

## BAB II

### DASAR TEORI

#### 2.1 Kajian Pustaka

Penelitian Teguh Abdul Latif pada tahun 2017 yang berjudul “Pengendalian Proyektor LCD dari Jarak Jauh dengan *Wi-Fi* dan *Raspberry Pi*” meneliti mengenai cara membuat sistem pengendali proyektor LCD dari jarak jauh pada ruang kelas di lingkungan Fakultas Komunikasi dan Informatika pada Universitas Muhammadiyah Surakarta dengan memanfaatkan *Wi-Fi*. Alat yang digunakan memakai *Raspberry Pi* sebagai *embedded controller*, *Infra Red (IR) receiver*, *Infra Red (IR) transmitter*, dan *smartphone android*. *Raspberry Pi* dan *smartphone android* dikoneksikan dengan *Wi-Fi*, sehingga halaman *web* yang ada di *web server Raspberry Pi* itu dapat diakses melalui *web browser* yang tersedia di *smartphone*[2].

Penelitian Adiet Puji Setiawan pada tahun 2017 yang berjudul “Sistem Monitoring Kontrol Penyejuk Udara dan Lampu Dengan Konsep *Internet of Things (IoT)* Menggunakan *Arduino Mega Berbasis Web*” meneliti tentang bagaimana cara membuat sistem monitoring penyejuk udara dan lampu pada ruang kelas dengan konsep *Internet of Things (IoT)*. Perangkat ini terdiri dari mikrokontroler *Arduino mega 2560*, *Ethernet shield*, sensor *PIR*, sensor *DHT11*, *light sensor BH1750*, dan rangkaian *relay*. Alat ini mempunyai *web server* yang berfungsi untuk memberikan perintah kepada sensor-sensor tersebut untuk melakukan eksekusi. Sistem ini akan bekerja sesuai jadwal yang sudah diatur pada *web server* atau sesuai keadaan lingkungan jika ada pergerakan keadaan suhu ruangan  $< 27^{\circ} \text{C}$  dan  $\text{lux} < 50 \text{ lx}$ , maka penyejuk udara dan lampu akan otomatis menyala. Pengguna dapat mengontrol kondisi penyejuk ruangan dan lampu melalui *website* yang telah dibuat melalui *web server*[3].

Penelitian oleh Aliffudin Ilham Ata pada tahun 2017 yang berjudul “Desain dan Implementasi Otomatisasi Pendingin Udara Berbasis *Arduino<sup>TM</sup>*” yang meneliti tentang bagaimana merancang dan mengimplementasikan sebuah alat untuk mengontrol suhu ruangan secara otomatis dimana hasil pembacaan suhu ruangan dapat ditampilkan dan diatur pada layar *LCD 16x2* dan pada *box* alat pengatur suhu ruangan. Komponen yang digunakan antara lain *Arduino mega 2560*

sebagai mikrokontroler, sensor suhu LM35, Infra Merah, *Liquid Crystal Display* (LCD), dan juga *push button*. Sensor LM35 mendeteksi suhu ruangan jika suhu ruangan lebih dari 28°C maka AC akan menyala secara otomatis. Untuk mengatur kenyamanan suhu ruangan dilakukan secara manual dengan menekan *push button* dan jika suhu kurang dari sama dengan 20°C maka AC akan *non-aktif* secara otomatis[4].

Penelitian selanjutnya oleh Dimas Arya Nugraha pada tahun 2019 yang berjudul “Monitoring Suhu dan Kelembaban Ruang *Server* Berbasis *Internet of Things* dengan Notifikasi Aplikasi *Android*” meneliti mengenai cara merancang suatu perangkat yang dapat memonitoring suhu dan kelembaban ruang *server* secara *realtime* menggunakan satu buah sensor suhu DHT11, *microcontroller NodeMCU ESP8266*, *MIT App Inventor* sebagai media pembuatan aplikasi, dan *firebase* sebagai *database*. Aplikasi *android* akan menampilkan data suhu dan kelembaban yang terbaca pada sensor suhu DHT11. Saat suhu yang terbaca melebihi 25°C maka disaat yang bersamaan *relay* dalam kondisi *on* dan akan menyalakan kipas untuk mendinginkan suhu ruangan *server*. Aplikasi pada *android* akan mengirimkan notifikasi secara terus menerus sampai suhu yang terbaca pada sensor DHT11 kurang dari 25°C. Notifikasi pada *android* berupa suara dan pemberitahuan pada layar *smartphone*. [5]

Penelitian M.Natsir, Dwi Bayu Rendra, dan Acep Derby Yudha Anggara pada tahun 2019 yang berjudul “Implementasi IoT untuk Sistem Kendali AC Otomatis Pada Ruang Kelas di Universitas Serang Raya” meneliti mengenai perancangan suatu alat yang mampu memantau dan mengendalikan suhu dan juga kelembaban udara ruangan berbasis *Internet Of Things* (IoT). Perangkat yang digunakan antara lain *Arduino mega 2560*, sensor *Infra Red (IR)*, sensor suhu *DS18B20*, *Ethernet shield W5500*, *Buzzer*, *Liquid Crystal Display (LCD)*. Alat ini bekerja ketika mahasiswa masuk ke ruang kelas dan sensor ultrasonik akan menghitung jumlah orang yang masuk sebagai *inputan* yang kemudian akan diproses oleh mikrokontroler *ATmega 2560* untuk men-*set* suhu ruang kelas yaitu *set* suhu biasa, *set* suhu sedang, dan *set* suhu tinggi sesuai dengan pembacaan suhu ruang kelas. Hasil *inputan* dari sensor suhu dan sensor ultrasonik akan diproses oleh mikrokontroler *ATmega 2560* yang akan mengambil keputusan untuk men-*set* suhu

ruang kelas. Setelah data diproses mikrokontroler akan mengirimkan sinyal IR untuk mengontrol AC [6].

## 2.2 Dasar Teori

### 2.2.1 *Internet of Things*

*Internet Of Things* (IoT) memungkinkan pengguna untuk mengelola dan mengoptimalkan elektronik dan peralatan listrik yang menggunakan internet. Hal ini berspekulasi bahwa disebagian waktu dekat komunikasi antara komputer dan peralatan elektronik mampu bertukar informasi diantara mereka sehingga mengurangi interaksi manusia. *Internet of Things* dalam penerapannya juga dapat mengidentifikasi, menemukan, melacak, memantau objek serta memicu *event* terkait secara otomatis dan *real time* [7].

*Internet of Things* menggunakan beberapa teknologi yang secara garis besar digabungkan menjadi satu kesatuan diantaranya sensor sebagai pembaca data, koneksi internet dengan bebarapa macam topologi jaringan, *radio frequency identification* (RFID), *wireless sensor network*, dan teknologi yang terus akan bertambah sesuai dengan kebutuhan [8].



Gambar 2. 1 Konsep *Internet of Things* [8]

### 2.2.2 NodeMCU ESP8266

*NodeMCU* pada dasarnya adalah pengembangan dari ESP8266 dengan *firmware* berbasis *e-Lua*. Pada *NodeMCU* dilengkapi dengan *micro usb port* yang berfungsi untuk pemrograman maupun *power supply* [9]. *NodeMCU* juga dilengkapi dengan tombol *push button* yaitu tombol *reset* dan *flash*, serta dilengkapi dengan modul ESP8266 yang merupakan sebuah *chip* yang digunakan sebagai pengiriman

data dari sebuah mikrokontroler menuju sebuah komputer *server* menggunakan media *wireless*. Menyediakan akses jaringan *Wi-Fi* secara transparan menggunakan interkoneksi serial ( UART. RX. TX ) [10].

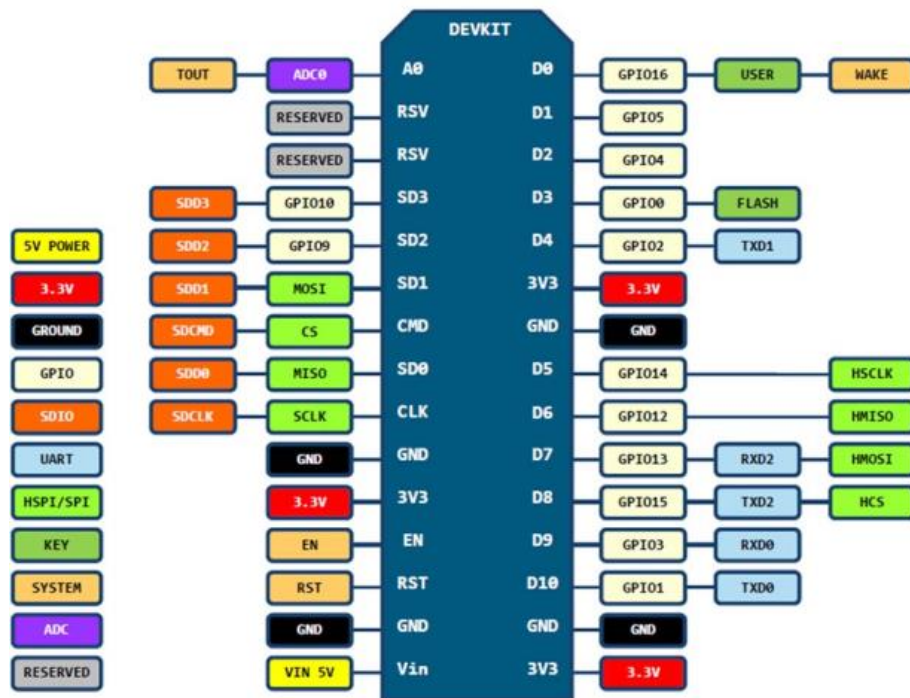


Gambar 2. 2 ESP8266 *NodeMCU* [8]

*NodeMCU* menggunakan bahasa pemrograman *Lua* yang merupakan *package* dari ESP8266. Bahasa *Lua* memiliki logika dan susunan pemrograman yang sama dengan *c* hanya berbeda *syntax*. Jika menggunakan bahasa *Lua* maka dapat menggunakan *tool Lua loader* maupun *Lua uploder*. Dibawah ini spesifikasi dari *NodeMCU V3* [9] :

Tabel 2. 1 Spesifikasi *NODEMCU V3*[8]

SPESIFIKASI	NODEMCU V3
Mikrokontroler	ESP8266
Ukuran Board	57mm x 30 mm
Tegangan Input	3.3 ~ 5V
GPIO	13 PIN
Kanal PWM	10 Kanal
10 bit ADC Pin	1 Pin
Flash Memory	4 MB
Clock Speed	10/26/24 MHz
Wi-Fi	IEEE 802.11 b/g/n
Frekuensi	2.4 Ghz – 22.5 Ghz
USB Port	Micro USB
Card Reader	Tidak Ada
USB to Serial Converter	CH340G



Gambar 2. 3 Skematik Posisi Pin *NodeMcu Dev Kit v3*[9]

### 2.2.3 Sensor DHT22

Sensor DHT22 adalah sensor digital yang dapat mengukur suhu  $-40^{\circ}\text{C}$  –  $125^{\circ}\text{C}$  dan kelembaban udara 0% -100% di sekitarnya [11]. DHT22 ini merupakan sensor pengukur suhu dan kelembaban yang memiliki 4 pin yang terdiri dari *power supply*, *data signal*, *null*, dan *ground*. Sensor DHT22 Memiliki tingkat stabilitas yang sangat baik serta fitur kalibrasi yang sangat akurat dengan galat relatif pengukuran suhu 4% dan kelembaban 18% [12]:



Gambar 2. 4 Sensor DHT22[12]

Tabel 2. 2 Spesifikasi Sensor DHT22 [13]

Model	AM2303
Power Supply	3.3 – 6V DC
Output Signal	Digital signal via single-bus
Sensing element	Polymer humidity capacitor & DS18B20 for detecting temperature
Measuring range	Humidity 0-100%RH; temperature -40~125Celcius
Accuracy	Humidity +-2%RH(Max +-5%RH); temperature +-0.2Celcius
Resolution or sensitivity	Humidity 0.1RH; temperature 0.1 Celcius
Repeatability	Humidity +-1%RH; temperature +-0.2 Celcius
Humidity hysteresis	+ - 0.3%RH
Long-term Stability	+ -0.5%RH / year
Sensing Period	Average : 2s
Interchangeability	Fully interchangeable

Tabel 2. 3 Karakteristik Listrik Sensor DHT22 [13]

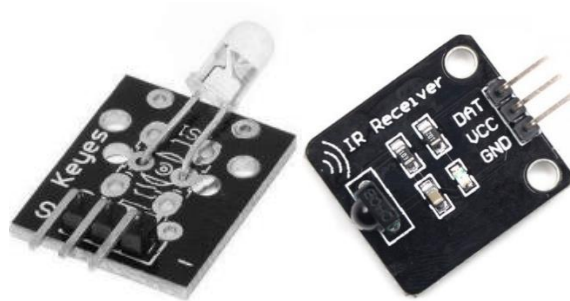
Item	Condition	Min	Typical	Max	Unit
Power supply	DC	3.3	5	5.5	V
Current Supply	Measuring	1.3	1.5	2.1	mA
Collecting period	Second	1.7		2	Second

#### 2.2.4 *Infra Red (IR)*

Cahaya infra merah merupakan cahaya yang tidak tampak. Jika dilihat dengan spektroskop cahaya maka radiasi cahaya infra merah akan terlihat pada spektrum elektromagnet dengan panjang gelombang di atas panjang gelombang

cahaya merah. Radiasi inframerah memiliki panjang gelombang antara 700 nm sampai 1 mm dan berada pada spektrum berwarna merah. Dengan panjang gelombang ini maka cahaya infra merah tidak akan terlihat oleh mata namun radiasi panas yang ditimbulkannya masih dapat dirasakan atau dideteksi.

Sinar infra merah yang dipancarkan oleh pemancar infra merah mempunyai aturan tertentu agar data yang dipancarkan dapat diterima dengan baik pada penerima. Oleh karena itu baik di IR *transmitter* maupun IR *receiver* harus mempunyai aturan yang sama dalam mentransmisikan (bagian pengirim) dan menerima sinyal tersebut kemudian mendekodekannya kembali menjadi data biner (bagian penerima)[6].



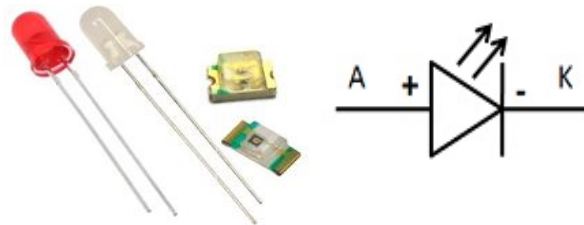
Gambar 2. 5 IR *transmitter* dan IR *receiver* [6]

### 2.2.5 LED (*Light Emitting Diode*)

*Light Emitting Diode* atau sering disingkat dengan LED adalah komponen elektronika yang dapat memancarkan cahaya monokromatik ketika diberikan tegangan maju. LED merupakan keluarga dioda yang terbuat dari bahan semikonduktor. Warna-warna cahaya yang dipancarkan oleh LED tergantung pada jenis bahan semikonduktor yang dipergunakannya. LED juga dapat memancarkan sinar inframerah yang tidak tampak oleh mata seperti yang sering dijumpai pada *Remote Control* TV ataupun *Remote Control* perangkat elektronik lainnya.

LED terdiri dari sebuah *chip* semikonduktor yang *didoping* sehingga menciptakan *junction* P (positif) dan N (negatif). Proses *doping* dalam semikonduktor adalah proses untuk menambahkan ketidakmurnian (*impurity*) pada semikonduktor yang murni sehingga menghasilkan karakteristik kelistrikan yang diinginkan. Ketika LED dialiri tegangan maju atau bias *forward* yaitu dari Anoda (P) menuju ke Katoda (K), maka kelebihan Elektron pada N-Type material akan

berpindah ke wilayah yang kelebihan *hole* (lubang) yaitu wilayah yang bermuatan positif (P-Type material). Saat Elektron berjumpa dengan *hole* akan melepaskan *photon* dan memancarkan cahaya monokromatik (satu warna). [14]

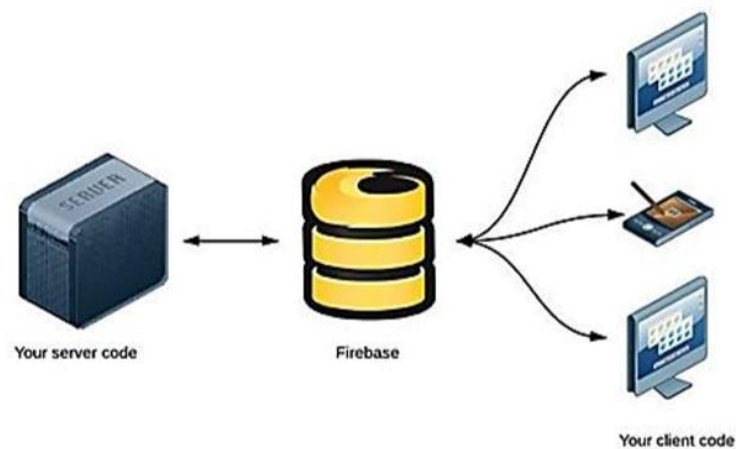


Gambar 2. 6 Bentuk dan Simbol LED[14]

### 2.2.6 *FireBase*

*Firebase* adalah penyedia layanan *realtime database* dan *backend* sebagai layanan. Suatu aplikasi yang memungkinkan pengembang membuat API untuk disinkronisasikan untuk *client* yang berbeda-beda dan disimpan pada *cloud*-nya.

*Firebase* memiliki banyak *library* yang memungkinkan untuk mengintegrasikan layanan ini dengan *android*, *Ios*, *Javacript*, *Java*, *Objective-C* dan *Node.JS*. *database firebase* juga bersifat dapat diakses lewat REST API. REST API tersebut menggunakan protokol *Server-Sent Event* dengan membuat koneksi HTTP untuk menerima *push notification* dari *server*. Pengembang menggunakan REST API untuk *post* data yang selanjutnya *firebase client library* yang sudah diterapkan pada aplikasi yang dibangun akan mengambil data secara *realtime* [15].



Gambar 2. 7 Arsitektur Sistem *Firebase* [15]



### 2.2.7 *App Inventor*

*App inventor* merupakan aplikasi *web* sumber terbuka yang awalnya dikembangkan oleh *Google* dan saat ini dikelola oleh *Massachusetts Institute of Technology (MIT)*. *App Inventor* memungkinkan pengguna baru untuk memprogram komputer untuk menciptakan aplikasi perangkat lunak bagi sistem operasi *android*.

*App inventor* juga sering disebut *visual block programming* karena akan dapat melihat, menggunakan, menyusun, dan *mendrag-drops* blok yang merupakan simbol-simbol perintah dan fungsi *even handler* tertentu dalam membuat aplikasi dengan cara sederhana dapat menyebutnya tanpa menuliskan kode program atau *coding less*.

Pada *app inventor* ini terdapat beberapa komponen yang terdiri dari:

- (1) Komponen desainer yang berjalan pada *browser* digunakan untuk memilih komponen yang diperlukan untuk mengatur propertinya. Pada komponen desainer sendiri terdapat 5 bagian, yaitu : *palette, viewer, component, media dan properties*.
- (2) *Block Editor* berjalan di luar *browser* dan digunakan untuk membuat serta mengatur *behaviour* dari komponen-komponen yang akan kita pilih dari komponen desainer.
- (3) *Emulator* yang digunakan untuk menjalankan dan menguji *project* yang telah dibuat [16].

### 2.2.8 *Wireshark*

*Wireshark* merupakan sebuah aplikasi perangkat lunak (*software*) yang digunakan untuk dapat melihat dan mencoba menangkap paket-paket jaringan dan berusaha untuk menampilkan semua informasi di paket tersebut serinci mungkin. Salah satu dari sekian banyak *tool Network Analyzer* yang banyak digunakan oleh *network administrator* untuk menganalisa kinerja jaringannya. *Wireshark* banyak disukai karena *interfacenya* yang menggunakan *Graphical User Interface (GUI)* atau tampilan grafis [17].

*Wireshark* dapat menganalisis paket data secara *real time* yang artinya aplikasi *wireshark* ini akan mengawasi semua paket data yang keluar masuk melalui antar muka yang telah di tentukan oleh *user* sebelumnya. Aplikasi *wireshark* akan

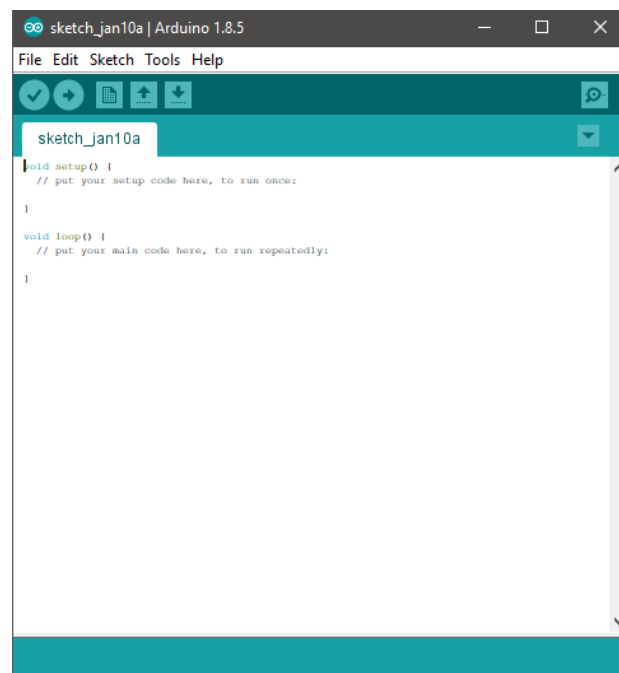
menampilkan banyak sekali paket data dan menimbulkan kebingungan karena ada begitu banyak paket data jaringan yang muncul jika komputer terhubung dengan jaringan kecepatan tinggi dan pada komputer sedang digunakan aplikasi berbasis jaringan,. Aplikasi *wireshark* dapat memfilter jenis *protocol* tertentu yang ingin ditampilkan [18].

### 2.2.9 Arduino IDE (*Integrated Development Environment*)

Arduino IDE (*Integrated Development Environment*) adalah program khusus yang digunakan untuk menulis *sketch* untuk arduino dengan bahasa yang sederhana dalam penulisan *sketch* menggunakan bahasa pemrograman C.

Arduino IDE (*Integrated Development Environment*) merupakan editor teks untuk menulis kode, sebuah pesan, konsol teks. Arduino IDE menghubungkan arduino dan *hardware* untuk meng-*upload* program dan berkomunikasi dengan alat elektronik.

Arduino IDE ini berguna sebagai *text editor* untuk membuat, mengedit, dan *mevalidasi* kode program. Bisa juga digunakan untuk meng-*upload* ke *board* arduino. Kode program yang digunakan pada arduino disebut dengan istilah arduino “*sketch*” atau disebut juga *source code* arduino dengan ekstensi *file source code.ino* [9]



Gambar 2. 8 Arduino IDE (*Integrated Development Environment*)[9]

### 2.2.10 Quality of Service

*Quality of Service (QoS)* adalah kemampuan suatu jaringan untuk menyediakan layanan yang baik dengan menyediakan *bandwith*, mengatasi *jitter* dan *delay*. *QoS* sangat ditentukan oleh kualitas jaringan yang digunakan. Terdapat beberapa faktor yang dapat menurunkan nilai *QoS* seperti redaman, *distorsi*, dan *noise*. [19]

Tujuan dari mekanisme *QoS* adalah mempengaruhi setidaknya satu diantara empat parameter dasar *QoS* yang telah ditentukan. *QoS* didesain untuk membantu *end user (client)* menjadi lebih produktif dengan memastikan bahwa *user* mendapatkan performansi yang handal dari aplikasi-aplikasi berbasis jaringan. *QoS* mengacu pada kemampuan jaringan untuk menyediakan layanan yang lebih baik pada trafik jaringan tertentu melalui teknologi yang berbeda-beda. [20] Parameter yang digunakan dalam mengukur *QoS* sebagai berikut:

- a. *Delay (Latency)* adalah waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. *Delay* dapat dipengaruhi oleh jarak, media fisik, kongesti atau juga waktu proses yang lama. Rata-rata *delay* dapat dicari dengan :

$$\text{Rata - rata delay} = \frac{\sum \text{delay}}{\sum \text{paket diterima}}$$

- b. *Throughput* yaitu kecepatan transfer data efektif yang diukur dalam bps. *Throughput* merupakan jumlah total kedatangan paket yang sukses diamati pada tujuan selama interval waktu tertentu dibagi oleh durasi waktu interval tersebut.

$$\text{Throughput} = \frac{\text{Paket data yang diterima}}{\text{lama pengamatan}}$$

- c. *Packet loss* merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang dapat terjadi karena *collision* dan *congestion* pada jaringan. Hal ini berpengaruh pada

semua aplikasi karena retransmisi akan mengurangi efisiensi jaringan secara keseluruhan meskipun jumlah *bandwidth* cukup tersedia untuk aplikasi tersebut.[19]

$$Packet\ loss = \frac{(\text{paket data dikirim} - \text{paket data diterima})}{\text{paket data dikirim}} \times 100\%$$