

## BAB II

### TINJAUAN PUSTAKA

Pada bab ini dijelaskan kajian pustaka dan dasar teori yang digunakan untuk menunjang penyelesaian skripsi ini. Kajian pustaka diperlukan untuk melakukan kajian terhadap karya ilmiah yang berkaitan dengan skripsi ini. Tinjauan pustaka yang dipaparkan meliputi: penggunaan *Application Programming Interface Server* sebagai jembatan antara aplikasi yang berjalan pada perangkat lain dengan arduino. Dasar teori yang diperlukan berdasarkan kajian pustaka untuk penyusunan skripsi ini adalah teori konsep jaringan komputer, protokol TCP/IP, protokol HTTP, konsep *Application Programming Interface Server*, *Hypertext Preprocessor (PHP)*, *Javascript Object Notation (JSON)* dan Arduino.

#### 2.1. Tinjauan Umum

Perkembangan teknologi telekomunikasi terutama telekomunikasi yang berbasis komunikasi data yang pesat memicu munculnya inovasi-inovasi baru dalam sistem komunikasi. Sistem telekomunikasi adalah proses penyaluran informasi dari suatu titik ke titik yang lain. komunikasi data adalah pertukaran data dari pengirim menuju penerima. Data yang dikirim dapat berupa teks, audio maupun video. Sedangkan jaringan komputer merupakan kumpulan komputer yang saling terhubung satu sama lain sehingga membentuk suatu topologi dengan menggunakan protokol komunikasi yang bersifat universal. Dengan memanfaatkan jaringan komputer pertukaran data menjadi lebih efektif dan efisien.

Sebuah protokol pada komunikasi data berfungsi sebagai pengatur agar dua buah komputer atau lebih dapat saling berkomunikasi. Dalam berkomunikasi, komputer-komputer ini harus menggunakan protokol yang sama. Di dalam konteks jaringan, protokol merupakan suatu aturan formal dan kesepakatan yang menentukan bagaimana komputer bertukar informasi melewati sebuah media jaringan.

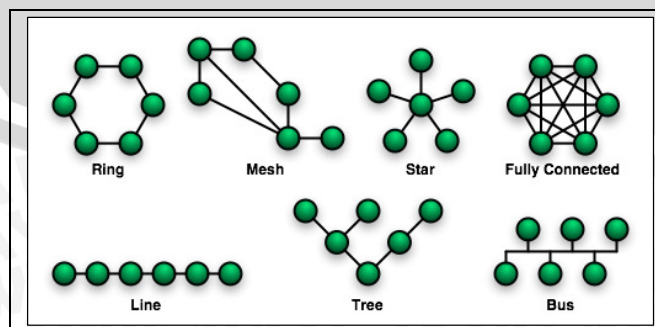
## 2.2 Konsep Jaringan Komputer

Jaringan komputer adalah kumpulan sekelompok komputer otonom yang saling berhubungan antar satu dengan lainnya menggunakan protokol komunikasi melalui media komunikasi sehingga dapat saling berbagi sumberdaya yang ada. Berbagai perangkat yang terhubung dalam jaringan komputer terkoneksi dengan topologi jaringan tertentu. Topologi jaringan adalah, hal yang menjelaskan hubungan geometris antara unsur-unsur dasar penyusun jaringan, yaitu node, link, dan station (Wikipedia, 2013). *Node* adalah perangkat – perangkat yang menjadi titik penghubung dalam jaringan komputer. *Link* adalah media transmisi yang digunakan, sedangkan station adalah *client* atau server dalam jaringan.

Dalam pembahasan konsep ini jarak menjadi hal yang penting dalam merancang sebuah jaringan komputer, karena untuk setiap jarak tertentu diperlukan teknik yang berbeda pula. Berdasarkan jarak dan area jangkauannya jaringan komputer di bagi menjadi tiga kelompok yaitu : *Local Area Network* (LAN), *Metropolitan Area Network* (MAN), dan *Wide Area Network* (WAN) .

### 2.2.1 Local Area Network (LAN)

LAN adalah jaringan yang mencakup area atau jangkauan yang sempit/tidak begitu luas, seperti jaringan dalam rumah, kampus, atau kantor di gunakan untuk menghubungkan komputer-komputer pribadi dan workstation menggunakan peralatan secara bersama-sama dan saling bertukar informasi. LAN dimiliki oleh suatu instansi tanpa menggunakan fasilitas yang telah di sediakan oleh penyedia line dari perusahaan telekomunikasi umum. LAN dibangun menggunakan topologi tertentu.. Topologi LAN yang di gunakan ada lima macam yaitu : *Linier Bus*, *Ring* , *Tree*, *Star*, dan *Mesh*



**Gambar 2.1** Network Topologies  
Sumber : Wikipedia

### 2.2.2 Metropolitan Area Network (MAN)

MAN merupakan versi LAN yang berukuran lebih besar dan memakai teknologi yang sama dengan LAN. MAN merupakan pilihan yang tepat untuk membangun infrastruktur jaringan komputer antar kantor dalam satu kota. MAN dapat mencakup perusahaan yang memiliki kantor-kantor yang letaknya sangat berdekatan. Jaringan ini mampu menjangkau cakupan area sampai dengan radius 10-50 km.

### 2.2.3 Wide Area Network (WAN)

*Wide Area Network* adalah sebuah jaringan yang memiliki jangkauan radius yang sangat luas, karena jangkauannya tidak lagi mencakup suatu kota, bahkan lebih yaitu bisa mencakup lintas negara atau lintas benua. Pada sebagian besar WAN, komponen yang dipakai dalam berkomunikasi terdiri dari dua komponen yaitu media transmisi dan perangkat *switching*. Media transmisi di gunakan untuk menyalurkan sinyal dan perangkat *switching* digunakan untuk meneruskan sinyal.

## 2.3. Protokol TCP/IP

TCP/IP (*Transmission Control Protocol/Internet Protocol*) merupakan sekelompok protokol yang mengatur komunikasi data komputer di internet. Karena menggunakan protokol yang sama, maka perbedaan jenis komputer dan sistem operasi antara dua buah komputer tidak menjadi masalah.

TCP/IP terdiri atas kumpulan protokol yang masing-masing bertanggung jawab atas bagian-bagian tertentu komunikasi data, sehingga tugas masing-masing protokol menjadi jelas dan sederhana. Protokol yang satu tidak perlu mengetahui cara kerja protokol yang lain, sepanjang ia masih bisa saling mengirim dan menerima data.

### 2.3.1. Arsitektur Protokol TCP/IP

TCP/IP terdiri dari empat *layer*, yaitu

- *Network Access/Interface Layer*
- *Internet Layer*

- *Host-to-Host Transport Layer*
- *Application Layer*

<b>4</b>	<b>APPLICATION LAYER</b> Terdiri dari aplikasi dan proses-proses yang menggunakan jaringan	Telnet, FTP, HTTP
<b>3</b>	<b>HOST-TO-HOST TRANSPORT LAYER</b> Menyediakan layanan pengiriman end-to-end	TCP, UDP
<b>2</b>	<b>INTERNET LAYER</b> Menetapkan datagram dan menangani routing data	ARP, IP, ICMP
<b>1</b>	<b>NETWORK INTERFACE LAYER</b> Terdiri dari prosedur-prosedur untuk mengakses jaringan fisik	Device Perangkat dan Kartu Interface

Gambar 2.2. *Layer-layer dalam Arsitektur Protokol TCP/IP*  
Sumber: Hunt, 1992:9

Masing-masing layer memiliki tugas yang berbeda :

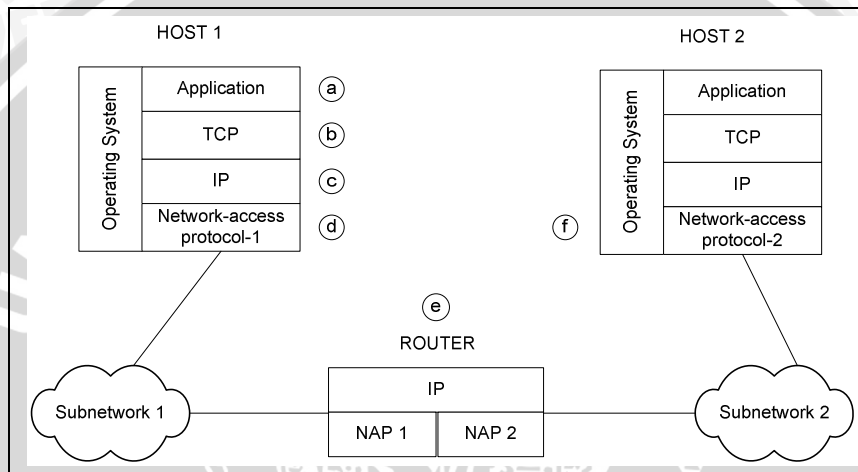
1. *Network Interface Layer*, bertanggung jawab mengirim dan menerima data ke dan dari media fisik. Media fisik ini dapat berupa kabel, serat optik, atau gelombang radio. Protokol pada *layer* ini harus mampu menerjemahkan sinyal listrik menjadi data digital yang dimengerti komputer, yang berasal dari peralatan lain yang sejenis.
2. *Internet Layer* , bertanggung jawab dalam proses pengiriman paket ke alamat yang tepat. Pada *layer* ini terdapat tiga macam protokol, yaitu: IP, ARP, dan ICMP. IP (*Internet Protocol*) berfungsi untuk menyampaikan paket data ke alamat yang tepat. ARP (*Address Resolution Protocol*) ialah protokol yang digunakan untuk menemukan alamat *hardware* dari *host*/komputer yang terletak pada jaringan yang sama. Sedangkan ICMP (*Internet Control Message Protocol*) ialah protokol yang digunakan untuk mengirimkan pesan dan melaporkan kegagalan pengiriman data.
3. *Host-to-Host Transport Layer*, berisi protokol-protokol yang bertanggung jawab untuk mengadakan komunikasi antara dua *host*/komputer. Kedua

protokol tersebut ialah TCP (*Transmission Control Protocol*) dan UDP (*User Datagram Protocol*).

#### 4. Application Layer

Pada *layer* ini terletak semua aplikasi yang menggunakan protokol TCP/IP, seperti: HTTP (*Hypertext Transfer Protocol*), Telnet, FTP (*File Transfer Protocol*) dan SMTP (*Simple Mail Transfer Protocol*).

### 2.3.2. Mekanisme TCP/IP



**Gambar 2.3. Proses Pengalamatan TCP/IP**  
**Sumber: Stallings, 2001:40**

Gambar 2.2. menunjukkan proses pengalamatan pada TCP/IP. Proses ini terjadi pada saat *host* 1 mengirim proses pada *host* 2. Proses-prosesnya adalah :

- Proses pengiriman akan membangkitkan sebuah blok data dan melewatkannya menuju TCP.
- TCP menambahkan informasi kontrol (*header*) sepanjang 20 *byte*, untuk masing-masing potongan data sehingga membentuk segmen TCP.
- TCP membawa masing-masing segmen ke IP dengan perintah transmisi data ke *host* 2.
- Masing-masing datagram IP dibawa ke *network access layer* untuk ditransmisikan ke tujuan. Di sini terjadi penambahan *network access header*.
- Data diterima di *host* 2 dan terjadi proses kebalikannya. Masing-masing *layer* melepaskan *headernya* sebelum melewatkan data ke *layer* di atasnya, dan

sisanya dilewatkan ke *layer* yang lebih tinggi sampai data pengguna asli dikirimkan ke proses tujuan

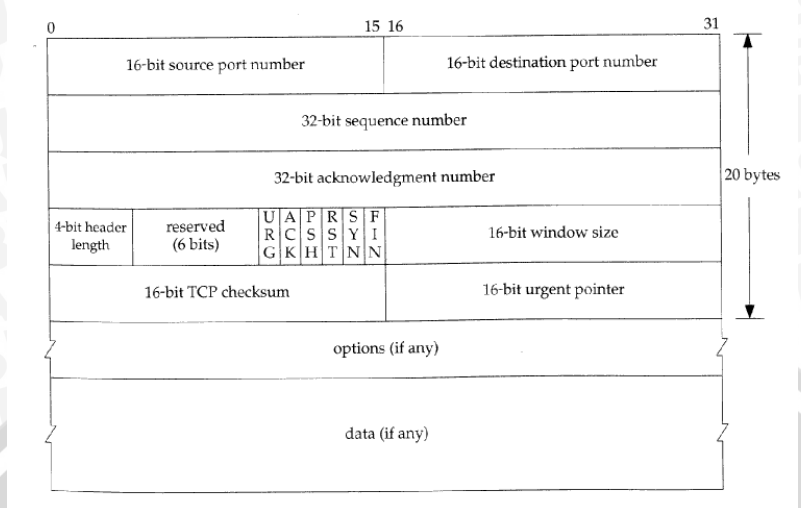
Pada TCP/IP, masing-masing *layer* menambahkan informasi kontrol (*header*) untuk menjamin pengiriman yang sesuai. Informasi kontrol ini diletakkan di depan setiap data yang akan ditransmisikan. Setiap *layer* memperlakukan seluruh informasi yang diterimanya sebagai data dan meletakkan *headernya* sendiri di depan informasi tersebut. Penambahan informasi yang dikirimkan pada tiap *layer* ini disebut dengan proses enkapsulasi.

### 2.3.3. UDP (User Datagram Protocol)

UDP merupakan protokol yang berada dilapisan selain TCP. Protokol ini bersifat *connectionless* dan *unreliable*. Dalam pengiriman data, *connectionless* berarti tidak diperlukan pembentukan suatu hubungan terlebih dahulu untuk pengiriman data. Sedangkan *unreliable*, berarti protokol ini, tidak menjamin data akan sampai pada tujuan dalam kondisi benar. Aplikasi yang memanfaatkan UDP sebagai protokol *transport*, dapat mengirimkan data tanpa melalui proses pembentukan koneksi terlebih dahulu. Lebih sering diimplementasikan untuk aplikasi seperti *multimedia streaming* dimana rugi-rugi paket data kecil masih bisa ditoleransi dari pada nilai *delay* yang lebih besar.

### 2.3.4. TCP (Transmission Control Protocol)

TCP merupakan protokol yang terletak di *layer transport* pada arsitektur protokol TCP/IP. TCP bertanggung jawab dalam menyediakan layanan *connection oriented*, *reliable*, *byte stream service* bagi *application layer*. *Connection oriented* berarti sebelum melakukan pertukaran data, dua aplikasi pengguna TCP harus melakukan pembentukan hubungan (*handshake*) terlebih dahulu. *Reliable* berarti TCP menerapkan proses deteksi kesalahan paket dan retransmisi. *Byte stream service* berarti paket dikirimkan dan sampai ke tujuan secara berurutan.



**Gambar 2.4. Format paket TCP**  
**Sumber: RFC 793**

**2.3.5. IP (Internet Protocol)**

IP merupakan inti protokol TCP/IP yang terletak pada *network layer*. IP melakukan pengiriman data yang bersifat *unreliable*, *connectionless*, dan *datagram delivery service*.

*Unreliable*/ketidakandalan berarti bahwa Protokol IP tidak menjamin datagram yang dikirim pasti sampai ke tempat tujuan. Protokol IP hanya berjanji akan melakukan usaha sebaik-baiknya (*best effort delivery service*), agar paket yang dikirim tersebut sampai ke tujuan. Jika di perjalanan paket tersebut terjadi hal-hal yang tidak diinginkan (salah satu jalur putus, *router* mengalami kongesti/macet, atau *host/network* tujuan sedang *down*), protokol IP hanya memberitahukan ke pengirim paket melalui protokol ICMP, bahwa terjadi masalah dalam pengiriman paket IP ke tujuan. Jika diinginkan keandalan yang lebih baik, keandalan itu harus disediakan oleh protokol yang berada di atas *layer* IP ini (yaitu TCP dan aplikasi pengguna).

Sedangkan kata *connectionless* berarti dalam mengirim paket dari tempat asal ke tujuan, pihak pengirim dan penerima paket IP sama sekali tidak mengadakan perjanjian (*handshake*) terlebih dahulu.

*Datagram delivery service* berarti setiap paket data yang dikirim adalah independen terhadap paket data yang lain. Akibatnya, jalur yang ditempuh oleh

masing-masing paket data IP ke tujuannya bisa jadi berbeda satu dengan lainnya. Karena jalur yang ditempuh berbeda, kedatangan paket pun bisa jadi tidak berurutan.

### 2.3.6. HTTP (*Hypertext Transfer Protocol*)

HTTP merupakan protokol yang terletak di *application layer* pada arsitektur protokol TCP/IP. HTTP tidak menentukan bagaimana data diproses tetapi HTTP bertanggung jawab menentukan bagaimana data ditransfer. HTTP tidak hanya dapat mentransfer halaman web saja tetapi dapat digunakan untuk mentransfer semua format data, bukan hanya web *browser* saja yang dapat menggunakan protokol ini, tetapi aplikasi lain juga dapat melakukan transfer data dengan menggunakan protokol HTTP.

Protokol HTTP bersifat *request-response*, yaitu dalam protokol ini *client* menyampaikan pesan *request* ke *server* dan *server* kemudian memberikan *response* yang sesuai dengan *request* tersebut. *Request* dan *response* dalam protokol HTTP disebut sebagai *request chain* dan *response chain*. Hubungan HTTP yang paling sederhana terdiri atas hubungan langsung antara *user-agent* dengan *server* asal.

#### 2.3.6.1 Format HTTP

HTTP menggunakan format URL (Universal Resource Locator) HTTP dalam bentuk :

```
http : // host [:port] / [resource_path]
```

Gambar 2.5. Format URL pada HTTP  
Sumber: Purbo, 1998:359

- Host adalah nama domain yang legal atau IP Address dari *server*.
- Port adalah bilangan yang menunjukkan port HTTP pada host, jika port tidak disebutkan maka port HTTP diasumsikan sebagai 80.
- Resource\_path menyatakan lokasi resource di dalam host.



### 2.3.6.2 HTTP Request

Komunikasi protokol HTTP terdiri dari pesan request yang diberikan oleh *user agent* dan response yang dikeluarkan oleh server. Setiap request dan response HTTP menggunakan format pesan yang generic seperti yang telah didefinisikan pada RFC 822. Pesan HTTP terdiri atas baris mulai, header pesan dan isi pesan beserta entity (opsional). Header pesan dan isi pesan dipisahkan oleh sebuah baris kosong, yaitu hanya berisi karakter CRLF.

GET /shop/price.html HTTP/1.1	Request Line
Host : www.google.com Accept : text/plain, text/html Accept-Language : en-us User-Agent : Mozilla/5.0 Content-Length : 55	Request Header
CRLF	Line Break
catID=23&prodID=123&price=100	Request Body

Gambar 2.6. Format HTTP Request  
Sumber: Mansfield, 2004:553

### 2.3.6.3 HTTP Method

Setiap HTTP request selalu diawali dengan baris mulai (*request line*). Baris mulai diawali dengan metode request, Request-URI dan versi protokolnya. Metode request menunjukkan metode apa yang hendak dilakukan atas resource yang ditunjuk oleh Request-URI. Ada beberapa metode yang didefinisikan oleh HTTP/1.1. Pada skripsi ini hanya dua metode saja yang digunakan :

1. **GET**, metode GET mengambil informasi apa saja dalam bentuk entity yang diidentifikasi oleh Request-URI.
2. **POST**, metode POST digunakan untuk meminta server menempatkan entity yang dikirim bersama request sebagai subordinat dari Request-URI yang dituju. Metode ini biasa digunakan dalam mengirimkan form.

#### 2.3.6.4 HTTP Response

Setelah menerima request, server harus memberikan response HTTP atas request tersebut, yang terdiri atas baris status, header-header, dan isi pesan. Baris status berisi kode-status yang berupa kode tiga digit dan frasa alasan, yaitu penjelasan singkat atas kode status tersebut. Digit pertama kode status menentukan kelas dari response. Protokol HTTP/1.1 mendefinisikan 5 nilai untuk digit pertama :

- 1XX : Informational – request diterima dan proses berlanjut
- 2XX : Success – request diterima dan dimengerti
- 3XX : Redirection – request membutuhkan tindakan lebih lanjut
- 4XX : Client Error – request mengandung sintaks yang salah
- 5XX : Server Error – server gagal melakukan tindakan sesuai request

Server HTTP dapat menghasilkan kode-status selain yang didefinisikan dalam RFC sepanjang digit pertama kode status tersebut dimengerti oleh aplikasi yang menggunakan protokol HTTP.

#### 2.4 Application Programming Interface Server

*Application Programming Interface* (API) adalah sekumpulan fungsi, perintah dan protokol yang dapat digunakan untuk menghubungkan satu aplikasi dengan aplikasi yang lain agar dapat berinteraksi. Seiring dengan perkembangan internet, API dapat diimplementasikan pada sisi server dan dapat digunakan oleh beberapa aplikasi yang dapat terhubung ke server dengan menggunakan protokol tertentu. Pada protokol HTTP, *Application Programming Interface* umumnya disebut sebagai *Web Application Programming Interface Server* atau *Web Service*.

##### 2.4.1 REST (*Representational State Transfer*)

*Representational State Transfer* adalah sebuah *web architectural style* dan bukan merupakan suatu protokol karena REST menggunakan protokol yang berada pada *application layer* termasuk protokol HTTP. Sebelum REST ditemukan, untuk membuat sebuah *Web Application Programming Interface Server* digunakan SOAP (*Simple Object Access Protocol*). SOAP merupakan

protokol untuk memanipulasi *resource* dengan cara membuat sebuah entity pesan dalam format XML (*Extensible Markup Language*) yang memenuhi standart yang dikeluarkan oleh W3C dan mengirimkan pesan request tersebut dengan metode POST ke server. Selanjutnya server harus melakukan *parsing* terhadap entity XML tersebut untuk menentukan request apa yang diinginkan oleh client.

Prinsip kerja dari REST berbeda dengan SOAP. Pada implementasinya REST jauh lebih cepat jika dibandingkan dengan SOAP, karena tidak diperlukan entity pesan yang harus dibuat dalam format XML, sehingga mempercepat proses persiapan oleh client dan tidak membutuhkan proses *parsing* XML pada server. Dalam REST, *resource* yang diidentifikasi sebagai Request-URI dapat dimanipulasi dengan cara menggunakan HTTP *method* yang sesuai. Setelah menerima request dari client, server akan mengirimkan representasi dari *resource* yang diminta oleh client dalam format apapun, namun sekarang ini format yang banyak dikenali oleh client adalah JSON.

Berikut ini adalah beberapa ciri dari REST :

- Penggunaan HTTP Method (*verb*) yang tepat untuk memanipulasi resource
- Request-URI tidak mengandung *query string* dan hanya terdiri dari element resource yang dipisahkan dengan karakter “/”
- Penggunaan Kode-status yang tepat pada HTTP Response

HTTP METHOD	Resource (URI)	Function
GET	/books	request list of books
POST	/books	create new books
GET	/books/123	request book #123 info
POST	/books/123	update book #123 info

Gambar 2.7. Kombinasi HTTP method dan resource pada REST

### 2.4.2 PHP (*Hypertext Preprocessor*)

*PHP: Hypertext Preprocessor* (PHP) merupakan bahasa berbentuk script yang ditempatkan dalam server dan diproses di server. Pada prinsipnya PHP mempunyai fungsi yang sama dengan *script-script* yang lain seperti ASP (Active Server Page), JSP (JavaServer Pages) dan lainnya.

Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilis kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP. Pada saat skripsi ini ditulis, versi terbaru dari PHP adalah PHP 5.4.14

### 2.4.3 JSON (*Javascript Object Notation*)

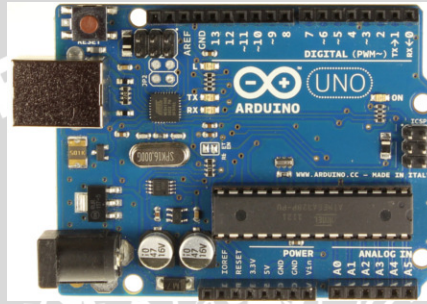
JSON adalah sebuah berkas yang umumnya digunakan sebagai pertukaran data pada internet. Berkas JSON berbasis teks sehingga mudah dikenali oleh berbagai macam bahasa pemrograman sehingga sangat ideal untuk digunakan dalam pertukaran data antar aplikasi yang berbeda bahasa pemrogramannya.

PHP Associative Array	JSON
<pre>array (   name =&gt; "Jon Snow",   home =&gt; array(     "The Wall",     "Winterfell"   )   id    =&gt; 630097 )</pre>	<pre>{   name : "Jon Snow",   home : [     "The Wall",     "Winterfell"   ]   id   : 630097 }</pre>

Gambar 2.8. Format JSON

## 2.5 Arduino

Arduino adalah *platform* untuk melakukan komputasi fisis yang berbasis mikrokontroler. Arduino dapat merasakan lingkungan sekitar dengan cara menghubungkan berbagai jenis sensor pada input dan dapat mengendalikan sesuatu dengan cara menghubungkan aktuator pada output. Salah satu kelebihan adalah arduino dapat dihubungkan dengan *board* yang lain atau biasa disebut *arduino shield* sehingga fungsi dari arduino tersebut dapat diperluas lagi.



Gambar 2.9. Arduino

### 2.5.1 Arduino UNO R3

Arduino memiliki beberapa jenis yang fitur dan fungsinya berbeda antara satu dengan yang lainnya. Arduino UNO R3 adalah salah satu jenis dari arduino yang ada. Berikut ini adalah spesifikasi lengkap dari Arduino UNO R3 :

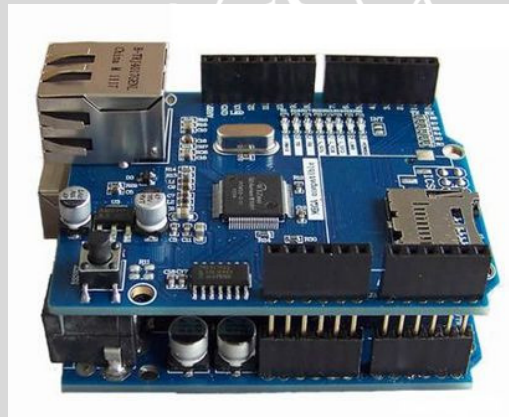
Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader

SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Tabel 2.1. Spesifikasi Arduino UNO R3

### 2.5.2 Ethernet Shield

Arduino dapat dihubungkan dengan *board* yang lain atau biasa disebut *arduino shield* sehingga fungsi dari arduino tersebut dapat diperluas lagi. Salah satu jenis *shield* yang ada adalah *ethernet shield*. Dengan menggunakan *ethernet shield*, arduino dapat terhubung dengan jaringan komputer. *Ethernet shield* yang banyak ditemui di pasaran adalah *ethernet shield* yang berbasis chip WIZNET W5100.



Gambar 2.10. Ethernet shield

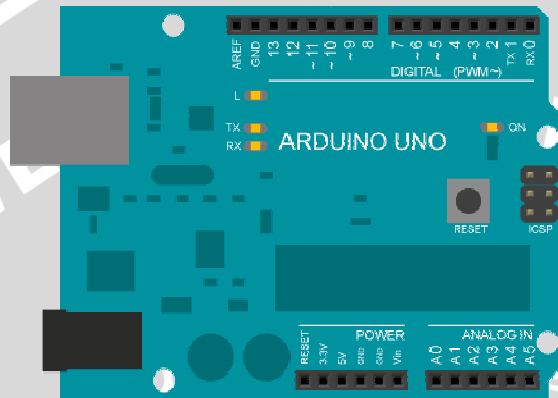
### 2.5.3 Input - Output pada Arduino

Arduino UNO R3 memiliki 14 Digital input atau output pin dan 6 analog input pin. Arduino memiliki standard penomoran pin-pin yang dimiliki. Berikut ini adalah penjelasan tentang fungsi pin-pin tersebut :

- Serial : 0 (RX) dan 1 (TX)
- External Interrupts : 2 dan 3
- PWM : 3, 5, 6, 9, 10 dan 11
- SPI : 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK)
- Analog Input : A0 – A5

Arduino Ethernet Shield membutuhkan beberapa pin pada arduino untuk bekerja. Sehingga pin tersebut tidak dapat dipakai pada aplikasi arduino. Berikut ini adalah daftar pin yang digunakan Ethernet Shield :

- D2 : Ethernet Interrupt
- D10 – D13 : SPI
- A0 – A1 : SD Card (Jika digunakan)



Gambar 2.11. Konfigurasi pin pada Arduino UNO R3

#### 2.5.4 Arduino Programming Language (APL)

Arduino dapat diprogram dengan menggunakan *Arduino Programming Language*. *Arduino Programming Language* (APL) mengimplemetasi Bahasa C++ dengan menyediakan fungsi-fungsi khusus untuk arduino hingga terbentuk sebuah *framework* yang dapat memudahkan pemrogram untuk memprogram arduino.

Struktur program APL mirip dengan struktur program bahasa C++, hanya saja pada implementasinya APL tidak mengenal fungsi *main* yang merupakan sebuah keharusan dalam bahasa C++. Struktur program APL adalah sebagai berikut :

- **Fungsi *setup***

Fungsi yang pertama kali dieksekusi oleh arduino. Fungsi *setup* ini hanya dieksekusi satu kali setiap arduino diaktifkan atau setelah melakukan *reset*.

- **Fungsi loop**

Fungsi yang dieksekusi berkali-kali sejak arduino diaktifkan hingga dimatikan.

```
1  include <SPI.h>
2
3  int pin = 13;
4
5  void setup(){
6      pinMode(pin, output);
7  }
8
9  void loop(){
10     digitalWrite(pin, HIGH);
11     delay(1000);
12     digitalWrite(pin, LOW);
13 }
```

Gambar 2.12. Contoh program pada arduino

### 2.5.5. Fungsi Input - Output pada arduino

Pada *Arduino Programming Language* (APL) terdapat beberapa fungsi input dan output. Berikut ini adalah beberapa fungsi yang dimiliki APL :

- **Pin Mode**

Fungsi untuk menentukan mode dari pin tertentu. Sintaks dari fungsi tersebut adalah *pinMode(pin, mode)*. Fungsi tersebut memiliki parameter nomor pin yang akan diubah mode nya dengan tipe data *integer* dan mode pin yang merupakan konstanta dengan nilai OUTPUT atau INPUT. Fungsi *pinMode* tidak memiliki nilai kembalian (*return*). Jika pada program tidak terdapat fungsi *pinMode* maka semua pin dianggap sedang dalam mode input.

- **Digital Read**

Fungsi untuk membaca nilai input dari pin digital tertentu. Sintaks dari fungsi tersebut adalah *digitalRead(pin)*. Fungsi tersebut memiliki parameter nomor pin digital yang akan dibaca dengan tipe data *integer*. Fungsi *digitalRead* mempunyai nilai kembalian (*return*) konstanta dengan nilai HIGH atau LOW.



- **Digital Write**

Fungsi untuk memberikan nilai keluaran pada pin digital tertentu. Sintaks dari fungsi tersebut adalah *digitalWrite(pin, value)*. Fungsi tersebut memiliki parameter nomor pin yang akan diberi nilai keluaran dengan tipe data *integer* dan nilai yang akan ditulis yang merupakan konstanta dengan nilai HIGH atau LOW. Fungsi *digitalWrite* tidak memiliki nilai kembalian (*return*).

- **Analog Read**

Fungsi untuk membaca nilai input dari pin analog tertentu. Sintaks dari fungsi tersebut adalah *analogRead(pin)*. Fungsi tersebut memiliki parameter nomor pin analog yang akan dibaca dengan tipe data *integer*.

Arduino memiliki 10-bit Analog-to-Digital Converter yang berarti tegangan 0 – 5 volt akan direpresentasikan dalam nilai 0 – 1023. Fungsi *analogRead* mempunyai nilai kembalian (*return*) 0 – 1023 dengan tipe data *integer*.

- **Analog Write**

Fungsi untuk memberikan nilai keluaran pada pin tertentu. Arduino akan membangkitkan sinyal PWM dengan frekuensi 490 Hz pada pin yang dituju. Sintaks dari fungsi tersebut adalah *analogWrite(pin, value)*. Fungsi tersebut memiliki parameter nomor pin yang akan diberi nilai keluaran dengan tipe data *integer* dan nilai representasi *duty cycle* dari sinyal PWM antara 0 – 255 dengan tipe data *integer*. Fungsi *analogWrite* tidak memiliki nilai kembalian (*return*).

Fungsi	Parameter	Return
<i>digitalRead</i>	Nomor pin	HIGH / LOW
<i>analogRead</i>	Nomor pin	0 - 1023
<i>digitalWrite</i>	- Nomor pin - Nilai (HIGH / LOW)	-
<i>analogWrite</i>	- Nomor pin - Duty Cycle (0 - 255)	-
<i>pinMode</i>	- Nomor pin - Mode (INPUT / OUTPUT)	-

Gambar 2.13. Fungsi input dan output pada arduino

### 2.5.6. Delay Total

*Delay* pada sistem ini adalah waktu yang dibutuhkan oleh client untuk mengirimkan sebuah request hingga menerima response. Adapun batasan *delay* yang terjadi bisa dikelompokkan sebagai berikut (Monteleus, 2001) :

- *Delay* di bawah 150 ms tidak akan terasa pengaruhnya oleh *client*
- *Delay* antara 150 – 400 ms masih dapat ditolerir
- *Delay* di atas 400 ms tidak dapat ditolerir.

UNIVERSITAS BRAWIJAYA

