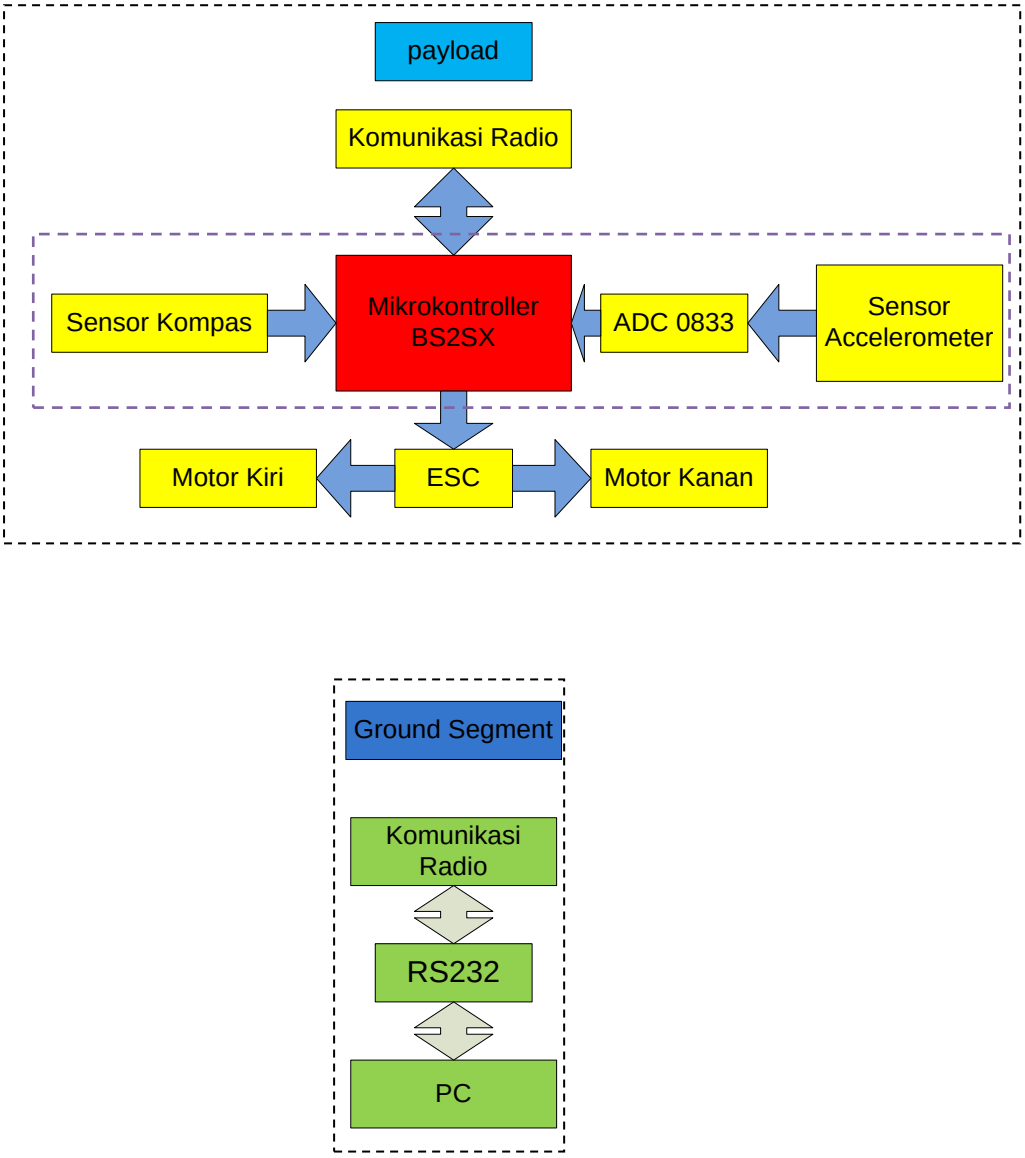
BAB III

PERANCANGAN SISTEM

3.1 Perancangan Sistem

Dalam tugas akhir ini, dibangun sistem telemetri sensor kompas dan sensor *accelerometer* pada *payload* roket. Dibawah ini merupakan diagram blok sistem yang akan dirancang.



Gambar 3.1 Block Diagram Sistem Telemetri *Payload*

Sistem telemetri *payload* bekerja jika ada perintah dari *ground segment* untuk mengaktifkan sistem *payload* melalui modul komunikasi radio. Sistem pada *payload* menerima perintah untuk mengaktifkan sistem, mikrokontroler yang sudah diprogram untuk menerima data dari sensor kompas dan sensor *accelerometer*, kemudian data diolah oleh mikrokontroler dari data analog menjadi data digital. Data dari sensor *accelerometer* harus diubah dari analog menjadi digital dengan bantuan ADC. Setelah data diolah, data tersebut dikirim oleh mikrokontroler ke ESC (*Electronic Speed Control*) untuk menggerakkan motor *brushless* agar *payload* berada pada posisi stabil, kemudian data dikirimkan ke *ground segment* melalui modul radio dan data dapat di analisa untuk mengukur *performance* atau percepatan *payload* roket secara *real time*.

Tabel 3.1 Deskripsi Blok Diagram Sistem *Payload*

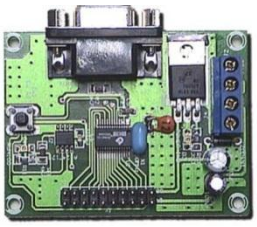
No	Blok	Keterangan
1	Komunikasi radio	Blok komunikasi yang menghubungkan seluruh sistem pada *payload* dan *ground segment*
2	Mikrokontroler BS2SX	Mengatur seluruh kerja blok sistem pada *payload*
3	Sensor Kompas	Mencari sudut arah mata angin, 0º dari utara
4	Sensor *Accelerometer*	Mengukur besaran percepatan *payload*
5	ADC	Mengubah besaran analog menjadi besaran digital
6	ESC	*Elektronik Speed Control*, mengatur kecepatan putaran motor
7	RS 232	Sebagai pengubah logika TTL antar *Hardware* dan PC
8	PC	Sebagai *interface* dari *Hardware*
9	Motor Kanan	Mendorong *payload* kearah kanan
10	Motor Kiri	Mendorong *payload* kearah kiri

3.2 Perancangan *Hardware*

Pada perancangan *Hardware* terdiri dari pemilihan komponen mikrokontroler, sensor-sensor, perancangan desain *payload* , aktuator dan catu daya serta komponen komunikasi radio.

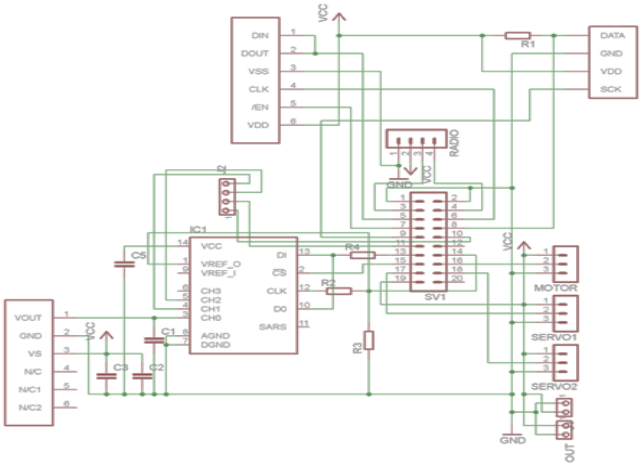
3.2.1 Rangkaian *Motherboard* Sensor dan Modul BS2SX

Mikrokontroler yang digunakan adalah sebuah modul BS2SX. Mikrokontroler ini memiliki jumlah pin yang banyak yaitu 20 pin. Kemudahan dalam pemrograman khususnya sebagai pengendali aktuator yang digunakan, berikut gambar dari BS2SX yang digunakan.



Gambar 3.2 Modul BS2SX

Untuk mempermudah koneksi antara sensor dengan pin-pin pada modul BS2SX maka dibuatlah sebuah PCB untuk meletakkan semua sensor yang langsung dapat terkoneksi dengan modul BS2SX. Berikut gambar rangkaian sensor-sensor.



Gambar 3.3 *Motherboard* Sensor-sensor

Tabel 3.2 Pin-pin yang digunakan Pada BS2SX

Pin	Keterangan
0	Clock ADC
1	Data Kompas
2	Clock ADC
3	Clock Kompas
4	CS ADC
5	Enable Kompas
6	Servo Kanan
7	RX Radio
8	Servo Kiri
9	TX Radio
10	Sleep MMA
11	Motor Kiri
12	Select 2
13	Motor Kanan
14	Select 1
15	-

3.2.2 Komunikasi Serial

Proses komunikasi antara mikrokontroler yang berada didalam *payload* dengan *ground segment* menggunakan modul radio YS1020UB. Dalam perancangan ini pin-pin pada basic stamp harus terhubung dengan pin pada modul radio. Pada modul radio pin yang terpakai hanya terdiri dari GND, Vcc, RXD/TTL dan TXD/TTL.

Dalam komunikasi ini harus melakukan penyesuaian *baudrate* pada basic stamp dengan modul radio yang digunakan, agar proses komunikasi berjalan dengan baik. Dibawah ini dapat dilihat tabel 3.3 penggunaan *baudrate* pada Basic Stamp.

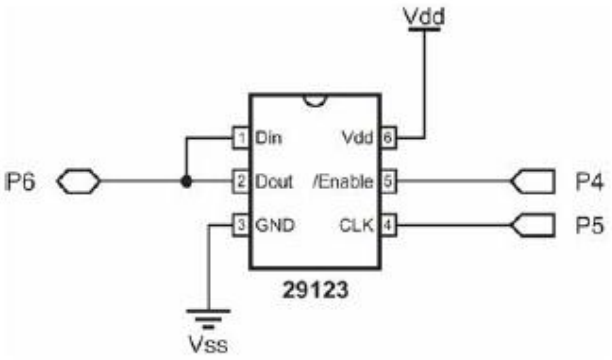
Tabel 3.3 Baudrate pada Basic Stamp

Baudrate	**8-bit No Parity Inverted**	**8-bit No Parity True**	**7-bit Even Parity Inverted**	**7-bit EvenParity True**
1200	18447	2063	26639	10255
2400	17405	1021	25597	9213
4800	16884	500	25076	8692
9600	16624	240	24816	8432

Pin RX pada radio digunakan sebagai pin untuk mengirimkan data dengan perintah dari Basic Stamp yaitu *Serout, Pin, Boudrate, [data yang dikirimkan]*, sedangkan untuk membaca data yang diterima maka pin TX (Pengirim) pada modul radio disambungkan dengan pin Basic Stamp dengan menggunakan perintah *Serin, Pin, baudrate, variable penyimpanan data*.

3.2.3 Rangkaian Sensor Kompas

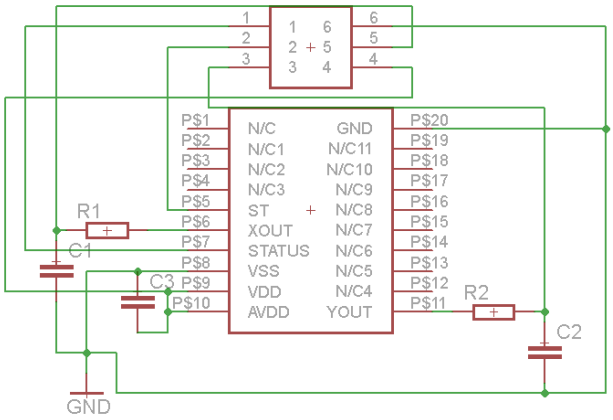
Sensor kompas bekerja pada tegangan input sebesar 5 V, pin data input dan data output disatukan untuk jalur data keluaranya, keluarannya berupa data digital. diperlukan juga sebuah data clock untuk mengambil data dari sensor kompas karena datanya bersifat serial. Dibawah ini gambar rangkaian sensor kompas.



Gambar 3.4 Rangkaian Sensor Kompas HM55b

3.2.4 Rangkaian Sensor Percepatan MMA3201

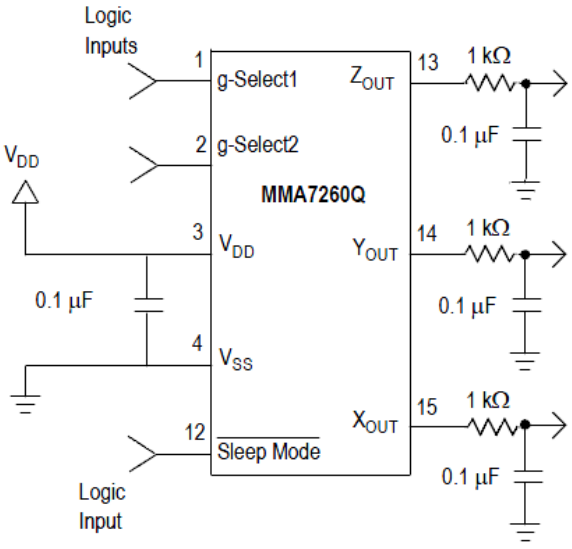
Sensor ini akan mengukur akselerasi gerak dari *payload* dengan satuan g. Data yang digunakan adalah data dari axis X dan juga axis Y, apabila kedua data ini akan dijadikan dalam besaran derajat maka dapat dihitung menggunakan rumus angle= ArcusTAN (X/Y), serta apabila data yang sudah menjadi besaran derajat ingin dirubah menjadi besaran dalam g kembali maka dapat digunakan rumus x=sin angle dan y=cos angle. Berikut gambar dari rangkaian *accelerometer* MMA3201.



Gambar 3.5 Rangkaian *Accelerometer* MMA3201

3.2.5 Rangkaian MMA7260Q

Sensor *accelerometer* ini beroperasi pada tegangan 2,2–3,6 volt dengan tegangan tipikal 3,3 volt (Vdd). Keluaran sensor *accelerometer* berupa tegangan analog yang merepresentasikan data percepatan dalam satuan gravitasi (g). Pengesetan dilakukan dengan memberikan input logika 0 pada pin 1 dan pin 2 pada pin *g-select1* dan *g-select2*.

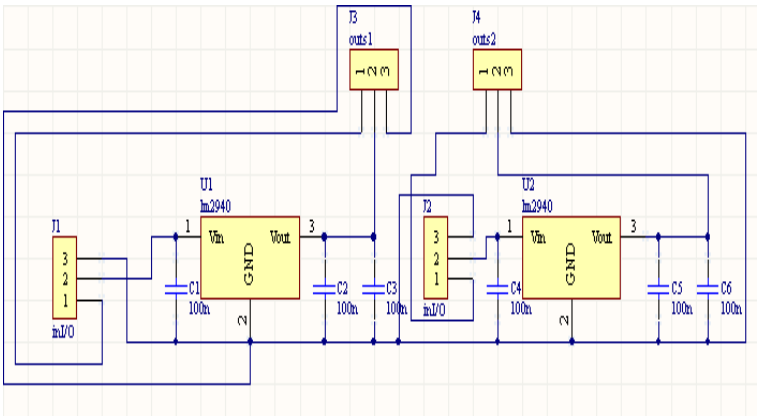


Gambar 3.6 Rangkaian Accelerometer MMA7260Q

3.2.6 Rangkaian Catu Daya

Catu daya merupakan faktor yang penting dalam perancangan sistem telemetri ini, karena tanpa adanya sebuah catu daya maka sistem ini tidak akan mampu untuk bekerja. Catu daya ini berfungsi untuk mengaktifkan mikrokontroler, semua sensor yang digunakan, aktuator serta radio komunikasinya.

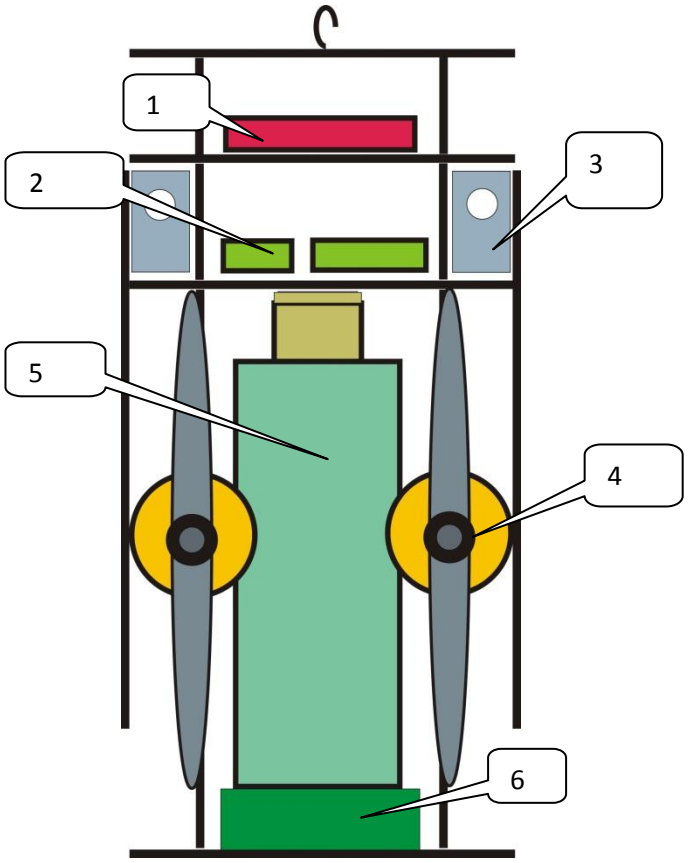
Catu daya pada aktuator menggunakan sumber tegangan sebesar 11,1V dc dan 5V dc sedangkan pada mikrokontoler dan sensor membutuhkan sumber sebesar 5 V dc. Pada mikrokontroler telah tersedia regulator 5 V sebagai penurun tegangan sedangkan pada aktuator tidak diperlukan sebuah rangkaian penurun tegangan.



Gambar 3.7 Rangkaian Catu Daya 5 volt

3.2.7 Desain *Payload* Menggunakan Anti Noise

Desain *payload* sangat berperan penting dalam tugas akhir ini, karena dalam mendesain perlu adanya perhitungan dari berbagai komponen dan jenis bahan yang digunakan dalam pembuatan *payload* ini agar *payload* memiliki kontruksi yang kuat. Kerangka *payload* ini dibuat dari akrilik yang ringan dan mudah untuk dibentuk. *Payload* ini dilengkapi dengan peredam getaran, peredam ini berguna untuk memperkecil getaran atau hentakan yang dapat menganggu kinerja dari sistem *payload*.



Gambar 3.8 Desain Awal *Payload*

Tabel 3.4 Deskripsi Gambar Desain *Payload*

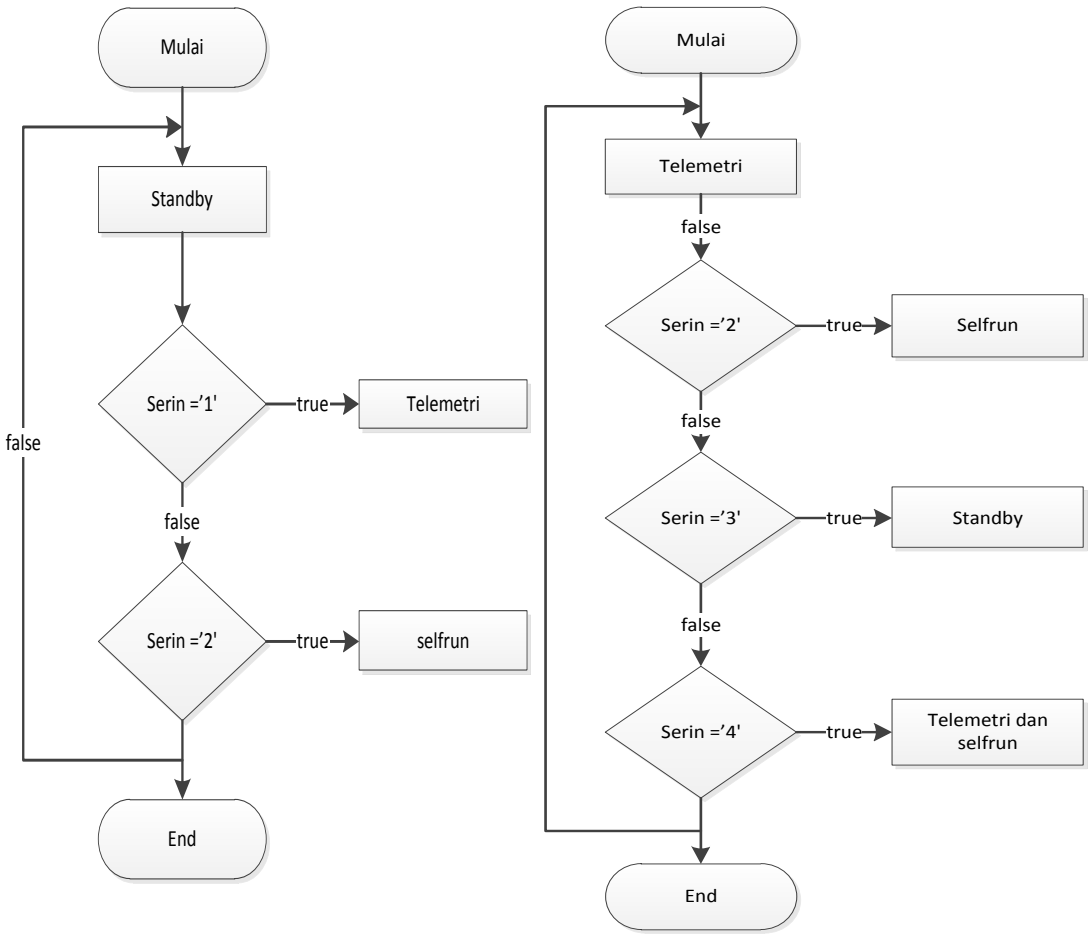
No	Keterangan
1	Mikrokontroler dan ADC
2	Sensor *Accelerometer* dan Kompas
3	Servo
4	Motor *Brushless* dan propeller
5	Antena radio dan Baterai Li-Po
6	Regulator

3.3 Perancangan *Software*

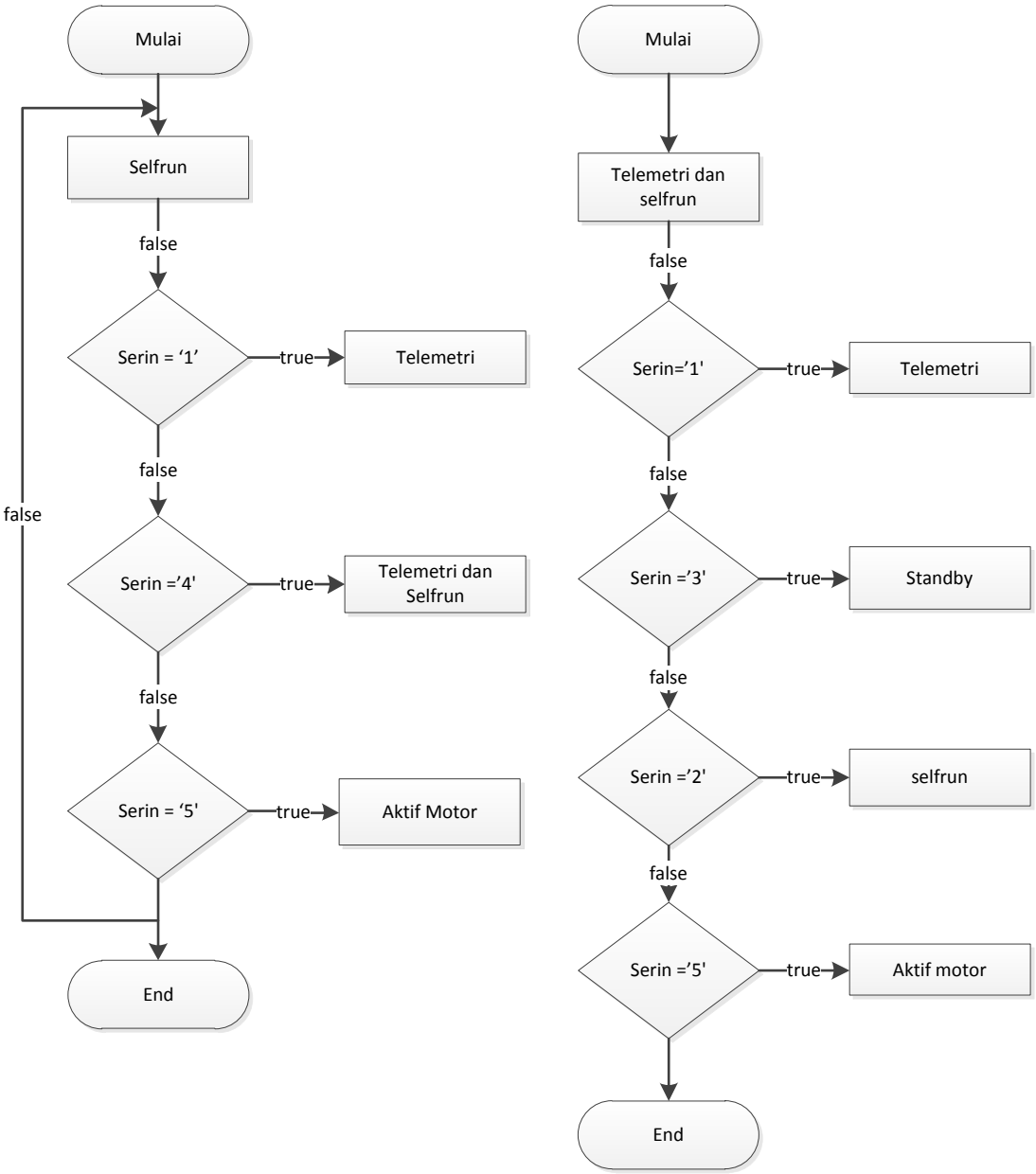
Perancangan *software* merupakan hal yang penting dalam perancangan sistem kendali pada *payload* . Pada bagian perancangan ini yang nantinya akan dijadikan sebagai algoritma kendali dari keseluruhan sistem kerja pada *payload* . Algoritma tersebut dituliskan pada sebuah Basic Stamp Editor v 2.2 dengan bahasa basic kemudian disimpan pada sebuah IC BS2SX, maka secara otomatis BS2SX akan mengerjakan seluruh program.

3.3.1 Algoritma Dasar

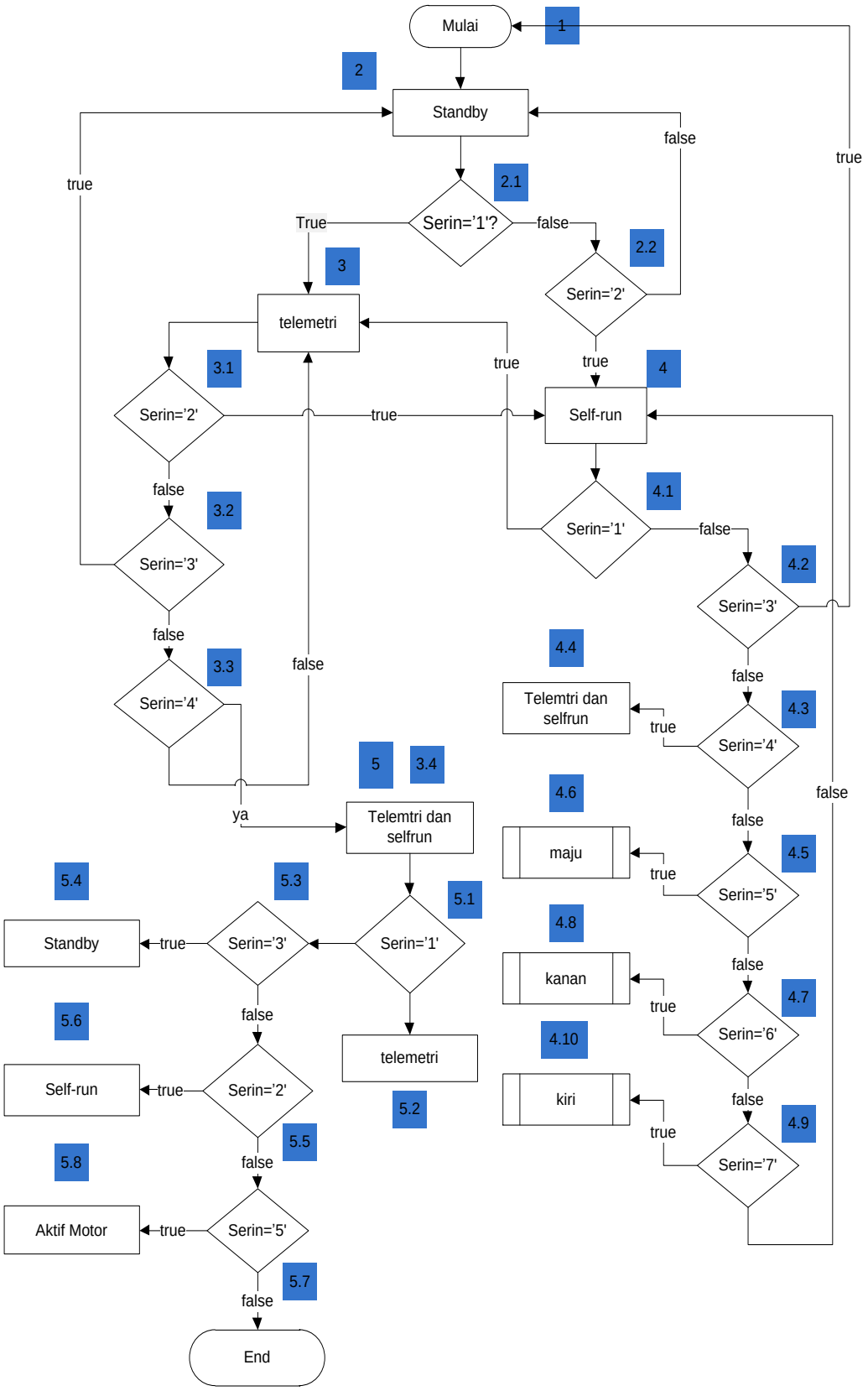
Algoritma dasar merupakan algoritma yang mendasari algoritma pada sistem *payload*. Algoritma dasar ini yang nantinya akan dikembangkan menjadi algoritma yang lebih komplek dan lebih spesifik lagi. Berikut ini merupakan diagram alir sederhana dari sistem *payload* yang dirancang.



Gambar 3.9 Diagram Alir Algoritma Standby dan Telemetri



Gambar 3.10 Diagram Alir Algoritma Selfrun dan Telemetri



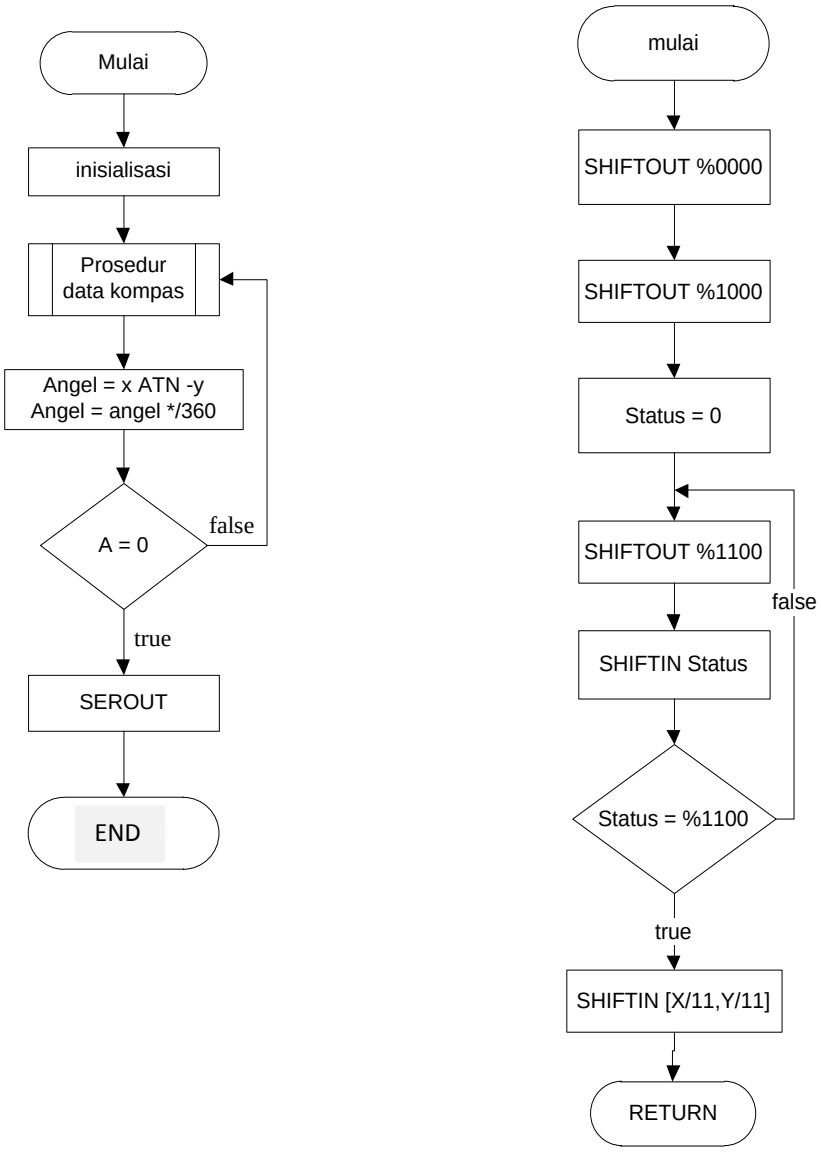
Gambar 3.11 Algoritma Dasar Dalam Bentuk Diagram Alir

Table 3.5 Deskripsi Alur Kerja Sistem *Payload*

No	Keterangan
1	Mulai Program dijalankan
2	Rutin standby, mikrokontroler tidak mengerjakan apapun
2.1	Jika data = ‘1’ maka lompat ke rutin telemetri jika tidak lanjut ke ’2.2’
2.2	Jika data = ‘2’ maka lompat ke rutin self run jika tidak balik ke mulai ‘1’
3	Rutin telemetri
3.1	Jika data = ‘2’ maka lompat ke rutin self run, jika tidak lanjut ke ‘3.2’
3.2	Jika data = ‘3’ maka lompat ke rutin standby, jika tidak lanjut ke ‘3.3’
3.3	Jika data = ‘4’ maka lompat ke rutin telemetri dan self run, jika tidak kembali ke rutin telemetri
4	Self run
4.1	Jika data = ‘1’ maka lompat ke rutin telemetri, jika tidak lanjut ke ‘4.2’
4.2	Jika data = ‘3’ maka lompat ke awal, jika tidak lanjut ke ‘4.3’
4.3	Jika data = ‘4’ maka lompat ke rutin telemetri dan self run (4.4), jika tidak lanjut ke ‘4.5’
4.4	Telemetri dan self run
4.5	Jika data = ‘5’ maka lompat ke prosedur maju (4.6), jika tidak lanjut ke ‘4.7’
4.6	Prosedur maju
4.7	Jika data = ’6’ maka lompat ke prosedur kanan (4.8), jika tidak lanjut ke ‘4.9’
4.8	Prosedur kanan
4.9	Jika data = ‘7’ maka lompat ke prosedur kiri (4.10), jika tidak lanjut ke 4
4.10	Prosedur kiri
5	Telemetri dan self run
5.1	Jika data = ‘1’ maka lompat ke rutin telemetri (5.2), jika tidak lanjut ke 5.3
5.2	Telemetri
5.3	Jika data = ‘3’ maka lompat ke rutin standby (5.4), jika tidak lanjut ke 5.5
5.4	Standby
5.5	Jika data = ‘2’ maka lompat ke rutin self run (5.6), jika tidak lanjut ke 5.7
5.6	Selfrun
5.7	Jika data =’5’ maka lompat ke rutin aktif motor (5.8), jika tidak lanjut ke end.

3.3.2 Prosedur Sensor Kompas HM55B

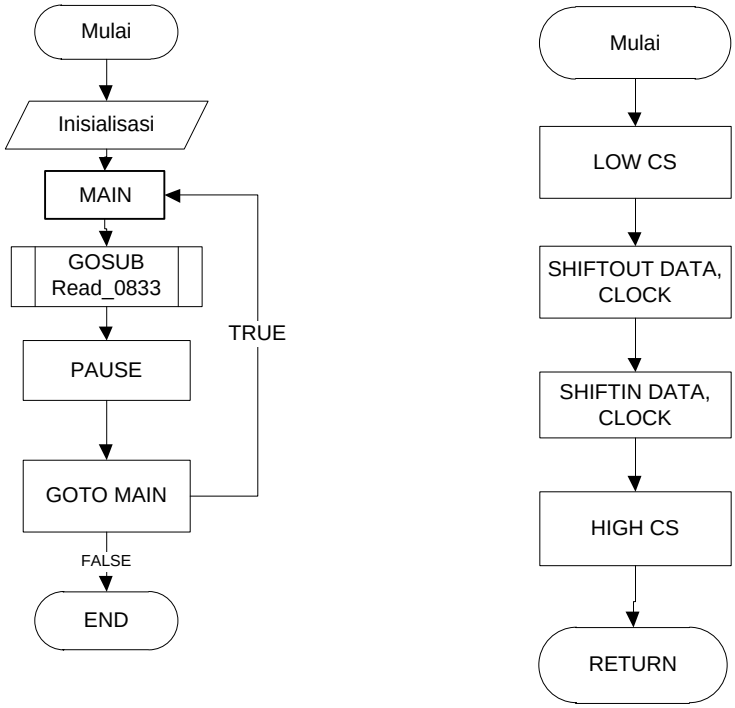
Algoritma prosedur sensor kompas digunakan dalam memprogram mikrokontroler. Prosedur ini akan di masukkan kedalam pembuatan program sehingga sensor kompas yang dipergunakan berjalan sesuai dengan keluaran yang dihasilkan. Dibawah ini Algoritma prosedur kompas yang dibuat penulis.



Gambar 3.12 Diagram Alir Prosedur Sensor Kompas

3.3.3 Prosedur Sensor Accelerometer MMA3201

Algoritma prosedur sensor *accelerometer* digunakan dalam memprogram mikrokontroler untuk mengetahui nilai percepatan *payload* roket. Dengan adanya prosedur ini, menghasilkan nilai percepatan yag sesuai dengan perolehan data yang nantinya akan dianalisa. Dibawah ini adalah prosedur sensor *accelerometer*.



Gambar 3.13 Diagram Alir Prosedur Sensor *Accelerometer*

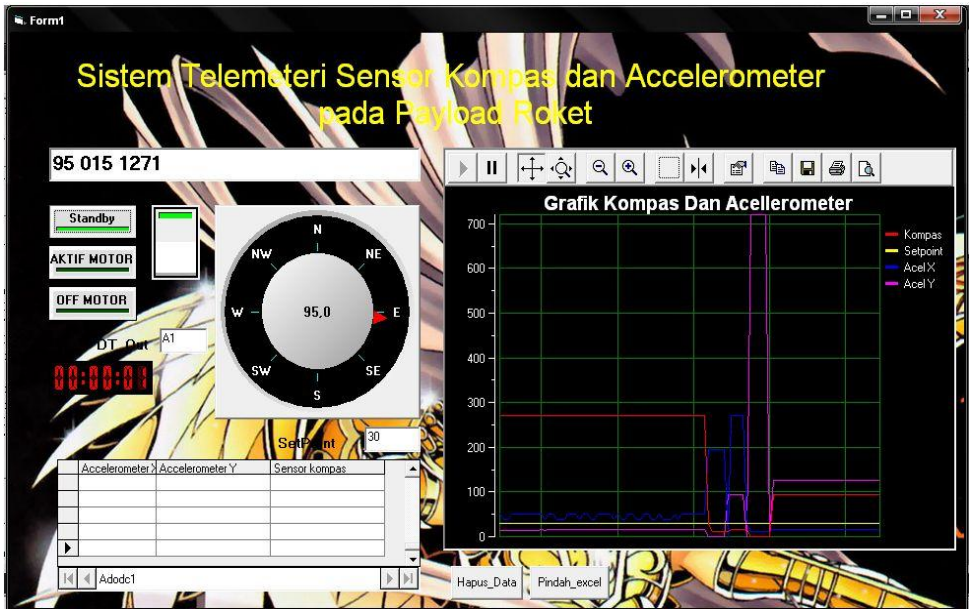
3.3.4 Perancangan Pada Komputer

Perancangan pada komputer merupakan pembuatan program antarmuka (*interface*) dengan menggunakan bahasa pemrograman visual basic 6.0. Perancangan pada perangkat lunak ini difokuskan pada pembuatan program antar muka (*interface*) yang mampu menyampaikan informasi yang sesuai. Selain itu, program yang dibuat harus bias berkomunikasi dengan perangkat keras lainnya pada *baudrate* tertentu.

Dalam pembuatan program antarmuka (*interface*), sistem perancangan telemetri ini menggunakan Visual Basic 6.0 pada *ground segment* dengan beberapa keunggulan dari bahasa pemrograman ini adalah :

1. Mudah untuk digunakan
2. Fasilitas dan fitur-fitur grafis yang diperlukan untuk program antar muka (*interface*) tersedia.
3. Program Visual Basic berbasis windows, sehingga program akan mudah untuk dipahami dan dimengerti.
4. Pemrograman ini mempunyai *database* sendiri yang digunakan untuk menyimpan data.

Program ini berguna ketika informasi yang dikirimkan *payload* roket dapat diolah di *ground segment* . Data yang masuk akan disimpan secara otomatis dalam sebuah *database* sederhana. Program ini juga berguna untuk mengendalikan semua aktifitas *payload* roket pada saat di *launcer* sampai keadaan separasi hingga proses telemetri berlangsung. Dibawah ini adalah gambar dari *ground segment* sebagai penerima data telemetri, serta *monitoring* pergerakan *payload* roket.



Gambar 3.14 Tampilan *Ground Segment*