

BAB I

PENDAHULUAN

1.1. Latar Belakang

Muatan roket merupakan salah satu produk elektronika yang saat ini sedang berkembang pesat. Perkembangan teknologi ini juga diikuti dengan perkembangan teknologi sensor yang dipakai pada muatan roket. Sensor merupakan komponen penting pada muatan roket karena dengan adanya sensor-sensor didalamnya maka muatan roket dapat beroperasi secara otomatis. LAPAN (Lembaga Penerbangan dan Antariksa Nasional) sebagai pengembang dalam dunia peroketan telah menyelenggarakan Kompetisi Muatan Roket Indonesia (KOMURINDO) tingkat perguruan tinggi. KOMURINDO merupakan suatu ajang kompetisi yang diadakan setiap tahun sebagai sarana pendidikan dan menarik minat, sekaligus untuk menyiapkan bibit unggul tenaga ahli peroketan. Diharapkan, kompetisi ini dapat meningkatkan kemampuan mahasiswa dalam merancang bangun teknologi peroketan pada bagian muatan roket mulai dari mendesain, membuat, menguji fungsional sampai dengan melaksanakan uji terbang muatan roket.

Terdapat tema yang berbeda-beda setiap tahunnya dalam pelaksanaan kompetisi ini. Pada Kompetisi Muatan Roket Indonesia, setiap muatan roket yang dilombakan menggunakan sensor-sensor untuk mengukur parameter-parameter meteorologi. Untuk itu, muatan roket dilengkapi dengan sensor seperti: sensor suhu, sensor percepatan, sensor kelembapan, sensor tekanan, sensor kamera dan sensor kompas digital. Sensor kompas digital merupakan salah satu sensor yang digunakan untuk menunjukkan arah muatan roket. Sehingga dengan menggunakan sensor kompas digital, kita dapat mengetahui kemana arah muatan roket menuju.

Dalam penelitian ini akan dibahas implementasi sensor kompas digital untuk memonitor arah muatan roket. Sensor kompas digital yang akan dipakai dalam penelitian ini adalah sensor kompas digital CMPS03. Proses pembuatan penelitian ini terbagi kedalam 2 tahap yaitu perancangan sistem muatan roket dan perancangan sistem *ground segmen*. Perancangan sistem muatan roket meliputi sensor kompas digital CMPS03 dan komponen-komponen elektronika. Sedangkan perancangan *ground segmen* meliputi software antarmuka untuk memonitor arah muatan roket.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan sebelumnya, dapat disusun rumusan masalah sebagai berikut:

- 1) Bagaimana merancang dan membuat sistem elektronika yang dapat menunjang penggunaan sensor kompas digital CMPS03 sebagai data arah muatan roket.
- 2) Bagaimana mengkalibrasi sensor kompas digital CMPS03 agar data keluaran dari sensor mendekati nilai sebenarnya.
- 3) Bagaimana merancang software antarmuka agar dapat menerima data arah dari sensor kompas digital CMPS03 untuk memonitor arah muatan roket secara *real time* dalam bentuk visual dan grafik.

1.3. Batasan Masalah

Dengan mengacu pada permasalahan yang telah dirumuskan, maka hal-hal yang berkaitan dengan perancangan akan diberi batasan sebagai berikut:

- 1) Muatan roket ini menggunakan modul sensor kompas digital CMPS03.
- 2) Mikrokontroler yang digunakan adalah ATmega8.
- 3) Muatan roket yang dirancang dapat menunjukkan arah mata angin (0° - 359°).
- 4) Komunikasi data menggunakan modul *transceiver* YS-1020.
- 5) Alat yang dirancang hanya muatan roket, tidak termasuk peluncur roketnya.

1.4. Tujuan

Penelitian ini bertujuan untuk mengimplementasikan sensor kompas digital yang dapat menunjukkan arah muatan roket (0° - 359°) dan memonitor arah muatan roket dengan menggunakan software antarmuka secara *real time* dalam bentuk visual dan grafik.

1.5. Sistematika Pembahasan

Penelitian ini terdiri atas enam bab dengan sistematika pembahasan sebagai berikut:

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, ruang lingkup, tujuan, dan sistematika pembahasan.

BAB II Tinjauan Pustaka

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

BAB III Metodologi

Berisi tentang metode-metode yang dipakai dalam melakukan perancangan, pengujian, dan analisis data.

BAB IV Perancangan

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

BAB V Pengujian dan Analisis

Memuat aspek pengujian meliputi penjelasan tentang cara pengujian dan hasil pengujian. Aspek analisis meliputi penilaian atau komentar terhadap hasil-hasil pengujian. Pengujian dan analisis ini terhadap alat yang telah direalisasikan berdasarkan masing-masing blok dan sistem secara keseluruhan.

BAB VI Kesimpulan dan Saran

Memuat intisari hasil pengujian dan menjawab rumusan masalah serta memberikan rekomendasi untuk perbaikan kualitas penelitian di masa yang akan datang.



BAB II

TINJAUAN PUSTAKA

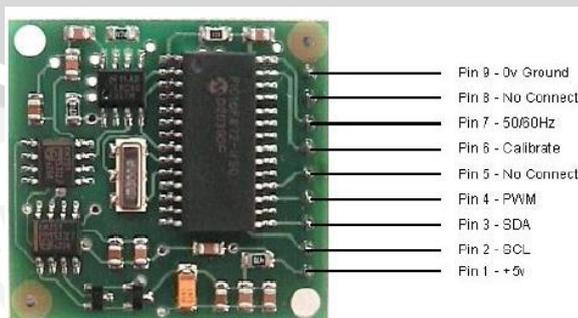
Muatan roket merupakan salah satu bentuk produk teknologi di bidang elektronika yang saat ini berkembang pesat. Perkembangan teknologi roket ini juga diikuti dengan perkembangan teknologi sensor yang semakin canggih. Sensor merupakan komponen penting pada muatan roket karena dengan adanya sensor-sensor didalamnya maka muatan roket dapat dimonitor secara jarak jauh.

Sensor kompas digital CMPS03 adalah salah satu sensor yang digunakan untuk menunjukkan arah. Sensor kompas ini diletakkan ke dalam muatan roket sehingga dapat diketahui arah muatan roket menuju. Selain itu, dengan menggunakan software antarmuka muatan roket dapat dimonitor arah muatan roket secara visual dan grafik.

2.1. Sensor Kompas Digital CMPS03

Dalam memonitor sistem pergerakan suatu muatan roket, diperlukan suatu sensor posisi untuk mengetahui kemana arah muatan roket yang akan dimonitor. Jadi dengan penggunaan sensor posisi pada muatan roket, diharapkan muatan roket tidak hanya bergerak ke kanan dan ke kiri saja tetapi juga dapat diketahui arah muatan roket tersebut.

CMPS03 merupakan modul sensor kompas digital dilengkapi dengan dua sensor medan magnet chip KMZ51 buatan Philips yang cukup peka untuk mendeteksi medan magnet bumi. Dua sensor ini dipasang saling bersilangan. Pada modul kompas telah dipasang rangkaian pengkondisi sinyal dan mikrokontroler. Sehingga kita dapat mengakses berapa derajat posisi kompas secara langsung. CMPS03 memiliki resolusi 0,1°, sehingga hasil keluaran akan menjadi lebih presisi. Bentuk fisik CMPS03 ditunjukkan dalam Gambar 2.1.



Gambar 2.1 Bentuk Fisik Sensor Kompas Digital CMPS03

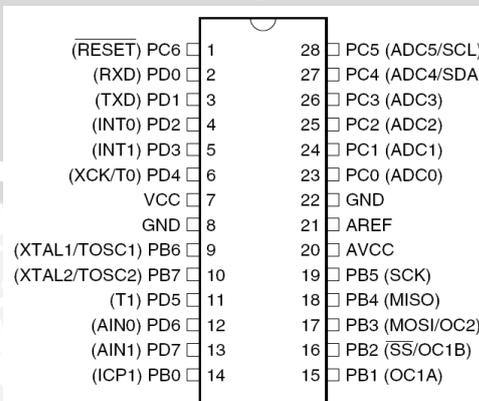
Modul kompas digital CMPS03 membutuhkan supply tegangan sebesar 5 V DC dengan konsumsi arus sebesar 15 mA. Pengambilan data arah dari modul kompas CMPS03 ke mikrokontroler dapat dilakukan dengan dua cara. Pertama menggunakan data PWM (Pulse Width Modulation), 1 ms (untuk 0°) sampai 36.99 ms (untuk 359°). Cara yang kedua dengan menggunakan I2C (Inter Integrated Circuit), yaitu dengan menghubungkan pin SDA (Serial Data) dan SCL (Serial Clock) yang dihubungkan ke supply 5 V DC melalui resistor *pull-up*. Metode ini dapat digunakan langsung sehingga data yang dibaca tepat 0°-359°.

2.2. ATmega8

ATmega8 adalah mikrokontroler CMOS 8-bit berarsitektur AVR RISC yang memiliki 8 KBytes *In-System Programmable Flash*. Fitur Utama AVR ATmega8 adalah:

- Mikrokontroler AVR 8 bit memiliki kemampuan tinggi dengan daya rendah.
- Sistem *self-programable flash* 8 kbyte.
- EEPROM sebesar 512 byte, dan 1 kbyte SRAM internal 23 saluran I/O dan 32 *general purpose register*.
- Dua timer/counter 8 bit dan satu timer/counter 16 bit dengan *prescaler* terpisah
- Internal dan eksternal *interrupt*.
- Pin SDA dan SCL untuk komunikasi I2C
- Tiga pin PWM.
- Enam pin ADC (*Analog to Digital Converter*) dengan ketepatan sebesar 10 bit.
- USART (*Universal Synchronous and Asynchronous Receiver Transmitter*).

Konfigurasi pin-pin pada ATmega8 dengan kemasan 28-pin DIP (*Dual In-Line Package*) ditunjukkan dalam Gambar 2.2.



Gambar 2.2 ATmega8

Sumber : ATmega8 Datasheet, 2003: 2

2.3. Komunikasi Data Serial

Ada 2 macam cara komunikasi data serial yaitu Sinkron dan Asinkron. Pada komunikasi data serial sinkron, *clock* dikirimkan bersama sama dengan data serial, tetapi *clock* tersebut dibangkitkan sendiri-sendiri baik pada sisi pengirim maupun penerima. Sedangkan pada komunikasi serial asinkron tidak diperlukan *clock* karena data dikirimkan dengan kecepatan tertentu yang sama baik pada pengirim/penerima.

Pada UART, kecepatan pengiriman data (*baudrate*) dan *fase clock* pada sisi transmitter dan sisi receiver harus sinkron. Untuk itu diperlukan sinkronisasi antara Transmitter dan Receiver. Hal ini dilakukan oleh bit “Start” dan bit “Stop”. Ketika saluran transmisi dalam keadaan *idle*, output UART adalah dalam keadaan logika “1”. Ketika Transmitter ingin mengirimkan data, *output* UART akan diset dulu ke logika “0” untuk waktu satu bit. Sinyal ini pada receiver akan dikenali sebagai sinyal “Start” yang digunakan untuk menyinkronkan *fase clock* nya sehingga sinkron dengan *fase clock* transmitter.

Selanjutnya data akan dikirimkan secara serial dari bit yang paling rendah (bit 0) sampai bit tertinggi. Selanjutnya akan dikirimkan sinyal “Stop” sebagai akhir dari pengiriman data serial. Sebagai contoh misalnya akan dikirimkan data huruf “A” dalam format ASCII (atau sama dengan 41 heksa atau 0100 0001). Kecepatan transmisi (*baud rate*) dapat dipilih bebas dalam rentang tertentu. *Baudrate* yang umum dipakai adalah 110, 135, 150, 300, 600, 1200, 2400, dan 9600 (bit/per detik). Dalam komunikasi data serial, *baudrate* dari kedua alat yang berhubungan harus diatur pada kecepatan yang sama. Selanjutnya harus ditentukan panjang data (6,7 atau 8 bit), paritas (genap, ganjil, atau tanpa paritas), dan jumlah bit “Stop” (1, 1 ½, atau 2 bit).

Mikrokontroler ATmega8 dilengkapi dengan fasilitas komunikasi serial USART (*Universal Synchronous and Asynchronous serial Receiver and Transmitter*) dengan fitur sebagai berikut:

- 1) Komunikasi *full-duplex* dengan register serial untuk penerima dan pengirim data.
- 2) Dapat dioperasikan pada mode komunikasi sinkron dan asinkron.
- 3) Pada operasi sinkron, *clock* berasal dari *master* atau *slave*.
- 4) Mempunyai resolusi tinggi untuk pembangkit *baudrate*.
- 5) Paritas genap atau ganjil dan didukung dengan pengecekan paritas oleh *hardware*.

- 6) Pendeteksi pengiriman kelebihan data.
- 7) Pendeteksi kesalahan pada format data yang dikirim
- 8) Memiliki filter *noise* yang terdiri atas pendeteksi kesalahan bit *start* dan *low pass filter*.
- 9) Memiliki 3 layanan *interrupt* yaitu TX *complete*, TX *data empty*, dan RX *complete*.
- 10) Mode komunikasi multi prosesor.
- 11) Mode komunikasi asinkron dengan dua kecepatan.

Untuk menghitung *baudrate* komunikasi serial digunakan rumus seperti ditunjukkan dalam Tabel 2.1.

Tabel 2.1. Rumus menghitung *baudrate*

| Operating Mode | Equation for Calculating Baud Rate | Equation for Calculating UBRR Value |
|--|---------------------------------------|-------------------------------------|
| Asynchronous Normal mode (U2X = 0) | $BAUD = \frac{f_{osc}}{16(UBRR + 1)}$ | $UBRR = \frac{f_{osc}}{16BAUD} - 1$ |
| Asynchronous Double Speed Mode (U2X = 1) | $BAUD = \frac{f_{osc}}{8(UBRR + 1)}$ | $UBRR = \frac{f_{osc}}{8BAUD} - 1$ |
| Synchronous Master Mode | $BAUD = \frac{f_{osc}}{2(UBRR + 1)}$ | $UBRR = \frac{f_{osc}}{2BAUD} - 1$ |

Sumber : Atmel, 2006:136

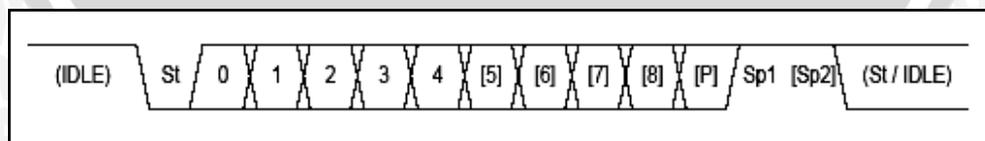
dengan:

f_{osc} = Frekuensi *clock* sistem osilator

UBRR = Register *baudrate* yang terdiri dari UBRRH dan UBRRH

BAUD = *Baudrate* dalam *bit per second* (bps)

Format pengiriman data secara serial menggunakan ATMega8 ditunjukkan dalam Gambar 2.3.



Gambar 2.3 Format data pengiriman

Sumber : Atmel, 2006:137

dengan:

St = Bit *start* selalu berlogika rendah

(n) = Banyaknya data yang dikirim (0-8)

P = Bit paritas (ganjil atau genap)

Sp = Bit *stop* selalu berlogika tinggi (bit stop bisa berjumlah 1 atau 2)
 IDLE = Tidak ada data yang ditransfer pada RX dan TX, IDLE selalu berlogika tinggi.

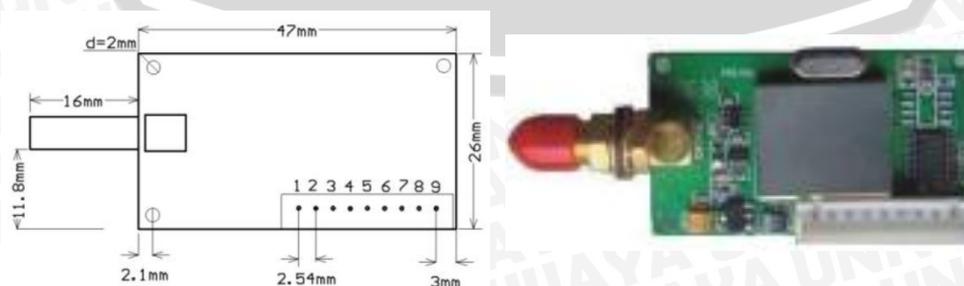
2.4. Radio Frekuensi YS-1020U

YS-1020 merupakan modul Radio Frekuensi menggunakan daya rendah yang dirancang untuk mentransmisikan data secara serial (UART) dalam jarak pendek. YS-1020 terdiri dari komponen terintegrasi (CC1020 RF) yang dapat bekerja dalam dua arah yaitu dapat sebagai pemancar (*Transmitter*) dan penerima (*Receiver*). Perangkat ini dapat secara langsung dikoneksikan dengan *processor*, PC, RS-485, RS-232, dan *port* antarmuka serial lain level TTL.

Fitur-fitur yang dimiliki YS-1020 adalah sebagai berikut:

- 1) Antarmuka yang dipakai menggunakan level RS-232 / RS-485 / TTL
- 2) Mempunyai 8 *channel* yang dapat diperluas hingga 16 / 32 *channel*.
- 3) *Baudrate* di udara adalah 1200 / 2400 / 4800 / 9600 / 19200 / 38400 bps.
- 4) Modulasi yang digunakan adalah GFSK (*Gaussian Frequency Shift Keying*).
- 5) Pemancar dan penerima terpadu.
- 6) Konsumsi daya rendah yaitu 50 mW.
- 7) Sensitivitas penerima sebesar 115 dBm (@9600 bps) dan 120 dBm (@1200 bps).
- 8) Memiliki jangkauan hingga kurang dari atau sama dengan 1 km ketika antena berada 2 m di atas tanah, dan BER-nya sebesar 10^{-3} @1200 bps.
- 9) Frekuensi kerjanya adalah 434,0325 MHz.
- 10) Membutuhkan catu daya 3,3 - 5 V DC.

Berikut adalah bentuk fisik Radio Frekuensi YS-1020U ditunjukkan dalam Gambar 2.4 dan keterangan masing-masing pin ditunjukkan dalam Tabel 2.2.



Gambar 2.4 Bentuk Dimensi RF YS-1020U

Sumber : ShenZhen YiShi Electronic Technology Development, 2003 : 2

Tabel 2.2. Keterangan pin-pin YS-1020U

| Pin | Pin Name | Description | Level | Connection With Terminal |
|-----|-----------------|------------------------------|-------------|--------------------------|
| 1. | GND | Grounding of Power Supply | | Ground |
| 2. | V _{CC} | Power Supply DC | 3.3 – 5.5 V | |
| 3. | RXD / TTL | Serial data receiving end | TTL | TXD |
| 4. | TXD / TTL | Serial data transmitting end | TTL | RXD |
| 5. | DGND | Digital Grounding | | |
| 6. | A(TXD) | A of RS-485 or TXD of RS-232 | | A(RXD) |
| 7. | B(TXD) | B of RS-485 or RXD of RS-232 | | B(TXD) |
| 8. | Sleep | Sleep control (input) | TTL | Sleep Signal |
| 9. | Test | Ex-factory testing | | |

Sumber : ShenZhen YiShi Electronic Technology Development, 2003 : 2

2.5. Delphi

Delphi adalah suatu bahasa pemrograman (*development language*) yang digunakan untuk merancang suatu aplikasi program OOP (*Object Oriented Programming*). OOP adalah metode pemrograman dengan membentuk sebuah aplikasi yang mendekati keadaan dunia yang sesungguhnya. Hal itu bisa dilakukan dengan cara mendesain object untuk menyelesaikan masalah. Ada tiga unsur dalam OOP yaitu:

1) *Encapsulation* atau pemodelan

Encapsulation adalah konsep penggabungan data dengan operator. Dalam konsep pemodelan data dan operasi menjadi satu kesatuan yang disebut *object*.

2) *Inheritance* atau penurunan

Inheritance adalah sebuah object yang dapat diturunkan menjadi object yang baru dengan tidak menghilangkan sifat asli dari object tersebut.

3) *Polymorphism*

Polymorphism merupakan penggunaan berbagai macam object yang berbeda tetapi secara fungsi bergantung pada satu object sebagai induk, dengan cara pelaksanaan yang berbeda-beda.

2.5.1 Delphi dan OOP (*Object Oriented Programming*)

Setiap merancang suatu aplikasi program, tanpa disadari telah menerapkan OOP, walaupun secara teori kurang memahami OOP dalam arti yang sebenarnya. Contoh sederhananya adalah ketika merancang suatu form (Tform1) baru, sadar atau tidak sebenarnya form yang aktifkan merupakan turunan dari Tform sebagai induknya

atau ketika mengaktifkan button pada form merupakan turunan dari Tbutton. Contoh dalam bahasa program adalah sebagai berikut:

Ketika merancang suatu label di form secara otomatis Delphi akan menuliskan label tersebut dalam jendela code editor tentang turunan dari label tersebut.

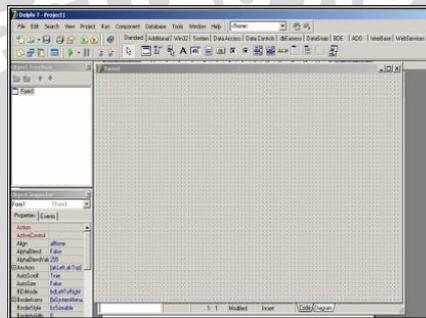
Type

```
Tform = class(tform)
```

```
Label1 = tlabel
```

```
End;
```

IDE (*Integrated Development Environment*) Delphi atau lingkungan pengembangan aplikasi sendiri ditunjukkan dalam Gambar 2.5.

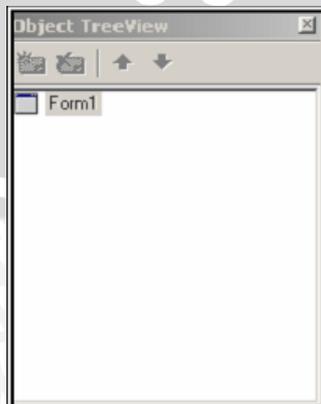


Gambar 2.5 Jendela Utama Delphi

Bagian-bagian dari jendela Delphi sebagai berikut:

1) Object Tree View

Merupakan sebuah diagram pohon yang menggambarkan hubungan logis menghubungkan semua komponen yang terdapat dalam suatu proyek program. Komponen tersebut meliputi form, modul atau frame. Fungsinya digunakan untuk menampilkan seluruh daftar komponen program dalam sebuah aplikasi program sesuai dengan penempatannya. Jendela *object tree view* ditunjukkan dalam Gambar 2.6.



Gambar 2.6 Jendela Object Tree View

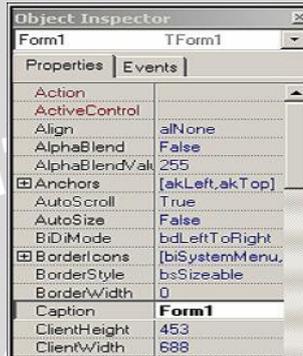
2) Object Inspector

Merupakan jendela yang digunakan untuk mengatur tampilan komponen pada form.

Secara Umum Object Inspector terbagi menjadi 2, yaitu:

a) Properties

Digunakan untuk mengatur tampilan pada sebuah komponen baik itu meliputi penggantian nama, warna, jenis huruf, *border*, dan lain-lain. Jendela *inspector* ditunjukkan dalam Gambar 2.7.



Gambar 2.7 Jendela *Properties* pada *Object Inspector*

b) Events

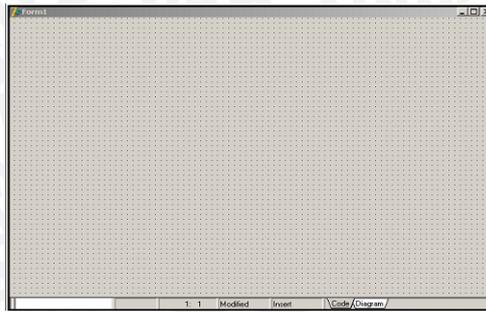
Merupakan jendela properties yang digunakan untuk memberikan fungsi yang lebih detail dari fungsi sebenarnya. Misalnya ketika tombol Simpan di klik maka program akan menjalankan perintah penyimpanan data. Dari kalimat tersebut ada event klik untuk mengeksekusi sebuah tombol simpan. Perintah event klik tersebut dapat diberikan melalui jendela events. Jendela *events* ditunjukkan dalam Gambar 2.8.



Gambar 2.8. Jendela *Events* pada *Object Inspector*

c) Form Designer

Merupakan tempat yang digunakan untuk merancang semua aplikasi program yang diambil dari komponen pallete. Jendela form *designer* ditunjukkan dalam Gambar 2.9.



Gambar 2.9. Jendela Form Designer

3) Component Palette

Merupakan kumpulan icon yang digunakan untuk merancang suatu aplikasi pada untuk membentuk sebuah aplikasi user interface. Dalam komponen palette semua icon dikelompokkan kedalam berbagai komponen sesuai dengan fungsi dan kegunaannya. Beberapa jendela komponen palette ditunjukkan dalam Gambar 2.10 dan Tabel 2.3.



Gambar 2.10 Jendela Komponen Palette

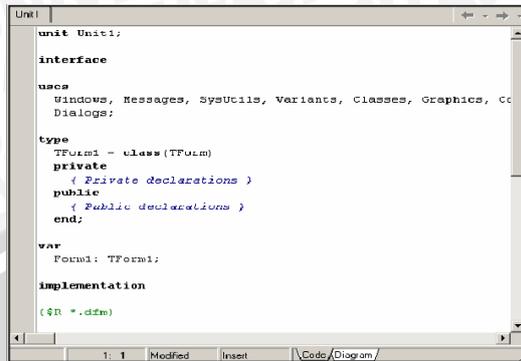
Tabel 2.3 Beberapa Komponen Palette dan Fungsinya

| No | Icon | Name | Fungsi |
|----|------|--------------|---|
| 1 | | Pointer | Mengembalikan fungsi mouse ke defaultnya |
| 2 | | Frame | Membentuk suatu frame terhadap obyek yang ada didalamnya |
| 3 | | Main menu | Membuat menu Utama |
| 4 | | Popup Menu | |
| 5 | | Label | Hanya untuk menampilkan Teks |
| 6 | | Edit | Untuk menampilkan dan input data (1 baris) |
| 7 | | Memo | Sama seperti edit tetapi mempunyai kapasitas lebih besar (lebih dari 1 baris) |
| 8 | | Button | Digunakan untuk melakukan eksekusi terhadap suatu proses |
| 9 | | Checkbox | Digunakan untuk menentukan pilihan lebih dari satu |
| 10 | | Radio Button | Digunakan untuk menentukan pilhan, tetapi hanya satu pilhan yang bisa digunakan |
| 11 | | List Box | Menmpilkan pilihan dalam bentuk list |
| 12 | | Combo Box | Menampilkan pilihan dalam bentuk popup |
| 13 | | Scroll Bar | Merupakan icon yang berupa baris status |
| 14 | | Group Box | Digunakan untuk mengelompokan suatu icon |
| 15 | | Radio Group | Digunakan untuk mengelompokan pilihan |



4) Code Editor

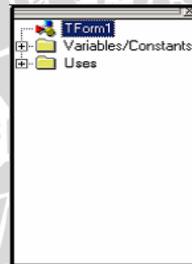
Bagian dari delphi yang digunakan untuk menuliskan kode program. Jendela *code editor* ditunjukkan dalam Gambar 2.11.



Gambar 2.11 Jendela *Code Editor*

5) Code Explorer

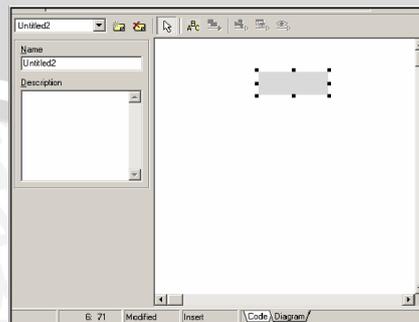
Merupakan jendela yang digunakan untuk menampilkan seluruh variabel, tipe, dan routine yang didefinisikan pada sebuah unit. Jendela *code explorer* ditunjukkan dalam Gambar 2.12.



Gambar 2.12 Jendela *Code Explorer*

6) Code Diagram

Merupakan fasilitas pada delphi yang digunakan untuk mendesain sebuah diagram atas komponen-komponen yang digunakan dalam suatu rancangan aplikasi. Jendela *code diagram* ditunjukkan dalam Gambar 2.13.



Gambar 2.13. Jendela *Code Diagram*

BAB III

METODOLOGI PENELITIAN

Penyusunan laporan ini didasarkan pada masalah yang bersifat aplikatif, yaitu perencanaan dan perealisasiian alat agar dapat bekerja sesuai dengan yang direncanakan dengan mengacu pada rumusan masalah. Langkah-langkah yang perlu dilakukan untuk merealisasikan alat yang dirancang adalah penentuan spesifikasi alat, perancangan alat, pembuatan alat, dan pengujian alat.

3.1. Penentuan Spesifikasi Alat

Spesifikasi alat secara global ditetapkan terlebih dahulu sebagai acuan dalam perancangan selanjutnya. Spesifikasi alat yang direncanakan yaitu:

- a) Alat terdiri atas dua unit, yaitu sistem muatan roket sebagai pengambil, pengolah, dan pengirim data arah muatan roket. Kedua sistem *ground segmen* sebagai penerima dan penampil data arah muatan roket secara visual dan grafik.
- b) Sensor yang digunakan adalah kompas digital CMPS03.
- c) Arah muatan roket adalah 0° - 359° .
- d) Jalur transmisi data dari unit pengirim data ke unit penerima data berupa modul *transceiver* YS-1020U.
- e) Sistem *ground segmen* berupa perangkat lunak (*interface software*) yang dapat memonitor arah muatan roket secara *real time* dalam bentuk visual dan grafik.
- f) Mikrokontroler yang digunakan adalah ATmega8.
- g) Sistem menggunakan catu daya dari baterai Lipoly 11,1 V.
- h) Bentuk fisik alat berupa sebuah muatan roket berbentuk tabung dengan diameter 9 cm dan panjang 15 cm.

3.2. Perancangan dan Perealisasiian Alat

Perancangan dan pembuatan alat dalam penelitian ini dibagi menjadi dua bagian, yaitu perancangan dan pembuatan perangkat keras (*hardware*) dan perancangan dan penyusunan perangkat lunak (*software*).

3.2.1. Perancangan dan Pembuatan Perangkat Keras (*Hardware*)

Dalam perancangan dan pembuatan perangkat keras terlebih dahulu merancang diagram blok sistem. Dari diagram blok dapat ditentukan apa saja spesifikasi yang alat

dibuat. Kemudian, perancangan mekanik muatan roket serta menentukan desain muatan roket. Selanjutnya, perancangan rangkaian catu daya, rangkaian sensor kompas digital CMPS03, perancangan rangkaian mikrokontroler ATmega8, perancangan pengirim dan penerima RF, dan rangkaian penyesuai level tegangan (MAX232).

3.2.2. Perancangan dan Penyusunan Perangkat Lunak (*Software*)

Setelah mengetahui seperti apa perangkat keras yang dirancang, dibutuhkan perangkat lunak untuk mengendalikan dan mengatur kerja dari alat ini. Desain dan parameter yang telah dirancang kemudian diterapkan ke dalam perancangan perangkat lunak kompas digital CMPS03 dan mikrokontroler ATmega8 dengan menggunakan *compiler CodeVisionAVR* dan. Sedangkan *software* antarmuka yang dapat memonitor arah muatan roket menggunakan Delphi7.

3.3. Pengujian Alat

Untuk menganalisis kinerja alat apakah sesuai dengan yang direncanakan maka dilakukan pengujian sistem. Pengujian dilakukan pada masing-masing blok dan secara keseluruhan.

3.4. Pengujian Perangkat Keras

Pengujian perangkat keras dilakukan dengan tujuan untuk menyesuaikan nilai tegangan dan arus yang diijinkan bekerja dalam komponen berdasarkan data sekunder komponen yang diambil dari buku data komponen elektronika maupun dari *datasheet*.

3.5. Pengujian Perangkat Lunak

Untuk pengujian perangkat lunak, pengujian dilakukan dengan cara mensimulasikan perangkat lunak pada mikrokontroler, kemudian dilakukan pengujian bersama perangkat keras untuk mengetahui respon yang dihasilkan. Pengujian perangkat lunak (*software interface*) bertujuan untuk mengetahui apakah perangkat lunak tersebut dapat mengolah data dalam bentuk visual dan grafik sesuai dengan yang diinginkan.

3.6. Pengujian Keseluruhan Sistem

Pengujian sistem secara keseluruhan dilakukan dengan tujuan untuk mengetahui unjuk kerja alat setelah perangkat keras dan perangkat lunak diintegrasikan bersama.

BAB IV

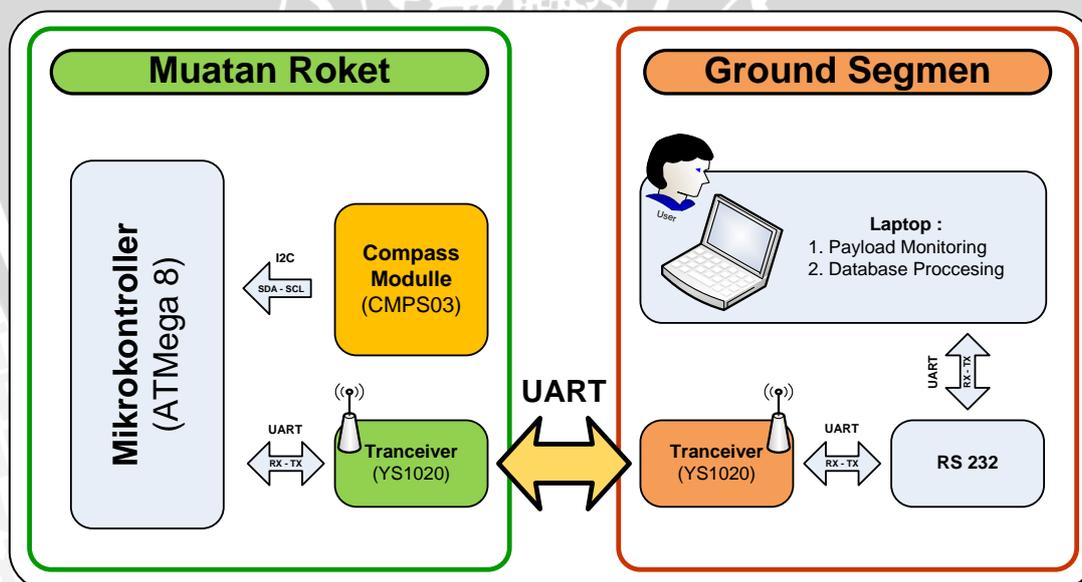
PERANCANGAN DAN PEMBUATAN ALAT

Perancangan alat ini dilakukan secara bertahap dalam bentuk blok sehingga akan memudahkan dalam analisis pada setiap bloknya maupun secara keseluruhan. Perancangan ini terdiri atas:

- Perancangan sistem.
- Perancangan mekanik muatan roket.
- Perancangan perangkat keras (perancangan rangkaian catu daya, rangkaian sensor kompas digital CMPS03, perancangan rangkaian mikrokontroler ATmega8, perancangan pengirim dan penerima RF, rangkaian penyesuai level tegangan (MAX232).
- Perancangan perangkat lunak (perancangan perangkat lunak kompas digital CMPS03, perancangan perangkat lunak *ground segmen*, dan perancangan perangkat lunak utama).

4.1. Perancangan Sistem

Diagram blok sistem yang dirancang ditunjukkan dalam Gambar 4.1.



Gambar 4.1 Diagram Blok Sistem

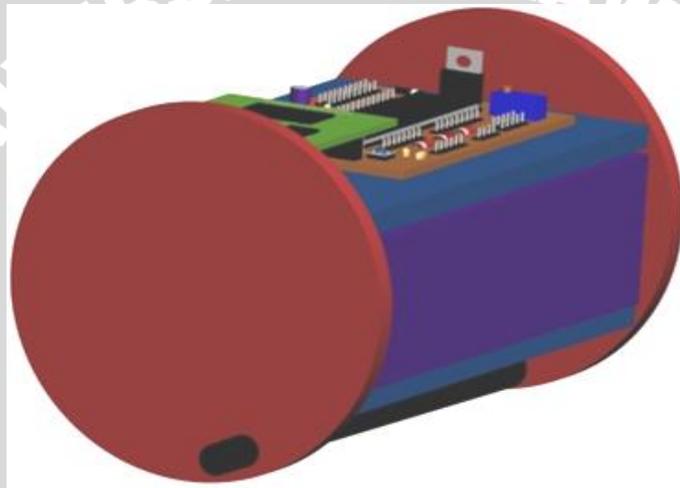
Sensor yang dipakai adalah modul kompas digital CMPS03, digunakan untuk menunjukkan arah dari muatan roket (0° - 359°). Data keluarannya berupa data digital yang kemudian masuk kedalam mikrokontroler ATmega8 dengan menggunakan jalur komunikasi *Inter Integrated Circuit* (I2C).

repository.ub.ac.id

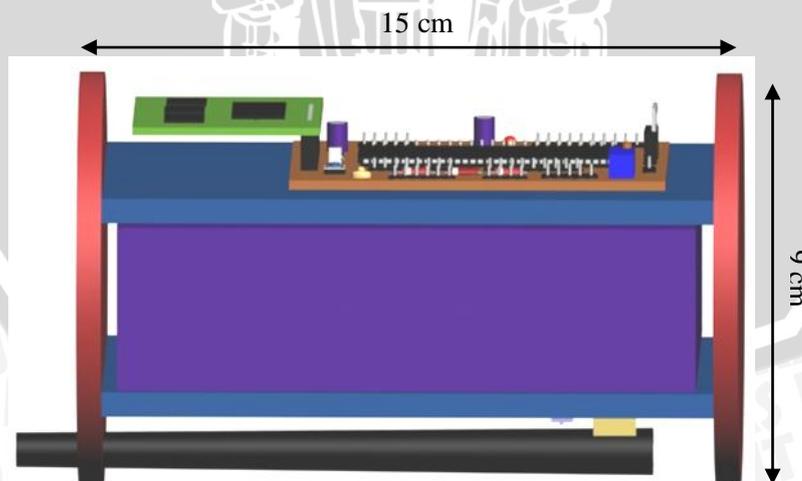
Dalam mikrontroller, dilakukan proses kalibrasi sensor hingga diperoleh nilai data arah kompas yang sebenarnya. Selanjutnya, data akan dikirim melalui RF (Radio Frequency) YS-1020 secara serial ke *ground segmen* dan ditampilkan secara *real time* dalam bentuk visual dan grafik. Data yang diterima *ground segmen* dapat disimpan ke dalam *database*.

4.2. Perancangan Mekanik Muatan Raket

Dalam perancangan mekanik muatan roket ini, bahan yang digunakan adalah mika *Acrylic* dengan tebal 3 mm. Muatan roket berbentuk silinder dengan dimensi yaitu panjang 15 cm dan diameter 9 cm. Gambar rancangan ukuran muatan roket ditunjukkan dalam Gambar 4.2 dan Gambar 4.3.



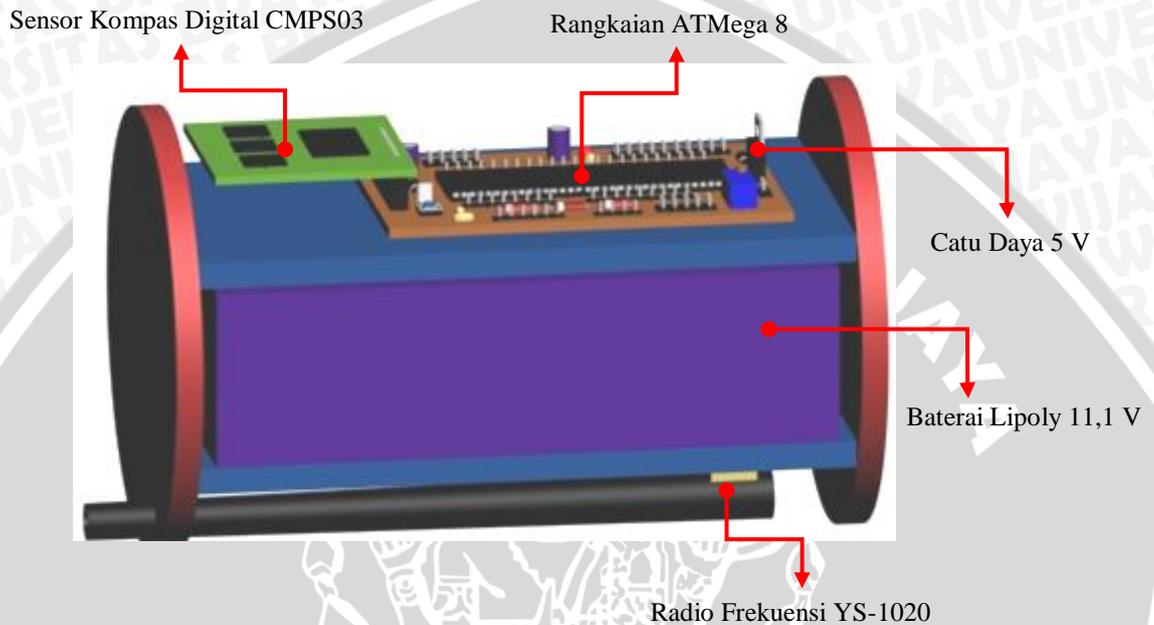
Gambar 4.2 Muatan Raket Tampak Depan



Gambar 4.3 Muatan Raket Tampak Samping

4.3. Perancangan Perangkat Keras

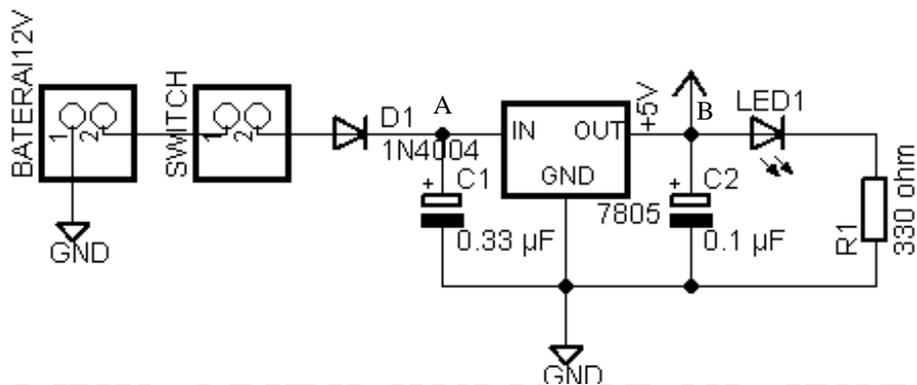
Perancangan dan pembuatan perangkat keras terdiri atas beberapa rangkaian, yaitu rangkaian catu daya, rangkaian antarmuka kompas digital CMPS03, rangkaian mikrokontroler ATmega8, pengirim dan penerima RF, dan rangkaian penyesuai level tegangan. Posisi perangkat keras tersebut pada muatan roket ditunjukkan dalam Gambar 4.4.



Gambar 4.4 Peletakan Perangkat Keras pada Muatan Roket

4.3.1. Perancangan Catu Daya

Pada perancangan catu daya tegangan yang dibutuhkan adalah 5 volt untuk catu mikrokontroler ATmega8, modul kompas digital CMPS03, dan RF YS-1020U. Rangkaian catu daya secara keseluruhan ditunjukkan dalam Gambar 4.5.



Gambar 4.5 Rangkaian Catu Daya 5 V

Catu sumber yang digunakan sebesar 11,1 volt. Dioda D1 berfungsi sebagai pengamanan rangkaian jika terjadi pemasangan polaritas catu terbalik. Sehingga, nilai tegangan pada titik A sebesar 10,4 volt yang perhitungannya ditunjukkan dalam Persamaan (4.1).

$$\begin{aligned} V_A &= 11,1 - V_{D1} ; & V_{D1} &= 0,7 \text{ volt} \\ V_A &= 11,1 - 0,7 \\ V_A &= 10,4 \text{ volt} \end{aligned} \tag{4.1}$$

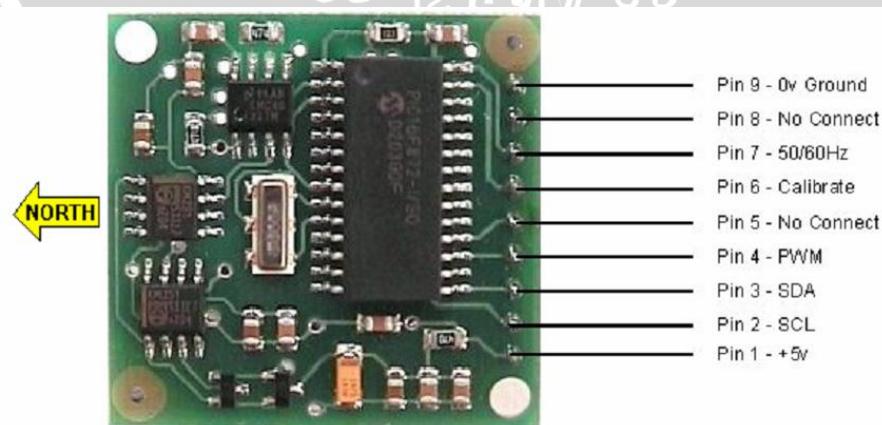
Pada titik B diinginkan tegangan sebesar 5 volt maka digunakanlah regulator LM7805. Pada titik B terdapat LED yang digunakan sebagai indikator catu. Perhitungan nilai resistor yang digunakan ditunjukkan dalam Persamaan (4.2).

$$\begin{aligned} V_B &= 5 \text{ volt} & V_{fLED1} &= 2,2 \text{ volt} & I_{f \max LED1} &= 30 \text{ mA} \\ R_1 &= \frac{V_B - V_{fLED1}}{I_{f \max LED1}} \\ R_1 &= \frac{5 - 2,2}{0,03} = 93,3 \Omega \end{aligned} \tag{4.2}$$

Dikarenakan nilai R_1 adalah nilai minimum dan nilai tersebut tidak terdapat di pasaran maka digunakan R_1 sebesar 330 Ω .

4.3.2. Perancangan Antarmuka Kompas Digital CMPS03

Sensor yang digunakan pada penelitian ini adalah kompas digital CMPS03 yang berfungsi mendeteksi medan magnet bumi pada muatan roket untuk mendeteksi arah muatan roket. Modul sensor kompas ini buatan Devantech dengan ukuran 4 x 4 cm. Bentuk fisik modul kompas digital CMPS03 ditunjukkan dalam Gambar 4.6.

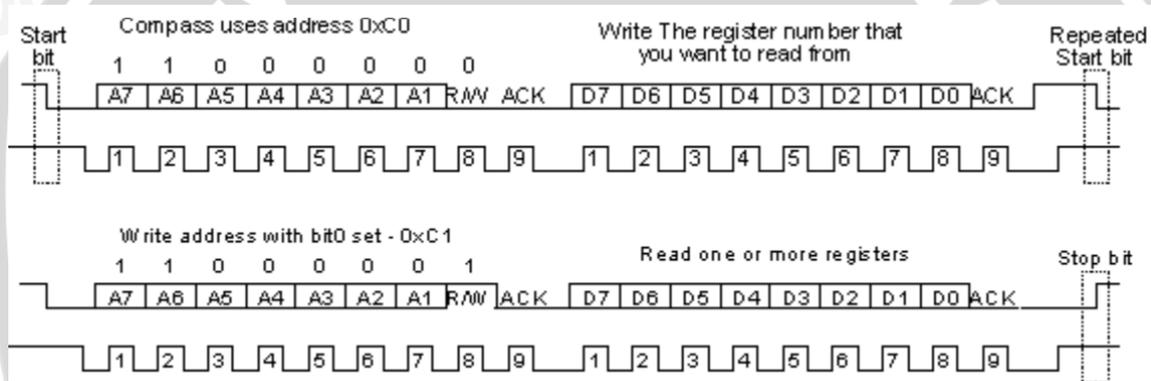


Gambar 4.6 Modul kompas digital CMPS03

Sumber: CMPS03 datasheet, 2008: 1

Kompas digital ini membutuhkan catu daya 5V DC dengan konsumsi arus 15mA. Pada CMPS03, arah mata angin dibagi dalam bentuk derajat, yaitu: Utara (0°), Timur (90°), Selatan (180°), dan Barat (270°). Ada dua cara untuk mendapatkan informasi arah dari modul kompas digital ini yaitu dengan membaca sinyal PWM (*Pulse Width Modulation*) pada pin 4 atau dengan membaca data interface I2C (*Inter Integrated Circuit*) pada pin 2 dan 3. Untuk penelitian kali ini akan menggunakan jalur I2C untuk membaca data kompas digital.

Komunikasi I2C dimulai dengan mengirimkan start bit, address modul kompas digital dengan read/write low (0xC0), kemudian nomor register yang akan dibaca. Selanjutnya diikuti dengan start bit lagi, address digital dengan read/write high (0xC1). Alamat register ini akan menghasilkan 0°-359°. Protokol komunikasi CMPS03 ditunjukkan dalam Gambar 4.7.



Gambar 4.7 Protokol komunikasi I2C
Sumber : CMPS03 Datasheet, 2008:2

Sensor CMPS03 ini memiliki 16 buah register yang masing-masingnya memiliki kapasitas sebesar 1 byte. Fungsi register-register CMPS03 ditunjukkan dalam Tabel 4.1.

Tabel 4.1 Register Sensor Kompas CMPS03

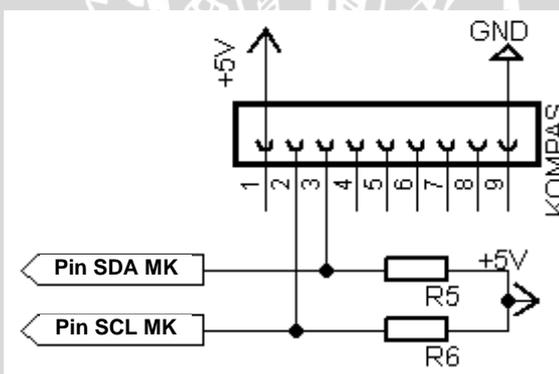
| Register | Fungsi |
|----------|---|
| 0 | Nomor revisi perangkat lunak |
| 1 | Compass bearing dalam byte, 0-255 untuk satu lingkaran penuh (menunjukkan 0°-359°) |
| 2,3 | Compass bearing dalam word, 0-3599 untuk satu lingkaran penuh (menunjukkan 0°-359,9° derajat) |
| 4,5 | Tes internal, sinyal diferensial sensor 1 – 16 bit <i>signed word</i> |
| 6,7 | Tes internal, sinyal diferensial sensor 2 – 16 bit <i>signed word</i> |
| 8,9 | Tes internal, nilai kalibrasi 1 – 16 bit <i>signed word</i> |
| 10,11 | Tes internal, nilai kalibrasi 2 – 16 bit <i>signed word</i> |

Berikut konfigurasi pin-pin sensor yang dipakai ditunjukkan dalam Tabel 4.2.

Tabel 4.2 Konfigurasi pin-pin Sensor Kompas CMPS03

| No. Pin | Nama Pin Sensor | Pin Mikrokontroler |
|---------|-----------------|--------------------|
| 1. | +5V | Vcc |
| 2. | SCL | PC5 |
| 3. | SDA | PC4 |
| 4. | PWM | - |
| 5. | No Connect | - |
| 6. | Calibrate | - |
| 7. | 50/60Hz | - |
| 8. | No Connect | - |
| 9. | Ground | Ground |

Komunikasi I2C membutuhkan 2 jalur, yaitu SDA (*Serial Data Line*) dan SCL (*Serial Clock Line*). Agar komunikasi I2C dapat berfungsi dengan baik maka sangat penting untuk menambahkan sepasang resistor *pull-up* pada masing-masing jalur SDA dan SCL. Resistor *pull-up* memiliki peran penting dalam meminimalisir kondisi *idle*. Skema rangkaian antarmuka CMPS03 ke ATmega8 ditunjukkan dalam Gambar 4.8.



Gambar 4.8 Rangkaian antarmuka CMPS03 ke ATmega8

Nilai resistor *pull-up* dapat dihitung berdasarkan nilai *sink current* dalam keadaan *low level output voltage*. Berdasarkan PHILIPS I2C *reference* diketahui bahwa untuk *sink current* sebesar 3mA dan nilai *low level output voltage* nya sebesar 0,4V. Nilai resistor *pull-up* dapat dihitung berdasarkan Persamaan (4.3).

$$R_p = \frac{V_{dd} - V_{OL}}{I_{sink}} \tag{4.3}$$

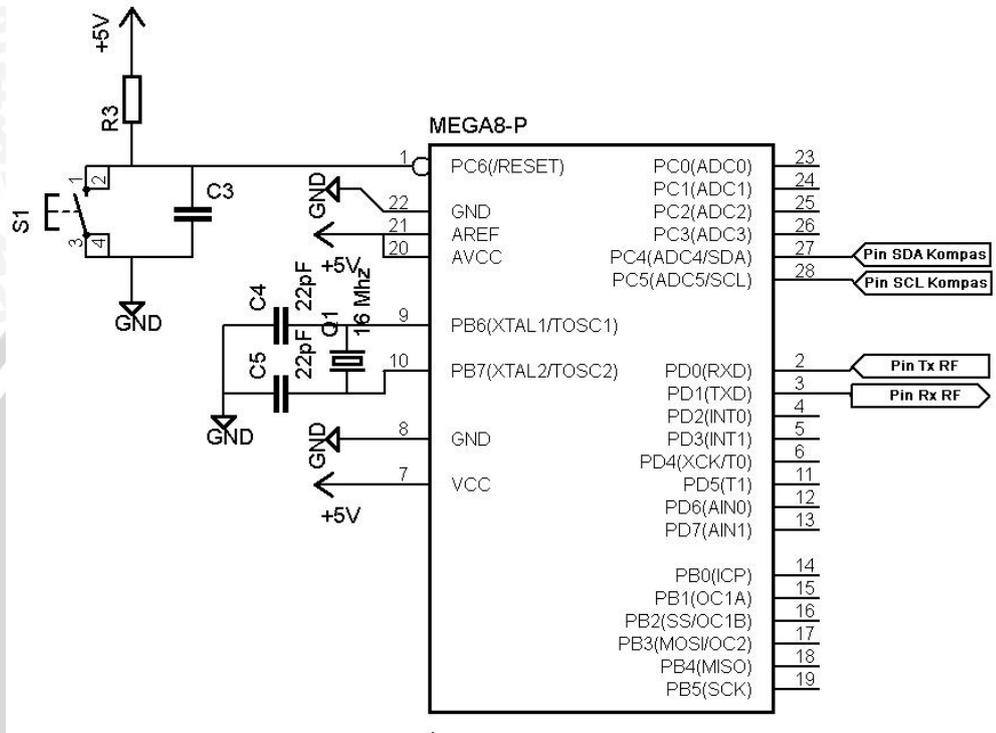
$$R_p = \frac{5V - 0,4V}{3mA} = 1533\Omega \approx 1,8k\Omega$$

Nilai resistor *pull-up* yang digunakan R5 dan R6 sebesar 1,8kΩ.



4.3.3. Perancangan Rangkaian Mikrokontroler ATmega 8

Mikrokontroler yang digunakan dalam alat ini adalah Atmega8. Mikrokontroler ini dirancang untuk melakukan pemrosesan dan pengolahan data dari sensor kompas digital CMPS03 kemudian dikirim ke *ground segmen* melalui Radio Frekuensi YS-1020. Rangkaian minimum sistem mikrokontroler ditunjukkan dalam Gambar 4.9.



Gambar 4.9 Rangkaian antarmuka CMPS03 ke ATmega8

Mikrokontroler ATmega8 terdiri atas tiga port I/O yang dapat diprogram dengan masing-masing port memiliki jumlah data 8 bit. Pada perancangan ini pin-pin yang digunakan ditunjukkan dalam Tabel 4.3.

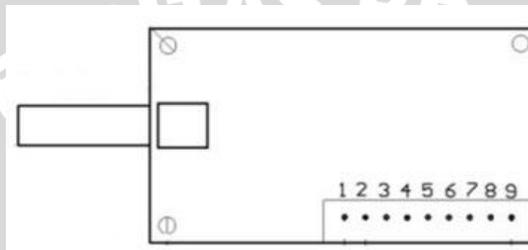
Tabel 4.3 Pin-pin yang digunakan pada mikrokontroler ATmega8

| No. | Fungsional | Pin Mikrokontroler | No. Pin | |
|-----|---|--------------------|---------|----|
| 1. | Jalur Reset | PC6 | 1 | |
| 2. | Jalur Kristal 16 MHz | PB6, PB7 | 9, 10 | |
| 3. | Jalur kompas digital CMPS03 | - SDA | PC4 | 27 |
| | | - SCL | PC5 | 28 |
| 4. | Jalur data UART (<i>input/output</i> menuju RF YS-1020U) | - Tx | PD1 | 3 |
| | | - Rx | PD0 | 2 |

4.3.4. Perancangan Pengirim dan Penerima Radio Frekuensi

4.3.4.1 Perancangan Pengirim Radio Frekuensi

Perancangan pengirim Radio Frekuensi menggunakan modul RF YS-1020U (*transmitter*) yang dipasang pada muatan roket. YS-1020U digunakan untuk komunikasi data antara muatan roket dengan *ground segmen*. Frekuensi kerja antara pengirim dan penerima di set sama yaitu 433,0325 MHz. RF ini menggunakan tegangan catu daya DC sebesar 5 volt. Agar waktu transmisi pengiriman data lebih cepat maka dipilih *baudrate* sebesar 9600 bps. Baudrate 9600 bps dirancang agar lamanya proses transmisi data tidak mempengaruhi proses pemantauan data pada komputer. Konfigurasi pin-pin pengirim RF YS-1020U ditunjukkan dalam Gambar 4.10 dan Tabel 4.4.



Gambar 4.10 Konfigurasi pin pemancar RF

Sumber : ShenZhen YiShi Electronic Technology Development, 2003 : 2

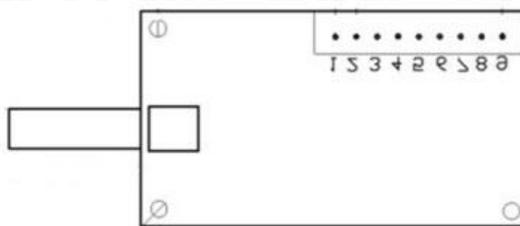
Tabel 4.4 Tabel konfigurasi pin yang dipakai

| No. Pin RF | Nama Pin RF | Pin Mikrokontroler |
|------------|-------------|--------------------|
| 1. | Ground | Ground |
| 2. | Vcc | Vcc |
| 3. | Rx (TTL) | Tx (TTL) |
| 4. | Tx (TTL) | Rx (TTL) |

4.3.4.2 Perancangan Penerima Radio Frekuensi

Perancangan penerima RF menggunakan modul RF YS-1020U (*receiver*) yang dipasang pada *ground segmen*. Data yang dikirim oleh bagian pengirim menggunakan modul RF YS-1020U (*transmitter*) akan didemodulasi 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 RF YS-1020U (*receiver*) di set frekuensi kerja yang sama antara penerima dan pengirim dengan frekuensi 433,0325 MHz. RF ini menggunakan tegangan catu daya DC sebesar 5 volt. Konfigurasi pin-pin penerima RF YS-1020U ditunjukkan dalam Gambar 4.11 dan Tabel 4.5.



Gambar 4.11 Nomor pin penerima RF

Tabel 4.5 Tabel konfigurasi pin yang dipakai

| No. Pin RF | Nama Pin RF | Pin Rangkaian Penyesuai Level Tegangan |
|------------|-------------|--|
| 1. | Ground | Ground |
| 2. | Vcc | Vcc |
| 3. | Rx (TTL) | Rx (TTL) |
| 4. | Tx (TTL) | Tx (TTL) |

4.3.5. Rangkaian Penyesuai Level Tegangan (MAX232)

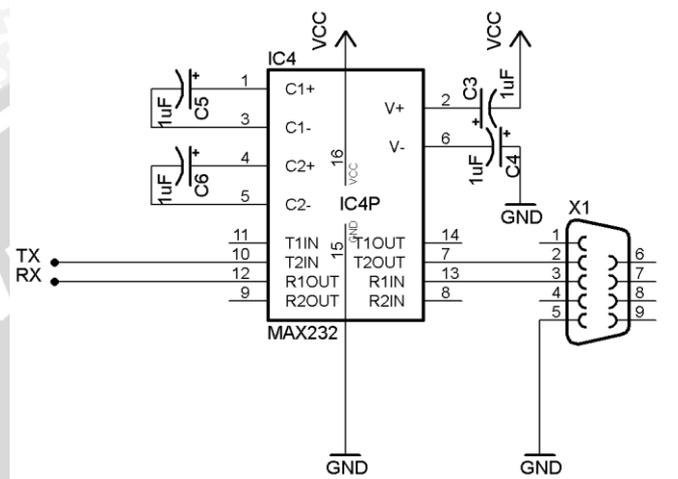
Data yang keluar masuk port data dari rangkaian penerima RF menggunakan level tegangan CMOS. Sedangkan keluaran atau masuk dari PC menggunakan standar komunikasi serial RS-232, level tegangan yang digunakan adalah level tegangan yang berkisar antara -3 volt dan -15 volt untuk kondisi logika “1” atau 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 1uF, nilai ini merupakan konfigurasi *datasheet* MAX232. Gambar 4.12 menunjukkan rangkaian MAX232. Lima kapasitor dalam rangkaian ini digunakan pada:

- C₁ terhubung pada pin 1 (+) dan pin 3(-), sebagai pengganda tegangan (+5 volt menjadi +10 volt). keluaran hasil proses pengganda tegangan ada di pin 2.
- C₂ terhubung pada pin 4(+) dengan pin 5(-), sebagai pembalik tegangan (+10 volt menjadi -10 volt), keluaran ada di pin 6.

- C₃ terhubung pada pin 6 (-) dan dengan *Ground* (+), untuk penstabil nilai arus yang mengalir.
- C₄ terhubung pada pin 2(+) dengan pin 16(-), untuk penstabil nilai arus yang mengalir.
- C₅ terhubung pada pin 16(+) dengan *Ground*(-), untuk penstabil nilai arus yang mengalir.



Gambar 4.12 Rangkaian Penyesuai Level Tegangan (MAX232)

4.4. Perancangan Perangkat Lunak

Perancangan perangkat lunak terdiri atas perancangan perangkat lunak kompas digital CMPS03, perancangan perangkat lunak *ground segmen*, dan perancangan perangkat lunak keseluruhan. Pemrograman dilakukan menggunakan *software* Code Vision AVR dengan memakai bahasa pemrograman C dan menggunakan Delphi7 dengan memakai bahasa pemrograman *pascal*.

4.4.1 Perancangan Perangkat Lunak Kompas Digital CMPS03

Ada dua cara untuk mengakses kompas digital CMPS03, melalui bus I2C dan PWM. Dalam penelitian ini, akan digunakan jalur I2C untuk mengakses kompas ini. Dalam mengakses kompas digital CMPS03 dirancang menggunakan *file header* "i2c.h".

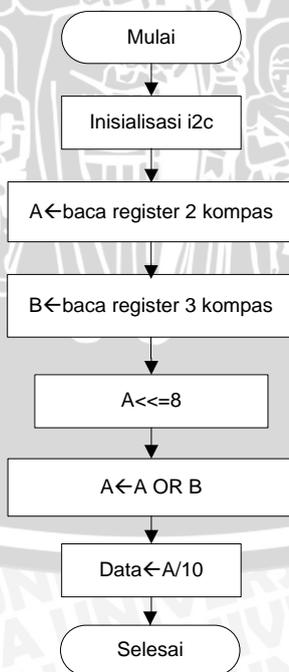
Perancangan perangkat lunak kompas digunakan untuk mengakses modul kompas sehingga dapat diolah data arah kompas. Terdapat dua cara pengolahan data, yaitu menggunakan *register* 1 atau gabungan *register* 2 dan 3. Data yang dihasilkan *register* 1 adalah 0-255 pada putaran penuh dengan resolusi 1,4 derajat. Sedangkan pada *register* 2 dan 3 dihasilkan data 0-3599 untuk satu putaran penuh yang merepresentasikan sudut 0°-359,9° dengan resolusi 0,1 derajat.

Pada penelitian ini digunakan *register 2* dan *3* untuk mengakses kompas. Register 2 untuk 8 bit teratas (MSB) dan register 3 untuk 8 bit terendah (LSB).

Perangkat lunak untuk mengolah dan mengakses kompas sebagai berikut:

```
#include <i2c.h>
#asm
    .equ __i2c_port=0x15 ;PORTC
    .equ __sda_bit=1
    .equ __scl_bit=0
#endasm
.....
    CMPSA= Akses_kompas(2);
    CMPSB= Akses_kompas(3);
    CMPSA= CMPSA<<8;
    CMPSA= CMPSA | CMPSB;
    DATA_CMPS= CMPSA/10;
.....
unsigned char Akses_kompas(char alamat)
{
    unsigned char Value;
    i2c_start();
    i2c_write(write);
    i2c_write(alamat);
    i2c_start();
    i2c_write(read);
    Value=i2c_read(0);
    i2c_stop();
    return Value;
}
```

Diagram alir akses kompas ditunjukkan dalam Gambar 4.13.

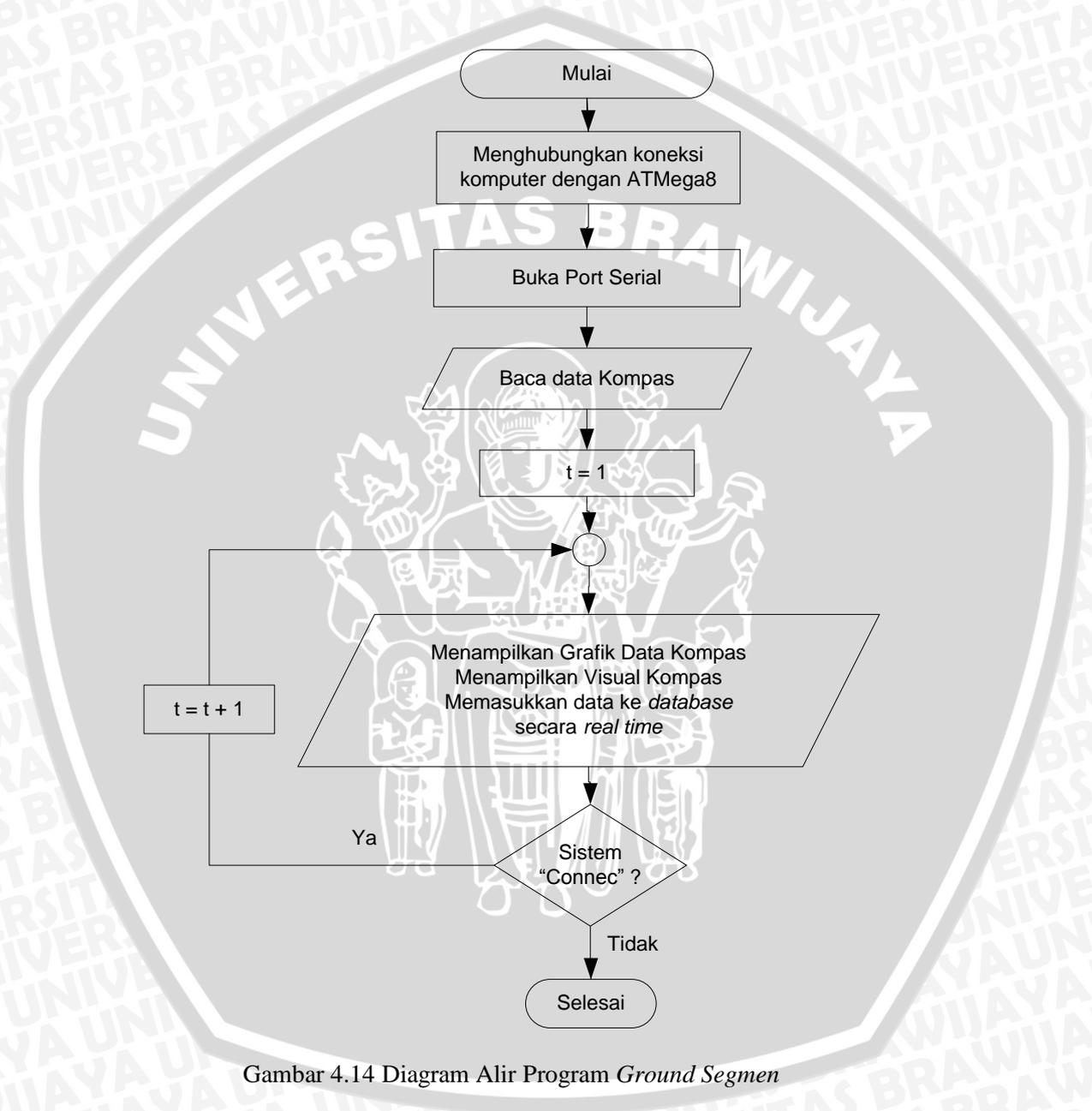


Gambar 4.13 Diagram Alir Perangkat Lunak Akses Kompas



4.4.2 Perancangan Perangkat Lunak *Ground Segmen*

Perancangan *ground segmen* ini merupakan program untuk menampilkan hasil keluaran akhir dari keseluruhan proses data sistem, yaitu berupa tampilan visual, grafik, dan *database* dengan menggunakan Delphi7. Diagram alir program ditunjukkan dalam Gambar 4.14.



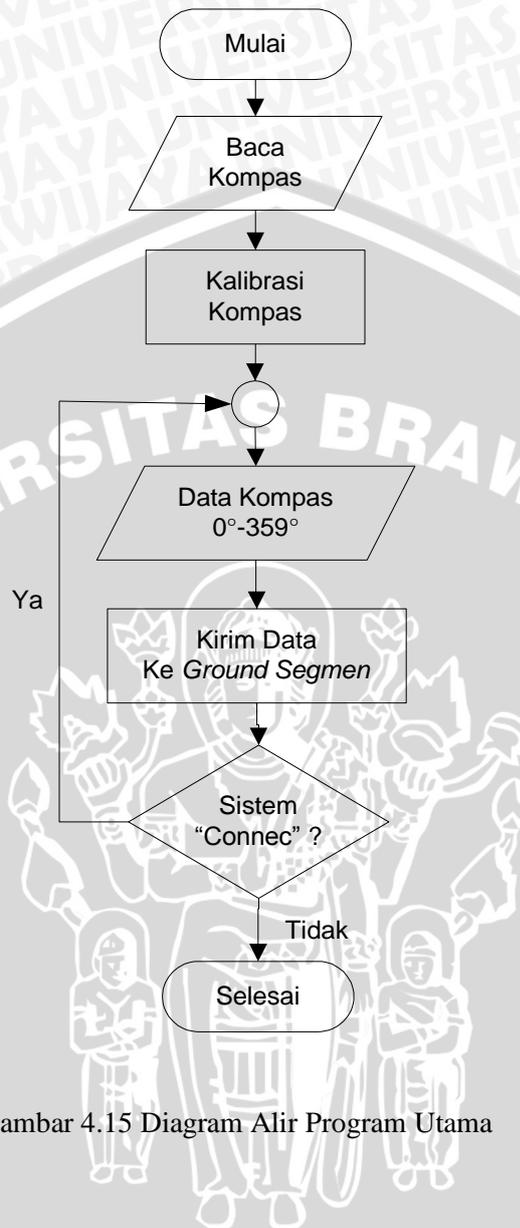
Gambar 4.14 Diagram Alir Program *Ground Segmen*

4.4.3 Perancangan Perangkat Lunak Utama

Perancangan perangkat lunak utama merupakan integrasi dari seluruh program pada mikrokontroller sebagai sistem kendali utama alat dengan menggunakan *software* Code Vision AVR. Keluaran dari program ini merupakan data jadi yang kemudian

repository.ub.ac.id

ditampilkan dalam bentuk visual dan grafik di laptop secara *real time* dengan waktu cuplik 100 ms. Diagram alir program utama ditunjukkan dalam Gambar 4.15.



Gambar 4.15 Diagram Alir Program Utama

BAB V

PENGUJIAN DAN ANALISIS

Pengujian dan analisis dilakukan untuk menganalisis alat yang telah dirancang dan direalisasikan apakah telah bekerja sesuai perancangan. Pengujian dilakukan tiap-tiap blok dengan tujuan untuk mengamati apakah tiap blok rangkaian sudah sesuai dengan perancangan, baru kemudian dilanjutkan dengan pengujian secara keseluruhan sistem. Adapun pengujian yang perlu dilakukan sebagai berikut:

- 1) Pengujian rangkaian catu daya.
- 2) Pengujian sensor kompas digital CMPS03
- 3) Pengujian pemancar dan penerima pada Radio Frekuensi YS-1020U
- 4) Pengujian *software* antarmuka *ground segmen*
- 5) Pengujian sistem secara keseluruhan

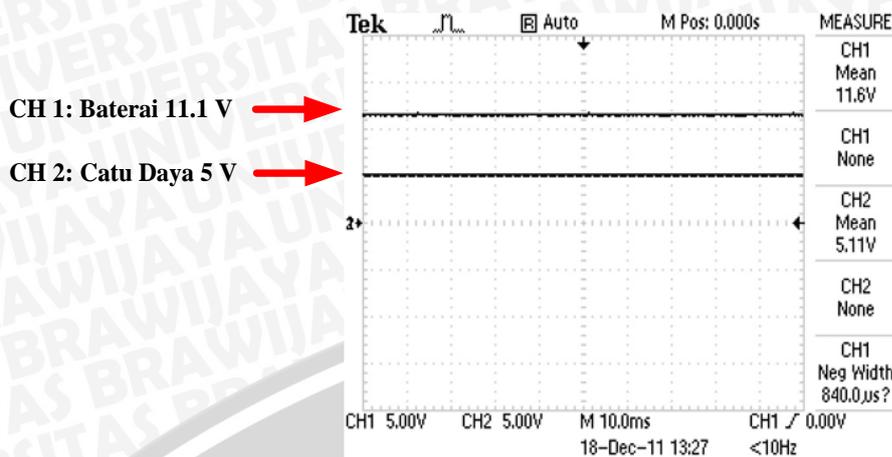
5.1. Pengujian Rangkaian Catu Daya

Pengujian rangkaian catu daya bertujuan untuk mengetahui kesesuaian tegangan keluaran dari catu daya. Pengujian ini diperlukan karena dari sumber tegangan baterai Lipoly 11,1 V DC akan digunakan untuk masukan rangkaian catu daya 5 V. Rangkaian ini akan digunakan untuk catu daya mikrokontroller ATmega8, kompas digital CMPS03, dan radio frekuensi YS-1020U. Pengujian dilakukan dengan cara menghubungkan masukan dan keluaran dari rangkaian catu daya dengan osiloskop untuk mengetahui nilai tegangannya. Diagram blok pengujian rangkaian catu daya 5 ditunjukkan dalam Gambar 5.1.



Gambar 5.1 Blok Diagram Pengujian Rangkaian Catu Daya 5 V

Pengujian rangkaian catu daya 5 V dilakukan dengan menggunakan osiloskop TEKTRONIX TDS-1012B. Channel 1 osiloskop dihubungkan dengan baterai Lipoly 11,1 V, sedangkan channel 2 osiloskop dihubungkan dengan keluaran rangkaian regulator LM7805. Hasil pengujian tegangan masukan dan tegangan keluaran pada rangkaian catu daya 5 V ditunjukkan dalam Gambar 5.2.



Gambar 5.2 Tegangan Masukan dan Keluaran pada Rangkaian Catu Daya 5 V

Dari hasil pengujian diperoleh nilai tegangan keluaran sebesar 5,11 V. Nilai tersebut adalah nilai tegangan tanpa beban, yang berarti rangkaian catu daya 5 V dapat menghasilkan tegangan maksimum (tanpa beban) sebesar 5,11 V. Namun, ketika rangkaian dibebani maka tegangan menjadi 5,03 V.

5.2. Pengujian Sensor Kompas Digital CMPS03

Pengujian sensor kompas digital CMPS03 bertujuan untuk mengetahui kemampuan serta ketepatan sensor dalam mengukur arah muatan roket. Pengujian ini dilakukan sejajar bidang datar dengan membandingkan hasil pembacaan arah mata angin oleh sensor CMPS03 terhadap pembacaan arah mata angin oleh perangkat lain yaitu kompas analog. Diagram blok pengujian sensor kompas digital CMPS03 ditunjukkan dalam Gambar 5.3.



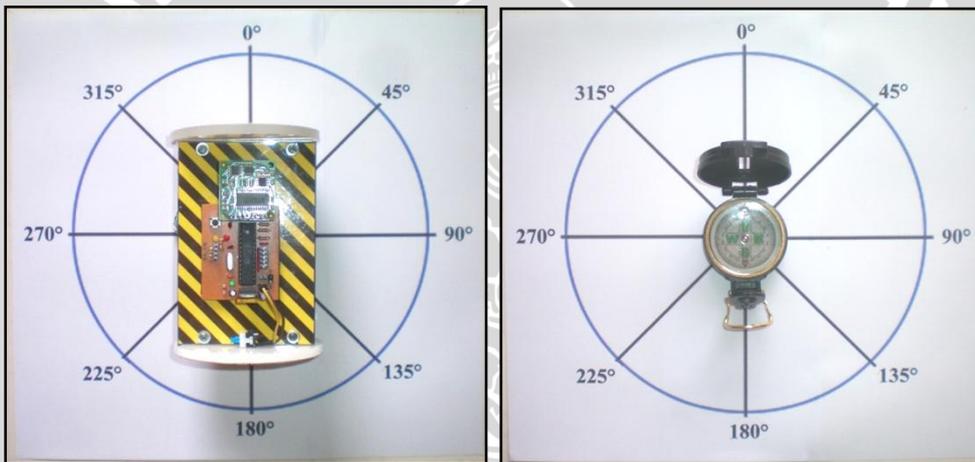
Gambar 5.3 Diagram Blok Pengujian Sensor Kompas Digital CMPS03

Dalam pengujian CMPS03 ini digunakan register kompas alamat 0x02 dan 0x03. Register ini kemudian digabungkan untuk menghasilkan data 0-3599 yang merepresentasikan 0-359,9°. Namun dalam penelitian ini data hanya memiliki resolusi 1° sesuai dengan rule KOMURINDO. Pengujian dilakukan dengan menggunakan sensor kompas digital CMPS03 sebagai pengkonversi besaran fisik arah mata angin menjadi besaran elektrik yang dapat dibaca oleh mikrokontroler ATMega8. ATMega8 sebagai unit pembaca serta pengolah data arah yang diberikan oleh sensor CMPS03 dan *ground segmen* untuk memonitor data arah dalam bentuk visual dan grafik.

Hasil yang diharapkan adalah data arah yang diberikan oleh sensor kompas digital CMPS03 memiliki nilai kesalahan (*error* dalam %) yang minimum ketika dibandingkan dengan hasil pembacaan arah dari kompas analog. Perhitungan nilai error pembacaan kompas ditunjukkan dalam Persamaan (5.1).

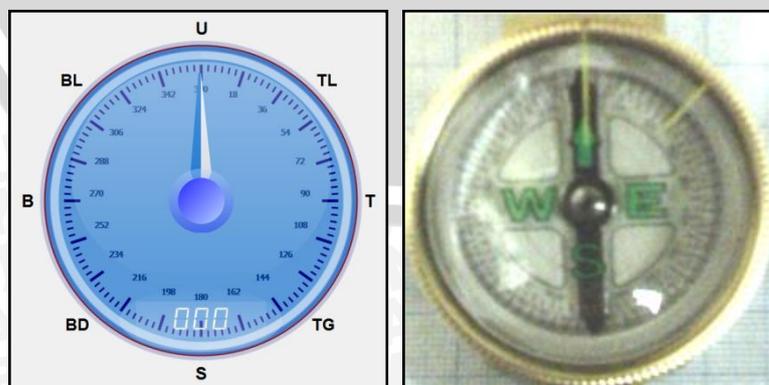
$$\% \text{ error} = \frac{|\text{nilai arah kompas digital} - \text{nilai arah kompas analog}|}{\text{nilai arah kompas analog}} \times 100\% \quad (5.1)$$

Pengujian kompas dilakukan dengan prosedur meletakkan muatan roket yang sudah terpasang sensor kompas pada sebuah karton berbentuk lingkaran yang sudah dilengkapi keterangan tiap sudutnya. Kompas diputar perlahan dengan sudut tertentu kemudian mengamati data hasil pengujian pada *software* antarmuka. Foto pengujian sensor kompas digital CMPS03 dan kompas analog ditunjukkan dalam Gambar 5.4.

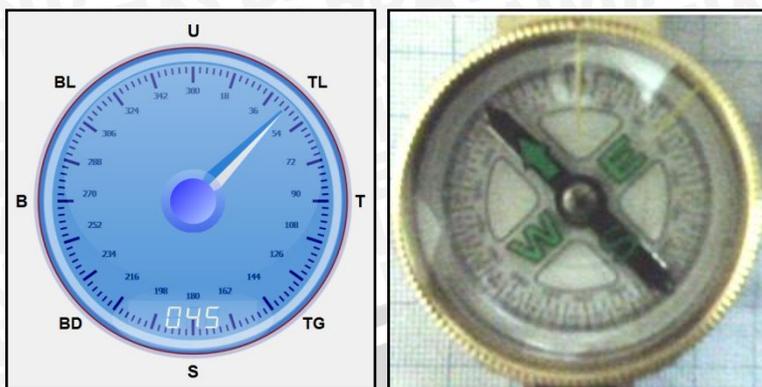


Gambar 5.4 Pengujian Sensor CMPS03 dibandingkan dengan Kompas Analog

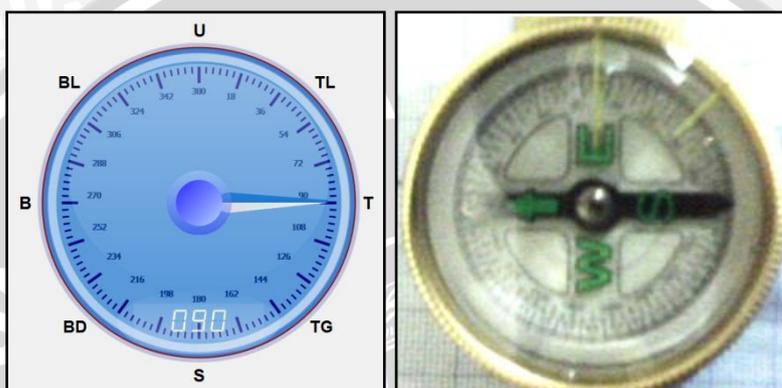
Hasil pembacaan kompas digital CMPS03 ditunjukkan dalam Gambar 5.5 sampai Gambar 5.12.



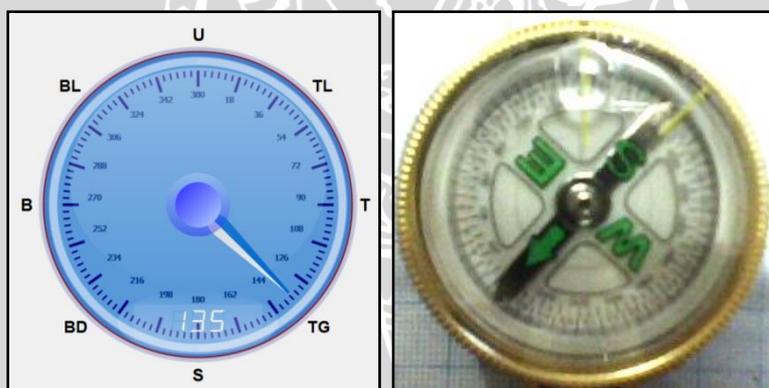
Gambar 5.5 Pembacaan Sensor CMPS03 dan Kompas Analog Arah Utara (0°)



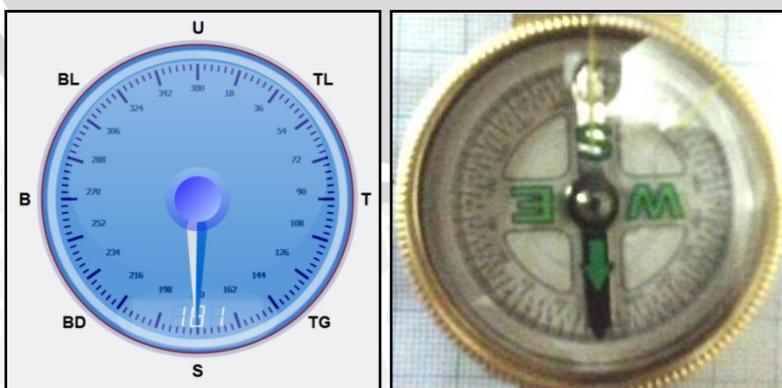
Gambar 5.6 Pembacaan Sensor CMPS03 dan Kompas Analog Arah Timur Laut (45°)



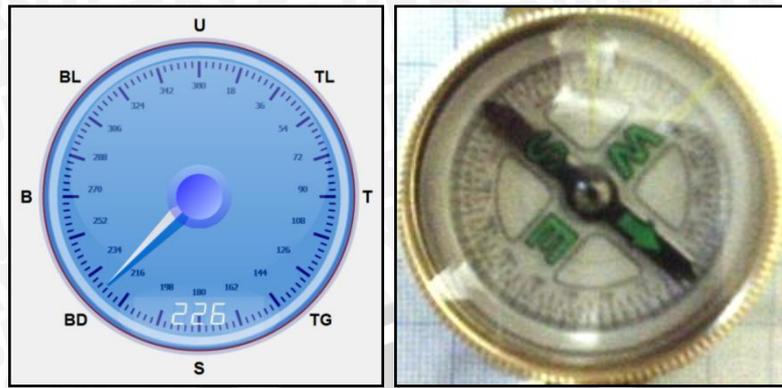
Gambar 5.7 Pembacaan Sensor CMPS03 dan Kompas Analog Arah Timur (90°)



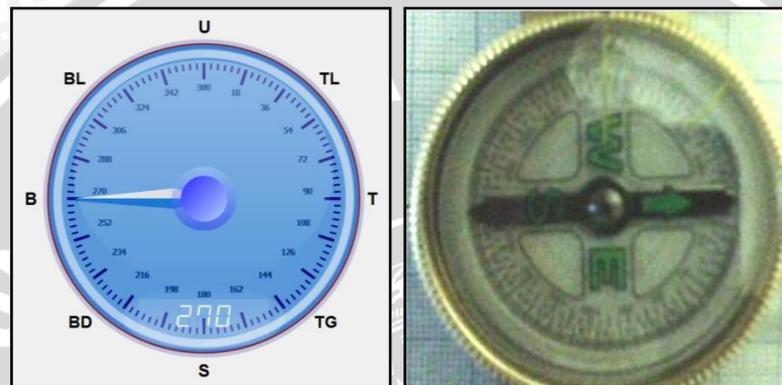
Gambar 5.8 Pembacaan Sensor CMPS03 dan Kompas Analog Arah Tenggara (135°)



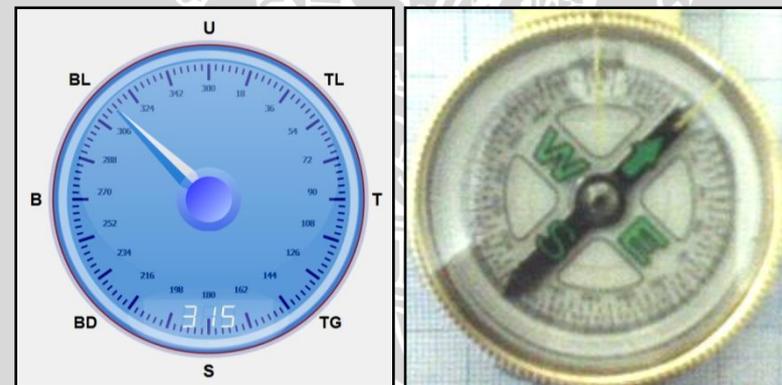
Gambar 5.9 Pembacaan Sensor CMPS03 dan Kompas Analog Arah Selatan (180°)



Gambar 5.10 Pembacaan Sensor CMPS03 dan Kompas Analog Arah Barat Daya (225°)



Gambar 5.11 Pembacaan Sensor CMPS03 dan Kompas Analog Arah Barat (270°)



Gambar 5.12 Pembacaan Sensor CMPS03 dan Kompas Analog Arah Barat Laut (315°)

Pengujian ini dilakukan sebanyak dua kali, yaitu pengujian pertama dengan meletakkan muatan roket ketika sejajar bidang datar dan pengujian kedua muatan roket dengan keadaan miring 45° terhadap bidang datar. Pengujian dilakukan dengan pengambilan data arah utara, timur laut, timur, tenggara, selatan, barat daya, barat, dan barat laut. Hasil pengujian pertama sensor kompas digital CMPS03 dibandingkan kompas analog ketika sejajar dengan bidang datar ditunjukkan dalam Tabel 5.1.

Tabel 5.1 Hasil Pengujian Arah Sensor Kompas Digital dibandingkan Kompas Analog Ketika Sejajar dengan Bidang Datar

| No. | Arah | Pembacaan Arah Kompas Digital (°) | Pembacaan Arah Kompas Analog (°) | Error (%) |
|----------------------------|------------|-----------------------------------|----------------------------------|-------------|
| 1. | Utara | 0 | 0 | 0 |
| 2. | Timur Laut | 45 | 45 | 0 |
| 3. | Timur | 90 | 90 | 0 |
| 4. | Tenggara | 135 | 135 | 0 |
| 5. | Selatan | 181 | 180 | 0,55 |
| 6. | Barat Daya | 226 | 225 | 0,55 |
| 7. | Barat | 270 | 270 | 0 |
| 8. | Barat Laut | 315 | 315 | 0 |
| Rata-Rata Kesalahan | | | | 0,14 |

Hasil pengujian arah oleh sensor kompas CMPS03 pada Tabel 5.1 menunjukkan bahwa rata-rata kesalahan pada pengukuran arah sebesar 0,14% ketika sensor kompas dalam keadaan sejajar dengan bidang datar.

Pengujian kedua muatan roket dalam keadaan miring 45° terhadap bidang datar. Hasil pengujian sensor kompas kompas digital CMPS03 ditunjukkan dalam Tabel 5.2.

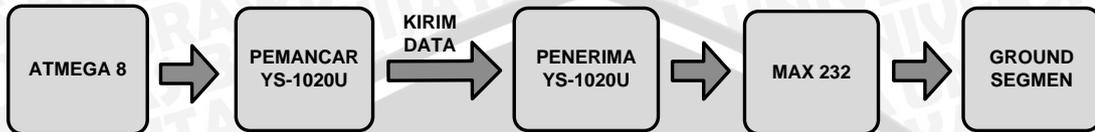
Tabel 5.2 Hasil Pengujian Arah Sensor Kompas Digital ketika Muatan Roket Miring 45° terhadap bidang datar

| No. | Arah | Arah (°) | Data Arah Kompas Digital Miring 45° (°) | Error (%) |
|----------------------------|------------|----------|---|------------|
| 1. | Utara | 0 | 3 | - |
| 2. | Timur Laut | 45 | 48 | 6,7 |
| 3. | Timur | 90 | 94 | 4,4 |
| 4. | Tenggara | 135 | 139 | 2,9 |
| 5. | Selatan | 180 | 183 | 1,7 |
| 6. | Barat Daya | 225 | 228 | 1,3 |
| 7. | Barat | 270 | 275 | 1,8 |
| 8. | Barat Laut | 315 | 318 | 0,9 |
| Rata-Rata Kesalahan | | | | 2,5 |

Hasil pengujian arah oleh sensor kompas CMPS03 ketika dalam keadaan miring 45° pada Tabel 5.2 menunjukkan bahwa rata-rata kesalahan pada pengukuran arah sebesar 2,5%. Jadi, dari kedua pengujian yang telah dilakukan dapat disimpulkan bahwa data arah kompas digital CMPS03 pada muatan roket memiliki kesalahan rata-rata 0,14% terhadap bidang datar. Namun, ketika sensor kompas CMPS03 dalam keadaan semakin miring terhadap bidang datar maka data arah kompas memiliki persentase *error* semakin besar yang akan menyebabkan data arah tidak valid.

5.3. Pengujian Pemancar dan Penerima pada Radio Frekuensi YS-1020U

Pengujian ini bertujuan untuk menganalisis keberhasilan transmisi data antara pemancar dan penerima dengan menggunakan Radio Frekuensi YS-1020U. Diagram blok pengujian pemancar dan penerima pada radio frekuensi YS-1020U ditunjukkan dalam Gambar 5.13.

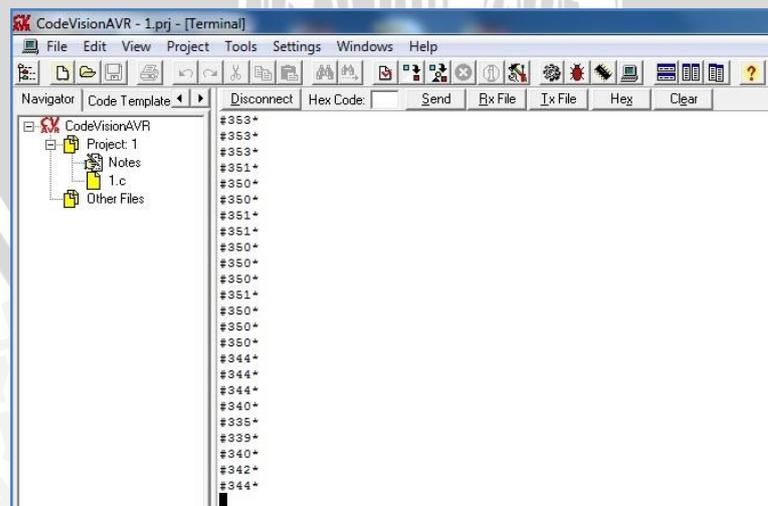


Gambar 5.13 Diagram Blok Pengujian Radio Frekuensi YS-1020U

Pada pengujian ini terlebih dahulu membuat program mikrokontroler ATmega8 untuk mengirim data arah dengan format “#datakompas*” dengan kecepatan transmisi 9600 bps. Kemudian mengamati tampilan data yang terkirim di *ground segmen* menggunakan *software Terminal* pada program *Code Vision AVR*.

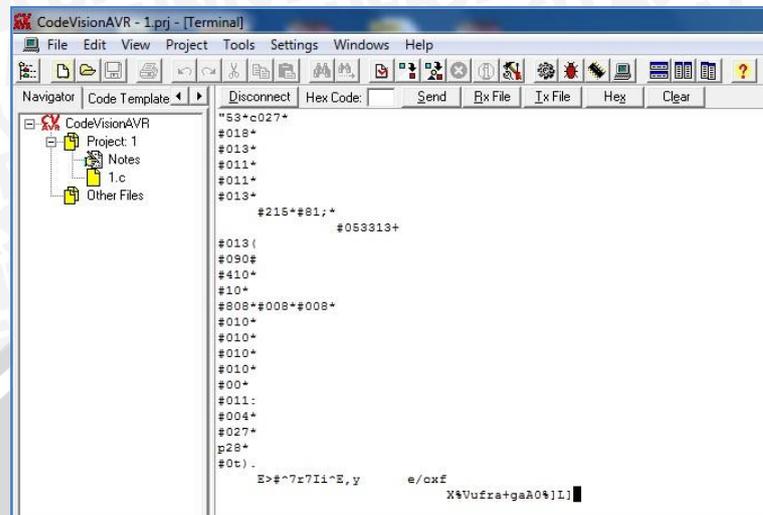
Pemancar dikondisikan terletak semakin lama semakin menjauhi penerima, kemudian menganalisis berapa batas maksimum jarak pengiriman data hingga terjadi kerusakan data pada penerima.

Hasil pengujian transmisi data pada pemancar dan penerima YS-1020U didapatkan bahwa data dapat dikirim dengan baik sesuai dengan format pengiriman data ketika jarak dari pemancar dan penerima hingga 250 meter. Hasil pengujian ditunjukkan dalam Gambar 5.14.



Gambar 5.14 Transmisi Data YS-1020U hingga Jarak 250 m

Hasil pengujian dengan jarak lebih dari 250 meter, data yang terkirim sudah tidak mengikuti format data yang dikirim. Hasil pengujian dengan jarak lebih dari 250 ditunjukkan dalam Gambar 5.15.

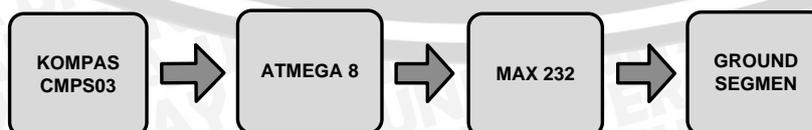


Gambar 5.15 Transmisi Data YS-1020U dengan Jarak Lebih dari 250 m

Berdasarkan hasil pengujian yang telah dilakukan, dapat dianalisis bahwa data dapat dikirim dengan baik tanpa ada karakter yang hilang dengan jarak maksimum 250 meter. Pengiriman data hingga terjadi kerusakan data terjadi pada jarak lebih dari 250 meter dari penerima.

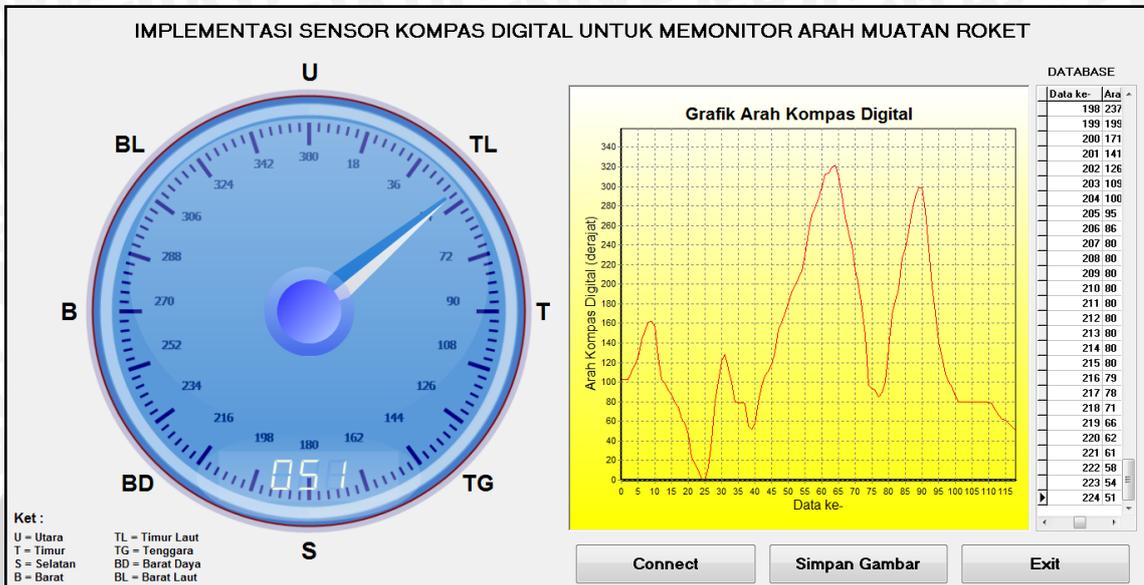
5.4. Pengujian *software* antarmuka *Ground Segmen*

Pengujian *software* antarmuka menggunakan aplikasi Delphi7. Pengujian ini dilakukan dengan tujuan untuk menganalisis kinerja aplikasi apakah sudah dapat menampilkan visual kompas, grafik, dan *database* secara *real time*. Pada pengujian ini mikrokontroler ATmega8 menerima data arah dari sensor kompas CMPS03. Kemudian data dikirim oleh mikrokontroler ATmega 8 melalui rangkaian MAX 232 untuk diterima dan ditampilkan pada *ground segmen*. Diagram blok pengujian *software* antarmuka *ground segmen* ditunjukkan dalam Gambar 5.16.



Gambar 5.16 Diagram Blok Pengujian *Software* Antarmuka *Ground Segmen*

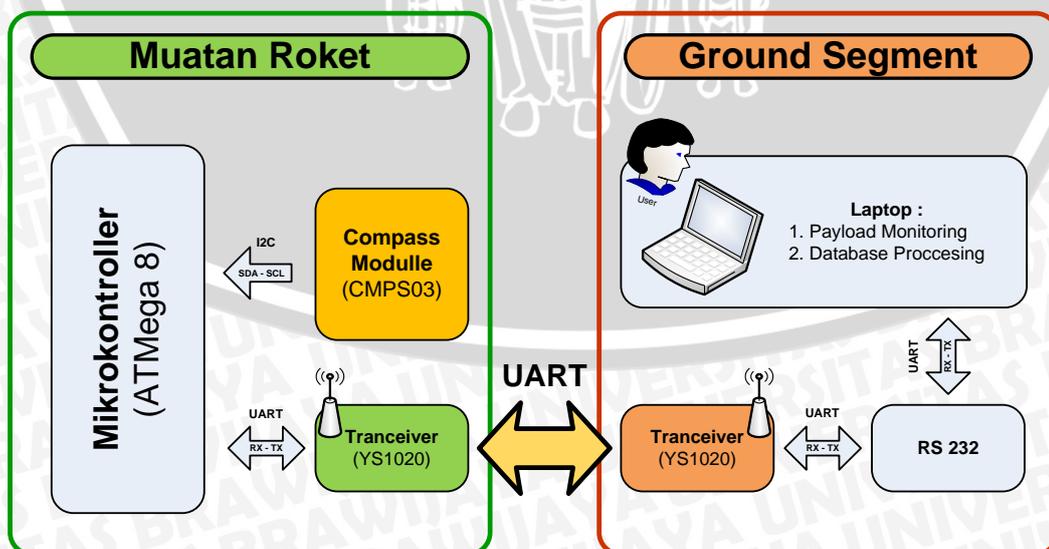
Dari hasil pengujian ini, dapat dianalisis bahwa program aplikasi komputer dapat berjalan dengan baik, yaitu dapat menampilkan visual kompas, grafik, dan database secara *real time*. Bentuk tampilan *software* antarmuka ditunjukkan dalam Gambar 5.17.



Gambar 5.17 Bentuk Tampilan Pengujian *Software* Antarmuka *Ground Segment*

5.5. Pengujian Sistem secara Keseluruhan

Pengujian sistem secara keseluruhan bertujuan untuk mengetahui kinerja sistem setelah seluruh bagian-bagian penyusun sistem digabungkan menjadi satu kesatuan utuh. Diagram blok pengujian sistem secara keseluruhan ditunjukkan Gambar 5.18.



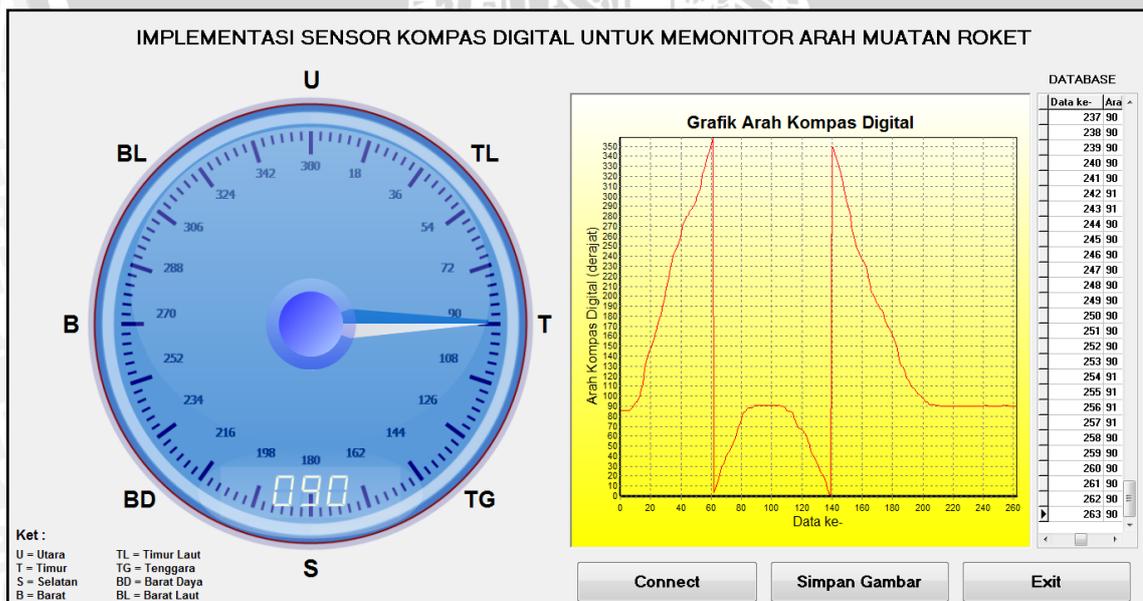
Gambar 5.18 Diagram Blok Sistem Secara Keseluruhan

Pengujian dilakukan dengan membuat program pada mikrokontroler ATmega8 untuk membaca data sensor kompas digital CMPS03. Kemudian, data arah yang masuk diterima oleh mikrokontroler ATmega8. Setelah data diterima maka data dikirim ke *ground segmen* melalui radio frekuensi YS-1020U. Pengiriman data dikirim tiap 100ms.

Hasil yang diharapkan adalah *ground segmen* dapat menerima data arah sehingga *ground segmen* dapat memonitor arah muatan roket dalam bentuk visual, grafik, dan database secara *real time*. Foto pengujian muatan roket yang sedang digantung, tampilan *software* antarmuka pada *ground segmen*, tampilan grafik arah muatan roket, dan database ditunjukkan dalam Gambar 5.19 sampai Gambar 5.23.



Gambar 5.19 Foto Pengujian Muatan Roket

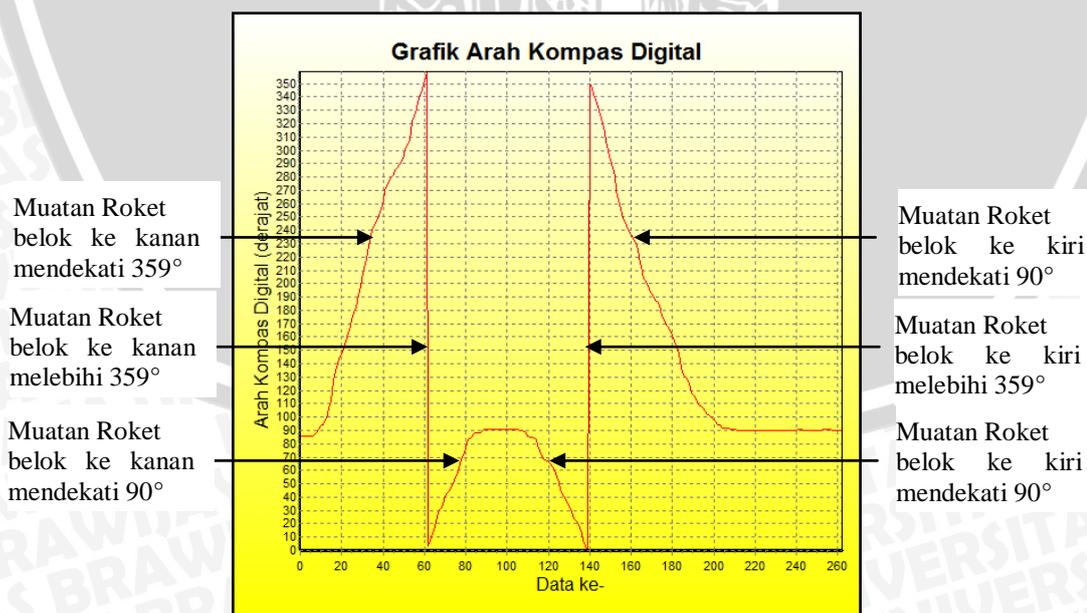


Gambar 5.20 Hasil Tampilan *Software* Antarmuka *Ground Segmen*



Gambar 5.21 Hasil Tampilan Software Antarmuka *Ground Segmen*

Dari hasil pengujian yang terlihat dalam Gambar 5.21, dapat dianalisis bahwa data arah muatan roket yang dikirimkan oleh Mikrokontroller ATMega8 dapat diterima dengan baik oleh visual kompas dengan data 0°-359° secara *real time*.



Gambar 5.22 Hasil Tampilan Grafik Software Antarmuka *Ground Segmen*

Dari hasil pengujian yang terlihat dalam Gambar 5.22, dapat dianalisis bahwa muatan roket mulanya menghadap arah 90° kemudian memutar ke kanan mengarah ke 359°. Setelah melewati 359° muatan roket tetap memutar ke kanan hingga mendekati arah 90°. Ketika sudah 90°, muatan roket lalu memutar ke arah kiri mengarah ke 0°. Setelah melewati 0° muatan roket tetap memutar ke kiri hingga mendekati arah 90°. Dari grafik data arah muatan roket yang dikirimkan oleh Mikrokontroler ATmega8 dapat disimpulkan bahwa tampilan grafik dapat menerima data arah muatan roket dengan baik.

| Waktu | Data ke- | Arah |
|------------|----------|------|
| 6:01:40 AM | 1 | 86 |
| 6:01:40 AM | 2 | 86 |
| 6:01:40 AM | 3 | 86 |
| 6:01:40 AM | 4 | 86 |
| 6:01:40 AM | 5 | 86 |
| 6:01:40 AM | 6 | 86 |
| 6:01:41 AM | 7 | 86 |
| 6:01:41 AM | 8 | 87 |
| 6:01:41 AM | 9 | 89 |
| 6:01:41 AM | 10 | 91 |
| 6:01:41 AM | 11 | 93 |
| 6:01:41 AM | 12 | 94 |
| 6:01:41 AM | 13 | 98 |
| 6:01:41 AM | 14 | 102 |
| 6:01:41 AM | 15 | 107 |
| 6:01:42 AM | 16 | 112 |
| 6:01:42 AM | 17 | 127 |
| 6:01:42 AM | 18 | 135 |
| 6:01:42 AM | 19 | 140 |
| 6:01:42 AM | 20 | 144 |
| 6:01:42 AM | 21 | 147 |
| 6:01:42 AM | 22 | 152 |
| 6:01:42 AM | 23 | 157 |
| 6:01:43 AM | 24 | 163 |
| 6:01:43 AM | 25 | 168 |
| 6:01:43 AM | 26 | 173 |
| 6:01:43 AM | 27 | 178 |
| 6:01:43 AM | 28 | 182 |
| 6:01:43 AM | 29 | 192 |
| 6:01:43 AM | 30 | 198 |
| 6:01:43 AM | 31 | 204 |
| 6:01:43 AM | 32 | 211 |
| 6:01:43 AM | 33 | 219 |

2 of 332

Sheet View

Gambar 5.23 Tampilan Database

Dari hasil pengujian yang terlihat dalam Gambar 5.23, maka dianalisis bahwa data arah muatan roket dapat disimpan ke dalam database. Dengan demikian dapat disimpulkan bahwa sistem secara keseluruhan dapat bekerja dengan baik.

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Berdasarkan hasil perancangan dan pengujian yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

- 1) Sensor Kompas Digital CMPS03 dapat diimplementasikan pada muatan roket dengan menunjukkan arah muatan roket. Data arah sensor kompas digital CMPS03 yang diakses secara I2C (*Inter Integrated Circuit*) ini cukup valid dan sesuai dengan kondisi yang sebenarnya karena rata-rata kesalahan pada pengukuran arah sebesar 0,14%. Namun, ketika muatan roket dalam keadaan miring 45° terhadap bidang datar maka rata-rata kesalahan pada pengukuran arah sebesar 2,5%.
- 2) Radio Frekuensi yang dipakai menggunakan tipe YS-1020U sebagai komunikasi data muatan roket dengan *ground segmen*. Data dapat diterima dengan baik oleh *software* antarmuka *ground segmen* sampai pada jarak 250 meter antara pemancar dan penerima pada lokasi terbuka. Pada jarak lebih dari 250 meter, data sudah tidak sesuai dengan format data yang dikirim sehingga data menjadi tidak valid.
- 3) Sistem elektronika yang menunjang penggunaan sensor kompas digital CMPS03 sebagai penunjuk arah muatan roket yaitu rangkaian catu daya, rangkaian sensor kompas digital CMPS03, rangkaian mikrokontroler ATmega8, rangkaian YS-1020U, dan *software* antarmuka yang dapat memonitor arah muatan roket dalam bentuk visual dan grafik secara *real time*.

6.2. Saran

Saran-saran dalam pengimplementasian maupun peningkatan unjuk kerja sistem ini dapat diuraikan sebagai berikut:

- 1) Dalam sistem yang telah dirancang, muatan roket belum dapat dikontrol arahnya. Oleh karena itu pada pengembangan selanjutnya dapat menggunakan motor pendorong agar muatan roket dapat diarahkan.
- 2) Muatan roket dapat diintegrasikan dengan sensor percepatan agar muatan roket dapat mengetahui jarak dan arah dari muatan roket ketika diluncurkan.

DAFTAR PUSTAKA

- Atmel. 2007. *8-bit AVR with 8K Bytes In-System Programmable Flash ATmega8*.
<http://www.datasheetcatalog.org/datasheet/atmel/2486S.pdf>. Diakses tanggal 28 September 2011.
- Digi-ware.2008. *CMPS03 – Devantech Magnetic Compass AN-09*.
www.digi-ware.com/file/AN-09.pdf. Diakses tanggal 27 September 2011.
- Giyuni.2009. *RF(Radio Frekuensi) Data Tranceiver*. <http://blog.giyuni.com/2009/06/rf-radio-frekuensi-data-transceiver.html>. Diakses tanggal 20 September 2011.
- Hanapi, Gunawan (penerjemah) Malvino A. P. 1996 *Prinsip-Prinsip Elektronika,Edisi Kedua*. Jakarta: Erlangga.
- Heryanto, Wisnu. 2010. *Pemrograman Bahasa C untuk Mikrokontroler ATmega 8535*. Yogyakarta: Andi.
- Laros-edu. 2007. *Komunikasi Seral Mikrokontroler dengan PC (Komputer)*.
<http://www.ibnubudir.files.wordpress.com/2008/08/komunikasi-serial-mikrokontroler-dengan-pc.pdf>. Diakses tanggal 4 September 2011.
- Rifai, Achmad. 2008. *Borland Delphi 7*.
<Http://www.downloads.ziddu.com/downloadfile/7283426/DelphiPart1.pdf.html>. Diakses tanggal 02 September 2011.
- ShenZen Yishi Electronic Technology Development. 2000. *YS-1020UB RF Data Transceiver*.<http://lapan.te.ugm.ac.id/download.php?f=YS-1020UB%20manual.pdf>. Diakses tanggal 27 September 2011.
- Soebhakti, Hendawan. 2008. *Digital Compass CMPS03*. <Http://www.robot-electronics.co.uk>. Diakses tanggal 02 September 2011.
- Sutrisno. 1987. *Elektronika, Teori dan Penerapannya*. Bandung:ITB
- Sudjadi. 2005. *Teori dan Aplikasi Microcontroller*. Yogyakarta: Graha Ilmu
- Wahana Komputer. 2009. *Aplikasi Cerdas Menggunakan DELPHI*. Yogyakarta: Andi.
- Winoto, Ardi. 2008. *Mikrokontroler AVR Atmega8/ 16/32/8535 dan Pemrogramannya dengan Bahasa C pada WinAVR*. Bandung: Informatika.

LAMPIRAN I

Dokumentasi Alat





Gambar 1. Muatan roket Tampak



Gambar 2. Muatan Roket Tampak Bawah



Gambar 3. Alat secara keseluruhan

LAMPIRAN II

Listing Program Mikrokontroler ATMEGA8



repository.ub.ac.id

```
/******
```

```
This program was produced by the  
CodeWizardAVR V1.24.8d Professional  
Automatic Program Generator  
© Copyright 1998-2006 Pavel Haiduc, HP InfoTech s.r.l.  
http://www.hpinfotech.com
```

```
Project :  
Version :  
Date : 03-Nov-2011  
Author : F4CG  
Company : F4CG  
Comments:
```

```
Chip type : ATmega8  
Program type : Application  
Clock frequency : 16.000000 MHz  
Memory model : Small  
External SRAM size : 0  
Data Stack size : 256
```

```
*****/
```

```
#include <mega8.h>  
#include <delay.h>  
// I2C Bus functions  
#asm  
    .equ __i2c_port=0x15 ;PORTC  
    .equ __sda_bit=4  
    .equ __scl_bit=5  
#endasm  
#define write 192  
#define read 193  
#include <i2c.h>  
// Standard Input/Output functions  
#include <stdio.h>  
char byte[3] = {0,0,0};  
// Declare your global variables here  
void partisi(int part)  
{  
    byte[0] = (part/1000)+48;  
    byte[1] = ((part% 1000)/100)+48;  
    byte[2] = ((part% 100)/10)+48;  
    printf("#%c%c%c*", byte[0], byte[1], byte[2]);  
}
```

```
unsigned char Akses_kompas(char alamat)  
{  
    unsigned char Value;  
    i2c_start();  
    i2c_write(write);  
    i2c_write(alamat);  
    i2c_start();  
    i2c_write(read);  
    Value=i2c_read(0);  
    i2c_stop();  
    return Value;  
}
```

```

void DATA_CMPS03()
{
Int CMPSA,CMPSB;
float DATA_CMPS;
  CMPSA= Akses_kompas(2);
  CMPSB= Akses_kompas(3);
  CMPSA= CMPSA<<8;
  CMPSA= CMPSA | CMPSB;
  DATA_CMPS= (float)CMPSA/10;
  partisi(DATA_CMPS);
}

```

```

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x00;

// Port C initialization
// Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
TCCR0=0x00;
TCNT0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;

```

```

TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

```

```

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

```

```

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
MCUCR=0x00;

```

```

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

```

```

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 9600
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x67;

```

```

// Analog Comparator initialization
// Analog Comparator: Off
ACSR=0x80;
SFIO=0x00;

```

```

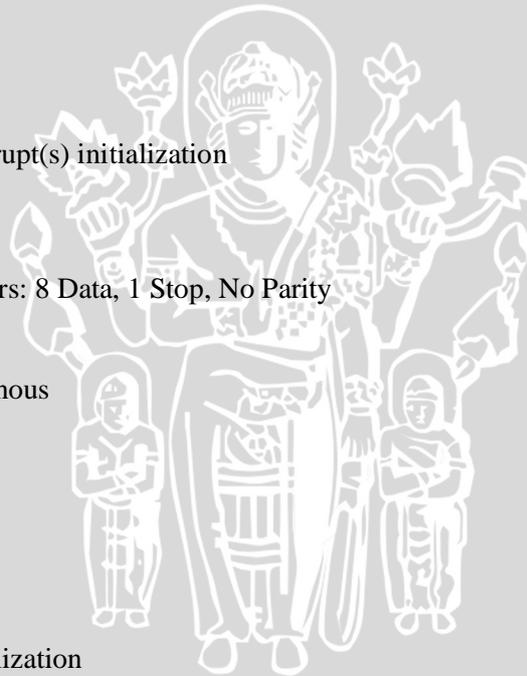
// I2C Bus initialization
i2c_init();

```

```

while (1)
{
// Place your code here
DATA_CMPS03();
delay_ms(100);
};
}

```



LAMPIRAN III

Listing Program Software Antarmuka Delphi



unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, AdvSmoothGauge, StdCtrls, XComDrv, XPMan, DBTables, DB, ADODB,
Provider, DBClient, AdvSmoothLedLabel, Grids, DBGrids, TeEngine, Series,
ExtCtrls, TeeProcs, Chart, ExtDlgs, jpeg;

type

```
TForm1 = class(TForm)
  AdvSmoothGauge1: TAdvSmoothGauge;
  XComm1: TXComm;
  Button1: TButton;
  XPManifest1: TXPManifest;
  DataSource1: TDataSource;
  ADOConnection1: TADOConnection;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Label5: TLabel;
  Chart1: TChart;
  SavePictureDialog1: TSavePictureDialog;
  Timer1: TTimer;
  Label6: TLabel;
  Label7: TLabel;
  Label8: TLabel;
  Label9: TLabel;
  Label10: TLabel;
  Label11: TLabel;
  Label12: TLabel;
  Label13: TLabel;
  Label14: TLabel;
  Label15: TLabel;
  Label16: TLabel;
  Label17: TLabel;
  Label18: TLabel;
  ADOTable1: TADOTable;
  DBGrid1: TDBGrid;
  Button3: TButton;
  Exit: TButton;
  Label19: TLabel;
  Series1: TLineSeries;
  procedure XComm1Data(Sender: TObject; const Received: Cardinal);
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure ExitClick(Sender: TObject);
```

private

{ Private declarations }

public

{ Public declarations }

end;

```

var
  Form1: TForm1;
  Buffer:widestring;
  data,ribuan, satuan,puluhan,ratusan, koma1, koma2:string;
  tanda,data_compass,i,d_satuan,d_puluhan,d_ratusan, d_ribuan, d_koma1, d_koma2:integer;
  connect:boolean;

```

implementation

```
{$R *.dfm}
```

```

procedure delay(lama:longint);
var
  ref:longint;
begin
  ref:=gettickcount;
  repeat
    application.ProcessMessages;
  until((gettickcount-ref)>=lama)
end;

```

```

procedure TForm1.Button1Click(Sender: TObject);

```

```

begin
  if(connect=false)then
  begin
    XComm1.OpenDevice;
    connect:=true;
    Button1.Caption:='Disconnect';
    i:=0;

```

```

    repeat
      delay(1000);
      inc(i);
      until((i=100)or(connect=false));
    end
  else
  begin
    XComm1.CloseDevice;
    connect:=false;
    Button1.Caption:='Connect';
  end;
end;

```

```

procedure TForm1.XComm1Data(Sender: TObject; const Received: Cardinal);

```

```

begin
  XComm1.ReadString(data);
  Buffer:=buffer+data;
  if (data=*) then begin
    inc(tanda);
    ratusan:= copy(Buffer,pos('#',Buffer)+1,1);
    puluhan:= copy(Buffer,pos('#',Buffer)+2,1);
    satuan:= copy(Buffer,pos('#',Buffer)+3,1);

```

```
d_satuan:=StrToInt(satuan);
d_puluhan:=StrToInt(puluhan);
d_ratusan:=StrToInt(ratusan);

data_compass:=(d_satuan*1)+(d_puluhan*10)+(d_ratusan*100);

AdvSmoothGauge1.Value:=data_compass;

Chart1.Series[0].AddXY(tanda-1,data_compass);

ADOTable1.Append;
ADOTable1.FieldByName('Waktu').AsDateTime:=now;
ADOTable1.FieldByName('Arah').AsString:=IntToStr(data_compass);
ADOTable1.Post;

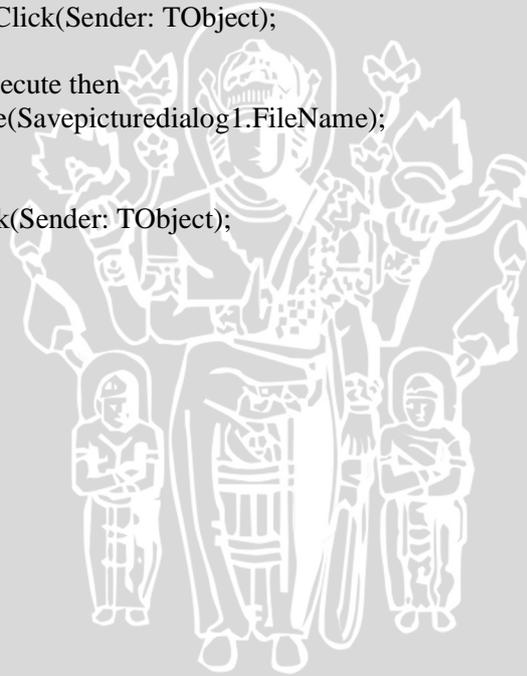
buffer:="";

end;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  If SavePictureDialog1.Execute then
    Chart1.SaveToBitmapFile(Savepicturedialog1.FileName);
end;

procedure TForm1.ExitClick(Sender: TObject);
begin
  close();
end;

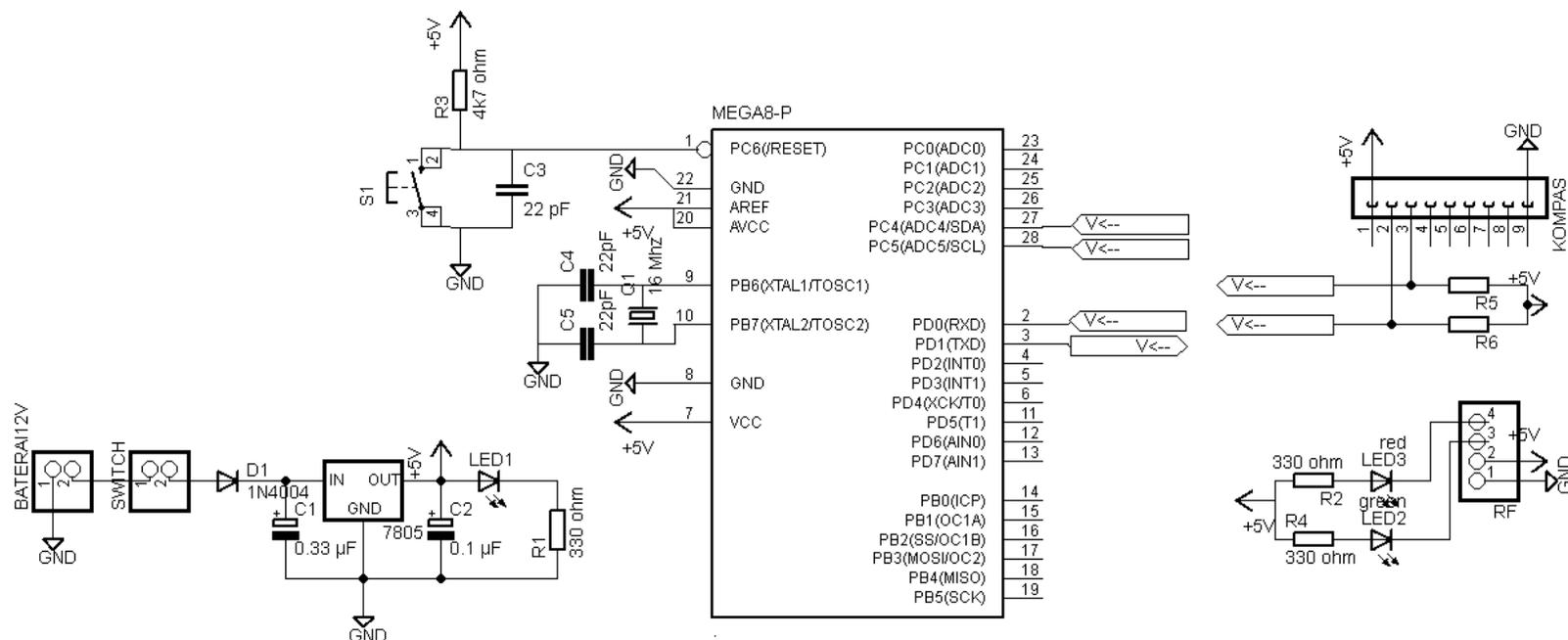
end.
```



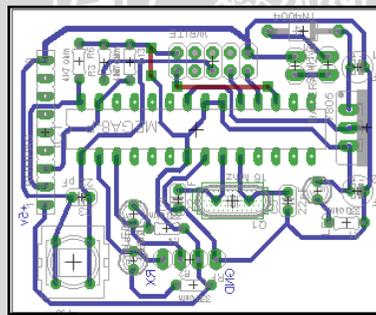
LAMPIRAN IV

Schematic Rangkaian Elektrik





Gambar 1. Schematic Rangkaian Elektrik Keseluruhan



Gambar 2. Board Elektrik

LAMPIRAN V

Datasheet

