

Bab 2

LANDASAN TEORI

2.1 Internet of Things

Internet of Things (IoT) merupakan kumpulan benda-benda (*things*), berupa perangkat fisik (*hardware/embedded system*) yang mampu bertukar informasi antar sumber informasi, operator layanan ataupun perangkat lainnya yang terhubung kedalam sistem sehingga dapat memberikan kemanfaatan yang lebih besar. Perangkat fisik (*hardware/embedded system*) dalam infrastruktur *Internet of Things* merupakan *hardware* yang tertanam (*embedded*) dengan elektronik, perangkat lunak, sensor dan juga konektivitas.

Perangkat *embedded system* melakukan komputasi untuk pengolahan data dari input sensor dan beroperasi dalam infrastruktur internet. "A *Things*" pada *Internet of Things* dapat didefinisikan sebagai subjek misalkan orang dengan implant jantung, hewan peternakan dengan transponder biochip, sebuah mobil yang telah dilengkapi *built-in sensor* untuk memperingatkan pengemudi ketika tekanan ban rendah. Sejauh ini, IoT paling erat hubungannya dengan komunikasi *machine-to-machine (M2M)* di bidang manufaktur dan listrik, perminyakan, dan gas. Produk dibangun dengan kemampuan komunikasi M2M yang sering disebut dengan sistem cerdas atau "smart". (contoh: *smart label, smart meter, smart grid sensor*).

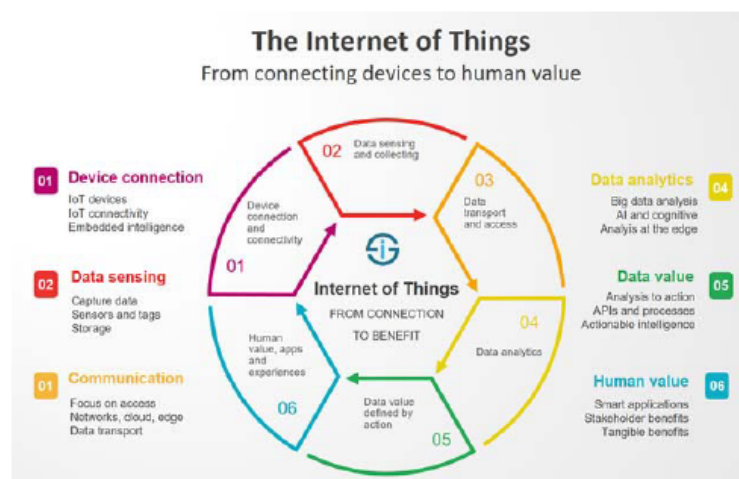
Menurut *Coordinator and support action for global RFID-related activities and standadisation* menyatakan *Internet of Things (IoT)* sebagai sebuah infrastruktur koneksi jaringan global, yang mengkoneksikan benda fisik dan virtual melalui eksploitasi *data capture* dan teknologi komunikasi. Infrastruktur IoT terdiri dari jaringan yang telah ada dan internet berikut pengembangannya. Hal ini menawarkan identifikasi obyek, identifikasi sensor dan kemampuan koneksi yang menjadi dasar untuk pengembangan layanan dan aplikasi koperatif yang berdiri secara independen, juga ditandai dengan tingkat otonomi *data capture* yang tinggi,

event transfer, konektivitas pada jaringan dan juga interoperabilitas.

Menurut IEEE(*Institute of Electrical and Electronics Engineers*) *Internet of Things (IoT)* didefinisikan sebagai sebuah jaringan dengan masing-masing benda yang tertanam dengan sensor yang terhubung kedalam jaringan internet. (IEEE "*Internet of things*" 2014)

Untuk membangun sistem *Internet of Things* membutuhkan komponen yaitu *device connection* dan *data sensing*. Selain komponen untuk membangun sistem IoT kemampuan berkomunikasi antara sistem juga dibutuhkan dalam IoT. Untuk menyimpan serta melakukan *data analytics* dari data hasil akusisi *data Sensing* digunakan server database. Komponen terakhir adalah pemanfaatan dari komunikasi yang dijalankan terus menerus antara *device connection* dengan *data sensing* yang mampu menyimpan serta melakukan data analytics dan digunakan untuk membantu manusia dalam hal tertentu.

"*Internet of Things*" dikenalkan pertama kali oleh visioner Inggris yaitu Kevin Ashton, pada tahun 1999. IoT merupakan teknologi yang diharapkan mampu menawarkan perangkat sistem canggih dengan kemampuan konektivitas, sehingga mampu melakukan komunikasi mesin ke mesin (M2M) dan mencakup berbagai protokol, domain, dan aplikasi. Interkoneksi pada perangkat ini tertanam (*embedded*) diharapkan untuk mengantarkan otomatisasi dalam hampir semua bidang.



Gambar 2.1: *Internet of Things*



Gambar 2.2: konsep *Internet of Things*

Konsep *Internet of Things* mencakup 3 elemen utama yaitu: benda fisik atau nyata yang telah diintegrasikan pada modul sensor, koneksi internet, dan pusat data pada server untuk menyimpan data ataupun informasi dari aplikasi. Penggunaan benda yang terkoneksi ke internet akan menghimpun data yang kemudian terkumpul menjadi *big data* untuk kemudian diolah, dianalisa baik oleh instansi pemerintah, perusahaan terkait, maupun instansi lain kemudian di manfaatkan bagi kepentingan masing-masing.

2.2 Cloud Computing

Komputasi awan (Bahasa Inggris: *cloud computing*) adalah gabungan pemanfaatan teknologi komputer ('komputasi') dan pengembangan berbasis Internet ('awan'). Awan (*cloud*) adalah metafora dari internet, sebagaimana awan yang sering digambarkan di diagram jaringan komputer. Sebagaimana awan dalam diagram jaringan komputer tersebut, awan (*cloud*) dalam *Cloud Computing* juga merupakan abstraksi dari infrastruktur kompleks yang disembunyikannya. Ia adalah suatu metoda komputasi di mana kapabilitas terkait teknologi informasi disajikan sebagai suatu layanan (*as a service*), sehingga pengguna dapat mengaksesnya lewat Internet ("di dalam awan") tanpa mengetahui apa yang ada didalamnya, ahli dengannya, atau memiliki kendali terhadap infrastruktur teknologi yang membanturnya. Menurut sebuah makalah tahun 2008 yang dipublikasi *IEEE Internet Computing* "*cloud computing* adalah suatu paradigma di mana informasi secara permanen tersimpan di server di internet dan tersimpan secara sementara di komputer pengguna (*client*) termasuk di dalamnya adalah desktop, komputer tablet, notebook, komputer tembok, handheld, sensor-sensor, monitor dan lain-lain."

Komputasi awan adalah suatu konsep umum yang mencakup SaaS, Web 2.0, dan tren teknologi terbaru lain yang dikenal luas, dengan tema umum berupa

ketergantungan terhadap internet untuk memberikan kebutuhan komputasi pengguna. Sebagai contoh, Google Apps menyediakan aplikasi bisnis umum secara daring yang diakses melalui suatu penjelajah web dengan perangkat lunak dan data yang tersimpan di server. Komputasi awan saat ini merupakan trend teknologi terbaru, dan contoh bentuk pengembangan dari teknologi *cloud Computing* ini adalah *iCloud*.

Cloud computing merupakan sebuah istilah baru dalam dunia teknologi informasi, *cloud* merupakan sebuah metode komputasi yang mengandalkan teknologi internet, *cloud* yang dalam bahasa Indonesia diartikan sebagai awan yang artinya dalam teknologi informasi merupakan lambang internet, dimana sumber daya bersama, perangkat lunak, dan informasi disediakan untuk komputer dan perangkat lain sesuai permintaan, seperti jaringan listrik. *cloud computing* adalah pergeseran paradigma mengikuti pergeseran dari mainframe ke *client server* pada awal tahun 1980. Jika *cloud* di implementasikan pada instansi pemerintahan akan mampu memberikan kemudahan dalam proses pelayanan dikarenakan dengan *cloud* akan mensentralisasikan semua data. Konsep penerapan *cloud* dapat mengikuti pola teknologi dan system informasi yang ada pada perbankan. Namun tidak semua layanan yang ada di internet masuk kedalam kategori *cloud computing*. **Hariguna dan Berililana (2011)**

Layanan yang ditawarkan *cloud computing* *Cloud computing* dapat diimplementasikan dengan cara menyediakan komponen-komponen berupa server, *hardware*, dan jaringan yang dibutuhkan. Pengguna *cloud computing* dapat melakukan instalasi aplikasi yang digunakannya pada infrastruktur tersebut. Pengguna *cloud computing* juga dapat memilih bagaimana menggunakan layanan *cloud computing* yang ditawarkan vendor sesuai kebutuhan.

Layanan-layanan yang bisa dipilih meliputi:

1. *Infrastructure as a service*

Layanan ini diberikan dengan cara menyediakan komponen-komponen berupa server, *hardware*, dan jaringan yang dibutuhkan dengan harga tertentu. Pengguna *cloud computing* dapat melakukan instalasi aplikasi yang digunakannya pada infrastruktur tersebut.

- Contoh terapan: hosting aplikasi web
- Contoh vendor penyedia jasa: semua penyedia hosting.

2. *Platform as a service*

Layanan yang menyediakan *software system* dan *software* pendukung yang diperlukan untuk membangun aplikasi yang akan dipasang pada server tersebut sesuai kebutuhan organisasi/instansi. Organisasi/instansi kemudian membangun aplikasi yang dibutuhkan pada platform ini dan menggunakannya. Analogi yang dapat digunakan untuk menggambarkan layanan ini adalah seperti menyewa rumah lengkap dengan isinya sehingga pengguna dapat langsung menggunakan rumah tersebut.

- Contoh terapan: *remote application development*.
- Contoh vendor: BizNet, Microsoft, LintasArta.

3. *Software as a service*

Layanan yang diberikan dengan menyediakan software maupun aplikasi yang dapat diakses pelanggan via internet. Penyedia layanan *cloud computing* berinteraksi dengan

pengguna dan pelanggan melalui sebuah *front-end* panel.

- Contoh layanan sederhana: e-mail, online documents.
- Contoh layanan (agak susah): SAP online Contoh vendor: Google, Amazon, SAP, dan lain-lain.

National Institute of Standards and Technology (NIST), Information Technology Laboratory memberikan dua buah catatan mengenai pengertian komputasi awan. Pertama, komputasi awan masih merupakan paradigma yang berkembang. Definisi, kasus penggunaan, teknologi yang mendasari, masalah, risiko, dan manfaat akan terus disempurnakan melalui perdebatan baik oleh sektor publik maupun swasta. Definisi, atribut, dan karakteristik akan berkembang dan berubah dari waktu ke waktu. Kedua, industri komputasi awan merupakan ekosistem besar dengan banyak model, vendor, dan pangsa pasar. NIST mengidentifikasi lima karakteristik penting dari komputasi awan (Mell & Grance, 2009) sebagai berikut:

1. *On-demand self-service*. Pengguna dapat memesan dan mengelola layanan tanpa interaksi manusia dengan penyedia layanan, misalnya dengan menggunakan, sebuah portal web dan manajemen antarmuka. Pengadaan dan perlengkapan layanan serta sumberdaya yang terkait terjadi secara otomatis pada penyedia.

2. *Broad network access*. Kemampuan yang tersedia melalui jaringan dan diakses melalui mekanisme standar, yang mengenalkan penggunaan berbagai platform (misalnya, telepon selular, laptop, dan PDA).
3. *Resource pooling*. Penyatuan sumberdaya komputasi yang dimiliki penyedia untuk melayani beberapa konsumen menggunakan model multipenyewa, dengan sumberdaya fisik dan virtual yang berbeda, ditetapkan secara dinamis dan ditugaskan sesuai dengan permintaan konsumen. Ada rasa kemandirian lokasi bahwa pelanggan umumnya tidak memiliki kontrol atau pengetahuan atas keberadaan lokasi sumberdaya yang disediakan, tetapi ada kemungkinan dapat menentukan lokasi di tingkat yang lebih tinggi (misalnya, negara, negara bagian, atau *data center*). Contoh sumberdaya termasuk penyimpanan, pemrosesan, memori, bandwidth jaringan, dan mesin virtual.
4. *Rapid elasticity*. Kemampuan dapat dengan cepat dan elastis ditetapkan.
5. *Measured Service*. Sistem komputasi awan secara otomatis mengawasi dan mengoptimalkan penggunaan sumberdaya dengan memanfaatkan kemampuan pengukuran (*metering*) pada beberapa tingkat yang sesuai dengan jenis layanan (misalnya, penyimpanan, pemrosesan, bandwidth, dan *account* pengguna aktif). Penggunaan sumberdaya dapat dipantau, dikendalikan, dan dilaporkan sebagai upaya memberikan transparansi bagi penyedia dan konsumen dari layanan yang digunakan.

Sedangkan tiga jenis model layanan dijelaskan oleh NIST (Mell dan Grance, 2009) sebagai berikut :

1. *Cloud Software as a Service (SaaS)*. Kemampuan yang diberikan kepada konsumen untuk menggunakan aplikasi penyedia dapat beroperasi pada infrastruktur awan. Aplikasi dapat diakses dari berbagai perangkat klien melalui antarmuka seperti web browser (misalnya, email berbasis web). Konsumen tidak mengelola atau mengendalikan infrastruktur awan yang mendasari termasuk jaringan, server, sistem operasi, penyimpanan, atau bahkan kemampuan aplikasi individu, dengan kemungkinan pengecualian terbatas terhadap pengaturan konfigurasi aplikasi pengguna tertentu.
2. *Cloud Platform as a Service (PaaS)*. Kemampuan yang diberikan kepada konsumen untuk menyebarkan aplikasi yang di buat konsumen atau diperoleh ke infrastruktur komputasi awan menggunakan bahasa pemrograman dan peralatan yang didukung oleh provider. Konsumen tidak mengelola atau mengendalikan infrastruktur awan yang mendasari termasuk jaringan, server, sistem

operasi, atau penyimpanan, namun memiliki kontrol atas aplikasi disebarkan dan memungkinkan aplikasi melakukan hosting konfigurasi.

3. *Cloud Infrastructure as a Service (IaaS)*. Kemampuan yang diberikan kepada konsumen untuk memproses, menyimpan, berjaringan, dan komputasi sumberdaya lain yang penting, dimana konsumen dapat menyebarkan dan menjalankan perangkat lunak secara bebas, dapat mencakup sistem operasi dan aplikasi. Konsumen tidak mengelola atau mengendalikan infrastruktur awan yang mendasari tetapi memiliki kontrol atas sistem operasi, penyimpanan, aplikasi yang disebarkan, dan mungkin kontrol terbatas komponen jaringan yang pilih (misalnya, *firewall host*)

Model penyebaran komputasi awan menurut NIST terdiri dari empat model (Mell dan Grance, 2009), yaitu:

1. *Private cloud*. Swasta awan. Infrastruktur awan yang semata-mata dioperasikan bagi suatu organisasi. Ini mungkin dikelola oleh organisasi atau pihak ketiga dan mungkin ada pada on premis atau off premis.
2. *Community cloud*. Masyarakat awan. Infrastruktur awan digunakan secara bersama oleh beberapa organisasi dan mendukung komunitas tertentu yang telah berbagi concerns (misalnya, misi, persyaratan keamanan, kebijakan, dan pertimbangan kepatuhan). Ini mungkin dikelola oleh organisasi atau pihak ketiga dan mungkin ada pada on premis atau off premis.
3. *Public cloud*. Infrastruktur awan yang dibuat tersedia untuk umum atau kelompok industri besar dan dimiliki oleh sebuah organisasi yang menjual layanan awan.
4. *Hybrid cloud*. Hybrid awan. Infrastruktur awan merupakan komposisi dari dua atau lebih awan (swasta, komunitas, atau publik) yang masih entitas unik namun terikat bersama oleh standar atau kepemilikan teknologi yang menggunakan data dan portabilitas aplikasi (*e.g., cloud bursting for load balancing between clouds*).

2.3 Blynk

Blynk adalah platform aplikasi yang dapat diunduh secara gratis untuk iOS dan Android yang berfungsi mengontrol Arduino, Raspberry Pi dan sejenisnya melalui internet. *Blynk* dirancang untuk *Internet of Things* dengan tujuan dapat mengontrol *hardware* dari jarak jauh, dapat menampilkan data sensor, dapat menyimpan data, visual dan melakukan banyak hal canggih lainnya. Ada tiga komponen utama dalam platform yaitu *Blynk App*, *Blynk Server*, dan *Blynk Library*.



Gambar 2.3: Ilustrasi *Blynk*

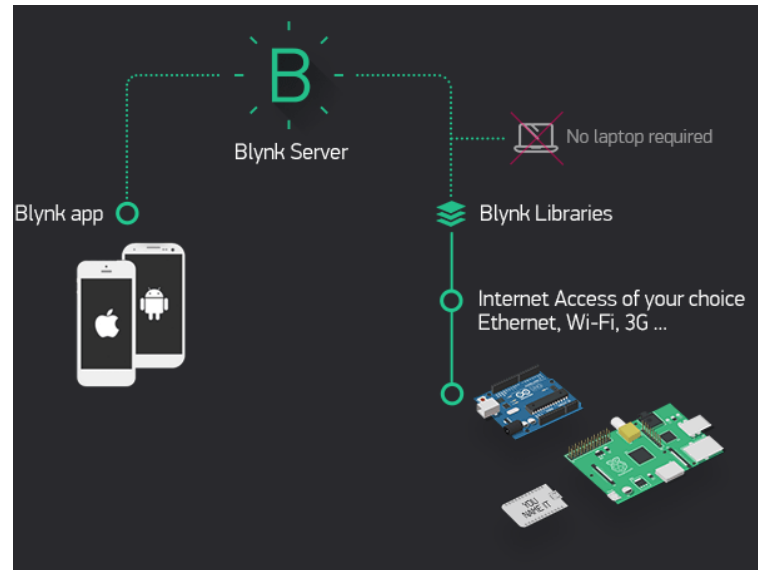
2.3.1 Cara kerja dari *cloud blynk*

Blynk di rancang untuk IoT *Internet of Things*. Dapat mengendalikan suatu hardware atau mikrokontroler yang berbasis IoT (*Internet of Things*), aplikasi ini dapat menampilkan data sensor, menyimpan data, dan lain-lain.

ada 3 hal utama yang ada pada *blynk*, diantaranya ialah:

- *Blynk App*. Aplikasi ini yang dapat di gunakan untuk membuat sebuah remote control yang berbasis IoT (*Internet of Things*).
- *Blynk Server*. Yang bertugas untuk komunikasi antara perangkat *smartphone* dengan mikrokontroler yang berbasis IoT (*Internet of Things*). Ini bersifat *open source*, dapat menangani ribuan perangkat dan bahkan dapat diluncurkan pada Raspberry Pi.
- *Blynk libraries*. Untuk mengaktifkan komunikasi dengan server, dan semua proses perintah masuk dan keluar.

setelah membuat *project* pada *blynk* aplikasi, maka ketika menekan sebuah tombol pada aplikasi *blynk* maka pesan tersebut berpindah atau terkirim ke *cloud blynk*, setelah berpindah atau di kirim ke *cloud blynk* maka pesan tersebut terkirim lagi ke sebuah mikrokontroler yang berbasis IoT (*Internet of Things*).



Gambar 2.4: cara kerja dari *cloud blynk*

2.4 Sistem Kendali dengan *NodeMCU*

NodeMCU adalah sebuah platform IoT yang bersifat *opensource*. Terdiri dari perangkat keras berupa *System On Chip* ESP8266 dari ESP8266 buatan **Espressif System**, juga *firmware* yang digunakan, yang menggunakan bahasa pemrograman *scripting Lua*. Istilah NodeMCU secara *default* sebenarnya mengacu pada *firmware* yang digunakan daripada perangkat keras *development kit*.

NodeMCU bisa dianalogikan sebagai *board* arduino-nya ESP8266. Dalam seri tutorial ESP8266 **embeddednesia** pernah membahas bagaimana memprogram ESP8266 sedikit merepotkan karena diperlukan beberapa teknik *wiring* serta tambahan modul *USB to serial* untuk mengunduh program. Namun NodeMCU telah *me-package* ESP8266 ke dalam sebuah *board* yang kompak dengan berbagai fitur layaknya mikrokontroler + kapabilitas akses terhadap Wifi juga chip komunikasi *USB to serial*. Sehingga untuk memprogramnya hanya diperlukan ekstensi kabel data USB persis yang digunakan sebagai kabel data dan kabel *charging smartphone Android*.

2.4.1 Spesifikasi Sejarah NodeMCU

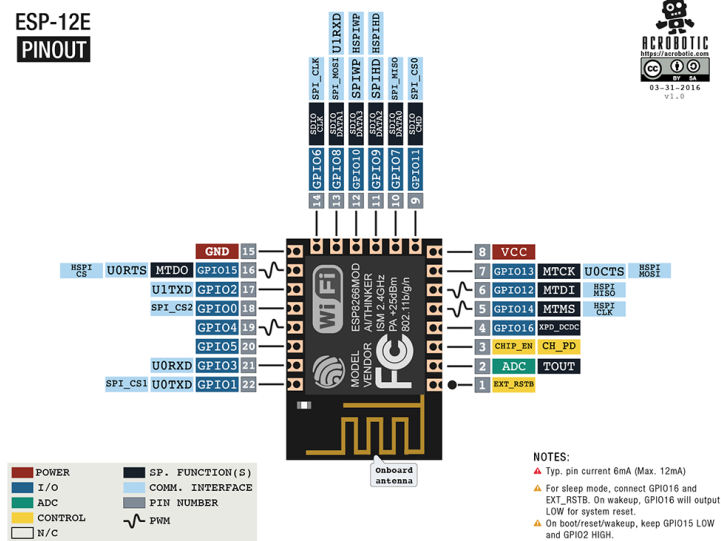
Sejarah lahirnya NodeMCU berdekatan dengan rilis ESP8266 pada 30 Desember 2013, *Espressif Systems* selaku pembuat ESP8266 memulai produksi ESP8266 yang merupakan SoC Wi-Fi yang terintegrasi dengan prosesor Tensilica Xtensa LX106. Sedangkan NodeMCU dimulai pada 13 Oktober 2014 saat Hong me-commit file pertama nodemcu-firmware ke Github. Dua bulan kemudian project tersebut dikembangkan ke platform perangkat keras ketika Huang R meng-commit file dari board ESP8266, yang diberi nama devkit v.0.9.

Berikutnya, di bulan yang sama. Tuan PM memporing pustaka *client* MQTT dari Contiki ke platform SOC ESP8266 dan di-commit ke *project* NodeMCU yang membuatnya mendukung protokol IOT MQTT melalui Lua. Pemutakhiran penting berikutnya terjadi pada 30 Januari 2015 ketika Devsaurus memporing u8glib ke *project* NodeMCU yang memungkinkan NodeMCU bisa mendrive display LCD, OLED, hingga VGA. Demikianlah, project NodeMCU terus berkembang hingga kini berkat komunitas *open source* dibaliknya, pada musim panas 2016 NodeMCU sudah terdiri memiliki 40 modul fungsionalitas yang bisa digunakan sesuai kebutuhan developer.

2.4.2 Spesifikasi ESP-12E

Karena jantung dari NodeMCU adalah ESP8266 (khususnya seri ESP-12, termasuk ESP-12E) maka fitur – fitur yang dimiliki NodeMCU akan kurang lebih sama ESP-12 (juga ESP-12E untuk NodeMCU v.2 dan v.3) kecuali NodeMCU telah dibungkus oleh API sendiri yang dibangun berdasarkan bahasa pemrograman eLua, yang kurang lebih cukup mirip dengan javascript. Beberapa fitur tersebut antara lain:

1. 10 Port GPIO dari D0 – D10.
2. Fungsionalitas PWM.
3. Antarmuka I2C dan SPI
4. Antarmuka 1 *Wire*
5. ADC

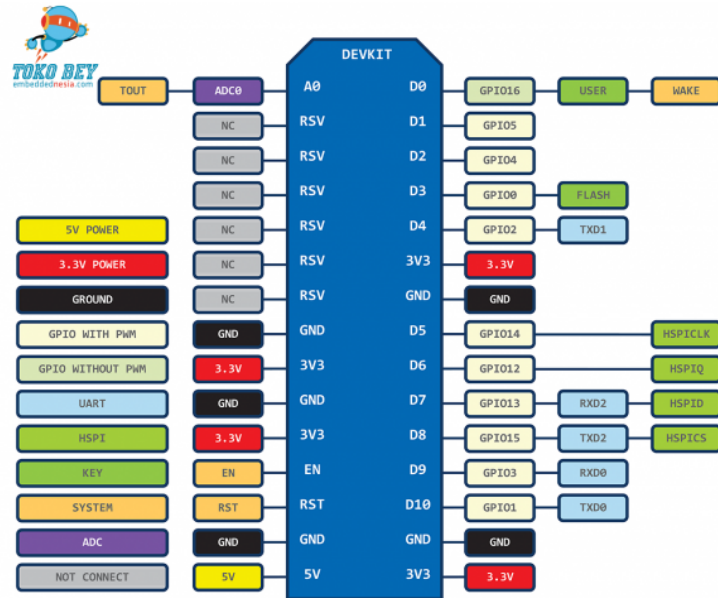
Gambar 2.5: Pin-pin dari *ESP-12E*

1. RST : Berfungsi mereset modul
2. ADC : *Analog Digital Converter*. Rentang tegangan masukan 0-1v, dengan skup nilai digital 0-1024
3. EN : *Chip Enable, Active High*
4. IO16 : GPIO16, dapat digunakan untuk membangunkan chipset dari *mode deep sleep*
5. IO14 : GPIO14, HSPI CLK
6. IO12 : GPIO12, HSPI MISO
7. IO13 : GPIO13, HSPI MOSI, UART0 CTS
8. VCC : Catu daya 3.3V (VDD)
9. CS0 : *Chip selection*
10. MISO : *Slave output, Main input*
11. IO9 : GPIO9
12. IO10 : GPIO10
13. MOSI : *Main output slave input*

- 14. SCLK : *Clock*
- 15. GND : *Ground*
- 16. IO15 : GPIO15, MTDO, HSPICS, UART0 RTS
- 17. IO2 : GPIO2, UART1 TXD
- 18. IO0 : GPIO0
- 19. IO4 : GPIO4
- 20. IO5 : GPIO5
- 21. RXD : UART0 RXD, GPIO3
- 22. TXD : UART0 TXD, GPIO1

2.4.3 Spesifikasi Tegangan Kerja

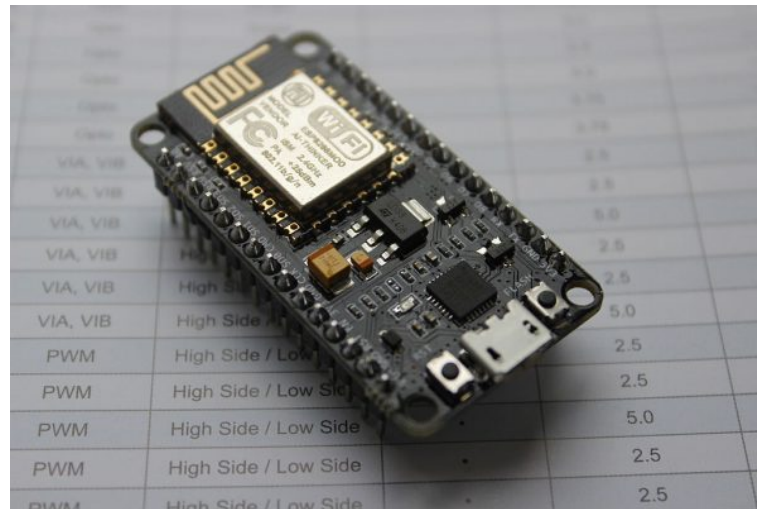
ESP8266 menggunakan standar tegangan JEDEC (tegangan 3.3V) untuk bisa berfungsi. Tidak seperti mikrokontroler AVR dan sebagian besar board Arduino yang memiliki tegangan TTL 5 volt. Meskipun begitu, NodeMCU masih bisa terhubung dengan 5V namun melalui port micro USB atau pin Vin yang disediakan oleh board-nya. Namun karena semua pin pada ESP8266 tidak toleran terhadap masukan 5V. Maka jangan sekali – kali langsung mencatunya dengan tegangan TTL jika tidak ingin merusak *board* anda. Anda bisa menggunakan ***Level Logic Converter*** untuk mengubah tegangan ke nilai aman 3.3v.



Gambar 2.7: Skematik posisi pin nodemcu devkit v.1

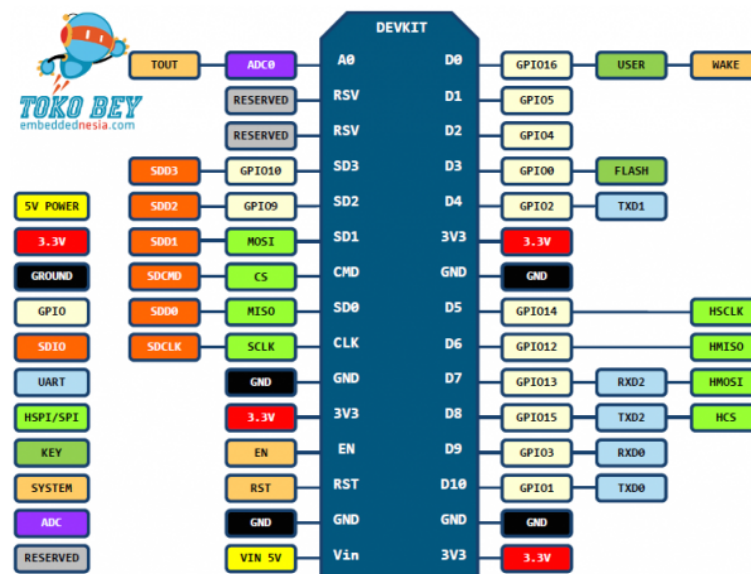
Namun beberapa produk juga ada yang menggunakan chip ESP-12E sebagai inti dari board v.0.9 dengan tampilan board berubah menjadi hitam.

2.4.6 Spesifikasi Generasi kedua / board v 1.0 (biasa disebut V2)



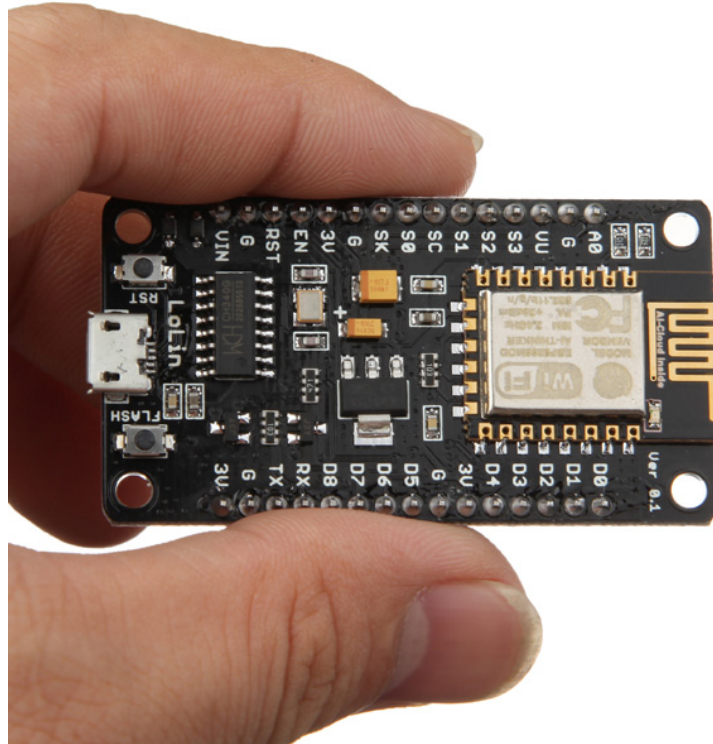
Gambar 2.8: NodeMCU Devkit v.1.0

Generasi kedua adalah pengembangan dari versi sebelumnya, dengan chip yang ditingkatkan dari sebelumnya ESP12 menjadi ESP12E. Dan IC Serial diubah dari CHG340 menjadi CP2102.



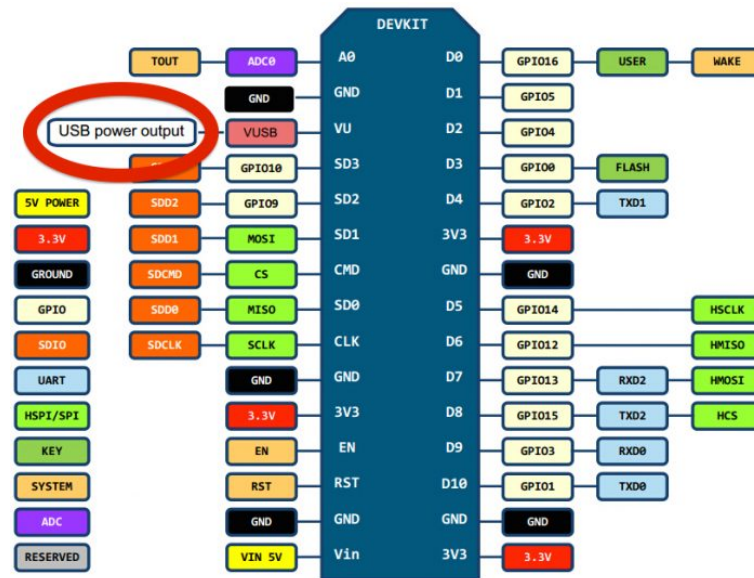
Gambar 2.9: Skematik posisi pin nodemcu devkit v.2

2.4.7 Spesifikasi Generasi ketiga / board v 1.0 (biasa disebut V3 Lolin)



Gambar 2.10: NodeMCU Devkit v.3.0 LoLin

Sedangkan untuk V3 sebenarnya bukanlah versi resmi yang dirilis oleh NodeMCU. Setidaknya sampai posting ini dibuat, belum ada versi resmi untuk V3 NodeMCU. V3 hanyalah versi yang diciptakan oleh produsen LoLin dengan perbaikan minor terhadap V2. Diklaim memiliki antarmuka USB yang lebih cepat.



Gambar 2.11: Skematik posisi pin nodemcu devkit v.3

Jika anda bandingkan dengan versi sebelumnya, dimensi dari board V3. Akan lebih besar dibanding V2. Lolin menggunakan 2 pin cadangan untuk daya USB dan yang lain untuk GND tambahan.

2.5 DC Motor

Motor dc adalah jenis motor listrik yang bekerja menggunakan sumber tegangan dc. Motor dc atau motor arus searah sebagaimana namanya, menggunakan arus langsung dan tidak langsung/*direct-unidirectional*. Motor dc digunakan pada penggunaan khusus dimana diperlukan penyalan torque yang tinggi atau percepatan yang tetap untuk kisaran kecepatan yang luas.



Gambar 2.12: DC Motor

2.5.1 Komponen Utama Motor DC

Gambar di atas memperlihatkan sebuah motor dc yang memiliki tiga komponen utama :

- Kutub medan magnet

Secara sederhana digambarkan bahwa interaksi dua kutub magnet akan menyebabkan perputaran pada motor dc. Motor dc memiliki kutub medan yang stasioner dan kumparan motor dc 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.

- Kumparan motor dc

Bila arus masuk menuju kumparan motor dc, maka arus ini akan menjadi elektromagnet. Kumparan motor dc yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakkan beban. Untuk kasus motor dc yang kecil, kumparan motor dc 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 kumparan motor dc.

- *Commutator dc motor*

Komponen ini terutama ditemukan dalam motor dc. Kegunaannya adalah untuk membalikan arah arus listrik dalam kumparan motor dc. Commutator juga membantu dalam transmisi arus antara kumparan motor dc dan sumber daya.

2.5.2 Kelebihan Motor DC

Keuntungan utama motor dc adalah dalam hal pengendalian kecepatan motor dc tersebut, yang tidak mempengaruhi kualitas pasokan daya. Motor ini dapat dikendalikan dengan mengatur :

- Tegangan kumparan motor dc – meningkatkan tegangan kumparan motor dc 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. Hubungan antara kecepatan, flux medan dan tegangan kumparan motor dc ditunjukkan dalam persamaan berikut :

Gaya elektromagnetik : $E = K \phi N$

Torque : $T = K \phi I_a$

Dimana:

E = gaya elektromagnetik yang dikembangkan pada terminal kumparan motor DC (volt).

ϕ = flux medan yang berbanding lurus dengan arus medan.

N = kecepatan dalam RPM (putaran per menit).

T = *torque electromagnetic*.

I_a = arus kumparan motor dc.

K = konstanta persamaan.

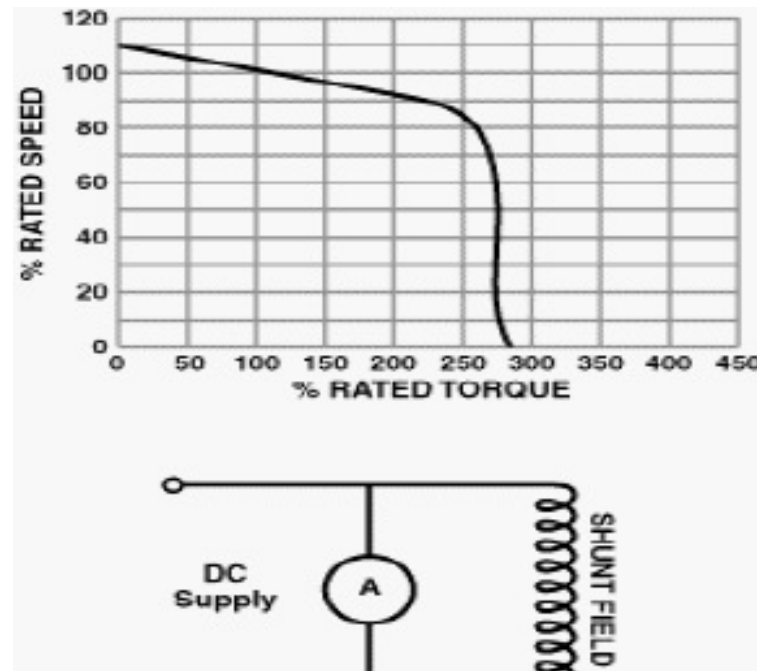
2.5.3 Jenis-Jenis Motor DC

1. Motor DC sumber daya terpisah/ *Separately Excited*

Jika arus medan di pasok dari sumber terpisah maka disebut motor dc sumber daya terpisah / *separately excited*.

2. Motor DC sumber daya sendiri/ *Self Excited: shunt motor*

Pada *motor shunt*, gulungan medan (medan shunt) disambungkan secara paralel dengan gulungan kumparan motor dc (A) seperti diperlihatkan dalam gambar dibawah. Oleh karena itu total arus dalam jalur merupakan penjumlahan arus medan dan arus kumparan motor dc.



Gambar 2.13: Karakteristik Motor DC Shunt

Berikut tentang kecepatan motor shunt (E.T.E., 1997):

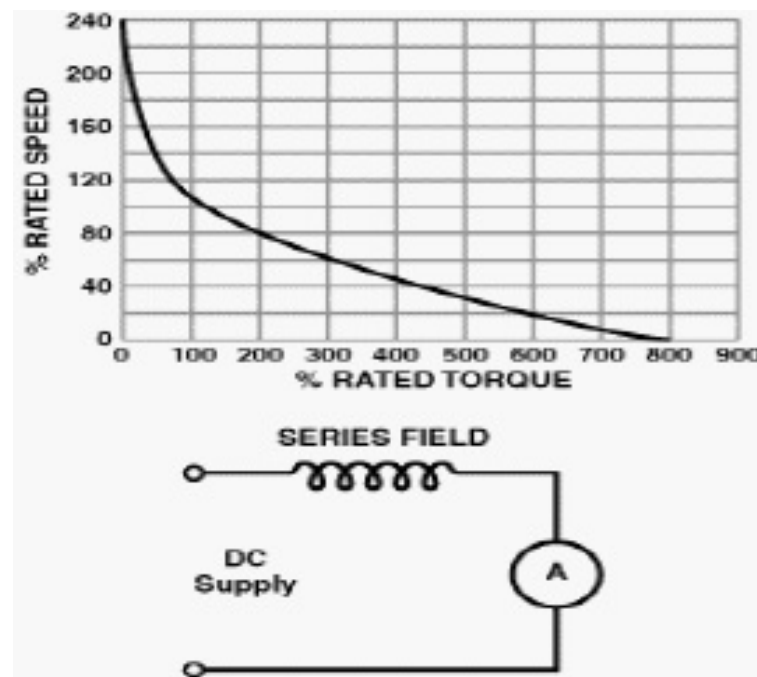
- Kecepatan pada prakteknya konstan tidak tergantung pada beban (hingga torque tertentu setelah kecepatannya berkurang, lihat Gambar diatas dan oleh karena itu cocok untuk penggunaan komersial dengan beban awal yang rendah, seperti peralatan mesin.
- Kecepatan dapat dikendalikan dengan cara memasang tahanan dalam susunan seri dengan kumparan motor DC (kecepatan berkurang) atau dengan memasang tahanan pada arus medan (kecepatan bertambah).

3. Motor DC daya sendiri: motor seri

Dalam motor seri, gulungan medan (medan shunt) dihubungkan secara seri dengan gulungan kumparan motor DC (A) seperti ditunjukkan dalam gambar dibawah. Oleh karena itu, arus medan sama dengan arus kumparan motor DC. Berikut tentang kecepatan motor seri (*Rodwell International Corporation, 1997; L.M. Photonics Ltd, 2002*) :

- Kecepatan dibatasi pada 5000 RPM.
- Harus dihindarkan menjalankan motor seri tanpa ada beban sebab motor akan mempercepat tanpa terkendali.

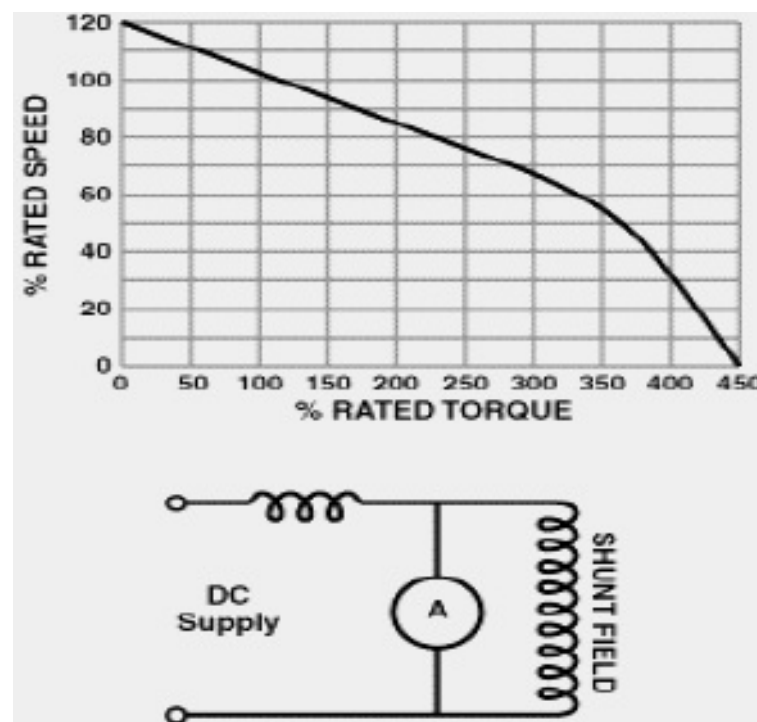
Motor-motor seri cocok untuk penggunaan yang memerlukan torque penyalan awal yang tinggi, seperti derek dan alat pengangkat hoist seperti pada gambar berikut.



Gambar 2.14: Karakteristik motor dc Seri

4. Motor DC Kompon/Gabungan

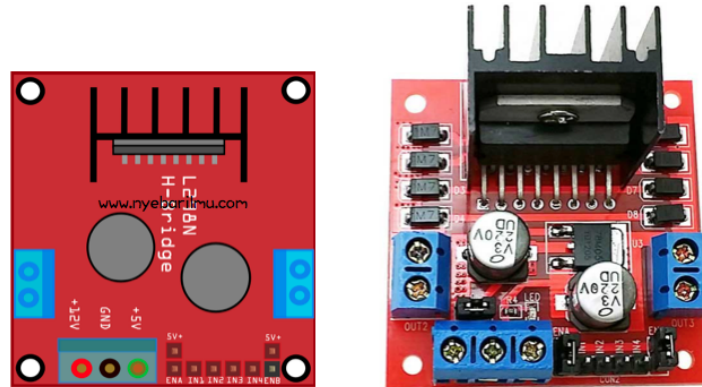
Motor Kompon DC merupakan gabungan motor seri dan shunt. Pada motor kompon, gulungan medan (medan shunt) dihubungkan secara paralel dan seri dengan gulungan kumparan motor DC (A) seperti yang ditunjukkan dalam gambar dibawah. Sehingga, motor kompon memiliki torque penyalan awal yang bagus dan kecepatan yang stabil. Makin tinggi persentase penggabungan (yakni persentase gulungan medan yang dihubungkan secara seri), makin tinggi pula torque penyalan awal yang dapat ditangani oleh motor ini. Contoh, penggabungan 40-50 % menjadikan motor ini cocok untuk alat pengangkat hoist dan derek, sedangkan motor kompon yang standar (12 %) tidak cocok (*myElectrical*, 2005).



Gambar 2.15: Karakteristik Motor DC Kompon

2.6 Motor Driver L298N

Driver motor L298N merupakan module driver motor DC yang paling banyak digunakan atau dipakai di dunia elektronika yang difungsikan untuk mengontrol kecepatan serta arah perputaran motor DC.



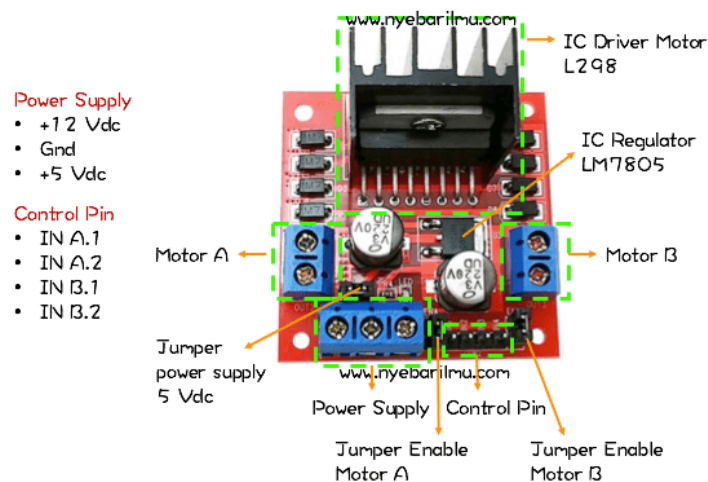
Gambar 2.16: Skematik L298N dan bentuk fisik L298N

IC L298 merupakan sebuah IC tipe H-bridge yang mampu mengendalikan beban-beban induktif seperti relay, solenoid, motor DC dan motor stepper.

Pada IC L298 terdiri dari transistor-transistor logik (TTL) dengan gerbang nand yang berfungsi untuk memudahkan dalam menentukan arah putaran suatu motor dc maupun motor stepper.

Untuk dipasaran sudah terdapat modul driver motor menggunakan ic l298 ini, sehingga lebih praktis dalam penggunaannya karena pin I/O nya sudah terpackage dengan rapi dan mudah digunakan.

Kelebihan akan modul driver motor L298N ini yaitu dalam hal kepresisian dalam mengontrol motor sehingga motor lebih mudah untuk dikontrol.



Gambar 2.17: *Pin out* dari driver motor L298N

Keterangan :

- *Enable A* : Berfungsi untuk mengaktifkan bagian *output* motor A
- *Enable B* : Berfungsi untuk mengaktifkan bagian *output* motor B
- *Jumper 5vdc* : Sebagai mode pemilihan sumber tegangan 5V-DC, jika tidak dijumpa maka akan ke mode sumber tegangan 12V-DC
- *Control Pin* : Sebagai kendali perputaran dan kecepatan motor yang dihubungkan ke mikrokontroler

Spesifikasi dari Modul Driver Motor L298N

- Menggunakan IC L298N (*Double H bridge Drive Chip*)
- Tegangan minimal untuk masukan power antara 5V-35V
- Tegangan operasional : 5V
- Arus untuk masukan antara 0-36mA
- Arus maksimal untuk keluaran per Output A maupun B yaitu 2A
- Daya maksimal yaitu 25W
- Dimensi modul yaitu 43 x 43 x 26mm
- Berat : 26g

2.7 Motor Servo



Gambar 2.18: Motor Servo

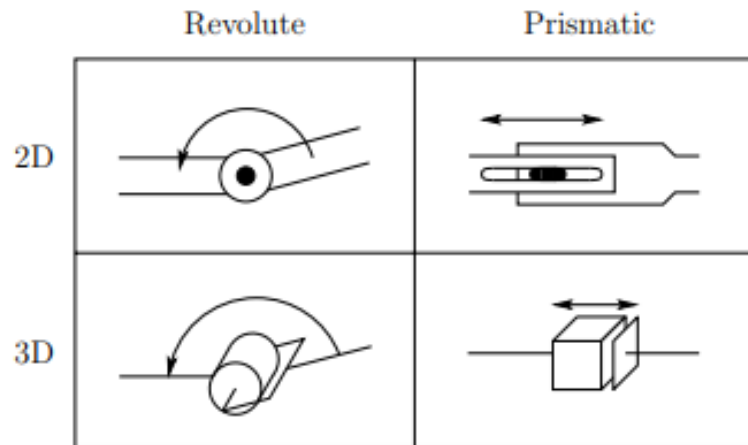
Motor servo adalah sebuah motor dengan sistem *closed feedback* di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor, serangkaian gear, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan 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 bagian bagian 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 2.18.

2.8 Komponen dan Struktur Robot

Robot Manipulator terdiri dari tautan yang dihubungkan oleh sambungan ke dalam rantai kinematik . Sambungan biasanya berputar (*revolute*) atau linier (prismatik). Sendi *revolute* seperti engsel dan memungkinkan rotasi relatif antara dua tautan. Sambungan prismatik memungkinkan kerabat linier gerakan antara dua tautan.



Gambar 2.19: Representasi simbolik sendi robot

Setiap sambungan mewakili interkoneksi antara dua tautan, katakan

$$l_i$$

dan

$$l_{i+1}$$

. Menunjukkan sumbu rotasi sendi *revolute*, atau sumbu sepanjang mana sendi prismatic meluncur oleh z_i jika sambungan tersebut adalah interkoneksi tautan i dan $i + 1$. Variabel-variabel sambungan, dilambangkan dengan

$$\theta_i$$

untuk sambungan *revolute* dan d_i untuk sambungan prismatic, merupakan perpindahan relatif antara tautan yang berdekatan.

1. Derajat Kebebasan dan Ruang Kerja

Jumlah sambungan menentukan derajat kebebasan (DOF) dari manipulator. Biasanya, seorang manipulator harus memiliki setidaknya enam DOF independen: tiga untuk penentuan posisi dan tiga untuk orientasi. Dengan kurang dari enam DOF lengan tidak dapat mencapai setiap titik dalam lingkungan kerjanya dengan orientasi sewenang-wenang. Aplikasi tertentu seperti reach sekitar atau di belakang rintangan membutuhkan lebih dari enam DOF. Kesulitan mengendalikan seorang manipulator meningkat dengan cepat dengan jumlah tautan. Seorang manipulator memiliki lebih dari enam tautan disebut sebagai manipulator berlebihan secara kinematis. Ruang kerja seorang manipulator adalah volume total yang disapu oleh *end-effector* sebagai manipulator menjalankan semua gerakan yang mungkin. Ruang kerja dibatasi oleh geometri manipulator serta kendala mekanis pada sendi. Misalnya, *revolute* sendi mungkin terbatas pada kurang dari 360 full penuh gerakan. Ruang kerja sering rusak menjadi ruang kerja yang terjangkau dan ruang kerja dekstro. Ruang kerja yang terjangkau adalah seluruh set poin yang dapat dijangkau oleh manipulator, sedangkan ruang kerja dekstro terdiri dari titik-titik yang dapat dicapai oleh manipulator dengan orientasi sewenang-wenang dari *end-effector*. Jelas ruang kerja dextrous adalah bagian dari ruang kerja yang bisa dijangkau. Itu ruang kerja beberapa robot diperlihatkan nanti dalam bab ini.

2. Klasifikasi Robot

Manipulator robot dapat diklasifikasikan berdasarkan beberapa kriteria, seperti sumber daya mereka, atau cara di mana sendi digerakkan, geometri,

atau struktur kinematik, yang dimaksudkan area aplikasi, atau metode kontrolnya. Klasifikasi semacam itu berguna terutama dalam untuk menentukan robot mana yang tepat untuk tugas yang diberikan. Misalnya, robot hidrolik tidak akan cocok untuk aplikasi penanganan makanan atau membersihkan kamar, sedangkan SCARA Robot tidak akan cocok untuk bekerja di pengecoran. Menjelaskan ini secara lebih rinci di bawah ini.

- Sumber Daya Biasanya, robot bertenaga listrik, hidrolik, atau pneumatik. Hidrolik aktuator tidak tertandingi dalam kecepatan respons dan kemampuan menghasilkan torsi. Karena itu robot hidrolik digunakan terutama untuk mengangkat beban berat. Kelemahan dari hidrolik robot adalah bahwa mereka cenderung bocor cairan hidrolik, membutuhkan lebih banyak peralatan periferal, seperti pompa, yang juga membutuhkan lebih banyak perawatan, dan berisik. Robot didorong oleh DC atau motor servo AC semakin populer karena lebih murah, lebih bersih dan lebih tenang. Robot pneumatik tidak mahal dan sederhana tetapi tidak dapat dikendalikan dengan tepat. Akibatnya, robot pneumatik terbatas dalam jangkauan aplikasi dan popularitasnya.

- Area Aplikasi

Proyeksi area aplikasi robot masa depan terbesar adalah dalam perakitan. Karena itu, robot sering diklasifikasikan berdasarkan aplikasi menjadi robot assembly dan non-assembly. Majelis robot cenderung kecil, digerakkan oleh listrik dan dapat berputar atau SCARA (dijelaskan di bawah) dalam desain. Area aplikasi nonassembly utama hingga saat ini adalah pengelasan, semprotan pengecatan, penanganan material, dan pemuatan dan pembongkaran mesin.

- Metode Kontrol

Robot diklasifikasikan berdasarkan metode kontrol menjadi robot servo dan non-servo. Robot yang paling awal adalah robot non-servo. Robot ini pada dasarnya adalah perangkat loop terbuka yang gerakan terbatas pada perhentian mekanis yang telah ditentukan, dan mereka berguna terutama untuk transfer bahan. Padahal, sesuai definisi yang diberikan sebelumnya, robot stop tetap hampir tidak memenuhi syarat sebagai robot. Robot servo menggunakan kontrol komputer loop tertutup untuk menentukan gerak dan karenanya mampu menjadi perangkat yang benar-benar multifungsi dan dapat diprogram ulang.

Robot yang dikontrol servo selanjutnya diklasifikasikan me-

nurut metode pengontrolnya digunakan untuk memandu efektor akhir. Jenis robot paling sederhana di kelas ini adalah *point-to-point robot*. Robot *point-to-point* dapat diajarkan satu set poin terpisah tetapi tidak ada *control* di jalur end-effector di antara poin yang diajarkan. Robot seperti itu biasanya diajarkan serangkaian poin dengan liontin mengajar. Poin-poin tersebut kemudian disimpan dan diputar ulang. Robot *point-to-point* sangat terbatas dalam jangkauan aplikasi mereka. Di robot jalur kontinu, di sisi lain, seluruh jalur end-effector dapat dikontrol. Misalnya, robot *end-effector* dapat diajarkan untuk mengikuti garis lurus antara dua titik atau bahkan mengikuti kontur seperti jahitan pengelasan. Selain itu, kecepatan dan / atau akselerasi efektor akhir seringkali dapat dikontrol. Ini adalah robot paling canggih dan membutuhkan pengendali komputer dan pengembangan perangkat lunak yang paling canggih.

- Geometris

Kebanyakan manipulator industri saat ini memiliki enam atau lebih sedikit kebebasan. Manipulator ini biasanya diklasifikasikan secara kinematis berdasarkan tiga sendi pertama lengan, dengan pergelangan tangan yang dijelaskan secara terpisah. Mayoritas manipulator ini jatuh ke dalam salah satu dari lima jenis geometris: artikulasi (RRR), bulat (RRP), SCARA (RRP), silindris (RPP), atau cartesian (PPP). Membahas masing-masing secara rinci di bawah ini. Masing-masing dari lima konfigurasi ini adalah tautan serial robot Kelas manipulator keenam dan berbeda secara fundamental adalah yang disebut paralel robot. Dalam konfigurasi paralel, tautan disusun secara tertutup daripada terbuka rantai kinematik. Menyertakan diskusi tentang robot paralel untuk kelengkapan sebagai paralel robot menjadi semakin umum.

3. Pengaturan Kinematik Umum

- Konfigurasi Artikulasi Manipulator yang diartikulasikan juga disebut *manipulator revolute*, atau antropomorfik. Lengan artikulasi ABB IRB1400 ditunjukkan pada Gambar 2.20 Desain

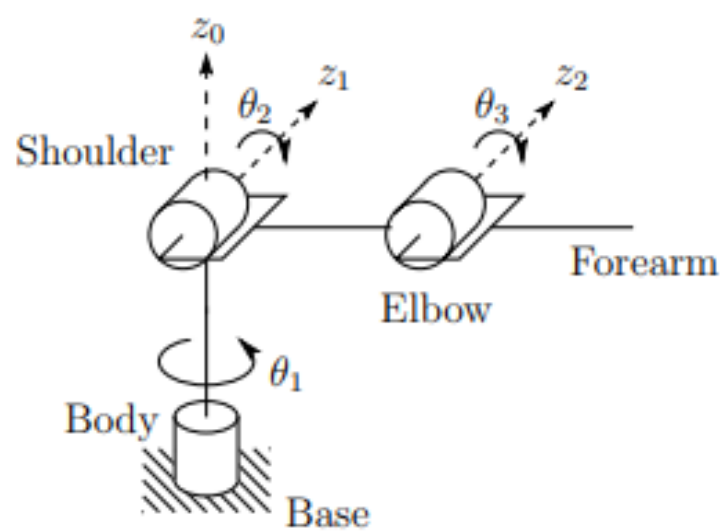


Gambar 2.20: Lengan artikulasi ABB IRB1400

sambungan *revolute* yang umum adalah tautan jajaran genjang seperti Motoman SK16, yang ditunjukkan pada Gambar 2.21. Di keduanya dari pengaturan ini sumbu gabungan z_2 sejajar dengan z_1 dan baik z_1 dan z_2 tegak lurus ke z_0 . Struktur dan terminologi yang terkait dengan manipulator siku ditampilkan pada Gambar 2.22. Ruang kerjanya ditunjukkan pada Gambar 2.23. Konfigurasi putaran menyediakan untuk kebebasan bergerak yang relatif besar dalam ruang yang kompak. Link jajaran genjang, meskipun kurang dekstroso daripada konfigurasi manipulator siku, namun



Gambar 2.21: Manipulator motoman SK16

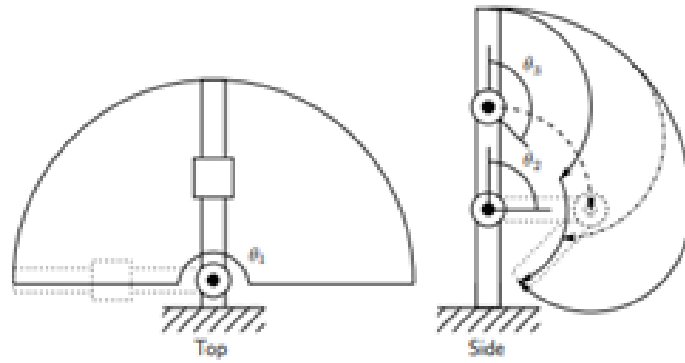


Gambar 2.22: Struktur manipulator siku

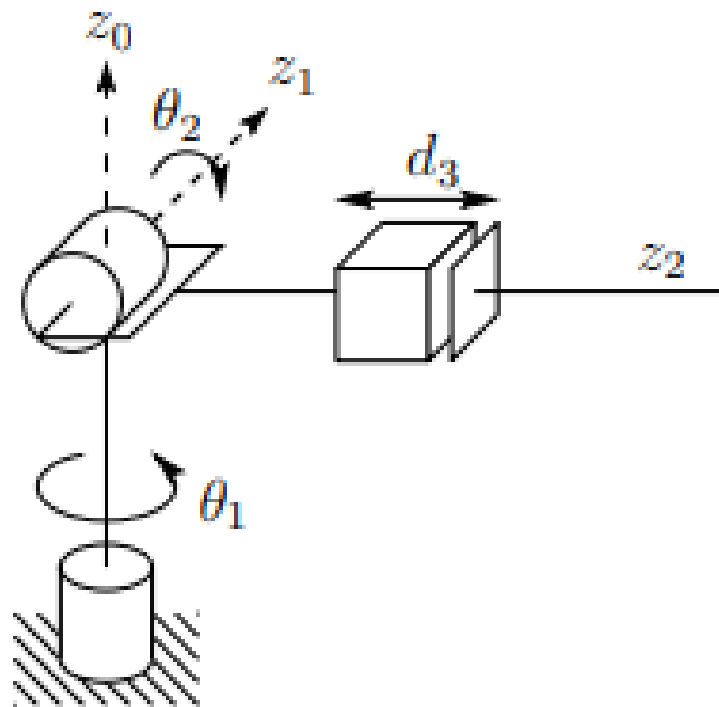
memiliki beberapa keunggulan yang menjadikannya desain yang menarik dan populer. Yang paling terkenal Fitur konfigurasi hubungan genjang adalah aktuator untuk sambungan 3 berada pada tautan 1. Karena bobot motor dilahirkan oleh tautan 1, tautan 2 dan 3 dapat dibuat lebih ringan dan motor itu sendiri bisa kurang kuat. Juga dinamika manipulator jajar genjang lebih sederhana daripada manipulator siku, sehingga membuatnya lebih mudah dikendalikan.

- Konfigurasi bulat

Dengan mengganti sendi ketiga atau siku dalam konfigurasi putaran oleh sambungan prismatic mendapatkan konfigurasi bola yang ditunjukkan pada Gambar 2.24. Istilah konfigurasi bola berasal dari fakta bahwa koordinat bola menentukan posisi efektor akhir sehubungan dengan bingkai yang asalnya terletak di persimpangan sumbu z_1 dan z_2 adalah sama dengan tiga variabel gabungan pertama. Gambar 2.25 menunjukkan Lengan Stanford, salah satunya kebanyakan robot bola yang terkenal. Ruang kerja dari manipulator bola ditunjukkan pada Gambar 2.26.



Gambar 2.23: Ruang kerja manipulator siku



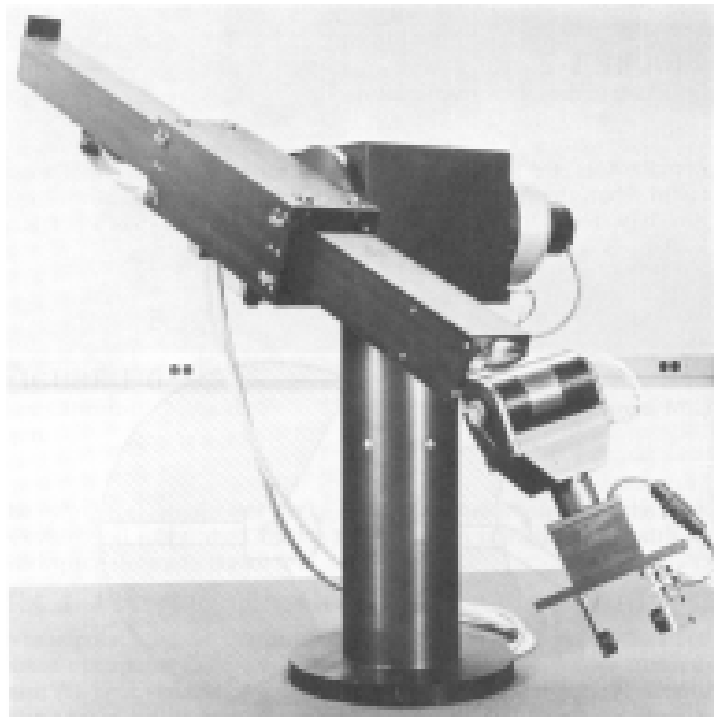
Gambar 2.24: Konfigurasi manipulator bola

- Konfigurasi SCARA (RRP)

Apa yang disebut SCARA (*for Selective Compliant Articulated Robot for Assembly*) ditunjukkan pada Gambar 2.27 adalah konfigurasi yang populer, yang, seperti namanya, dirancang untuk perakitan operasi. Meskipun SCARA memiliki struktur RRP, sangat berbeda dari konfigurasi bola dalam tampilan dan rentang aplikasinya. tidak seperti desain bola, yang memiliki z_0 , z_1 , z_2 saling tegak lurus, SCARA memiliki z_0 , z_1 , z_2 paralel. Gambar 2.28 menunjukkan Epson E2L653S, manipulator jenis ini. SCARA ruang kerja manipulator ditunjukkan pada Gambar 2.29.

- Konfigurasi Silinder (RPP)

Konfigurasi silinder ditunjukkan pada Gambar 2.30. Sambungan pertama adalah *revolute* dan menghasilkan rotasi di sekitar pangkalan, sedangkan sambungan kedua dan ketiga bersifat prismatik. Seperti namanya Konfigurasi Silinder ditunjukkan pada Gambar 2.30. Sambungan pertama adalah revolute dan menghasilkan rotasi di sekitar pangkalan, sedangkan sambungan kedua dan ketiga bersifat prismatik. Seperti namanya



Gambar 2.25: Lengan Stanford

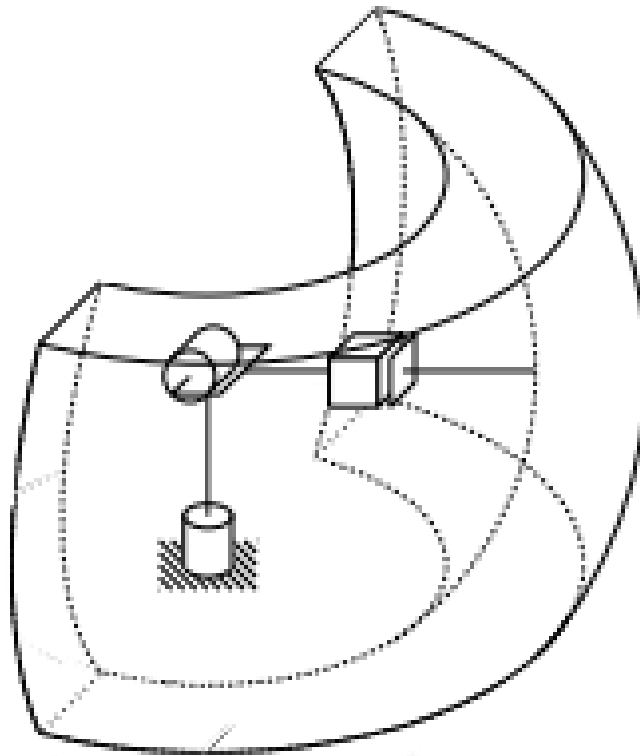
menunjukkan, variabel gabungan adalah koordinat silinder dari *end-effector* sehubungan dengan dasar. Robot silindris, Seiko RT3300, ditunjukkan pada Gambar 2.31, dengan ruang kerjanya ditunjukkan pada Gambar 2.32.

- Konfigurasi kartesius (PPP)

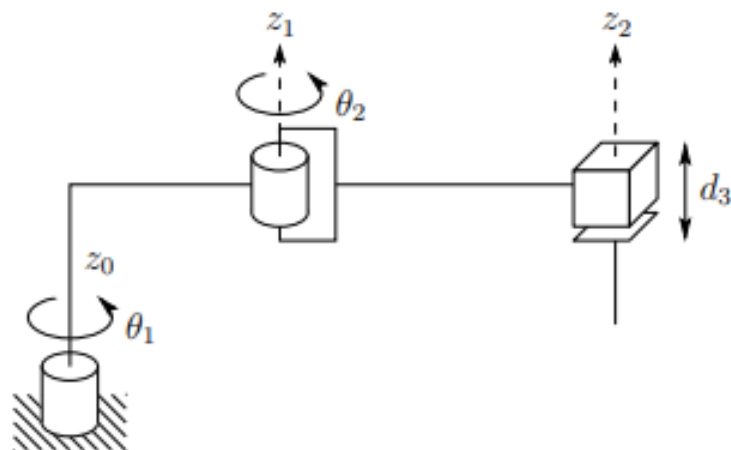
Seorang manipulator yang tiga persendian pertamanya prismatic dikenal sebagai manipulator kartesius, ditunjukkan pada Gambar 1.16. Untuk manipulator Cartesian, variabel gabungan adalah koordinat Cartesian dari *end-efektor* sehubungan dengan pangkalan. Seperti yang mungkin diharapkan deskripsi kinematik dari manipulator ini adalah yang paling sederhana dari semua konfigurasi. Konfigurasi kartesius berguna untuk aplikasi perakitan di atas meja dan, sebagai robot gantry, untuk transfer material atau kargo. Contoh robot kartesius, dari Epson-Seiko, ditunjukkan pada Gambar 2.34. Ruang kerja dari manipulator Cartesian ditunjukkan pada Gambar 2.35.

- Manipulator Paralel

Manipulator paralel adalah yang mana tautannya membentuk rantai tertutup. Lebih khusus, a manipulator paralel memiliki dua atau lebih rantai kinematik independen yang menghubungkan basis efektor akhir. Gambar 2.36 menunjukkan robot ABB IRB 940 Tricept, yang memiliki paralel konfigurasi. Kinematika rantai tertutup dari robot paralel dapat menghasilkan struktur yang lebih besar kekakuan, dan karenanya akurasi yang lebih besar, daripada robot rantai terbuka. Deskripsi kinematik dari robot paralel pada dasarnya berbeda dari robot tautan serial dan karenanya memerlukan metode analisis yang berbeda.



Gambar 2.26: Ruang kerja manipulator bola



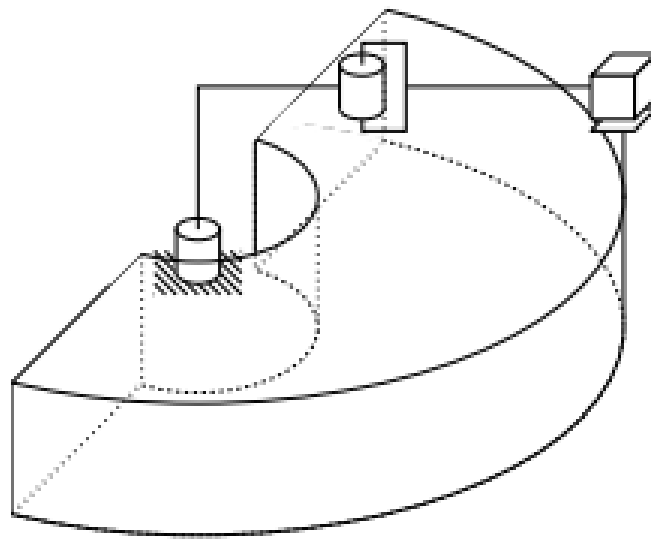
Gambar 2.27: *The SCARA (Selective Compliant Articulated Robot for Assembly).*

4. Sistem Robot

Seorang manipulator robot harus dipandang lebih dari sekadar serangkaian hubungan mekanis. Itu lengan mekanik hanya satu komponen untuk Sistem Robot keseluruhan, ditunjukkan pada Gambar 2.37, yang terdiri dari lengan, sumber daya eksternal, perkakas *end-of-arm*, eksternal dan sensor internal, antarmuka komputer, dan kontrol komputer. Bahkan deprogram perangkat lunak harus dianggap sebagai bagian integral dari keseluruhan sistem, karena cara dimana robot diprogram dan dikendalikan dapat memiliki dampak besar pada kinerjanya dan berbagai aplikasi selanjutnya.



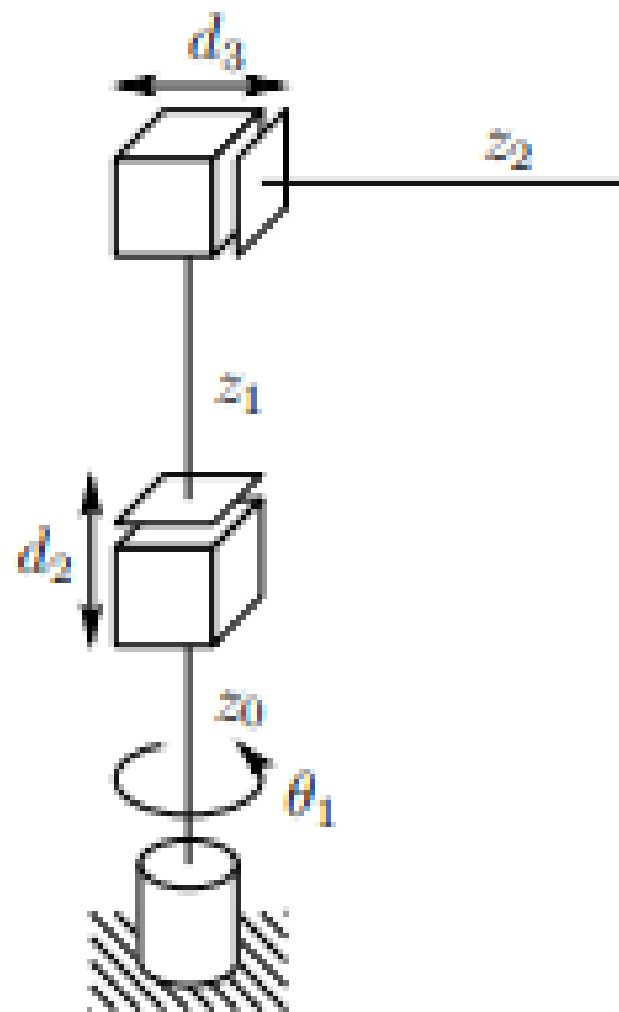
Gambar 2.28: Robot Epson E2L653S SCARA



Gambar 2.29: Ruang kerja manipulator SCARA

5. Akurasi dan pengulangan

Keakuratan seorang manipulator adalah ukuran seberapa dekat manipulator dapat mencapai titik tertentu dalam ruang kerjanya. Repeatability adalah ukuran seberapa dekat seorang manipulator dapat kembali ke titik yang diajarkan sebelumnya. Kebanyakan manipulator saat ini sangat tinggi berulang tetapi tidak terlalu akurat. Metode utama merasakan kesalahan penentuan posisi di kebanyakan case adalah dengan posisi encoders yang terletak di sambungan, baik di poros motor yang menggerakkan sendi atau pada sendi itu sendiri. Biasanya tidak ada pengukuran langsung posisi dan orientasi end-effector. Seseorang harus mengandalkan asumsi geometri manipulator dan kekakuannya untuk menyimpulkan (mis., untuk menghitung) posisi end-effector dari sudut sendi diukur. Karena itu akurasi dipengaruhi oleh kesalahan komputasi, pemesianan ketepatan dalam konstruksi manipulator, efek fleksibilitas seperti pembengkokan link di bawah beban gravitasi dan lainnya, serangan balik roda gigi, dan sejumlah statis dan lainnya efek dinamis. Terutama karena alasan inilah robot dirancang dengan sangat kekakuan tinggi. Tanpa kekakuan tinggi, akurasi hanya dapat ditingkatkan dengan semacam direct



Gambar 2.30: Konfigurasi manipulator silinder

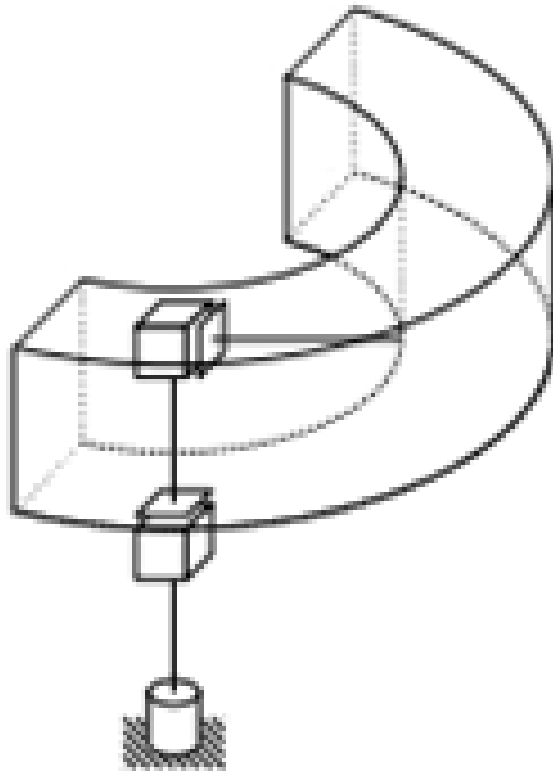


Gambar 2.31: Robot Seiko RT3300

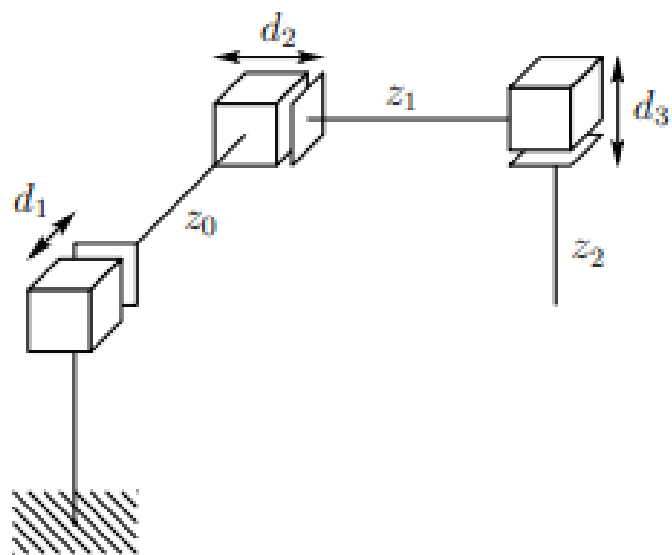
Penginderaan langsung dari posisi *end-effector*, seperti dengan penglihatan. Namun, begitu suatu poin diajarkan kepada manipulator, katakanlah dengan liontin mengajar, di atas efek diperhitungkan dan nilai-nilai encoder yang tepat diperlukan untuk kembali ke yang diberikan titik disimpan oleh komputer pengendali. Oleh karena itu pengulangan dipengaruhi oleh resolusi pengontrol. Resolusi pengontrol berarti peningkatan gerakan terkecil bahwa pengontrol dapat merasakan. Resolusi dihitung sebagai total jarak yang ditempuh oleh ujung dibagi

$$2^n$$

Dimana n adalah jumlah bit akurasi encoder. Dalam hal ini, sumbu linier, yaitu, sambungan prismatic, biasanya memiliki resolusi lebih tinggi daripada sambungan revolute, karena jarak garis lurus dilalui oleh ujung sumbu linear antara dua titik tersebut kurang dari arclength yang sesuai dilacak oleh ujung link rotasi. Selain itu, seperti yang akan kita lihat di bab-bab selanjutnya, sumbu rotasi biasanya menghasilkan besar jumlah kopling kinematik dan dinamis di antara tautan dengan akumulasi yang dihasilkan kesalahan dan masalah kontrol yang lebih sulit.



Gambar 2.32: Ruang kerja manipulator silinder



Gambar 2.33: Konfigurasi manipulator kartesian

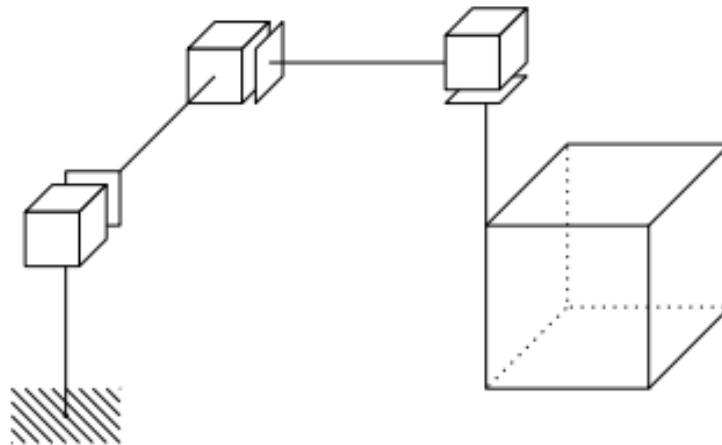
Untuk rentang gerak yang sama, tautan rotasi dapat dibuat jauh lebih kecil dari tautan dengan gerak linier. Dengan demikian manipulator yang terbuat dari sendi revolute menempati kerja yang lebih kecil volume daripada manipulator dengan sumbu linier. Ini meningkatkan kemampuan manipulator bekerja di ruang yang sama dengan robot, mesin, dan orang lain. Pada saat yang sama berputar manipulator bersama lebih mampu bermanuver di sekitar rintangan dan memiliki jangkauan yang lebih luas aplikasi yang mungkin.

6. Pergelangan Tangan dan *End-Effectors*

Pergelangan manipulator mengacu pada sendi dalam rantai kinematik di antara lengan dan tangan. Sendi pergelangan tangan hampir selalu semua berputar. Ini semakin umum terjadi manipulator desain dengan pergelangan bola, yang dimaksud adalah pergelangan tangan yang memiliki tiga sumbu sendi berpotongan pada titik yang sama.



Gambar 2.34: Robot Cartesian Epson



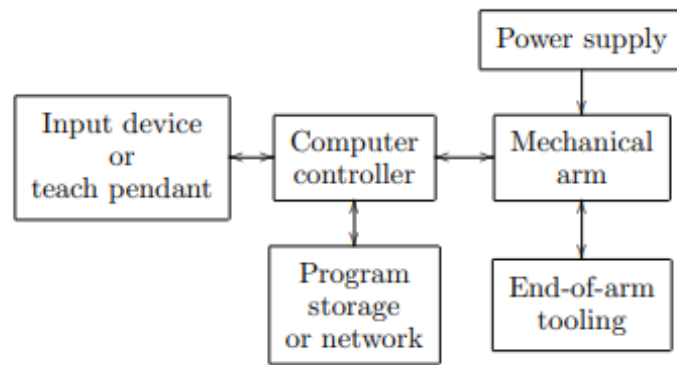
Gambar 2.35: Ruang kerja manipulator kartesian

Oleh karena itu biasanya, manipulator akan memiliki tiga derajat kebebasan posisi, yang mana diproduksi oleh tiga atau lebih sendi di lengan. Jumlah orientasi-derajat kebebasan kemudian akan tergantung pada derajat kebebasan pergelangan tangan. Adalah umum untuk menemukan pergelangan tangan memiliki satu, dua, atau tiga derajat kebebasan tergantung dari aplikasi. Sebagai contoh, robot SCARA yang ditunjukkan pada Gambar 2.28 memiliki empat derajat kebebasan: tiga untuk lengan, dan satu untuk pergelangan tangan, yang hanya memiliki gulungan tentang sumbu z akhir. Dikatakan bahwa robot hanya sebagai tangan atau end-effectornya. Lengan dan rakitan pergelangan tangan robot digunakan terutama untuk memposisikan end-effector dan alat apa pun mungkin membawa.

Ini adalah end-effector atau alat yang benar-benar melakukan pekerjaan. Meskipun ini memadai untuk transfer bahan, beberapa di antaranya penanganan bagian, atau memegang alat sederhana, itu tidak memadai untuk tugas-tugas lain seperti pengelasan, perakitan, penggilingan, dll. Oleh karena itu banyak penelitian yang dikhususkan untuk desain efektor tujuan akhir khusus serta alat yang dapat dengan cepat diubah sesuai dengan tugas yang ditentukan. Ada juga banyak penelitian yang dikhususkan untuk pengembangan tangan antropomorfik.



Gambar 2.36: *ABB IRB940 Tricept Parallel Robot*



Gambar 2.37: Komponen sistem robot

Tangan seperti itu sedang dikembangkan baik untuk penggunaan prostetik dan untuk digunakan dalam pembuatan. Sejak peduli dengan analisis dan kontrol dari manipulator itu sendiri dan bukan di aplikasi tertentu atau *end-effector*.