09:44:21 - @,00103,#	09:45:17 - @,00103,#	96	
09:44:26 - @,00104,#	09:45:22 - @,00104,#	96	
09:44:31 - @,00105,#	09:45:27 - @,00105,#	96	
09:44:36 - @,00106,#	09:45:32 - @,00106,#	96	
09:44:41 - @,00107,#	09:45:37 - @,00107,#	96	
09:44:46 - @,00108,#	09:45:42 - @,00108,#	96	
09:44:51 - @,00109,#	09:45:47 - @,00109,#	96	
09:44:56 - @,001010,#	09:45:52 - @,001010,#	96	
09:45:01 - @,001011,#	09:45:57 - @,001011,#	56	
09:45:06 - @,001012,#	09:46:02 - @,001012,#	96	
09:45:11 - @,001013,#	09:46:07 - @,001013,#	96	
09:45:16 - @,001014,#	09:46:12 - @,001014,#	96	
09:45:21 - @,001015,#	09:46:17 - @,001015,#	96	
09:45:26 - @,001016,#	Gagal	Gagal	
09:45:31 - @,001017,#	09:46:27 - @,001017,#	96	
09:45:36 - @,001018,#	Gagal	Gagal	
09:45:41 - @,001019,#	Gagal	Gagal	
09:45:46 - @,001020,#	09:46:42 - @,001020,#	96	
09:45:51 - @,001021,#	09:46:47 - @,001021,#	96	
09:45:57 - @,001022,#	09:46:52 - @,001022,#	95	
09:46:02 - @,001023,#	09:46:57 - @,001023,#	55	
09:46:07 - @,001024,#	09:47:02 - @,001024,#	95	
09:46:12 - @,001025,#	09:47:07 - @,001025,#	95	
09:46:17 - @,001026,#	09:47:12 - @,001026,#	95	
09:46:22 - @,001027,#	09:47:17 - @,001027,#	95	
09:46:27 - @,001028,#	09:47:22 - @,001028,#	95	
09:46:32 - @,001029,#	09:47:27 - @,001029,#	95	
09:46:37 - @,001030,#	09:47:32 - @,001030,#	95	
09:46:42 - @,001031,#	09:47:37 - @,001031,#	95	
09:46:47 - @,001032,#	09:47:42 - @,001032,#	95	
09:46:52 - @,001033,#	09:47:47 - @,001033,#	95	
09:46:57 - @,001034,#	09:47:52 - @,001034,#	95	
09:47:02 - @,001035,#	09:47:57 - @,001035,#	55	
09:47:07 - @,001036,#	09:48:02 - @,001036,#	95	
09:47:12 - @,001037,#	09:48:07 - @,001037,#	95	
09:47:17 - @,001038,#	09:48:12 - @,001038,#	95	
09:47:22 - @,001039,#	09:48:17 - @,001039,#	95	
09:47:27 - @,001040,#	09:48:22 - @,001040,#	95	

09:47:32 - @,001041,#	09:48:27 - @,001041,#	95	
09:47:37 - @,001042,#	09:48:32 - @,001042,#	95	
09:47:42 - @,001043,#	09:48:37 - @,001043,#	95	
09:47:47 - @,001044,#	09:48:42 - @,001044,#	95	
09:47:52 - @,001045,#	09:48:47 - @,001045,#	95	
09:47:57 - @,001046,#	09:48:52 - @,001046,#	95	
09:48:02 - @,001047,#	09:48:57 - @,001047,#	55	
09:48:07 - @,001048,#	Gagal	Gagal	
09:48:12 - @,001049,#	09:49:08 - @,001049,#	96	
09:48:17 - @,001050,#	09:49:12 - @,001050,#	95	
		4133	
		45	91,84444

