**BAB II**

**TEORI DASAR**

Pada bab ini dibahas tentang mekatronika, robot, motor servo, mikrokontroler ATMega8535, *CodeVisionAVR* dan *Visual Basic*.

* 1. **Mekatronika**

Mekatronika adalah teknologi atau rekayasa yang menggabungkan teknologi tentang mesin, elektronika, dan informatika yang bertujuan untuk merancang, memproduksi, mengoperasikan, dan memelihara sistem untuk mencapai tujuan yang diinginkan. Berdasarkan definisi tersebut dapat disimpulkan bahwa mekatronika adalah gabungan disiplin iptek teknik mesin, teknik elektro, teknik informatika, dan teknik kendali.

Pengertian mekatronika lebih mengarah pada teknologi kendali numerik yaitu mengendalikan mekanisme menggunakan aktuator untuk mencapai tujuan tertentu dengan memonitor informasi kondisi gerak mesin menggunakan sensor, dan memasukkan informasi tersebut ke dalam mikrokontroler/mikroprosesor. Teknologi mekatronika dapat mengubah skenario kontrol secara fleksibel dan dapat memiliki fungsi pengambilan keputusan tingkat tinggi. Hal inilah yang menjadi kelebihan mekatronika dibandingkan dengan kontrol otomatis menggunakan instrumen analog.

Berdasarkan penjelasan di atas struktur mekatronika dapat dibagi menjadi dua bagian yaitu mekanika dan elektronika. Mekanika terdiri dari mekanisme mesin sebagai objek yang dikendalikan, sedangkan elektronika terdiri dari sensor, kontroler, rangkaian penggerak, aktuator, dan sumber energi.

Dengan demikian mekatronika dapat diartikan juga sebagai ilmu yang mempelajari tentang pengontrolan sistem mekanik dengan menggunakan rangkaian-rangkaian elektronika. Rangkaian-rangkaian elektronika yang mengontrol sistem mekanik tersebut dinamakan kontroler.

Dewasa ini semua orang sering melihat barang-barang mekatronik seperti robot, mesin bubut NC, kamera digital, printer, dan lain-lain. Persamaan dari barang mekatronik itu adalah objek yang dikendalikan adalah gerakan mesin. Jika dibandingkan dengan gerakan mesin konvensional maka gerakan mesin tersebut lebih fleksibel dan lebih memiliki kecerdasan.

* 1. **Robot**

Istilah robot berasal dari kata *robota* (bahasa Czech) yang berarti bekerja. Kamus Webster mendefinisikan robot sebagai “perangkat automatik yang menunjukkan fungsi yang dapat dilakukan manusia”. Robot Institute of America mendeskripsikan istilah robot sebagai berikut: “Suatu robot adalah manipulator multi fungsi yang dirancang untuk memindahkan benda, peralatan atau perangkat khusus dengan pergerakan yang dapat diprogram secara berulang untuk menunjukkan berbagai macam tugas”.

Salah satu jenis robot yang sangat populer dan paling sederhana adalah sistem lengan robot (*robot arm system*). Sistem ini mengadopsi pergerakan lengan manusia, yang terdiri dari pangkal bahu, lengan, sendi dan telapak tangan. Pada dunia industri sistem ini banyak dipakai untuk memindahkan benda dari satu tempat ke tempat lain.

* + 1. **Klasifikasi Umum Robot**

Berdasarkan fungsinya robot dapat dikelompokan menjadi dua bagian, yaitu:

1. *Industrial* robot

*Industrial* robot digunakan pada industri manufaktur seperti membantu pekerjaan manusia di industri dalam proses perakitan kendaraan, mengelas dan sebagainya.

1. *Service* robot

*Service* robot adalah robot yang berfungsi untuk mempermudah manusia dalam melakukan tugasnya seperti membersihkan rumah, membantu dokter dalam melakukan operasi dan lain-lain.

Berdasarkan sifatnya robot dapat dikelompokan menjadi dua, yaitu:

1. *Fixed* robot

*Fixed* robot adalah robot yang memiliki ruang kerja yang terbatas (*spatial space*). Dikatakan terbatas, karena bagian dasar atau *base* diletakkan pada benda tetap seperti lantai, dinding, meja dan lain- lain. Bentuk *fixed* robot dapat dilihat pada gambar 2.1.



**Gambar 2.1 *Fixed* robot**

1. *Mobile* robot

*Mobile* robot adalah robot yang memiliki ruang kerja yang luas. Dikatakan luas, karena bagian dasar dari robot ini diletakkan alat untuk bergerak, seperti roda atau kaki. Bentuk dari *mobile* robot dapat dilihat pada gambar 2.2.



**Gambar 2.2 *Mobile* robot**

* + 1. **Robot Lengan**

Robot lengan adalah sebuah manipulator dengan tiga atau lebih sumbu, yang dikontrol secara otomatis, yang dapat diprogram ulang, dengan banyak tujuan. Robot lengan diletakkan pada tempat yang tetap atau dapat bergerak untuk keperluan dan aplikasi-aplikasi otomasi industri.

Manipulator adalah suatu struktur mekanik yang terdiri atas beberapa badan yang kaku (*link*), yang dihubungkan dengan sendi (*joint*). Manipulator terdiri atas lengan (*arm*) yang melakukan gerakan, pergelangan (*wrist*) yang memberikan kecekatan serta *end effector* yang melakukan tugas yang diinginkan, seperti misalnya pencekam (*gripper*).

* + 1. **Komponen Dasar Robot Lengan**

Sistem robot lengan memiliki empat komponen dasar, yaitu: manipulator, *end effector*, aktuator, dan kontroler.

1. Manipulator

Manipulator pada robot lengan memiliki 3 bagian, yaitu bagian dasar (*base*), bagian lengan (*arm*), dan bagian pergelangan (*wrist*). Bagian-bagian manipulator pada robot lengan dapat dilihat pada gambar 2.3.



Pergelangan (*wrist*)

Lengan (*arm*)

Dasar (*base*)

**Gambar 2.3 Manipulator pada robot lengan**

Bagian dasar (*base*) manipulator bisa kaku terpasang pada lantai area kerja ataupun terpasang pada rel. Rel berfungsi sebagai *path* atau alur sehingga memungkinkan robot untuk bergerak dari satu lokasi ke lokasi lainnya dalam satu area kerja. Bagian lengan (*arm*) berfungsi untuk memposisikan *end effector*, dan bagian pergelangan (*wrist*) berfungsi untuk mengatur orientasi dari *end effector.* Bagian ujung pada robot lengan terpasang *end effector* yang berfungsi sebagai fungsi kerja dari robot lengan.

1. *End Effector*

Efektor (*end effector*) dapat ditemukan hampir semua aplikasi robot, walaupun keberadaannya bukan merupakan komponen dasar dari sistem robot. Efektor berfungsi sebagai bagian terakhir yang menghubungkan antara manipulator dengan objek yang akan dijadikan kerja dari robot. Sebagai contoh efektor dapat berupa peralatan las, penyemprot cat ataupun hanya berupa pencekam objek. *End effector* berupa pencekam objek dapat dilihat pada gambar 2.4.



**Gambar 2.4 *End Effector* berupa pencekam objek**

1. Aktuator

Aktuator berfungsi sebagai sumber tenaga untuk menggerakkan manipulator. Aktuator pada robot dapat memakai sistem hidrolik, sistem pneumatik, motor DC, motor AC, motor stepper dan berbagai jenis penggerak lainnya. Aktuator jenis motor DC dapat dilihat pada gambar 2.5.



**Gambar 2.5 Aktuator jenis motor DC**

1. Kontroler

Kontroler merupakan jantung dari sistem robot sehingga keberadaannya sangat penting. Kontroler menyimpan informasi yang berkaitan dengan data-data robot, dalam hal ini data gerakan robot yang telah diprogram sebelumnya.

Kontroler berfungsi untuk mengontrol pergerakan dari manipulator. Kontroler sendiri diatur oleh sebuah informasi atau program yang diisikan dengan menggunakan bahasa pemrograman tertentu. Informasi tersebut kemudian disimpan didalam memori. Data dalam memori dapat di keluarkan atau di edit sesuai dengan yang dibutuhkan. Gabungan antara robot dan kontroler dapat diihat pada gambar 2.6.



**Gambar 2.6 Robot dan kontroler**

* 1. **Motor Servo**

Motor servo adalah sebuah alat yang dapat mengubah energi listrik menjadi energi mekanik. Motor servo sebenarnya digerakkan oleh sebuah motor DC. Karena motor servo digerakkan oleh motor DC maka jenis arus yang dipergunakan adalah jenis arus searah. Selain terdapat komponen motor DC, di dalam motor servo pun terdapat roda gigi yang berfungsi untuk mereduksi putaran dan untuk memperbesar torsi yang dihasilkan oleh motor DC tersebut.

Berbeda dengan motor DC yang akan langsung berputar ketika diberi tegangan, motor servo tidak akan berputar tanpa adanya perintah tertentu walaupun telah diberi tegangan. Motor servo dapat bergerak bila dialiri arus DC dan kaki sinyal diberi isyarat sinyal atau pulsa berupa arus listrik. Dengan diberi pulsa tertentu, motor servo akan berputar pada sudut tertentu.

* + 1. **Jenis – Jenis Motor Servo**

Jenis motor servo ada 2 yaitu jenis motor servo *continuous* dan motor servo *standard*. Kedua motor servo ini tidak jauh berbeda hanya saja pada putarannya. Berikut ini adalah penjelasan kedua jenis motor servo tersebut.

1. Motor Servo *Standard* 180°

Motor servo jenis ini hanya mampu bergerak dua arah (CW dan CCW) dengan masing-masing sudut mencapai 90° sehingga total sudut dari kanan – tengah – kiri adalah 180°. Motor servo *standard* lebih mudah dikontrol jika dibandingkan dengan motor servo *continuous* karena motor servo *standard* dapat diatur sudutnya sesuai dengan yang diinginkan (tidak berputar secara kontinyu). Contoh motor servo *standard* dapat dilihat pada gambar 2.7.

1. Motor Servo *Continuous*

Motor servo jenis ini mampu bergerak dua arah (CW dan CCW) tanpa batasan sudut putar (dapat berputar secara kontinyu).

 

**Gambar 2.7 Motor Servo *Standard* Hitec HS-311**

* + 1. **Konfigurasi Pin Motor Servo**

Motor servo hanya memiliki 3 kabel yang mana masing-masing fungsinya terdiri dari positif (Vcc), negatif (*Ground*) dan kontrol (*Signal*). Motor servo mampu bergerak searah jarum jam ataupun berlawanan arah jarum jam tanpa membalik pin konektor pada motor servo, hal ini disebabkan bahwa pada motor servo telah terdapat *driver* untuk membalik polaritas motor DC yang ada pada motor servo. Konfigurasi pin pada motor servo dapat dilihat pada gambar 2.8.



**Kuning (*Signal*)**

**Hitam (-)**

**Merah (+)**

**Gambar 2.8 Konfigurasi pin pada motor servo**

* + 1. **Prinsip Kerja Motor Servo**

Salah satu jenis motor servo adalah motor servo jenis standar. Motor servo jenis standar hanya dapat berputar 180° searah atau berlawanan arah jarum jam. Prinsip kerja motor servo standar adalah kaki sinyal motor servo diberi sinyal digital dengan lebar sinyal antara 0,60 milidetik sampai 2,00 milidetik. Sinyal akan dideteksi setiap 20 milidetik. Apabila dalam selang waktu lebih dari 20 milidetik motor servo tidak mendeteksi sinyal maka motor servo akan slip. Bentuk sinyal pengontrolan motor servo dapat dilihat pada gambar 2.9.



**Gambar 2.9 Bentuk sinyal pengontrolan motor servo**

Untuk memposisikan poros motor servo pada sudut tertentu dan selama waktu tertentu, lebar dan jumlah pulsa tertentu harus diberikan pada kaki sinyal motor servo. Agar motor servo dapat diposisikan pada sudut tertentu terlebih dahulu lebar pulsa untuk posisi 0° dan posisi 180° harus diketahui. Setelah mengetahui lebar pulsa pada kedua posisi tersebut, lebar pulsa untuk posisi sudut yang lain bisa diketahui dengan cara interpolasi.

Karena lamanya pulsa yang diberikan sangat singkat maka pemberian pulsa pada kaki sinyal tidak mungkin dilakukan secara manual. Selain itu sejumlah pulsa harus diberikan selama selang waktu tertentu. Oleh karena itu, pemberian pulsa pada kaki sinyal motor servo dilakukan dengan menggunakan mikrokontroler. Mikrokontroler akan terus memberikan sejumlah pulsa ke kaki sinyal pada motor servo sesuai dengan program. Skematis pengiriman pulsa dapat dilihat pada gambar 2.10.



**Gambar 2.10 Skematis pengiriman pulsa**

* 1. **Mikrokontroler ATMega8535**

Mikrokontroler merupakan perangkat elektronika yang didalamnya terdapat rangkaian kontrol, mikroprosesor, memori, dan *input/output*. Mikrokontroler dapat diprogram menggunakan berbagai macam bahasa pemrograman. Bahasa pemrograman yang biasa digunakan untuk memprogram mikrokontroler diantaranya adalah bahasa *assembler*, bahasa C, bahasa *basic* dan lain-lain.

Mikrokontroler biasanya digunakan untuk mengendalikan suatu proses secara otomatis seperti sistem kontrol mesin, *remote* kontrol, kontrol alat berat, kontrol robot dan lain-lain. Dengan menggunakan mikrokontroler sistem kontrol akan menjadi lebih ringkas, lebih mudah, dan lebih ekonomis.

Salah satu jenis mikrokontroler yang banyak digunakan untuk aplikasi kontrol adalah ATMega8535. ATMega8535 merupakan salah satu mikrokontroler keluaran Atmel. Atmel adalah salah satu vendor yang bergerak dibidang mikroelektrika. ATMega8535 memiliki beberapa fitur yang dapat digunakan untuk aplikasi kontrol. Skema dan bentuk mikrokontroler ATMega8535 dapat dilihat pada gambar 2.11.



**Gambar 2.11 Skema dan bentuk Mikrokontroler ATMega8535**

* 1. ***CodeVisionAVR***

Ada beberapa program yang dapat digunakan sebagai *editor* dan *compiler* untuk mikrokontroler, salah satunya adalah menggunakan program *CodeVisionAVR*. *CodeVisionAVR* adalah salah satu alat bantu pemrograman (*programming tool*) yang bekerja dalam lingkungan pengembangan perangkat lunak yang terintegrasi (*Integrated Development Environment*, IDE). Seperti aplikasi IDE lainnya, *CodeVisionAVR* dilengkapi dengan *source code editor*, *compiler*, *linker*, dan dapat memanggil Atmel AVR Studio untuk *debugger* nya. Tampilan awal pada program *CodeVisionAVR* dapat dilihat pada gambar 2.12.



**Gambar 2.12 Tampilan awal program *CodeVisionAVR***

* + 1. **Input/Output**

Fasilitas *input*/*output* merupakan fungsi mikrokontroler untuk dapat menerima sinyal masukan (*input*) dan memberikan sinyal keluaran (*output*). Sinyal *input* maupun sinyal *output* berupa data digital 1 (*high*) yang mewakili tegangan 5 volt dan data digital 0 (*low*) yang mewakili tegangan 0 volt. Mikrokontroler ATMega8535 memiliki 4 buah port 8 *bit bidirectional* yang dapat difungsikan sebagai port *input* maupun sebagai port *output*. Port-port tersebut adalah portA, portB, portC, dan portD.

Setiap port pada mikrokontroler mempunyai tiga buah *register bit*, yaitu DDRx.n, PORTx.n, dan PINx.n. huruf x mewakili nama port, sedangkan n mewakili nama *bit*. *Bit* DDx.n terdapat I/O *address* PORTx sedangkan *bit* PINx.n terdapat pada I/O *address* PINx. *Register* PORTx.n digunakan untuk mengaktifkan *pull-up* resistor (pada saat pin difungsikan sebagai *input*), dan memberi nilai keluaran pin *high*/*low* (pada saat difungsikan sebagai *output*).

* + 1. **Timer**

*Timer* merupakan fasilitas pada mikrokontroler yang dapat digunakan untuk mengukur selang waktu antara dua kejadian yang tidak bersamaan. *Timer* dapat dianalogikan sebagai suatu penampung yang diisi dengan suatu pulsa dengan kecepatan pengisian pulsa (frekuensi pengisian pulsa) tertentu. Ketika penampung tersebut penuh, penampung akan dikosongkan kembali dan mikrokontroler akan mengeksekusi suatu fungsi yang terkait dengan *timer*.

Pada mikrokontroler ATMega8535 terdapat tiga buah fasilitas *timer* yaitu *timer*0, *timer*1, dan *timer*2. *Timer*0 mempunyai kapasitas maksimum 8 bit, *timer*1 mempunyai kapasitas maksimum 16 bit, dan *timer*2 mempunyai kapasitas maksimum 8 bit. Frekuensi pengisian pulsa pada *timer* dapat dilihat pada tabel 2.1.

**Tabel 2.1 Frekuensi pengisian pulsa pada *timer***

|  |  |  |  |
| --- | --- | --- | --- |
| Frekuensi (kHz) | Timer0 | Timer1 | Timer2 |
| 16000 |  |  |  |
| 2000 |  |  |  |
| 500 | - | - |  |
| 250 |  |  |  |
| 125 | - | - |  |
| 62.5 |  |  |  |
| 15.625 |  |  |  |

Kapasitas *timer* dapat diatur dengan cara memberikan nilai awal pada *timer* tersebut. Bila *timer*0 diberi nilai awal 5, maka kapasitas *timer* tersebut yang semula 255 (8 bit) akan berubah menjadi 250. Bila frekuensi *timer* yang dipilih adalah 250 kHz, maka waktu yang diperlukan untuk mengisi *timer* adalah 0.001 detik atau 1 mili detik. Secara umum, waktu yang diperlukan untuk mengisi suatu *timer* mulai dari kondisi kosong sampai penuh dapat ditentukan dengan persamaan:

$$t = \frac{kapasitas timer-nilai awal}{frekuensi pengisian timer}$$

* 1. ***Visual Basic***

*Visual basic* adalah salah satu bahasa pemrograman komputer. Bahasa pemrograman adalah bahasa perintah yang dapat dimengerti oleh komputer untuk melakukan perintah-perintah tertentu. Bahasa pemrograman *visual basic* yang dikembangkan oleh *Microsoft*, merupakan pengembangan dari pendahulunya yaitu bahasa pemrograman BASIC (Beginner’s Allpurpose Symbolic Instruction Code) yang dikembangkan pada era tahun 1950-an. Program komputer yang menggunakan bahasa *basic* adalah *Microsoft Visual Basic* 6.0.

*Microsoft Visual Basic* 6.0 merupakan salah satu *development tool* yaitu alat bantu untuk membuat berbagai macam program komputer, khususnya program aplikasi berbasis sistem operasi *Windows*. Agar dapat menggunakan fasilitas *Microsoft Visual Basic* 6.0, perlu diketahui lingkungan kerja program tersebut yang didalamnya terdapat komponen-komponen utama. Komponen utama pada *visual basic* yang biasa digunakan yaitu *project*, *form*, dan *toolbox*.

* + 1. ***Project***

*Project* adalah sekumpulan modul yang didalamnya terdapat *form* beserta kodenya. *Project* dapat disimpan dalam *file* berformat .vbp. *File* ini akan menyimpan seluruh komponen program termasuk pilihan *project*, pilihan *environment*, dan pilihan *file* .exe. Pada jendela *project* terdapat tiga *icon*, yaitu *icon view code*, *icon view object* dan *icon toggle folder*.

*Icon view code* digunakan untuk menampilkan jendela editor kode program, *icon view object* digunakan untuk menampilkan bentuk *form*, dan *icon toggle folder* digunakan untuk menampilkan *folder* atau tempat penyimpanan *folder*. Tampilan *project* pada program *Visual Basic 6* dapat dilihat pada gambar 2.13.



**Toggle Folder**

**View Object**

**View Code**

**Gambar 2.13 Bentuk *project***

* + 1. ***Form***

*Form* adalah objek yang digunakan sebagai tempat perancangan program. Biasanya pada *form* terdapat garis titik-titik yang disebut *grid*. *Grid* sangat berguna untuk membantu pengaturan tata letak objek yang dimasukkan ke dalam *form*, karena gerakan penunjuk *mouse* akan sesuai (tepat) pada titik-titik *grid*. Bentuk *form* pada program *Visual Basic 6* dapat dilihat pada gambar 2.14.



**Gambar 2.14 Bentuk *Form***

* + 1. ***Toolbox***

*Toolbox* adalah kotak alat yang berisi *icon-icon* tertentu yang akan dimasukkan ke dalam jendela *form*. *Toolbox* dapat dimodifikasi misalnya menambah *icon* dengan cara meng-klik kanan pada *toolbox*, kemudian memilih *Components* atau *Add* tab. Beberapa objek yang biasa digunakan adalah *TextBox*, *CommandButton*, *ListBox*, *MSCommonDialog*, *MSChart* dan *MSComm*. Tampilan pada *toolbox* dapat dilihat pada gambar 2.15.



**Gambar 2.15 Komponen *toolbox***

* + - 1. ***Textbox***

*TextBox* merupakan objek pada *Visual Basic* yang biasanya digunakan untuk memasukkan nilai yang diperlukan oleh suatu aplikasi program. *Textbox* dapat juga digunakan untuk menampilkan suatu nilai hasil perhitungan maupun nilai suatu pengukuran.

* + - 1. ***CommandButton***

*CommandButton* merupakan objek yang digunakan untuk mengeksekusi perintah tertentu. *CommandButton* hampir selalu muncul pada semua aplikasi. Bentuk *CommandButton* tampak seperti segi empat dengan teks diatasnya.

* + - 1. ***MSComm***

*MSComm* merupakan objek yang digunakan untuk melakukan komunikasi serial. Objek ini disimbolkan dengan gambar telepon. Agar dapat melakukan komunikasi secara serial, beberapa properti objek *MSComm* perlu diubah. Beberapa properti tersebut adalah:

* *CommPort*, properti ini diisi dengan jalur komunikasi serial yang akan digunakan,
* *RThreshold*, properti ini diisi harga satu,
* *SRhreshold*, properti ini diisi harga satu, dan
* *Settings*, properti ini diisi sesuai dengan kecepatan pengiriman data dan jenis komunikasi serial yang dipilih.

Penerimaan data secara serial dilakukan dengan cara memindahkan karakter yang ada pada properti *input* ke memori. Sintaks penulisan perintah penerimaan data secara serial adalah buffer = MSComm1.Input.

Pengiriman data secara serial dilakukan dengan cara mengisi properti output dengan karakter yang dikirim. Sintaks penulisan perintah pengiriman data secara serial adalah MSComm1.Output = chr (angka).

Ketika komputer selesai mengirim atau menerima data secara serial, program akan secara otomatis mengeksekusi MSComm.OnComm(). Bila komputer selesai menerima data, maka harga properti ComEvent = ComEvReceive. Bila komputer selesai mengirim data, maka harga properti ComEvent = ComEvSend.