BAB 2
LANDASAN TEORI

2.1 KAJIAN PUSTAKA

Penilitian Heri Arya Supriyatna, Yuwaldi Away, Zulhelmi pada tahun 2019 yang berjudul "Desain Sistem Internet of Things (IoT) Untuk Pemantauan Dan Prediksi Gejala Serangan Jantung" melakukan pengujian pada sensor ECG AD8232 menggunakan perangkat lunak Parallax Data Acquisition dan Matlab telah berhasil dilakukan dengan adanya data berupa bentuk sinyal gelombang P, Q, R, S, T dan nilai tegangan dari masing-masing sinyal gelombang tersebut. Pengujian buzzer dan thinkspeak telah berhasil dilakukan dengan adanya data yang tersimpan pada tampilan channel yang dibuat di server thinkspeak. Keseluruhan sistem internet of things untuk pemantauan dan prediksi gejala serangan jantung telah berhasil dilakukan dengan adanya hail pengujian berupa data detak jantung dan data buzzer.[2]

Penelitian Ary Sulistyo Utomo, Erda Hermono Puspo Negoro, Mohamad Sofie pada tahun 2019 yang berjudul "Monitoring Heart Rate dan Saturasi Oksigen Melalui Smartphone". Hasil pembacaan heart rate dan saturasi oksigen dihasilkan dari pembacaan sensor max 3000 yang dibaca menggunakan Arduino uno. Arduino uno akan mengirim data pembacaan melalui perangkat smartphone melalui perangkat ESP8266 ESP01 ke smartphone. Dari hasil pengujian menunjukan selisih nillai heart rate dan saturasi oksigen pada alat dan pasien monitoring terbesar 0,8% untuk heart rate dan 1% untuk oksigen.[6]

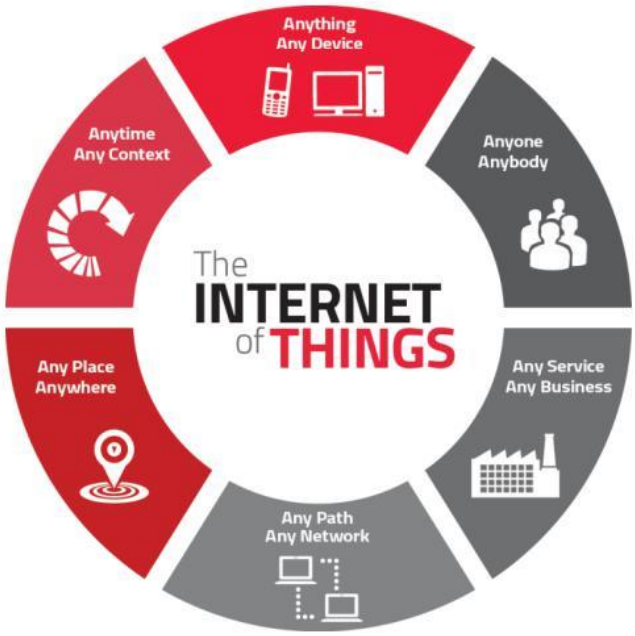
Penelitian Yosef Febri Wiryawan, Dany Primanita Kartikasari, Mahendra Data pada tahun 2018 yang berjudul "Implementasi Constrained Application Protocol (CoAP) pada Sistem Pengamatan Kelembaban Tanah". protokol CoAP sebagai protokol transfer diimplementasikan pada sistem pengamatan kelembaban tanah. Pada sisi perangkat menggunakan mikrokontroler Wemos DI dan sensor YL-69 yang digunakan untuk mendeteksi kelembaban tanah dan mengirimkannya ke server menggunakan modul bloetooth HC-05. Dari hasil pengujian menunjukkan bahwa kenaikan delay dapat berubah secara signifikan jika ukuran muatan (payload) lebih dari 64 byte, hal ini terjadi dikarenakan terjadi mekanisme blockwise pada komunikasi menggunakan protokol CoAP. Selain itu jarak antara sensor dengan modul bluetooth yang merupakan terjadinya proses pertukaran client dan server akan berpengaruh pada tinggi rendahnya delay.[7]

Penilitian Nailis Dyanningrum, Dodi Zulherman, Herryawan Pujiharsono pada tahun 2018 yang berjudul "Analisis Rancangan Sistem Pengukuran Denyut Nadi Berbasis Internet of Things". Pengujian mengukur denyut nadi menggunakan pulse sensor dan secara manual. Nilai hasil dari sensor akan ditampilkan dalam bentuk angka pada LCD 16x2 dan akan ditampilkan dalam bentuk grafik pada platform thingspeak.[8]

2.2 DASAR TEORI

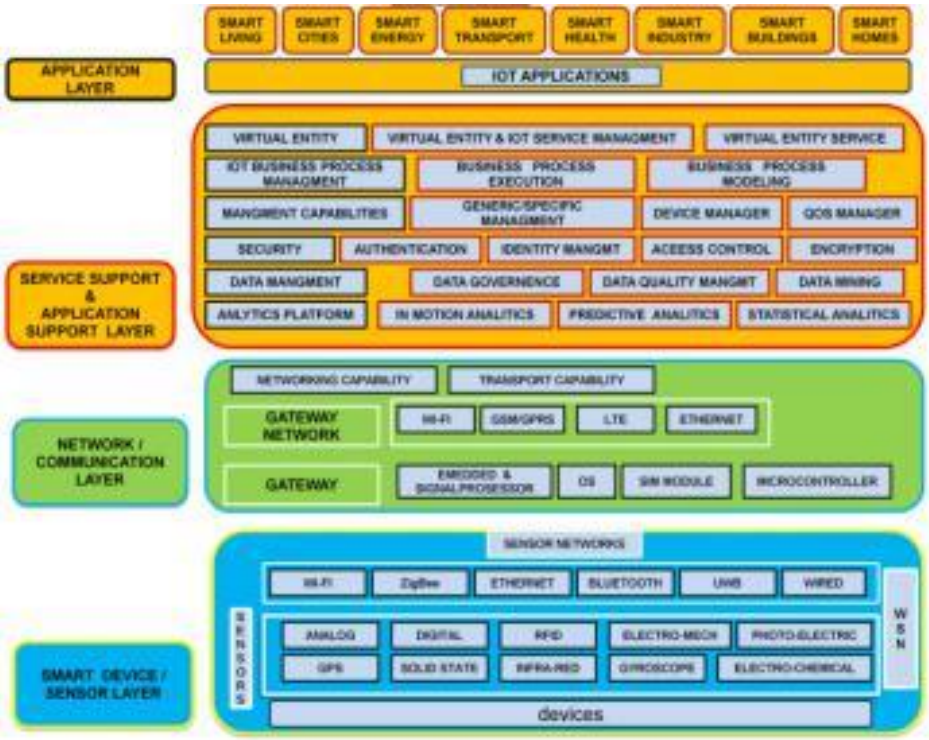
2.2.1 Interner of Things (IoT)

Internet of Things atau disingkat dengan sebutan IoT jika dilihat dari singkatannya, IoT atau *Internet of Things* didefinisikan sebagai kumpulan benda fisik atau piranti yang saling berhubungan dengan piranti lainnya dengan memanfaatkan konektivitas jaringan internet yang tersambung dan saling berkolaborasi satu sama lain sehingga mampu menjalankan perintah – perintah kerja yang tepat, efektif dan efisien. [9]



Gambar 2. 1 *1Internet of Things (IoT)* [9]

Secara umum arsitektur teknologi IoT terdiri dari beberapa lapisan, hal ini menggambarkan bahwa beragamnya teknologi yang saling berhubungan satu sama lain untuk berkomunikasi. Arsitektur IoT dapat dibagi menjadi 4 bagian, seperti pada gambar 2.2 berikut.



Gambar 2. 2 Arsitektur *IoT* [9]

a. *Smart Device / Sensor Layer*

Merupakan lapisan terendah dari arsitektur IoT yang terdiri dari benda – benda pintar yang terintegrasi dengan sensor. Sensor – sensor pada layer ini memungkinkan adanya interkoneksi fisik dengan dunia digital yang mana terjadi proses pengumpulan informasi dan selanjutnya akan diproses. Sensor memiliki kapasitas untuk mengambil pengukuran seperti pengukuran suhu, kelembaban atau kualitas udara, tekanan, aliran, gerakan atau getaran, dll. Sensor dikelompokkan sesuai dengan tujuan masing – masing dari sensor itu sendiri, seperti berdasarkan lingkungan sensor, sensor tubuh, peralatan rumah atau ruangan dan sensor kendaraan, dan lain - lain. [9]

Biasanya kebanyakan dari sensor memerlukan suatu konektivitas dengan *gateway sensor*, ini biasanya dilakukan pada koneksi menggunakan *Local Area Network* (LAN) yaitu koneksi *Ethernet* atau *WiFi.* Sementara itu untuk sensor yang tidak memerlukan konektivitas ke sensor agregator, konektivitas sensor dengan server dapat menggunakan jaringan *Wide Area Network* (WAN) contohnya seperti GSM, GPRS dan LTE. Sensor dalam penggunaannya memiliki daya rendah dan juga *data rate* yang rendah, maka dari itu biasanya sensor membentuk suatu

jaringan yang dikenal dengan sebutan jaringan sensor *wireless* atau *Wireless Sensor Network* (WSN). WSN sangat populer dikarenakan kemampuannya dalam mengakomodasi lebih banyak *node* sensor dan konsumsi daya baterai yang rendah dalam cakupan yang luas. [9]

b. *Gateway and Networks*

Data dengan *volume* besar dari sensor akan diproduksi atau diproses, hal ini tentunya membutuhkan media transportasi kabel yang kuat dan memiliki kinerja yang tinggi atau kualitas dari infrastruktur jaringan nirkabel *(wireless)*. Arus jaringan atau kecepatan layanan pengiriman informasi ditentukan oleh jenis protokol yang digunakan, tentunya protokol tersebut sudah mendukung jaringan *machine-to-machine* (M2M). [9]

Beberapa jaringan dengan berbagai teknologi dan akses protokol diperlukan untuk bekerja satu sama lain dalam satu konfigurasi heterogen. Jaringan tersebut bisa dibentuk secara *private*, *public* atau *hybrid* dan dibangun untuk menunjang persyaratan komunikasi seperti latensi, *bandwidth* atau *security*. Berbagai jenis *gateway* yang digunakan seperti mikrokontroler, mikroprosesor, dll., dan jaringan *gateway* seperti *Wi-Fi*, GSM, GPRS atau LTE. [9]

c. *Management Service Layer*

Manajemen data adalah kemampuan untuk mengelola informasi aliran data. Dengan *management service layer*, informasi dapat diakses, terintegrasi dan dikendalikan. *Management service* membuat pemrosesan informasi yang mungkin melalui analitik, kontrol keamanan, pemodelan proses dan manajemen perangkat. [9]

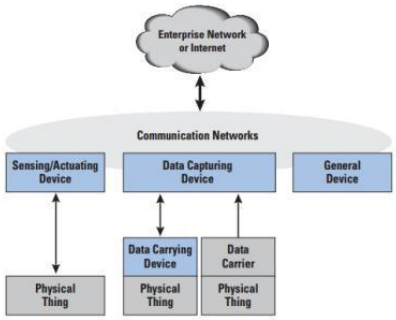
Salah satu fitur penting dalam *management service layer* adalah bisnis dan proses aturan mesin. IOT membawa koneksi dan interaksi objek, dan sistem bersama – sama memberikan informasi dalam bentuk peristiwa atau data kontekstual seperti suhu barang, lokasi terkini atau data lalu lintas. Beberapa peristiwa ini memerlukan penyaringan atau *routing* ke *post-processing* sistem, seperti menangkap data sensorik periodik. Sementara yang membutuhkan *respons* terhadap situasi langsung seperti reaksi terhadap keadaan darurat yaitu pada kondisi kesehatan pasien. Aturan mesin mendukung perumusan logika keputusan dan

memicu proses interaktif dan otomatis untuk mengaktifkan lebih banyak lagi sistem responsif IOT. [9]

d. *Application Layer*

Dalam pengaplikasiannya IoT dapat mencakup pada *Smart Environment* atau *Smart Spaces* yang meliputi *Smart City, Smart Transport, Smart Industry, Smart Living, Smart Building, Smart Homes, Smart Health, Smart Energy,* dan lain – lain.

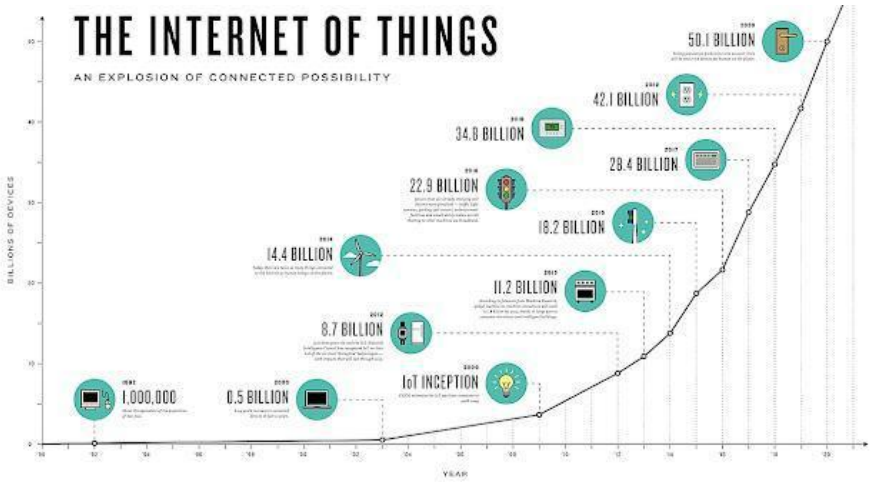
Aspek yang unik dari IoT ini adalah terdapat pada piranti dan banyaknya jumlah benda fisik, selain dari perangkat komputasi dan pemrosesan data. Berikut ini merupakan tipe – tipe perangkat IoT mengacu pada model ITU-T. [10]



Gambar 2. 3 Tipe perangkat *Internet of Things* [10]

Cakupan dari teknologi IoT ini sangat luas, bahkan tidak terbatas dan akan terus mengalami perkembangan setiap tahunnya. Maka dari itu banyak *developer* yang melakukan riset dan menemukan hal – hal baru pada teknologi ini. Tujuannya tidak lain adalah untuk mempermudah dalam memenuhi kebutuhan manusia, dan sampai mana batasan dari teknologi ini bisa dibilang belum ada batasannya. [10]

Pada gambar 2.4 berikut merupakan prediksi CISCO mengenai perkembangan jumlah piranti IoT sampai dengan tahun 2020.



Gambar 2. 4 Perkembangan Teknologi *Internet of Things (IoT)* [11]

Karakteristik yang mendasar dari teknologi *Internet of Things* (IoT) adalah sebagai berikut [9] :

a. *Interconnectivity*

Dengan menggunakan teknologi IoT, benda apapun bisa interkoneksi dan saling berkomunikasi, bertukar informasi dengan menggunakan infrastruktur global.

b. *Things-related services*

IoT mampu menyediakan layanan yang berkaitan dengan *"things"* seperti perlindungan privasi dan konsistensi antara *physical things* dan *virtual things* yang terkait.

c. *Heterogeneity*

Perangkat IoT bersifat heterogen. Hal ini berdasarkan *platform* perangkat keras dan jaringan. Perangkat atau *platform* dapat berinteraksi dengan yang lainnya melalui jaringan yang berbeda.

d. *Dynamic changes*

Perangkat berubah secara dinamis, seperti *on* dan *off*, terhubung dan terputus dan juga jumlah perangkat dapat berubah secara dinamis.

e. *Enormous Scale*

IoT bisa mencapai skala besar. Jumlah perangkat yang diperlukan dan dikelola untuk berkomunikasi satu sama lain lebih besar dibandingkan dengan perangkat yang terhubung ke internet saat ini.

f. *Safety*

Kemanan dirancang untuk menjaga kemanan data pribadi data kemanan fisik.

g. *Connectivity*

Konektivitas dalam IoT memungkinkan untuk aksesbilitas untuk mendapatkan jaringan, dan kompabilitas yaitu kemampuan untuk mengkonsumsi dan memproduksi data.

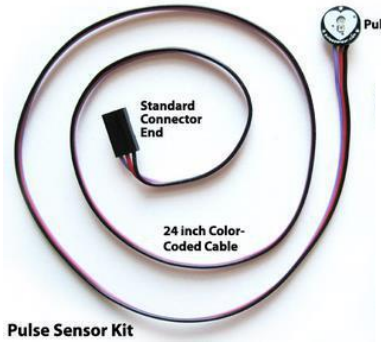
2.2.2 Denyut Nadi (Jantung)

Jantung adalah organ vital dan merupakan pertahanan terakhir untuk hidup selain otak. Denyut yang ada di jantung ini tidak bisa dikendalikan oleh manusia. Denyut jantung biasanya mengacu pada jumlah waktu yang dibutuhkan oleh detak jantung per satuan waktu. Secara umum hal tersebut direpresentasikan sebagai beats per minute (BPM) karena waktu standar yang dapat digunakan untuk mengukur berapa denyut

jantung manusia, yaitu berdasarkan menit, tepatnya 1 menit. Denyut jantung manusia dewasa ratarata yaitu: 60–100 bpm. Jika memang denyut jantung di bawah atau di atas standar, maka terdapat kemungkinan organ jantung mengalami masalah. [12]

2.2.3 Pulse Sensor

Pulse Sensor pada dasarnya adalah alat medis yang berfungsi untuk memantau kondisi denyut jantung manusia. Rangkaian dasar dari sensor ini dibangun menggunakan photodioda dan LED. Sensor ini bekerja berdasarkan prinsip pantulan sinar LED. Kulit dipakai sebagai permukaan reflektif untuk sinar LED. Kepadatan darah pada kulit akan mempengaruhi reflektifitas sinar LED. Aksi pemompaan jantung mengakibatkan kepadatan darah meningkat. Pada saat jantung memompa darah, maka darah akan mengalir melalui pembuluh arteri dari yang besar hingga kecil seperti di ujung jari. Volume darah pada ujung dari bertambah maka intensitas cahaya yang mengenai photodioda akan kecil karena terhalang oleh volume darah, begitu pula sebaliknya. Keluaran sinyal dari photodioda kemudian dikuatkan oleh sebuah Op-Amp sehingga dapat dibaca oleh ADC mikrokontroler. Gambar 2 menunjukkan komponen perangkat *pulse sensor*.[13]



Gambar 2. 5 Komponen Perangkat *Pulse Sensor* [13]

Perangkat Pulse Sensor terdiri dari [13] :

1. Kabel 24-inch color-coded, dengan konektor standar (0,1” pitch). Dengan kabel ini perangkat pulse sensor pun semakin mudah dipasang dengan perangkat mikrokontroler, tanpa adanya proses menyolder.
2. Klip untuk telinga. Dimana alat dapat ditempelkan pada telinga sehingga denyut pada telinga dapat diperoleh dan diolah menjadi grafik pada layar display.
3. 2 buah Velcro dots (tempelan pada kain)

4. 3 buah stiker transparan. Dimana penggunaannya berfungsi untuk menutup komponen depan pada pulse sensor pada tangan yang basah agar komponen tidak rusak.

2.2.4 LCD (Liquid Cristal Display)

LCD merupakan salah satu perangkat penampil yang sekarang ini mulai banyak digunakan. Penampil LCD mulai dirasakan menggantikan fungsi dari penampil CRT (*Cathode Ray Tube*), yang sudah berpuluh-puluh tahun digunakan manusia sebagai penampil gambar/text baik monokrom (hitam dan putih), maupun yang berwarna. Teknologi LCD memberikan keuntungan dibandingkan dengan teknologi CRT, karena pada dasarnya, CRT adalah tabung triode yang digunakan sebelum transistor ditemukan.[14]

Beberapa keuntungan LCD dibandingkan dengan CRT adalah konsumsi daya yang relative kecil, lebih ringan, tampilan yang lebih bagus, dan ketika berlama-lama di depan monitor, monitor CRT lebih cepat memberikan kejenuhan pada mata dibandingkan dengan LCD. [14]

Gambar 2. 6 LCD 16x2 [14]

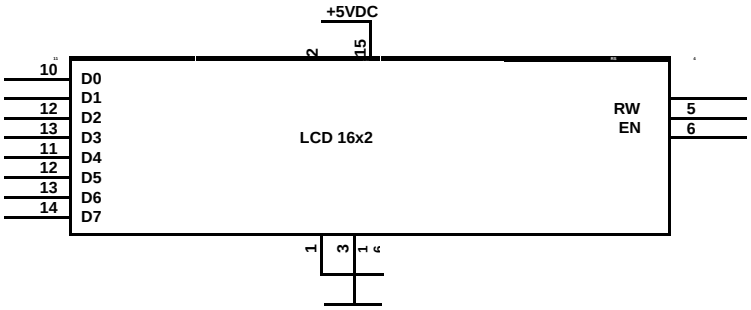
LCD memanfaatkan silicon atau gallium dalam bentuk Kristal cair sebagai pemendar cahaya. Pada layar LCD, setiap matrik adalah susunan dua dimensi piksel yang dibagi dalam baris dan kolom. Dengan demikian, setiap pertemuan baris dan kolom adalah sebuah LED terdapat sebuah bidang latar (backplane), yang merupakan lempengan kaca bagian belakang dengan sisi dalam yang ditutupi oleh lapisan elektroda trasparan. Dalam keadaan normal, cairan yang digunakan memiliki warna cerah. Daerah-daerah tertentu pada cairan akan berubah warnanya menjadi hitam ketika tegangan diterapkan antara bidang latar dan pola elektroda yang terdapat pad sisi dalam lempeng kaca bagian depan. [14]

Keunggulan LCD adalah hanya menarik arus yang kecil (beberapa microampere), sehingga alat atau sistem menjadi portable karena dapat menggunakan catu daya yang

kecil. Keunggulan lainnya adalah tampilan yang diperlihatkan dapat dibaca dengan mudah di bawah terang sinar matahari. Di bawah sinar cahaya yang remang-remang dalam kondisi gelap, sebuah lampu (berupa LED) harus dipasang dibelakang layar tampilan. [14]

LCD yang digunakan adalah jenis LCD yang mena mpilkan data dengan 2 baris tampilan pada display. Keuntungan dari LCD ini adalah :

1. Dapat menampilkan karakter ASCII, sehingga dapat memudahkan untuk membuat program tampilan.
2. Mudah dihubungkan dengan port I/O karena hanya mengunakan 8 bit data dan 3 bit control.
3. Ukuran modul yang proporsional.
4. Daya yang digunakan relative sangat kecil.



Gambar 2. 7 Konfigurasi Pin LCD [14]

Operasi dasar pada LCD terdiri dari empat, yaitu instruksi mengakses proses internal, instruksi menulis data, instruksi membaca kondisi sibuk, dan instruksi membaca data. ROM pembangkit sebanyak 192 tipe karakter, tiap karakter dengan huruf 5x7 dot matrik. Kapasitas pembangkit RAM 8 tipe karakter (membaca program), maksimum pembacaan 80x8 bit tampilan data. Perintah utama LCD adalah Display Clear, Cursor Home, Display ON/OFF, Display Character Blink, Cursor Shift, dan Display Shift. Tabel 2.1 menunjukkan operasi dasar LCD. [14]

Tabel 2. 1 Operasi Dasar LCD

RS	R/W	Operasi
0	0	Input Instruksi ke LCD
0	1	Membaca Status Flag (DB7) dan alamat counter (DB0 ke DB6)
1	0	Menulis Data

1	1	Membaca Data

Tabel 2. 2 Konfigurasi LCD

Pin	Bilangan biner	Keterangan
RS	0	Inisialisasi
	1	Data
RW	0	Tulis LCD / W (write)
	1	Baca LCD / R (read)
E	0	Pintu data terbuka
	1	Pintu data tertutup

Tabel 2. 3 Konfigurasi Pin LCD

Pin No.	Keterangan	Konfigurasi Hubung
1	GND	Ground
2	VCC	Tegangan +5VDC
3	VEE	Ground
4	RS	Kendali RS
5	RW	Ground
6	E	Kendali E/Enable
7	D0	Bit 0
8	D1	Bit 1
9	D2	Bit 2
10	D3	Bit 3
11	D4	Bit 4
12	D5	Bit 5
13	D6	Bit 6
14	D7	Bit 7

15	A	Anoda (+5VDC)
16	K	Katoda (Ground)

Lapisan film yang berisis Kristal cair diletakkan di antara dua lempeng kaca yang telah ditanami elektroda logam transparan. Saat teganga dicatukan pada beberapa pasang elektroda, molekul – molekul Kristal cair akan menyusun diri agar cahaya yang mengenainya akan dipantulkan atau diserap. Dari hasil pemantulan atau penyerapan cahaya tersebut akan terbentuk pola huruf, angka, atau gambar sesuai bagian yang di aktifkan. [14]

LCD membutuhkan tegangan dan daya yang kecil sehingga sangat popular untuk aplikasi pada kalkulator, arloji digital, dan instrument elektronika lain seperti *Global Positioning System* (GPS), baragraph display dan multimeter digital. LCD umumnya dikemas dalam bentuk *Dual In Line Package* (DIP) dan mempunyai kemampuan untuk menampilkan beberapa kolom dan baris dalam satu panel. Untuk membentuk pola, baik karakter maupun gambar pada kolom dan baris secara bersamaan digunakan metode *Screening.* [14]

Metode *screening* adalah mengaktifkan daerah perpotongan suatu kolo dan suatu baris secara bergantian dan cepat sehingga seolah-olah aktif semua. Penggunaan metode ini dimaksudkan untuk menghemat jalur yang digunakan untuk mengaktifkan panel LCD. Saat ini telah dikembangkan berbagai jenis LCD, mulai jenis LCD biasa, Passive Matrix LCD (PMLCD), hingga Thin-Film Transistor Active Matrix (TFT-AMLCD). Kemampuan LCD juga telah ditingkatkan dari yang monokrom hingga yang mampu menampilkan ribuan warna.[14]

2.2.5 NodeMCU ESP8266 V3

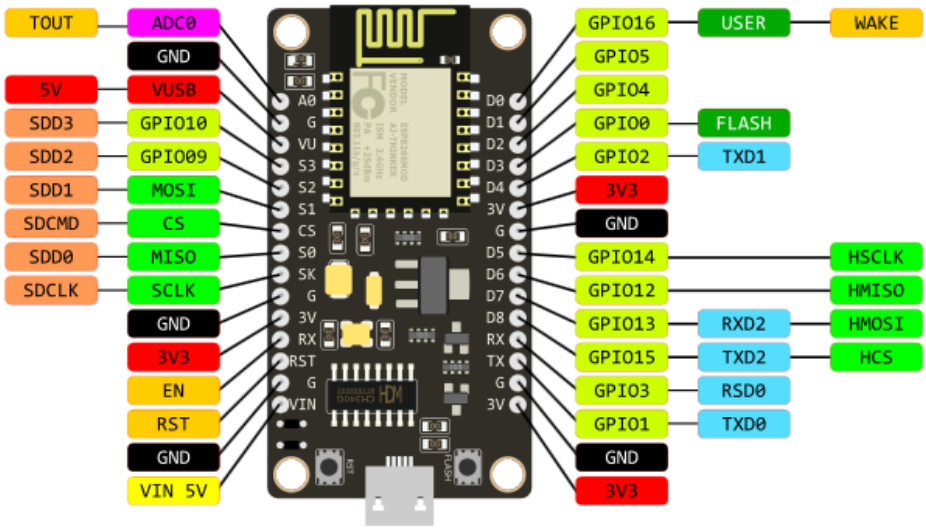
NodeMCU merupakan sebuah modul pengembangan dari modul *platform Internet of Things* (IoT) yang berbasis chip ESP8266 tipe ESP-12 yang memiliki firmware berbasis e-Lua. NodeMCU bekerja sama halnya dengan Arduino, namun yang membedakan dengan keduanya adalah NodeMCU diciptakan khusus untuk platform yang terkoneksi dengan internet atau sudah memiliki modul wifi sendiri didalamnya. NodeMCU telah me-package ESP8266 ke dalam sebuah board yang kompak dengan berbagai fitur layaknya mikrokontroler dan 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. [15]

NodeMCU juga bisa di Program menggunakan Arduino IDE, software yang digunakan untuk memprogram board Arduino. Namun ada perbedaan antara board NodeMCU dengan board Arduino. Sehingga dalam memprogram NodeMCU menggunakan Arduino IDE harus memperhatikan penomoran GPIO sesuai board arduino bukan seperti yang tertulis pada fisik NodeMCU. [15]

Pada gambar 2.8 merupakan tampilan fisik dari modul NodeMCU V3 berbasis chip ESP8266.

Gambar 2. 8 NodeMCU ESP8266 V3 [15]

Pada gambar 2.9 berikut merupakan konfigurasi pin NodeMCU ESP8266 V3 berikut dengan keterangan pin yang digunakan pada penelitian ini.

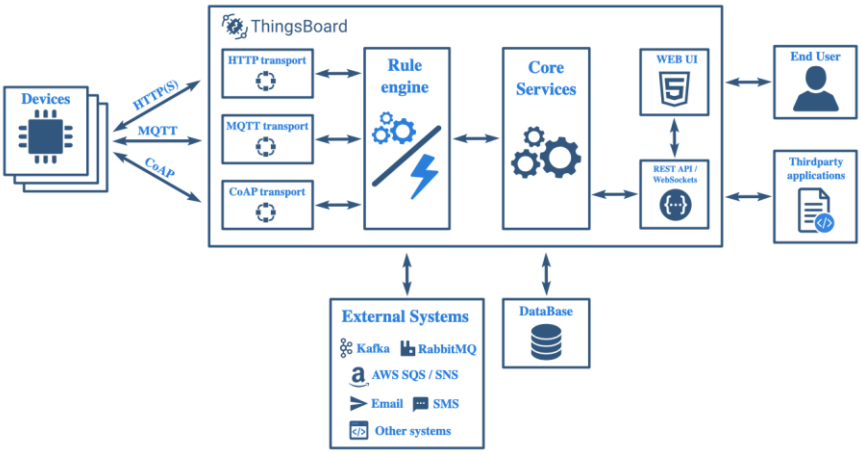


Gambar 2. 9 Konnfigurasi pin NodeMCU ESP8266 V3 [15]

2.2.6 Thingsboard

ThingsBoard adalah platform IoT sumber terbuka untuk pengumpulan, pemrosesan, visualisasi, dan manajemen perangkat. Ini memberikan solusi cloud IoT atau solusi lokal yang siap pakai untuk memungkinkan infrastruktur sisi server untuk berbagai aplikasi IoT. Dibangun pada platform Java 8, ThingsBoard menyediakan dukungan 100 persen untuk protokol IoT standar untuk konektivitas perangkat, termasuk MQTT, CoAP, dan HTTP(S), dan saat ini mendukung tiga opsi basis data yang berbeda: SQL, NoSQL, dan Database hibrida. Platform ThingsBoard menggunakan database ini untuk menyimpan entitas (seperti perangkat, aset, dasbor, pengguna, alarm, pelanggan, dll.), dan data telemetri (atribut, sensor deret waktu bacaan, statistik, peristiwa, dll.). Telemetri adalah rangkaian waktu dari pasangan data nilai kunci yang terkait dengan perangkat tertentu, dan ThingsBoard menyimpan data yang diterima sebagai telemetri. Aset adalah wadah untuk menata ulang data yang diterima, dan dapat digunakan untuk mengunggah hasil pengolahan data. Atribut, di sisi lain, biasanya mewakili fitur perangkat seperti versi firmware, perangkat keras spesifikasi, dll. yang ditetapkan ke perangkat dan aset terdaftar dalam bentuk pasangan nilai kunci. Database SQL seperti PostgreSQL menyimpan semua entitas dan telemetri dalam database SQL, sedangkan opsi NoSQL seperti Cassandra menyimpan semua entitas dan telemetri dalam database NoSQL. Dalam Hibrida opsi basis data, semua entitas disimpan dalam basis data SQL sementara semua telemetri disimpan di NoSQL basis data [11, 30, 38, 44]. Dalam proyek ini, database PostgreSQL diinstal pada ThingsBoard server untuk entitas dan penyimpanan telemetri. [15]

ThingsBoard memiliki dua edisi berbeda, Edisi Komunitas, yang gratis dan sepenuhnya terbuka source, dan Professional Edition, yang memiliki fitur lebih canggih. Dalam proyek ini, Edisi Komunitas digunakan. Edisi Komunitas ini adalah open source, dan tersedia gratis di situs web resmi ThingsBoard dan di platform pengembangan perangkat lunak GitHub [15]. Dengan back-end ThingsBoard yang ditulis dalam Java, dan beberapa layanan mikronya berdasarkan Node.js, Arsitektur ThingsBoard dirancang agar dapat diskalakan, toleran terhadap kesalahan, kuat dan efisien, dapat disesuaikan, dan tahan lama. Arsitektur dasar ThingsBoard ditunjukkan pada Gambar 2.9 [15].



Gambar 2. 10 Arsitektur Dasar *Thingsboard* [15]

2.2.7 Constrained Application Protocol (CoAP)

Constrained Aplication Protocol (CoAP) merupakan sebuah protokol yang berada pada *Application Layer*, yang mana memiliki sifat *request/response.* CoAP merupakan protocol yang dikembangkan oleh IETF *(International Engineering Task Force)*, protokol ini termasuk ke dalam standar RFC 7252. CoAP digunakan pada perangkat atau *device* yang memiliki keterbatasan *(Constrainer Device)* seperti pada penggunaan energi, memori ataupun yang lainnya. Berikut adalah tabel 2.1, merupakan klasifikasi protokol dalam layer yang berbeda.[18]

Tabel 2. 4 Protokol dalam layer yang berbeda [18]

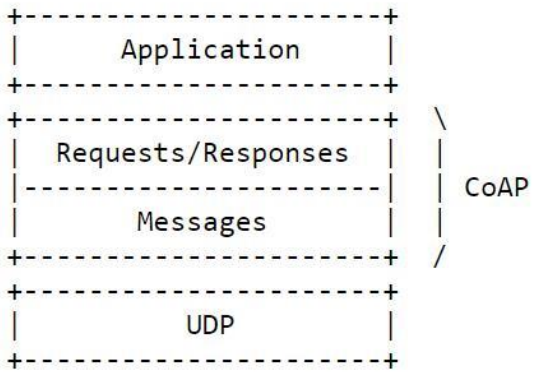
Application Layers	HTTP, CoAP, EBHTTP, LTP, SNMP, Ipfix, DNS, NTP, SSH, DLMS, COSEM, DNP, MODBUS.
Network/Communication Layers	Ipv6/IPv4, RPL, TCP/UDP, uIP, SLIP, 6LoWPAN.
PHY/ MAC layer	IEEE 802.11 Series, 802.15 Series, 802.3, 802.16, WirelessHART, Z-WAVE, UWB, IrDA, PLC, LonWorks, KNX.

Dilihat dari sisi kelebihannya, CoAP dapat menggunakan *power* yang rendah, seperti dalam penggunaan sesnsor, *relay, switch* atau perangkat yang membutuhkan daya rendah yang memerlukan pengendalian. Berikut adalah fitur yang dimiliki oleh protokol CoAP :

- Protokol yang memiliki standar M2M pada perangkat yang memiliki keterbatasan *(Constrained Device).*
- Mendukung permintaan *unicast* dan *multicast.*

- Tipe *messaging Asynchronous*
- *Low header overhead* dan *parsing complexity*.
- *Simple proxy* dan *caching capabilities.*
- Pengikatan keamanan pada *Datagram Transport Layer Security* (DTLS)

Protokol CoAP secara logis menggunakan pendekatan dua *layer*, yaitu UDP dan *Application.* Sebuah *CoAP messaging layer* digunakan untuk menangani UDP dan interaksi *asynchronous*, dan interaksi *request/response* menggunakan metode dan kode tanggapan. Namun CoAP merupakan sebuah protokol tunggal, dengan *messaging* dan *request/response* hanya sebagai fitur dari *header* CoAP [18].



Gambar 2. 11 *Abstract Layering pada Protokol CoAP* [18]

Sama seperti protokol HTTP, protokol CoAP ini mengadopsi berbagai *pattern* dari protokol HTTP seperti pada *resource abstraction*, *URIs*, *RESTful interactions* dan *extended header*. Tetapi pada protokol CoAP ini menggunakan *binary* dalam melakukan representasinya, sementara dalam melakukan transimisi CoAP menggunakan protokol UDP, sehingga protokol CoAP ini memiliki sifat *multicast*, yaitu sebuah protokol yang dapat melakukan komunikasi *one-to-many*. [18]

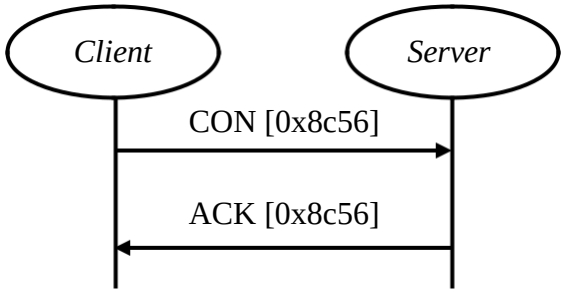
a. *Messaging Model*

Cara kerja protokol CoAP secara keseluruhan hampir sama dengan protokol HTTP. Pada protocol CoAP *layer message* mendukung 4 bagian, yaitu *Confirmable* (CON), *Non-Confirmable* (NON), *Acknowledgement* (ACK), dan *Reset* (RST), yang mana *method code* dan *response code* sudah termasuk didalamnya [18].

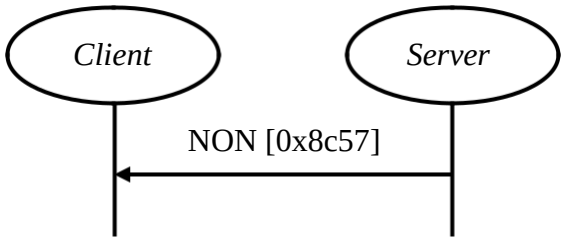
- Confirmable (CON), merupakan pesan yang berisi *request* dan memerlukan *Acknowledgement* (ACK).

- Non-Confirmable (NON), merupakan pesan yang didalamnya berisi *Acknowledgement* (ACK).
- *Acknowledgement* (ACK), merupakan pesan yang didalamnya berisi *response.*
- *Reset* (RST), merupakan pesan yang digunakan apabila pesan yang dikirimkan oleh CON tidak diterima dengan baik atau ada sebagian yang hilang.

Pesan yang dikirim oleh protokol CoAP memiliki dua bentuk pesan, yaitu *Reliability* dan yang kedua yaitu *Non-Realibility*. *Reliability* pada CoAP digunakan pada *Confirmable message*, yang mana pesan akan dipastikan sampai pada sisi penerima. *Confirmable message* akan terus dikirim secara berulang apabila terjadi *timeout* pada proses *messaging* hingga penerima mengirim ACK dengan *message ID* yang sama. Sementara itu *Non-Realibility* merupakan *message* yang tidak memerlukan *reliable* transmisi (Pada umumnya message ini digunakan untuk sensor). *Message* akan dikirim menggunakan *Non-confirmable message* (NON) yang mana tidak memiliki ACK, tetapi tetap memiliki *message ID*. [18]



Gambar 2. 6 *Reliable Message Transport*



Gambar 2. 12 *Unreliable Message Transport* [18]

Request/response yang ada pada protokol CoAP *message* bersama *method code* atau *response code*. *Request* dibawa oleh *Confirmable* (CON) atau *Non-Confirmable message*, sementara itu *response* dibawa oleh *Acknowledge message* (ACK) yang dikirim kembali pada sisi penerima. *Request Method* yang digunakan

protokol CoAP sama seperti protokol *HTTP*, yaitu GET, POST, PUT dan juga DELETE. [18]

- GET

  GET digunakan untuk melakukan permintaan data kepada web *server* dengan mengambil data yang diterima dari *response server*

- POST

  POST digunakan untuk melakukan pengiriman sebuah data ke *server* melalui sebuah *blank space.*

- PUT

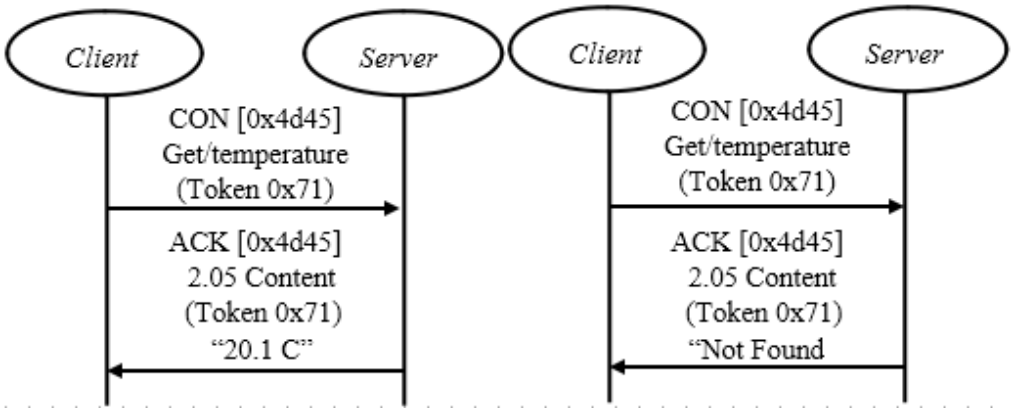
  Metode PUT ini hampir sama dengan metode POST, namun yang membedakannya yaitu, PUT ini sudah ditetapkan sesuai dengan URI sehingga dalam pengiriman sebuah data PUT sesuai atau asli dengan sumber URI-nya.

- DELETE

  Metode yang digunakan untuk menghapus data pada *server*

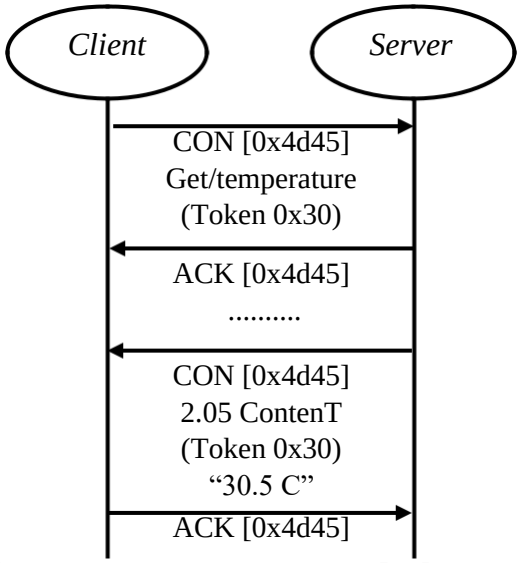
b. *Request/Response Model*

Respons dilakukan dalam sebuah tipe pesan *Confirmable* (CON) atau *Non-confirmable* (NON). Jika tersedia, *respons* terhadap *request* akan menghasilkan *Acknowledgement* (ACK). Ini disebut *respons* yang didukung atau disebut *piggybacked response.* Pada gambar berikut dua contoh untuk permintaan GET dengan *piggybacked response*. Satu berhasil menghasilkan respons dan satu lagi tidak menghasilkan *(Not found).* [18]



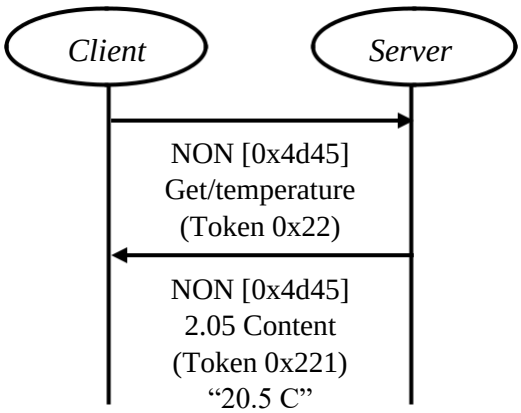
Gambar 2. 13 *Piggybacked Response* [18]

Jika *server* tidak merespon *request* secara langsung, tetapi *server* menerima sebuah *confirmable message* (CON), *server* akan mengirimkan pesan ACK kosong, maka *client* akan mengirimkan ulang kembali pesannya. Ketika *server* sudah siap mengirim CON baru kepada *client,* ACK akan dikirimkan oleh *client* sebagai bentuk konfirmasi dari *client.* Kasus tersebut dinamakan *separate response.* [18]



Gambar 2. 14 *Separate Response* [18]

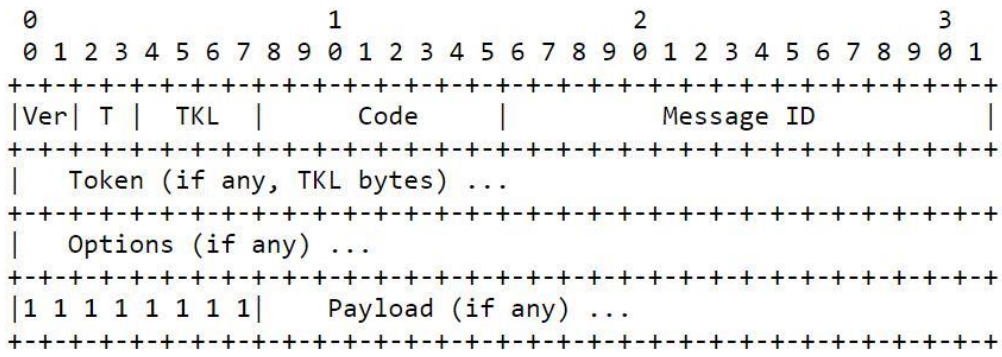
Model *request/response* selanjutnya merupakan kebalikan dari *piggybacked response,* yaitu *Non-confirmable request and response. Client* melakukan *request* dan mengirimkan pesan tipe NON, yang berarti *server* tidak perlu melakukan konfirmasi. Selanjutnya *server* akan mengirimkan pesan tipe NON bersama dengan *respon.* [18]



Gambar 2. 15 *Non-Confirmable Request and Response* [18]

c. *Message Format*

Model *messaging* atau pertukaran *message* pada protokol CoAP dilakukan melalui UDP. Pesan CoAP dikodekan dalam format biner sederhana, dengan format *message* yang digunakan yaitu *short fixed-length binary hearder* (4 *bytes*) kemudian diikuti oleh *compact binary options* dan muatanannya *(payload)* . Setiap pesan berisi message ID yang mendeteksi duplikasi dan untuk *reliability*. [18]



Gambar 2. 16 *Format Pesan CoAP* [18]

d. *Response Code*

Response code pada protokol CoAP terbagi menjadi tiga kelas atau bagian, yaitu *success 2.xx, Client Error 4.xx* dan *Server Error 5.xx.* Kode respon *Success* ditandai dengan kode 2.xx diindikasikan bahwa permintaan dari *client* sukses, dimengerti atau diterima oleh *server.* Kode *client error* ditandai dengan angka 4.xx, diindikasikan bahwa telah terjadi kekeliruan pada sisi *client.* Selanjutnya yaitu kode 5.xx yang muncul akibat *server error.* Kode – kode tersebut memiliki definisi masing – masing yang dapat dilihat pada tabel berikut. [18]

Tabel 2. 5 *Response Code Success* [18]

Response Code	***Definition***
2.01	*Created*
2.02	*Deleted*
2.03	*Valid*
2.04	*Changed*
2.05	*Conten*

Tabel 2. 6 *Respons Code Code Client Error* [18]

Response Code	***Definition***
4.00	*Bad Request*
4.01	*Unauthorized*
4.02	*Bad Option*
4.03	*Forbidden*
4.04	*Not Found*
4.05	*Method Not Allowed*
4.06	*Not Acceptable*
4.12	*Precondition Failed*
4.13	*Request Entity too Large*
4.15	*Unsupported Content-Format*

Tabel 2. 7 *Response Code Server Error* [18]

Response Code	***Definition***
5.00	*Internal Server Error*
5.01	*Not Implemented*
5.02	*Bad Gateway*
5.03	*Service Unavailable*
4.04	*Gateway Timeout*
5.05	*Proxiying Not Supported*

2.2.8 Quality of Service (QoS)

a. *Throughput*

Throughput merupakan kecepatan rata – rata data yang diterima oleh suatu *node* dalam waktu pengamatan tertentu. Atau juga merupakan *bandwidth* aktual ketika melakukan koneksi. Satuan dari *throughput* adalah *bit per second* (bps).

Rumus untuk melakukan perhitungan pada *throughput* adalah sebagai berikut [19**]**:

$$Throughtput = \frac{Jumlah\ data\ yang\ dikirim}{Waktu\ pengiriman\ data} \quad \text{.....(2.1)}$$

Berdasarkan penelitian ETSI LTN, besarnya throughput untuk M2M adalah<50 kbps.[20]

b. **Waktu Tunda *(Delay)***

Delay merupakan waktu tunda sebuah paket pada saat proses transmisi paket tersebut ke node satu ke node lainnya atau tujuan akhir paket tersebut. Delay direpresentasikan dalam satuan *second.*

Untuk melakukan perhitungan rata – rata delay, dapat digunakan rumus berikut [19] :

$$\text{Rata} - \text{rata } delay = \frac{\text{Total } delay}{\text{Total paket yang diterima}} \text{.....(2.2)}$$

Menurut *Eurocom Institute* dalam publikasi *LOLA Project,* besarnya waktu tunda *(delay)* pada *Machine to Machine* (M2M) adalah < 1290 ms. [21]

c. ***Packet Loss***

Packet Loss merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang dapat terjadi karena collision dan congestion pada jaringan. Persamaan perhitungannya dibawah ini [19].

$$Packet\ Loss = \frac{\text{Data yang dikirim} - \text{Paket data yang diterima}}{Paket\ data\ yang\ dikirim} \text{.....(2.3)}$$

2.2.9 WIRESHARK

Wireshark adalah salah satu dari sekian banyak *tool Network Analyzer* yang biasa digunakan oleh *Network Administrator* untuk mengontrol lalu lintas data di jaringan yang dkelola dan untuk menganalisa kinerja jaringan. *Software wireshark* memiliki interface yang menggunakan *Graphical User Interface* (GUI) sehingga banyak peminat yang menggunkannya. *Wireshark* juga banyak digunakan oleh professional untuk keperluan analisis, *troubleshooting*, pengembangan software dan protokol, dan juga unutk tujuan edukasi. *Wireshark* dapat menganalisa dan menangkap paket - paket data yang ada pada jaringan tersebut dengan segala jenis paket informasi dalam berbagai format protokol [8].

Fitur – fitur yang terdapat pada *wireshark* yaitu tersedia untuk *windows, unix, linux* dan *mac,* dapat mengcapture atau menangkap paket -paket data

secara langsung dari sebuah interface *network* dengan berbagai macam paket informasi dalam berbagai format protokol sehingga dapat menampilkan informasi yang detail mengenai hasil *capture* atau tangkapan tersebut, dapat mencari paket yang diinginkan dengan berbagai jenis filter, dan *wireshark* juga dapat menampilkan data statistik. Selain itu, *tool* pada *wireshark* dapat menganalisa transmisi paket data yang berada dalam jaringan serta proses koneksi dan trasnmisi data antar komputer. Namun pada *wireshark* juga terdapat kekurangan yaitu pada OS *windows* yang tidak terdeteksi *driver wireless* karena pada *library wincap* tidak dapat mendeteksinya [8].