BAB 2
TINJAUAN PUSTAKA

2.1 Profil STV

Sunda Televisi merupakan buah kreativitas dari Prima Entertainment yang berada dibawah naungan PT Pasundan Utama Televisi dan berdiri pada tahun 2005. Memposisikan sebagai televisi lokal di Jawa Barat yang berfokus pada seni budaya Sunda pada khususnya dan kebudayaan di Jawa pada umumnya. Pemilihan wilayah Jawa Barat sebagai wilayah penyiaran STV adalah dengan mempertimbangkan bahwa Jawa Barat merupakan wilayah yang penduduknya lebih padat dibandingkan dengan wilayah Jawa Tengah dan Jawa Timur.

2.1.1 Sejarah STV

Sejarah STV Bandung dimulai dari pertama kalinya mengudara pada tanggal 18 Maret 2005 di frekuensi 34 UHF (*Ultra High Frekuensi*). STV Bandung pertama kali melakukan siaran selama 5 jam dimulai dari pukul 17.00 WIB sampai dengan pukul 22.00 WIB. Namun saat ini, setelah mengalami perkembangan luar biasa yaitu antusiasnya konsumen atau masyarakat terhadap program-program yang ditayangkan STV Bandung, maka STV Bandung memulai memperpanjang jam tayangnya dari pukul 09.00 WIB sampai dengan tengah malam yaitu pukul 00.00 WIB.

Tagline STV adalah One Tune Hade, yang dapat diartikan secara harfiah satu tune/channel yang bagus atau bisa juga diartikan secara pelafalan (Wantun Hade) berani tampil bagus (dalam bahasa sunda, hade artinya bagus, wanton artinya berani). STV berafiliasi dengan TV edukasi sejak tanggal 18 Maret 2011. Logo STV tidak lagi menggunakan S atau dihilangkan. Sebagai gantinya, logo STV yang sekarang hanyalah tulisan STV. Penggantian logo tersebut adalah sebagai wujud STV Bandung sebagai stasiun televisi yang terbuka, tanpa adanya batasan.

STV Bandung sekarang merupakan jaringan televisi kompas TV dan mulai tanggal 9 September 2011, seluruh program STV Bandung akan diisi oleh tayangan Kompas TV sebanyak 70 persen , dan 30 persen berasal dari tayangan local. Namun konten local ini akan berbeda dengan acara STV Bandung sebelumnya. Sehingga sebagian program STV Bandung yang lama akan dihapus.

STV memiliki wilayah jangkauan di Kota Bandung, Kabupaten Bandung, Kota Cimahi, Kabupaten Cianjur, sebagian Majalengka, Sumedang, Tasikmalaya, Subang, Garut, dan Sukabumi. Sebagai televisi lokal, STV Bandung sangat peduli dengan berbagai perkembangan di wilayah Jawa Barat sesuai dengan visi STV Bandung yaitu mengangkat dan menyuarakan kepentingan rakyat Jawa Barat.

2.1.2 Visi dan Misi STV

2.1.2.1 Visi

Visi dari STV adalah sebagai satu-satunya menjadi sumber inspirasi, informasi dan hiburan bagi anak-anak, anak muda, dan keluarga yang mengerti serta memahami keinginan dan kebutuhan pemirsa yang sekaligus menjadi media terefektif bagi agencies dan pemasang iklan.

2.1.2.2 Misi

1. Membangun dan mengembangkan Industri Televisi Lokal yang sehat dan berdaya guna bagi pembangunan Jawa Barat.
2. Memberikan informasi, pendidikan dan hiburan yang sesuai dengan karakter dan kondisi sosial, ekonomi, budaya dan politik masyarakat Jawa Barat, sehingga dapat mendorong terciptanya masyarakat yang mandiri, demokratis, adil dan sejahtera, serta beriman dan bertaqwa.
3. Membangun dan membina Sumber Daya Manusia bidang pertelevisian yang berkualitas dan bertanggung jawab.

4. Mendorong dan Memotivasi masyarakat untuk berperan aktif dalam pembangunan Jawa Barat menjadi propinsi yang aman dan termaju.
5. Menggali dan mengangkat berbagai potensi daerah Jawa Barat sebagai materi utama Siaran.

2.1.3 Logo STV Bandung

Seperti kita ketahui semua perusahaan, organisasi ataupun lembaga pasti mempunyai sebuah logo yang mencerminkan dan membedakan dengan yang lain. Begitu juga dengan STV Bandung, STV Bandung mempunyai sebuah logo yang membedakan dengan televise-televisi local yang ada di Bandung. Berikut ini adalah logo dari STV dapat dilihat dalam gambar 2.1 berikut ini:

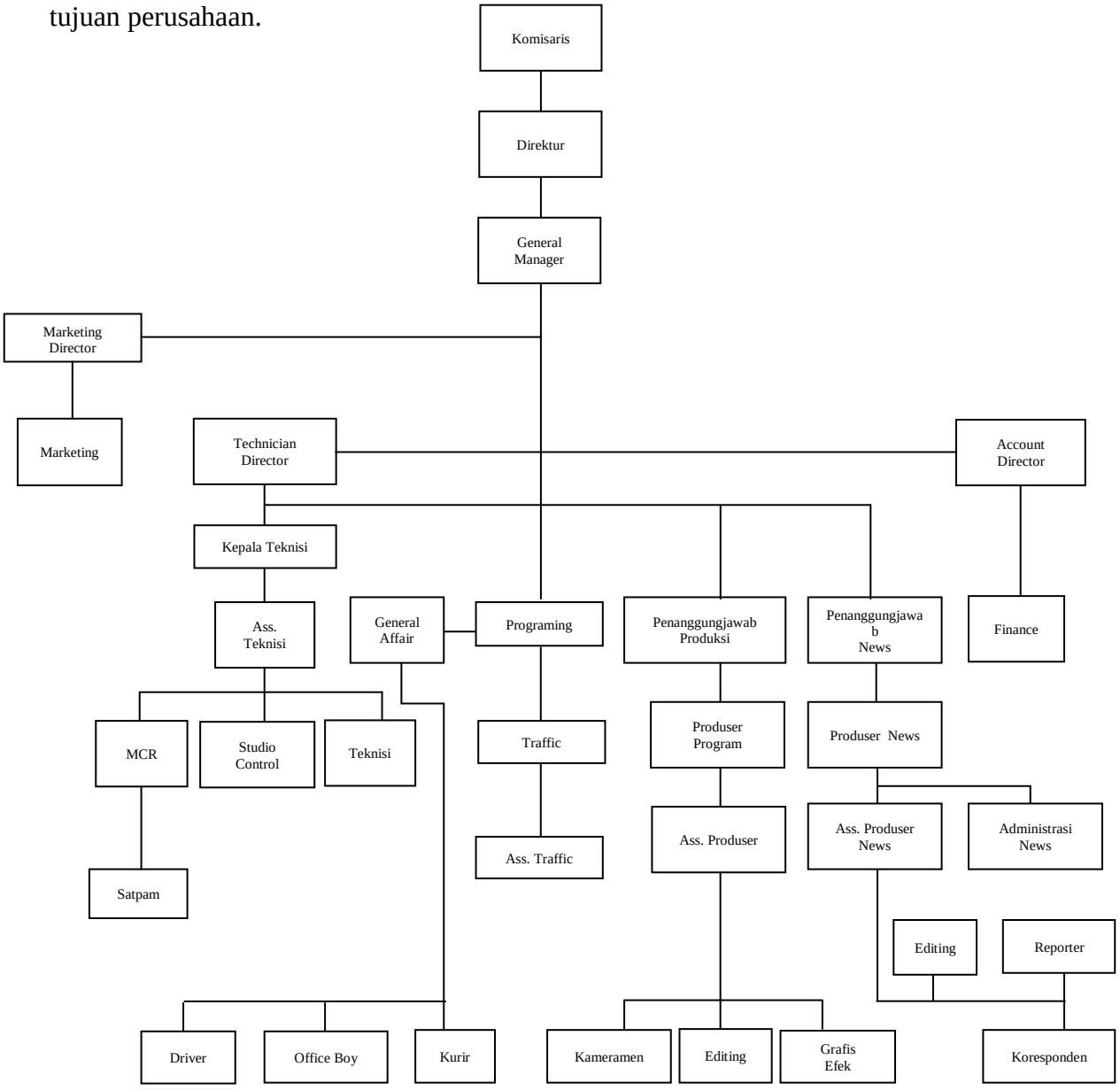


Gambar 2. 1 Logo STV Bandung

Logo di atas menggambarkan ciri khas dari STV, lambang warna biru membentuk seperti oval di tambah dengan huruf S yang terbalik membuat khas icon STV sendiri. Mengapa di ambil warna biru, karena biru melambangkan kesejukan, disini STV ingin menyejukan para penontonnya dengan sajian-sajian yang di suguhkan. Sedangkan huruf S pada tulisan STV berwarna merah yang artinya STV harus tetap semangat memberikan informasi kepada penonton yang ada di Jawa Barat. Warna hijau pada huruf T maksudnya adalah mengenai Jawa Barat itu sendiri, bersih, hijau, berbunga. Jadi, secara keseluruhan logo STV itu dapat diartikan mampu memberikan tontonan yang berwawasan, teraktual, dan tidak melupakan budaya di Jawa Barat.

2.1.4 Struktur Organisasi STV Bandung

Untuk memperoleh gambaran yang nyata mengenai wewenang, pembagian tugas, dan tangung jawab kepada masing-masing jabatan yang ada serta kerjasama antara atasan dan bawahan didalam suatu perusahaan yang disesuaikan dengan keadaan dan perkembangan perusahaan ini mempunyai pengaruh penting karena suatu garis wewenang dan tanggung jawab yang nyata akan menghasilkan garis kerja yang efektif dan efisien untuk mencapai tujuan perusahaan.



Gambar 2. 2 Struktur Organisasi STV Bandung

2.1.5 *Job Description* STV Bandung

Job Description adalah daftar tugas-tugas umum, fungsi dan tanggung jawab dari sebuah posisi. *Job description* biasanya dikembangkan dengan melakukan analisis pekerjaan, yang meliputi pemeriksaan tugas dan urutan tugas yang diperlukan untuk melakukan pekerjaan.

a. Pengarah Acara (Program Director)

   Bertanggungjawab terhadap hasil produksi sebuah program acara.

b. Pimpinan Produksi (Producer)

   Seseorang yang bertugas mengepalai sebuah produksi, baik di bidangartistik, lighting, dan lain-lain serta mengkoordinir dan mengawasi semua faset-faset produksi. Selain itu, produser juga berfungsi menyiapkan program acara siaran dan bertanggung jawab terhadap keberhasilan faktor ekonomi (keuntungan).

c. Penata Artistic (Art Director)

   Orang yang bertanggung jawab terhadap desain kreatif (artistik) dari program materi tayangan berupa desain set, set lokasi, art grafis, program ID, title yang sesuai dengan pendekatan artistik

d. Pimpinan Teknik (Technical Director)

   Orang yang bertanggung jawab dalam mempersiapkan dan menyetel semua perlatan yang digunakan, supaya alat yang satu dengan yang lainya bisa singkron, bertugas mengawasi crew teknis dan peralatan teknis lainnya.

e. Penata Kamera (Cameraman)

   Orang yang bertanggung jawab dan membantu sutradara dalam menterjemahkan skenario kedalam bahasa visual agar tercapai hasil yang sesuai dengan konsep desain produksi. Selain itu, sebagi perekam unsur visual dengan cine camera baik magnetik maupun elektronik, serta bertanggung jawab atas kualitas teknik, artistik dari rekaman tersebut.

f. Penata Suara (Audioman)

Orang yang bertanggung jawab terhadap kualitas audio dan scoring music pada setiap program acara.

g. Para Pemeran (Performers)

Orang yang menjadi pengisi acara dalam suatu proses produksi dan juga salah satu yang menentukan suksesnya suatu program yang dapat dinikmati penonton.

2.2 Perangkat Lunak / Aplikasi

Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media.

Adapun pengertian lain, aplikasi adalah penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan. Aplikasi dapat diartikan juga sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu. Aplikasi software yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi 2 (dua) yaitu:

1. Aplikasi software spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
2. Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

2.3 Internet

Internet (*kependekan dari interconnected-networking)* ialah sistem global dari seluruh jaringan komputer yang saling terhubung menggunakan standar *Internet Protocol Suite (TCP/IP)* untuk melayani miliaran pengguna di seluruh dunia. Manakala Internet ialah sistem komputer umum, yang berhubung secara global dan menggunakan TCP/IP sebagai protokol pertukaran paket *(packet*

switching communication protocol). Rangkaian internet yang terbesar dinamakan Internet. Cara menghubungkan rangkaian dengan kaedah ini dinamakan *internetworking*. Internet dijaga oleh perjanjian bilateral atau multilateral dan spesifikasi teknikal (protokol yang menerangkan tentang perpindahan data antara rangkaian). Protokol-protokol ini dibentuk berdasarkan perbincangan *Internet Engineering Task Force (IETF)* yang terbuka kepada umum.

2.4 Unified Modelling Language (UML)

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. Unified Modeling Language (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut.

2.4.1 Diagram

Diagram menampilkan representasi sebagian dari sistem, dan semantiks dengan view yang lain. Dalam UML, ada dua tipe diagram, yaitu :

a. Static Diagram : *use case diagram, class diagram, object diagram,component diagram dan deployment diagram.*
b. Dinamic Diagram : *sequence diagram, collaboration diagram, activity diagram dan statechart diagram.*

a. Use Case Diagram

Menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang atau sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Use case diagram dapat sangat membantu bila kita sedang menyusun requirement sebuah sistem,

mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua feature yang ada pada sistem.

Sebuah use case dapat meng-include fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa use case yang di-include akan dipanggil setiap kali use case yang meng-include dieksekusi secara normal. Sebuah use case dapat di-include oleh lebih dari satu use case lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang common. Sebuah use case juga dapat meng-extend use case lain dengan behaviour-nya sendiri. Sementara hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain.

b. Activity Diagram

Video Activity Diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana aktivitas itu berakhir. Activity diagram juga dapat menggambarkan proses pararel yang mungkin terjadi pada beberapa eksekusi. Activity diagram merupakan state diagram khusus, dimana sebagian besar state adalah action dan sebagian besar transisi di triger oleh selesainya state sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak , tetapi lebih menggambarkan proses-proses dan jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama dengan state, standar UML menggunakan segi empat dengan sudut membulat untuk menggambarkan aktivitas. Decision digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses pararel *(fork on join)* digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. Activity diagram dapat dibagi menjadi beberapa

object swimlane untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

c. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan teradap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

Masing-masing objek, termasuk aktor, memiliki lifeline vertikal. *Messag*e digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari class. Actiation bar menunjukan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk *objek boundary*, *controller* dan *persistent entity.*

d. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut

3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. Private, tidak dapat dipanggil dari luar class yang bersangkutan.
2. Protected, hanya dapat dipanggil oleh class yang bersangkutan dan anak anak yang mewarisinya.
3. Public, dapat dipanggil oleh siapa saja.

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

a. Hubungan Antar Class

1. Asosiasi, yaitu hubungan statis antar *class* . Umumnya menggambarkan *class* yang memiliki atribut berupa class lain, atau *class* yang harus mengetahui eksistensi class lain. Panah navigability menunjukkan arah query antar *class* .
2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
3. Pewarisan, yaitu hubungan hirarkis antar *class* . *Class* dapat diturunkan dari class lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di- passing dari satu class kepada class lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

2.4.2 Faktor Yang Mendorong Dibuatnya UML

Saat ini piranti lunak semakin luas dan besar lingkupnya, sehingga tidak bisa lagi dibuat asal-asalan. Piranti lunak saat ini seharusnya dirancang dengan memperhatikan hal-hal seperti *scalability, security*, dan eksekusi yang robust walaupun dalam kondisi yang sulit. Selain itu arsitekturnya harus

didefinisikan dengan jelas, agar bug mudah ditemukan dan diperbaiki, bahkan oleh orang lain selain programmer aslinya. Keuntungan lain dari perencanaan arsitektur yang matang adalah dimungkinkannya penggunaan kembali modul atau komponen untuk aplikasi piranti lunak lain yang membutuhkan fungsionalitas yang sama. Pemodelan (*modeling*) adalah proses merancang piranti lunak sebelum melakukan pengkodean (*coding*). Model piranti lunak dapat dianalogikan seperti pembuatan blueprint pada pembangunan gedung. Membuat model dari sebuah sistem yang kompleks sangatlah penting karena kita tidak dapat memahami sistem semacam itu secara menyeluruh. Semakin komplek sebuah sistem, semakin penting pula penggunaan teknik pemodelan yang baik.

Dengan menggunakan model, diharapkan pengembangan piranti lunak dapat memenuhi semua kebutuhan pengguna dengan lengkap dan tepat, termasuk faktor-faktor seperti *scalability, robustness, security*, dan sebagainya. Kesuksesan suatu pemodelan piranti lunak ditentukan oleh tiga unsur, yang kemudian terkenal dengan sebutan segitiga sukses *(the triangle for success*). Ketiga unsur tersebut adalah metode pemodelan (*notation*), proses (*process*) dan tool yang digunakan.

2.4.3 Tujuan UML

Pada dunia *software engineering*, UML merupakan singkatan dari *Unified Modelling Language*, yaitu suatu metode modeling generasi ketiga dan bahasa spesifikasi yang sifatnya non-proprietary. Sebenarnya penggunaan dari UML itu sendiri tidak terbatas hanya pada dunia software modeling, tetapi bisa pula digunakan untuk *modeling hardware (engineering systems)* dan sering digunakan sebagai *modeling* untuk proses bisnis dan juga modeling untuk struktur organisasi. UML adalah suatu metode terbuka yang digunakan untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan artifak-artifak dari suatu pengembangan sistem software yang berbasis pada objek. UML merupakan hasil kompilasi best *engineering practice* yang sudah terbukti sukses dalam pemodelan sistem yang besar,

sistem yang kompleks, khususnya pada level arsitektural. UML mengintegrasikan konsep dari Booch, OMT, OOSE dan juga Class-Relation dengan menggabungkan mereka menjadi suatu kesatuan bahasa modeling yang bisa berguna bagi siapa saja. UML bertujuan untuk menjadi standar bahasa modeling yang mampu untuk memodelkan sistem yang konkuren dan juga terdistribusi. Tujuan yang ingin dicapai dari penggunaan UML adalah:

1. Memberikan model yang siap pakai, bahasa permodelan visual yang ekspresif untuk mengembangkan dan saling menukar model yang mudah dan dimengerti secara umum.
2. Memberikan bahasa permodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
3. Menyatukan praktek-praktek terbaik yang terdapat dalam permodelan.

Dan bahasa pemodelan *Unified Modeling Language (UML)* ini juga biasa digunakan untuk :

1. Menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum, dibuat dengan use case dan actor.
2. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan interaction diagrams.
3. Menggambarkan representasi struktur statik sebuah sistem dalam bentuk class diagrams.
4. Membuat model behavior yang menggambarkan kebiasaan atau sifat sebuah sistem dengan state transition diagrams.
5. Menyatakan arsitektur implementasi fisik menggunakan component and development diagrams.
6. Menyampaikan atau memperluas fungsionalitas dengan *stereotypes.*

2.4.4 Cakupan UML

Model UML dapat mencakup banyak perbendaharaan. Diantaranya adalah sebagai berikut :

1. Things :

a. *Structural*

1. *Use case*, deskripsi interaksi dengan *external actor*.
2. C*lass*, deskripsi untuk objek-objek.
3. *Interface*, kumpulan operasi yang memberikan service tertentu untuk class atau component.
4. *Component*, bagian sistem yang dapat diganti *(replaceable)* dan realisasinya sesuai dengan interface.

b. *Behaviour*

1. Interaksi (message sequence chart), pertukaran messages antar objek.
2. *State machine*, urutan state dari objek dalam berinteraksi dengan objek lain.

c. *Grouping*

Package, mekanisme untuk mengumpulkan elemen ke dalam satu set (group).

d. Anotasi

Catatan atau keterangan (teks) sebagai dokumentasi .

2. *Relationship :*

a. Dependency

Hubungan antar elemen dimana perubahan pada elemen yang satu dapat mempengaruhi elemen yang lain *(dependent).*

b. Association

Hubungan struktur antara elemen dan bertindak sebagai link.

c. Generalization

Hubungan dimana elemen yang special *(child)* mewarisi elemen yang umum *(parent).*

d. Realization

1. Hubungan (semantik) antara 2 elemen, dimana elemen yang satu memberikan kontrak dan elemen yang lain menjamin realisasi kontrak tersebut.
2. Dimana elemen yang special *(child)* mewarisi elemen yang umum *(parent).*
3. Diagram *: use case diagram, class diagram, msc diagram*, dan lain–lain.

4. *UML Extensible:* dapat didefinisikan perbendaharaan baru berdasarkan yang telah ada.

Selain cakupan–cakupan di atas, ada tiga aspek utama dalam pemodelan sistem yang mampu didukung oleh UML:

1. *Functional Model*, untuk menunjukkan fungsionalitas dari suatu sistem dari sudut pandang user atau pengguna. Ini dicapai dengan menggunakan *Use Case Diagram*.
2. *Object Model*, untuk menunjukkan struktur dan substruktur dari suatu sistem dengan menggunakan object, atribut, operasi dan juga asosiasi. Ini dicapai dengan menggunakan *Class Diagram.*
3. *Dynamic Model*, menunjukkan *internal behavior* dan suatu sistem. Ini dicapai dengan menggunakan *Sequence Diagram, Activity Diagram* dan juga *Statechart Diagram.*

2.4.5 Perbedaan UML Dengan Bahasa Pemodelan Lain

Perbedaan yang paling mendasar jika kita menggunakan pemograman terstruktur seperti DFD dengan UML adalah, Data Flow Diagram sebagai tools design system berdasarkan procedural dan Unified Modeling Language sebagai tools design OOP. Salah satu kelebihan dari digunakannya Unified Modelling Language adalah kedekatan antara artifak yang sesungguhnya dengan model yang kita buat. Mental model dan cara berpikir kita apabila dibuat sedekat mungkin dengan artifak yang sesungguhnya memudahkan kita untuk berpikir 'objek-objek' apa saja dan interelasi antar objek yang terdapat dalam sebuah model. Sedangkan kekurangannya adalah tidak semua model bisa dilakukan dengan mudah dengan pendekatan ini. Misalnya scientific model, dan semacamnya.

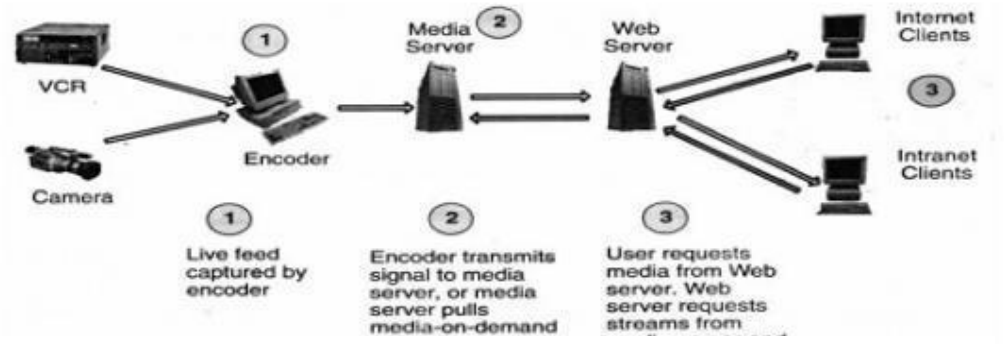
Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan

piranti lunak dalam bahasa pemrograman berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

2.5 Streaming

2.5.1 Pengertian Streaming

Streaming adalah suatu teknologi untuk memainkan file audio atau video secara langsung maupun dengan prerecord dari sebuah mesin server (web server). File audio atau video yang terletak pada sebuah server dapat secara langsung dijalankan pada komputer client sesaat setelah ada permintaan dari pengguna sehingga proses download file tersebut yang menghabiskan waktu cukup lama dapat dihindari. Saat file tersebut di-stream maka akan terbentuk sebuah buffer di komputer client dan data audio atau video tersebut akan mulai di-download ke dalam buffer yang telah terbentuk pada mesin client. Setelah buffer terisi dalam waktu hitungan detik, maka secara otomatis file video ataupun audio akan di jalankan oleh sistem. Sistem akan membaca informasi dari buffer sambil tetap melakukan proses download file sehingga proses streaming tetap berlangsung ke mesin client.



Gambar 2. 3 Konsep Video Streaming

dalam beberapa bagian, mentransmisikan pake tersebut, kemudian pada pihak penerima (client) dapat men-decode dan memainkan potongan paket file video tanpa harus menunggu seluruh file terkirim ke mesin penerima. Video

atau audio dapat diencode untuk keperluan komunikasi secara real time atau dapat juga di pre-encoded dan disimpan dalam format CD-DVD untuk dijalankan pada saat dibutuhkan. Salah satu aplikasi yang membutuhkan real time encoding adalah videophone dan video conferencing. Sedangkan aplikasi yang membutuhkan pre-encoded antara lain DVD, VCD, yang dikenal dengan penyimpanan secara local atau Video on Demand (VoD), yang penyimpanannya dilakukan secara remote di server yang dikenal dengan video streaming.

Sebuah file video yang akan ditampilkan di user dapat menggunakan dua metode transfer file. Pertama, dengan mendownload file video tersebut dan yang kedua dengan melakukan proses streaming. Kedua metode ini memiliki keunggulan dan kekurangannya masing-masing. Sebuah file video yang diambil dari server dengan cara download tidak dapaat ditampilkan dalam sebuah file video tersebut selesai tersalin ke buffer. Metode ini memerlukan media penyimpanan yang cukup besar dan waktu yang diperlukan untuk proses download cukup lama karena file video biasanya berukuran besar. Metode kedua yang dapat digunakan adalah proses streaming. Metode ini berusaha untuk mengatasi masalah yang terdapat dalam metode download. Ide dasar dari video streaming ini adalah membagi paket video menjadi beberapa bagian, mentransmisikan paket data tersebut, kemudian penerima (receiver) dapat mendecode dan memainkan potongan paket video tersebut tanpa harus menunggu keseluruhan file selesai terkirim ke mesin penerima.

Secara garis besar, konsep video streaming di bagi ke dalam tiga tahap, antara lain:

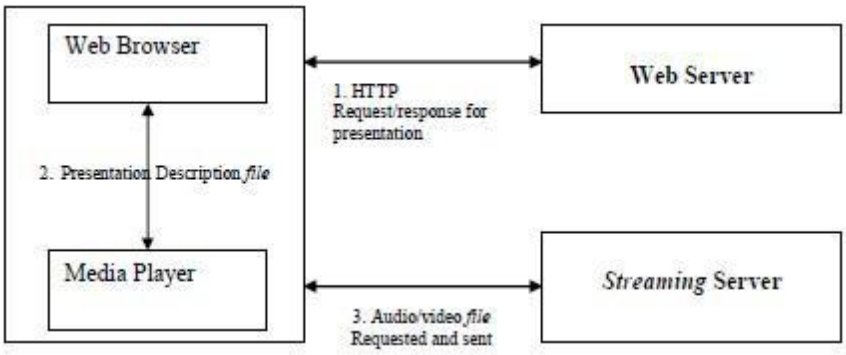
1. Mempartisi atau membagi data video yang telah terkompresi ke dalam paket - paket data.
2. Pengiriman paket - paket data video.
3. Pihak penerima (client) mulai men-decode dan menjalankan video walaupun paket data yang lain masih dalam proses pengiriman ke PC client.

2.5.2 Format Streaming

Hingga kini setidaknya ada tiga jenis format streaming yang banyak digunakan di situs-situs Internet. Format tersebut adalah keluaran Real Media (.rm / .ra / .ram), Windows Media (.asf / .wmf / .asx) dan QuickTime (.mov). Masing-masing format tersebut memiliki kekurangan dan kelebihan sendiri-sendiri. Format Real Media dan Windows Media sangat handal di proses streaming, tetapi tidak terlalu handal untuk proses editing dan playback lokal. Sedangkan format QuickTime rata-rata cukup handal untuk proses streaming, editing dan playback lokal.

2.5.3 Arsitektur Streaming

Sistem streaming tersusun dari kombinasi server , player, transmisi dan metode encoding yang digunakan. Berikut ini bagian hubungan setiap komponen penyususun sistem streaming. Untuk framework aplikasi video streaming protokol pada layer network akan menggunakan model *IP (Internet Protokol),* sedangkan unutk layer transpot protokol yang dipakai merupakan gabungan antara TCP (Transport Control Protokol) dan UDP *(User Datagram Protocol).*



Gambar 2. 4 Arsitektur Streaming

2.5.4 Metode Streaming

Dalam proses streaming terdapat beberapa metode, antara lain:

2.5.4.1 *Broadcast*

Broadcast adalah bentuk komunikasi one to many (yang pada dasarnya adalah *one to all*) yang paling dikenal, salah satu contohnya adalah siaran TV. Dengan menggunakan *broadcast* maka informasi yang akan dikirim berasal

dari satu sumber atau titik kepada semua penerima yang tergabung dalam jaringan. Pada *broadcast* semua penerima mau tidak mau akan menerrima informasi ini.

2.5.4.2 *Multicast*

Multicast juga merupakan bentuk komunikasi *one to many*, tetapi tidak seperti broadcast. Pada multicast informasi yang dikirimkan berasal dari satu sumber atau titik kepada semua penerima yang menginginkan informasi tersebut. *Server* akan membuat *stream* satu kali kemudian *stream* ini diduplikasi dan dikirimkan ke setiap *client.* Setiap *client* akan menerima *stream* yang sama dengan *client* lainnya. Salah satu contoh penggunaan *multicast* adalah *live video*.

2.5.4.3 *Unicast*

Unicast merupakan bentuk komunikasi *one to one* atau *point to point*. Pada *unicast* informasi yang dikirimkan berasal dari satu sumber atau titik ke satu titik lainnya. *Server* akan mengirimkan file *streaming* ke komputer *client* berkali-kali bergantung pada banyaknya jumlah permintaan. Setiap *client* akan menerima file *streaming* yang terpisah dari *client* lainnya. Contoh penggunaan *unicast* adalah *video on demand*.

2.5.5 Komponen-komponen Dalam Streaming Media

Streaming media merupakan suatu sistem yang terdiri dari komponen-komponen yang saling mendukung. Komponen-komponen yang diperlukan untuk melakukan streaming media adalah sebagai berikut:

1. Media Source

   Media Source yaitu sumber yang akan menampilkan suatu content presentasi. Media source dapat beruba sumber yang sifatnya live, seperti kamera video dan microphone.

2. Encoder

Encoder adalah program yang digunakan untuk mengubah media source ke format yang sesuai untuk streaming. Biasanya memiliki kompresi yang cukup tinggi untuk mengatasi keterbatsan bandwidth jaringan.

3. Media Srver

   Media server digunakan untuk mendistribusikan video streaming ke client. Juga bertanggungjawab untuk mencatat semua aktivitas streaming, yang nantinya digunakan untuk statistik.

4. Player

   Player dibutuhkan untuk menampilkan atau mempresentasikan content multimedia (data stream) yang diterima dari media server. File-file khusus yang disebut metafile digunakan untuk mengaktifkan player dari halaman web. Metafile berisi keterangan dari content multimedia. Browser web mendownload dan meneruskan ke player yang tepat untuk mempresentasikannya. Selain itu, juga berfungsi untuk melakukan dekompresi.

2.5.6 Masalah Streaming Multimedia

Dalam melakukan streaming multimedia, untuk menghasilkan presentasi yang baik seringkali timbul kendala. Kendala-kendala yang dapat terajadi dalam melakukan streaming multimedia adalah sebagai berikut:

1. Bandwidth

   Bandwidth sangat berpengaruh terhadap kualitas presentasi suatau data stream. Disamping kondisi jaringan junga mempengaruhi bandwidth, hal yang perlu diperhatikan adalah ukuran data stream harus sesuai dengan kapasitas bandwidth jaringan. Untuk mengatasinya digunakan kompresi data penggunaan buffer.

2. Delay

   Delay merupakan waktu tunda antara data terkirim dengan data diterima. Delay selain diakibatkan oleh sistem yang digunakan juga bisa disebabkan oleh gangguan pada transmisi dan juga bisa disebabkan oleh kesalahan pada bias. Yang merupakan delay yang disebabkan oleh sistem antara lain

adalah proses encoder, masuknya sinyal data ke server sebelum diteruskan ke client, dan juga karena masuknya ke buffer streaming selama beberapa detik.

3. Sinkronisasi

   Agar media yang berbeda sampai dan dipresentasikan pada user seperti aslinya, maka media tersebut harus tersinkronisasikan sesuai dengan timeline presentasi tersebut dan delay seminimal mungkin. Adanya kerugian sinkronisasi dan delay dapat disebabkan oleh kondisi jaringan yang buruk, sehingga mengakibatkan timeline presentasi menjdi kacau.

4. Interoperability

   Idealnya adalah presentasi yang kita buat harus dapat dimainkan oelh semua jenis client, CPU yang berbeda, sistem operasi yang berbeda, media player lainnya dan smartphone yang digunakan.

Khusus untuk masalah yang berhubungan dengan kondisi bandwidth jaringan dapat diatasi dengan menerapkan teknologi HSDPA atau 3G. Teknologi ini dapat menyediakan lintasan data dengan kecepatan transfer yang tinggi. Dengan menerapkan teknologi 3G diharapkan mampu menyelenggarakan live vide streaming dengan kualitas yang baik sekaliguas menjadi alternative untuk melakukan berbagai macam aplikasi seperti TV on demand, live TV, maupun belajar jarah jauh (e-learning).

2.5.7 Protokol Streaming

Protokol merupakan sebuah format yang distandarkan untuk mentransmisikan data antara dua buah device. Beberapa jenis protokol yang digunakan dalam teknologi streaming, antara lain:

a. MMS (Microsoft Media Service) Protocol

Protokol MMS merupakan protokol streaming yang dikembangkan oleh Microsoft. Protocol MMS digunakan untuk mengirimkan informasi secara unicast pada Windows Media Service.

b. RTSP (Real Time Streaming Protocol)

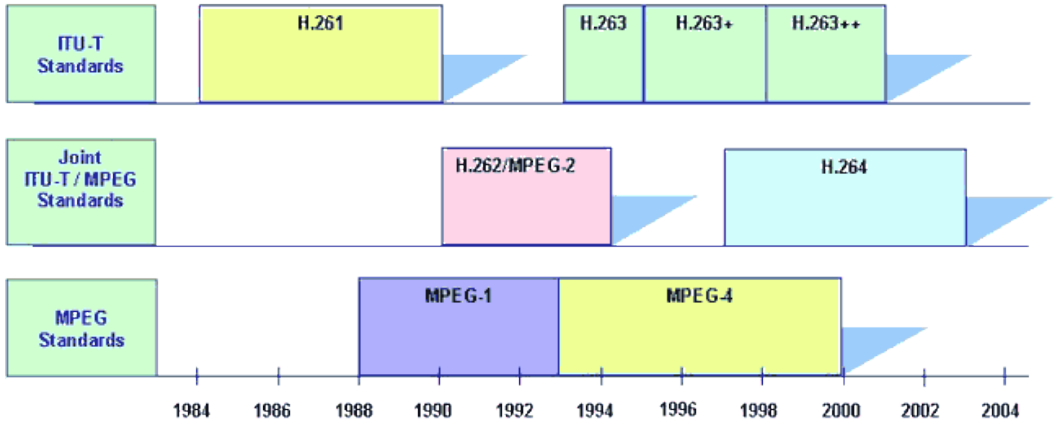
Protokol RTSP merupakan protokol yang digunakan untuk mengontrol paket data secara real time. RTSP tidak mengatur masalah transmisi pengiriman paket data, biasanya server yang menggunakan RTSP memakai RTP (Real time Transport Protocol) sebagai protokol untuk pengiriman paket data.

c. HTTP (Hypertext Transport Protocol)

Protokol HTTP dapat digunakan untuk pengiriman paket data dalam layanan streaming. Protokol HTTP sangat berguna untuk pengiriman data melewati firewall, sebab protokol HTTP menggunakan port 80, dimana kebanyakan firewall tidak mem-block port ini.

2.6 Standar Kompresi Video

Standardisasi terhadap kompresi informasi audio-visual diperlukan untuk memfasilitasi pertukaran data berupa video digital secara global. Sebuah standardisasi pengkodean dikatakan efisien bila mendukung algoritma kompresi yang baik dan mengimplementasikan disain enkoder dan dekoder yang efisien. Untuk komunikasi multimedia, terdapat dua organisasi standard yang utamayaitu ITU-T dan International Organization for Standardization (ISO). Selama beberapa dekade belakangan ini, sejumlah standard dari ITU-T dan ISO, seperti MPEG-1, MPEG-2, MPEG-4, H.261, H.263 dan H.264 telah dikembangkan untuk banyak domain aplikasi. Standard-standard tersebut mendefinisikan bitstream dari data audio-visual dan menentukan sekumpulan aturan yang harus dipatuhi dalam pengembangan hardware maupun software untuk solusi kompresi.



Gambar 2. 5 Perkembangan Standar Kompresi Video

Standardisasi kompresi video tidak dengan tegas menentukan proses pengkodean, tetapi mengambil kelebihan dari riset-riset dalam bidang teknik kompresi yang banyak dilakukan dan merekomendasikan sejumlah agoritma untuk memperoleh teknik kompresi dan dekompresi yang efisien.

Tabel 2. 1 Perbandingan Standar Kompresi Video

	H.261	H.263	H.264	MPEG-1	MPEG-2	MPEG-4
Format	CIF/QCIF 29.97 fps (max)	Fleksibel (pada umumnya QCIF)	Fleksibel	SIF 30 fps	Fleksibel	Fleksibel
Bit Rate Kompresi	*px*64kbps *p=1,2,...,30*	<28.8kbps	Fleksibel	1.5 Mbps	>2 Mbps	Fleksibel
Aplikasi	Videophone/ Videoconfer encing	Videophone	Multimedia Web Video	Video Cassette Recorder	Berbagai macam aplikasi video	Multimedia Web Video
Transport	N-ISDN	Kabel Telpon dan Wireless	Berbagai macam media	ADSL	DVD dan Digital TV Boradcast	Berbagai macam media

2. 7 Android

2.7.1 Sejarah Android

Android adalah sebuah sistem operasi untuk perangkat mobile bebasis linux yang mencakup sistem operasi, middleware dan aplikasi. Android menyediakan platform yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, google Inc membeli Android Inc. yang merupakan pendatang baru yang membuat piranti lunak untuk ponsel/smartphone. Kemudian untuk mengembangkan android dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan piranti lunak, piranti keras 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 pengembangkan open source pada perangkat mobile. Di lain pihak, google meilis kode-kode

android dibawah lisensi Apache, sebuah aplikasi lisensi perangkat lunak dan open platform perangkat seluler.



Gambar 2. 6 Maskot Android

2.7.2 Arsitektur Android

Secara garis besar arsitektur android dapat dijelaskan dan digambarkan sebagai berikut:

1. Applications dan Widgets

Applications dan Widgets ini adalah layer dimana pengguna berhubungan dengan aplikasi saja, dimana biasanya pengguna download aplikasi kemudian pengguna lakukan instalasi dan jalankan aplikasi tersebut. Di layer terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak dan lain-lain. Semua aplikasi dikembangkan menggunakan bahasa pemrograman Java.

2. Applications Framework

Android adalah Open development Platform yaitu android menawarkan kepada pengembang atau member kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi resources, menjalankan service background, mengatur alarm dan menambahkan status notifications dan sebagainya. Pengembang memiliki akses penuh menuju API Framework seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan.

Sehingga bisa disimpulkan applications frameworks ini adalah layer dimana para pengembang aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi android, karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti content providers yang berupa SMS dan panggilan telepon.

Komponen-komponen yang termasuk didalam applications frameworks adalah sebagai berikut:

a. Views
b. Content Provider
c. Resource Manager
d. Notification Manager
e. Activity Manager

3. Libraries

Libraries adalah layer dimana firut-fitur android berada, biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya. Berjalan diatas kernel, layer ini meliputi berbagai library C/C++ inti seperti Libc dan SSL, serta :

1. libraries media untuk pemutaran media audio dan video
2. libraries untuk manajemen tampilan
3. libraries graphics mencakup SGL dan OpenGL untuk grafis 2D dan 3D
4. libraries SQLite untuk dukungan database
5. libraries SSL dan WebKit terintegrasi dengan web browser dan security
6. libraries LiveWebcore mencakup modern web browser dengan engine embedded wev view
7. libraries 3D yang mencakup implementasi OpenGL ES 1.0 API’s

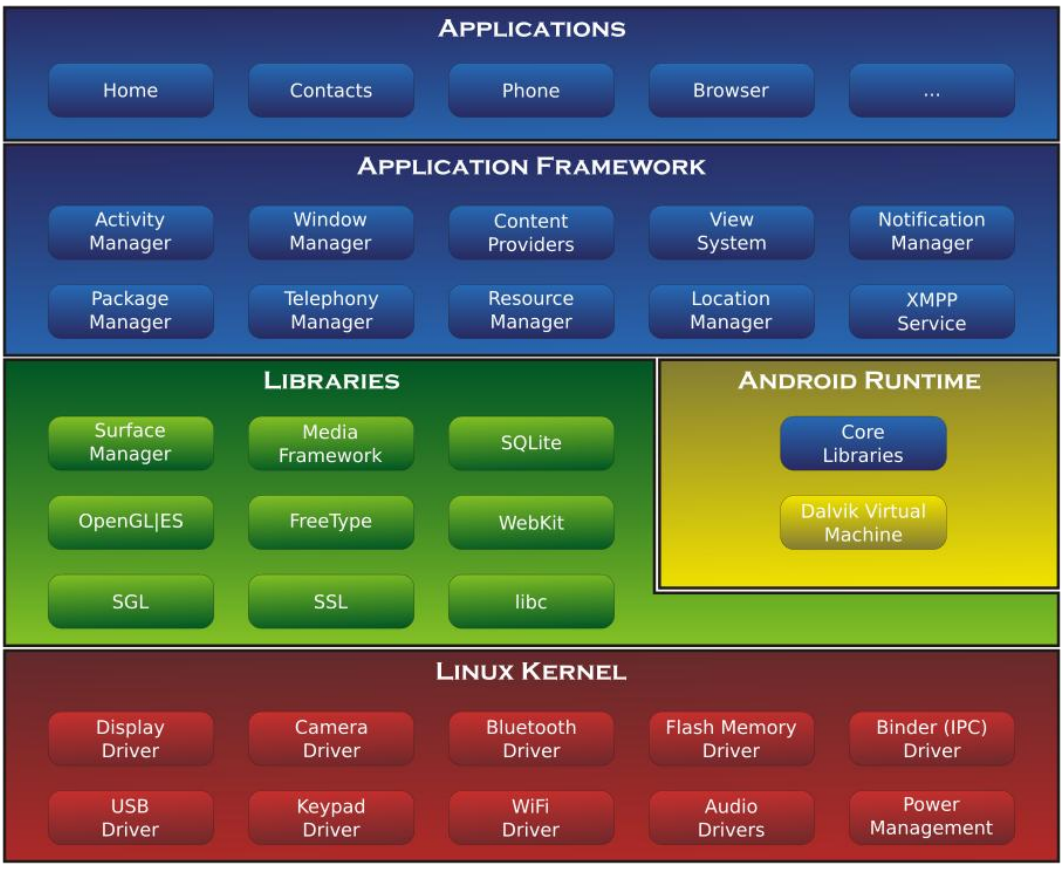
4. Android Run Time

Layer yang membuat aplikasi android dapat dijalankan dimana dalam prosesnya menggunakan implementasi linux. Dalvik Virtual Machine (DVM) merupakan mesin yang membaut dasar kerangka aplikasi android. Didalam android run time dibagi kedalam dua bagian yaitu:

1. Core Libraries: Aplikasi android dibangun dalam bahasa java, sementara Dalvik sebagai virtual mesinnya bukan Virtual Machine Java, sehingga diperlukan sebuah libraries yang berfungsi untuk menterjemahkan bahasa java/c yang ditangani oleh Core Libraries.
2. Dalvik Virtual Machine: Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efiesien, dimana merupakan pengembangan yang mampu membuat linux kernel untuk menjalankan threading dan manajemen tingkat rendah.

5. Linux Kernel

Linuk kernel adalah layer dimana inti dari operating sistem dari android itu berada. Berisi file-file sistem yang mengatur sistem processing, memory, resource, drivers dan sistem-sistem android lainnya. Linux kernel yang digunakan android adalah linux kernel release 2.6.



Gambar 2. 7 Arsitektur Android

2.7.3 Versi Android

Telepon pertama yang memakai sistem operasi android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2010 diperkirakan hampir semua vendor seluler di dunia menggunakan android sebagai *operating system*. Android memakai berbagai *codename* makanan agar mudah diingat setiap versinya. Adapun versi-versi android yang pernah dirilis adalah sebagai berikut:

Tabel 2. 2 Versi Android

Android	Code Name	Tanggal Peluncuran
Android versi 1.1	-	09 Maret 2009
Android versi 1.5	Cupcake	30 April 2009
Android versi 1.6	Donut	15 September 2009
Android versi 2.0/2.1	Eclair	26 Oktober 2009
Android versi 2.2	Frozen Yoghurt (Froyo)	20 Mei 2010
Android versi 2.3	Gingerbread	06 Desember 2010
Android versi 3.0/3.1	Honeycomb	22 Februari 2011
Android versi 4.0	Ice Cream Sandwich (ICS)	19 Oktober 2011
Android versi 4.1/4.2	Jellybean	09 Juli 2012
Android versi 4.4	Kitkat	31 Oktober 2013

3.7.4 Kelebihan Android

Berbagai *platform* seluler muncul pada saat ini, namun ada beberapa hal yang menjadi kelebihan android dibandingkan dengan *platform* lain walaupun beberapa fitur-fitur yang ada telah muncul sebelumnya pada *platform* lain, android adalah yang pertama menggabungkan hal seperti berikut:

1. Keterbukaan, Bebas pengembangan tanpa dikenakan biaya terhadap system karena berbasiskan Linux dan *open source*. Pembuat perangkat menyukai hal ini karena dapat membangun *platform* yang sesuai yang diinginkan tanpa harus membayar royality. Sementara pengembang software

menyukai karena Android dapat digunakan diperangkat manapun dan tanpa terikat oleh vendor manapun.

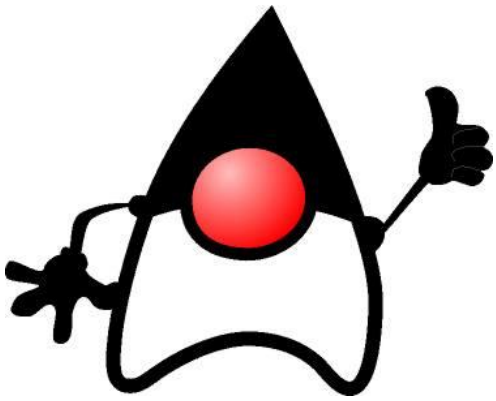
2. Arsitektur komponen dasar Android terinspirasi dari teknologi internet *Mashup*. Bagian dalam sebuah aplikasi dapat digunakan oleh aplikasi lainnya, bahkan dapat diganti dengan komponen lain yang sesuai dengan aplikasi yang dikembangkan.
3. Banyak dukungan service, kemudahan dalam menggunakan berbagai macam layanan pada aplikasi seperti penggunaan layanan pencarian lokasi, database SQL, browser dan penggunaan peta. Semua itu sudah tertanam pada Android sehingga memudahkan dalam pengembangan aplikasi.
4. Siklus hidup aplikasi diatur secara otomatis, setiap program terjaga antara satu sama lain oleh berbagai lapisan keamanan, sehingga kerja system menjadi lebih stabil. Pengguna tak perlu kawatir dalam menggunakan aplikasi pada perangkat yang memorinya terbatas.
5. Dukungan grafis dan suarat terbaik, dengan adanya dukungan 2D grafis dan animasi yang diilhami oleh *Flash* menyatu dalam 3D menggunakan *OpenGL* memungkinkan membuat aplikasi maupun game yang berbeda.
6. Portabilitas aplikasi, aplikasi dapat digunakan pada perangkat yang ada saat ini maupun yang akan datang. Semua program ditulis dengan menggunakan bahas pemrograman Java dan dieksekusi oleh mesin virtual Dalvik, sehingga kode program portabel antara ARM, X86, dan arsitektur lainnya. Sama halnya dengan dukungan masukan seperti penggunaan *Keyboard,* layar sentuh, *trackball* dan resolusi layar semua dapat disesuaikan dengan program.

3.7.5 Kekurangan Android

Ancaman dari malware/virus pada sistem operasi android, karena android adalah open source dan pengguna dapat menginstall aplikasi yang bukan berasal dari Google Play, perangkat android lebih rentan terkena *malware* dan aplikasi bajakan/palsu.

2.8 Java

Bahasa java merupakan karya Sun Microsystem Inc. Rilis resmi beta dilakukan pada November 1995. Dua bulan berikutnya Netscape menjadi perusahaan pertama yang memperokleh lisensi bahasa java dari Sun. Maskot



Gambar 2. 8 Maskot java

Pada tahun *1996, Sun mengeluarkan JSDK (Java Software Development Kit),* kemudian secara berturut-turut mengeluarkan versi yang berbeda, diantaranya sebagai berikut:

1. Versi 1.02 yang mendukung basis data dan objek-objek yang tersebar.
2. Versi 1.1 pada tahun 1997 ditambahkan model kejadian (event model) yang handal, internationalization dan model komponen JavaBeans.
3. Versi 1.2, dirilis pada tahun 1998 mempunyai banyak peningkatan diantaranya user interface toolkit "Swing" yang memungkinkan pemrograman membuat aplikasi berbasis GUI yang sepenuhnya portable. Sejak ini disebut sebagai Java 2.
4. Versi 1.3 dirilis tahun 2000, dengan banyak peningkatan.
5. Versi 1.4 ditambahkan fasilitas asersi untuk dukungan design-by-contact.
6. Versi 1.5 ditambahkan fitur-fitur baru di level bahasa diantaranya generics (parameterized types), enumeration dan metadata. Versi ini disebut JDK 5.
7. Versi 1.6 ditambahkan lightweigh database system yaitu derby. Derby merupakan hasil pengembangan dari proyek basis data Apache. Derby mulanya merupakan CloudScape dari IBM.

2.8.1 Sejarah Java

Pada tahun 1991, sekelompok insinyur Sun Microsystem Inc., dipimpin oleh Patrick Naughton dan James Gosling merancang bahasa untuk perangkat consumer seperti cable TV box. Karena perangkat tidak mempunyai banyak memori, bahasa harus berukuran kecil dan menghasilkan kode yang liat. Karena pabrikan-pabrikan berbeda memilih pemroses-pemroses berbeda maka bahasa harus netral arsitekturnya manapun. Proyek ini diberi nama kode "Green".

Kebutuhan untuk kecil, lihat dan netral platform mengantar tim mempelajari implementasi Pascal yang pernah dicoba. Niklaus Wirth, pencipta Pascal telah merancang bahasa kode antara (intermediate code) portabel untuk mesin hipotesis. Kode antara ini kemudian dapat digunakan di sembarang mesin yang memiliki interpreter. Proyek Green menggunakan mesin maya untuk mengatasi isu netral terhadapt arsitektur mesim.

Karena orang-orang di proyek green berbasis C++ bukan Pascal maka kebanyakan sintaks diambil dari C++ serta mengadopsi orientasi objek bukan procedural, Mulainya bahasa yang diciptakan diberi nama "Oak", kemudian diganti "Java" karena telah ada bahasa yang bernama "Oak".

Produk pertama proyek Green adalah *7, sebuah kendali jauh yang cerdas. Karena pasar belum tertarik dengan produk consumer cerdas maka proyek Green harus menemukan pasar lain. Penerapan mengarah menjadi teknologi di web. Pada tahun 1995, NetScape memutuskan membuat browser yang dilengkapi Java. Setelah itu diikuti IBM, Symantec bahkan Microsoft.

Setelah itu gaung Java mulai terdengar. Berbagai industri meliriknya. Dengan strategi terbuka, banyak industri menyambutnya. Berbagai universitas Amerika, Jepang dan Eropa mengubah pengenalan bahasa pemograman menjadi Java meninggalkan C++. Java lebih sederhana dan mengakomodasikan hampir seluruh fitur penting bahasa-bahasa pemograman yang diciptakan manusia.

2.8.2 Kelebihan Java

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, Mac OS, dan Sun Solari. Penyebabnya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs Java) untuk menginterpretasikan bytecode tersebut.

2. **OOP (Object Oriented Programming)**

   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 (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 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).

2.8.3 Kekurangan Java

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 di 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/direverse-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.

2.9 Tools Implementasi

Perangkat lunak (software) yang digunakan untuk membangun aplikasi video streaming ini diantaranya:

2.9.1 Android SDK

Menurut Safaat, 2011, Android SDK tools API (application Programming Interface) yang diperlukan untuk mulai mengembangkan aplikasi pada platform android menggunakan bahasa pemograman Java. Saat ini disediakan SDK sebagai alat bantu dan API untuk mulai mengembangkan aplikasi.

2.9.2 Eclipse

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

1. Multi-platform: Dapat berjalan pada berbagai platform seperti, 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, Phyton, 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 yang difavoritkan karena gratis dan open source, yang berarti setiap orang boleh melihat kode program perangkat lunak. Selain itu, kelebihan dari Eclipse yang

membuatnya popular adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponan yang dinamakan plug-in.

2.9.2.1 Sejarah Eclipse

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak IBM Visual Age for Java 4.0. Produk ini diluncurkan oleh IBM pada tanggal, 5 november 2001, yang menginvestasikan sebanyak US$40 juta untuk pengembangannya. Semenjak itu konsursium Eclipse Foundation mengambil alih untuk pengembangan lebih lanjut dan pengaturan organisasinya.

2.9.2.2 Arsitektur Eclipse

Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah kernel, yang mengangkut plug-in. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari plug-in yang sudah diinstal. Ini merupakan basis dari Eclipse yang dinamakan Rich Client Platform (RCP). Berikut ini adalah komponen yang membentuk RCP :

1. Core Platform
2. OSGi
3. SWT (Standard Widget Toolkit)
4. JFace
5. Eclipse Workbench

Standard pengoperasian Eclipse selalu dilengkapi dengan JDT *(Java Development Tools)*, *plug-in* yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE *(Plug-in Development Environment)* untuk mengembangkan *plug-in* baru. Eclipse beserta *plug-in* nya diimplementasikan dalam bahasa pemrograman Java.

Konsep Eclipse adalah IDE yang terbuka (open), mudah diperluas (extensible) untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi, Eclipse tidak saja mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal plug-in yang dibutuhkan. Apabila ingin mengembangkan program C/C++ terdapat

plug-in CDT (C/C++ Development Tools). 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.

2.9.2.3 Perkembangan Eclipse

Sejak tahun 2006, Eclipse Foundation mengkoordinasikan peluncuran Eclipse secara rutin dan simultan yang dikenal dengan nama Simultaneous Release. Setiap versi peluncuran terdiri dari Eclipse Platform dan juga sejumlah proyek yang terlibat dalam proyek Eclipse. Tujuan dari sistem ini adalah untuk menyediakan distribusi Eclipse dengan fitur-fitur dan versi yang terstandarisasi. Hal ini juga dimaksudkan untuk mempermudah pengembangan (deployment) dan perawatan (maintenance) untuk sistem enterprise, serta untuk kenyamanan. Peluncuran simultan dijadwalkan pada bulan Juni setiap tahunnya.

Tabel 2. 3 Perkembangan Eclipse

Kode Peluncuran	**Tanggal Peluncuran**	**Platform**	**Nama Proyek**
Eclipse 3.0	21 Juni 2004	3.0	-
Eclipse 3.1	28 Juni 2005	3.1	-
Callisto	30 Juni 2006	3.2	Callisto projects
Europa	29 Juni 2007	3.3	Europa projects
Ganymede	25 Juni 2008	3.4	Ganymede projects
Galileo	24 Juni 2009	3.5	Galileo projects

Helios	23 Juni 2010	3.6	Indigo projects
Juno	27 Juni 2012	4.2	Juno projects