

## PENGANTAR

Segala pujian hanya milik Allah, Rabb semesta alam. Saya bersaksi tiada Illah yang berhak diibadahi kecuali hanya Allah semata, tidak ada sekutu baginya dan saya bersaksi bahwa Muhammad adalah hamba dan utusan-Nya. Hanya karena pertolongan Allah semata-mata penulis mampu menyelesaikan penyusunan skripsi ini. Yaa Rabb, Engkaulah Dzat yang membolak-balikkan hati manusia, tetapkanlah hati hamba-Mu ini di atas agama-Mu, Amien. Semoga shalawat dan salam senantiasa tercurahkan kepada Nabi Muhammad SAW, keluarga beliau, para sahabat dan pengikut beliau sampai hari kiamat nanti.

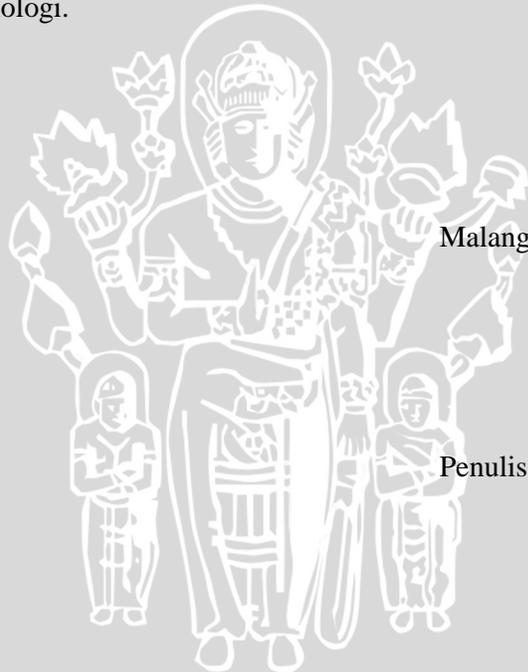
Skripsi berjudul “Sistem *Downloader* Data Memori Mikrokontroler Atmel-AVR Melalui Radio frekuensi: Sisi Target” ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Universitas Brawijaya.

Penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, dengan ketulusan dan kerendahan hati penulis menyampaikan terima kasih kepada:

- Ibu dan Bapak atas segala kasih sayang, perhatian serta kesabarannya dalam mendidik dan membesarkan penulis, yang telah mendoakan, membimbing dan mendukung penulis hingga terselesaikannya skripsi ini,
- Rudy Yuwono, ST., MSc selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya,
- Muhammad Aziz Muslim, ST., MT., PhD. selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya,
- Ir. M. Julius St, MS selaku Ketua Kelompok Dosen Keahlian Elektronika Jurusan Teknik Elektro Universitas Brawijaya,
- Ir. Nanang Sulistiyanto selaku Dosen Pembimbing I atas segala bimbingan, pengarahan, gagasan, ide, saran serta motivasi yang telah diberikan,

- Waru Djuriatno, ST., MT selaku Dosen Pembimbing II atas segala bimbingan, saran dan masukan yang telah diberikan.
- Anang Fakhruudin selaku partner penelitian yang selalu menemani dalam pengerjaan penelitian dan penyelesaian alat maupun dalam penyusunan laporan.
- Segala pihak yang memberikan support untuk pengerjaan penelitian baik itu secara langsung ataupun tidak langsung yang tidak dapat disebut satu persatu.

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



Malang,

Penulis

## DAFTAR ISI

PENGANTAR.....	i
DAFTAR ISI.....	iii
DAFTAR GAMBAR.....	vi
DAFTAR TABEL.....	viii
ABSTRAK.....	ix
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Ruang Lingkup.....	3
1.4 Tujuan.....	3
1.5 Sistematika Penulisan.....	3
TINJAUAN PUSTAKA.....	5
2.1 Dokumen Heksadesimal Intel ( <i>Intel Hex File</i> ) 8 bit.....	5
2.2 Transmisi Data.....	7
2.3 Unit RF (Radio Frekuensi).....	8
2.3.1 Fitur Utama Produk.....	8
2.3.2 Spesifikasi.....	9
2.4 Protokol Komunikasi.....	10
2.4.1 Komunikasi Serial.....	11
2.4.2 MAX 232.....	14
2.5 Metode Deteksi Kesalahan pada Transmisi Data menggunakan <i>Checksum</i> .....	15
2.6 <i>Compiler</i> Delphi.....	16
METODE PENELITIAN.....	18
3.1 Studi Literatur.....	18
3.2 Perancangan dan Pembuatan Alat.....	18
3.3 Pengujian Alat.....	18
3.4 Pengambilan Kesimpulan dan Pemberian Saran.....	19
PERANCANGAN DAN PEMBUATAN ALAT.....	20
4.1 Spesifikasi Sistem.....	20

4.2 Perancangan Sistem.....	21
4.2.1 Perancangan Sistem Secara Keseluruhan.....	21
4.2.2 Perancangan Sisi Pemrograman.....	23
4.3 Perancangan Perangkat Keras.....	25
4.3.1 Rangkaian Penerima dan Pemancar RF.....	25
4.3.2 Rangkaian Antarmuka MAX232.....	26
4.4 Perancangan Komunikasi Data.....	27
4.4.1 Perancangan Protokol Komunikasi Data Radio Frekuensi.....	27
4.5 Perancangan dan Pembuatan Perangkat Lunak.....	35
<b>PENGUJIAN DAN ANALISIS.....</b>	<b>41</b>
5.1 Pengujian Pemancar dan Penerima RF.....	41
5.1.1 Tujuan.....	41
5.1.2 Peralatan Pengujian.....	41
5.1.3 Proses Pengujian.....	42
5.1.4 Hasil dan Analisis.....	43
5.2 Pengujian Rangkaian MAX232.....	44
5.2.1 Tujuan.....	44
5.2.2 Peralatan Pengujian.....	44
5.2.3 Proses Pengujian.....	44
5.2.4 Hasil dan Analisis.....	45
5.3 Pengujian Perangkat Lunak.....	45
5.3.1 Tujuan.....	45
5.3.2 Peralatan Pengujian.....	45
5.3.3 Proses Pengujian.....	45
5.3.4 Hasil dan Analisis.....	46
5.4 Pengujian Keseluruhan.....	46
5.4.1 Pengujian Pengiriman Instruksi.....	47
5.4.2 Pengujian Jarak Jangkauan Maksimal Transmisi Data.....	51
5.4.3 Pengujian Kecepatan Operasi Memori.....	53
5.4.4 Pengujian Penerimaan Data.....	54
5.4.5 Pengujian Perangkat Lunak.....	55
<b>KESIMPULAN DAN SARAN.....</b>	<b>59</b>

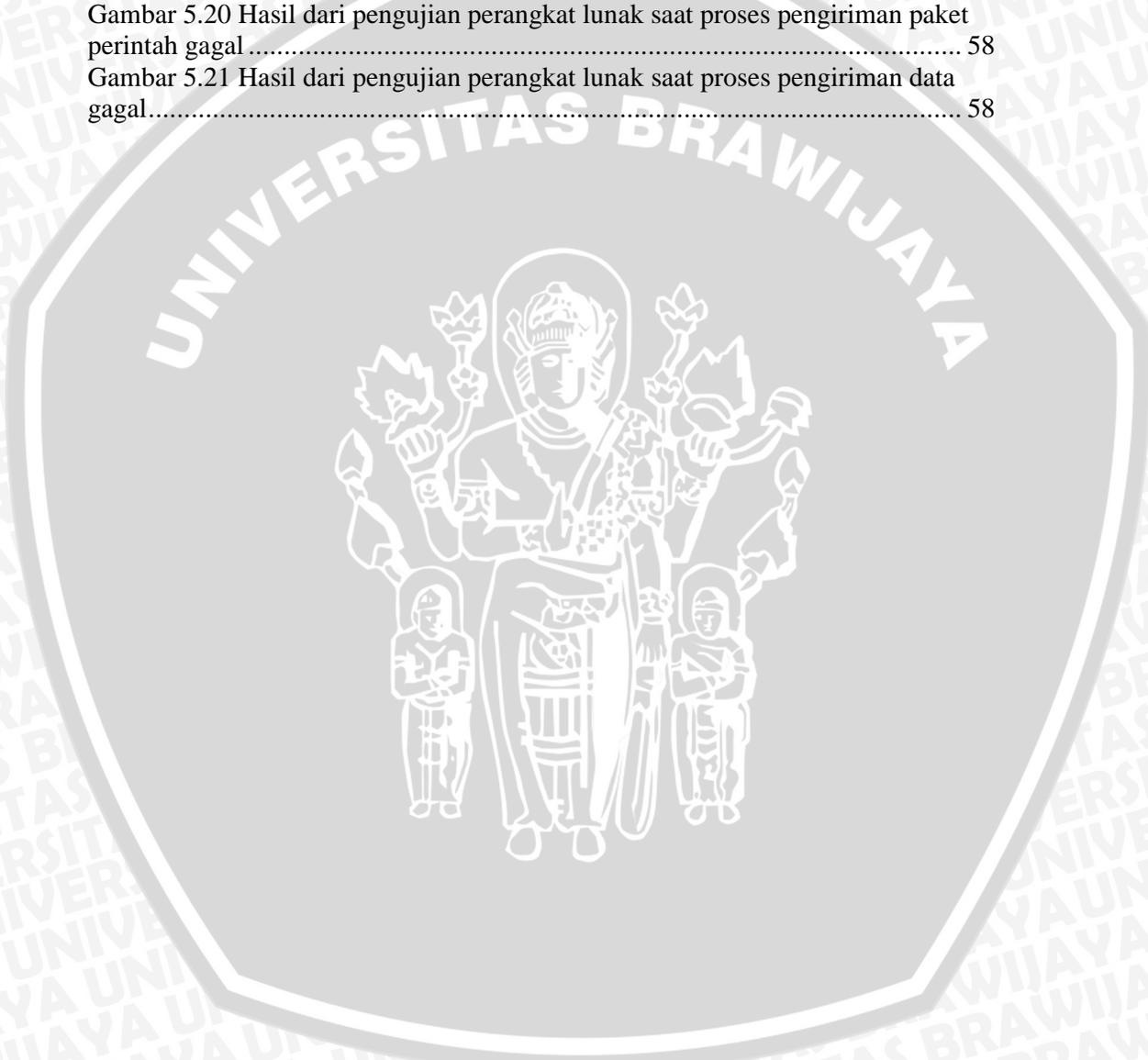
6.1 Kesimpulan.....	59
6.2 Saran.....	59
DAFTAR PUSTAKA.....	61



## DAFTAR GAMBAR

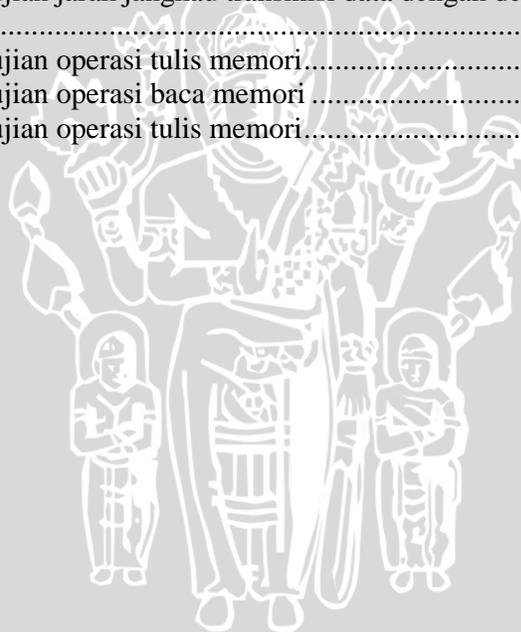
Gambar 2.1 Bagian-bagian yang terdapat dalam satu baris pada dokumen hex Intel.....	6
Gambar 2.2 Ys-1020 (Radio Frekuensi).....	8
Gambar 2.3 Konfigurasi Pin YS-1020.....	10
Gambar 2.4 Format data logika TTL/CMOS.....	13
Gambar 2.5 Format data pada konverter level tegangan RS232.....	13
Gambar 2.6 IC MAX232 dan Rangkaian minimum sistemnya.....	14
Gambar 2.7 IDE Delphi.....	17
Gambar 4.1 Sistem secara keseluruhan.....	21
Gambar 4.2 Aliran data di sisi pemrogram.....	22
Gambar 4.3 Aliran data di sisi target.....	23
Gambar 4.4 Blok diagram sisi pemrogram.....	23
Gambar 4.5 Diagram blok dari keseluruhan perangkat keras.....	25
Gambar 4.6 Rangkaian pemancar dan penerima RF.....	26
Gambar 4.7 Rangkaian Penyesuai Level Tegangan (MAX232).....	27
Gambar 4.8 Format frame header paket transmisi.....	28
Gambar 4.9 Skenario protokol sesi pilih mikrokontroler target.....	32
Gambar 4.10 Skenario protokol sesi tulis memori target.....	33
Gambar 4.11 Skenario protokol sesi baca memori target.....	34
Gambar 4.12 Diagram alir utama aplikasi perangkat lunak PC sisi pemrogram..	35
Gambar 4.13 Diagram alir sisi pemrogram pilih mikrokontroler.....	36
Gambar 4.14 Diagram alir sisi pemrogram transmisi sesi tulis <i>flash</i> /EEPROM target.....	37
Gambar 4.15 Diagram alir sisi pemrogram transmisi sesi baca <i>flash</i> /EEPROM target.....	38
Gambar 4.16 Diagram alir sisi pemrogram cek koneksi.....	39
Gambar 4.17 Diagram alir sisi pemrogram hapus mikrokontroler.....	39
Gambar 4.18 Diagram alir sisi pemrogram <i>load file</i> .....	40
Gambar 5.1 Diagram blok pengujian pengaturan <i>baudrate</i> dan <i>channel</i> modul RF.....	42
Gambar 5.2 Tampilan program untuk pengaturan baudrate dan setting modul RF.....	42
Gambar 5.3 Diagram blok pengujian software penerimaan data secara serial.....	43
Gambar 5.4 Hasil dari pengujian software penerimaan dan pengiriman data secara serial.....	44
Gambar 5.5 Diagram blok pengujian MAX232.....	45
Gambar 5.6 Diagram blok pengujian perangkat lunak pisah paket data.....	46
Gambar 5.7 Hasil dari pengujian perangkat lunak pisah paket data dan pengecekan <i>error</i> .....	46
Gambar 5.8 Blok pengujian sistem secara keseluruhan.....	47
Gambar 5.9 Hasil pengujian operasi pilih mikrokontroler.....	48
Gambar 5.10 Hasil pengujian operasi baca memori EEPROM.....	48
Gambar 5.11 Hasil pengujian operasi hapus memori.....	49
Gambar 5.12 Hasil pengujian operasi tulis memori <i>flash</i> .....	50
Gambar 5.13 Hasil pengujian operasi baca memori EEPROM.....	51

Gambar 5.14 Blok pengujian jarak jangkauan transmisi ..... 52  
Gambar 5.15 Hasil dari pengujian perangkat lunak saat *load file hex* ..... 55  
Gambar 5.16 Hasil dari pengujian perangkat lunak saat memilih mikrokontroler  
dari sisi target ..... 56  
Gambar 5.17 Hasil dari pengujian perangkat lunak saat pemilihan mikrokontroler  
dari sisi target gagal..... 56  
Gambar 5.18 Hasil dari pengujian perangkat lunak saat sisi target tidak aktif..... 57  
Gambar 5.19 Hasil dari pengujian perangkat lunak saat proses penulisan *flash*  
sukses ..... 57  
Gambar 5.20 Hasil dari pengujian perangkat lunak saat proses pengiriman paket  
perintah gagal ..... 58  
Gambar 5.21 Hasil dari pengujian perangkat lunak saat proses pengiriman data  
gagal..... 58



## DAFTAR TABEL

Tabel 2.1 Konfigurasi YS-1020 .....	10
Tabel 2.2 Fungsi pin pada DB25 dan DB9 .....	12
Tabel 4.1 Format <i>frame</i> perintah pilih mikrokontroler.....	29
Tabel 4.2 ID mikrokontroler dari sisi pemrogram .....	29
Tabel 4.3 Format <i>frame</i> paket perintah dari sisi pemrogram saat sesi tulis target	29
Tabel 4.4 Format <i>frame</i> paket perintah dari sisi pemrogram saat sesi baca target	30
Tabel 4.5 Format <i>frame</i> paket perintah dari sisi target saat sesi baca target .....	30
Tabel 4.6 Format <i>frame</i> paket data dari sisi pemrogram .....	30
Tabel 4.7 Format <i>frame</i> paket data dari sisi target .....	30
Tabel 4.8 Format <i>frame</i> paket konfirmasi.....	31
Tabel 4.9 ID Perintah sesuai jenis operasi memori mikrokontroler target .....	31
Tabel 4.10 Informasi ukuran data dan id <i>device</i> .....	31
Tabel 5.1 Hasil pengujian pengaturan <i>baudrate</i> dan <i>setting</i> RF YS-1020 .....	43
Tabel 5.2 Hasil pengujian rangkaian MAX232 .....	45
Tabel 5.3 Hasil pengujian jarak jangkauan transmisi data dengan dengan memori sebesar 288 <i>bytes</i> .....	52
Tabel 5.4 Hasil pengujian operasi tulis memori.....	53
Tabel 5.5 Hasil pengujian operasi baca memori .....	54
Tabel 5.6 Hasil pengujian operasi tulis memori.....	54



## ABSTRAK

**Kun Ubaidan Fahmi, Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, November 2010, Sistem *Downloader* Data Memori Mikrokontroler Atmel-AVR Melalui *Radio frequency* Menggunakan Sistem Multi target : Sisi Pemrogram, Dosen Pembimbing: Ir. Nanang Sulistiyanto dan Waru Djuriatno, ST., MT.**

Penggunaan mikrokontroler keluaran Atmel-AVR mencakup berbagai aspek dalam perancangan sistem elektronika, hal ini dikarenakan penggunaannya yang praktis, ekonomis dan cukup handal. Penggunaan Atmel-AVR oleh tim robot dari Teknik Elektro Universitas Brawijaya (TEUB) dalam ajang utuk berkompetisi dalam KRI (Kontes Robot Indonesia) ataupun KRCI (Kontes Robot Cerdas Indonesia) menggunakan banyak mikrokontroler untuk tiap proses kerja robot, mulai dari sebagai pengolah hasil sensor dan transduser, pengolah logika pemrogram, pengatur pergerakan dan posisi robot.

Dalam mencari performa robot secara maksimal dibutuhkan adanya proses *trial* dan *error* berulang-ulang. Dalam pengujiannya dibutuhkan pemindahan robot dari lapangan uji ke komputer untuk proses pemrograman memori mikrokontroler, hal ini menyita banyak waktu dan tenaga karena bobot rata-rata robot sekitar sepuluh kilogram. Penggunaan media kabel panjang sebagai solusi terhadap permasalahan ini dianggap tidak praktis.

Alternatif solusi dari permasalahan tersebut adalah menggunakan *downloader* dengan media nirkabel sehingga dirancanglah “Sistem *Downloader* Data Memori Mikrokontroler Atmel-AVR Melalui Radio frekuensi Menggunakan Sistem Multi Target” yang terdiri atas sisi pemrogram dan sisi target. Sistem yang terdapat pada sisi target bertugas untuk menangani pemrograman memori mikrokontroler yang berbasis atmel-AVR. Komunikasi sisi target dan sisi pemrogram menggunakan protokol yang dibangun diatas protokol data serial USART dengan menggunakan media transmisi Radio frekuensi.

Operasi pemrograman memori atmel-AVR meliputi hapus memori, baca tulis memori *flash*, baca tulis memori EEPROM, baca tulis fuse bit serta operasi baca *device signature* mikrokontroler target. Ukuran memori maksimal yang dapat ditangani oleh sistem ini adalah 32 Kbyte. Sisi target juga dapat berkomunikasi dengan baik secara *half-duplex* dengan sisi pemrogram. Baud rate transmisi data melalui radio frekuensi yaitu 9600 bps.

**Kata Kunci:** *downloader*, mikrokontroler AVR, radio frekuensi, operasi memori, sistem multitarget, sisi target.

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Penggunaan mikrokontroler untuk berbagai tujuan seperti contohnya pendidikan, riset, perancangan sistem elektronika dan digital mulai dari yang sederhana sampai yang kompleks sudah bukan merupakan suatu hal yang baru lagi. Berbagai fasilitas dan kemudahan yang ditawarkan oleh produsen mikrokomputer/mikrokontroler pada sebuah *chip* mikrokontroler dari waktu ke waktu dengan didukung pula aspek ekonomis, kepraktisan dan kehandalan dari mikrokontroler semakin membuat orang-orang lebih memilih mikrokontroler sebagai jantung/pengendali utama dari sistem elektronika yang dibuat.

Kontes Robot Indonesia (KRI) merupakan suatu *event* perlombaan robotika nasional yang diselenggarakan oleh Direktorat Jenderal Pendidikan Tinggi melalui Direktorat Pembinaan Penelitian dan Pengabdian kepada Masyarakat. Kegiatan KRI ini bertujuan untuk menumbuhkembangkan dan meningkatkan kreatifitas mahasiswa di Perguruan Tinggi, mengaplikasikan Ilmu Pengetahuan dan Teknologi ke dalam dunia nyata, meningkatkan kepekaan mahasiswa dalam pengembangan bidang teknologi robotika dan membudayakan iklim kompetitif di lingkungan perguruan tinggi. Salah satu Instansi Perguruan Tinggi yang ikut berpartisipasi aktif dengan mengirimkan perwakilan timnya dalam KRI adalah Teknik Elektro Universitas Brawijaya.

Mikrokontroler dipilih sebagai jantung pengendali utama dari sistem elektronika yang terdapat dalam sebuah robot KRI terutama untuk robot-robot otomatis. Hal ini dikarenakan dari segi biaya mikrokontroler yang murah, praktis dan performanya yang cukup handal bagi sebuah robot otomatis untuk menjalankan misi dalam pertandingan KRI. Segi kehandalan dan kepraktisan dari sebuah mikrokontroler bagi sistem kendali robot menjadikan mikrokontroler sebagai pilihan utama. Kecepatan dalam proses pemrograman ulang (baca, tulis atau hapus memori) (*Memory Programming/Downloading*) mikrokontroler selama proses *development* program robot otomatis KRI sangatlah penting.

Penggunaan mikrokontroler dalam sistem elektronik robot dari team TEUB mencakup banyak hal, ini dikarenakan untuk team yang sekarang sudah mengembangkan penggunaan banyak mikrokontroler dalam sistem keseluruhan yang nantinya menunjang pergerakan robot atau disebut dengan multi-mikrokontroler. Hal ini bertujuan untuk mempercepat proses eksekusi dan menghasilkan pergerakan robot yang sigap dan tangkas sehingga mampu berkompetisi dalam perlombaan.

Dalam proses untuk mempersiapkan robot kondisi siap lomba maka diperlukan *trial* dan *error* untuk menemukan sistem pergerakan yang optimal dengan cara mencoba setiap parameter-parameter logika terapan pemrogram untuk dijalankan pada lapangan simulasi lomba. Dalam proses tersebut sering terdapat perubahan parameter dalam tahap menyesuaikan dengan kondisi lomba atau dengan kata lain perubahan kode program untuk kondisi tertentu dalam pencarian performa pergerakan robot secara optimal. Sehingga diperlukan adanya proses baca tulis dan hapus memori dari mikrokontroler Atmel-AVR secara berulang-ulang.

Untuk lebih menunjang segi kepraktisan pada saat proses baca, tulis dan hapus memori mikrokontroler khususnya dalam hal ini dibatasi untuk mikrokontroler Atmel-AVR maka sebagai skripsi ini penulis rancang dan buatlah “Sistem *Downloader* Data Memori Mikrokontroler Atmel-AVR Melalui Radio Frekuensi Menggunakan Sistem Multi Target: Sisi Pemrogram”. Penelitian ini merupakan pengembangan dari skripsi saudara Hendy Eka Hardana (0301063059) yang berjudul ”Sistem *Downloader* Data Memori Mikrokontroler Atmel-AVR Melalui Inframerah: Sisi Pemrogram”. Dengan pengembangan dari segi peningkatan kecepatan transfer data dan kompatibel dengan sistem multi-mikrokontroler.

## 1.2 Rumusan Masalah

Dalam perancangan ini rumusan masalah ditekankan pada:

- 1) Bagaimana merancang rangkaian pemancar dan penerima data dengan menggunakan gelombang radio pada frekuensi radio menggunakan modul RF.
- 2) Bagaimana merancang protokol komunikasi data dengan menggunakan gelombang radio pada frekuensi radio menggunakan serial komputer.

- 3) Bagaimana merancang perangkat lunak yang dapat memilih mikrokontroler target.
- 4) Bagaimana merancang sebuah sistem pusat data *downloader* memori mikrokontroler Atmel-AVR melalui komunikasi dengan menggunakan gelombang radio pada frekuensi radio untuk mikrokontroler Atmel-AVR yang dapat menggantikan keberadaan media kabel untuk PC berbasis Windows.

### 1.3 Ruang Lingkup

Pada skripsi ini dibatasi oleh hal-hal sebagai berikut:

- 1) Modul pemancar dan penerima yang digunakan adalah YS-1020.
- 2) Jarak jangkauan komunikasi dengan menggunakan radio frekuensi diasumsikan mencapai 20 meter dalam kondisi ruang terbuka. Hal ini akan menjadi dasar dalam pengujian transfer data unit RF modul.
- 3) Pembahasan difokuskan pada proses pengiriman perintah dan data untuk operasi memori yang ditujukan ke memori mikrokontroler target seri Atmel AVR.
- 4) Jumlah mikrokontroler yang terpasang dalam sistem yaitu empat buah, hal ini dikondisikan dengan pemakaian rata-rata untuk tiap robot otomatis di lomba KRI/KRCI.

### 1.4 Tujuan

Tujuan skripsi ini adalah merancang dan membuat sistem penerima dan pengolah data. Sistem penerima dan pengolah data ini merupakan salah satu bagian “Sistem *Downloader* Data Memori Mikrokontroler Atmel-AVR Melalui Radio Frekuensi Menggunakan Sistem Multi Target”.

### 1.5 Sistematika Penulisan

Penulisan skripsi ini terdiri dari enam bagian dengan sistematika sebagai berikut:

#### **Bab I: Pendahuluan**

Berisi tentang uraian latar belakang, tujuan, batasan masalah, rumusan masalah serta sistematika penulisan.

**Bab II: Tinjauan Pustaka**

Membahas teori-teori yang mendukung dalam perencanaan dan pembuatan sistem.

**Bab III: Metode Penelitian**

Membahas tentang metode yang digunakan dalam penyusunan skripsi ini.

**Bab IV: Perancangan dan Pembuatan Alat**

Berisi perancangan dan perealisasiian alat yang meliputi spesifikasi, perencanaan blok diagram, prinsip kerja dan realisasi alat.

**Bab V: Pengujian dan Analisis**

Membahas tentang proses pengujian dan analisis data yang diperoleh dari sistem yang telah dibuat.

**Bab VI: Kesimpulan dan Saran**

Memuat kesimpulan dan saran dari penulisan skripsi ini.



## BAB II

### TINJAUAN PUSTAKA

Untuk merencanakan dan merealisasikan Sistem *Downloader* Data Memori Mikrokontroler Atmel-AVR Melalui Radio Frekuensi Menggunakan Sistem Multi Target: Sisi Pemrogram, dibutuhkan pemahaman tentang berbagai hal yang mendukung. Pemahaman ini akan bermanfaat untuk merancang perangkat keras dan perangkat lunak sistem. Pengetahuan yang mendukung perencanaan dan realisasi alat meliputi:

1. Spesifikasi dokumen heksadesimal Intel (*Intel Hex File*) 8 bit
2. Transmisi data
3. RF (Radio Frekuensi)
4. Metode deteksi kesalahan pada transmisi data
5. Protokol komunikasi.
6. Bahasa pemrograman delphi

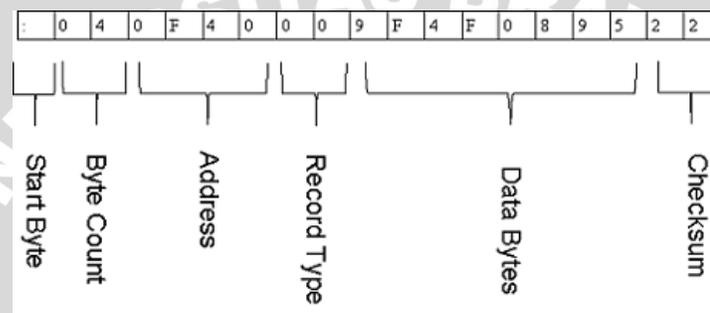
#### 2.1 Dokumen Heksadesimal Intel (*Intel Hex File*) 8 bit

Format dokumen *Hex* Intel 8 bit adalah format dokumen *hex* yang paling umum digunakan di seluruh dunia. Terdapat pula format dokumen *hex* dari Motorola dan beberapa yang lain. Format dokumen heksadesimal memenuhi syarat sebagai *input* bagi pemrogram PROM ataupun perangkat *emulator*. Format heksadesimal adalah sebuah cara untuk merepresentasikan dokumen objek (*binary object file*) ke format ASCII. Format *hex* Intel ini diikuti pula oleh *compiler-compiler* program mikrokontroler Atmel AVR.

Terdapat 3 format dokumen objek untuk masing-masing jenis mikroprosesor Intel yakni format 8 bit, 16 bit dan 32 bit. Format dokumen objek heksadesimal 8 bit memungkinkan untuk menempatkan kode dan data di dalam 16 bit alamat memori pada prosesor Intel 8 bit. Format dokumen objek heksadesimal 16 bit memungkinkan untuk menempatkan dalam 20 bit segmen memori pada prosesor Intel 16 bit. Format dokumen objek heksadesimal 32 bit memungkinkan untuk menempatkan kode dan data di dalam 32 bit alamat memori pada prosesor Intel 32 bit.

Representasi heksadesimal dari biner dikodekan menjadi karakter-karakter alfanumerik ASCII. Sebagai contoh, nilai biner 8 bit 0011-1111 adalah 3F dalam heksadesimal. Untuk menjadikannya ke format ASCII, kode ASCII dari karakter '3' adalah 0011-0011 atau 33H sedangkan kode ASCII dari karakter 'F' adalah 0100-0110 atau 46H. Sehingga menjadi 3346H dalam representasi ASCII. Representasi ASCII ini membutuhkan *byte* dua kali lebih banyak daripada representasi nilai sebenarnya dalam biner atau heksadesimal itu sendiri.

Tiap-tiap baris pada dokumen hex Intel terdiri atas 6 bagian, seperti yang ditunjukkan dalam Gambar 2.1 berikut ini.



**Gambar 2.1** Bagian-bagian yang terdapat dalam satu baris pada dokumen hex Intel  
**Sumber:** Scienceprog, 2006

Penjelasan untuk tiap-tiap bagian yang ditunjukkan dalam Gambar 2.1 adalah sebagai berikut:

- *Start*, (satu digit) selalu dimulai dengan karakter ':'
- *Byte Count*, (dua digit) menunjukkan jumlah byte data dalam setiap baris
- *Address*, (empat digit) menunjukkan posisi data pada memori
- *Record Type*, (dua digit) menunjukkan tipe data
- *Data*, data sejumlah n byte ( $2^n$  digit)
- *Checksum*, (dua digit) sebagai akhir dari satu baris. Merupakan komplemen 2 dari penjumlahan (dalam heksadesimal) dari seluruh bagian sebelumnya. Contoh perhitungan checksum:

Satu baris pada dokumen hex adalah:  
 100010004EC04DC04CC04BC04AC049C048C047C08C

Maka

$$10+00+10+00+4E+C0+4D+C0+4C+C0+4B+C0+4A+C0+49+C0+48+C0+47+C0= 874$$

Diambil 2 digit akhir yakni 74. Komplemen dua dilakukan dengan  $100-74= 8C$ . Checksum yang didapat adalah 8C.

Format dokumen *hex* Intel memiliki 6 tipe data (*Record Type*), yaitu:

- 00 – *Data record, record data*.
- 01 – *End of file record*, akhir dari dokumen, yaitu “:00000001FF”.
- 02 – *Extended Segment address record*, segmen alamat *extended*. Menunjukkan basis segmen alamat ketika 16 bit tidak mencukupi untuk pengalamatan memori.
- 03 – *Start segment address record*, awal segmen alamat. Menandakan basis alamat awal.
- 04 – *Extended Linear Address Record*, alamat linier *extended*. Mengijinkan pengalamatan dengan 32 bit.
- 05 – *Start Linear Address Record*.

## 2.2. Transmisi Data

Transmisi data merupakan proses pengiriman data dari satu sumber ke penerima data. Ada tiga komponen utama dalam proses transmisi data, yaitu: sumber data (*source*), media transmisi (*transmission media*), dan penerima (*receiver*). Media transmisi yang dapat digunakan sebagai jalur transmisi atau carrier dari data yang dikirimkan dapat berupa kabel maupun gelombang elektromagnetik. Bila sumber dan penerima data jaraknya cukup jauh, media transmisinya dapat digunakan gelombang elektromagnetik yang dipancarkan melalui udara terbuka yang dapat berupa sistem laser, gelombang radio atau sistem satelit.

Pengiriman data serial melalui media udara menggunakan gelombang radio sebagai pembawa data. Secara sederhana proses pengiriman data menggunakan gelombang radio adalah, sinyal informasi atau data yang akan dikirimkan ditumpangkan terlebih dahulu ke sinyal pembawa.

Proses menumpangkan sinyal informasi ini disebut dengan modulasi. Gabungan antara kedua sinyal tersebut kemudian akan dipancarkan oleh transmiiter. Pada receiver, gelombang pembawa yang membawa sinyal informasi tersebut diterima, kemudian dipisahkan antara gelombang pembawa dan sinyal informasi, sehingga diperoleh kembali sinyal informasi. Proses ini disebut demodulasi.

### 2.3 Unit RF (Radio Frekuensi)

Untuk bagian penerima dari Sistem *Downloader* ini akan menggunakan modul RF YS-1020 ditunjukkan dalam Gambar 2.2.



**Gambar 2.2** Ys-1020 (Radio Frekuensi)  
**Sumber:** ShenZhen Yishi, 2009

Modul Low power RF seri YS-1020 didesain untuk sistem transmisi data tanpa kabel. Ys-1020 mengadopsi IC Chipeon CC1020RF, bekerja pada aliran pita frekuensi, penerima dan pentransmisi yang sudah terintegrasi pada modul. Alat ini langsung dapat dihubungkan dengan prosesor monolithic, pc, perangkat RS485, dan komponen UART dengan RS-232, RS-485 dan UART/TTL port level interface.

#### 2.3.1 Fitur Utama Produk:

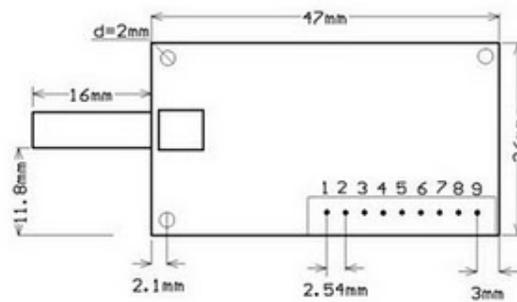
- 1) Menggunakan Frekuensi pembawa: 433/450/868 Mhz atau pilihan lainnya
- 2) Perangkat penghubung: RS-232/RS-485/TTL
- 3) Mendukung banyak saluran komunikasi: 8 saluran, dapat diperluas untuk 16/32 saluran
- 4) Baud-rate di udara: 1200/2400/4800/9600 bps, Pengiriman data secara transparan: apa yang telah diterima persis dengan apa yang dikirim, cocok dengan standarisasi atau bukan standarisasi protokol pengguna
- 5) Format penghubung: ketetapan untuk pengguna 8N1/8E1/801, atau disesuaikan dengan format penghubung lainnya

- 6) Pemodulasian: GFSK. Berdasarkan pada modulasi GFSK, tahan terhadap gangguan yang besar dan Bit Error Rate yang rendah
- 7) Half duplex: penggabungan untuk penerima dan pengirim yang menjadi satu didalam satu alat, kemampuan merubah dengan sendirinya untuk menerima dan mengirim selama 10ms
- 8) Penggunaan daya yang rendah dan fungsi terjaga
- 9) Batas temperatur : -35 derajat celsius sampai 75 derajat celsius (-31 ~ 167 F)
- 10) Batas kelembaban: 10% ~ 90% kelembaban relatif tanpa pengembunan

### 2.3.2 Spesifikasi

- 1) Daya untuk RF:  $\leq 10\text{mW}/10\text{dBm}$
- 2) Arus yang diterima:  $\leq 25\text{mA}$
- 3) Arus mengirim:  $\leq 40\text{mA}$
- 4) Arus sleep mode:  $20\mu\text{A}$
- 5) Suplai daya: Tegangan DC 5V atau 3.3V
- 6) Sensitivitas dalam menerima :
  - 115 dBm (@9600bps)
  - 120 dBm (@1200bps)
- 7) Ukuran: 47mm x 26mm x 10mm (tanpa antenna)
- 8) Jangkauan :
  1.  $\leq 0.5\text{m}$  (BER= $10^{-3}$ @9600bps, saat antena berada 2 meter diatas permukaan tanah pada area terbuka)
  2.  $\leq 0.8\text{m}$  (BER= $10^{-3}$ @1200bps, saat antena berada 2 meter diatas permukaan tanah pada area terbuka)

Sedangkan untuk konfigurasi pin YS-1020 ditunjukkan dalam Gambar 2.3. Dan untuk penjelasan dari tiap-tiap pin ditunjukkan dalam Tabel 2.1.



**Gambar 2.3** Konfigurasi Pin YS-1020  
**Sumber:** ShenZhen Yishi, 2009

**Tabel 2.1** Konfigurasi YS-1020

No. pin	Nama pin	Penjelasan	Level	Koneksi dengan terminal
1	GND		Ground	
2	Vcc	$+3,3 \pm 5,5$ V		
3	RXD/TTL	TTL	TxD	
4	TXD/TTL	TTL	RxD	
5	DGND			
6	A dari RS-485 atau TXD dari RS-232		A(RxD)	
7	B dari RS-485 atau RXD dari RS-232		B(TxD)	
8	Sleep control (input)	TTL	Sleep signal	Low level sleep
9	Ex-factory testing			

**Sumber:** ShenZhen Yishi, 2009

#### 2.4 Protokol Komunikasi

Protokol adalah sebuah aturan atau standar yang mengatur atau mengizinkan terjadinya hubungan, komunikasi, dan perpindahan data antara dua atau lebih perangkat komputer. Protokol dapat diterapkan pada perangkat keras, perangkat lunak atau kombinasi dari keduanya. Pada tingkatan yang terendah, protokol mendefinisikan koneksi perangkat keras.

Protokol perlu diutamakan pada penggunaan standar teknis, untuk menspesifikasi bagaimana membangun komputer atau menghubungkan peralatan

perangkat keras. Protokol secara umum digunakan pada komunikasi *real-time* dimana standar digunakan untuk mengatur struktur dari informasi untuk penyimpanan jangka panjang.

Kebanyakan protokol memiliki salah satu atau beberapa dari hal berikut:

- Melakukan deteksi adanya koneksi fisik atau ada tidaknya komputer atau perangkat lainnya.
- Melakukan metode "jabat-tangan" (*handshaking*).
- Bagaimana mengawali dan mengakhiri suatu pesan.
- Bagaimana format data yang digunakan.
- Yang harus dilakukan saat terjadi kerusakan pesan atau pesan yang tidak sempurna.
- Mengakhiri suatu koneksi.

Dalam membuat protokol ada tiga hal yang harus dipertimbangkan, yaitu efektivitas, kehandalan, dan kemampuan dalam kondisi gagal. Agar protokol dapat dipakai untuk komunikasi diberbagai perangkat maka dibutuhkan standarisasi protokol.

#### **2.4.1 Komunikasi Serial**

Komunikasi yang digunakan sebagai penghubung antara penerima-pemancar frekuensi radio dengan komputer adalah komunikasi serial. Komunikasi serial dipilih berdasarkan kebutuhan pada perancangan sistem, dengan pengiriman data secara berurutan.

Devais yang menggunakan kabel serial untuk komunikasinya ada dua macam, DTE (*Data Terminal Equipment*) dan DCE (*Data Communications Equipment*). Devais yang termasuk DTE adalah modem, plotter, dan sebagainya, sementara devais yang termasuk DCE adalah komputer atau terminal.

Spesifikasi dari port serial tercantum dalam EIA (*Electronics Industry Association*) standar RS232C. Beberapa parameter yang dicantumkan diantaranya:

- 1). Logika nol (disebut *spacing*), berada pada level tegangan antara +3 dan +25 volt.
- 2). Logika satu (disebut *marking*), berada pada level tegangan antara -3 dan -25 volt.

- 3). Level tegangan antara -3 sampai +3 volt adalah level tegangan yang tidak valid.
- 4). Tegangan rangkaian terbuka tidak boleh melebihi 25 volt.
- 5). Arus rangkaian hubung singkat tidak boleh melebihi 500 mA.

Ada dua macam port serial yang digunakan, yaitu tipe D dengan 25 konektor (DB25) dan tipe D dengan 9 konektor (DB9). Pada komputer, kedua port ini merupakan jenis *male*. Fungsi dari pin-pin yang terdapat pada DB25 dan DB9 ditunjukkan dalam Tabel 2.2

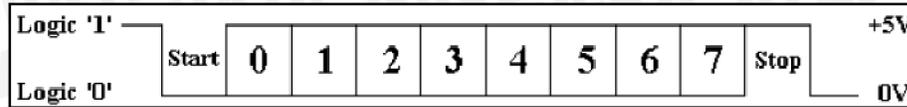
**Tabel 2.2** Fungsi pin pada DB25 dan DB9

Pin DB25	Pin DB9	Nama Pin	Fungsi
Pin 2	Pin 3	TD ( <i>Transmit Data</i> )	Data Keluaran Serial (TXD)
Pin 3	Pin 2	RD ( <i>Receive Data</i> )	Data Masukan Serial (RXD)
Pin 4	Pin 7	RTS ( <i>Request to Send</i> )	Komputer siap berkomunikasi
Pin 5	Pin 8	CTS ( <i>Clear to Send</i> )	Devais siap berkomunikasi
Pin 6	Pin 6	DSR ( <i>Data Set Ready</i> )	Devais siap berhubungan dengan komputer
Pin 7	Pin 5	SG ( <i>Signal Ground</i> )	<i>Ground</i>
Pin 8	Pin 1	CD ( <i>Carrier Detect</i> )	Aktif ketika devais membaca "carrier" pada devais lainnya
Pin 20	Pin 4	DTR ( <i>Data Terminal Ready</i> )	Fungsinya berkebalikan dengan DSR
Pin 22	Pin 9	RI ( <i>Ring Indicator</i> )	Aktif ketika devais mendeteksi sinyal bunyi pada PSTN

Sumber : Peacock, 1998: 5

Komunikasi serial yang digunakan dalam perancangan ini adalah dengan sistem transmisi data RS-232 (*Recommended Standard 232*) yang merupakan standar penghubung antara DTE dan DCE. Komunikasi dengan RS232 merupakan komunikasi asinkron, yaitu sinyal *clock* tidak dikirim bersamaan

dengan data. Setiap kata yang dikirim memiliki bit *start*, bit *stop*, dan *clock* internal.



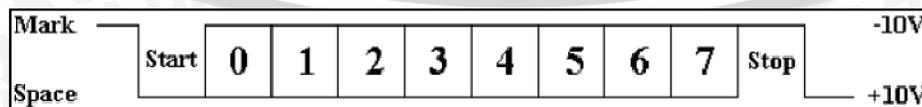
Gambar 2.4 Format data logika TTL/CMOS  
 Sumber : Peacock,1998: 33

Gambar 2.4 menunjukkan format data untuk logika TTL/CMOS pada UART. Format data di atas menggunakan format standar 8N1, yaitu 8 bit data, tidak ada bit paritas, dan 1 bit stop. Pada RS232, keadaan *idle* merupakan logika 1 (*marking*). Transmisi data dimulai dengan bit *start* yang berlogika rendah, kemudian bit data dikirim secara berurutan satu demi satu. Bit yang pertama dikirim adalah LSB. Untuk menandakan data telah terkirim, bit *stop* (logika 1) dikirim.

Gambar 2.4 menunjukkan bit setelah bit stop berlogika rendah. Hal ini menandakan ada bit *start* baru, dan ada data baru yang akan dikirim. Jika tidak ada data baru yang akan dikirim, maka kondisi akan menjadi *idle* (logika 1). Jika saluran data tetap berlogika 0 selama waktu yang diperlukan untuk mengirim 1 format data, kondisi ini disebut *break*. Untuk menghindari terjadinya *break*, kondisi saluran data harus dikembalikan ke kondisi *idle*.

Format data dengan bit start dan bit stop ini disebut *frame* data. Kondisi *framing error* terjadi jika bit stop diterima sebagai logika 0. Hal ini dapat terjadi jika kecepatan yang digunakan pada pengirim dan penerima data berbeda.

Gambar 2.4 di atas hanya relevan pada UART, sedangkan level tegangan RS232 antara +3 sampai +25 volt pada logika 0 dan -3 sampai -25 volt pada logika 1. Untuk dapat berkomunikasi antara UART dan RS232 diperlukan konverter tegangan RS232.

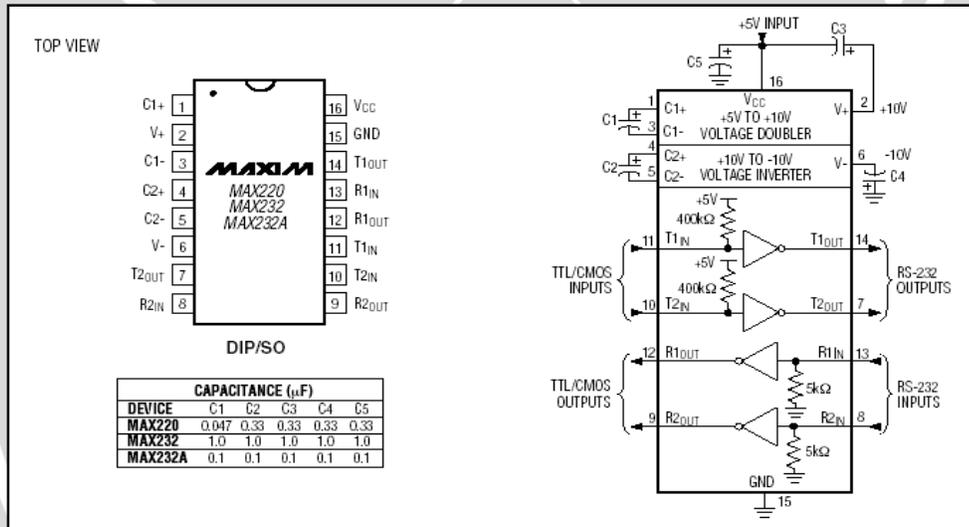


Gambar 2.5 Format data pada konverter level tegangan RS232  
 Sumber :Peacock, 1998: 33

Gambar 2.5 menunjukkan format data pada saluran pengirim dan penerima pada port RS232. Perbedaan tegangan pada devais dan komputer dapat diatasi dengan konverter level tegangan RS232.

### 2.4.2 MAX 232

Level Tegangan dari RS232 adalah +3 sampai +15 volt untuk logika “0” sedangkan -3 sampai -15 volt untuk logika “1”. Level tegangan ini berbeda dengan level tegangan logika dari mikrokontroler yang bertipe TTL/CMOS dengan *supply* 5 V yang memiliki keluaran untuk logika tinggi minimal 2,4 volt dan logika rendah maksimal 0,4 volt, sehingga dibutuhkan IC MAX 232 yang berfungsi sebagai penyetara level tegangan logika. Gambar IC MAX 232 dan rangkaian minimum sistemnya ditunjukkan dalam Gambar 2.6.



Gambar 2.6 IC MAX232 dan Rangkaian minimum sistemnya  
 Sumber : Maxim, 2003: 14

IC MAX 232 memiliki empat bagian yaitu *dual charge pump*, konverter tegangan, RS 232 *driver* dan RS 232 *receiver*. *Dual charge pump* mengubah tegangan masukan +5V menjadi ±10V (tak terbebani) pada RS 232 *driver*. Konverter pertama menggunakan kapasitor C1 untuk menggandakan tegangan +5V menjadi +10V di C3 pada keluaran V+. Konverter kedua menggunakan kapasitor C2 untuk membalik +10V menjadi -10V di C4 pada keluaran V-.

Keluaran RS 232 *driver* berayun dari  $\pm 8V$  ketika dibebani dengan 5k (nominal) dengan Vcc sebesar 5V. *Pull up* resistor yang tersambung dengan Vcc menyebabkan keluaran *driver* yang tidak digunakan pada kondisi rendah karena semua *driver* adalah terbalik. Spesifikasi EIA/TIA-232 E dan V.28 menentukan bahwa level tegangan yang lebih dari 3V adalah berlogika 0. Jadi semua *receiver* adalah terbalik. *Input Threshold* ditentukan pada 0,8V dan 2,4V sehingga keluaran *receiver* akan sesuai dengan level tegangan TTL.

### 2.5 Metode Deteksi Kesalahan pada Transmisi Data menggunakan *Checksum*

Tujuan dari metode deteksi kesalahan adalah untuk membuat penerima mampu untuk memeriksa apakah data yang ditransmisikan melalui sebuah media yang rawan terhadap noise benar atau tidak. Untuk melakukan hal ini, pemancar memilih suatu nilai (disebut sebagai *checksum*) yang kemudian mengkombinasikannya dengan data yang dikirim menggunakan algoritma tertentu. Dengan algoritma yang sama pula, maka penerima melakukan perhitungan terhadap data yang diterima untuk dibandingkan dengan *checksum* dan pada akhirnya dapat diketahui apakah data yang diterima benar atau tidak.

Sebagai contoh deteksi kesalahan (*checksum*) sederhana adalah sebagai berikut:

Data: 6 23 4

Data dengan checksum: 6 23 4 33

Data setelah transmisi: 6 27 4 33

Keterangan: lebar 1 *byte* data adalah 0-256

Dari contoh di atas, *byte* kedua dari data yang diterima rusak/salah karena nilainya berubah menjadi 27. Akan tetapi penerima dapat mendeteksinya dengan membandingkan *checksum* (33) yang diterima dengan *checksum* perhitungan yakni 37 (6+27+4). Namun ketika *checksum* itu sendiri mengalami kerusakan, akan memungkinkan data yang seharusnya benar dianggap sebagai sebuah kesalahan. Hal yang paling fatal adalah bila data dan *checksum* kesemuanya mengalami kerusakan/kesalahan. Hal ini menyebabkan proses transmisi data menjadi tidak konsisten. Seluruh kemungkinan di atas memang tidak dapat dihindari, cara terbaik yang dapat dilakukan adalah memperkecil kemungkinan

dengan menambah jumlah data pada *checksum* (sebagai contoh, memperlebar data *checksum* dari 1 *byte* menjadi 2 *byte*).

Dari contoh perhitungan *checksum* di atas dengan lebar data 1 *byte*, terdapat 1/256 kemungkinan *error* tidak dapat terdeteksi. Ketika lebar data kita ganti menjadi 2 *byte* (0-65536) maka kemungkinan terjadi *error* akan lebih kecil menjadi 1/65535. Namun tetap saja hal ini tidaklah cukup karena metode *checksum* di atas masih terlalu sederhana dan tidak cukup acak. *Checksum* tidak dapat mendeteksi *error* yang terjadi pada banyak paket (*byte* data) dan tidak dapat mendeteksi *error* pada *checksum* yang diberikan itu sendiri.

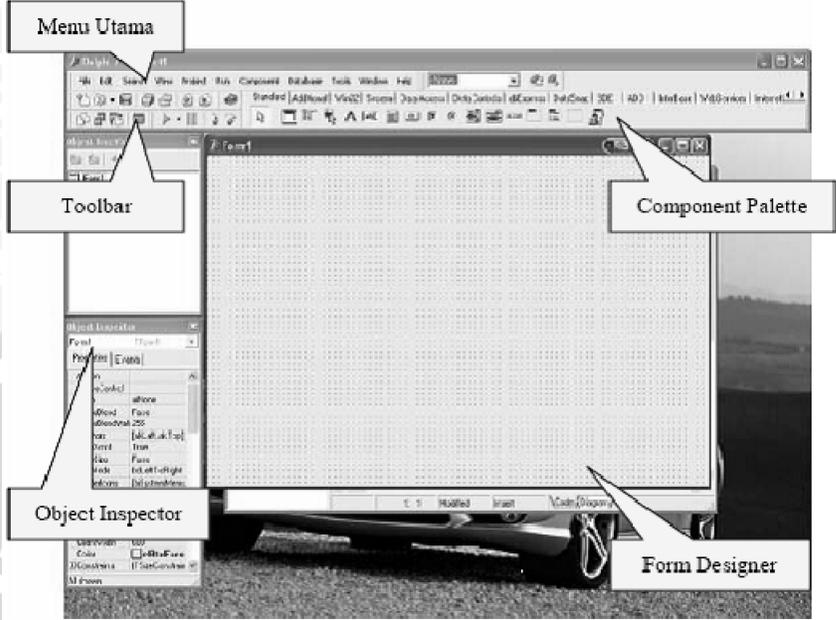
## 2.6 Compiler Delphi

Borland Delphi atau biasa disebut Delphi merupakan perangkat lunak pengembangan aplikasi yang sangat populer di lingkungan Linux maupun Windows. Delphi merupakan bahasa pemrograman yang mempunyai cakupan kemampuan yang luas dan canggih. Berbagai jenis aplikasi dapat dibuat dengan Delphi, termasuk aplikasi untuk mengolah teks, grafik, angka, dan aplikasi *web*. Secara umum, kemampuan Delphi adalah menyediakan komponen-komponen dan bahasa pemrograman yang handal, sehingga memungkinkan untuk membuat program aplikasi sesuai dengan keinginan, dengan tampilan dan kemampuan yang canggih.

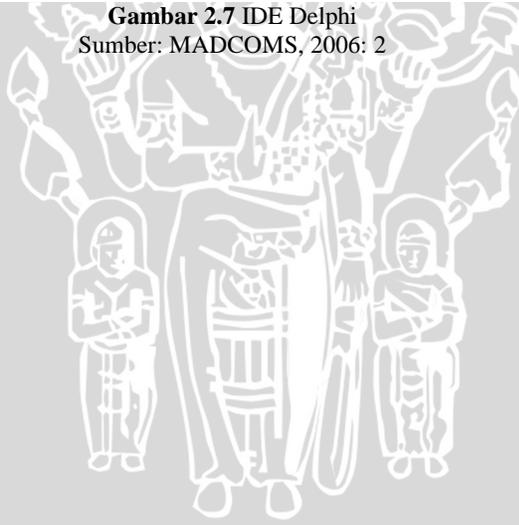
Untuk mempermudah pemrograman dalam membuat program aplikasi, Delphi menyediakan fasilitas pemrograman yang sangat lengkap. Fasilitas pemrograman tersebut dibagi dalam dua kelompok, yaitu *object* dan bahasa pemrograman. Secara ringkas, *object* adalah suatu komponen yang mempunyai bentuk fisik dan biasanya dapat dilihat (*visual*). *Object* biasanya dipakai untuk melakukan tugas-tugas tertentu dan mempunyai batasan-batasan tertentu. Sedangkan bahasa pemrograman secara singkat dapat disebut sebagai sekumpulan teks yang mempunyai arti tertentu dan disusun dengan aturan tertentu serta untuk menjalankan tugas tertentu. Delphi menggunakan struktur bahasa pemrograman *Object Pascal* yang sudah sangat dikenal di kalangan pemrograman profesional.

Khusus untuk pemrograman, Delphi menyediakan *object* yang canggih, dan lengkap, sehingga memudahkan pemrogram merancang, membuat, dan menyelesaikan aplikasi yang diinginkan. Delphi dapat menangani data dalam

berbagai format , misalnya format MS-Access, Oracle, MySQL, MS SQL, DB2, dll. Tampilan IDE Delphi ditunjukkan dalam Gambar 2.7.



Gambar 2.7 IDE Delphi  
Sumber: MADCOMS, 2006: 2



## BAB III

### METODE PENELITIAN

Metode yang digunakan dalam penulisan skripsi ini adalah sebagai berikut:

#### 3.1 Studi Literatur

Studi literatur dilakukan untuk mendapatkan pengetahuan yang dapat mendukung penyusunan skripsi ini. Studi literatur dalam skripsi ini meliputi Dokumen Heksadesimal Intel 8 Bit, Unit RF(*Radio Frequency*), Metode Deteksi Kesalahan Pada Transmisi Data, Sistem antarmuka, dan Penampilan data pada komputer menggunakan pemrograman visual Delphi.

#### 3.2 Perancangan dan Pembuatan Alat

Pada tahap perancangan alat, dibuat suatu blok diagram fungsional dari rangkaian yang direncanakan. Perancangan rangkaian dilakukan pada tiap-tiap blok untuk mempermudah perancangan serta penentuan nilai komponen yang digunakan. Perancangan perangkat lunak dilakukan dengan membuat diagram alir untuk program utama dan sub program. Secara garis besar perancangan alat dilakukan dalam tahap berikut:

1. Penentuan spesifikasi alat.
2. Pembuatan diagram blok sistem keseluruhan.
3. Perancangan perangkat keras yang terdiri atas rangkaian penerima dan pemancar RF 434A dan rangkaian antarmuka komputer.
4. Perancangan perangkat lunak yang menggunakan software computer.

#### 3.3 Pengujian Alat

Pengujian alat ini bertujuan untuk mengetahui kesesuaian kinerja alat hasil perancangan dengan spesifikasi yang direncanakan. Adapun bentuk pengujian yang dilakukan adalah sebagai berikut:

- a. Pengujian Perangkat Keras

Pengujian dilakukan tiap blok perancangan untuk memudahkan analisis.

Pembuatan alat untuk perangkat keras meliputi pembuatan PCB

(pembuatan *lay out*, pengetesan dan pengeboran). Perakitan komponen dan penyolderan dilakukan pada PCB.

b. Pengujian Perangkat Lunak

Pengujian ini bertujuan untuk mengetahui kinerja dari perangkat lunak PC yang telah dibuat apakah telah sesuai dengan spesifikasi yang direncanakan.

c. Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem dilakukan dengan menghubungkan bagian-bagian perangkat keras dan mengoperasikan perangkat lunak pada PC sehingga menjadi kesatuan sistem secara keseluruhan, yakni sisi pemrogram dan sisi target.

### 3.4 Pengambilan Kesimpulan dan Pemberian Saran

Tahap berikutnya dari penelitian adalah pengambilan kesimpulan dari alat yang dibuat. Pengambilan kesimpulan ini didasarkan pada kesesuaian antara teori dan praktek.

Tahap terakhir dari penulisan adalah pemberian saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang mungkin terjadi, kemungkinan pengembangan serta penyempurnaan penulisan.

## BAB IV

### PERANCANGAN DAN PEMBUATAN ALAT

Perancangan dan pembuatan alat yang akan dibahas di sini merupakan bagian dari “Sistem *Downloader* Data Memori Mikrokontroler Atmel-AVR Melalui Radio Frekuensi Menggunakan Sistem Multi Target”, yakni sisi pemrogram. Perancangan dan pembuatan alat untuk sisi pemrogram terdiri atas dua bagian, yaitu perancangan dan pembuatan perangkat keras serta perancangan dan pembuatan perangkat lunak. Perancangan dan pembuatan alat dilakukan secara bertahap. Hal ini untuk memudahkan proses analisis sistem. Aspek lain yang perlu dijelaskan dalam bab ini adalah penentuan spesifikasi alat yang dirancang, blok diagram dan prinsip kerja sistem.

#### 4.1 Spesifikasi Sistem

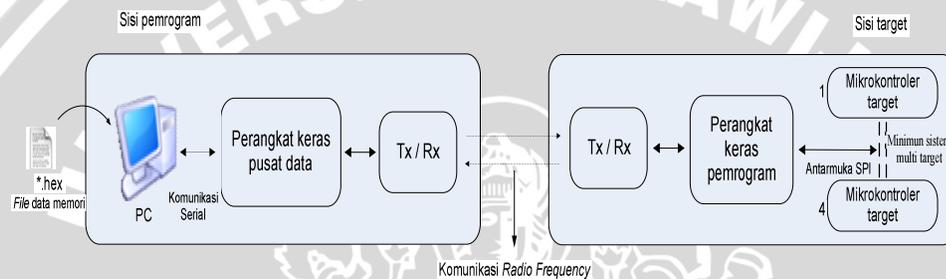
Spesifikasi sistem yang dirancang untuk sisi pemrogram adalah sebagai berikut:

- a. Perangkat lunak pada PC berbasis Windows dirancang untuk mengatur seluruh proses operasi memori (baca, tulis atau hapus memori) yang akan ditujukan ke mikrokontroler keluarga Atmel AVR di sisi target.
- b. Alat yang dirancang menggunakan komunikasi antara perangkat RF dengan komputer dilakukan secara serial dengan RS232.
- c. Komunikasi data serial menggunakan protokol USART dengan *baudrate* 9600 bps.
- d. Pemancar dan penerima frekuensi radio yang digunakan adalah modul YS-1020.
- e. Mampu berkomunikasi dengan empat mikrokontroler pada bagian sisi target.
- f. Perangkat keras menggunakan catu daya DC sebesar 5 V dan arus maksimum 50 mA yang bersumber dari *port* USB PC.

## 4.2 Perancangan Sistem

### 4.2.1 Perancangan Sistem Secara Keseluruhan

Sistem keseluruhan terdiri atas dua bagian. Bagian pertama adalah sisi pemrogram yang mencakup perangkat lunak PC dan perangkat keras sisi pemrogram dengan antarmuka serial ke PC. Bagian kedua adalah sisi target yang mencakup perangkat lunak dan perangkat keras sistem target dengan menggunakan antarmuka SPI (*Serial Peripheral Interface*) untuk proses serial *downloading* mikrokontroler target. Ilustrasi sistem secara keseluruhan ditunjukkan dalam Gambar 4.1.

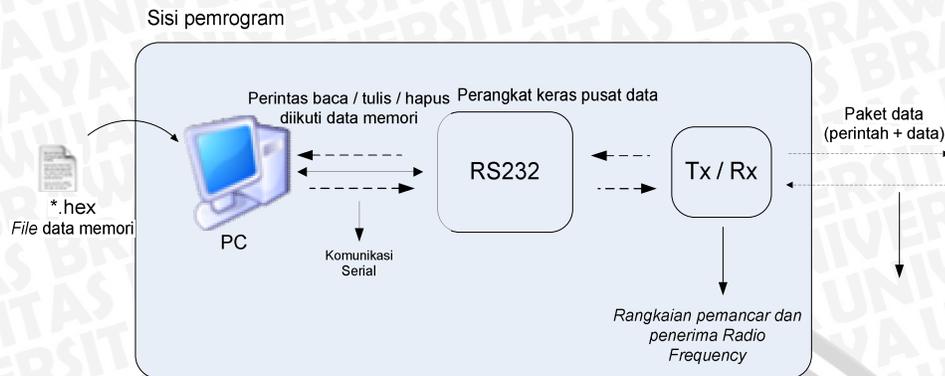


**Gambar 4.1** Sistem secara keseluruhan

Prinsip kerja sistem yang terdapat pada sisi target secara global bersifat pasif dan menunggu instruksi maupun data memori dari sisi pemrogram. Proses yang terjadi pada sisi target dikendalikan oleh sisi pemrogram. Perintah maupun data dikirim oleh sisi pemrogram secara serial melalui media radio frekuensi. Prinsip kerja sistem adalah sebagai berikut:

Sisi pemrogram terdiri atas:

- Perangkat lunak pada PC, bertindak sebagai antarmuka dengan pengguna untuk mengatur seluruh proses operasi memori (baca, tulis atau hapus memori) yang ditujukan ke mikrokontroler target.
- Perangkat lunak pada PC, bertindak untuk memilih mikrokontroler target mana yang akan dituju untuk melakukan proses operasi memori tersebut.
- Pada perangkat keras sisi pemrogram terdapat perangkat lunak yang berfungsi untuk proses transfer data-data (data perintah maupun data memori) dari/ke PC untuk diteruskan/diterima melalui komunikasi radio frekuensi dari/ke sisi target. Aliran data pada sisi pemrogram dapat ditunjukkan dalam Gambar 4.2.

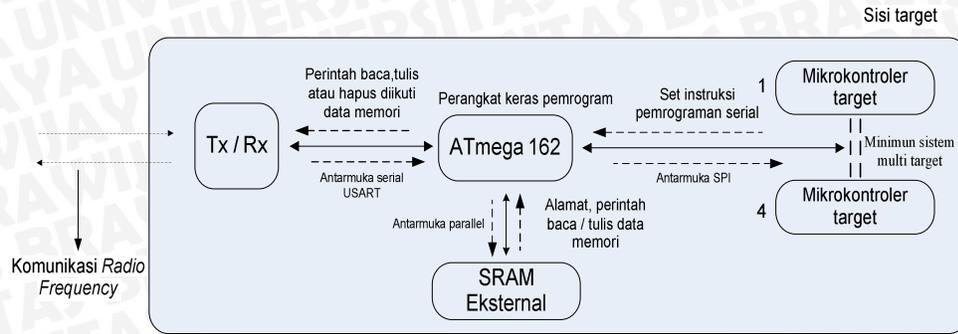


**Gambar 4.2** Aliran data di sisi pemrogram

- d) Perangkat keras radio frekuensi terdiri atas dua bagian, pemancar dan penerima. Berfungsi sebagai media pendukung komunikasi dengan radio frekuensi. Komunikasi data berlangsung dua arah secara serial pada bagian sisi pemrogram.

Sisi target terdiri atas:

- a) Perangkat keras unit radio frekuensi terdiri atas dua bagian, pemancar dan penerima. Berfungsi sebagai media pendukung komunikasi dengan radio frekuensi. Komunikasi data berlangsung dua arah secara serial dengan memanfaatkan fasilitas serial USART pada mikrokontroler sisi target.
- b) Di dalam mikrokontroler pada perangkat keras pemrogram terdapat pula perangkat lunak yang berfungsi untuk proses transfer data-data (data perintah dan data memori) dari/ke sisi pemrogram melalui komunikasi radio frekuensi, menampung sementara data-data tersebut ke SRAM eksternal, mengolah data untuk kemudian diteruskan ke mikrokontroler target yang dipilih dari sisi pemrogram. Aliran data pada sisi target dapat ditunjukkan dalam Gambar 4.3.

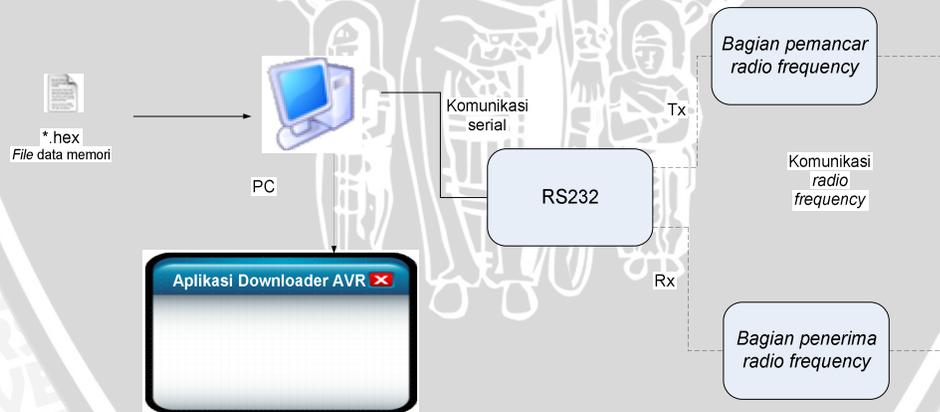


Gambar 4.3 Aliran data di sisi target

Mikrokontroler target yang diinginkan direncanakan untuk mikrokontroler keluarga Atmel AVR. Bagian ini sebenarnya bukan merupakan bagian utama dari sistem akan tetapi turut ditampilkan sebagai pendukung penjelasan.

#### 4.2.2 Perancangan Sisi Pemrograman

Penelitian ini dibagi menjadi dua bagian, sisi pemrogram dan sisi target. Yang akan penulis bahas di sini adalah sisi pemrogram. Sisi target dibahas oleh peneliti yang lain. Blok diagram perangkat keras sisi pemrogram secara lebih spesifik ditunjukkan dalam Gambar 4.4.



Gambar 4.4 Blok diagram sisi pemrogram

Melalui perangkat lunak PC yang terdapat di sisi pemrogram, maka pengguna akan memilih operasi memori (baca, tulis atau hapus memori program) yang ditujukan ke mikrokontroler target di sisi target dan juga memilih mikrokontroler mana yang akan digunakan.

Operasi memori dibedakan menjadi tiga bagian:

a. Operasi baca memori

Proses ini akan membaca isi memori yang terdapat dalam mikrokontroler target untuk ditampilkan di perangkat lunak pada PC sisi pemrogram. Data memori akan disimpan dalam format *file* Intel Hex.

b. Operasi tulis memori

Proses ini akan menulis memori sisi target dari perangkat lunak pada PC. Data dari *file* Intel Hex diambil dan kemudian dikirimkan ke sisi target.

c. Operasi hapus memori

Proses ini akan menghapus seluruh isi memori program dan EEPROM mikrokontroler target.

Ketiga operasi memori di atas dibedakan dengan cara memberikan data perintah yang berbeda-beda untuk masing-masing proses sesuai kesepakatan antara sisi pemrogram dengan sisi target.

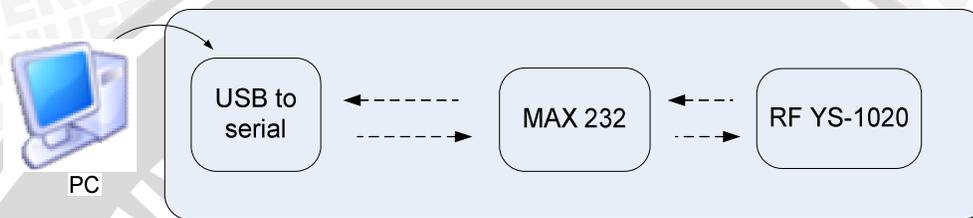
Proses pengiriman dan penerimaan data: di sisi pengirim data, data-data akan dikirimkan ke unit radio frekuensi. Unit pemancar radio frekuensi akan mengirimkan data ke sisi penerima data melalui komunikasi radio frekuensi. Unit penerima radio frekuensi di sisi penerima akan menerima data-data dari sisi pengirim untuk kemudian ditampung dan diolah. Proses pengiriman berlangsung hingga seluruh data telah selesai dikirimkan dari sisi pengirim ke penerima.

Proses konfirmasi data: konfirmasi berfungsi sebagai penanda berlangsungnya proses penerimaan data. Selama berlangsungnya proses pengiriman dan penerimaan data, juga terjadi proses konfirmasi. Proses konfirmasi data dilakukan oleh sisi penerima data dengan mengirimkan status data yang diterima ke sisi pengirim, yakni apakah berhasil atau gagal. Apabila gagal maka sisi penerima akan meminta pengulangan kembali pengiriman data oleh sisi pengirim dan apabila berhasil maka proses pengiriman dapat diteruskan untuk data berikutnya.

Pada sisi pemrogram ini, antarmuka yang digunakan antara PC dan mikrokontroler menggunakan antarmuka serial. Perangkat lunak pada PC berbasis Windows, dibuat dengan bahasa pemrograman Pascal menggunakan *compiler* Delphi.

### 4.3 Perancangan Perangkat Keras

Perancangan perangkat keras terdiri tiga bagian, yaitu USB to serial, rangkaian penerima dan pemancar RF yang bertujuan menerima data yang dikirim oleh pemancar RF dan memancar data yang dikirim oleh pemancar RF dan rangkaian MAX232 yang menyetarakan level logika data keluaran yang berupa data digital dengan level logika computer. Diagram blok dari keseluruhan perangkat keras ini ditunjukkan dalam Gambar 4.5.

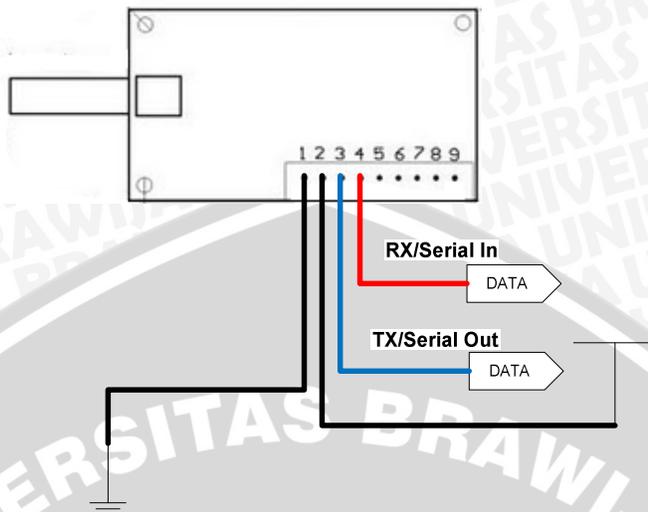


Gambar 4.5 Diagram blok dari keseluruhan perangkat keras

#### 4.3.1 Rangkaian Penerima dan Pemancar RF

Perancangan rangkaian pemancar dan penerima RF menggunakan modul RF YS-1020 yang dalam satu modul terdapat pemancar dan penerima data. Data yang dikirim oleh bagian pemancar yang menggunakan modul RF YS-1020 akan didemodulasi oleh modul YS-1020 kedua sehingga diperoleh sinyal data. Keluaran dan pembacaan data dari modul RF ini yaitu berupa data digital sehingga nantinya dapat dikomunikasikan dengan sistem komunikasi data serial.

Modul YS-1020 menggunakan modulasi didigital GFSK (*Gaussian frequency Shift Keying*). Modul ini memiliki frekuensi kerja yang beragam yaitu 433,92 MHz, 450 MHz, 868 MHz. Menggunakan tegangan catu daya DC dengan range antara 3.3–5.0 volt. Mempunyai daya output RF sebesar 10mW. Kecepatan data serial *typical* yang ditransmisikan adalah sebesar 9,6 kbps. Perancangan untuk rangkaian RF ditunjukkan dalam Gambar 4.6.



Gambar 4.6 Rangkaian pemancar dan penerima RF

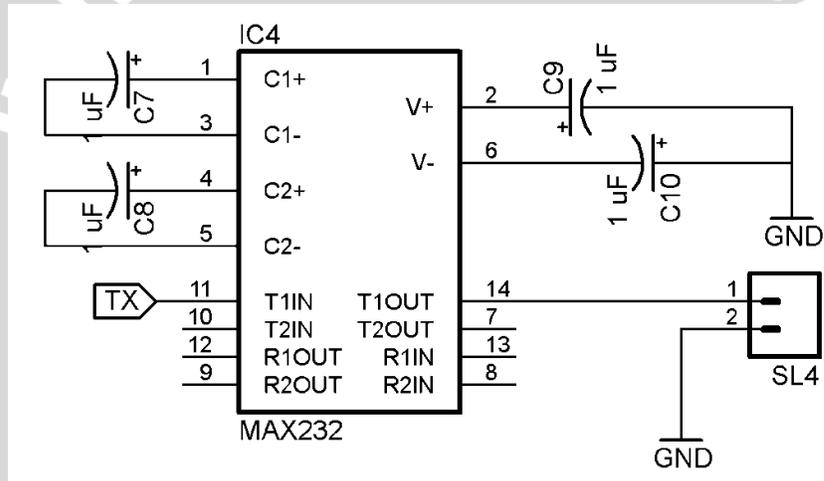
#### 4.3.2 Rangkaian Antarmuka MAX232

Data yang keluar masuk port data dari rangkaian penerima RF menggunakan level tegangan CMOS. Sedangkan keluaran atau masuk dari PC menggunakan standar komunikasi serial RS-232, level tegangan yang digunakan adalah level tegangan yang berkisar antara  $-3$  volt dan  $-15$  volt untuk kondisi logika “1” atau yang disebut dengan keadaan *mark* dan antara  $+3$  volt dan  $+15$  volt untuk kondisi logika “0” atau disebut dengan keadaan *space* atau dengan kata lain standar RS-232 menggunakan logika negatif/terbalik. Oleh karena itu dibutuhkan sebuah rangkaian untuk mengkonversi perbedaan level tegangan untuk masing-masing logika pada kedua bagian ini, dalam hal ini digunakan komponen MAX232.

IC MAX232 mempunyai 2 *receivers* yang berfungsi sebagai penyesuai level tegangan level RS-232 ke level *Transistor Transistor Logic* (TTL) ataupun CMOS, dan juga mempunyai 2 *drivers* yang berfungsi mengubah level tegangan dari level TTL/CMOS ke level RS-232. Pasangan *driver/receiver* ini berfungsi sebagai *TX* dan *RX*, sedangkan pasangan yang lainnya berfungsi sebagai *CTS* dan *RTS*.

Nilai kapasitor (C) yang digunakan telah ditentukan sebesar  $1\mu\text{F}$ , nilai ini merupakan konfigurasi *datasheet* MAX232. Gambar 4.7 menunjukkan rangkaian MAX232. Lima kapasitor dalam rangkaian ini digunakan pada:

- $C_1$  terhubung pada pin 1 (+) dan pin 3(-), sebagai pengganda tegangan (+5 volt menjadi +10 volt). keluaran hasil proses pengganda tegangan ada di pin 2.
- $C_2$  terhubung pada pin 4(+) dengan pin 5(-), sebagai pembalik tegangan (+10 volt menjadi -10 volt), keluaran ada di pin 6.
- $C_3$  terhubung pada pin 6 (-) dan dengan *Ground* (+), untuk penstabil nilai arus yang mengalir.
- $C_4$  terhubung pada pin 2(+) dengan pin 16(-), untuk penstabil nilai arus yang mengalir.
- $C_5$  terhubung pada pin 16(+) dengan *Ground*(-), untuk penstabil nilai arus yang mengalir.



Gambar 4.7 Rangkaian Penyesuai Level Tegangan (MAX232)

#### 4.4 Perancangan Komunikasi Data

Pada sisi pemrogram, terjadi proses pertukaran data dengan sisi target. Pertukaran data ditangani langsung oleh program aplikasi *downloader* pada PC di sisi pemrogram sendiri melalui rangkaian antarmuka max 232. Oleh karena itu di sisi pemrogram terdapat komunikasi data yang dalam perancangan ini diistilahkan komunikasi data radio frekuensi.

##### 4.4.1 Perancangan Protokol Komunikasi Data Radio Frekuensi

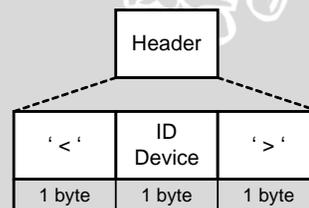
Proses pertukaran data antara sisi pemrogram dengan sisi target dapat berlangsung maka perlu dibuat sebuah protokol komunikasi data yang akan mengatur transmisi data secara serial melalui radio frekuensi. Protokol bekerja

dengan menggunakan fasilitas serial USART mikrokontroler yang berarti protokol ini bekerja satu tingkat di atas protokol USART. Protokol ini diterapkan pada perangkat lunak di kedua sisi sistem, baik di sisi pemrogram maupun di sisi target.

Berdasarkan operasinya maka pada protokol ini terdapat tiga sesi transmisi, yakni: sesi pilih mikrokontroler target, sesi tulis target dan sesi baca target. Sesi pilih mikrokontroler target adalah saat protokol menangani proses transfer data dari sisi pemrogram ke sisi target untuk operasi pilih mikrokontroler pada minimum sistem multi target. Sesi tulis target adalah saat dimana protokol menangani proses transfer data dari sisi pemrogram ke sisi target untuk operasi tulis memori. Sesi baca target adalah saat protokol menangani proses transfer data dari sisi target ke sisi pemrogram untuk operasi baca memori.

#### 4.4.1.1. Format *Frame* Paket Transmisi

Pada protokol komunikasi terdapat beberapa paket transmisi yang terdiri atas paket perintah pilih mikrokontroler, paket perintah operasi memori, paket data dan paket konfirmasi. Seluruh paket-paket tersebut tersusun atas *header*, isi dan *checksum*, kecuali untuk paket konfirmasi. *Header* berisi data yang berfungsi sebagai awalan sebuah paket. Informasi penting yang terdapat dalam *header* adalah id *device*, yang berfungsi sebagai identitas khusus bagi sistem sehingga hanya satu pasang perangkat keras sisi pemrogram dan sisi target saja yang akan saling mengenal id dan dapat melakukan komunikasi data. Protokol ini menggunakan pula satu *bytes checksum*, hal ini untuk menyiapkan pendeteksian kesalahan dalam lebar data yang panjang. *Frame header* ditunjukkan dalam Gambar 4.8.



**Gambar 4.8** Format frame header paket transmisi

Seperti yang telah dijelaskan sebelumnya, paket transmisi terdiri atas paket perintah pilih mikrokontroler, paket perintah operasi memori, paket data dan

paket konfirmasi. Format *frame* tiap-tiap paket tersebut dan fungsinya adalah sebagai berikut:

a. Paket Perintah Pilih Mikrokontroler

Paket perintah ini berisi informasi antara lain kode mikrokontroler dari sisi pemrogram ke sisi target yang diproses oleh sisi target untuk memilih mikrokontroler mana yang akan dilakukan proses operasi memori pada salah satu mikrokontroler target. Format *frame* paket perintah ditunjukkan dalam tabel 4.1 dan untuk id mikrokontroler ditunjukkan dalam tabel 4.2.

**Tabel 4.1** Format *frame* perintah pilih mikrokontroler

3 byte	1 byte	1 byte	1 byte
Header	ID mikrokontroler	0x7C	checksum

**Tabel 4.2** ID mikrokontroler dari sisi pemrogram

No	Nama mikrokontroler	ID mikrokontroler (heksadesimal)	Panjang Data (byte)
1	Mikrokontroler 1	0x01	1
2	Mikrokontroler 2	0x02	1
3	Mikrokontroler 3	0x03	1
4	Mikrokontroler 4	0x04	1

b. Paket Perintah operasi memori

Paket perintah dikirim ke bagian yang akan menerima data sebagai tanda/persiapan untuk proses transmisi data. Paket perintah berisi informasi penting antara lain id perintah, jumlah paket data yang akan ditransmisikan dan *device signature* mikrokontroler target. Untuk sesi tulis target maka sisi pemrogram akan mengirimkan paket perintah yang berisikan data id perintah, *device signature* dan jumlah paket data. Sedangkan untuk sesi baca target maka paket perintah berisi id perintah dan *device signature*. Khusus untuk sesi baca target maka sisi target juga akan mengirimkan paket perintah untuk mengawali proses transmisi data pembacaan memori mikrokontroler target. Format *frame* paket perintah ditunjukkan dalam Tabel 4.3, Tabel 4.4 dan Tabel 4.5.

**Tabel 4.3** Format *frame* paket perintah dari sisi pemrogram saat sesi tulis target

3 byte	1 byte	1 byte	2 byte	1 byte	3 byte	1 byte	1 byte
header	ID perintah	0x7C	Jumlah paket	0x7C	Device signature	0x7C	checksum

**Tabel 4.4** Format *frame* paket perintah dari sisi pemrogram saat sesi baca target

3 byte	1 byte	1 byte	2 byte	1 byte	3 byte	1 byte	1 byte
header	ID perintah	0x7C	N/A	0x7C	Device signature	0x7C	checksum

**Tabel 4.5** Format *frame* paket perintah dari sisi target saat sesi baca target

3 byte	1 byte	1 byte	2 byte	1 byte	1 byte	1 byte	1 byte
Header	ID perintah	0x7C	Jumlah paket	0x7C	ID mikrokontroler	0x7C	checksum

c. Paket Data

Paket data terdiri atas dua jenis, yaitu paket data dari sisi target dan paket data dari sisi pemrogram. Paket data dari sisi target berisi data memori program mikrokontroler target yang dikirimkan secara langsung, sehingga dalam pengirimannya akan menyimpan keseluruhan *bytes* data dari mikrokontroler target ke sisi pemrogram. Sedangkan paket data dari sisi pemrogram berisi data memori program mikrokontroler yang dikirimkan secara langsung, sehingga dalam pengirimannya akan mengirim keseluruhan *bytes* data ke mikrokontroler target. Dalam paket data, terdapat informasi id perintah dan data memori mikrokontroler. Format *frame* paket data ditunjukkan dalam Tabel 4.6 dan Tabel 4.7.

**Tabel 4.6** Format *frame* paket data dari sisi pemrogram

3 byte	n byte	1 byte	1 byte
header	Data Kiriman	0x7C	checksum

**Tabel 4.7** Format *frame* paket data dari sisi target

3 byte	1 byte	1 byte	n byte	1 byte	1 byte
header	ID perintah	0x7C	Data Kiriman	0x7C	checksum

d. Paket Konfirmasi

Paket konfirmasi ditransmisikan oleh sisi penerima transmisi data. Seluruh paket yang dikirimkan mulai dari paket perintah dan paket data diberikan balasan berupa sebuah paket konfirmasi. Paket konfirmasi terbagi menjadi dua jenis, yaitu paket ok, paket gagal. Format *frame* paket konfirmasi ditunjukkan dalam Tabel 4.8. Beberapa informasi yang dirancang dalam *frame-frame* paket transmisi antara lain untuk menunjukkan jenis operasi memori (id perintah), id *device*, jumlah paket dan id paket ditunjukkan dalam Tabel 4.9 dan Tabel 4.10.

**Tabel 4.8** Format *frame* paket konfirmasi

Paket OK	0x7C	0xAA	0x7C	0xAA
Paket gagal	0x7C	0x55	0x7C	0x55

**Tabel 4.9** ID Perintah sesuai jenis operasi memori mikrokontroler target

No	Jenis Operasi Memori	ID Perintah (heksadesimal)	Panjang Data (byte)
1	Tulis memori <i>flash</i>	0x10	1
2	Baca memori <i>flash</i>	0x20	1
3	Tulis memori EEPROM	0x30	1
4	Baca memori EEPROM	0x40	1
5	Tulis <i>fuse bits</i>	0x50	1
6	Baca <i>fuse bits</i>	0x60	1
7	Hapus memori	0x70	1

**Tabel 4.10** Informasi ukuran data dan id *device*

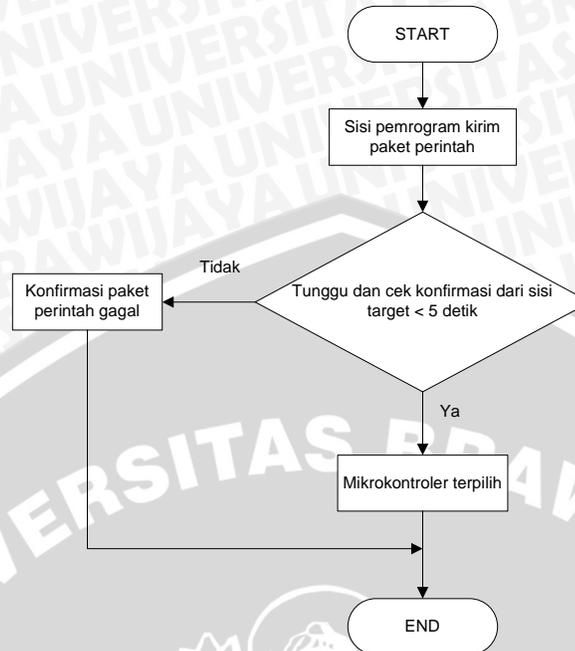
No	Informasi	Kode Instruksi (heksadesimal)	Panjang Data (byte)
1	ID Device Sisi Pemrogram	0x81	1
2	ID Device Sisi Target	0x82	1

#### 4.4.1.2. Skenario Protokol Komunikasi

Untuk mempermudah analisis protokol komunikasi, maka perlu dibuat sebuah diagram skenario protokol komunikasi. Skenario protokol dibedakan berdasarkan sesi operasinya, yakni sesi pilih mikrokontroler target, sesi tulis target dan sesi baca target.

##### a. Skenario protokol untuk sesi pilih mikrokontroler target

Sesi pilih mikrokontroler target diawali oleh pengiriman paket perintah pilih mikrokontroler oleh sisi pemrogram. Sisi target memberikan balasan berupa paket konfirmasi ok ketika paket perintah berhasil diterima dan diolah maka mikrokontroler target telah terpilih. Skenario protokol untuk sesi pilih mikrokontroler target ditunjukkan dalam Gambar 4.9.



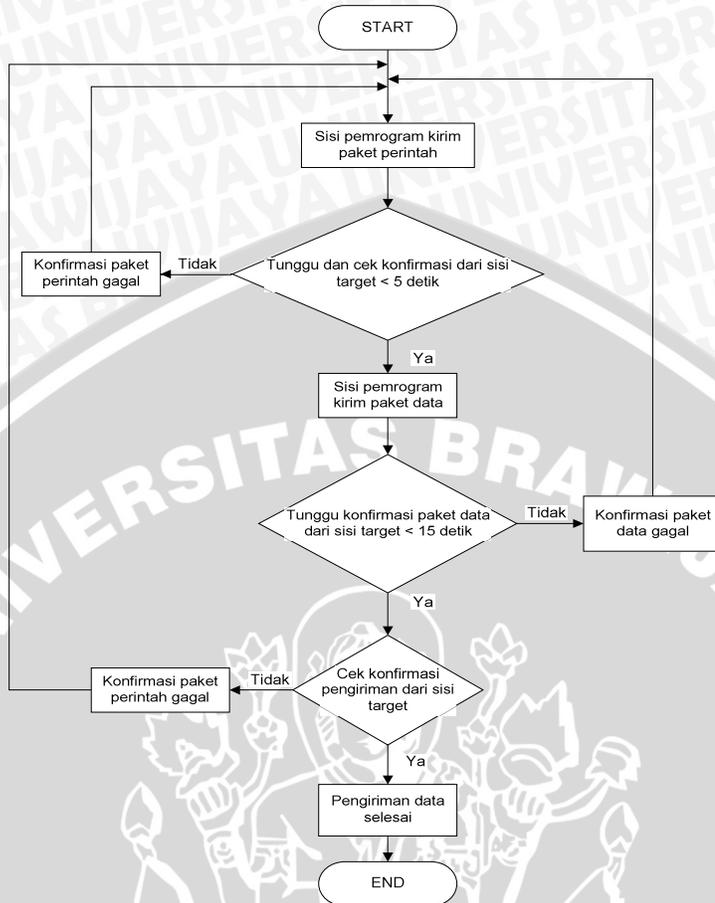
**Gambar 4.9** Skenario protokol sesi pilih mikrokontroler target

#### b. Skenario protokol untuk sesi tulis memori target

Sesi tulis memori target diawali oleh pengiriman paket perintah tulis target oleh sisi pemrogram. Sisi target memberikan balasan berupa paket konfirmasi ok ketika paket perintah berhasil diterima dan diolah. Skenario protokol untuk sesi tulis memori target ditunjukkan dalam Gambar 4.10.

Sisi pemrogram melanjutkan proses berikutnya yaitu pengiriman paket data dan sisi target akan membalas dengan mengirimkan paket konfirmasi berhasil ataukah gagal. Apabila paket data berhasil diterima dengan baik tanpa terdapat kesalahan data *checksum* maka sisi target akan mengirimkan konfirmasi paket ok ke sisi pemrogram. Sedangkan bila terjadi kesalahan dalam penerimaan maka sisi target akan mengirimkan konfirmasi paket gagal dan sisi pemrogram harus mengulang pengiriman paket data yang gagal tersebut. Proses di atas diulang hingga seluruh data memori yang akan dituliskan ke mikrokontroler target telah diterima dengan baik oleh sisi target.

Apabila telah selesai maka sisi target mengirimkan balasan konfirmasi paket ok ke sisi pemrogram. Dengan diterimanya konfirmasi paket ok oleh sisi pemrogram maka sesi tulis memori target telah selesai.



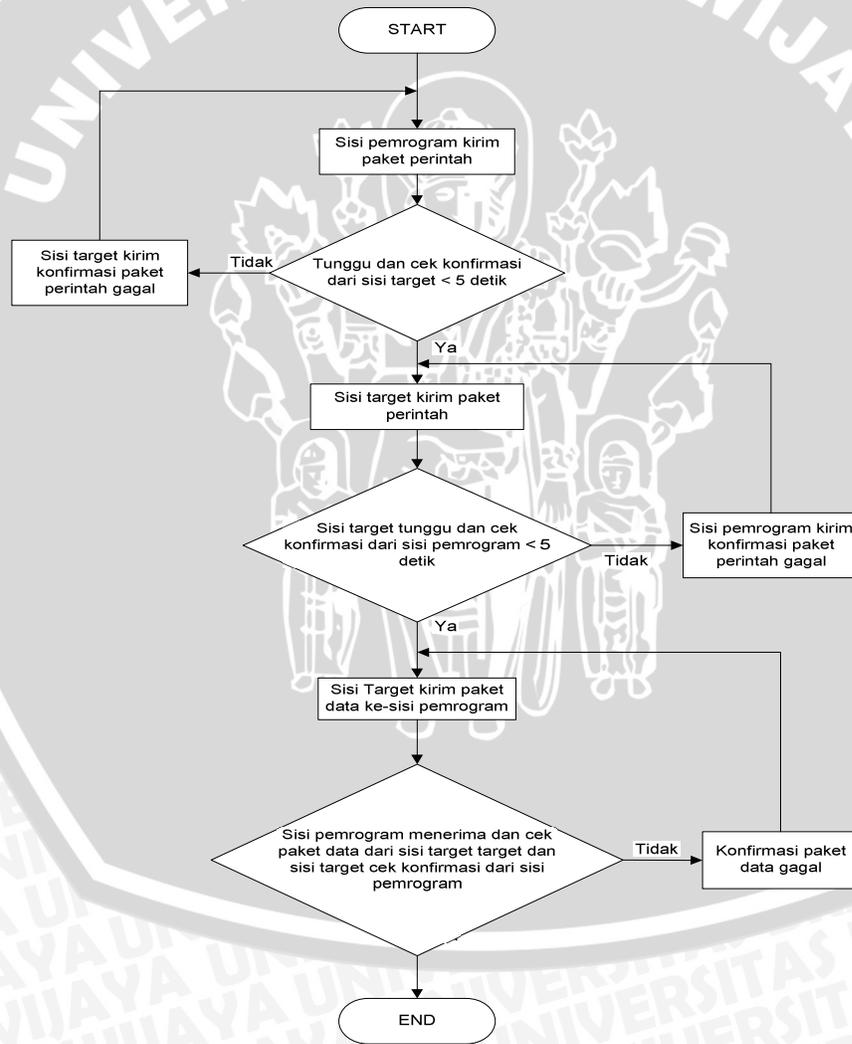
**Gambar 4.10** Skenario protokol sesi tulis memori target

b. Skenario protokol untuk sesi baca memori target

Sesi baca target diawali oleh pengiriman paket perintah baca target oleh sisi pemrogram. Sisi target memberikan balasan berupa konfirmasi paketok ketika paket perintah berhasil diterima dan diolah. Setelah sisi pemrogram menerima balasan konfirmasi paket ok dari sisi target, sisi pemrogram akan berganti menunggu pengiriman paket perintah dari sisi target. Skenario protokol untuk sesi baca memori target ditunjukkan dalam Gambar 4.11.

Sisi target melanjutkan proses berikutnya yaitu pengiriman paket perintah ke sisi pemrogram yang berisi jumlah paket yang akan dikirimkan. Sisi pemrogram akan memberikan balasan berupa konfirmasi paket ok. Sisi target akan melanjutkan dengan pengiriman paket data dan sisi pemrogram akan membalas dengan mengirimkan paket konfirmasi sesuai dengan kondisi paket yang diterimanya, berhasil ataukah gagal.

Apabila paket data berhasil diterima dengan baik tanpa terdapat kesalahan data *checksum* maka sisi pemrogram akan mengirimkan konfirmasi paket ok ke sisi target. Sedangkan bila terjadi kesalahan dalam penerimaan maka sisi pemrogram akan mengirimkan konfirmasi paket gagal dan sisi target harus mengulang pengiriman paket data kembali. Proses di atas diulangi hingga seluruh data memori yang dibaca dari mikrokontroler target telah diterima dengan baik oleh sisi pemrogram. Sisi pemrogram akan menampilkan hasil pembacaan memori mikrokontroler target di perangkat lunak sisi pemrogram. Dengan diterimanya konfirmasi paket ok oleh sisi target maka sesi baca memori target telah selesai.

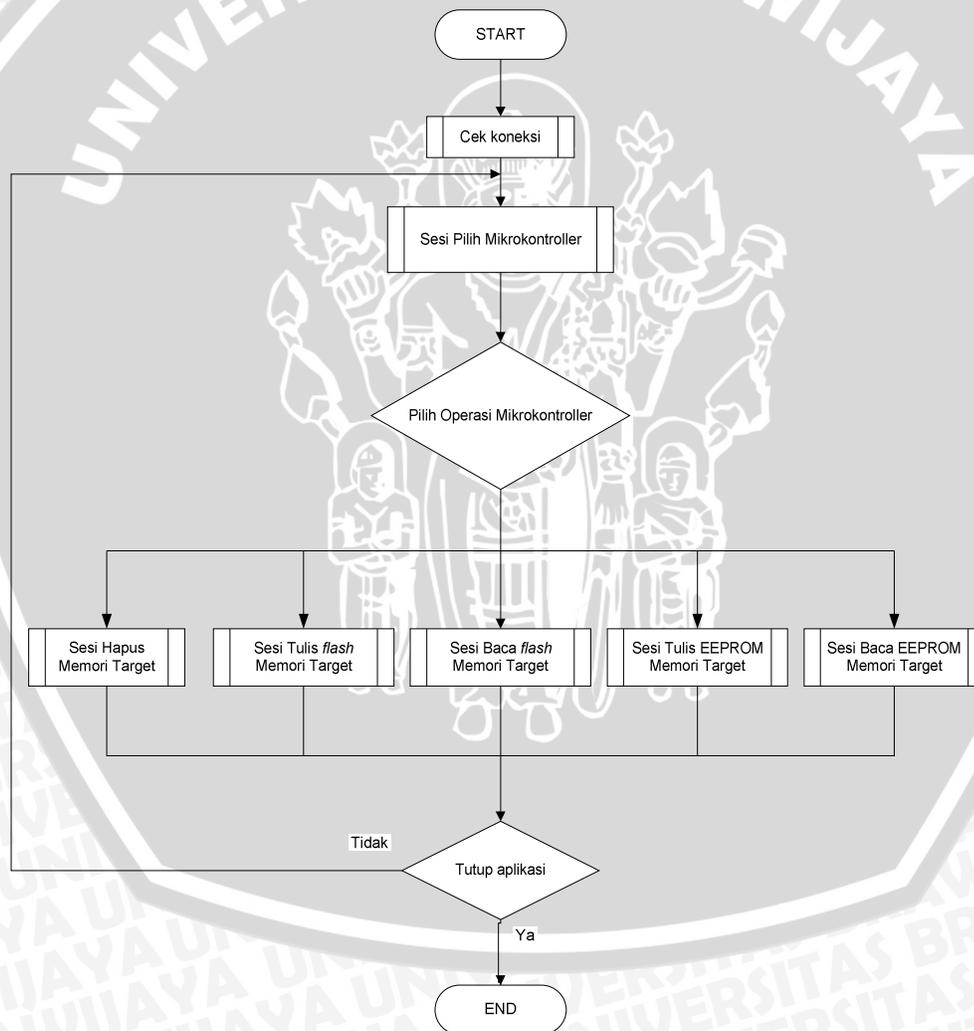


Gambar 4.11 Skenario protokol sesi baca memori target

#### 4.5 Perancangan dan Pembuatan Perangkat Lunak

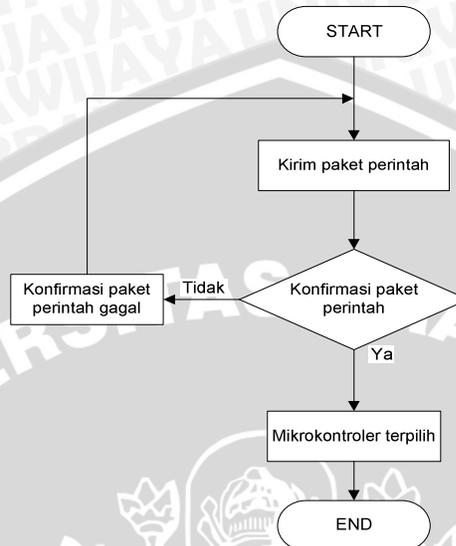
Dalam perancangan perangkat lunak PC menggunakan metode pendekatan struktural. Perancangan dilakukan dengan pembuatan diagram alir (*flowchart*), diikuti beberapa diagram bantu serta penjelasan untuk membantu memperjelas konsep perancangan.

Proses yang dirancang pada perangkat lunak PC selalu diawali oleh sesi pilih mikrokontroler. Kemudian diikuti oleh penanganan transmisi berdasarkan sesi-sesi yang diinginkan. Proses Utama dari perancangan perangkat lunak ditunjukkan dalam Gambar 4.12.



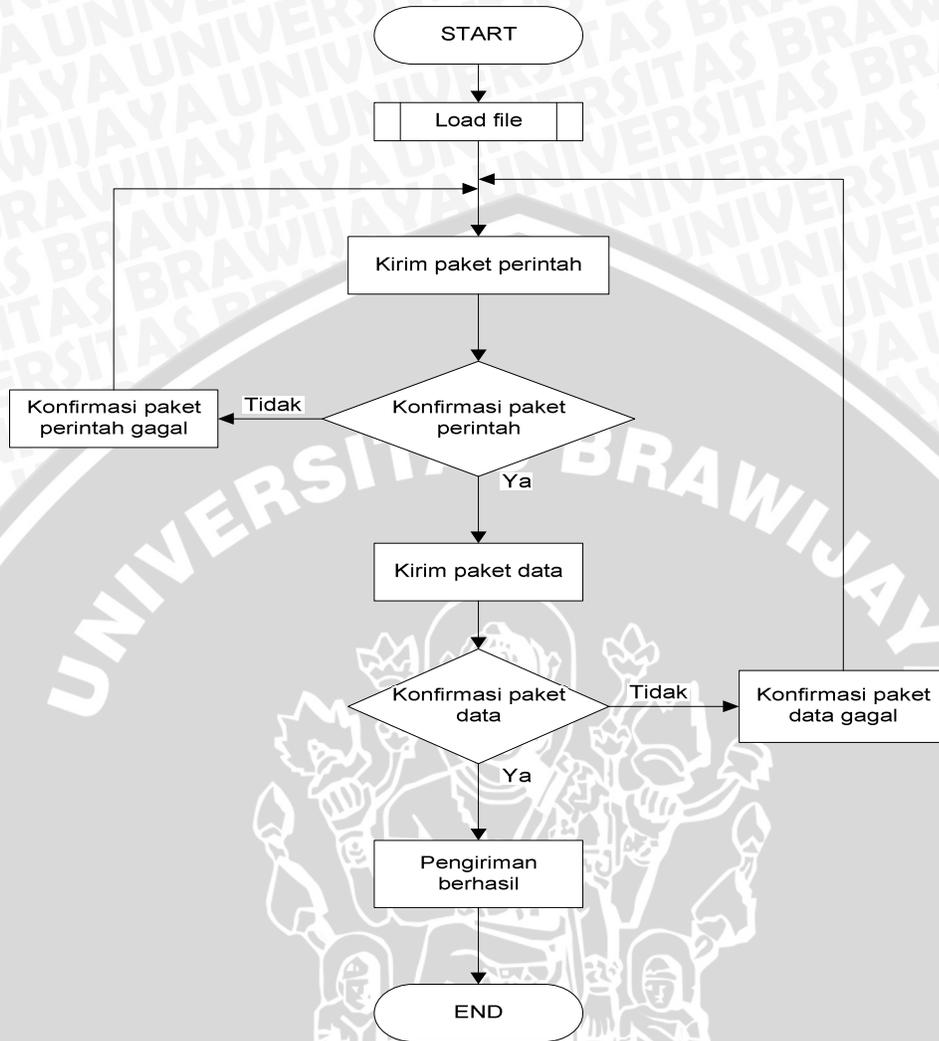
**Gambar 4.12** Diagram alir utama aplikasi perangkat lunak PC sisi pemrogram

Dalam diagram alir yang dirancang sesuai dalam perancangan komunikasi data, proses transmisi dibagi berdasarkan sesi-sesi. Diagram alir sesi pilih mikrokontroler ditunjukkan dalam Gambar 4.13.



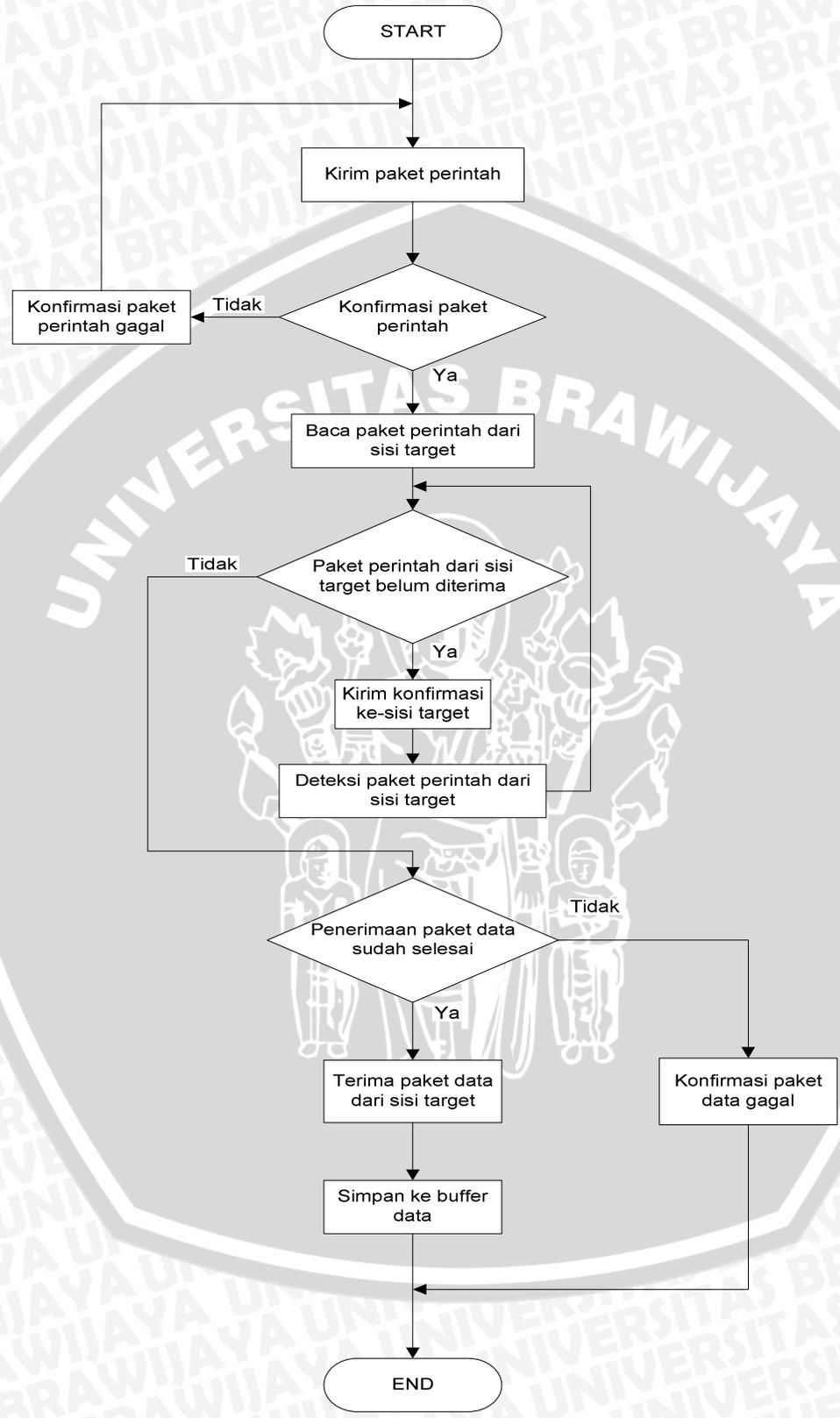
**Gambar 4.13** Diagram alir sesi pemrogram pilih mikrokontroler

Diagram alir transmisi sesi tulis target dapat dilihat dalam Gambar 4.10. Sesi tulis target terdiri dari dua proses, yakni: proses pengiriman paket perintah dan proses pengiriman data. Aliran data berasal dari sisi pemrogram menuju ke sisi target. Detail proses dalam diagram alir transmisi sesi tulis target ditunjukkan dalam Gambar 4.14.



**Gambar 4.14** Diagram alir sisi pemrograman transmisi sesi tulis *flash*/EEPROM target

Diagram alir transmisi sesi baca target dapat dilihat dalam Gambar 4.11. Sesi baca target terdiri dari dua proses, yakni: proses pengiriman paket perintah dan proses penerimaan paket perintah dan paket data. Detail proses dalam diagram alir transmisi sesi baca target ditunjukkan dalam Gambar 4.15.



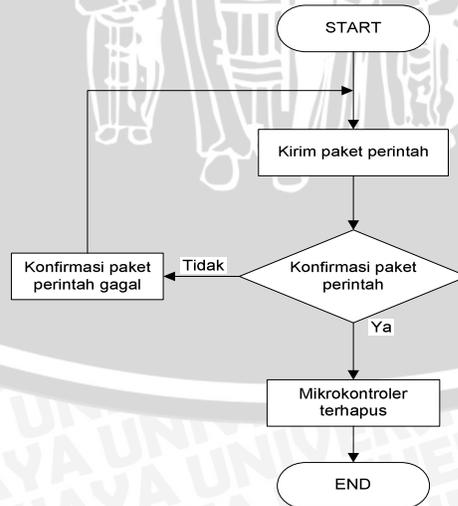
Gambar 4.15 Diagram alir sisi pemrogram transmisi sesi baca flash/EEPROM target

Dalam cek koneksi terdiri atas setting port dimana dalam proses ini memasukkan setting port yang diinginkan baru kemudian melakukan cek port tersebut digunakan atau tidak. Detail proses dalam diagram alir transmisi sesi cek koneksi ditunjukkan dalam Gambar 4.16.



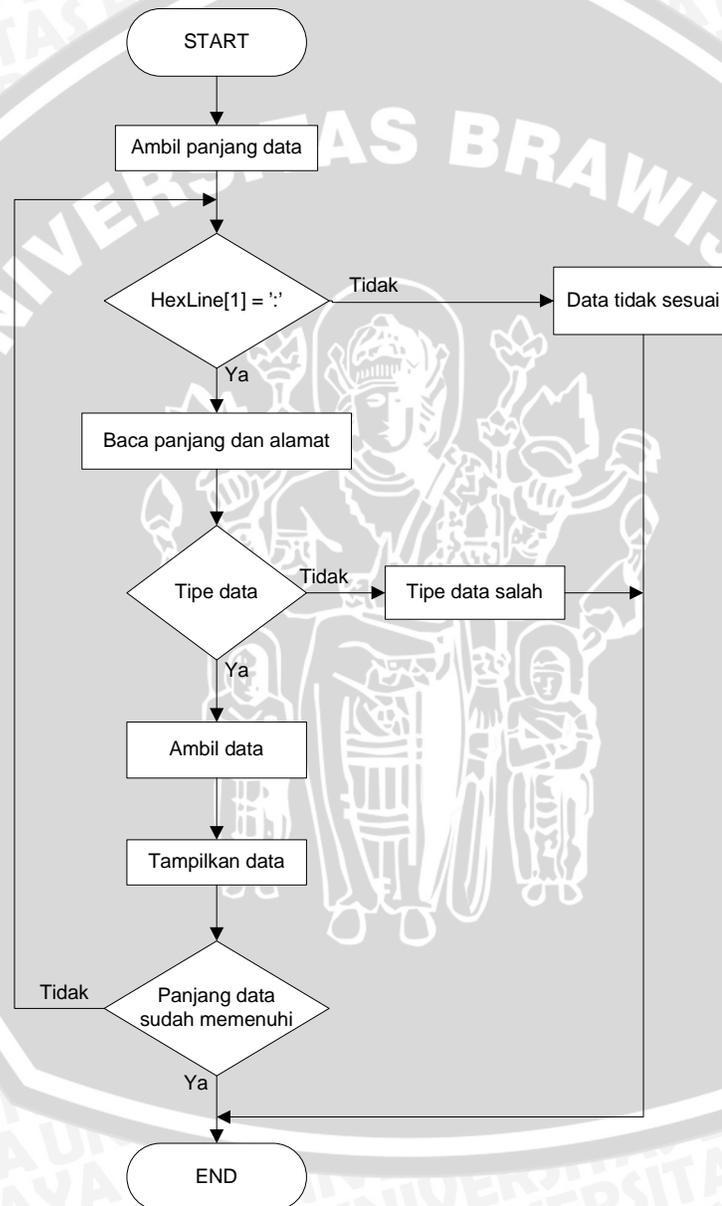
**Gambar 4.16** Diagram alir sisi pemrogram cek koneksi

Proses hapus target terdiri atas proses pengiriman paket perintah hapus mikrokontroler. Detail proses dalam diagram alir transmisi sesi hapus mikrokontroler target ditunjukkan dalam Gambar 4.17.



**Gambar 4.17** Diagram alir sisi pemrogram hapus mikrokontroler

Dalam proses *load file* terdiri atas proses pengambilan panjang data dari *file* yang digunakan. Kemudian dari setiap baris dari panjang data itu diseleksi dan diambil data yang diinginkan yaitu *file hex* dan *file eeprom*. Detail proses dalam diagram alir *load file* ditunjukkan dalam Gambar 4.18.



Gambar 4.18 Diagram alir sisi pemrogram *load file*

## BAB V

### PENGUJIAN DAN ANALISIS

Pengujian dan analisis dilakukan untuk mengetahui tingkat kesesuaian hasil perancangan dan pembuatan alat dengan hasil yang diharapkan. Pengujian dilakukan tiap blok perancangan untuk memudahkan analisis data. Setelah seluruh blok perancangan diuji, maka dilakukan pengujian sistem secara keseluruhan.

Pengujian dan analisis dilakukan pada tiap-tiap blok maupun alat secara keseluruhan yang meliputi:

1. Pengujian Pemancar dan Penerima RF
2. Pengujian Rangkaian MAX 232
3. Pengujian perangkat lunak
4. Pengujian sistem secara keseluruhan

#### 5.1 Pengujian Pemancar dan Penerima RF

##### 5.1.1 Tujuan

Pengujian ini bertujuan untuk mengetahui apakah rangkaian pemancar dan penerima RF dapat menerima dan mengirimkan data dengan baik. Dan bagaimana cara pengaturan *baudrate* dan *channel* pada modul RF. Diagram blok pengujian pengaturan *baudrate* dan *channel* ditunjukkan dalam Gambar 5.1. Pengujian berikutnya dilakukan dengan cara mengirimkan data secara serial ke komputer menggunakan *hyperterminal* program *Code Vision AVR* pada PC sisi pemrogram dan menerima data-data tersebut menggunakan *hyperterminal* program *Code Vision AVR* pada PC sisi target dan sebaliknya. Dalam proses pengiriman data berupa string yang terdiri atas karakter-karakter huruf dan angka. Kemudian *hyperterminal* akan menampilkan data yang dikirim oleh PC. Diagram blok pengujian penerimaan data secara serial ini ditunjukkan dalam Gambar 5.3.

##### 5.1.2 Peralatan Pengujian

- 1) Rangkaian Pemancar RF YS-1020
- 2) Rangkaian Penerima RF YS-1020
- 3) USB to Serial
- 4) Rangkaian MAX 232

## 5) PC

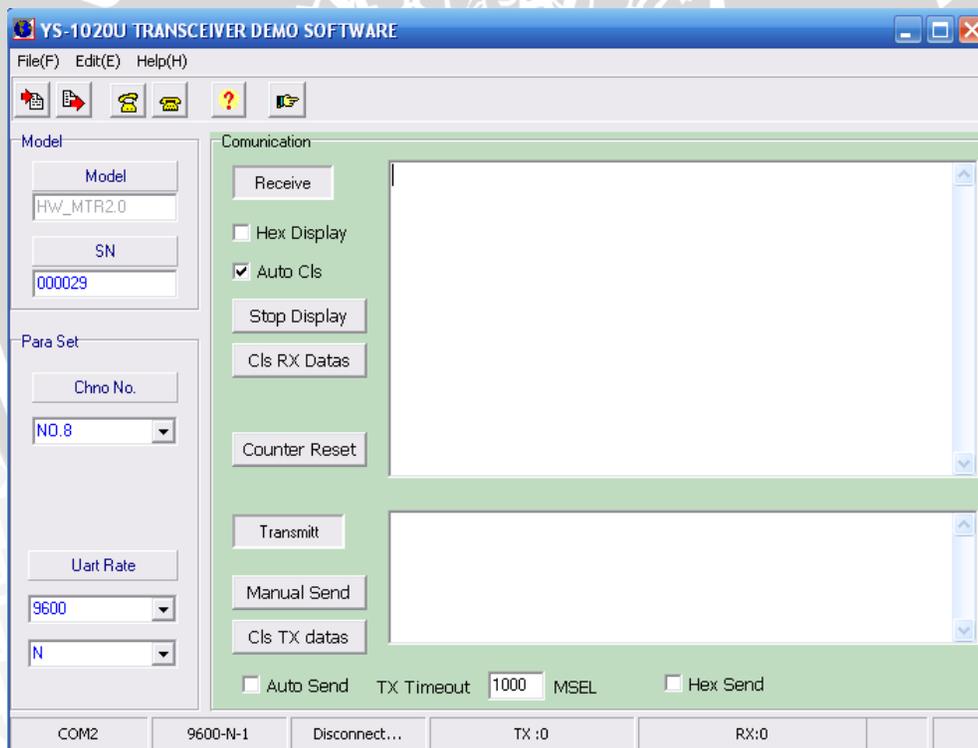
**5.1.3 Proses Pengujian**

Pengujian dilakukan dengan cara menghubungkan peralatan-peralatan pengujian seperti yang ditunjukkan dalam Gambar 5.1 dan Gambar 5.3.

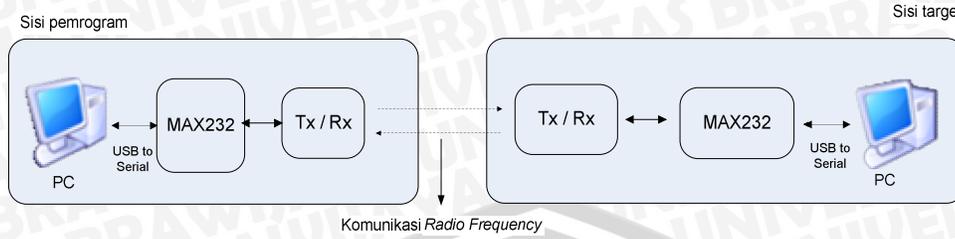


**Gambar 5.1** Diagram blok pengujian pengaturan *baudrate* dan *channel* modul RF

Untuk program pengaturan *baudrate* dan *channel* ditunjukkan dalam Gambar 5.2.



**Gambar 5.2** Tampilan program untuk pengaturan baudrate dan setting modul RF



**Gambar 5.3** Diagram blok pengujian software penerimaan data secara serial

### 5.1.4 Hasil dan Analisis

Hasil pengujian pengaturan *baudrate* dan *channel* antara rangkaian pemancar dan penerima modul radio frekuensi ini ditunjukkan dalam Tabel 5.1 berikut.

**Tabel 5.1** Hasil pengujian pengaturan *baudrate* dan *setting* RF YS-1020

Kondisi	Hasil
Baudrate dan channel sama	Pengiriman sukses
Baudrate beda dan channel sama	Pengiriman gagal
Baudrate sama dan channel beda	Pengiriman gagal
Baudrate beda dan channel beda	Pengiriman gagal

Berdasarkan hasil pengujian dalam Tabel 5.1 dapat dianalisis bahwa pengaturan *baudrate* dan *channel* dari kedua modul RF YS-1020 sisi pemrogram dan sisi target harus sama diantara dua modul RF YS-1020 agar proses pengiriman data dapat berhasil dengan lancar.

Hasil dari pengujian perangkat lunak penerimaan dan pengiriman data secara serial ditunjukkan dalam Gambar 5.4.

```

CodeVisionAVR - [Terminal]
File Edit View Project Tools Settings Windows Help
Connect
CodeVisio
No Pri
Other
ABCDEF GHIJKLMN OPQRST UVWXYZ abcdefghijklmnopqrstuvwxyz01234567890
567890 ABCDEF GHIJKLMN OPQRST UVWXYZ abcdefghijklmnopqrstuvwxyz01234
opqrstuvwxyz01234567890 ABCDEF GHIJKLMN OPQRST UV
XYZ abcdefghijklmnopqrstuvwxyz01234567890 ABCDEF
GHIJKLMN OPQRST UVWXYZ abcdefghijklmnopqrstuvwxyz01234567890
ABCDEF GHIJKLMN OPQRST UVWXYZ abcdefghijklmnopqrstuvwxyz01234567890
01234567890 ABCDEF GHIJKLMN OPQRST UVWXYZ abcdefghijklmnopqrstuvwxyz
jklmnopqrstuvwxyz01234567890 ABCDEF GHIJKLMN OPQR
STUVWXYZ abcdefghijklmnopqrstuvwxyz01234567890 A
BCDEF GHIJKLMN OPQRST UVWXYZ abcdefghijklmnopqrstuvwxyz01234567890
ABCDEF GHIJKLMN OPQRST UVWXYZ abcdefghijklmnopqrstu
vwxyz01234567890 ABCDEF GHIJKLMN OPQRST UVWXYZ abcd
efghijklmnopqrstuvwxyz01234567890 ABCDEF GHIJKLM
NOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz01234567890
ABCDEF GHIJKLMN OPQRST UVWXYZ abcdefghijklmnopqrstuvwxyz01234567890
7890 ABCDEF GHIJKLMN OPQRST UVWXYZ abcdefghijklmnopqrstuvwxyz0123456
qrstuvwxyz01234567890 ABCDEF GHIJKLMN OPQRST UVWXYZ
Z abcdefghijklmnopqrstuvwxyz01234567890 ABCDEF GHIJKLMN OPQRST UVWXYZ
IJKLMN OPQRST UVWXYZ abcdefghijklmnopqrstuvwxyz01234567890
ABCDEF GHIJKLMN OPQRST UVWXYZ abcdefghijklmnopqrstuvwxyz01234567890
ABCDEF GHIJKLMN OPQRST UVWXYZ a

```

**Gambar 5.4** Hasil dari pengujian software penerimaan dan pengiriman data secara serial

Pada tampilan program *Hyperterminal Code Vision AVR* tampak bahwa pengujian komunikasi data dengan mengirimkan karakter dapat berhasil dengan baik dan tidak ada data yang hilang. Karakter yang dikirim oleh rangkaian pemancar RF dapat diterima dengan baik oleh rangkaian penerima RF.

## 5.2 Pengujian Rangkaian MAX232

### 5.2.1 Tujuan

Pengujian ini bertujuan untuk memeriksa apakah rangkaian MAX232 dapat bekerja dengan baik untuk mengkonversi sinyal dari mikrokontroler yang bertipe CMOS menjadi sinyal dengan tipe RS232 yang dapat diterima oleh komputer.

### 5.2.2 Peralatan Pengujian

Peralatan yang digunakan dalam pengujian adalah rangkaian MAX232, multimeter digital dan catu daya +5V dan 0V.

### 5.2.3 Proses Pengujian

Pengujian ini dilakukan dengan cara memberikan tegangan masukan sebesar 0V dan 5V secara bergantian dan mengukur tegangan keluaran rangkaian MAX232. Diagram blok pengujian MAX232 ditunjukkan dalam Gambar 5.5



Gambar 5.5 Diagram blok pengujian MAX232

## 5.2.4 Hasil dan Analisis

Hasil dari pengujian rangkaian MAX232 dapat dilihat dalam tabel 5.2.

Tabel 5.2 Hasil pengujian rangkaian MAX232

Tegangan Masukan MAX232	Tegangan Keluaran MAX232
0 volt	9,11 volt
4,9 volt	-9,20 volt

Berdasarkan hasil pengujian dalam Tabel 5.1 dapat dianalisis bahwa tegangan yang dihasilkan dengan masukan 4,9V adalah -9,20V. Nilai tegangan ini sudah sesuai dengan level logika "high" pada RS232. Sedangkan pada masukan 0V tegangan yang dihasilkan adalah 9,11V. Nilai tegangan ini sudah sesuai dengan level logika "low" pada RS232.

## 5.3 Pengujian Perangkat Lunak

### 5.3.1 Tujuan

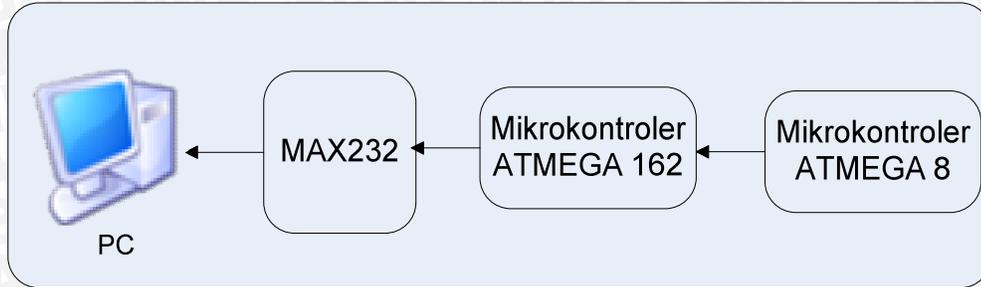
Pengujian ini bertujuan untuk mengetahui apakah perangkat lunak yang dirancang sudah dapat memisah paket data yang dirancang dengan baik.

### 5.3.2 Peralatan Pengujian

Peralatan yang digunakan dalam pengujian adalah mikrokontroler, MAX232 dan komputer.

### 5.3.3 Proses Pengujian

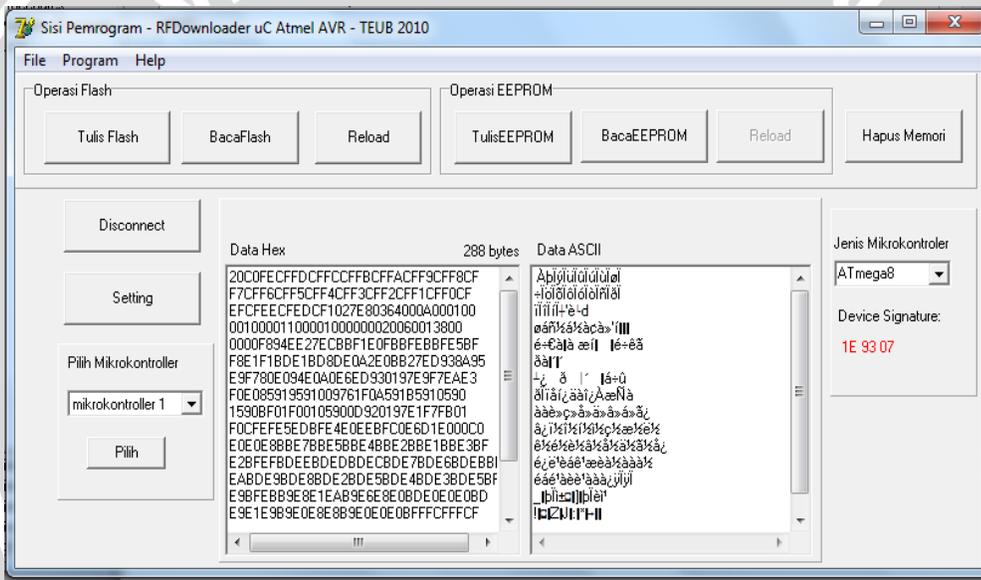
Pengujian ini dilakukan dengan cara mikrokontroler diprogram untuk mengirimkan data sesuai paket data yang dirancang. Data yang dikirim akan dibaca oleh perangkat lunak yang telah dirancang. Kemudian pada tampilan software dilihat apakah data yang ditampilkan sudah sesuai yang diinginkan yaitu memisah paket data yang dikirim oleh mikrokontroler. Gambar 5.6 menunjukkan diagram blok pengujian perangkat lunak pemisahan data.



Gambar 5.6 Diagram blok pengujian perangkat lunak pisah paket data

### 5.3.4 Hasil dan Analisis

Hasil dari pengujian perangkat lunak delphi pemisahan paket data ditunjukkan dalam Gambar 5.7.



Gambar 5.7 Hasil dari pengujian perangkat lunak pisah paket data dan pengecekan error

Dalam pengujian ini perangkat lunak Delphi menerima sebuah paket data yang dikirim mikrokontroler secara serial. Pada Gambar 5.7 menunjukkan bahwa paket data telah dapat dipisah dan hasil data yang diterima ditampilkan.

### 5.4 Pengujian Keseluruhan

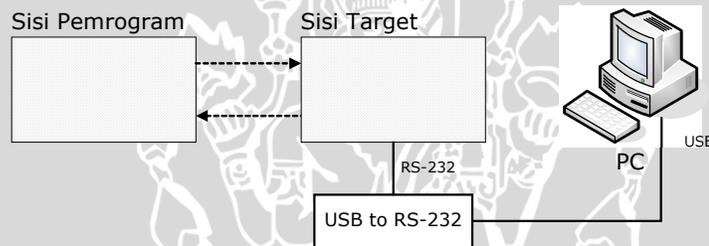
Pengujian keseluruhan sistem dilakukan untuk mengetahui kesesuaian sistem dengan spesifikasi yang diinginkan. Urutan pengujian sistem keseluruhan adalah sebagai berikut:

### 5.4.1 Pengujian Pengiriman Instruksi

Pengujian keseluruhan sistem dilakukan untuk mengetahui kesesuaian sistem dengan spesifikasi yang diinginkan. Urutan pengujian sistem keseluruhan adalah sebagai berikut:

#### 5.4.1.1 Pengujian Pengiriman dari Sisi Pemrogram ke Sisi Target

Pengujian pengiriman instruksi dan data memori ke sisi target dilakukan dengan menggunakan perangkat *USB to RS-232 converter*. Seluruh data instruksi yang terdapat pada paket perintah, data memori pada paket data serta paket akhir yang diterima oleh sisi target dikirim ke *Hyperterminal* PC. Hal ini bertujuan untuk merekam setiap data yang diterima sisi target kemudian disimpan dan dianalisis. Prosedur pengujian yang dilakukan adalah dengan menyusun sistem seperti ditunjukkan dalam Gambar 5.8.

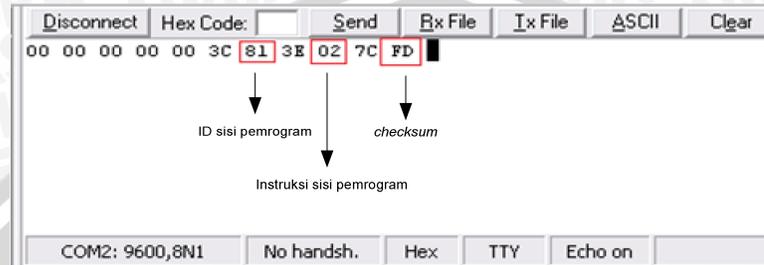


Gambar 5.8 Blok pengujian sistem secara keseluruhan

Operasi berdasarkan instruksi sisi pemrogram meliputi operasi pilih mikrokontroler dan seluruh operasi *serial downloading* untuk proses tulis baca memori *flash* maupun EEPROM, operasi tulis baca *Fuse bits* serta operasi hapus chip memori. Hasil pengujian penerimaan perintah dan data dari sisi pemrogram yang ditampilkan pada layar monitor komputer adalah pengujian operasi pilih mikrokontroler, pengujian operasi baca memori *flash*, pengujian operasi tulis memori *flash* serta pengujian operasi hapus memori chip. Keempat hasil pengujian tersebut cukup mewakili dari seluruh operasi yang ada. Sedangkan untuk operasi memori yang lain dapat diwakili oleh operasi memori baca tulis *flash* dan operasi hapus memori karena operasi memori yang lain memiliki format data yang sama dan hanya dibedakan oleh instruksi operasi memori tersebut.

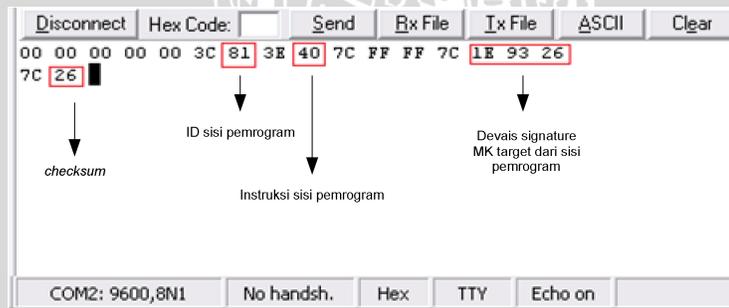
Hasil pengujian terhadap penerimaan instruksi berupa operasi pilih mikrokontroler ditunjukkan dalam Gambar 5.9. Format penerimaan data sesuai

dengan protokol hasil perancangan. Mikrokontroler target yang digunakan pada pengujian ini adalah mikrokontroler kedua yaitu mikrokontroler ATmega8535. Format penerimaan data terdiri atas *id device* sisi pemrogram yaitu 0x81, *id device* mikrokontroler kedua yaitu 0x02, serta checksum.



**Gambar 5.9** Hasil pengujian operasi pilih mikrokontroler

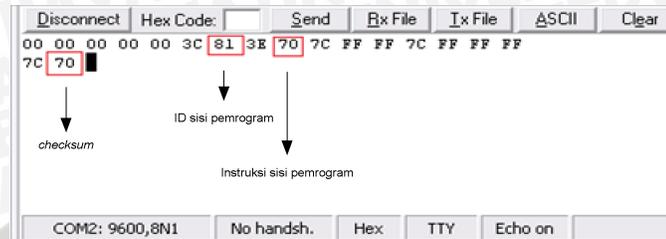
Hasil pengujian terhadap penerimaan instruksi berupa operasi baca memori EEPROM ditunjukkan dalam Gambar 5.10. Operasi baca memori EEPROM mewakili operasi memori sejenisnya yaitu operasi baca *flash* serta baca *Fuse bits*. Format penerimaan data sesuai dengan protokol hasil perancangan. Mikrokontroler target yang digunakan pada pengujian ini adalah mikrokontroler ATmega8535. Format penerimaan data terdiri atas *id device* sisi pemrogram yaitu 0x81, instruksi sisi pemrogram untuk operasi baca memori EEPROM yaitu 0x40, *3 bytes device signature* ATmega8535 yaitu 0x1E 0x93 0x08, dan checksum.



**Gambar 5.10** Hasil pengujian operasi baca memori EEPROM

Hasil pengujian terhadap penerimaan instruksi berupa operasi hapus memori ditunjukkan dalam Gambar 5.11. Pada dasarnya format penerimaan data untuk operasi hapus memori sama dengan format penerimaan data pada operasi baca data memori. Perbedaannya terletak pada tidak adanya *device signature* pada format operasi hapus memori karena informasi ini tidak diperlukan di dalam

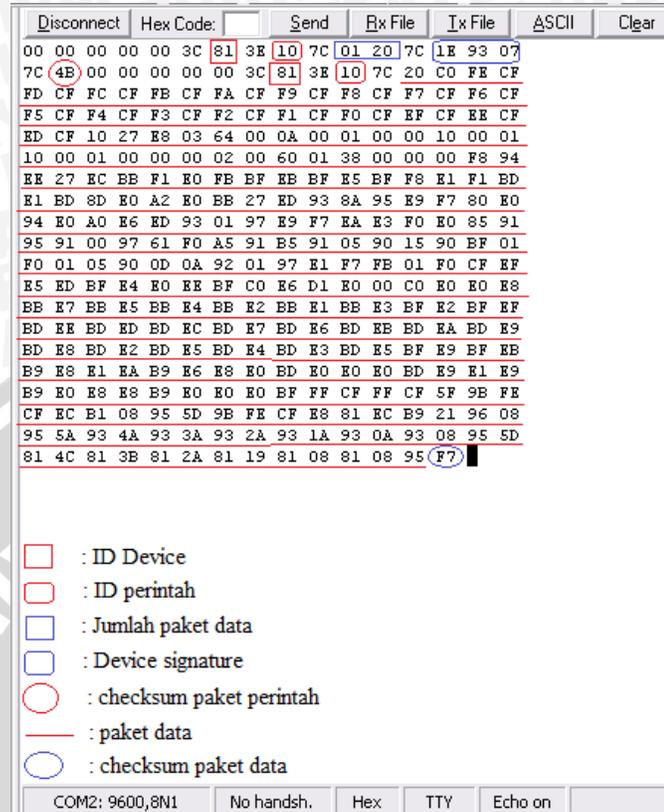
operasi ini. Data yang seharusnya berisi *device signature* diisi dengan sembarang nilai data oleh sisi pemrogram dan diabaikan pada sisi target.



**Gambar 5.11** Hasil pengujian operasi hapus memori

Format penerimaan data untuk operasi tulis memori mikrokontroler target terdiri atas paket perintah dan paket data. Hasil pengujian terhadap penerimaan instruksi berupa operasi tulis memori *flash* ditunjukkan dalam Gambar 5.12. Operasi tulis memori *flash* mewakili operasi memori sejenisnya yaitu operasi tulis EEPROM serta tulis *Fuse bits*. Format penerimaan data pada paket perintah sama dengan format paket perintah pada operasi hapus memori yaitudua *byte* berisi jumlah paket data yang dikirimkan oleh sisi pemrogram dan tiga *byte* berisi informasi *device signature*.

Pengujian ini menggunakan *file* memori *flash* berukuran 288 *bytes*. Dari hasil pengujian, diketahui bahwa penerimaan instruksi dan data memori dari sisi pemrogram dapat diterima dengan baik oleh sisi target.



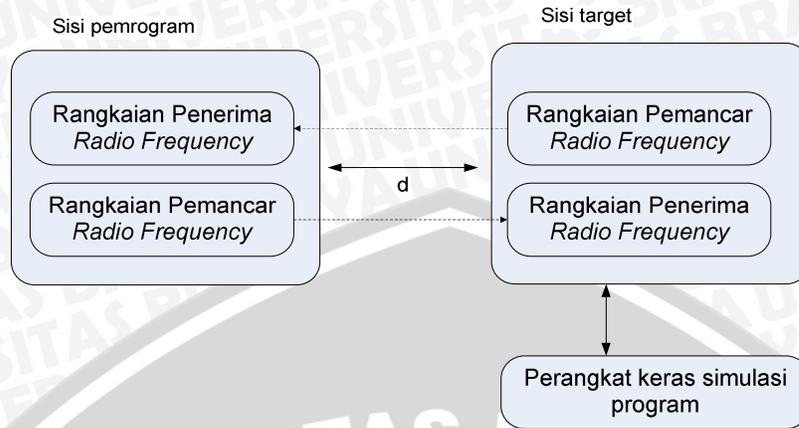
Gambar 5.12 Hasil pengujian operasi tulis memori flash

#### 5.4.1.2 Pengujian Penerimaan dari Sisi Target ke Sisi Pemrogram

Pengujian ini bertujuan untuk melihat respon sisi target terhadap permintaan data dari sisi pemrogram. Instrumen yang digunakan maupun susunan sistem sama dengan pengujian sebelumnya. Hasil pengujian untuk operasi baca memori EEPROM ditunjukkan dalam Gambar 5.13.

Dari hasil pengujian yang ditunjukkan dalam Gambar 5.13 dapat diamati bahwa *id device* sisi target adalah 0x82, kode instruksi untuk operasi baca memori EEPROM adalah 0x40, dua *byte* jumlah paket data, satu *byte id* mikrokontroler. Mikrokontroler target yang digunakan pada pengujian ini adalah ATmega8535 yang memiliki EEPROM berukuran 512 *bytes* dan satu *byte* checksum. Dari hasil pengujian, diketahui bahwa sisi target dapat merespon dengan baik permintaan data memori dari sisi pemrogram.





Gambar 5.14 Blok pengujian jarak jangkauan transmisi

- b. Transmisi data yang dilakukan adalah operasi tulis memori *flash* dengan data memori sebesar 288 bytes. Terdiri dari paket perintah dan paket data.
- c. Melakukan perubahan jarak jangkauan transmisi ( $d$ ) per 1 meter kemudian mengamati proses transmisi. Untuk setiap jarak jangkauan dilakukan pengujian transmisi data sebanyak tiga kali. Hasil pengujian ditunjukkan dalam Tabel 5.3.

Tabel 5.3 Hasil pengujian jarak jangkauan transmisi data dengan dengan memori sebesar 288 bytes

No	d (meter)	Status Transmisi	Keterangan
1	1	Berhasil	Seluruh transmisi data berhasil
2	2	Berhasil	Seluruh transmisi data berhasil
3	3	Berhasil	Seluruh transmisi data berhasil
4	4	Berhasil	Seluruh transmisi data berhasil
5	5	Berhasil	Seluruh transmisi data berhasil
6	6	Berhasil	Seluruh transmisi data berhasil
7	7	Berhasil	Seluruh transmisi data berhasil
8	8	Berhasil	Seluruh transmisi data berhasil
9	9	Berhasil	Seluruh transmisi data berhasil
10	10	Berhasil	Seluruh transmisi data berhasil
11	11	Berhasil	Seluruh transmisi data berhasil
12	12	Berhasil	Seluruh transmisi data berhasil
13	13	Berhasil	Seluruh transmisi data berhasil
14	14	Berhasil	Seluruh transmisi data berhasil
15	15	Berhasil	Seluruh transmisi data berhasil
16	16	Berhasil	Seluruh transmisi data berhasil
17	17	Berhasil	Seluruh transmisi data berhasil
18	18	Berhasil	Seluruh transmisi data berhasil
19	19	Berhasil	Seluruh transmisi data berhasil
20	20	Berhasil	Seluruh transmisi data berhasil
21	21	Berhasil	Seluruh transmisi data berhasil
22	22	Berhasil	Seluruh transmisi data berhasil
23	23	Berhasil	Seluruh transmisi data berhasil
24	24	Berhasil	Seluruh transmisi data berhasil
25	25	Berhasil	Seluruh transmisi data berhasil

26	26	Berhasil	Seluruh transmisi data berhasil
27	27	Berhasil	Seluruh transmisi data berhasil
28	28	Berhasil	Seluruh transmisi data berhasil
29	29	Berhasil	Seluruh transmisi data berhasil
30	30	Berhasil	Seluruh transmisi data berhasil

Dari hasil pengujian, maka jarak jangkauan transmisi data diketahui bisa mencapai sejauh 30 meter dan ini sudah melebihi target dari perancangan semula yaitu 20 meter.

#### 5.4.3 Pengujian Kecepatan Operasi Memori

Operasi memori yang telah berhasil dilakukan dengan baik adalah hapus chip, operasi baca dan tulis memori *flash*, operasi baca dan tulis memori EEPROM, serta baca dan tulis *fuse bits*.

Pengujian ini bertujuan untuk mengetahui waktu yang diperlukan untuk operasi tulis maupun baca memori mikrokontroler target. Pengujian ini berada dalam kondisi ideal tanpa gangguan serta jarak antara sisi pemrogram dengan sisi target terpisah sejauh satu meter. Mikrokontroler target adalah tipe ATmega8535. Frekuensi *clock* SCK yang digunakan di sisi target adalah 1382,4 kHz. Hasil pengujian terhadap waktu rata-rata yang diperlukan untuk operasi tulis memori tercantum di dalam Tabel 5.4 sedangkan waktu rata-rata yang dibutuhkan untuk operasi baca memori tercantum di dalam Tabel 5.5.

Tabel 5.4 Hasil pengujian operasi tulis memori

No	Ukuran data (byte)	Jenis memori	Waktu yang dibutuhkan (detik)	Status transmisi
1	256	Flash	1,5	Berhasil
2	512	Flash	3,5	Berhasil
3	1024	Flash	6	Berhasil
4	2048	Flash	11	Berhasil
5	3072	Flash	16,5	Berhasil
6	4096	Flash	22,5	Berhasil
7	5120	Flash	26,5	Berhasil

8	6144	Flash	32	Berhasil
9	7168	Flash	37,5	Berhasil
10	8192	Flash	42,5	Berhasil
11	256	EEPROM	1	Berhasil
12	512	EEPROM	1,5	Berhasil
13	3	Fuse Bits	1	Berhasil

Tabel 5.5 Hasil pengujian operasi baca memori

No	Ukuran data (byte)	Jenis memori	Waktu yang dibutuhkan (detik)	Status transmisi
1	8192	Flash	9	Berhasil
2	512	EEPROM	1,5	Berhasil
3	3	Fuse Bits	1	Berhasil

#### 5.4.4 Pengujian Penerimaan Data

Pengujian ini bertujuan untuk mengetahui tingkat keberhasilan sistem ketika mentransmisikan data memori hingga berukuran 32 KBytes. Mikrokontroler target adalah tipe Atmega32. Frekuensi *clock* SCK yang digunakan di sisi target adalah 344 kHz. Jenis operasi yang dipilih pada pengujian ini adalah operasi tulis memori *flash*. Hasil pengujian terhadap waktu rata-rata yang diperlukan untuk operasi tulis memori *flash* tercantum di dalam Tabel 5.6.

Tabel 5.6 Hasil pengujian operasi tulis memori

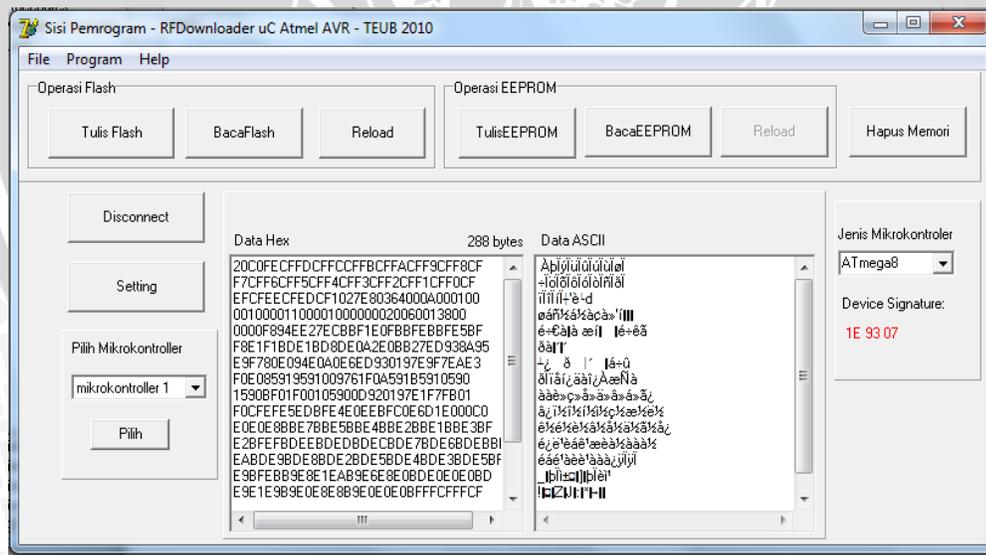
No	Ukuran data (KBytes)	Jenis memori	Waktu yang dibutuhkan (detik)	Status transmisi
1	2	Flash	11	Berhasil
2	4	Flash	25	Berhasil
3	6	Flash	32	Berhasil
4	8	Flash	42,5	Berhasil
5	12	Flash	46	Berhasil

6	16	Flash	62,5	Berhasil
7	24	Flash	93	Berhasil
8	32	Flash	118	Berhasil

Dari hasil pengujian dapat disimpulkan bahwa sistem mampu melakukan operasi tulis memori dengan ukuran memori maksimal mencapai 32 KBytes.

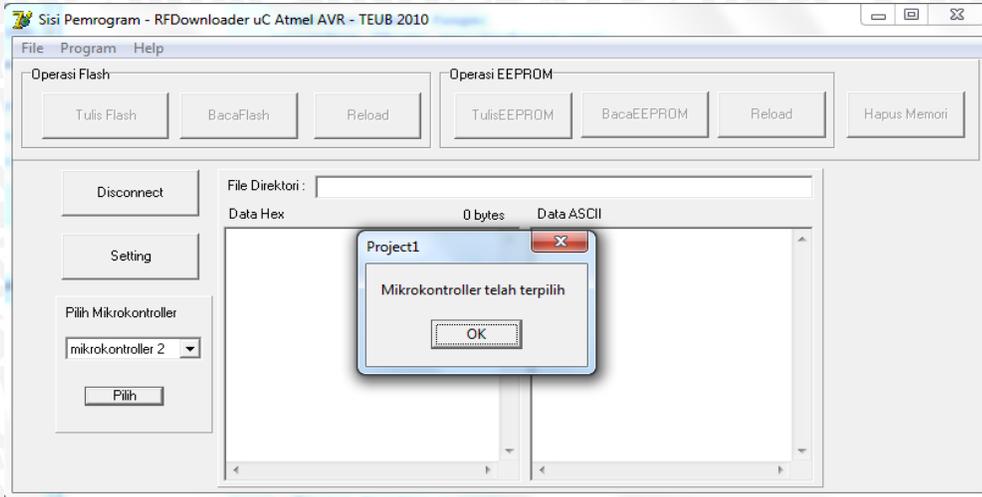
### 5.4.5 Pengujian Perangkat Lunak

Pengujian perangkat lunak pada sistem keseluruhan bertujuan untuk mengetahui bagaimana hasil dari program di perangkat lunak saat pengiriman data, penerimaan data, pemilihan mikrokontroler dan saat *load file hex*. Pengujian ini dilakukan saat proses *load file hex*, saat pemilihan mikrokontroler berhasil, saat proses pemilihan mikrokontroler gagal, saat proses sisi target tidak aktif, saat penulisan *flash* ke mikrokontroler target berhasil, saat proses pengiriman paket perintah pada saat penulisan *flash* ke mikrokontroler gagal, saat proses penulisan *flash* ke mikrokontroler gagal.



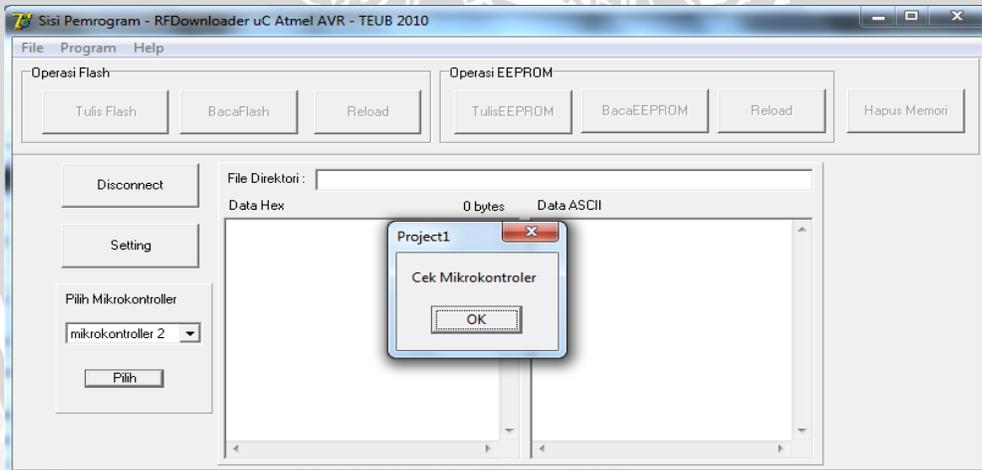
Gambar 5.15 Hasil dari pengujian perangkat lunak saat *load file hex*

Hasil tampilan program ditunjukkan dalam Gambar 5.15. Dapat dilihat bahwa program telah berhasil melakukan proses seleksi dan menampilkan file hex.



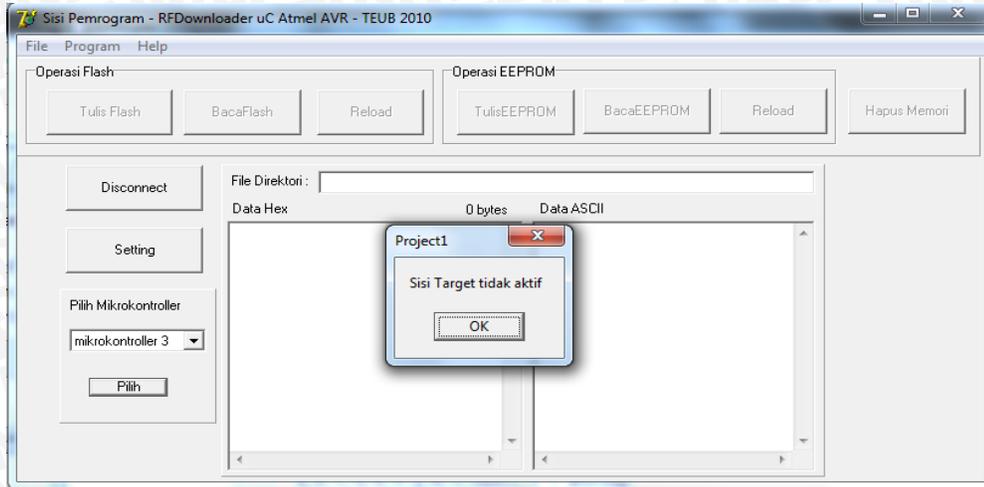
**Gambar 5.16** Hasil dari pengujian perangkat lunak saat memilih mikrokontroler dari sisi target

Hasil tampilan program ditunjukkan dalam Gambar 5.16. Dapat dilihat bahwa program telah berhasil melakukan proses pemilihan mikrokontroler dengan sukses.



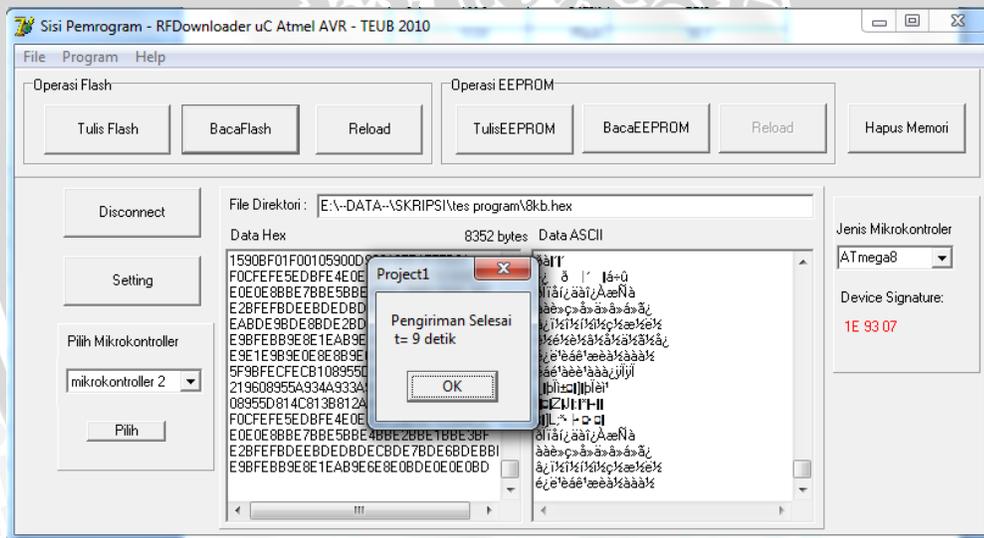
**Gambar 5.17** Hasil dari pengujian perangkat lunak saat pemilihan mikrokontroler dari sisi target gagal

Hasil tampilan program ditunjukkan dalam Gambar 5.17. Dapat dilihat bahwa program gagal melakukan proses pemilihan mikrokontroler.



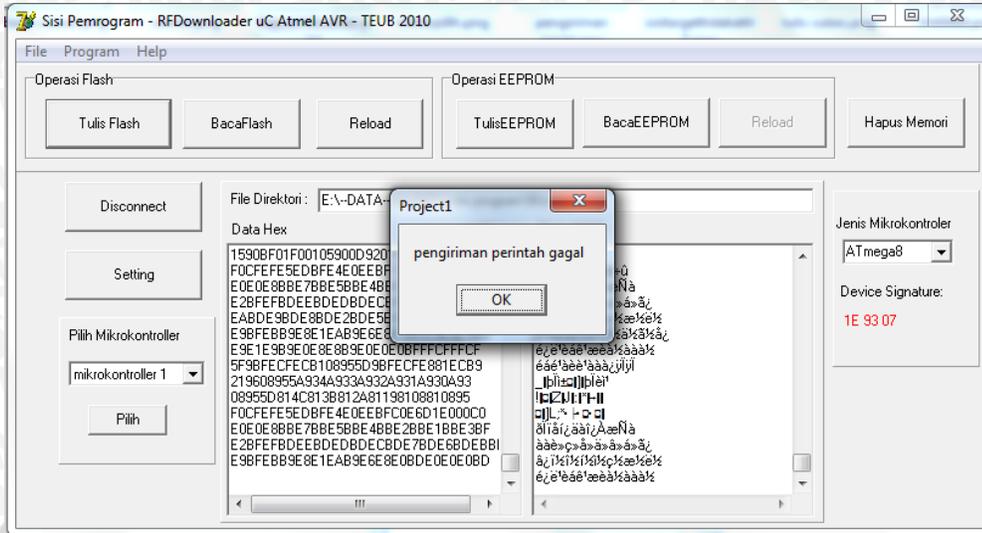
Gambar 5.18 Hasil dari pengujian perangkat lunak saat sisi target tidak aktif

Hasil tampilan program ditunjukkan dalam Gambar 5.18. Dapat dilihat bahwa program tidak menemukan koneksi dengan sisi target atau saat sisi target tidak aktif.

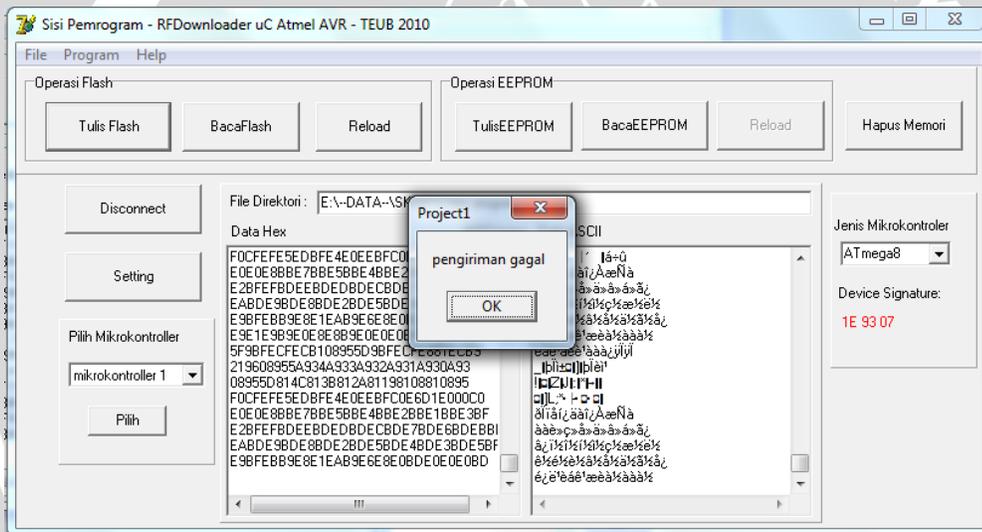


Gambar 5.19 Hasil dari pengujian perangkat lunak saat proses penulisan flash sukses

Hasil tampilan program ditunjukkan dalam Gambar 5.19. Dapat dilihat bahwa program telah berhasil melakukan proses penulisan flash mikrokontroler dengan sukses.



Gambar 5.20 Hasil dari pengujian perangkat lunak saat proses pengiriman paket perintah gagal



Gambar 5.21 Hasil dari pengujian perangkat lunak saat proses pengiriman data gagal

Hasil tampilan program ditunjukkan dalam Gambar 5.20 dan 5.21. Dapat dilihat bahwa program gagal melakukan proses penulisan *flash* mikrokontroler dikarenakan paket perintah yang gagal diterima dan paket data yang tidak berhasil diterima oleh sisi target. Dari hasil pengujian dapat disimpulkan bahwa perangkat lunak sudah sesuai dengan perancangan yang diinginkan.

## BAB VI

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

1. Antarmuka yang menghubungkan pemancar dan penerima RF dengan komputer telah berfungsi baik dengan komunikasi serial. Unit radio frekuensi yang terdiri dari satu pemancar dan penerima radio frekuensi YS-1020 dapat bekerja dengan baik dengan *baudrate* transmisi 9600 bps dan jarak jangkauan transmisi hingga 30 meter atau lebih dalam kondisi ideal.
2. Protokol komunikasi data melalui media radio frekuensi dibangun satu tingkat di atas protokol USART pada mikrokontroler. Paket transmisi yang terdiri atas paket perintah, paket data dan paket konfirmasi. Transmisi data serial menggunakan fasilitas USART0 dengan format frame terdiri atas 1 bit start, 8 bit data serta 1 bit stop.
3. Sistem mampu menangani operasi memori empat buah mikrokontroler secara bergantian dengan adanya sesi pilih mikrokontroler target.
4. Program *downloader* di sisi pemrogram dapat bekerja dengan baik untuk operasi-operasi memori mikrokontroler Atmel AVR yaitu operasi memori flash, eeprom dan fusebits dapat memberikan kemudahan pada pengguna saat pengoperasian alat.

#### 6.2 Saran

Terdapat beberapa hal yang perlu diperhatikan dalam pengembangan alat ini di kemudian hari. Meskipun alat ini sudah dapat bekerja dengan baik dan sesuai dengan spesifikasi yang diinginkan masih terdapat hal-hal yang perlu ditingkatkan. Hal-hal tersebut antara lain:

1. Sistem dikembangkan agar dapat menangani devais dengan kapasitas memori lebih dari 32 kbyte.
2. Media transmisi data selain radio frekuensi yang memungkinkan transmisi data dengan *baudrate* yang lebih tinggi dan lebih handal

3. Dimungkinkan sistem dapat dikembangkan untuk menangani operasi memori pada mikrokontroler selain dari keluarga Atmel AVR.



## DAFTAR PUSTAKA

- Atmel. 2000. *AVR910: In-System Programming*. San Jose: Atmel.  
[http://www.atmel.com/dyn/resources/prod\\_documents/doc0943.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc0943.pdf).  
Diakses tanggal 28 September 2010
- Atmel. 2002. *AVR236: CRC Check of Program Memory*. San Jose: Atmel.  
[http://www.atmel.com/dyn/resources/prod\\_documents/doc1143.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc1143.pdf).  
Diakses tanggal 28 September 2010
- Atmel. 2008. *AVRISP Help*. San Jose: Atmel.  
[http://www.atmel.org/dyn/resources/prod\\_documents/AVRISP.chm](http://www.atmel.org/dyn/resources/prod_documents/AVRISP.chm).  
Diakses tanggal 28 September 2010
- Cantu, Marco. 2003. *Mastering Delphi 7*. Sybex Inc.
- Forouzan, Behrouz A. 2000. *Data Communications and Networking 2nd edition*. New York. Mc Graw-Hill International Edition.
- Gadre, Dhananjay V. 2001. *Programming And Customizing The AVR Microcontroller*. New York: McGraw-Hill
- Hardana, HendyEka .2008. *Sistem Downloader Data Memori Mikrokontroler Atmel-AVR Melalui Inframerah : Sisi pemrogram*. Malang. Departemen Pendidikan Nasional Universitas Brawijaya Fakultas Teknik Jurusan Teknik Elektro
- Intel. 1988. *Intel Hexadecimal Object File Format Specification*. Santa Clara: Intel
- Lamont, John. Ralph Patterson. 2001. *Universal RF Communication System Project Plan*. ECPE Senior Design.
- Maxim. 2003. *MAXIM RS 232 Drivers/Receivers*. USA: MAXIM Integrated Products.
- Prasad, Dr. K. V. 2003. *Principles of Digital Communication Systems and Computer Networks*. Hingham. Charles River Media, Inc.
- Schmidt, Thomas. 2000. *CRC Generating And Checking*. Chandler: Microchip Technology.  
<http://ww1.microchip.com/downloads/en/AppNotes/00730a.pdf>. Diakses tanggal 25 September 2010
- Scienceprog. 2006. *Shelling The Intel 8-bit Hex File Format*.  
<http://www.scienceprog.com/shelling-the-intel-8-bit-hex-file-format>.  
Diakses tanggal 20 September 2010
- ShenZen YiShi. 2009. *YS-1020UA RF Data Transceiver*. Hongkong. ShenZhen Yishi Electronic Technology Development Co., Ltd.
- Williams, Ross N. 1993. *A Painless Guide to CRC Error Detection Algorithms*. Adelaide: Rocksoft Pty