BAB II
DASAR TEORI

2.1 Pengertian dan Sejarah Satelit

Perkembangan teknologi satelit semakin maju, hal tersebut ditandai dengan perubahan desain satelit yang berukuran besar menjadi berukuran kecil dengan mempunyai teknologi yang sama. Satelit mempermudah kita dalam mendapatkan informasi secara cepat memperoleh berita serta berbagai kejadian yang diperlukan manusia melalui udara, tanpa harus melibatkan manusia dalam bahaya.

Satelit buatan manusia pertama adalah Sputnik 1, diluncurkan oleh Soviet pada tanggal 4 Oktober 1957. Sputnik 2 diluncurkan pada tanggal 3 November 1957 dan membawa awak mahluk hidup pertama ke dalam orbit yaitu seekor anjing bernama Laika. Satelit buatan manusia terbesar pada saat ini yang mengorbit bumi adalah *International Space Station*. Mikro satelit merupakan salah satu jenis satelit yang ringan dan kecil sehingga biaya pembuatannya lebih murah dibandingkan satelit yang lainnya. Mikro satelit yang telah diluncurkan adalah satelit Lapan-TubSat pada tahun 2007 dan dikembangkan juga satelit kembar yakni Saletit Lapan-A2 dan Satelit Lapan-A3 yang akan diluncurkan pada tahun 2011.



Gambar 2.1. Satelit Lapan-TubSat

2.2 Perangkat Keras (Hardware)

2.2.1 Mikrokontroler Basic Stamp

Basic stamp adalah suatu mikrokontroler yang dikembangkan oleh *Parallax Inc* yang diprogram menggunakan bahasa pemrogrograman *basic* dan populer sekitar pada tahun 1990an. Mikrokontroler *basic stamp* membutuhkan *power supply* saat men-*download* dan program di-*download* melaui *port* serial.

Kode PBasic (pemrograman *basic*) disimpan di dalam EEPROM serial pada *board basic stamp*. EEPROM digunakan dalam *basic stamp* 1 dan 2 yang dijamin menyimpan selama 40 tahun ke depan dan mampu ditulisi ulang 10.000.000 kali perlokasi memori.

Mikrokontroler *basic stamp* memiliki beberapa versi yang berbeda – beda, yaitu *basic stamp* 1, *basic stamp* 2, *basic stamp* 1e, *basic stamp* 2P, *basic stamp* 2Pe dan *basic stamp* 2sx. *Basic stamp* jalan pada tegangan DC 5 sampai 15 volt. *Basic stamp* yang dipakai adalah *basic stamp* BS2P40 yang mempunyai 40 pin I/O. Pemilihan *basic stamp* ini karena membutuhkan banyak *input* atau *output* yang dipakai dalam pengontrolan prototipe satelit. Berikut ini adalah tampilan *basic stamp* BS2P40.



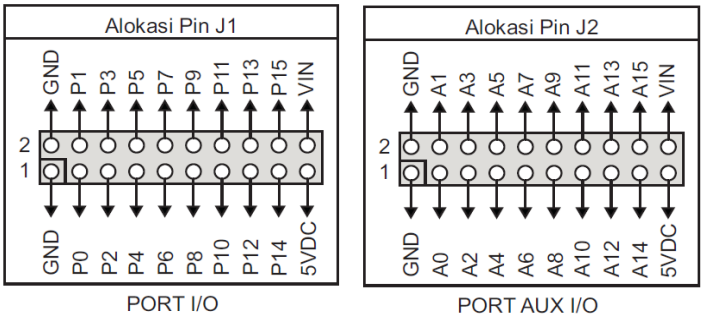
Gambar 2.2. Modul *basic stamp* (BS2P40)

Modul *basic stamp* 2p40 mempunyai spesifikasi *hardware* sebagai berikut:

1. Mikrokontroler *basic stamp* 2P40 *Interpreter Chip* (PBASIC48W/P40)
2. 8 x 2Kbyte EEPROM yang mampu menampung hingga 4000 instruksi.
3. Kecepatan prosesor 20MHz Turbo dengan kecepatan eksekusi program hingga 12000 instruksi per detik.

4. RAM sebesar 38*byte* (12 I/O, 26 variabel) dengan *Scratch Pad* sebesar 128 *byte*.
5. Jalur *input* / *output* sebanyak 32 pin.
6. Tersedia jalur komunikasi serial UART RS-232 dengan konektor DB9.
7. Tegangan *input* 9 – 12 V_{DC} dengan tegangan *output* 5 V_{DC}.

Berikut ini adalah alokasi pin yang terdapat pada mikrokontroler *basic stamp* BS2P40.



Gambar 2.3. Alokasi pin *basic stamp*

Adapun hubungan antara komputer dengan modul BS2P40 memiliki konfigurasi sebagai berikut :

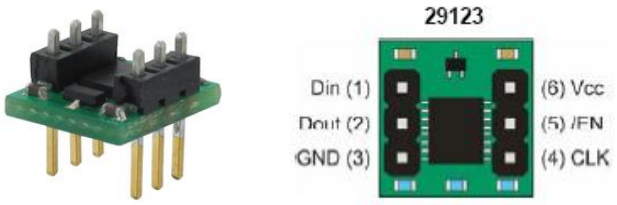
Tabel 2.1. Hubungan pin antara komputer dengan BS2P40 (DB9)

COM Port Komputer DB9	**Modul BS2P40 DB9**
RX (Pin 2)	RX (Pin 2)
TX (pin 3)	TX (pin 3)
DTR (pin 4)	DTR (pin 4)
GND (pin 5)	GND (pin 5)
DSR (Pin 6)	DSR (Pin 6)
RTS (Pin 7)	RTS (Pin 7)

2.2.2 Sensor Kompas HM55B

Kompas adalah alat yang menunjukkan arah mata angin, yaitu utara, selatan, barat, timur. Kompas hitachi HM55B merupakan salah satu kompas digital yang dikembangkan oleh parallax yang mempunyai keluaran digital sebayak 2 axis yaitu axis X dan axis Y. Kompas ini memiliki regulator *onboard* yang mengubah

menjadi 3 V dan itu merupakan daerah kerja kompas ini. Pada saat program dijalankan kecepatan sensitivitas 30-40 ms. Dengan mikrokontroler memudahkan untuk menampilkan arah dalam format 360^0. Berikut ini adalah tampilan dan konfigurasi pin kompas hitachi HM55B.



(a) (b)

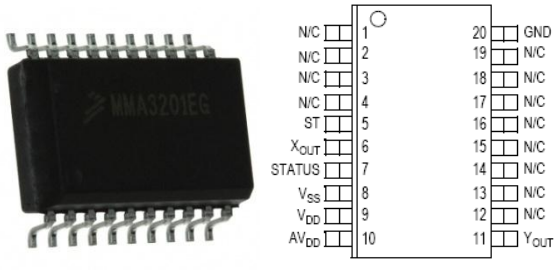
Gambar 2.4. (a) Bentuk fisik sensor kompas HM55B, (b) konfigurasi pin sensor kompas HM55B

Berikut di bawah ini spesifikasi modul Hitachi HM55B:

1. Tegangan sumber: 3V (*Onboard regulator*).
2. Sensitivitas: 1.0 - 1.6 uT/lsb.
3. Resolusi: 6 bit (64 *direction*).
4. Waktu konversi: 30 - 40 ms.
5. *Synchronous serial interface*.
6. Dimensi: 0.3 inci, 6-pin DIP *package*.
7. *Range* pengoperasian: 0 - 70^oC.

2.2.3 Sensor Accelerometer MMA3201EG

Accelerometer berfungsi untuk mengukur percepatan, mendeteksi getaran, dan mengukur percepatan akibat gravitasi bumi. Pada satelit *accelerometer* berguna untuk mendeteksi getaran yang terjadi karena gangguan seperti angin dan menyetimbangkan satelit sampai berbeda-beda posisi yang sempurna untuk melanjutkan ke proses selanjutnya. *Accelerometer* MMA3201 merupakan salah satu sensor *accelerometer* yang mengeluarkan data analog. Sensor ini memiliki kisaran pengukuran (-40 hingga 40) g, dimana 1(satu) g = 9,8 m/s^2. Nilai akurasinya sebesar ±0,2 g. Tampilan sensor *accelerometer* MMA3201 dapat dilihat pada gambar 2.5



(a) (b)

Gambar 2.5. (a) Bentuk fisik Sensor *accelerometer* MMA3201EG, (b) konfigurasi pin sensor *accelerometer* MMA3201EG

Berikut ini adalah tabel yang menjelaskan konfigurasi pin pada sensor MMA3201EG diperlihatkan pada tabel 2.2.

Tabel 2.2. Konfigurasi pin MMA3201EG

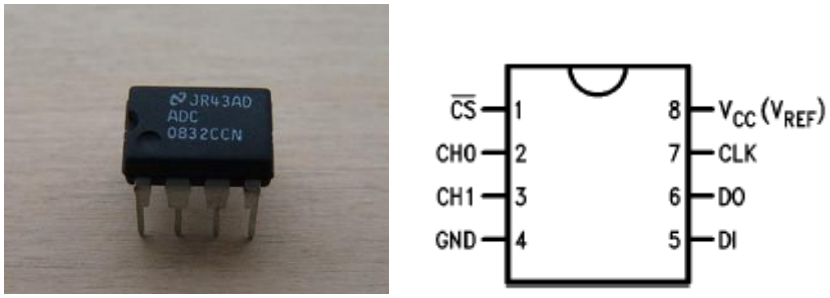
No. Pin	Nama Pin	Keterangan
1 sampai 3	-	*Leave unconnected*.
4	-	Tidak ada koneksi internal. *Leave unconnected.*
5	ST	Logika masukan pin yang digunakan untuk memulai *self-test*.
6	X_{OUT}	Data *output accelerometer*, untuk arah X.
7	STATUS	Logika keluaran untuk menujukan kesalahan.
8	V_{SS}	*The power supply ground*.
9	V_{DD}	*The power supply input*.
10	AV_{DD}	*Power supply input (Analog).*
11	Y_{OUT}	Data *output accelerometer*, untuk arah Y.
12 sampai 16	-	*Used factory trim. Leave unconnected*.
17 sampai 19	-	Tidak ada koneksi *internal*. *Leave unconnected*.
20	GND	*Ground.*

2.2.4 ADC (Analog to Digital Converter)

ADC (*Analog to Digital Converter*) digunakan untuk mengubah keluaran sensor yang masih berupa analog menjadi besaran digital. Resolusi pada ADC merupakan ketelitian nilai hasil konversi, ADC 8 bit akan memiliki output 8 bit data digital, ini berarti sinyal input dapat dinyatakan dalam 255 (2n -1) nilai

diskrit. Karena prinsip kerja ADC adalah mengkonversi sinyal analog ke dalam bentuk besaran yang merupakan rasio perbandingan sinyal input dan tegangan referensi. Sebagai contoh, bila tegangan referensi 5 volt, tegangan input 3 volt, rasio input terhadap referensi adalah 60%. Jadi, jika menggunakan ADC 8 bit dengan skala maksimum 255, akan didapatkan sinyal digital sebesar 60% x 255 = 153 (bentuk desimal) atau 10011001 (bentuk biner).

ADC yang digunakan adalah ADC0832, ADC ini merupakan sebuah ADC serial yang datanya langsung dapat diterima mikrokontroler pada satu pin saja. ADC ini memiliki resolusi sampai dengan 8 bit dengan 2 *channel* analog *multiplaxer*, dapat bekerja dengan *supply* tegangan sebesar 0-5 Volt. Berikut konfigurasi pin IC ADC0832 seperti pada gambar 2.6.



(a) (b)

Gambar 2.6. (a) Bentuk fisik IC ADC0832, (b) konfigurasi pin IC ADC0832

Spesifikasi yang dimiliki ADC0832 sebagai berikut:

1. Jangkauan *input* berkisar 0-5 volt dengan satu buah catu daya 5 *volt*
2. Mempunyai 2 *channel multiplexer* dengan 2 buah alamat logika.
3. Mudah *interface* untuk semua mikroprosesor.
4. Beroperasi dengan *link* data serial.
5. Mudah untuk digunakan bersama rangkaian mikroprosessor.
6. Tidak diperlukan penyesuaian yang rumit.

2.2.5 Penggerak Prototipe satelit

2.2.5.1 Brushless motor

Brushless motor merupakan motor yang mempunyai permanen magnet pada bagian "rotor" sedangkan elektro-magnet pada bagian "stator"-nya. Secara umum, kecepatan putaran *brushless* motor yang keluar dari ESC diatur oleh pulsa dari mikrokontroler, sehingga berbeda dengan *brushed*.



Gambar 2.7. *Brushless* motor

Keuntungan dari *brushless* motor sebagai berikut :

1. Komputer dapat mengatur kecepatan motor lebih baik sehingga membuat *brushless* motor lebih efisien.
2. Tidak adanya *storing* atau *electrical noise*.
3. Tidak menggunakan *brushes* yang dapat rusak setelah lamanya pemakaian.
4. Dengan posisi *electromagnets* dibagian *stator*, maka pendinginan motor menjadi lebih mudah.
5. Jumlah *electromagnets* di *stator* dapat sebanyak mungkin untuk mendapatkan kontrol yang lebih akurat.

2.2.5.2 ESC (Electronic Speed Control)

ESC (*Electronic Speed Control)* berfungsi sebagai pengatur kecepatan motor, juga untuk menaikan jumlah arus yang diperlukan oleh motor. ESC dapat dikatakan juga sebagai driver motor dengan mengeluarkan pulsa untuk *brushless* motor yang berasal dari mikrokontroler.



Gambar 2.8. ESC (*Electronic Speed Control*)

2.2.5.3 Propeler

Baling-baling atau propeler merupakan jenis kipas yang menghasilkan tenaga dengan mengkonversi gerakan rotasi menjadi daya dorong untuk menggerakkan sebuah benda atau kendaraan. Propeler tersebut adalah salah satu yang sering dipakai pada *aermodelling* untuk daya dorong.



Gambar 2.9. Propeler

2.2.6 Catu Daya dan Regulator

Catu daya merupakan bagian penting dalam perancangan prototipe satelit. Sehingga tanpa bagian catu daya, prototipe sateli tidak akan berfungsi. Pemilihan catu daya sangat penting karena jika salah dalam pemilihan maka prototipe satelit tidak akan berfungsi dengan maksimal.

Penentuan sistem catu daya yang akan digunakan ditentukan oleh beberapa faktor, diantaranya :

1. Tegangan

   Setiap aktuator tidak memiliki tegangan yang sama. Hal ini akan berpengaruh terhadap desain catu daya.

2. Arus

   Arus memiliki satuan Ah (*Ampere-hour*). Semakin besar Ah, semakin lama daya tahan baterai bila digunakan pada beban yang sama.

3. Teknologi Baterai

   Salah satu teknologi baterai pada saat ini yaitu baterai diisi ulang. Isi ulang

dapat dilakukan hanya apabila benar-benar kosong, dan ada pula yang dapat diisi ulang kapan saja tanpa harus menunggu baterai tersebut benar-benar kosong.

Baterai yang digunakan pada prototipe satelit ini berjenis *lithium polymer* (LiPo). Baterai ini dapat diisi ulang (*rechargeable*). Baterai yang digunakan memiliki tegangan 11,1 Volt dan arus sebesar 2200 mAh dengan 3 *cell* di dalamnya. *Cell* merupakan teknologi konversi energi elektrokimia yang mampu mengubah senyawa hidrogen dan oksigen menjadi air, dan dalam prosesnya menghasilkan listrik. Berikut ini adalah contoh sebuah *lithium polymer*.



Gambar 2.10. *Lithium Polymer*

Pemakaian baterai jenis ini harus dihentikan atau dilepas jika tegangan baterai turun mendekati batas tegangan 11,1 Volt, sehingga harus diisi ulang agar melebihi tegangan 11,1 Volt. Selain jenis baterai *lithium polymer* terdapat juga jenis baterai lainnya, antara lain baterai Ni-cd, Alkaline, Lead Acid dan sebagainya.

Komponen-komponen yang terdapat pada prototipe satelit seperti sensor membutuhkan regulator untuk mengatur tegangan yang dikeluarkan *lithium polymer* agar turun dan mempunyai nilai yang tetap. Karena komponen tersebut hanya dapat bekerja dengan tegangan tertentu yaitu di bawah 11,1 Volt. Untuk menurunkan tegangan yang tetap, maka digunakan IC regulator yaitu LM2940. Berikut bentuk fisik LM2940 :



Gambar 2.11. Bentuk fisik LM2940

2.2.7 Motor Servo

Sebuah motor *servo* adalah alat yang dapat mengendalikan posisi, dapat membelokkan dan menjaga suatu posisi berdasarkan penerimaan pada suatu *signal* elektronik itu. Pada perancangan prototipe satelit digunakan 2 buah motor *servo* pada bagian kanan dan kiri. Pada saat pra prototipe dijalankan motor *servo* akan mengangkat dan membentangkan bagian yang seperti sayap dengan beban *brushless* motor dan propeler.



Gambar 2.12. Hitec HS-5245MG

HS-5245MG adalah motor *servo* yang dibuat oleh *hitec*. Berikut merupakan spesifikasi dari motor *servo* tersebut :

1. Jenis motor adalah 3 *pole ferrite*
2. Jenis b*earing* adalah d*ual ball bearing*
3. Torsi 4.8/6.0v adalah 4,4kg atau 5.5kg
4. Speed 4.8/6.0 v adalah 0.15 detik atau 0.12 detik
5. Ukuran 32.4 X 16.8 X 30.8mm
6. Berat 32g

Karena motor *servo* merupakan alat untuk mengubah energi listrik menjadi energi mekanik, maka magnet permanen motor *servo* yang mengubah energi listrik ke dalam energi mekanik melalui interaksi dari dua medan magnet.



Gambar 2.13. *Servo standard*

Bagian-bagian dari sebuah motor *servo standard* adalah sebagai berikut:

1. Konektor yang digunakan untuk menghubungkan motor *servo* dengan Vcc, *Ground* dan *signal input* yang dihubungkan ke *basic stamp*.
2. Kabel menghubungkan Vcc, *Ground* dan *signal input* dari konektor ke motor *servo*.
3. Tuas menjadi bagian dari motor *servo* yang kelihatan seperti suatu bintang *four-pointed*. Ketika motor *servo* berputar, tuas motor *servo* akan bergerak ke bagian yang dikendalikan sesuai dengan program.

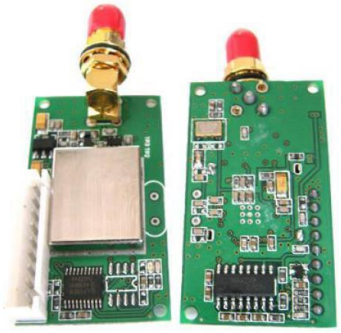
Cassing berisi bagian untuk mengendalikan kerja motor *servo* yang pada dasarnya berupa motor DC dan *gear*. Bagian ini bekerja untuk menerima instruksi dari *basic stamp* dan mengkonversi ke dalam sebuah pulsa untuk menentukan arah atau posisi motor *servo*.

2.2.8 Komunikasi Radio

Sistem komunikasi radio tidak menggunakan kabel dalam penyampaian informasi atau data, melainkan melalui udara sebagai pengantarnya. Dalam komunikasi radio memiliki sebuah pemancar Tx yang memencarkan dayanya menggunakan antena ke arah tujuan, sinyal yang dipancarkan berbentuk gelombang elektromagnetik. Pada penerima gelombang elektromagnetik ini diterima oleh sebuah antena yang sesuai. Sinyal yang diterima kemudian diteruskan ke sebuah pesawat penerima Rx. Jenis komunikasi dapat dibedakan berdasarkan aliran datanya, antara lain :

1. *Simplex communication* merupakan komunikasi satu arah, aliran data hanya satu arah. Contoh sistem komunikasi TV, Radio *broadcast*.
2. *Half duplex communication* merupakan komunikasi dua arah, data dapat mengalir kedua arah secara bergantian, hanya satu arah saja pada suatu saat. Contoh pada Sistem Walkie-talkies,
3. *Full duplex communication* merupakan komunikasi dua arah secara simultan, pada saat yang sama data mengalir ke kedua arah secara bersamaan. Contoh akses internet dan telepon lewat saluran TV *cable*, pada saat bersamaan.

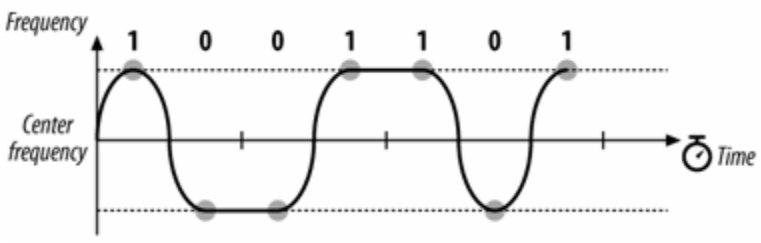
Pada perancangan prototipe satelit menggunakan *half duplex* dengan radio yang dipakai dalam pengiriman data ini adalah YS-1020UB.



Gambar 2.14. Modem Radio YS-1020UB

Modem radio ini mempunyai spesifikasi sebagai berikut :

1. Mempunyai 8 *channel* untuk pengiriman atau penerimaan data
2. Dapat menggunakan *level* TTL (*Transistor-Transistor Logic*).
3. Integrasi antara *receiver* dan *transmiter* memerlukan waktu 10 ms antara pengiriman dan penerima.
4. Tipe modulasi yang dipakai adalah GFSK (*Gaussian Frequensy Shift Keying*). GFSK adalah jenis modulasi Frequency Shift Keying (FSK) yang menggunakan Gaussian Filter untuk melancarkan penyimpangan frekuensi positif/negatif, yang diwakili dengan biner 1 atau 0. Gaussian Filtering merupakan salah satu cara yang sangat standar untuk mengurangi lebar spectral yang disebut dengan " pulse shaping ".



Gambar 2.15. Sinyal modulasi GFSK

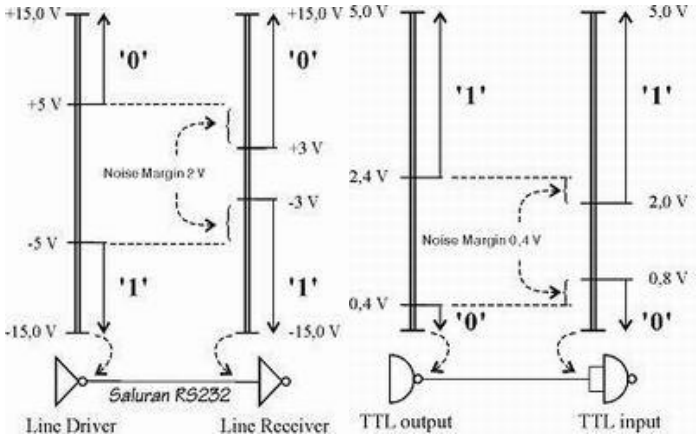
2.2.9 IC Max232

IC Max232 adalah sebuah IC yang berfungsi untuk mengubah tegangan dari TTL menjadi level RS232. Komunikasi pada radio menggunakan serial pada level TTL sedangkan komputer pada level RS232. Berikut tampilan dan rangkaian IC max232 :



Gambar 2.16. Bentuk fisik IC MAX232

Dalam *standard* RS232, tegangan antara +3 sampai +15 Volt pada *input line receiver* dianggap sebagai level tegangan ‘0’ dan tegangan antara –3 sampai –15 Volt dianggap sebagai level tegangan ‘1’. Agar *output line driver* bisa dihubungkan dengan baik, tegangan *output line driver* antara +5 sampai +15 Volt untuk menyatakan level tegangan ‘0’, dan antara –5 sampai –15 Volt untuk menyatakan level tegangan ‘1’.



(a) (b)

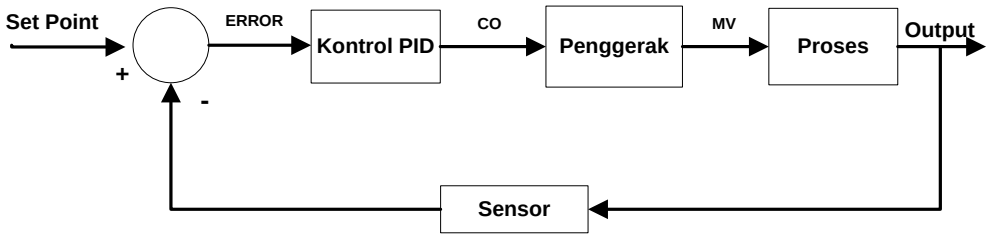
Gambar 2.17. (a) Level tegangan RS232, (b) Level tegangan TTL

IC digital, termasuk mikrokontroler, umumnya bekerja pada level tegangan TTL, yang dibuat atas dasar tegangan catu daya +5 Volt. Rangkaian input TTL menganggap tegangan kurang dari 0,8 Volt sebagai level tegangan ‘0’

dan tegangan lebih dari 2.0 Volt dianggap sebagai level tegangan ‘1’. Level tegangan ini sering dikatakan sebagai level tegangan TTL. Untuk menjamin output bisa diumpankan ke input dengan baik, tegangan output TTL saat level ‘0’ yaitu lebih rendah dari 0,4 Volt. Sedangkan tegangan output TTL pada saat level ‘1’ yaitu lebih tinggi dari 2,4 Volt

2.2.10 Pengontrol PID

Sistem kontrol merupakan proses pengendalian *error* dengan cara memasukkan nilai *error* tersebut untuk dibandingkan dengan sistem pengendalian. Tujuannya untuk mengurangi *error* tersebut dan menghasilkan *output* atau keluaran yang sesuai dengan *setpoint* yang akan dicapai. Pada perancangan prototipe satelit menggunakan cara pengontrolan PID *(Proporsional Integral dan Derivatif).* Secara umum, pengontrolan PID menggunakan *close loop* atau umpan balik yaitu program diolah pada mikrokontroler lalu menjalankan *actuator* setelah itu mengeluarkan *output*. Hasil *output* akan terus diperiksa oleh sensor agar dapat mencapai *setpoint* atau keluaran yang diinginkan. Jika tidak sesuai dengan *setpoint* yang diinginkan maka proses dari prototipe satelit akan terus berputar sampai bertemu dengan *setpoint* yang diinginkan seperti pada gambar 2.16 :



Gambar 2.18. Pengontrolan PID dengan *close loop*

Berikut adalah penjelasan gambar di atas:

1. *Setpoint* merupakan nilai proses yang diinginkan, terdapat dua pengaturan *setpoint* pada prototipe satelit yaitu dari PC (manual) atau sudah berada dalam program tanpa mengaturnya dari PC (*autonomous*).

2. *Error* merupakan nilai *set point* yang dikurangi dengan hasil pengukuran sensor.
3. Kontrol PID di atas menggunakan *close loop* atau umpan balik pada prosesnya dan terjadi pada mikrokontroler.
4. *Conroller Output* (CO) merupakan keluaran dari kontrol.
5. Penggerak merupakan elemen akhir sistem kontrol yang menerjemahkan sinyal kontrol yang keluar dari unit kontol ke plant atau proses, berupa *brushless* motor dan propeler.
6. *Manipulated variable* (MV) merupakan variabel yang besarannya secara langsung dapat dimanipulasi oleh kontroler.
7. Proses merupakan kejadian fisis yang variabelnya ingin dikontrol.
8. *Output* merupakan hasil keluaran dari proses. Hasil *output* akan terus diperiksa oleh sensor agar dapat mencapai *setpoint* yang diinginkan dan kesetimbangan yang tepat. Jika tidak sesuai dengan *setpoint* yang diinginkan maka proses dari benda yang dibuat akan terus berputar seperti pada blok diagram di atas, sehingga proses tersebut dapat dikatakan *cloose loop* (terjadi umpan balik untuk sensor) dan terjadi pada kontrol PID.
9. Sensor merupakan alat yang melakukan hubungan langsung dengan proses yang ingin ditinjau. Pada tahap ini sensor akan terus menerus memproses sampai *set point* yang diinginkan tercapai. Serta memeperkecil *error* yang terjadi sebelum masuk ke pengontrolan PID.

Berikut merupakan persamaan dari kontrol PID yang diperlihatkan pada persamaan:

$$CO(t) = K_p e(t) + K_I \int_0^t e(t) + K_D \frac{de(t)}{dt} \quad \text{.....2.1}$$

Keterangan :

$CO(t)$: *output* kontroler
K_p : *gain proporsional*
K_i : *gain integral*
K_d : *gain derivatif*
$e(t)$: *error*

Persamaan kontrol PID dapat diimplementasikan ke dalam bahasa pemrograman, berikut penjabaran persamaan kontrol PID ke dalam bahasa pemrograman.

1. Kontrol Proposional
   a. Persamaan proposional :

$$CO(t) = K_p e(t) \quad \text{.....2.2}$$

   b. Jika dalam bahasa pemrograman menjadi :

      P = KP*$error$($current$)

2. Kontrol Integral
   a. Persamaan integral :

$$CO(t) = K_I \int_0^t e(t) = K_I * [e(0) + e(1) + \cdots + e(t-1) + e(t) \quad \text{.....2.3}$$

   b. Jika dalam bahasa pemrograman menjadi :

      $error$($accummulator$) = $error$($accummulator$)+ $error$($current$)

      I=KI* $error$($accummulator$)

3. Kontrol Derivatif
   a. Persamaan derivatif :

$$CO(t) = K_D \frac{de(t)}{dt} = K_D \frac{e(t) - e(t-1)}{dt} \quad \text{.....2.4}$$

   b. Jika dalam bahasa pemrograman menjadi:

      $error$($delta$) = $error$(current)-$error$($previous$)

      D = KD*$error$($delta$)

      $error$($previous$) = $error$($current$)

Pada dasarnya aksi kontrol PID bertujuan untuk menggabungkan kelebihan-kelebihan yang dimiliki komponen PID. Komponen yang dimiliki antara lain:

1. Pengontrol *proporsional* berfungsi untuk mempercepat respon
2. Pengontrol *Integral* berfungsi untuk menghilangkan *error steady*.
3. Pengontrol *derivatif* berfungsi untuk memperbaiki sekaligus mempercepat *respon transient*.

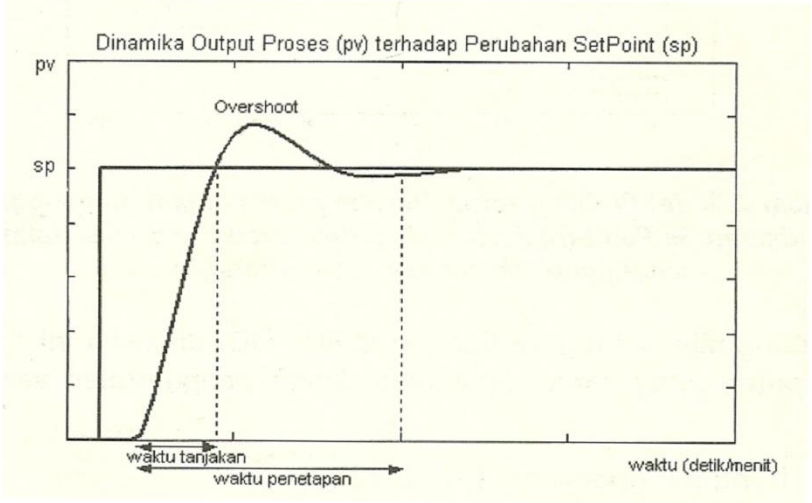
Dalam pengaturan *tunning* dalam kontrol PID akan berpengaruh terhadap kerja proses yang dikeluarkannya. Pengaruh dari tuning dapat dilihat pada tabel :

Tabel 2.3. Pengaruh tunning salah satu parameter PID terhadap kerja proses

Nilai	Waktu Tanjakan	Overshoot	Waktu Penetapan	Kestabilan
Pembesaran K_P	Berkurang	Bertambah	Sedikit bertambah	Menurun
Pembesaran K_I	Sedikit berkurang	Bertambah	Bertambah	Menurun
Pembesaran K_D	Sedikit berkurang	Berkurang	Berkurang	Meningkat

Dari tabel 2.3 dapat dijelaskan parameter PID terhadap kerja proses dan dapat dilihat pada gambar 2.18

1. Waktu tanjakan adalah waktu yang diperlukan respon untuk naik dari 0 sampai 100% harga akhirnya.
2. *Overshoot* adalah lonjakan maksimum yang dialami oleh respon proses.
3. Waktu penetapan adalah waktu yang diperlukan respon untuk mencapai dan menetap disekitar 95%-98% dari harga akhirnya.



Gambar 2.19. Parameter PID terhadap kerja proses

Dalam sistem kontrol PID terdapat fenomena *windup*. *Windup* adalah sebuah kejadian yang disebabkan terjadinya saturasi pada penggerak. Saturasi dapat terjadi karena beban yang dikontrol sudah di luar kemampuan penggerak.

Jika kontroler tidak memiliki anti *windup* dengan terjadinya beban berlebihan, maka *output* integrator pada kontroler PID akan terus menerus membesar. Untuk menghindari kejadian tersebut dalam kontroler dapat dipasang anti *windup*. Anti *windup* dapat diimplementasikan ke dalam bahasa pemrograman menjadi :

If (*input*<=min) then
Output = min
Else if (*input*>=max)
Output = max
Else
Output =input

2.3 Perangkat Lunak (Software)

2.3.1 Pengenalan Basic Stamp Editor

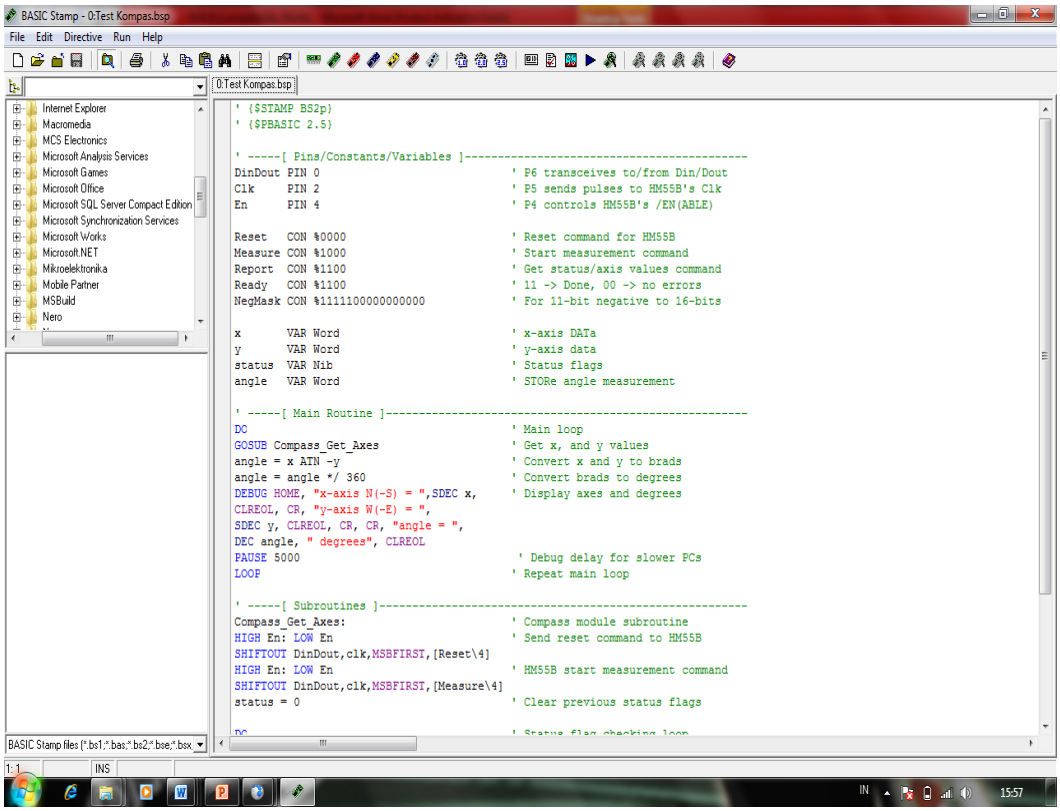
Perangkat lunak merupakan faktor penting dalam tahap perancangan prototipe satelit. Perangkat lunak ini merupakan algoritma atau *listing* program yang ditanamkan kedalam mikrokontroler. Program dapat bermacam - macam bentuk dan bahasanya sesuai dengan spesifikasi dari mikrokontroler yang digunakan.

Mikrokontroler *basic stamp* BS2P40 menggunakan bahasa pemrograman *basic*. *Software* yang digunakan adalah *basic stamp* editor. *Basic stamp editor* adalah sebuah *editor* yang dibuat oleh *Paralax Inc* untuk menulis *listing* program, meng-c*ompile* dan men-*download*-nya ke mikrokontroler keluarga *basic stamp*. Program ini memungkinkan penggunanya memprogram *basic stamp* dengan bahasa *basic* yang relatif ringan dibandingkan bahasa pemrograman lainnya. Berikut ini beberapa instruksi-instruksi dasar yang dapat digunakan pada mikrokontroler *basic stamp*.

Tabel 2.4. Beberapa instruksi dasar *basic stamp*

Instruksi	Keterangan
DO...LOOP	Perulangan
GOSUB	Memanggil prosedur
FOR...NEXT	Perulangan

PAUSE	Waktu tunda milidetik
IF...THEN	Perbandingan
PULSOUT	Pembangkit pulsa
PULSIN	Menerima pulsa
GOTO	Loncat ke alamat memori tertentu
HIGH	Menset pin I/O menjadi 1
LOW	Menset pin I/O menjadi 0
SEROUT	Mengirim data secara serial
SERIN	Menerima data secara serial



Gambar 2.20. Tampilan *basic stamp* editor

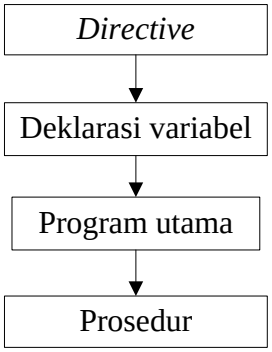
2.3.2 Memprogram Basic Stamp

Dalam membuat sebuah program secara umum, dapat dibagi menjadi empat bagian penting, yaitu :

1. *Header*
2. Variabel
3. Program utama

4. Prosedur

Pemrograman dalam *basic stamp editor*, secara blok dibagi menjadi empat bagian penting.



Gambar 2.21. Urutan bagian dari program dalam *basic stamp*

2.3.2.1 Directive

Directive ditulis paling awal listing program yang dibuat. Bagian ini menentukan *tipe prosesor* yang digunakan dan *versi dari compiler PBASIC* yang digunakan untuk meng-*compile bahasa basic* menjadi bahasa mesin. Tampilannya adalah seperti gambar berikut :

```
' {$STAMP BS2sx}     ' pemilihan tipe prosesor
' {$PBASIC 2.5}      ' pemilihan versi kompiler PBASIC
```

Gambar 2.22. Tampilan bagian *directive*

2.3.2.2 Menentukan Variabel

Beberapa ketentuan untuk mendeklarasikan variabel dalam mikrokontroler yaitu :

1. PIN : PIN dari mikrokontroler (0-15)
2. VAR : Variabel
3. CON : Konstanta

PIN yang digunakan sudah ditentukan sesuai dengan konfigurasi *hardware* / *mainboard* yang digunakan yaitu BS2P40.

```
' -----[ Pins/Constants/Variables ]-------
DinDout PIN 0
Clk     PIN 2
En      PIN 4

Reset   CON %0000
Measure CON %1000
Report  CON %1100
Ready   CON %1100
NegMask CON %1111100000000000

x       VAR Word
y       VAR Word
status  VAR Nib
angle   VAR Word
```

Gambar 2.23. Tampilan bagian deklarasi variabel

2.3.2.3 Bagian Program Utama

Pada bagian program utama bisa melakukan dua mode, yaitu mode pengetikan langsung atau mode pemanggilan prosedur. Mode pengetikan langsung akan efektif jika program tidak terlalu banyak dan kasus yang sederhana. Tetapi jika program sudah mulai banyak atau rumit, maka sebaiknya program utama memanggil prosedur. Pemanggilan prosedur akan mempermudah dalam pemeriksaan dan lebih terkendali. *Listing* programnya dapat dilihat pada gambar berikut.

```
' -----[ Main Routine ]------------------
DO
GOSUB Compass_Get_Axes
angle = x ATN -y
angle = angle */ 360
DEBUG HOME, "x-axis N(-S) = ",SDEC x,
CLREOL, CR, "y-axis W(-E) = ",
SDEC y, CLREOL, CR, CR, "angle = ",
DEC angle, " degrees", CLREOL
PAUSE 5000
LOOP
```

Gambar 2.24. Tampilan bagian program utama yang memanggil prosedur

2.3.2.4 Bagian Prosedur

Berikut adalah blok prosedur memperoleh data dari sumbu X dan Y yang dipanggil oleh program utama.

```
' -----[ Subroutines/procedure ]----------
Compass_Get_Axes:
HIGH En: LOW En
SHIFTOUT DinDout,clk,MSBFIRST,[Reset\4]
HIGH En: LOW En
SHIFTOUT DinDout,clk,MSBFIRST,[Measure\4]
status = 0

DO
HIGH En: LOW En
SHIFTOUT DinDout,clk,MSBFIRST,[Report\4]
SHIFTIN DinDout,clk,MSBPOST,[Status\4]
LOOP UNTIL status = Ready
SHIFTIN DinDout,clk,MSBPOST,[x\11,y\11]
HIGH En
IF (y.BIT10 = 1) THEN y = y | NegMask
IF (x.BIT10 = 1) THEN x = x | NegMask
RETURN
```

Gambar 2.25. Tampilan bagian prosedur

Sebuah prosedur harus mempunyai nama prosedur yang disimpan dibagian paling atas prosedur itu sendiri, serta harus diakhiri dengan *return* agar kembali lagi ke program utama dan melanjutkan kembali urutan program berikutnya.

2.3.2.5 Memeriksa Sintaks Program

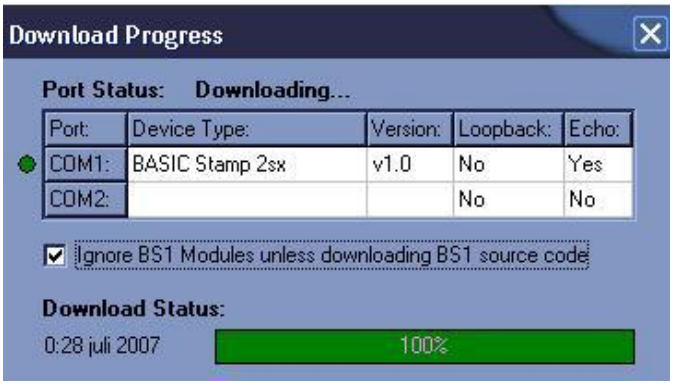
Sangat penting untuk memeriksa sintaks program, hal ini kita lakukan untuk memastikan semua *syntax* sudah benar. Untuk memeriksa sintaks ini bisa pilih menu *RUN*, Cek *Syntax* atau kombinasi tombol CTRL+T. Berikut ini adalah tampilan jika *listing* program yang kita buat sudah benar.



Gambar 2.26. Hasil pemeriksaan *syntax* yang sukses (*tokenize successful*)

2.3.3 Menjalankan Program

Setelah program selesai, program siap di-*download* ke modul *basic stamp*. Cara untuk menjalankan program dapat memilih menu RUN atau kombinasi tombol CTR+R. Berikut adalah tampilan jika men-*downlod* program sukses.

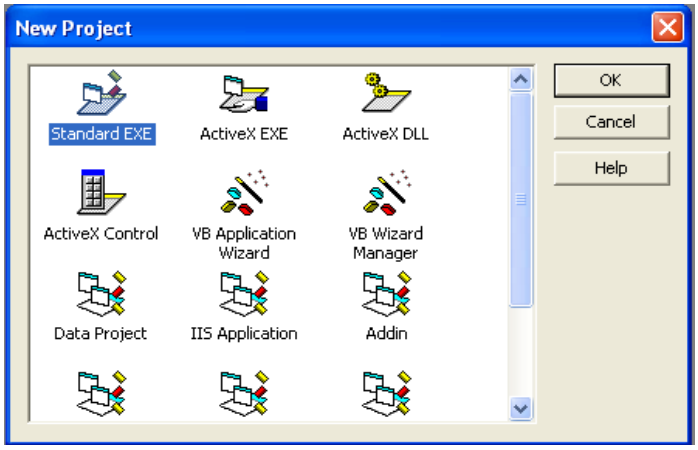


Gambar 2.27. Tampilan jika program sukses di-*download*

2.3.4 Visual Basic 6.0

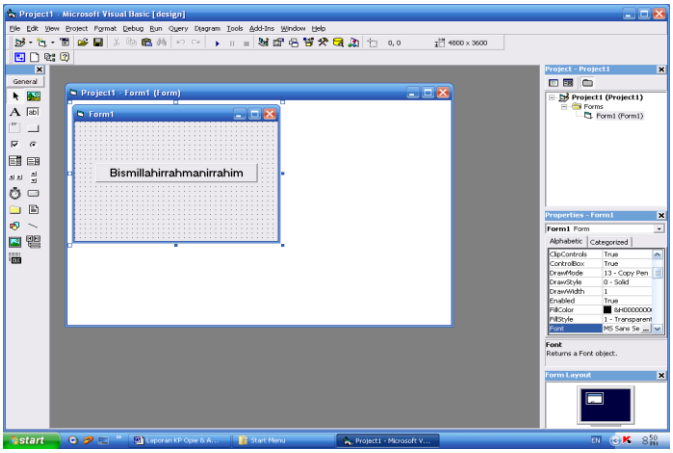
Visual basic 6.0 selain disebut sebagai bahasa pemrograman, juga sering disebut sebagai sarana untuk menghasilkan program-program aplikasi berbasiskan *windows*. *Visual basic* pada dasarnya adalah bahasa pemrograman komputer. Bahasa pemrograman adalah perintah-perintah atau instruksi-instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. *Visual basic* yang dikembangkan oleh *Microsoft* sejak tahun 1991 merupakan pengembangan dari pendahulunya yaitu bahasa pemrograman BASIC (*Baginners All-purpose Symbolic Instruction Code*) yang dikembangkan pada era 1950-an. Beberapa kemampuan atau manfaat dari *visual basic* diantaranya :

1. Untuk membuat program aplikasi seperti *windows.*
2. Untuk membuat objek-objek pembantu program seperti misalnya : kontrol *active*X, *file help*, aplikasi internet, dan sebagainya.
3. Menguji program (*debugging*) dan menghasilkan program EXE yang bersifat *executable*, atau dapat langsung dijalankan.



Gambar 2.28. Tampilan *new project* pada *visual basic*

Tampilan *Itegrated Development Environment* (IDE) pada sebuah *project Visual Basic* dengan sebuah *form*, label dan *command button* terlihat pada gambar di bawah ini :



Gambar 2.29. Tampilan IDE *Visual Basic*

Pada perancangan prototipe satelit *visual basic* berguna sebagai *interface* dan penghubung antara komputer dengan prototipe satelit. Sehingga *user* dapat mengontrol *setpoint* yang baru atau gerak prototipe satelit dengan tampilan di *visual basic*.