BAB II
TINJAUAN PUSTAKA

2. 1. Profil Perusahaan

2. 1. 1. Sejarah Perusahaan

Era Teknologi informasi saat ini telah mengubah tatanan hidup manusia dalam berkomunikasi dan bersosialisasi. Teknologi informasi memungkinkan adanya komunikasi tanpa adanya batas waktu dan tempat. Pesatnya perkembangan teknologi informasi mendorong manusia untuk mengubah gaya hidupnya, sehingga manusia senantiasa berinovasi menciptakan sesuatu yang baru tidak hanya untuk memenuhi kebutuhan hidupnya saat ini, melainkan menciptakan era baru yang "memaksa" manusia untuk berada pada era dimana teknologi menjadi kekuatan besar dalam "*change of life style*". Pesatnya perkembangan dunia teknologi informasi ini, bukan hanya dalam hitungan hari melainkan lebih cepat dari itu.

Teknologi *Mobile* adalah salah satu contoh teknologi yang paling cepat perkembangannya diantara pesatnya perkembangan teknologi Internet. Perkembangan kedua teknologi ini harus tetap diimbangi dengan ilmu dan pengetahuan yang cukup pada sisi pengguna, supaya tujuan positif dari penggunaan teknologi informasi bisa tercapai dan dirasakan oleh semua pihak yang bersangkutan sesuai dengan tujuan awal yaitu memberikan banyak manfaat untuk kebutuhan hidup manusia.

Indonesia saat ini menjadi salah satu negara dengan pengguna teknologi informasi terbesar di dunia. Sebut saja *Facebook* dan *Twitter* dalam bidang jejaring sosial, Indonesia disebutkan sebagai peringkat 3 terbesar dalam kategori pengguna dan diprediksikan akan terus meningkat. Selain itu Indonesia juga tercatat sebagai pengguna Blackberry dan pengunduh *software* aplikasi terbanyak untuk beberapa aplikasi. Dan tentu saja masih banyak prestasi Indonesia dari sisi konsumsi produk teknologi di kancah global.

Apabila kita berbicara mengenai teknologi *mobile*, maka sudah barang tentu kita bisa membayangkan seberapa pesat dan tingginya kebutuhan teknologi *mobile* saat ini. Bukan hanya tenar di kalangan dewasa, penggunaan *mobile* di kalangan anak-anak pun sudah menjadi hal yang lumrah. Telekomunikasi *mobile* sudah selayaknya mendapatkan perhatian yang lebih dari semua kalangan agar penggunaan telekomunikasi *mobile* menjadi tepat sasaran dan tidak melenceng dari tujuan.

Berlatar dari kondisi inilah, paseban.com hadir sebagai solusi edukasi bagi para pengguna teknologi Informasi di Indonesia khususnya pengguna *mobile*. Dengan kata lain, paseban.com mempunyai visi untuk memberikan informasi yang bersifat edukasi seakurat mungkin kepada para pengguna *mobile* di Indonesia. Paseban.com adalah *website* portal yang berbasis komunitas dan mengupas tuntas secara *detail* mengenai perkembangan teknologi *mobile* saat ini, dengan harapan dapat menciptakan masyarakat *mobile* indonesia yang cerdas, bertanggung jawab dan peka terhadap perkembangan teknologi informasi.

Paseban.com memberikan informasi dunia telekomunikasi yang lengkap, relevan, akurat dan juga dapat dipertanggung jawabkan.

Paseban.com akan menjadi tempat bertemunya komunitas *mobile* di Indonesia, mulai dari Pengguna akhir layanan telekomunikasi, Pelaku industri telekomunikasi, dan Pemerintah.

Paseban.com lahir sebagai bentuk partisipasi masyarat *mobile* Indonesia untuk menciptakan sebuah komunitas positif yang menghubungkan para pengguna *mobile* di Indonesia dalam satu lingkup komunitas, saling berbagi informasi, memberikan kontribusi positif satu sama lain, dan memenuhi kebutuhan bersosialisasi dari setiap individu penggunanya.

Paseban.com menciptakan kedekatan antara konsumen dan produsen *mobile* serta memberikan masukan kepada produsen akan kebutuhan dan keinginan dari tiap konsumennya, sehingga memungkinkan produsen untuk tetap berkewajiban menjaga dan meningkatkan kualitas pelayanan. Hal tersebut dapat terpenuhi dengan adanya informasi lengkap mengenai data konsumen, keluhan, testimoni dan informasi lainnya yang berasal dari konsumen. Disisi lain konsumen akan mendapatkan informasi cerdas mengenai jasa dan layanan yang diberikan oleh produsen, aktivitas produsen, promo, bahkan sampai dengan informasi harga produk dan layanan yang ditetapkan oleh produsen, semuanya akan tersaji *detail* dan lengkap di paseban.com

Paseban.com mempermudah pemerintah dalam mengawasi kebijakan-kebijakan yang diterapkan oleh pihak produsen *mobile* kepada konsumen *mobile*

serta memudahkan pemeritah untuk mensosialisasikan kebijakan-kebijakannya kepada masyarakat telekomunikasi di Indonesia. Lebih dari itu paseban.com memfasilitasi pemerintah untuk mengetahui hal-hal yang menjadi isu terhangat di lingkungan masyarakat *mobile* Indonesia sehingga kedepannya pemerintah dapat membuat sebuah aturan atau kebijakan yang tepat dan sesuai dengan kondisi masyarakat *mobile* di Indonesia.

Dengan demikian tujuan dari adanya kebijakan yang diterapkan oleh pemerintah yaitu untuk menciptakan suatu kondisi regulasi yang bukan hanya menguntungkan masyarakat tetapi juga mewujudkan tercapainya industri *mobile* di Indonesia.

Paseban.com akan menjadi tempat bertemunya komunitas *mobile* di Indonesia, mulai dari Pengguna akhir layanan telekomunikasi, Pelaku industri telekomunikasi, dan Pemerintah.

2. 1. 2. Logo Perusahaan

Logo instansi merupakan suatu gambar yang memiliki fungsi dan arti dari sebuah instansi untuk menggambarkan instansi itu sendiri.

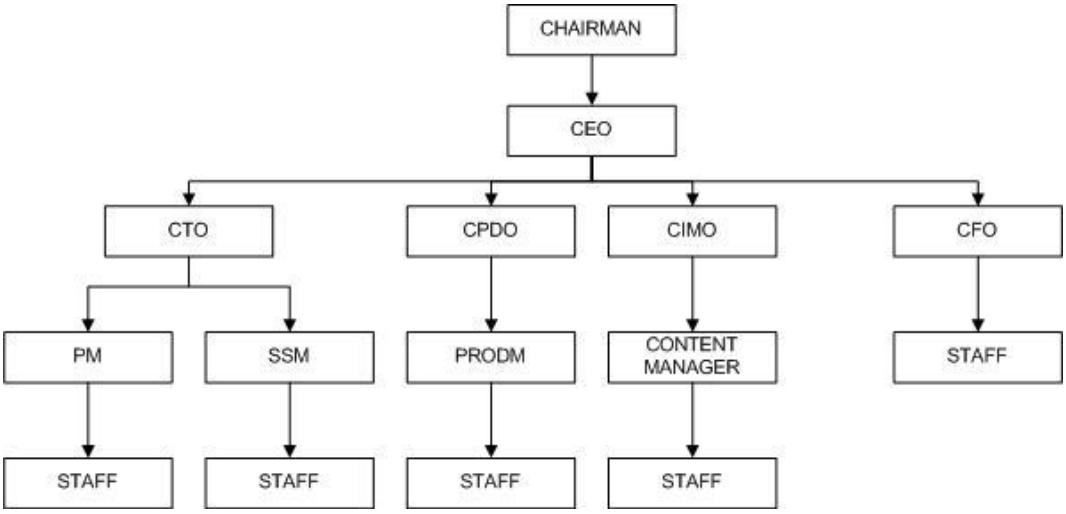
Logo Instansi dapat dilihat pada gambar 2.1:



Gambar 2. 1. Logo Perusahaan

2. 1. 3. Struktur Organisasi

Struktur organisasi sangatlah diperlukan demi kelancaran proses kinerja suatu organisasi atau perusahaan. Tingkatan atau hierarki dari kekuasaan yang tergambar, akhirnya akan menentukan hak dan kewajiban yang dimiliki oleh masing-masing individu dalam organisasi atau perusahaan tersebut. Berikut ini adalah struktur organisasi yang ada di Paseban.



Gambar 2. 2. Struktur Organisasi Paseban

2. 1. 3. 1. Job Description Perusahaan

Jabatan menandakan tingkat atau pangkat karyawan dalam hirarki organisasi. Jabatan juga menginformasikan tentang tanggung jawab kerja dari karyawan yang memegang jabatan.

1. *Chairman*

   Direktur utama (*Chairman*) adalah jabatan yang ditunjuk dan memberi laporan kepada Dewan Direksi / *Board of Director* (BOD)

   a. Memimpin seluruh dewan atau komite eksekutif.
   b. Menawarkan visi dan imajinasi di tingkat tertinggi (biasanya bekerjasama dengan MD atau CEO).
   c. Memimpin rapat umum, dalam hal: untuk memastikan pelaksanaan tata-tertib, keadilan dan kesempatan bagi semua untuk berkontribusi secara tepat; menyesuaikan alokasi waktu per item masalah; menentukan urutan agenda; mengarahkan diskusi ke arah konsensus; menjelaskan dan menyimpulkan tindakan dan kebijakan.
   d. Bertindak sebagai perwakilan organisasi dalam hubungannya dengan dunia luar.
   e. Memainkan bagian terkemuka dalam menentukan komposisi dari board dan sub-komite, sehingga tercapainya keselarasan dan efektivitas.
   f. Mengambil keputusan sebagaimana didelegasikan oleh BOD atau pada situasi tertentu yang dianggap perlu, yang diputuskan, dalam meeting-meeting BOD.

g. Menjalankan tanggung jawab dari direktur perusahaan sesuai dengan standar etika dan hokum.

2. CEO (*Chief Executive Officer*)

Chief Executive Officer (CEO) adalah manajer eksekutif tertinggi dalam sebuah perusahaan atau organisasi. CEO memiliki tanggung jawab tertentu tergantung pada kebutuhannya atau organisasinya. Tugas utama dari seorang CEO bervariasi oleh organisasi.

a. Bertindak sebagai perwakilan organisasi dalam hubungannya dengan dunia luar.
b. Memainkan bagian terkemuka dalam menentukan komposisi dari *board* dan sub-komite, sehingga tercapainya keselarasan dan efektivitas.
c. Mengambil keputusan sebagaimana didelegasikan oleh BOD atau pada situasi tertentu yang dianggap perlu, yang diputuskan, dalam pertemuan-pertemuan BOD.

3. CTO (*Chief Technology Officer*)

Chief Technology Officer (CTO) adalah eksekutif tertinggi yang bertanggung jawab untuk pengembangan dan penerapan teknologi. CTO mengawasi insinyur dan profesional teknis dalam penelitian dan desain produk baru dan sistem. Mereka mengelola semua aspek teknis seperti pengembangan perangkat lunak, keamanan jaringan dan operasi internet. Posisi CTO bertanggung jawab untuk memimpin kepala departemen lain

untuk mengembangkan strategi bisnis secara keseluruhan dan tujuan garis besar bagi suatu organisasi.

a. Mengembangkan dan menyusun strategi dan rencana TI perusahaan untuk jangka pendek, menengah, dan panjang, untuk memastikan kesesuaian dengan strategi dan rencana perusahaan dalam mencapai sasaran usaha
b. Menganalisa dan mengkaji perkembangan dan tren teknologi informasi serta pengaruhnya ke industri dan bidang usaha berkecimpung, untuk menjadi landasan bagi penetapan dan penyusunan rekomendasi pengembangan TI secara korporat.
c. Menyusun anggaran bagian TI dan mengontrol penggunaan dan realisasi dari anggaran tersebut untuk memastikan efektifitas dan efisiensinya.
d. Mengkordinir perencanaan arsitektur TI dengan membangun suatu pendekatan arsitektural TI untuk keseluruhan sistem perusahaan/ korporat, menyiapkan kapasitas perencanaan sehingga pelayanan kepada pelanggan terpelihara secara konsisten dan tidak kompromis, mempertimbangkan kreasi-nilai dalam membangun suatu arsitektur perusahaan aplikasi, dan selalu mengupdate pengetahuan tentang perkembangan TI yang mutakhir.
e. Mengarahkan dan mengkordinir penciptaan manajemen pelayanan TI yang efektif untuk seluruh cabang, unit dan korporat, dengan:

1. Bermitra dengan *provider* yang melayani perusahaan untuk menjamin perolehan pelayanan yang benar dan yang dibutuhkan oleh kegiatan usaha agar sejalan dengan proses-proses dalam perusahaan.
2. Mengkordinir pelaksanaan SLA (*service level agreement*) TI di seluruh perusahaan.
3. Menjamin pemberian manajemen pelayanan SLA yang tepat, demikian pula untuk pelayanan internal.
4. Menyiapkan rencana pemulihan pada upaya antisipasi terhadap tiap serangan bahaya yang sama artinya dengan menjamin sistem TI secara maksimum.
5. Mempertimbangkan biaya dan dasar-dasar dalam negosiasi kontrak.
6. menyeleksi, merekomendasikan dan memberikan pelatihan TI yang tepat kepada staf TI di perusahaan.

f. Mengarahkan dan mengkordinir pendayagunaan *software* dan *hardware* untuk mencapai kinerja optimum di seluruh perusahaan

g. Mengkordinir perencanaan dan pelaksanaan proyek TI yang besar di seluruh perusahaan, untuk memastikan integrasi korporat dan pencapaian target penyelesaian proyek dengan tepat waktu.

4. CFO (*Chief Financial Marketing*)

Chief Financial Officer (CFO) menyediakan dukungan operasional dan program bagi organisasi. CFO membawahi unit keuangan dan

merupakan juru bicara kepala keuangan bagi organisasi. CFO bertanggung jawab langsung kepada Presiden / *Chief Executive Officer* (CEO) dan langsung membantu *Chief Operating Officer* (COO) dalam semua masalah strategis dan taktis yang berkaitan dengan pengelolaan anggaran, analisis biaya manfaat, kebutuhan peramalan dan mengamankan pendanaan baru.

a. Mengkoordinir perumusan Strategi Jangka Panjang sebagai dasar perumusan Rencana Kerja dan Anggaran perusahaan (RKAP) dengan bekerja sama dengan Direksi lainnya.
b. Memberlakukan langkah-langkah yang dapat mengurangi dan menanggulangi berbagai jenis risiko finansial yang dapat dihadapi oleh perusahaan dengan berkoordinasi dengan Direksi lainnya.
c. Memastikan agar seluruh unit usaha dan wilayah kerja perusahaan mematuhi *policy* dan *standard operating procedure* (SOP) keuangan yang berlaku untuk masing-masing fungsi sesuai dengan rencana yang telah disetujui (*business units oversight*).
d. Membangun sinergi dan berusaha mencapai hasil bisnis yang optimal dari pelaksanaan seluruh usaha perusahaan.
e. Memastikan ketersediaan dana operasional yang dibutuhkan oleh perusahaan untuk kegiatan operasional sehari-hari, dengan melakukan koordinasi erat dengan para pimpinan unit usaha.
f. Memastikan konsolidasi keuangan yang akurat dan tepat waktu untuk keperluan pelaporan kepada Direksi dan Komisaris Perusahaan.

5. PM (*Promotion Manager*)
    a. Menyiapkan bahan atau materi promosi sesuai kebutuhan promosi suatu produk secara menyeluruh.
    b. Menganalisa spesifikasi setiap produk untuk dapat menemukan spesifikasi dan keunikan setiap produk yang akan digunakan sebagai materi promosi.
    c. Mengkoordinasikan, memonitor dan menganalisa strategi marketing secara nasional setiap jangka waktu tertentu sesuai kebijakan perusahaan.
    d. Menyusun usulan strategi penjualan secara nasional melalui hasil analisa pasar dan kesiapan promosi.
    e. Mengarahkan dan mengkoordinasikan kegiatan-kegiatan yang diikuti sebagai sarana promosi perusahaan.
    f. Menjalankan tugas-tugas terkait lainnya dalam upaya pencapaian target promosi.
6. SSM (*Sales Solution Marketing*)
    a. Menentukan harga jual, produk yang akan dilaunching, jadwal kunjungan serta system promosi untuk memastikan tercapainya target penjualan.
    b. Memonitor perolehan *order* serta merangkumkan *forecast* untuk memastikan kapasitas produksi terisi secara optimal

c. Memonitor jumlah *stock* seluruh Dept. Sales & Marketing untuk memastikan umur *stock* perusahaan tidak melebihi target yang telah ditentukan.
d. Menganalisa dan mengembangkan strategi marketing untuk meningkatkan jumlah pelanggan dan area sesuai dg target yang ditentukan.
e. Menganalisa dan memberikan arah pengembangan design & warna, untuk memastikan pengembangan produk sesuai dengan kebutuhan pasar.
f. Melakukan evaluasi kepuasan pelanggan dari hasil survey seluruh sales *team* untuk memastikan tercapainya target kepuasan pelanggan yang ditentukan.
g. Menerapkan budaya, sistem, dan peraturan intern perusahaan serta menerapkan manajemen biaya, untuk memastikan budaya perusahaan dan sistem serta peraturan dijalankan dengan optimal.

7. PRODM (*Production Manager*)
   a. Mempersiapkan program kerja dan mendapatkan persetujuan dari kepala dinas.
   b. Mengatur dan mengawasi kegiatan operasional harian di departemen.
   c. Mengadakan pertemuan rutin dengan laporan langsung untuk mengatasi masalah teknis.
   d. Menjamin semua proses permasalahan.
   e. Bernegosiasi dengan pemasok dalam produksi.

f. Secara teratur mengunjungi pemasok di berbagai tempat.

g. Meningkatkan pengetahuan produk dan setiap kemungkinan inovasi dalam produksi.

8. Staf

a. Melakukan pemeliharaan dan perbaikan pada setiap unit alat kerja (komputer) sesuai dengan waktu yang diberikan.

b. Mampu menyelesaikan pekerjaan administrasi dengan baik.

c. Menjalin kerjasama yang baik dengan rekan-rekan sekerja.

d. Bertanggung jawab kepada atasan langsung & Manager IT

2. 1. 4. Visi Dan Misi Perusahaan

a. Visi

To Be The Biggest Mobile Portal & Community Website In Indonesia

b. Misi

1. Memberikan informasi dunia telekomunikasi yang lengkap, relevan, akurat dan dapat dipertanggung jawabkan.
2. Menciptakan media informasi yang interaktif dan menciptakan sarana edukasi untuk para pengguna mobile di Indonesia
3. Membuat sarana untuk memenuhi kebutuhan serta keinginan pengguna layanan telekomunikasi dalam aplikasi dan *tools* yang bermanfaat
4. Fasilitator pengembangan bisnis industri telekomunikasi

2. 2. Sistem Informasi

Pada dasarnya sistem informasi bukanlah sistem yang selalu terkait dengan komputer. Adapun beberapa penjelasan terkait dengan sistem informasi adalah pengertian sistem, pengertian data, pengertian informasi, dan pengertian sistem informasi itu sendiri.

2. 2. 1. Pengertian Sistem

Pengertian sistem bisa dilihat dari dua pendekatan, yaitu berdasarkan prosedur dan komponennya. Sistem dilihat dari pendekatan prosedur berarti kumpulan prosedur yang saling bekerja sama untuk mencapai suatu tujuan tertentu sedangkan sistem dilihat dari pendekatan komponennya berarti sistem terdiri dari komponen-komponen yang saling berinteraksi untuk mencapai suatu tujuan tertentu.

2. 2. 2. Pengertian Data

Data adalah kenyataan yang menggambarkan suatu kejadian–kejadian dan kesatuan yang nyata. Atau data adalah : representasi dunia nyata yang mewakili suatu objek seperti manusia (pegawai, mahasiswa, pelanggan), hewan, peristiwa, konsep, keadaan dll, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya.

2. 2. 3. Pengertian Informasi

Informasi merupakan data yang memiliki nilai lebih bagi penerimanya karena telah melewati proses pengolahan tertentu sehingga bisa dimanfaatkan untuk mengambil keputusan baik itu untuk saat ini ataupun untuk masa yang akan

datang atau informasi didefinisikan sebagai data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

2. 2. 4. Pengertian Sistem Informasi

Berdasarkan pemaparan sebelumnya mengenai sistem, data, dan informasi maka dapat diambil kesimpulan bahwa pengertian dari sistem informasi adalah sekumpulan komponen-komponen dan prosedur-prosedur yang saling bekerja sama untuk mencapai suatu tujuan tertentu. Berikut pengertian-pengertian lainnya dari sistem informasi:

1. *A set of people, data, procedures, hardware and software that work together to achieve the common goal of information management*.
2. Keterkaitan komponen-komponen dalam pengumpulan (atau menerima), proses, menyimpan dan mendistribusikan informasi untuk mendukung pembuatan keputusan (*decision making*), mengkoordinasi serta mengkontrol
3. Sistem informasi juga membantu, manajer dan para pegawai (pekerja) menganalisa masalah, menggambarkan 'subyek' yang kompleks dan membuat (menghasilkan) produk-produk baru
4. Dasar (pondasi) dari bisnis pada umumnya.
5. Ada tiga aktifitas pada sistem informasi menghasilkan informasi pada organisasi yang diperlukan untuk membuat suatu keputusan, pengkontrolan, analisa masalah-masalah dan membuat produk baru atau layanan. Aktifitas itu adalah *input*, *process* dan *output*.

Input berarti menangkap atau pengumpulkan data mentah (*raw data*) didalam organisasi atau dari luar organisasi – *environment* / lingkungan. Proses

diartikan dengan menterjemahkan data *input* kedalam bentuk yang lebih berarti. Sedangkan *output* adalah mentransfer informasi yang telah diproses kepada manusia dan aktivitasnya.

2. 3. Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Dikembangkan oleh *Sun Microsystems* dan diterbitkan tahun 1995. Java tidak boleh disalahpahami sebagai *JavaScript*. *JavaScript* adalah bahasa *scripting* yang digunakan oleh *web browser* (Fikri, 2005).

2. 3. 1. Sejarah Singkat

Pada 1991, sekelompok insinyur Sun dipimpin oleh Patrick Naughton dan James Gosling ingin merancang bahasa komputer untuk perangkat konsumer seperti *cable TV Box*. Karena perangkat tersebut tidak memiliki banyak memori, bahasa harus berukuran kecil dan mengandung kode yang liat. Juga karena manufaktur – manufaktur berbeda memilih *processor* yang berbeda pula, maka bahasa harus bebas dari manufaktur manapun. Proyek diberi nama kode ”*Green*”.

Kebutuhan untuk fleksibilitas, kecil, liat dan kode yang netral terhadap *platform* mengantar tim mempelajari implementasi Pascal yang pernah dicoba. Niklaus Wirth, pencipta bahasa Pascal telah merancang bahasa portabel yang menghasilkan *intermediate code* untuk mesin hipotesis. Mesin ini sering disebut dengan mesin maya *(virtual machine)*. Kode ini kemudian dapat digunakan di

sembarang mesin yang memiliki *interpreter*. Proyek *Green* menggunakan mesin maya untuk mengatasi isu utama tentang netral terhadap arsitektur mesin.

Karena orang – orang di proyek *Green* berbasis C++ dan bukan Pascal maka kebanyakan sintaks diambil dari C++, serta mengadopsi orientasi objek dan bukan prosedural. Mulanya bahasa yang diciptakan diberi nama ”Oak” oleh James Gosling yang mendapat inspirasi dari sebuah pohon yang berada pada seberang kantornya, namun dikarenakan nama Oak sendiri merupakan nama bahasa pemrograman yang telah ada sebelumnya, kemudian SUN menggantinya dengan JAVA. Nama JAVA sendiri terinspirasi pada saat mereka sedang menikmati secangkir kopi di sebuah kedai kopi yang kemudian dengan tidak sengaja salah satu dari mereka menyebutkan kata JAVA yang mengandung arti asal bijih kopi. Akhirnya mereka sepakat untuk memberikan nama bahasa pemrograman tersebut dengan nama Java.

Produk pertama proyek *Green* adalah *Star* 7 (*7), sebuah kendali jarak jauh yang sangat cerdas. Dikarenakan pasar masih belum tertarik dengan produk konsumer cerdas maka proyek Green harus menemukan pasar lain dari teknologi yang diciptakan. Pada saat yang sama, implementasi WWW dan Internet sedang mengalami perkembangan pesat. Di lain pihak, anggota dari proyek *Green* juga menyadari bahwa Java dapat digunakan pada pemrograman internet, sehingga penerapan selanjutnya mengarah menjadi teknologi yang berperan di web.

Sebagai sebuah bahasa pemrograman, Java dapat membuat seluruh bentuk aplikasi, *desktop, web* dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain.

Java adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai *platform* sistem operasi. Perkembangan Java tidak hanya terfokus oada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source.* Sebagai sebuah peralatan pembangun, teknologi Java menyediakan banyak *tools* : *compiler, interpreter,* penyusun dokumentasi, paket kelas dan sebagainya. Aplikasi dengan teknologi Java secara umum adalah aplikasi serbaguna yang dapat dijalankan pada seluruh mesin yang memiliki *Java Runtime Environment* (JRE).

Terdapat dua komponen utama dari *Deployment Environment*. Yang pertama adalah JRE, yang terdapat pada paket J2SDK, mengandung kelas – kelas untuk semua paket teknologi Java yang meliputi kelas dasar dari Java, komponen GUI dan sebagainya. Komponen yang lain terdapat pada *Web Browser*. Hampir seluruh *Web Browser* komersial menyediakan *interpreter* dan *runtime environment* dari teknologi Java.

2. 3. 2. Versi Awal

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:

a. Java.lang: Peruntukan kelas elemen-elemen dasar.

b. Java.io: Peruntukan kelas *input* dan *output*, termasuk penggunaan berkas.

c. Java.util: Peruntukan kelas pelengkap seperti kelas struktur data dan kelas kelas penanggalan.

d. Java.net: Peruntukan kelas TCP/IP, yang memungkinkan berkomunikasi dengan komputer lain menggunakan jaringan TCP/IP.

e. Java.awt: Kelas dasar untuk aplikasi antarmuka dengan pengguna (GUI)

f. Java.applet: Kelas dasar aplikasi antar muka untuk diterapkan pada penjelajah web.

2. 3. 3. Kelebihan

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 *operating system Linux* tetapi dijalankan dengan baik di atas *Microsoft Windows*. *Platform* yang didukung sampai saat ini adalah *Microsoft Windows, Linux, Mrac OS* dan *Sun Solaris*. Penyebanya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs Java) untuk menginterpretasikan *bytecode* tersebut.

2. OOP (*Object Oriented Programming* - Pemrogram Berorientasi Objek) yang artinya semua aspek yang terdapat di Java adalah Objek. Java merupakan

salah satu bahasa pemrograman berbasis objek secara murni. Semua tipe data diturunkan dari kelas dasar yang disebut *Object*. Hal ini sangat memudahkan pemrogram untuk mendesain, membuat, mengembangkan dan mengalokasi kesalahan sebuah program dengan basis Java secara cepat, tepat, mudah dan terorganisir. Kelebihan ini menjadikan Java sebagai salah satu bahasa pemograman termudah, bahkan untuk fungsi fungsi yang *advance* seperti komunikasi antara komputer sekalipun.

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. 3. 4. Kekurangan

1. Tulis sekali, perbaiki 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. 3. 5. Tahap Kompilasi Java

Pada tahap kompilasi untuk program java sendiri, terdapat beberapa tahapan, berikut akan diuraikan tahap-tahap untuk penkompilasian program java :

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 tidak bisa di pakai.

2. 4. Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti 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 2000, 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. versi android terbaru yaitu versi 3.0. Pada tahun 2005 Google mengakusisi Android Inc yang pada saat itu dimotori oleh Andy Rubin, Rich Miner, Nick Sears dan Crish White. Yang kemudian pada tahun itu juga memulai membangun *platform* Android secara lebih intensif. Kemudian pada tanggal 12 November 2007 Google bersama *Open Handset Aliance* (OHA) yaitu konsorium perangkat lunak mobile terbuka, merilis Google Android SDK, setelah mengumumkannya seminggu sebelumnya. Dan sambutannya sangat luar biasa , hampir semua media berita tentang IT dan

Programming memberitakan tentang dirilisnya Android SDK (*Software Developtment Kit*). Ini dikarenakan ide dari *flatform* Android sangat menarik untuk *developer* dan *programmer* di seluruh penjuru dunia. Di sisi lain, produsen pembuat *handset* ponsel juga segera berlomba-lomba membuat *handset* dengan *platform* Android ini. Google bersama dengan OHA merilis paket *software* SDK yang lengkap untuk mengembangkan aplikasi pada perangkat *mobile*, yaitu : Sistem Operasi, *Middleware* dan Aplikasi utama untuk perangkat *mobile*. Sebagai *Programmer* atau *Developer* kita bisa melakukan segalanya, mulai dari membuat aplikasi pengiriman SMS hanya dengan dua baris kode, hingga mengganti *even* pada *Home Screen* perangkat Android. Selain itu, bahkan dengan mudah kita bisa membuatdan mengkostumisasi Sistem Operasinya, atau mengganti semua aplikasi *default* dari google.

Semua aplikasi yang dibuat untuk android akan memiliki akses setara dalam mengakses seluruh kemampuan *handset*, tanpa membedakan apakah itu merupakan aplikasi inti atau aplikasi pihak ketiga. Dalam kata lain dengan *platform* android ini, *Programer* atau *Developer* secara penuh akan bisa mengkustomisasi perangkat androidnya. Android *built in* pada Linux kernel (*Open Linux Kernel*), dengan sebuah mesin virtual yang telah didesain dan untuk mengoptimalkan penggunaan sumber daya memori dan *handware* pada lingkungan perangkat *mobile* (*Mobile Environment*). Dalvik adalah nama dari Android Virtual Mesin, yang merupak interpreter virtual mesin yang akan mengeksekusi *file* kedalam *format* Dalvik Executable (*.dex). Sebuah format

yang dirancang untuk ruang penyimpanan yang efisien dan eksekusi memori yang terpetakan (*memory-mappable execution*).

Dlavik Virtual Mesun (Dalvik VM) berbasi register , dan dapat mengeksekusi kelas (class) yang telah terkompilasi pada compier bahasa java, kemudian di transformasikan kedalam *native format* dengan menggunakan *Tool* "dx" yang telah terintegrasi. Kita mungkin telah mengenal JavaVM (Java Virtual Mesin), yang saat ini bisa kita temukan pada setiap komputer *desktop*. Berbeda dengan DalvikVM, JavaVM berbasis *stack* (*Stack-based Virtual Machine*), DalvikVM memiliki keunggulan dengan menggunakan *Registred Based*, ini karena pada prosesor perangkat genggam telah dioptimasi untuk eksekusi berbasis *register* . Android saaat ini tidak hanya berjalan pada *handphone*, beberapa vendor menanamkan Android pada *tablet, Internet Tablet, E-Book Reader*, Laptop, dan *gadget* lainnya.

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 *mobile* yang merupakan modifikasi kernel Linux 2.6. 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. Beberapa versi dari android yang sudah diluncurkan diantaranya:

1. Andorid 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. 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.

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; kamera, *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.

4. Android versi 2.0/2,1 (Eclair)

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*.

5. Android versi 2.2 (Froyo: *Frozen Youghurt*)

   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 portabel*, dan kemampuan *auto update* dalam aplikasi Android *Market*.

6. Android versi 2.3 (Gingerbread)

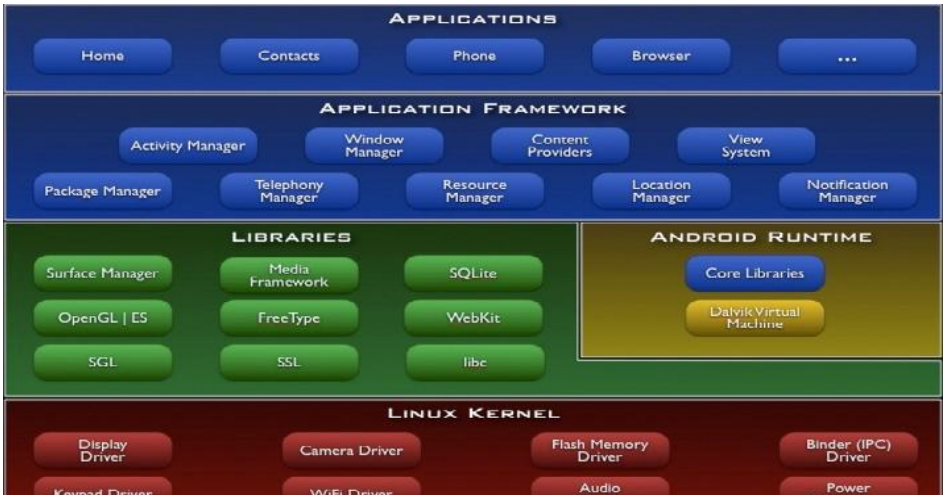
   Pada 6 Desember 2010, Android versi 2.3 (*Gingerbread*) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (*gaming*), peningkatan fungsi *copy paste*, layar antar muka (*User Interface*) didesain ulang, dukungan *format video* VP8 dan 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. 4. 1. Features

Features yang tersedia pada *platform* Android saat ini antara lain :

a. *Framework* Aplikasi yang mendukung penggantian komponen dan *reusable*. (Mulyadi, 2012)
b. Mesin Virtual Dalvik berjalan diatas Linux kernel dan dioptimalkan untuk perangkat *mobile*.
c. *Integrated browser* berdasarkan *open source engine WebKit.*
d. Grafis yang dioptimalkan dan didukung oleh *library* grafis 2D yang terkostumisasi, grafis 3D berdasarkan spesifikasi *openGL ES 1,0* (Opsional akselarasi *hardware*).
e. *SQLite* untuk penyimpanan data.
f. *Media support* yang mendukung *audio, video*, dan gambar (MPGE, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
g. GSM *Telephony* (tergantung *hardware*).
h. *Bluetooth, EDGE, 3G*, dan *WIFI* (tergantung *hardware*).
i. *Multi-touch* : kemampuan layaknya handset modern yang dapat menggunakan dua jari atau lebih untuk berinteraksi dengan perangkat.
j. Lingkungan *Development* yang lengkap dan kaya termasuk perangkat *emulator*, *tools* untuk *debugging*, profil, dan kinerja memori, dan *plugin* untuk *Eclips IDE*.
k. *Market* : Seperti kebanyakan *Handphone* yang memiliki tempat penjualan aplikasi, *market* pada android merupakan katalog aplikasi yang dapat diunduh dan di *install* pada *handphone* melalui internet. (Mulyadi, 2012)

2. 4. 2. Arsitektur Android



Gambar 2. 3. Arsitektur Android

2. 3. 2. 1. Linux Kernel

Android bukan Linux, akan tetapi android dibangun diatas Linux Kernel yaitu versi 2.6 sehingga kehandalannya dapat di percaya. Untuk inti sistem *service linux* yang digunakan seperti keamanan, manajemen memori, proses manajemen, *network*, dan *drive model*. Seperti yang terlihat digambar, Linux Kernel menyediakan *Driver Layer*, Kamera, *Keypad*, Kamera, *Wifi, Flash Memory*, *Audio*, dan IPC (*Interprocess Communication*) untuk mengatur aplikasi dan keamanan. Kernel juga bertindak sebagai lapisan abstrak antara *hardware* dan *software stack*-nya. (Mulyadi, 2012)

2. 3. 2. 2. Libraries

Android menyertakan *libraries* C/C++ yang digunakan oleh berbagai komponen dari sistem Android. Kemampuan ini disediakan kepada *Developer* aplikasi melalui *framework* aplikasi Android. beberapa inti *libraries* tercantum di bawah ini :

A. *System* C *library* – Variasi dari implementasi BSD-berasal pelaksanaan sistem standar C *library* (*libe*), sesuai untuk perangkat *embedded* berbasis linux. (Mulyadi, 2012)

B. *Media libraries* – Paket*Video* berdasarkan Open CORE; *library* mendukung pemutaran rekaman dan populer banyak *format audio* dan *video*, serta *file* gambar, termasuk MPEG, H.264, MP3, AAC, AMR, JPG. Dan PNG. (Mulyadi, 2012)

C. *Surface Manager* – mengelola akses ke subsistem layar. (Mulyadi, 2012)

D. *LibWebCore* – mesin *web modern* yang *powerfull* yang baik *browser* Android *embeddable web*. (Mulyadi, 2012)

E. SGL – mesin grafis 2D. (Mulyadi, 2012)

F. 3D *libraries* – sebuah pelaksanaan berdasarkan openGL ES 1,0 API; perpustakaan baik menggunakan perangkat keras akselarasi 3D (apabila tersedia) atau yang disertakan, sangat optimal 3D *software rasterizer*. (Mulyadi, 2012)

G. *FreeType – bitmap* dan vektor *font rendering* SQLite – mesin *database* yang kuat dan ringan, dan penghubung tersedia untuk semua aplikasi. (Mulyadi, 2012)

2. 3. 2. 3. Android-Runtime

Android terdiri satu *set core libraries* yang menyediakan sebagian besar fungsi yang sama dengan yang terdapat pada *core libraries* bahasa pemograman java. Setiap aplikasi menjalankan prosesnya sendiri dalam Android. Setiap aplikasi menjalankan prosesnya sendiri dalam Android, dengan masing-masing instan dari mesin virtual Dalvik (Dalvik VM). Dalvik dirancang agar perangkat dapat menjalankan *multiple* VMs secara efisien. Mesin Virtual Dalvik

mengeksekusi *file* dalam Dalvik *executable* (.dex), sebuah format yang dioptimalkan untuk memori yang kecil. Dalvik VM berbasis, berjalan dan dikompilasi oleh *compiler* bahasa Java yang telah ditransformasikan ke dalam .dex *format* oleh *tool* "dx" yang telah di sertakan, Dalvik VM bergantung pada kernel Linux untuk berfungsi seperti *threading* dan manajemen memori tingkat rendahnya. (Mulyadi, 2012)

2. 3. 2. 4. *Framework*-Aplikasi

Pengembang memiliki akses penuh menuju API *framework* yang sama, dengan yang digunakan oleh aplikasi inti. Arsitektur aplikasi dirancang agar komponen dapat digunakan kembali (*reuse*) denga mudah. Setiap aplikasi dapat memanfaatkan kemampuan ini dan aplikasi yang lain mungkin akan memanfaatkan kemampuan ini (sesuai dengan batasan keamanan yang didefinisikan oleh *framework*). Mekanisme yang sama memungkinkan komponen untuk diganti oleh pengguna. (Mulyadi, 2012)

Semua pengguna merupakan rangkaian set layanan dan sistem, termasuk:

A. *View* yang kaya dan *extensible* yang dapat digunakan untuk membangun aplikasi, termasuk *list, grid*, kotak teks, tombol, dan bahkan sebuah *embeddable web*. (Mulyadi, 2012)

B. *Content provider* yang memungkinkan aplikasi untuk mengakses data (seperti dari daftar kontak *help*) atau membagi data tersebut. (Mulyadi, 2012)

C. *Resource Manager*, yang menyediakan akses ke kode non sumber lokal seperti *string*, gambar, dan tata letak *file*. (Mulyadi, 2012)

D. *Notification Manager* yang memungkinkan semua aplikasi menampilkan *alert* yang bisa dikostumisasi didalam status bar. (Mulyadi, 2012).

E. *Activity Manager* yang mengelola siklus hidup aplikasi dan menyediakan navigasi umum *backstack*.

2. 3. 2. 5. Application

Android telah menyertakan aplikasi inti seperti *email elient*, SMS, kalender, peta, *browser*, kontak, dan lain-nya. Semua aplikasi tersebut ditulis dengan menggunakan bahasa pemrograman Java. Pada *layer* inilah *developer* atau kita menempatkan aplikasi yang dibuat. Yang istimewa adalah pada android semua aplikasi baik aplikasi inti (*native*) maupun aplikasi pihak ketiga berjalan pada *layer* aplikasi yang dibuat untuk android akan memiliki akses yang setara dalam mengakses seluruh kemampuan *handset*, tanpa membedakan apakah itu merupakan aplikasi inti atau aplikasi pihak ketiga. Dalam kata lain dengan *platform* android ini, *Programmer* atau *developer* secara penuh akan bisa mengkostumisasi perangkat androidnya. (Mulyadi, 2012)

2. 5. *Upload* dan *Download*

Upload dan *download* adalah dua istilah yang digunakan dalam komputasi untuk merujuk untuk mentransfer data. Singkatnya, *file* dianggap *upload* ketika mereka ditransfer dari komputer ke server pusat. *File download* bila mereka dipindahkan dari server ke komputer. Ketika data sedang ditransfer dari satu sistem ke sistem lain, ini bisa berupa *upload* atau *download* tergantung dari situasi.

Upload dan *download* yang digunakan setiap hari oleh konsumen internet. Email yang diunduh dari server sementara balasan diunggah untuk dikirim keluar. Para kode sumber dari suatu halaman web diunduh ke komputer pengguna akhir sehingga ia dapat melihat konten. Konsumen lain menengunduh *game*, *file audio* , atau *video* untuk hiburan pribadi. Konsumen yang *host* situs Web mereka sendiri dapat menggunakan khusus *File Transfer Protocol* (FTP) program untuk *Upload file* ke server mereka, atau mengunduh *file* dari lokasi lain.

Pada dasarnya setiap informasi yang dapat didigitalkan dapat diunggah atau diunduh. Istilah merujuk pada informasi bergerak sekitar sebuah web informasi, apakah itu jaringan rumah atau kantor atau seluruh Internet. Sejumlah program telah muncul untuk membantu konsumen dengan *upload* dan *download*, juga disebut *file sharing* .

2. 5. 1. Upload

Upload biasanya mengacu pada menyalin *file* dari sistem perifer yang lebih kecil ke sistem pusat yang lebih besar. Sebagai contoh, pengguna dapat *Upload file* dari ponsel-nya ke komputer pribadi. Data ini pada gilirannya mungkin akan *upload* lagi ke server pusat. *Upload file* kecil mungkin mengambil hanya beberapa detik, sedangkan yang lebih besar seperti *file* grafis yang besar dapat mengambil waktu jam. Dalam kebanyakan kasus, *upload* dapat dicapai saat komputer melakukan tugas-tugas lain, meskipun mungkin memperlambat komputer jika menangani *file* besar.

2. 5. 2. Download

Download mengacu pada transfer data dari sistem pusat ke sistem yang lebih kecil. Sebagai contoh, *file* yang sama *upload* ke server di atas mungkin diunduh untuk dilihat oleh pengguna lain. Ketika menengunduh *file*, *file* yang lebih besar, semakin banyak waktu yang dibutuhkan untuk menyelesaikan transfer, seperti diunggah. *File* yang diuduh biasanya tetap di server kecuali pengguna menentukan lain, yang berarti bahwa orang lain dapat mengakses informasi

2. 6. Internet

Internet merupakan media yang akan digunakan sebagai media interaksi anatara pihak merchant dan konsumen. Untuk dapat menjalankan bisnis *online*, maka *merchant* harus melakukan koneksi terhadap internet. Demikian pula konsumen yang akan membeli produk atau jasa secara elektronik harus menyambungkan komputernya ke internet melaui ISP (*Internet Service Provider*) tertentu. (Cantoni & Tardini, 2006)

2. 6. 1. Pengertian Internet

Internet adalah sistem informasi global berbasis komputer. Internet terbentuk dari jaringan-jaringan komputer-komputer yang terkoneksi satu sama lain. Setia jaringan mungkin menghubungkan puluhan, ratusan, hingga ribuan komputer, yang memungkinkan komputer-komputer itu saling berbagi data dan informasi satu sama lain dan untuk saling berbagi sumber daya komputasi yang mahal dan langka seperti komputer-komputer super (*Super Computer*) yang

sangat kuat dan sistem-sistem basis data yang berisi informasi-informasi yang bermanfaat. (Cantoni & Tardini, 2006)

2. 6. 2. Perkembangan Internet

Penggunaan internet berkembang pesat sejak penemuannya. Alih-alih menghubungkan jaringan-jaringan secara terbatas pada tipe komputer tertentu, teknologi internet memungkinkan koneksi terjadi diantara berbagai jenis komputer. Antar berbagai sistem operasi (baca:tidak terikat *platform* perangkat keras maupun perangkat lunak). Tidak ada jaringan yang terlalu cepat atau lambat, terlalu besar atau terlalu kecil sehingga tidak bisa dikoneksikan. Internet dapat menghubungkan jaringan-jaringan sederhana yang terdiri dari beberapa komputer di suatu ruangan hingga jaringan-jaringan canggih yang merentang antar benua dan menghubungkan ribuan (bahkan jutaan) komputer. (Cantoni & Tardini, 2006)

2. 7. 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. Teknologi XML adalah teknologi keturunan dari SGML (*Standard Generalized Markup Language,* ISO 8879) yang dikembangkan pada tahun 1980-an. XML adalah suatu bahasa *Markup*, yaitu bahasa yang berisikan kode-kode berupa tanda-tanda tertentu dengan aturan tertentu untuk

memformat dokumen teks dengan *tag* sendiri agar dapat dimengerti. XML adalah *language* untuk mengidentifikasi dan menandai data terstruktur

XML didesain untuk mampu 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.

XML memiliki tiga tipe *file***,** diantaranya:

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.

Keunggulan XML bisa diringkas sebagai berikut :

1. Pintar (*Intelligence*). XML dapat menangani berbagai tingkat (*level*) kompleksitas.
2. Dapat beradaptasi. Dapat mengadaptasi untuk membuat bahasa sendiri. Seperti Microsoft membuaat bahasa MSXML atau Macromedia mengembangkan MXML.
3. Mudah pemeliharaannya.

4. Sederhana. XML lebih sederhana.

Mudah dipindah-pindahkan (*Portability*). XML mempunyai kemudahan perpindahan (portabilitas) yang lebih bagus.

2. 8. *Tools* Implementasi

Tools implementasi merupakan peralatan-peralatan yang dibutuhkan dalam membangun perangkat lunak. Adapun *tools* implementasi yang digunakan untuk membuat aplikasi *mobile* jejaring sosial Kampus *Online* Unikom berbasis Android adalah JDK sebagai perangkat pembangun Java, Eclipse IDE sebagai lingkungan pengembangan perangkat lunak, dan SQL sebagai sistem manajemen basis data.

2. 8. 1. JDK

JDK (*Java Development Kit*) adalah sebuah perangkat peralatan yang digunakan untuk membangun perangkat lunak dengan menggunakan bahasa pemrograman Java. JDK berjalan diatas sebuah *virtual machine* yang dinamakan JVM (*Java Virtual Machine*).

Dokumentasi JDK berisi spesifikasi API, deskripsi fitur, panduan pengembang, referensi halaman untuk perkakas JDK dan utilitas, demo, dan *link* ke informasi terkait. Dokumentasi ini juga tersedia dalam bundel yand dapat diunduh dan diinstal pada komputer.

2. 8. 2. Eclipse IDE

Eclipse adalah sebuat komunitas proyek *open source* yang difokuskan pada membangun sebuah *platform* pengembangan yang dapat diperluas, *runtime* dan kerangka aplikasi untuk membangun, menyebarkan dan mengelola perangkat lunak di seluruh daur hidup perangkat lunak.

Komunitas Eclipse memiliki lebih dari 60 proyek *open source*. Proyek-proyek ini secara konsep terbagi menjadi 7 kategori :

1. *Enterprise Development*
2. *Embedded and Device Development*
3. *Rich Client Platform*
4. *Rich Internet Applications*
5. *Application Frameworks*
6. *Application Lifecycle Management (ALM)*
7. *Service Oriented Architecture (SOA)*

Komunitas Eclipse juga didukung oleh ekosistem besar dan dinamis dari penyedia solusi teknologi informasi utama, *start-up* inovatif, universitas dan lembaga penelitian, dan individu yang memperluas, mendukung, dan melengkapi *platform* Eclipse.

2. 8. 3. SQLite

SQLite merupakan sebuah perpustakaan proses yang menerapkan kemandirian, tanpa server, tidak adanya konfigurasi mesin, dan basis data SQL

transaksional. Kode untuk SQLite berada dalam domain publik dengan demikian bebas untuk digunakan untuk tujuan apapun, komersial atau swasta.

SQLite adalah mesin basis data SQL yang tertanam. Tidak seperti kebanyakan basis data SQL lainnya, SQLite tidak memiliki proses server yang terpisah. SQLite membaca dan menulis secara langsung ke *file disk* biasa. Basis data SQL lengkap dengan berbagai tabel, indeks, *triggers*, dan *views* terkandung dalam sebuah *file disk* tunggal.

2. 9. *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. (Fowler, 2004)

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*). (Fowler, 2004)

2. 9. 1. Sejarah UML

Sampai era tahun 1990-an metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi booch, metodologi coad, metodologi OOSE, metodologi OMT, metodologi shlaer-mellor, 18

metodologi *wirfs-brock* dan masih banyak metodologi lainnya. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan kelompok/perusahaan lain yang menggunakan metodologi yang berlainan.

Dimulai pada bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan mempelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 di-*release draft* pertama dari UML (versi 0.8). Sejak tahun 1996

pengembangan tersebut dikoordinasikan oleh *Object Management Group* (OMG – http://www.omg.org).

2. 9. 2. Diagram UML

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

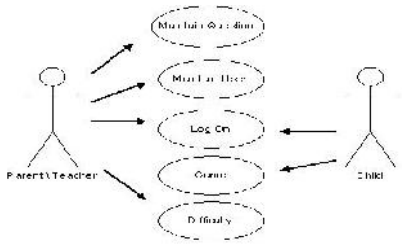
1. *Use Case* Diagram untuk memodelkan proses bisnis.
2. *Conceptual* Diagram untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. *Sequence* Diagram untuk memodelkan pengiriman pesan (*message*) antar objek.
4. *Collaboration* Diagram untuk memodelkan interaksi antar objek.
5. *State* Diagram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

Berikut akan dijelaskan 4 macam diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi objek, yaitu *use case diagram*, *sequence diagram, collaboration diagram,* dan *class diagram.*

2. 9. 3. Usecase Diagram

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

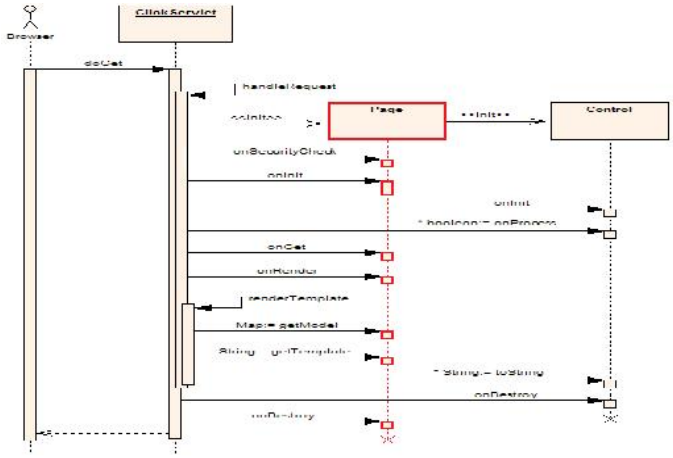
Usecase merepresentasikan operasi-operasi yang dilakukan oleh *actor. Usecase* digambarkan berbentuk elips dengan nama operasi dituliskan didalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke *use case.*



Gambar 2. 4. *Usecase* Diagram

2. 9. 4. Sequence Diagram

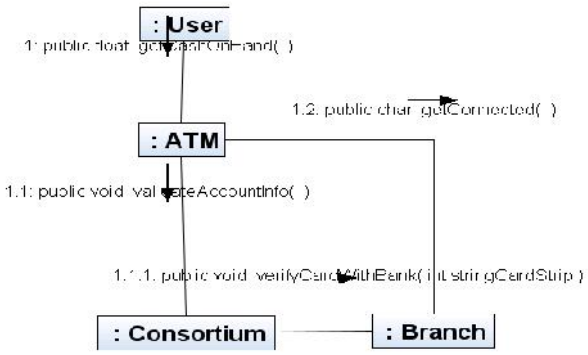
Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case.* Interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.



Gambar 2. 5. Sequence Diagram

2. 9. 5. Collaboration Diagram

Collaboration diagram dipakai untuk memodelkan interaksi antar objek di dalam sistem. Berbeda dengan *sequence diagram* yang lebih menonjolkan kronologis dari operasi-operasi yang dilakukan, *collaboration diagram* lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan oleh objek.

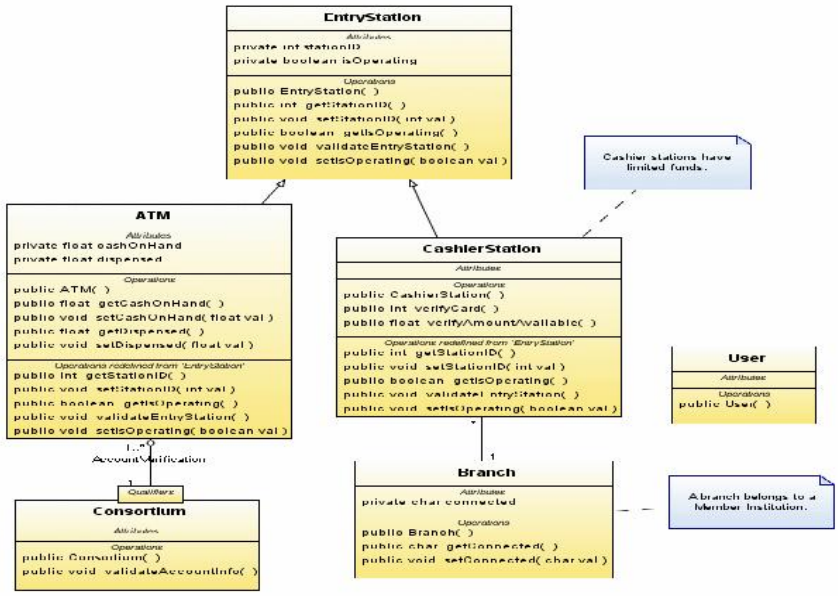


Gambar 2. 6. Collaboration Diagram

2. 9. 6. *Class Diagram*

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. *Class* diagram menunjukkan hubungan antar *class* dalam

sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.



Gambar 2. 7. Class Diagram

2. 10. *Personal Home Page* (PHP)

PHP: *Hypertext Preprocessor* adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk memrogram situs *web* dinamis. PHP dapat digunakan untuk membangun sebuah CMS.

2. 10. 1. Sejarah PHP

Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web*.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, *interpreter* PHP sudah diimplementasikan dalam program 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 rilis tersebut sebagai PHP 3.0 dan singkatan PHP dirubah menjadi akronim berulang *PHP: Hypertext Preprocessing*.

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 pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi *web* kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek.

2. 10. 2. Kelebihan PHP

Seluruh aplikasi berbasis *web* dapat dibuat dengan PHP. Namun kekuatan yang paling utama PHP adalah pada konektivitasnya dengan *system database* di dalam *web*. Sistem *database* yang dapat didukung oleh PHP adalah Oracle, MySQL , Sybase, PostgreSQL, dan lainnya. PHP dapat berjalan di berbagai sistem operasi seperti windows 98/NT, UNIX/LINUX, solaris maupun macintosh.

Software ini juga dapat berjalan pada *web server* seperti PWS (*Personal Web Server*), Apache, 2S, AOL *Server*, fhttpd, phttpd dan sebagainya. PHP juga merupakan bahasa pemograman yang dapat kita kembangkan sendiri seperti untuk menambah fungsi-fungsi baru. Keunggulan lainnya dari PHP adalah bahwa PHP juga mendukung komunikasi dengan layanan seperti *protocol* IMAP, SNMP, NNTP, POP3 dan bahkan HTTP. PHP dapat diinstal sebagai bagian atau modul dari apache *web server* atau sebagai CGI script yang mandiri.

Banyak keuntungan yang dapat diperoleh jika menggunakan PHP sebagai modul dari apache, di antaranya adalah :

1. Tingkat keamanan yang cukup tinggi
2. Waktu eksekusi yang lebih cepat dibandingkan dengan bahasa pemograman *web* lainnya yang berorientasi pada *server*-side *scripting*.

   Akses ke sistem *database* yang lebih fleksibel. seperti MySQL.