

BAB II

TINJAUAN PUSTAKA

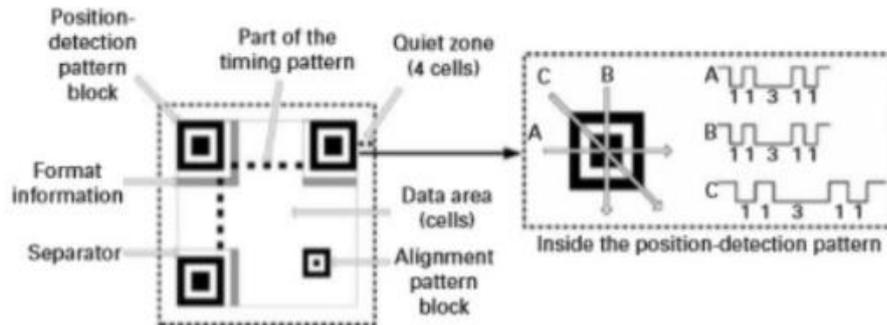
2.1 Presensi

Presensi adalah kegiatan pengambilan data guna mengetahui jumlah kehadiran seseorang dalam suatu instansi, institusi atau perusahaan [1]. Presensi merupakan kehadiran suatu objek dalam hal ini adalah orang, dimana orang tersebut terlibat dalam suatu organisasi yang mengharuskan adanya pemberitahuan tentang keadaan atau kehadiran atau ketidakhadirannya dalam ruang lingkup organisasi tersebut [9]. Presensi kehadiran bagi siswa maupun mahasiswa merupakan salah satu indikator yang penting dalam proses belajar mengajar baik di sekolah umum maupun di perguruan tinggi. Presensi mahasiswa menunjukkan seberapa aktif mahasiswa tersebut dalam mengikuti perkuliahan. Presensi juga dijadikan sebagai salah satu poin penilaian akhir mahasiswa dalam proses belajar mengajar tersebut [1].

2.2 *Quick Response (QR) Code*

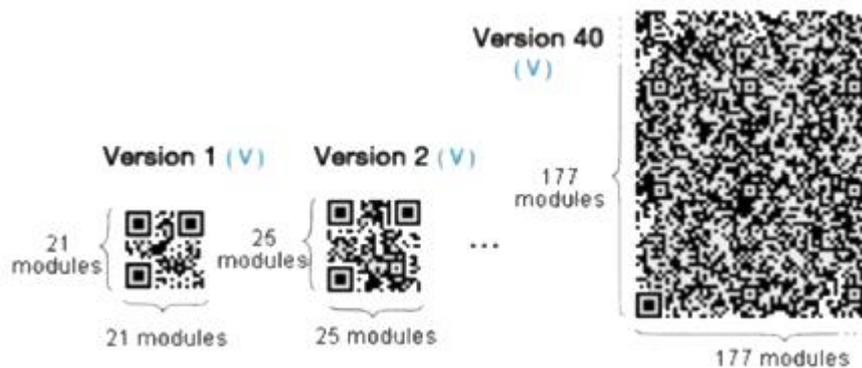
QR merupakan *barcode* dua dimensi berbentuk matrik yang dikembangkan oleh Denso Wave pada tahun 1994 [10]. QR bertujuan untuk menyampaikan informasi dengan cepat dan mendapatkan respon yang cepat pula. Berbeda dengan *barcode*, yang hanya menyimpan informasi secara horizontal, QR *code* dapat menyimpan informasi secara horizontal dan vertical. Oleh karena itu, secara otomatis QR *code* dapat menyimpan informasi lebih dari *barcode* [2]. QR *code* merupakan susunan matrik dua dimensi yang memiliki pola deteksi tiga sudut. QR *code* dikembangkan dengan tujuan untuk meningkatkan kecepatan dan kemudahan untuk dibaca oleh peralatan *scanner* serta meningkatkan kapasitas penyimpanan data.

Secara umum bentuk dari QR *code* adalah persegi dengan tiga *position detection pattern* yang merupakan salah satu fitur khas yang dimiliki QR *code*. Dalam proses pembacaan data QR *code* ketiga *pattern detection* inilah yang pertama kali di baca oleh alat pembaca. Struktur dari QR *code* dapat dilihat pada gambar 2.1.



Gambar 2.1 Struktur QR Code [10]

QR code memiliki beberapa versi dimulai dari versi satu (1) sampai dengan versi empat puluh (40). Semakin tinggi versi QR code maka semakin banyak modul yang terdapat dalam sebuah QR code sehingga berpengaruh pada besarnya kapasitas penyimpanan dari QR code tersebut. Perbandingan modul QR code dapat dilihat pada gambar 2.2.



Gambar 2.2 Perbandingan Modul QR Code [10]

Dalam proses pembuatan QR code kurang lebih terdapat tujuh (7) tahap yang harus dilakukan. Berikut merupakan contoh dari proses *encoding* dengan data *alphanumeric* “HELLO” dan versi 1 (ECC level M) [10]:

1. Data analisis

Melakukan pemilihan mode *indicator* yang sesuai dengan data, hal ini digunakan untuk menunjukkan jenis data dalam QR code. Jenis mode *indicator* dalam QR code dapat dilihat pada tabel 2.1:

Tabel 2.1 Jenis Mode *Indicator* [10]

Mode	Indicator
ECI	0111
<i>Numeric</i>	0001
<i>Alphanumeric</i>	0010
8-bit Byte	0100
Kanji	1000
<i>Structure Append</i>	001
FNCI	0101 (<i>First Position</i>) 1001 (<i>Second Positon</i>)
<i>Terminator (End of Message)</i>	0000

Menambahkan *character count indicator* yang sesuai dengan data dalam QR code. Banyaknya bit yang digunakan untuk *character count indicator* dapat dilihat pada tabel 2.2:

Tabel 2.2 Indikator Jumlah Karakter [10]

Character Count Indicator	Version		
	1-9	10-26	27-40
<i>Numeric</i>	10 bit	12 bit	14 bit
<i>Alphanumeric</i>	9 bit	11 bit	13 bit
<i>Binary</i>	8 bit	16 bit	16 bit
<i>Japanese</i>	8 bit	10 bit	12 bit

Berdasarkan data yang telah di input adalah *alphanumeric* sejumlah lima karakter maka di dapatkan *string* biner 0010000000101.

2. Pengkodean data

Dalam proses *encoding* dengan tipe data *alphanumeric* memiliki beberapa aturan diantaranya:

- a. Mengelompokkan huruf dan angka yang berpasangan.
- b. Kemudian lakukan perhitungan (45 x Nilai Karakter) + Nilai Karakter 2 dan ubah menjadi biner. Data dari nilai karakter dapat dilihat pada tabel 2.3:

Tabel 2.3 Kode *Alphanumeric* [10]

<i>Char</i>	<i>Value</i>	<i>Char</i>	<i>Value</i>
0	0	N	23
1	1	O	24
2	2	P	25
3	3	Q	26
4	4	R	27
5	5	S	28
6	6	T	29
7	7	U	30
8	8	V	31
9	9	W	32
A	10	X	33
B	11	Y	34
C	12	Z	35
D	13	SP	36
E	14	\$	37
F	15	%	38
G	16	*	39
H	17	+	40
I	18	-	41
J	19	_	42
K	20	/	43
L	21	:	44
M	22		

- c. Jika terdapat karakter yang tidak memiliki pasangan ubah nilai yang diperoleh dari tabel dan ubah dalam bentuk biner.

Proses *encoding* data “HELLO” dapat dilihat pada tabel 2.4:

Tabel 2.4 Proses *Encoding* data “HELLO” [10]

HE	LL	O
$(45 \cdot 17) + 14$ 770 01100001011	$(45 \cdot 21) + 21$ 966 01111000110	24 24 011000
<i>String Biner</i>	0010 000000101 01100001011 01111000110 011000	

Selanjutnya melakukan penambahan *terminator mode indicator* di belakang barisan dan kelompokkan menjadi barisan biner 8 bit, sehingga menjadi seperti berikut:

```
00100000 00101011 00001011 0111100011001100 00000.
```

Karena pada barisan terakhir tidak genap 8 bit, maka tambahkan *padding* berupa bit “0” hingga barisan tersebut berjumlah 8 bit, seperti berikut:

```
00100000 00101011 00001011 01111000 11001100 00000000.
```

Kemudian karena QR code versi 1 dengan *error correction* M memiliki data *codeword* (barisan data sejumlah 8 bit) sejumlah 9, maka perlu ditambahkan *padding*. Untuk *codeword* pada QR code menggunakan *padding* bernilai “11101100” dan “00010001” yang digunakan berselang-seling, sehingga didapat nilai biner sebagai berikut:

```
00100000 00101011 00001011 01111000 11001100 00000000 11101100
0001000 11110110.
```

3. Pengkodean *error correction*

Error correcting merupakan salah satu fitur yang dimiliki QR code untuk memperbaiki data yang salah atau rusak. Pada bagian ini akan ditambahkan barisan bit-bit *error correction* menggunakan metode Reed Solomon. Secara singkat selesai melewati tahap ini akan diperoleh *string* biner sebagai berikut:

```
00100000 00101011 00001011 00111100 01100110
00000000 11101100 00010001 11101100 01101101
10010101 10011100 00010010 11011001 00101001
11110110 00100100 00101010 01010100 00101110
11100001 10111110 11011010 11111011 00011011
11000100
```

4. Struktur akhir pesan

Tahap ini akan dikelompokkan string biner yang didapat sesuai dengan jumlah *block* yang berisi data *codeword* dan *error correction codeword*. Hasil pengelompokan data *codeword* dan *error correction codeword* dapat dilihat pada tabel 2.5:

Tabel 2.5 Pengelompokkan Data *Codeword* dan *Error Correction Codeword* [10]

<i>Data Codeword</i>		<i>Error Correction Codeword</i>			
ID	DC	ID	ECC	ID	ECC
D1	00100000	E1	01101101	E10	01010100
D2	00101011	E2	10010101	E11	00101110
D3	00001011	E3	10011100	E12	11100001
D4	01111000	E4	00010010	E13	10111110
D5	11001100	E5	11011001	E14	11011010
D6	00000000	E6	00101001	E15	11111011
D7	11101100	E7	11110110	E16	00011011
D8	00010001	E8	00100100	E17	11000100
D9	11101100	E9	00101010		

5. Penempatan modul pada matrik

Proses berikutnya adalah pembentukan *raw* pada QR *code* berdasarkan data yang telah ada sebelumnya. Pada pembentukan *raw* ini data akan diwakilkan dengan dua warna yaitu hitam dan putih dimana untuk bit “0” diwakili oleh warna putih sedangkan bit “1” diwakili oleh warna hitam. Hasil dari penempatan modul pada matrik dapat dilihat pada gambar 2.3.

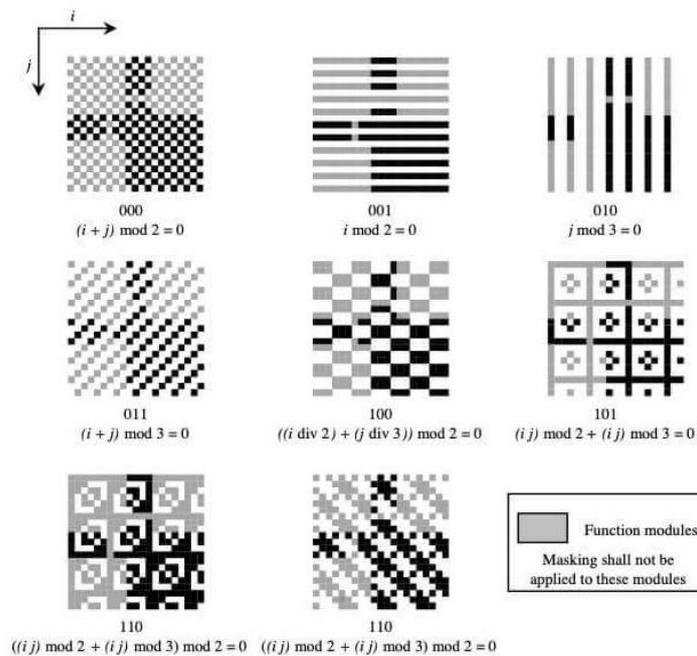
Gambar 2.3 Hasil Modul *Placement* in *Matrix* [10]

6. *Masking*

Tahap ini akan dilakukan proses *masking* yang bertujuan agar posisi hitam dan putih seimbang, sehingga memudahkan *reader* dalam melakukan pembacaan. Tabel aturan untuk *mask pattern* dapat dilihat pada tabel 2.6 dan ilustrasi *masking* pada QR *code* dapat dilihat pada gambar 2.4.

Tabel 2.6 Referensi Pola *Masking* [10]

<i>Mask Pattern Reference</i>	<i>Condition</i>
000	$(i+j) \bmod 2 = 0$
001	$i \bmod 2 = 0$
010	$j \bmod 3 = 0$
011	$(i+j) \bmod 3 = 0$
100	$((i \div 2) + (j \div 3)) \bmod 2 = 0$
101	$(ij) \bmod 2 + (ij) \bmod 3 = 0$
110	$((ij) \bmod 2 + (ij) \bmod 3) = 0$
111	$((ij) \bmod 3 + (ij) \bmod 2) = 0$

Gambar 2.4 Ilustrasi *Masking* pada QR Code Versi 1 [10]

7. Peletakan format serta informasi versi.

Format information merupakan deretan bit dengan jumlah 15 yang berisikan informasi mengenai *error correcting* yang digunakan dan *mask pattern*. 15 dalam format *information* menyatakan 5 bit data dan 10 bit *error correcting* dengan BCH code.

Tabel 2.7 Indikator *Error Correction* [10]

<i>Error Correcting Level</i>	<i>Binary Indicator</i>
L	01
M	00
Q	11
H	10

Dalam hal ini akan digunakan *error correcting level* H dengan *aprox.*15% dan *masking* 000, maka dapat dinyatakan dalam format *information* sebagai berikut:

- a. Melakukan Perhitungan BCH

$$F(x) = x^{14}$$

$$G(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$$

$$\text{Remainder} = F(x)/G(x)$$

$$R(x) = x^9 + x^7 + x^4 + x^3 + x + 1$$

- b. *Unmasked Format Information* : 100001010011011
 c. *Masked Pattern Format Information* : 101010000010010
 d. Hasil XOR (*Format Information*) : 001011010001001

8. Tahap Akhir Pembuatan QR Code

Berikut merupakan hasil pembuatan QR code versi 1 dengan data HELLO yang dapat dilihat pada gambar 2.5.



Gambar 2.5 Hasil QR Code Versi 1 dengan Data HELLO [10]

Berikut beberapa fitur yang dimiliki oleh QR code [10]:

1. Mendukung berbagai jenis data

QR code mampu menampung banyak jenis data diantaranya data dengan bentuk *numeric*, *alphanumeric*, *byte*, kanji dan kana.

2. Memiliki daya tampung yang besar

QR code memiliki daya tampung yang besar yang dapat dilihat pada tabel 2.8 berikut ini:

Tabel 2.8 Kapasitas Penyimpanan Data pada QR Code [10]

Kapasitas Data pada QR Code	
<i>Numeric</i>	Maksimal 7089 karakter
<i>Alphanumeric</i>	Maksimal 4296 karakter
<i>Binary (8 bit)</i>	Maksimal 2953 karakter
Kanji dan Kana	Maksimal 1817 karakter

3. Pembacaan dapat dilakukan dari segala arah

QR code memiliki *position detection pattern* pada ketiga ujungnya sebagai patokan, dengan adanya *position detection pattern* dimungkinkan pembacaan dari segala sudut.

4. Tahan terhadap kotoran dan kerusakan

Dengan menggunakan fitur *error correcting* yang dimiliki oleh QR code, dimungkinkan QR code mampu memperbaiki data yang salah atau rusak pada saat pembacaan dengan menggunakan bantuan metode Reed Solomon. QR code memiliki empat level *error correcting* yaitu level L, M, Q dan H yang masing-masing memiliki kemampuan berbeda dalam memperbaiki data. Data level *error correcting* yang dimiliki QR code dapat dilihat pada tabel 2.9:

Tabel 2.9 Level Error Correction [10]

ECC Level	Recovery Capacity % (Approx)	Data Translation Ratio (cell needed to encode)
L	7	3.3 cell/character
M	15	5.5 cell/character
Q	25	8 cell/character
H	30	13 cell/character

2.3 International Mobile Equipment Identity (IMEI)

IMEI merupakan nomor identitas khusus bersifat unik yang dimiliki oleh setiap perangkat mobile. IMEI terdiri dari 15 hingga 16 digit nomor yang dihasilkan dari delapan digit *Type Allocation Code* yang dikeluarkan oleh *Global System for Mobile Association* (GSMA) untuk tiap slot kartu *Subscriber Identity Module* (SIM) yang berguna untuk mengidentifikasi alat dan/atau perangkat telekomunikasi yang tersambung ke jaringan bergerak seluler [11][12].

2.4 Internet Protocol Address (IP Address)

IP address adalah metode pengalamatan unik yang diberikan pada peralatan jaringan yang terkoneksi ke jaringan berbasis TCP/IP. Tiap-tiap komputer atau peralatan jaringan akan memiliki alamat IP yang unik dan masing-masing berbeda dengan yang lainnya. Hal tersebut untuk mencegah terjadinya kesalahan pada saat transfer data ketika berlangsung [13]. *IP address* merupakan deretan angka biner antara 32 bit sampai dengan 128 bit yang digunakan sebagai alamat identifikasi untuk tiap komputer *host* dalam jaringan internet [14].

2.4.1 Jenis IP Address

Berikut merupakan jenis-jenis dari *IP address* yang dibagi menjadi dua bagian, yaitu:

1. IPv4 (*IP Address* Versi 4)

IPv4 terdiri dari 32 bit dan bisa menampung lebih dari 4.294.967.296 *host* di seluruh dunia. IPv4 dibagi menjadi dalam 4 buah oktet dalam ukuran 8 bit sehingga nilai yang dipakai berkisar dari 0 sampai dengan 255 dan dipisahkan oleh notasi titik. Contoh pengalamatan IPv4 adalah 192.168.0.1. Apabila *host* di seluruh dunia melebihi angka 4.294.967.296 maka dibuatlah IPv6 [13][14].

2. IPv6 (*IP Address* Versi 6)

IPv6 diciptakan untuk menjawab kekhawatiran akan kemampuan IPv4 yang hanya menggunakan 32 bit untuk menampung *IP address* di seluruh dunia. Semakin banyaknya pengguna jaringan internet dari hari ke hari di seluruh dunia, IPv4 dinilai suatu saat akan mencapai batas maksimum yang dapat ditampungnya. Oleh karena itu, IPv6 yang terdiri dari 128 bit diciptakan. IP ini 4 kali dari IPv4 namun jumlah

host yang bisa ditampung bukanlah 4 kali dari 4.294.967.296 melainkan 4.294.967.296 pangkat 4. Jadi hasilnya 340.282.366.920.938.463.463.374.607.431.768.211.456 [14].

2.4.2 Pembagian Kelas IP Address

IPv4 terdiri dari 4 oktet, nilai 1 oktet adalah 255. Karena ada 4 oktet maka jumlah IP *address* yang tersedia adalah $255 \times 255 \times 255 \times 255$. IP *address* sebanyak ini harus dibagikan keseluruhan pengguna jaringan internet di seluruh dunia. Untuk mempermudah proses pembagiannya, IP *address* harus di kelompokkan dalam kelas-kelas.

IP *address* dikelompokkan dalam empat kelas, yaitu kelas A, B, C, D. Perbedaan terletaknya ukuran dan jumlah. IP *address* kelas A dan kelas B digunakan untuk jaringan berukuran besar dan sedang. IP *address* kelas C untuk pembagian jaringan yang banyak, namun masing-masing jaringan memiliki anggota yang sedikit. IP *address* kelas D dan E juga didefinisikan seperti kelas C, tetapi tidak digunakan dalam penggunaan normal, kelas D diperuntukan bagi jaringan *multicast*. Pembagian kelas-kelas IP *address* didasarkan pada dua hal, yaitu *network identity* (id) dan *host id* dari suatu IP *address*. Setiap IP *address* merupakan suatu pasangan *network id* (identitas jaringan) dan *host id* (identitas *host* dalam suatu jaringan). Masing-masing komputer atau *router* di suatu jaringan *host id*-nya harus unik dan harus berbeda dengan komputer yang lain [14]. Pembagian kelas IP *address* dapat dilihat pada tabel 2.10.

Tabel 2.10 Pembagian Kelas IP Address [14]

<i>Class</i>	<i>Network Bits</i>	<i>Host Bits</i>	<i>Decimal Address Range</i>	<i>Subnet Mask</i>
<i>Class A</i>	8 bits	24 bits	1-126	255.0.0.0
<i>Class B</i>	16 bits	16 bits	128-191	255.255.0.0
<i>Class C</i>	24 bits	8 bits	192-223	255.255.255.0
<i>Class D</i>	<i>Reserves for Multicasting</i>		224-239	N/A

2.5 Protokol Komunikasi

Protokol komunikasi adalah sistem aturan atau standar yang mengatur atau mengizinkan terjadinya hubungan, komunikasi, dan perpindahan data antara dua perangkat atau lebih [15]. Hal yang mendasari protokol komunikasi adalah menggunakan prinsip pengiriman suatu benda dalam kehidupan sehari-hari dengan analogi seseorang mengirim pesan. Komunikasi bisa dianalogikan sebagai berikut:

Source (pengirim) – *Transmitter* (pengirim ke pos pengiriman) – *Transmission System* (sistem yang mengirimkan pesan seperti dengan mobil, pesawat, dll) – *Receiver* (penerima pesan) – *Destination* (alamat pesan yang dituju).

Analogi komunikasi di atas dapat diserap dan diimplementasikan pada komunikasi elektronik. Komunikasi elektronik membutuhkan suatu protokol komunikasi, protokol adalah sebuah aturan yang menentukan format, *frame* dan *packet* dalam pengiriman pesan. Beberapa hal yang perlu diperhatikan dalam *protocol engineering* [16] adalah:

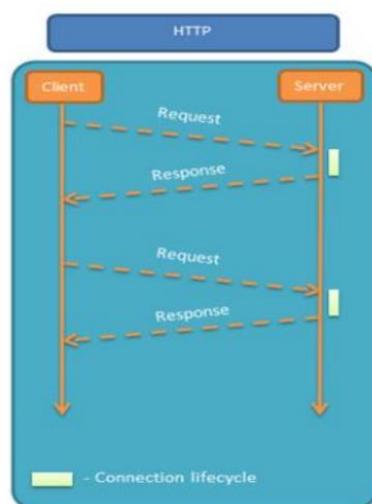
1. Proses pengembangan protokol komunikasi.
2. Rancangan protokol.
3. Verifikasi protokol.
4. Evaluasi kinerja protokol.
5. Implementasi protokol.
6. Pengujian protokol.

2.6 Hypertext Transfer Protocol (HTTP)

HTTP adalah protokol aplikasi yang berjalan di atas protokol *Transmission Control Protocol/Internet Protocol* (TCP/IP.) HTTP merupakan sebuah protokol yang bersifat *stateless* yang dapat digunakan untuk sistem informasi terdistribusi, kolaboratif, dan berbasis pada *hypertext* yang diperkenalkan pada awal tahun 90an [17]. HTTP adalah sebuah protokol yang berjalan pada *port* 80 [5].

Masalah utama dari protokol HTTP adalah proses pengiriman HTTP *request* dan HTTP *response* dilakukan tanpa ada pengaman sama sekali dalam komunikasi antara *client* dan server, sehingga seseorang yang memiliki akses di jaringan mampu menyadap informasi yang dikirimkan (*traffic sniffing*) dan bahkan bisa

melakukan modifikasi (*data tampering*) tanpa diketahui oleh kedua belah pihak. Penyerangan cukup memiliki akses ke salah satu infrastruktur jaringan yang dilalui dan memasang aplikasi yang mampu menyedap seperti halnya *wireshark*. Setiap terjadi permintaan akses melalui protokol HTTP maka penyerang akan dapat melihat seluruh data yang dikirimkan, seperti *username* dan *password* yang seharusnya menjadi informasi rahasia dari masing-masing pengguna [17].



Gambar 2.6 Mekanisme Kerja Protokol HTTP [5]

Gambar 2.6 menunjukkan bahwa protokol HTTP membutuhkan *request* dan *response* setiap kali proses komunikasi [5]. Pesan yang dikirim HTTP bergerak dalam satu arah (*half duplex*) seperti mengalir dari hulu dan hilir, baik dalam hal *request* atau *response*. Sebuah pesan HTTP merupakan sebuah data sederhana, yang dibentuk dalam format tertentu. Setiap pesan selalu berisi *request* ataupun *response*, yang dibagi ke dalam tiga bagian [18]:

1. Bagian “*Start Line*” yang berisi deskripsi pesan.
2. Bagian “*Header*” yang berisi daftar atribut dari pesan.
3. Bagian “*Body*” yang berisi data.

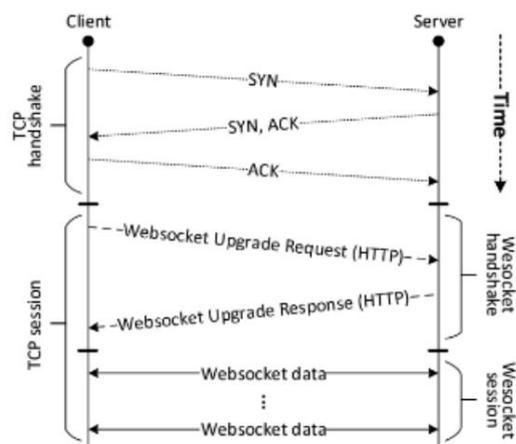
2.7 WebSocket

WebSocket adalah salah satu cara untuk komunikasi antar komputer, umumnya melalui *network* atau internet [19], yang merupakan salah satu protokol yang berjalan pada layer aplikasi atau pada layer tujuh di dalam *Open System*

Interconnection (OSI) layer [5]. WebSocket merupakan protokol komunikasi yang mulai dikembangkan pada HTML 5 yang mendukung komunikasi *full duplex* dimana komunikasi terjadi dua arah (*bidirectional*) melalui satu koneksi TCP antara *client* dan server, yang merupakan standar baru untuk komunikasi *real time* pada web dan aplikasi *mobile*. WebSocket dirancang untuk diterapkan di browser web dan server web, tetapi dapat juga digunakan pada aplikasi *client* maupun server [3][4][5].

Protokol ini adalah hasil peningkatan dari HTTP dan berdiri pada protokol TCP [4]. Namun, WebSocket menggunakan HTTP sebagai mekanisme *transport*-nya [20]. WebSocket dirancang untuk menggantikan teknologi komunikasi dua arah yang mulanya menggunakan HTTP dengan memanfaatkan infrastruktur yang ada (*proxies, filtering, authentication*), layer yang digunakan pada arsitektur WebSocket antara lain FIN, RSV1, RSV2, RSV3, *opcode, mask, payload length*, dan *extend payload length* sehingga minimal paket yang dikirim oleh protokol WebSocket sebesar 2 byte. Teknologi ini diimplementasikan sebagai keputusan antara efisiensi dan kehandalan, karena HTTP memiliki permasalahan pada metode pengiriman pesan secara *real time* yang meminta permintaan kepada server secara terus-menerus yang menyebabkan server menjadi sibuk, tujuan dari diciptakannya WebSocket adalah mengatasi komunikasi dua arah dengan menggunakan infrastruktur HTTP yang berjalan pada saluran 80, 433 dan mendukung *proxy* HTTP [21]. Keuntungan utama menggunakan WebSocket adalah mampu berinteraksi antara *client-server* dengan *overheads* yang rendah, dapat memfasilitasi kebutuhan transfer data *stream* secara aktual [3], dan memiliki *header* yang kecil [4].

Dengan menggunakan mekanisme satu koneksi TCP maka WebSocket dapat membangun satu buah jalur koneksi antara *client* dan server. Dimana, WebSocket menggunakan metode *handshake* untuk membuat sebuah koneksi yang terbuka antara *client* dan server. Gambar 2.7 menunjukkan proses *handshake* antara *client* dengan server pada protokol WebSocket.



Gambar 2.7 Mekanisme Kerja Protokol WebSocket [22]

Gambar 2.7 menunjukkan bahwa proses pertama mekanisme kerja protokol WebSocket adalah *client* mengirimkan *handshake* kepada server kemudian server membalas *handshake* tersebut lalu terciptalah sebuah koneksi TCP antara *client* dan server. Koneksi TCP tersebutlah yang mendukung komunikasi *bidirectional* antara *client* dan server sampai koneksi tersebut diputus oleh *client* ataupun server [5].

Protokol WebSocket sudah di standardisasi oleh *Internet Engineering Task Force* (IETF) sebagai RFC 6455 dan *Application Programming Interface* (API) WebSocket di IDL di standarisasi oleh W3C. Manfaat WebSocket [19] adalah:

- a. WebSocket memungkinkan *server* untuk mendorong data kepada *client* yang terhubung.
- b. Mengurangi *traffic* jaringan yang tidak perlu dan *latency* menggunakan *full duplex* melalui koneksi tunggal.
- c. Streaming melalui *proxy* dan *firewall*, mendukung komunikasi simultan hulu dan hilir.
- d. Kompatibel dengan pre-WebSocket dengan cara beralih dari koneksi HTTP ke WebSocket.

2.8 Socket.io

Socket.io adalah salah satu *library* javascript yang digunakan dalam Node.js. Socket.io menggunakan WebSocket sebagai mekanisme layer aplikasinya. Socket.io digunakan untuk membuat sebuah sistem antara *client* dan server dapat

berkomunikasi secara *real time*, *bidirectional* (dua arah), dan komunikasi berbasis *event*. Socket.io terbagi menjadi dua bagian, yaitu:

1. *Client-side library*, yang berjalan di browser *client*.
2. *Server-side library*, yang dapat digunakan sebagai modul untuk Node.js.

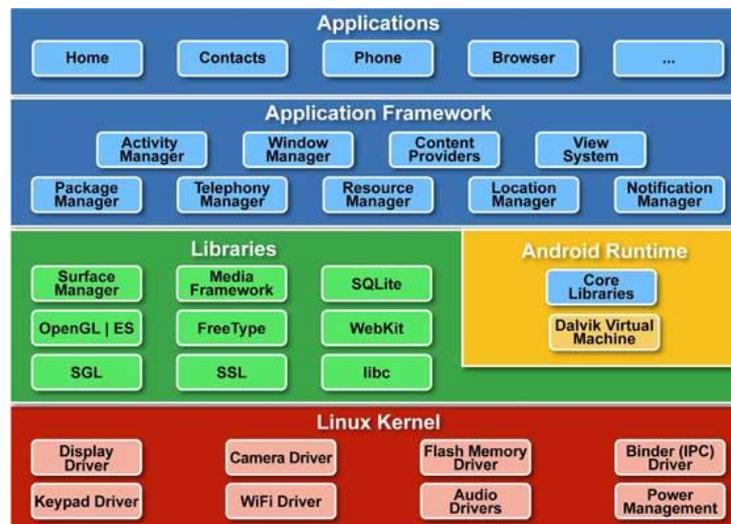
Kedua komponen mempunyai API yang sama. Seperti Node.js, Socket.io juga bersifat *event-driven*. Socket.io menggunakan protokol websocket dengan *polling* sebagai opsi *fallback*. Meskipun Socket.io merupakan ‘pembungkus’ untuk soket web, namun ia memiliki banyak fitur, antara lain *broadcast* ke banyak soket, dan *Input/Output (I/O)* yang *asynchronous* [5][23][24][25].

2.9 Android

Android adalah sebuah *software* untuk perangkat *mobile* yang mencakup sistem operasi, *middleware* dan aplikasi kunci. *Android Software Development Kit (SDK)* menyediakan alat dan API yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* android menggunakan bahasa pemrograman java [26]. Android merupakan sistem operasi yang dikembangkan untuk perangkat *mobile* berbasis Linux. Awalnya sistem operasi ini dikembangkan oleh Android Inc., kemudian dibeli oleh Google pada tahun 2005 dan pada tahun 2007 dibentuklah *Open Handset Alliance (OHA)*, sebuah konsorsium dari beberapa perusahaan, yaitu Texas Instruments, Broadcom Corporation, Google, HTC, Intel, LG, Marvell Technology Group, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, dan T-Mobile dengan tujuan untuk mengembangkan standar terbuka untuk perangkat *mobile*. Sistem operasi ini dibuat *open source* atau terbuka sehingga memungkinkan penggunaanya untuk menciptakan aplikasi mereka sendiri [27].

2.9.1 Struktur Aplikasi Android

Struktur aplikasi android atau fundamental aplikasi, ditulis dalam bahasa pemrograman java. Kode java dikompilasi bersama dengan *resource file* yang dibutuhkan oleh aplikasi, dimana prosesnya di-*package* oleh *tools* yang dinamakan *apk tools* ke dalam paket android. Sehingga menghasilkan *file* dengan ekstensi *apk*. *File* *apk* ini yang disebut dengan aplikasi, dan nantinya dapat dijalankan pada peralatan *mobile*.



Gambar 2.8 Arsitektur Sistem Android [28]

Gambar 2.8 menunjukkan gambaran dari arsitektur sistem android yang secara garis besar terdiri dari [28]:

1. *Application and Widget*

Merupakan layer yang digunakan sebagai untuk instalasi dan menjalankan aplikasi. Layer ini terdapat aplikasi inti seperti *e-mail*, sms, kalender, browser, dan lain-lain.

2. *Application Framework*

Merupakan layer yang digunakan sebagai tempat para pembuat aplikasi melakukan pengembangan dan pembuatan aplikasi yang akan dijalankan pada sistem operasi android.

3. *Libraries*

Merupakan layer tempat fitur-fitur android berada, contohnya seperti *libraries* media untuk memutar media video dan audio, *libraries graphic* dan masih banyak lainnya.

4. *Android Run Time*

Merupakan layer yang membuat aplikasi android dapat dijalankan.

5. *Linux Kernel*

Merupakan tempat inti dari sistem operasi android.

2.9.2 Android Studio

Android studio adalah *Integrated Development Environment* (IDE) untuk pengembangan atau pembuatan aplikasi android yang bersifat *open source* atau gratis. Selain itu, android studio merupakan hasil pengembangan dari IDE sebelumnya yaitu eclipse. Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan eclipse disertai dengan *Android Development Tools* (ADT) *plugin*. Android studio memiliki beberapa fitur, yaitu [26]:

1. Projek berbasis pada *Gradle Build*.
2. *Refactory* dan pembenahan *bug* yang cepat.
3. Mendukung *Proguard and App-signing* untuk keamanan.
4. Memiliki *Graphic User Interface* (GUI).
5. Didukung oleh *Google Cloud Platfrom* untuk setiap aplikasi yang dikembangkan.

Berikut merupakan beberapa *software* pendukung agar android studio dapat berjalan:

1. *Software Development Kit* (SDK)

SDK terdiri atas *platform*, *tools*, *sample code* dan dokumentasi yang dibutuhkan untuk pengembangan aplikasi android. Selain itu, *emulator* yang terdapat pada SDK digunakan untuk mengimplementasikan beberapa fitur pada perangkat android, seperti menampilkan browser dan menghubungkan ke internet. Sebagai *tools API*, SDK menggunakan bahasa pemrograman java untuk mengembangkan aplikasi di *platform* android [26].

2. *Java Development Kit* (JDK)

JDK adalah paket fungsi API untuk bahasa pemrograman java, meliputi *Java Runtime Environment* (JRE) dan *Java Virtual Machine* (JVM). JDK merupakan perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode java menjadi *bytecode* yang dapat dimengerti dan dapat dijalankan oleh JRE. JRE merupakan komponen utama yang bertugas untuk menjalankan aplikasi berbasis java. Aplikasi java dapat berjalan di suatu perangkat komputer apabila di perangkat komputer tersebut sudah terinstal JRE.

JDK merupakan perangkat pengembangan aplikasi java yang bisa diunduh secara gratis. Perangkat ini diperlukan untuk membangun aplikasi android, mengingat aplikasi android berbasis java [26].

3. Java

Java diciptakan pada tahun 1991, oleh James Gosling yaitu seorang *developer* dari Sun Microsystem yang kemudian berkembang dan digunakan untuk menciptakan *executable content* untuk didistribusikan melalui jaringan. Java merupakan bahasa berorientasi objek dengan unsur seperti bahasa C++ dan bahasa lainnya, digunakan dalam pengembangan aplikasi untuk perangkat-perangkat cerdas yang dapat berkomunikasi melalui internet atau jaringan komunikasi, aplikasi mandiri, serta aplikasi berbasis internet. Terdapat dua program yang berbeda dalam java, yaitu *applet* dan *application*. *Applet* merupakan program yang yang disimpan pada web server sehingga dapat dijalankan melalui web browser. Sedangkan program yang dapat dijalankan secara langsung disebut *application* [26].

4. *eXtensible Markup Language* (XML)

XML merupakan bahasa web turunan dari *Standard Generalized Markup Language* (SGML) yang ada sebelumnya. XML hampir sama dengan HTML, dimana kedua-duanya diturunkan dari SGML. Secara sederhana XML adalah suatu bahasa yang digunakan untuk mendeskripsikan dan memanipulasi dokumen secara terstruktur [29]. XML didefinisikan sebagai suatu bahasa *markup* yang menggunakan *tag* sebagai penanda untuk mengkategorikan dan menjelaskan teks atau data dalam sebuah dokumen agar lebih spesifik [15]. Jadi, XML merupakan kumpulan aturan untuk mendesain format teks, sehingga format teks lebih terstruktur dan lebih mudah dibaca oleh komputer [30].

2.9.3 Volley

Volley adalah *library* HTTP yang digunakan untuk membantu pertukaran data antara server dengan *client*. Volley membuat pertukaran data menjadi lebih mudah dan lebih cepat untuk aplikasi android. Secara *default* semua jaringan volley menghasilkan *asynchronous* sehingga tidak perlu khawatir menggunakan

AsyncTask. Volley dilengkapi dengan banyak fitur [31][32][33], diantaranya adalah:

1. Melakukan sebuah *request queuing and prioritization* (mengutamakan prioritas dalam antrean).
2. Efektif untuk melakukan *cache* dan manajemen memori.
3. Perluasan dan kustomisasi *library* untuk kebutuhan pengguna.
4. Dapat melakukan pembatalan dalam sebuah *request*.

2.10 Web Server

Web server adalah sebuah perangkat lunak server yang berfungsi untuk menerima permintaan *request* data yang dikirimkan dari *client* menggunakan web browser dengan protokol komunikasi yang telah ditentukan kemudian mengirimkan kembali hasilnya dalam bentuk halaman web yang umumnya berbentuk dokumen *Hypertext Markup Language* (HTML) sebagai bentuk respon data. Halaman web yang diminta bisa terdiri dari berkas teks, video, gambar, *file*, dan banyak lagi. Jadi, web server adalah sebuah servis untuk menerima *request* dari *client* dan mengirim *response* kepada *client* [34][35][36][37].

2.10.1 Virtual Private Server (VPS)

VPS adalah suatu metode virtual membagi *resource* yang meliputi *Central Processing Unit* (CPU), *Random Access Memory* (RAM) dan *storage* pada *physical* server menjadi beberapa virtual server, yang keseluruhan *resource*-nya hanya digunakan oleh satu pengguna saja dan tidak dipengaruhi oleh pengguna lain. Pengguna dapat mengelola secara penuh semua konfigurasi dan *resource* yang ada pada VPS. Jadi, seolah-olah pengguna memiliki sebuah komputer pribadi atau server yang sewaktu-waktu dapat dikontrol.

VPS terasa seperti sebuah *dedicated* server. Jika dibandingkan dengan *shared hosting*, menyewa VPS akan mendapatkan *resource* yang lebih baik sehingga tidak terganggu jika ada *problem* pada website yang dikelola. Selain itu, VPS menyediakan konfigurasi layaknya *dedicated* server dan akses *full root* terhadap *user root* sehingga lebih leluasa dalam mengkustomasi server sesuai kebutuhan. Pengguna bisa mengakses dan mengelola seluruh aspek *software* dari

server termasuk akses administrator di sistem operasi server sampai aplikasi yang akan di implementasikan di server tersebut.

Jadi, teknologi yang digunakan VPS adalah virtualisasi *hardware* server fisik yang kemudian dibagi menjadi beberapa *resource* berbeda. VPS disebut virtual karena pembagian *resource* dilakukan dengan menggunakan *software* sehingga dalam satu server fisik bisa terdapat beberapa VPS yang dijalankan [38][39].

2.10.2 Webuzo

Webuzo adalah *single user control* panel yang biasa digunakan pada layanan VPS. Webuzo sedikit berbeda dengan *Web Host Manager* (WHM) yang dapat digunakan untuk membuat beberapa akun *hosting* cPanel. Webuzo dapat digunakan layaknya WHM sekaligus sebagai *hosting*. Panel Webuzo banyak digunakan oleh pengguna VPS karena fiturnya yang lengkap dan *user-friendly*.

Webuzo dapat diinstal pada virtual server, *dedicated* server, atau menerapkannya dalam peralatan virtual. Webuzo berfungsi sebagai manajemen server. Terdapat beberapa fitur pada Webuzo seperti manajemen domain, manajemen *user File Transfer Protocol* (FTP), manajemen *database* dengan *My Structured Query Language* (MySQL) dan Rockmongo, manajemen *Domain Name System* (DNS) *zone*, manajemen *MXEntry records*, manajemen CRON, keamanan menggunakan *IP block*, CSR, dan sebagainya. Webuzo dapat dengan mudah diinstal pada server virtual. Server virtual yang didukung oleh Xen PV atau HVM, Linux-KVM, OpenVZ, dan yang lainnya. Secara *default* Webuzo akan menginstal MySQL, PHP 5.5, FTP dan beberapa *service* penting lainnya [40].

2.10.3 Apache HTTP Server

Apache HTTP server dikelola dan dikembangkan oleh *Apache Software Foundation*. Apache HTTP server adalah web server yang dapat dijalankan di banyak sistem operasi, seperti Unix, BSD, Linux, Windows, dan Novell Netware serta *platform* lainnya yang berguna untuk melayani dan memfungsikan situs web [35]. Protokol yang digunakan untuk melayani fasilitas web ini adalah HTTP. Apache tidak hadir dalam bentuk server fisik, melainkan *software* yang

menjalankan sebuah server. Server dan *client* berinteraksi melalui protokol HTTP. Sementara itu Apache bertanggung jawab terhadap kelancaran serta keamanan komunikasi antara server dengan *client*.

Apache awalnya dikembangkan berbasis kode pada NCSA HTTPD 1.3 yang kemudian diprogram ulang menjadi sebuah web server yang paling banyak digunakan saat ini. Apache kini menjadi web server yang paling populer dan banyak digunakan lebih dari 42% dari berbagai domain website yang ada di internet. Apache memiliki fitur yang sangat lengkap mulai dari performa yang tinggi, fungsionalitas, efisiensi, serta kecepatan. Apache juga merupakan web server berbasis *open source* [41]. Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigurasi, autentikasi berbasis basis data dan lain-lain. Selain itu Apache juga bisa digunakan sebagai *load balancer* dengan menambahkan beberapa modul konfigurasi tertentu [35].

2.10.4 *CodeIgniter*

CodeIgniter adalah sebuah *framework* PHP yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi web berbasis PHP dibanding jika menulis semua kode program dari awal. Sehingga dalam pengembangan aplikasi akan menjadi lebih terstruktur dan terorganisasi. Adapun beberapa keuntungan menggunakan *CodeIgniter* [24][42], diantaranya:

1. Gratis

CodeIgniter berlisensi dibawah Apache/BSD *open source*.

2. Ditulis menggunakan PHP 4

Meskipun *CodeIgniter* dapat berjalan di PHP 5, namun sampai saat ini kode program *CodeIgniter* masih dibuat dengan menggunakan PHP 4.

3. Berukuran kecil

Ukuran *CodeIgniter* yang kecil merupakan keunggulan tersendiri, jika dibandingkan dengan *framework* lain yang berukuran besar.

4. Menggunakan konsep *Model View Controller* (MVC)

CodeIgniter menggunakan konsep MVC yang memungkinkan pemisahan *layer application-logic* dan *presentation*.

5. *Uniform Resource Locator (URL)* yang sederhana

Secara *default*, URL yang dihasilkan *CodeIgniter* sangat bersih dan *Search Engine Friendly (SEF)*.

6. Memiliki paket *library* yang lengkap

CodeIgniter mempunyai *library* yang lengkap untuk mengerjakan operasi-operasi yang umum dibutuhkan oleh sebuah aplikasi berbasis web, misalnya mengakses *database*, mengirim *e-mail*, memvalidasi *form*, menangani *session* dan sebagainya.

7. *Extensible*

Sistem dapat dikembangkan dengan mudah menggunakan *plugin* dan *helper*, atau dengan menggunakan *hooks*.

8. Tidak memerlukan *template engine*

Meskipun *CodeIgniter* dilengkapi dengan *template parser* sederhana yang dapat digunakan, tetapi hal ini tidak mengharuskan kita untuk menggunakannya.

9. Dokumentasi lengkap dan jelas

CodeIgniter adalah satu-satunya *framework* dengan dokumentasi yang lengkap dan jelas

10. Komunitas

Komunitas *CodeIgniter* saat ini berkembang pesat. Salah satu komunitasnya bisa dilihat di (<http://codeigniter.com/forum/>).

2.10.5 Node.js

Node.js adalah salah satu *framework* dari javascript yang banyak digunakan untuk melakukan pemrograman server atau pemrograman jaringan. Node.js dibangun diatas Chrome's V8 Javascript Engine. Chrome's V8 Javascript Engine adalah *open source high performance javascript engine* yang ditulis dalam bahasa C++. Node.js dapat digunakan sebagai *framework* bahasa pemrograman yang mengatur *event* dari *input user* serta Node.js dapat sangat cepat dan ringan dalam mengatur I/O model dikarenakan Node.js menggunakan javascript sebagai bahasa pemrogramannya sehingga bersifat *framework*. Salah satu kelebihan Node.js adalah penyelesaian dalam permasalahan *deadlock* proses dimana setiap proses yang

dilakukan tanpa *thread* akan terjadi *deadlock* namun *threading programming* relatif tidak efisien dan sulit untuk diimplementasikan. Karena Node.js bersifat *non blocking* maka tidak ada proses yang menjadi *deadlock*. Maka *user* dapat dengan mudah mengembangkan sistem dengan skala yang besar didalam Node.js. [5].

2.10.6 My Structured Query Language (MySQL)

MySQL adalah salah satu jenis *database server* yang bersifat *open source*. MySQL dapat dijalankan pada semua *platform* baik Windows maupun Linux. MySQL adalah salah satu *Database Management System* (DBMS) dari sekian banyak DBMS seperti Oracle, MS SQL, Postagre SQL, dan lainnya. MySQL dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user* (*multi-user*), dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*). Sesuai namanya, bahasa standar yang digunakan MySQL adalah SQL. SQL singkatan dari *Structure Query Language*, yang merupakan bahasa standar untuk pengolahan *database*. SQL menyediakan perintah untuk membuat *database*, *field*, ataupun *index* untuk menambah atau menghapus data [43][42].

2.10.7 Hypertext Markup Language (HTML)

HTML merupakan sebuah bahasa struktur yang menggunakan *markup* atau penanda untuk membuat halaman web. HTML berperan untuk menentukan struktur konten dan tampilan dari sebuah web. HTML bukanlah bahasa pemrograman (*programming language*), tetapi bahasa *markup* (*markup language*). HTML disebut *hypertext* karena di dalam HTML sebuah teks biasa dapat membuatnya menjadi *link* yang dapat berpindah dari satu halaman ke halaman lainnya hanya dengan mengklik teks tersebut. Kemampuan teks inilah yang dinamakan *hypertext*, walaupun pada implementasinya nanti tidak hanya teks yang dapat dijadikan *link*. HTML juga disebut *markup language* karena bahasa HTML menggunakan tanda (*mark*), untuk menandai bagian-bagian dari teks. Misalnya, teks yang berada di antara tanda tertentu akan menjadi tebal, dan jika berada di antara tanda lainnya akan tampak besar. Tanda ini di kenal sebagai HTML *tag* [44].

2.10.8 *Cascading Style Sheets (CSS)*

CSS adalah kumpulan kode program yang digunakan untuk mendesain atau mempercantik tampilan halaman HTML. CSS dapat merubah desain dari teks, warna, gambar dan latar belakang dari (hampir) semua kode *tag* HTML. CSS biasanya selalu dikaitkan dengan HTML, karena keduanya memang saling melengkapi. HTML ditujukan untuk membuat struktur, atau konten dari halaman web. Sedangkan, CSS digunakan untuk tampilan dari halaman web tersebut. Istilahnya, “*HTML for content, CSS for Presentation*” [44].

2.10.9 *Hypertext Processor (PHP)*

PHP adalah bahasa *server-side-scripting* yang menyatu dengan HTML. Karena PHP merupakan *server-side-scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di server kemudian hasilnya akan dikirimkan ke browser dengan format HTML. Awalnya, PHP merupakan singkatan dari *Personal Home Page* yang merupakan bahasa skrip yang tertanam dalam HTML untuk dieksekusi bersifat *server side*. PHP termasuk dalam *open source product*, sehingga *source code* PHP dapat diubah dan didistribusikan secara bebas.

PHP adalah bahasa pemrograman yang berjalan dalam sebuah web server dan berfungsi sebagai pengolahan data pada sebuah server. PHP makin populer dan banyak orang karena punya banyak kelebihan. Berikut beberapa kelebihan yang dimiliki PHP [43], antara lain:

1. Aksesnya cepat karena ditulis di tengah kode HTML sehingga waktu respon programnya lebih cepat.
2. Murah, bahkan gratis. Tidak perlu membayar *software* ini untuk menggunakannya.
3. Mudah dipakai, fitur dan fungsinya lengkap, serta cocok dipakai untuk membuat halaman web dinamis.
4. Dapat dijalankan di berbagai sistem operasi, seperti Windows, Linux, Mac OS, dan lain sebagainya.
5. Dukungan teknis banyak tersedia. Banyak forum dan situs didedikasikan untuk *trouble shooting* berbagai masalah seputar PHP.

2.10.10 Javascript

Javascript merupakan salah satu bahasa pemrograman yang tergolong handal dan cepat dalam memproses perintah dikarenakan sifat yang non-blok. Javascript adalah bahasa pemrograman yang bersifat *Client Side Programming Language*. *Client Side Programming Language* adalah tipe bahasa pemrograman yang pemrosesannya dilakukan oleh *client*. Aplikasi *client* yang dimaksud merujuk kepada web browser seperti Google Chrome dan Mozilla Firefox. Bahasa pemrograman *Client Side* berbeda dengan bahasa pemrograman *Server Side* seperti PHP, dimana untuk *server side* seluruh kode program dijalankan di sisi server. Untuk menjalankan javascript, kita hanya membutuhkan aplikasi *text editor* dan web browser. Tetapi, pengembangan javascript sangatlah pesat. Javascript dapat digunakan untuk pemrograman *server side* dimana pemrograman tersebut berjalan pada sisi server dan javascript pun digunakan untuk pemrograman *network* seperti *framework javascript Node.js* [5][44].

2.10.11 PuTTY

PuTTY adalah sebuah *software remote console* atau terminal yang bersifat *open source* yang dapat digunakan untuk melakukan protokol jaringan *Secure Shell* (SSH) dan Telnet. Protokol ini dapat digunakan untuk menjalankan sesi *remote* pada sebuah komputer melalui sebuah jaringan, baik itu *Local Area Network* (LAN), maupun internet. Program ini banyak digunakan oleh para pengguna komputer tingkat menengah ke atas, yang biasanya digunakan untuk menyambungkan, mensimulasi, atau mencoba berbagai hal yang terkait dengan jaringan [45].

2.10.12 Sublime Text

Sublime *text* adalah editor teks untuk berbagai bahasa pemrograman termasuk pemrograman PHP. Sublime *text* merupakan *editor text* lintas *platform* dengan Python API. Sublime *text* juga mendukung banyak bahasa pemrograman dan bahasa *markup*, dan fungsinya dapat ditambah dengan *plugin*.

Sublime *text* pertama kali dirilis pada tanggal 18 januari 2008. Sublime *text* mendukung sistem operasi, seperti Linux, Mac Os X, dan juga Windows.

Terdapat banyak fitur yang tersedia pada sublime *text*, diantaranya *minimap*, membuka *script* secara *side by side*, *bracket highlight*, kode *snippets*, dan lain sebagainya [44].

2.11 *Black Box Testing*

Black box testing merupakan pengujian yang berfokus pada kebutuhan fungsionalitas sebuah perangkat lunak atau program tanpa menguji desain dan kode program untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sudah sesuai dengan spesifikasi yang dibutuhkan. Jadi, pengujian ini dilakukan tanpa mengetahui detail struktur internal dari perangkat lunak yang di uji. Pengujian ini memandang perangkat lunak atau program seperti layaknya sebuah “kotak hitam” yang benar-benar tidak peduli dengan struktur internal dari program atau sistem, tetapi cukup dikenai proses *testing* di bagian luar. Pengujian ini dilakukan untuk meyakinkan semua *input* diterima dengan tepat, dan *output* yang dihasilkan juga tepat dan berjalan dengan baik. Jadi, metode pengujian *black box* adalah pengujian yang dilakukan untuk mengetes hubungan antar program dalam sebuah sistem. Kelebihan dan kelemahan pada *black box testing* dapat dilihat pada tabel 2.11 [46][47][48][49]:

Tabel 2.11 Kelebihan dan Kelemahan *Black Box Testing* [49]

Kelebihan	Kelemahan
Perincian aplikasi dapat ditentukan di awal, dan pengujian dilakukan berdasarkan perincian spesifikasi tersebut.	Apabila keperluan perangkat lunak yang akan dikembangkan tidak begitu jelas, pembuatan dokumentasi yang tepat akan sedikit sulit.
Dapat dipakai untuk menilai konsistensi suatu aplikasi, dan tidak perlu melihat kode program secara detail.	Pengguna akan kurang merasa yakin dengan perangkat lunak yang diuji apakah lolos dalam standar pengujian.

2.12 *Stress Test*

Stress test adalah berupa simulasi serangan “*brute force*” yang menjalankan muatan atau permintaan secara berlebihan menuju sistem [50]. *Stress test* dikenal sebagai pengujian daya tahan yang digunakan untuk menguji stabilitas dan

keandalan sistem. Pengujian ini dapat menentukan sistem pada ketahanan dan penanganan kesalahan dalam kondisi beban yang sangat berat. Bahkan pengujian di luar titik operasi normal dan mengevaluasi bagaimana sistem bekerja di bawah kondisi ekstrim tersebut. *Stress test* dilakukan untuk memastikan bahwa sistem tidak akan *crash* di bawah situasi krisis. Penggunaan *stress test* yang paling menonjol adalah untuk menentukan batas maksimum suatu sistem saat diberi beban berat. Pengujian ini juga memeriksa apakah sistem menunjukkan manajemen kesalahan yang efektif dalam kondisi ekstrim. Berikut beberapa contoh skenario pada *stress testing*, antara lain [51]:

1. Jika pemindai virus dimulai di semua mesin secara bersamaan.
2. Sejumlah besar pengguna *login* pada saat yang bersamaan.
3. Jika *database offline* ketika sedang diakses dari situs web.
4. Ketika sejumlah besar data dimasukkan ke *database* secara bersamaan.

Pegujian *stress test* dalam tugas akhir ini diujikan pada protokol WebSocket untuk mendapatkan nilai *Round Trip Time* (RTT) dan mendapatkan berapa banyak *request* mampu dilayani oleh sistem presensi perkuliahan untuk mengetahui kelayakan penggunaan protokol Websocket sebagai pengatur komunikasi pada proses presensi perkuliahan di Jurusan Teknik Elektro Politeknik Negeri Sriwijaya. Berikut merupakan rumus yang dipakai untuk mendapatkan nilai rata-rata RTT untuk setiap parameter yang akan diujikan.

$$\text{Rata - Rata Round Trip Time} = \frac{\text{Jumlah RTT Setiap Request yang dikirim}}{\text{Jumlah Request yang dikirim}} \quad (2.1)$$

2.13 Penelitian Terdahulu

Sebagai bahan pertimbangan pembuatan tugas akhir, diperlukan perbandingan penelitian untuk mengetahui penelitian-penelitian yang telah dilakukan sebelumnya yang dapat dilihat pada tabel 2.12.

Tabel 2.12 Perbandingan Penelitian Sebelumnya

No	Judul	Tahun	Penulis	Keterangan
1	Rancang Bangun Sistem Kontrol dan Monitoring Berbasis WebSocket pada Perangkat Arduino	2016	Adhitya Bhawiyuga	Tidak melakukan pengujian fungsional terhadap aplikasi web yang dibangun,.
2	Implementasi <i>Smart Metering</i> Menggunakan <i>Internet Of Things</i> Dengan <i>Transport Protocol</i> Web Socket Berbasis OpenMTC di Universitas Telkom.	2017	Ibnu Fajar Muhammad	Tidak membahas secara mendalam tentang performa protokol yang digunakan.
3	Analisis Perbandingan Performasi Antara WebSocket dengan Long Polling pada Aplikasi <i>Chatting</i> Berbasis Web.	2017	Dhian Galang Pradana	Tidak melakukan pengujian stabilitas dan keandalan sistem jika di bebaskan dengan jumlah <i>client</i> yang banyak.
4	Pembangunan dan Pengujian Protokol MQTT & WebSocket untuk Aplikasi IoT Rumah Cerdas berbasis Android.	2018	Muhammad Adzhar Amrullah	Parameter yang digunakan untuk pengujian kurang bervariasi.
5	Implementasi Protokol WebSocket pada Perangkat Non IP Berbasis NRF24L01 (Studi Kasus: Sistem Monitoring Suhu dan Kontroling Lampu LED)	2018	Labib Imam Hidayat	Jumlah <i>request</i> yang diujikan pada sistem kurang bervariasi, yang mana pengujian dilakukan dengan WebSocket <i>stress test</i> .
6	Pemanfaatan Teknologi QR Code pada Sistem Presensi Mahasiswa Berbasis <i>Mobile</i> .	2019	Yiyi Supendi	Untuk melakukan proses presensi, mahasiswa harus menghadap kepada dosen satu-persatu. Hal tersebut dianggap kurang efektif.