

BAB II

LANDASAN TEORI

2.1 Internet

Menurut Sugeng (2010:31), Internet (*interconnection-networking*) merupakan suatu sistem global dari seluruh jaringan komputer yang saling terhubung menggunakan standar *internet protocol suite* (TCP/IP) untuk melayani miliaran pengguna di seluruh dunia. Jadi, dapat disimpulkan bahwa internet adalah suatu jaringan komunikasi antar komputer yang besar, yang mencakup seluruh dunia dan berbasis pada sebuah protokol yang disebut TCP/IP (*Transmission Control Protocol / Internet Protocol*).

2.1.1 TCP/IP

Menurut Sugeng (2010:61), TCP/IP (*Transmission Control Protocol / Internet Protocol*) standar komunikasi yang digunakan oleh komunitas internet dalam proses pertukaran data dari satu komputer ke komputer lain di dalam jaringan internet. TCP/IP menggunakan banyak protokol didalamnya, namun protokol utamanya adalah TCP. Data diimplementasikan dalam bentuk perangkat lunak (*software*) di sistem operasi. Istilah yang diberikan kepada perangkat lunak ini disebut dengan TCP/IP *stack*.

TCP/IP mempunyai beberapa layer, antara lain :

1. IP (*Internet Protocol*) yang berperan pada perpindahan paket data dari *node* ke *node*. IP akan meneruskan paket dengan basis empat *byte address* tujuan

2. (nomor IP). Internet memberi otoritas kelompok penomoran IP untuk setiap organisasi yang berbeda. IP bekerja pada mesin *gateway* yang mengunggah data dari departemen organisasi kemudian ke region dan kemudian keseruluh dunia.
3. TCP (*Transmission Control Protocol*) berperan atas pengujian penyerahan data dari klien ke *server*. Data dapat saja hilang di antara jaringan. TCP memiliki dukungan untuk melakukan deteksi *error* atau kehilangan data dan memungkinkan memperbaiki *error* atau mengembalikan kehilangan data tersebut, sehingga data dapat diterima pada sisi penerima kembali sepenuhnya.
4. *Socket* merupakan suatu nama yang diberikan pada paket dari *subroutine* guna menyediakan akses ke TCP/IP pada banyak sistem.

2.1.2 Internet Protocol Version 4 (IPv4)

Menurut Sugeng (2010:81), *Internet protocol address* (alamat IP) adalah suatu komponen vital dalam dunia internet karena alamat IP dapat dikatakan sebagai identitas dari pemakai internet, sehingga antara satu alamat dengan alamat lainnya tidak boleh sama. IP versi 4 digunakan dalam protokol jaringan TCP/IP. IP versi ini memiliki keterbatasan yakni hanya mampu mengalami sebanyak 4 miliar *host* komputer di seluruh dunia.

Pada IPv4 ada 5 jenis kelas, tergantung dari besarnya bagian *host*, yaitu :

1. Kelas A (1 bit pertama IP *address*-nya “0”) alamat *unicast* untuk jaringan skala besar. Nomor urut bit tertinggi di dalam alamat IP kelas A selalu diset dengan nilai 0 (nol). Tujuh bit berikutnya untuk melengkapi oktet pertama akan membuat sebuah *network identifier*. 24 bit sisanya (atau tiga oktet

terakhir) merepresentasikan *host identifier*. Ini mengizinkan kelas A memiliki hingga 126 jaringan, dan 16,777,214 *host* tiap jaringannya.

2. Kelas B (2 bit pertama IP *address*-nya “10”) alamat *unicast* untuk jaringan skala menengah hingga skala besar. Dua bit pertama di dalam oktet pertama alamat IP kelas B selalu diset ke bilangan biner 10. 14 bit berikutnya (untuk melengkapi dua oktet pertama), akan membuat sebuah *network identifier*. 16 bit sisanya (dua oktet terakhir) merepresentasikan *host identifier*. Kelas B dapat memiliki 16,384 *network*, dan 65,534 *host* untuk setiap *network*-nya.
3. Kelas C (3 bit pertama IP *address*-nya “110”) alamat *unicast* untuk jaringan skala kecil. Tiga bit pertama di dalam oktet pertama alamat kelas C selalu diset ke nilai biner 110. 21 bit selanjutnya (untuk melengkapi tiga oktet pertama) akan membentuk sebuah *network identifier*. 8 bit sisanya (sebagai oktet terakhir) akan merepresentasikan *host identifier*. Ini memungkinkan pembuatan total 2,097,152 buah *network*, dan 254 *host* untuk setiap *network*-nya.
4. Kelas D (4 bit pertama IP *address*-nya “1110”) alamat *multicast* (bukan alamat *unicast*). sehingga berbeda dengan tiga kelas di atas. Empat bit pertama di dalam IP kelas D selalu diset ke bilangan biner 1110. 28 bit sisanya digunakan sebagai alamat yang dapat digunakan untuk mengenali *host*. Untuk lebih jelas mengenal alamat ini, lihat pada bagian alamat *multicast* IPv4.
5. Kelas E (4 bit pertama IP *address*-nya “1111”) umumnya digunakan sebagai alamat percobaan (eksperimen) dan dicadangkan untuk digunakan pada masa depan. Empat bit pertama selalu diset kepada bilangan biner 1111. 28

bit sisanya digunakan sebagai alamat yang dapat digunakan untuk mengenali *host*.

2.1.3 Network Address Translation (NAT)

Terbatasnya alamat IPv4 adalah permasalahan pada jaringan global atau internet. Untuk memaksimalkan penggunaan alamat IP yang diberikan oleh *internet service provider* (ISP), maka dapat digunakanlah *network address translation* (NAT). NAT membuat jaringan yang menggunakan alamat lokal (*private*), alamat yang tidak boleh ada pada tabel *routing* internet dan dikhususkan untuk jaringan lokal/intranet, agar dapat berkomunikasi ke jaringan internet dengan cara meminjam alamat IP internet yang diberikan oleh ISP.

Tabel 2.1 Keuntungan dan Kerugian dari Penggunaan NAT (Sugeng 2010:86):

Keuntungan	Kerugian
Menghemat alamat IP ilegal yang ditetapkan oleh NIC atau <i>service provider</i>	Translasi menimbulkan <i>delay switching</i>
Mengurangi terjadinya duplikat alamat jaringan	Menghilangkan kemampuan <i>trace (traceability) end to end IP</i>
Meningkatkan fleksibilitas untuk koneksi ke internet	Aplikasi tertentu tidak dapat berjalan jika menggunakan NAT
Menghidarkan proses pengalamatan kembali (<i>readdressing</i>) pada saat jaringan berubah.	

Apabila suatu paket dialihkan dengan *Network Address Translation* (NAT) pada suatu *link*, maka pada saat ada paket kembali dari tujuan maka *link* ini akan mengingat darimana asal dari paket itu, sehingga komunikasi akan berjalan seperti biasa. Penggunaan utama dari *Network Address Translation* (NAT) adalah untuk

membatasi jumlah alamat IP publik suatu organisasi atau perusahaan menggunakan IP publik baik untuk tujuan ekonomi maupun tujuan keamanan.



Gambar 2.1 Proses *Network Address Translation*
(sumber : www.medium.com)

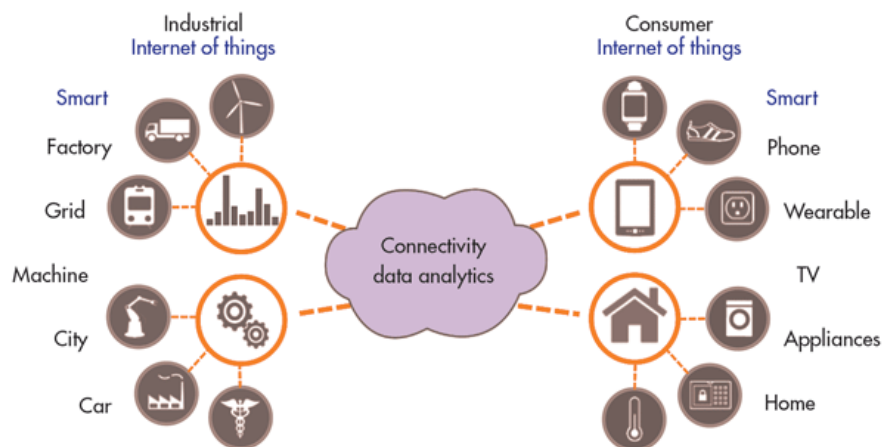
Pada jaringan komputer, proses *Network Address Translation* (NAT) adalah proses penulisan ulang (*masquerade*) pada alamat IP asal (*source*) dan/atau alamat IP tujuan (*destination*), setelah melalui *router* atau *firewall*. NAT digunakan pada jaringan dengan *workstation* yang menggunakan IP *Private* supaya dapat terkoneksi ke internet dengan menggunakan satu atau lebih IP *public*.

2.1.4 DNS (*Domain Name System*)

Menurut Sugeng (2010:143), *Domain Name System* adalah teknik penamaan pada alamat IP *address*. Penggunaan DNS sebagai penamaan alamat IP dikarenakan IP *address* merupakan sederetan angka sehingga sulit untuk diingat oleh manusia. *Routing* paket IP berbasis TCP/IP sebenarnya tidak memerlukan teknik DNS, cukup dengan IP *address*. Agar internet lebih mudah digunakan, diperlukan suatu cara untuk memetakan dari IP *address* ke nama *host* komputer dan sebaliknya, dan ini yang dinamakan DNS. Jadi, DNS dapat didefinisikan sebagai suatu sistem yang memungkinkan nama suatu *host* pada jaringan komputer atau internet ditranslasikan menjadi IP *address* atau sebaliknya.

2.2 Internet of Things

Menurut Rouse (2016), *internet of things* (IoT) adalah sistem perangkat komputasi yang saling terkait, mesin mekanis dan digital, objek, hewan atau orang-orang yang dilengkapi dengan pengenalan unik dan kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan akses manusia ke manusia atau interaksi manusia ke-komputer .



Gambar 2.2 Ilustrasi *Internet of Thing*
(sumber : *National Instruments, based on Moor Insights*)

Pada gambar 2.2 merupakan gambaran *internet of things* dimana semua perangkat terkoneksi ke jaringan internet dan dapat dikendalikan melalui jaringan. IoT akan mengubah dunia dan berdampak pada kehidupan sehari – hari baik siang ataupun malam akan dikelilingi oleh konektivitas yang cerdas merespon apa yang kita inginkan dan apa yang kita butuhkan yang diistilahkan *Digital Sixth Sense*. Cisco mendefinisikan bahwa IoT (*Internet of Tthing*) sebagai penyatu masyarakat , proses , data , dan yang lainnya menjadi terhubung bersama-sama dan menjadi lebih berharga dari sebelumnya.

2.3 Router

Menurut Athailah (2013:2), *router* adalah sebuah alat yang digunakan untuk mengatur rute sinyal atau data yang ada pada jaringan komputer sehingga dapat diarahkan menuju ke rute tertentu yang telah diatur sebelumnya dan menghasilkan suatu hubungan antar jaringan komputer itu sendiri.



Gambar 2.3 *Router*
(sumber : www.tp-link.co.id)

2.3.1 Fungsi Router

Adapun fungsi dari *router* ini adalah :

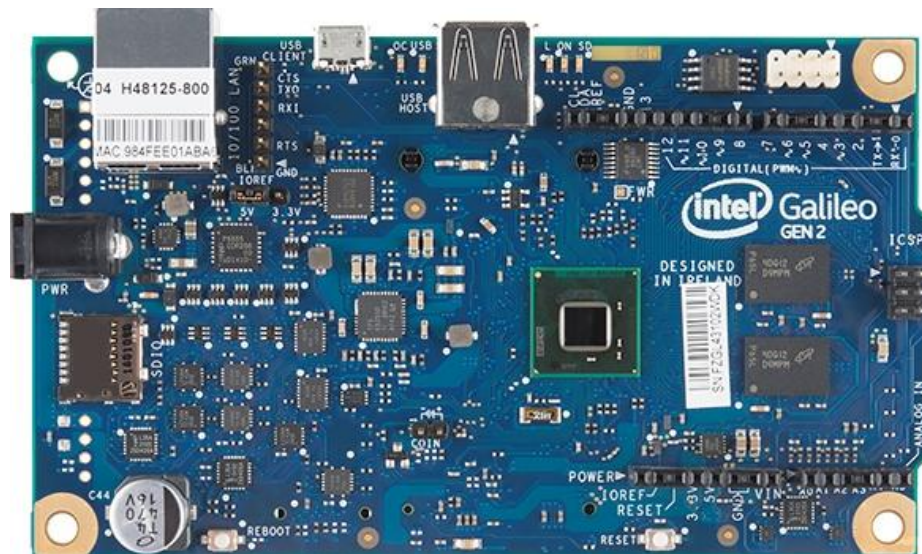
- a. Membaca IP *source* dan *destination* untuk menentukan *routing* dari suatu *local area network* ke *local area network* lain nya
- b. Menyimpan *routing table* untuk dapat menentukan rute terbaik antara *local area network* ke *wide area network*
- c. Bekerja pada layer 3 OSI layer

Router dapat bekerja dengan cara merutekan paket atau data informasi yang disebut dengan *routing*. Dengan teknik *routing* tersebut, *router* dapat mengetahui arah rute perjalanan informasi tersebut akan dituju, apakah berada pada satu jaringan yang sama atau berbeda. Jika informasi yang dituju mengarah kepada jaringan yang berbeda, maka *router* akan meneruskannya kepada jaringan

tersebut, sebaliknya apabila paket yang dituju adalah jaringan yang sama, maka *router* akan menghalangi paket keluar serta meneruskan paket tersebut dengan *routing* di jaringan yang sama sampai terkirim ke tujuan.

2.4 Intel Galileo

Menurut Ramon (2014), Intel Galileo adalah *embedded board* yang merupakan gabungan dari Arduino Uno dengan prosesor *Intel Quark SoC X1000*. Prosesor *Quark SoC X1000* merupakan prosesor yang dibuat Intel dan didesain khusus untuk pengembangan perangkat *internet of things*.



Gambar 2.4 Intel Galileo Gen 2
(sumber : www.communities.intel.com)

Intel Galileo bisa disebut dengan Arduino dengan *high speed processor* dan banyak fitur untuk pengembangan *project internet of thing*. Salah satunya bisa diinstallkan sistem operasi seperti *windows* ataupun *linux*. Semua program dari Arduino IDE yang tersedia untuk mikrokontroler 8 bit bisa digunakan ke dalam Intel Galileo. Sintak pemograman Arduino bisa menjalankan perintah-perintah *linux* seperti mengkopi file, menghapus file, pindah direktori dan lain-lain

Tabel 2.2 Spesifikasi *Board Intel Galileo Gen 2*

Spesifikasi	
Prosesor	- Intel® Quark™ SoC X1000
Kecepatan	- 400 MHz
Arsitektur	- 32-bit Intel® Pentium® <i>processor-compatible ISA</i>
Memori	- <i>Firmware/Bootloader</i> (8 MB) - SRAM 512 kb (<i>embedded</i>) - DRAM 256 MB DDR3 - SD Card Sampai 32 GB
Komunikasi	- USB 2.0 <i>host port</i> (standar tipe A) - USB 2.0 <i>client port</i> (<i>micro</i> USB tipe B) - TTL UART 6-pin <i>header</i>
Tegangan Operasi	- 5 Volt
Tegangan Masukan (rekomendasi)	- 7-12 Volt
I/O	- 20 digital I/O - 6 analog input - 6 pwm dengan resolusi 12-bit

2.4.1 Input dan Output Intel Galileo

Masing-masing dari 14 pin digital pada Intel Galileo dapat digunakan sebagai *input* atau *output*, menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Mereka beroperasi di tegangan 5 V. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor *pull-up* internal dari 20-50 Kilo Ohm. Selain itu, beberapa pin memiliki fungsi khusus:

1. Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data TTL serial.
2. Eksternal Interupsi: 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu interupsi pada nilai yang rendah, tepi naik atau jatuh, atau perubahan nilai.

Lihat *attach Interrupt ()* fungsi untuk rincian.

3. PWM: 3, 5, 6, 9, 10, dan 11. Menyediakan 12-bit *output* PWM dengan *analogWrite ()* fungsi.
4. SPI: 10 (SS), 11 (mosi), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI.
5. LED: 13. Ada *built-in* LED terhubung ke pin digital 13. Ketika pin adalah nilai *high*, LED menyala, ketika pin adalah *low*. Intel Galileo memiliki 6 input analog, diberi label A0 sampai A5, masing-masing menyediakan sampai 12 bit resolusi dengan nilai 4096. Secara *default* sistem mengukur dari 0 sampai 5 V

2.5 Arduino IDE

Arduino IDE adalah sebuah perangkat lunak pemrograman yang memudahkan kita mengembangkan aplikasi mikrokontroler mulai dari menuliskan *source* program, kompilasi, upload hasil kompilasi, dan uji coba secara terminal serial.

Arduino IDE bisa dijalankan di komputer dengan berbagai macam *platform* karena didukung atau berbasis Java. *Source* program yang dibuat untuk aplikasi mikrokontroler adalah bahasa C/C++ dan dapat digabungkan dengan *assembly*. Tentunya terdapat *style* khusus yang membedakannya yaitu:

- a. *void main(void)* sebagai fungsi program utama diganti dengan *void loop()* .

Perbedaannya pada c biasa tidak terjadi *loop*, jadi harus ada looping yang ditambahkan misalnya *while(1){.....}*. Dalam Arduino secara otomatis fungsi *loop()* akan kembali lagi dari awal jika sudah dieksekusi intruksi paling bawahnya.

- b. Ditambahkan fungsi *void setup(void)*, fungsi ini digunakan untuk inisialisasi mikrokontroler sebelum fungsi utama *loop()* dieksekusi.

Tidak direpotkan dengan *setting register*, karena arduino sudah memasukkannya kedalam *library*-nya dan secara otomatis disesuaikan dengan jenis *board* arduino berkenaan jenis mikrokontrolernya. Sehingga setup perangkat keras menjadi mudah.

2.6 PWM (*Pulse With Modulation*)

Menurut Recktenwald (2011), *Pulse Width Modulation* atau PWM adalah teknik untuk mengatur tegangan listrik ke beban yang dimilikinya dengan respon yang reaktif lambat. Sinyal suplai terdiri dari sederetan tegangan pulsa sedemikian rupa sehingga lebar pulsa dapat mengendalikan tegangan efektif pada beban. Baik sinyal AC dan DC yang dapat disimulasikan dengan PWM.

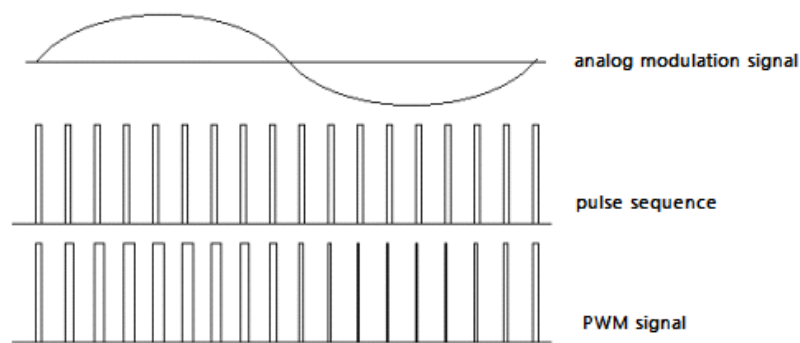
PWM merupakan sebuah mekanisme untuk membangkitkan sinyal keluaran yang periodenya berulang antara *high* dan *low* dimana kita dapat mengontrol durasi sinyal *high* dan *low* sesuai dengan yang kita inginkan. *Duty cycle* merupakan presentase periode sinyal *high* dan periode sinyal *low*, presentase *duty cycle* akan bebanding lurus dengan tegangan rata-rata yang dihasilkan. Berikut ilustrasi sinyal PWM, misalkan kondisi *high* 5 V dan kondisi *low* 0 V.

Pengaturan lebar pulsa modulasi atau PWM merupakan salah satu teknik yang “ampuh” yang digunakan dalam sistem kontrol (*control system*) saat ini. Pengaturan lebar modulasi dipergunakan di berbagai bidang yang sangat luas, salah satu diantaranya adalah: *speed control* (kendali kecepatan), *power control*

(kendali sistem tenaga), *measurement and communication* (pengukuran atau instrumentasi dan telekomunikasi).

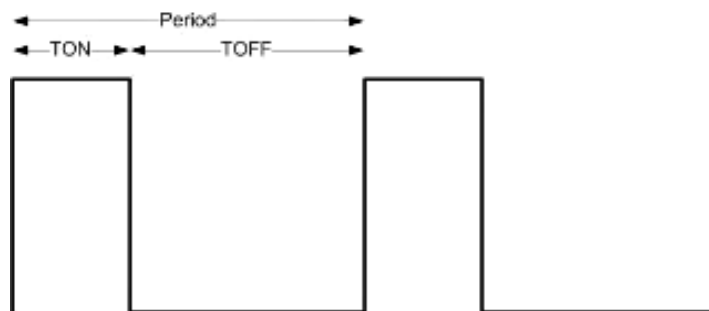
2.6.1 Prinsip Dasar PWM

Lebar pulsa (PWM) diperoleh dengan bantuan sebuah gelombang kotak yang mana siklus kerja (*duty cycle*) gelombang dapat diubah untuk mendapatkan sebuah tegangan keluaran yang bervariasi yang merupakan nilai rata – rata dari gelombang tersebut.



Gambar 2.5 Proses Modulasi Lebar Pulsa
(sumber : *web.tiscali.it*)

Pada gambar 2.6 diatas terlihat bahwa dalam proses modulasi ini ampitudo pulsa gelombang pembawa konstan, yang berubah – ubah adalah lebar pulsa yang bersangkutan dibandingkan dengan sebelum modulasi.



Gambar 2.6 Bentuk Gelombang Kotak (Pulsa)
(sumber : *www.ebldc.com/?p=48*)

T_{on} adalah waktu dimana tegangan keluaran berada pada posisi tinggi (*high* atau 1) dan T_{off} adalah waktu dimana tegangan keluaran berada pada posisi rendah (*low* atau 0). Untuk mendapatkan periode satu gelombang maka (T_{Total}) dilakukan penjumlahan antara T_{on} dengan T_{off} .

$$T_{Total} = T_{on} + T_{off} \dots\dots\dots(2.1)$$

Siklus kerja atau *duty cycle* sebuah gelombang didefinisikan sebagai,

$$D = \frac{T_{on}}{T_{on} + T_{off}} \times 100 \% \dots\dots\dots(2.2)$$

Tegangan keluaran dapat bervariasi dengan *duty-cycle* dapat dirumuskan sebagai berikut,

$$V_{out} = D \times V_m \dots\dots\dots(2.3)$$

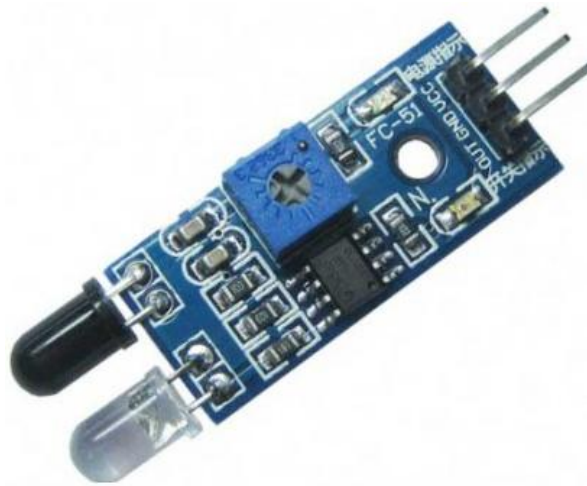
Dari rumus diatas ditarik kesimpulan bahwa tegangan keluaran dapat diubah-ubah secara langsung dengan mengubah nilai T_{on} , apabila T_{on} adalah 0, maka V_{out} juga akan 0. Apabila T_{on} adalah T_{total} maka V_{out} adalah V_{in} dengan kata lain nilai maksimumnya.

2.7 Sensor Inframerah

Menurut muvida (2010), sensor merupakan piranti yang mengubah besaran-besaran fisis seperti magnetik, radiasi, mekanik dan termal menjadi besaran listrik. Komponen utama pembentuk sensor adalah struktur sensor, teknologi manufaktur, dan algoritma pengolah sinyal, hal inilah yang menentukan kemampuan suatu sensor.

Sensor inframerah menggunakan catu daya 5V, berguna untuk mendeteksi halangan dengan memancarkan gelombang inframerah dan tidak terpengaruh oleh

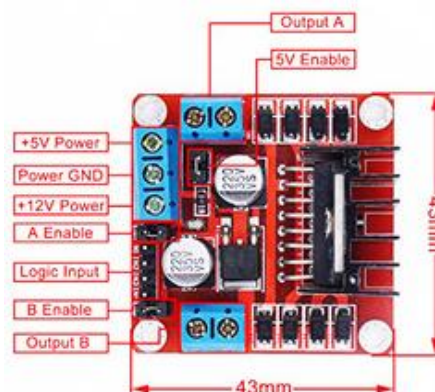
cahaya lampu maupun sinar matahari. Sensor inframera terpasang satu pemancar dan satu penerima.



Gambar 2.7 Sensor Inframerah
(sumber : www.ecdio.com)

2.8 Driver Motor L298N

Driver motor L298N merupakan *driver* motor yang paling populer digunakan untuk mengontrol kecepatan dan arah pergerakan motor terutama pada *robot line foller / line tracer*. Kelebihan dari *driver* motor L298N ini adalah cukup presisi dalam mengontrol motor. Selain itu, kelebihan *driver* motor L298N adalah mudah untuk dikontrol.



Gambar 2.8 *Driver* Motor L298N
(sumber : www.wholesaler.alibaba.com)

Untuk mengontrol *driver* L298N ini dibutuhkan 6 buah pin mikrokontroller. Dua buah untuk pin *enable* (satu buah untuk motor pertama dan satu buah yang lain untuk motor kedua. Karena *driver* L298N ini dapat mengontrol dua buah motor DC) 4 buah untuk mengatur kecepatan motor motor tersebut. Skematik rangkaian *driver* motor L298N harus ditambahkan beberapa komponen lagi agar dapat bekerja.

Yang pertama berupa rangkaian regulator yang berada dibagian atas skematik. dan yang kedua adalah rangkaian pendukung *driver* motor yang berupa beberapa dioda. *Output* dari rangkaian ini sudah berupa dua pin untuk masing masing motor. Pada prinsipnya rangkaian *driver* motor L298N ini dapat mengatur tegangan dan arus sehingga kecepatan dan arah motor dapat diatur (Kadir 2015:190).

2.9 Motor Servo

Motor *servo* adalah sebuah motor DC kecil yang diberi *gear* dan potensiometer dan dapat menempatkan horn *servo* pada posisi yang dikehendaki. Motor *servo* menggunakan sistem *close loop* sehingga posisi *horn* bisa dipertahankan. Ada 2 jenis motor *servo*, yaitu motor *servo* standar dan *continuous*. Motor *servo* standar sering dipakai pada sistem robotika. Misalnya dalam pembuatan lengan robot. Motor *servo continuous* sering dipakai untuk *mobile* robot. Pada motor *servo* tertulis tipe *servo* yang bersangkutan (Malik 2009 :152)



Gambar 2.9 Motor *Servo*
(sumber : www.servodatabase.com)

2.10 IP Camera

IP camera adalah kamera digital yang digunakan sebagai pemantau keamanan serta dapat mengirim dan menerima data melalui jaringan IP seperti LAN, intranet atau internet dan memungkinkan pengguna untuk melihat dan mengelola kamera menggunakan *web browser* standar atau *software management* video pada setiap komputer lokal atau *remote* pada jaringan. Kamera jaringan membuat orang – orang menjadi lebih praktis untuk memonitoring area atau ruangan yang di pantau (www.stealth.co.id).



Gambar 2.10 IP Camera
(sumber : www.dlink.com)

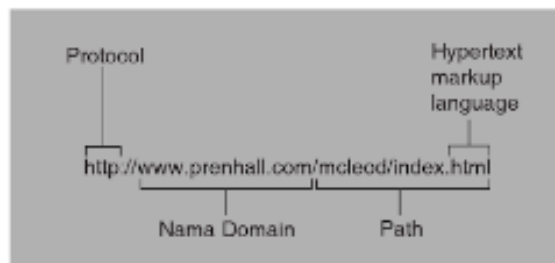
IP camera merupakan perkembangan dari CCTV dan yang membedakan *IP camera* dengan CCTV biasa adalah setiap kamera memiliki IP sendiri sehingga

bisa memilih kamera mana yang mau dilihat. *IP camera* memiliki beberapa jenis yaitu:

1. *IP camera* terpusat, jenis ini membutuhkan pusat *Network Video Recorder* (NVR) untuk merekam video dan manajemen alarm.
2. *IP camera* desentralisasi, jenis *IP camera* ini tidak memerlukan NVR karena telah memiliki fungsi perekam *built-in* dan dapat merekam secara langsung ke media penyimpanan seperti *SD card*, komputer ataupun *server*.

2.11 Web

Menurut Mcleod (2008 : 78), web adalah informasi yang dapat diakses melalui internet dimana dokumen-dokumen *hypermedia* disimpan dan kemudian diambil dengan menggunakan metode penentuan alamat yang unik. Internet memberikan arsitektur jaringan, dan web memberikan metode untuk menyimpan dan mengambil dokumen – dokumennya. *World Wide Web* adalah kumpulan komputer yang bertindak sebagai server yang menyimpan dokumen – dokumen yang diformat untuk memungkinkan dilihatnya teks, grafik, dan audio dan link dokumen – dokumen lainnya.



Gambar 2.11 Terminologi *World Wide Web*
(sumber : www.book.google.co.id)

Pada gambar 2.8 mengilustrasikan bagaimana protokol, nama domain, dan *path* digunakan.

2.12 Wireshark

Menurut Kurniawan (2012:15), Wireshark adalah tool yang ditujukan untuk penganalisaan paket data jaringan. Wireshark melakukan pengawasan paket secara *real time* dan kemudian menangkap data dan menampilkannya selengkap mungkin. Wireshark memungkinkan seorang pengguna mengamati sebuah data jaringan yang sedang beroperasi atau dari sebuah data yang ada pada *disk*, dan langsung melihat dan mensortir data yang tertangkap. Informasi singkat dan detail bagi masing-masing paket, termasuk *full header* dan porsi data, bisa diperoleh.

Wireshark memiliki beberapa fitur termasuk *display filter language* yang kaya dan kemampuan untuk merekonstruksi kembali sebuah aliran data yang telah di *capture* oleh wireshark. Paket *sniffer* sendiri dapat di artikan sebagai sebuah *program* atau *tool* yang memiliki kemampuan untuk mencegat dan melakukan pencatatan terhadap *traffic* data dalam jaringan. Wireshark sebagai salah satu packet *sniffer* diprogram sedemikian rupa untuk mengenali berbagai macam protokol jaringan.

Wireshark memiliki seperangkat fitur yang meliputi sebagai berikut :

- a. Tersedia untuk *Unix*, *Linux* dan *Windows*
- b. Menangkap paket data dari antar muka jaringan
- c. Wireshark dapat menangkap lalu lintas dari banyak jenis jaringan media yang berbeda, termasuk LAN nirkabel juga. Media jenis tersebut didukung, tergantung pada banyak hal seperti sistem operasi yang digunakan.
- d. Tampilan paket dengan informasi protokol yang sangat rinci.

2.13 App Inventor

App Inventor adalah sebuah *tool* untuk membuat aplikasi Android yang berbasis *visual block programming* menggunakan *Web based Graphical User Interface (GUI) builder*. App Inventor disediakan` oleh Google dan sekarang dikelola oleh Massachusetts Institute of Technology (MIT). MIT Media Lab merilis Android App Inventor untuk membangun aplikasi Android tanpa koding, karena tools ini berbasis *visual block programming* yang *open source*.

Membuat suatu aplikasi Android dengan App Inventor tidak perlu menjadi seorang pengembang profesional (*professional developer*), hal ini dapat dirancang dengan semudah mungkin bagi siapapun baik programmer maupun non-programmer, karena secara visual dapat langsung merancang aplikasi menggunakan blok-blok program yang sudah memiliki program dan fungsi-fungsi tertentu yang berbentuk potongan-potongan puzzel.

App Inventor *Designer* digunakan untuk melakukan desain atau mengatur tata letak komponen dari tampilan aplikasi yang ingin dibuat, *App Inventor Block Editor* digunakan untuk melakukan pemrograman terhadap komponen-komponen sesuai fungsi masing-masing yang diinginkan sedangkan Android Emulator atau Android Phone digunakan untuk melakukan simulasi terhadap aplikasi yang telah dibuat bisa menggunakan emulator atau bisa langsung menggunakan Android phone. Hasil dari perancangan aplikasi kemudian dapat didownload setelah di kompile menjadi file aplikasi Android *Package* yang berekstensi .apk.