

BAB II

TINJAUAN PUSTAKA

2.1. Konsep Sistem Informasi

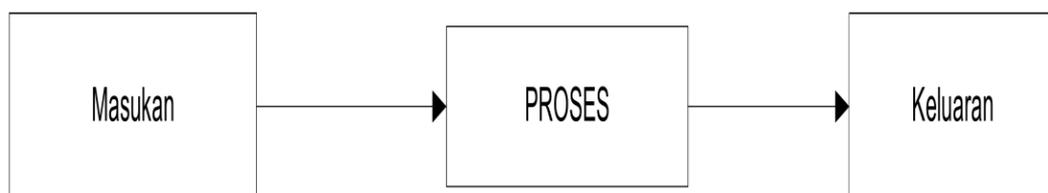
2.1.1. Sistem

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sistem ini menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata adalah suatu objek nyata, seperti tempat benda, dan orang-orang yang betul-betul ada dan terjadi.[2]

Kata "sistem" sendiri sudah banyak digunakan dalam percakapan sehari-hari, dalam forum diskusi maupun dokumen ilmiah. Kata ini digunakan untuk banyak hal dan pada banyak bidang pula, sehingga maknanya menjadi beragam. Sesuatu sistem juga bisa dibidang berdasarkan atas objek-objek, unsur-unsur, atau komponen-komponen yang berkaitan dan berhubungan satu sama lainnya sedemikian rupa sehingga unsur-unsur tersebut merupakan suatu kesatuan pengolahan yang tertentu.[2]

Jadi dari pengertian sistem di atas dapat disimpulkan bahwa sistem merupakan suatu entitas (kesatuan) yang terdiri dari elemen-elemen atau prosedur-prosedur yang saling berinteraksi yang disusun dengan syarat skema yang menyeluruh (terintegrasi) digunakan untuk melaksanakan atau menjalankan suatu kegiatan atau fungsi utama dari perusahaan yang bertujuan untuk mencapai tujuan tertentu. Sasaran sistem mempengaruhi dan sering mengendalikan konten masukan menjadi keluaran.[3]

Pada dasarnya rangkaian unsur dalam suatu sistem meliputi:



Gambar 2.1 Contoh Rangkaian unsur dalam suatu Sistem.

2.1.2. Informasi

Informasi adalah suatu data yang disampaikan baik secara langsung maupun tidak langsung, akurat, lengkap, tepat pada waktu dan sesuai kebutuhannya oleh si penerima.[2]

Atribut Informasi Yang Berkualitas, di lihat dari beberapa dimensi :

1. Dimensi Waktu :

Ketepatan waktu : Informasi harus disajikan ketika diperlukan

Kondisi keberadaan : Informasi harus terbaru (*up-to-date*) ketika disajikan

Frekwensi : Informasi harus disajikan setiap kali Informasi tersebut diperlukan.

Periode Waktu : dapat disajikan tentang masa lampau, masa kini, dan periodewaktu masa yang akan datang.

2. Dimensi Isi (*content*):

a. Ketelitian : Informasi harus bebas dari kesalahan

b. Keterkaitan : Informasi harus dihubungkan dengan kebutuhan dari penerima informasi yang spesifik dalam suatu situasi yang spesifik pula.

c. Kelengkapan : Harus bisa menyajikan semua informasi yang diperlukan.

d. Kepadatan : Hanya menyajikan informasi yang diperlukan saja
Lingkup Informasi Ruang lingkupnya luas atau sempit, fokus kepada informasi eksternal atau internal

e. Kinerja/ *Performance* : Informasi dapat mengungkapkan kinerja dengan mengukur penyelesaian aktivitas , kemajuan yang dicapai, atau sumber daya yang terkumpulkan.

3. Dimensi Bentuk (*Format*):

a. Kejelasan : Informasi harus disajikan dalam bentuk atau format yang mudah untuk difahami

b. *Detil* / rincian : Informasi dapat disajikan secara rinci atau dalam format ringkasan

c. *Order*: Informasi dapat diatur dalam suatu urutan tertentu.

- d. Presentasi : Informasi dapat ditampilkan dalam bentuk narasi, angka, gambar / grafis, atau bentuk lain
- e. Media : Informasi dapat disajikan dalam wujud catatan dokumen atau kertas yang tercetak, tayangan video, atau media lainnya.

2.1.3. Sistem Informasi

Sistem informasi adalah data yang dikumpulkan, dikelompokkan dan di olah sedemikian rupa sehingga menjadi sebuah satu kesatuan informasi yang saling terkait dan saling mendukung sehingga menjadi suatu informasi yang berharga bagi yang menerimanya.[2]

Sistem informasi dapat didefinisikan sebagai suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media prosedur-prosedur dan pengendalian yang ditunjukkan untuk mendapatkan jalur komunikasi penting, melakukan *proses* tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan.[3]

Informasi dalam suatu lingkungan sistem informasi, harus mempunyai persyaratan umum sebagai berikut :

- a. Harus sesuai dengan kebutuhan yang ada dalam *proses* pembuatan / pengambilan keputusan
- b. Harus menilai *surprise* yaitu yang sudah diketahui hendaknya jangan diberikan
- c. Harus dapat menuntun pemakai untuk membuat keputusan, Suatu keputusan tidak selalu menuntun adanya tindakan.

2.2. Bisnis

2.2.1. Mengenal Bisnis

Bisnis merupakan organisasi yang menjual barang atau jasa kepada konsumen atau bisnis lainnya, untuk mendapatkan laba sebesar-besarnya. Secara historis, bisnis berasal dari kata business yang berasal dari kata dasar busy yang berarti “sibuk”. Dalam artian, sibuk mengerjakan aktivitas dan pekerjaan yang

mendatangkan keuntungan. Dalam ekonomi kapitalis, kebanyakan bisnis dimiliki oleh pihak swasta, bisnis dibentuk untuk mendapatkan profit dan meningkatkan kemakmuran para pemiliknya.[4]

2.2.2. Pengertian Jasa dan Tipe Jasa

Jasa adalah bentuk produk yang hadir dalam bentuk nilai kepuasan konsumen, yang dapat dari aksi dan bukan dari benda fisik. Sebuah bentuk jasa dapat dilihat dari dua dimensi. Dimensi yang pertama adalah jenis dari jasa tersebut, dan dimensi yang kedua adalah cara penyampaian jasa tersebut.

Pada dimensi yang pertama, Jasa dapat dibagi menjadi beberapa tipe antara lain:

1. *Physical Service*

Jasa yang berhubungan langsung dengan benda fisik. Contoh signifikan dari Jasa Fisik adalah jasa pengangkutan barang, panti pijat, terapi kecantikan dan lainnya.

2. *Information Service*

Jasa yang berhubungan langsung dengan informasi. Contohnya pencatatan data akademis, pencatatan data medis, dan lainnya.

3. *Knowledge Service*

Jasa yang berhubungan dengan penyaluran pengetahuan. Contohnya, perkuliahan dan bimbingan belajar.

Cara penyampaian jasa antara lain:

1. *Personal*

Jasa yang disampaikan oleh pihak yang identitasnya wajib diketahui oleh konsumen. Misalnya konsultasi kesehatan dengan seorang dokter spesialis.

Konsumen harus yakin bahwa dokter yang dihadapi adalah dokter dengan spesialisasi yang dibutuhkan.

2. *Impersonal*

Kebalikan dari personal, identitas pihak penyampai jasa bukan merupakan hal yang penting. Contohnya perbaikan mobil di bengkel. Konsumen tidak perlu mengetahui spesialisasi atau kemampuan para pekerja yang melakukan servis.

3. *Automated*

Jasa bersifat *automated* bila disampaikan oleh mesin, contohnya komputer atau lainnya.

2.2.3. Persaingan Pasar

Keberhasilan bisnis salah satunya ditentukan oleh kemampuan memahami pesaing. Output dari kemampuan tersebut menopang manajemen dalam memutuskan dimana akan bersaing dan bagaimana posisi diantara pesaing. Demikian karena, persaingan merupakan sebuah usaha untuk mengidentifikasi ancaman, kesempatan atau permasalahan strategis (*strategy question*) yang terjadi sebagai akibat dari perubahan pesaing pasar secara potensial bersifat dinamis.[15]

Maraknya pertumbuhan usaha saat ini, ternyata berhasil menciptakan persaingan pasar yang semakin ketat, keadaan ini tentu saja memaksa para pengusaha untuk menggunakan berbagai cara dalam strategi pemasaran mereka. Banyak cara yang mereka lakukan agar usahanya tidak kalah bersaing dengan peluang usaha lainnya, sehingga masih bisa bertahan bahkan berkembang ditengah persaingan pasar yang semakin kompetitif. [15]

2.3. Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan *platform* yang terbuka bagi para pengembang untuk menciptakan aplikasinya. Android Inc. merupakan pendatang baru dalam pembuatan piranti lunak untuk ponsel atau *smartphone* yang dibeli oleh Google Inc. dan untuk mengembangkan *Android*, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan piranti keras, piranti lunak dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile dan Nvidia [7]

2.3.1. Sejarah Android

Pada tahun 2005 Google mengakuisi *Android* Inc yang pada saat itu dimotori oleh Rubin, Rich Miner, Nick Sears, dan Chris White. Yang kemudian pada tahun itu juga memulai membangun platform *Android* secara intensif. Kemudian pada tanggal 12 november 2007 Google bersama open Handset Alliance yaitu sebuah konsersium perangkat *mobile* terbuka, merilis Google *Android* SDK, setelah mengumumkan seminggu sebelumnya. Dan sambutannya sangat luar biasa, hampir semua media

berita tentang IT dan Programming memberitakan tentang di lirisnya *Android SDK (Software Development Kit)* ini.[7]

Ini dikarenakan IDE dari platform *Android* ini sangat menarik untuk developer dan programmer di seluruh penjuru dunia. Di sisi lain, produsen pembuat handset ponsel juga berlomba-lomba membuat handset dengan platform *Android* ini.

Google bersama dengan OHA merilis paket *software SDK* yang lengkap untuk mengembangkan aplikasi pada perangkat *mobile*. Yaitu : *sistem operasi, middleware* dan aplikasi Utaman untuk perangkat *mobile*. Sebagai programmer atau *developer* bisa melakukan segalanya, mulai dari membuat aplikasi pengiriman SMS hanya dengan dua baris kode, hingga mengganti event pada *Home Screen* perangkat *Android*. Selain itu, bahkan dengan mudah kita bisa membuat dan merubah sistem operasi-nya atau mengganti semua aplikasi default dari google.[7]

2.3.2. Versi Sistem Operasi Android

Berikut ini adalah beberapa dari versi versi Sistem Operasi *Android* :

1. Versi 4.0.3 – 4.0.4 (*Ice Cream Sandwich*) Perubahan akan mencakup :
 - a. Peningkatan fungsionalitas grafis, berbasis data, pemeriksa ejaan dan bluetooth
 - b. Peningkatan fitur kalender
 - c. Fitur aplikasi kamera yang meningkatkan stabilisasi video dan resolusi QVGA
 - d. Perbaikan aksesibilitas seperti penigkatan akses konten bagi pembaca layar.
 - e. Rotasi layar yang lebih halus
 - f. Dukungan fitur identifikasi nomor telepon.



Gambar 2.2 Contoh Ilustrasi Logo Android Versi 4.0.3- 4.0.4 *Ice Sandwich*

2. Versi 4.1 (Jelly Bean) Perubahan akan mencakup :
 - a. Kemampuan untuk menambahkan widget aplikasi tanpa akses root.
 - b. Audio USB (sebagai suara eksternal DACs)
 - c. Audio chaining
 - d. Transfer data Bluetooth bagi *Android* Bean
 - e. Perluasan notifikasi
 - f. Teks dua bahasa dan dukungan bahasa lainnya
 - g. Triple buffering pada grafis
 - h. Antar muka yang lebih halus
 - i. Waktu vsync pada animasi UI dikelola oleh kerangka kerja *Android*, termasuk reaksi aplikasi, efek sentuh, komposisi layar dan penyegaran tampilan.
 - j. Papan ketik yang bias dimodifikasi oleh pengguna



Gambar 2.3 Contoh *Ilustrasi Logo Android Versi 4.1 Jelly Bean*

3. Versi 4. (KitKat) perubahan akan mencakup :
 - a. Pembaruan antar muka dengan bar status dan navigasi transparan pada layar depan.
 - b. Kinerja perangkat dengan spesifikasi yang lebih rendah.
 - c. NFC Host Card Emulation sebagai emulator kartu pintar.
 - d. Kerangka kerja baru untuk transisi *UI*
 - e. Inframerah
 - f. Peningkatan aksestabilitas API.
 - g. Mesin virtual eksperimental baru, ART

h. Dukungan Bluetooth Message Access profil (MAP)



Gambar 2.4 Contoh *Ilustrasi Logo Android Versi 4 KitKat*

2.3.3. Tools

2.3.3.1. Android SDK

Android SDK merupakan *Software Development Kit* yang memungkinkan pengembangan untuk membuat aplikasi untuk *platform Android*. *SDK Android* mencakup proyek sampel dengan source code, alat-alat pembangunan, sebuah emulator, dan perpustakaan yang dibutuhkan untuk membangun aplikasi *Android*. Aplikasi yang ditulis menggunakan bahasa pemrograman Java dan berjalan di Dalvik. Mesin virtual yang dirancang khusus untuk penggunaan embedded yang berjalan di atas kernel Linux.[14]



Gambar 2.5 Contoh *Android SDK*

2.3.3.2. Android Studio

Android studio adalah sebuah lingkungan pengembangan terpadu (IDE) untuk mengembangkan untuk platform *Android*. Hal itu di sampaikan pada tanggal 16 Mei, 2013 pada Google I / O conference oleh Product Manager Google, Katherine Chou. Android Studio tersedia secara bebas di bawah Lisensi Apache 2.0.

Android Studio pada awal tahap preview akses mulai dari versi 0.1 Mei 2013, kemudian memasuki tahap beta mulai dari versi 0.8 yang dirilis pada bulan 2014. Yang pertama membangun stabil dirilis pada bulan Desember 2014, mulai dari versi 1.0.

Android Studio dirancang khusus untuk pengembangan *Android*. Ini tersedia untuk di-download pada Windows, Mac OS X dan Linux, dan di ganti Eclipse Pengembangan Android Tools (ADT) sebagai IDE utama Google untuk pengembangan aplikasi *Android*.



Gambar 2.6 Contoh Android Studio

2.4. Pengembangan Aplikasi Mobile

Perkembangan perangkat *mobile* tumbuh pesat dalam beberapa tahun belakangan ini, khususnya pada perangkat yang memiliki sistem operasi *Android*. Aplikasi *Android* dapat di kembangkan melalui 3 pendekatan yang masing-masing menghasilkan aplikasi *native*, aplikasi *web* dan aplikasi *hybrid*. Masing-masing pendekatan ini mempunyai kelebihan dan kekurangan serta cara pengembangan yang berbeda satu sama lain. Perbedaan antara aplikasi *native*, aplikasi *web* dan aplikasi *hybrid* digambarkan.[8]



Gambar 2.7 Contoh Perbedaan Struktur Aplikasi

2.4.1. Aplikasi Web

Pada awalnya aplikasi web dibangun dengan hanya menggunakan bahasa yang disebut HTML (Hyper Text Markup Language). Pada perkembangan berikutnya, sejumlah skrip dan obyek yang dikembangkan untuk memperluas kemampuan pengembangan HTML seperti PHP dan ASP pada skrip dan Applet pada obyek. [8] Aplikasi Web dapat dibagi menjadi dua jenis yaitu aplikasi web statis dan dinamis.

Web statis dibentuk dengan menggunakan HTML. Kekurangan aplikasi seperti ini terletak pada keharusan untuk memelihara program secara terus menerus yang mengikuti setiap perkembangan yang terjadi. Kelemahan ini dibatasi oleh model aplikasi web dinamis.

2.4.2. Aplikasi Native

Aplikasi *native* adalah aplikasi yang secara khusus ditujukan untuk *platform mobile* tertentu dan menggunakan bahasa pemrograman serta perangkat pengembangan sesuai dengan *platform* tersebut. Aplikasi *Android* ditulis menggunakan bahasa pemrograman Java dengan menggunakan Eclipse IDE, sementara iOS (iphone) ditulis dengan menggunakan bahasa Objective-C dan Xcode.

Aplikasi *native* menghasilkan tampilan antarmuka yang alami dan cocok dengan *platform* yang dikembangkan. Performa dari aplikasi *native* sangat baik karena ditulis menggunakan bahasa pemrograman yang sesuai dengan Sistem operasi yang dituju. Kelebihan lain dari aplikasi *native* adalah adanya dukungan akses ke semua fitur perangkat keras *smartphone* seperti *info device*, kamera, *compass*, *file* dan GPS. Pengembangan aplikasi *native* hanya dapat berjalan pada *platform* yang



sudah dispesifikasikan pada tahap awal pengembangan aplikasi, sehingga terbatas hanya untuk satu sistem operasi.

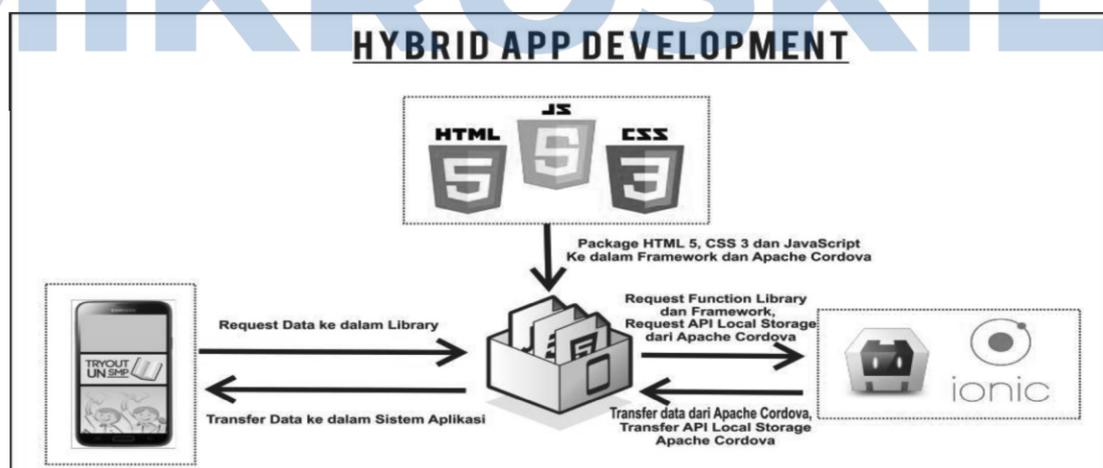
2.4.3. Aplikasi Hybrid

Intuisi dari aplikasi *hybrid* adalah menanamkan aplikasi *mobile* HTML5 ke dalam konten aplikasi *native*. Aplikasi *hybrid* merupakan kombinasi dari kelebihan yang dimiliki antara aplikasi *web* dan aplikasi *native*. Aplikasi *hybrid* mengkonversi aplikasi *web mobile* HTML5 ke aplikasi *native smartphone* yang ditargetkan. Konversi aplikasi *web mobile* ke dalam aplikasi *native* memerlukan dukungan perangkat lunak spesifik semacam *framework* pengembangan aplikasi *mobile*.

Kelebihan aplikasi *hybrid* adalah tahap pengembangannya yang relative mudah karena memanfaatkan standar teknologi *web*. Aplikasi *hybrid* dapat berjalan pada hampir semua *platform mobile* yang dituju, artinya aplikasi *hybrid* dapat didistribusikan secara aplikasi *native* menuju ke *platform mobile* yang ditargetkan oleh pengembang. Distribusi aplikasi yang secara *native* memungkinkan aplikasi *hybrid* untuk dapat mengakses fitur-fitur perangkat keras seperti kamera, *wifi*, kalender dan *file*.

2.5. Pengembangan Aplikasi Hybrid Android

Aplikasi *hybrid* adalah gabungan antara aplikasi *native* dan aplikasi *web*. Pengembangan aplikasi *hybrid* menggunakan HTML5, JavaScript, CSS3 dan Ionic Framework yang di dalamnya terdapat *compiler* Apache Cordova atau PhoneGap.



Gambar 2.8 Contoh *Tools* yang Dipakai untuk Pengembangan Aplikasi *Hybrid*

2.5.1. HTML 5

Hybrid Mark up Language atau biasa dikenal dengan HTML merupakan suatu metode untuk mengimplementasikan konsep *hypertext* dalam suatu naskah atau dokumen. HTML bukan tergolong pada suatu bahasa pemrograman karena sifatnya yang hanya memberikan tanda (*marking up*) pada suatu naskah teks dan bukan sebagai program. HTML [10] merupakan protokol yang digunakan untuk mentransfer data atau dokumen dari *web server* ke dalam *browser*. [10]

HTML memiliki banyak versi dan versi terbaru saat ini adalah HTML5. HTML5 adalah versi terbaru teknologi *hypertext* atau *web* yang sekarang ini masih dalam tahap pengembangan. Munculnya versi terbaru dari HTML yakni HTML versi ke-5, membawa beberapa elemen-elemen baru untuk mewujudkan struktur halaman *web* yang lebih baik. Elemen HTML sendiri merupakan semua hal yang terdapat diantara *start tag* sampai *end tag*. Beberapa elemen-elemen baru. [10]

Tabel 2.1 Elemen-Elemen HTML

Tag	Keterangan
<article>	Digunakan untuk menspesifikasikan konten yang bersifat independen, seperti artikel, <i>blog post</i> , <i>forum post</i> dan
<aside>	Digunakan untuk sebuah subkonten. Biasanya digunakan di dalam tag <article>
<bdi>	Digunakan untuk teks yang tidak boleh terikat pada arah teks-elemen induknya
<command>	Sebuah <i>button</i> atau <i>radiobutton</i> atau <i>checkbox</i> .
<details>	Digunakan untuk menjelaskan detail tentang sebuah dokumen atau sebagian dari dokumen.
<summary>	Digunakan pada sebuah ringkasan dan sejenisnya di dalam tag <details>
<figure>	Digunakan untuk mengelompokkan sekumpulan <i>section</i> , biasanya berupa video.
<figcaption>	Berisi <i>caption</i> atau keterangan yang ditempatkan di dalam tag <figure>

<footer>	Digunakan sebagai <i>footer</i> dari sebuah halaman.
<header>	Digunakan sebagai <i>header</i> dari sebuah halaman.
<hgroup>	Digunakan untuk sekumpulan <i>heading</i>
<mark>	Digunakan pada teks yang di <i>highlight</i> .
<meter>	Digunakan untuk pengukuran, yang nilai maksimal dan minimalnya telah ditentukan.
<nav>	Digunakan untuk sekumpulan navigasi.
<progress>	Digunakan untuk membuat <i>progress bar</i> .
<ruby>	Digunakan untuk anotasi ruby.
<rt>	Menjelaskan anotasi ruby.
<rp>	Menunjukkan elemen jika <i>browser</i> tidak mendukung ruby.
<section>	Menunjukkan sebuah <i>section</i> di dalam halaman seperti bab, <i>footer</i> dan sebagainya.
<time>	Digunakan untuk mendefinisikan waktu dan tanggal.
<wbr>	<i>Word Break</i> digunakan untuk memisahkan suatu kata bila diperlukan.

Versi terbaru dari HTML memiliki beberapa fitur baru yang belum ada pada versi-versi sebelumnya. Fitur-fitur baru yang dimiliki HTML5 [10] adalah:

1. **Canvas**

Fitur yang satu ini memungkinkan *developer* untuk memasukkan objek 2D atau 3D ke dalam halaman *web*. *Canvas* dapat membuat *render* grafik, diagram, gambar dan animasi secara dinamis.

2. **Doctype**

HTML lama biasanya memiliki *doctype* yang panjang, sedangkan HTML5 memiliki *doctype* yang lebih singkat.

3. **HTML5 Audio**

Salah satu fitur baru HTML5 lainnya adalah disediakannya *tag* baru yang memungkinkan untuk memutar audio tanpa menggunakan *plugin* tambahan seperti Flash Player.

4. **HTML5 Video**

Fitur multimedia lainnya yang didukung oleh HTML5 adalah video, jika pada HTML selama ini harus menggunakan Flash Player untuk memasukkan konten video, maka pada HTML5 tidak perlu lagi menggunakannya.

5. Geolocation

Sebuah API (*Application Programming Interface*) yang menarik, yang memungkinkan pengguna untuk berbagi lokasi dengan aplikasi *web*.

6. Media Penyimpanan (Storage)

Halaman *web* dapat menyimpan data secara lokal dalam *browser* pengguna yang sebelumnya hal ini dilakukan dengan *cookie*, namun berimbas pada penyimpanan *web* yang lebih aman dan lebih cepat. Data ini tidak disertakan dengan setiap permintaan *server*, tetapi digunakan hanya ketika meminta. Hal ini juga memungkinkan untuk menyimpan data dalam jumlah besar, tanpa mempengaruhi kinerja *website*. Data disimpan dalam kunci atau nilai pasangan dan halaman *web* hanya dapat mengakses data yang disimpan dengan sendirinya. HTML5 mendukung beberapa jenis media penyimpanan baru yaitu *Web Storage (name or value pairs)*, *Indexed DB* dan *Web SQL Database*. Keuntungan dari adanya berbagai macam media penyimpanan ini adalah peningkatan kecepatan dari aplikasi.

7. WebSocket API

Fitur komunikasi yang paling *powerfull* dispesifikasi HTML5 adalah *WebSocket*, yang mendefinisikan saluran komunikasi *full-duplex* yang beroperasi melalui soket tunggal dalam *web*. *WebSocket* tidak hanya peningkatan tambahan untuk konvensional komunikasi HTTP, melainkan

8. Semantic atau Struktur Element yang Baru

Web menjadi lebih dinamis dan menarik akibat fitur ini. Fitur ini menjadikan *website* yang menggunakan teknologi HTML5 lebih mudah dikenali oleh *Google Search Engine* sehingga *web* sering muncul pada mesin pencari *Google* dan lebih sering diakses serta cepat diketahui oleh orang lain.

2.5.2. CSS 3

Cascading Style Sheet (CSS) merupakan salah satu bahasa pemrograman *web* untuk mengatur beberapa komponen atau elemen dalam sebuah *web* sehingga lebih

terstruktur dan seragam dengan HTML atau *script*. CSS memiliki 3 versi, yaitu CSS1, CSS2 dan CSS3. [10]

CSS3 melakukan penataan terhadap komponen HTML maupun XHTML pada halaman *web* sehingga menghasilkan tampilan yang ramah dimata atau *retina friendly*. CSS pada mulanya dipelopori dan dikembangkan serta distandarisasi oleh *World Wide Web Consortium* atau W3C pada tahun 1996.

CSS versi ketiga melakukan berbagai pemutakhiran terhadap kemampuannya beradaptasi dengan teknologi multimedia yang semakin modern, diantara pemutakhiran tersebut adalah :

1. CSS3 mampu membuat efek animasi secara independen tanpa membutuhkan *software* tambahan seperti Adobe Flash dan Microsoft Silverlight.
2. CSS3 mampu memberikan efek grafis terhadap teks, kolom, tabel serta penataan huruf.
3. CSS3 mampu menampilkan berbagai macam dan jenis huruf selain *web-safe fonts*.
4. CSS3 mampu merekonstruksi secara visual tampilan *block*, seperti merubah ukuran kotak, transformasi 2D atau 3D, memberikan efek sudut dan bayangan. CSS3 dapat memanipulasi secara visual warna, desain maupun tekstur.

2.5.3. JavaScript

Perkembangan JavaScript diperkenalkan pertama kali oleh Netscape pada tahun 1995. JavaScript merupakan bahasa yang berbentuk kumpulan *script* yang berfungsi untuk memberikan tampilan yang tampak lebih interaktif pada dokumen *web*. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan ke dalam bahasa pemrograman HTML dengan mengijinkan pengekseskuan perintah-perintah pada sisi *client* dan bukan pada sisi *server* dokumen *web*. [10]

Bahasa pemrograman JavaScript berisi skrip yang pemasangannya terselip di sebuah dokumen HTML, sehingga bahasa JavaScript tidaklah memerlukan sebuah kompilator atau penerjemah khusus. JavaScript dapat membuat sebuah halaman *web* menjadi lebih dinamis dan interaktif terhadap *user* karena halaman *web* mampu

berfungsi sebagai sebuah program aplikasi yang dapat melakukan *proses* masukan yang diberikan *user* dan memberikan hasil sesuai dengan yang telah diprogramkan. JavaScript membuat beban server menjadi lebih ringan dan halaman *web* akan jauh lebih cepat merespon, bahkan pada koneksi *Internet* yang paling rendah sekalipun.

JavaScript termasuk kategori bahasa yang *case sensitive* artinya sangat membedakan penamaan variabel dan fungsi dalam penggunaan antara huruf besar dan huruf kecil. Kelebihan dari penggunaan JavaScript adalah sebagai berikut :

- Lebih praktis dan mudah sebab bahasa pemrograman JavaScript memiliki sedikit sintaks.
- Koneksi lebih cepat sebab peletakan program terdapat di sisi *client*, ukuran *file* sangat kecil dan dapat langsung dijalankan di *browser*.

Selain memiliki beberapa kelebihan, JavaScript juga memiliki beberapa kekurangan antara :

1. Pengelolaan objek dalam JavaScript sangat terbatas.
2. Penggunaan skrip dapat di-*copy* langsung melalui *web browser*, sehingga setiap orang dapat menggunakan program JavaScript yang telah dibuat.

2.5.4. Ionic Framework

Ionic adalah kerangka ponsel HTML5 dengan fokus pada kinerja yang memanfaatkan akselerasi *hardware* dan tidak memerlukan pihak ketiga seperti JS library. Ionic bekerja bersama-sama dengan Angular.js untuk membangun sebuah aplikasi interaktif.

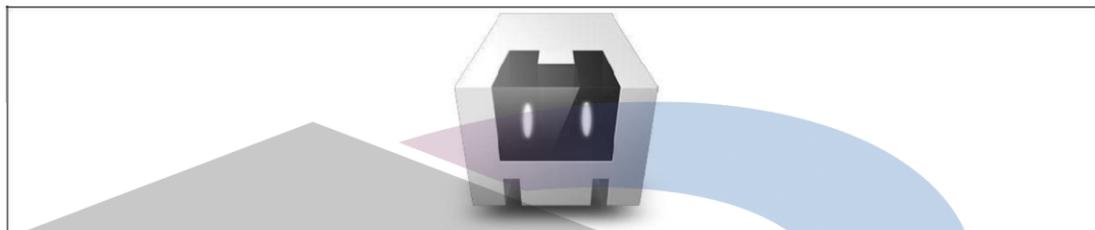


Gambar 2.9 Contoh Logo Ionic

2.5.4.1. Apache Cordova atau PhoneGap

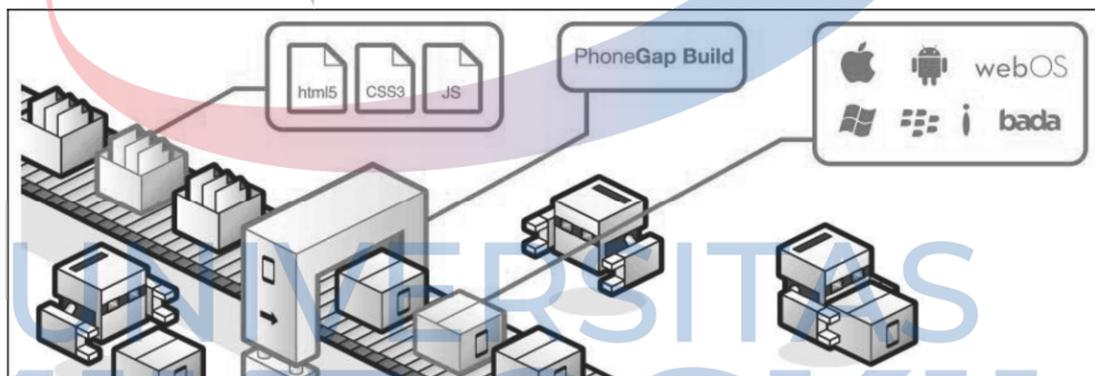
Apache Cordova adalah *framework* untuk membuat aplikasi pada berbagai macam *platform* seperti Android, Blackberry, Iphone atau Windows Phone

menggunakan HTML5, JQuery, JQuery Mobile dan CSS3. Membangun aplikasi untuk *device* yang berbeda seperti Android, Iphone, Windows Mobile dan lainnya dibutuhkan *framework* dan bahasa pemrograman yang berbeda, seperti pada *Android* menggunakan bahasa pemrograman Java, blackberry dengan Java, Iphone dengan Basic C dan Windows Phone dengan C#. [11]



Gambar 2.10 Contoh Logo Apache Cordova

Framework dari PhoneGap digunakan oleh beberapa *platform* aplikasi *mobile* seperti Worklight dan appMobi sebagai tulang punggung mesin pengembangan *mobile* klien nya. Adobe resmi mengumumkan akuisisi *software*.



Gambar 2.11 Contoh Pengembangan Aplikasi Mobile Menggunakan PhoneGap

2.5.4.2. Angular JS

AngularJS adalah sebuah *framework* MVC *full frontend* untuk aplikasi *web* JavaScript. AngularJS dibangun oleh Google dan menyediakan sebuah metode cepat untuk membangun aplikasi *web* laman tunggal. Penggunaan AngularJS dimasukkan ke dalam sebuah laman *web* dengan menggunakan *tag* `<script>` dan ditulis dalam JavaScript. AngularJS dimaksudkan sebagai sebuah *framework* untuk membangun sebuah aplikasi *web* utuh. AngularJS mengandung sebuah versi minimal jQuery secara *default*.

2.6. Web Service

2.6.1. Definisi

Web Service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. *Web service* diartikan sebagai sebuah antar muka (*interface*) yang menggambarkan sekumpulan operasi-operasi yang dapat diakses melalui jaringan, misalnya *Internet* dalam bentuk pesan *XML (eXtensible Markup Language)*. [16]

Menurut Michael C. Daconta (2005), *Web Service* adalah aplikasi perangkat lunak yang dapat ditemukan, diuraikan, dan diakses berdasarkan pada *XML* dan protocol *standard Web* pada *intranet, extranet, dan Internet*.

Web service menyediakan standar komunikasi diantara berbagai aplikasi *software* yang berbeda-beda, dan dapat berjalan diberbagai *platform* maupun *framework*. *Web service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu web untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*service*) yang disediakan oleh suatu sistem yang menyediakan *web service*. [16]

2.6.2. Komponen Web Service

Komponen – komponen Web Service yaitu :

a. *Extensible Markup Language (XML)*

XML merupakan dasar yang penting atas terbentuknya *Web services*. *Web services* dapat berkomunikasi dengan aplikasi-aplikasi yang memanggilnya dengan menggunakan *XML*, karena *XML* berbentuk teks sehingga mudah untuk ditransportasikan menggunakan protokol HTTP. Selain itu, *XML* juga bersifat *platform* independen sehingga informasi didalamnya bisa dibaca oleh aplikasi apapun pada *platform* apapun selama aplikasi tersebut menerjemahkan tag-tag *XML*.

b. *Simple Object Access Protocol (SOAP)*

XML saja tidak cukup agar *Web services* dapat berkomunikasi dengan aplikasi yang lainnya. *XML* yang digunakan untuk saling bertukar informasi antara *web services* dengan aplikasi yang lainnya harus menggunakan sebuah format standard yang dapat dimengerti oleh keduanya. Format itulah yang dikenal dengan nama

SOAP. SOAP (*Simple Object Access Protocol*) merupakan suatu format standard dokumen berbentuk *XML* yang digunakan untuk melakukan *proses* request dan responses antara *web services* dengan aplikasi yang memanggilnya. Dokumen SOAP digunakan untuk melakukan request disebut dengan SOAP *request* sedangkan dokumen SOAP yang diperoleh dari *Web services* disebut dengan SOAP *responses*.

c. *Web service Definition Language* (WSDL)

Sebelum mengakses sebuah *Web services* pastinya perlu mengetahui *method-method* apa saja yang disediakan oleh *Web services* tersebut, untuk mengetahuinya memerlukan sebuah dokumen yang bernama WSDL. WSDL (*Web services Description Language*) adalah sebuah dokumen dalam format *XML* yang isinya menjelaskan informasi detail sebuah *Web services*.

2.6.3. REST (*Representational State Transfer*)

REST (*Representational State Transfer*) merupakan jenis arsitektur yang terdapat pada *web* untuk melayani suatu *service*. REST merepresentasikan cara interaksi antara *server* dan *client* untuk melakukan *proses* pertukaran informasi melalui media yang sama.[16]

REST dapat digunakan sebagai *interface* dari API untuk mengakses suatu *resource*. API yang mengikuti prinsip dari REST *architecture* memberikan kemudahan bagi *developer* untuk tidak perlu mengetahui bagaimana struktur dari API di dalam *server*. Dalam hal ini, *server* akan memberikan informasi bagaimana agar *client* dapat mengakses *service* melalui API yang telah disediakan.

Pengguna protokol HTTP pada REST *architecture* untuk komunikasi antara *client* dan *server* terletak pada HTTP *method*, yaitu GET, POST, PUT, dan DELETE. *Method* ini dapat digunakan untuk mengakses *resources* yang ada pada *server*, bergantung dari instruksi yang diberikan oleh *server*. Dengan menggunakan protokol HTTP, URI dapat dijadikan sebagai media yang digunakan untuk mengakses *resource* dari *server*. Hal ini disebut dengan URI *tunneling*[16].

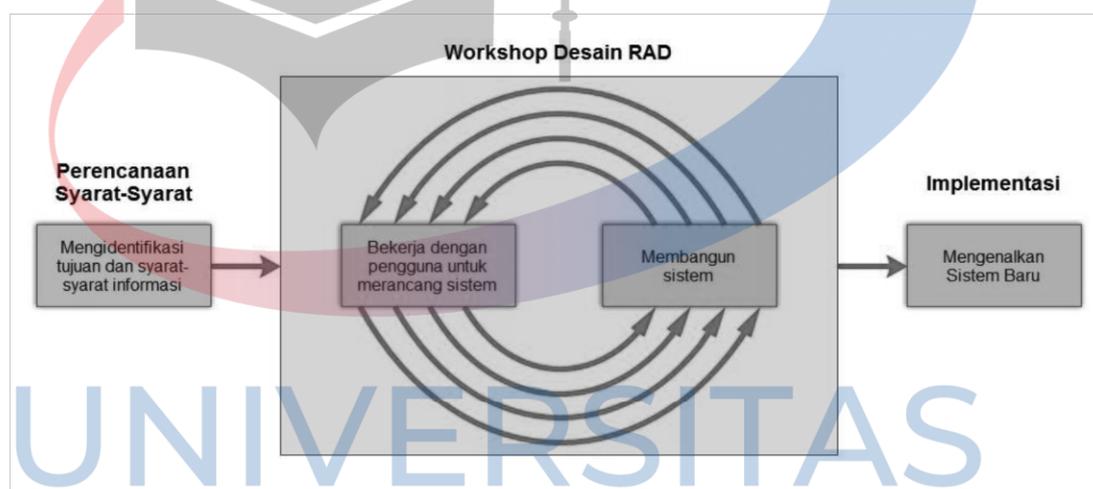
URL *tunneling* mempergunakan URI untuk mentransfer informasi pada antar sistem yang dalam jaringan dengan melakukan *encode* pada URI itu sendiri. Dengan mengirim HTTP *method* yang telah disebutkan sebelumnya, *server* dapat melakukan

eksekusi terhadap suatu program yang menghasilkan atau mengambil suatu *resource* dan mengirimkannya kembali ke *client*.

2.7. Pengembangan Sistem Informasi

2.7.1. RAD (*Rapid Application Development*) Menurut Kendall

RAD (*Rapid Application Development*) adalah suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat-perangkat lunak. RAD bertujuan mempersingkat waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem tradisional antara perancangan dan penerapan suatu sistem informasi. Pada akhirnya, RAD sama-sama berusaha memenuhi syarat-syarat bisnis yang berubah secara cepat.[12]

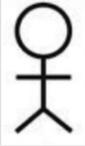


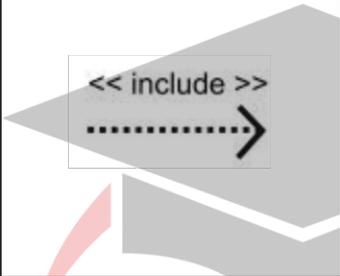
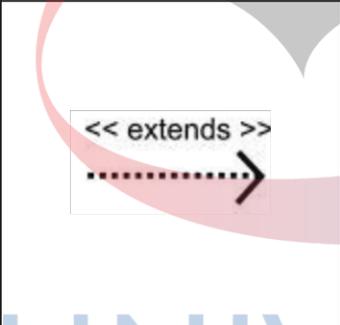
Gambar 2.12 Contoh RAD Menurut Kendall

2.7.2. Use Case Diagram

Use Case Diagram adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case diagram* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem untuk mencapai tujuan tertentu. Setiap skenario menjelaskan urutan kejadian. *Use case diagram* adalah serangkaian skenario yang digabungkan bersama-sama oleh tujuan umum pengguna. *Use case diagram* biasanya menggunakan *actors*. *Actor* adalah sebuah peran yang dapat dimainkan oleh pengguna dalam interaksinya dengan sistem.[13].

Tabel 2.2 Simbol Use Case Diagram

Simbol	Nama	Keterangan
	Aktor	Seorang aktor yang behaved classifier yang menentukan peran yang dimainkan oleh entitas eksternal yang berinteraksi dengan subjek (misalnya, dengan bertukar sinyal dan data), pengguna manusia dari sistem yang dirancang, beberapa sistem lain atau perangkat keras menggunakan layanan dari subjek.
	Use Case	Penggunaan kasus yang memungkinkan untuk menangkap kebutuhan sistem di bawah desain atau pertimbangan, menjelaskan fungsi yang disediakan oleh sistem-sistem, dan menentukan persyaratan sistem.
	Relasi Asosiasi	Setiap kasus relasi merupakan unit fungsi yang berguna yang menyediakan keterhubungan untuk aktor. Hubungan antara aktor dan use case menunjukkan bahwa aktor dan kasus

		<p>penggunaan entah bagaimana berinteraksi atau berkomunikasi satu sama lain .</p> <p>.</p>
	Include Relationship	<p>Relasi cakupan memungkinkan sesuatu <i>use case</i> untuk menggunakan fungsionalitas yang disediakan oleh para <i>use case</i> yang lainnya.</p>
	Extends Relationship	<p>Memungkinkan suatu <i>use case</i> memiliki kemungkinan untuk memperluas fungsionalitas yang disediakan <i>use case</i> yang lainnya.</p>

UNIVERSITAS
MIKROSKIL