BAB 2
LANDASAN TEORI

Landasan teori menjelaskan dasar-dasar teori yang digunakan dalam penelitian yang dulakukan pada tugas akhir ini. Teori-teori yang digunakan meliputi keamanan sistem, kriptografi, algoritma RSA, teknologi *mobile*, GSM, SMS, Android, Eclipse, UML, XML, dan Java.

2.1 Keamanan Sistem

Masalah keamanan merupakan salah satu aspek penting dari sebuah sistem informasi. Sayang sekali masalah keamanan ini sering kali kurang mendapat perhatian dari para pemilik dan pengelola sistem informasi. Seringkali masalah keamanan berada di urutan kedua, bahkan di urutan terakhir dalam daftar hal-hal yang dianggap penting. Apabila menggangu performansi dari sistem, seringkali keamanan dikurangi/ditiadakan. Informasi saat ini sudah menjadi sebuah komoditi yang sangat penting. Bahkan ada yang mengatakan bahwa sudah berada di sebuah *"information-based society*". Kemampuan untuk mengakses dan menyediakan informasi secara cepat dan akurat menjadi sangat esensial bagi sebuah organisasi, baik yang berupa organisasi komersial (perusahaan), perguruan tinggi, lembaga pemerintahan, maupun individual (pribadi). Hal ini dimungkinkan dengan perkembangan pesat di bidang teknologi komputer & telekomunikasi.

Sangat pentingnya nilai sebuah informasi menyebabkan seringkali informasi diinginkan hanya boleh diakses oleh orang-orang tertentu. Jatuhnya informasi ke tangan pihak lain (misalnya pihak lawan bisnis) dapat menimbulkan kerugian bagi pemilik informasi. Untuk itu keamanan dari sistem informasi yang digunakan harus terjamin dalam batas yang dapat diterima [9].

2.1.1 Pengertian Keamanan Sistem

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul, bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Keamanan adalah suatu kinerja dalam menghadapi masalah baik internal maupin eksternal yang terjadi terhadap suatu ruang lingkup demi terciptaanya keadaan yang seharusnya [8].

2.2 Kriptografi

Kriptografi berasal dari kata Yunani, yaitu *Crypto* yang berarti rahasia dan *Grapho* yang berarti menulis. Menurut (Schneier , 1996) secara umum Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan (*Cryptography is the art and science of keeping messages secure*). Menurut (Menezes, 1996) sebagai pembanding, terdapat pula definisi yang berbeda, Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi. Kata "seni" di dalam definisi di atas berasal dari fakta sejarah bahwa pada masa-masa awal sejarah kriptografi, setiap orang mungkin mempunyai cara yang unik untuk merahasiakan pesan. Cara-cara unik tersebut mungkin berbeda-beda pada setiap pelaku kriptografi sehingga setiap cara menulis pesan rahasia pesan mempunyai nilai estetika tersendiri sehingga kriptografi berkembang menjadi sebuah seni merahasiakan pesan (kata "*graphy*" di dalam "*cryptography*" itu sendiri sudah menyiratkan sebuah seni).

Menurut catatan sejarah, kriptografi sudah digunakan oleh bangsa Mesir sejak 4000 tahun yang lalu oleh raja-raja Mesir pada saat perang untuk mengirimkan pesan rahasia pada panglima perangnya melalui kurir-kurirnya. Orang yang melakukan penyandian ini disebut *Cryptographer*, sedangkan orang yang mendalami ilmu dan seni dalam membuka atau memecahkan suatu algoritma kriptografi tanpa harus mengetahui kuncinya disebut kriptanalis.

Seiring dengan perkembangan teknologi, algoritma kriptografi pun mulai berubah menuju arah algoritma kriptografi yang lebih rumit dan kompleks. Kriptografi mau tidak mau harus diakui mempunyai peranan yang paling penting dalam peperangan sehingga algoritma kriptografi berkembang cukup pesat pada

saat perang dunia I dan perang dunia ke II. Menurut catatan sejarah, terdapat beberapa algoritma kriptografi yang pernah digunakan dalam peperangan, diantaranya adalah ADFVGX yang dipakai oleh Jerman pada perang dunia I, Sigaba/M-134 yang digunakan oleh Amerika Serikat pada perang dunia II, *Typex* oleh Inggris, dan *Purple* oleh Jepang. Selain itu jerman juga mempunyai mesin legendaris yang dipakai untuk memecahkan sandi yang dikirim oleh pihak musuh dalamn peperangan yaitu, Enigma [8].

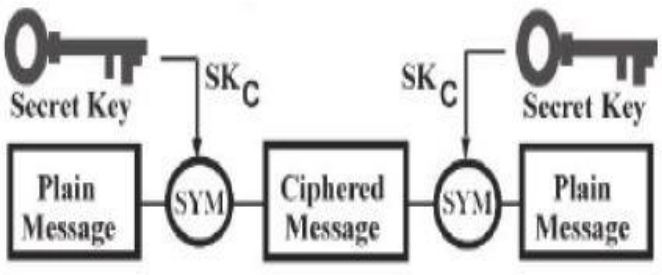
Algoritma kriptografi yang baik tidak ditentukan oleh kerumitan dalam mengelolah data atau pesan yang akan disampaikan, yang penting, algoritma tersebut memiliki 4 persyaratan berikut [8]:

1. **Kerahasiaan**. Pesan *(plaintext)* hanya dapat dibaca oleh dua pihak kewenangan.
2. **Autentifikasi.** Pengirim pesan harus dapat diindentifikasi dengan pasti, penyusup harus dipastikan tidak bisa berpura – pura menjadi orang lain.
3. **Integritas.** Penerima Pesan harus dapat memastikan bahwa pesan yang dia terima tidak dimodifikasi ketika dalam melakukan proses transmisi data.
4. ***Non – Repudiation***. Pengirim pesan harus tidak bisa menyangkal pesan yang dia kirim.

2.2.1 Sistem Kriptografi

Kriptografi pada dasarnya terdiri dari dua proses, yaitu proses enkripsi dan proses dekripsi. Proses enkripsi adalah proses penyandian pesan terbuka menjadi pesan rahasia (*ciphertext*). *Ciphertext* inilah yang nantinya akan dikirimkan melalui saluran komunikasi terbuka. Pada saat *ciphertext* diterima oleh penerima pesan, maka pesan rahasia tersebut diubah lagi menjadi pesan terbuka melalui proses dekripsi sehingga pesan tadi dapat dibaca kembali oleh penerima pesan.

Menurut (Sadikin, 2012) secara umum, proses enkripsi dan dekripsi dapat dilihat pada gambar II.1 [8]:



Gambar 2.1 Proses Enkripsi dan Dekripsi

Dalam sistem komputer, pesan terbuka (*plaintext*) diberi lambang M, yang merupakan singkatan dari *message*. *Plaintext* ini dapat berupa tulisan, foto, atau video yang berbentuk data biner. *Plaintext* inilah yan nantinya akan dienkripsi menjadi pesan rahasia atau *ciphertext* yang dilambangkan dengan C (*ciphertext*).

Secara matematis, fungsi enkripsi ini dinotasikan dengan :

$$E(M)=C$$

Sedangkan fungsi deskripsi adalah proses pengembalian dari *chiphertext* menjadi *plaintext* kembali, secara matematis dinotasikan sebagai berikut :

$$D(C)=M$$

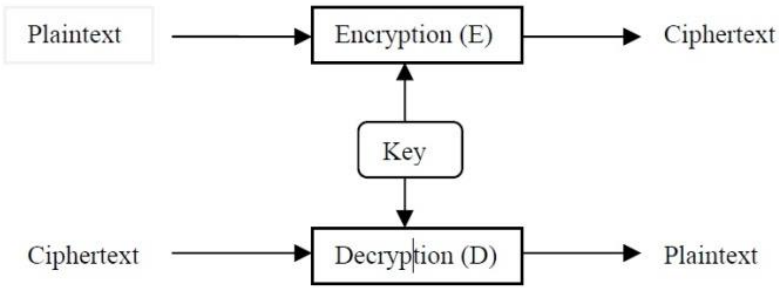
$$D(E(M))=M$$

2.2.2 Kriptografi Kunci Rahasia dan Kunci Publik

Menurut (Sadikin, 2012) pada dasarnya terdapat dua jenis algoritma kriptografi berdasarkan kunci yang digunakan. Pertama adalah kriptografi dengan mengunakan *secret key* dan yang kedua adalah kriptografi yang menggunakan *public key*. *Public key* menggunakan dua kunci yang berbeda dimana satu kunci digunakan untuk melakukan enkripsi dan kunci yang lain digunakan untuk melakukan dekripsi [8].

2.2.2.1 Kriptografi Kunci Rahasia

Kriptografi kunci rahasia atau *secret key* adalah kriptografi yang hanya melibatkan satu kunci dalam satu proses enkripsi dan dekripsi. Proses dekripsi dalam kriptografi *secret key* ini adalah kebalikan dari proses enkripsi.



Gambar 2.2 Kriptografi Simetris

Menurut (Chey Cobb, CISSP, 2004) kriptografi *secret key* seringkali disebut sebagai kriptografi konvensional atau kriptografi simetris (*Symmetric Cryptography*) dimana proses dekripsi adalah kebalikan dari proses enkripsi dan menggunakan kunci yang sama.

Kriptografi simetris dapat dibagi menjadi dua, yaitu penyandian blok *(block cipher*) dan penyandian alir (*stream cipher*) . Penyandian blok bekerja pada suatu data yang terkelompok menjadi blok-blok data atau kelompok data dengan panjang data yang telah ditentukan. Pada penyandian blok, data yang masuk akan dipecah-pecah menjadi blok data yang telah ditentukan ukurannya. Penyandian alir bekerja pada suatu data bit tunggal atau terkadang dalam satu *byte*. Jadi format data yang mengalami proses enkripsi dan dekripsi adalah berupa aliran bit-bit data [9].

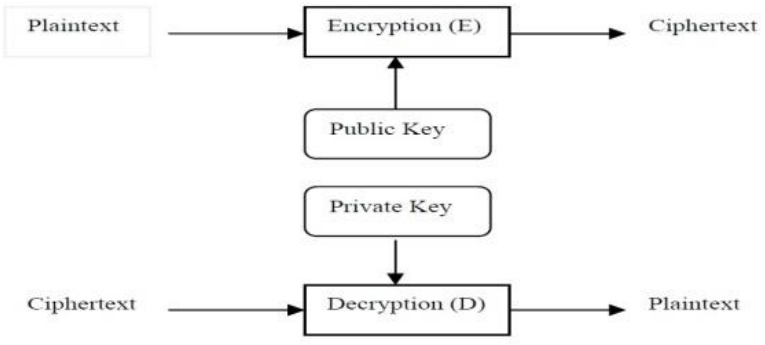
Algoritma yang ada saat ini kebanyakan bekerja untuk penyandian blok karena kebanyakan proses pengiriman data pada saat ini menggunakan blok-blok data yang telah ditentukan ukurannya untuk kemudian dikirim melalui saluran komunikasi [2].

2.2.2.2 Kriptografi Kunci Publik

Sistem kriptografi kunci publik atau sering juga disebut sebagai kriptografi kunci asimentrik pertama kali diusulkan oleh Deffie & Hellman pada tahun 1976 (Deffie & Hellman, 1976). Ide kriptografi kunci publik sebenarnya mirip dengan cara kerja kunci gembok. Bayangkan terdapat sebuah peti yang berisi pesan rahasia lalu peti itu dikunci dengan gembok. Kemudian peti ini dikirim ke penerima yang memiliki kunci untuk membuka gembok. Penerima dapat membuka gembok apabila kunci yang dipegang olehnya merupakan pasangan gembok.

Kriptografi kunci publik sering disebut dengan kriptografi asimetris. Berbeda dengan kriptografi kunci rahasia, kunci yang digunakan pada proses enkripsi dan proses dekripsi pada kriptografi public key ini berbeda satu sama lain. Jadi dalam kriptografi *public key*, suatu *key generator* akan menghasilkan dua kunci berbeda dimana satu kunci digunakan untuk melakukan proses enkripsi dan kunci yang lain digunakan untuk melakukan proses dekripsi.

Kunci yang digunakan untuk melakukan enkripsi akan dipublikasikan kepada umum untuk dipergunakan secara bebas. Oleh sebab itu, kunci yang digunakan untuk melakukan enkripsi disebut juga sebagai *public key*. Sedangkan kunci yang digunakan untuk melakukan dekripsi akan disimpan oleh pembuat kunci dan tidak akan dipublikasikan kepada umum. Kunci untuk melakukan dekripsi ini disebut *private key*.



Gambar 2.3 Kriptografi Asimetris

Dengan cara demikian, semua orang yang akan mengirimkan pesan kepada pembuat kunci dapat melakukan proses enkripsi terhadap pesan tersebut, sedangkan proses dekripsi hanya dilakukan oleh pembuat atau pemilik kunci dekripsi. Dalam kenyataannya, kriptografi asimetris ini dipakai dalam suatu layanan untuk mengakses suatu *server* [2].

2.3 Algoritma RSA

Menurut (Burnett & Paine, 2001) dari sekian banyak metode kriptografi asimetris yang pernah dibuat, algoritma yang paling popular adalah algoritma RSA. Algoritma RSA diciptakan oleh 3 orang peneliti dari MIT *(Masschussets Institute of Technology)* pada tahun 1976, yaitu Ron (R)ivest, Adi (S)hamir dan Leonard (A)dleman. Keamanan algoritma RSA terletak pada sulitnya menfaktorkan bilangan yang besar menjadi faktor – faktor prima. Penfaktoran dilakukan untuk memperoleh private key. Selama penfaktoran bilangan besar menjadi bilangan prima belum tentu menemukan algoritma yang benar, maka selama itu pula keamanan algoritma RSA terjamin [1].

Algoritma RSA memiliki besaran seperti berikut :

1. p dan q bilangan prima (rahasia)
2. n = p.q (tidak rahasia)
3. $\Phi(n) = (p-1)(q-1)$ (rahasia)
4. e = kunci enkripsi (tidak rahasia)
5. d = kunci dekripsi (rahasia)
6. m = *plaintext* (rahasia)
7. c = chipertext (tidak rahasia)

2.3.1 Perumusan Algoritma RSA

Algoritma RSA didasarkan pada teorema Euler yang menyatakan bahwa

$$\mathbf{a^{\Phi(n)} = 1 \bmod n} \qquad \mathbf{(I)}$$

dengan syarat :

1. a harus relatif prima terhadap n
2. $\Phi(n) = n(1-1/p_1)\ \ n(1-1/p_2) \ldots n(1-1/p_r)$ dalam hal ini $p_1, p_2, \ldots p_r$ adalah faktor prima dari n. $\Phi(n)$ adalah fungsi yang menentukan berapa banyak dari bilangan – bilangan 1,2,3,..n yang relatif prima terhadap n.

Berdasarkan sifat $a^k = a^b \pmod n$ untuk k bilangan bulat ≥ 1, maka persamaan (I) dapat ditulis menjadi

$$\mathbf{a^{\Phi k(n)} = 1^k \pmod n}$$

atau

$$\mathbf{a^{\Phi k(n)} = 1 \pmod n} \qquad \textbf{(II)}$$

Bila a diganti m,maka persamaan (II) dapat ditulis menjadi

$$\mathbf{m^{\Phi k(n)} = 1 \pmod n} \qquad \textbf{(III)}$$

Berdasarkan sifat ac = bc (mod n) maka bila persamaan (III) dikali dengan m, maka menjadi

$$\mathbf{m^{\Phi k(n)+1} = m \pmod n} \qquad \textbf{(IV)}$$

yang dalam hal ini m relatif prima dengan n.

Misalkan e dan d dipilih sedemikian sehingga

$$\mathbf{e \,.\, d = 1 \pmod{\Phi(n)}} \qquad \textbf{(V)}$$

atau

$$\mathbf{e \,.\, d = k\Phi(n) + 1} \qquad \textbf{(V)}$$

Masukkan (V) ke dalam persamaan (VI) menjadi

$$\mathbf{me \,.\, d = m \pmod n} \qquad \textbf{(VI)}$$

Persamaan (VI) dapat ditulis kembali menjadi

$$\mathbf{(m^e)^d = m \pmod n} \qquad \textbf{(VII)}$$

Yang artinya perpangkatan m dengan e diikuti dengan perpangkatan dengan d yang menghasilkan kembali m semula. Berdasarkan persamaan (VII), maka enkripsi dan dekripsi dirumuskan sebagai berikut

$$\mathbf{E_e\,(m) = c = m^e \bmod n} \qquad \mathbf{(VIII)}$$

$$\mathbf{D_d(c) = m = c^d \bmod n} \qquad \mathbf{(IX)}$$

Karena e . d = d . e maka enkripsi dilakukan dengan dekripsi ekivalen dengan dekripsi diikuti enkripsi

$$\mathbf{D_d(E_e(m)) = E_e\,(D_d(m)) = m^d \bmod n} \qquad \mathbf{(X)}$$

Oleh karena $m^d \bmod n = (m + jn)^d \bmod n$ untuk sembarang bilangan bulat j, maka setiap *plaintext* m, m + n, m + 2n, …, menghasilkan *chipertext* yang sama, Dengan kata lain, transformasinya dari banyak ke satu, maka m harus dibatasi dalam himpunan {0, 1, 2, … n-1} sehingga enkripsi dan dekripsi tetap benar seperti persamaan (VIII) dan (IX) [1].

2.4 Teknologi *Mobile*

Tingginya mobilitas masyarakat telah membantu mengembangkan teknologi mobile *(mobile technology)* di dunia ini. Banyak aktivitas/pekerjaan yang perlu dilakukan dan sedikitnya waktu yang tersedia membuat orang-orang ingin tetap berhubungan dengan informasi dimanapun dan kapanpun. *Mobile Technology* merupakan inovasi yang muncul sebagai jawaban atas kebutuhan-kebutuhan tersebut. Kini *mobile technology* memiliki karakteristik yang mencerminkan kemampuan-kemampuan seperti *ubiquity, convenience, instant connection,* and *personalization.*

Ubiquity berarti dimana saja, kapan saja, sehingga dapat memenuhi kebutuhan untuk *real time information. Convenience* adalah mudah dan cepat untuk digunakan. *Instant connection* berarti bisa melakukan koneksi dengan internet atau

alat-alat *mobile* lainnya dengan cepat dan mudah, sedangkan *personalization* berarti bisa memberikan *costumized information* sesuai karakteristik pribadi seseorang.

Kemunculan teknologi *mobile* ini, selain didorong oleh kebutuhan juga didorong oleh faktor-faktor lainnya, antara lain : semakin banyaknya jenis *mobile devices* yang tersedia, *Vendor/Market Push*, semakin murahnya harga *mobile technology*, semakin banyak fungsi *mobile* yang ditawarkan semakin baik dan luasnya jaringan yang tersedia. Kini telash tersedia perangkat teknlogi *mobile* seperty laptop, PDA, dan ponsel.

Ponsel pada awalnya muncul dari kebutuhan untuk berkomunikasi dimana saja dan kapan saja. Seiring dengan perkembangan waktu, teknologi komunikasi ini mulai diintegrasikan dengan berbagai macam teknologi dan fungsi lainnya, seperti kamera dan internet [4].

2.5 Global System For Mobile (GSM)

GSM (*Global system for Mobile*) adalah generasi kedua dari standar system selular. Sistem selular yang tengah dikembangkan untuk mengatasi problem fragmentasi yang terjadi pada standar pertama di negara Eropa. GSM adalah sistem standar selular pertama didunia yang menspesifikasikan *digital modulation* dan *network level architectures and service*. Sebelum muncul standar GSM ini negara-negara di Eropa menggunakan standar yang berbeda-beda, sehingga pada saat itu tidak memungkinkan seorang pelanggan menggunakan *single subscriber* unit untuk menjangkau seluruh benua Eropa.

Pada awalnya sistem GSM ini dikembangkan untuk melayani sistem seluler di Eropa dan menjanjikan jangkauan network yang lebih luas seperti halnya penggunaan ISDN. Pada perkembangaannya sistem GSM ini mengalami kemajuan pesat dan menjadi standar yang paling populer di seluruh dunia untuk sistem seluler. Bahkan pertumbuhannya diprediksikan akan mencapai 20 sampai 50 juta pelanggan pada tahun 2000.

Penggunaan alokasi frekuensi 900 MHz oleh GSM ini diambil berdasarkan rekomendasi GSM *(Gropue special Mobile*) *comitte* yang merupakan salah satu grup kerja pada *confe'rence Europe'ene Postes des Telecommunication* (CEPT). Namun pada akhirnya untuk alasan marketing GSM berubah namanya menjadi *yhe Global System for Mobile Communication,* sedangkan standar teknisnya diambil dari *European Technical Standards Institute* (ETSI).

GSM pertama kali diperkenalakan di Eropa pada tahun 1991 kemudian pada akhir 1993, beberapa negara non Amerika seperti Amerika Selatan, Asia dan Australia mulai mengadopsi GSM yang akhirnya menghasilkan standar baru yang mirip yaitu DCS 1800, yang mendukung PCS (*Personal Communication Service*) pada frekuensi 1,8 GHz sampai 2 GHz [4].

2.5.1 Definisi GSM

Global System for Mobile Communication (GSM) adalah sebuah teknologi komunikasi selular yang bersifat digital. Teknologi GSM banyak diterapkan pada komunikasi bergerak, khususnya telepon genggam. Teknologi ini memanfaatkan gelombang mikro dan pengiriman sinyal yang dibagi berdasarkan waktu, sehingga sinyal informasi yang dikirim akan sampai pada tujuan. GSM dijadikan standar global untuk komunikasi selular sekaligus sebagai teknologi selular yang paling banyak digunakan orang di seluruh dunia [4].

2.5.2 Sejarah Perkembangan GSM

Teknologi komunikasi selular sebenarnya sudah berkembang dan banyak digunakan pada awal tahun 1980-an, diantaranya sistem C-NET yang dikembangkan di Jerman dan Portugal oleh Siemens, sistem RC-2000 yang dikembangkan di Perancis, sistem NMT yang dikembangkan di Belanda dan Skandinavia oleh Ericsson, serta sistem TACS yang beroperasi di Inggris. Namun teknologinya yang masih analog membuat sistem yang digunakan bersifat regional sehingga sistem antara negara satu dengan yang lain tidak saling kompatibel dan menyebabkan mobilitas pengguna terbatas pada suatu area sistem teknologi tertentu saja (tidak bisa melakukan roaming antar negara).

GSM muncul pada pertengahan 1991 dan akhirnya dijadikan standar telekomunikasi selular untuk seluruh Eropa oleh ETSI *(European Telecomunication Standard Institute)*. Pengoperasian GSM secara komersil baru dapat dimulai pada awal kuartal terakhir 1992 karena GSM merupakan teknologi yang kompleks dan butuh pengkajian yang mendalam untuk bisa dijadikan standar. Pada September 1992, standar *type approval* untuk handphone disepakati dengan mempertimbangkan dan memasukkan puluhan item pengujian dalam memproduksi GSM. Pada awal pengoperasiannya, GSM telah mengantisipasi perkembangan jumlah penggunanya yang sangat pesat dan arah pelayanan per area yang tinggi, sehingga arah perkembangan teknologi GSM adalah DCS *(Digital Cellular System)* pada alokasi frekuensi 1800 MHz. Dengan frekuensi tersebut, akan dicapai kapasitas pelanggan yang semakin besar per satuan *cell*. Selain itu, dengan luas *cell* yang semakin kecil akan dapat menurunkan kekuatan daya pancar handphone, sehingga bahaya radiasi yang timbul terhadap organ kepala akan dapat di kurangi. Pemakaian GSM kemudian meluas ke Asia dan Amerika, termasuk indonesia. Indonesia awalnya menggunakan sistem telepon selular analog yang bernama AMPS *(Advances Mobile Phone System)* dan NMT *(Nordic Mobile Telephone)*. Namun dengan hadir dan dijadikannnya standar sistem komunikasi selular membuat sistem analog perlahan menghilang, tidak hanya di Indonesia, tapi juga di Eropa. Pengguna GSM pun semakin lama semakin bertambah. Pada akhir tahun 2005, pelanggan GSM di dunia telah mencapai 1,5 triliun pelanggan. Akhirnya GSM tumbuh dan berkembang sebagai sistem telekomunikasi seluler yang paling banyak digunakan di seluruh dunia [4].

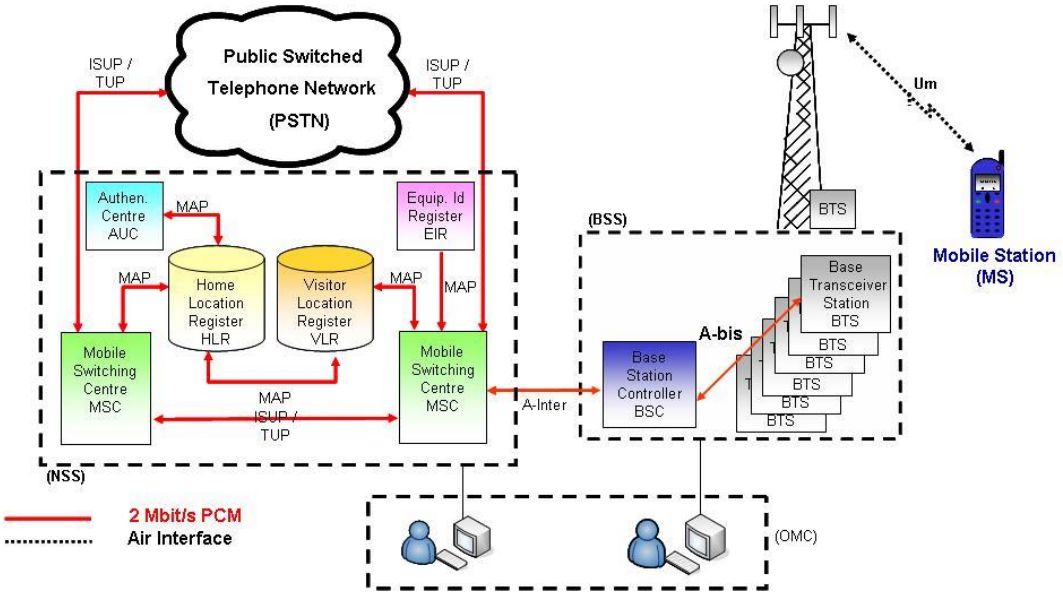
2.5.3 Spesifikasi Teknis GSM

Di Eropa, pada awalnya GSM didesain untuk beroperasi pada frekuensi 900 MHz. Pada frekuensi ini, frekuensi *uplinks*-nya digunakan frekuensi 890–915 MHz, sedangkan frekuensi *downlink* nya menggunakan frekuensi 935–960 MHz. *Bandwith* yang digunakan adalah 25 MHz (915–890 = 960–935 = 25 MHz), dan lebar kanal sebesar 200 kHz. Dari keduanya, maka didapatkan 125 kanal, dimana 124 kanal digunakan untuk suara dan satu kanal untuk sinyal. Pada

perkembangannya, jumlah kanal 124 semakin tidak mencukupi dalam pemenuhan kebutuhan yang disebabkan pesatnya pertambahan jumlah pengguna. Untuk memenuhi kebutuhan kanal yang lebih banyak, maka regulator GSM di Eropa mencoba menggunakan tambahan frekuensi untuk GSM pada band frekuensi di *range* 1800 MHz dengan frekuensi 1710-1785 MHz sebagai frekuensi *uplinks* dan frekuensi 1805-1880 MHz sebagai frekuensi *downlinks*. GSM dengan frekuensinya yang baru ini kemudian dikenal dengan sebutan GSM 1800, yang menyediakan *bandwidth* sebesar 75 MHz (1880-1805 = 1785–1710 = 75 MHz). Dengan lebar kanal yang tetap sama yaitu 200 kHz sama, pada saat GSM pada frekuensi 900 MHz, maka pada GSM 1800 MHz ini akan tersedia sebanyak 375 kanal. Di Eropa, standar-standar GSM kemudian juga digunakan untuk komunikasi *railway*, yang kemudian dikenal dengan nama GSM-R [4].

2.5.4 Arsitektur Jaringan GSM

Teknologi GSM menggunakan sistem CDMA dengan alokasi kurang lebih sekitar delapan pengguna di dalam satu *channel* frekuensi sebesar 200 kHz per satuan waktu. Awalnya, frekuensi yang digunakan adalah 900 MHz. Pada perkembangannya frekuensi yang digunakan adalah 1800 MHz dan 1900 MHz. Kelebihan dari GSM adalah *interface* yang lebih bagi para *provider* maupun para penggunanya. Selain itu, kemampuan *roaming* antar sesama *provider* membuat pengguna dapat bebas berkomunikasi. Arsitektur jaringan GSM seperti ditunjukkan pada gambar 2.4.



Gambar 2.4 Arsitektur Jaringan GSM

Pada Gambar II.4 Arsitektur Jaringan GSM terdiri dari perangkat-perangkat yang saling mendukung, dari 4 subsistem yang terkoneksi dan berinteraksi antar sistem dan dengan *user* melalui *network interface*, 4 subsistem tersebut yaitu : MS (*Mobile Station*), BSS (*Base Station Subsystem*), NSS (*Network Sub-System*) dan OSS (*Operation and Support System*) [4].

2.5.5 Keunggulan GSM Sebagai Teknologi Generasi Kedua (2G)

GSM sebagai sistem telekomunikasi selular digital memiliki keunggulan yang jauh lebih banyak dibanding sistem analog, di antaranya:

1. Kapasitas sistem lebih besar, karena menggunakan teknologi digital dimana penggunaan sebuah kanal tidak hanya diperuntukkan bagi satu pengguna saja. Sehingga saat pengguna tidak mengirimkan informasi, kanal dapat digunakan oleh pengguna lain.
2. Sifatnya yang sebagai standar internasional memungkinkan international roaming.
3. Dengan teknologi digital, tidak hanya mengantarkan suara, tapi memungkinkan servis lain seperti teks, gambar, dan video.

4. Keamanan sistem yang lebih baik.
5. Kualitas suara lebih jernih dan peka.

Bagaimanapun, keunggulan GSM yang beragam pantas saja membuatnya menjadi sistem telekomunikasi selular terbesar penggunanya di seluruh dunia [4].

2.6 Short Message Service (SMS)

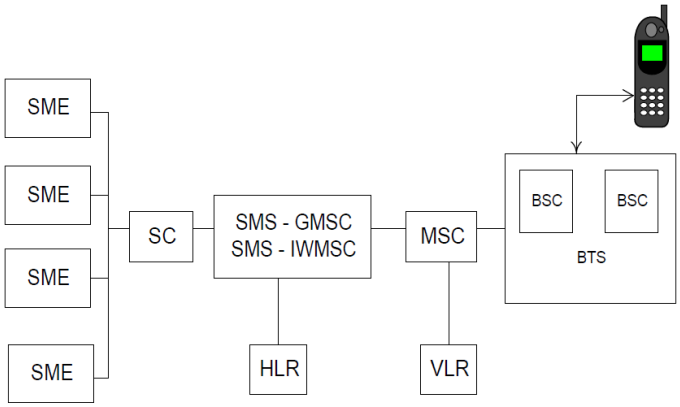
Short Message Sevice (SMS) adalah salah satu fasilitas dari teknologi GSM yang memungkinkan mengirim dan menerima pesan pesan singkat berupa text dari Mobile Station (MS). Layanan SMS juga memungkinkan pengiriman pesan dalam bentuk *alphanumeric,* layanan SMS ini banyak diaplikasikan pada sistem komunikasi tanpa kabel (*wireless).* Teknologi wireless dipelopori dari kawasan Eropa yang diawali pada kebutuhan bersama terhadap satu sistem jaringan baru yang dapat menjadi standard jaringan yang berlaku dan dapat diterapkan di seluruh kawasan Eropa. Dalam sistem baru juga harus terdapat kemampuan yang dapat mengantisipasi mobilitas pengguna serta kemampuan melayani lebih banyak pengguna untuk menampung penambahan jumlah pelanggan baru.

Karena hal ini tidak dapat dilakukan dengan mempertahankan sistem analog, maka kemudian diputuskan untuk merombak sistem dan menggantinya dengan sistem digital. Standard baru diperkenalkan dengan nama *Global Standard for Mobile Communications* (GSM). GSM pada awalnya adalah kepanjangan dari *Groupe Speciale Mobile*, sebuah badan gabungan dari para ahli yang melakukan studi bersama untuk menciptakan standard GSM tersebut.

Pada bulan Desember 1992, dilakukan pengiriman pesan menggunakan SMS dari sebuah *Personal Computer* (PC) ke telepon *mobile* (bergerak) dalam jaringan GSM milik Vodafone Inggris, kemudian merambah ke benua Amerika yang dipelopori oleh beberapa operator komunikasi bergerak berbasis digital seperti BellSouth Mobility, PrimeCo dan operator lainnya. Teknologi yang digunakan dari pengiriman SMS yaitu *Store* and *forward service*, jadi SMS yang di kirim akan simpan sementara di server SMS center kemudian dialihkan ke nomor tujuan [4].

2.6.1 Arsitektir Jaringan SMS

Layanan SMS dibangun dari berbagai entitas yang saling terkait dan mempunyai fungsi dan tugas masing-masing. Tidak ada satupun dalam sistem SMS yang dapat bekerja secara parsial. Entitas dalam jaringan SMS ini disebut juga elemen SMS. Di bawah ini merupakan arsitektur SMS dengan beberapa elemen-elemen yang saling terkait [4]:



Gambar 2.5 Arsitektur SMS

Elemen-elemen dasar pada jaringan SMS :

1. *SME (Short Message Entity)*, merupakan tempat penyimpanan dan pengiriman message yang akan dikirimkan ke MS tertentu.
2. *SC (Service Centre)*, bertugas untuk menerima message dari SME dan melakukan forwarding ke alamat MS yang dituju.
3. *SMS-GMSC (Short Message Service Gateway SMC*), melakukan penerimaan message dari SC dan memeriksa parameter yang ada. Selain itu GMSC juga mencari alamat MS yang dituju dangan bantuan HLR, dan mengirimkannya kembali ke MSC yang dimaksud.
4. *SMS IWMSC (Short Message Service Interworking MSC*), berperan dalam SMS Message Origiating, yaitu menerima pesan dari MSC
5. *Home Location Register (HLR)* merupakan sebuah database yang digunakan sebagai tempat penyimpanan permanen data dan profil

pelanggan. Bila diminta oleh SMSC, maka HLR dapat memberikan informasi routing dari pelanggan tertentu. HLR juga dapat memberikan informasi status tujuan apakah aktif atau tidak.

6. *Visitor Location Register (VLR*) merupakan sebuah database tempat menyimpan informasi sementara yang berisi data pelanggan dari sebuah HLR yang roaming pada HLR lain.
7. *MSC* merupakan sebuah sistem yang melakukan fungsi switching dan mengontrol panggilan telepon dalam sebuah jaringan komunikasi bergerak. MSC inilah yang akan mengirimkan sebuah *short message* ke suatu tujuan tertentu melalui base station yang sesuai.
8. *Base Station Sistem (BSS)* Merupakan kesatuan sistem yang bertanggung jawab mengatur transmisi sinyal elektromagnetik untuk membawa data dari MSC ke perangkat telepon bergerak. Base Station terdiri dari Base Station Controler (BSC) dan Base Tranceiver Station (BTS) dan juga dikenal dengan nama *cell cite* atau sederhananya cell. Sebuah BSC biasanya menangani satu atau lebih BTS dan bertanggung jawab menangani pelanggan saat berpindah dari satu cell ke cell lainnya.
9. *Mobile Device* merupakan perangkat yang mempunyai kemampuan mengirimkan dan menerima short message, biasanya berupa telepon seluler dengan teknologi digital. Akan tetapi, saat ini jenis terminal berkembang sesuai aplikasi dan kebutuhan seperti POS, laptop dan Personal Digital Assistant (PDA).

2.6.2 Layanan Aplikasi SMS

Layanan aplikasi SMS merupakan sebuah layanan yang bersifat *none real time* dimana sebuah *short message* dapat di submit ke suatu tujuan, tidak peduli apakah tujuan tersebut aktif atau tidak. Bila dideteksi bahwa tujuan tidak aktif, maka sistem akan menunda pengiriman ke tujuan hingga tujuan aktif kembali. Pada dasarnya sistem SMS akan menjamin *delivery* dari suatu *short message* hingga sampai tujuan. Kegagalan pengiriman yang bersifat sementara seperti tujuan tidak

aktif akan selalu teridentifikasi sehingga pengiriman ulang *short message* akan selalu dilakukan kecuali apabila diberlakukuan aturan bahwa *short message* yang telah melampaui batas waktu tertentu harus dihapus dan dinyatakan gagal kirim. Berdasarkan mekanisme distribusi pesan SMS diperoleh aplikasi sms, terdapat 4 macam mekanisme pengantaran pesan yaitu :

1. *Pull*, yaitu pesan yang dikirimkan ke pengguna berdasarkan permintaan pengguna.
2. *Push Event based*, yaitu pesan yang diaktivasi oleh aplikasi berdasarkan kejadian yang berlangsung
3. *Push - Schedule*, yaitu pesan yang diaktivasi oleh aplikasi berdasarkan waktu yang telah terjadwal.
4. *Push Personal Profile*, yaitu pesan yang diaktivasi oleh aplikasi berdasarkan profile dan preference dari pengguna.

2.7 Android

Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh berbagai macam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat piranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Pada saat perilisan perdana android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode–kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler. Di dunia ini terdapat dua jenis distributor sistem operasi Android, pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD).

Pada Juli 2005, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc. bekerja pada Google, di antaranya Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi Android Inc. hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler.

Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan *Open Handset Alliance*, OHA mengumumkan produk perdana mereka, Android, perangkat bergerak (*mobile*) yang merupakan modifikasi kernel Linux 2.6 [9].

2.7.1 Versi Android

Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan *bug* dan penambahan fitur baru. Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android. Berikut versi-versi android yang penah dirilis [10]:

2.7.1.1 Android Versi 1.1

Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2.7.1.2 Android Versi 1.5 (Cupcake)

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (*Software Development Kit*) dengan versi 1.5 (Cupcake). Terdapat beberapa pembaruan termasuk juga penambahan beberapa

fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan *Bluetooth* A2DP, kemampuan terhubung secara otomatis ke *headset Bluetooth*, animasi layar, dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

2.7.1.3 Android Versi 1.6 (Donut)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus, *camcorder* dan galeri yang dintegrasikan; CDMA / EVDO, 802.1x, VPN, *Gestures*, dan *Text-to-speech engine*; kemampuan dial kontak; teknologi *text to change speech* (tidak tersedia pada semua ponsel; pengadaan resolusi VWGA.

2.7.1.4 Android Versi 2.0/2.1 (Éclair)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah pengoptimalan *hardware*, peningkatan Google Maps 3.1.2, perubahan UI dengan *browser* baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1. Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi *mobile* terbaik (*killer apps* - aplikasi unggulan).Kompetisi ini berhadiah $25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik.Dengan semakin berkembangnya dan semakin bertambahnya jumlah *handset* Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi Android.Aplikasi terkenal yang diubah ke dalam sistem operasi Android adalah Shazam, Backgrounds, dan WeatherBug.Sistem operasi Android dalam situs internet juga dianggap penting untuk menciptakan aplikasi Android asli, contohnya oleh MySpace dan Facebook.

2.7.1.5 Android Versi 2.2 (Froyo : Frozen Yoghurt)

Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, intergrasi V8 JavaScript*engine* yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi *Hotspot portable*, dan kemampuan *auto update* dalam aplikasi *Android Market*.

2.7.1.6 Android Versi 2.3 (Gingerbread)

Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (*gaming*), peningkatan fungsi *copy paste*, layar antar muka (*User Interface*) didesain ulang, dukungan *formatvideo* VP8 dan WebM, efek audio baru (*reverb, equalization, headphone virtualization*, dan *bass boost*), dukungan kemampuan *Near Field Communication* (NFC), dan dukungan jumlah kamera yang lebih dari satu.

2.7.1.7 Android Versi 3.0/3.1 (Honeycomb)

Android Honeycomb dirancang khusus untuk *tablet*. Android versi ini mendukung ukuran layar yang lebih besar.*User Interface* pada Honeycomb juga berbeda karena sudah didesain untuk *tablet.* Honeycomb juga mendukung *multi prosesor* dan juga akselerasi perangkat keras (*hardware*) untuk grafis. *Tablet* pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom.

2.7.1.8 Android Versi 4.0 (ICS : Ice Cream Sandwitch)

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb untuk *smartphone* dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara *offline*, dan berbagi informasi dengan menggunakan NFC.

2.7.1.9 Android Versi 4.1/4.2 (Jelly Bean)

Android 4.1 Jelly Bean diumumkan pada 27 Juni 2012 pada konferensi Google l/O. Versi ini adalah yang tercepat dan terhalus dari semua versi Android. Jelly Bean 4.1 meningkatkan kemudahan dan keindahan tampilan dari Ice Cream Sandwich dan memperkenalkan pengalaman pencarian Google yang baru di Android. Android 4.2 *Jelly Bean* diumumkan pada 29 Oktober 2012, versi ini menawarkan peningkatkan kecepatan dan kemudahan Android 4.1 serta mencakup semua fitur baru seperti *Photo Sphere* dan desain baru aplikasi kamera, *keyboard Gesture Typing*, *Google Now* dan lainnya.

2.8 SQLite

2.8.1 Penjelasan Umum

SQLite merupakan sebuah system manajemen basis data relasional yang bersifat ACID - compliant dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. SQLite merupakan proyek yang bersifat public domain yang dikerjakan oleh D. Richard Hipp. Tidak seperti pada paradigma *client-server* umumnya, Inti SQLite bukanlah sebuah sistem yang mandiri yang berkomunikasi dengan sebuah program, melainkan sebagai bagian integral dari sebuah program secara keseluruhan. Sehingga protokol komunikasi utama yang digunakan adalah melalui pemanggilan API secara langsung melalui bahasa pemrograman. Mekanisme seperti ini tentunya membawa keuntungan karena dapat mereduksi *overhead*, *latency times*, dan secara keseluruhan lebih sederhana. Seluruh elemen basisdata (definisi data, tabel, indeks, dan data) disimpan sebagai sebuah file. Kesederhanaan dari sisi disain tersebut bisa diraih dengan cara mengunci keseluruhan file basis data pada saat sebuah transaksi dimulai [6].

2.8.2 SQLite di Android

SQLite adalah *Database Open Source* yang tertanam ke Android. SQLite mendukung fitur *database relasional* standar seperti sintaks SQL, *transactions* dan *prepared statements*. Selain itu hanya memerlukan sedikit memori pada saat runtime (sekitar 250 KByte).

Menggunakan SQLite di Android tidak memerlukan setup *database* atau administrasi. Kita menentukan SQL untuk bekerja dengan database dan database secara otomatis dikelola untuk kita. Bekerja dengan database di Android bisa lambat karena diperlukan I / O. Oleh karena itu dianjurkan untuk melakukan work ini dalam sebuah *AsyncTask*.

SQLite mendukung tipe data TEXT (mirip dengan String di Java), INTEGER (mirip dengan yang lama di Java) dan REAL (mirip dengan ganda di Java). Semua jenis lain harus dikonversi ke pada bidang ini sebelum menyimpannya dalam database. SQLite sendiri tidak memvalidasi apakah jenis ditulis ke kolom sebenarnya dari jenis didefinisikan, Kita dapat menulis sebuah integer ke dalam kolom string.

Jika aplikasi Anda menciptakan database database ini disimpan di direktori "DATA / data / APP_NAME / database / FILENAME". "DATA" adalah jalan yang Environment.getDataDirectory () mengembalikan, "APP_NAME" adalah nama aplikasi Anda dan "FILENAME" adalah nama yang Anda berikan pada saat pembuatan database. Environment.getDataDirectory () biasanya kembali SD card sebagai lokasi.Database SQLite adalah swasta untuk aplikasi yang menciptakan itu. Jika Anda ingin berbagi data dengan aplikasi lain Anda dapat menggunakan *Content Provider* [6].

2.9 Eclipse

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

1. *Multi-platform:* Target sistem operasi Eclipse adalah *Microsoft Windows, Linux, Solaris, AIX, HP-UX* dan *Mac OS X*.
2. *Multi-language*: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis

bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.

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

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan open source, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

Konsep Eclipse adalah IDE yang terbuka *(open),* mudah diperluas *(extensible)* untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi, Eclipse tidak saja untuk mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal *plug-in* yang dibutuhkan. Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh Eclipse, *plug-in* UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat *plug-in* sesuai dengan keinginannya [9].

2.10 Unfield Modeling Language (UML)

UML menurut Booch (1999: 14) dalam Jurnal Gintoro, Andreyus, Emilia dan Richard William (2010 , Hal : B-30) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk menvisualisasi, menspesifikasi, merancang dan mendokumentasi sistem piranti lunak [13].

Unified Modelling Language (UML) menawarkan sebuah standar untuk merancang model sebuah sistem. Tujuan UML adalah :

1. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.

2. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrogramandan proses rekayasa. Menyatukan praktik-praktik terbaik yang terdapat dalam pemodelan.
3. Mentatukan praktik-praktik terbaik yang terdapat dalam pemodelan.

UML menyediakan beberapa notasi dan *artifact standard* yang bisa digunakan sebagai alat komunikasi bagi para pelaku dalam proses analisis dan desain. *Artifact* didalam UML didefinisikan sebagai informasi dalam bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat. Contohnya adalah *source code* yang dihasilkan oleh proses pemrograman [13].

2.10.1 Diagram UML

Terdapat beberapa diagram UML, diantaranya sebagai berikut [13] :

2.10.1.1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Use Case dilakukan oleh satu actor yang digambarkan dengan simbol orang yang dihubungkan dengan garis yang menunjukan hubungan komunikasi Setiap Use Case harus diberi nama yang menyatakan apa hal yang dicapai dari hasil interaksi dengan Actor. Nama Use Case boleh terdiri dari beberapa kata dan tidak boleh ada Use Case yang memiliki nama yang sama.

2.10.1.2 Class Diagram

Diagram kelas menggambarkan tipe-tipe objek dalam *system* dan berbagai jenis hubungan atau relasi statis yang ada diantara mereka. Diagram ini memberikan gambaran umum dari *system*. Seperti tipe-tipe dari objek dengan menunjukan kelasnya dan *relationship* yang diantara mereka, serat penjelasan detail tiap-tiap kelas ke dalam model suatu *system*. *Class diagram* bersifat *static* (tidak berubah) yang akan menunjukan apa itu interaksi tapi tidak menjelaskan apa yang terjadi ketika mereka melakukan interaksi.

2.10.1.3 Activity Diagram

Activity diagram memodelkan alur kerja *(work flow)*. Sebuah proses bisnis dan urutan aktifitas dalam suatu proses. Diagram ini sangat mirip dengan sebuah *flowchart* karena kita dapat memodelkan sebuah alur kerja dari aktifitas keaktifitas lainnya atau dari suatu aktifitas kedalam keadaan sesaat *(state).*

2.10.1.4 Sequence Diagram

Sequence diagram merupakan gambaran interaksi antar objek di dalam dan di sekitar sistem berupa *message* yang digambarkan terhadap waktu. Diagram ini secara khusus berasosiasi dengan *Use Case*. *Sequence Diagram* juga digunakan untuk menggambarkan skenario atau rangkain langkah-langkah apa yang seharusnya terjadi sebagai respons dari sebuah event untuk menghasilkan sesuatu didalam *Use Case* sebagai *output*. Untuk *message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* dipetakan menjadi operasi/metode dari *class. Activation bar* menunjukan lamanya eksekusi sebuat proses, biasanya diawali dengan sebuah *message*.

2.11 XML

XML (*Extensible Markup Language*) adalah bahasa markup untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen markup keperluan pertukaran data antar sistem yang beraneka ragam.XML merupakan kelanjutan dari HTML (*HyperText Markup Language*) yang merupakan bahasa standar untuk melacak Internet.

XML didesain untuk mempu menyimpan data secara ringkas dan mudah diatur.Kata kunci utama XML adalah data (jamak dari datum) yang jika diolah bisa memberikan informasi. XML menyediakan suatu cara terstandarisasi namun bisa dimodifikasi untuk menggambarkan isi dari dokumen. Dengan sendirinya, XML dapat digunakan untuk menggambarkan sembarang *view database*, tetapi dengan suatu cara yang standar [3].

2.11.1 Tipe XML

XML memiliki tiga tipe, diantaranya [3]:

1. XML, merupakan standar format dari struktur berkas (*file*).
2. XSL, merupakan standar untuk memodifikasi data yang diimpor atau diekspor.
3. XSD, merupakan standar yang mendefinisikan struktur *database* dalam XML.

2.11.2 Keunggulan XML

Keunggulan XML bisa diringkas sebagai berikut [3] :

1. Pintar (*Intelligence*). XML dapat menangani berbagai tingkat (*level*) kompleksitas.
2. Dpat beradaptasi. Dapat mengadaptasi untuk membuat bahasa sendiri. Seperti Microsoft membuat bahasa MSXML atau Macromdia mengembangkan MXML.
3. Mudah pemeliharaannya.
4. Sederhana. XML lebih sederhana.
5. Mudah dipindah-pindahkan (*Portability*). XML mempunyai kemudahan perpindahan (*Portabilitas)* yang lebih bagus.

2.12 Java

Java adalah bahasa pemrograman berorientasi objek yang dikembangkan oleh Sun Microsystems sejak tahun 1991. Bahasa ini dikembangkan dengan model yang mirip dengan bahasa C++ dan *Smalltalk*, namun dirancang agar lebih mudah dipakai dan *platform independent*, yaitu dapat dijalankan di berbagai jenis sistem operasi dan arsitektur komputer [11].

2.12.1 Sejarah Java

Bahasa pemrograman Java terlakhir dari The Green Project, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut

belum menggunakan versi yang dinamakan Oak.Proyek ini dimotori oleh Patrick Naughton, Mike Sheridan, James Gosling dan Bill Joy, beserta sembilan pemrogram lainnya dari Sun Microsystems. Salah satu hasil proyek ini adalah maskot Dukeyang dibuat oleh Joe Palrang.

Pertemuan proyek berlangsung di sebuah gedung perkantoran Sand Hill Road di Menlo Park. Sekitar musim panas 1992 proyek ini ditutup dengan menghasilkan sebuah program Java Oak pertama, yang ditujukan sebagai pengendali sebuah peralatan dengan teknologi layar sentuh (*touch screen*), seperti pada PDA sekarang ini. Teknologi baru ini dinamai "*7" (*Star Seven*).

Setelah era *Star Seven* selesai, sebuah anak perusahaan Tv kabel tertarik ditambah beberapa orang dari proyek The Green Project. Mereka memusatkan kegiatannya pada sebuah ruangan kantor di 100 Hamilton Avenue, Palo Alto. Perusahaan baru ini bertambah maju: jumlah karyawan meningkat dalam waktu singkat dari 13 menjadi 70 orang. Pada rentang waktu ini juga ditetapkan pemakaian Internet sebagai medium yang menjembatani kerja dan ide di antara mereka.Pada awal tahun 1990-an, Internet masih merupakan rintisan, yang dipakai hanya di kalangan akademisi dan militer.

Mereka menjadikan perambah (*browser*) Mosaic sebagai landasan awal untuk membuat perambah Java pertama yang dinamai Web Runner, terinsipirasi dari film 1980-an, Blade Runner. Pada perkembangan rilis pertama, Web Runner berganti nama menjadi Hot Java.

Pada sekitar bulan Maret1995, untuk pertama kali kode sumber Java versi 1.0a2 dibuka. Kesuksesan mereka diikuti dengan untuk pemberitaan pertama kali pada surat kabar *San Jose Mercury News* pada tanggal 23 Mei1995. Sayang terjadi perpecahan di antara mereka suatu hari pada pukul 04.00 di sebuah ruangan hotel Sheraton Palace.Tiga dari pimpinan utama proyek, Eric Schmidt dan George Paolini dari Sun Microsystems bersama Marc Andreessen, membentuk Netscape.

Nama Oak, diambil dari pohon oak yang tumbuh di depan jendela ruangan kerja "Bapak Java", James Gosling. Nama Oak ini tidak dipakai untuk versi release Java karena sebuah perangkat lunak lain sudah terdaftar dengan merek dagang tersebut, sehingga diambil nama penggantinya menjadi "Java". Nama ini diambil

dari kopi murni yang digiling langsung dari biji (kopi tubruk) kesukaan Gosling. Konon kopi ini berasal dari Pulau Jawa. Jadi nama bahasa pemrograman Java tidak lain berasal dari kata Jawa (bahasa Inggris untuk Jawa adalah Java) [11].

2.12.2 Versi Awal Java

Versi awal Java ditahun 1996 sudah merupakan versi *release* sehingga dinamakan Java Versi 1.0. Java versi ini menyertakan banyak paket standar awal yang terus dikembangkan pada versi selanjutnya [11]:

1. java.lang: Peruntukan kelas elemen-elemen dasar.
2. java.io: Peruntukan kelas *input* dan *output*, termasuk penggunaan berkas.
3. java.util: Peruntukan kelas pelengkap seperti kelas struktur data dan kelas-kelas penanggalan.
4. java.net: Peruntukan kelas TCP/IP, yang memungkinkan berkomunikasi dengan kompiter lain menggunakan jaringan TCP/IP.
5. java.awt: Kelas dasar untuk aplikasi antarmuka dengan pengguna (GUI).
6. java.applet: Kelas dasar aplikasi antarmuka untuk diterapkan pada penjelajahan web.

2.12.3 Kelebihan Java

Berikut kelebihan-kelebihan Java [11]:

1. *Multiplatform*. Kelebihan utama dari Java ialah dapat dijalankan di beberapa *platform* / sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja. Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin / *bytecode*) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis java dikerjakan diatas *operatingsystem* Linux tetapi dijalankan dengan baik di atas Microsoft Windows. *Platform* yang didukung sampai saat ini adalah Microsoft Windows, Linux, Mac OS dan Sun Solaris. Penyebanya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang

dapat diunduh dari situs Java) untuk meninterpretasikan *bytecode* tersebut.

2. OOP (*Object Oriented Programming* - Pemrogram Berorientasi Objek).
3. Perpustakaan kelas yang lengkap, Java terkenal dengan kelengkapan *library*/perpustakaan (kumpulan program program yang disertakan dalam pemrograman Java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas Java yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.
4. Bergaya C++, memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak pemrogram C++ untuk pindah ke Java. Saat ini pengguna Java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke Java. Universitas-universitas di Amerika Serikat juga mulai berpindah dengan mengajarkan Java kepada murid-murid yang baru karena lebih mudah dipahami oleh murid dan dapat berguna juga bagi mereka yang bukan mengambil jurusan komputer.
5. Pengumpulan sampah otomatis, memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas).

2.12.4 Kekurangan Java

Berikut kekurangan-kekurangan Java [11]:

1. Tulis sekali, jalankan di mana saja - Masih ada beberapa hal yang tidak kompatibel antara *platform* satu dengan *platform* lain. Untuk J2SE, misalnya *SWT-AWT bridge* yang sampai sekarang tidak berfungsi pada Mac OS X.
2. Mudah didekompilasi. Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber. Ini dimungkinkan karena kode jadi Java merupakan *bytecode* yang menyimpan banyak atribut bahasa

tingkat tinggi, seperti nama-nama kelas, metode, dan tipe data. Hal yang sama juga terjadi pada Microsoft .NET *Platform*. Dengan demikian, algoritma yang digunakan program akan lebih sulit disembunyikan dan mudah dibajak/di-*reverse-engineer*.

3. Penggunaan memori yang banyak. Penggunaan memori untuk program berbasis Java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal (lebih spesifik lagi, Delphi dan Object Pascal). Biasanya ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena trend memori terpasang makin murah), tetapi menjadi masalah bagi mereka yang masih harus berkutat dengan mesin komputer berumur lebih dari 4 tahun.

2.12.5 Tahap Kompilasi Java

Berikut tahap kompilasi Java [5]:

1. Tulis / Ubah. Pemrogram menulis program dan menyimpannya di media dalam bentuk berkas '.java'.
2. Kompilasi. Pengkompilasi membentuk *bytecodes* dari program menjadi bentuk berkas '.class'.
3. Muat. Pemuat kelas memuat *bytecodes* ke memori.
4. Verifikasi. Peng-verifikasi memastikan *bytecodes* tidak mengganggu sistem keamanan Java.
5. Jalankan. Penerjemah menerjemahkan *bytecodes* ke bahasa mesin.