

**KOMUNIKASI PENGIRIMAN DATA DENGAN PROTOKOL ZIGBEE
MENGUNAKAN MEKANISME INTERNET CHECKSUM UNTUK
MENGUBAH PARAMETER MODUL ARDUINO**

SKRIPSI

**Diajukan untuk Memenuhi Persyaratan
Memperoleh Gelar Sarjana Komputer**



Disusun Oleh :

ALDIM IRFANI VIKRI

NIM. 0910683104

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER**

MALANG

2014

LEMBAR PERSETUJUAN

KOMUNIKASI PENGIRIMAN DATA DENGAN PROTOKOL ZIGBEE MENGUNAKAN MEKANISME INTERNET CHECKSUM UNTUK MENGUBAH PARAMETER MODUL ARDUINO

SKRIPSI

Untuk memenuhi sebagian persyaratan untuk mencapai gelar Sarjana Komputer



Disusun Oleh:

ALDIM IRFANI VIKRI

NIM. 0910683104

Skrripsi ini telah diperiksa dan disetujui oleh dosen pembimbing:

Dosen Pembimbing I

Dosen Pembimbing II

Sabriansyah Rizqika Akbar, ST., M.Eng.

NIP. 820809 06 1 1 0084

Eko Sakti P., S.Kom., M.Kom.

NIP. 860805 06 1 1 0252

LEMBAR PENGESAHAN

**KOMUNIKASI PENGIRIMAN DATA DENGAN PROTOKOL ZIGBEE
MENGUNAKAN MEKANISME INTERNET CHECKSUM UNTUK
MENGUBAH PARAMETER MODUL ARDUINO**

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh :

Aldim Irfani Vikri

NIM. 0910683104

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 13 Januari 2014

Penguji I

Penguji II

R. Arief Setyawan, ST., MT.
NIP. 19750819 199903 1 001

Ir. Heru Nurwarsito, M.Kom.
NIP. 196504021990021001

Penguji III

Achmad Basuki, ST.,M.MG.,Ph.D
NIP. 19741118 200312 1 002

Mengetahui

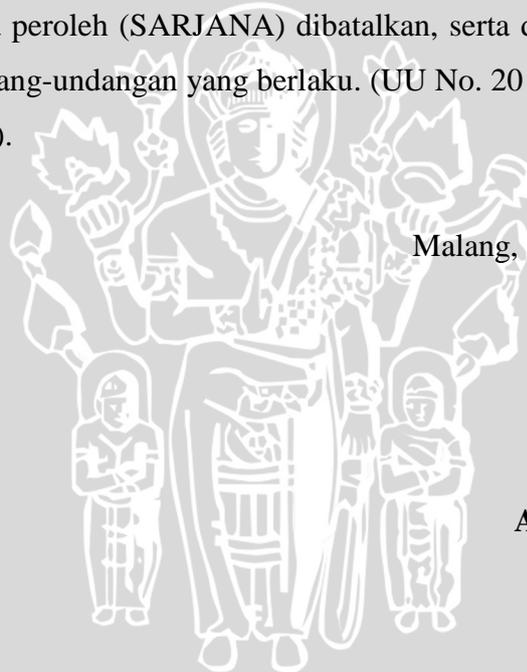
Ketua Program Studi Teknik Informatika

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

PERNYATAAN**ORISINALITAS SKRIPSI**

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

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



Malang, 27 Desember 2013

Mahasiswa,

Aldim Irfani Vikri

0910683104

KATA PENGANTAR

Puji Syukur kehadiran Tuhan Yang Maha Esa yang telah mencurahkan kasih dan rahmat, sehingga Proposal skripsi yang berjudul “ KOMUNIKASI PENGIRIMAN DATA DENGAN PROTOKOL ZIGBEE MENGGUNAKAN MEKANISME INTERNET CHECKSUM UNTUK MENGUBAH PARAMETER MODUL ARDUINO” ini dapat diselesaikan.

Dalam menyelesaikan skripsi ini, penulis telah banyak mendapat bantuan dari berbagai pihak. Pada kesempatan kali ini, penulis mengucapkan banyak terima kasih kepada:

1. Allah SWT atas rahmat dan hidayah yang telah diberikan.
2. Rasulullah Muhammad SAW, semoga shalawat serta salam selalu tercurah kepada beliau.
3. Kedua orang tua penulis, Ayahanda Munidzom Soetikgiyoto, S.E dan Ibunda Siti Hadijah S.H yang selalu tidak lepas dari do'a dan harapan untuk terselesaikannya skripsi ini dan terus memberikan dorongan moral, material dan kasih sayangnya tiada akhir.
4. Kakak penulis Priaz Rizka Fardhani S.Psi serta Adik penulis Tito Syaqui Ramadhan yang senantiasa memberi do'a dan motivasi. Tak lupa juga penulis berterima kasih untuk segenap keluarga besar penulis.
5. Bapak Drs. Marji, MT. dan Bapak Issa Arwani S.Kom, MSc. selaku Ketua Program dan Sekretaris Program Studi Teknik Informatika, segenap Bapak/Ibu Dosen dan seluruh Staff Teknik Informatika Universitas Brawijaya.
6. Bapak Sabriansyah Rizqika Akbar, ST., M.Eng. selaku Dosen Pembimbing I yang telah banyak memberikan bimbingan, masukan dan arahan dalam penyusunan skripsi ini.

7. Bapak Eko Sakti P, ST., MT. selaku Dosen Pembimbing II yang telah banyak memberikan bimbingan, masukan dan arahan dalam penyusunan skripsi ini.
8. Keluarga besar Sufri Mundzir dan Abdul Aziz yang selalu memberikan motivasi, dukungan, semangat, dan doa selama penulis menempuh dan menyelesaikan studi di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
9. Dian Ayu Novitasari, yang selalu menjadi motivasi penulis, memberikan doa dan semangat agar skripsi ini cepat selesai dan lulus.
10. Sahabat penulis Eriyanto Wibowo S.Kom, Arief Azwar Alfian S.Kom, Alfi Aulia S.Kom, dan Rico Maulana S.Kom yang telah memberi motivasi, dukungan, semangat, dan doa selama penulis menempuh dan menyelesaikan studi di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
11. Teman penulis Ichsan Wahyudi S.Kom, Dyah Ayu M G W S.Kom, Tika Rahmadian S.Kom, Ardy Purnama Putra S.Kom, Ryan Nanda S.Kom, M. Nurweseiso Wibisono S.Kom, Yusuf Oktofiani S.Kom, Yoga Pradana S.Kom, Dhimas Sawung P S.Kom, Thierry Rahman S.Kom dan Delis Sukmawati S.Kom yang selalu bertukar pikiran dan semangat selama pengerjaan skripsi ini serta senior penulis.
12. Mas Broto dan Mas Didit selaku Laboran Laboratorium Jaringan dan Robotika yang selalu bersedia meminjamkan lab dan alat untuk melaksanakan percobaan dan pengujian.
13. Teman-teman Arsenal Indonesia Suporter Regional Malang (AIS MALANG) yang telah memberikan motivasi agar skripsi ini cepat selesai dan lulus.
14. Teman-teman kelas TIF C yang telah memberikan sebuah cerita dan persahabatan yang tak pernah ternilai harganya selama penulis kuliah di Universitas Brawijaya.
15. Teman-teman Angkatan 2008, 2009, 2010, dan 2011 Informatika, terima kasih atas segala bantuannya selama menempuh studi di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.

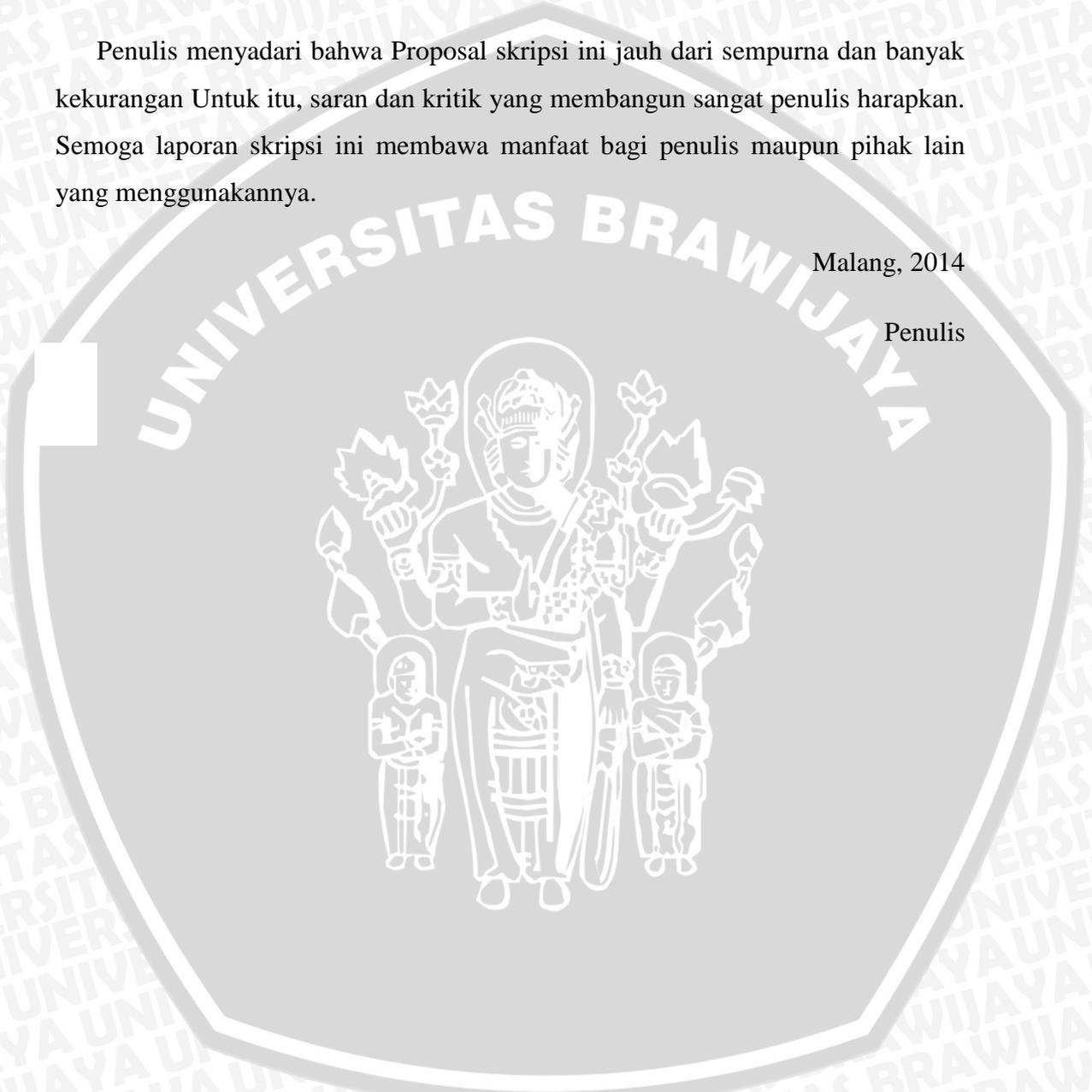
16. Pihak lain yang tidak bisa penulis sebutkan satu persatu yang terlibat langsung maupun tidak langsung demi terselesaikannya skripsi ini.

Hanya doa yang bisa penulis berikan dan semoga Allah SWT memberikan pahala serta balasan kebaikan yang berlipat. Amin.

Penulis menyadari bahwa Proposal skripsi ini jauh dari sempurna dan banyak kekurangan Untuk itu, saran dan kritik yang membangun sangat penulis harapkan. Semoga laporan skripsi ini membawa manfaat bagi penulis maupun pihak lain yang menggunakannya.

Malang, 2014

Penulis



ABSTRAKSI

Aldim Irfani Vikri. 2014. Komunikasi Pengiriman Data Dengan Protokol Zigbee Menggunakan Mekanisme Internet Checksum Untuk Mengubah Parameter Modul Arduino. Skripsi Program Studi Teknik Informatika. PTIIK Universitas Brawijaya. Sabriansyah Rizqika Akbar, ST., M.Eng. dan Eko Sakti P, S.Kom., M.Kom.

Seiring berjalannya waktu, teknologi tanpa kabel atau wireless telah mengalami perkembangan teknologi yang sangat pesat dan telah banyak dimanfaatkan untuk WSN (*Wireless Sensor Network*). WSN kebanyakan ditempatkan pada tempat yang tidak dapat di jangkau oleh manusia. Pada setiap WSN memiliki parameter sensor yang dapat bekerja sesuai kebutuhan sensor tersebut. Pada beberapa kasus tertentu dimana suatu wireless sensor akan diubah parameternya sesuai kebutuhan dibutuhkan suatu komunikasi nirkabel berdaya rendah dan jarak jangkauan yang cukup jauh untuk dapat mendukung pengiriman parameter tersebut. Protokol ZigBee yang menerapkan standar IEEE 802.15.4 merupakan solusi kedepan untuk itu. ZigBee memiliki kelebihan pada pengoperasiannya yang memiliki jarak maksimal komunikasi hingga 100 meter, bentuknya kecil, murah dan membutuhkan daya yang sangat rendah. Dalam skripsi “Komunikasi Pengiriman Data Dengan Protokol Zigbee Menggunakan Mekanisme Internet Checksum Untuk Mengubah Parameter Modul Arduino” dikaji akan dilakukan penelitian dan pembuatan prototipe sebuah sistem yang dapat melakukan pembaharuan parameter pada sensor. Pengujian dilakukan melalui dua node dengan salah satu fungsi sebagai sender dan satunya berfungsi sebagai receiver. Pengiriman data secara serial dapat diandalkan, karena pada pengiriman data dengan jarak 1m – 100m tidak mengalami kehilangan paket, akan tetapi jarak satu node dengan node yang lain mempengaruhi kecepatan dari pengiriman data. File kode hex yang diterima oleh receiver dapat di upload kedalam flash memory.

Kata Kunci : *Komunikasi Wireless, Protokol ZigBee, Wireless Sensor Network*

ABSTRACT

Aldim Irfani Vikri. 2014. Data Delivery On Zigbee Communication With Internet Checksum Mechanism To Change Parameters of Microcontroller. Skripsi Informatics Technology / Computer Science Study. Information Technology and Computer Science Program Brawijaya University. Supervisor: Sabriansyah Rizqika Akbar, ST., M.Eng. and Eko Sakti P, S.Kom., M.Kom.

As time goes by, wireless technology has technology without cable or wireless has rapidly develop and be used for WSN (Wireless Sensor Network) so much. Mostly, WSN used in a place that cannot reached by peoples. In every WSN has a sensor parameter that can work appropriate with the sensor needed. In some particular case the parameter of wireless sensor will be changed, so that can appropriate with the requirement, that need a low-power wireless communication and far enough range which supporting the transmission parameter. ZigBee protocol that apply the standard of IEEE 802.15.4 are the solution for it. ZigBee have advantages in operating that have a maximal communication to 100 meters, tiny shape, cheap, and a lowest power need. In “Firmware Updates In Wireless Sensor Utilizing Low-power Wireless Communications”’s essay will be researched and be made a system prototype that can do a renewal parameter in sensor. The testing is do with two nodes which one of them is as sender function and the other one as receiver. The data transmission by serial can be relied, because transmit data in 1-100m will not be losses, but the distance from one node to another will affect the speed of data transmission. Hex code file that accepted by receiver can be uploaded into flash memory.

Keyword : *Wireless Communication, Protocol ZigBee, Wireless Sensor Network*

DAFTAR ISI

KATA PENGANTAR.....	I
ABSTRAKSI.....	IV
ABSTRACT.....	V
DAFTAR ISI.....	VI
DAFTAR GAMBAR.....	IX
DAFTAR TABEL	XI
BAB I.....	1
1.1. LATAR BELAKANG	1
1.2. RUMUSAN MASALAH	2
1.3. BATASAN MASALAH	3
1.4. TUJUAN	3
1.5. MANFAAT	4
1.6. SISTEMATIKA PEMBAHASAN	4
BAB II	6
2.1. KAJIAN PUSTAKA	6
2.2. PROTOKOL ZIGBEE 802.15.4	7
2.2.1. <i>Modul XBee</i>	7
2.2.2. <i>Prinsip Kerja Modul XBee</i>	8
2.2.3. <i>Modem ZigBee</i>	10
2.3. MIKROKOMPUTER	11
2.3.1. <i>Raspberry Pi</i>	11
2.4. KOMUNIKASI SERIAL	13
2.5. XMODEM	14
2.6. CHECKSUM.....	15
2.6.1. <i>Internet Checksum</i>	15
2.7. MIKROKONTROLER	16
2.7.1. <i>Arduino</i>	16



2.7.2.	<i>Kelebihan Arduino</i>	17
2.7.3.	<i>File Kode Hex</i>	17
2.7.4.	<i>AVRDUDE</i>	18
2.8.	PENGUJIAN PERANGKAT LUNAK	18
BAB III		19
3.1.	METODE PENELITIAN	19
3.1.1.	<i>Studi Literatur</i>	20
3.1.2.	<i>Analisis Kebutuhan</i>	20
3.1.3.	<i>Perancangan</i>	22
3.1.4.	<i>Implementasi</i>	23
3.1.5.	<i>Pengujian dan Analisis</i>	23
3.1.6.	<i>Pengambilan Kesimpulan dan Saran</i>	25
3.2.	PERANCANGAN	25
3.2.1.	<i>Diskripsi Kerja Sistem Secara Keseluruhan</i>	26
3.2.2.	<i>Perancangan Perangkat Lunak Pengiriman File</i>	27
3.2.3.	<i>Perancangan Perangkat Lunak Penerima File</i>	32
3.2.4.	<i>Perancangan Proses Unggah File Kode Hex</i>	36
BAB IV		37
4.1.	SPESIFIKASI SISTEM	38
4.1.1.	<i>Spesifikasi Perangkat Keras</i>	38
4.1.2.	<i>Spesifikasi Perangkat Lunak</i>	39
4.2.	IMPLEMENTASI PERANGKAT KERAS	40
4.2.1.	<i>Implementasi Perangkat Keras Komputer Sender</i>	40
4.2.2.	<i>Implementasi Perangkat Keras Komputer Receiver</i>	41
4.3.	IMPLEMENTASI KONFIGURASI MODUL XBEE	41
4.3.1.	<i>Konfigurasi Modul XBee Sender</i>	42
4.3.2.	<i>Konfigurasi Modul XBee Receiver</i>	43
4.4.	IMPLEMENTASI PERANGKAT LUNAK	43
4.4.1.	<i>Implementasi Komunikasi Serial</i>	44
4.4.2.	<i>Implementasi Pengiriman File</i>	45

4.4.3. Implementasi Mengunggah File.....	51
4.5. KENDALA DAN SOLUSI.....	52
BAB V.....	53
5.1. PENGUJIAN KOMUNIKASI SERIAL.....	53
5.2. PENGUJIAN PENGIRIMAN FILE.....	54
5.2.1. Pengujian Pengiriman File Kode Hex.....	54
5.2.2. Pengujian Menerima File Kode Hex.....	56
5.2.3. Pengujian Pengiriman String ACK.....	57
5.2.4. Pengujian Algoritma Pengecekan (Internet Checksum).....	58
5.3. PENGUJIAN UNGGAH FILE KEDALAM FLASH MEMORI.....	59
5.4. PENGUJIAN KESELURUHAN SISTEM.....	61
5.3.1. Pengujian Skenario 1.....	61
5.3.2. Pengujian Skenario 2.....	62
BAB VI.....	64
6.1. KESIMPULAN.....	64
6.2. SARAN.....	64
DAFTAR PUSTAKA.....	66
LAMPIRAN.....	68
LAMPIRAN 1. KODE PROGRAM KOMPUTER SENDER.....	68
LAMPIRAN 2. KODE PROGRAM KOMPUTER RECEIVER.....	71

DAFTAR GAMBAR

	Halaman
GAMBAR 2.1 BENTUK FISIK MODUL XBEE IEEE 802.15.4.....	8
GAMBAR 2.2 DIAGRAM DATA <i>FLOW</i> INTERNAL MODUL XBEE.....	9
GAMBAR 2.3 ILUSTRASI PRINSIP KERJA DUA MODUL XBEE	9
GAMBAR 2.4 KONFIGURASI XBEE MELALUI <i>AT COMMAND</i>	10
GAMBAR 2.5 RASPBERRY PI.....	11
GAMBAR 2.6 BENTUK FISIK ARDUINO	17
GAMBAR 3.1 DIAGRAM ALIR METODOLOGI PENELITIAN	19
GAMBAR 3.2 DIAGRAM BLOK PERANCANGAN SISTEM	22
GAMBAR 3.3 BLOK DIAGRAM ALIR PENGUJIAN.....	24
GAMBAR 3.4 POHON ANALISA DAN PERANCANGAN SISTEM.....	26
GAMBAR 3.5 DIAGRAM BLOK PERANCANGAN APLIKASI SECARA UMUM.....	26
GAMBAR 3.6 SISTEM PLAN PROSES PENGIRIMAN FILE.....	27
GAMBAR 3.7 SISTEM PLAN PROSES KOMUNIKASI SERIAL.....	28
GAMBAR 3.8 SISTEM PLAN PROSES PENGIRIMAN FILE.....	29
GAMBAR 3.9 SISTEM PLAN PROSES ALGORITMA <i>INTERNET CHECKSUM</i>	31
GAMBAR 3.10 SISTEM PLAN PROSES PENERIMAAN FILE.....	32
GAMBAR 3.11 SISTEM PLAN PROSES KOMUNIKASI SERIAL.....	33
GAMBAR 3.12 SISTEM PLAN PROSES PENERIMAAN FILE KODE HEX.....	34
GAMBAR 3.13 SISTEM PLAN PROSES ALGORITMA <i>INTERNET CHECKSUM</i>	35
GAMBAR 4.1 POHON IMPLEMENTASI.....	37
GAMBAR 4.2 PERANGKAT KERAS KOMPUTER <i>SENDER</i>	40
GAMBAR 4.3 PERANGKAT KERAS KOMPUTER <i>RECEIVER</i>	41
GAMBAR 4.4 KONFIGURASI MODUL XBEE <i>SENDER</i>	42
GAMBAR 4.5 KONFIGURASI MODUL XBEE <i>RECEIVER</i>	43
GAMBAR 4.6 IMPOR <i>LIBRARY</i> PYSERIAL.....	44
GAMBAR 4.7 KONFIGURASI SERIAL PORT	44
GAMBAR 4.8 IMPORT <i>LIBRARY</i> XMODEM	45

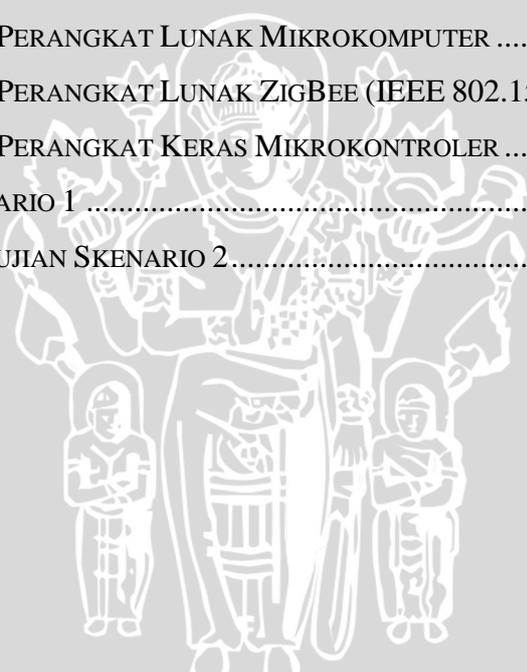
GAMBAR 4.9 FUNGSI MEMBACA DATA	46
GAMBAR 4.10 PENGIRIMAN FILE.....	47
GAMBAR 4.11 FUNGSI MEMBACA DATA DAN ALGORITMA PENGECEKAN	48
GAMBAR 4.12 PROSES PENGIRIMAN STRING ACK.....	50
GAMBAR 4.13 PROSES MENERIMA ACK DAN ALGORITMA PENGECEKAN	50
GAMBAR 4.14 KONFIGURASI AVRDUDE	52
GAMBAR 5.1 KOMUNIKASI SERIAL KOMPUTER <i>RECEIVER</i>	53
GAMBAR 5.2 KOMUNIKASI SERIAL KOMPUTER <i>SENDER</i>	54
GAMBAR 5.3 SERIAL PORT TIDAK DITEMUKAN.....	54
GAMBAR 5.4 PENGIRIMAN FILE.....	55
GAMBAR 5.5 KOMPUTER <i>SENDER</i> MELAKUKAN PERULANGAN	55
GAMBAR 5.6 KOMPUTER <i>SENDER TRANSMITTED</i>	56
GAMBAR 5.7 PROSES MENERIMA FILE.....	56
GAMBAR 5.8 KOMPUTER <i>SENDER</i> BERHASIL MENERIMA ACK.....	57
GAMBAR 5.9 KOMPUTER <i>SENDER</i> GAGAL MENERIMA ACK	57
GAMBAR 5.10 KODE PROGRAM INTERNET CHECKSUM.....	58
GAMBAR 5.11 PENGUJIAN ALGORITMA INTERNET CHECKSUM.....	59
GAMBAR 5.12 KONDISI AWAL MODUL ARDUINO	59
GAMBAR 5.13 KONDISI MODUL ARDUINO SETELAH UNGGAH FILE.....	60
GAMBAR 5.14 PROSES UNGGAH FILE KEDALAM FLASH MEMORI.....	60
GAMBAR 5.15 MIKROKONTROLER ARDUINO ATMEGA	61



DAFTAR TABEL

Halaman

TABEL 1.1 PERBEDAAN ANTARA UWB, BLUETOOTH, DAN ZIGBEE.....	2
TABEL 2.1 SPESIFIKASI RASPBERRY PI.....	12
TABEL 4.1 SPESIFIKASI PERANGKAT KERAS KOMPUTER	38
TABEL 4.2 SPESIFIKASI PERANGKAT KERAS MIKROKOMPUTER.....	38
TABEL 4.3 SPESIFIKASI PERANGKAT KERAS ZIGBEE (IEEE 802.15.4)	38
TABEL 4.4 SPESIFIKASI PERANGKAT KERAS MIKROKONTROLER.....	39
TABEL 4.5 SPESIFIKASI PERANGKAT LUNAK KOMPUTER.....	39
TABEL 4.6 SPESIFIKASI PERANGKAT LUNAK MIKROKOMPUTER.....	39
TABEL 4.7 SPESIFIKASI PERANGKAT LUNAK ZIGBEE (IEEE 802.15.4).....	39
TABEL 4.8 SPESIFIKASI PERANGKAT KERAS MIKROKONTROLER	40
TABEL 5.1 HASIL SKENARIO 1	62
TABEL 5.2 HASIL PENGUJIAN SKENARIO 2.....	63



BAB I

PENDAHULUAN

1.1. Latar Belakang

Teknologi tanpa kabel atau wireless telah mengalami perkembangan yang pesat dan telah banyak dimanfaatkan sebagai WSN (*Wireless Sensor Network*). WSN merupakan suatu jaringan nirkabel yang terdiri dari beberapa sensor yang kebanyakan diletakkan pada tempat yang sulit dijangkau oleh manusia untuk memonitoring kondisi suatu plan seperti suhu, suara, getaran, gelombang elektromagnetik, tekanan, gerakan, dll. Pada setiap WSN memiliki parameter sensor yang dapat bekerja sesuai kebutuhan sensor tersebut. Parameter tersebut perlu diubah untuk disesuaikan dengan kebutuhan dan untuk keperluan pemeliharaan. Pada beberapa kasus, jika ingin mengubah sebuah parameter sensor memungkinkan seorang *programmer* sensor untuk mengambil sensor tersebut dan diubah parameternya lalu dikembalikan pada tempat asal sensor tersebut ditanam. Namun cara tersebut sangat tidak efisien, mengingat sensor diletakkan ditempat yang sulit dijangkau oleh manusia. Oleh karena itu dibutuhkan suatu komunikasi nirkabel berdaya rendah dan jarak jangkauan yang cukup jauh untuk dapat mendukung pengiriman parameter tersebut.

Pada IEEE telah terdapat standarisasi teknologi wireless yaitu 802.15. Terdapat 3 teknologi wireless yang telah menggunakan standar IEEE 802.15 yaitu Bluetooth (802.15.1), UWB/*Ultra-Wideband* (802.15.3), dan ZigBee (802.15.4). Namun yang telah banyak digunakan oleh masyarakat adalah Bluetooth dan UWB, karena kedua perangkat ini sudah banyak diaplikasikan pada laptop, dan beberapa gadget lainnya. Bluetooth dan UWB merupakan sebuah standar yang bekerja untuk kecepatan transfer tingkat tinggi, kecepatannya mencapai 1 Mbps untuk Bluetooth dan 50 Mbps untuk UWB sehingga cocok digunakan untuk pengiriman data yang besar. Sedangkan untuk sebuah kecepatan transfer rendah dapat gunakan standar dari protokol ZigBee.

Protokol ZigBee juga masuk dalam standar keluarga IEEE 802.15 bersama Bluetooth (802.15.1) dan UWB (802.15.3) dengan kode standar IEEE 802.15.4.

ZigBee memiliki kelebihan pada pengoperasiannya yang memiliki jarak maksimal komunikasi hingga ± 100 meter, bentuknya kecil, murah dan membutuhkan daya yang sangat rendah dibandingkan dengan kedua keluarganya yang lain, Bluetooth dan UWB [WIN-12].

Tabel 1.1 Perbedaan antara UWB, Bluetooth, dan ZigBee

Sumber : [RIA-09]

Teknologi	Bandwidth	Konsumsi Daya	Jarak	Fungsi	Aplikasi
UWB	54 mbps	400 + mA TX	± 50 M	Pembacaan data yang cepat	Internet browsing, pertukaran data
Bluetooth	1 mbps	400 mA TX	$\pm 0,5$ M	Interoperabilitas, Pengganti Kabel	Handset, Headset
ZigBee	250 kbps	30 mA TX	± 100 M	Baterai yang tahan lama, biaya murah	sensor

Dengan memanfaatkan teknologi dari protokol ZigBee yang mempunyai daya konsumsi rendah dan jarak jangkauan yang cukup jauh, pada skripsi “Komunikasi Pengiriman Data Dengan Protokol Zigbee Menggunakan Mekanisme Internet Checksum Untuk Mengubah Parameter Modul Arduino” akan dilakukan penelitian dan pembuatan prototipe sebuah sistem yang dapat melakukan pembaharuan parameter pada modul arduino.

1.2. Rumusan Masalah

Berdasarkan permasalahan yang diangkat pada bagian latar belakang, maka rumusan masalah dikhususkan pada :

1. Bagaimana membuat prototipe sebuah sistem yang mampu merubah parameter modul arduino secara nirkabel dengan memanfaatkan komunikasi data dengan protokol ZigBee (IEEE 802.15.4).
2. Bagaimana mekanisme deteksi kesalahan menggunakan Internet Checksum untuk pengiriman file pada protokol ZigBee.
3. Bagaimana modul arduino mampu merubah parameter setelah menerima file.

1.3. Batasan Masalah

Agar permasalahan yang dirumuskan dapat lebih terfokus, maka pada penelitian ini dibatasi dalam hal:

1. Protokol komunikasi nirkabel yang digunakan adalah protokol ZigBee.
2. Menggunakan bentuk fisik dari protokol ZigBee yaitu dua alat Modul XBee, satu sebagai koordinator dan satu lagi sebagai router.
3. Menggunakan perintah *AT Command* untuk konfigurasi Modul XBee.
4. Perangkat lunak yang dibuat berbasis Terminal/*console*.
5. Menggunakan satu mikrokomputer Raspberry Pi sebagai penerima dengan sistem operasi berbasis Linux Debian yang sudah terkoneksi dengan satu modul XBee dan mikrokontroler.
6. Menggunakan mikrokontroler Arduino Mega sebagai notifikasi sensor.
7. Pengiriman file kode hex kedalam mikrokontroler.
8. File yang dikirim oleh modul XBee adalah file kode hexa yang berisi kode untuk menyalakan lampu LED setiap 8 detik sekali.
9. Menggunakan *Internet Checksum* untuk pengecekan data yang diterima sama atau tidak dengan yang dikirim.
10. File kode hexa diunggah kedalam flash memori mikrokontroler dengan memanfaatkan teknologi perangkat lunak AVRDUDE.

1.4. Tujuan

Sesuai dengan latar belakang dan rumusan masalah, tujuan dari pengembangan perangkat lunak ini adalah mengetahui bagaimana pembuatan

sistem yang mampu berkomunikasi dalam hal pengiriman data pada protokol ZigBee (IEEE 802.15.4) dengan algoritma pengecekan menggunakan mekanisme Internet Checksum untuk merubah parameter dari sebuah modul arduino.

1.5. Manfaat

Penulisan skripsi ini diharapkan mempunyai manfaat yang baik dan berguna bagi pembaca dan penulis. Adapun manfaat yang diharapkan adalah sebagai berikut:

a. Bagi penulis

1. Menerapkan ilmu yang telah didapatkan di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
2. Mengetahui bagaimana mekanisme deteksi kesalahan pengiriman file pada protokol ZigBee.
3. Sebagai salah satu referensi belajar bagi mahasiswa untuk meningkatkan pengetahuannya dalam bidang perancangan khususnya perancangan aplikasi perangkat lunak.

b. Bagi pengguna

1. Menyediakan aplikasi yang dapat digunakan untuk mengirim file dengan protokol ZigBee dan proses unggah file kode hex kedalam flash memori mikrokontroler.
2. Untuk mempermudah programmer dalam pembaharuan dari sebuah wireless sensor tanpa harus mengambil sensor tersebut.

1.6. Sistematika Pembahasan

Sistematika pembahasan ditunjukkan untuk memberikan gambaran dan uraian dari penulisan skripsi ini secara garis besar yang meliputi beberapa bab, sebagai berikut :

Bab I : Pendahuluan

Memuat latar belakang, rumusan masalah, tujuan, batasan masalah, manfaat, metodologi pembahasan, dan sistematika pembahasan.

Bab II : Tinjauan Pustaka dan Dasar Teori

Menguraikan tentang referensi dan dasar teori yang mendasari pembuatan aplikasi komunikasi pengiriman data dengan protokol zigbee menggunakan mekanisme internet checksum untuk mengubah parameter modul arduino.

Bab III : Metode Penelitian dan Perancangan

Menguraikan dan membahas langkah kerja yang dilakukan dalam penulisan skripsi yang terdiri dari studi literatur, implementasi, perancangan, analisis kebutuhan dan pengujian komunikasi pengiriman data dengan protokol zigbee menggunakan mekanisme internet checksum untuk mengubah parameter modul arduino.

Bab IV : Implementasi

Membahas implementasi dari aplikasi komunikasi pengiriman data dengan protokol zigbee menggunakan mekanisme internet checksum untuk mengubah parameter modul arduino yang sesuai dengan perancangan sistem yang telah dibuat.

Bab V : Pengujian dan Analisis

Memuat hasil pengujian dan analisis terhadap perangkat lunak yang telah direalisasikan.

Bab VI : Penutup

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan dalam skripsi ini serta saran – saran untuk pengembangan lebih lanjut.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

Bab ini berisi tinjauan pustaka yang meliputi kajian pustaka dan dasar teori yang diperlukan untuk penelitian. Kajian pustaka adalah membahas penelitian yang telah ada dan yang diusulkan. Dasar teori adalah membahas teori yang diperlukan untuk menyusun penelitian yang diusulkan.

Kajian pustaka pada penelitian ini adalah membahas penelitian sebelumnya yang berjudul ‘Impelentasi *Wireless Sensor Network* (WSN) Untuk Monitoring Beberapa Parameter Penting Yang Berhubungan Dengan Permasalahan Daya Serta Keperluan Pemeliharaan’. Teori dasar yang akan dibahas pada bab ini yaitu membuat sebuah WSN yang terdiri dari beberapa sensor node yang diletakkan pada tempat yang berbeda-beda untuk memonitoring kondisi suatu plan dengan metode pengiriman data ke PC menggunakan komunikasi nirkabel, yang selanjutnya data parameter dikirim ke PC yang selanjutnya simpan pada database.

2.1. Kajian Pustaka

Kajian pustaka pada penelitian ini adalah membahas penelitian sebelumnya yang berjudul ‘Impelentasi *Wireless Sensor Network* (WSN) Untuk Monitoring Beberapa Parameter Penting Yang Berhubungan Dengan Permasalahan Daya Serta Keperluan Pemeliharaan’. Penelitian ini membahas tentang sistem monitoring pada *Wireless Sensor Network* (WSN) yang merupakan jaringan nirkabel dan terdiri dari beberapa sensor node yang diletakkan pada tempat yang berbeda-beda untuk memonitoring kondisi suatu plan. Dalam penelitian ‘Impelentasi *Wireless Sensor Network* (WSN) Untuk Monitoring Beberapa Parameter Penting Yang Berhubungan Dengan Permasalahan Daya Serta Keperluan Pemeliharaan’ dikaji penggunaan protokol ZigBee IEEE 802.15.4 pada *Wireless Sensor Network* (WSN) dengan cara mengirim beberapa parameter ke PC untuk memonitoring kinerja beban daya industri dan keperluan. Pada pengimplementasiannya, penelitian tersebut menggunakan 4 modul XBee

yang saling berkomunikasi dan pengiriman filenya memanfaatkan perangkat lunak Visual Basic 6.0 serta menggunakan database untuk penyimpanan datanya.

Perbedaan yang dibuat penulis pada penelitian “Komunikasi Pengiriman Data Dengan Protokol Zigbee Menggunakan Mekanisme Internet Checksum Untuk Mengubah Parameter Modul Arduino” dalam pengimplementasiannya adalah pengiriman data yang menggunakan dua modul XBee yang saling berkomunikasi dan data yang dikirimkan adalah berupa file kode hex yang berisi parameter dari sebuah sensor. Aplikasi yang dibuat adalah aplikasi berbasis Terminal/*console*.

2.2. Protokol ZigBee 802.15.4

Khamdan Amin Bisyrri [BIS-12:2] mengatakan, “ZigBee adalah protokol jaringan nirkabel yang ditargetkan untuk otomasi dan aplikasi *remote control* dengan teknologi data rate rendah, konsumsi daya rendah, dan murah.”

ZigBee adalah standar dari IEEE 802.15.4 yang telah banyak dimanfaatkan untuk sensor yang wireless karena mempunyai kelebihan dengan komunikasi yang tidak memakan banyak daya sehingga dapat bertahan beberapa bulan atau tahun. Selain itu juga kelebihan protokol ZigBee adalah memiliki kemampuan untuk mengatur jaringan sendiri ataupun mengatur pertukaran data.

Pada skripsi “Komunikasi Pengiriman Data Dengan Protokol Zigbee Menggunakan Mekanisme Internet Checksum Untuk Mengubah Parameter Modul Arduino” dalam pengimplementasiannya menggunakan bentuk fisik dari protokol ZigBee IEEE 802.15.4 yaitu modul XBee.

2.2.1. Modul XBee

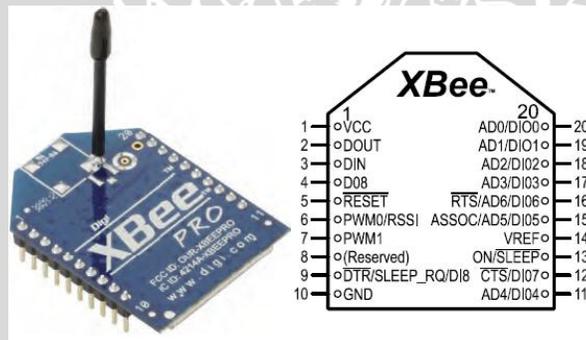
Modul XBee adalah bentuk fisik dari protokol ZigBee IEEE 802.15.4 merupakan modul *transceiver* yaitu modul yang dapat memancarkan dan menerima sinyal. Modul XBee ini beroperasi pada frekuensi 2.4 GHz adalah teknologi yang memfokuskan pada data rate rendah, konsumsi daya rendah, dan jarak jangkauannya yang cukup jauh bisa mencapai ± 100 Meter. Fathurrohman [FAT-13:3] mengatakan, “Modul ini memerlukan tegangan suplai 2.8 V sampai dengan 3.3 V saat mengirim data, modul ini akan membebani dengan arus 270

mA, dan arus 55 mA untuk penerimaan data”. Pada modul XBee terdapat beberapa pin, dan pin yang digunakan sementara ini adalah 6 pin dari 20 pin. 6 pin yang digunakan adalah

- VCC dan GND berfungsi untuk tegangan suplai
- DOUT berfungsi untuk *Transmit(TX)*
- DIN merupakan pin penerima/*Receive (RX)*
- Reset merupakan pin reset modul XBee
- PWMO/RSSI berfungsi untuk indikator bahwa ada penerimaan data

Modul XBee dapat digunakan sebagai pengganti kabel serial, dapat juga digunakan sebagai mode perintah untuk suatu *broadcast*, dan pilihan menghubungkan suatu jaringan. Radio frekuensi *tranciever* ini merupakan sebuah modul yang terdiri dari RF *receiver* dan RF *trnasmiter* dengan sistem antar muka serial *Universal Asynchronous Receiver-Transmitter (UART) asynchronous*.

Pada gambar 2.1 ditunjukkan bentuk fisik dari XBee PRO sebagai berikut:



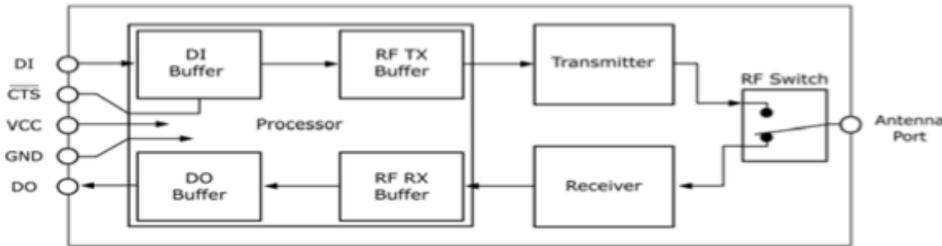
Gambar 2.1 Bentuk Fisik Modul XBee IEEE 802.15.4

Sumber : [HEB-11]

2.2.2. Prinsip Kerja Modul XBee

Khamdan Amin Bisyrri [BIS-12:3] mengatakan, “Pada dasarnya, modul XBee merupakan komunikasi serial. Akan tetapi, apabila mode API (*Application Programming Interface*) digunakan, dibutuhkan pemaketan data RF”. Dalam mengirim atau menerima data pada internal modul XBee data akan di-*buffer* terlebih dahulu dan diubah menjadi paket RF. Pada modul XBee data akan masuk ke DI *buffer* setelah itu input data akan diteruskan ke pin RF TX *buffer* dan ditransmisikan sehingga posisi RF *switch* menjadi *transmitter*. Begitu juga sebaliknya, apabila ada data yang diterima, posisi RF *switch* menjadi 4 *receiver*

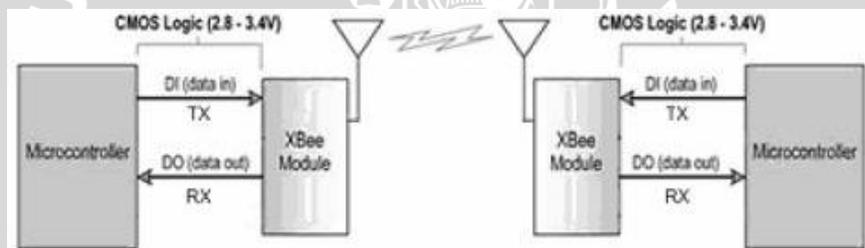
lalu data akan masuk RF RX *buffer*, kemudian data diteruskan ke DO *buffer* lalu menjadi data output (DO), kemudian DO diteruskan dari XBee ke host. Diagram data flow internal XBee dapat dilihat pada Gambar 2.2.



Gambar 2.2 Diagram Data Flow Internal Modul XBee

Sumber : [BIS-12]

Contoh dari prinsip kerja dari dua modul XBee yang saling berkomunikasi dapat dilihat pada gambar 2.3 berikut ini:



Gambar 2.3 Ilustrasi Prinsip Kerja Dua Modul XBee

Sumber : [BIS-12]

Dari ilustrasi di atas dapat dilihat bahwa pin tx dan Rx dapat dikoneksikan langsung dengan pin DIN dan Dout dari modul XBee. sebelum ditransmisikan via port antena menuju modul XBee yang lain, data yang diterima oleh modul XBee masuk melalui pin DIN dan disimpan pada pin DI-*buffer* dan RF Tx-*buffer*. Begitu juga sebaliknya dengan data yang diterima melalui port antena.

Sebelum menggunakan modul XBee ada beberapa parameter yang harus diatur agar dapat berkomunikasi dengan modul lain. Terdapat 24 perintah (*AT Command*) untuk mengatur parameter dari modul XBee. Cara mengatur parameter modul XBee sebagai berikut:

```
request=+++ //membuka atcommand
```

```
response=OK
request=atmy1 //alamat diri =1
response=OK
request=atdl2 //alamat yang dikirim
response=OK
request=atchc //channel RF connection
response=OK
request=atid123 //Personal Area Network
response=OK
request=atwr //menyimpan di memori xbee pro
response=OK
request=atcn //menutup atcommand
response=OK
```

Gambar 2.4 Konfigurasi XBee Melalui AT Command

Sumber : [ABR-11]

2.2.3. Modem ZigBee

Pada skripsi “Komunikasi Pengiriman Data Dengan Protokol Zigbee Menggunakan Mekanisme Internet Checksum Untuk Mengubah Parameter Modul Arduino” menggunakan dua node modul XBee yaitu node pertama berfungsi sebagai *sender* dan node kedua berfungsi sebagai *receiver*. Penggunaan dua node modul XBee dapat diartikan menggunakan komunikasi dua arah dan modul XBee dapat difungsikan sebagai modem.

Modem adalah Modulator - Demodulator merupakan suatu alat perangkat keras yang mempunyai bentuk fisik, kecepatan sebuah modem diukur dengan satuan bps (bit per second) atau kbps (kilobit per second) (Nursyamsiah Mughny Zahira, Penyambungan Internet, 2013, Hal. 1). Pengertian modulator adalah suatu rangkaian yang berfungsi melakukan proses modulasi, yaitu proses merubah bentuk data yang berbentuk sinyal digital dirubah menjadi sinyal analog. Demodulator mempunyai fungsi kebalikan dari modulator (demodulasi), yaitu proses mendapatkan kembali data atau proses membaca data dari sinyal yang diterima dari pengirim. Data yang berupa sinyal Analog diberubah kembali menjadi sinyal digital agar bisa terbaca di komputer penerima. Modem

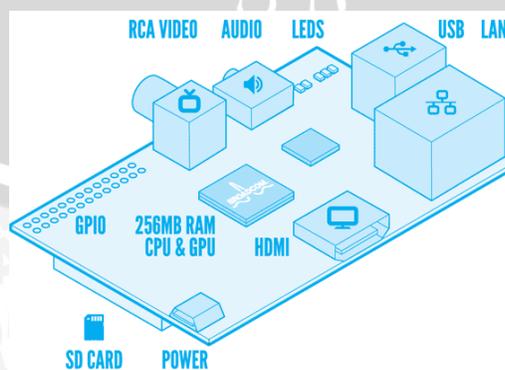
merupakan penggabungan dari kedua sistem modulator dan demodulator, sehingga modul XBee merupakan modem yang memiliki prinsip komunikasi dua arah.

2.3. Mikrokomputer

Gilang Hardiyanto [HAR-11:1] mengatakan, “Mikrokomputer adalah interkoneksi antara mikroprosesor (CPU) dengan memori utama (main memory) dan antarmuka input-output (I/O devices) yang dilakukan dengan menggunakan sistem interkoneksi bus Mikrokomputer dapat dikatakan pula sebagai sebuah mikroprosesor (CPU) dengan ditambahkan unit memori serta sistem I/O”. Pada skripsi “Komunikasi Pengiriman Data Dengan Protokol Zigbee Menggunakan Mekanisme Internet Checksum Untuk Mengubah Parameter Modul Arduino” menggunakan mikrokomputer Raspberry Pi sebagai komputer *receiver* penghubung antara modul XBee *receiver* dengan mikrokontroler sensor.

2.3.1. Raspberry Pi

Malik Abdillah Ibnul Hakim [HAK-13:1] mengatakan, “Raspberry Pi (juga dikenal sebagai RasPi) adalah sebuah SBC (*Single Board Computer*) seukuran kartu kredit yang dikembangkan oleh Yayasan Raspberry Pi di Inggris (UK) dengan maksud untuk memicu pengajaran ilmu komputer dasar di sekolah-sekolah”. Raspberry Pi terdapat dua tipe yaitu Raspberry Pi rev A dan Raspberry Pi rev B. Tidak menggunakan hard disk, namun menggunakan SD Card untuk proses booting dan penyimpanan data jangka-panjang. Berikut gambar Raspberry Pi dan spesifikasinya :



Gambar 2.5 Raspberry Pi

Sumber : [raspberrypi.org]

Spesifikasi Raspberry Pi :

Tabel 2.1 Spesifikasi Raspberry Pi

Sumber : [NIS-13]

	Model A	Model B
Harga	US\$ 25	US\$ 35
SoC	Broadcom BCM2835 (CPU, GPU, DSP, and SDRAM)	
CPU	700 MHz ARM1176JZF-S core (ARM11 family)	
GPU	Broadcom VideoCore IV, OpenGL ES 2.0, MPEG-2 & VC-1 (dengan lisensi), 1080p30 h.264/MPEG-4 AVC high-profile decoder dan encoder	
Memori (SRDAM)	256 MB (berbagi-pakai dengan GPU)	
USB 2.0 ports	1	2 (melalui USB hub)
Luaran Video	Composite RCA (PAL & NTSC), HDMI (rev 1.3 & 1.4), raw LCD Panels via DSI 14 HDMI resolutions from 640×350 to 1920×1200 plus various PAL and NTSC standards.	
Luaran Audio	3.5 mm jack, HDMI	
Media Penyimpanan	SD / MMC / SDIO card slot	
Jaringan	None	10/100 Ethernet (RJ45)
Periferal	8 × GPIO, UART, I ² C bus, SPI bus with two chip selects, +3.3 V, +5 V, ground	
Daya	300 mA (1.5 W)	700 mA (3.5 W)
Catu Daya	5 volt via MicroUSB or GPIO header	
Ukuran	85.60 × 53.98 mm (3.370 × 2.125 in)	
Berat	45gram	
Sistem	Debian GNU/Linux, Fedora, Arch Linux ARM, RISC OS	

Operasi

Untuk dapat menggunakan Raspberry Pi ada beberapa alat yang dibutuhkan, diantaranya adalah

1. SD Card sebagian pengganti Harddisk. Semua system operasi (debian) termasuk data-data akan disimpan disini.
2. Mouse dan Keyboard USB
3. Adaptor / Power Supply (5V 1A)
4. Monitor LCD beserta kabel adapter HDMI

2.4. Komunikasi Serial

Pada skripsi “Komunikasi Pengiriman Data Dengan Protokol Zigbee Menggunakan Mekanisme Internet Checksum Untuk Mengubah Parameter Modul Arduino” agar suatu komputer dapat berkomunikasi dengan sebuah modul XBee dibutuhkan komunikasi melalui serial port. Radix Satrio Ramadhan [RAM-12:3] mengatakan, “Komunikasi serial adalah komunikasi dimana pengiriman data dilakukan per bit, sehingga lebih lambat dibandingkan komunikasi paralel seperti pada port printer yang mampu mengirim 8bit sekaligus dalam sekali detak”. Beberapa contoh komunikasi serial adalah mouse, keyboard, printer dan sistem akuisisi data yang terhubung ke port COM1 / COM2.

Pada prinsipnya, komunikasi serial adalah komunikasi dimana proses transmisi data dilakukan per bit. Komunikasi serial lebih menghemat pin jika dibandingkan dengan komunikasi interface parallel karena komunikasi interface serial hanya membutuhkan jalur yang lebih sedikit yaitu 2 jalur. Python termasuk yang menyediakan dukungan untuk komunikasi secara serial port. Sebelum dapat melakukan komunikasi secara serial port pada python dibutuhkan sebuah proses instalasi *library* Python yang mendukung komunikasi serial port, yaitu PySerial.

PySerial adalah *library* Python siap pakai dan gratis yang dibuat untuk memudahkan dalam membuat program komunikasi data serial RS232 dalam bahasa Python. Agar dapat menggunakan fungsi-fungsi PySerial dalam program, harus mengintegrasikannya terlebih dahulu dengan perintah :

```
import serial
```

Dan juga dapat mem-*binding* modul XBee dengan serial port dengan settingan baudrate 115200.

```
ser = serial.Serial("/dev/ttyUSB0", 115200)
```

2.5. XModem

Pada skripsi “Komunikasi Pengiriman Data Dengan Protokol Zigbee Menggunakan Mekanisme Internet Checksum Untuk Mengubah Parameter Modul Arduino” setelah dapat berkomunikasi secara serial selanjutnya dibutuhkan *library* Python yang mendukung agar dapat mengirim file menggunakan modul XBee. Bahasa pemrograman Python mempunyai *library* untuk mendukung pengiriman file, yaitu menggunakan *library* Python XModem.

Sebuah Xmodem pada sisi pengirim prinsipnya mengambil cara kerja dari modem yaitu mentransmisikan data digital dari komputer ke sinyal analog yang berkomunikasi dua arah. XModem di sisi penerima, mengkonversikan sinyal analog yang diterima dari jalur telepon menjadi data yang akan dibaca oleh komputer. Ukuran kecepatan dari data yang akan ditransmisikan dan dikirim dinyatakan dalam Kbps (kilobit persecond).

Seluruh protokol data bekerja berdasarkan potongan blok-blok dari bit-bit data yang dibawa dan dikirimkan ke modem tujuan, yang selanjutnya akan mengecek kesalahan di masing-masing blok data, dan kemudian, berdasarkan hasilnya dan tipe protokolnya, modem penerima tadi akan mengembalikan sinyal positif *Acknowledgement* (ACK) maupun *Negative Acknowledgement* (NAK) ke modem pengirim.

Contoh data flow dari cara kerja X-Modem

<i>Sender</i>	<i>Receiver</i>
	← NAK
SOH 01 FD Data[8]	→
	← CSUM
	ACK
SOH 02 FC Data[8]	→
	← CSUM
	ACK
SOH 03 FB Data[8]	→
	← ACK
EOT	→

CSUM

← ACK

2.6. Checksum

Jonathan Stone [STO-10:10] mengatakan, “*Checksum* adalah skema kesalahan deteksi sederhana di mana setiap pesan yang dikirim yang menghasilkan nilai numerik berdasarkan byte dalam pesan.” Penerima menghitung panjang data yang diterima dari pengirim, dan jika jumlah data yang diterima benar dengan yang dikirim maka pengiriman data berhasil. Karena kebanyakan kesalahan transmisi ini, protokol jaringan sangat sering menggunakan *checksum* untuk mendeteksi kesalahan tersebut.

Ada beberapa jenis *checksum*, sebuah contoh algoritma *checksum* yang sering digunakan adalah *Parity Checksum*, *Internet Checksum*, dan CRC32. Pada skripsi “Komunikasi Pengiriman Data Dengan Protokol Zigbee Menggunakan Mekanisme Internet Checksum Untuk Mengubah Parameter Modul Arduino” algoritma *checksum* yang digunakan adalah *Internet Checksum*.

2.6.1. Internet Checksum

Internet Checksum dapat diartikan sebagai pengecekan dilakukan dengan melakukan penjumlahan pada sekumpulan data kemudian mengkomplemen jumlah tersebut, dan hasil komplemen tersebut ditambahkan pada data sebagai sebuah variabel. Kemudian pada *reciever*, akan dihitung ulang *checksum*-nya dan dilakukan perbandingan jumlah data yang dikirimkan dengan *checksum*. Bila terjadi perbedaan nilai antara kedua nilai ini, maka terjadi kesalahan dalam pengiriman data.

Pada dasarnya metode ini mirip dengan *Parity Check*, perbedaannya adalah jumlah bit pada *sums* lebih besar dan hasil dari penjumlahan data dengan *checksum* harus selalu dibuat nol.

Menurut Diyah Deviyanti [DEV-12:5], “Kelebihan dari *Internet Checksum* sendiri adalah mudah diimplementasikan kedalam software, dan memiliki kehandalan sistem yang cukup tinggi, yaitu sekitar 90%, namun memiliki kelemahan deteksi kesalahan yang masih lemah (walaupun lebih handal dibandingkan dengan *Parity Check*), karena tidak dapat mendeteksi unit data

(bytes/words) yang urutannya berantakan, tidak dapat mendeteksi unit data mana yang mengalami kesalahan”.

2.7. Mikrokontroler

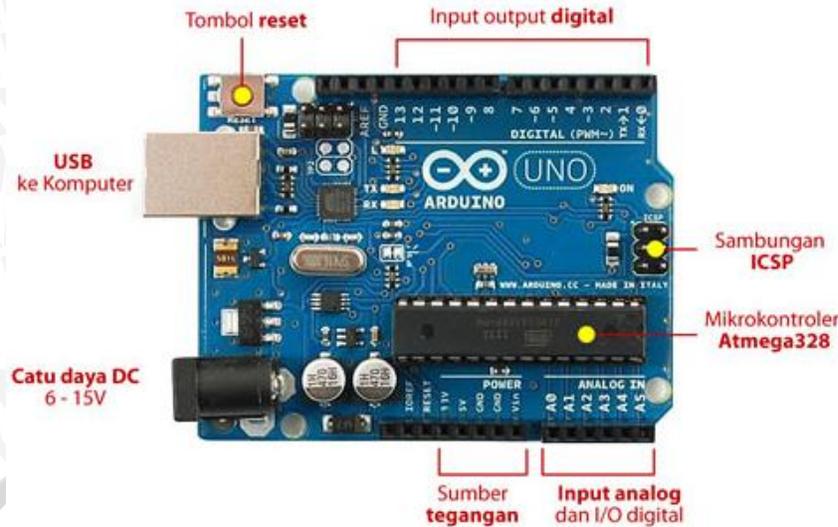
Jati Lestari [LES-11:2] mengatakan, “Mikrokontroler adalah suatu kontroler yang digunakan untuk mengontrol suatu proses atau aspek-aspek dari lingkungan”. Mikrokontroler dibangun menggunakan seluruh kontroler yang masuk kedalam PCB yang kecil. Dalam pengaplikasiannya mikrokontroler juga sering digunakan untuk sensor.

Pada skripsi “Komunikasi Pengiriman Data Dengan Protokol Zigbee Menggunakan Mekanisme Internet Checksum Untuk Mengubah Parameter Modul Arduino” menggunakan mikrokontroler Arduino sebagai sensor untuk notifikasi berhasil atau tidaknya suatu proses unggah file kedalam flash memori dari sebuah mikrokontroler/sensor.

2.7.1. Arduino

Jati Lestari [LES-11:2] mengatakan, “Arduino adalah kit elektronik atau papan rangkaian elektronik *open source* yang di dalamnya terdapat komponen utama yaitu sebuah chip mikrokontroler dengan jenis AVR dari perusahaan Atmel”. Tujuan menanamkan program pada mikrokontroler adalah agar dapat membaca input dan memprosesnya sesuai dengan parameter yang telah dibuat. Jadi mikrokontroler bertugas sebagai ‘otak’ yang mengendalikan input, proses dan output sebuah rangkaian elektronik.

Arduino kebanyakan dipakai pada keperluan mengendalikan robot baik robot mainan, maupun robot industri. Karena komponen utama Arduino adalah mikrokontroler, maka Arduino pun dapat diprogram menggunakan komputer sesuai kebutuhan. Bentuk dari arduino dapat dilihat pada gambar 2.6.



Gambar 2.6 Bentuk Fisik Arduino

Sumber : [BIS-12]

2.7.2. Kelebihan Arduino

Arduino memiliki kelebihan yaitu :

1. Tidak perlu perangkat chip programmer karena di dalamnya sudah ada *bootloader* yang akan menangani upload program dari komputer.
2. Sudah memiliki sarana komunikasi USB yang didukung dengan driver UART.
3. Bahasa pemrograman menggunakan C++ dan relatif mudah karena software Arduino dilengkapi dengan kumpulan *library* yang cukup lengkap.
4. Memiliki modul siap pakai (shield) yang bisa ditancapkan pada board Arduino.

2.7.3. File Kode Hex

Pada bahasa pemrograman Assembler, C, Basic atau yang lain menghasilkan kode hex yang diciptakan oleh kompiler. File kode hex adalah file yang dibaca oleh programmer dan kode-kode mesin-nya dimasukkan kedalam flash memory mikrokontroler untuk memberikan perintah kepada mikrokontroler. Pada setiap program yang menggunakan *software* Arduino IDE terdapat file kode hex dan letaknya jika menggunakan sistem operasi linux berada pada direktori /tmp.

2.7.4. AVRDUDE

Pada arduino juga terdapat fungsi mikrokontroler yang memanfaatkan flash memori, yang berarti sebuah program dapat diupload kedalam mikrokontroler, pada skripsi “Komunikasi Pengiriman Data Dengan Protokol Zigbee Menggunakan Mekanisme Internet Checksum Untuk Mengubah Parameter Modul Arduino” penulis memanfaatkan teknologi tersebut dengan memasukkan file kode hex dari program yang sudah dibuat dengan aplikasi AVRDUDE.

Nuzulyani [NUZ-12:9] mengatakan, “AVRDUDE adalah aplikasi perangkat lunak yang berguna untuk pemrograman mikrokontroler Atmel AVR secara *in-system programming* (ISP)”. Karena AVRDUDE merupakan program berbasis command-line, maka semua instruksi harus diberikan dalam bentuk baris perintah.

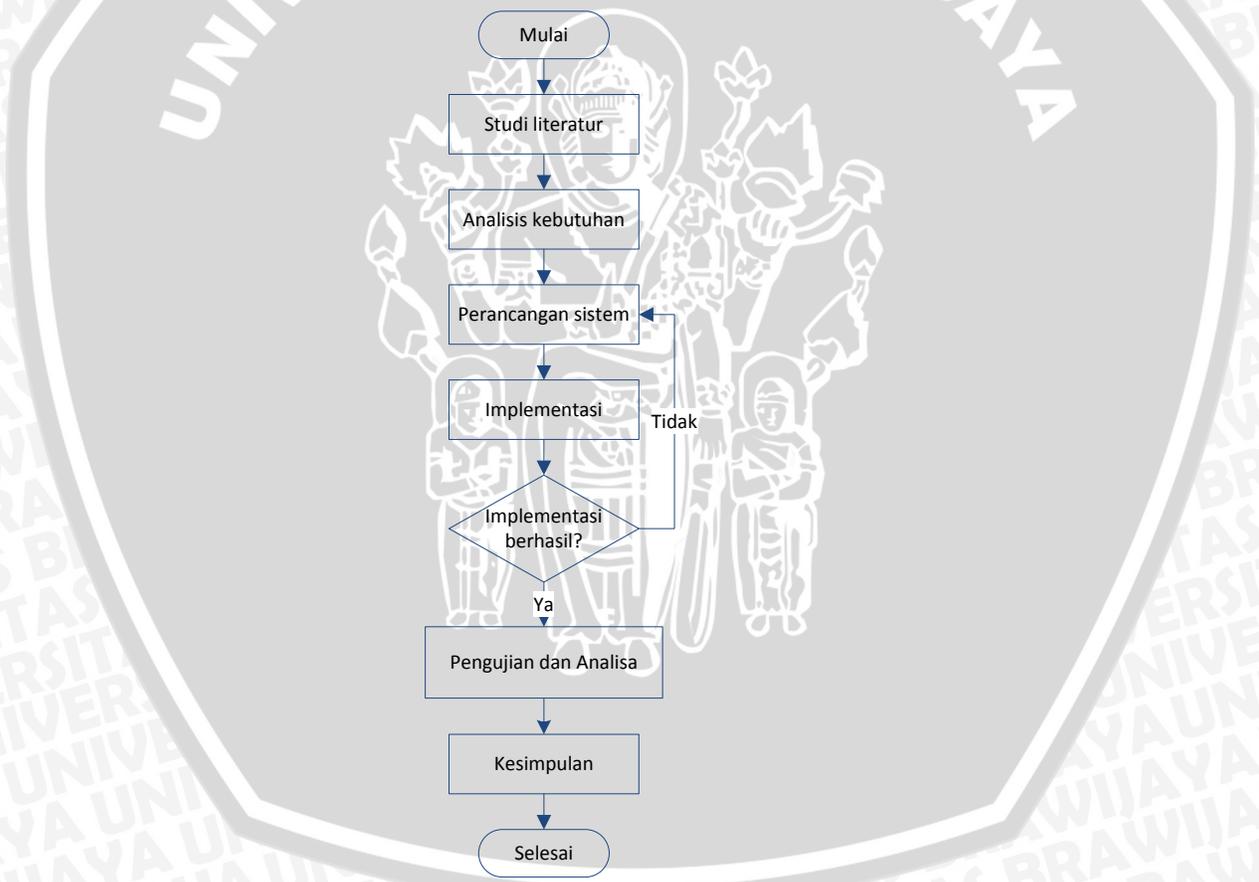
2.8. Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan suatu penilaian terhadap suatu implementasi kemudian dibandingkan dengan kriteria standar yang telah ditetapkan dan melihat tingkat dari suatu keberhasilannya. Dari pengujian kemudian akan tersedia informasi mengenai sejauh mana suatu penilaian sehingga bisa diketahui bila terdapat selisih antara standar yang telah ditetapkan dengan hasil yang bisa dicapai. Pengujian yang digunakan pada penelitian “Komunikasi Pengiriman Data Dengan Protokol Zigbee Menggunakan Mekanisme Internet Checksum Untuk Mengubah Parameter Modul Arduino” adalah dengan melakukan pengiriman file dari satu komputer ke komputer lain dan file tersebut secara dinamis dimasukkan kedalam flash memory dari mikrokontroler.

BAB III METODE PENELITIAN DAN PERANCANGAN

3.1. Metode Penelitian

Pada bab ini dijelaskan langkah-langkah yang akan dilakukan dalam pengerjaan tugas akhir, yaitu studi literatur, penyusunan dasar teori, metode penelitian dan perancangan, implementasi, analisis dan pengujian dari aplikasi perangkat lunak yang akan dibuat, hingga penulisan laporan. Kesimpulan dan saran disertakan sebagai catatan atas aplikasi dan kemungkinan arah pengembangan aplikasi selanjutnya. Gambar 3.1 menunjukkan desain penelitian secara umum.



Gambar 3.1 Diagram Alir Metodologi Penelitian

Sumber : [Perancangan]

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, jurnal, e-book, penelitian sebelumnya, dan dokumentasi proyek. Teori – teori pendukung tersebut meliputi :

1. Protokol ZigBee IEEE 802.15.4
 - a. Modul XBee
 - b. Prinsip Kerja Modul XBee
 - c. Modem
2. Mikrokomputer
 - a. Raspberry Pi
3. Komunikasi Serial
4. XModem
5. Checksum
 - a. Internet Checksum
6. Mikrokontroler/sensor
 - a. Arduino
 - b. Manfaat Arduino
 - c. Kelebihan Arduino
 - d. AVRDUDE
7. Pengujian Perangkat Lunak

3.1.2. Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan oleh sistem yang akan dibangun. Analisis kebutuhan dilakukan dengan mengidentifikasi kebutuhan sistem dan siapa saja yang terlibat di dalamnya. Berikut analisis kebutuhan dalam skripsi “Komunikasi Pengiriman Data Dengan Protokol Zigbee Menggunakan Mekanisme Internet Checksum Untuk Mengubah Parameter Modul Arduino”

A. Kebutuhan Antar Muka

Kebutuhan-kebutuhan untuk pengembangan perangkat lunak ini sebagai berikut :

1. Program yang akan dibangun berbasis *console* yang hanya berjalan pada terminal untuk sistem operasi berbasis Linux atau command prompt untuk sistem operasi berbasis Windows.
2. Program yang dibangun harus mampu mendeteksi *kesalahan* pengiriman file pada protokol ZigBee..
3. Memanfaatkan program perangkat lunak yang mampu mengupload file kode hex kedalam flash memori mikrokontroler.

B. Analisa Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini meliputi:

1. Satu unit komputer sebagai komputer *sender*.
2. Satu unit Raspberry Pi sebagai komputer *receiver*.
3. Dua unit modul XBee beserta adaptornya. Satu modul XBee berfungsi sebagai *sender* dan satu modul XBee lagi berfungsi sebagai *receiver*.
4. Tiga unit kabel serial, dua diantaranya berfungsi sebagai penghubung antara komputer dengan modul XBee dan satu diantaranya berfungsi penghubung antara komputer Raspberry Pi dengan sensor.
5. Satu unit Arduino Mega berfungsi sebagai notifikasi sensor.

C. Analisa Perangkat Lunak

Kebutuhan perangkat lunak yang akan digunakan dalam pelaksanaan penelitian ini adalah:

1. X-CTU

X-CTU adalah sebuah perangkat lunak yang berfungsi untuk mengkonfigurasi suatu modul XBee agar dapat berkomunikasi melalui AT Command pada terminal X-CTU.

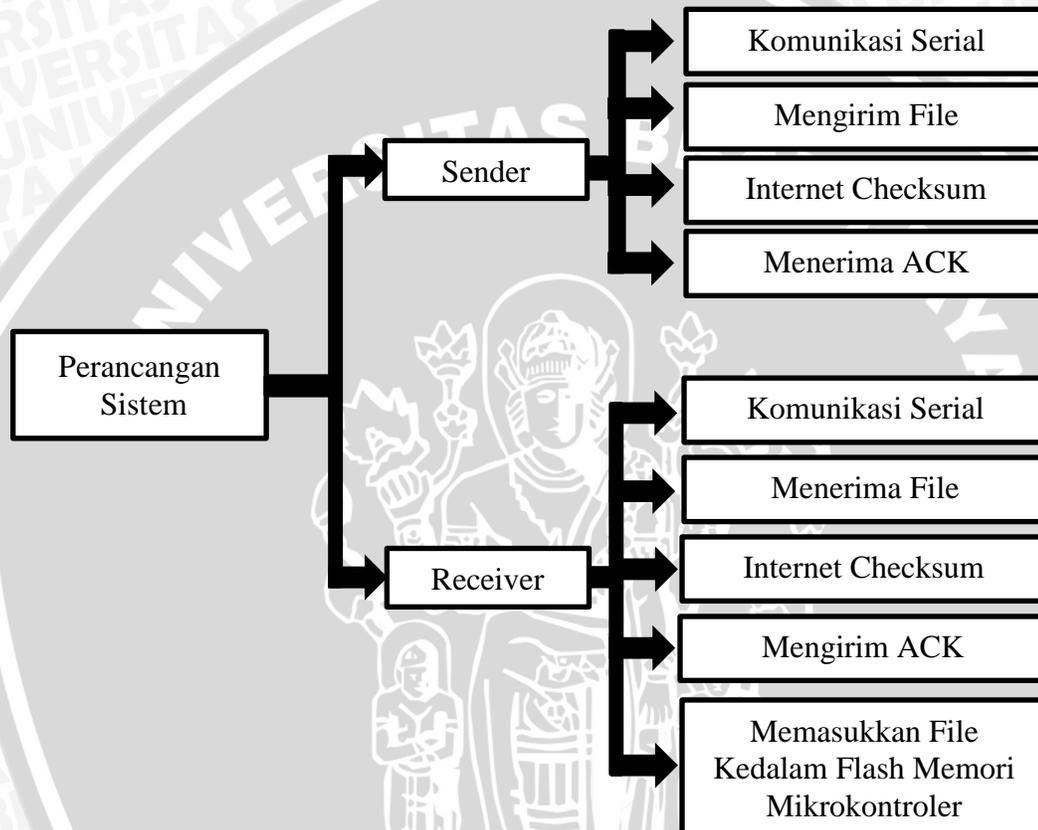
2. AVRDUDE

AVRDUDE adalah sebuah perangkat lunak berbasis *command-line* atau *console* yang berfungsi untuk memasukkan kode perintah suatu program kedalam mikrokontroler. Pada penelitian ini AVRDUDE digunakan untuk

memasukkan file kode hex yang diterima oleh Raspberry Pi menuju mikrokontroler/sensor.

3.1.3. Perancangan

Perancangan aplikasi dilakukan setelah semua kebutuhan sistem diperoleh melalui tahap analisis kebutuhan. Perancangan aplikasi digambarkan pada diagram blok gambar 3.2 berikut ini:



Gambar 3.2 Diagram Blok Perancangan Sistem

Sumber : [Perancangan]

Pada perancangan sistem, hal yang perlu dirancang pada sisi komputer sender meliputi komunikasi serial antara modul XBee dengan komputer, pengiriman file, algoritma pengecekan menggunakan Internet Checksum, dan menerima ACK yang dikirim oleh komputer Receiver. Sedangkan pada sisi komputer receiver yang perlu dirancang adalah komunikasi serial antara komputer dengan modul XBee, menerima file yang dikirim oleh komputer sender, algoritma

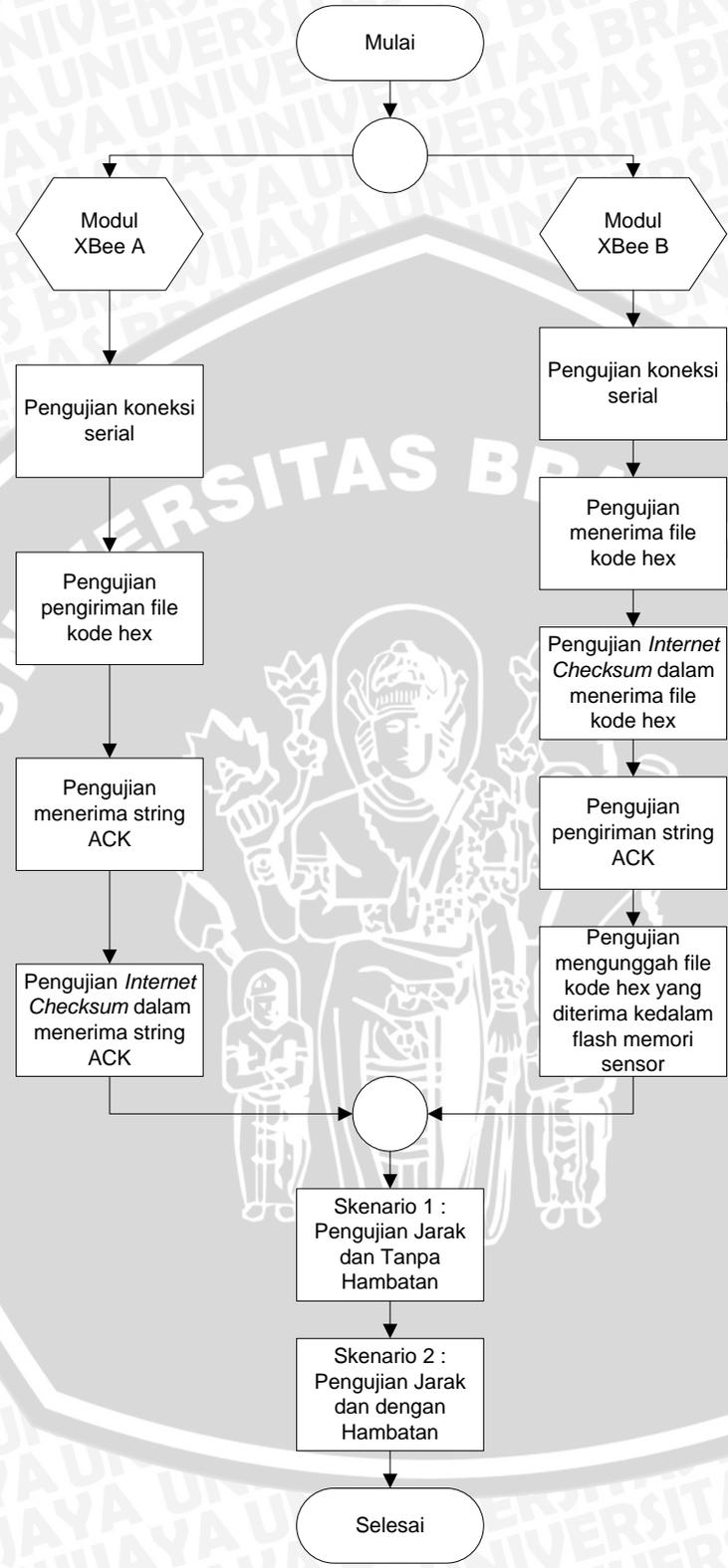
pengecekan Internet Checksum, mengirim ACK kepada komputer sender, dan memasukkan file kode hexa kedalam flash memori dari mikrokontroler.

3.1.4. Implementasi

Implementasi perangkat lunak mengacu kepada perancangan perangkat lunak. Perancangan perangkat lunak digunakan sebagai dasar acuan implementasi perangkat lunak yang akan dilakukan. Implementasi perangkat lunak diawali dengan melakukan instalasi dan konfigurasi perangkat lunak. Implementasi perangkat lunak dilakukan dengan menggunakan bahasa pemrograman Python. Pada bagian ini ada beberapa konfigurasi yang dilakukan antara lain konfigurasi modul XBee. Setelah modul XBee sudah dapat berkomunikasi antara satu dengan yang lainnya, tahapan berikutnya adalah instalasi *library* Python yang diantaranya adalah *library* Pyserial dan Xmodem. Tahap terakhir dari implementasi adalah implementasi mengunggah file kode hex kedalam flash memori mikrokontroler dengan memanfaatkan perangkat lunak AVRDUDE berbasis *command-line* atau *console*.

3.1.5. Pengujian dan Analisis

Pada tahapan ini dilakukan pengujian terhadap pengiriman file kode hex dan proses unggah file kode hex kedalam flash memori dari mikrokontroler dengan mengeluarkan setiap fungsi yang digunakan didalam perancangan aplikasi pengiriman file. Setelah setiap fungsi dikeluarkan dan diuji, diuji pula kinerja dari keseluruhan fungsi aplikasi menggunakan skenario pengujian dengan membandingkan pengiriman filenya dengan jarak jangkauan antara 1 hingga 100 meter, hambatan dan tanpa hambatan. Dengan demikian akan didapatkan kinerja dari setiap fungsi apakah dapat berjalan baik atau tidak dan juga dapat dilihat kerja dari aplikasi secara global. Tahap pengujian pengiriman file dan pembaharuan dari sebuah sensor dapat dilihat pada Gambar 3.3.



Gambar 3.3 Blok Diagram Alir Pengujian

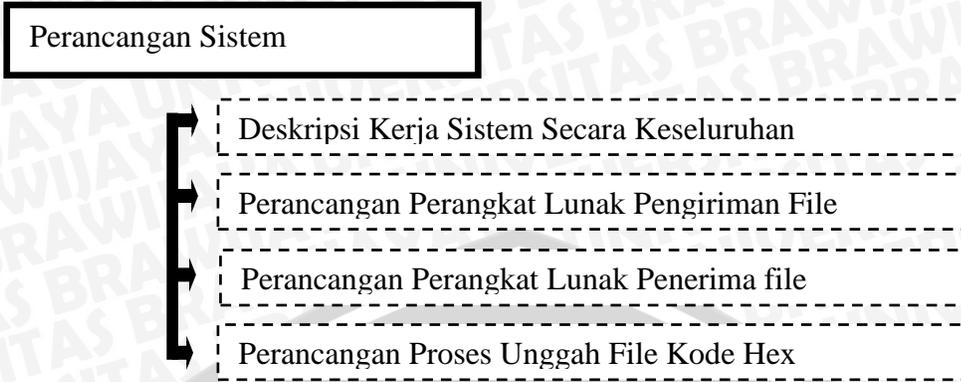
Sumber : [Perancangan]

3.1.6. Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan perangkat lunak, implementasi perangkat lunak, dan pengujian perangkat lunak telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem yang dibangun. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakan penulisan serta untuk memberikan pertimbangan atas pengembangan perangkat lunak lebih lanjut.

3.2. Perancangan

Perancangan ini bertujuan menjelaskan rancangan sistem komunikasi pengiriman data dengan protokol zigbee menggunakan mekanisme internet checksum untuk mengubah parameter modul arduino. Dua hal yang akan dibahas dari perancangan sistem ini yaitu proses analisis kebutuhan dan proses perancangan perangkat lunak. Tahap analisis kebutuhan terdiri atas dua tahap yaitu melakukan penjabaran tentang gambaran umum pengiriman file menggunakan protokol ZigBee beserta proses unggah file kode hex kedalam sensor dan membuat daftar kebutuhan perangkat lunak dan perangkat keras yang dibutuhkan. Proses perancangan perangkat lunak memiliki tiga tahap yaitu perancangan pemrograman pengiriman file, perancangan pemrograman penerima file dan perancangan proses unggah file kode hex kedalam sensor dengan memanfaatkan software *AVRDUDE*. Berikut pohon analisa dan perancangan sistem dijelaskan pada gambar 3.4.

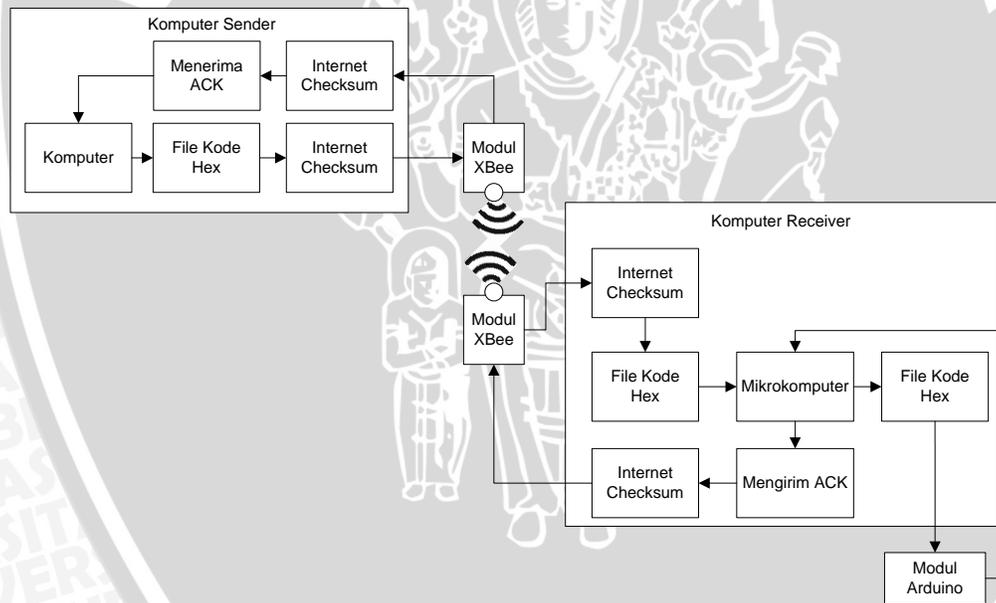


Gambar 3.4 Pohon Analisa dan Perancangan Sistem

Sumber : [Implementasi]

3.2.1. Diskripsi Kerja Sistem Secara Keseluruhan

Diagram alir menggunakan notasi-notasi untuk menggambarkan arus data yang membantu dalam proses memahami jalannya aplikasi. Gambar 3.5 menunjukkan diagram blok aplikasi secara umum.



Gambar 3.5 Diagram Blok Perancangan Aplikasi Secara Umum

Sumber : [Perancangan]

Dalam gambar diatas di jelaskan bagaimana cara aplikasi ini bekerja. Pertama, komputer mengirim file kode hex dengan algoritma pengecekan melalui modul XBee. Sistem kemudian akan mengirim file kode hex tersebut menuju modul Xbee yang lain dengan memanfaatkan *library* Pyserial dan Xmodem dari Python dan dicek dengan algoitma pengecekan yaitu Internet Checksum. Setelah

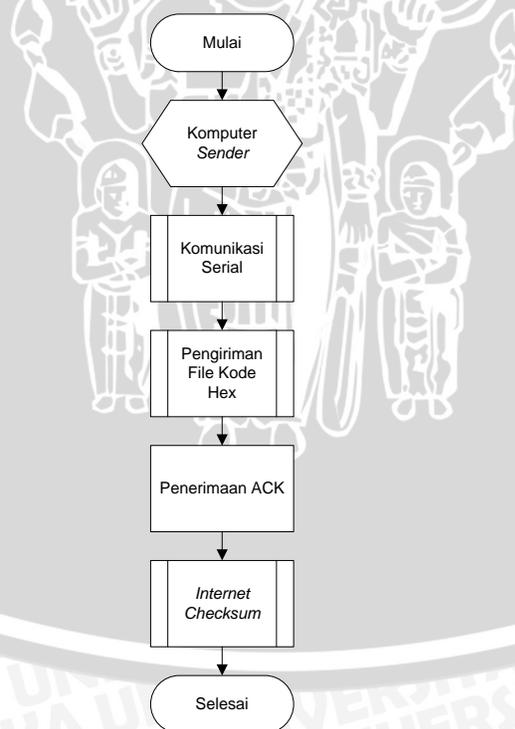
terkirim, komputer receiver mengirim ACK yang menunjukkan bahwa pengiriman telah berhasil menuju komputer sender. Kerja dari komputer receiver tidak sampai disitu, selanjutnya file kode hexa yang diterima akan diteruskan menuju flash memori dari sebuah mikrokontroler.

3.2.2. Perancangan Perangkat Lunak Pengiriman File

Perancangan pengiriman file merupakan bagian perancangan perangkat lunak. Pengiriman file kode hex dilakukan dengan modul XBEE memanfaatkan *library* Pyserial dalam komunikasinya dan *library* XModem dalam pengiriman filenya. Disamping itu menggunakan algoritma *Internet Checksum* dalam pengecekan datanya.

A. Proses Pengiriman file

Proses pengiriman file dilakukan oleh komputer *sender* dengan beberapa fungsi diantaranya adalah komunikasi serial, pengiriman file kode hex, penerimaan ACK, dan proses pengecekan menggunakan algoritma *Internet Checksum*. Diagram alir dari proses pengiriman file dijelaskan pada gambar 3.6.



Gambar 3.6 Sistem Plan Proses Pengiriman File

Sumber : [Perancangan]

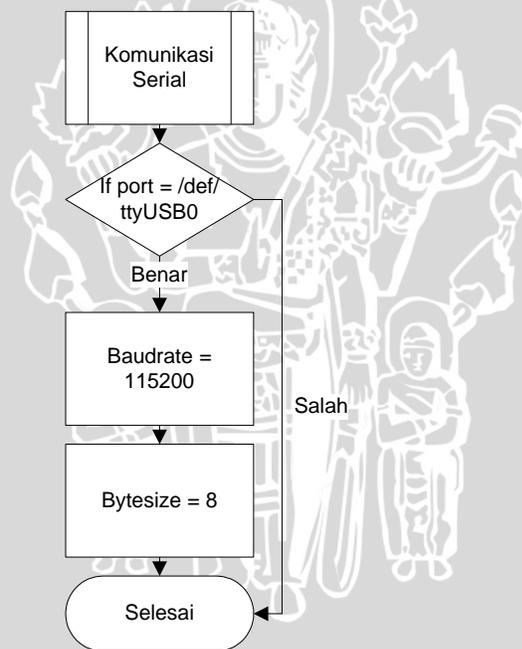
B. Komputer *Sender*

Komputer *sender* adalah komputer programmer sensor untuk pembuatan program pengiriman file kode hex menuju komputer *receiver*.

C. Komunikasi Serial

Agar suatu komputer dapat berkomunikasi dengan sebuah modul XBee dibutuhkan komunikasi melalui serial port. Python termasuk yang menyediakan dukungan untuk komunikasi secara serial port. Sebelum dapat melakukan komunikasi secara serial port pada python dibutuhkan sebuah proses instalasi *library* Python yang mendukung komunikasi serial port, yaitu PySerial.

PySerial adalah *library* Python siap pakai dan gratis yang dibuat untuk memudahkan *programmer* dalam membuat program komunikasi data serial RS232 dalam bahasa Python. Perancangan proses komunikasi serial dapat dilihat pada gambar 3.7.



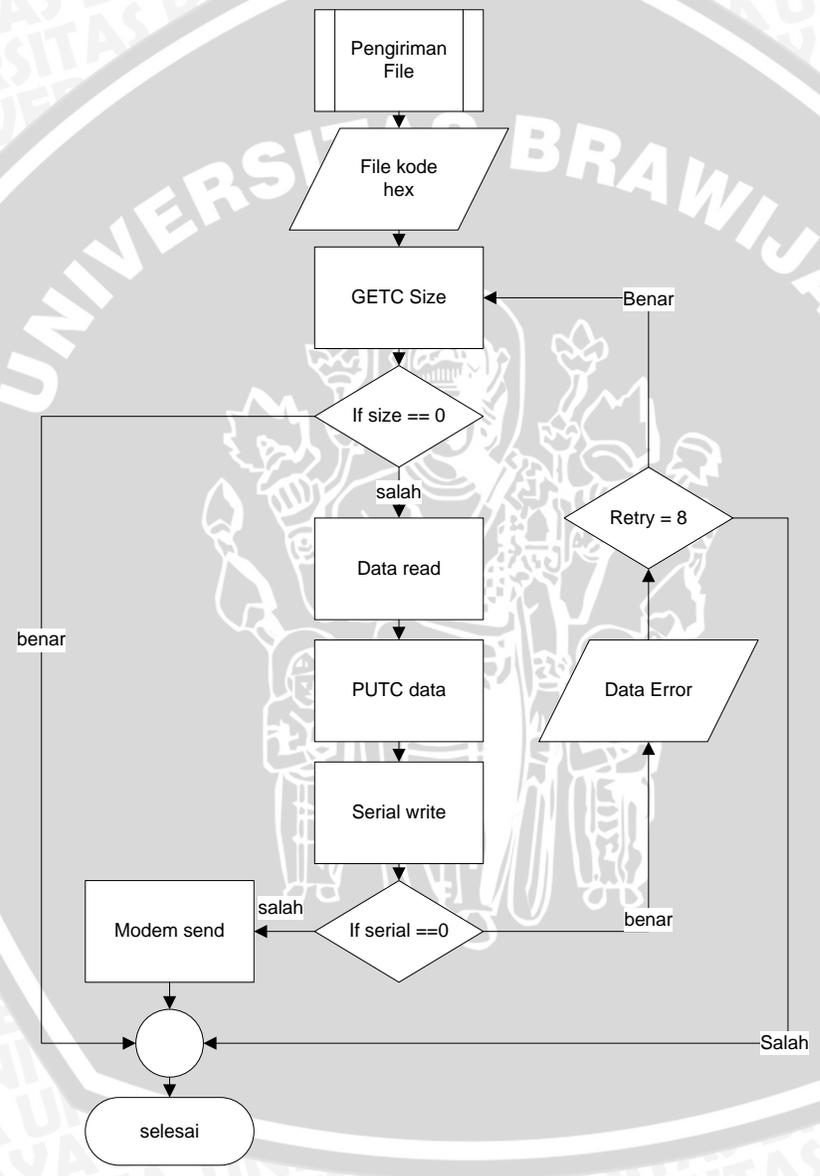
Gambar 3.7 Sistem Plan Proses Komunikasi Serial

Sumber : [Perancangan]

Pada proses komunikasi serial dapat dilihat bahwa port berada pada alamat `/dev/ttyUSB0`, jika benar maka baudrate di-*setting* 115200, dan bytesize tiap pengiriman adalah 8 bit, namun jika salah sistem langsung *terminated*.

D. Pengiriman File Kode Hex

Pada proses pengiriman file kode hex pada penelitian ini memanfaatkan *library* Python XModem. Proses dari kerja XModem adalah peralatan yang mentranslasikan data digital dari komputer ke sinyal analog yang akan dilewatkan jalur telepon. XModem di sisi yang lain, mengkonversikan sinyal analog yang diterima dari jalur telepon menjadi data yang akan dibaca oleh komputer. Proses kerja dari *library* XModem dapat dilihat pada gambar 3.8.



Gambar 3.8 Sistem Plan Proses Pengiriman File

Sumber : [Perancangan]

Pada proses pengiriman file, dapat dilihat file yang dikirim adalah file kode hex. Terdapat fungsi *getc* dan *putc*, adalah fungsi untuk membaca *size* dan

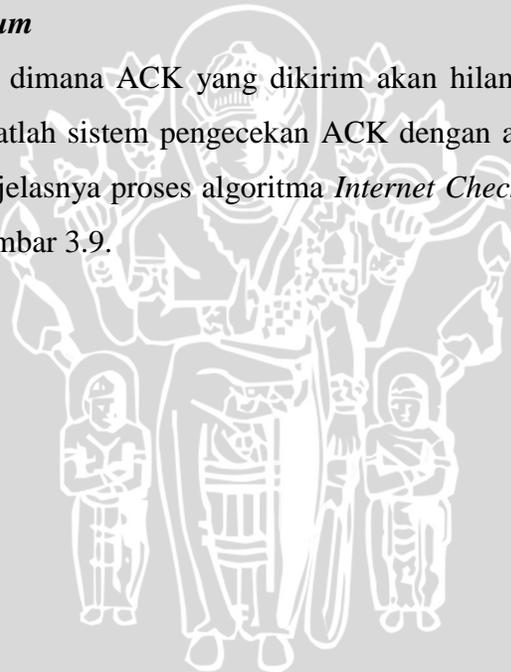
data dari file kode hex. Pertama sistem akan membaca besar *size* dari file kemudian sistem baru membaca panjang data, jika besar *size* = 0 maka sistem langsung selesai. Selanjutnya sistem memasukkan data yang telah dibaca tadi menuju serial dan langsung dikirim melalui modem/modul XBee. Akan tetapi jika port serial tidak ditemukan sistem akan memberikan notifikasi bahwa telah terjadi kesalahan dan akan mengulanginya lagi sebanyak 8 kali, jika tetap tidak ditemukan port serialnya, maka sistem akan *terminatted*.

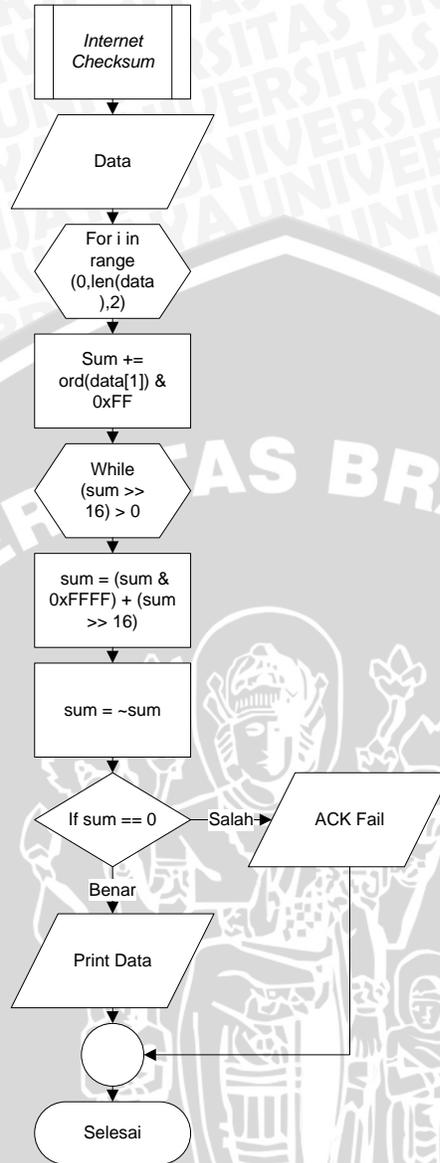
E. Penerima ACK

Proses penerimaan ACK adalah proses dimana setelah pengiriman file, komputer *receiver* mengirim ACK berupa string kepada komputer *sender* agar diketahui bahwa file yang telah dikirim telah diterima.

F. Internet Checksum

Ada suatu kasus dimana ACK yang dikirim akan hilang, oleh sebab itu pada penelitian ini dibuatlah sistem pengecekan ACK dengan algoritma *Internet Checksum*. Untuk lebih jelasnya proses algoritma *Internet Checksum* pada sistem ini dapat dilihat pada gambar 3.9.





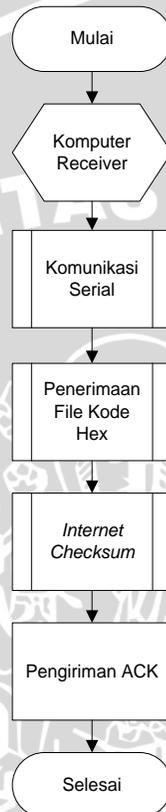
Gambar 3.9 Sistem Plan Proses Algoritma *Internet Checksum*

Sumber : [Perancangan]

Pada gambar 3.10 proses algoritma *Internet Checksum* diatas dapat dilihat bahwa sistem menerima data setiap 8 bit sebanyak 2 kali sehingga total 16 bit, setiap 16 bit yang diterima akan ditambahkan dengan 16 bit berikutnya lalu hasilnya akan di komplenkan dan ditambahkan dengan hasil sebelum dikomplemen, dan didapatlah hasil 0. Jika hasil yang didapat tidak bernilai 0 maka string ACK yang diterima salah, dan jika hasil yang didapat bernilai 0 maka hasil yang diterima sama dengan hasil yang dikirim.

3.2.3. Perancangan Perangkat Lunak Penerima File

Proses pengiriman file dilakukan oleh komputer *receiver* dengan beberapa fungsi diantaranya adalah komunikasi serial, penerimaan file kode hex, proses pengecekan menggunakan algoritma *Internet Checksum*, dan pengiriman ACK. Diagram alir dari proses penerimaan file dijelaskan pada gambar 3.10.



Gambar 3.10 Sistem Plan Proses Penerimaan File

Sumber : [Perancangan]

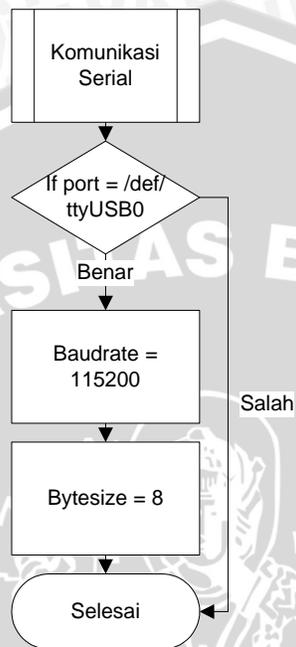
A. **Komputer Receiver**

Komputer *receiver* adalah komputer Raspberry Pi yang berfungsi sebagai penerima data yang dikirim oleh komputer *sender* melalui modul XBee.

B. **Komunikasi Serial**

Agar suatu komputer dapat berkomunikasi dengan sebuah modul XBee dibutuhkan komunikasi melalui serial port. Python termasuk yang menyediakan dukungan untuk komunikasi secara serial port. Sebelum dapat melakukan komunikasi secara serial port pada python dibutuhkan sebuah proses instalasi *library* Python yang mendukung komunikasi serial port, yaitu PySerial.

PySerial adalah *library* Python siap pakai dan gratis yang dibuat untuk memudahkan *programmer* dalam membuat program komunikasi data serial RS232 dalam bahasa Python. Perancangan proses komunikasi serial dapat dilihat pada gambar 3.11.



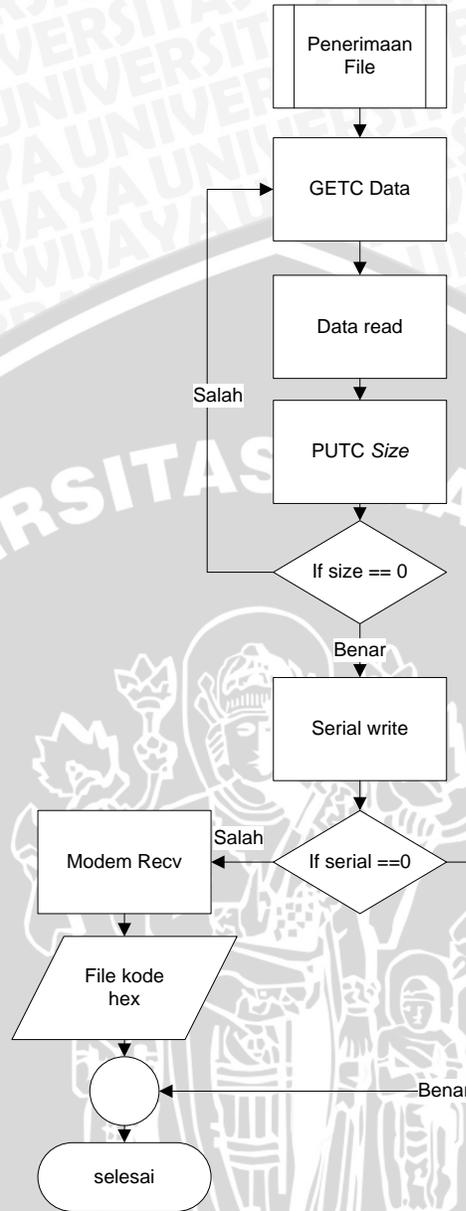
Gambar 3.11 Sistem Plan Proses Komunikasi Serial

Sumber : [Perancangan]

Pada proses komunikasi serial dapat dilihat bahwa port berada pada alamat `/dev/ttyUSB0`, jika benar maka baudrate disetting 115200, dan bytesize tiap pengiriman adalah 8 bit, namun jika salah sistem langsung *terminated*.

C. Penerimaan File Kode Hex

Pada proses penerimaan file kode hex pada penelitian ini memanfaatkan library python XModem. Proses dari kerja XModem adalah peralatan yang mentranslasikan data digital dari komputer ke sinyal analog yang akan dilewatkan jalur telepon. XModem di sisi yang lain, mengkonversikan sinyal analog yang diterima dari jalur telepon menjadi data yang akan dibaca oleh komputer. Proses kerja dari penerimaan library XModem dapat dilihat pada gambar 3.12.



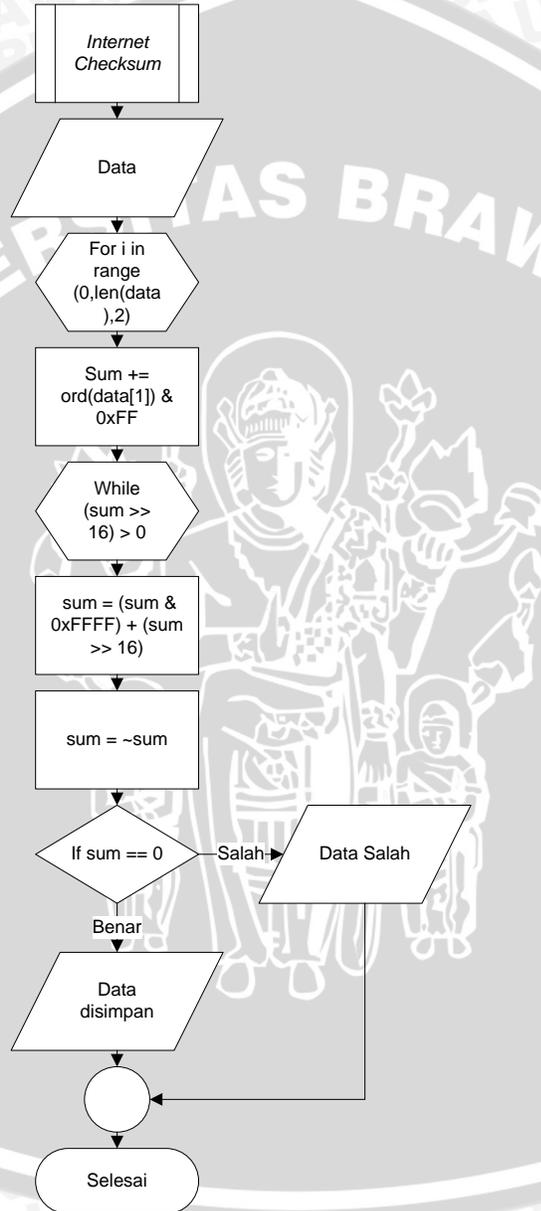
Gambar 3.12 Sistem Plan Proses Penerimaan File Kode Hex

Sumber : [Perancangan]

Proses awal dalam penerimaan file adalah membaca panjang dari data kemudian diambil besar *size* dari data tersebut, proses itu terus berulang hingga *size* bernilai 0. Setelah *size* bernilai 0, sistem akan koneksi serial, jika serial tidak terhubung/bernilai 0, maka sistem akan *terminated*. Akan tetapi jika serial terkoneksi maka data tersebut akan diterima.

D. Internet Checksum

Pada proses penerimaan file digunakan fungsi pengecekan untuk mengetahui jumlah bit yang dikirim sama dengan yang diterima. Pada penelitian kali ini digunakan algoritma *Internet Checksum* dalam pengecekan bit. Proses algoritma *Internet Checksum* dapat dilihat pada gambar 3.13.



Gambar 3.13 Sistem Plan Proses Algoritma *Internet Checksum*

Sumber : [Perancangan]

Pada gambar 3.13 proses algoritma *Internet Checksum* diatas dapat dilihat bahwa sistem menerima data setiap 8 bit sebanyak 2 kali sehingga total 16 bit,

setiap 16 bit yang diterima akan ditambahkan dengan 16 bit berikutnya lalu hasilnya akan di komplementkan dan ditambahkan dengan hasil sebelum dikomplement, dan didapatlah hasil 0. Jika hasil yang didapat tidak bernilai 0 maka paket data yang diterima salah, dan jika hasil yang didapat bernilai 0 maka hasil yang diterima sama dengan hasil yang dikirim dan pengiriman data berhasil.

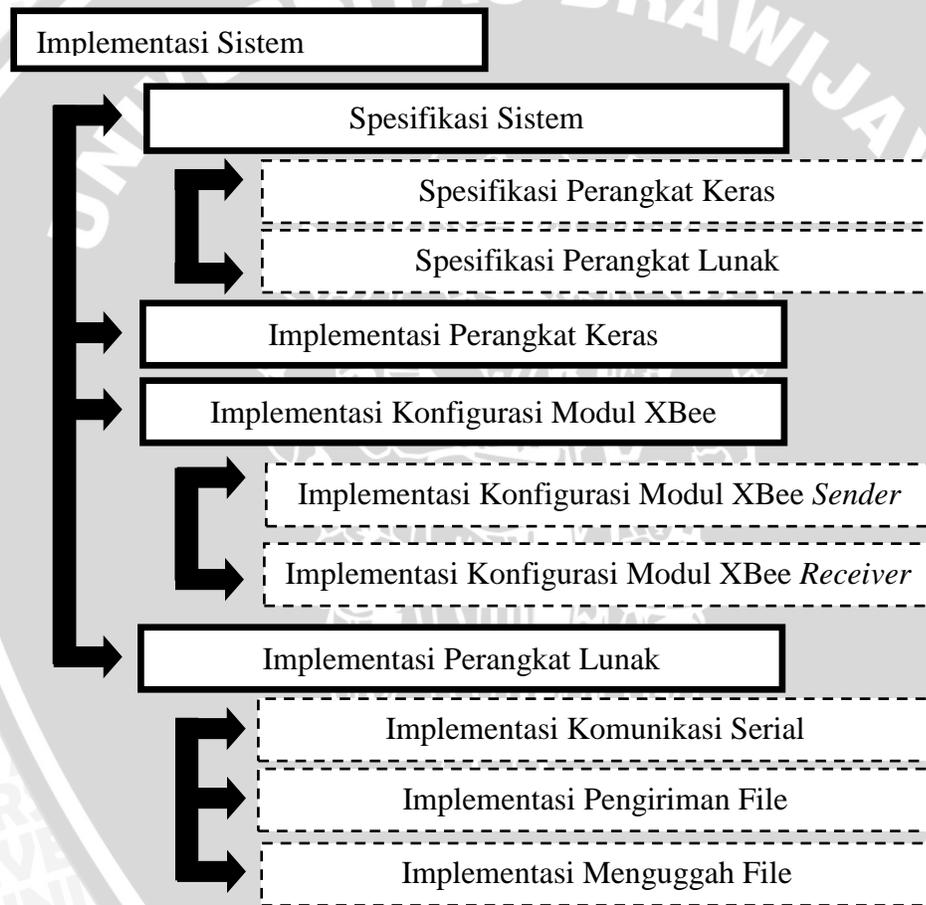
3.2.4. Perancangan Proses Unggah File Kode Hex

Pada proses unggah file memanfaatkan perangkat lunak *AVRDUDE* berbasis *command-line/console*. Pada penelitian ini file yang akan diupload adalah file kode hex menyalakan LED yang telah disimpan oleh komputer *receiver*. Sensor yang digunakan disini adalah mikrokontroler arduino mega digunakan untuk notifikasi sebagai parameter keberhasilan suatu proses unggah file kedalam flash memori pada suatu mikrokontroler.



BAB IV IMPLEMENTASI

Pada bab implementasi, akan dibahas pembuatan sistem yang mampu melakukan *upgrade* firmware/parameter sensor dengan memanfaatkan protokol ZigBee IEEE 802.15.4. Sesuai dengan rancangan pada Bab III, sistem ini dibangun terdiri atas penjelasan tentang spesifikasi sistem, implementasi perangkat keras, implementasi konfigurasi modul XBee, implementasi perangkat lunak.



Gambar 4.1 Pohon Implementasi

Sumber : [Implementasi]

4.1. Spesifikasi Sistem

Perangkat lunak sistem komunikasi pengiriman data dengan protokol zigbee menggunakan mekanisme internet checksum untuk mengubah parameter modul arduino dikembangkan dalam lingkungan implementasi yang terdiri dari perangkat keras dan perangkat lunak.

4.1.1. Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang dipakai dalam proses pengembangan dijelaskan pada Tabel 4.1 sampai Tabel 4.4.

Tabel 4.1 Spesifikasi Perangkat Keras Komputer

Sumber : [Implementasi]

Laptop Dell Inspiron N4050	
<i>Processor</i>	Intel Core i3-2310M (3M Cache, 2.10 GHz)
<i>Memory (RAM)</i>	4GB DDR3
<i>Harddisk</i>	500GB SATA
<i>Motherboard</i>	Dell Inspiron Intel Motherboard
<i>Graphic Card</i>	Integrated Intel(R) HD Graphics
<i>Monitor</i>	14.1 Inch Wide LED HD (1366 x 768) Truelife

Tabel 4.2 Spesifikasi Perangkat Keras Mikrokomputer

Sumber : [Implementasi]

Raspberry Pi Model B	
<i>CPU</i>	700 MHz ARM1176JZF-S core (ARM11 family)
<i>Memory (RAM)</i>	256 MB
<i>Harddisk</i>	8 GB Sand Disk Class 10
<i>Graphic</i>	Raw LCD Panels via DSI 14 HDMI resolutions from 1920×1200

Tabel 4.3 Spesifikasi Perangkat Keras ZigBee (IEEE 802.15.4)

Sumber : [Implementasi]

XBee-Pro RF Module	
<i>Frequency</i>	2.4 Ghz
<i>Indoor/Urban</i>	Up to 100 m
<i>Outdoor</i>	Up to 200 m
<i>Transmit Power</i>	100 mW (20 dBm) EIRP
<i>Receiver Sensitivity</i>	-100 dBm
<i>RF Data Rate</i>	50,000 bps
<i>TX Current Power</i>	270 mA
<i>RX Current</i>	55 mA

Tabel 4.4 Spesifikasi Perangkat Keras Mikrokontroler

Sumber : [Implementasi]

Arduino Mega 2560	
<i>Operating Voltage</i>	5 V
<i>Digital I/O Pins</i>	54 (of which 15 provide PWM output)
<i>Analog Input Pins</i>	16
<i>Flash Memory</i>	256 KB of which 8 KB used by bootloader
<i>SRAM</i>	8 KB
<i>Clock Speed</i>	16 Hz

4.1.2. Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang dipakai dalam proses pembuatan *prototype* sebuah sistem yang dapat melakukan *upgrade* firmware/parameter pada sensor dengan memanfaatkan komunikasi nirkabel berdaya rendah (ZigBee IEEE 802.15.4) dijelaskan pada tabel 4.5.

Tabel 4.5 Spesifikasi Perangkat Lunak Komputer

Sumber : [Implementasi]

Laptop Dell Inspiron N4050	
<i>Operating System</i>	Ubuntu 11.10 (Oneiric Ocelot)
<i>Programming Language</i>	Python
<i>Compiler</i>	Geany, Terminal

Tabel 4.6 Spesifikasi Perangkat Lunak Mikrokomputer

Sumber : [Implementasi]

Raspberry Pi Model B	
<i>Operating System</i>	Raspbian 'Wheezy'
<i>Programming Language</i>	Python
<i>Compiler</i>	Terminal
<i>Upload Program</i>	AVRDUDE

Tabel 4.7 Spesifikasi Perangkat Lunak ZigBee (IEEE 802.15.4)

Sumber : [Implementasi]

XBee-Pro RF Module	
<i>Driver</i>	FTDI Encoder Driver
<i>Configuration</i>	X-CTU, AT Command

Tabel 4.8 Spesifikasi Perangkat Keras Mikrokontroler

Sumber : [Implementasi]

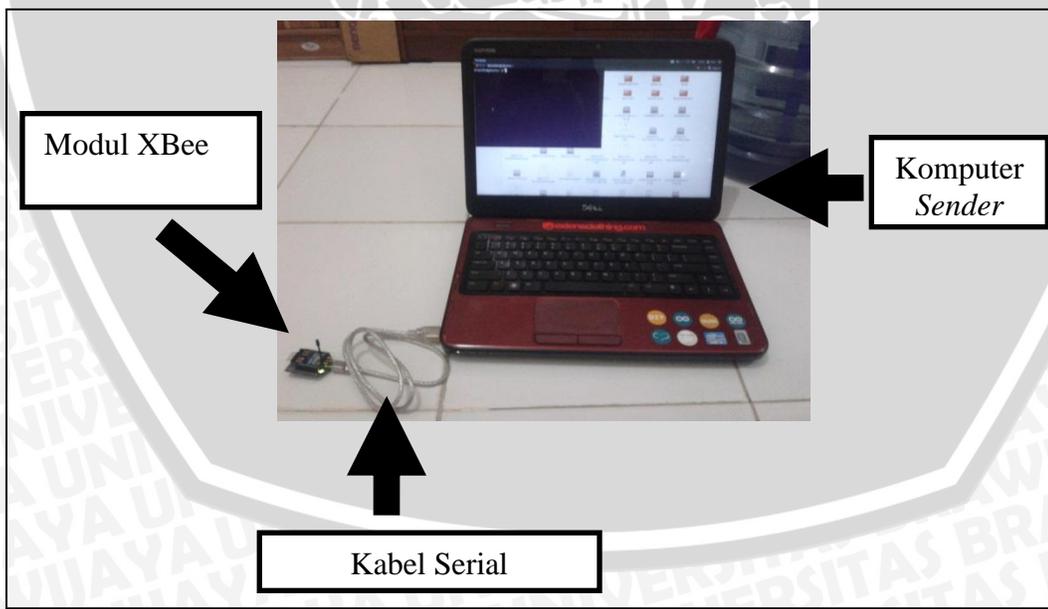
Arduino Mega 2560	
<i>Driver</i>	FTDI Encoder Driver
<i>Compiler</i>	Arduino IDE

4.2. Implementasi Perangkat Keras

Implementasi perangkat keras dilakukan seperti BAB III pada perancangan arsitektur aplikasi. Terdapat 2 implementasi pada perangkat keras, yaitu implementasi perangkat keras komputer *sender* dan implementasi perangkat keras komputer *receiver*. Implementasi perangkat keras dilakukan untuk mendukung agar sebuah aplikasi yang dibangun mampu meng-*upgrade* firmware. Berikut gambar implementasi arsitektur perangkat keras.

4.2.1. Implementasi Perangkat Keras Komputer *Sender*

Pada implementasi perangkat keras komputer *sender* dilakukan dengan menggunakan 1 buah komputer/laptop, 1 kabel serial, dan 1 modul XBee. Implementasi perangkat keras komputer *sender* dapat dilihat pada gambar 4.2.

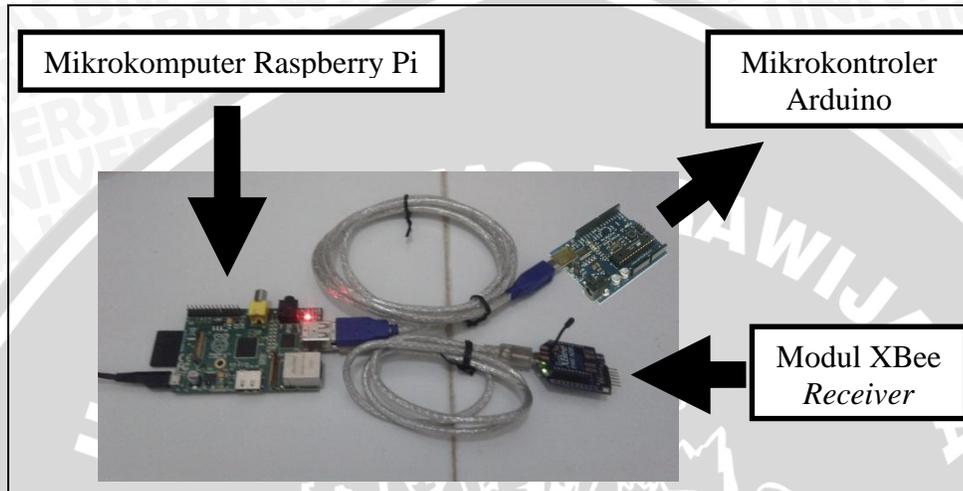


Gambar 4.2 Perangkat Keras Komputer *Sender*

Sumber : [Implementasi]

4.2.2. Implementasi Perangkat Keras Komputer *Receiver*

Pada implementasi perangkat keras komputer *receiver* dilakukan dengan menggunakan 1 buah mikrokomputer, 2 kabel serial, 1 modul XBee, dan 1 mikrokontroler. Implementasi perangkat keras komputer *receiver* dapat dilihat pada gambar 4.3.



Gambar 4.3 Perangkat Keras Komputer *Receiver*

Sumber : [Implementasi]

4.3. Implementasi Konfigurasi Modul XBee

Implementasi konfigurasi modul XBee dilakukan untuk memberikan alamat pada modul XBee agar modul XBee dapat mengetahui node yang akan dituju. Konfigurasi modul XBee sendiri dilakukan dengan memanfaatkan perintah *AT-Command* pada perangkat lunak X-CTU. Pada setiap konfigurasi modul XBee terdapat beberapa parameter yang harus dirubah diantaranya sebagai berikut:

1. *Baudrate* : Kecepatan data digital yang dikirim
2. *PAN ID* : Mengatur jaringan XBee untuk berkomunikasi
3. *Serial High* : Serial number pada tiap node modul XBee
4. *Serial Low* : Serial number pada tiap tipe modul XBee
5. *Destination High* : Serial number tujuan pada node modul XBee
6. *Destination Low* : Serial number tujuan pada tipe modul XBee
7. *Node Identifier* : Nama dari sebuah node modul XBee

Pada penelitian “Komunikasi Pengiriman Data Pada Zigbee Dengan Mekanisme Internet Checksum Untuk Mengubah Parameter Dari Mikrokontroler”

menggunakan dua modul XBee. Modul XBee pertama *node identifier* bernama ROUTER, dan modul XBee kedua *node identifier* bernama COORDINATOR. Berikut implementasi konfigurasi pada modul XBee pertama dan kedua.

4.3.1. Konfigurasi Modul XBee *Sender*

Modul XBee pertama digunakan sebagai ROUTER yang berfungsi untuk mengirim sebuah data menuju komputer *receiver*. Implementasi konfigurasi modul XBee pertama dapat dilihat pada gambar 4.4.

```
→ +++ // Berfungsi membuka AT
Command
← OK
→ ATBD 7 // Menyimpan Baudrate 115200
← OK
→ ATID 123 // Menyimpan PAN ID 123
← OK
→ ATSH // Menampilkan Serial High
13A200
← 13A200
→ ATSL // Menampilkan Serial Low
40865547
← 40865547
→ ATDH 13A200 // Menyimpan Destination High
13A200
← OK
→ ATDL 40865524 // Menyimpan Destination High
40865524
← OK
→ ATNI ROUTER // Menyimpan Node Identifier
Router
← OK
→ ATWR // Menyimpan seluruh
konfigurasi di XBee
← OK
→ ATCN // Keluar dari AT-Command
← OK
```

Gambar 4.4 Konfigurasi Modul XBee *Sender*

Sumber : [Implementasi]

4.3.2. Konfigurasi Modul XBee Receiver

Modul XBee kedua digunakan sebagai COORDINATOR yang berfungsi untuk menerima sebuah data dari komputer *sender*. Implementasi konfigurasi modul XBee pertama dapat dilihat pada gambar 4.5.

```
→ +++ // Berfungsi membuka AT
Command
← OK
→ ATBD 7 // Menyimpan Baudrate 115200
← OK
→ ATID 123 // Menyimpan PAN ID 123
← OK
→ ATSH // Menampilkan Serial High
13A200
← 13A200
→ ATSL // Menampilkan Serial Low
40865524
← 40865524
→ ATDH 13A200 // Menyimpan Destination High
13A200
← OK
→ ATDL 40865547 // Menyimpan Destination High
40865547
← OK
→ ATNI COORDINATOR // Menyimpan Node Identifier
COORDINATOR
← OK
→ ATWR // Menyimpan seluruh
konfigurasi di XBee
← OK
→ ATCN // Keluar dari AT-Command
← OK
```

Gambar 4.5 Konfigurasi Modul XBee Receiver

Sumber : [Implementasi]

4.4. Implementasi Perangkat Lunak

Implementasi pembuatan sistem yang mampu merubah parameter sensor secara nirkabel dengan memanfaatkan komunikasi nirkabel berdaya

rendah/ZigBee (IEEE 802.15.4) menggunakan komunikasi serial dan XModem untuk pengiriman filenya berdasarkan perancangan sistem pada BAB III. Pada implementasi perangkat lunak terdapat lima bagian yaitu implementasi komunikasi serial, implementasi pengiriman file, implementasi algoritma pengecekan file, implementasi pengiriman string ack, implementasi algoritma pengecekan string ack, dan implementasi mengunggah file kedalam mikrokontroler. Bahasa yang digunakan adalah bahasa pemrograman Python.

4.4.1. Implementasi Komunikasi Serial

Pada proses implementasi komunikasi serial, terdapat dua proses yaitu proses integrasi dan proses konfigurasi serial port. Implementasi komunikasi serial digunakan pada kedua modul XBee dengan kode program yang sama.

A. Proses Integrasi

Pada proses integrasi untuk bisa menggunakan fungsi-fungsi pada *library* Pyserial harus impor terlebih dahulu agar kode yang dibuat dapat mengenali fungsi-fungsi yang terdapat pada *library* PySerial. Untuk mengimpor menggunakan syntax dibawah ini:

1.	<code>import serial</code>
----	----------------------------

Gambar 4.6 Impor *Library* Pyserial

Sumber : [Implementasi]

B. Proses Konfigurasi Serial Port

Proses konfigurasi serial port digunakan untuk memberikan alamat pada serial yang digunakan dan beberapa parameter agar serial port bekerja. Pada implementasi komunikasi serial, untuk mengkonfigurasi dan memanggil serial port digunakan kode dibawah ini:

1.	<code>s = serial.Serial(port='/dev/ttyUSB0', baudrate=115200,</code>
2.	<code>bytesize=8, timeout=None, xonxoff=False, rtscts=100)</code>
3.	<code>s.open()</code>
4.	<code>s.close()</code>

Gambar 4.7 Konfigurasi Serial Port

Sumber : [Implementasi]

Penjelasan implementasi kode program pada gambar 4.7 yaitu:

1. Pada baris 1 dan 2 digunakan untuk mengkonfigurasi serial port dengan parameter sebagai berikut :
 - a. Alamat port berada pada port `/dev/ttyUSB0`.
 - b. Baudrate yang digunakan adalah 115200.
 - c. Bytesize yang dikirim sebanyak 8 bit.
 - d. Tidak terjadi timeout selama proses menunggu.
 - e. Fungsi `rtscts` bernilai 100 yang berarti jika paket di atas 100 byte maka mekanisme RTS/CTS akan di aktifkan. Paket di bawah 100 byte dapat dikirim tanpa mekanisme RTS/CTS.
2. Pada baris 3 berfungsi untuk membuka serial port.
3. Pada baris 4 berfungsi untuk menutup serial port.

4.4.2. Implementasi Pengiriman File

Pada proses implementasi pengiriman file, terdapat beberapa proses yaitu proses integrasi, proses pengiriman file, proses penerima file dan algoritma pengecekan, proses pengiriman string ACK dan algoritma pengecekan.

A. Proses Integrasi

Pada proses integrasi untuk bisa menggunakan fungsi-fungsi pada *library* XModem harus meng-*import* terlebih dahulu agar kode yang dibuat dapat mengenali fungsi-fungsi yang terdapat pada *library* XModem. Untuk meng-*import* menggunakan syntax dibawah ini:

1.	<code>xmodem import XMODEM</code>
----	-----------------------------------

Gambar 4.8 Import *Library* XModem

Sumber : [Implementasi]

B. Proses Pengiriman File

Pada proses pengiriman file, penulis memanfaatkan *library* XModem untuk pengiriman file. Agar dapat melakukan pengiriman file, langkah pertama adalah membuat sebuah fungsi `getc` dan `putc` untuk membaca suatu panjang data dan besaran suatu data.

1.	<code>def getc(size, timeout=3):</code>
2.	<code>w,t,f = select.select([s], [], [], timeout)</code>

```
3.         if w:
4.             data = s.read(size)
5.         else:
6.             data = None
7.         return data
8. def putc(data, timeout=3):
9.     w,t,f = select.select([], [s], [], timeout)
10.    if t:
11.        s.write(data)
12.    else:
13.        size = None
14.    return size
```

Gambar 4.9 Fungsi Membaca Data

Sumber : [Implementasi]

Penjelasan fungsi membaca data pada gambar 4.9 adalah:

1. Pada baris 1 terdapat fungsi bernama `getc` untuk membaca panjang data dengan *timeout* selama 3 detik.
2. Pada baris 2 terdapat variabel `w`, `t`, dan `f` yang berfungsi untuk membaca, menyimpan, dan pengecualian kondisi. Pada kode program, variabel yang berisi suatu nilai yang di eksekusi, dan variabel yang bernilai adalah variabel 'w' yaitu membaca suatu panjang data.
3. Pada baris 3 – 6 berfungsi untuk membaca panjang dari data.
4. Pada baris 7 berfungsi untuk mengembalikan suatu nilai kepada bagian program yang memanggil fungsi.
5. Pada baris 8 terdapat fungsi bernama `putc` untuk menyimpan besaran dari suatu data dengan *timeout* selama 3 detik.
6. Pada baris 9 terdapat variabel `w`, `t`, dan `f` yang berfungsi untuk membaca, menyimpan, dan pengecualian kondisi. Pada kode program, variabel yang berisi suatu nilai yang di eksekusi, dan yang bernilai adalah variabel 't' yaitu menyimpan suatu besaran data.
7. Pada baris 10 -13 berfungsi untuk menyimpan besaran suatu data dari panjang data.

8. Pada baris 14 berfungsi untuk mengembalikan suatu nilai kepada bagian program yang memanggil fungsi.

Setelah membuat fungsi `getc` dan `putc` langkah selanjutnya adalah membuat mekanisme pengiriman file dari modul XBee *sender*. Kode program dari mekanisme pengiriman file dapat dilihat pada gambar 4.10.

1.	<code>modem = XMODEM(getc, putc)</code>
2.	<code>stream = file('Blink.cpp.hex', 'rb')</code>
3.	<code>modem.send(stream, retry = 8, timeout = 40)</code>
4.	<code>stream.close()</code>

Gambar 4.10 Pengiriman File

Sumber : [Implementasi]

Penjelasan pengiriman file pada gambar 4.10 adalah:

1. Pada baris 1 terdapat variabel `modem` yang memanggil *library* XModem yang didalamnya terdapat fungsi `getc` dan `putc` yang telah dibuat sebelumnya.
2. Pada baris 2 terdapat variabel '`stream`' yang berfungsi untuk membuka file yang akan dikirim.
3. Pada baris 3 `modem` mengirimkan file yang ada pada variabel `stream`. `retry` berfungsi untuk melakukan perulangan sebanyak 8 kali jika seandainya file tidak berhasil dikirim, dan mempunyai *timeout* selama 40 detik.
4. Pada baris 4 berfungsi untuk menutup variabel `stream`.

C. Proses Menerima File dan Algoritma Pengecekan

Setelah pengiriman file pada modul XBee *sender*, langkah selanjutnya dilakukan proses menerima file pada sisi modul XBee *receiver*. Selain proses menerima file, pada tahap ini terdapat proses algoritma pengecekan file.

Seperti halnya pada pengiriman file, pada tahap awal proses menerima file dibuat fungsi `getc` dan `putc` untuk membaca panjang dan besaran ukuran suatu data. Didalam fungsi `putc` terdapat juga proses algoritma pengecekan data yang berfungsi untuk menghitung besaran data apakah jumlah bit yang diterima oleh modul XBee *receiver* sama dengan jumlah bit yang dikirim oleh modul XBee *sender*. Fungsi `getc` dan `putc` dapat dilihat pada gambar 4.11.

```
1. def getc(size, timeout=3):
2.     w,t,f = select.select([s], [], [], timeout)
3.     if w:
4.         data = s.read(size)
5.     else:
6.         data = None
7.     return data
8. def putc(data, timeout=3):
9.     w,t,f = select.select([], [s], [], timeout)
10.    if t:
11.        pos = len(data)
12.        if (pos & 1):
13.            pos -= 1
14.            sum = ord(data[pos])
15.        else:
16.            sum = 0
17.        while pos > 0:
18.            pos -= 2
19.            sum += (ord(data[pos + 1])
20.            sum = (sum >> 16) + (sum & 0xffff)
21.            sum += (sum >>16)
22.            result = (~ sum) & 0xffff
23.            result = result >> 8|(( result &
0xff) << 8)
24.            return result
25.            s.write(data)
26.            s.flush()
27.            size = len(data)
28.        else :
29.            size = None
30.    return size
```

Gambar 4.11 Fungsi Membaca Data dan Algoritma Pengecekan

Sumber : [Implementasi]

Penjelasan proses fungsi membaca data dan algoritma pengecekan pada gambar 4.11 adalah :

1. Pada baris 1 terdapat fungsi bernama `getc` untuk membaca panjang data dengan *timeout* selama 3 detik.

2. Pada baris 2 terdapat variabel `w`, `t`, dan `f` yang berfungsi untuk membaca, menyimpan, dan pengecualian kondisi. Pada kode program, variabel yang berisi suatu nilai yang di eksekusi, dan variabel yang bernilai adalah variabel '`w`' yaitu membaca dari suatu panjang data.
3. Pada baris 3 – 6 berfungsi untuk membaca panjang dari data.
4. Pada baris 7 berfungsi untuk mengembalikan suatu nilai kepada bagian program yang memanggil fungsi.
5. Pada baris 8 terdapat fungsi bernama `putc` untuk menyimpan besaran dari suatu data dengan `timeout` selama 3 detik.
6. Pada baris 9 terdapat variabel `w`, `t`, dan `f` yang berfungsi untuk membaca, menyimpan, dan pengecualian kondisi. Pada kode program, variabel yang berisi suatu nilai yang di eksekusi, dan yang bernilai adalah variabel '`t`' yaitu menyimpan besaran dari suatu data.
7. Pada baris 10 – 24 adalah fungsi Algoritma pengecekan yaitu *Internet Checksum*.
8. Pada baris 11 – 16 berfungsi untuk membaca panjang dari sebuah data.
9. Pada baris 17 – 21 berfungsi untuk perulangan perhitungan panjang data yang diperoleh dan ditambahkan nilai panjang data yang kedua.
10. Pada baris 22 – 23 berfungsi untuk mengklomplemen hasil yang diperoleh pada penjumlahan pada baris 21.
11. Pada baris 24 berfungsi untuk mengembalikan suatu nilai kepada bagian program yang memanggil fungsi.
12. Pada baris 25 -29 berfungsi untuk menyimpan besaran suatu data dari panjang data.
13. Pada baris 30 berfungsi untuk mengembalikan suatu nilai kepada bagian program yang memanggil fungsi.

D. Proses Pengiriman String ACK dan Algoritma Pengecekan

Pada proses pengiriman string ACK berfungsi untuk mengirim sebuah notifikasi bahwa pengiriman file telah berhasil. Pengiriman string ACK dilakukan dari modul XBee *receiver* menuju modul XBee *sender*. Pada proses ini juga terdapat algoritma pengecekan berfungsi untuk mengetahui bahwa tidak

terjadinya paket yang hilang. Pada proses ini terdapat dua kode program yang berbeda antara modul XBee *receiver* sebagai pengirim string ACK dan modul XBee *sender* sebagai penerima string ACK. Kode program dari proses pengiriman string ACK, menerima string ACK dan algoritma pengecekan dapat dilihat pada gambar 4.12 dan gambar 4.13 .

1.	s.flush()
2.	s.write ("transfer file complete")

Gambar 4.12 Proses Pengiriman String ACK

Sumber : [Implementasi]

Proses pengiriman string ACK pada gambar 4.12 dilakukan pada modul XBee *receiver*. Penjelasan proses pada gambar 4.12 adalah :

1. Pada baris 1 berfungsi untuk mengirimkan semua output/data yang berada di internal *buffer* melalui serial port.
2. Pada baris 2 berfungsi untuk menulis string dan dikirimkan melalui port.

1.	def ichecksum(data, sum=0):
2.	for i in range(0,len(data),2):
3.	if i + 1 >= len(data):
4.	sum += ord(data[i]) & 0xFF
5.	else:
6.	w=((ord(data[i])<<8)&0xFF00)+(ord(data[i+1])&0xFF
7.	sum += w
8.	while (sum >> 16) > 0:
9.	sum = (sum & 0xFFFF) + (sum >> 16)
10.	sum = ~sum
11.	return sum & 0xFFFF
12.	s.flushInput()
13.	if data:
14.	check = ichecksum(data)
15.	csum = ichecksum(data, check)
16.	if csum == 0:
17.	print data
18.	else:
19.	print 'ACK fail'

Gambar 4.13 Proses Menerima ACK dan Algoritma Pengecekan

Sumber : [Implementasi]

Penjelasan proses menerima ACK dan algoritma pengecekan pada Gambar 4.13 adalah:

1. Pada baris 1 membuat fungsi untuk algoritma pengecekan yang bernama *ichecksum*.
2. Pada baris 2 – 11 adalah fungsi dari proses algoritma pengecekan yaitu *Internet Checksum*.
3. Pada baris 2 – 7 adalah fungsi perulangan untuk membaca panjang data yang diterima.
4. Pada baris 8 – 9 adalah fungsi perulangan untuk menambahkan jumlah 16 bit data pertama yang diterima dengan jumlah 16 bit data kedua yang diterima.
5. Pada baris 10 berfungsi untuk menkomplement hasil dari baris 8 - 9.
6. Pada baris 11 berfungsi untuk mengembalikan suatu nilai kepada bagian program yang memanggil fungsi.
7. Pada baris 12 berfungsi untuk menerima seluruh data yang ada pada *buffer*.
8. Pada baris 13 – 15 berfungsi untuk pemanggilan fungsi *ichecksum* untuk pengecekan data yang masuk.
9. Pada baris 16 – 19 berfungsi untuk mencetak string ACK jika pengecekan benar (sum bernilai 0) dan jika tidak akan mencetak 'ACK Fail'.

4.4.3. Implementasi Mengunggah File

Pada proses implementasi mengunggah file berfungsi untuk memasukkan file kode hex yang dikirim oleh modul XBee kedalam flash memori dari mikrokontroler. Pada penelitian "Komunikasi Pengiriman Data Dengan Protokol Zigbee Menggunakan Mekanisme Internet Checksum Untuk Mengubah Parameter Modul Arduino" digunakan mikrokontroler Arduino Uno sebagai sensor yang kondisi awalnya telah diformat (tanpa ada parameter yang dimasukkan). File kode hex adalah parameter yang dikirim kedalam flash memori untuk menyalakan LED pada mikrokontroler yang hanya untuk notifikasi bahwa proses unggah file kedalam flash memori telah berhasil. Proses mengunggah file memanfaatkan perangkat lunak AVRDUDE berbasis *command-line/console*.

Konfigurasi untuk menjalankan perangkat lunak AVRDUDE dapat dilihat pada gambar 4.14.

```
avrdude -p m32 -c stk500 -P /dev/ttyACM0 -U  
flash:w:Blink.cpp.hex:i -D
```

Gambar 4.14 Konfigurasi AVRDUDE

Sumber : [Implementasi]

Penjelasan dari gambar 4.14 konfigurasi AVRDUDE adalah :

1. avrdude : digunakan untuk memanggil program AVRDUDE.
2. -p m32 : menentukan jenis mikro yang akan diprogram, dalam hal ini arduino atmega32.
3. -c stk500 : menentukan jenis *programmer* yang digunakan, dalam hal ini *programmer* stk500.
4. -P /dev/ttyACM0 : menentuका port yang digunakan untuk berkomunikasi dengan programmer, dalam hal ini alamat port adalah /dev/ttyACM0.
5. -U : instruksi untuk membaca/menulis chip.
6. flash:w:Blink.cpp.hex:i : mengunggah file test.hex ke memori flash.
7. -D : untuk menon-aktifkan penghapusan chip sebelum proses programming.

4.5. Kendala dan Solusi

Dalam penelitian ini ada beberapa kendala yang dihadapi. Salah satu kendala yang cukup sulit adalah proses menerima file. Ketika modul XBee menerima file dalam bentuk hex data yang diterima terdapat karakter/symbol sebanyak 128 bit pada EOF(*End of File*), itu terjadi karena pada tidak terdapat *buffer* dalam pemanggilan fungsi `getc` dan `putc`.

Untuk mensiasati kendala yang dihadapi dalam proses menerima file digunakan fungsi *cleaned* pada bahasa pemrograman Python. Pada fungsi ini digunakan untuk menghapus 128 bit terakhir yang diterima oleh modul XBee.

BAB V

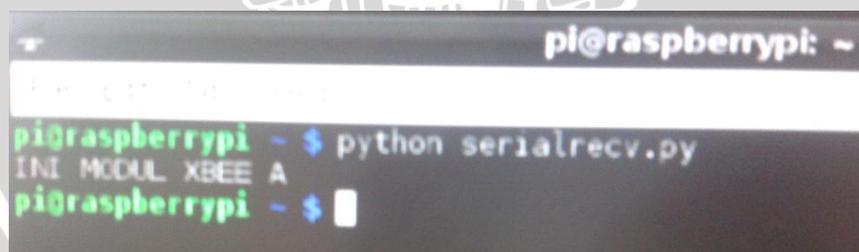
PENGUJIAN DAN ANALISIS

Pada bab ini akan dijelaskan bagaimana cara melakukan pengujian dan analisa terhadap sistem yang telah dibuat. Pengujian Unit dilakukan dalam 4 tahap, yang pertama pengujian komunikasi serial, pengujian pengiriman data, pengujian mengunggah file kedalam flash memori dari mikrokontroler, dan pengujian sistem secara keseluruhan. Pengujian dan analisis sistem perlu dilakukan untuk mengetahui apakah sistem yang telah dirancang dan diimplementasikan telah bekerja sesuai dengan tujuan dan dapat melakukan analisis terhadap kondisi jaringan yang digunakan untuk pengujian.

5.1. Pengujian Komunikasi Serial

Pengujian komunikasi serial digunakan untuk mengetahui apakah sistem yang dibangun telah dapat berkomunikasi antara modul XBee satu dengan yang lainnya. Pengujian dilakukan dengan mengirim string antar modul XBee. Prosedur pengujian komunikasi serial adalah sebagai berikut:

1. Modul XBee *sender* mengirim string “Ini Modul XBee A”.
2. Modul XBee *receiver* menerima string dari modul XBee *sender* dan menampilkan string yang diterima, lalu modul XBee *receiver* mengirim string balasan “OKE, Ini Modul XBee B”.



```
pi@raspberrypi: ~
python serialrecv.py
INI MODUL XBEE A
pi@raspberrypi: ~
```

Gambar 5.1 Komunikasi Serial Komputer *Receiver*

Sumber : [Pengujian]

3. Modul XBee *sender* menerima string dari modul XBee *receiver* dan menampilkan string yang diterima.

```
dimaldim@ubuntu: ~/6
dimaldim@ubuntu:~/6$ python serialsend.py
OKE, Ini Modul XBee B
dimaldim@ubuntu:~/6$
```

Gambar 5.2 Komunikasi Serial Komputer *Sender*

Sumber : [Pengujian]

4. Kondisi ketika sebuah serial port tidak ditemukan maka akan keluar notifikasi bahwa port tidak ditemukan.

```
dimaldim@ubuntu: ~/9
dimaldim@ubuntu:~/9$ python sender.py
ls: cannot access /dev/ttyUSB0: No such file or directory
dimaldim@ubuntu:~/9$
```

Gambar 5.3 Serial Port tidak Ditemukan

Sumber : [Pengujian]

5.2. Pengujian Pengiriman File

Pada pengujian pengiriman file dilakukan untuk mengetahui apakah sistem yang dibangun dapat mengirimkan file atau tidak. Pengujian pengiriman file didalamnya terdapat beberapa pengujian yaitu pengujian pengiriman file kode hex, pengujian menerima file kode hex, pengujian pengiriman string ACK, dan pengujian algoritma pengecekan.

5.2.1. Pengujian Pengiriman File Kode Hex

Pengujian pengiriman file kode hex dilakukan dengan modul XBee yang telah diintegrasikan dengan komputer *sender*. Mekanisme pengujian yaitu mengirim file kode hex dan menampilkan proses kerja dari pengiriman file. Mekanisme tersebut bertujuan agar mengetahui bagaimana kerja dari pengiriman file. Prosedur pengujian pengiriman file adalah sebagai berikut:

1. Modul XBee yang telah terkoneksi dengan komputer *sender* mengirim file berupa kode hex.
2. Pada kode program *sender* proses pengiriman ditampilkan dengan fungsi `print` untuk mengetahui proses pengiriman tiap 8 bit.
3. Kondisi ketika komputer *sender* mengirim file kode hex.

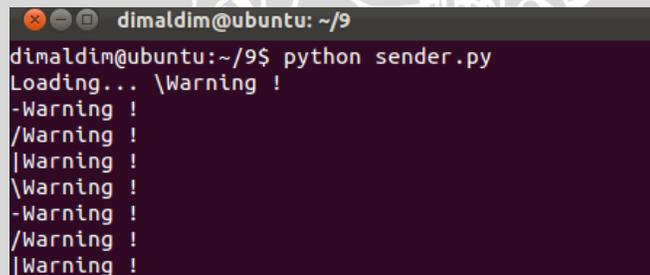
A terminal window titled 'dimaldim@ubuntu: ~/9' showing the execution of a Python script named 'sender.py'. The output consists of a series of 'putc' function calls. The first call is '\putc(" 1)', followed by several '\putc(" 1)' and '\putc(" 128)' calls, indicating the transmission of data to a receiver.

```
dimaldim@ubuntu: ~/9
dimaldim@ubuntu:~/9$ python sender.py
Loading... \putc( " 1 )
putc( " 1 )
putc( " 1 )
putc( " 128 )
putc( " 1 )
putc( " 128 )
putc( " 1 )
putc( " 128 )
putc( " 1 )
putc( " 128 )
putc( " 1 )
```

Gambar 5.4 Pengiriman File

Sumber : [Implementasi]

4. Pada kode program *sender* terdapat fungsi *waiting* jika node *receiver* tidak ditemukan, dan fungsi *waiting* akan melakukan perulangan sebanyak 8 kali.
5. Kondisi ketika komputer *sender* menunggu dan melakukan perulangan sebanyak 8 kali.

A terminal window titled 'dimaldim@ubuntu: ~/9' showing the execution of the same Python script 'sender.py'. The output consists of a series of 'Warning!' messages, indicating that the receiver was not found and the sender entered a waiting loop.

```
dimaldim@ubuntu:~/9$ python sender.py
Loading... \Warning !
-Warning !
/Warning !
|Warning !
\Warning !
-Warning !
/Warning !
|Warning !
```

Gambar 5.5 Komputer *Sender* Melakukan Perulangan

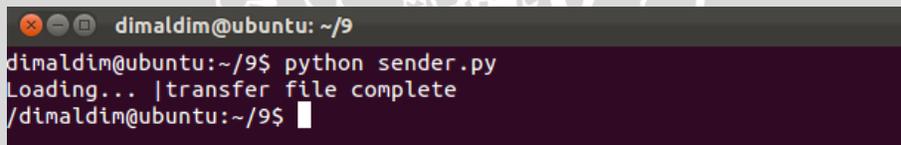
Sumber : [Pengujian]

6. Jika sampai perulangan ke 8 tidak ditemukan node *receiver*, maka pada sistem node *sender* akan *timeout*.
7. Kondisi ketika komputer *sender timeout*.

5.2.3. Pengujian Pengiriman String ACK

Pengujian proses pengiriman string ACK dilakukan dengan dua modul XBee yang telah terkoneksi dengan komputer *sender* dan *receiver*. Mekanisme pengujian yaitu komputer *sender* menerima string ACK yang dikirim oleh komputer *receiver* setelah pengiriman file berhasil. Prosedur pengujian pengiriman string ACK adalah sebagai berikut:

1. Setelah menerima file, pada komputer *receiver* akan mengirim sebuah string yang menandakan bahwa pengiriman file telah berhasil.
2. String yang dikirim oleh komputer *receiver* adalah “Transfer File Complete”.
3. Komputer *sender* akan menerima string yang dikirim oleh komputer *receiver*.
4. Pada proses menerima ACK terdapat proses algoritma pengecekan *Internet Checksum* yang berfungsi untuk mengecek apakah jumlah bit yang diterima sama dengan yang dikirim.
5. Kondisi dimana komputer *sender* berhasil menerima string ACK yang dikirim oleh komputer *receiver*.

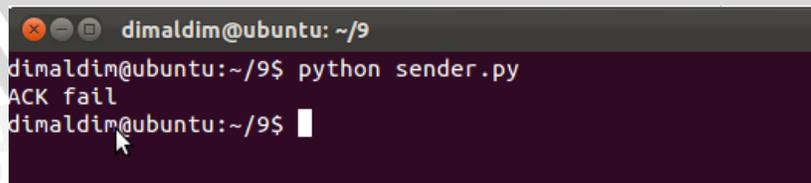


```
dimaldim@ubuntu: ~/9
dimaldim@ubuntu:~/9$ python sender.py
Loading... |transfer file complete
dimaldim@ubuntu:~/9$
```

Gambar 5.8 Komputer *Sender* Berhasil Menerima ACK

Sumber : [Pengujian]

6. Kondisi dimana komputer gagal menerima string ACK yang dikirim oleh komputer *receiver*.



```
dimaldim@ubuntu: ~/9
dimaldim@ubuntu:~/9$ python sender.py
ACK fail
dimaldim@ubuntu:~/9$
```

Gambar 5.9 Komputer *Sender* Gagal Menerima ACK

Sumber : [Pengujian]

5.2.4. Pengujian Algoritma Pengecekan (*Internet Checksum*)

Pengujian algoritma pengecekan dengan memanfaatkan algoritma dari *Internet Checksum*. Pengujian dilakukan untuk melihat apakah implementasi dari algoritma *Internet Checksum* telah bekerja dengan benar atau tidak. Mekanisme pengujian dilakukan dengan melakukan pembacaan data dan dilakukan perhitungan sesuai dengan cara kerja dari algoritma *Internet Checksum*. Prosedur dari pengujian algoritma pengecekan (*Internet Checksum*) adalah sebagai berikut :

1. Pada sistem pengujian telah dibuat nilai data berupa string “abcdefghijklmnopqrstuvwxyz”. Berikut kode pengujian untuk Algoritma pengecekan :

```

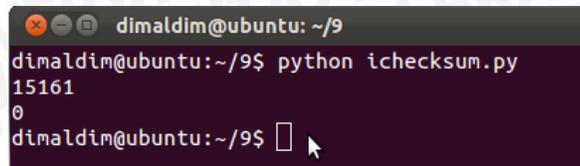
1. def ichecksum(data, sum=0):
2.     for i in range(0, len(data), 2):
3.         if i + 1 >= len(data):
4.             sum += ord(data[i]) & 0xFF
5.         else:
6.             w = ((ord(data[i]) << 8) & 0xFF00) +
7. (ord(data[i+1]) & 0xFF)
8.             sum += w
9.             while (sum >> 16) > 0:
10.                sum = (sum & 0xFFFF) + (sum >> 16)
11.            sum = ~sum
12.            return sum & 0xFFFF
13.
14. check = ichecksum('abcdefghijklmnopqrstuvwxyz')
15. print check
16. print ichecksum('abcdefghijklmnopqrstuvwxyz', check)

```

Gambar 5.10 Kode Program Internet Checksum

Sumber : [Pengujian]

2. Pada algoritma *Internet Checksum*, data 8 bit pertama ditambahkan dengan 8 bit kedua. Lalu hasil dari penjumlahan tersebut dikomplement dan ditambahkan dengan hasil awal dan hasilnya adalah 0.
3. Berikut kondisi awal setelah di tambahkan dengan 8 bit dan hitung dengan algoritma *Internet Checksum* sehingga hasilnya menjadi 0.



```
dimaldim@ubuntu: ~/9
dimaldim@ubuntu:~/9$ python ichecksum.py
15161
0
dimaldim@ubuntu:~/9$
```

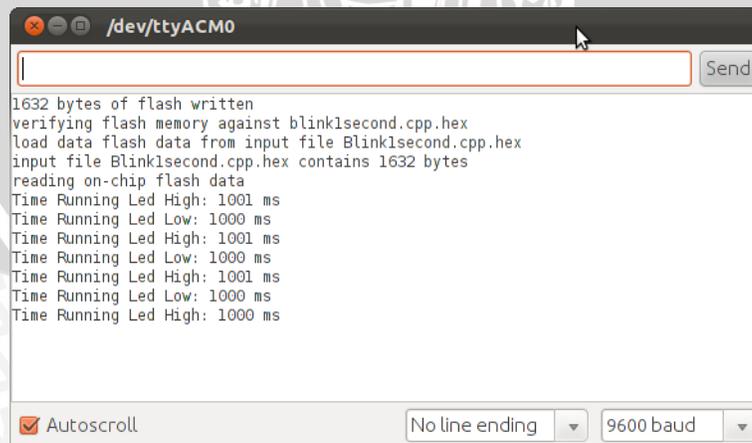
Gambar 5.11 Pengujian Algoritma Internet Checksum

Sumber : [Pengujian]

5.3. Pengujian Unggah File Kedalam Flash Memori

Pengujian unggah file dilakukan untuk mengetahui apakah file kode hex dapat di-*upload* kedalam flash memori mikrokontroler. Proses unggah file memanfaatkan perangkat lunak AVRDUDE berbasis *command-line/console*. Kondisi awal pada mikrokontroler arduino uno adalah terdapat program menyalakan lampu LED setiap 1 detik sekali. Selanjutnya, akan diubah fungsi dari mikrokontroler arduino uno dengan mengirim program berupa file kode hex yang berisi kode untuk menyalakan led setiap 8 detik sekali sebagai notifikasi bahwa file kode hex berhasil diunggah kedalam flash memori. Prosedur dari pengujian unggah file kedalam flash memori dari sebuah mikrokontroler adalah sebagai berikut:

1. Mikrokomputer Raspberry Pi memanggil program AVRDUDE sesuai dengan BAB IV implementasi.
2. Kondisi awal modul arduino yang berisi file kode hexa yang bernama `Blink1second.cpp.hex` untuk menghidupkan lampu setiap 1 detik sekali.



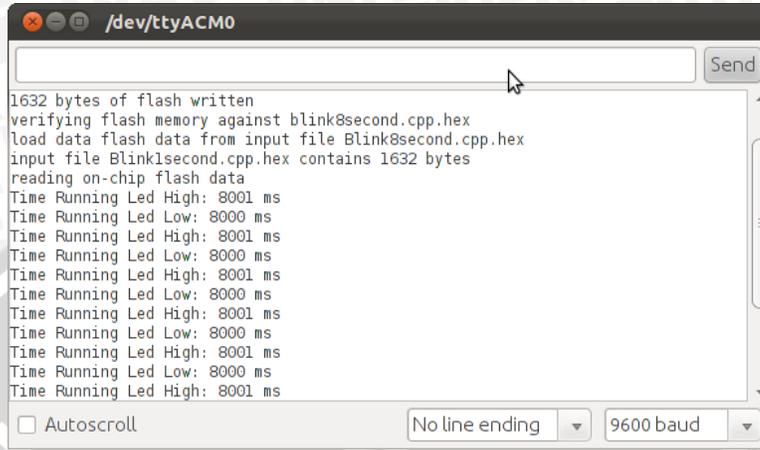
```
/dev/ttyACM0
1632 bytes of flash written
verifying flash memory against blink1second.cpp.hex
load data flash data from input file Blink1second.cpp.hex
input file Blink1second.cpp.hex contains 1632 bytes
reading on-chip flash data
Time Running Led High: 1001 ms
Time Running Led Low: 1000 ms
Time Running Led High: 1001 ms
Time Running Led Low: 1000 ms
Time Running Led High: 1001 ms
Time Running Led Low: 1000 ms
Time Running Led High: 1000 ms
Time Running Led High: 1000 ms

Autoscroll No line ending 9600 baud
```

Gambar 5.12 Kondisi Awal Modul Arduino

Sumber : [Pengujian]

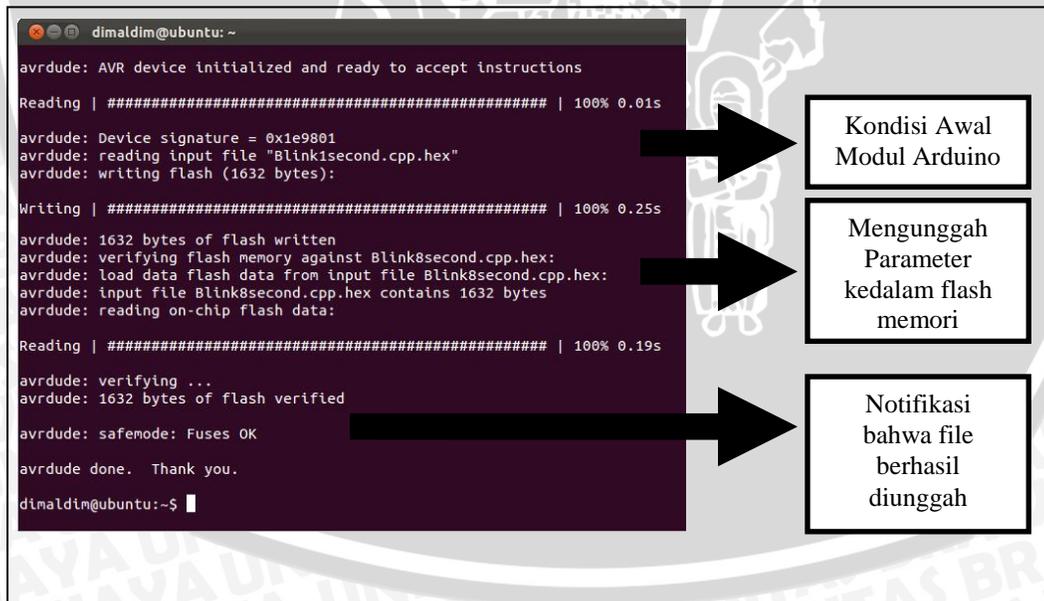
- 3. Kondisi ketika modul Arduino berhasil unggah file kode hexa yang bernama Blink8second.cpp.hex untuk menyalakan lampu LED selama 8 detik.



Gambar 5.13 Kondisi Modul Arduino Setelah Unggah File

Sumber : [Perancangan]

- 4. Proses berhasil unggah file kode hexa yang bernama Blink.8second.cpp.hex untuk menghhidupkan lampu setiap 8 detik sekali kedalam kedalam flash memori pada modul Arduino.



Gambar 5.14 Proses Unggah File Kedalam Flash Memori

Sumber : [Pengujian]

5. Kondisi pada mikrokontroler Arduino Mega, lampu LED menyala sesuai instruksi pada file kode hex yaitu menyala setiap 8 detik sekali.



Gambar 5.15 Mikrokontroler Arduino Atmega

Sumber : [Pengujian]

5.4. Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem dimaksudkan untuk menguji kinerja dari keseluruhan sistem mulai dari komunikasi serial, pengiriman file, pengiriman string ACK, algoritma pengecekan, dan unggah file kedalam flash memori dari mikrokontroler. Pengujian keseluruhan sistem menggunakan dua skenario pengujian. Skenario pertama diuji keseluruhan sistem dengan jarak dan tanpa hambatan. Skenario kedua diuji keseluruhan sistem dengan jarak dan dengan hambatan.

5.3.1. Pengujian Skenario 1

Pada pengujian skenario 1, diuji dengan keseluruhan sistem dari proses komunikasi serial, pengiriman file, algoritma pengecekan, pengiriman ACK hingga proses mengunggah file kedalam flash memori dari mikrokontroler. Pada pengujian ini dilakukan pengujian sistem dengan jarak antara 1 sampai 100 meter dan tanpa hambatan. Berikut hasil yang didapat dari pengujian skenario 1. Pengujian dilakukan pada lorong gedung PTIIK Universitas Brawijaya.

Tabel 5.1 Hasil Skenario 1

Sumber : [Pengujian]

Jarak (m)	File			ACK			Mengunggah pada flash memory
	yang dikirim	yang diterima	% losses	yang dikirim	yang diterima	% losses	
1	File hex	File hex	0	String	String	0	Berhasil
10	File hex	File hex	0	String	String	0	Berhasil
20	File hex	File hex	0	String	String	0	Berhasil
30	File hex	File hex	0	String	String	0	Berhasil
40	File hex	File hex	0	String	String	0	Berhasil
50	File hex	File hex	0	String	String	0	Berhasil
60	File hex	File hex	0	String	String	0	Berhasil
70	File hex	File hex	0	String	String	0	Berhasil
80	File hex	File hex	0	String	String	0	Berhasil
90	File hex	File hex	0	String	String	0	Berhasil
100	File hex	File hex	0	String	String	0	Berhasil

Pengujian pada skenario 1 dari jarak antara 1 meter hingga 100 meter berhasil mengirim file kode hex tanpa *losses* dan berhasil diunggah kedalam flash memori dari sebuah mikrokontroler.

5.3.2. Pengujian Skenario 2

Pada pengujian skenario 2, diuji dengan keseluruhan sistem dari proses komunikasi serial, pengiriman file, algoritma pengecekan, pengiriman ACK hingga proses mengunggah file kedalam flash memori dari mikrokontroler. Pada pengujian ini dilakukan pengujian sistem dengan jarak antara 1 sampai 100 meter dan dengan hambatan. Pengujian dilakukan pada gedung PTIIK Universitas Brawijaya.

Hambatan yang dimaksud berupa hambatan dinding ruangan yang terbuat dari tembok beton. Pengujian pada skenario ini, node *sender* diletakkan pada

gedung B PTIIK tepatnya pada laboratorium Jaringan Komputer. Sedangkan node *receiver* diletakkan secara bertahap pada gedung B PTIIK diluar ruangan laboratorium sampai gedung D PTIIK. Berikut hasil yang didapat dari pengujian skenario 2.

Tabel 5.2 Hasil Pengujian Skenario 2

Sumber : [Pengujian]

Jarak (m)	File			ACK			Mengunggah pada flash memory
	yang dikirim	yang diterima	% losses	yang dikirim	yang diterima	% losses	
1	File hex	File hex	0	String	String	0	Berhasil
10	File hex	File hex	0	String	String	0	Berhasil
20	File hex	File hex	0	String	String	0	Berhasil
30	File hex	File hex	0	String	String	0	Berhasil
40	File hex	File hex	0	String	String	0	Berhasil
50	File hex	File hex	0	String	String	0	Berhasil
60	File hex	File hex	0	String	String	0	Berhasil
70	File hex	File hex	100%	String	String	100%	Gagal
80	File hex	File hex	100%	String	String	100%	Gagal
90	File hex	File hex	100%	String	String	100%	Gagal
100	File hex	File hex	100%	String	String	100%	Gagal

Pengujian pada skenario 2 dari jarak antara 1 meter hingga 100 meter dengan hambatan hanya berhasil dari jarak 1 meter hingga 60 meter mengirim file kode hex tanpa losses dan berhasil diunggah kedalam flash memori dari sebuah mikrokontroler. Pada jarak 70 meter hingga 100 meter, node *sender* tidak dapat menemukan sinyal dari node *receiver*, sehingga proses pengiriman ACK dan proses mengunggah file kedalam flash memori gagal. Pada pengujian ini didapat juga kecepatan dari pengiriman file lebih lambat karena adanya hambatan.

BAB VI PENUTUP

6.1. Kesimpulan

Berdasarkan hasil perancangan, implementasi dan pengujian yang dilakukan, maka diambil kesimpulan sebagai berikut :

1. Dalam pembuatan sebuah prototipe sistem digunakan komunikasi serial antara modul XBee dengan komputer, dan modul XBee difungsikan sebagai modem dalam komunikasi datanya.
2. Dalam pengiriman parameter menggunakan modul XBee jarak antara 1 meter hingga 100 meter diluar ruangan/tanpa hambatan dapat berhasil. Namun pada pengiriman parameter dari jarak antara 70 meter hingga 100 meter antar ruangan satu dengan yang lain atau dengan hambatan yang berupa tembok beton tidak dapat berhasil.
3. Algoritma pengecekan *Internet Checksum* dapat diandalkan, karena pada protokol *end to end* cara kerjanya adalah dengan menjumlahkan semua deretan bilangan integer yang dikirim. Namun memiliki kelemahan deteksi *kesalahan* yang tidak dapat mendeteksi unit data (*words*) yang urutannya acak dan tidak dapat mendeteksi unit data yang mengalami kesalahan.
4. Dalam proses mengubah parameter kedalam flash memori pada modul arduino, dapat berhasil jika file kode hexa yang diterima oleh mikrokomputer berhasil 100%, sehingga parameter pada modul arduino dapat diubah.

6.2. Saran

Saran yang dapat diberikan untuk pengembangan perangkat lunak ini antara lain :

1. Dilakukan pengiriman file dengan menggunakan node yang lebih dari dua dengan membangun sebuah topologi jaringan.
2. Pengujian dilakukan dengan menggunakan metode routing.

3. Algoritma pengecekan yang digunakan Algoritma CRC32 atau *Parity Check*.



DAFTAR PUSTAKA

- [ABR-11] Abriansyah, Okki Dwi Bagus. 2011. "Implementasi Wireless Sensor Network (WSN) Untuk Monitoring Beberapa Parameter Penting yang Berhubungan dengan Permasalahan Daya Serta Kebutuhan Pemeliharaan", Jurnal : Institut Teknologi Sepuluh Nopember
- [ALL-07] Alliance, ZigBee. 2007. "*ZigBee and Wireless Radio Frequency Coexistence*", White paper.
- [BIS-12] Bisyr, Khamdan Amin. 2012. "Rancang Bangun Komunikasi Data Wireless Mikrokontroler Menggunakan Modul Xbee (IEEE 802.15.4)", Institut Pertanian Bogor.
- [DEV-12] Deviyanti, Diah. 2012. "Perancangan Aplikasi Error Detection Menggunakan Metode Checksum Dengan Bahasa Pemrograman C++", STMIK Pontianak.
- [DWI-11] Dwi, Nofianti. 2011. "Simulasi Kinerja WPAN 802.15.4 (zigbee) dengan algoritma routing". (Online), (Diakses dari <http://eprints.undip.ac.id/32020/1/>, Tanggal Akses: 24-02-2013).
- [FAT-13] Fathurrohman. 2013. "Implementasi GPS Pada Mobile Robot Untuk Memantau Lokasi Kebocoran Gas". Jurnal : Institute Teknologi Surabaya.
- [HAK-31] Hakim, Malik Abdillah Ibnul. 2013. "Pemanfaatan mini pc raspberry PI sebagai pengontrol lampu jarak jauh berbasis web pada rumah". Jurnal: Perpustakaan Universitas Komputer Bandung.
- [HAR-11] Hardiyanto, Gilang. 2011. "Definisi Komputer Dengan Gambar. (Online) <http://gilanghardiyanto.blogspot.com/>. Diakses : 20 Nopember 2013.
- [HEB-11] Hebel, Martin. 2011. "*Getting Started with XBee RF Modules: A Tutorial for BASIC Stamp and Propeller Microcontrollers*", Jurnal : Parallax, Inc.
- [JOH-08] John Bellantoni, Liaison. 2008. "*ZigBee-Enabled RFID Reader*

- Network*". WJ Communications, Inc. Worcester Polytechnic Institute.
- [KIN-09] Kinney, Patric. 2009. "*Zigbee Technology: Wireless Control that Simply Works*". Chair of IEE 802.15.4 Task Group. Communications Design Conference.
- [LES-11] Lestari, Jati. 2011. "*Webcam Monitoring Ruangan Menggunakan Sensor Gerak Pir (Passive Infra Red)*". Jakarta : Universitas Budi Luhur.
- [LON-08] Lonn Johan, dan Jonas Olsson. 2008. "*Zigbee for wireless networking*". Jurnal : Department of Science and Technology.
- [NIS-13] Nishanthini. 2013. "*Wireless E-Organize And Seek Physical Items Using Raspberry Pi*". Jurnal : International Journal of Advanced Engineering and Global Technology.
- [RAM-12] Ramadhan, Radix Satrio. 2012. "*Perancangan Wireless Network Control System Pada Kendali Kecepatan Motor Dengan Metode Pid*". Bandung : Telkom Institute of Technology.
- [RIA-09] Riaz, Ahamed. 2009. "*The Role Of Zigbee Technology In Future Data Communication System*", Sathak Institute of Technology.
- [STO-10] Stone, Jonathan, Michael Greenwald, Craig Partridge, and Jim Hughes. 2009. "*Performance of Checksums and CRCs over Real Data*". California : Stanford University.
- [UPT-12] Upton, Eben. 2012. "*Programming the Raspberry Pi*". Jurnal : Raspberry Pi Foundation.
- [WIN-12] Winardi. 2012. "*Mengenal Teknologi ZigBee Sebagai Standar Pengiriman Data Secara Wireless*". (Online), (Diakses dari <http://comp-eng.binus.ac.id/files/2012/06/>, Tanggal Akses: 21-02-2013).
- [ZAH-13] Zahira, Nursyamsiah. 2013 "*Penyambungan Internet*". (online). (Diakses dari <http://www.scribd.com/>, Tanggal Akses: 19/12/2013).

LAMPIRAN

Lampiran 1. Kode Program Komputer Sender

```
1. import serial
2. import select
3. import re
4. import threading
5. import sys
6. import time
7. from xmodem import XMODEM
8. from time import sleep
9. s = serial.Serial(port='/dev/ttyUSB0', baudrate=115200,
10. bytesize=8, parity='N', stopbits=1, timeout=None,
11. xonxoff=False, rtscts=100)
12. try:
13.     s.open()
14. except Exception, e:
15.     print "error open serial port: " + str(e)
16. exit()
17. if s.isOpen():
18.     try:
19.         s.flush()
20.         s.write(" ")
21.         def getc(size, timeout=3):
22.             w,t,f = select.select([s], [], [], timeout)
23.             if w:
24.                 data = s.read(size)
25.             else:
26.                 data = None
27.             return data
28.         def putc(data, timeout=3):
29.             w,t,f = select.select([], [s], [], timeout)
30.             if t:
31.                 s.write(data)
32.                 s.flush()
33.                 size = len(data)
34.                 data = re.search('(?!<=\\xff)', '\\xff')
35.                 sleep(0.001)
```

```
36.         else:
37.             size = None
38.         return size
39.     class progress_bar_loading(threading.Thread):
40.     def run(self):
41.         global stop
42.         global kill
43.         print 'Loading.... ',
44.         sys.stdout.flush()
45.         i = 0
46.         while stop != True:
47.             if (i%4) == 0:
48.                 sys.stdout.write('\b/')
49.             elif (i%4) == 1:
50.                 sys.stdout.write('\b-')
51.             elif (i%4) == 2:
52.                 sys.stdout.write('\b\\')
53.             elif (i%4) == 3:
54.                 sys.stdout.write('\b|')
55.             sys.stdout.flush()
56.             time.sleep(0.2)
57.             i+=1
58.             if kill == True:
59.                 print '\b\b\b\b ABORT!',
60.                 kill = False
61.                 stop = False
62.         p = progress_bar_loading()
63.         p.start()
64.         try:
65.             modem = XMODEM(getc, putc)
66.             stream = file('Blink.cpp.hex', 'rb')
67.             modem.send(stream,retry=8,quiet=0,timeout=40)
68.             stream.close()
69.         def ichecksum(data, sum = 0):
70.             for i in range(0,len(data),2):
71.                 if i + 1 >= len(data):
72.                     sum += ord(data[i]) & 0xFF
73.                 else:
```

```
74.         w = ((ord(data[i]) << 8) & 0xFF00) +
75.             (ord(data[i+1]) & 0xFF)
76.         sum += w
77.         while (sum >> 16) > 0:
78.             sum = (sum & 0xFFFF) + (sum >> 16)
79.             sum = ~sum
80.         return sum & 0xFFFF
81.         s.flushInput()
82.         data = s.read(22)
83.         if data:
84.             check = ichecksum(data)
85.             csum = ichecksum(data, check)
86.             if csum == 0:
87.                 print data
88.             else:
89.                 print 'ACK fail'
90.                 time.sleep(1)
91.                 stop = True
92.                 s.close()
93.         except KeyboardInterrupt or EOFError:
94.             kill = True
95.             stop = True
96.         except Exception, el:
97.             print "error..." + str(el)
98.     else:
99.         print "cannot open serial port"
```

Lampiran 2. Kode Program Komputer Receiver

```
1. # -*- coding: UTF-8 -*-
2. import select
3. import select
4. import re
5. from xmodem import XMODEM
6. from time
7. import sleep
8. s = serial.Serial(port='/dev/ttyUSB0', baudrate=115200,
9. bytesize=8, parity='N', stopbits=1, timeout=None,
10. xonxoff=False, rtscts=100)
11. try:
12.     s.open()
13. except Exception, e:
14.     print "error open serial port: " + str(e)
15. exit()
16. if s.isOpen():
17.     try:
18.         s.flush()
19.         data = s.read(1)
20.         def getc(size, timeout=3):
21.             w,t,f = select.select([s], [], [], timeout)
22.             if w:
23.                 data = s.read(size)
24.             else:
25.                 data = None
26.             return data
27.         def putc(data, timeout=3):
28.             w,t,f = select.select([], [s], [], timeout)
29.             if t:
30.                 pos = len(data)
31.                 if (pos & 1):
32.                     pos -= 1
33.                     sum = ord(data[pos])
34.                 else:
35.                     sum = 0
36.                 while pos > 0:
```

```
37.         pos -= 2
38.         sum += (ord(data[pos + 1]) << 8) + ord(data[pos])
39.         sum = (sum >> 16) + (sum & 0xffff)
40.         sum += (sum >>16)
41.         result = (~ sum) & 0xffff
42.         result = result >> 8 | (( result & 0xff) << 8)
43.         return result
44.         s.write(data)
45.         s.flush()
46.         size = len(data)
47.     else:
48.         size = None
49.     return size
50.     modem = XMODEM(getc, putc)
51.     sleep(2)
52.     stream = file('Blink.cpp.hex', 'wb')
53.     modem.recv(stream, retry = 8, delay = 1)
54.     stream.close()
55.     txt_file = open('Blink.cpp.hex')
56.     source_content = txt_file.read()
57.     cleaned = source_content.replace("\xff", "")
58.     cleaned
59.     target = open('Blink.cpp.hex', 'w') ()
60.     target.write(cleaned)
61.     target.close()
62.     s.flush()
63.     s.write ("transfer file complete")
64.     s.close()
65. except Exception, el:
66.     print "error..." + str(el)
67. else:
68.     print "cannot open serial port"
```