



MODUL PERKULIAHAN

Robotika

Apa itu Mekatronika?

Fakultas

Fakultas Teknik

Program Studi

Teknik Elektro

Tatap Muka

01

Kode MK

14043

Disusun Oleh

Julpri Andika, ST., M.Sc.

Abstract

Modul ini memperkenalkan mahasiswa tentang seluk beluk mekatronika. Pembahasan ditekankan kepada latar belakang, definisi, elemen system, manfaat dan aplikasi mekatronika.

Kompetensi

Mahasiswa memiliki wawasan tentang mekatronika dan tahu hal-hal yang berhubungan dengan mekatronika.

1.1. Latar Belakang

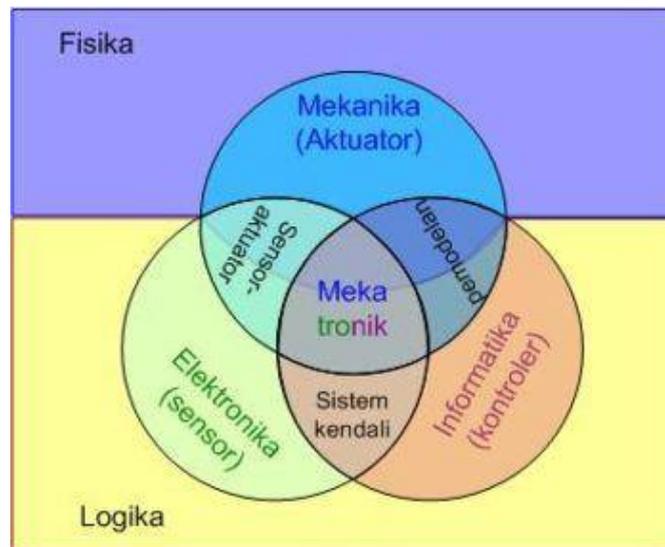
Mekatronika adalah sebuah kata baru yang lahir di Jepang pada awal tahun 1970-an yang merupakan gabungan antara 2 kata yaitu mekanika dan elektronika. Sekarang dapat dilihat di sekeliling kita, barang-barang mekatronika seperti robot, mesin bubut NC, kamera digital, printer dan lain sebagainya.

Persamaan prinsip dari barang-barang mekatronik ini adalah bahwa objek yang dikendalikan adalah gerakan mesin. Jika dibandingkan dengan gerakan mesin konvensional maka gerakan mesin tersebut lebih bersifat fleksibel dan lebih memiliki kecerdasan. Hal ini dimungkinkan karena memanfaatkan kemajuan bidang micro-electronics. Yang berarti, dengan bantuan peralatan micro-electronics mesin dapat bergerak dengan lebih cerdas. Jika seseorang memberikan sebuah perintah, lalu semua dapat dipasrahkan ke mesin yang dapat bergerak secara otomatis. Ini sangat membantu menciptakan mesin atau alat yang praktis dan mudah digunakan. Sehingga sumber daya pada manusia seperti waktu dan otak dapat dipakai untuk pekerjaan yang lain untuk lebih menciptakan nilai tambah.

Dari keterangan di atas, terdapat beberapa faktor yang menyebabkan timbulnya bidang mekatronika. Faktor-faktor tersebut antara lain adalah:

- a. Kemajuan pada bidang semikonduktor dan fabrikasi IC, yang mengarahkan perkembangan pada produk baru dengan cara mengintegrasikan sistem mekanik dan elektronik
- b. Perkembangan teknologi komputer (*microcomputer*, *embedded computer*), teknologi informasi dan perancangan perangkat lunak sehingga menjadikan mekatronika sebagai *leading technology* pada abad ke-21 ini.

Mekatronika (Inggris: *Mechatronic*) berasal dari kata mekanika, elektronika dan informatika. Secara sederhana pembentukan ilmu mekatronika terdiri atas dua lapisan ilmu dasar, yaitu: fisika dan logika serta tiga ilmu utama, yaitu: elektronika, informatika dan mekanika. Dengan melihat asal katanya dapat dengan mudah dipahami, bahwa ilmu ini menggabungkan atau mensinergikan disiplin ilmu Mekanika, ilmu Elektronika dan Informatika. Pendekatan ini diilustrasikan pada Gambar 1.1.



Gambar 1.1 Ilustrasi Asal Ilmu Mekatronika

Sementara itu, istilah *Mechatronics* (*Mechanical Engineering-Electronic Engineering*) pertama kali dikenalkan pada tahun 1969 oleh perusahaan Jepang *Yasakawa Electric Cooperation*. Pada awalnya, perusahaan ini berkembang dalam bidang *Feinwerktechnik*, yaitu cabang dari teknik yang mengedepankan aspek ketelitian. Misalnya pada pembuatan jam, alat optik dan sebagainya. Lalu ditambahkan sistem informatika, setelah munculnya informatika sebagai disiplin ilmu baru.

Pada awalnya, bidang mekatronik diarahkan pada 3 target yaitu:

- penghematan energi (*energi saving*),
- pengecilan dimensi dan peringan berat (*size reducing*), serta
- peningkatan kehandalahan (*reliability*).

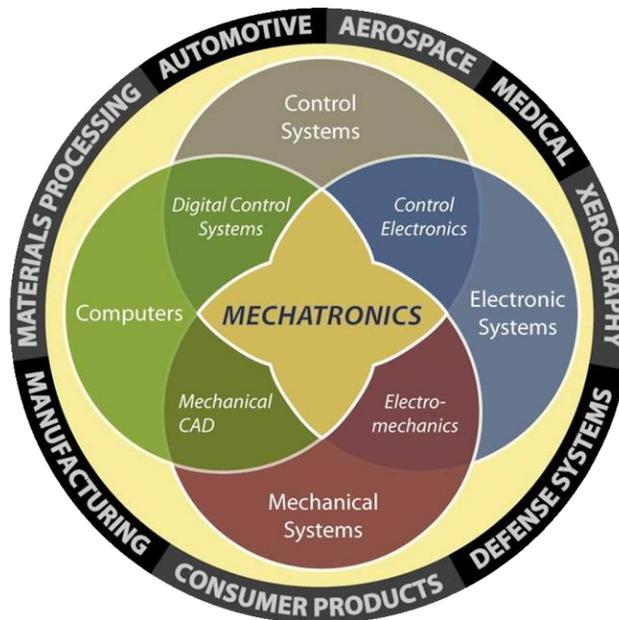
Sekarang, setelah 30 tahun lebih berlalu dari kelahirannya, perlu dirumuskan kembali arah mekatronik sesuai dengan perkembangan jaman. Dan khususnya untuk Indonesia sebagai negara yang masih berkembang dengan segudang permasalahannya, rasanya arah mekatronik perlu ditentukan agar dapat membantu memecahkan masalah-masalah yang ada dengan tetap memperhatikan lingkungan regional dan global.

Hingga saat ini, mekatronika dipandang sebagai hubungan antara ilmu Mekanik, Elektronik dan Informatika. Dalam masa yang akan datang, aplikasi mekatronika akan digunakan hampir disemua bidang, seperti Otomotif, Pemutar CD, Stasiun luar angkasa atau pada fasilitas produksi. Mekatronika dikategorikan oleh Majalah Technology Review pada tahun 2003 sebagai 10 Teknologi yang dalam waktu dekat dapat mengubah hidup kita!



1.2. Definisi

Menurut IEEE (*IEEE Mechatronics Transaction*, 1996), definisi mekatronika adalah sebagai berikut: *Mechatronics is the synergistic integration of mechanical engineering with electronics and intelligent computer control in the design and manufacturing of industrial products and processes*. Definisi ini dilengkapi dengan ilustrasi perkembangan mekatronika dalam bentuk diagram, seperti yang dapat dilihat pada Gambar 1.2.



Gambar 1.2 Definisi dan Keterkaitan Ilmu di Mekatronika

Sementara itu, berdasarkan hasil Musyawarah Nasional Mekatronika di Bandung 28 Juli 2006, Komunitas Mekatronika Indonesia merekomendasikan definisi Mekatronika sebagai berikut: *Mekatronika adalah sinergis IPTEK teknik mesin, teknik elektronika, teknik informatika dan teknik pengaturan (atau teknik kendali) untuk merancang, membuat atau memproduksi, mengoperasikan dan memelihara sebuah sistem untuk mencapai tujuan yang diinginkan*.

Dari definisi-definisi di atas, dapat disimpulkan beberapa bidang ilmu yang menyumbangkan peranannya dalam perkembangan ilmu mekatronika, adalah:

- Sistem Mekanika
- Sistem Elektronika
- Sistem Kontrol, dan
- Sistem Komputer

Penggabungan beberapa bidang di atas, menghasilkan beberapa ilmu baru, yaitu: Elektromekanika yang merupakan penggabungan ilmu mekanika dan elektronika, Kontrol

Elektronika yang memadukan ilmu elektronika dengan sistem kontrol, Sistem Kontrol Digital yang mempertemukan ilmu sistem kontrol dengan sistem komputer dan Mekanika CAD (*Computer Aided Design*) yang merupakan perpaduan ilmu mekanika dengan sistem komputer. Kemudian, integrasi keseluruhan ilmu tersebutlah yang menjadi akar dari perkembangan bidang mekatronika.

Secara sempit pengertian mekatronika mengarah pada teknologi kendali numerik yaitu teknologi mengendalikan proses mekanik menggunakan aktuator untuk mencapai tujuan tertentu dengan memonitor informasi kondisi gerak mesin menggunakan sensor, dan memasukkan informasi tersebut ke dalam mikro-prosesor. Ini menyumbangkan kemajuan yang spektakuler jika dibandingkan dengan kontrol otomatis menggunakan instrumen analog, karena dapat merubah skenario kontrol secara fleksibel dan dapat memiliki fungsi pengambilan keputusan tingkat tinggi.

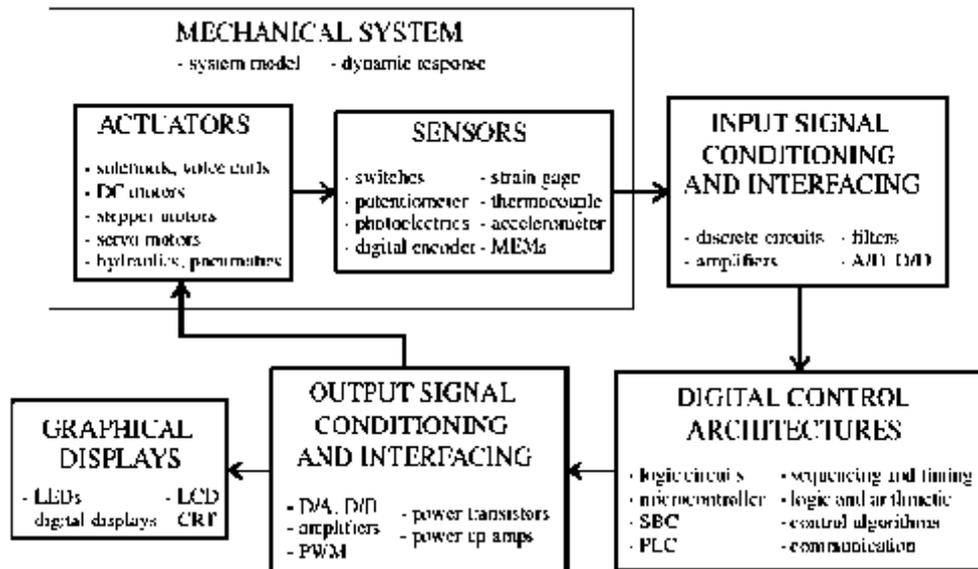
Contoh klasik barang mekatronik adalah lengan robot dan mesin bubut kontrol numerik. Barang-barang ini dapat melakukan pekerjaan-pekerjaan yang berbeda-beda dengan cara merubah program mereka sesuai kondisi yang diminta, karena telah ditambahkan kemampuan kendali aktif yang canggih terhadap mekanisme yang telah ada.

1.3. Sistem Mekatronika

Sistem mekatronika terdiri dari beberapa bagian sebagaimana digambarkan dalam blok diagram Sistem Mekatronik di Gambar 1.3. Dari gambar tersebut, sistem mekatronika terdiri dari: Aktuator, Sensor, Pengkondisian Signal dan Antarmuka, Pengendalian serta Display.

Pertama adalah blok Aktuator. Aktuator merupakan unsur penggerak sistem mekanik dari keseluruhan sistem. Sistem ini terdiri dari motor-motor, solenoida, dan komponen-komponen penggerak lainnya, baik yang digerakkan oleh elektronik, hidrolik atau pneumatik.

Setelah itu, terdapat blok Pengkondisian Signal dan Antarmuka. Blok ini terdapat pada bagian input dan juga output sebuah sistem. Setiap signal yang masuk akan dikondisikan sehingga dapat diproses oleh sistem dan akan dikondisikan kembali agar dapat ditampilkan kepada dalam bentuk output kepada pengguna. Dari blok satu ke blok yang lain, terjadi pengkondisian signal, sehingga seluruh sistem dapat berjalan dengan baik.



Gambar 1.3. Blok Diagram Sistem Mekatronika

Blok yang merupakan blok utama dalam sistem mekatronika adalah blok Pengendalian. Blok ini berintikan sebuah sistem pengendalian, yang pada umumnya menggunakan peralatan digital, seperti *Personal Computer (PC)*, *Programmable Logic Controller (PLC)*, mikroprosesor dan sejenisnya. Pengendali akan bekerja seperti otak manusia, dimana ia menerima hasil bacaan lingkungan dan perintah dari pengguna, mengolahnya dan menghasilkan respon dan keluaran yang sesuai dengan kehendak pengguna.

Terakhir, blok yang amat dekat dengan pengguna adalah blok penampil (Display). Blok ini merupakan antarmuka antara pengguna dengan keseluruhan sistem mekatronika, sehingga pengguna dapat melihat apa yang terjadi di dalam sistem dan menganalisa keluaran dari sistem tersebut. Blok Pengkondisian Signal dan Antarmuka, Blok Pengendali dan Blok Display merupakan blok-blok pembentuk sistem elektronika dari keseluruhan sistem mekatronika yang ada.

1.3. Manfaat Sistem Mekatronika

Beberapa manfaat penerapan mekatronik adalah sebagai berikut:

1. Meningkatkan fleksibilitas

Manfaat terbesar yang dapat diperoleh dari penerapan mekatronik adalah meningkatkan fleksibilitas mesin dengan menambahkan fungsi-fungsi baru yang mayoritas merupakan kontribusi mikro-prosesor. Sebagai contoh, lengan robot industri dapat melakukan berbagai jenis pekerjaan dengan merubah program peranti lunak di mikro-

prosesornya seperti halnya lengan manusia. Ini yang menjadi faktor utama dimungkinkannya proses produksi produk yang beraneka ragam tipenya dengan jumlah yang sedikit-sedikit.

2. Meningkatkan kehandalan

Pada mesin-mesin konvensional (manual) muncul berbagai masalah yang diakibatkan oleh berbagai jenis gesekan pada mekanisme yang digunakan seperti: keusangan, masalah sentuhan, getaran dan kebisingan. Pada penggunaan mesin-mesin tersebut diperlukan sarana dan operator yang jumlahnya banyak untuk mencegah timbulnya masalah-masalah tersebut. Dengan menerapkan switch semikonduktor misalnya, maka masalah-masalah akibat sentuhan tersebut dapat diminimalkan sehingga meningkatkan kehandalan. Selain itu, dengan menggunakan komponen-komponen elektronika untuk mengendalikan gerakan, maka komponen-komponen mesin pengendali gerak bisa dikurangi sehingga meningkatkan kehandalan.

3. Meningkatkan presisi dan kecepatan

Pada mesin-mesin konvensional (manual) yang sebagian besar menggunakan komponen-komponen mesin sebagai pengendali gerak, tingkat presisi dan kecepatan telah mencapai garis saturasi yang sulit untuk diangkat lagi. Dengan menerapkan kendali digital dan teknologi elektronika, maka tingkat presisi mesin dan kecepatan gerak mesin dapat diangkat lebih tinggi lagi sampai batas tertentu. Batas ini misalnya adalah rigiditas mesin yang menghalangi kecepatan lebih tinggi karena munculnya getaran. Hal ini melahirkan tantangan baru yaitu menciptakan sistem mesin yang memiliki rigiditas lebih tinggi.

1.5. Aplikasi Mekatronika

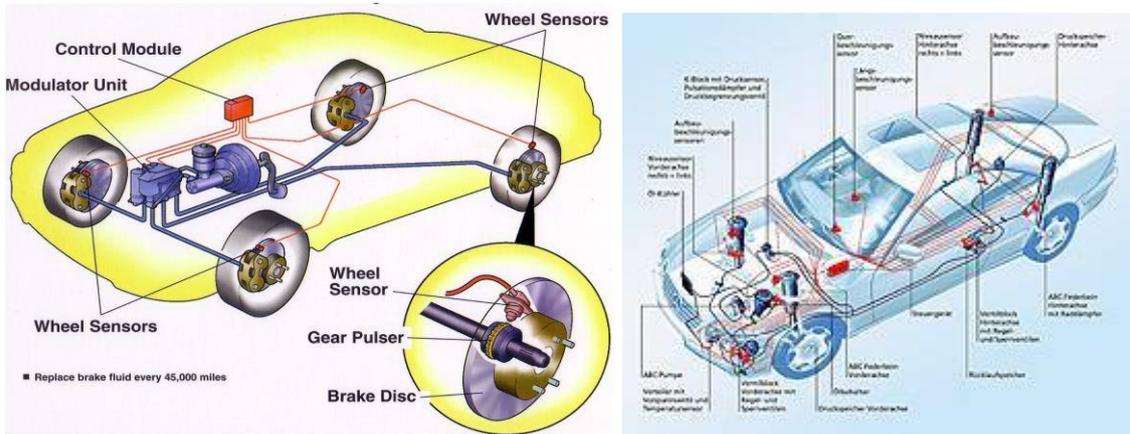
Begitu banyaknya penggunaan sistem mekatronika dalam kehidupan. Beberapa bidang yang merupakan perkembangan teknologi mekatronika adalah:

- Otomotif
- Aerospace
- Medikal
- Xerografi
- Sistem Pertahanan
- Produk-produk Konsumen
- Manufaktur
- Pemrosesan Material

Beberapa contoh dari aplikasi mekatronika adalah sebagai berikut:

a. Teknik Otomotif

Sebagai contoh sistem mekatronika pada kendaraan bermotor adalah sistem rem ABS (*Anti-lock Breaking system*) atau sistem pengereman yang menghindari terkuncinya roda sehingga mobil tetap dapat dikendalikan dalam pengereman mendadak, ESP (*Elektronik Stability Programm*), ABC (*Active Body Control*) dan Motor-Management-System.

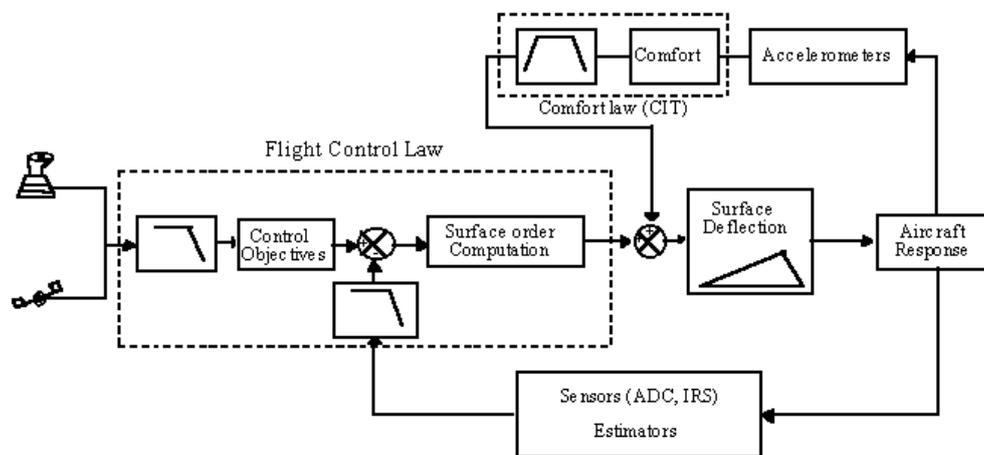


(a) (b)

Gambar 1.4. Mekatronika pada sistem otomotif
(a) Anti-lock Breaking System dan (b) Active Body Control

b. Teknologi Penerbangan

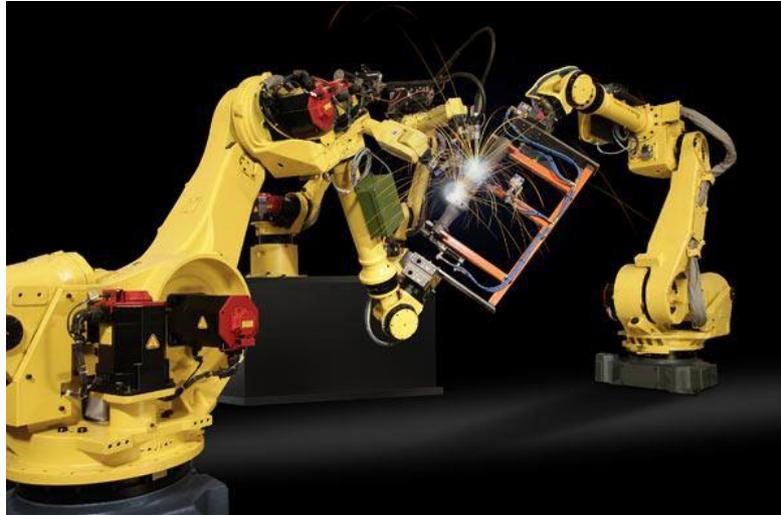
Dalam teknologi penerbangan modern digunakan *Comfort-In-Turbulence System* sehingga dapat meningkatkan kenyamanan penumpang walau ketika terjadi turbulensi. *Gust Load Alleviation* serta banyak contoh lainnya.



Gambar 1.5. Blok Diagram Confor In Turbulence System pada Pesawat

c. Teknik Produksi

Contoh dalam teknik produksi adalah penggunaan sensor pada robot. Sistem kendali umpan balik pada elektromotor berkecepatan rotasi tinggi dengan ‘pemegang as’ tenaga magnet.



Gambar 1.6. Sinkronisasi Pergerakan Robot-Majemuk

d. Alat-alat Perkantoran

Seperti pemutar CD, *Harddisk* serta mesin pencetak berkecepatan tinggi, atau alat-alat elektronika yang biasa kita gunakan sehari-hari aplikasi mekatronika akan sangat sering kita jumpai.



(a)



(b)

Gambar 1.7. (a) Pemutar CD dan (b) Hard Disk, Peralatan Mekatronika yang terdapat di dalam Komputer Pribadi



(a)



(b)

Gambar 1.8. Peralatan Perkantoran yang menggunakan Prinsip Mekatronika (a) Mesin Photocopy dan (b) Mesin Fax

Daftar Pustaka

1. G. C. Onwobolu, *Mechatronics: Principles and Application*, Elsevier Butterworth-Heinemann, UK, 2005.
2. C. W. de Silva, *Mechatronics: A Foudation of Course*, CRS Press, NY, USA, 2010



MODUL PERKULIAHAN

Robotika

So, Robotika itu Apa?

Fakultas

Fakultas Teknik

Program Studi

Teknik Elektro

Tatap Muka

02

Kode MK

MK14043

Disusun Oleh

Julpri Andika, ST., M.Sc.

Abstract

Modul ini menjelaskan robotika sebagai salah satu aplikasi praktis dari Mekanika. Penertian, konstruksi umum dan jenis-jenis robot akan dibahas.

Kompetensi

Mahasiswa memiliki wawasan tentang robotika dan tahu hal-hal yang berhubungan dengan robotika. Selain itu, mahasiswa juga mampu menjelaskan konstruksi umum sebuah robot sederhana.

2.1. Mekatronika dan Robotika

Seperti sudah disebutkan pada modul terdahulu, bahwa terdapat banyak sekali aplikasi mekatronika dalam kehidupan sehari-hari. Salah satu aplikasi mekatronika yang tampak dengan jelas adalah bidang robotika. Karena, dalam bidang robotika ini, terlihat ilmu-ilmu, elemen-elemen sistem dan blok-blok yang sama dengan mekatronika. Robotika memanfaatkan ilmu-ilmu dari bidang mekanika, elektronika, sistem kontrol dan sistem komputer digital, seperti mekatronika. Selain itu, sebuah robot akan terdiri dari sistem mekanika, sistem sensor, sistem pengkondisian signal dan antar-muka, sistem pengendalian serta sistem display, sebagaimana yang ada pada sistem mekatronika. Jadi, dapat dikatakan bahwa robotika adalah aplikasi yang kongkrit dari mekatronika.

Dengan demikian, ketika seseorang sedang mempelajari sistem robotika atau sedang mencoba untuk merancang sebuah sistem robotika, berarti dia sedang mempelajari dan sedang merancang sebuah sistem mekatronika. Diharapkan, seseorang yang mampu merancang sebuah robot sederhana, maka ia akan mampu pula merancang sebuah sistem mekatronika yang kompleks.

2.2. Pengertian Robot

Robot adalah sebuah alat elektro-mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dulu melalui kecerdasan buatan. Robot biasanya digunakan untuk tugas yang berat, berbahaya, pekerjaan yang berulang dan kotor. Biasanya kebanyakan robot industri digunakan dalam bidang produksi. Penggunaan robot lainnya termasuk untuk pembersihan limbah beracun, penjelajahan bawah air dan luar angkasa, pertambangan, pekerjaan "cari dan tolong" (*search and rescue*), dan untuk pencarian tambang. Belakangan ini robot mulai memasuki pasaran konsumen di bidang hiburan, dan alat pembantu rumah tangga, seperti penyedot debu, dan pemotong rumput.

Kata *robot* berasal dari kata *robota* terdapat pada sebuah teater fiksi sains berjudul *Rossum's Universal Robots* (RUR), karya Karel Capek dari Czech (1921) yang berarti *bekerja* atau *pekerja*. Kemudian, istilah robot berkembang dari waktu ke waktu sesuai dengan teknologi yang berkembang pada waktu tersebut.



Gambar 2.1. Contoh Aplikasi Robot dalam Proses *pick and place*

Secara formal, robot didefinisikan sebagai berikut: *Robot is an automatic device that performs functions ordinarily ascribed to human beings* (Webster's Dictionary) atau *any automatically operated machine that replaces human effort, though it may not resemble human beings in appearance or perform functions in humanlike manner* (Encyclopaedia Britanica). Definisi ini merujuk kepada fungsi utama robot pada waktu itu, yaitu sebagai alat bantu otomatis untuk mengerjakan pekerjaan-pekerjaan manusia yang sulit, berbahaya dan berulang.

Kemudian, definisi ini berkembang sejalan dengan pengaplikasian robot di dunia industri yang semakin kompleks dan beragam. Definisi robot berdasarkan pendekatan ini adalah: *An industrial robot is a reprogrammable and multifunctional manipulator designed to move materials, parts, tools, or special devices through variable programmed motions for the performance of variety tasks.* (Robotics Industry Association, RIA, 1985). Secara teknologi, definisi robot pada waktu ini sudah memasukkan unsur 'reprogrammable'. Artinya, pada waktu itu telah dikembangkan suatu alat yang mampu memprogram dan dapat diprogram ulang yang digunakan sebagai 'otak' dari robot untuk mengerjakan pekerjaan-pekerjaan industri. Kemudian, diketahui bahwa alat tersebut adalah mikroprosesor.

Setelah berkembang pula teknologi material/bahan, sensor dan ilmu kecerdasan buatan (*artificial intelligence*), maka definisi robot pun berubah pula. Beberapa definisi robot yang ada pada waktu terakhir ini adalah sebagai berikut: *A robot is a system that interacts with the world already, and learning as method of improving its performance* (Brooks, 91) dan *A robot is a machine able to extract information from its environment and use knowledge about its world to move safely in a meaningful and purposive manner* (Arkin, 98). Pada waktu ini, robot sudah mampu berinteraksi dengan lingkungannya dan mengambil informasi darinya, untuk kemudian melakukan proses pembelajaran sendiri sehingga mampu meresponnya dalam bentuk suatu tindakan dalam rangka mengerjakan fungsi tertentu. Artinya, robot sudah harus mampu untuk berinteraksi dan mengambil informasi

dari lingkungannya melalui sistem sensor tertentu. Selain itu, pada robot juga sudah memiliki sistem kecerdasan buatan berupa algoritma tertentu dalam mikroprosesornya untuk menentukan tindakan yang akan diambil olehnya.

Dari beberapa definisi diatas, dapat ditarik kesimpulan berupa beberapa sifat dan karakteristik robot masa kini, yaitu:

- bergerak tanpa harus dikendalikan langsung oleh manusia
- bergerak secara multi-aksis (rotasi dan translasi)
- dapat diprogram ulang
- dapat mengambil keputusan tertentu secara otomatis
- dapat berinteraksi, mengambil informasi dan memanipulasi lingkungannya
- memiliki sistem kecerdasan buatan

Untuk menghindari hal-hal yang dikhawatirkan oleh manusia akibat 'perkembangan' yang pesat dari robot, maka sejak awal telah dibuat Tiga Hukum Robot. Tiga Hukum Robot dalam genre cerita fiksi ilmiah adalah tiga buah peraturan yang ditulis oleh Isaac Asimov, yang harus dipatuhi oleh hampir semua robot-robot positroniknya, yang terdapat dalam karya-karya cerita fiksinya. Meskipun dalam berbagai cerita sebelumnya pernah disebutkan secara selintas, Tiga Hukum Robot pertama kali diperkenalkan secara lengkap pada tahun 1942 dalam cerita pendek "Runaround", yang menyatakan sebagai berikut:

1. Robot tidak boleh melukai manusia, atau dengan berdiam diri, membiarkan manusia menjadi celaka (*A robot may not injure a human being or, through inaction, allow a human being to come to harm*).
2. Robot harus mematuhi perintah yang diberikan oleh manusia kecuali bila perintah tersebut bertentangan dengan Hukum Pertama (*A robot must obey orders given to it by human beings except where such orders would conflict with the First Law*).
3. Robot harus melindungi keberadaannya sendiri selama perlindungan tersebut tidak bertentangan dengan Hukum Pertama atau Hukum Kedua (*A robot must protect its own existence as long as such protection does not conflict with the First or Second Law*).

Belakangan, Asimov menambahkan Hukum Ke-Nol: "Robot tidak boleh mencelakakan umat manusia, atau dengan berdiam diri, membiarkan umat manusia menjadi celaka" (*A robot may not harm humanity, or, by inaction, allow humanity to come to harm*); hukum-hukum selanjutnya dapat disesuaikan secara berturut-turut, untuk mengakomodasi hukum ini.

2.3. Konstruksi Robot

Terdapat berbagai macam jenis robot. Pada umumnya, robot diklasifikasikan terdiri dari 2 (dua) jenis, yaitu: robot industri (*industrial robot*) dan robot bergerak (*mobile robot*). Klasifikasi ini dilihat dari sisi perkembangan dan struktur robot tersebut.

Robot industri adalah robot yang awal-awal dibangun oleh para peneliti. Robot ini memiliki ciri khas sebagai berikut:

- Banyak didominasi oleh sistem mekanikal
- Konstruksi dasar (*base*) robot tetap (*stationary*) berbentuk tangan manusia
- Menggunakan sensor yang jumlah dan variasi terbatas
- Menggunakan sistem mikroprosesor dan dapat diprogram ulang

Sedangkan robot bergerak adalah kelanjutan dari proses pembangunan robot.

Robot bergerak ini memiliki ciri khas sebagai berikut:

- Banyak didominasi oleh sistem elektrik
- Konstruksi dasar (*base*) robot yang bergerak dan beroda atau berkaki
- Menggunakan sensor yang banyak jumlah dan variasinya
- Menggunakan sistem mikroprosesor yang memiliki kecerdasan buatan

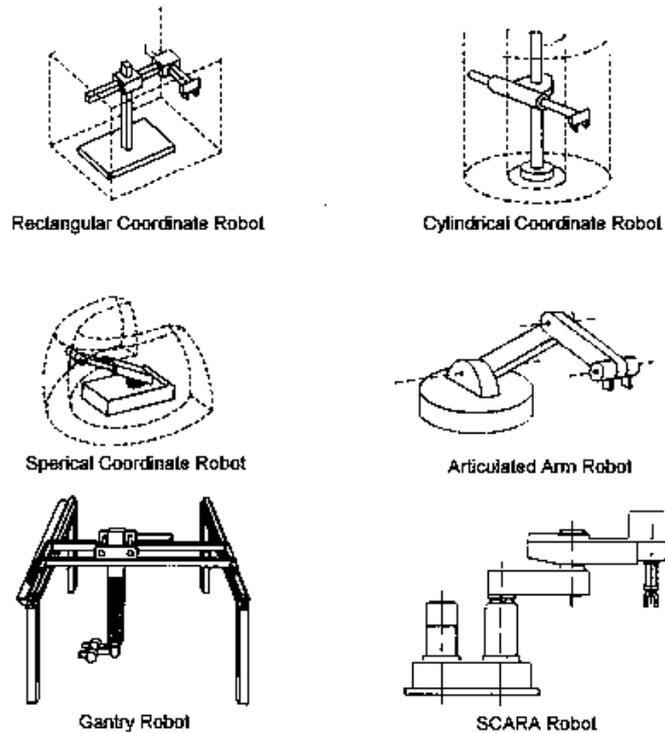
Namun, seiring dengan perkembangan teknologi, robot bergerak berkembang dengan sangat pesat. Perkembangan ini meliputi sisi konstruksi turunannya, kemampuan gerak dan areal pergerakannya serta pengoperasiannya.

Berikut ini akan dibahas secara global masing-masing jenis konstruksi robot tersebut.

2.4. Robot Industri

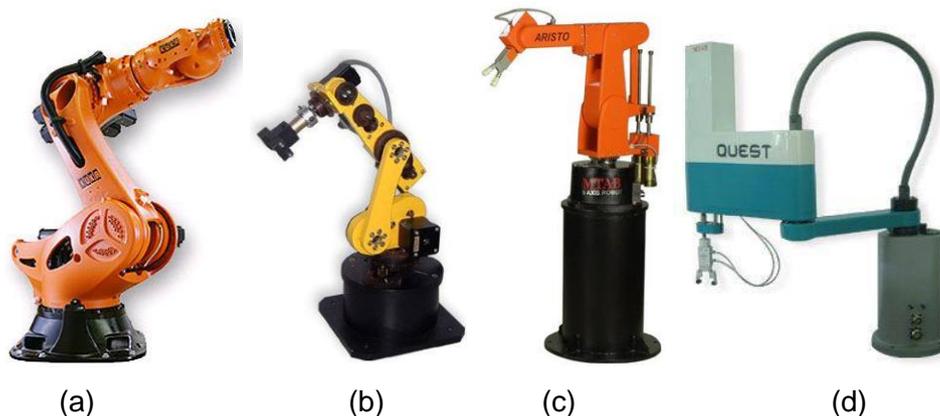
Sebagaimana dikatakan di atas, bahwa robot industri merupakan robot yang pertama-tama dibangun oleh para peneliti. Robot ini mirip dengan tangan manusia yang bekerja untuk pekerjaan yang sebelumnya dilakukan oleh manusia. Karena konstruksinya yang berbentuk tangan, robot ini juga sering disebut dengan istilah *Arm Robot*.

Terdapat beberapa jenis robot industri berdasarkan areal kerjanya (*envelope work*), yaitu: *rectangular coordinate robot*, *cylindrical coordinate robot*, *spherical coordinate robot*, *articulate arm robot*, *gantry robot* dan *scara robot*, sebagaimana yang tampak pada Gambar 2.2



Gambar 2.2. Robot Industri

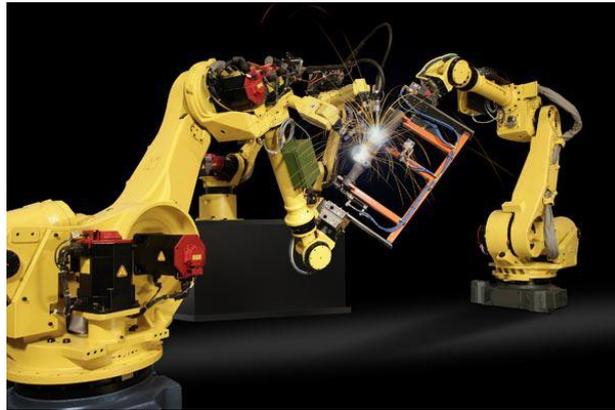
Beberapa contoh robot industri yang biasa digunakan dalam proses industri secara aplikatif tampak pada Gambar 2.3.



Gambar 2.3 Contoh Robot Industri: (a) dan (b) Spherical Robot, (c) Articulated Arm Robot dan (d) Scara Robot

Robot industri juga berkembang dengan cepat. Salah satu bentuk perkembangan robot industri adalah pengaplikasian robot secara bersama (*sychromotion*). Robot diatur sedemikian rupa sehingga bergerak secara serempak sehingga proses industri berjalan lebih

efektif dan efisien. Salah satu contoh pasangan robot industri yang bekerja secara *sychromotion* ditampilkan pada Gambar 2.4.



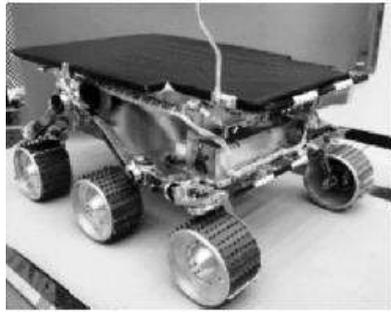
Gambar 2.4. *Sychromotion* Robot

2.5. Robot Beroda

Ketika para pencipta robot pertama kali mencoba meniru manusia dan hewan, mereka menemukan bahwa hal tersebut sangatlah sulit; membutuhkan tenaga penghitungan yang jauh lebih banyak dari yang tersedia pada masa itu. Jadi, penekanan perkembangan diubah ke bidang riset lainnya. Robot sederhana beroda digunakan untuk melakukan eksperimen dalam tingkah laku, navigasi, dan perencanaan jalur. Teknik navigasi tersebut telah berkembang menjadi sistem kontrol robot otonom yang tersedia secara komersial.

Robot Beroda atau *Wheeled Mobile Robot* adalah konstruksi robot yang ciri khasnya mempunyai aktuator berupa roda untuk menggerakkan keseluruhan badan robot tersebut, sehingga robot tersebut dapat melakukan perpindahan posisi dari satu titik ke titik yang lain. Robot beroda ini sangat disukai bagi orang yang mulai mempelajari robot. Hal ini karena membuat robot mobil tidak memerlukan kerja fisik yang berat. Untuk dapat membuat sebuah robot beroda minimal diperlukan pengetahuan tentang mikrokontroler dan sensor-sensor elektronik.

Base robot beroda dapat dengan mudah dibuat dengan menggunakan plywood /triplek, akrilik sampai menggunakan logam aluminium. Robot beroda dapat dibuat sebagai pengikut garis (*Line Follower*) atau pengikut dinding (*Wall Follower*) ataupun pengikut cahaya. Beberapa robot beroda diperlihatkan pada Gambar 2.5.



(a)



(b)

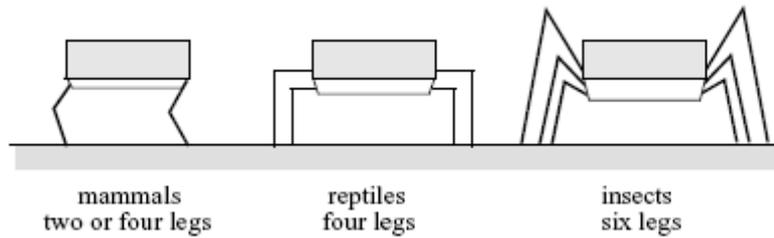


(c)

Gambar 2.5. Robot Beroda: (a) Robot Beroda 6, (b) Robot Beroda (4) dan (c) Robot Beroda dengan Belt

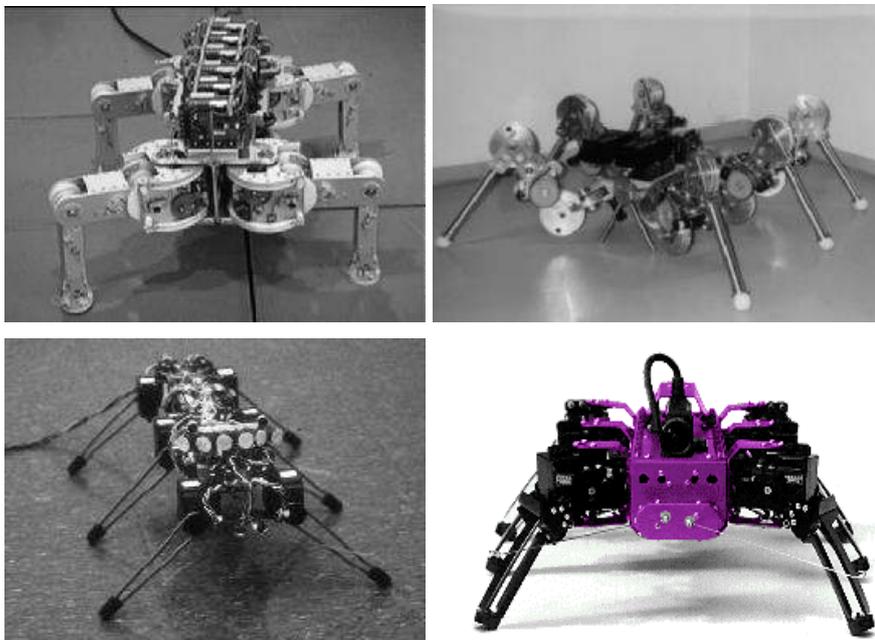
2.6. Robot Berkaki

Ketika para teknisi siap untuk mencoba robot berjalan kembali, mereka mulai dengan heksapoda dan platform berkaki banyak lainnya, yang dikenal dengan istilah *Legged Robot*. Robot-robot tersebut meniru serangga dan arthropoda dalam bentuk dan fungsi. Tren menuju jenis badan tersebut menawarkan fleksibilitas yang besar dan terbukti dapat beradaptasi dengan berbagai macam lingkungan, tetapi biaya dari penambahan kerumitan mekanikal telah mencegah pengadopsian oleh para konsumen. Dengan lebih dari empat kaki, robot-robot ini stabil secara statis yang membuat mereka bekerja lebih mudah. Prinsip robot berkaki dapat dilihat pada Gambar 2.6 di bawah ini.



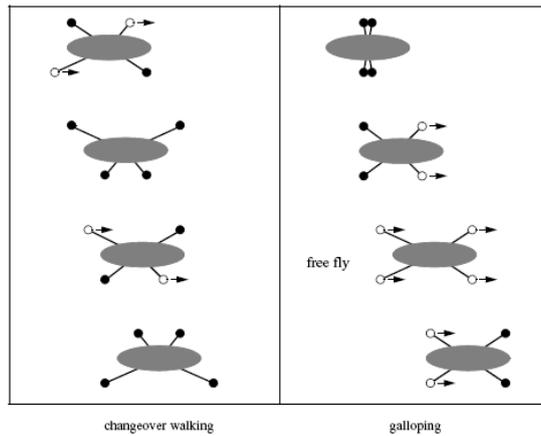
Gambar 2.6. Robot Berkaki

Kelebihan robot ini dibanding dengan robot beroda adalah kemampuannya berjalan di berbagai permukaan. Sehingga, untuk penggunaan di permukaan yang tidak datar, seperti berbatu, berumput, berpasir dan permukaan yang tidak terkirakan sebelumnya, digunakan robot berkaki. Beberapa jenis robot berkaki, seperti berkaki 4 atau berkaki 6 diperlihatkan pada Gambar 2.7 berikut ini.



Gambar 2.7. Robot Berkaki

Namun, kekurangan robot ini adalah kerumitan pemrogram untuk pergerakan robot. Karena, harus dikendalikan dengan cermat pergerakan setiap kakinya sehingga dapat terjadi pergerakan robot secara keseluruhan sesuai yang diinginkan. Pengaturan kaki ini dikenal dengan istilah *gait analysis*. Contoh *gait analysis* dapat dilihat pada Gambar 2.8 yang merupakan pergerakan robot berkaki 4.

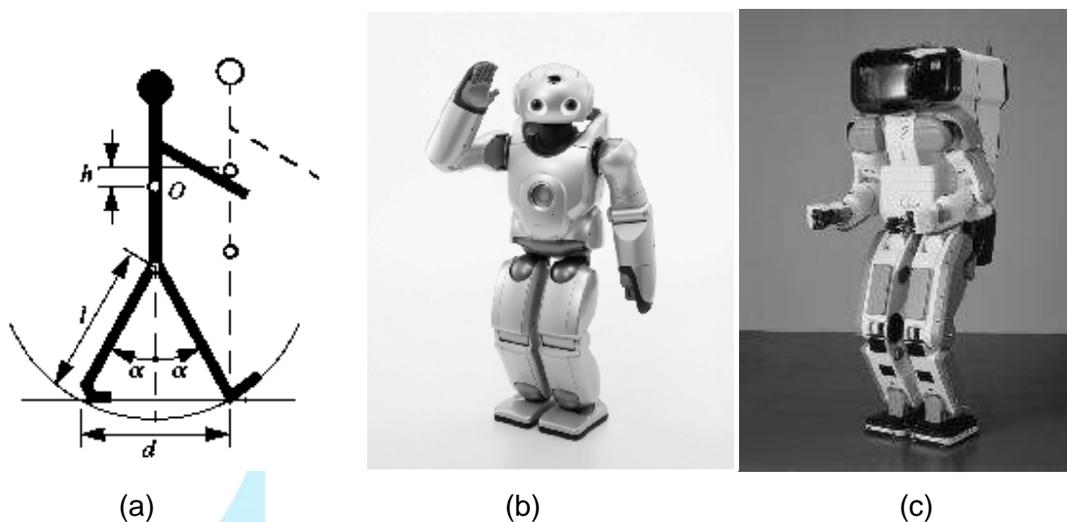


Gambar 2.8. Gait Analysis Robot Berkaki 4

2.7. Humanoid

Robot bergerak yang kesulitannya tinggi adalah robot berkaki dua, atau dikenal dengan istilah *humanoid*, karena mirip seperti manusia. Tujuan dari riset robot berkaki dua adalah mencapai gerakan berjalan menggunakan gerakan pasif-dinamik yang meniru gerakan manusia. Kesulitannya adalah disamping harus mengatur gerakan agar robot dapat berjalan dengan baik juga harus diperhitungkan keseimbangan dan kestabilan robot yang terus berubah, termasuk sumber daya yang harus disiapkan secara ringan untuk dapat dibawa oleh robot.

Hingga saat ini telah terdapat beberapa humanoid yang ditampilkan oleh beberapa industri. Salah satu humanoid yang cukup terkenal adalah ASIMO yang telah memiliki beberapa versi. Contoh humanoid dapat dilihat pada Gambar 2. 9.



Gambar 2.9. Humanoid : (a) Prinsip, (b) dan (c) Versi-versi Humanoid

2.8. Robot Fungsi Khusus

Selain robot-robot diatas, telah dirancang pula berbagai jenis robot yang memiliki fungsi tertentu. Robot-robot tersebut antara lain adalah :

a. Tour Guide Robot

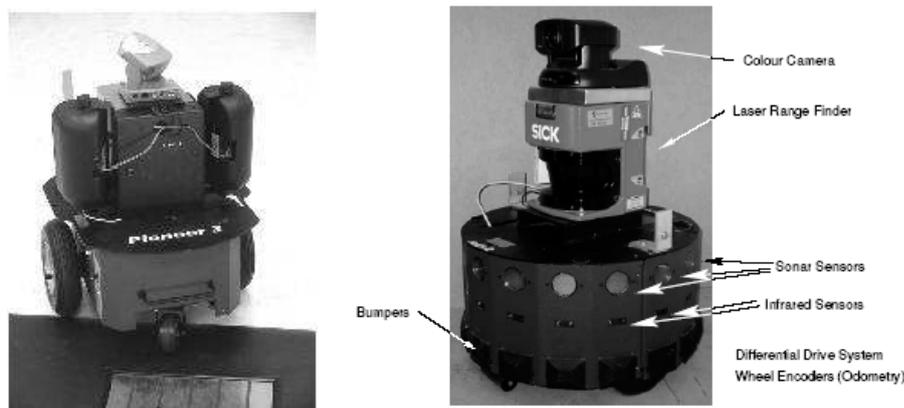
Robot yang berfungsi untuk mengantar pengunjung melihat-lihat pameran, museum atau galery. Sehingga tidak diperlukan lagi manusia untuk menjelaskan isi dari sebuah tempat.



Gambar 2.10. Tour Guide Robbot

b. Robot Laboratorium

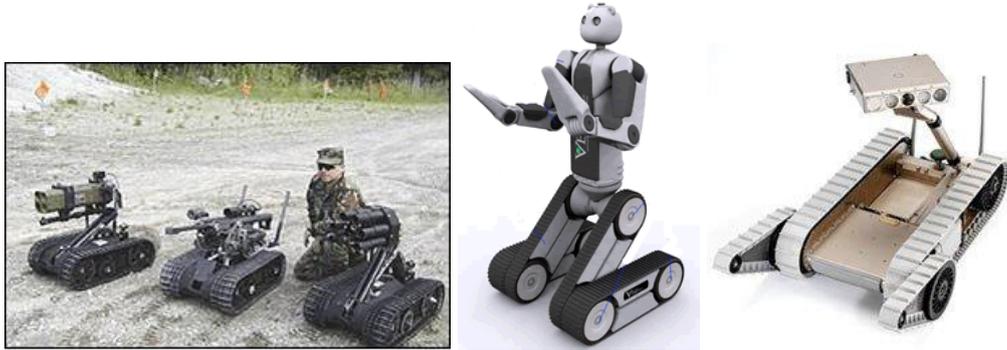
Robot yang digunakan di lab sebagai bahan pengajaran dan praktikum pelajar atau mahasiswa. Pada umumnya, robot ini telah dilengkapi dengan berbagai macam sarana, seperti berbagai jenis sensor, mikroprosesor, camera dan lain-lain. Mahasiswa atau pelajar hanya perlu memprogram robot tersebut agar dapat berfungsi sesuai yang dikehendaki.



Gambar 2.11. Robot Laboratorium

c. Robot Militer

Robot-robot ini diharapkan dapat menjadi bahan pengkajian dan strategi militer, dimana robot dijadikan sebagai tentara, atau sistem patroli, pencarian korban perang / bencana alam dan lain-lain.



Gambar 2.12 Robot-robot Militer

d. Robot Masa Depan

Beberapa riset telah mencoba mengaplikasikan robot ke aplikasi yang lebih rumit. Beberapa bidang pengaplikasian ini antara lain: bidang elektro-medis, bidang bio-medis. Contoh dari aplikasi ini adalah robot bedah, pencarian penyakit menggunakan mini-robot, implementasi robot pada tubuh manusia dan lain-lain.



Gambar 2.13. Robot untuk Bio-Electro-Medis

Daftar Pustaka

1. K. S. Fu, R. C. Gonzales, C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*, Mc Graw-Hill Book Company, US, 1987.
2. J. G. Keramas, *Robot Technology*, Delmar Publisher, 1998.



MODUL PERKULIAHAN

Robotika

Mikrokontroler

Fakultas

Fakultas Teknik

Program Studi

Teknik Elektro

Tatap Muka

03

Kode MK

MK14043

Disusun Oleh

Julpri Andika, ST., M.Sc.

Abstract

Modul ini menjelaskan salah satu jenis komponen utama robot, yaitu mikrokontroler. Arsitektur umum, elemen-elemen yang ada serta teknik penggunaan mikrokontroler akan dijelaskan juga.

Kompetensi

Mahasiswa memiliki wawasan yang jelas tentang mikrokontroler serta mampu untuk merancang sebuah system minimum mikrokontroler.

3.1. Apa itu Mikrokontroler

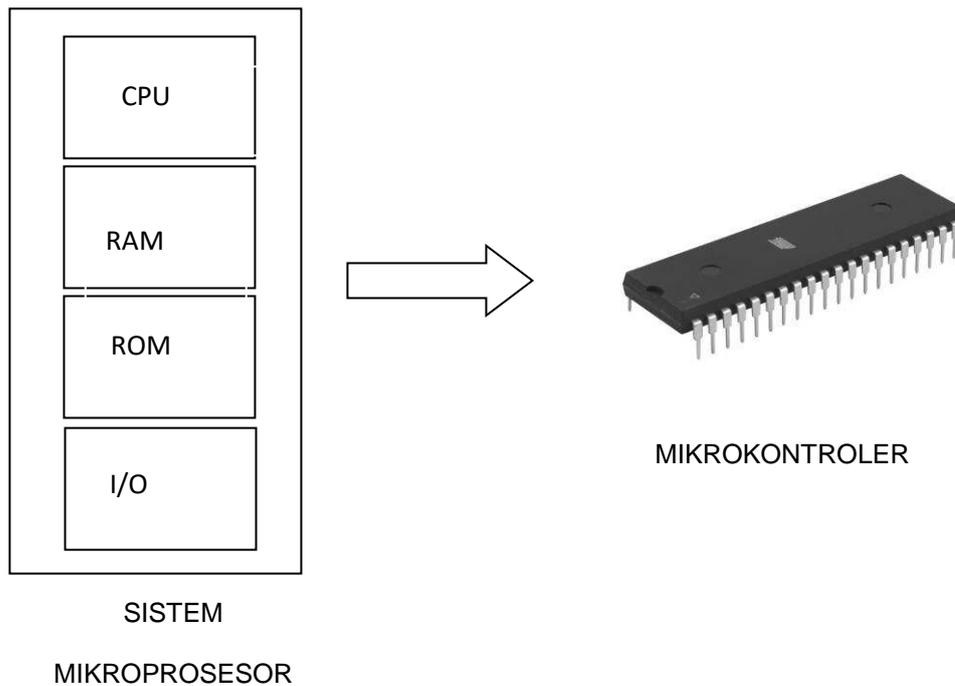
Mikrokontroler adalah komputer berpiranti chip tunggal (*single chip computer*) yang sangat murah. Piranti chip tunggal maksudnya adalah keseluruhan sistem yang diperlukan untuk sebuah komputer telah tersedia dalam satu chip saja berupa material semikonduktor silikon yang telah terenkapsulasi dalam suatu keping rangkaian terpadu (*Integrated Circuit, IC*). Sebuah mikrokontroler dapat serupa dengan sebuah komputer pribadi (*Personal Computer, PC*).

Sebuah mikrokontroler berisikan sebuah CPU (*Central Processing Unit*), serangkaian memori: RAM (*Random Access Memory*) dan ROM (*Read-Only Memory*), jalur I/O (*Input/Output*), port-port serial and parallel, beberapa mode *timer*, dan beberapa peripheral tambahan seperti konversi Analog ke Digital (A/D) dan konversi Digital ke Analog (D/A). Jadi, sebuah mikrokontroler yang berbentuk kecil dan fleksibel ini dapat mengendalikan suatu perangkat yang dengan mudah dimampatkan ke dalam sebuah system mekatronika. Piranti ini juga dapat dengan mudah deprogram untuk menjalankan fungsi-fungsi yang luas dalam kerja-kerja pengendalian. Fitur kunci dari mikrokontroler adalah kemampuannya untuk memasukkan, menyimpan dan menjalankan sebuah program tertentu.

Dibandingkan dengan IC yang semirip dengan mikrokontroler ini, yaitu mikroprosesor, maka jelas penggunaan mikrokontroler akan lebih ekonomis, efisien dan efektif. Karena, ketika menggunakan mikroprosesor, yang tersedia baru sebuah CPU (*Central Processing Unit*). Bagian-bagian yang lain harus menggunakan piranti-piranti tersendiri, yaitu IC memori, baik RAM maupun ROM, IC I/O, IC A/D serta D/A dan dibuat sendiri jalur-jalur antara blok yang satu dengan blok-blok yang lain. Sedangkan sebuah chip mikrokontroler umumnya sudah memiliki fitur:

- *Central Processing Unit* - mulai dari prosesor 4-bit yang sederhana hingga prosesor kinerja tinggi 64-bit.
- input/output antarmuka jaringan seperti port serial (UART)
- antarmuka komunikasi serial lain seperti I²C, Serial Peripheral Interface and Controller Area Network untuk sambungan sistem
- periferal seperti timer dan watchdog
- RAM untuk penyimpanan data
- ROM, EPROM, EEPROM atau Flash memory untuk menyimpan program komputer
- pembangkit clock - biasanya berupa resonator rangkaian RC
- pengubah analog-ke-digital

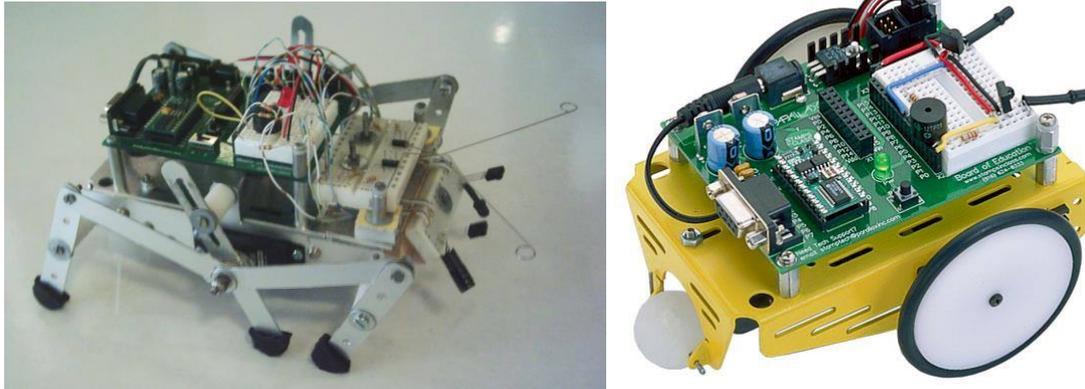
Perbandingan mikroprosesor dengan mikrokontroler diperlihatkan pada Gambar 4.1.



Perbandingan Mikroprosesor dan Mikrokontroler

Selain kemudahannya, sistem mikrokontroler juga mudah untuk diaplikasikan ke perancangan sistem elektronika, baik sistem elektronika sederhana maupun sistem elektronika kompleks. Kemampuannya untuk menyimpan dan menjalankan program membuat sistem ini amat berguna. Misalnya, seseorang dapat menggunakan mikrokontroler untuk sistem yang harus mengambil keputusan dengan segera atau berfungsi berdasarkan situasi dan keadaan tertentu. Biasanya, keputusan-keputusan tersebut dapat berupa fungsi-fungsi matematika atau logika. Dapat juga keputusan tersebut menggunakan sistem Jaringan Syaraf Tiruan (*Artificial Neural Network*) atau Pengendali Logika Fuzzy (*Fuzzy Logic Controller*). Mikrokontroler telah menjadi bagian utama dalam peralatan elektronika dan menjadi 'otak' pada sistem elektronika yang cerdas.

Saat ini, begitu banyak artikel-artikel elektronika yang menampilkan penggunaan mikrokontroler pada sistem elektronika, termasuk sistem mekatronika. Oleh karena, pemahaman tentang mikrokontroler menjadi amat esensial sebagai landasan untuk dapat merancang sebuah sistem mekatronika.

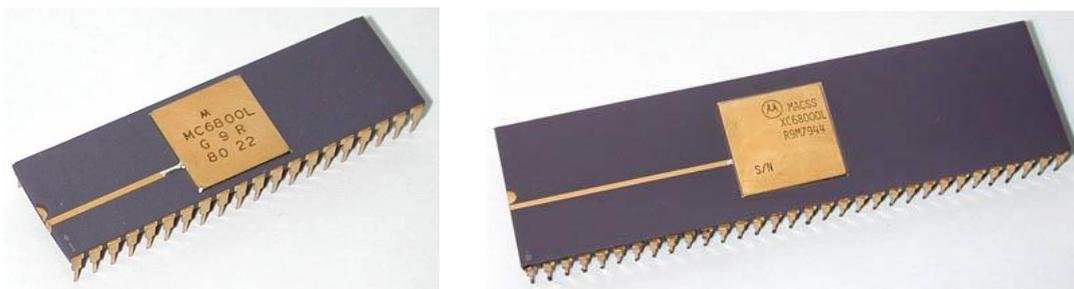


Gambar 4.2. Penggunaan Mikrokontroler pada Robot

4.2. Jenis-jenis Mikrokontroler

Saat ini, begitu banyak jenis, variasi dan merek-merek mikrokontroler yang tersedia di pasaran. Paling tidak, terdapat 3 (tiga) produsen besar yang memproduksi mikrokontroler, yaitu: Intel, Motorola, Microchip (dengan merek PIC) dan Atmel.

Motorola merupakan produsen yang cukup lama di dunia IC. Setelah lama memproduksi mikroprosesor, seperti mikroprosesor seri 6800, Motorola juga memproduksi mikrokontroler seri 68000. Namun, jenis mikrokontroler ini jarang digunakan di Indonesia.

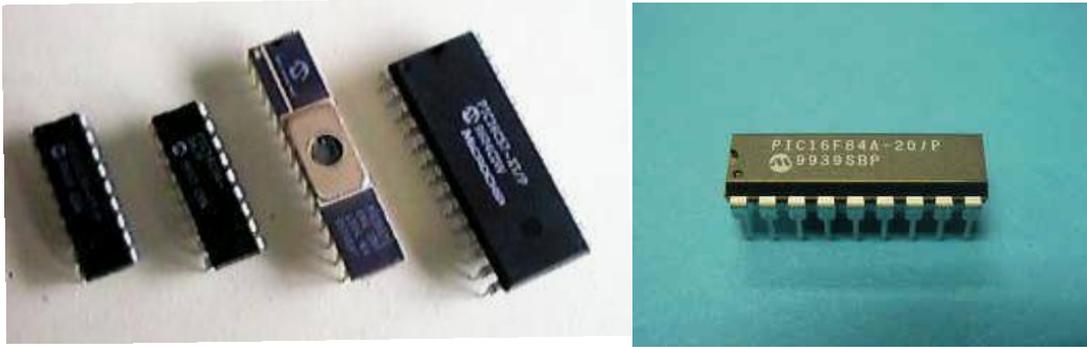


(a)

(b)

Gambar 4.3. Mikrokontroller Motorola: (a) M6800 dan (b) M68000

Sementara itu, Microchip juga memproduksi mikrokontroler yang dikenal dengan nama PIC (*Programmable Interface Controller*). Terdapat beberapa tipe PIC yang terdapat di pasaran, seperti PIC16F84 dan PIC 16F877. Pada umumnya, tipe-tipe yang ada berhubungan dengan kapasitas memori dan metoda penyimpanannya. PIC ini termasuk mikrokontroler yang banyak digunakan oleh para pecinta elektronika.



Gambar 4.4. Mikrokontroler PIC

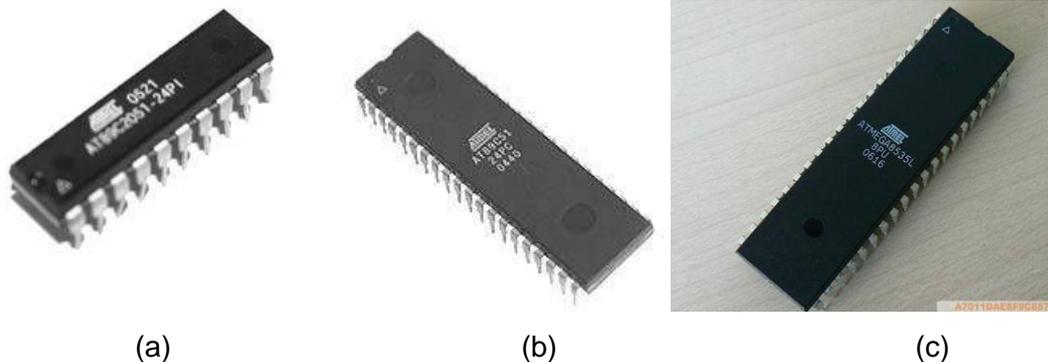
Terakhir, mikrokontroler yang juga banyak dipakai secara praktis adalah produksi Atmel. Seperti juga PIC, Atmel juga memiliki banyak tipe dan jenisnya. Tipe-tipe yang ada berhubungan dengan kapasitas memori dan metoda penyimpanannya. Secara umum, minimal terdapat 2 (dua) keluarga besar mikrokontroler versi Atmel, yaitu keluarga MCS 51 dan keluarga AVR.

Keluarga MCS 51 mengikuti secara persis struktur dan set instruksi mikrokontroler Intel 8051. Terdapat banyak turunan dari keluarga MCS 51 ini, seperti: AT89S51, AT89S52, AT89S8252, AT89C2051, AT89C4051, AT89C51, AT89C52, atau sering disebut keluarga AT89xx.

Sementara itu, terdapat pula keluarga Atmel AVR. Atmel AVR adalah jenis mikrokontroler yang paling sering dipakai dalam bidang elektronika dan instrumentasi. Mikrokontroler AVR ini memiliki arsitektur RISC (Reduce Instruction Set Computing) delapan bit, di mana semua instruksi dikemas dalam kode 16-bit (16 bits word) dan sebagian besar instruksi dieksekusi dalam 1 (satu) siklus clock. Secara umum, AVR dapat dikelompokkan dalam 7 kelas, yaitu keluarga ATTiny, keluarga AT90Sxx, keluarga ATmega, keluarga ATXMega, keluarga ATUSBxx, keluarga ATPWMxx dan AT86RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, peripheral dan fungsinya. Salah satu mikrokontroler AVR yang sering dipakai adalah ATmega8 dan ATmega 8535.

Terdapat 2 metoda pengisian program pada mikrokontroler Atmel, yaitu menggunakan downloader melalui kabel serial atau menggunakan In System Programming (ISP) melalui kabel paralel. Metoda pertama memerlukan alat tambahan yaitu downloader, dimana chip harus dicabut dan dipasang ulang antara sistem dan downloader. Sedangkan dengan metoda kedua, chip tidak perlu dicabut dan dipasang ulang, karena pengisian dapat dilakukan secara langsung ke sistem.

Contoh beberapa IC Atmel dapat dilihat pada Gambar 4.5.



Gambar 4.5. Mikrokontroler Atmel: (a) AT89C2051, (b) AT89C51 dan (c) AVR8535

4.3. Mikrokontroler ATMega 328

Salah satu mikrokontroler yang sekarang banyak digunakan adalah mikrokontroler produksi Atmel generasi AVR. Ada dua versi kepanjangan dari AVR, yaitu **Advanced Versatile RISC** yang maknanya adalah mikroprosesor berteknologi tinggi RISC atau **Alfa and Vegard's RISC** yang artinya prosesor yang dirancang oleh dua mahasiswa Norwegia bernama Alf-Egil Bogen dan Vegard Wollan dengan teknologi RISC.

RISC (*Reduced Instruction Set Computers*) adalah sebuah tren teknologi terbaru mikroprosesor untuk mempercepat proses pelaksanaan instruksi sehingga berefek kepada perubahan desain dan ukurannya. Berbeda dengan teknologi sebelumnya yaitu CISC (*Complex Instruction Set Computers*) yang membutuhkan 12 siklus clock setiap instruksi, teknologi RISC hanya memerlukan 1 siklus clock setiap instruksi.

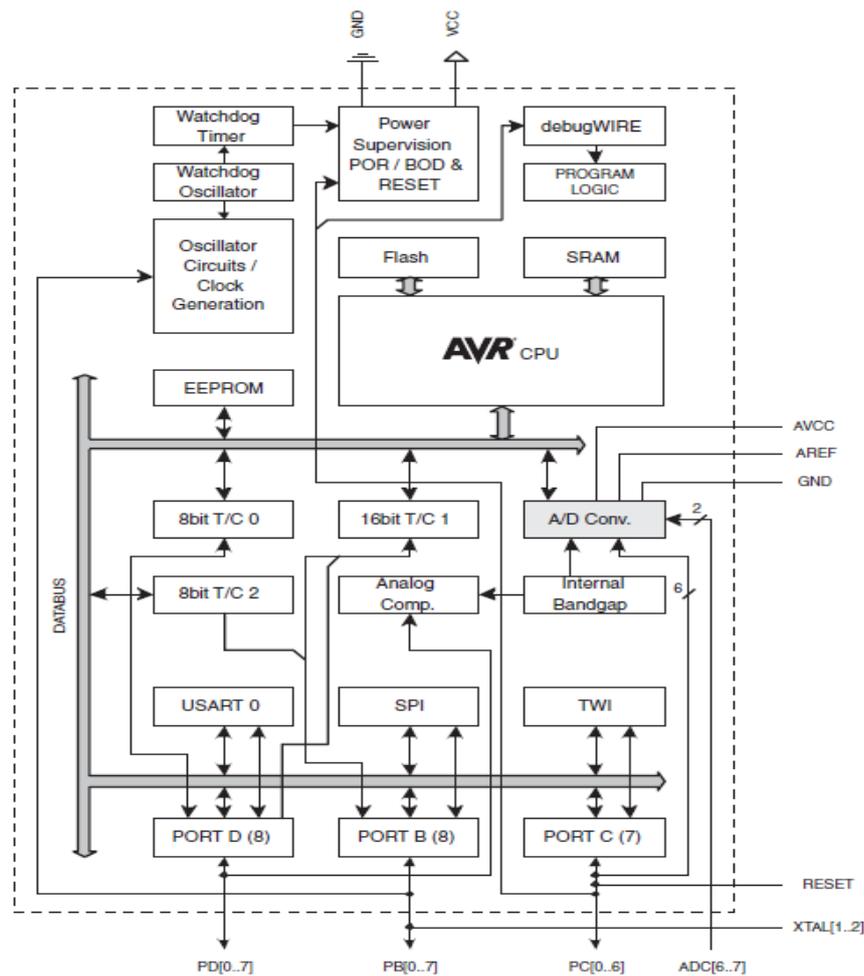
Secara umum AVR dapat dikelompokkan menjadi 4 golongan, yaitu: keluarga ATTiny, AT90Sxx, ATMega dan AT86RFxx. Yang membedakan masing-masing golongan adalah kapasitas memori, peripheral, dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan keseluruhan golongan tersebut hampir sama.

OK, sekarang kembali ke ATMega 328 sebagai komponen utama Arduino! ATMega 328 adalah mikrokontroler AVR produk Atmel yang harganya relatif murah dan mudah didapatkan di pasaran serta memiliki fasilitas yang lengkap. Fitur yang terdapat dalam ATMega 328 antara lain adalah:

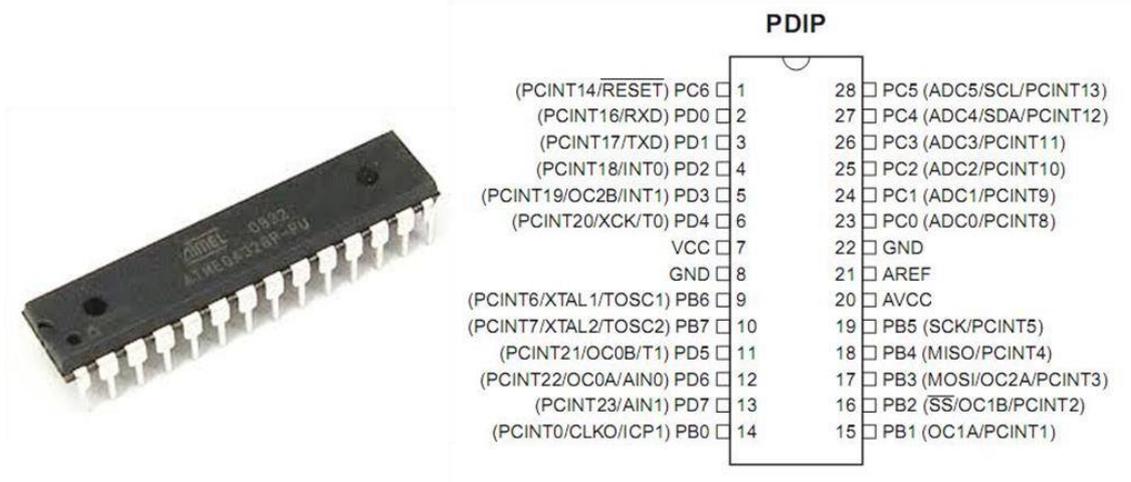
1. Sistem mikroprosesor 8 bit berbasis RISC dengan kecepatan maksimal 16 MHz.
2. Kapasitas memori *flash* 32 KB, SRAM 2 KB byte dan EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 1 KB.
3. Memiliki 19 pin Input/Output Digital dan 6 pin Input Analog

4. Analog to Digital Converter (ADC) internal dengan fidelitas 10 bit sebanyak 8 *channel*, 2 buah Timer/Counter 8 bit, sebuah Timer/Counter 16 bit dan 6 buah kanal PWM
5. Portal komunikasi serial (USART) dengan kecepatan maksimal 2.5Mbps, *Serial Interface* dan I2C.

Kalau mau tahu arsitektur mikrokontroler ATmega 328 lengkap, dapat dilihat pada Gambar 4.6. Sementara, bentuk fisik dan konfigurasi pin ditampilkan pada Gambar 4.7.



Gambar 4.6. Arsitektur Internal Mikrokontroler ATmega 328



Gambar 4.7. Konfigurasi Pin Mikrokontroler ATmega 328

Secara garis besar, sebagaimana terlihat pada Gambar 2.2, mikrokontroler ATmega dengan kemasan DIP (*Dual In-line Package*) memiliki 28 buah pin. Fungsi masing-masing pin tersebut adalah sebagai berikut:

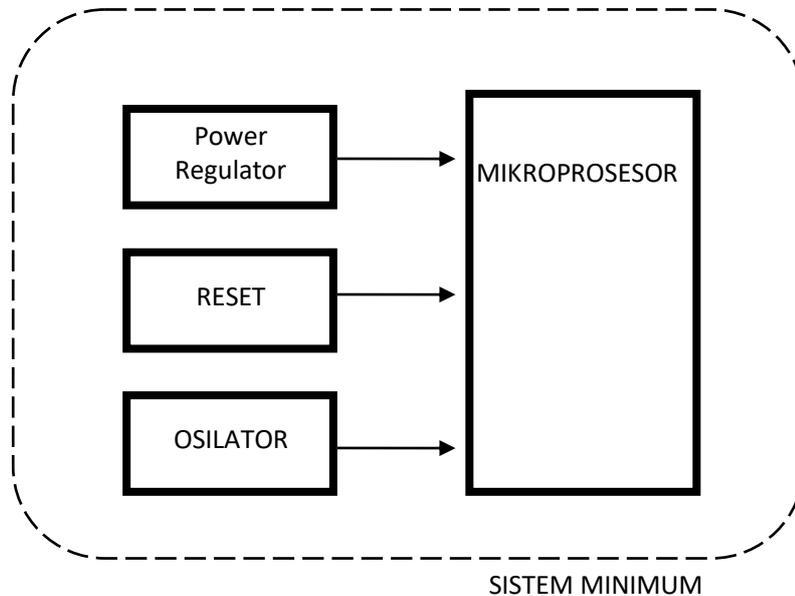
1. VCC merupakan pin yang berfungsi sebagai pin masukan catu daya.
2. GND merupakan pin ground.
3. Port C (PC0 ... PC6) merupakan I/O(Input Output) dua arah dan pin masukan ADC (*Analog to Digital Converter*).
4. Port B (PB0 ... PB7) merupakan pin I/O digital dua arah dan pin fungsi khusus, yaitu *interrupt*, XTAL1, XTAL2, serta MOSI, MISO dan SCK untuk komunikasi serial
5. Port D (PD0 ... PD7) merupakan pin I/O dua arah dan fungsi pin khusus, yaitu *interrupt* dan Tx/RX sebagai transceiver
6. RESET merupakan pin yang digunakan untuk me-reset mikrokontroler.
7. XTAL1 dan XTAL2 merupakan pin masukan *clock* eksternal.
8. AREF merupakan pin masukan tegangan referensi ADC.

Sekali lagi, tenang! Jangan terburu-buru bingung. Semua ini akan tampak sangat jelas, ketika mikrokontroler mulai diaplikasikan pada rangkaian aplikasi tertentu.

4.4. Sistem Minimum

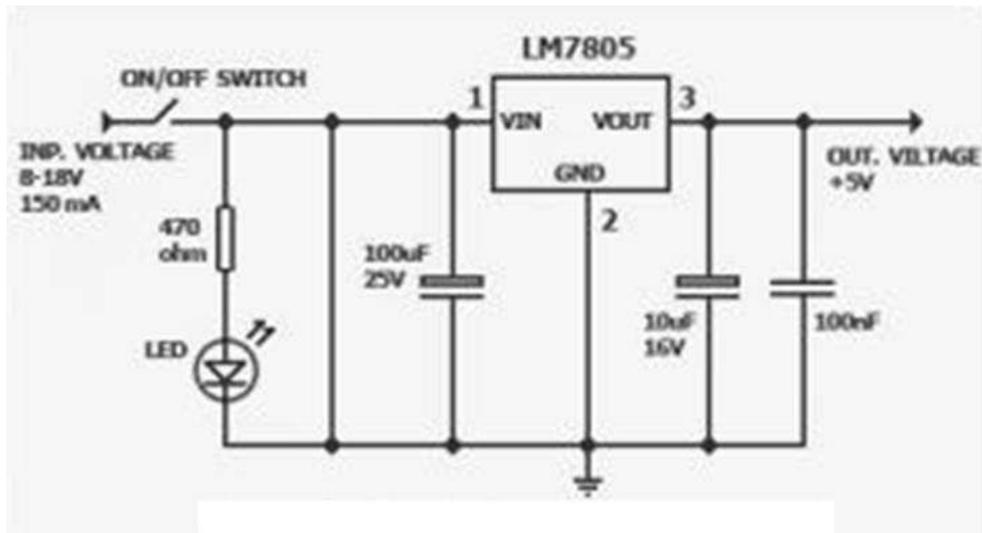
Sebuah sistem mikrokontroler tidak dapat dioperasikan dengan hanya sebuah mikrokontroler saja. Namun, untuk menjalankannya diperlukan serangkaian komponen yang terbentuk dalam sebuah *sistem minimum mikrokontroler*. Jika mikrokontroler sudah terangkai dalam sebuah sistem minimum, maka barulah sistem tersebut sudah siap untuk diaplikasikan dalam menjalankan fungsi tertentu.

Sebuah sistem minimum mikrokontroler terdiri dari tiga blok rangkaian, yaitu: rangkaian *power regulator*, rangkaian reset dan rangkaian osilator. Gambar 4.8 mengilustrasikan blok-blok sebuah sistem minimum.

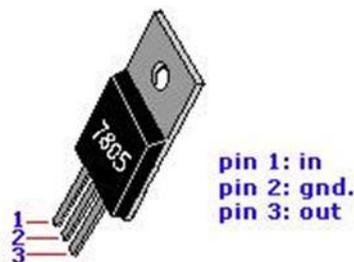


Gambar 4.8. Sistem Minimum

Blok pertama dalam sebuah sistem minimum adalah rangkaian *power regulator*. Rangkaian ini diperlukan agar tegangan yang masuk ke dalam rangkaian mikrokontroler dalam kondisi yang stabil, yaitu +5 volt. Pada umumnya, rangkaian power regulator terdiri dari beberapa buah dioda sebagai penyearah tegangan, beberapa keping *capacitor* sebagai penyaring, sebuah chip regulator LM7805 serta pasangan LED dan resistor sebagai indikator. Rangkaian lengkap power regulator dan bentuk fisik chip regulator LM7805 diperlihatkan pada Gambar 4.9 dan 4.10.

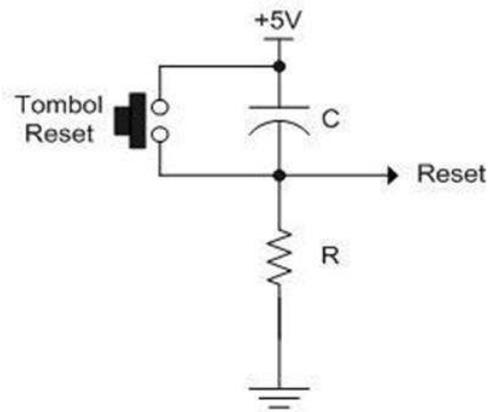


Gambar 4.9. Rangkaian Power Regulator



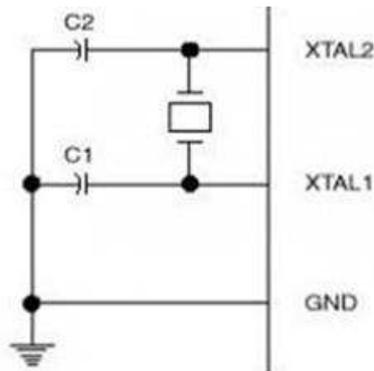
Gambar 4.10. IC LM 7805

Rangkaian lain yang dibutuhkan adalah rangkaian reset. Rangkaian ini berguna bagi mengembalikan program sistem mikrokontroler ke nilai awalnya (*Address Memory = 0*). Mikrokontroler bekerja dengan cara melaksanakan instruksi yang disimpan pada *Address Memory*. Pelaksanaan instruksi ini dimulai dari *Address Memory = 0*. Setelah sebuah instruksi dijalankan maka instruksi *Address Memory* berikutnya akan dipanggil dan dijalankan pula. Dengan demikian, jika saklar reset ini ditekan berarti mikrokontroler akan mengulangi program dari awal lagi. Biasanya, saklar reset ditekan apabila terdapat kesalahan atau gangguan pada saat menjalankan program. Sebuah rangkaian reset diperlihatkan pada Gambar 4.11.



Gambar 4.11. Rangkaian Reset

Blok terakhir dalam sebuah sistem minimum adalah rangkaian osilator. Rangkaian osilator ini berfungsi untuk menghasilkan suatu gelombang osilasi. Gelombang ini digunakan sebagai *clock* (detak) bagi mikrokontroler. Dengan detak ini, mikrokontroler berfungsi secara sinkron (serempak) dari satu instruksi ke instruksi berikutnya. Rangkaian osilator terdiri dari sebuah kristal (XTAL) dan 2 buah capacitor. Interkoneksi rangkaian reset dengan mikrokontroler ditampilkan pada Gambar 4.12.



Gambar 4.12. Rangkaian Osilator

Dari rangkaian-rangkaian diatas, maka dirakitlah sistem minimum mikrokontroler. Dengan sistem minimum ini, berarti sistem mikrokontroler telah siap untuk dioperasikan untuk melaksanakan fungsi tertentu. Contoh sebuah sistem minimum mikrokontroler, yang telah dilengkapi dengan berbagai konektor, diperlihatkan pada Gambar 4.13.

Daftar Pustaka

1. A. P. Mathur, *Introduction to Microprocessors*, 24th Reprinted., McGraw-Hill Publishing Company Ltd., USA, 2006.



MODUL PERKULIAHAN

Robotika

Pemrograman I/O Digital

Fakultas
Fakultas Teknik

Program Studi
Teknik Elektro

Tatap Muka

05

Kode MK
MK14043

Disusun Oleh
Julpri Andika, ST., M.Sc.

Abstract

Modul ini menjelaskan teknik pemrograman Input dan Output Digital. Beberapa komponen Input dan Output Digital dibahas, lengkap dengan rangkaian dan contoh programnya.

Kompetensi

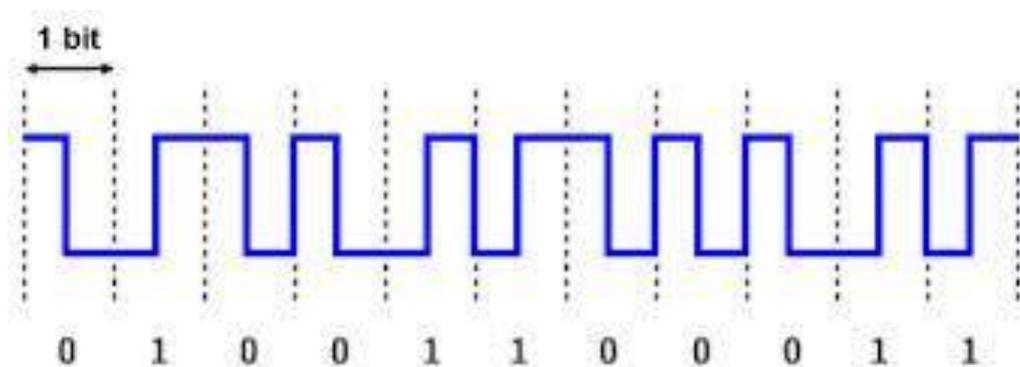
Mahasiswa mampu untuk memprogram dengan menggunakan komponen input dan output digital.

5.1. Signal Digital

Sistem yang banyak digunakan pada saat ini adalah sistem digital. Hampir setiap perangkat baru yang dikeluarkan oleh berbagai manufaktur produksi adalah perangkat yang berbasis digital, seperti: jam digital, kamera digital, TV digital, penyimpan music/fil digital, dan lain-lain. Dapat dikatakan bahwa saat ini adalah saat dimana perangkat digital menjadi primadona.

Sistem Mikroprosesor atau Mikrokontroler adalah contoh sebuah sistem digital. Pada system ini, mikrokontroler mengambil input data digital, memproses data secara digital dan mengeluarkan output digital.

Digital merupakan penggambaran dari suatu keadaan data atau signal yang terdiri dari angka 0 dan 1, atau *off* dan *on* (bilangan [biner](#)). Dapat disebut juga dengan istilah [Bit](#) (*Binary Digit*). Contoh sebuah signal digital diperlihatkan pada Gambar 5.1.



Gambar 5.1. Signal Digital

5.2. Komponen Digital

Berdasarkan definisi di atas, berarti komponen digital adalah perangkat yang memiliki dua keadaan dalam proses kerjanya, yaitu dalam keadaan '0' atau keadaan '1'. Contoh komponen digital yang praktis dan banyak digunakan adalah LED dan switch.

LED atau *Light Emitting Diode* adalah perangkat yang bahan dasarnya adalah semikonduktor merupakan komponen output digital. LED ini beroperasi pada dua keadaan saja, yaitu kondisi MENYALA atau ON dan kondisi PADAM atau OFF. LED akan menyala jika terdapat beda potensial positif antara elektroda-elektrodanya, yaitu kaki Anoda dan Katoda. Pemrograman mengatur signal keluaran mikrokontroler untuk mendapatkan signal sehingga LED berada pada kondisi yang diinginkan. Gambar 5.2 menampilkan contoh LED yang berwarna merah.



Gambar 5.2 Komponen Output Digital: LED

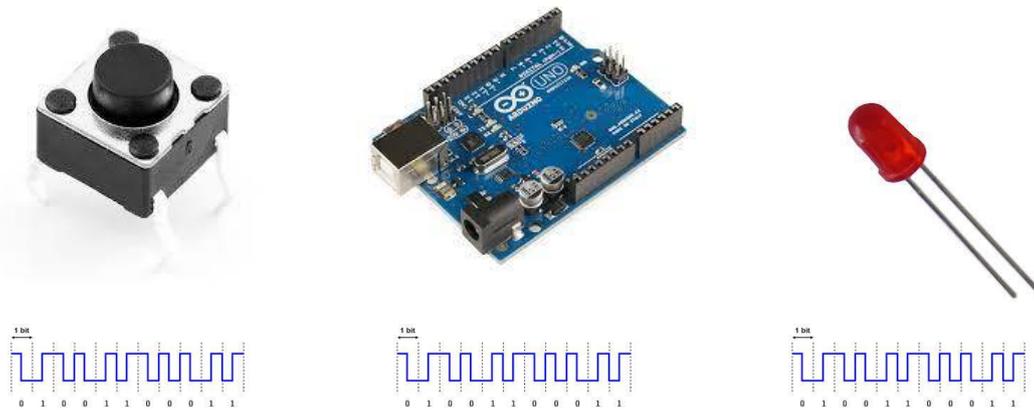
Contoh komponen input digital yang praktis adalah switch. Komponen dirancang sedemikian rupa sehingga memiliki kemampuan untuk menghubungkan atau memutuskan arus listrik. Jadi, komponen switch juga memiliki dua kondisi. Jika komponen input ini TERHUBUNG atau CLOSE maka arus listrik mengalir dan jika TERPUTUS atau OPEN maka arus listrik terputus. Mikrokontroler akan dirprogram untuk mempunyai kemampuan membaca kondisi switch, apakah dalam kondisi TERHUBUNG atau dalam kondisi TERPUTUS. Bentuk dan dimensi switch diperlihatkan pada Gambar 5.3.



Gambar 5.3. Bentuk dan dimensi switch

5.3. Sistem Input/Output Digital

Sebuah sistem digital adalah system yang seluruhnya diproses dalam bentuk signal digital. Jadi, system tersebut terdiri dari komponen input digital, pemroses data digital dan komponen output digital. Gambar 5.4 memperlihatkan sebuah system digital, dimana inputnya witch yang merupakan komponen digital, prosesesornya adalah mikrokontroler Arduino dan ouputnya adalah LED yang juga merupakan komponen digital.



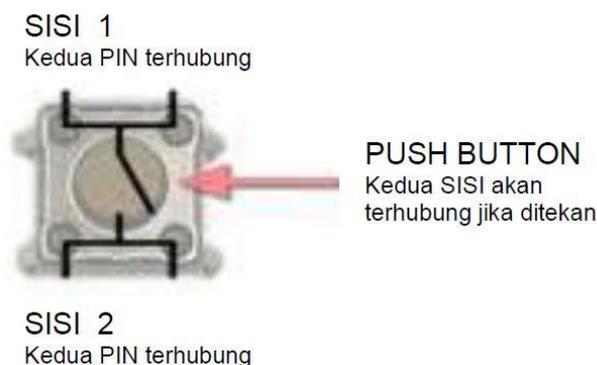
Gambar 5.4. Sistem Digital

Pada system digital ini, komponen input digital memberikan masukan kepada pemroses digital. Komponen input digital berupa switch memberikan signal digital yang terdiri dari dua kondisi saja, yaitu kondisi switch terhubung atau kondisi switch tidak terhubung. Kemudian, pemroses data berupa mikrokontroler Arduino akan memproses signal input tersebut dengan algoritma tertentu. Algoritma tersebut akan menghasilkan signal output digital. Signal output itu akan mengaktifkan output digital berupa LED. Aktifasi LED dapat berupa kondisi HIDUP atau dalam kondisi PADAM.

5.4. Pemrograman Input/Output Digital

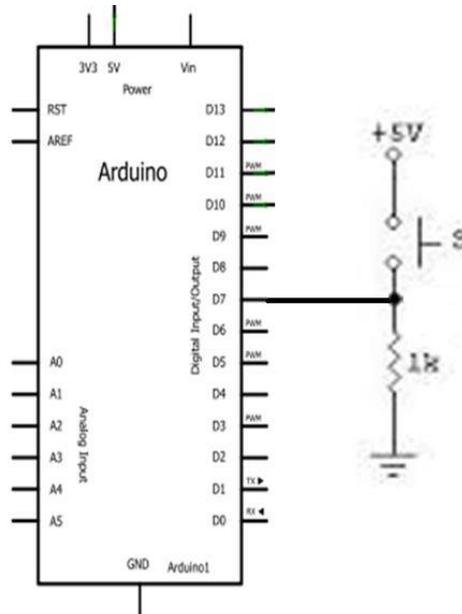
a. Rangkaian Input

Rangkaian input berupa push button yang dihubungkan dengan pin 7 pada Arduino. Jika tombol ditekan, maka sisi 1 dan sisi 2 dari push button akan terhubung. Sebaliknya, jika tombol tidak ditekan, maka kedua sisinya akan terpisah. Proses kerja push button diperlihatkan pada Gambar 5.5.



Gambar 5.5 Interkoneksi push button dengan Arduino

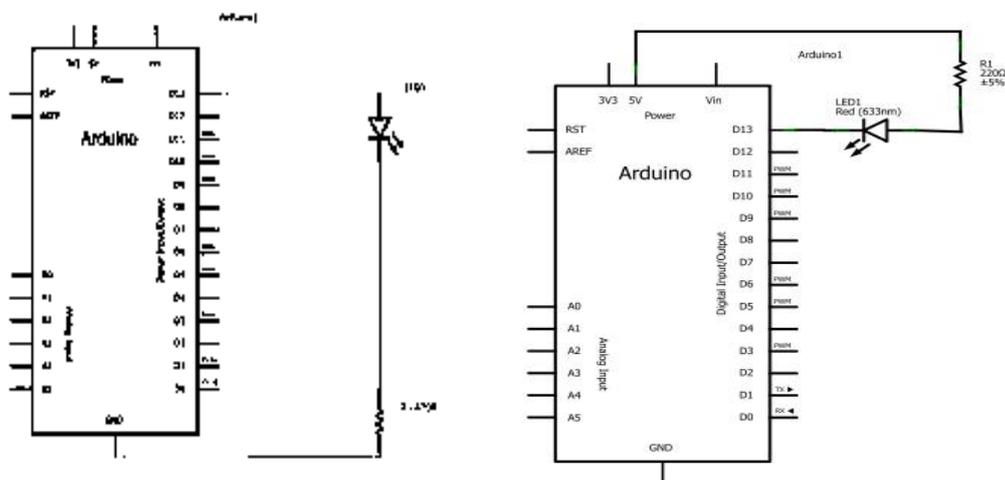
Sedangkan pada Gambar 5.6 memperlihatkan interkoneksi push button dengan Arduino, Berdasarkan prinsip rangkaian, maka jika tombol S ditekan, rangkaian push button akan terhubung. Akibatnya, Logika pada pin 7 adalah HIGH. Namun, jika tombol S tidak ditekan, maka rangkaian switch akan terputus. Akibatnya, Logika pada pin 7 adalah LOW.



Gambar 5.6 Interkoneksi push button dengan Arduino

b. Rangkaian Output

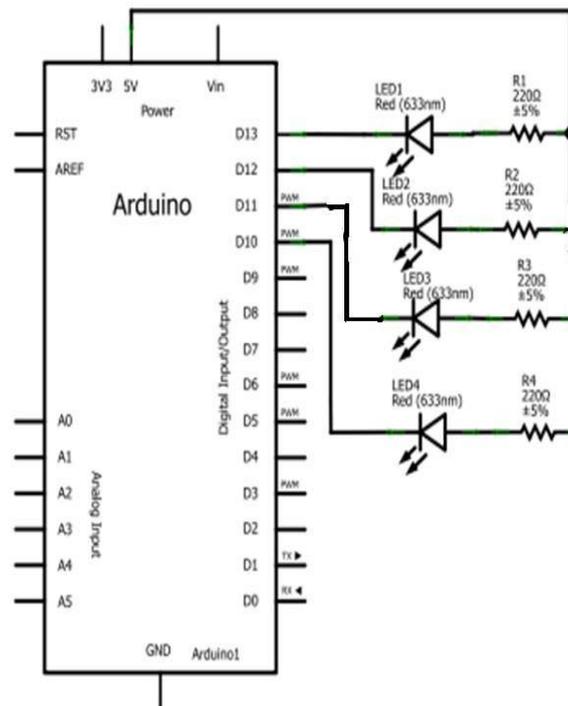
LED kembali digunakan sebagai rangkaian output karena mudah cara kerjanya dan mudah pula diamati perubahannya. Terdapat dua mekanisme peinterkonesian LED dengan mikrokontroler, yaitu rangkaian active high dan rangkaian active low. Gambar 5.7 Interkoneksi LED dengan Arduino.



Gambar 5.7 Jenis Interkoneksi LED dengan Arduino: (a) Active High dan (b) Active Low

Pada rangkaian active high, LED akan menyala ketika output berharga HIGH dan sebaliknya LED padam ketika output berharga LOW. Sedangkan rangkaian active low adalah lawan dari rangkaian active high. LED akan menyala ketika output berharga LOW dan sebaliknya LED padam ketika output berharga HIGH.

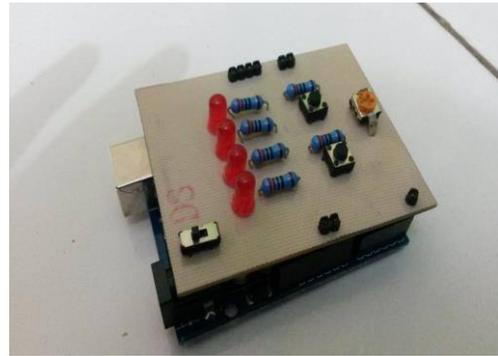
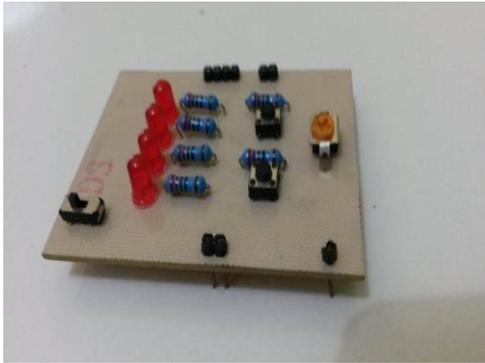
Rangkaian output menggunakan susunan 4 buah LED secara parallel, dengan secara keseluruhan digunakan rangkaian active low. Rangkaian diperlihatkan pada Gambar 5.8.



Gambar 5.8. Rangkaian susunan 4 LED

c. Modul Pelatihan I/O Digital

Rangkaian input dan output digital digabungkan dalam satu kesatuan yang disebut modul pelatihan I/O digital. Pada modul tersebut terdapat dua buah push button, empat buah LED dan sebuah resistor variable. Modul pelatihan dirancang sedemikian rupa sehingga dapat dihubungkan ke Arduino dalam bentuk plug and play. Pada hubungan ini tidak digunakan solder untuk menghubungkan antar komponen. Gambar 5.9 memperlihatkan modul pelatihan dan penggabungan modul tersebut dengan Arduino.



Gambar 5.9 Modul Pelatihan dan Penggabungan dengan Arduino.

d. Bahasa Pemrograman I/O Digital

Dirancang sebuah program yang melibatkan input dan output digital. Apabila push button tidak ditekan, LED menyala dengan pola penyalan bergantian masing-masing 2 buah LED. Sedangkan bila push button ditekan, LED menyala dengan pola penyalan yang lain, yaitu bergantian masing-masing 4 buah LED

Program #2 Multi LED – Dasar

Program blink 4 buah LED, bergantian 4 buah LED menyala dan 4 buah LED padam dengan delay waktu selama 1 detik.

```
/*  
Program 4 LED Dasar Kelap-kelip per 1 detik  
Program ini dibuat oleh Andi Adriansyah  
*/  
void setup()  
{  
  pinMode(10, OUTPUT); Inisialisasi pin-pin  
  pinMode(11, OUTPUT); 10, 11,12 dan 13  
  pinMode(12, OUTPUT); sebagai Output  
  pinMode(13, OUTPUT);  
}  
void loop()  
{  
  digitalWrite(10, HIGH); // LED padam  
  digitalWrite(11, HIGH); // LED padam
```

```

digitalWrite(12, HIGH); // LED padam
digitalWrite(13, HIGH); // LED padam
delay(1000);           // tunggu 1 detik

digitalWrite(10, LOW); // LED nyala
digitalWrite(11, LOW); // LED nyala
digitalWrite(12, LOW); // LED nyala
digitalWrite(13, LOW); // LED nyala
delay(1000);           // tunggu 1 detik
}

```

Program #3 Multi LED – 1

Program blink 4 buah LED, terdapat 2 pola blink, yaitu pola 1 bergantian 4 buah LED menyala dan 4 buah LED padam, dan pola 2 bergantian 2 buah LED menyala dan 2 buah LED padam dan sebaliknya, dengan delay waktu sama yaitu selama 1 detik.

```

/*
Program 4 LED Dasar Kelap-kelip per 1 detik
Program ini dibuat oleh Andi Adriansyah
*/
void setup()
{
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
}
void pola1()
{
  digitalWrite(10, HIGH); // LED padam
  digitalWrite(11, HIGH); // LED padam
  digitalWrite(12, HIGH); // LED padam
  digitalWrite(13, HIGH); // LED padam
  delay(1000);           // tunggu 1 detik
  digitalWrite(10, LOW); // LED menyala
  digitalWrite(11, LOW); // LED menyala

```

```

digitalWrite(12, LOW); // LED menyala
digitalWrite(13, LOW); // LED menyala
delay(1000);           // tunggu 1 detik
}

```

```

void pola2()

```

```

{
digitalWrite(10, HIGH); // LED padam
digitalWrite(11, HIGH); // LED padam
digitalWrite(12, LOW);  // LED menyala
digitalWrite(13, LOW);  // LED menyala
delay(1000);            // tunggu 1 detik
digitalWrite(10, LOW);  // LED menyala
digitalWrite(11, LOW);  // LED menyala
digitalWrite(12, HIGH); // LED padam
digitalWrite(13, HIGH); // LED padam
delay(1000);            // tunggu 1 detik
}

```

```

void loop()

```

```

{
pola1();
pola1();
pola1();
pola2();
pola2();
pola2();
}

```

Program #4 Multi LED – 2

Sama dengan Program #3, tetapi menggunakan perintah pengulangan FOR, dengan jumlah pengulangan yang berbeda

```

void setup()

```

```

{
pinMode(10, OUTPUT);
pinMode(11, OUTPUT);
}

```

```

pinMode(12, OUTPUT);
pinMode(13, OUTPUT);
}
void pola1()
{
for(int i;i<7;i++)
{
digitalWrite(10, HIGH); // LED padam
digitalWrite(11, HIGH); // LED padam
digitalWrite(12, HIGH); // LED padam
digitalWrite(13, HIGH); // LED padam
delay(100);           // tunggu 0.1 detik
digitalWrite(10, LOW); // LED menyala
digitalWrite(11, LOW); // LED menyala
digitalWrite(12, LOW); // LED menyala
digitalWrite(13, LOW); // LED menyala
delay(100);           // tunggu 0.1 detik
}
}

```

```

void pola2()
{
for(int i;i<5;i++)
{
digitalWrite(10, HIGH); // LED padam
digitalWrite(11, HIGH); // LED padam
digitalWrite(12, LOW); // LED menyala
digitalWrite(13, LOW); // LED menyala
delay(1000);           // tunggu 1 detik
digitalWrite(10, LOW); // LED menyala
digitalWrite(11, LOW); // LED menyala
digitalWrite(12, HIGH); // LED padam
digitalWrite(13, HIGH); // LED padam
delay(1000);           // tunggu 1 detik
}
}

```

```

void loop()

```

```

{
  pola1();
  pola2();
}

```

Program #5 Multi LED – Switch

Nyalanya LED tergantung kepada kondisi push button S. Jika S ditekan, 2 buah LED akan menyala dan 2 LED lainnya akan padam. Jika S tidak ditekan, kondisi LED akan berkebalikan.

```

void setup()
{
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
  pinMode(7, INPUT);
}

void loop()
{
  int tombol = digitalRead(7);
  if (tombol==HIGH)
  {
    digitalWrite(10, HIGH); // LED padam
    digitalWrite(11, HIGH); // LED padam
    digitalWrite(12, LOW); // LED menyala
    digitalWrite(13, LOW); // LED menyala
    delay(1000);           // tunggu 1 detik
  }
  else

  {
    digitalWrite(10, LOW); // LED padam
    digitalWrite(11, LOW); // LED padam
    digitalWrite(12, HIGH); // LED menyala
    digitalWrite(13, HIGH); // LED menyala
  }
}

```

```

    delay(1000);           // tunggu 1 detik
  }
}

```

Program #6 Multi LED - Switch

Pola menyala LED tergantung kepada kondisi push button S. Jika S ditekan maka LED menyala dengan pola1. Jika S tidak ditekan, maka LED menyala dengan pola2.

```

void setup()
{
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
}
void pola1()
{
  for(int i;i<5;i++)
  {
    digitalWrite(10, HIGH); // LED padam
    digitalWrite(11, HIGH); // LED padam
    digitalWrite(12, HIGH); // LED padam
    digitalWrite(13, HIGH); // LED padam
    delay(100);           // tunggu 0.1 detik
    digitalWrite(10, LOW); // LED menyala
    digitalWrite(11, LOW); // LED menyala
    digitalWrite(12, LOW); // LED menyala
    digitalWrite(13, LOW); // LED menyala
    delay(100);           // tunggu 0.1 detik
  }
}
void pola2()
{
  for(int i;i<3;i++)
  {
    digitalWrite(10, HIGH); // LED padam
    digitalWrite(11, HIGH); // LED padam
    digitalWrite(12, LOW); // LED menyala
    digitalWrite(13, LOW); // LED menyala
    delay(1000);           // tunggu 1 detik
  }
}

```

```

digitalWrite(10, LOW); // LED menyala
digitalWrite(11, LOW); // LED menyala
digitalWrite(12, HIGH); // LED padam
digitalWrite(13, HIGH); // LED padam
delay(1000);          // tunggu 1 detik
}
}
void loop()
{
  int tombol = digitalRead(7);
  if (tombol==HIGH) pola1();
  else pola2();
}

```

Berdasarkan program di atas, terdapat beberapa prinsip pemrograman, yaitu:

- pinMode(13, OUTPUT); menginisialisasi pin
- digitalWrite(13, HIGH); mengirimkan signal output digital
- digitalRead(7); membaca signal input digital
- delay(1000); melakukan penundaan
- if (kondisi) else; perintah bercabangan kondisi bersyarat
- for (kondisi); perintah pengulangan
- int variabel; integer, type data bilangan
- void nama_fungsi(); sub program atau fungsi

Daftar Pustaka

1. M. Margoilis, *Arduino Cookbook*, O'Reilly, USA 2011
2. M. McRoberts, *Beginning Arduino*, Apress, USA, 2010
3. M. Schmidt, *Arduino A Quick-Start Guide*, Pragmatic Programmers, USA, 2011.



MODUL PERKULIAHAN

Robotika

Pemrograman I/O Analog

Fakultas
Fakultas Teknik

Program Studi
Teknik Elektro

Tatap Muka

06

Kode MK
MK14043

Disusun Oleh
Julpri Andika, ST., M.Sc.

Abstract

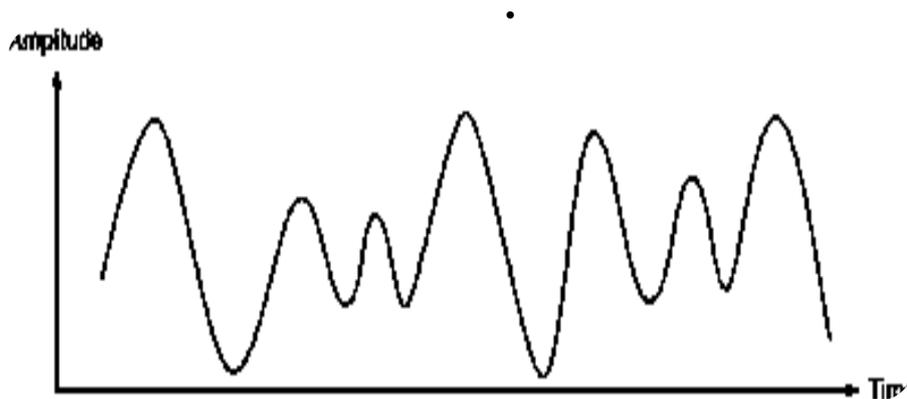
Modul ini menjelaskan teknik pemrograman Input dan Output Analog. Beberapa komponen Input dan Output Analog dibahas, lengkap dengan rangkaian dan contoh programnya.

Kompetensi

Mahasiswa mampu untuk memprogram dengan menggunakan komponen input dan output analog.

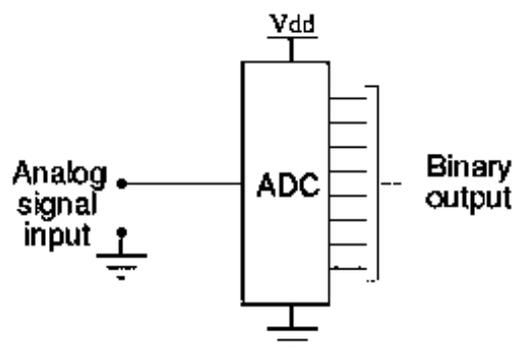
6.1. Signal Analog

Sinyal analog adalah yang dapat mengambil sejumlah nilai, tidak seperti sinyal digital yang hanya memiliki dua nilai: TINGGI dan RENDAH. Untuk mengukur nilai sinyal analog, Arduino memiliki built-in analog-ke-digital converter (ADC). ADC mengubah tegangan analog menjadi nilai digital. Fungsi yang Anda gunakan untuk mendapatkan nilai dari sinyal analog adalah `analogRead (pin)`. Fungsi ini mengubah nilai tegangan pada pin input analog dan mengembalikan nilai digital 0-1023, relatif terhadap nilai referensi.. Contoh sebuah signal analog diperlihatkan pada Gambar 6.1.



Gambar 6.1. Signal Analog

Alat yang mengubah dari Data Analog ke Data Digital dikenal dengan **Analog to Digital Converter (ADC)**. Hasilnya adalah beberapa buah bit data digital.



Gambar 6.2 Analog to Digital Converter

Arduino memiliki 6 buah input ADC (A0-A5), membaca tegangan input antara 0 - 5 volt, resolusi digital 10-bit (1024 keadaan), mengubahnya menjadi bilangan 0-1023 (desimal), tegangan terkecil yang dapat diukur: $5/1024 = 4.8 \text{ mV}$

5.2. Komponen Analog

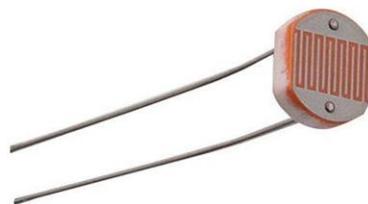
Berdasarkan definisi di atas, berarti komponen analog adalah perangkat yang mengambil sejumlah nilai untuk pengukurannya. Contoh komponen analog yang praktis dan banyak digunakan adalah potensiometer dan photo resistor.

Potentiometer adalah resistor tiga terminal yang nilai tahanannya dapat diubah dengan cara memutar tuasnya. Penggunaan tuas dimaksudkan bahwa rangkaian yang menggunakan potensiometer ini sering digunakan sebagai pengaturan yang dilakukan oleh pemakai atau perubahan fisik. Gambar 6.2 menampilkan contoh potensiometer.



Gambar 6.2 Komponen Potensiometer

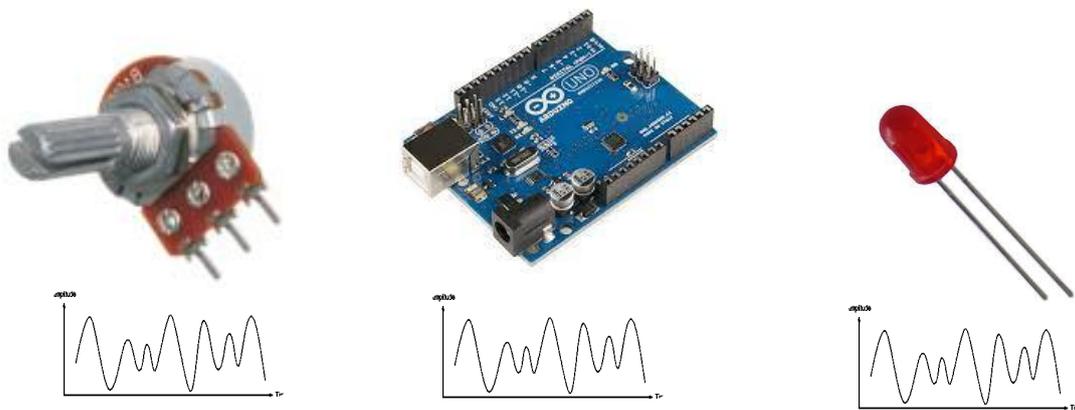
Sebuah photoresistor (atau resistor yang tergantung terhadap cahaya, LDR, atau fotosel) adalah variabel resistor pengendali cahaya. Hambatan dari photoresistor berkurang seiring dengan meningkatnya intensitas cahaya; dengan kata lain, itu menunjukkan fotokonduktivitas. Sebuah photoresistor dapat diterapkan dalam rangkaian detektor yang sensitif terhadap cahaya, dan sirkuit yang dapat aktif dan non-aktif terhadap cahaya. Komponen photoresistor diperlihatkan pada Gambar 6.3.



Gambar 6.3. Komponen Photoresistor

6.3. Sistem Input/Output Analog

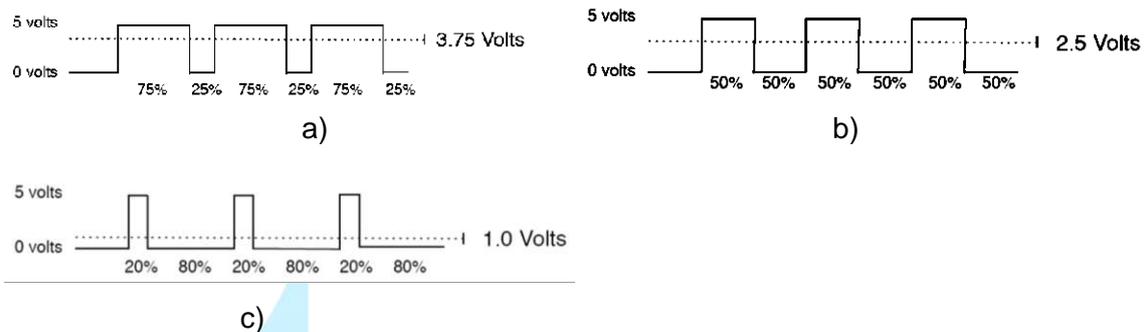
Sebuah sistem analog adalah system yang proses awalnya dalam bentuk signal analog yang kemudian dijadikan signal digital menggunakan ADC. Jadi, system tersebut terdiri dari komponen input analog, pemroses data digital dan komponen output analog. Gambar 6.4 memperlihatkan sebuah system analog, dimana inputnya potensiometer yang merupakan komponen analog, prosesesornya adalah mikrokontroler Arduino dan ouputnya adalah motor DC (kecepatan).



Gambar 6.4. Sistem Analog

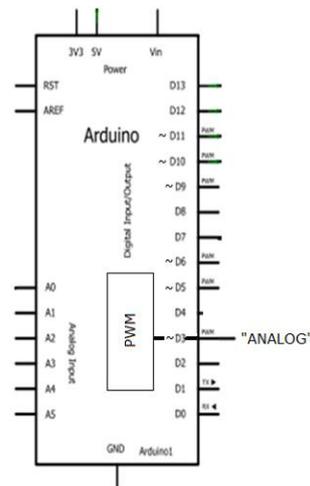
Mikrokontroler hanya mengenal sinyal digital dan akan menghasilkan sinyal digital, maka perlu trik untuk menghasilkan sinyal output analog dengan memanfaatkan proses Pulse Width Modulation (PWM), merupakan salah satu jenis modulasi yang dilakukan dengan cara merubah lebar pulsa dari suatu pulsa data. Total pulsa dalam PWM adalah tetap, namun terjadi perubahan perbandingan pulsa positif (T_{on}) terhadap total pulsa (T). Tegangan akhir dari proses modulasi adalah:

$$V_{out} = (T_{on}/T) * V_{max}$$



Gambar 6.5 a) $T_{on} = 75\%$, maka $V = 3.75 V$, b) $T_{on} = 50\%$, maka $V = 2.5 V$, c) $T_{on} = 20\%$, maka $V = 1.0 V$

Arduino memiliki komponen PWM (bertanda: ~) yang menghasilkan sinyal ANALOG 8 bit. Keluaran PWM ini, seperti sinyal ANALOG, dengan nilai tegangan bervariasi dari 0 Volt – 5 volt, setara dengan nilai 0 – 255. Tersedia 6 keluaran PWM pada Arduino: pin 3, 5, 6, 9, 10 dan 11.



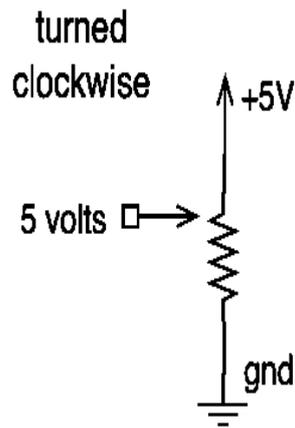
Gambar 6.6 Pin PWM pada Arduino

Pada system analog ini, komponen input analog memberikan masukan kepada pemroses analog. Komponen input analog berupa potensiometer memberikan signal analog yang mengatur besarnya tegangan. Kemudian, pemroses data berupa mikrokontroler Arduino akan memproses signal input (proses analog ke digital melalui ADC) tersebut dengan algoritma tertentu. Algoritma tersebut akan menghasilkan signal output analog (proses digital ke analog melalui PWM). Signal output itu akan mengaktifkan output berupa LED dimmer. Aktifasi LED dimmer dapat berupa kondisi menerang atau meredup.

6.4. Pemrograman Input/Output Digital

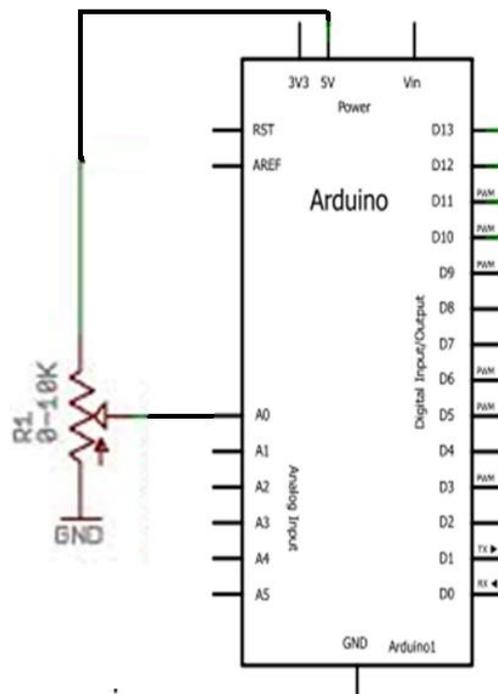
a. Rangkaian Input

Rangkaian input berupa potensiometer yang dihubungkan dengan pin A0 pada Arduino. Jika potensiometer diputar berlawanan arah jarum jam hingga hujung, maka tegangan keluarannya adalah 0 volt, Jika potensiometer diputar searah jarum jam mendekati tengah, maka tegangan keluarannya adalah 2.3 volt, Jika potensiometer diputar searah jarum jam hingga hujung, maka tegangan keluarannya adalah 5 volt. Proses kerja potensiometer diperlihatkan pada Gambar 6.7.



Gambar 6.7 Proses kerja potensiometer

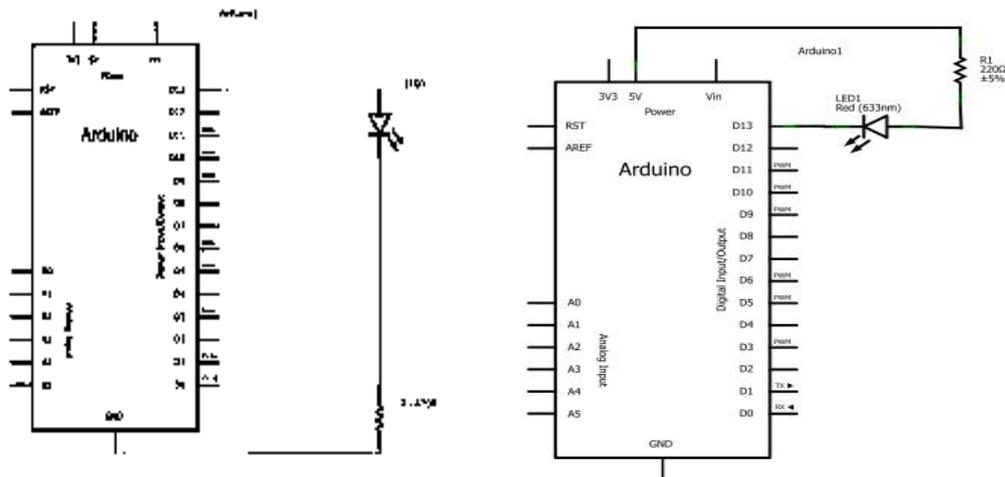
Sedangkan pada Gambar 6.8 memperlihatkan interkoneksi potensiometer dengan Arduino, Berdasarkan prinsip rangkaian, maka jika potensiometer diputar maka akan merubah harga tegangan yang akan diberikan Arduino ke lampu LED, sehingga membuat lampu LED akan menerang atau meredup.



Gambar 6.8 Interkoneksi potensiometer dengan Arduino

b. Rangkaian Output

LED kembali digunakan sebagai rangkaian output karena mudah cara kerjanya dan mudah pula diamati perubahannya. Terdapat dua mekanisme peinterkonesian LED dengan mikrokontroler, yaitu rangkaian active high dan rangkaian active low. Gambar 6.9 Interkoneksi LED dengan Arduino.



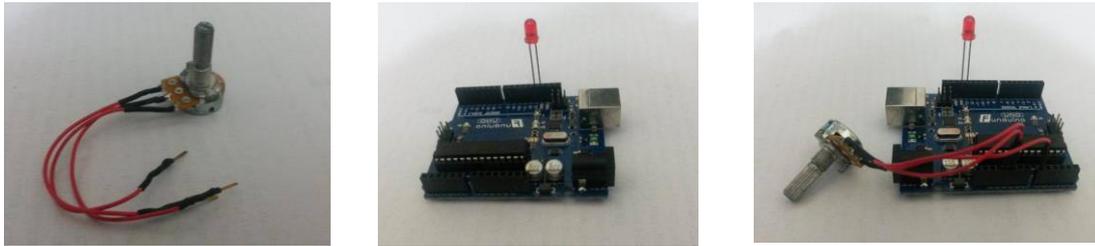
Gambar 6.9 Jenis Interkoneksi LED dengan Arduino: (a) Active High dan (b) Active Low

Pada rangkaian active high, LED akan menyala ketika output berharga HIGH dan sebaliknya LED padam ketika output berharga LOW. Sedangkan rangkaian active low adalah lawan dari rangkaian active high. LED akan menyala ketika output berharga LOW dan sebaliknya LED padam ketika output berharga HIGH.

Rangkaian output menggunakan susunan 1 buah LED dengan secara keseluruhan digunakan rangkaian active low.

c. Modul Pelatihan I/O Digital

Rangkaian input dan output analog digabungkan dalam satu kesatuan yang disebut modul pelatihan I/O analog. Pada modul tersebut terdapat satu buah potensiometer, dan sebuah lampu LED. Modul pelatihan dirancang sedemikian rupa sehingga dapat dihubungkan ke Arduino dalam bentuk plug and play. Pada hubungan ini tidak digunakan solder untuk menghubungkan antar komponen. Gambar 6.10 memperlihatkan modul pelatihan dan penggabungan modul tersebut dengan Arduino.



Gambar 6.10 Modul Pelatihan dan Penggabungan dengan Arduino.

d. Bahasa Pemrograman I/O Analog

Dalam contoh ini kita menggunakan resistor variabel (potensiometer), kita membaca nilainya menggunakan satu input analog dari papan Arduino dan mengubah tingkat kedipan LED. Nilai analog resistor dibaca sebagai tegangan karena ini adalah bagaimana input analog bekerja.

Program #7 Potensiometer

LED akan blink (menyala dan padam) dengan waktu delay berdasarkan tegangan potensiometer yang berubah-ubah sesuai dengan putarannya

```
int sensorValue = 0; //inisialisasi harga bacaan potentiometer awal
void setup()
{
  // Inisialisasi Pin Digital sebagai OUTPUT.
  // Pin 13 telah menghubungkan LED ke Arduino:
  pinMode(13, OUTPUT);
}
void loop()
{
  sensor Value = analogRead(A0); //membaca tegangan dari potensiometer
  digitalWrite(13, HIGH); // LED menyala
  delay(sensorValue); // waktu tunggu sesuai harga tegangan dari potensiometer
  digitalWrite(13, LOW); // LED padam
  delay(sensorValue); // waktu tunggu sesuai harga tegangan dari potensiometer
```

```
}
```

Program #8 Potensiometer

LED akan menyala bila tegangan potentiometer lebih besar dari harga batas tertentu dan padambila tegangan potentiometer kurang dari harga batas tertentu

```
int sensorValue = 0; //inisialisasi harga bacaan potentiometer awal
int batas = 500;     //inisialisasi harga bacaan potentiometer awal
void setup()
{
  // Inisialisasi Pin Digital sebagai OUTPUT.
  // Pin 13 telah menghubungkan LED ke Arduino:
  pinMode(13, OUTPUT);
}
void loop()
{
  sensor Value = analogRead(A0); //membaca tegangan dari potentiometer
  if (sensorValue>batas) //cek harga, apakah lebih besar dari harga batas?
  {
    digitalWrite(13, HIGH); // LED menyala
    delay(1000);           // tunggu 1 s
  }
  else
  {
    digitalWrite(13, LOW); // LED padam
    delay(1000);           // tunggu 1s
  }
}
```

Program #9 Potensiometer + LED

LED akan menyala dengan kecerahan berbeda, sesuai dengan harga tegangan dari potentiometer. Ini menggambarkan pengendalian nilai analog dengan masukan dari nilai digital.

```
int sensorValue = 0; //inisialisasi harga bacaan potentiometer awal
```

```

int outputValue = 0; //inisialisasi harga output awal
void setup()
{
  // Inisialisasi Pin Digital sebagai OUTPUT.
  // Pin 13 telah menghubungkan LED ke Arduino:
  pinMode(13, OUTPUT);
}
void loop()
{
  sensorValue = analogRead(0); //membaca tegangan dari potentiometer
  outputValue = sensorValue /4; //harga output disesuaikan dengan harga ADC
  analogWrite(0,outputValue); // LED menyala sesuai dengan harga output
  delay(1000); // waktu tunggu 1 s
}

```

Berdasarkan program di atas, terdapat beberapa prinsip pemrograman, yaitu:

- pinMode(13, OUTPUT); menginisialisasi pin
- digitalWrite(13, HIGH); mengirimkan signal output digital
- digitalRead(7); membaca signal input digital
- analogWrite(0,255); mengirimkan signal output analog (0 – 255)
- analogRead(0); membaca signal input analog (0 – 1023)
- delay(1000); melakukan penundaan
- if (kondisi) else; perintah bercabangan kondisi bersyarat
- for (kondisi); perintah pengulangan
- int variabel; integer, type data bilangan
- void nama_fungsi(); sub program atau fungsi

Daftar Pustaka

1. M. Margoilis, *Arduino Cookbook*, O'Reilly, USA 2011
2. M. McRoberts, *Beginning Arduino*, Apress, USA, 2010
3. M. Schmidt, *Arduino A Quick-Start Guide*, Pragmatic Programmers, USA, 2011.



MODUL PERKULIAHAN

Robotika

Serial Monitor

Fakultas

Fakultas Teknik

Program Studi

Teknik Elektro

Tatap Muka

07

Kode MK

14043

Disusun Oleh

Julpri Andika, ST., M.Sc.

Abstract

Motor listrik adalah alat untuk mengubah energi listrik menjadi energi mekanik, Motor listrik merupakan tulang punggung utama dalam aplikasi mekatronika.

Kompetensi

Mahasiswa memiliki wawasan tentang motor-motor listrik, motor DC, motor stepper, motor servo, motor DC dan robot bergerak

7.1 Serial Monitor

7.1 Pengertian

Secara lebih sederhananya. Komunikasi serial adalah sebuah komunikasi yang terjadi dengan mengirimkan data per-bit secara berurutan dan bergantian.

Dengan adanya komunikasi serial, maka Arduino tak hanya bisa mengolah data dari pin input dan outputnya saja, tetapi juga bisa dikomunikasikan secara dua arah dengan perangkat komputer untuk menampilkan hasil pengolahan datanya.

Bahkan [Arduino](#) pun bisa melakukan komunikasi serial dengan perangkat Android melalui koneksi *bluetooth*.

Di Arduino IDE sendiri sudah disediakan fitur untuk komunikasi dua arah melalui serial monitor yang bisa digunakan untuk berbagai keperluan. Dengan adanya fitur ini, maka komputer bisa mengirim dan menerima data dari Arduino.

Komunikasi serial Arduino dengan PC memungkinkanmu mengontrol Arduino melalui komputer atau memantau sesuatu yang terjadi padanya.

Misalnya kita dapat mengirim isyarat untuk menghidupkan lampu atau memantau suhu yang dideteksi sensor suhu melalui serial monitor di komputer. Umumnya semua [board Arduino](#) telah memiliki minimal 1 *port* serial yang biasa dikenal dengan jenis UART atau USART.

Proses komunikasi serial menggunakan dua pin, yaitu pin RX untuk menerima data dan pin TX untuk mengirimkan data.

Itulah alasan mengapa kadang ada yang menyebutkan istilah komunikasi RX TX Arduino. Biasanya pin RX di Arduino adalah Pin 0 dan pin TX adalah Pin 1.

Ada satu istilah yang perlu kamu ketahui dalam komunikasi serial, yaitu *baudrate*. Pengertian *baudrate* adalah kecepatan transfer data dalam bit per satuan detik atau *bit per second* (bps).

Beberapa pilihan *baudrate* pada Arduino yaitu 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, dan 115200. Tetapi kebanyakan yang sering digunakan adalah 9600.

7.2 Jenis-jenis Perintah Komunikasi Serial

1. If (Serial)

Perintah ini berfungsi untuk mengecek apakah *port* serial sudah siap atau belum.

2. Serial.available()

Berfungsi untuk menghasilkan jumlah *byte*, angka, atau karakter di *port* serial yang belum terbaca. Apabila *port* serial dalam keadaan kosong, maka fungsi ini akan menghasilkan nilai nol.

```
int kirimdata = 0;
void setup()
{ Serial.begin(9600);}

void loop()
{
if (Serial.available() > 0)
{
dataterkirim = Serial.read();
Serial.print("data terbaca: ");
Serial.println(kirimdata, DEC);
}
}
```

3. Serial.begin()

Fungsi serial begin pada Arduino adalah menentukan kecepatan pengiriman dan penerimaan data melalui *port* serial. Biasanya kecepatan yang sering digunakan adalah 9600 bps (*bit per second*).

Tetapi untuk Arduino Uno sendiri bisa mendukung hingga kecepatan 115.200 bps. Contoh format penulisannya yang benar adalah:

```
void setup() { Serial.begin(9600);}
```

4. Serial.end()

Perintah ini berfungsi untuk menghentikan komunikasi pada Arduino atau menutup *port* komunikasi serial serta menonaktifkan pin RX dan TX sebagai fungsi serial lalu mengubahnya jadi pin input output. Cara penulisannya yang benar yaitu:

```
Serial.end()
```

5. Serial.find()

Perintah ini digunakan untuk mencari *string* dalam *buffer* data. Contoh penulisan yang benar yaitu:

```
Serial.find()
```

6. **Serial.findUntil()**

Fungsi perintah ini adalah untuk mencari *buffer* data hingga data sesuai kriteria yang diberikan bisa ditemukan.

7. **Serial.print(data)**

`Serial.print` berfungsi untuk mengirimkan data ke *port* serial dan menampilkannya di serial monitor dalam satu baris saja. Apabila argumen format dimasukkan ke perintah, maka data yang dikirim akan menyesuaikan dengan format tersebut. Contoh format penulisan yang benar yaitu:

```
Serial.print(78) // mencetak "78"  
Serial.print(1.23456) // mencetak "1.23"  
Serial.print('N') // mencetak "N"  
Serial.print("Hello world.") // mencetak "Hello world."  
Serial.print(78, BIN) // mencetak "1001110"  
Serial.print(78, OCT) // mencetak "116"  
Serial.print(78, DEC) // mencetak "78"  
Serial.print(78, HEX) // mencetak "4E"
```

8. **Serial.println(data)**

Fungsi `serial.println` pada Arduino yaitu mengirimkan data untuk ditampilkan di layar serial monitor atau di LCD pada baris baru. Contoh penulisannya yaitu:

```
Serial.println(1.23456, 0) // mencetak "1"  
Serial.println(1.23456, 2) // mencetak "1.23"  
Serial.println(1.23456, 4) // mencetak "1.2346"
```

9. **Serial.read()**

Berfungsi membaca satu *byte* data yang diterima pada *port* serial. Biasanya setelah pemanggilan di `Serial.read()`, maka jumlah data di *port* serial akan berkurang satu. Contoh penulisan yang benarnya yaitu:

```
void loop()
{
  if (Serial.available() > 0)
  {
    dataterkirim = Serial.read();
    Serial.print("saya menerima data: ");
    Serial.println(dataterkirim, DEC);
  }
}
```

10. Serial.readBytes()

Perintah ini digunakan untuk membaca data *byte* yang diterima.

11. Serial.write()

Fungsi perintah ini adalah untuk membaca data biner dari *port* serial. Biasanya data yang terkirim dalam bentuk *byte* atau deretan data *byte*. Contoh penulisan perintah yang benar yaitu:

```
void setup(){
  Serial.begin(9600);
}

void loop(){
  Serial.write(45);
  int bytesSent = Serial.write("hello");
}
```

12. Serial.setTimeout()

Penggunaan perintah ini biasanya untuk mengatur batas maksimum waktu tunggu (*timeout*) proses transmisi data.

13. **Serial.flush()**

Perintah ini berfungsi untuk mengosongkan data pembacaan yang tersisa di *port* serial.

14. **Serial.parseFloat()**

Parsefloat Arduino fungsinya untuk mengambil data *float* (bilangan pecahan) pertama dari data di *buffer* serial.

15. **Serial.peek()**

Berfungsi untuk mengambil data berikutnya di *buffer* penerima.

16. **Serial.parseInt()**

Fungsinya untuk mengambil data integer (bilangan bulat) pertama dari data di *buffer* serial

17. **Serial.serialEvent()**

Perintah ini berfungsi layaknya interupsi serial yang biasanya dipanggil jika data datang atau diterima.

7.3 Contoh Format Konversi Data Pada Komunikasi Serial

1. **Serial.print(65);**

Hasil yang tertampil sesuai angka yang di dalam perintah

2. **Serial.print(65, DES);**

Hasil yang tampil berupa format desimal atau basis 10 yaitu 65

3. **Serial.print(65, HEX);**

Hasil yang tampil berupa format hexadesimal atau basis 16 yaitu 41

4. **Serial.print(65, OCT);**

Hasil yang tampil berupa format oktal atau basis 8 yaitu 101

5. Serial.print(65, BIN);

Hasil yang tampil berupa format biner atau basis 2 yaitu 1000001

```
int led1 = 13;

int led2 = 12;

int led3 = 11;

int out;

void setup()
{
  Serial.begin(9600);
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
}

void loop()
{
  if (Serial.available()>0)
  {
    int baca = Serial.read();

    if (baca == 'a')
    {
      out=1;
    }

    if (baca == 'b')
```

```

{
  out=2;
}

if (baca == 'c')
{
  out=3;
}

if(out==1)
jalan: //LED akan berjalan running
{
  digitalWrite(led1, HIGH);
  delay(500);
  digitalWrite(led1, LOW);
  delay(500);
  digitalWrite(led2, HIGH);
  delay(500);
  digitalWrite(led2, LOW);
  delay(500);
  digitalWrite(led3, HIGH);
  delay(500);
  digitalWrite(led3, LOW);
  delay(500);
  Serial.println("LED running");
  loop();
}

```

```

goto jalan;

}

if(out==2)

start:                                // LED akan berjalan flip flop
{
digitalwrite(led1, HIGH);
digitalwrite(led2, HIGH);
digitalwrite(led3, HIGH);
delay(500);
digitalwrite(led1, LOW);
digitalwrite(led2, LOW);
digitalwrite(led3, LOW);
delay(500);
Serial.println("LED flip flop");
loop();
goto start;
}

if(out==3)

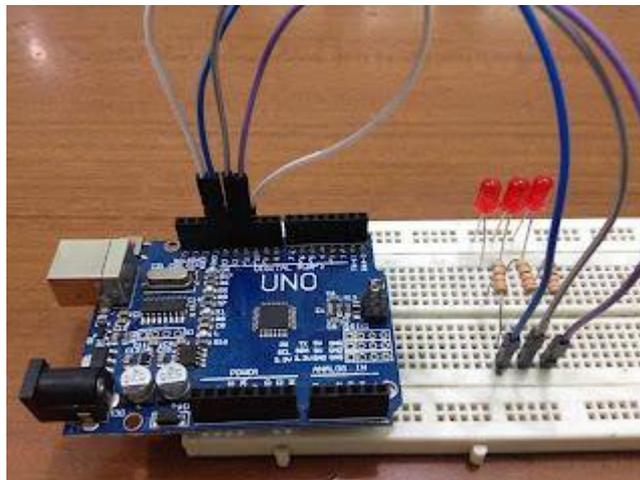
stop:                                  // LED akan mati
{
digitalwrite(led1,LOW);
digitalwrite(led2,LOW);
digitalwrite(led3,LOW);

```

```
Serial.println("LED mati");  
  
loop();  
  
goto stop;  
  
}  
  
}  
  
}
```

Setelah membuat program diatas dibawah ini adalah langkah - langkah mengaktifkan LED menggunakan serial port :

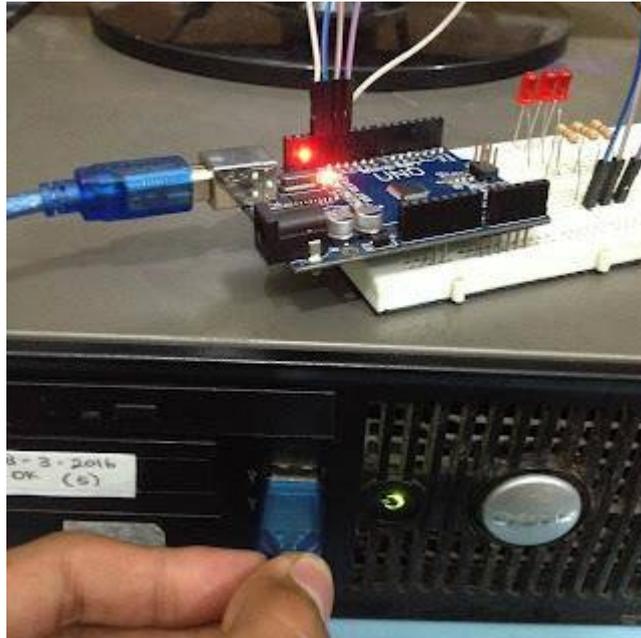
1. Rangkailah rangkaian menggunakan arduino. Gunakan output di nomor 13,12,11. LED aktif high.



Gambar 7.1

2. Sambungkan kabel downloader ke PC.

(sebelumnya pastikan kabel downloader sudah terinstal terlebih dahulu di PC anda, jika belum terinstal installah dulu. baca artikel saya sebelumnya tentang cara menginstal downloader arduino).



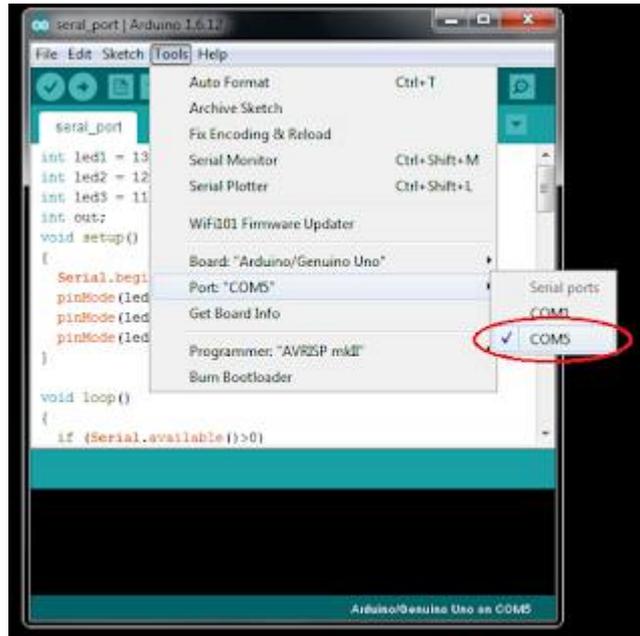
Gambar 7.2

3. Buka Aplikasi Arduino, kemudian buat program sesuai perintah di atas. atau tinggal copas program yang sudah saya buat.

```
serial_port | Arduino 1.6.12
File Edit Sketch Tools Help
serial_port
int led1 = 13;
int led2 = 12;
int led3 = 11;
int out;
void setup()
{
  Serial.begin(9600);
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
}
void loop()
{
  if (Serial.available() > 0)
```

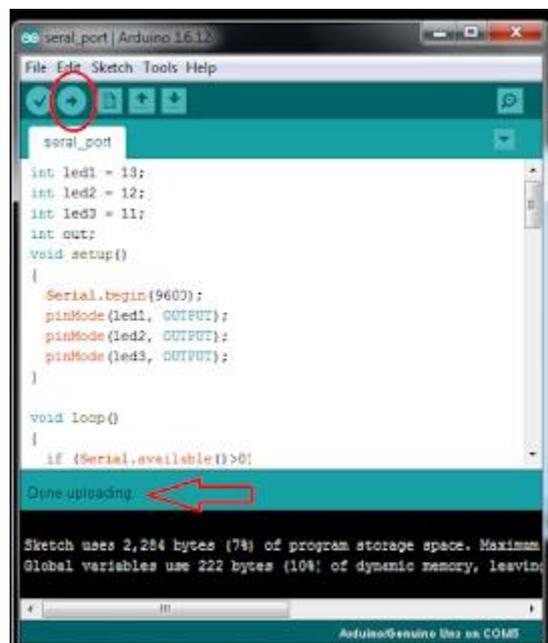
Gambar 7.3

4. Pastikan port berada pada serial port COM5.



Gambar 7.4

5. Upload program sampai tertera tulisan Done Uploading.



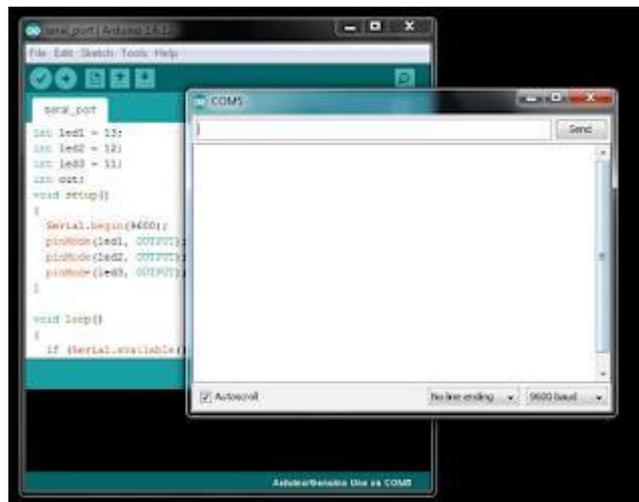
Gambar 7.5

6. Buka serial monitor di pojok kanan atas.



Gambar 7.6

7. Ketikkan "a" untuk menjalankan perintah running. "b" untuk menjalankan perintah flip flop. "c" untuk mematikan semua program.



Gambar 7.7

Daftar Pustaka

1. A. P. Mathur, *Introduction to Microprocessors*, 24th Reprinted., McGraw-Hill Publishing Company Ltd., USA, 2006.
2. ar Publisher, 1998.
3. J. G. Keramas, *Robot Technology*, Delmar Publisher, 1998.
4. K. S. Fu, R. C. Gonzales, C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*, Mc Graw-Hill Book Company, US, 1987.
5. M. Margoilis, *Arduino Cookbook*, O'Reilly, USA 2011
6. M. McRoberts, *Beginning Arduino*, Apress, USA, 2010
7. M. Schmidt, *Arduino A Quick-Start Guide*, Pragmatic Programmers, USA, 2011.



MODUL PERKULIAHAN

Robotika

Penggunaan Motor Pada Sistem Mekatronika

Fakultas

Fakultas Teknik

Program Studi

Teknik Elektro

Tatap Muka

08

Kode MK

14043

Disusun Oleh

Julpri Andika, ST., M.Sc.

Abstract

Motor listrik adalah alat untuk mengubah energi listrik menjadi energi mekanik, Motor listrik merupakan tulang punggung utama dalam aplikasi mekatronika.

Kompetensi

Mahasiswa memiliki wawasan tentang motor-motor listrik, motor DC, motor stepper, motor servo, motor DC dan robot bergerak

8.1 Motor-motor Listrik

Motor listrik adalah alat untuk mengubah energi listrik menjadi energi mekanik. Alat yang berfungsi sebaliknya, mengubah energi mekanik menjadi energi listrik disebut generator atau dinamo. Motor listrik merupakan tulang punggung utama dalam aplikasi mekatronika. Motor listrik dapat ditemukan pada peralatan rumah tangga seperti kipas angin, mesin cuci, pompa air dan penyedot debu.

Pada motor listrik tenaga listrik dirubah menjadi tenaga mekanik. Perubahan ini dilakukan dengan merubah tenaga listrik menjadi magnet yang disebut sebagai elektro magnet. Sebagaimana kita ketahui bahwa kutub-kutub dari magnet yang senama akan tolak-menolak dan kutub-kutub tidak senama akan tarik-menarik. Maka akan diperoleh gerakan jika kita menempatkan sebuah magnet pada sebuah poros yang dapat berputar dan magnet yang lain pada suatu kedudukan yang tetap.

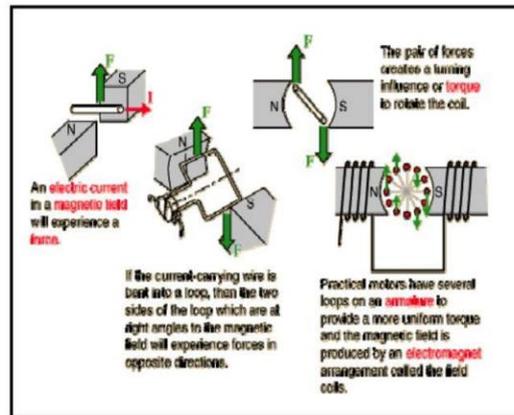


Gambar 8.1. Contoh beberapa Jenis Motor Listrik

Untuk lebih jelasnya, mekanisme kerja untuk seluruh jenis motor secara umum sama, sebagaimana diperlihatkan pada Gambar 8.2, adalah sebagai berikut:

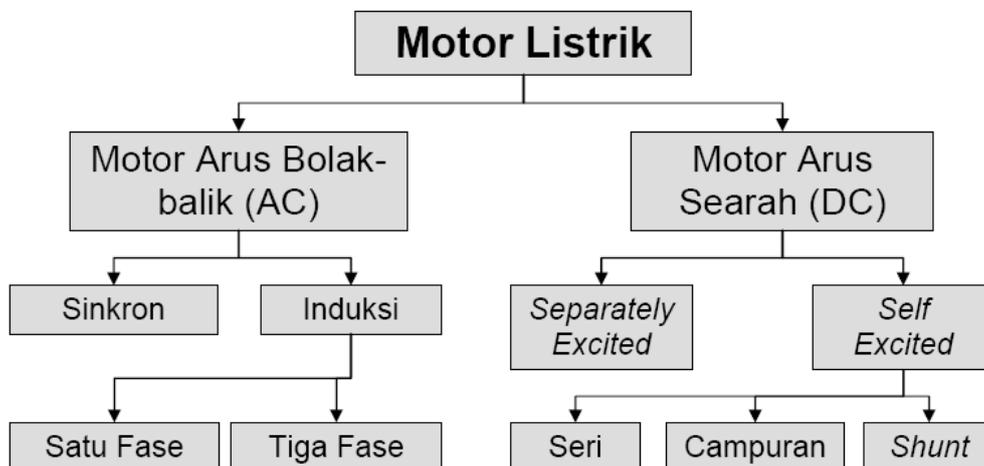
- Arus listrik dalam medan magnet akan memberikan gaya
- Jika kawat yang membawa arus dibengkokkan menjadi sebuah lingkaran/*loop*, maka kedua sisi *loop*, yaitu pada sudut kanan medan magnet, akan mendapatkan gaya pada arah yang berlawanan.
- Pasangan gaya menghasilkan tenaga putar/ *torque* untuk memutar kumparan

- Motor-motor memiliki beberapa *loop* pada dinamonya untuk memberikan tenaga putaran yang lebih seragam dan medan magnetnya dihasilkan oleh susunan elektromagnetik yang disebut kumparan medan.



Gambar 8.2. Prinsip Kerja Motor

Terdapat beberapa jenis motor yang diaplikasikan dalam sistem mekatronika, antara lain: motor AC, motor DC, motor DC *brushless*, motor servo, motor stepper dan motor linear. Termasuk dalam kategori motor AC antara lain adalah motor induksi, motor serempak (sinkron) dan motor tak serempak (asinkron). Salah satu klasifikasi jenis motor yang ada diperlihatkan pada Gambar 8.3.



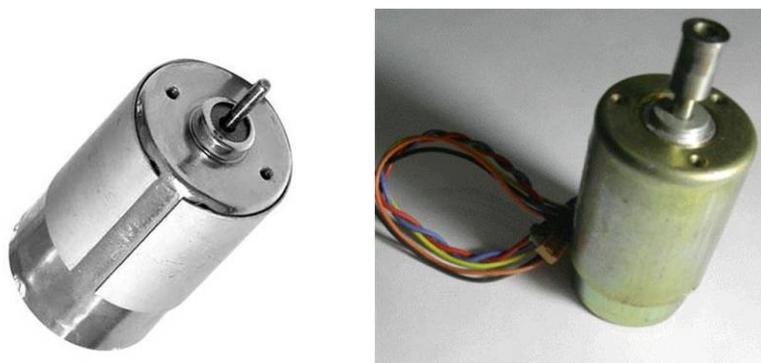
Gambar 8.3. Klasifikasi Jenis Motor

Dalam memahami sebuah motor, penting untuk mengerti apa yang dimaksud dengan beban motor. Beban mengacu kepada keluaran tenaga putar/ *torque* sesuai dengan kecepatan yang diperlukan. Beban umumnya dapat dikategorikan kedalam tiga kelompok (BEE India, 2004):

- **Beban *torque* konstan** adalah beban dimana permintaan keluaran energinya bervariasi dengan kecepatan operasinya namun *torque* nya tidak bervariasi. Contoh beban dengan *torque* konstan adalah *conveyors*, *rotary kilns*, dan pompa *displacement* konstan.
- **Beban dengan variabel *torque*** adalah beban dengan *torque* yang bervariasi dengan kecepatan operasi. Contoh beban dengan variabel *torque* adalah pompa sentrifugal dan fan (*torque* bervariasi sebagai kwadrat kecepatan).
- **Beban dengan energi konstan** adalah beban dengan permintaan *torque* yang berubah dan berbanding terbalik dengan kecepatan. Contoh untuk beban dengan daya konstan adalah peralatan-peralatan mesin.

8.2. Motor DC

Motor DC (Motor Arus Searah), sebagaimana namanya, menggunakan arus langsung yang tidak searah (*direct-unidirectional*). Motor ini merupakan hasil perancangan tertua dibandingkan dengan motor jenis lainnya. Saat ini, motor DC telah menjadi pilihan utama dalam aplikasi industri, khususnya yang memiliki karakter kecepatan yang berbeda, percepatan tetap dan pengendalian torsi.



Gambar 8.4. Motor DC

Motor DC yang memiliki tiga komponen utama, yaitu:

- **Kutub medan.** Secara sederhana digambarkan bahwa interaksi dua kutub magnet akan menyebabkan perputaran pada motor DC. Motor DC memiliki kutub medan yang stasioner dan dinamo yang menggerakkan *bearing* pada ruang diantara kutub medan. Motor DC sederhana memiliki dua kutub medan: kutub utara dan kutub selatan. Garis magnetik energi membesar melintasi bukaan diantara kutub-kutub dari utara ke selatan. Untuk motor yang lebih besar atau lebih kompleks terdapat satu atau lebih elektromagnet. Elektromagnet menerima listrik dari sumber daya dari luar sebagai penyedia struktur medan.
- **Dinamo.** Bila arus masuk menuju dinamo, maka arus ini akan menjadi elektromagnet. Dinamo yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakkan beban. Untuk kasus motor DC yang kecil, dinamo berputar dalam medan magnet yang dibentuk oleh kutub-kutub, sampai kutub utara dan selatan magnet berganti lokasi. Jika hal ini terjadi, arusnya berbalik untuk merubah kutub-kutub utara dan selatan dinamo.
- **Commutator.** Komponen ini terutama ditemukan dalam motor DC. Kegunaannya adalah untuk membalikan arah arus listrik dalam dinamo. *Commutator* juga membantu dalam transmisi arus antara dinamo dan sumber daya.

Terdapat beberapa keuntungan penggunaan motor DC dibandingkan motor lain. Motor DC sangat sederhana desainnya. Sebagaimana dijelaskan di atas, bagian medan magnet permanen motor DC dibangun oleh stator, yang terdiri dari permanen magnet atau elektro-magnetik *winding*. Selain itu, pengendalian kecepatan pada motor DC cukup sederhana. Semakin besar tegangan armaturnya akan semakin cepat putarannya. Hubungan ini linier sehingga mencapai kecepatan maksimumnya. Kemudian, pengendalian torsi juga sederhana. Torsi pada motor DC mempunyai hubungan yang proporsional dengan arusnya. Jika arus dibatasi, maka torsi motor juga dapat dibatasi. Hal ini membuat motor DC dapat digunakan untuk kebutuhan industri yang memerlukan torsi rendah, seperti pabrik tekstil. Keuntungan lain yang dimiliki motor DC adalah kebutuhan terhadap *driver* yang murah dan sederhana. Telah terdapat banyak *driver* motor DC yang dijual secara komersial.

Jadi, motor ini dapat dikendalikan dengan mengatur:

- Tegangan dinamo – meningkatkan tegangan dinamo akan meningkatkan kecepatan
- Arus medan – menurunkan arus medan akan meningkatkan kecepatan.

Motor DC tersedia dalam banyak ukuran, namun penggunaannya pada umumnya dibatasi untuk beberapa penggunaan berkecepatan rendah, penggunaan daya rendah hingga sedang seperti peralatan mesin dan *rolling mills*, sebab sering terjadi masalah dengan perubahan arah arus listrik mekanis pada ukuran yang lebih besar. Juga, motor tersebut dibatasi hanya untuk penggunaan di area yang bersih dan tidak berbahaya sebab resiko percikan api pada sikatnya. Motor DC juga relatif mahal dibanding motor AC.

Untuk memperkuat torsi sebuah motor yang biasanya dinyatakan dalam kg-cm digunakan gear reduksi. Torsi diukur berdasarkan kemampuan sebuah tuas sepanjang 1 cm untuk menggerakkan benda sebesar x kg. Semakin lambat putaran motor akibat penambahan gear maka semakin kuat torsi yang dihasilkan. Perubahan putaran ini berbanding terbalik dengan perbedaan diameter gear. Kecepatan motor akan turun dua kali lipat untuk gear yang dua kali lebih besar. Perlu diperhatikan bahwa gear yang digunakan harus memiliki ukuran gigi yang sama persis.



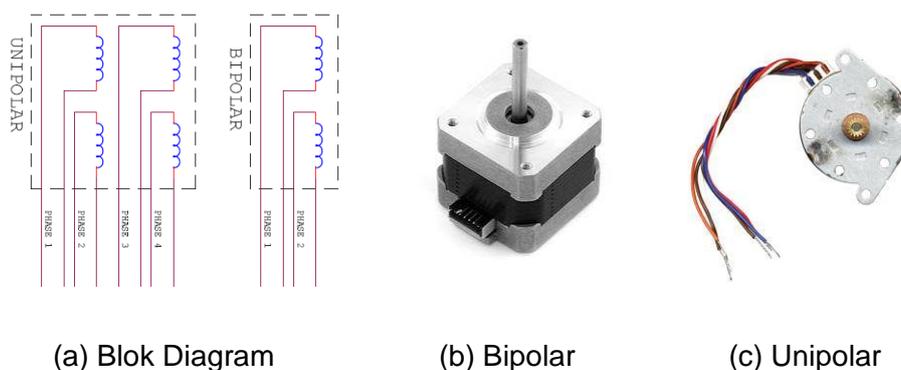
Gambar 8.5. Motor DC dengan Gear

Gambar 8.5. adalah Delta Robo Motor & Wheel Set yang dilengkapi dengan gearbox perbandingan 1 : 32 yang akan meningkatkan torsi motor 32 kali lebih besar daripada yang dihasilkan oleh motor itu sendiri. Untuk mengatur kecepatan gerak dari motor DC digunakan teknik PWM yaitu pengaturan lebar pulsa dalam setiap detiknya. Semakin besar pulsa fase ON dari sebuah motor akan semakin besar pula kecepatan motor tersebut.

8.3. Motor Stepper

Torsi motor stepper tidak sebesar motor DC, namun motor ini mempunyai tingkat presisi yang sangat tinggi dalam gerakannya. Kecepatan gerak motor ini dinyatakan dalam *step per second* atau jumlah step gerakan dalam setiap detiknya.

Secara umum terdapat dua jenis motor stepper yaitu bipolar dan unipolar. Motor stepper unipolar terdiri dari dua buah motor yang masing-masing mempunyai dua buah kumparan sedangkan motor stepper bipolar terdiri dari sebuah motor dengan dua buah kumparan.



Gambar 8.6. Motor Stepper Bipolar dan Unipolar

Pengendalian motor stepper dilakukan dengan mengaktifkan setiap kumparan secara bergantian. Untuk motor stepper unipolar yang terdiri dari 4 kumparan terdapat 4 phase sedangkan untuk motor stepper bipolar yang terdiri dari 2 kumparan terdapat 2 phase. Seringkali untuk menghemat kabel, pada motor stepper unipolar ada beberapa polaritas kumparan yang digabung seperti tampak pada gambar 4.

Pada sebuah robot, motor stepper lebih sering digunakan pada aplikasi penggerak lengan, tuas penjepit dan lain-lain yang tidak memerlukan torsi dan kecepatan yang terlalu besar namun dibutuhkan ketelitian saja.

8.4. Motor Servo

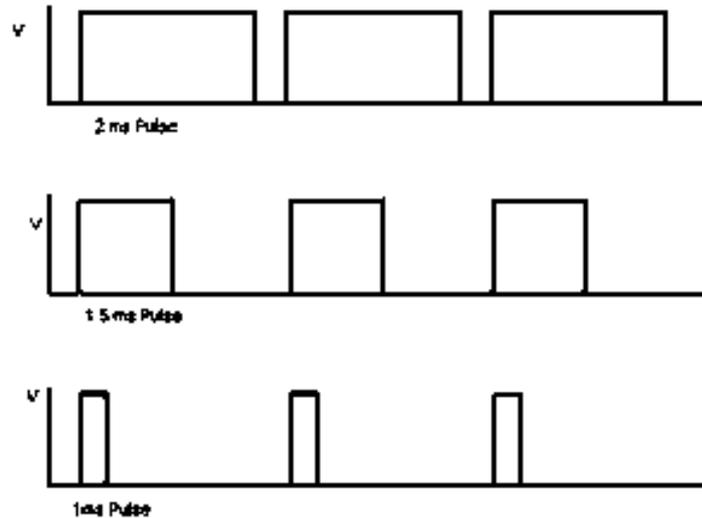
Berbeda dengan motor DC dan motor Stepper, motor servo adalah sebuah motor dengan sistem *closed feedback* di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor, serangkaian gear, potensiometer dan rangkaian kontrol.



Gambar 8.7. Motor Servo

Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor.

Pengontrollan motor servo dilakukan dengan memberikan pulsa secara kontinyu selama 1 ms sampai 2 ms, tergantung posisi yang ingin dicapai. Untuk menjaga posisinya maka pulsa 1 ms sampai 2 ms itu diulang-ulang sekitar 50 sampai 60 kali. Pulsa 1 ms akan memposisikan servo pada 0 derajat, sedangkan 2 ms akan memposisikan servo pada maximum putaran. Jika diberi pulsa 1,5 ms maka servo akan berputar setengah putaran. Hal ini diilustrasikan pada Gambar 8.8.



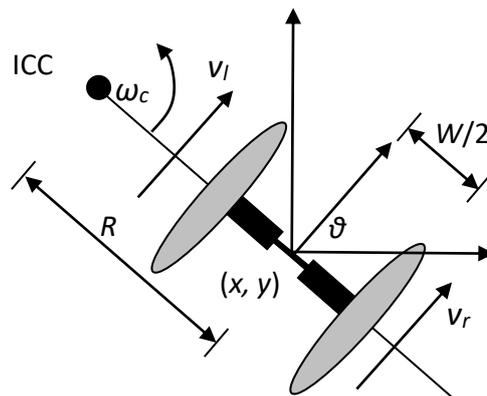
Gambar 8.8. Pembangkitan Sinyal untuk Kendali Motor Servo

Motor servo biasanya hanya bergerak mencapai sudut tertentu saja dan tidak kontinyu seperti motor DC maupun motor stepper. Walau demikian, untuk beberapa keperluan tertentu, motor servo dapat dimodifikasi agar bergerak kontinyu.

Pada robot, motor ini sering digunakan untuk bagian kaki, lengan atau bagian-bagian lain yang mempunyai gerakan terbatas dan membutuhkan torsi cukup besar.

8.5. Motor DC dan Robot Bergerak

Berdasarkan perbandingan, keuntungan dan kemudahan yang ada, motor DC banyak digunakan sebagai penggerak pada robot jenis bergerak (*mobile robot*). Pada umumnya, motor ini digunakan sekaligus 2 (dua) buah untuk mengendalikan pergerakan robot bergerak dengan mengaplikasikan metode *differential drive model*. Pada model ini, gerakan robot dikendalikan dengan mengkombinasikan pergerakan kedua motor.

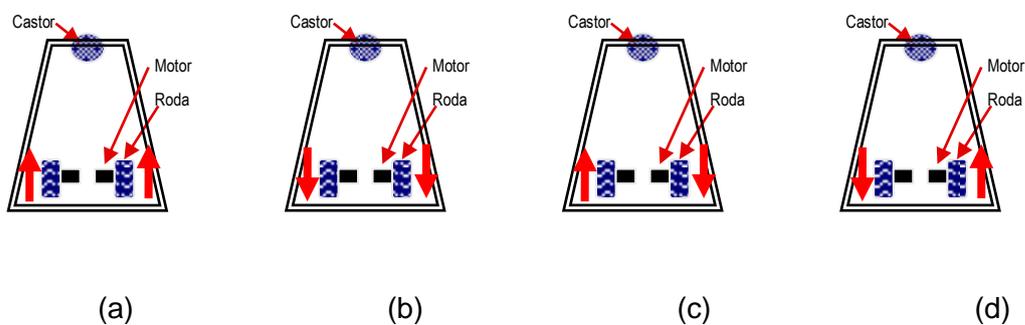


Gambar 8.9. Prinsip *Differential Drive Model*

Terdapat tiga kasus menarik dari kombinasi pergerakan kedua motor tersebut. Yaitu:

1. Jika kecepatan motor roda kanan sama dengan kecepatan motor roda kiri, maka robot akan bergerak ke depan
2. Jika kecepatan motor roda kanan berlawanan dengan kecepatan motor roda kiri, maka robot akan bergerak berputar dengan pusat di tengah-tengah dadan robot
3. Jika salah satu motor tidak bergerak maka robot akan berputar dengan berpusat pada motor yang tidak bergerak tersebut.

Berdasarkan prinsip tersebutlah dapat ditentukan pergerakan robot sesuai dengan keadaan sekitarnya, sebagaimana ditampilkan pada Gambar 8.10.



Gambar 8.10. Algoritma Pergerakan Robot:

(a) Maju, (b) Mundur, (c) Belok Kanan dan (d) Belok Kiri

Namun, untuk dapat mengendalikan motor DC dengan baik menggunakan sistem mikrokontroler dibutuhkan suatu koneksi tertentu. Hal ini disebabkan arus yang keluar dari port keluaran mikrokontroler bernilai kecil untuk dapat menggerakkan motor DC. Selain itu, perputaran arah gerakan motor DC juga harus dikendalikan dengan baik agar tidak menimbulkan efek tegangan EMF balik atau keausan.

Oleh karena itu, modul berikutnya akan menjelaskan mengenai proses pengendalian pergerakan motor DC, antarmuka dan pemrogramannya dengan sistem mikrokontroler.

Daftar Pustaka

1. A. P. Mathur, *Introduction to Microprocessors*, 24th Reprinted., McGraw-Hill Publishing Company Ltd., USA, 2006.
2. ar Publisher, 1998.
3. J. G. Keramas, *Robot Technology*, Delmar Publisher, 1998.
4. K. S. Fu, R. C. Gonzales, C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*, Mc Graw-Hill Book Company, US, 1987.
5. M. Margoilis, *Arduino Cookbook*, O'Reilly, USA 2011
6. M. McRoberts, *Beginning Arduino*, Apress, USA, 2010
7. M. Schmidt, *Arduino A Quick-Start Guide*, Pragmatic Programmers, USA, 2011.



MODUL PERKULIAHAN

Robotika

Pengendalian Arah Putaran Motor DC

Fakultas

Fakultas Teknik

Program Studi

Teknik Elektro

Tatap Muka

09

Kode MK

14043

Disusun Oleh

Julpri Andika, ST., M.Sc.

Abstract

Untuk aplikasi pembuatan robot agar robot tersebut dapat bergerak, biasanya digunakan beberapa buah aktuator. Salah satu aktuator yang paling umum digunakan adalah motor DC. Untuk dapat mengaplikasikan motor DC pada sistem mekatronika, dibutuhkan pengendalian motor DC tersebut.

Kompetensi

Mahasiswa memiliki wawasan tentang pengendalian motor DC, pengendalian arah putaran motor DC dan Rangkaian H-Bridge berbentuk IC

9.1 Pengendalian Motor DC

Dewasa ini pengendalian kecepatan motor listrik memegang peranan penting pada dunia industri, karena motor DC mempunyai karakteristik kopel-kecepatan yang menguntungkan dibandingkan dengan motor AC. Tujuan dari pengontrol kecepatan motor adalah agar dapat menjalankan motor sesuai dengan kecepatan yang diinginkan, meskipun ada perubahan beban. Pada umumnya penerapan motor listrik bekerja pada tiga aspek operasi yaitu pada saat starting, pengendalian kecepatan dan pengereman. Dalam sistem pengendalian dikenal dua sistem yaitu pengendalian sistem loop terbuka ("open loop controller" dan pengendalian sistem loop tertutup ("Closed loop Controller". Pada tulisan ini akan disimulasikan pengaturan kecepatan motor dc dengan sistem pengendalian loop terbuka, ("tanpa umpan balik" yaitu mengatur kecepatan motor dengan menambah tahanan seri pada jangkar (juga menganalisa pengaruh tahanan seri terhadap arus starting dan stop pada motor dc" dan juga pengaturan tegangan jangkar (pengaruh perubahan tegangan jangkar terhadap putaran dan arus jangkar).

Secara sederhana motor listrik bekerja dengan prinsip bahwa dua buah medan magnet dapat dibuat berinteraksi untuk menghasilkan gerakan. Tujuan motor adalah untuk menghasilkan gaya yang menggerakkan yaitu torsi. Biasanya motor dipasang untuk mengerjakan pekerjaan tertentu yang memerlukan kecepatan dan arah putaran yang tepat, sehingga kecepatan putar dan arahnya bisa diatur sedemikian rupa sesuai dengan tujuan penggunaan motor.

Untuk aplikasi pembuatan robot agar robot tersebut dapat bergerak, biasanya digunakan beberapa buah aktuator. Salah satu aktuator yang paling umum digunakan adalah motor DC. Beberapa kemudahan yang ada pada motor DC mengakibatkan jenis motor ini banyak digunakan. Untuk dapat mengaplikasikan motor DC pada sistem mekatronika, dibutuhkan pengendalian motor DC tersebut. Terdapat 2 (dua) jenis pengendalian yang harus dilakukan untuk dapat mengaplikasikan motor DC, yaitu:

1. Pengendalian arah putaran motor DC
2. Pengendalian kecepatan motor DC

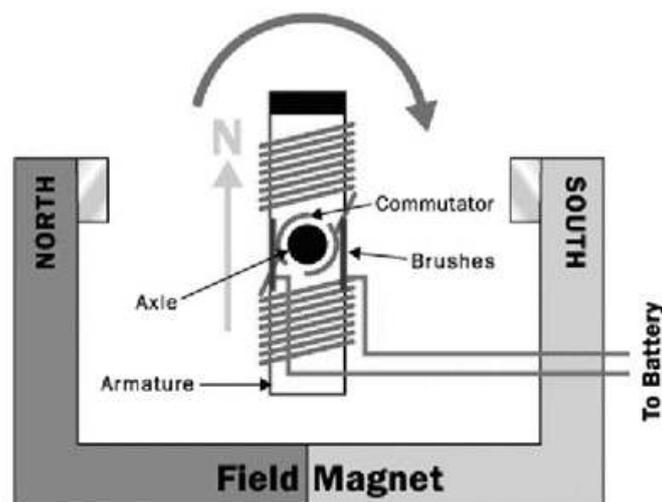
Dengan pengendalian tersebut di atas, motor DC dapat berputar, berhenti berputar, mengubah arah putaran atau bahkan mengubah kecepatan putaran. Kombinasi tersebut akan dimanfaatkan oleh sistem mekatronika untuk melakukan tugas-tugas tertentu.

9.2 Pengendalian Arah Putaran Motor DC

Untuk mengetahui proses pengendalian arah putaran motor, perlu diketahui terlebih dahulu prinsip kerja motor DC. Elemen utama motor DC adalah:

- Magnet
- Armatur atau rotor
- Commutator
- Sikat (*Brushes*)
- As atau poros (*Axle*)

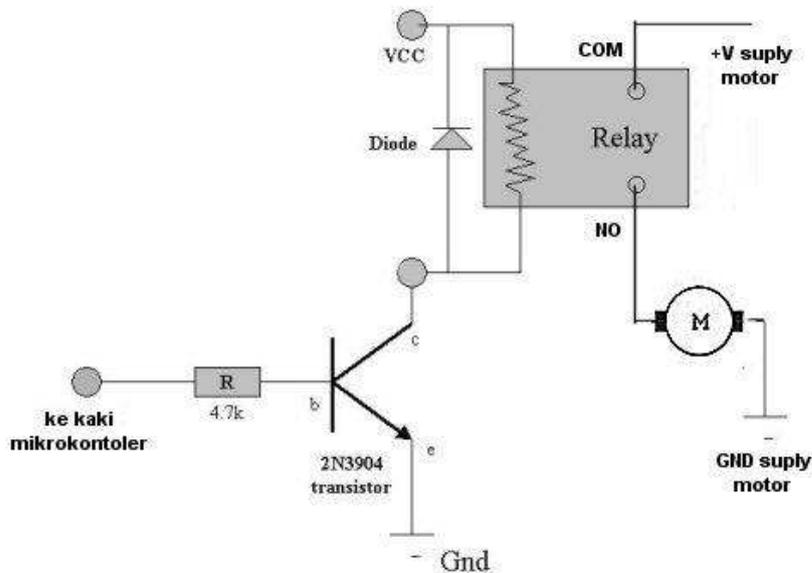
Elemen-elemen ini diperlihatkan pada Gambar 9.1. Dari gambar tersebut, tampak bahwa motor DC berputar sebagai hasil saling interaksi dua medan magnet, yaitu magnet North dan magnet South. Interaksi ini terjadi disebabkan arus yang mengalir pada kumparan.



Gambar 9.1 Elemen Motor DC

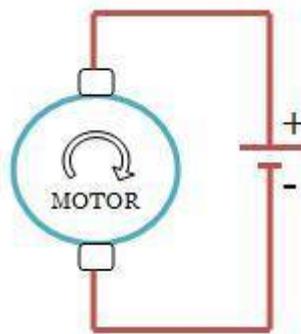
Metode pengendalian motor DC yang paling sederhana adalah dengan memanfaatkan ada – tidaknya arus yang mengalir pada kumparan. Dalam metode ini, arus kepada kedua terminal motor DC dengan beda tegangan sesuai spesifikasi tegangan motor DC dialirkan dan diputus. Kecepatan motor DC yang didapatkan akan maksimal (100%). Biasanya pada metode ini banyak digunakan relay sebagai alat bantu bagi mikrokontroler untuk switch ON/OFF dalam rangka mengalirkan dan memutuskan arus. Relay SPDT ini

dikendalikan melalui transistor penguat, karena arus dari kaki mikrokontroler biasanya tidak cukup kuat untuk mengendalikan relay secara langsung.



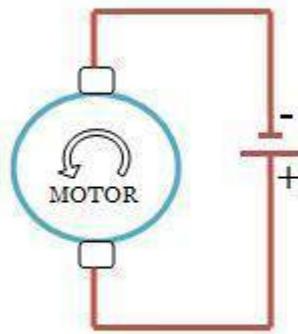
Gambar 9.2. Rangkaian Pengendalian Gerak Motor DC menggunakan Transistor

Namun rangkaian tersebut hanya dapat menggerakkan motor DC dengan satu arah putaran saja. Untuk mendapatkan putaran searah jarum jam (*clockwise, c*), maka motor DC harus diberikan tegangan dengan polaritas bagian atas motor DC positif dan bagian bawahnya negatif. Polarisasi dan arah putaran motor searah jarum jam dapat dilihat pada Gambar 9.3.



Gambar 9.3. Polarisasi Tegangan untuk Putaran Motor Searah Jarum Jam

Untuk mengubah arah tersebut menjadi berlawanan arah dengan jarum jam (*counter clockwise, ccw*), maka polarisasi tegangannya harus dibalik, dimana bagian atas motor DC diberikan tegangan negatif sedangkan bagian bawahnya diberikan tegangan negatif. Gambar 9.4. memperlihatkan polarisasi dan arah putaran motor berlawanan arah jarum jam.

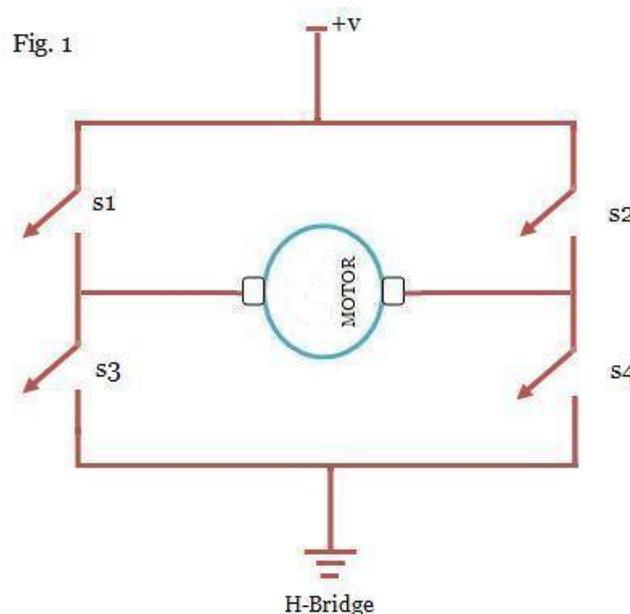


Gambar 9.4. Polarisasi Tegangan untuk Putaran Motor Berlawanan Arah Jarum Jam

Untuk aplikasi elektronika sederhana, perubahan ini dapat dilakukan secara manual dan tidak memerlukan rangkaian tambahan. Namun, untuk aplikasi yang lebih kompleks, seperti pada rangkaian pengendali robot, maka perubahan arah putaran motor DC memerlukan rangkaian tambahan yang lain. Rangkaian yang umum digunakan untuk dapat mengubah arah putaran motor DC secara otomatis dikenal dengan istilah rangkaian *H-Bridge*.

9.3 Pengendalian Arah Motor DC dengan H-Bridge

Dengan menambahkan rangkaian H-Bridge, arah putaran motor DC dapat diubah secara otomatis. Secara umum, rangkaian dasar dari pengendalian H-Bridge merupakan susunan 4 (empat) buah saklar yang disusun dalam formasi jembatan (*bridge*) berbentuk huruf H (oleh karena itu dinamakan H-Bridge) dengan motor DC berada pada titik tengahnya. Posisi keempat buah saklar tersebut disusun sedemikian rupa sehingga dapat memberikan polarisasi tegangan sumber yang melintas pada motor DC berubah-ubah. Prinsip dasar rangkaian H-Bridge diperlihatkan pada Gambar 9.5. Berdasarkan gambar tersebut, maka ada beberapa kemungkinan hal yang terjadi pada motor DC.

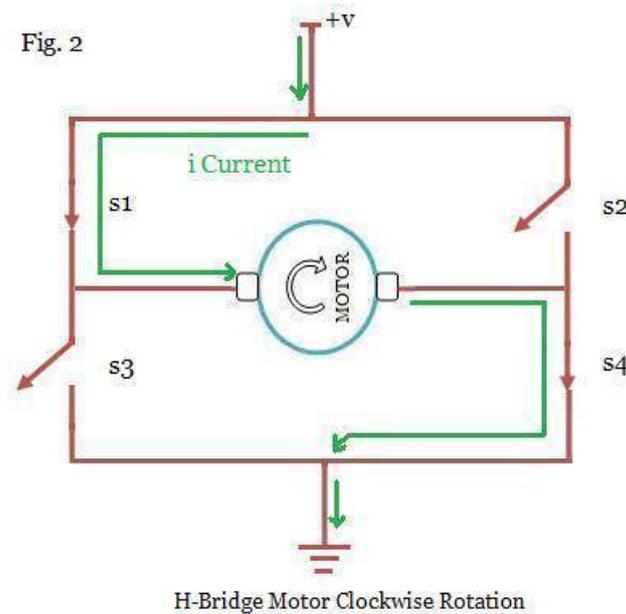


Gambar 9.5 Prinsip Dasar Rangkaian Pengendali H-Bridge

Jika semua saklar dalam keadaan terbuka (seperti yang tampak pada Gambar 9.5), maka motor DC tidak mendapatkan suplai tegangan dari sumbernya, +V. Hal ini mengakibatkan motor DC tidak bergerak sedikitpun.

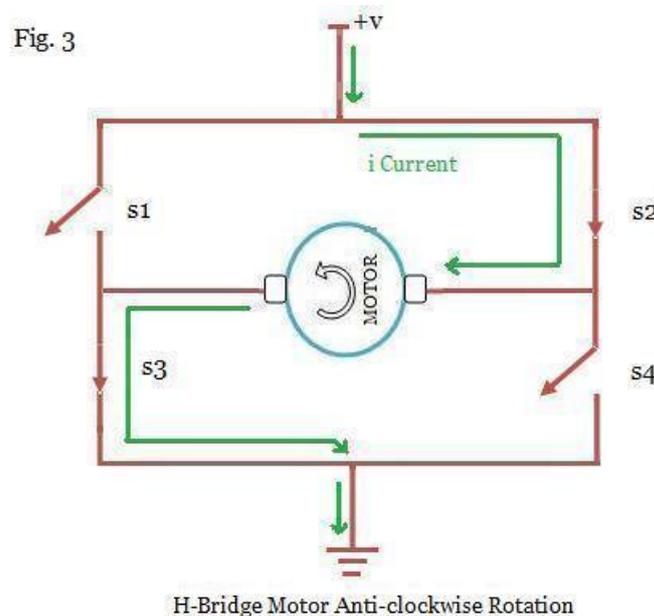
Namun, jika saklar S1 dan saklar S4 dihubungkan, dan saklar S2 serta saklar S3 dibiarkan dalam keadaan terbuka, keadaannya menjadi berbeda. Polarisasi tegangan yang mengakibatkan perbedaan tegangan yang melintasi motor DC diperlihatkan dengan warna hijau pada Gambar 9.6. Pada posisi ini, terminal motor DC sebelah kanan akan mendapatkan tegangan negatif sedangkan terminal motor DC sebelah kiri akan

mendapatkan tegangan positif. Motor DC akan mendapatkan polaritas tegangan yang mengakibatkan motor akan bergerak dengan putaran searah jarum jam.



Gambar 9.6 Posisi Saklar untuk Putaran Motor Searah Jarum Jam

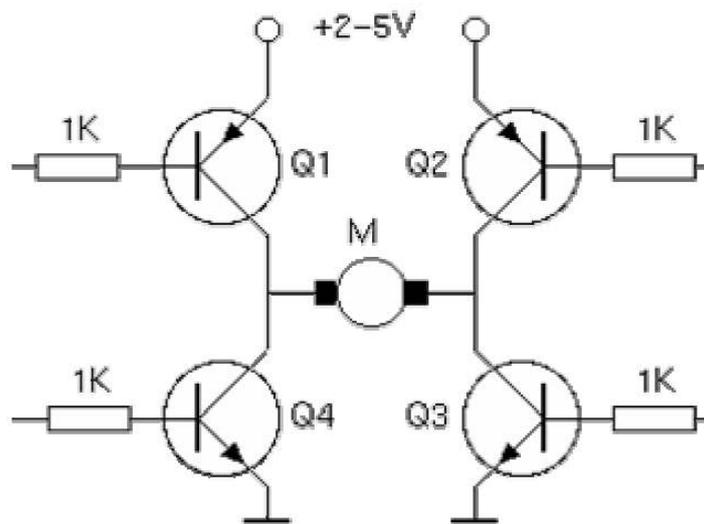
Sebaliknya, putaran motor DC yang berlawanan arah dengan jarum jam, dapat dihasilkan dengan menutup saklar S2 dan S3, serta membiarkan saklar S1 dan S4 terbuka, sebagaimana diperlihatkan pada Gambar 9.7.



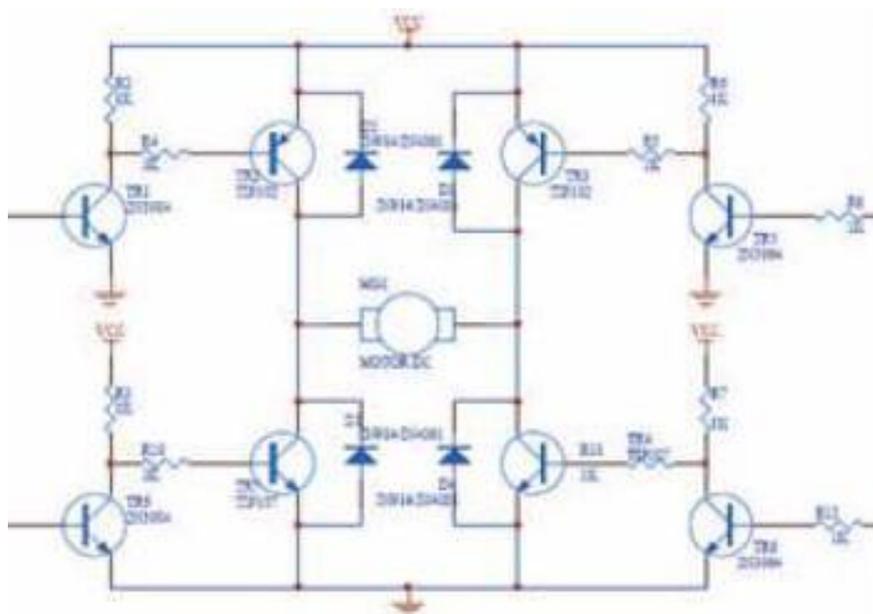
Gambar 9.7 Posisi Saklar untuk Putaran Motor Berlawanan Arah Jarum Jam

Pada posisi ini, terminal motor DC sebelah kanan akan mendapatkan tegangan positif sedangkan terminal motor DC sebelah kiri akan mendapatkan tegangan negatif. Arah polarisasi ini akan mengakibatkan perbedaan tegangan yang melintasi motor DC membuat ia bergerak berlawanan arah jarum jam.

Awalnya, rangkaian H-Bridge ini terdiri dari banyak komponen, seperti transistor (baik transistor BJT, J-FET atau MOSFET) dengan tambahan relay atau beberapa optocoupler. Namun, saat ini, rangkaian tersebut telah diintegrasikan dalam bentuk komponen Operational Ampleifier (OpAmp) bahkan dalam sebuah IC.



(a)



(b)

Gambar 9.8. Rangkaian H-Bridge menggunakan Transistor BJT

Untuk dapat menggunakan Transistor BJT dalam implementasi rangkaian H-Bridge, diperlukan spesifikasi transistor sebagai berikut:

- a. Tegangan kerja maksimum
- b. Arus yang mengalir dari Kolektor ke Emitor
- c. Tegangan dan arus minimum pada Basis untuk mengaktifkan transistor
- d. Frekuensi switching transistor

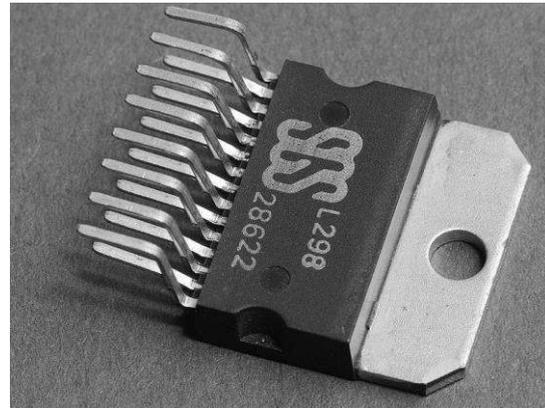
Gambar 9.8(a) memperlihatkan rangkaian dasar H-Bridge dengan menggunakan transistor jenis NPN saja. Secara prinsip rangkaian ini dapat bekerja sesuai teori, dimana transistor bekerja sebagai saklar otomatis. Namun, ketika diaplikasikan rangkaian tersebut tidak dapat bekerja karena tidak memiliki karakter yang dibutuhkan. Oleh karena itu, dikembangkanlah rangkaian H-Bridge seperti tampak pada Gambar 9.8 (b). Rangkaian ini menggunakan gabungan transistor bertingkat jenis NPN dan PNP. Selain itu, rangkaian H-Bridge pada Gambar 9.8(b) telah digunakan 4 buah dioda. Dioda-dioda ini digunakan untuk melindungi tegangan dan arus yang dihasilkan oleh kumparan pada motor DC.

9.4 Rangkaian H-Bridge berbentuk IC

Saat ini, telah tersedia rangkaian H-Bridge dalam bentuk rangkaian terintegrasi (*Integrated Circuit*), sehingga memudahkan perancangan. Beberapa IC yang didalamnya berisi rangkaian H-Bridge antara lain adalah: IC L293D dan IC L298, seperti tampak pada Gambar 9.9. IC seperti ini sering disebut dengan istilah *Motor Driver*.



(a)



(b)

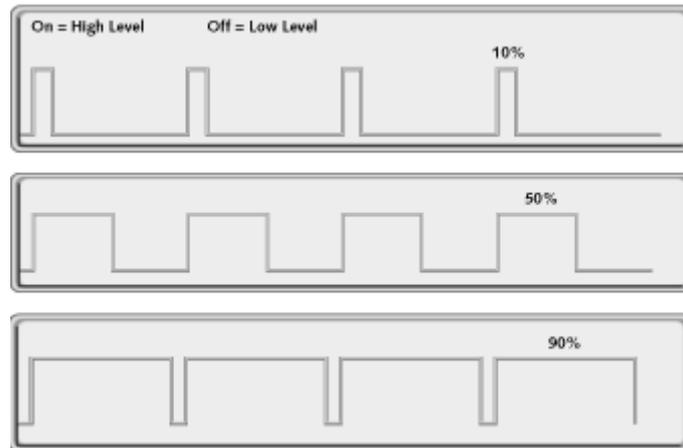
Gambar 9.9. IC H-Bridge: (a) L293D dan (b) L298

Untuk mengimplementasikan IC H-Bridge ini, tidak bisa langsung dihubungkan ke output yang diambil dari pin I/O mikrokontroler. Sebab output dari mikrokontroler hanya mempunyai daya yang sangat kecil. Sedangkan untuk motor DC sendiri, kadang-kadang membutuhkan daya yang tidak kecil (misalnya 200 mA, 1 A atau bahkan lebih). Jika dipaksakan menghubungkan output digital dari mikrokontroler langsung ke motor, bisa jadi merusak mikrokontroler itu sendiri.

Oleh karena itu, terdapat beberapa jenis IC yang dapat digunakan untuk men-*drive* motor DC. Pada dasarnya, IC tersebut adalah memiliki prinsip kerja yang sama, sebagaimana dijelaskan pada bagian sebelumnya. Perbedaannya terletak pada arus yang dapat dikeluarkan untuk mensuplay motor DC. IC L293D, misalnya, dapat mensuplay arus rata-rata hingga 600 mA. Sedangkan IC L298 dapat mensuplay arus total hingga 4 A.

Cara Pengendalian Motor Dc Dengan Mikrokontroler

Metode Pulse Width Modulation (PWM) adalah metode yang cukup efektif untuk mengendalikan kecepatan motor DC. PWM ini bekerja dengan cara membuat gelombang persegi yang memiliki perbandingan pulsa high terhadap pulsa low yang telah tertentu, biasanya diskalakan dari 0 hingga 100%. Gelombang persegi ini memiliki frekuensi tetap (biasanya max 10 KHz) namun lebar pulsa high dan low dalam 1 periode yang akan diatur. Perbandingan pulsa high terhadap low ini akan menentukan jumlah daya yang diberikan ke motor DC.



Gambar 9.10 PWM

Daftar Pustaka

1. A. P. Mathur, *Introduction to Microprocessors*, 24th Reprinted., McGraw-Hill Publishing Company Ltd., USA, 2006.
2. ar Publisher, 1998.
3. J. G. Keramas, *Robot Technology*, Delmar Publisher, 1998.
4. K. S. Fu, R. C. Gonzales, C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*, Mc Graw-Hill Book Company, US, 1987.
5. M. Margoilis, *Arduino Cookbook*, O'Reilly, USA 2011
6. M. McRoberts, *Beginning Arduino*, Apress, USA, 2010
7. M. Schmidt, *Arduino A Quick-Start Guide*, Pragmatic Programmers, USA, 2011.



MODUL PERKULIAHAN

Robotika

Motor Servo

Fakultas

Fakultas Teknik

Program Studi

Teknik Elektro

Tatap Muka

10

Kode MK

MK14043

Disusun Oleh

Julpri Andika, ST., M.Sc.

Abstract

Motor servo adalah sebuah motor dengan sistem closed feedback di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo.

Kompetensi

Mahasiswa mengetahui tentang prinsip dasar mengenai servo.

10.1 Pendahuluan

Motor servo adalah sebuah motor dengan sistem closed feedback di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor, serangkaian gear, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor. Tampak pada gambar dengan pulsa 1.5 mS pada periode selebar 2 mS maka sudut dari sumbu motor akan berada pada posisi tengah. Semakin lebar pulsa OFF maka akan semakin besar gerakan sumbu ke arah jarum jam dan semakin kecil pulsa OFF maka akan semakin besar gerakan sumbu ke arah yang berlawanan dengan jarum jam.

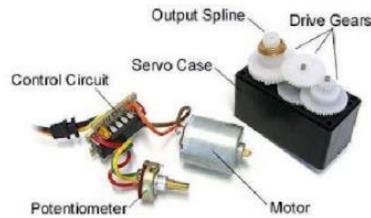
Motor servo biasanya hanya bergerak mencapai sudut tertentu saja dan tidak kontinyu seperti motor DC maupun motor stepper. Walau demikian, untuk beberapa keperluan tertentu, motor servo dapat dimodifikasi agar bergerak kontinyu. Pada robot, motor ini sering digunakan untuk bagian kaki, lengan atau bagianbagian lain yang mempunyai gerakan terbatas dan membutuhkan torsi cukup besar.

Motor servo adalah motor yang mampu bekerja dua arah (CW dan CCW) dimana arah dan sudut pergerakan rotornya dapat dikendalikan hanya dengan memberikan pengaturan duty cycle sinyal PWM pada bagian pin kontrolnya. Motor Servo tampak pada gambar 10.1.

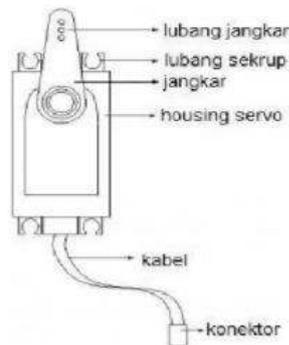


Gambar 10.1 Motor Servo

Motor Servo merupakan sebuah motor DC yang memiliki rangkaian control elektronik dan internal gear untuk mengendalikan pergerakan dan sudut angularnya. Komponen bagian dalam Motor Servo tampak pada gambar 10.2 dan 10.3 pada bagian luarnya.



Gambar 10.2. Komponen bagian dalam Motor Servo



Gambar 10.3. Komponen bagian dalam Motor Servo

Motor servo ini terdiri dari beberapa bagian sebagai berikut:

- Jangkar untuk menghubungkan motor servo dengan obyek-obyek yang akan digerakkan.
- Lubang Jangkar bagian ini berfungsi untuk menempatkan sekrup yang mengaitkan jangkar ke obyek-obyek yang akan digerakkan. Pada gambar tampak lubang jangkardihubungkan ke obyek dengan sekrup untuk gerakan memutar.
- Lubang Sekrup yang berfungsi untuk mengaitkan motor servo dengan tubuh robot
- Housing Servo, di dalam bagian ini terdapat motor DC, gearbox dan rangkaian pengatur sudut servo
- Kabel, kabel yang menghubungkan rangkaian servo dengan pengendali servo

Konektor, konektor 3 pin yang terdiri dari input tegangan positif (+), input tegangan negatif (GND) dan input pulsa (Signal)

Motor servo adalah motor yang berputar lambat, dimana biasanya ditunjukkan oleh rate putarannya yang lambat, namun demikian memiliki torsi yang kuat karena internal gearnya.

Lebih dalam dapat digambarkan bahwa sebuah motor servo memiliki :

- 3 jalur kabel : power, ground, dan control
- Sinyal control mengendalikan posisi
- Operasional dari servo motor dikendalikan oleh sebuah pulsa selebar ± 20 ms, dimana lebar pulsa antara 0.5 ms dan 2 ms menyatakan akhir dari range sudut maksimum.
- Konstruksi didalamnya meliputi internal gear, potensiometer, dan feedback control.

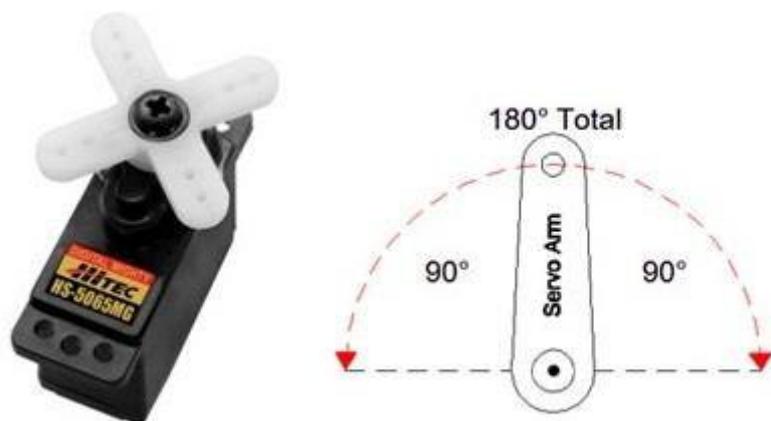
Keunggulan Motor Servo

Keunggulan dari penggunaan motor servo adalah :

- Tidak bergetar dan tidak ber-resonansi saat beroperasi.
- Daya yang dihasilkan sebanding dengan ukuran dan berat motor.
- Penggunaan arus listrik sebanding dengan beban yang diberikan.
- Resolusi dan akurasi dapat diubah dengan hanya mengganti encoder yang dipakai.
- Tidak berisik saat beroperasi dengan kecepatan tinggi.

Aplikasi Motor Servo

Motor servo dapat dimanfaatkan pada pembuatan robot, salah satunya sebagai penggerak kaki robot. Motor servo dipilih sebagai penggerak pada kaki robot karena motor servo memiliki tenaga atau torsi yang besar, sehingga dapat menggerakkan kaki robot dengan beban yang cukup berat. Pada umumnya motor servo yang digunakan sebagai penggerak pada robot adalah motor servo 180°.



Gambar 10.4 Aplikasi Motor Servo

10.2 Jenis-jenis Motor Servo

Ada dua jenis motor servo, yaitu motor servo AC dan DC. Motor servo AC lebih dapat menangani arus yang tinggi atau beban berat, sehingga sering diaplikasikan pada mesin-mesin industri. Sedangkan motor servo DC biasanya lebih cocok untuk digunakan pada aplikasi-aplikasi yang lebih kecil. Dan bila dibedakan menurut rotasinya, umumnya terdapat dua jenis motor servo yang terdapat di pasaran, yaitu motor servo rotation 180° dan servo rotation continuous.

- Motor servo standard:
Motor servo jenis ini hanya mampu bergerak dua arah (CW dan CCW) dengan defleksi masing-masing sudut mencapai 90° sehingga total defleksi sudut dari kanan – tengah – kiri adalah 180°
- Motor servo continuous:
Motor servo jenis ini mampu bergerak dua arah (CW dan CCW) tanpa batasan defleksi sudut putar (dapat berputar secara kontinyu). Sudut putarannya bisa mencapai 360°.

Motor servo standard sering dipakai pada sistem robotika misalnya untuk membuat “Robot Arm” (Robot Lengan) sedangkan motor servo Continuous sering dipakai untuk Mobile Robot.

10.3 Jenis-jenis Horn

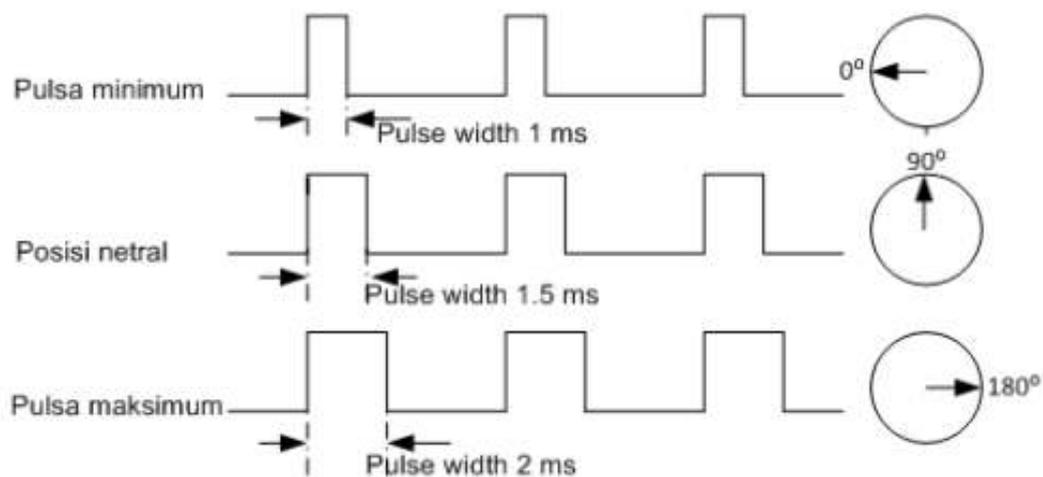
Motor servo merupakan sebuah motor dc kecil yang diberi sistem gear dan potensiometer sehingga dia dapat menempatkan “horn” servo pada posisi yang dikehendaki. Karena motor ini menggunakan sistem close loop sehingga posisi “horn” yang dikehendaki bisa dipertahankan. “Horn” pada servo ada dua jenis. Yaitu Horn “ X” dan Horn berbentuk bulat (seperti pada gambar di samping).



Gambar 10.5 Servo dengan Horn Bulat (kiri) dan Servo dengan Horn X (Kanan)

10.3 Prinsip Kerja Motor Servo

Motor servo dikendalikan dengan memberikan sinyal modulasi lebar pulsa (Pulse Wide Modulation / PWM) melalui kabel kontrol. Lebar pulsa sinyal kontrol yang diberikan akan menentukan posisi sudut putaran dari poros motor servo. Sebagai contoh, lebar pulsa dengan waktu 1,5 ms (mili detik) akan memutar poros motor servo ke posisi sudut 90°. Bila pulsa lebih pendek dari 1,5 ms maka akan berputar ke arah posisi 0° atau ke kiri (berlawanan dengan arah jarum jam), sedangkan bila pulsa yang diberikan lebih lama dari 1,5 ms maka poros motor servo akan berputar ke arah posisi 180° atau ke kanan (searah jarum jam). Lebih jelasnya perhatikan gambar 10.5 dibawah ini.

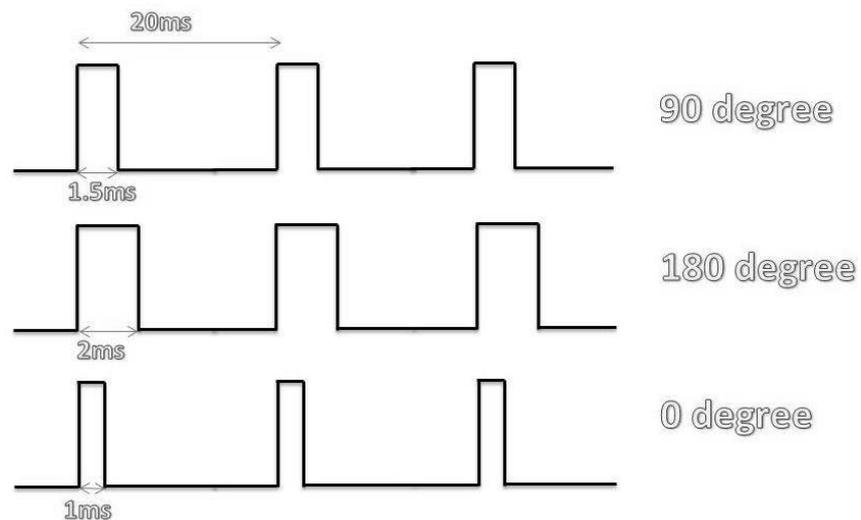


Gambar 10.6 PWM

Ketika lebar pulsa kendali telah diberikan, maka poros motor servo akan bergerak atau berputar ke posisi yang telah diperintahkan, dan berhenti pada posisi tersebut dan akan tetap bertahan pada posisi tersebut. Jika ada kekuatan eksternal yang mencoba memutar atau mengubah posisi tersebut, maka motor servo akan mencoba menahan atau melawan dengan besarnya kekuatan torsi yang dimilikinya (rating torsi servo). Namun motor servo tidak akan mempertahankan posisinya untuk selamanya, sinyal lebar pulsa kendali harus diulang setiap 20 ms (mili detik) untuk menginstruksikan agar posisi poros motor servo tetap bertahan pada posisinya.

Cara Mengendalikan Motor Servo

Untuk menjalankan atau mengendalikan motor servo berbeda dengan motor DC. Karena untuk mengendalikan motor servo perlu diberikan sumber tegangan dan sinyal kontrol. Besarnya sumber tegangan tergantung dari spesifikasi motor servo yang digunakan. Sedangkan untuk mengendalikan putaran motor servo dilakukan dengan mengirimkan pulsa kontrol dengan frekuensi 50 Hz dengan periode 20ms dan duty cycle yang berbeda. Dimana untuk menggerakkan motor servo sebesar 90° diperlukan pulsa dengan ton duty cycle pulsa positif 1,5ms dan unjuk bergerak sebesar 180° diperlukan lebar pulsa 2ms. Berikut bentuk pulsa kontrol motor servo dimaksud.



Gambar 10.7 Lebar Pulsa

10.4 Pemrograman Motor Servo

```
#####Program 1#####
```

```
#include <Servo.h>
Servo myservo;
void setup()
{
  myservo.attach(11);
}
void loop()
{
  myservo.write(0);
  delay(1000);
  myservo.write(90);
  delay(1000);
  myservo.write(180);
  delay(1000);
}
```

```
#####Program 2#####
```

```
include <Servo.h>
Servo myservo;
int pos;

void setup()
{
  myservo.attach(11);
}
void loop()
{
  for (pos=0; pos <= 180; pos +=20)
  {
    myservo.write(pos);
    delay(100);
  }
}
```

```

delay(1000);
for (pos = 180; pos >= 0; pos-=20)
{
myservo.write(pos);
delay(100);
}
delay(1000);
}

```

#####Program 3#####

```

#define rightMotor_Dir 4
#define rightMotor_Vel 5
#define leftMotor_Dir 7
#define leftMotor_Vel 6
#include <Servo.h>
Servo myservo;
int pos;
void setup()
{
myservo.attach(11);
Serial.begin(9600);
pinMode (4, OUTPUT);
pinMode (5, OUTPUT);
pinMode (6, OUTPUT);
pinMode (7, OUTPUT);
}
//fungsi arah putar motor kanan
void leftMotor(int pwm, boolean arah)
{ analogWrite(rightMotor_Vel,pwm);
if(arah) digitalWrite (rightMotor_Dir,HIGH);
else digitalWrite (rightMotor_Dir,LOW);
}
//fungsi arah putar motor kiri
void rightMotor(int pwm, boolean arah)

```

```

{
  analogWrite(leftMotor_Vel,pwm);
  if(arah) digitalWrite (leftMotor_Dir,HIGH);
  else digitalWrite (leftMotor_Dir,LOW);
}
//fungsi gerakan maju
void maju()
{
  rightMotor(255, false);
  leftMotor(255, false);
}
void servo()
{
  myservo.write(0);
  delay(1000);
  myservo.write(90);
  delay(1000);
  myservo.write(180);
  delay(1000);
}
//fungsi gerakan mundur
void mundur()
{
  rightMotor(255, true);
  leftMotor(255, true);
}
//fungsi gerakan belok kanan
void kiri()
{
  rightMotor(255, false);
  leftMotor(0, true);
}
//fungsi gerakan belok kiri
void kanan()
{
  rightMotor(0, true);
  leftMotor(255, false);
}

```

```
}  
void berhenti ()  
{  
  digitalWrite(rightMotor_Vel,LOW);  
  digitalWrite(leftMotor_Vel,LOW);  
}  
void loop()  
{  
  servo ();  
}
```

Daftar Pustaka

1. A. P. Mathur, *Introduction to Microprocessors*, 24th Reprinted., McGraw-Hill Publishing Company Ltd., USA, 2006.
2. ar Publisher, 1998.
3. J. G. Keramas, *Robot Technology*, Delmar Publisher, 1998.
4. K. S. Fu, R. C. Gonzales, C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*, Mc Graw-Hill Book Company, US, 1987.
5. M. Margoilis, *Arduino Cookbook*, O'Reilly, USA 2011
6. M. McRoberts, *Beginning Arduino*, Apress, USA, 2010
7. M. Schmidt, *Arduino A Quick-Start Guide*, Pragmatic Programmers, USA, 2011.



MODUL PERKULIAHAN

Robotika

Limit Switch Sensor

Fakultas

Fakultas Teknik

Program Studi

Teknik Elektro

Tatap Muka

11

Kode MK

MK14043

Disusun Oleh

Julpri Andika, ST., M.Sc.

Abstract

Limit switch merupakan jenis saklar yang dilengkapi dengan katup yang berfungsi menggantikan tombol. Prinsip kerja limit switch sama seperti saklar Push ON yaitu hanya akan menghubungkan pada saat katupnya ditekan pada batas penekanan tertentu yang telah ditentukan dan akan memutuskan saat katup tidak ditekan. Limit switch termasuk dalam kategori sensor mekanis yaitu sensor yang akan memberikan perubahan elektrik saat terjadi perubahan mekanik pada sensor tersebut. Penerapan dari limit switch adalah sebagai sensor posisi suatu benda (objek) yang bergerak.

Kompetensi

Mahasiswa mengetahui tentang prinsip umum sensor, sensor dalam robotika dan sensor limit switch.

11.1 Saklar/Push Button

Dalam dunia industri terdapat berbagai macam jenis-jenis mesin dengan cara kerja yang berbeda-beda dan fungsi yang berbeda pula sehingga menghasilkan output atau hasil yang bervariasi, untuk menggerakkan suatu mesin diperlukan suatu alat yang sangat banyak jenis dan ragamnya, salah satunya adalah Push Button atau saklar tekan.

Pada Push Button (PB), terdapat kontak-kontaknya, yang berupa normaly close (NC) dan normaly open (NO), atau ada juga PB yang memiliki jumlah kontak lebih banyak

Ini adalah gambar Push Button atau dalam bahasa Indonesianya yaitu saklar tekan yang artinya alat ini akan bekerja dengan cara ditekan, alat ini sangat umum, banyak digunakan diberbagai mesin baik itu diindustri ataupun diinstansi pendidikan lainnya, alat ini juga paling mudah untuk dipelajari atau dipahami karena fungsi dan cara kerjanya yang sangat sederhana, pada bagian atasnya terdapat knop yang berfungsi sebagai area penekan (warna merah), lalu disamping kiri dan kanan terdapat terminal, kontak normally open (no) dan normally close (nc) berfungsi sebagai terminal wiring yang dihubungkan dengan alat listrik lainnya, mempunyai kapasitas beban sekitar 5 A



Gambar 11.1 Saklar

Cara Kerja

Alat ini berfungsi sebagai pemberi sinyal masukan pada rangkaian listrik, ketika / selama bagian knopnya ditekan maka alat ini akan bekerja sehingga kontak-kontaknya akan terhubung untuk jenis normally open dan akan terlepas untuk jenis normally close, dan sebaliknya ketika knopnya dilepas kembali maka kebalikan dari sebelumnya, untuk

membuktikannya pada terminalnya bisa digunakan alat ukur tester / ohm meter, pada umumnya pemakaian terminal jenis NO digunakan untuk menghidupkan rangkaian dan terminal jenis NC digunakan untuk mematikan rangkaian, namun semuanya tergantung dari kebutuhan mesin.

Penggunaan

Seperti telah kita ketahui, alat ini sangat banyak digunakan, dalam sebuah operation panel bisa terdapat beberapa Push Button tergantung dari keperluan, alat ini juga memiliki kode warna pada bagian knopnya untuk membedakan fungsi dari masing-masing alat, seperti warna merah digunakan untuk tombol berhenti/stop, lalu warna hitam digunakan untuk tombol jalan/start kemudian warna kuning digunakan untuk tombol reset atau alarm stop, ada beberapa contoh penggunaan Push Button seperti untuk menjalankan motor/pompa, menjalankan conveyor, menghidupkan lampu, mereset alarm, menyalakan bell, menghidupkan cylinder dan masih banyak lagi.

11.2 Limit Switch

Limit switch (saklar pembatas) adalah saklar atau perangkat elektromekanis yang mempunyai tuas aktuator sebagai pengubah posisi kontak terminal (dari Normally Open/ NO ke Close atau sebaliknya dari Normally Close/NC ke Open). Posisi kontak akan berubah ketika tuas aktuator tersebut terdorong atau tertekan oleh suatu objek. Sama halnya dengan saklar pada umumnya, limit switch juga hanya mempunyai 2 kondisi, yaitu menghubungkan atau memutuskan aliran arus listrik. Dengan kata lain hanya mempunyai kondisi ON atau Off.

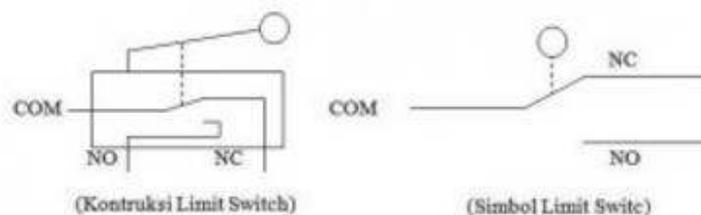


Gambar 11.2 Limit Switch

Limit switch atau dalam bahasa Indonesianya bisa juga disebut sensor pembatas, dalam artian mendeteksi gerakan dari suatu mesin sehingga bisa mengontrolnya atau memberhentikan gerakan dari mesin tersebut sehingga dapat membatasi gerakan mesin dan tidak sampai kebablasan, pemakaiannyapun sangat umum dan banyak, juga mempunyai prinsip kerja yang sederhana, sehingga sangat mudah untuk dipelajari, baik itu oleh pelajar ataupun praktisi dibidangnya, hampir setiap mesin-mesin produksi yang ada di industri menggunakannya, sehingga andaikan ada seorang siswa yang melakukan praktek kerja lapang (PKL) di sebuah industri pasti akan dengan mudah menemukannya.

Limit switch merupakan jenis saklar yang dilengkapi dengan katup yang berfungsi menggantikan tombol. Prinsip kerja limit switch sama seperti saklar Push ON yaitu hanya akan menghubungkan pada saat katupnya ditekan pada batas penekanan tertentu yang telah ditentukan dan akan memutuskan saat saat katup tidak ditekan. Limit switch termasuk dalam kategori sensor mekanis yaitu sensor yang akan memberikan perubahan elektrik saat terjadi perubahan mekanik pada sensor tersebut. Penerapan dari limit switch adalah sebagai sensor posisi suatu benda (objek) yang bergerak.

Limit switch umumnya digunakan untuk : Memutuskan dan menghubungkan rangkaian menggunakan objek atau benda lain. Menghidupkan daya yang besar, dengan sarana yang kecil. Sebagai sensor posisi atau kondisi suatu objek. Prinsip kerja limit switch diaktifkan dengan penekanan pada tombolnya pada batas/daerah yang telah ditentukan sebelumnya sehingga terjadi pemutusan atau penghubungan rangkaian dari rangkaian tersebut. Limit switch memiliki 2 kontak yaitu NO (Normally Open) dan kontak NC (Normally Close) dimana salah satu kontak akan aktif jika tombolnya tertekan. Konstruksi dan simbol limit switch dapat dilihat seperti gambar 11.3.



Gambar 11.3 Konstruksi Dan Simbol Limit Switch

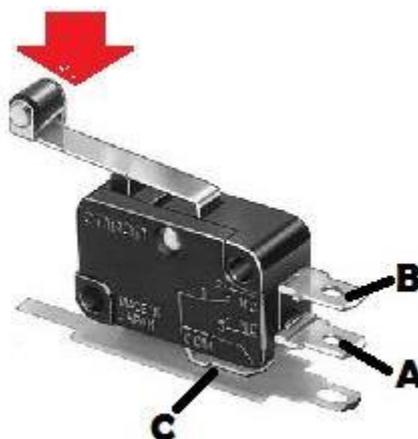
Ada berbagai jenis dan model Limit switch yang ada. Limit switch adalah salah satu sensor yang akan bekerja jika pada bagian actuator nya tertekan suatu benda, baik dari samping kiri ataupun kanan, mempunyai micro switch dibagian dalamnya yang berfungsi untuk mengontakkan atau sebagai pengontak, actuator yang diikat dengan sebuah baud, berfungsi untuk menerima tekanan dari luar, roda berfungsi agar pada saat limit switch menerima tekanan , bisa bergerak bebas, kemudian mempunyai tiga lubang pada body nya berfungsi untuk tempat dudukan baud pada saat pemasangan di mesin.



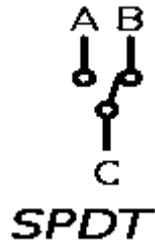
Gambar 11.4 Macam-macam Jenis Limit Switch

Cara Kerja

Ketika actuator dari Limit switch tertekan suatu benda baik dari samping kiri ataupun kanan sebanyak 45 derajat atau 90 derajat (tergantung dari jenis dan type limit switch) maka, actuator akan bergerak dan diteruskan ke bagian dalam dari limit switch, sehingga mengenai micro switch dan menghubungkan kontak-kontaknya, pada micro switch terdapat kontak jenis NO dan NC seperti juga sensor lainnya, kemudian kontak nya mempunyai beban kerja sekitar 5 A, untuk dihubungkan ke perangkat listrik lainnya, dan begitulah seterusnya, selain itu limit switch juga mempunyai head atau kepala tempat dudukan actuator pada bagian atas dari limit switch dan posisinya bisa dirubah-rubah sesuai dengan kebutuhan.



Gambar 11.5 Bentuk Limit Switch



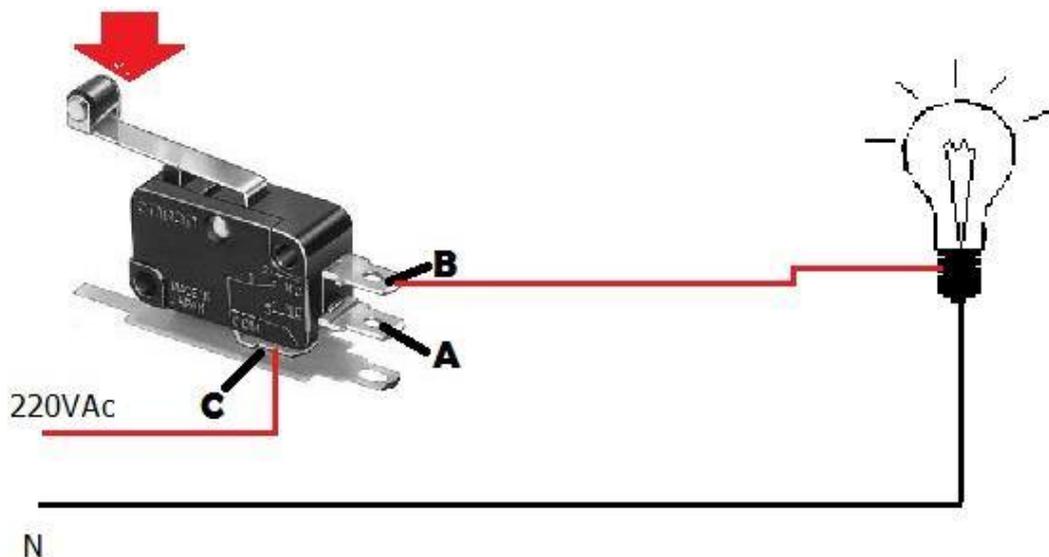
Gambar 11.6 Skema Hubungan Kontak

Cara kerja limit switch adalah bekerja berdasarkan tekanan atau sentuhan benda kerja. Misalnya pada saat pintu pagar bergeser, pada saat pagar sudah mencapai titik tertentu atau menyentuh batas yang diinginkan akan menekan pelat pemicu yang menyebabkan kontak limit switch membuka.

Perhatikan gambar 11.5 diatas.

Pada posisi awal, kontak C-B menutup atau disebut kondisi NC (Normally Close) dan kontak C-A terbuka atau NO (Normally Open). Pada saat saklar limit switch mendapat tekanan pada kaitnya, maka kontak C-B akan membuka dan kontak C-A akan menutup.

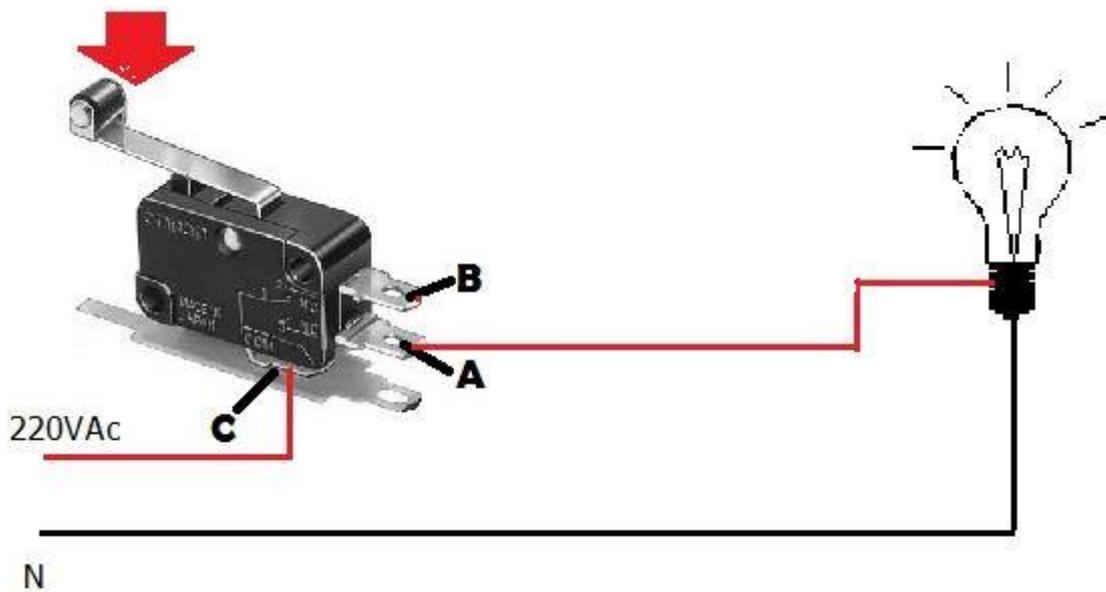
Contoh aplikasi 1 :



Gambar 11.7 Contoh Aplikasi 1

Kontak C disambung ke tegangan listrik 220VAc, dan kontak B disambung ke lampu pijar. Pada aplikasi ini, lampu pijar akan menyala jika pada kait tidak ada tekanan. Dan akan padam jika ada tekanan. Contoh aplikasi pada lampu ruangan yaitu lampu akan menyala jika pintu dibuka. Dan lampu akan padam jika pintu ditutup.

Contoh aplikasi 2.



Gambar 11.8 Contoh Aplikasi 2

Kontak C disambung ke tegangan listrik 220VAc, dan kontak A disambung ke lampu pijar. Pada aplikasi ini, lampu pijar akan menyala jika pada kait mendapat tekanan. Contoh aplikasi pada lampu kamar mandi yaitu lampu akan menyala jika pintu ditutup. Dan lampu akan padam jika pintu dibuka.

Limit switch biasa digunakan pada aplikasi seperti:

1. Pintu gerbang otomatis, dimana limit switch berguna untuk mematikan motor listrik sebelum pintu gerbang itu menabrak pagar pembatas saat membuka atau menutup.
2. Pada pintu panel listrik sebagai saklar otomatis apabila pintu panel dibuka maka lampu akan nyala untuk penerangan (seperti pada kulkas).
3. Pada hoist sebagai pembatas pengangkatan barang.
4. Pada tutup/cover mesin sebagai safety apabila cover dibuka maka mesin akan mati.
5. Pada sistem transfer seperti pada trolley dan conveyor sebagai pembatas maju dan mundurnya (forward reverse).
6. Pada sistem kontrol mesin sebagai sensor untuk mengetahui posisi up/down.

```

##Program##
//Tanggal 13 Desember 2012

#define rightMotor_Dir 4 // arah putar (Motor 1 / kanan)
#define rightMotor_Vel 5 // kontrol kecepatan (Motor 1 / Kanan)
#define leftMotor_Dir 7 // arah putar (Motor 2 / kiri)
#define leftMotor_Vel 6 // kontrol kecepatan (Motor 2 / Kiri)
int limitkanan = 12;
int limitkiri = 13;

void setup()
{
  pinMode(4,OUTPUT); //Setting pin 4 sebagai output
  pinMode(5,OUTPUT); //Setting pin 5 sebagai output
  pinMode(6,OUTPUT); //Setting pin 6 sebagai output
  pinMode(7,OUTPUT); //Setting pin 7 sebagai output
  pinMode(12, INPUT);
  pinMode(13, INPUT);
}

//fungsi arah putar motor kanan
void rightMotor(boolean arah)
{
  digitalWrite(rightMotor_Vel,HIGH); //Motor diberi HIGH agar EN=full
  if(arah) digitalWrite (rightMotor_Dir,HIGH);
  else digitalWrite (rightMotor_Dir,LOW);
}

//fungsi arah putar motor kiri
void leftMotor(boolean arah)
{
  digitalWrite(leftMotor_Vel,HIGH); //Motor diberi HIGH agar EN=full
  if(arah) digitalWrite (leftMotor_Dir,HIGH);
  else digitalWrite (leftMotor_Dir,LOW);
}

//fungsi gerakan maju

```

```

void maju()
{
  rightMotor(false);
  leftMotor(false);
}

//fungsi gerakan mundur
void mundur()
{
  rightMotor(true);
  leftMotor(true);
}

//fungsi gerakan belok kanan
void kiri()
{
  rightMotor(false);
  leftMotor(true);
}

//fungsi gerakan belok kiri
void kanan()
{
  rightMotor(true);
  leftMotor(false);
}

//fungsi berhenti
void berhenti ()
{
  digitalWrite(rightMotor_Vel,LOW); //Motor diberi HIGH agar EN=full speed
  digitalWrite(leftMotor_Vel,LOW); //Motor diberi HIGH agar EN=full speed
}

void loop()
{
  limitkanan = digitalRead (12);

```

```
limitkiri = digitalRead (13);

if (limitkanan == HIGH)
{
  mundur ();
  delay (1000);
  kiri ();
  delay(750);
}

if (limitkiri == HIGH)
{
  mundur ();
  delay (1000);
  kanan ();
  delay (750);
}

if (limitkanan == HIGH && limitkiri == HIGH)
{
  mundur();
  delay (1000);
  kanan();
  delay (1500);

}

else
{
  maju();
}
}
```

Daftar Pustaka

1. A. P. Mathur, *Introduction to Microprocessors*, 24th Reprinted., McGraw-Hill Publishing Company Ltd., USA, 2006.
2. ar Publisher, 1998.
3. J. G. Keramas, *Robot Technology*, Delmar Publisher, 1998.
4. K. S. Fu, R. C. Gonzales, C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*, Mc Graw-Hill Book Company, US, 1987.
5. M. Margoilis, *Arduino Cookbook*, O'Reilly, USA 2011
6. M. McRoberts, *Beginning Arduino*, Apress, USA, 2010
7. M. Schmidt, *Arduino A Quick-Start Guide*, Pragmatic Programmers, USA, 2011.



MODUL PERKULIAHAN

Robotika

Rangkaian Sensor (Flame)

Fakultas
Fakultas Teknik

Program Studi
Teknik Elektro

Tatap Muka

12

Kode MK
MK14043

Disusun Oleh
Julpri Andika, ST., M.Sc.

Abstract

Hasil penelitian menunjukkan secanggih apapun sistem kontrol yang dipakai akan sangat tergantung kepada sensor atau transduser yang digunakan. Sensor dan transduser merupakan peralatan atau komponen yang mempunyai peranan penting dalam sebuah sistem pengaturan otomatis. Ketepatan dan kesesuaian dalam memilih sebuah sensor atau transduser akan sangat menentukan kinerja dari sistem pengaturan secara otomatis.

Kompetensi

Mahasiswa mengetahui tentang prinsip umum sensor, sensor dalam robotika dan sensor flame.

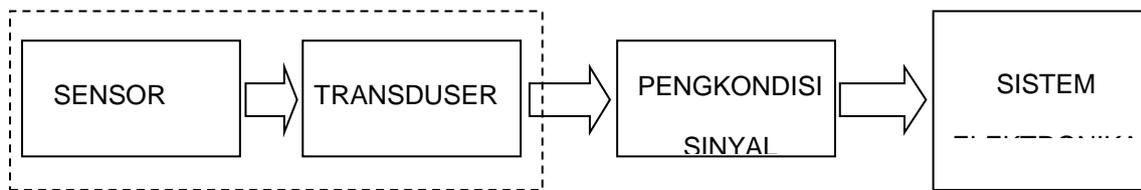
12.1 Prinsip Umum Sensor

Perkembangan otomatisasi manufaktur dan sistem mekatronika sangat bergantung kepada sistem kontrol yang digunakan. Hasil penelitian menunjukkan secanggih apapun sistem kontrol yang dipakai akan sangat tergantung kepada sensor atau transduser yang digunakan. Sensor dan transduser merupakan peralatan atau komponen yang mempunyai peranan penting dalam sebuah sistem pengaturan otomatis. Ketepatan dan kesesuaian dalam memilih sebuah **sensor** atau **transduser** akan sangat menentukan kinerja dari sistem pengaturan secara otomatis.

Besaran masukan pada kebanyakan sistem kontrol adalah bukan besaran listrik, namun dalam bentuk besaran fisika, kimia, mekanis dan sebagainya. Untuk dapat dimanfaatkan, besaran-besaran tersebut haruslah terlebih dahulu dikonversi ke besaran listrik, agar besaran tersebut dapat dimanipulasi dalam rangka fungsi pengontrolannya. Maka, alat pengkonversi sebuah besaran yang bukan listrik menjadi besaran listrik disebut *transducer*. Sedangkan, alat yang merasakan gejala-gejala fisik, kimia, mekanis dan sebagainya tersebut dikenal dengan istilah *sensor*. Terkadang, untuk memudahkan, sensor dan transduser dianggap sebagai satu kesatuan yang disebut sensor atau transduser.

Terdapat banyak definisi mengenai sensor. Salah satu definisi yang cukup komprehensif menyebutkan bahwa sensor adalah untuk mendeteksi dan mengukur suatu kejadian alami, seperti gerak, panas, sinar, variasi mekanis atau kimia dan mengubahnya menjadi representasi tegangan dan arus listrik, baik dalam bentuk analog maupun digital. Sensor menjadi alat yang cukup berperan sangat penting dalam sistem elektronika karena dengannya dapat menjadikan peralatan yang dahulunya sulit untuk diukur atau dideteksi kini menjadi mudah dan otomatis.

Secara umum, sensor digunakan dalam sistem kontrol dan instrumentasi untuk mengubah besaran fisik (seperti temperatur, gaya, kecepatan putaran) menjadi besaran listrik yang proporsional. Blok diagram umum pengaplikasian sensor dan transduser dalam sistem elektronika diperlihatkan pada Gambar 12.1.



Gambar 12.1. Blok Umum Aplikasi Sensor dan Transduser

Agar sistem elektronika yang memanfaatkan sensor dapat berfungsi secara sempurna, maka sensor yang digunakan haruslah dalam kondisi yang optimal. Beberapa persyaratan yang harus dimiliki sensor, antara lain adalah:

1. Linieritas
2. Tidak tergantung temperatur
3. Kepekaan
4. Waktu tanggapan
5. Batas frekuensi terendah dan tertinggi
6. Stabilitas waktu
7. Histerisis

Secara umum berdasarkan fungsi dan penggunaannya sensor dapat dikelompokkan menjadi 3 bagian yaitu:

- a. sensor thermal (panas)
- b. sensor mekanis
- c. sensor optik (cahaya)

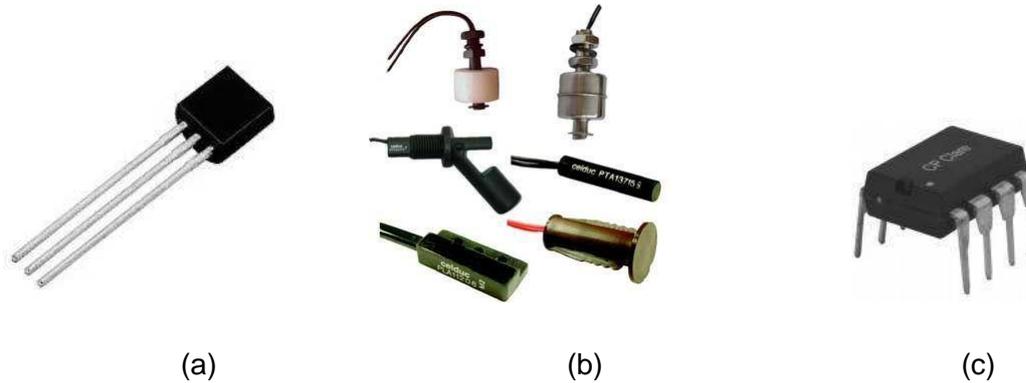
Sensor thermal adalah sensor yang digunakan untuk mendeteksi gejala perubahan panas/temperature/suhu pada suatu dimensi benda atau dimensi ruang tertentu. Contohnya; *bimetal, termistor, termokopel, RTD, infrared pyrometer, hygrometer, dsb.*

Sensor mekanis adalah sensor yang mendeteksi perubahan gerak mekanis, seperti perpindahan atau pergeseran atau posisi, gerak lurus dan melingkar, tekanan, aliran, level dsb. Contoh; *strain gage, linear variable deferential transformer (LVDT), proximity, potensiometer, load cell, bourdon tube, dsb.*

Sedangkan sensor optik atau cahaya adalah sensor yang mendeteksi perubahan cahaya dari sumber cahaya, pantulan cahaya ataupun bias cahaya yang mengenai benda

atau ruangan. Contoh; *photo cell*, *photo transistor*, *photo diode*, *photo voltaic*, *photo multiplier*, *pyrometer optic*, dsb.

Contoh jenis-jenis sensor diperlihatkan pada Gambar 12.2.

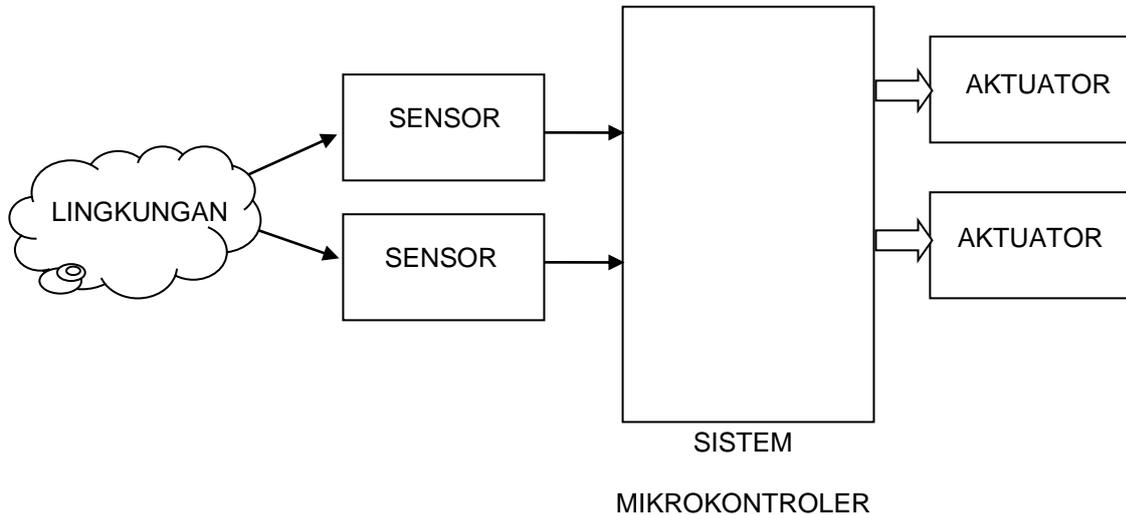


Gambar 12.2. Jenis-jenis Sensor:

(a) Sensor Suhu LM35, (b) Sensor Proximity dan (c) Sensor Optik Optocoupler

12.2 Sensor dalam Robotika

Perkembangan sensor dan transduser sangat cepat sesuai kemajuan teknologi otomasi. Maka, semakin kompleks suatu sistem otomasi dibangun maka semakin banyak jenis sensor yang digunakan. Robotika adalah sebagai contoh penerapan sistem otomasi yang kompleks dimana sensor memiliki peranan yang amat penting. Dalam lingkungan sistem pengendali dan robotika, sensor memberikan kesamaan yang menyerupai mata, pendengaran, hidung, lidah yang kemudian akan diolah oleh kontroler sebagai otaknya. Blok diagram sistem robotika yang menggunakan sensor dapat dilihat pada Gambar 12.3.



Gambar 12.3 Blok Diagram Sistem Robotika dengan Sensor

Cara kerja blok diagram tersebut lebih kurang adalah sebagai berikut. Subsistem sensor menyediakan pengukuran kuantitatif terhadap kenyataan di dalam lingkungan. Pemilihan sensor sebaiknya disesuaikan dengan misi yang akan dijalankan. Selanjutnya subsistem persepsi melakukan proses ekstraksi informasi dari sensor dan interpretasi informasi. Hasil pemrosesan memberikan deskripsi tentang lingkungan secara terbatas sesuai dengan sensor yang dipakai. Keluarannya lalu diberikan ke subsistem basis pengetahuan untuk menentukan aksi yang akan dilakukan sesuai misinya. Proses pengestrakan dan interpretasi informasi, pengkondisian sinyal, basis pengetahuan dan penentuan aksi oleh aktuator dikerjakan di dalam sebuah sistem mikrokontroler. Oleh subsistem perencanaan dan kendali, perintah tersebut diproses lebih lanjut untuk mengendalikan subsistem aktuasi.

Dalam sistem robotika, sensor dapat dikategorikan sebagai berikut:

- a. *Internal sensor*, yaitu sensor yang dipasang di dalam bodi robot.
- b. *External sensor*, yaitu sensor yang dipasang diluar bodi robot.

Sensor internal diperlukan untuk mengamati posisi, kecepatan, dan akselerasi berbagai sambungan mekanik pada robot, dan merupakan bagian dari mekanisme servo. Sedangkan sensor eksternal diperlukan karena dua macam alasan yaitu:

- 1) Untuk keamanan dan
- 2) Untuk penuntun.

Yang dimaksud untuk keamanan” adalah termasuk keamanan robot, yaitu perlindungan terhadap robot dari kerusakan yang ditimbulkannya sendiri, serta keamanan untuk peralatan, komponen, dan orang-orang dilingkungan dimana robot tersebut digunakan. Berikut ini adalah dua contoh sederhana untuk mengilustrasikan kasus diatas.

Contoh pertama: andaikan sebuah robot bergerak ke posisinya yang baru dan ia menemui suatu halangan, yang dapat berupa mesin lain misalnya. Apabila robot tidak memiliki sensor yang mampu mendeteksi halangan tersebut, baik sebelum atau setelah terjadi kontak, maka akibatnya akan terjadi kerusakan.

Contoh kedua: sensor untuk keamanan diilustrasikan dengan problem robot dalam mengambil sebuah telur. Apabila pada robot dipasang pencengkram mekanik (*gripper*), maka sensor harus dapat mengukur seberapa besar tenaga yang tepat untuk mengambil telur tersebut. Tenaga yang terlalu besar akan menyebabkan pecahnya telur, sedangkan apabila terlalu kecil telur akan jatuh terlepas.

Kini bagaimana dengan sensor untuk penuntun atau pemandu?. Katogori ini sangatlah luas, tetapi contoh berikut akan memberikan pertimbangan.

Contoh pertama: komponen yang terletak diatas ban berjalan tiba di depan robot yang diprogram untuk menyemprotnya. Apa yang akan terjadi bila sebuah komponen hilang atau dalam posisi yang salah?. Robot tentunya harus memiliki sensor yang dapat mendeteksi ada tidaknya komponen, karena bila tidak ia akan menyemprot tempat yang kosong. Meskipun tidak terjadi kerusakan, tetapi hal ini bukanlah sesuatu yang diharapkan terjadi pada suatu pabrik.

Contoh kedua: sensor untuk penuntun diharapkan cukup canggih dalam pengelasan. Untuk melakukan operasi dengan baik, robot haruslah menggerakkan tangkai las sepanjang garis las yang telah ditentukan, dan juga bergerak dengan kecepatan yang tetap serta mempertahankan suatu jarak tertentu dengan permukaannya.

Sesuai dengan fungsi sensor sebagai pendeteksi sinyal dan menginformasikan sinyal tersebut ke sistem berikutnya, maka peranan dan fungsi sensor akan dilanjutkan oleh transduser. Karena keterkaitan antara sensor dan transduser begitu erat maka pemilihan transduser yang tepat dan sesuai juga perlu diperhatikan.

12.3 Sensor Flame

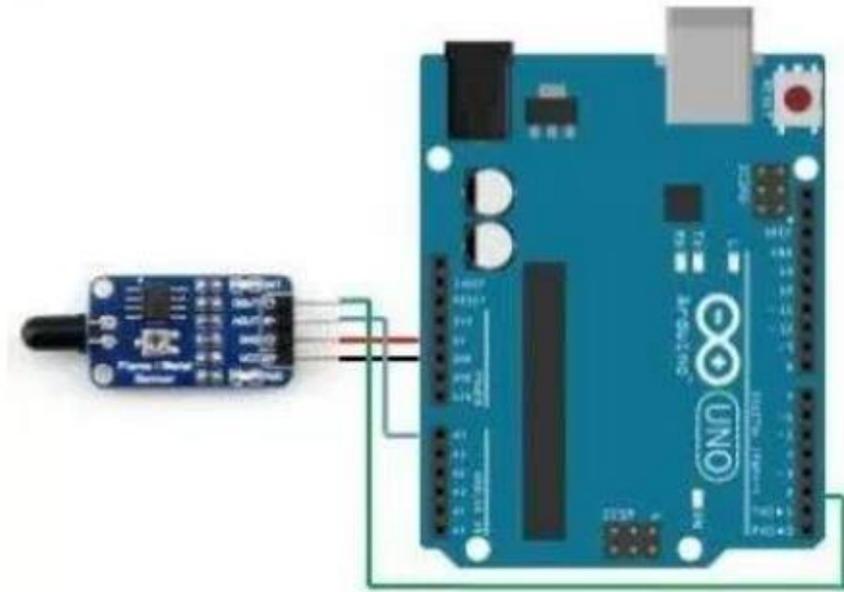
Flame sensor merupakan sensor yang dapat mendeteksi nyala api yang memiliki panjang gelombang antara 760 nm ~ 1100 nm. Sensor ini menggunakan infrared sebagai transduser dalam mensensing kondisi nyala api. Dalam kebanyakan pertandingan kompetisi robot, pendeteksian akan nyala api menjadi salah satu aturan yang umum dalam perlombaan yang tidak akan pernah ditinggalkan.

Oleh sebab itu sensor ini sangat berguna, yang dapat dijadikan sebagai “mata” bagi robot untuk dapat mendeteksi sumber nyala api. Biasanya digunakan pada kompetisi robot fire-fighting maupun soccer robot. Selain itu sensor ini sering juga digunakan untuk mendeteksi api pada ruangan di perkantoran, apartemen, maupun di perhotelan.

Sensor nyala api ini mempunyai sudut pembacaan sebesar 60 derajat, dan beroperasi normal pada suhu 25 – 85 derajat Celcius. Dengan memperhatikan bahwa jarak pembacaan antara sensor dengan objek yang akan dideteksi tidak boleh terlalu dekat, guna menghindari kerusakan sensor (lifetime sensor).

12.4 Fitur dari Sensor Flame

1. Dapat mendeteksi api atau sumber cahaya dengan panjang gelombang antar 760nm s/d 1100nm. Api korek api / mancis dapat dideteksi dari jarak 80cm. Semakin besar api, semakin jauh jarak yang bisa dideteksi.
2. Sudut deteksi sekitar 60 derajat, sensitif terhadap spektrum lidah api.
3. Sensitivitas dapat diatur (dengan potensiometer)
4. Hasil output comparator bersih dengan gelombang yang baik, lebih dari 15mA
5. Tegangan kerja 3.3V-5V
6. Output: digital (0 dan 1) dan analog (tegangan)
7. Dengan lobang baut untuk instalasi
8. Ukuran 3.2cm x 1.4cm
9. Menggunakan pembanding LM393 comparator yang stabil



Skema Pemasangan Flame Sensor dengan Arduino

12.5 Rangkaian Komparator

```
#include <Servo.h>

Servo myservo;

int pos;

void setup()
{
  Serial.begin(9600);
  myservo.attach(11);
}

void loop()
{
  myservo.write(0);
```

```
delay(50);

int sensorValue0=analogRead(A5);

Serial.println(sensorValue0, DEC);

delay (1000);

myservo.write(45);

delay(50);

int sensorValue45=analogRead(A5);

Serial.println(sensorValue45, DEC);

delay (1000);

myservo.write(90);

delay(50);

int sensorValue90=analogRead(A5);

Serial.println(sensorValue90, DEC);

delay (1000);

myservo.write(135);

delay(50);

int sensorValue135=analogRead(A5);

Serial.println(sensorValue135, DEC);

delay (1000);

myservo.write(180);

delay(50);
```

```
int sensorValue180=analogRead(A5);  
  
Serial.println(sensorValue45, DEC);  
  
delay (1000);  
  
}
```

Daftar Pustaka

1. A. P. Mathur, *Introduction to Microprocessors*, 24th Reprinted., McGraw-Hill Publishing Company Ltd., USA, 2006.
2. ar Publisher, 1998.
3. J. G. Keramas, *Robot Technology*, Delmar Publisher, 1998.
4. K. S. Fu, R. C. Gonzales, C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*, Mc Graw-Hill Book Company, US, 1987.
5. M. Margoilis, *Arduino Cookbook*, O'Reilly, USA 2011
6. M. McRoberts, *Beginning Arduino*, Apress, USA, 2010
7. M. Schmidt, *Arduino A Quick-Start Guide*, Pragmatic Programmers, USA, 2011.



MODUL PERKULIAHAN

Robotika

Bluetooth

Fakultas
Fakultas Teknik

Program Studi
Teknik Elektro

Tatap Muka

13

Kode MK
MK14043

Disusun Oleh
Julpri Andika, ST., M.Sc.

Abstract

Bluetooth adalah suatu peralatan media komunikasi yang dapat digunakan untuk menghubungkan sebuah perangkat komunikasi dengan perangkat komunikasi lainnya, bluetooth umumnya digunakan di handphone, komputer atau pc, tablet, dan lain-lain. Fungsi bluetooth yaitu untuk mempermudah berbagi atau sharing file, audio, menggantikan penggunaan kabel dan lain-lain. Saat ini sudah banyak sekali perangkat yang menggunakan bluetooth.

Kompetensi

Mahasiswa mengetahui tentang bluetooth dan modul bluetooth.

13.1 Bluetooth

Bluetooth adalah suatu peralatan media komunikasi yang dapat digunakan untuk menghubungkan sebuah perangkat komunikasi dengan perangkat komunikasi lainnya, bluetooth umumnya digunakan di handphone, komputer atau pc, tablet, dan lain-lain. Fungsi bluetooth yaitu untuk mempermudah berbagi atau sharing file, audio, menggantikan penggunaan kabel dan lain-lain. Saat ini sudah banyak sekali perangkat yang menggunakan bluetooth.

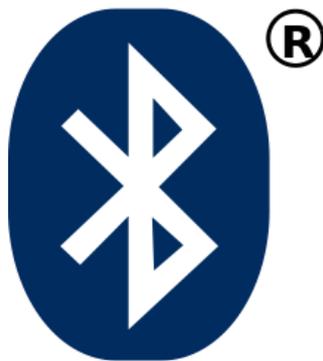
Atau definisi bluetooth yang lainnya adalah sebuah teknologi komunikasi wireless atau tanpa kabel yang beroperasi dalam pita frekuensi 2,4 GHz (antara 2.402 GHz s/d 2.480 GHz) dengan menggunakan sebuah frequency hopping tranceiver yang mapu menyediakan layanan komunikasi data dan juga suara secara real-time antara host-host bluetooth dengan jarak jangkauan layanan yang terbatas.

Pada dasarnya teknologi bluetooth ini diciptakan bukan hanya untuk menggantikan atau menghilangkan penggunaan media kabel dalam melakukan pertukaran data atau informasi, tetapi juga mampu menawarkan fitur yang bagus atau baik untuk teknologi mobile wireless atau tanpa kabel, dengan biaya yang relatif rendah, konsumsi daya rendah, interoperability yang sangat menjanjikan, mudah dalam pengoperasiannya dan juga mampu menyediakan berbagai macam layanan.

Nama "bluetooth" berasal dari nama raja di akhir abad sepuluh, Harald Blatand (Abad 10) yang di Inggris juga dijuluki Harald Bluetooth kemungkinan karena memang giginya berwarna gelap. Ia adalah raja Denmark yang telah berhasil menyatukan suku-suku yang sebelumnya berperang, termasuk suku dari wilayah yang sekarang bernama Norwegia dan Swedia. Bahkan wilayah Scania di Swedia, tempat teknologi bluetooth ini ditemukan juga termasuk daerah kekuasaannya. Kemampuan raja itu sebagai pemersatu juga mirip dengan teknologi bluetooth sekarang yang bisa menghubungkan berbagai peralatan seperti komputer personal dan telepon genggam. Sedangkan logo bluetooth berasal dari penyatuan dua huruf Jerman yang analog dengan huruf H dan B (singkatan dari Harald Bluetooth), yaitu (Hagall) dan (Blatand) yang kemudian digabungkan.

Awal mula dari Bluetooth adalah sebagai teknologi komunikasi wireless (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 GHz unlicensed ISM (Industrial, Scientific and Medical) dengan menggunakan sebuah frequency hopping tranceiver yang mampu menyediakan layanan komunikasi data dan suara secara real-time antara host-host bluetooth dengan jarak jangkauan layanan yang terbatas (sekitar 10 meter). Bluetooth berupa card yang menggunakan frekuensi radio standar IEEE 802.11 dengan jarak layanan yang terbatas dan kemampuan data transfer lebih rendah dari card untuk Wireless Local Area Network (WLAN).

Pembentukan Bluetooth dipromotori oleh 5 perusahaan besar Ericsson, IBM, Intel, Nokia dan Toshiba membentuk sebuah Special Interest Group (SIG) yang meluncurkan proyek ini. Pada bulan Juli 1999 dokumen spesifikasi bluetooth versi 1.0 mulai diluncurkan. Pada bulan Desember 1999 dimulai lagi pembuatan dokumen spesifikasi bluetooth versi 2.0 dengan tambahan 4 promotor baru yaitu 3Com, Lucent Technologies, Microsoft dan Motorola. Saat ini, lebih dari 1800 perusahaan di berbagai bidang bergabung dalam sebuah konsorsium sebagai adopter teknologi bluetooth. Walaupun standar Bluetooth SIG saat ini 'dimiliki' oleh grup promotor tetapi ia diharapkan akan menjadi sebuah standar IEEE (802.15).



Gambar 2.2 Logo Bluetooth

Prinsip Kerja

Sistem bluetooth terdiri atas: sebuah radio transceiver, baseband link Management dan Control, Baseband (processor core, SRAM, UART, PCM USB Interface), flash dan voice codec.

- Baseband link controller menghubungkan hardware atau perangkat keras radio ke baseband processing dan juga layer protokol fisik.
- Link manager melakukan aktivitas protokol tingkat tinggi, yaitu seperti melakukan link setup, autentikasi dan juga konfigurasi.

Kelebihan:

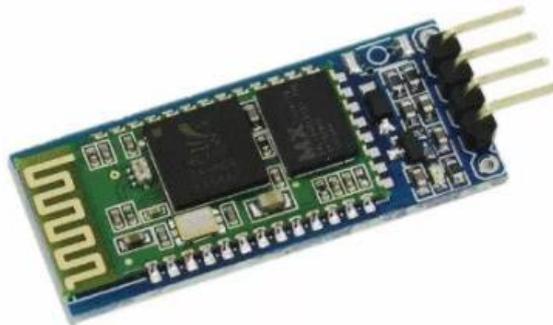
- Bisa menembus rintangan, misalnya seperti dinding, kotak, dan sebagainya. Walaupun jarak transmisinya hanya 10 M.
- Tidak memerlukan media kabel ataupun kawat.
- Dapat mensinkronisasi data dari Handphone ke Komputer atau laptop.
- Dapat dipakai sebagai perantara modem.
- Praktis dan tidak ribet dalam penggunaannya.

Kekurangan:

- Memakai frekuensi yang sama dengan gelombang WiFi.
- Kalau terlalu banyak koneksi bluetooth didalam satu ruangan, akan sulit untuk menemukan penerima yang dituju.
- Sering beredar virus-virus yang disebarkan melalui bluetooth, khususnya dari handphone.
- Cukup banyak mekanisme keamanan yang harus diperhatikan untuk mencegah kegagalan pengiriman data atau penerimaan data maupun informasi.
- Kecepatan dalam transfer data tidak tetap, tergantung dari perangkat yang dipakai untuk mengirim dan yang menerima data maupun informasi.

13.2 Modul Bluetooth

Bluetooth HC-05 adalah salah satu modul bluetooth yang sering digunakan oleh kebanyakan user mikrokontroler seperti arduino dikarenakan pemakaiannya yang mudah, harganya murah, dan banyak tersedia di pasaran. Kelebihan dari modul bluetooth HC-05 ini adalah bisa digunakan sebagai master maupun digunakan sebagai slave, berbeda dengan bluetooth tipe HC-06 yang hanya bisa digunakan sebagai slave.



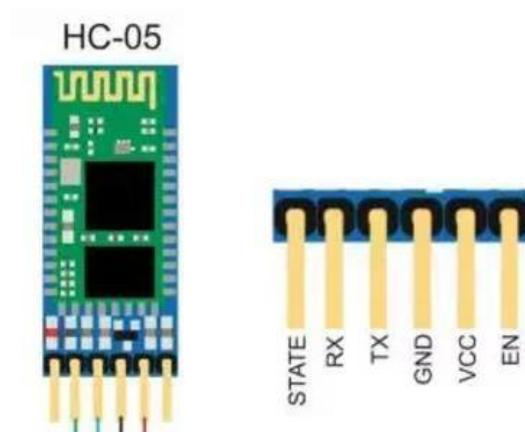
Gambar 13.1 Modul Bluetooth HC-05

Spesifikasi :

- Bluetooth protocol: Bluetooth Specification v2.0+EDR
- Frequency: 2.4GHz ISM band
- Modulation: GFSK(Gaussian Frequency Shift Keying)
- Emission power: 4dBm, Class 2
- Sensitivity: -84dBm at 0.1% BER
- Speed: Asynchronous: 2.1Mbps(Max) / 160 kbps, Synchronous: 1Mbps/1Mbps
- Security: Authentication and encryption
- Profiles: Bluetooth serial port

- Power supply: +3.3VDC 50mA
- Working temperature: -20 ~ +75 Centigrade
- Dimension: 3.57cm x 1.52cm

Modul Bluetooth HC-05 terdiri dari 6 pin konektor, yang setiap pin konektor memiliki fungsi yang berbeda – beda, untuk lebih jelasnya bisa dilihat gambar di bawah ini :



Gambar 13.2 Pin-pin Modul HC-05

Pin bluetooth HC-05 yaitu :

- EN fungsinya untuk mengaktifkan mode AT Command Setup pada modul HC-05. Jika pin ini ditekan sambil ditahan sebelum memberikan tegangan ke modul HC-05, maka modul akan mengaktifkan mode AT Command Setup. Secara default, modul HC-05 aktif dalam mode Data.
- Vcc adalah pin yang berfungsi sebagai input tegangan. Hubungkan pin ini dengan sumber tegangan 5V.
- GND adalah pin yang berfungsi sebagai ground. Hubungkan pin ini dengan ground pada sumber tegangan.
- TX adalah pin yang berfungsi untuk mengirimkan data dari modul ke perangkat lain (mikrokontroler). Tegangan sinyal pada pin ini adalah 3.3V sehingga dapat langsung dihubungkan dengan pin RX pada arduino karena tegangan sinyal 3.3V dianggap sebagai sinyal bernilai HIGH pada arduino.
- RX adalah pin yang berfungsi untuk menerima data yang dikirim ke modul HC-05. Tegangan sinyal pada pin sama dengan tegangan sinyal pada pin TX, yaitu 3.3V. Untuk keamanan, sebaiknya gunakan pembagi tegangan jika menghubungkan pin ini dengan arduino yang bekerja pada tegangan 5V. Pembagi tegangan tersebut menggunakan 2 buah resistor. Resistor yang digunakan sebagai pembagi tegangan

pada tutorial ini adalah 1K ohm dan 2K ohm. Untuk lebih jelasnya, dapat dilihat pada bagian implementasi koneksi antara modul HC-05 dan arduino UNO.

- STATE adalah pin yang berfungsi untuk memberikan informasi apakah modul terhubung atau tidak dengan perangkat lain.

Seperti dijelaskan di atas, modul HC-05 memiliki dua mode kerja yaitu mode AT Command dan mode Data. Modul HC-05 menggunakan mode Data secara default. Berikut ini adalah keterangan untuk kedua mode tersebut:

AT Command. Pada mode ini, modul HC-05 akan menerima instruksi berupa perintah AT Command. Mode ini dapat digunakan untuk mengatur konfigurasi modul HC-05. Perintah AT Command yang dikirimkan ke modul HC-05 menggunakan huruf kapital dan diakhiri dengan karakter CRLF (`\r\n` atau `0x0d 0x0a` dalam heksadesimal).

Data. Pada mode ini, modul HC-05 dapat terhubung dengan perangkat bluetooth lain dan mengirimkan serta menerima data melalui pin TX dan RX. Konfigurasi koneksi serial pada mode ini menggunakan baudrate: 9600 bps, data: 8 bit, stop bits: 1 bit, parity: None, handshake: None. Adapun password default untuk terhubung dengan modul HC-05 pada mode Data adalah 0000 atau 1234.

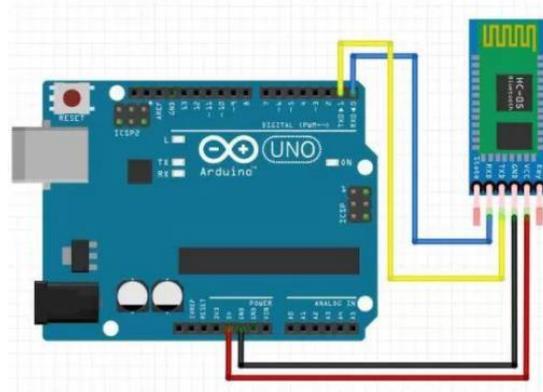
Dari 6 pin yang terdapat pada HC 05, hanya digunakan 4 pin saja yaitu Vcc, Gnd, Tx, dan Rx.

Alat dan komponen :

- mikrokontroler Arduino
- kabel jumper
- project board
- Modul bluetooth HC-05

Sebelum menggunakan bluetooth HC-05 alangkah baiknya dilakukan konfigurasi terlebih dahulu, yaitu masuk ke mode AT-Command. Apa itu AT Command? AT Command adalah perintah-perintah yang digunakan dalam komunikasi dengan serial port, hal ini juga berlaku untuk modul HC 05 sehingga user bisa mengubah status master dan slave, nama bluetooth, baud rate, dan password nya. Sebenarnya ada banyak cara untuk masuk ke mode AT-Command HC 05 tetapi mungkin bagi sebagian user yang pemula mungkin terkadang akan terasa membingungkan, maka dari itu admin memberikan salah satu cara yang mudah dan efektif untuk masuk ke mode AT Command bluetooth HC 05. Cara masuk ke AT command di bawah ini sangat mudah dan simpel dikarenakan tidak membutuhkan program ataupun software khusus.

Rangkaianlah seperti gambar di bawah ini :



GAMBAR 13.3 Pemasangan HC-05 dengan Arduino

- hubungkan pin RX ke RX
- hubungkan pin TX ke TX
- hubungkan VCC ke 5 volt DC dari arduino
- hubungkan GND ke gnd arduino
- untuk masuk ke mode AT command ini , RX hubungkan ke RX TX hubungkan ke TX

Setelah itu masukkan program kosong dari arduino IDE ke board arduino seperti di bawah ini n:

```
sketch_dec30a | Arduino 1.6.8
File Edit Sketch Tools Help
sketch_dec30a
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
```

Gambar 13.4 Program Kosong

- setelah selesai,cabut kabel arduino
- tekan dan tahan tombol pada modul bluetooth HC-05 dan pasang kabel arduino ke port usb komputer/laptop anda.



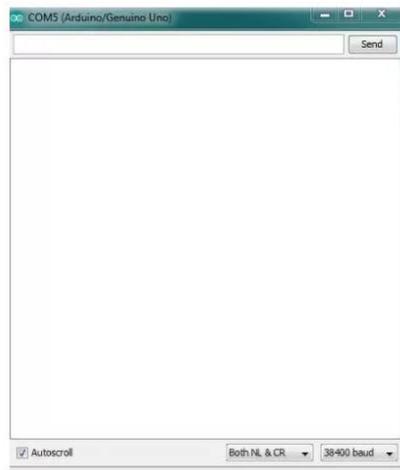
Gambar 13.5 Tombol reset

- lepas setelah terpasang selama 3 detik
- maka lampu pada modul bluetooth akan berkedip secara perlahan
- kemudian masuklah ke serial monitor arduino ide di pojok kanan atas



Gambar 13.6 Serial Monitor

- setting seperti di bawah ini yaitu pilih Both NL & CR dan 38400 baud

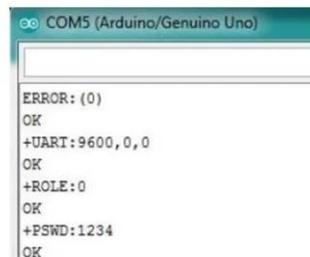


Gambar 13.7

- ketik AT kemudian send
- akan muncul pesan error,tapi itu bukan masalah. Ketik saja lagi AT



*akan muncul pesan OK



Gambar 13.8

Daftar perintah di AT Command :

AT+ROLE “untuk cek baud status master atau slave”

AT+NAME “untuk cek nama”

AT+UART “untuk cek baud rate”

AT+ROLE=0 “mengeset modul ke mode slave”

AR+ROLE=1 “mengeset modul ke mode master ”

AT+PSWD=(password kamu) “mengeset pasword ke pasword yang diinginkan”

AT+UART=9600,0,0 “mengeset baud rate ke 9600”

###Program###

```
#include <Servo.h>
#define rightMotor_Dir 7
#define rightMotor_Vel 6
#define leftMotor_Dir 4
#define leftMotor_Vel 5
Servo myservo;
int pos;
int pos1=0;
int LSKanan=8;
int LSKiri=9;
int val1=0;
int val2=0;
int val3=0;
int val4=0;
int n=0;
long int timer1=0;
long int t1=0;
long int t2=0;
char dataIn = 'S';
int vel=0;
boolean stat=0;
```

```

void setup(){
  Serial.begin(9600);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(LSKanan, INPUT);
  pinMode(LSKiri, INPUT);
  myservo.attach(11);}

void leftMotor(int pwm, boolean arah){
  analogWrite(rightMotor_Vel,pwm);
  if(arah) digitalWrite (rightMotor_Dir,HIGH);
  else digitalWrite (rightMotor_Dir,LOW);}

void rightMotor(int pwm, boolean arah){
  analogWrite(leftMotor_Vel,pwm);
  if(arah) digitalWrite (leftMotor_Dir,HIGH);
  else digitalWrite (leftMotor_Dir,LOW);}

void maju(){
  rightMotor(75, false);
  leftMotor(75, false);
  n+=1;}

void mundur(){
  rightMotor(75, true);
  leftMotor(75, true);}

void kiri(){
  rightMotor(75, false);
  leftMotor(0, true);}

void kanan(){
  rightMotor(0, true);
  leftMotor(75, false);}

void berhenti(){
  digitalWrite(rightMotor_Vel, LOW);
  digitalWrite(leftMotor_Vel, LOW);}

void scanning(){
  myservo.write(0);
  delay(50);
  for (pos=0; pos<=180; pos+=5){
    myservo.write(pos);
    int val3 = analogRead (A5);
    if (val3>200){
      if (val4<val3){

```

```

        val4=val3;pos1=pos;}
    }
    delay(50);
}
for(pos=180; pos>=0; pos-=5){
    myservo.write(pos);
    int val3 = analogRead (A5);
    if (val3>200){
        if (val4<val3){
            val4=val3;pos1=pos;}
        }
    delay(50);
}
}

void flame(){
    berhenti();
    scanning();
    delay(1000);
    if (pos1>90){
        kiri();
        t1=pos1-90;
        t2=t1*1000;
        timer1=t2/90;
        delay(timer1);
        pos1=0;val4=0;timer1=0;t1=0;t2=0;
    }
    else if ((pos1<90) && (pos1!=0)){
        kanan();
        t1=90-pos1;
        t2=t1*1000;
        timer1=t2/90;
        delay(timer1);
        pos1=0;val4=0;timer1=0;t1=0;t2=0;
    }
}

void otomatis(){
    maju();
    dataIn = Serial.read();
    if ((dataIn == 'x')&&(stat == 1)){stat = 0;berhenti();}

    pos1=0;
    val2 = digitalRead (LSKanan);
    val1 = digitalRead (LSKiri);

    if ((val1==HIGH)&&(val2==HIGH)){
        berhenti();delay(500);
        mundur();delay(600);
        berhenti();delay(500);
    }
}

```

```

        kanan();delay(740);
        n=10000;
    }
    else if ((val1==HIGH)&&(val2==LOW)){
        berhenti();delay(500);
        mundur();delay(600);
        kiri();delay(740);
        n=10000;
        berhenti();
    }
    else if ((val1==LOW)&&(val2==HIGH)){
        berhenti();delay(500);
        mundur();delay(600);
        kanan();delay(740);
        n=10000;
        berhenti();
    }
    else if ((val1==LOW)&&(val2==LOW)&&(n>=10000)){
        flame();
        n=0;
    }
}

```

```

void loop(){
    if (Serial.available() > 0){dataIn = Serial.read();}
    if (dataIn == 'F'){maju();}
    if (dataIn == 'B'){mundur();}
    if (dataIn == 'L'){kiri();}
    if (dataIn == 'R'){kanan();}
    if (dataIn == 'S'){berhenti();}
    if (dataIn == '0'){vel = 0;}
    if (dataIn == '1'){vel = 25;}
    if (dataIn == '2'){vel = 50;}
    if (dataIn == '3'){vel = 75;}
    if (dataIn == '4'){vel = 100;}
    if (dataIn == '5'){vel = 125;}
    if (dataIn == '6'){vel = 150;}
    if (dataIn == '7'){vel = 175;}
    if (dataIn == '8'){vel = 200;}
    if (dataIn == '9'){vel = 200;}
    if (dataIn == 'q'){vel = 200;}
    if ((dataIn == 'x')&&(stat == 1)){stat = 0;} //x=120
    if (dataIn == 'X'){stat = 1;} //X=88
    if (stat == 1){otomatis ();}
}

```

Daftar Pustaka

1. A. P. Mathur, *Introduction to Microprocessors*, 24th Reprinted., McGraw-Hill Publishing Company Ltd., USA, 2006.
2. ar Publisher, 1998.
3. J. G. Keramas, *Robot Technology*, Delmar Publisher, 1998.
4. K. S. Fu, R. C. Gonzales, C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*, Mc Graw-Hill Book Company, US, 1987.
5. M. Margoilis, *Arduino Cookbook*, O'Reilly, USA 2011
6. M. McRoberts, *Beginning Arduino*, Apress, USA, 2010
7. M. Schmidt, *Arduino A Quick-Start Guide*, Pragmatic Programmers, USA, 2011.



MODUL PERKULIAHAN

Robotika

Perakitan Robot (Perangkat Lunak)

Fakultas
Fakultas Teknik

Program Studi
Teknik Elektro

Tatap Muka

14

Kode MK
MK14043

Disusun Oleh
Julpri Andika, ST., M.Sc.

Abstract

Hasil penelitian menunjukkan secanggih apapun sistem kontrol yang dipakai akan sangat tergantung kepada sensor atau transduser yang digunakan. Sensor dan transduser merupakan peralatan atau komponen yang mempunyai peranan penting dalam sebuah sistem pengaturan otomatis. Ketepatan dan kesesuaian dalam memilih sebuah sensor atau transduser akan sangat menentukan kinerja dari sistem pengaturan secara otomatis.

Kompetensi

Mahasiswa mengetahui tentang sensor dalam robotika, sensor pendeteksi garis dan rangkaian komparator

Program # 1 Gerak Servo dengan sudut 1 derajat

```
#include <Servo.h>
Servo myservo;
int pos;

void setup()
{
  myservo.attach(11);
}

void loop()
{
  for (pos=0; pos <= 180; pos +=20)
  {
    myservo.write(pos);
    delay(100);
  }
  delay(1000);
  for (pos = 180; pos >= 0; pos-=20)
  {
    myservo.write(pos);
    delay(100);
  }
  delay(1000);
}
```

Program 2 # Servo gerak 0 90 180

```
#include <Servo.h>
Servo myservo;

void setup()
{
  myservo.attach(11);
}

void loop()
{
  myservo.write(0);
  delay(1000);
  myservo.write(90);
  delay(1000);
  myservo.write(180);
  delay(1000);
}
```

Program 3 # Gerak Motor & Serial Motor

```
// Motor shield ini menggunakan pin 4 dan 5 untuk mengontrol 1 motor DC
// Sambungkan motor DC ke M1+, M1-
// Upload sketch ke Arduino
// Last Modified : 27 November 2012
// By ...
```

```
#define rightMotor_Dir 4 // arah putar (Motor 1 / kanan)
#define rightMotor_Vel 5 // kontrol kecepatan (Motor 1 / Kiri)
int val;

void setup()
```

```

{
  pinMode(4,OUTPUT); //Setting pin 4 sebagai output
  pinMode(5,OUTPUT); //Setting pin 5 sebagai output
  Serial.begin(9600);
}

//fungsi arah putar motor kanan
void rightMotor(boolean arah)
{
  digitalWrite(rightMotor_Vel,HIGH); //Motor diberi HIGH agar EN=full
  if(arah) digitalWrite (rightMotor_Dir,HIGH);
  else digitalWrite (rightMotor_Dir,LOW);
}

//fungsi berhenti
void berhenti ()
{
  digitalWrite(rightMotor_Vel,LOW); //Motor diberi HIGH agar EN=full speed
}

void loop()
{
  val = Serial.read ();
  if (val=='f')
  {
    rightMotor(true);
    delay(5000);
  }
  else if (val=='r')
  {
    rightMotor(false);
    delay(5000);
  }
  else

```

```
{  
  berhenti ();  
}  
}
```

Program 4 # Gerak Motor 2

//Tanggal 13 Desember 2016

```
#define rightMotor_Dir 4 // arah putar (Motor 1 / kanan)  
#define rightMotor_Vel 5 // kontrol kecepatan (Motor 1 / Kiri)  
#define leftMotor_Dir 7 // arah putar (Motor 2 / kanan)  
#define leftMotor_Vel 6 // kontrol kecepatan (Motor 2 / Kiri)  
  
void setup()  
{  
  pinMode(4,OUTPUT); //Setting pin 4 sebagai output  
  pinMode(5,OUTPUT); //Setting pin 5 sebagai output  
  pinMode(6,OUTPUT); //Setting pin 6 sebagai output  
  pinMode(7,OUTPUT); //Setting pin 7 sebagai output  
}  
  
//fungsi arah putar motor kanan  
void leftMotor(int pwm, boolean arah)  
{  
  analogWrite(rightMotor_Vel,pwm); //Motor diberi HIGH agar EN=full  
  if(arah) digitalWrite (rightMotor_Dir,HIGH);  
  else digitalWrite (rightMotor_Dir,LOW);  
}  
  
//fungsi arah putar motor kiri  
void rightMotor(int pwm, boolean arah)  
{  
  analogWrite(leftMotor_Vel,pwm); //Motor diberi HIGH agar EN=full
```

```
if(arah) digitalWrite (leftMotor_Dir,HIGH);  
else digitalWrite (leftMotor_Dir,LOW);  
}
```

```
//fungsi gerakan maju
```

```
void maju()  
{  
  rightMotor(200, false);  
  leftMotor(200, false);  
}
```

```
//fungsi gerakan mundur
```

```
void mundur()  
{  
  rightMotor(200, true);  
  leftMotor(200, true);  
}
```

```
//fungsi gerakan belok kanan
```

```
void kiri()  
{  
  rightMotor(200, false);  
  leftMotor(0, true);  
}
```

```
//fungsi gerakan belok kiri
```

```
void kanan()  
{  
  rightMotor(0, true);  
  leftMotor(200, false);  
}
```

```
//fungsi berhenti
```

```
void berhenti ()
```

```

{
  digitalWrite(rightMotor_Vel,LOW); //Motor diberi HIGH agar EN=full speed
  digitalWrite(leftMotor_Vel,LOW); //Motor diberi HIGH agar EN=full speed
}

void loop()
{
  maju();
  delay(7000);
  kiri();
  delay (750);
  maju();
  delay(7000);
  berhenti();
  while(1);
}

```

Program 5 # Servo & Sensor Flame

```

#include <Servo.h>
Servo myservo;
int pos;

void setup()
{
  Serial.begin(9600);
  myservo.attach(11);
}

void loop()
{
  for (pos=0; pos <= 180; pos +=1)

```

```

{
myservo.write(pos);
delay(50);
int sensorValue=analogRead(A5);
Serial.println(sensorValue, DEC);
delay (10);
}
delay(1000);
for (pos = 180; pos >= 0; pos-=1)
{
myservo.write(pos);
delay(50);
int sensorValue=analogRead(A5);
Serial.println(sensorValue, DEC);
delay (10);
}
delay(1000);
}

```

Program 6 # Motor & Limit Switch

```

#define rightMotor_Dir 4
#define rightMotor_Vel 5
#define leftMotor_Dir 7
#define leftMotor_Vel 6
#include <Servo.h>
Servo myservo;
int pos;

void setup()
{
myservo.attach(11);
Serial.begin(9600);
pinMode (4, OUTPUT);

```

```

pinMode (5, OUTPUT);
pinMode (6, OUTPUT);
pinMode (7, OUTPUT);
}

void stp()
{

}

//fungsi arah putar motor kanan
void rightMotor(int pwm,boolean arah)
{
  analogWrite(rightMotor_Vel, pwm);
  if(arah) digitalWrite(rightMotor_Dir, HIGH);
  else digitalWrite(rightMotor_Dir, LOW);
}

//fungsi arah putar motor kiri
void leftMotor (int pwm, boolean arah)
{
  analogWrite(leftMotor_Vel, pwm);
  if(arah) digitalWrite(leftMotor_Dir, HIGH);
  else digitalWrite(leftMotor_Dir, LOW);
}

//fungsi gerakan maju
void maju()
{
  rightMotor(200, true);
  leftMotor(200, true);
}

//fungsi gerakan mundur

```

```

void mundur()
{
  rightMotor(200, false);
  leftMotor(200, false);
}

//fungsi gerakan belok kanan
void kanan()
{
  rightMotor(0, false);
  leftMotor(200, true);
}

//fungsi gerakan belok kiri
void kiri()
{
  rightMotor(200, true);
  leftMotor(0, false);
}

void loop()
{
  int sensor_kanan = digitalRead(12);
  int sensor_kiri = digitalRead(13);

  if (sensor_kanan == HIGH)
  {
    mundur();
    delay (500);
    kiri();
    delay (500);
  }

  if (sensor_kiri == HIGH)

```

```
{  
  mundur();  
  delay(500);  
  kanan();  
  delay(500);  
}
```

```
if (sensor_kanan == HIGH && sensor_kiri == HIGH)
```

```
{  
  mundur();  
  delay(500);  
  kanan();  
  delay(1000);  
}
```

```
else
```

```
{  
  maju();  
}  
}
```

Daftar Pustaka

1. A. P. Mathur, *Introduction to Microprocessors*, 24th Reprinted., McGraw-Hill Publishing Company Ltd., USA, 2006.
2. ar Publisher, 1998.
3. J. G. Keramas, *Robot Technology*, Delmar Publisher, 1998.
4. K. S. Fu, R. C. Gonzales, C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*, Mc Graw-Hill Book Company, US, 1987.
5. M. Margoilis, *Arduino Cookbook*, O'Reilly, USA 2011
6. M. McRoberts, *Beginning Arduino*, Apress, USA, 2010
7. M. Schmidt, *Arduino A Quick-Start Guide*, Pragmatic Programmers, USA, 2011.