

BAB II

LANDASAN TEORI

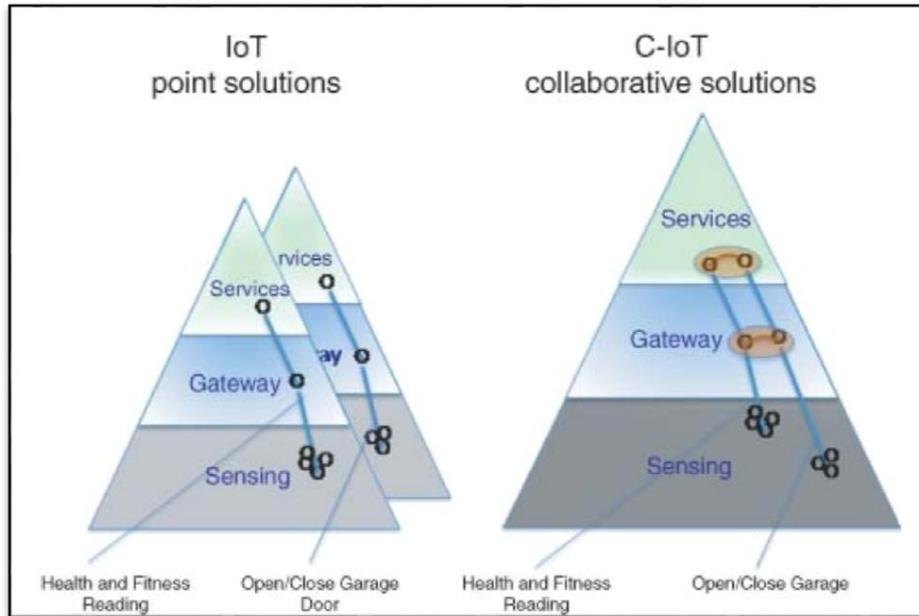
1.1 IoT (Internet of Thing)

Menurut Fawzi Behmann dan Kwok Wu : *Internet of Thing* atau IoT adalah sebuah istilah yang dimaksudkan dalam penggunaan internet yang lebih besar , mengadopsi komputasi yang bersifat mobile dan konektivitas kemudian menggabungkannya kedalam kesehari-harian dalam kehidupan kita.

IoT berkaitan dengan DoT (*Disruption of Things*) dan sebagai pengantar perubahan atau transformasi penggunaan internet dari sebelumnya *Internet of People* menjadi *Internet of M2M* (*Maching-to-Machine*).

Sedangkan C-IoT adalah singkatan dari *Collaborative Internet Of Thing* adalah sebuah hubungan dari dua point solusi menjadi tiga point secara cerdas, sebagai contohnya adalah iWatch salah satu *smartwatch* tidak hanya memanage kesehatan dan kebugaran tetapi juga dapat menyesuaikan suhu ruangan pada AC mobil.

Pada model C-IoT dalam bentuk sederhanya terdiri dari *Sensing*, *Gateway*, dan *Services*. Pengindraan (*Sensing*) akan memasukan apa yang di anggap penting, *Gateway* akan menambah kecerdasan dan konektifitas untuk tindakan yang akan di ambil baik tingkatan lokal atau menyampaikan informasinya ke Cloud level, sedangkan *Services* akan menangkap informasi dan mencerna, menganalisa, dan mengembangkan wawasan untuk membantu meningkatkan kualitas hidup atau improve business operation.

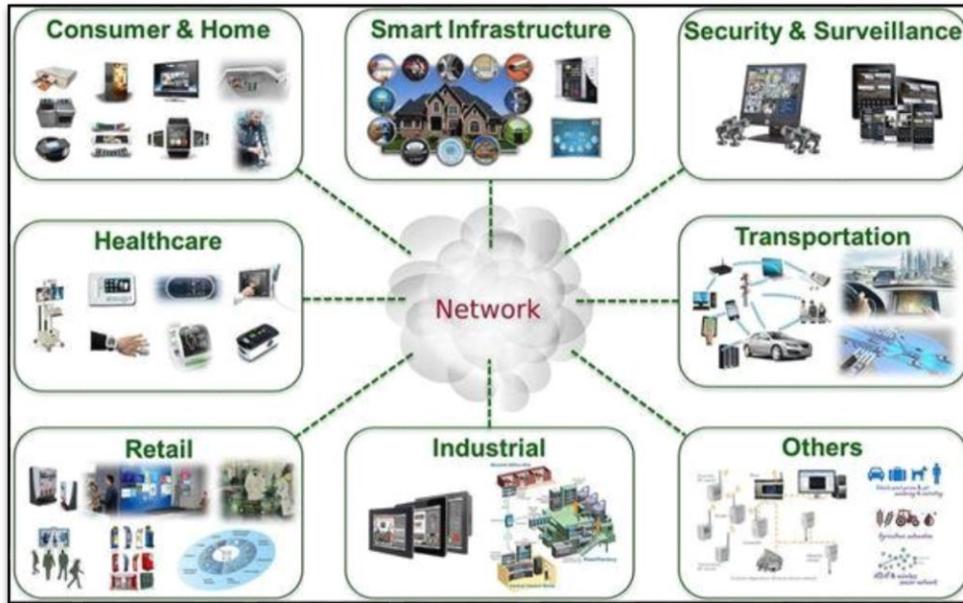


Gambar 2.1 Perbedaan IOT dan C-IoT

(Source : *E-Book Collaborative Internet of Things (C-IoT) for Future Smart Connected Life and Business*)

a) IoT (*Internet Of Thing*) & IoE (*Internet of Everything*)

IoE (*Internet Of Everything*) adalah konsep hal perangkat cerdas yang terhubung bersama-sama dan melakukan apa yang tidak bisa dilakukan untuk teknologi sebelumnya. Pergeseran paradigma tersebut menandai era dimana kesempatan untuk menjadi lebih baik untuk semua orang, mulai dari konsumen dan pelaku bisnis sampai pemerintahan.



Gambar 2.2 Ilustrasi Semua Terhubung Internet

(Source : *E-Book Collaborative Internet of Things (C-IoT) for Future Smart Connected Life and Business*)

IoE (*Internet of Everything*) akan mengubah dunia dan berdampak pada kehidupan sehari – hari baik siang ataupun malam akan dikelilingi oleh konektivitas yang cerdas merespon apa yang kita inginkan dan apa yang kita butuhkan yang diistilahkan *Digital Sixth Sense*.

Cisco mendefinisikan bahwa IoE (*Internet of Everything*) sebagai penyatu masyarakat , proses , data , dan yang lainnya menjadi terhubung bersama-sama dan menjadi lebih berharga dari sebelumnya.

1.2 UML (*Unified Modeling Language*)

Menurut Sri Dwarwiyanti dan Romi Satria Wahono, UML adalah sebuah bahasa yang telah menjadi standart dalam industri untuk visualisasi, merancang, dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membentuk model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman.

UML lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, java, dan sebagainya, namun UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C. UML mendefinisikan notasi dan *syntax / semantik*. Notasi UML merupakan sekumpulan bentuk khusus yang menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat di kombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada, yaitu : *Object-Oriented Design, Object-Modeling Technique* dan *Object-Oriented Software Engineering*.

Adapun konsepsi dasar dari UML adalah sebagai berikut :

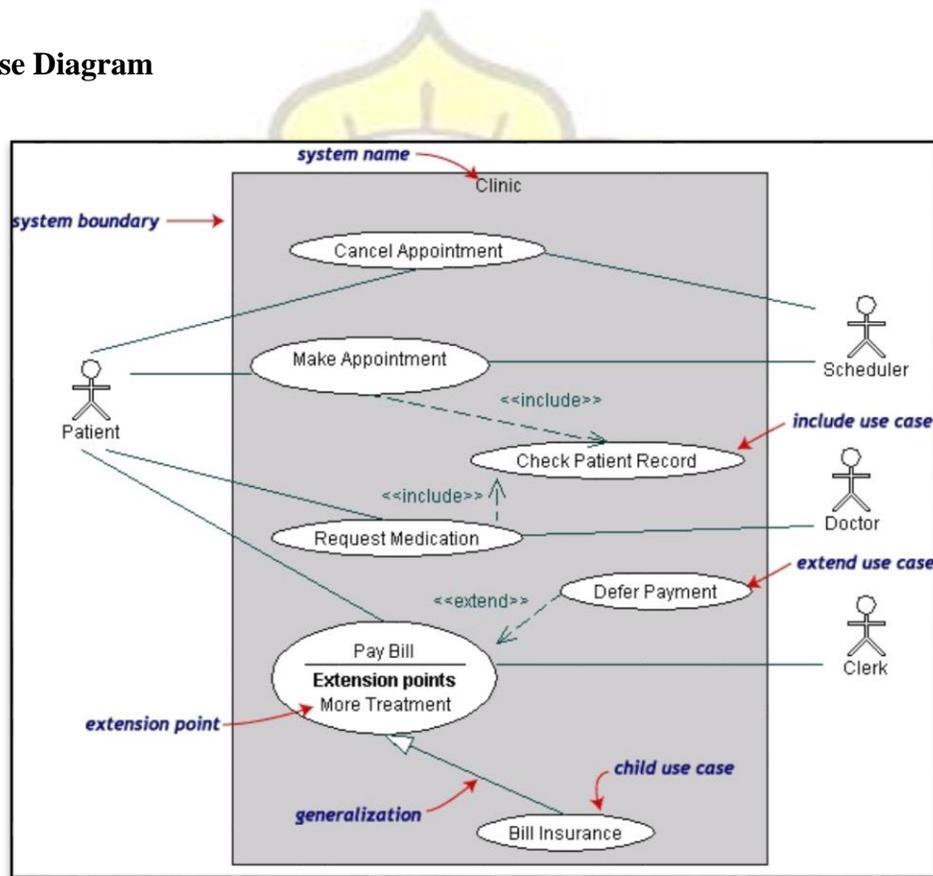
Tabel 2.1 Konsep Dasar UML

<i>Major Area</i>	<i>View</i>	<i>Diagrams</i>	<i>Main Concepts</i>
<i>Structural</i>	<i>Static View</i>	<i>Class Diagram</i>	<i>class, association, generalizations, dependency, realization, interface</i>
	<i>Use Case View</i>	<i>Use Case Diagram</i>	<i>use case, actor, association, extend, include, use case generalization</i>
	<i>Implementation View</i>	<i>Component Diagram</i>	<i>component, interface, dependency, location</i>
	<i>Deployment View</i>	<i>Deployment Diagram</i>	<i>node, component, dependency, locations</i>
<i>Dynamic</i>	<i>State Machine View</i>	<i>Statechart Diagram</i>	<i>state, event, transition, action</i>
	<i>Activity View</i>	<i>Activity Diagram</i>	<i>state, activity, completion transition, fork, join</i>
	<i>Interaction View</i>	<i>Sequence Diagram</i>	<i>interaction, object, message, activation</i>

		<i>Collaboration Diagram</i>	<i>collaboration, interaction, collaboration role, message</i>
<i>Model Management</i>	<i>Model Management View</i>	<i>Class Diagram</i>	<i>package, subsystem, model</i>
<i>Extensibility</i>	<i>all</i>	<i>All</i>	<i>constraint, stereotype, tagged values</i>

Abstraksi konsep dasar UML terdiri dari *structural classification, dynamic behavior, model management*. Sedangkan *main concept* dipandang sebagai yang akan muncul pada saat kita membuat diagram. adapun diagram-diagram untuk mendefinisikan UML adalah :

1.2.1 Use Case Diagram



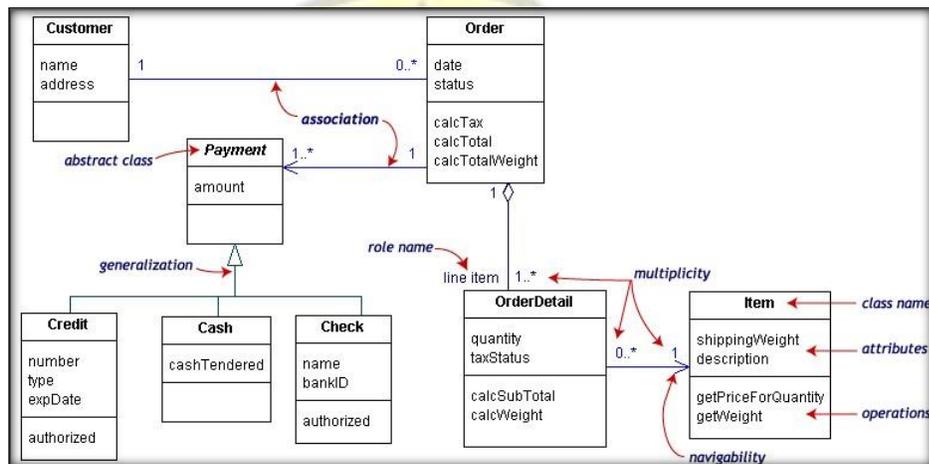
Gambar 2.3 Contoh Use Case Diagram

(Source : E-Book Pengantar Unified Modeling Language (UML))

Use Case diagram menggambarkan fungsionalitas yang di harapkan dari sebuah sistem, yang ditekankan adalah “apa” yang di perbuat sistem, dan bukan “bagaimana”.

Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat di-include oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*.

2.2.2 Class Diagram



Gambar 2.4 Contoh Class Diagram

(Source : E-Book Pengantar Unified Modeling Language (UML))

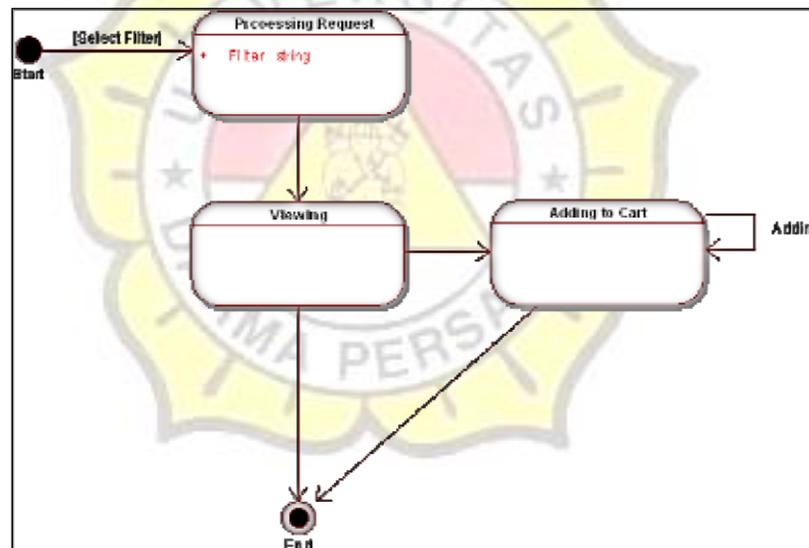
Class Diagram sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merukan inti dari pengembangan dan desain berorientasi objek. Class Diagram menggambarkan keadaan (attribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu

sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok Nama, Atribut , dan Metoda.

Atribut dan metoda dapat memiliki salah satu sifat berikut :

- a) *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
- b) *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
- c) *Public*, dapat dipanggil oleh siapa saja

2.2.3 Statechart diagram



Gambar 2.5 Contoh Statechart Diagram

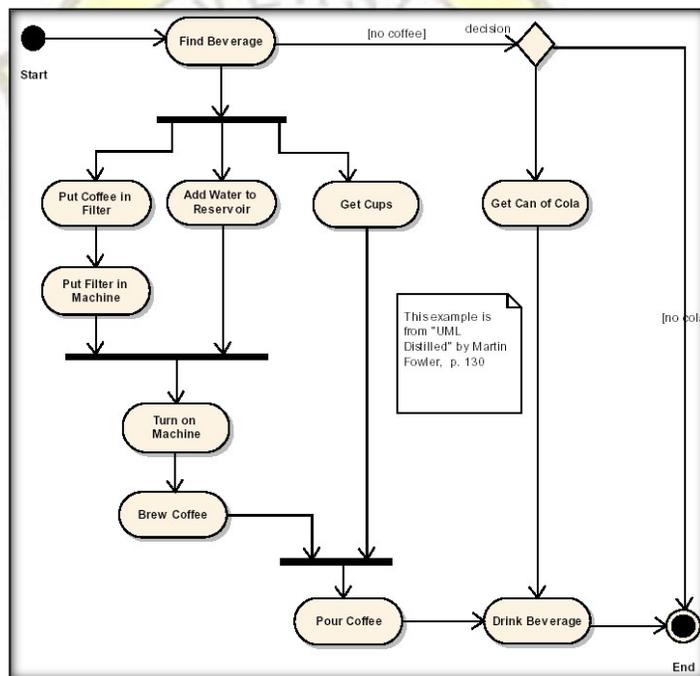
(Source : E-Book Pengantar Unified Modeling Language (UML))

State Diagram menggambarkan transisi dan perubahan keadaan (dari suatu state ke state lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada

umumnya statechart diagram menggambarkan class tertentu (satu class dapat memiliki lebih dari satu statechart diagram).

Dalam UML, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring, Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah.

2.2.4 Activity diagram



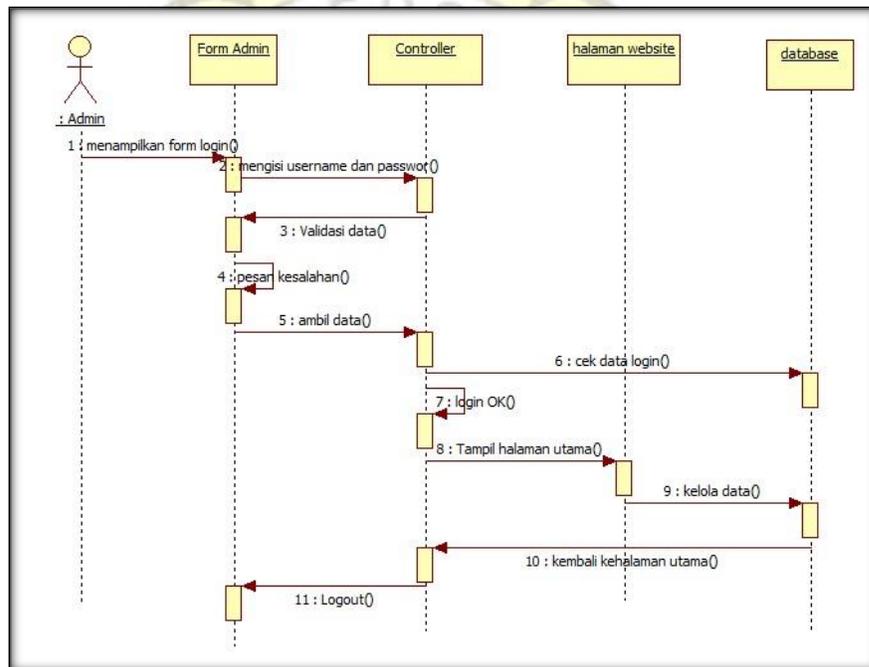
Gambar 2.6 Contoh Activity Diagram

(Source : E-Book Pengantar Unified Modeling Language (UML))

Activity Diagram menggambarkan alir aktivitas dalam sistem yang sedang dirancang bagaimana masing masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir.

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum

2.2.5 Sequence diagram



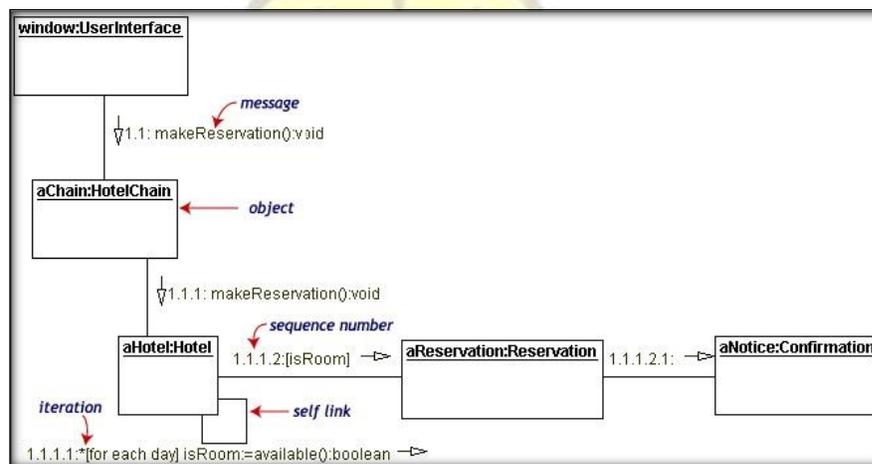
Gambar 2.7 Contoh Squence Diagram

(Source : E-Book Pengantar Unified Modeling Language (UML))

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu.

2.2.6 Collaboration diagram

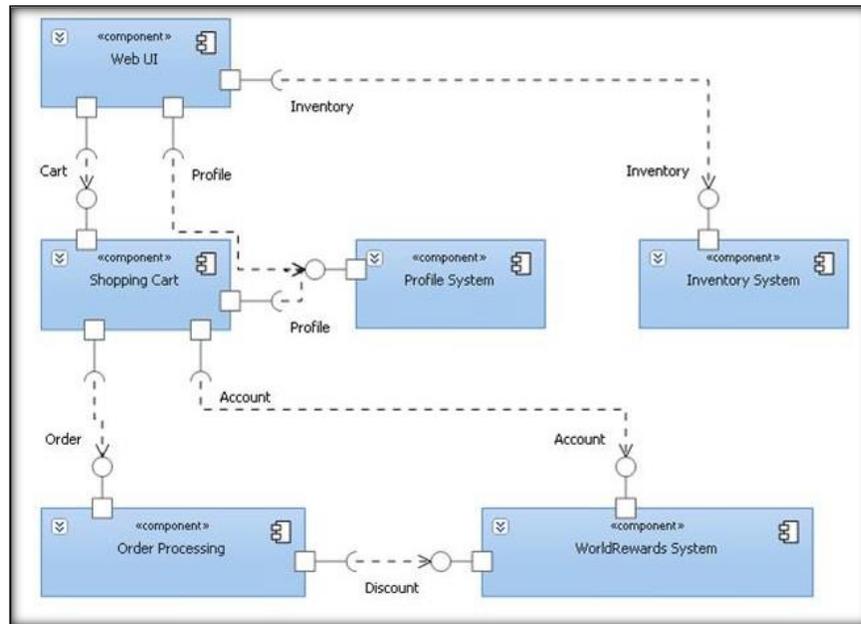


Gambar 2.8 Contoh Collaboration Diagram

(Source : E-Book Pengantar Unified Modeling Language (UML))

Collaboration diagram menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor 1. Messages dari level yang sama memiliki prefiks yang sama.

2.2.7 Component diagram

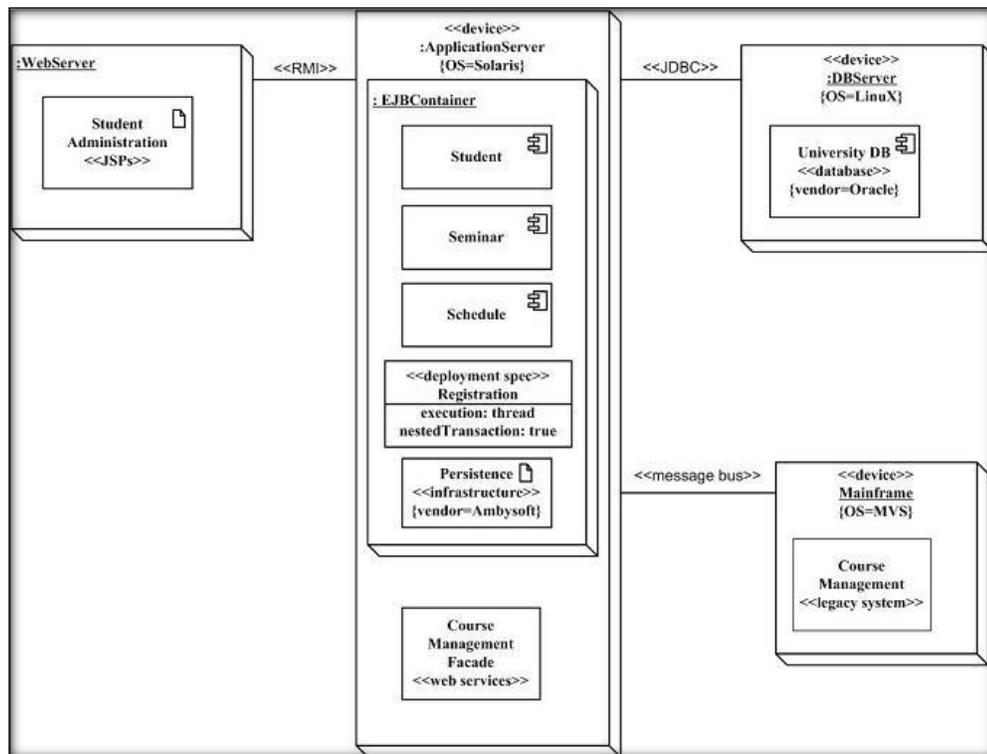


Gambar 2.9 Contoh Componen Diagram

(Source : *E-Book Pengantar Unified Modeling Language (UML)*)

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya.

2.2.8 Deployment diagram



Gambar 2.10 Contoh Deployment Diagram

(Source : E-Book Pengantar Unified Modeling Language (UML))

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisik.

1.3 Smart Home

Menurut Fawzi Behmann dan Kwok Wu : Smart home adalah sebuah konsep rumah cerdas yang memanfaatkan teknologi secara maksimal, dengan kontrol terpusat untuk

mengendalikan dua atau lebih sistem individu yang bertujuan meningkatkan efisiensi, kenyamanan dan keamanan.



Gambar 2.11 Ilustrasi Smart Home – Remote

(Source : E-Book Pengantar Unified Modeling Language (UML))

beberapa contoh penerapan smart home yang di bagi 4 kategori :

a) Kontrol Lingkungan

Penerapan konsep smart home pada area lingkungan rumah seperti pada taman dan pekarangan. memberikan beberapa informasi seperti suhu, kelembapan udara, dektektor banjir, tirai elektrik, dan sebagainya.

b) Efisiensi Energy

Penerapan konsep smart home untuk mengefisiensi sumber energi seperti listrik , air , gas, lemari es, kompor, pencuci piring, mesin cuci, dan sebagainya. Baik secara otomatis atau remote.

c) Keamanan dan Keselamatan (Indor / Outdoor)

Penerapan konsep smat home untuk keamana dan keseleamatan seperti detektor gerak, kunci pintu, jendela , garasi,dll. Baik secara otomatis atau remote.

d) Kesehatan dan Kebugaran

Penerapan konsep smart home untuk kesehatan dan kebugaran seperti iWatch yang dapat menyesuaikan suhu ruangan berdasarkan suhu badan, mencegah penyakit kronis dengan memberikan sebuah informasi kesehatan, dan sebagainya

1.4 Mikrokontroler Arduino

Mikrokontroler menurut Gunther Gridling dan Bettina Weiss adalah sebuah prosesor dengan memory dan kompenen lainnya yang dikemas dalam satu chip / IC. Adapun *Basic Layout* dari mikrokontroler adalah :

- a) Inti Prosesor
- b) Memori
- c) Interupsi Kontroler
- d) Timer / Counter
- e) Digital Input / Output
- f) Analog Input / Output
- g) Interfaces
- h) Wachdog Timer
- i) Debugging Unit

Adapun secara garis besar mikrokontroler adalah (serangkaian) prosesor yang dilengkapi dengan memori, timer, (paralel) I / O pin dan chip peripheral lainnya.

Menurut Jazi Eko Prasetyo : Teknologi mikrokontroler sebagai salah satu produk teknologi semikonduktor berkontribusi besar untuk menunjang aktivitas manusia. Sejumlah parameter kondisi bisa dipantau menggunakan teknologi mikrokontroler sebagai front-end device dengan melibatkan perangkat berspesifikasi tinggi.



Gambar 2.12 Arduino Uno

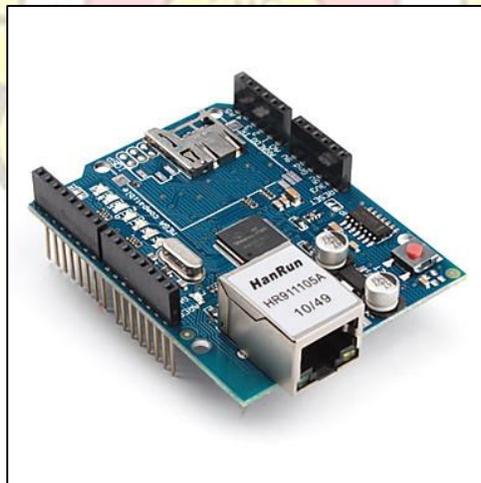
Arduino merupakan salah satu dari sekian produk edukasi mikrokontroler sebagai proyek rintisan berlisensi terbuka dan mampu difungsikan sebagai produk akhir. Dalam modul juga disertai komponen elektronika, struktur serta antarmuka Arduino yang sederhana memberi kemudahan pengguna dalam memahami parameter (visualisasi maupun non-visual) seperti konsep sensor atau penerapan sensor elektronik yang tidak selalu bisa di amati langsung seperti infra merah.

Konsep bahasa Arduino dengan mentargetkan ke pin tertentu (secara fisik) menjadikan lebih mudah dipahami oleh berbagai kalangan. Bagi programmer tingkat lanjut, hal ini mungkin merupakan halangan, misalnya optimasi kecepatan eksekusi pada kisaran mikrodetik yang hanya bisa dilakukan menggunakan bahasa C atau Assembler. Meskipun demikian, pengguna bahasa

Arduino dapat menjadi nilai tambah, yaitu efisiensi waktu desain prototype dan kemudahan penerapan yang lebih luas tanpa perlu pemahaman dasar fungsi-fungsi mikrokontroler.

Arduino juga memberikan kemudahan ekspansi sistem menggunakan sistem komunikasi yang sederhana dan efektif. Dalam penerapan memakai konsep “Shield”, sebab modul-modul eksternal arduino bersifat simple constumable dan sebagian besar produk modul ekspansi tidak perlu pensolderan sama sekali.

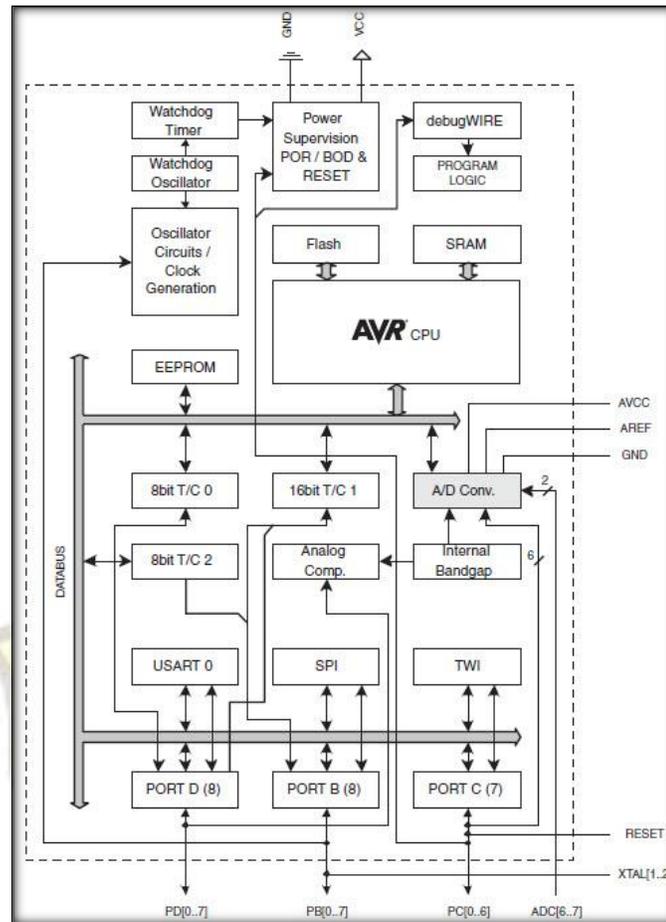
Konsep Shield pada arduino dapat di ibaratkan dengan sebuah plug-in pada browser, arduino shield adalah device yang dapat dengan mudah dibuka dan dilepas pada device inti arduino contoh penggabungan seperti arduino uno dan arduino ethernet shield yang penulis akan terapkan, secara umum arduino uno tidak memiliki fitur device untuk colokan kabel rj45 namun dengan fitur tambahan dari arduino ethershield memungkinkan menerapkan konsep jaringan melalui ethernet rj45.



Gambar 2.13 Ethernet Shield

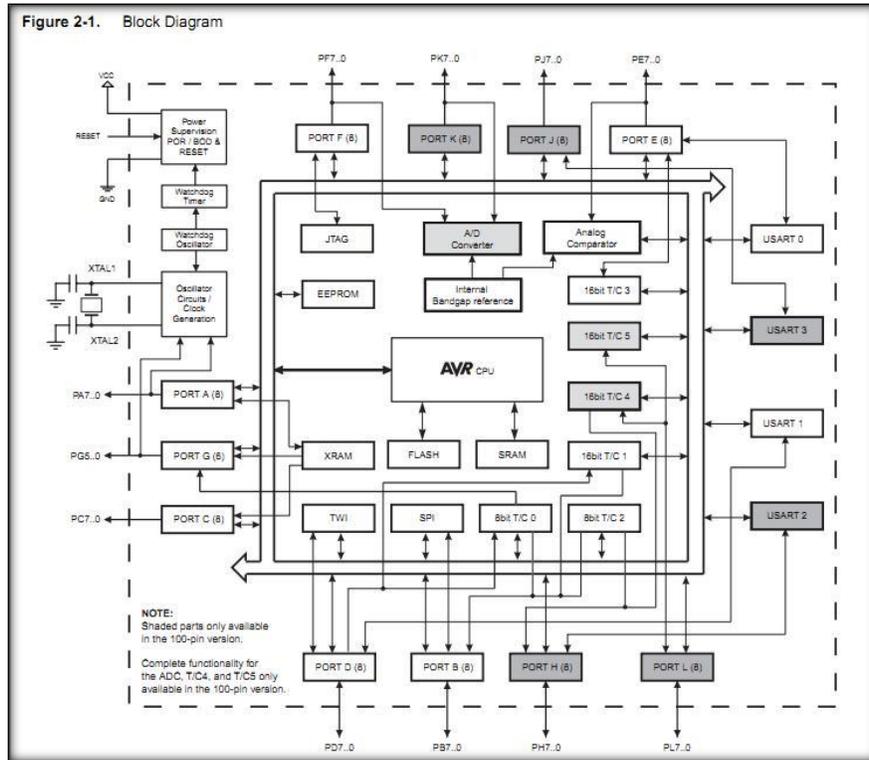
Seluruh mikrokontroler yang diimplementasikan pada produk Arduino menggunakan ATmega keluaran AVR. Sala satunya seri ATmega 328 dengan sejumlah fitur di antara *On-Chip*

System Debug, 5 Mode Sleep, 6 saluran ADC yang mendukung reduksi derau,, ragam hemat daya (Power-save Mode, Power-down), dan Ragam Standby Mode.



Gambar 2.14 ATmega 328

Mikrokontroler ATmega 328 paling umumnya digunakan pada board Arduino seperti UNO, Duemilanove, Nano, dsb. Sedangkan ATmega 1280 dipakai pada tipe Arduino Mega.



Gambar 2.15 ATmega 1280

baik ATmega 328 maupun ATmega 1280, keduanya menggunakan kristal 16Mhz sebagai pembangkit clock. Keduanya juga memiliki block memory Flash untuk penyimpanan instruksi program, SRAM untuk penyimpanan variable data sementara dan EEPROM sebagai media penyimpanan data yang tetap tersimpan meskipun mikrokontroler dalam kondisi tidak di catu. Fitur mikokontroler AVER seri lainnya seperti ATmega 168 atau 2560 tidak jauh berbeda dengan 328 atau 1280, kecuali pada ukuran kapasitas neniru EEPROM, Flash, dan SRAM.

1.5 PEMROGRAMAN

2.5.1 HTML

Menurut buku Abdul kadir : "HTML (*Hypertext Markup Language*) adalah bahasa script yang digunakan untuk menyusun halaman Web. Sejauh ini ada sejumlah standart HTML, antara lain HTML 4, XHTML, HTML 5 ". Contoh script HTML :

```
<html>
<head>
  <title>Contoh</title>
</head>
<body>
<h1> Abdul Jabar Hakim</h1>
</body>
</html>
```

Penulisan script HTML biasa diawali dengan `<html>` dan `</html>` sebagai penutup script, sebagian script HTML selalu memiliki tanda pembuka dan penutup seperti `<title>.....</title>` script yang berfungsi untuk membuat judul pada dokument HTML. Extension HTML pada umumnya adalah “.html”.

2.5.2 CSS

Menurut buku Abdul kadir : “CSS (*Cascading Style Sheet*) adalah skrip yang di tunjukan secara khusus untuk mengatur tampilan aplikasi web”.

Penulisan script css memiliki dua teknik, Internal dan External. penulisan script Internal adalah penggunaan script css yang dimasukan secara langsung pada halaman website yang sedang di design dengan menambahkan `<style> </style>` di dalam `<head>`, sedangkan penulisan script External adalah penggunaan script css yang diletakan terpisah pada halaman website yang sedang di design. Contoh script CSS :

```
.table{
  text-align: center;
}

thead{
```

```
background-color: #f8f8f8;
}
```

Penulisan script CSS memiliki sebuah aturan seperti *Selector*, *Declaration*, *Property*, dan *Value*. Untuk gambar di atas yang dimaksud dengan *selector* adalah *body*, *table*, *tr*, *td*, *.container*. *declaration* adalah `background-color: blue;` , `padding: 5px` , `text-align :center;` . sedangkan *Property* adalah `background-color`, `padding`, `text-align` dan *Value* adalah `blue`, `5px`, `center`. Extension CSS pada umumnya adalah “.css” .

2.5.3 PHP

Menurut buku Lukman Hakim : “PHP adalah singkatan dari *Personal Home Page* yaitu bahasa pemrograman yang di gunakan untuk membuat aplikasi web. PHP disebut sebagai *server side scripting*, artinya skrip PHP dijalankan di sisi server, dimana setelah skrip PHP di olah di server, hasilnya dikirimkan ke broser (klien).”. Contoh script PHP

```
<?php
$nama = “Hakim”;
$umur = 22;
echo “Hai nama saya “.$nama.”<br />”;
echo “Umur saya saat ini ”.$umur;
?>
```

Penulisan script PHP selalu diawali dengan `<?php` dan `?>` sebagai penutup script. Salah satu kelebihan PHP adalah dapat berjalan di berbagai macam sistem operasi seperti Linux, Windows, atau Mac. Extension untuk file PHP pada umumnya adalah “.php”.

2.5.4 Mysql

Menurut buku Lukman Hakim : “Mysql adalah salah satu software database relasional yang di design khusus untuk menangani pengelolaan database di lingkungan web”, walaupun pada kenyataannya mysql di gunakan juga untuk aplikasi berbasis desktop contohnya adalah VB atau Visual Basic. Contoh script Mysql :

```
SELECT * FROM mahasiswa where nama="hakim";
```

Penggunaan sintaks Mysql hampir sama dengan SQL karena Mysql merupakan turunan salah satu konsep SQL. Penyisipan sintaks mysql pada PHP biasa di kenal dengan Query. Contoh penerapan Mysql pada PHP :

```
$contohquery = "INSERT INTO log (username, waktu, nama_kontrol, posisi_terakhir) VALUES ('$username', '$waktu', 'Lampu 1', 'on')"
```

```
$stampung = mysql_query($contohquery);
```

2.5.5 Mysqli

Menurut Buku Lukman Hakim : “Ekstensi mysql paling populer dikalangan programer PHP, namun sejak PHP versi 5.5 tidak di anjurkan lagi untuk di gunakan, sehingga ada peringatan resmi dari pengembang PHP mengenai bahwa extention mysql dalam proses *depreceted* (akan di hapus pada versi berikutnya)”. Oleh karena itu penulis menggunakan mysqli sebagai extention php. Contoh penerapan Mysqli pada PHP:

```
$contohquery = "INSERT INTO log (username, waktu, nama_kontrol, posisi_terakhir) VALUES ('$username', '$waktu', 'Lampu 1', 'on')"
```

```
$stampung = mysqli_query($koneksi,$contohquery);
```

Terlihat digambar penggunaan sintaks mysql dan mysqli tidak begitu berbeda di PHP, sehingga pengguna yang terbiasa menggunakan mysql tidak akan kesulitan untuk berpindah ke mysqli. Contoh apabila mysql menggunakan mysql_query untuk mengeksekusi query maka dengan mysqli menggunakan mysqli_query, perbedaannya hanya menambahkan “ i “ .

2.5.6 JavaScript

Menurut buku Abdul Kadir : JavaScript adalah bahasa script yang di gunakan untuk mengontrol tindakan tindakan yang di butuhkan di halaman web. Perkembangan javascript cukup pesat sehingga muncul library yang di dasarkan oleh javascript seperti JQuery serta teknik pemrograman yang makin populer seperti Ajax dan JSON. Contoh script JavaScript :

```
<script>  
var x = 5;  
var y = 6;  
var z = x + y;  
document.getElementById("demo").innerHTML = z;  
</script>
```

Seperti layaknya CSS penulisan script javascript memiliki dua teknik, Internal dan External. penulisan script Internal adalah penggunaan script javascript yang dimasukan secara langsung pada halaman website dengan menambahkan <script> untuk pembuka dan </script> untuk penutup, sedangkan penulisan script External adalah

penggunaan script javascript yang diletakan terpisah pada halaman. Extention javascript secara umum adalah “.js”.

JQuery pada dasarnya adalah daftar pustaka dari JavaScript, artinya penggunaan JQuery adalah untuk menyederhanakan script-script Javascript menjadi lebih pendek. Contoh script JQuery :

```
$("#button").click(function(){  
    $("#p").hide("slow",function(){  
        alert("The paragraph is now hidden");  
    });  
});
```

Jquery dirilis pada awal tahun 2006 dengan lisensi ganda MIT dan GPL. Dengan adanya JQuery tersebut munculah beberapa metode-metode untuk membuat sebuah website menjadi lebih interaktif seperti Ajax dan JSON.

Contoh penggunaan Ajax seperti Berikut :

```
$("#lampuhidup1").click(function(){  
$.ajax({  
    url: "proses/lampu1hidup.php",  
    success: function(response)  
    {  
        if(response == "success") {  
            $("#lampu1mati").fadeOut();  
        }  
    }  
});
```

```

        $("#lampu1hidup").fadeIn();
    }
    else
        alert("Terjadi Kesalahan");
    }
});
return false;
});

```

Maksud dari penggunaan ajax di atas adalah apabila ID dengan nama lampu1hidup1 di klik maka akan menjalankan proses yang berada pada file proses dengan nama file lampu1hidup.php, apabila lampu1hidup.php memberikan output “*success*” atau proses berhasil dilakukan maka akan menyembunyikan ID dengan nama lampu1mati dan memunculkan ID dengan nama lampu1hidup, apabila proses tidak berhasil atau tidak ada output “*success*” maka akan muncul sebuah alert atau pop up yang berulisan “Terjadi Kesalahan”. Terlihat jelas di sini bahwa dengan menggunakan jquery dan ajax maka akan membuat website kita menjadi lebih interaktif.

Sedangkan JSON singkatan dari *JavaScript Object Notation* dengan cara kerja dan fungsi utama seperti XML untuk mentransfer data antar device / OS / bahasa pemrograman yang berbeda dan sebagai media penyimpanan data. Contoh penggunaan JSON pada JQuery :

```

var data_table = "";
var data_user = "";
$.ajax({
    url: "json/json_master_guru.php",
    dataType: "json",
    success: function(data) {
        for (var i =0; i<data.length; i++){

```

```

data_table += "<tr
                class='"+data[i].id_guru+"' id='"+data[i].id_guru+"-1'><td> "
                +data[i].id_guru+"</td> <td>"
                +data[i].nama+"</td> <td>"
                +data[i].mata_pelajaran+"</td><td>
                "</tr>";
                data_user += "<tr class='"+data[i].id_user+"'
                id='"+data[i].id_user+"-2'><td> "
                +data[i].id_user+"</td> <td>"
                +data[i].username+"</td> <td>"
                +data[i].password+"</td> <td>"
                +data[i].level+"</td></tr>";
            }
            $('#master_guru').append(data_table);
            $('#master_user').append(data_user);
        }
    });

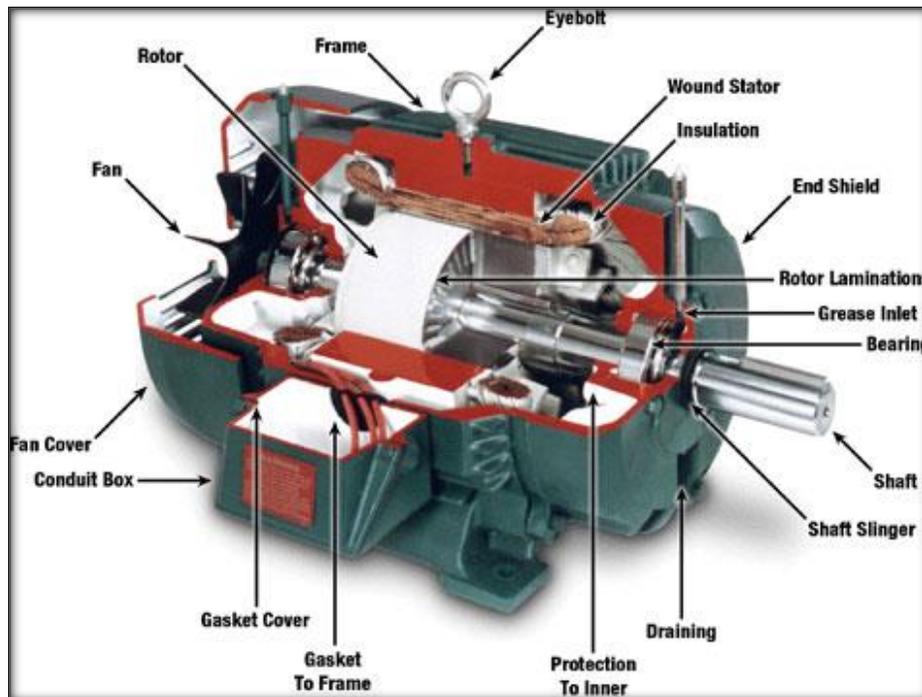
```

Dengan mengkombinasikan Ajax dan JSON maka akan membuat sebuah web lebih interaktif khususnya pada pengelolaan data. untuk mereload sebuah data yang di proses pada umumnya seseorang harus mererefresh browser, tapi dengan bantuan JSON setiap proses bisa di atur ketika berhasil yang di reload adalah datanya bukan halaman.

1.6 Motor DC DAN Motor Stepper

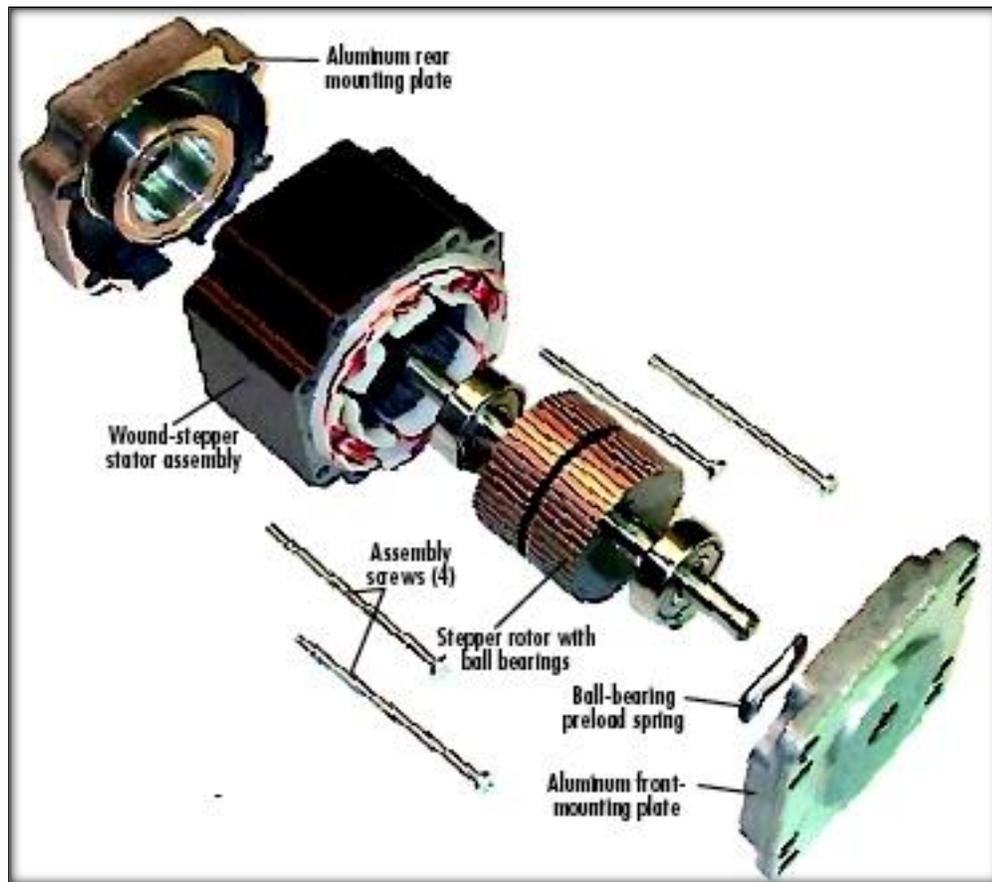
Menurut Jazi Eko Istiyanto : Cara kerja motor DC sama seperti dinamo pada kipas pending pada PC atau dinamo pada mainan remote control. Motor akan berputar ke satu arah jika di beri masukan tegangan DC, Sedangkan motor Stepper hanya berputar jika diberikan masukan pulsa paralel berurutan dari 0 hingga nilai bit maksimal (sesuai jumlah koneksi yang tersedia) jika ada 4 konektor maka putaran motor stepper akan dimulai dengan masukan logika 0000, 0001, 0010, hingga 1111 kemudian kembali ke 0000 lagu dan seterusnya. Jika dimasukan nilai bernilai acak atau tidak berurutan, motor

stepper tidak akan bergerak berputar. Pada umumnya motor DC beroperasi pada tegangan 1,5V – 4,5 V, dengan putaran maksimal sekitar 12800rpm yang dapat dicapai jika dicatu pada tegangan 4,5V dengan arus paling sedikit 320Ma.



Gambar 2.16 Motor DC

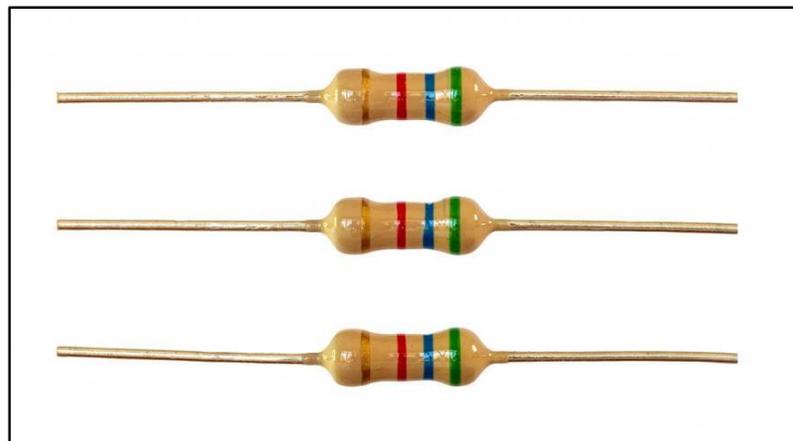
Sedangkan motor Stepper bisa di gerakan menggunakan mekanisme yang berbeda dengan motor DC, sehingga di perlukan rancangan skema rancangan serta sketch Arduino yang berbeda dari penggerak motor DC. Motor stepper bisa berputar di kedua arah, dengan memakai IC L293D atau SN7754410 dengan saluran OUT1 sampai OUT 4 untuk masukan IN1 samapi IN4 motor stepper. Pin kontrol enable ENA dan ENB terhubung ke VCC = 5V, sedangkan IN1, IN2, IN3, IN4 secara berurutan di hubungkan ke D5,D4,D3,dan D2. Apabila menggunakan model SM-42BYG011-25 harus bertipe bipolar.



Gambar 2.17 Motor Stepper

1.7 KOMPONEN ELEKTRIK

2.7.1 RESISTOR

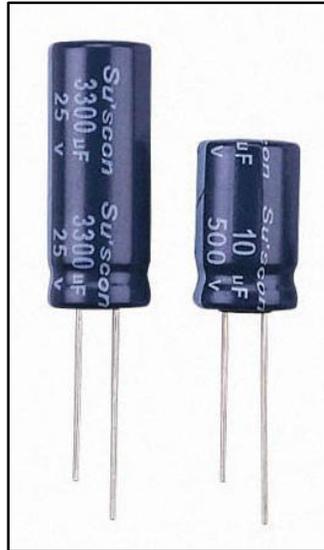


Gambar 2.18 Resistor

Menurut Jazi Eko Istiyanto : Fungsi Resistor sebagai peredam tegangan DC atau AC, Resistor berbahan dasar karbon film atau metal film, dengan besaran satuan resistans ohm berkisaran 0,1 ohm dan M-ohm (1 Mega = $1. 10$ pangkat 6 ohm).

Resistor tidak memiliki polaritas, atau disebut sebagai non-polar, resistor dapat di bagi menjadi enam tipe di bedakan berdasarkan pada kualitas dan nilai keakuratannya, antara lain Resistor tipe 5%, memiliki ketepatan nilai resistansi rill 5% dari besaran resistansinya yang tertera. Resistor tipe 10% memiliki nilai nyata tidak kurang / lebih dari 10%. Resistor tipe 20% memiliki keakuratan terendah 20%. Resistor tipe 1% memiliki keakuratas 1% dengan nilai selisih tidak kurang atau melampaui 1%. Resistor tipe 2% memiliki keakuratan 2% seperti resistor 1%. Dan terakhir adalah Variable Resistor (VR) merupakan resistor yang besar resistansinya dapat di variasikan 0 ohm hingga nilai maksimal yang tertera pada VR tersebut, untuk VR sendiri memiliki dua tipe yaitu potensiometer yang memiliki ukuran cukup besar dengan tuas pengatur dan tipe trimpot tanpa tuas pengatur.

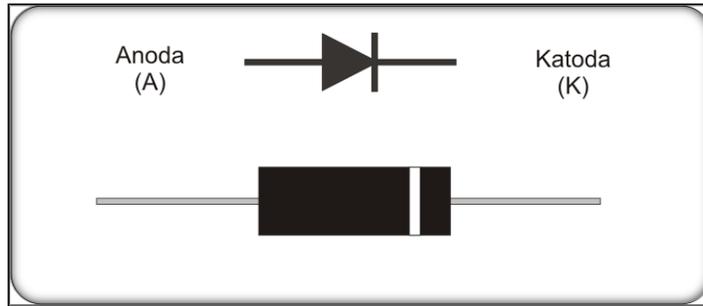
2.7.2 KAPASITOR



Gambar 2.19 Kapasitor

Menurut Jazi Eko Istiyanto : Fungsi kapasitor memiliki banyak fungsi di antaranya sebagai penguat tegangan DC untuk rangkaian catu daya atau tapis gelombang AC. Selain itu bisa di aplikasikan sebagai generator pembangkit gelombang kotak maupun sinusoidal. Kapasitor menggunakan satuan Farad (F), jangkauannya antar 1 pico-Farad (pF) atau 1×10 pangkat 12 Farad hingga 1 F. beberapa jenis kapasitor ada yang bertipe polar dan non-polar. Terdapat beberapa jenis kapasitor yang beredar pada umumnya seperti kapasitor keramik, kapasitor yang bersifat non-polar dan berbentuk seperti keramik. Kapasitor Polyester (Mylar) kapasitor dengan kategori kelas II dan bersifat non-polar. Kapasitor Elektrolit yang tersdidi dari dua lembar plat metal alumunium yang di pisahkan oleh bahan cariran dielektrik dan bersifat polar. Kapasitor Tantalum adalah kapasitor dengan elektroda yang terbuat dari material tantalum dan berpolaritas positif.

2.7.3 Dioda dan LED



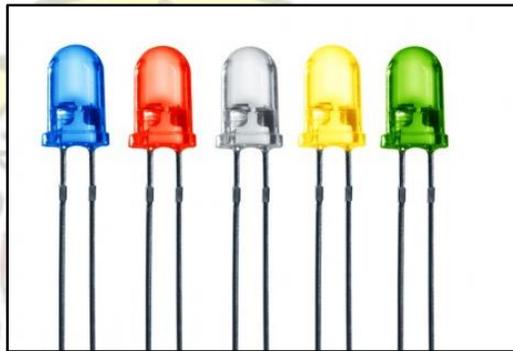
Gambar 2.20 Contoh Simbol dan Bentuk Dioda

Menurut Jazi Eko Istiyanto : Dioda adalah komponen semikonduktor yang hanya mengalirkan arus searah. Bahan dasar dari dioda terdapat germanium dan silikon yang dapat diklasifikasikan menjadi tiga jenis yaitu,

- a) Dioda Kontak Titik sebuah dioda yang diaplikasikan sebagai pengubah sinyal berfrekuensi tinggi ke sinyal yang berfrekuensi rendah, Umumnya di gunakan pada rangkaian radio penerima gelombang AM atau FM. Contoh : dioda tipe 0A 71, 0A 90 dan 1N 60.
- b) Dioda Hubungan sebuah dioda yang berfungsi sebagai penyearah tegangan, memiliki batas tegangan dan arus maksimal, jika dialiri melebihi batas akan mengakibatkan rusaknya dioda. Contoh : dioda tipe 1N4001 memiliki batas arus dan tegangan hingga 1A 50V
- c) Dioda Zenar sebuah dioda dengan berbagai tipe tegangan dan arus pembatasnya. Misalnya dioda zener 6V8 mampu membatasi tegangan luaran pada 6,8V meskipun tegangan masukan sebesar 12V atau lebih. Sebaliknya, saat tegangan masukan kurang dari 6,8V tegangan luaran akan sama dengan tegangan masukan,

Pembatasan teganan ini terwujud dari pemanfaatan karakteristik sisi eksklusif daerah tegangan *breakdown* (daerah kerja reverse bias)

- d) Light Emitting Diode (LED) adalah dioda yang dapat memancarkan cahaya dengan tegangan catu minimal 1,8V. selain itu LED berfungsi sebagai penyearah arus dan dapat digunakan berbagai rangkain elektronika, sebagai indikator lampu penerangan dengan efesiensi yang tinggi dan ketahanan jauh lebih baik dibandingkan lampu bohlam maupun neon PL.



Gambar 2.21 Bentuk Led

LED juga dapat digunakan sebagai pembatas tegangan dari tegangan catu yang tidak terlalu besar (3V sampai 5V), jika di catu melebihi 5V akan merusak LED