BAB 2
TINJAUAN PUSTAKA

2.1 Tinjauan Tempat Penelitian

2.1.1 Sejarah Perusahaan

Museum Geologi pertama kali diresmikan pada tanggal 16 Mei 1929 Masehi, awalnya museum ini berfungsi sebagai tempat penyimpanan koleksi hasil penyelidikan geologi yang dilakukan oleh para ahli yang dikomandoi oleh pemerintah Belanda pada zaman dahulu.

Setelah Indonesia merdeka pada Tahun 1945, pengelolaan Museum Geologi beradadibawah Pusat Djawatan Tambang dan Geologi (PDTG/1945-1950). Pada tanggal 19 September 1945, pasukan sekutu pimpinan Amerika Serikat dan Inggris yang diboncengi oleh Netherlands Indiës Civil Administration (NICA) tiba di Indonesia (mendarat di Tanjungpriuk, Jakarta). Di Bandung mereka berusaha menguasai kembali kantor PDTG yang sudah dikuasai oleh para pegawai Indonesia. Tekanan yang dilancarkan oleh pasukan Belanda memaksa kantor PDTG dipindahkan ke Jl. Braga No. 3 dan No. 8 Bandung pada tanggal 12 Desember 1945. Kepindahan kantor PDTG rupanya terdorong pula oleh gugurnya seorang pengemudi bernama Sakiman dalam rangka berjuang mempertahankan kantor PDTG . Pada waktu itu, Tentara Republik Indonesia Divisi III Siliwangi mendirikan Bagian Tambang, yang tenaganya diambil dari PDTG. Setelah kantor di Rembrandt Straat ditinggalkan oleh pegawai PDTG, pasukan Belanda pun di tempat itu mendirikan lagi kantor yang bernama Geologische Dienst. Di mana-mana terjadi pertempuran, maka sejak Desember 1945 sampai dengan Desember 1949, selama 4 tahun kantor PDTG terlunta-lunta pindah dari satu tempat ke tempat lain. Pemerintah Indonesia berusaha menyelamatkan dokumen- dokumen hasil penelitian geologi sehingga harus berpindah pindah tempat dari Bandung – Tasikmalaya- Solo – Magelang - Yogyakarta, baru pada Th 1950 kembali ke Bandung. Dalam usaha menyelamatkan dokumen dokumen tersebut, pada tanggal 7 mei 1949, Kepala PUSAT JAWATAN TAMBANG DAN GEOLOGI, Arie

Frederik Lasut, diculik dan dibunuh tentara belanda dan gugur sebagai kusuma bangsa di Desa Pakem Yogyakarta. Sekembalinya ke Bandung, Museum Geologi mulai mendapat perhatian dari pemerintah RI, terbukti pada tahun 1960 Museum Geologi dikunjung oleh Presiden pertama RI , Ir. Soekarno. Pengelolaan Museum Geologi yang tadinya dibawah PUSAT DJAWATAN TAMBANG DAN GEOLOGI (PDTG) berganti nama menjadi: Djawatan Pertambangan Republik Indonesia (1950-1952), Djawatan Geologi (1952- 1956), Pusat Djawatan Geologi (1956-1957), Djawatan Geologi (1957-1963), Direktorat Geologi (1963-1978), Pusat Penelitian dan Pengembangan Geologi (1978 - 2005) , Pusat Survei Geologi mulai akhir tahun 2005 sampai sekarang. Seiring dengan perkembangan jaman, pada tahun 1999 Museum Geologi mendapat bantuan dari Pemerintah Jepang senilai 754,5 juta yen untuk direnovasi. Setelah ditutup selama satu tahun, Museum Geologi dibuka kembali dan pembukaannya diresmikan pada tanggal 20 Agustus Tahun 2000 oleh Wakil Presiden RI waktu itu Ibu Megawati Soekarnoputri yang didampingi oleh Menteri Pertambangan dan Energi Bapak Susilo Bambang Yudhoyono.

2.1.2 Logo Perusahaan

Berikut adalah foto bagian depan gedung Museum Geologi Bandung :



Gambar 2.1 Museum Geologi

Berikut adalah logo dari Museum Geologi Bandung :



Gambar 2.2 Logo Perusahaan

2.1.3 Visi dan Misi

Adapun visi dan misi dari CV. Mitra Jaya adalah :

1. Visi

Terwujudnya sumber informasi geologi (dokumentasi koleksi warisan geologi Indonesia) yang profesional untuk masyarakat.

2. Misi
    a. Memperagakan & mengkomunikasikan koleksi museum
    b. Menyediakan informasi & materi edukasi geologi
    c. Mendokumentasikan & mengkonservasi koleksi museum
    d. Melakukan penelitian koleksi & pengembangan museum
    e. Melakukan pameran museum & geologi
    f. Melakukan penyuluhan & sosialisasi geologi
    g. Melakukan kerjasama dengan instansi & sekolah
    h. Melakukan pengelolaan museum secara professional
    i. Memberikan pelayanan jasa permuseuman

2.1.4 Struktur Organisasi Perusahaan

Berdasarkan keputusan Menteri Energi dan Sumber Daya Mineral No. 1725 tahun 2002 tentang organisasi dan tata kerja Museum Geologi serta berdasarkan peraturan Menteri Energi dan Sumber Daya Mineral No.0030 tahun 2005 tentang organisasi dan tata kerja Departemen Energi dan Sumber Daya Mineral, susunan organisasi Museum Geologi memili tugas, yaitu :



Gambar 2.3 Struktur Organisasi

Dan berikut ini merupakan tugas dan wewenang dari masing-masing bagian pada struktur organisasi perusahaan diatas :

a. Subbagian Tata Usaha

Melaksanakan, menyiapkan bahan penyusunan program dan laporan, urusan ketatausahaan, kepegawaian, keuangan serta rumah tangga Museum Geologi.

b. Seksi Dokumentasi

Melaksanakan penyiapan bahan dan evaluasi rencana, program pengelolaan, pengembangan dokumentasi dan publikasi koleksi, kerjasama serta pelayanan jasa permuseuman.

c. Seksi Peragaan

Melaksanakan penyiapan bahan dan evaluasi rencana, program pengelolaan, pengembangan dokumentasi dan publikasi koleksi, kerjasama serta pelayanan jasa permuseuman.

d. Kelompok Fungsional

Terdiri dari peneliti, penyelidik bumi, dan jabatan fungsional lainnya memiliki tugas melaksanakan penyelidikan, perekayasaan, penelitian dan pengembangan koleksi museum geologi.

2.2. Landasan Teori

Landasan teori ini akan menjelaskan teori – teori yang digunakan dalam pembangunan sistem ini.

2.2.1 Pengertian Data

Menurut Jogianto (2005), sumber data dapat diperoleh dari berbagai sumber untuk memperolehnya. Sumber data diklasifikasikan sebagai sumber data internal, sumber data personal, dan sumber data eksternal.

1. Data Internal sumbernya adalah orang, produk, layanan, dan proses. Data internal umumnya disimpan dalam basis data perusahaan dan biasanya dapat diakses.
2. Data Personal, sumber data personal bukan hanya berupa fakta, tetapi dapat juga mencakup konsep, pemikiran dan opini.
3. Data Eksternal, sumber data ekternal dimulai dari basis data komersial hingga sensor dan satelit. Data ini tersedia di *compact disk*, *flashdisk* atau media lainnya dalam bentuk film, suara gambar, atlas, dan televise[1].

Hirarki data dapat diorganisasikan menjadi beberapa level, antara lain sebagai berikut:

1. Elemen Data adalah satuan data terkecil yang tidak dapat dipecah lagi menjadi unit lain yang bermakna. Istilah lain dari elemen data dalam basis data telasional adalah field, kolom, item, dan atribut.
2. *Record* adalah gabungan sejumlah elemen data yang saling terkait. Istilah lain dari rekaman dalam basis data relasional adalah baris atau tupel.
3. *File* adalah kumpulan *record* sejenis yang mempunyai panjang atribut sama, namun berbeda isinya. Istilah lain dari *file* dalam basis data relasional adalah berkas, tabel, dan relasi[1].

Menurut Hasan (2009) data adalah, bentuk jamak dari datum. Data merupakan keterangan-keterangan tentang suatu hal, dapat berupa sesuatu yang diketahui atau dianggap, jadi data dapat diartikan sebagai sesuatu yang diketahui atau yang dianggap atau anggapan.

Berdasarkan pengertian dari para ahli diatas, data adalah fakta mentah atau rincian peristiwa yang belum diolah, maka dari itu data harus diolah terlebih dahulu menjadi informasi untuk dapat di terima oleh penerima. Data dapat berupa angka, karakter, simbol, gambar, suara, atau tanda-tanda yang dapat digunakan untuk dijadikan informasi. Suatu informasi bisa saja menjadi data apabila informasi tersebut digunakan kembali untuk pengolahan sistem informasi selanjutnya. Dalam dunia komputer data adalah segala sesuatu yang disimpan di dalam memori menurut format tertentu.

2.2.1.1 Pengertian Informasi

Menurut Jogiyanto (2005) mengatakan bahwa sistem informasi merupakan suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manejerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Untuk dapat berguna maka informasi harus didukung oleh tiga pilar sebagai berikut :

1. Tepat kepada orangnya atau relevan (*Relevancy*)

Berarti informasi tersebut mempunyai manfaat untuk pemakaiannya. Relevan informasi untuk tiap-tiap orang satu dengan yang lain berbeda.

2. Tepat Waktu *(Time Lines*)

Berarti informasi yang datang pada penerimaan tidak boleh terlambat, informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan didalam pengambilan keputusan.

3. Tepat nilainya atau akurat (*Accurate*)

Berarti informasi harus bebas dari kesalahan-kesalahan. Akurat juga berari informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi ke penerimaan kemungkinan banyak terjadi[2].

Menurut Nilai dari informasi ditentukan oleh dua hal yaitu manfaat dan biaya mendapatkannya. Jika suatu informasi manfaatnya lebih efektif dari biaya mendapatkannya maka informasi ini dapat dikatakan bernilai. Oleh karena itu, suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya dan sebagian besar informasi tidak dapat ditaksir keuntungannya dengan satuan nilai uang tetapi dapat ditaksir nilai efektifitasnya. Pengukuran dapat menggunakan analisis *cost effectiveness* atau *cost benefit.*

Menurut Kertahadi (2007) menyebutkan bahwa sistem informasi merupakan sebuah alat yang digunakan untuk menyajikan informasi sedemikian rupa sehingga bermanfaat bagi penerimanya. Tujuannya dari pembuatan alat berupa sistem informasi tesebut adalah untuk memberikan informasi dalam perencanaan, memulai, pengorganisasian, operasional sebuah perusahaan yang melayani sinergi organisasi dalam proses mengendalikan pengambilan keputusan.

Berdasarkan pengertian menurut para ahli secara umum informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan.

2.2.2 Basis Data (*Database)*

Basis data (*Database*) adalah sekumpulan informasi bermanfaat yang diorganisasikan kedalam tata cara yang khusus.

Database adalah kumpulan data, umumnya mendeskripsikan aktivitas suatu organisasi yang berhubungan atau. Misalnya, database suatu *minimarket* atau toko mungkin berisi informasi mengenai hal breikut :

- Entitas produk, entitas transaksi, entitas detail transaksi, entitas karyawan, entitas pemasok barang.

- Hubungan antar entitas, seperti produk yang dijual akan dicatat di detail transaksi, detail transaksi akan di catat di trasansaksi, setiap transaski akan disahkan oleh karyawan, dan pemasok akan memasukan produk – produk ke toko.

2.2.3. Pengertian *Web Application*

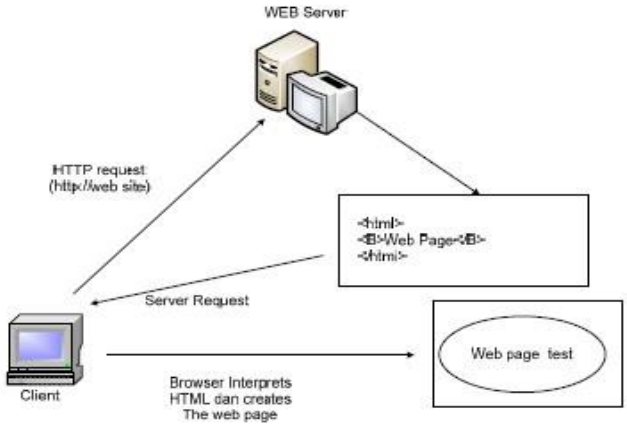
Web application dikenal sebagai aplikasi yang diakses melalui *web browser* dan melalui jaringan seperti internet atau intranet. Kemampuan untuk memperbarui dan memelihara aplikasi web tanpa harus mendistribusikan dan menginstal perangkat lunak pada kemungkinan ribuan komputer klien merupakan keunggulan teknologi ini, selain juga untuk *cross-platform compatibility*. Termasuk aplikasi *web common webmail*, penjualan ritel online, online pelelangan, wiki dan banyak fungsi lainnya.

Banyak situs internet yang memiliki halaman dengan sifat statis seperti profil perusahaan, artikel, dan keterangan-keterangan lain. Situs ini mempunyai dokumen dengan teks yang sederhana, image dan *hyperlinks* ke dokumen yang dimilikinya. Untuk mengembangkan situs yang bersifat statis, kita menggunakan teknologi client side. HTML dan *Cascading Style Sheet* (CSS) dapat digunakan untuk mengatur struktur dan menampilkan halaman isi. Seandainya ingin diperindah dapat ditambahkan *script* yang sifatnya *client side*, seperti JavaScript, Jscript ataupun VBScript.

Dengan berkembangnya internet, situs yang ada di internet tidak hanya berfungsi untuk mempresentasikan content tetapi cenderung berupa aplikasi yang kebanyakan terhubung ke suatu basis data. Pada tahapan ini situs akan bersifat dinamis, karena content yang dipresentasikan akan bervariasi dan berubah-ubah sesuai dengan data yang diminta dan action dari user. Untuk mengembangkan situs yang dinamis diperlukan teknologi *server side* seperti PHP, ASP, Perl dan CGI yang lain. Dengan teknologi *server side* kita dapat mengembangkan suatu aplikasi berbasis internet yang dapat mengahsilkan dan menampilkan content secara dinamis.

Pada Gambar 2.2 diperlihatkan cara kerja situs dengan content yang statis. Pada saat terjadi suatu *request* dari *browser*, *server web* selanjutnya membaca

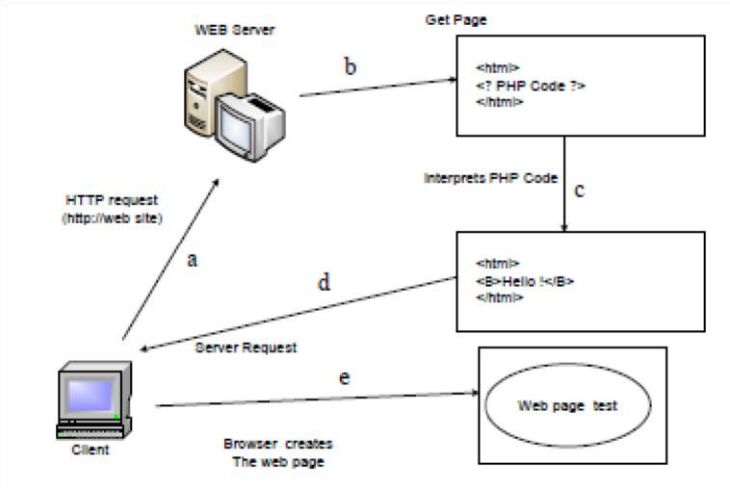
request yang dikirim oleh *browser*, mencari dan menemukan halaman di *server*, dan mengirim halaman yang diminta melalui internet ke *browser*. Selanjutnya halaman tersebut ditampilkan di *browser*.



Gambar 2.4 Web yang Statis

Pada Gambar 2.4 diperlihatkan cara kerja situs dengan *content* yang dinamis. Kronologis gambar adalah sebagai berikut :

a. *Browser* (*client*) melakukan *request* halaman web ke WEB Server.
b. WEB Server kemudian membaca request yang dikirim dari *browser*, mencari dan menemukan halaman di *server*.
c. Menterjemahkan perintah yang diberikan oleh bahasa program *server-side* menjadi halaman HTML.
d. Setelah konversi ke halaman HTML selesai kemudian mengirim halaman yang diminta ke *browser*.
e. Setelah halaman HTML yang dikirim dari WEB server sampai, selanjutnya *browser client* akan menampilkan halaman web.



Gambar 2.5 Web yang Dinamis

Dengan demikian aplikasi ini akan dibuat berbasis web agar mempermudah penggunaan, karena tidak memerlukan instalasi, mempermudah pengaksesan bagi manager dan pengunjunag museum.

2.2.3.1 Web Mapping

Web mapping merupakan proses dari perancangan, pemasangan, pembangunan dan penyampaian peta dalam *world wide web* dan produk didalamnya []. Dengan *web mapping,* bisa mendapatkan secara gratis mapping technologies dan geodata, dan memungkinkan orang yang mempunyai skill dapat membuat web mapsnya sendiri.

Wikipedia dalam situsnya mengelompokan tipe-tipe dari web maps, yaitu sebagai berikut :

1. *Analytic web maps*, web map ini menyediakan analisis SIG dengan geodata yang sudah tersedia, atau yang di upload oleh map user. Pada umumnya bagian analisis ada pada *serverside* SIG dan di client tampilan dari hasil analisis.
2. *Animated web maps*, web map yang menampilkan perubahan dalam map dengan animasi dari *graphical* atau *variable* sementara. Biasanya menggunakan berbagai data dan format multimedia seperti SVG, Adobe Flash, Java, Wuicktime, dll. Contoh *animated web maps* adalah weather maps.

3. *Collaborative web maps*, web map ini masih baru, sehingga masih belum stabil. Metode dari web map ini yaitu sama seperti Wikipedia project, yaitu untuk membuat dan mengembangkan peta dalam web. Web map ini memungkinkan petanya dapat di edit melalui web secara bersama-sama. Contoh *Collaborative web maps* adalah : OpenStreetMap, WikiMapia.
4. *Customisable web maps*, web map ini biasanya mempunyai sistem *web mapping* yang lebih kompleks, yang menyediakan API untuk digunakan oleh website dan produk lain. Sebagai contoh Open Layers Framework, Yahoo! Maps, dan Google Maps.
5. *Distributed web maps*, web map ini terbuat dari *data source* yang terdistribusi. Menggunakan WMS protocol, yang memungkinkan pengumpulan *data source*, menampilkan map baru, jika dibutuhkan dan mengirimkannya sebagai gambar yang terkombinasi yang berisi map layer yang diminta. Sistem kerjanya, satu server menyediakan topografi peta, dan server lain menyediakan *layer thematic*.
6. *Dynamically created web maps*, web map dibuat saat diminta setiap user reload halaman web. Data yang digunakan biasanya dari *data source* yang *dynamic* seperti *databases*. Cara kerjanya *webserver* generalisasi map menggunakan web map server atau suatu software sendiri. Sebagai contoh adalah Bhoosampada oleh Indian Space Research Organizations
7. *Hyper Maps*, merupakan segala macam yang menyediakan *planar presentation* dari sebagian *n-dimensional orthogonal web map* struktur dengan opsi untuk memilih axis untuk *depiction* dari dimensi tersebut.
8. *Interactive web maps*, yang paling menonjol dalam web map ini adalah *interactivity*. Membantu untuk menutupi kekurangan dari peta konvesional dan peta berbasis web (*web maps*). Secara teknik menggabungkan event, script, dan manipulasi DOM, untuk membuat peta yang interaktif. Sebagai contoh adalah Client Side Technoogies.
9. *Online atlases*, merupakan atlas yang dionline kan. Seperti halnya di atlas, pemetaan berdasarkan judul peta tersebut. Dan Peta tersebut static, hanya berupa gambar.

10. *Personalized web maps*, memungkinkan penguna peta untuk menerapkan filter mereka sendiri, pemilihan conten, dan pemilihan style dari penampilan peta dan simbolnya.
11. *Realtime web maps*, menunjukan situasi dan fenomena secara hampir realtime (dengan delay beberapa detik atau menit). Data didapatkan dari sensor dan peta *degenerated* atau di *update* pada interval yang telah diatur, atau saat diminta. Sebagai contoh weather maps, traffic maps, atau vehicle monitoring systems.
12. *Static web maps*, merupakan web map yang *view only*, hanya dapat dilihat, tidak ada animasi dan interaktif. Biasanya dibuat sekali, dan secara manual diupdate. Grapics format yang digunakan untuk peta pada umumnya berupa PNG, JPEG, GIF, atau TIFF untuk raster files.

2.2.4 Internet

Menurut Sidharta (1996) menyatakan bahwa internet adalah suatu bentuk interkoneksi dari sebuah jaringan komputer, dimana dapat memberikan bentuk layanan informasi secara lengkap. Dijelaskan lagi lebih lanjut, bahwa internet juga terbukti sebagai rekan maya atau virtual yang ampuh, yang biasa digunakan dalam media bisnis, politik, dan bahkan untuk hiburan semata.

Walaupun secara fisik Internet merupakan suatu bentuk interkoneksi antar jaringan komputer namun secara umum Internet harus dapat dipandang dan juga dipahami sebagai sebuah sumber daya informasi.

Menurut Sarwono (2012) internet merupakan sebuah kumpulan jaringan yang memiliki skala global. Bahkan, lebih lanjut Sarwono mengatakan bahwa tidak ada satupun orang yang mampu bertanggung jawab untuk menjalankan internet itu sendiri. Mulanya internet hanya digunakan dalam kebutuhan militer, namun kini internet merambah pada keperluan masyarakat sipil dan juga bentuk hiburan

Berdasarkan pengertian para ahli diatas, Internet adalah salah satu jaringan besar, yang seringkali dipandang sebagai sebuah pusat dari berkembangnya informasi. Internet bekerja dengan cara menghubungkan seluruh komputer di dunia, ke dalam satu jaringan, sehingga dapat saling berkomunikasi satu sama lain.

2.2.5 QR-CODE

Kode Batang atau yang biasa disebut *barcode* merupakan representasi data yang dapat dibaca oleh mesin optik yang berhubungan dengan objek yang terpasang. Pada awalnya, *barcode* mewakili data dengan memvariasikan lebar dan jarak dari garis pararel secara sistematis, dan dapat disebut sebagai linear atau satu dimensi (1D). Kemudian *barcode* berevolusi menjadi persegi panjang, titik, segi enam dan pola geometris lainnya dalam dua dimensi (2D). Meskipun sistem 2D menggunakan berbagai simbol, tapi pada umumnya tetap disebut sebagai *barcode* juga atau dikenal juga dengan sebutan kode matriks. Kelebihan dari kode matriks ini adalah dapat merepresentasikan lebih banyak data untuk setiap area unitnya. Awalnya *barcode* dipindai oleh pemindai (*scanner*) optic khusus yang disebut pembaca *barcode* (*barcode scanner*). Kemudian, *scanner* dan *software* interpretative mulai tersedia pada perangkat termasuk printer *desktop* dan *smartphones*.

Quick Respons Code atau biasa disebut QR *code* merupakan salah satu jenis kode matriks yang pertama kali dirancang oleh industry otomotif di Jepang. Sebuah QR *code* menggunakan empat mode standar *encoding* (yaitu numerik, alfanumerik, *byte* / biner dan kanji) untuk menyimpan data secara efisien. Sistem QR *code* pun telah manjadi populer di luar industri otomotif karena pembacaan yang cepat dan kapasitas penyimpanan yang lebih besar dibandingkan dengan standar *barcode* UPC.



Gambar 2. 1 Contoh QR Code

QR *code* terdiri dari modul hitam (titik persegi) diatur dalam kotak persegi pada latar belakang putih yang dapat dibaca oleh perangkat pencitraan seperti

kamera dan diolah menggunakan koreksi kesalahan Reed-Solomon hingga gambar dapat dengan tepat di interpretasikan. Data tersebut kemudian diekstraksi dari pola yang ada dari kedua komponen horizontal dan komponen vertical pada gambar.

2.2.5.1 Standar

Ada beberapa standar yang mencakup pengkodean data sebagai QR-*code*:

a. October 1997 – AIM (*Association for Automatic Identification and Mobility*) *International*

b. January 1999 – JIS X 0510

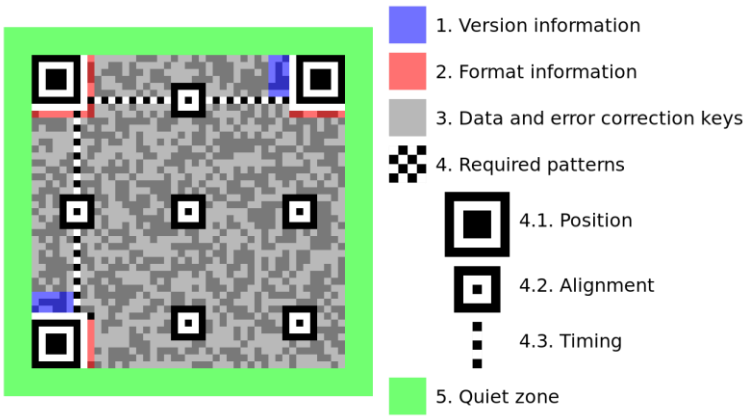
c. June 2000 – ISO/IEC 18004:2000 *Teknologi informasi - Indentifikasi otomatis dan teknik data capture – Simbologi barcode – QR code*

d. 1 September 2006 – ISO/IEC 18004:2006 *Teknologi informasi – Indetifikasi otomatis dan teknik data capture – QR code 2005 spesifikasi simbologi barcode.* Mendefinisikan simbol QR *code* 2005, perpanjangan QR *code* model 2. Tidak menentukan cara membaca simbol QR *code* model 1.

Pada lapisan aplikasi, terdapat beberapa variasi antara sebagian besar implementasi. NTT Docomo Jepangtelah membentuk standar de facto untuk pengkodean URL, informasi kontak dan beberapa jenis data lainnya. Proyek *opensource* "ZXing" menyimpan daftar tipe data QR *code.*

2.2.5.2 Desain

Tidak seperti *barcode* satu dimensi yang dirancang secara mekanis untuk dipindai oleh sinar sempit cahaya (*narrow beam of light*), QR *code* dideteksi oleh sensor gambar digital dua dimensi dan kemudian dianalisis secara digital oleh prosesor yang terprogram. Prosesor menempatkan togak kotak khas di sudut-sudut gambar QR *code* menggunakan persegi yang lebih kecil (atau beberapa kotak) dekat sudut keempat untuk menormalkan gambar untuk ukuran, orientasi dan sudut pandang. Titik-titik kecil di seluruh QR *code* tersebut kemudian dikonversi ke bilangan biner dan divalidasi dengan kode koreksi kesalahan.



Gambar 2. 2 Struktur QR Code dan Elemen Fungsionalnya

2.2.6 *Audio*

Audio adalah suara atau bunyi yang dihasilkan oleh getaran suatu benda, agar dapat tertangkap oleh telinga manusia getaran tersebut harus kuat minimal 20 kali/detik. Suara yaitu suatu getaran yang dihasilkan oleh gesekan , pantulan dan lain-lain, antara benda-banda. Sedangkan gelombang yaitu suatu getaran yang terdiri dari Amplitudo dan juga waktu. Suara dibangun oleh periode, Apabila Tidak Berarti itu bukanlah Suara.

Definisi audio yang lainnya adalah merupakan salah satu elemen yang penting, karena ikut berperan dalam membangun sebuah sistem Komunikasi dalam bentuk suara, ialah suatu sinyal elektrik yang akan membawa unsur-unsur bunyi didalamnya. Audio itu terbentuk melalui beberapa tahap, diantaranya: tahap pengambilan atau penangkapan suara, sambungan transmisi yang membawa bunyi, amplifier dan lain-lain.

2.2.7 Perangkat Lunak Pembangun Sistem

Dalam membangun aplikasi ini menggunakan PHP sebagai bahasa pemrograman untuk mengimplementasikan fungsi-fungsi yang dirancang, serta dibantu dengan MYSQL yang digunakan sebagai penyimpanan data atau database dari aplikasi yang dibangun, *Codeigniter* sebagai *framework* PHP, XAMPP sebagai server untuk menjalanan website dan *Sublime Text* sebagai editor yang digunakan

untuk pengkodean PHP. Berikut ini penjelasan dari perangkat lunak yang digunakan:

2.2.7.1 Pengertian PHP

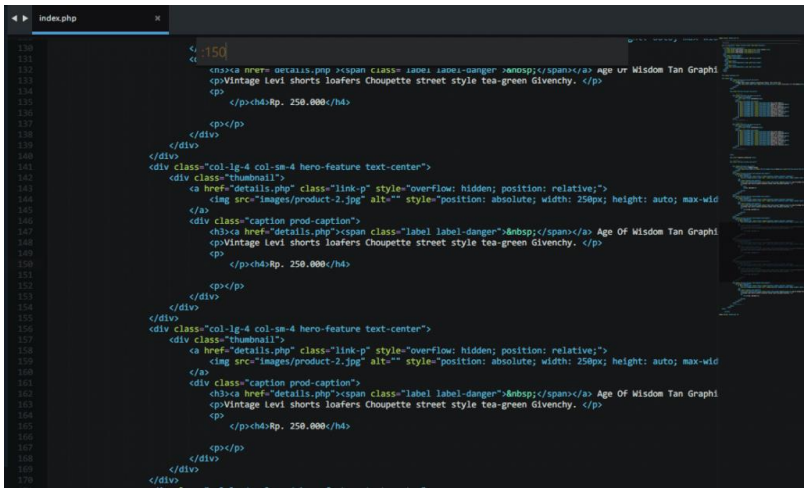
PHP adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memrogram situs web dinamis, walaupun tidak tertutup kemungkinam untuk digunakan untuk pemakaian lain. PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP bernama FI (*Form Interpreted*). Pada saat tersebut PHP adalah sekumpulan *script* yang digunakan untuk mengolah data *form* dari web. Perkembangan selanjutnya adalah Rasmus melepaskan kode sumber tersebut dan menamakannya PHP/ FI, pada saat tersebut kepanjangan dari PHP/ FI adalah *Personal Home Page*/ *Form Interpreter*. Dengan pelepasan kode sumber ini menjadi *open source*, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP. Pada November 1997, dirilis PHP/ FI 2.0. Pada rilis ini interpreter sudah diimplementasikan dalam C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/ FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend, menulis ulang interpreter PHP menjadi lebih bersih, lebih baik dan lebih cepat. Kemudian pada Juni 1998 perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan nama rilis tersebut menjadi PHP 3.0.Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai. Versi ini banyak dipakai sebab versi ini mampu dipakai untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan proses dan stabilitas yang tinggi.Pata Juni 2004, Zend merilis PHP 5.0. Versi ini adalah versi mutakhir dari PHP. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Dalam versi ini juga dikenalkan model pemrograman berorientasi objek baru untuk menjawab perkembangan bahasa pemrograman kearah pemrograman berorientasi objek.

PHP merupakan bahasa pemrograman web, penulisan *script* PHP dapat dijadikan satu dengan kode html dan disimpan dalam satu *file* yang sama ataupun

berdiri sendiri dengan nama *file* berekstensi *.php. Ciri khas PHP diawali dengan tanda “<?php“ dan diakhiri dengan “?>”. PHP memiliki fasilitas untuk melakukan koneksi ke *database*. Script program PHP disimpan dan dijalankan di sisi *server* sehingga keamanan data lebih terjamin.

Dengan demikian PHP digunakan dalam aplikasi ini karena PHP paling banyak digunakan dan tidak sulit untuk penerapan pada *server*.



Gambar 2.8 Contoh Coding PHP

2.2.7.2 Pengertian MySQL

MySQL merupakan aplikasi *database server*. Perkembangannya disebut SQL yang merupakan kepanjangan dari *Structured Query Language*. SQL merupakan bahasa terstruktur yang digunakan untuk mengolah *database*. MySQL dapat digunakan untuk membuat dan mengelola database beserta isinya. Kita dapat memanfaatkan MySQL untuk menambahkan, mengubah, dan menghapus data yang berada dalam database.

MySQL adalah sebuah sistem manajemen *database* relasi (*relation database management system*) yang bersifat *open source.* MySql merupakan buah pikiran dari Michael “Monty” Widenius, Daveid axmark dan Allan Larson yang dimulai tahun 1995.Mereka bertiga kemudian mendirikan perusahaan bernama MySQL AB di Swedia.

MySQL versi 1.0 rilis pada Mei 1996 dan penggunaannya hanya terbatas di kalangan perusahaan saja. Barulah pada bulan Oktober 1996, MySQL versi 3.11.0

di rilis ke masyarakat luas. MySQL menggunakan bahasa standar SQL (*Structure Query Languange*) sebagai bahasa interaktif dalam mengelola data. MySQL memiliki kinerja, kecepatan proses dan ketangguhan yang tidak kalah dibanding *database-database* besar lainnya yang komersil seperti ORACLE, Sysbase, Unify dan sebagainya. MySQL dapat berjalan di atas banyak sistem operasi Linux, Windows, Solaris, FreeBSD, Mac OS X, dan lain sebagainya.

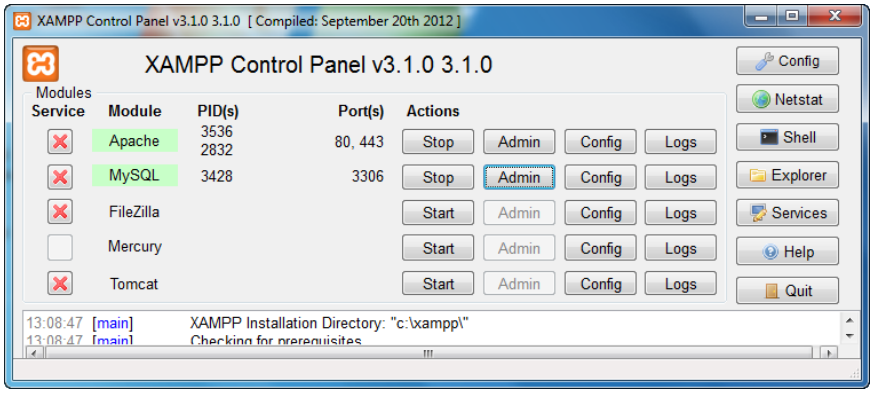
Dengan demikian MySQL sebagai database yang digunakan dalam aplikasi ini karena sangat cocok diterapkan pada PHP.

2.2.7.3 Pengertian XAMPP

XAMPP adalah *software* web *server apache* yang di dalamnya tertanam *server* MySQL yang didukung dengan bahasa pemrograman PHP untuk membuat website yang dinamis. XAMPP sendiri mendukung dua system operasi yaitu windows dan Linux. Untuk linux dalam proses penginstalanny menggunakan *command line* sedangkan untuk windows dalam proses penginstalannya menggunakan interface grafis sehingga lebih mudah dalam penggunaaan XAMPP di Windows di banding dengan Linux.

Apache merupakan web *server* yang digunakan untuk menampilkan website di internet seperti menggunakan Mozilla fire fox, Google Crome, IE, Safari, dan lain-lain berdasarkan kode-kode yang di tulis di dalam website tersebut baik menggunakan bahasa pemrograman HTML maupun PHP yang mengambil suatu *database* yang dibangun di MySQL, sehingga terbentuklah sebuah website yang dapat di lihat di Mozilla fire fox dan kawan-kawannya. *Apache* sendiri bersifat *open source* sehingga dapat digunakan oleh siapa saja dan dikembangkan oleh siapa saja tentunya bagi yang mampu mengembagkannya.

Dengan demikian XAMPP dipilih menjadi server untuk menjalankan dan mengembangkan aplikasi ini, karena lengkap dan memenuhi semua perangkat lunak yang digunakan.



Gambar 2.9 Xampp

2.2.7.4 Pengertian Codeigniter

Codeigniter adalah sebuah *framework* PHP yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi web berbasis PHP dibandingkan jika menulis semua kode program dari awal.

Codeigniter menyediakan banyak *library* untuk mengerjakan tugas-tugas yang umumnya ada pada sebuah aplikasi berbasis web. Selain itu, struktur dan susunan logis dari *codeigniter* membuat aplikasi yang dibuat menjadi semakin teratur dan rapi. Dengan demikian *developer* dapat fokus pada fitur-fitur apa yang dibutuhkan aplikasi dengan membuat kode program seminimal mungkin.

Codeigniter pertama kali dibuat oleh Rick Ellis, CEO Ellislab, Inc. sebuah perusahaan yang memproduksi sebuah CMS (*content management system*) yang cukup handal, yaitu *expression engine*. Saat ini, *codeigniter* dikembangkan dan dimaintain oleh *expression engine development team*.

Dengan demikian *Codeigniter* digunakan sebagai *framework* untuk membuat aplikasi ini, karena dengan menggunakan *codeigniter* tidak perlu lagi membuat struktur dan fungsi-fungsi kewangka awal, sehingga akan mempercepat dalam proses pembuatan aplikasi ini.

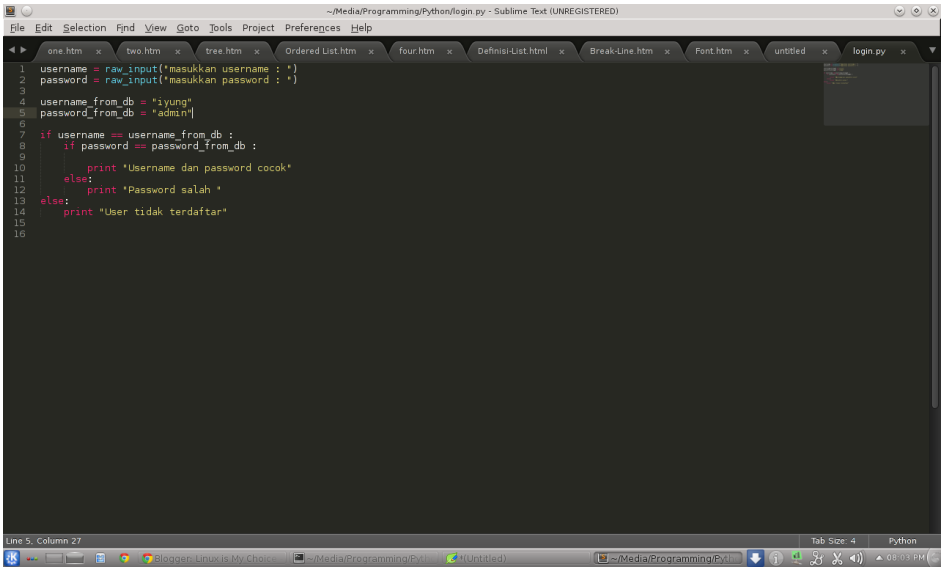
2.2.7.5 Pengertian Sublime Text

Sublime Text itu adalah salah satu *text editor* yang biasa digunakan oleh para *programmer*, khususnya *Web Developer*. Bisa diibaratkan sebagai senjata-nya *Web Developer*. Setiap *web developer* wajib untuk menggunakan *Sublime Text* sebagai

senjata hariannya. Dengan banyaknya fitur-fitur yang bagus menjadikan *Sublime Text* pilihan para Web Developer. Beberapa fiturnya sebagai berikut :

1. *Multiple Selection*, fitur untuk merubah banyak text dengan satu kali ketik.
2. *Find in Project*, fitur untuk mencari file dalam project
3. *Go to line*, fitur untuk berpindah baris dengan cepat
4. *Package Control dan Themes*, fitur untuk menambahkan fitur dan mengganti tema

Dengan demikian *Sublime Text* digunakan sebagai *editor* untuk membuat aplikasi ini, karena dengan beberapa fitur yang sangat membantu untuk mempercepat dalam pengkodean.



Gambar 2.10 Sublime Text

1. Bootstrap

Bootstrap merupakan sebuah framework css yang memudahkan pengembang untuk membangun website yang menarik dan responsif. Tidak konsistensinya terhadap aplikasi individual membuat sulitnya untuk mengembangkan dan pemeliharaannya. Bootstrap adalah css tetapi dibentuk dengan LESS, sebuah pre-prosessor yang memberi fleksibilitas dari css biasa. Bootstrap memberikan solusi rapi dan seragam terhadap solusi yang umum, tugas interface yang setiap pengembang hadapi. Bootstrap dapat dikembangkan dengan

tambahan lainnya karena ini cukup fleksibel terhadap pekerjaan design butuhkan (Otto, 2011).

2. CSS

CSS adalah singkatan dari Cascading Style Sheet. Style sheet mengacu pada dokumen itu sendiri.Style sheet telah digunakan untuk mendesain dokumen selama bertahun-tahun.Mereka adalah spesifikasi teknis untuk layout, apakah cetak atau online. Desainer cetak menggunakan style sheet untuk memastikan bahwa desain mereka dicetak persis dengan spesifikasi yang telah mereka buat. Sebuah style sheet untuk halaman Web melayani tujuan yang sama, tetapi dengan fungsionalitas tambahan juga mengatakan bahwa Engine Viewing (Web browser) bagaimana untuk membuat dokumen yang dilihat (Kyrnin, What is CSS?, 2012).

2.2.8 Perancangan Sistem

Dalam membangun aplikasi ini perancangan yang digunakan adalah perancangan berorientasi objek dengan menggunakan pemodelan *Unified Modelling Language* (UML), serta pemrograman berorientasi objek. Berikut ini penjelasan dari perancangan sistem yang digunakan:

2.2.8.1 Pengertian Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat 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 apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari tiga notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

UML ini menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu :

1. *Use Case Diagram* untuk memodelkan proses bisnis.
2. *Conceptual Diagram* untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. *Sequence Diagram* untuk memodelkan pengiriman pesan (*message*) antar objek.
4. *Collaboration Diagram* untuk memodelkan interaksi antar objek.
5. *State Diagram* untuk memodelkan perilaku objek di dalam sistem.
6. *Activity Diagram* untuk memodelkan perilaku userdan objek di dalam sistem.
7. *Class Diagram* untuk memodelkan struktur kelas.
8. *Objek Diagram* untuk memodelkan struktur objek.
9. *Component Diagram* untuk memodelkan komponen objek.
10. *Deployment Diagram* untuk memodelkan distribusi aplikasi.

Berikut akan dijelaskan 6 macam diagram yang digunakan dalam perancangan aplikasi ini, yaitu *use case diagram, activity diagram, class diagram, sequence diagram, deployment diagram,* dan *component diagram.*

1. *Use case diagram*

Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use case diagram* terdiri atas *diagram* untuk *use case* dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi.

Use case merepresentasikan operasi-operasi yang dilakukan oleh *actor*.*Use case* digambarkan berbentuk elips dengan nama operasi dituliskan didalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke *use case.*

2. *Activity Diagram*

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *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.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa object swimlane untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu. []

3. *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/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 :

a) Nama (dan *stereotype*)
b) Atribut
c) 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 anakanak yang mewarisinya
c) *Public* , dapat dipanggil oleh siapa saja

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*. Sesuai dengan perkembangan class model, class dapat dikelompokkan menjadi package.

4. *Sequence Diagram*

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 atar 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. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada *fase* desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*.

5. *Deployment Diagram*

Deployment 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 fisikal Sebuah node adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men- *deploy* komponen dalam lingkungan sebenarnya. Hubungan antar node (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.

6. *Component Diagram*

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

2.2.8.2 Pemrograman Berorientasi Objek

Analisis dan desain berorientasi objek adalah cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep sekitar dunia nyata. Dasar pembuatan adalah objek, yang merupakan kombinasi antara struktur data dab perilaku dalam suatu entitas. Model berorientasi objek bermanfaat untuk memahami masalah, komunikasi dengan ahli aplikasi, pemodelan suatu organisasi, meyiapkan dokumentasi serta perancangan program dan basis data. Pertama-tama suatu model analisis dibuat untuk menggambarkan aspek dasar dari domain aplikasi, dimana model tersebut berisi objek yang terdapat dalam domain aplikasi termasuk deskripsi dari keterangan objek dan perilakunya.

Secara spesifik, pengertian berorientai objek berarti bahwa mengorganisasi perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data dan perilakunya. Hal ini yang membedakan dengan pemograman konvensional dimana struktur data dan perilaku hanya berhubungan secara terpisah. Terdapat beberapa cara untuk menentukan karateristik dalam pendekatan berorientasi objek, tetapi secara umum mencakup empat hal, yaitu identifikasi, klasifikasi, *polymorphism* (polimorfisme) dan *inheritance* (pewarisan).

Metodologi pengembangan sistem berorientasi objek mempunyai tiga karateristik utama, yaitu :

a. Encapsulation

Encapsulation (pengkapsulan) merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses. Data dan prosedur atau fungsi dikemas dalam bersama-sama dalam suatu objek, sehingga prosedur atau fungsi lain dari luar tidak dapat mengaksesnya. Data terlindung dari prosedur atau objek lain kecuali prosedur yang berada dalam objek itu sendiri.

b. Inheritance

Inheritance (pewarisan) adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi atribut dan metoda dari induknya langsung. Atribut dan metoda dari objek induk diturunkan kepada anak objek, demikian seterusnya. Pendefinisian objek dipergunakan untuk membangun suatu hirarki dari objek turunannya, sehingga tidak perlu membuat atribut dan metoda lagi pada anaknya, karena telah mewarisi sifat induknya.

c. Polymorphism

Polymorphism (polimorfisme) yaitu konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku berbeda. Polimorfisme mempunyai arti bahwa operasi yang sama mungkin mempunyai perbedaan dalam kelas yang berbeda.

2.2.9 Android

Menurut Safaat (2012 : 1), Android adalah sistem operasi berbasis Linux bagi telepon seluler seperti telepon pintar dan komputer tablet. Android juga menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk berbagai macam piranti gerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat piranti lunak untuk ponsel. kemudian dalam pengembangan 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.

2.2.9.1 Versi Android

Safaat (2012 : 10) memaparkan penjelasan berbagai macam versi Android seperti di

bawah ini :

1. Android versi 1.1
2. Android versi 1.5 (Cupcake)
3. Android versi 1.6 (Donut)
4. Android versi 2.0/2.1 (Eclair)
5. Android versi 2.2 (Froyo: Frozen Yoghurt)
6. Android versi 2.3 (Gingerbread)
7. Android versi 3.0/3.1 (Honeycomb)
8. Android versi 4.0 (ICS: Ice Cream Sandwich)
9. Android versi 4.1 (Jelly Bean)
10. Android versi 4.4 (KitKat)
11. Android versi 5.0 (Lollipop)
12. Android versi 6.0 (Marshmallow)

2.2.9.2 Fitur Android

Adapun beberapa fitur – fitur menurut Safaat (2012 : 5) yang tersedia di Android adalah sebagai berikut ini :

- Kerangka aplikasi: memungkinkan penggunaan dan penghapusan komponen yang tersedia.
- Dalvik mesin virtual: mesin virtual dioptimalkan untuk perangkat telepon seluler.
- Grafik: grafik di 2D dan grafis 3D berdasarkan pustaka OpenGL.
- SQLite: untuk penyimpanan data.
- Mendukung media: audio, video, dan berbagai format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- GSM, Bluetooth, EDGE, 3G, 4G dan WiFi (tergantung piranti keras)
- Kamera, *Global Positioning System* (GPS), kompas, NFC dan *accelerometer* (tergantung piranti keras)

2.2.9.3 Android SDK

Menurut Safaat (2012 : 5), Android SDK adalah *tools* API (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi pada platform Android yang menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di-*release* oleh Google. Saat ini disediakan Android SDK (Software Development Kit) sebagai alat bantu dan API untuk mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Sebagai *platform* Android aplikasi – netral, Android memberi Anda kesempatan untuk membuat Aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan *Handphone/Smartphone*.

2.2.10 IDE Eclipse

Menurut Fatimah (2011), Eclipse adalah sebuah IDE (*Integrated Development*
Environment) untuk mengembangkan perangkat lunak agar dapat dijalankan di semua platform (*platform-independent*). Berikut ini adalah sifat dari Eclipse:

- *Multi-platform*: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
- *Mulit-language*: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Java, Cobol, Python, Perl, PHP, dan lain sebagainya.

Multi-role: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun dapat digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.