# BAB I
# PENDAHULUAN

## 1.1 Latar Belakang Masalah

Perkembangan ilmu pengetahuan yang semakin pesat mendorong lahirnya berbagai macam teknologi yang membawa dampak positif bagi kehidupan manusia. Telah banyak penemuan teknologi yang membantu kehidupan manusia sehingga berbagai jenis objek pekerjaan dapat terselesaikan dengan cepat, semakin teratur dan sistematis. Dalam hal ini kemajuan teknologi komputer dan informasi memegang peranan yang sangat penting dalam mewujudkan hal tersebut, dari mulai perindustrian, bisnis, jasa, bahkan multimedia. Mungkin beberapa tahun yang lalu tidak dapat terbayangkan bagaimana kita dapat berkomunikasi dan mendapatkan informasi ataupun hiburan secara cepat dengan jarak yang sangat jauh dari berbagai belahan dunia.

Pada saat ini hal tersebut sudah dapat terealisasikan melalui internet. Hanya dengan menghubungkan kabel telepon dan komputer, dengan perantara modem atau *wifi*, kita sudah dapat *surfing* di internet mendapatkan informasi yang kita butuhkan kapanpun kita inginkan. Sayangnya sebagian besar media informasi di internet (*website*) hanya menyediakan layanan dalam bentuk gambar atau tulisan tidak dalam bentuk video. Beberapa informasi atau hiburan misalnya olah raga, berita, video klip ataupun lainnya akan lebih nyaman jika disampaikan tidak dalam bentuk tulisan melainkan 'tayangan'. Sebetulnya informasi dalam bentuk pemutaran audio visual tersebut dapat diwujudkan melalui tv satelit, namun bagaimana jika kita menginginkan program acara dengan waktu yang dapat kita tentukan sendiri, rasanya dengan tv satelit akan sulit karena hierarki jaringan tv satelit hanya satu arah, sehingga sangat sulit melakukan komunikasi dua arah dan secara *online*. Untuk merealisasikan hal tersebut internet merupakan salah satu alternatif sebagai mediator yang paling memungkinkan. Internet menyediakan komunikasi dua arah lewat tulisan (halaman *web*) sehingga kapan saja bisa memilih suatu acara yang kita inginkan lalu menayangkan dalam bentuk *audio visual* ("*Video On Demand*").

Berdasarkan hal tersebut penulis merasa tertarik untuk membuat suatu media informasi di web yang menyediakan layanan tidak hanya dalam bentuk tulisan melainkan juga dalam bentuk penayangan video. Melalui aplikasi ini *user* akan mendapatkan informasi yang diinginkan kapanpun mereka perlukan. Selain itu, dalam aplikasi ini *user* yang telah menjadi *acount* di beri kesempatan untuk menyampaikan informasi yang mereka miliki untuk ditampilkan dalam web serta mempromosikan produk, usaha ataupun jasa yang mereka miliki.

Dalam menyediakan informasi dalam bentuk video diperlukan suatu proses pengiriman data sehingga informasi dapat diterima secara baik oleh *user*, jika *file* berukuran kecil pengiriman data melalui proses *download* akan cepat dan tidak akan mengalami masalah, namun rata-rata *file* video berukuran besar sehingga akan memakan waktu yang lama, selain itu informasi setiap waktu pasti berubah sehingga akan tidak efisien (memakan *space hardisk*) jika dilakukan melalui proses *download*. Proses *download file* juga akan menimbulkan masalah berupa pelanggaran hak cipta jika *user* menggandakan *file* tersebut untuk diperjual belikan secara ilegal. Berdasarkan beberapa paparan diatas penulis merasa tertarik dan akan mengangkat tema tugas akhir tentang "Video On Demand Melalui Web Dengan Menggunakan Proses Streaming (Studi kasus penyampain kebutuhan informasi baik tulisan maupun audio visual melalui web)".

## 1.2 Rumusan Masalah.

Berdasarkan latar belakang di atas penulis mencoba untuk merumuskan beberapa masalah yaitu sebagai berikut :

1. Bagaimana membuat suatu media yang dapat digunakan sebagai penyaluran informasi dalam bentuk video yang tidak terikat dengan waktu.
2. Teknologi apa yang tepat untuk mentransfer *file* berukuran besar seperti *file* video agar dapat disampaikan dengan cepat namun tidak tersimpan di *hardisk user* (jika tersimpan akan tidak efisien karena informasi setiap waktu pasti berkembang/berubah dan akan menimbulkan pelanggaran hak cipta jika diperbanyak tanpa izin).

## 1.3 Maksud dan Tujuan.

Tujuan dari tugas akhir yang berjudul "Video On Demand Melalui Web Browser Dengan Menggunakan Proses Streaming (Studi kasus penyampain kebutuhan informasi audio visual dalam web)" untuk memenuhi salah satu persyaratan kelulusan program S1 jurusan Teknik Informatika di Univesitas Widyatama. Adapun tujuan khususnya adalah sebagai berikut :

1. Menyediakan media yang dapat digunakan oleh *user* sehingga dapat memenuhi kebutuhan informasi tidak hanya dalam bentuk tulisan namun juga dalam bentuk video, dengan cara seperti ini *user* dapat memperoleh tayangan informasi yang lebih lengkap dan lebih tersampaikan dengan waktu yang tidak terikat.
2. Mencari dan mempelajari teknologi pengiriman data selain *download* yang memungkinkan sehingga dapat digunakan secara cepat dan efisien untuk proses pengiriman data video.

## 1.4 Batasan Masalah.

Banyaknya hal-hal yang berkaitan dengan pembuatan tugas akhir ini dapat menimbulkan pembahasan menjadi luas. Untuk itu diperlukan batasan-batasan masalah yang mencegah terjadinya hal tersebut. Penulis akan menentukan batasan masalah pembahasan yaitu sebagai berikut:

1. Tidak membahas secara mendalam mengenai jaringan yang terkait, serta perhitungan kecepatan pengiriman data.
2. Aplikasi ini berbasis *web*, untuk teknik pengiriman data akan digunakan komponen-komponen pendukung sesuai dengan kebutuhan yang diperlukan dalam proses pengiriman data tersebut.

## 1.5 Metode Penelitian.

Dalam pengembangan aplikasi, penulis menggunakan paradigma teknologi berorientasi objek dengan metoda *Unified Software Development Process* (USDP) sebagai metoda pengembangan perangkat lunak. dimana tahapan pengembangan sistemnya adalah sebagai berikut :

1. *Requirement* sistem yaitu mencari tahu kebutuhan sistem dengan langkah-langkah berikut :
   a) Studi observasi, yaitu peninjauan secara langsung tentang proses pengiriman data *streaming* sehingga dapat di akses melalui halaman web.
   b) *Interview* (wawancara), yaitu dengan melakukan wawancara langsung dengan berbagai pihak yang memahami tentang teknologi web dan pemrosesan data *streaming*.
   c) Studi pustaka, yaitu mempelajari buku-buku serta referensi-referensi yang berkaitan dengan pembuatan aplikasi video on demand melalui web browser dengan teknologi *streaming*.
2. *Analysis*, data-data yang telah di peroleh kemudian di analisa untuk mengetahui kebutuhan sistem kemudian menentukan objek-objek yang diperlukan.
3. *Design*, tahapan ini dimulai dari perancangan arsitektur sistem, proses, antar muka, dan interaksi sistem dengan pengguna.
4. *Implementation*, Hasil rancangan yang telah dibuat kemudian direalisasikan kedalam kode program yang siap digunakan.
5. *Test*, setelah selesai maka dilakukan serangkaian tes untuk menjamin bahwa sistem dapat berjalan dengan baik.

## 1.6 Sistematika Penulisan.

Penulisan ini terbagi menjadi beberapa bab berdasarkan pada urutan pembahasannya, dimana masing-masing bab memiliki keterhubungan satu dengan yang lainnya, yang terdiri dari :

**Bab satu pendahuluan**, membahas mengenai latar belakang masalah, rumusan masalah, maksud dan tujuan, batasan masalah, metode penelitian dan sistematika penulisan.

**Bab dua landasan teori**, membahas tentang teori-teori yang digunakan penulis untuk membangun aplikasi dan membahas secara singkat mengenai bahasa pemrograman yang digunakan dalam pengimplementasian aplikasi yang akan dibangun.

**Bab tiga analisis,** pada bab ini berisi tentang penganalisaan terhadap kebutuhan aplikasi dalam suatu proses sistem yang telah ada.

**Bab empat perancangan,** berisi mengenai perancangan sistem berdasarkan analisa yang telah didapatkan untuk memenuhi kebutuhan aplikasi yang akan dibangun.

**Bab lima implementasi,** bab implementasi ini akan menerjemahkan hasil analisis dan perancangan yang telah dibuat kedalam bentuk program Video On Demand Melalui Web Browser Dengan Menggunakan Proses Streaming.

**Bab enam penutup,** merupakan hasil kesimpulan dan penyelesaian masalah yang dihadapi serta saran-saran dari penulis.

# BAB II
# DASAR TEORI

Informasi merupakan suatu kebutuhan manusia yang sangat penting untuk menunjang berbagai jenis objek kehidupan. Saat ini kita bisa mendapatkan informasi dari berbagai jenis media seperti cetak, elektronik dan virtual. Saat ini penyedia layanan informasi dalam media virtual (internet) dinilai lebih baik dibandingkan kedua media penyedia layanan informasi lainnya, dikarenakan media ini merupakan media tanpa batas, bisa di-akses kapanpun dan dimanapun kita berada.

Dalam membuat suatu aplikasi di web kita memerlukan bahasa pemrograman web untuk merealisasikan hal tersebut. Untuk tugas akhir ini penulis menggunakan bahasa pemrograman PHP, PHP sengaja dipilih karena berbagai kelebihannya sebagai bahasa pemrograman untuk pengembangan *web*. Selain itu dalam aplikasi ini penulis menyediakan informasi tidak hanya dalam bentuk tulisan saja namun juga dalam bentuk penayangan video yang dilakukan dengan teknologi *streaming*. Berikut akan dibahas mengenai beberapa teori mulai dari jaringan internet, bahasa pemrograman, konsep pengiriman data video dengan teknologi *streaming* dan juga pembahasan teori lainnya yang berhubungan dengan tugas akhir ini.

## 2.1 Internet[2]

*Internet* berasal dari kata *interconnection networking* yang mempunyai arti hubungan berbagai komputer dan beberapa tipe komputer yang membentuk sistem jaringan yang mencakup seluruh dunia (jaringan global) melalui jalur telekomunikasi seperti telepon, *wireless* dan yang lainnya. WWW adalah jaringan beribu-ribu komputer yang dikategorikan menjadi dua yaitu *client* dan *server*.

*Client* dan *server* dengan menggunakan *software* khusus membentuk sebuah jaringan yang disebut jaringan *client-server*. Dalam cara kerja dari www ada dua hal yang terpenting yaitu *software web server* dan *software web browser*. *Server* menyimpan/menyediakan informasi dan memproses permintaan dari *client*, apabila ada *client* yang meminta informasi maka *server* mengirimkannya.

Informasi yang diakses dapat berupa teks, gambar dan suara. *Server* juga mengirimkan perintah-perintah ke *client* tentang bagaimana cara menampilkan semua informasi tersebut. Instalasi tersebut dalam bentuk HTML (*Hyper Text Mark Up Language). Client* membuat permintaan informasi dan kemudian menangani pengaksesan informasi tersebut kepada *end user* (pemakai akhir).

Komunikasi jaringan komputer diatur dengan bahasa/*software* standar yang disebut dengan protokol yang memungkinkan beragam jenis komputer yang berbeda untuk berkomunikasi. Protokol ini secara resmi dikenal sebagai TCP/IP *(Transmission Control Protokol Internet Protokol)* merupakan cara standar untuk mempaketkan dan menyelamatkan data komputer (sinyal elektronik)sehingga data tersebut dapat dikirim ke komputer yang lain.

### 2.1.1 Website

*Website* (situs *web*) adalah merupakan alamat (URL) yang berfungsi sebagai tempat penyimpanan data dan informasi dengan berdasarkan topik tertentu. *Webpage* merupakan halaman khusus dari situs *web* tertentu yang tersimpan dalam bentuk *file*. Dalam *webpage* tersimpan berbagai informasi dan *link* yang menghubungkan satu informasi ke informasi lain baik itu dalam *web page* yang sama ataupun *web page* lain pada *web site* yang berbeda.

*Home page* merupakan halaman pertama atau sampul dari suatu *website* yang biasanya berisi apa dan siapa dari perusahaan atau instansi atau organisasi pemilik *website* tersebut. Jadi pada dasarnya *home page* merupakan sarana dasar untuk memperkenalkan secara singkat tentang apa yang menjadi isi dari keseluruhan *web site* dari suatu organisasi atau pribadi. *Web* adalah fasilitas hiperteks untuk menampilkan data berupa teks, data, suara, animasi dan data multimedia lainnya, yang diantara data tersebut saling berhubungan satu sama lain.

## 2.2 PHP[3]

PHP merupakan *script* untuk pemrograman *script web server-side*, *script* yang membuat dokumen HTML secara *on the fly*, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan

menggunakan editor teks atau editor HTML. Dengan menggunakan PHP maka *maintenance* suatu situs *web* menjadi lebih mudah. Proses *update* data dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan *script* PHP. PHP/FI merupakan nama awal dari PHP. *Personal Home Page Form Interface* dibuat pertama kali oleh Rasmus Lerdoff. PHP awalnya merupakan program CGI yang dikhususkan untuk menerima *input* melalui *form* yang ditampilkan dalam *browser web*. *Software* ini disebarkan dan dilisensikan sebagai perangkat lunak *open source.*

PHP secara resmi merupakan kependekan dari PHP *Hyper Text Preprocessor*, merupakan bahasa *script server*-side yang disisipkan pada HTML. Contoh *script* PHP berbeda dengan *script* yang dituliskan dengan bahasa lain seperti C atau Perl. Pemrogram tidak harus menuliskan semua dokumen HTML sebagai bagian dari keluaran dari *script* PHP, cukup menuliskan bagian mana saja yang berupa *tag* HTML dan bagian mana yang harus ditulis atau dihasilkan dari program *script* PHP diapit dengan menggunakan *tag* awal dan *tag* akhir yang khusus, yang memungkimkan pemrograman untuk masuk dan keluar dari mode *script* PHP.

### 2.2.1 Kemampuan PHP

PHP secara mendasar dapat mengerjakan semua yang dapat dikerjakan oleh program CGI , seperti mendapatkan data dari *form*, menghasilkan isi halaman *web* yang dinamik, dan menerima *cookies*. Kemampuan (*feature*) PHP yang paling diandalkan dan signifikan adalah dukungan kepada banyak *database*. Membuat halaman *web* yang menggunakan data dari *database* dengan sangat mudah dapat dilakukan. *Database* yang dapat mendukung PHP adalah Adabas D, dBase, Empress, FilePro(read only), FrontBase, HyperWave, IBM DB2, Informix, Ingres, InterBase, MSQL, Direct MS SQL, ODBC, Oracle(OCI7 dan OCI8), Ovrimos, PostgreSQL, Solid, SQLite, Sybase, Velocis, Unix DBM.

PHP merupakan bahasa *scripting* yang bisa diandalkan untuk menyelesaikan masalah-masalah yang ada. PHP memiliki beberapa kelebihan yang diantaranya:

- Bersifat *Open Source*, sehingga didukung oleh komunitas *programmer* dari seluruh dunia. Selain itu banyak tersedia program-program PHP yang siap pakai tanpa harus mengeluarkan biaya sedikit pun.
- Dapat berjalan di banyak sistem operasi. Beberapa diantaranya yang terkenal seperti Linux, varian UNIX(HP-UX, Solaris dan OpenBSD), Mac OS, dan Microsoft Windows.
- Dapat bekerja hampir dengan semua aplikasi *database* seperti Oracle, MySQL, Postgre SQL, Microsoft SQL Server, IBM DB2, dan database lainnya.
- Dapat berkomunikasi dengan layanan lainnya melalui protokol-protokol yang tersedia. Protokol yang didukung antara lain, LDAP, IMAP, SNMP, NNTP, POP3, HTTP, dan COM (pada Windows).
- *Text Processing* yang *powerfull*. Dengan POSIX Extended atau Perl *Regular Expressions* untuk mem-*parser* dokumen XML. Untuk mengakses dokumen XML PHP mendukung *standart* DOM dan SAX.
- Tidak hanya menghasilkan kode HTML saja, melainkan dapat menghasilkan gambar, dokumen PDF, dan animasi Flash secara *on-the-fly* dari kode-kode PHP.

Selain kelebihan diatas, beberapa ekstension telah ditulih menambah fungsi PHP seperti, kompresi (Gzip & bz2), IRC Gateway,Calendar Convertion, dan ekstension lainnya.

### 2.2.2 PHP dan *Database*

Salah satu keunggulan dari PHP sebagai bahasa pemrograman *script* adalah banyak fasilitas (librari fungsi) yang memungkinkan untuk mengakses *database*. Kecepatan akses dengan menggunakan *engine/drive* yang khusus untuk setiap *database* merupakan satu kelebihan dan kekurangan. Kelebihannya adalah dari sisi kecepatan tidak dapat disangkal, karena dibuat khusus fungsinya. Kekurangannya adalah karena ketidakseragaman nama fungsi(perintah), sehingga sulit bagi aplikasi yang dihasilkan dikatakan *independent* terhadap *database* yang digunakan.

PHP mendukung ODBC, suatu standar untuk mengakses *database*, akan tetapi belum semua aplikasi yang ada mendukung ODBC yang dibutuhkan oleh PHP. Prosedur standar untuk melakukan operasi akses *database* adalah:

1. Open *database*.
2. Eksekusi SQL
3. Proses *recordset* yang dihasilkan
4. Klose *database*

Proses inti dari manipulasi *database* adalah pada pembangunan perintah SQL yang digunakan untuk melakukan *query, insert, update*, ataupun *delete* data untuk *database*.

### 2.2.3 PHP untuk MySQL

MySQL merupakan *software database* yang termasuk paling populer di lingkungan Linux, kepopuleran ini karena ditunjang oleh performansi *query* dari *database*nya yang saat ini bisa dikatakan paling cepat dan jarang bermasalah. Berangkat dari *software* yang *shareware*, kini MySQL mengeluarkan versi 3.23 yang merupakan *software open source* yang berarti *free*. MySQLdapat digunakan untuk kepentingan komersial ataupun personal (*non profit*).

MySQL bisa berjalan dilingkungan windows, dimana PHP di windows secara *default* telah mendukung penggunaan MySQL untuk pengelolaan datanya.

### 2.2.4 Server MySQL

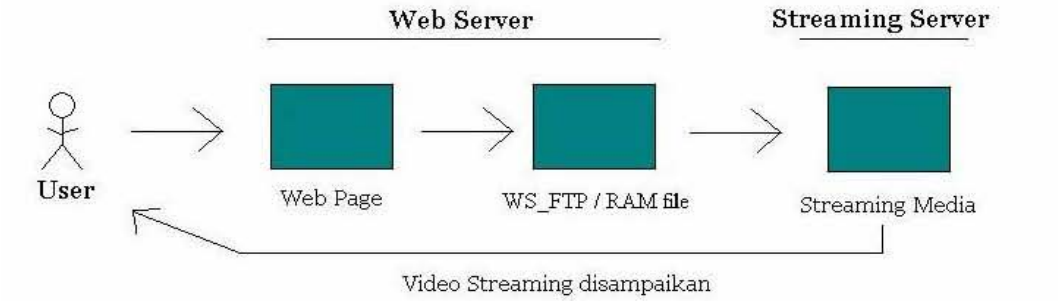
Untuk mengaktifkan MySQL dilingkungan windows maka harus dijalankan *software server* MySQL. *Software server* MySQL dilingkungan windows dibedakan menurut sistem operasi windows yang digunakan, secara umum ada dua yaitu mysqld.exe dan mysqld-nt.exe. Untuk lingkungan Windows NT atau Windows 2000 *software* ini dapat berjalan sebagai *service*, sehingga dapat secara otomatis hidup pada saat *server* dinyalakan.

Jika belum dijalankan maka mysqld sebagai software *server database* harus dijalankan terlebih dahulu. Program ini terletak pada direktori bin pada direktori mysql terpasang, secara default akan terinstal pada c:\mysql\bin.

## 2.3 Konsep pengiriman data dengan teknologi streaming[4].

Teknologi *streaming* audio atau video memungkinkan *user* dapat menonton atau mendengar lagu-lagu atau video/film tanpa perlu menyelesaikan proses *download* (pengambilan *file* dari *server*) ke komputer *user*. Dapat dibayangkan dengan *file* video sebesar 100 MB jika harus di-*download* ke komputer *user* akan menghabiskan waktu yang sangat lama untuk menayangkannya, sedangkan dengan teknologi *streaming* melalui proses *buffering* dalam beberapa detik, sedikit demi sedikit *file* video bisa ditayangkan.

Setiap *file* video yang akan di-*streaming* harus melalui kompresi(*convert*) *file* terlebih dahulu sehingga *file* tersebut memungkinkan untuk bisa di-*streaming* biasanya *file* yang dapat di-*streaming* berekstesi rm, mpeg2 dan sebagainya. Untuk menayangkan data *streaming* biasanya digunakan media player, seperti QuickTime, RealOne Player, atau Windows Media Player untuk memutar media yang di-*stream* dari sebuah *web server* yang disertakan pada halaman web. Biasanya, pertama-tama sistem akan meminta *metafile*, yang meliputi keterangan alamat letak data yang di-*stream* berada (URL), dari data yang di-*stream*. *Metafile* ini nantinya akan dikirim ke *web browser user*, kemudian *browser* akan membuka berkas yang dimaksud dengan memilih media *player* yang sesuai dengan jenis media yang dispesifikasikan (di-*embed* pada *script*) oleh *metafile*. Gambar 2.1 menunjukkan ilustrasi tentang proses pengiriman data *streaming* melalui web:




Gambar 2.1 Ilustrasi penyampaian data *streaming*.

Untuk lebih jelasnya berikut ini adalah contoh sederhana pembuatan RAM *file* dan halaman *web* untuk proses *streaming*: [6]

1. Di dalam program text editor, misalnya notepad tuliskan kode berikut:
   rtsp: //www.infonet.com/infonet/video/tes video.rm

Kode program diatas merupakan pemisalan url tempat lokasi *file video* berekstensi rm berada.

Kemudian simpan dengan nama contohfile.ram

RAM *file* diatas merupakan suatu *metafile* yang membuka *file video* berekstensi rm dengan menggunakan protokol rtsp untuk penyaluran datanya.

2. Buat *web page* untuk mengakses RAM *file* tersebut dengan menggunakan fasilitas *link* yang ada pada *web*.

   Misalnya : <a href = "contohfile.ram">Play movie </a>

   Simpan dengan nama Tes streaming.html

   Untuk menayangkan *file video* yang dimaksud, kita memerlukan suatu *player* yang di-*embed* (disertakan) pada web. Dalam hal ini berarti *player* yang mendukung *video* yang telah didefinisikan sebelumnya oleh *metafile* (contohfile.ram yang mendefinisikan *video* berekstensi realmedia *file* berarti memerlukan *player* dengan *default* rm *file*). Berikut ini merupakan *script* untuk meng-*embed* realmedia *player* dalam *web*:

   <embed src = "http ://www.infonet.com/infonet/video/contohfile.ram" type = "audio/x-pn-realaudio-plugin" control = "ImageWindow,ControlPanel" width=200 height=176></embed>.

3. Buka Tes streaming.html, kemudian jalankan *link* Play movie.

## 2.4 Protokol Transfer

Protokol *transfer* adalah suatu protokol yang digunakan untuk pengiriman informasi di internet. HTTP adalah merupakan protokol standar untuk suatu dokumen *web*. Selain HTTP di internet juga dikenal beberapa protokol transfer lain diantaranya :

1. **FTP** (*File Transfer Protokol)* protokol ini dirancang untuk memungkinkan pemakai mentransfer *file* dalam format text atau *binary* dalam suatu *server* komputer di internet.
2. **Ghopher** protokol ini dirancang untuk mengakses *server* gopher yang menyediakan informasi dengan menggunakan suatu sistem menu atau melalui hubungan telnet.

3. **News NNTP** (*Network News Transfer Protokol)* ini adalah protokol yang digunakan untuk mendistribusikan berita di USENet. USENet adalah suatu sistem yang dirancang sebagai forum diskusi dengan berdasarkan pada topik-topik yang disebut *news group.*
4. **Telnet,** protokol ini digunakan untuk *login* ke suatu *server* komputer.

### 2.4.1 RTSP (Real Time Streaming Protokol)[4]

Cara kerja dari media yang di-*stream* dapat dikirim oleh seorang klien dari sebuah standar web server dengan pendekatan yang menggunakan hypertext transport protocol atau HTTP yang merupakan protokol yang biasa digunakan untuk mengirim dokumen dari web browser. Biasanya, seorang klien menggunakan media player, seperti QuickTime, RealPlayer, atau Windows Media Player untuk memutar kembali media yang di-*stream* dari sebuah web server. Biasanya, pertama-tama klien akan meminta metafile, yang meliputi keterangan alamat letak data yang di-*stream* berada (URL), dari data yang di-*stream*. Metafile ini nantinya akan dikirim ke web browser klien, kemudian browser akan membuka berkas yang dimaksud dengan memilih media player yang sesuai dengan jenis media yang dispesifikasikan oleh metafile.

Masalah yang dihadapi dari pengiriman data yang di-*stream* pada standar web server adalah bahwa HTTP merupakan protokol yang tidak memiliki status. Sehingga, web server tidak dapat menjaga status dari koneksinya dengan klien. Keadaan ini mengakibatkan kesulitan yang dialami klien manakala ia ingin melakukan *pause* saat pengiriman data yang distream. Hal ini dikarenakan pelaksanaan *pause* membutuhkan pengetahuan (biasanya status) dari web browser, sehingga ketika klien ingin memulai kembali mengirim data melalui streaming web server dapat dengan mudah memutar kembali berdasarkan status yang disimpan tersebut.

Strategi alternatif yang dapat dilakukan untuk menanggulangi hal diatas adalah dengan menggunakan server *streaming* khusus yang didesain untuk men-*streaming* media, yaitu real-time streaming protocol (RTSP). RTSP didesain untuk melakukan komunikasi antara server yang melakukan *streaming* dengan media player. Keuntungan RTSP adalah bahwa protokol ini menyediakan koneksi

yang memiliki status antara server dan klien, yang dapat mempermudah klien ketika ingin melakukan *pause* atau mencari posisi random dalam *stream* ketika memutar kembali data.

RTSP memiliki empat buah perintah. Perintah ini dikirim dari klien ke sebuah server *streaming* RTSP. Keempat perintah tersebut adalah:

1. *Setup.* Server mengalokasikan sumber daya kepada *client session.*
2. *Play.* Server mengirim sebuah *stream* ke *client session* yang telah dibangun dari perintah *setup* sebelumnya.
3. *Pause.* Server menunda pengiriman *stream* namun tetap menjaga sumber daya yang telah dialokasikan.
4. *Teardown.* Server memutuskan koneksi dan membebastugaskan sumber daya yang sebelumnya telah digunakan.

## 2.5 Konsep Analisa dan Perancangan Berbasis Objek[1]

Obyek dalam *software analysis & design* adalah sesuatu berupa konsep (*concept*), benda (*thing*), dan sesuatu yang membedakannya dengan lingkungannya. Secara sederhana obyek adalah mobil, manusia, *alarm* dan lainlainnya. Tapi obyek dapat pula merupakan sesuatu yang abstrak yang hidup didalam sistem seperti tabel, *database, event, system messages.*

Obyek dikenali dari keadaannya dan juga operasinya. Sebagai contoh sebuah mobil dikenali dari warnanya, bentuknya, sedangkan manusia dari suaranya. Ciri-ciri ini yang akan membedakan obyek tersebut dari obyek lainnya. Alasan mengapa saat ini pendekatan dalam pengembangan software dengan *object-oriented*, pertama adalah *scalability* dimana obyek lebih mudah dipakai untuk menggambarkan sistem yang besar dan komplek. Kedua *dynamic modeling*, yaitu dapat dipakai untuk permodelan sistem dinamis dan *real time.*

Dalam dunia pemodelan, metodologi implementasi obyek walaupun terikat kaidah-kaidah standar, namun teknik pemilihan obyek tidak terlepas pada subyektifitas *software analyst & designer.* Beberapa obyek akan diabaikan dan beberapa obyek menjadi perhatian untuk diimplementasikan di dalam sistem. Hal ini sah-sah saja karena kenyataan bahwa suatu permasalahan sudah tentu memiliki

lebih dari satu solusi. Ada 3 (tiga) teknik atau konsep dasar dalam OOA/D, yaitu pemodulan (*encapsulation*), penurunan (*inheritance)* dan *polymorphism.*

a. Pemodulan (*Encapsulation*)

   Pada dunia nyata, seorang ibu rumah tangga menanak nasi dengan menggunakan *rice cooker*, ibu tersebut menggunakannya hanya dengan menekan tombol. Tanpa harus tahu bagaimana proses itu sebenarnya terjadi. Disini terdapat penyembunyian informasi milik *rice cooker*, sehingga tidak perlu diketahui seorang ibu. Dengan demikian menanak nasi oleh si ibu menjadi sesuatu yang menjadi dasar bagi konsep *information hiding.*

b. Penurunan (*Inheritance*)

   Obyek-obyek memiliki banyak persamaan, namun ada sedikit perbedaan. Contoh dengan beberapa buah mobil yang mempunyai kegunaan yang berbeda-beda. Ada mobil bak terbuka seperti truk, bak tertutup seperti sedan dan minibus. Walaupun demikian obyek-obyek ini memiliki kesamaan yaitu teridentifikasi sebagai obyek mobil, obyek ini dapat dikatakan sebagai obyek induk (*parent*). Sedangkan minibus dikatakan sebagai obyek anak (*child*), hal ini juga berarti semua operasi yang berlaku pada mobil berlaku juga pada minibus.

c. *Polymorphism*

   Pada obyek mobil, walaupun minibus dan truk merupakan jenis obyek mobil yang sama, namun memiliki juga perbedaan. Misalnya suara truk lebih keras dari pada minibus, hal ini juga berlaku pada obyek anak (*child*) melakukan metoda yang sama dengan algoritma berbeda dari obyek induknya. Hal ini yang disebut *polymorphism*, teknik atau konsep dasar lainnya adalah ruang lingkup/pembatasan. Artinya setiap obyek mempunyai ruang lingkup kelas, atribut, dan metoda yang dibatasi.

## 2.6 Unified Modelling Language (UML)

UML merupakan salah satu alat bantu yang sangat handal dalam pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan *visual* yang memungkinkan bagi pengembang

sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, *Object Modelling Technique* (OMT) dan *Object Oriented Software Engineering* (OOSE). Metode Booch dari Grady Booch sangat terkenal dengan nama metode *Design Object Oriented.* Metode ini menjadikan proses analisis dan *design* ke dalam empat metode iteratif yaitu:

1. Identifikasi objek dan kelas
2. Identifikasi semantik dari hubungan objek dan kelas
3. Perincian *interface*
4. Implementasi

Keunggulan metode Booch adalah pada detil dan kayanya dengan notasi dan elemen.

Pemodelan OMT yang dikembangkan oleh Rumbaugh didasarkan pada analisis terstruktur dan pemodelan *entity-relationship.* Tahapan utama dalam metodologi ini adalah:

1. Analisis
2. *Design* Sistem
3. *Design* Objek
4. Implementasi

Keunggulan metode ini adalah pada penotasian yang mendukung semua konsep *object oriented.*

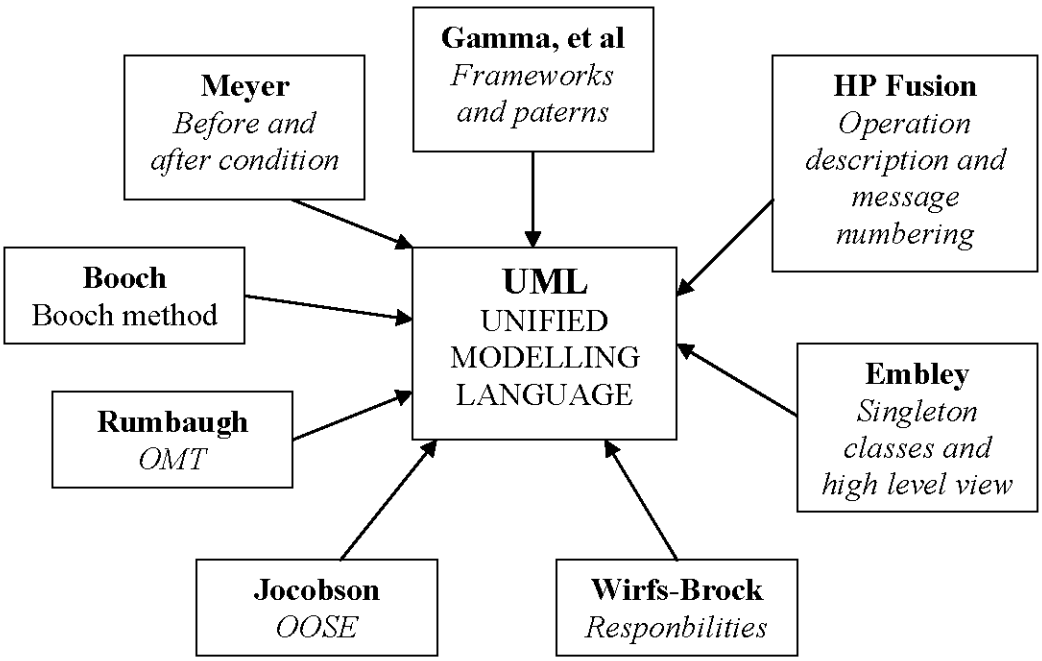
Metode OOSE dari Jocobson lebih memberi penekanan pada *use case.* Jocobson memiliki tiga tahapan yaitu:

a. Membuat model *requirement* dan analisis
*b. Design*
c. Implementasi

Keunggulan metode ini adalah mudah dipelajari karena memiliki notasi yang sederhana namun mencakup seluruh tahapan dalam rekayasa perangkat lunak.

Dengan UML, metode Booch, OMT, dan OOSE digabungkan dengan membuang elemen-elemen yang tidak praktis ditambah dengan elemen-elemen

dari metode lain yang lebih efektif dan elemen-elemen baru yang belum ada pada metode terdahulu. Hal ini membuat UML lebih ekspresif dan seragam daripada metode yang lain. Berikut adalah gambar unsur-unsur yang merupakan pembentuk UML.




Gambar 2.2 Unsur-unsur pembentuk UML

UML diagram memiliki beberapa model diagram yang digunakan berdasarkan fungsinya. Diagram-diagram tersebut memiliki ketergantungan satu sama lain. Beberapa jenis diagram yang paling sering digunakan antara lain:

***a. Use Case Diagram***

*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/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

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. 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 yang dideklarasikan di dalam class yaitu :

- Nama (dan *stereotype*)
- Atribut
- Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut:

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

Antar kelas bisa terjadi hubungan dimana suatu *class* terhubung dengan satu kelas atau lebih. Hubungan-hubungan tersebut adalah:

- 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*.
- Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas”).

- 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..

***c. Statechart Diagram***

*Statechart diagram* menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

Dalam UML, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring.

***d. Activity Diagram***

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

*Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case*

menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

*Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

***e. Sequence Diagram***

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

*Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary, controller* dan *persistent entity*.

***f. Collaboration Diagram***

*Collaboration diagram* juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor dan *messages* dari *level* yang sama memiliki prefiks yang sama.

***g. Component Diagram***

*Component diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya.

Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time, link time*, maupun *run time*.

Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

***h. Deployment Diagram***

*Deployment/physical diagram* menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisikal.

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

## 2.7 *Unified Software Development Process* (USDP) *Methodology*

Dalam pengembangan aplikasi diperlukan manajemen dan organisasi agar dapat bekerja secara efektif dan efisien. Pada dasarnya sebuah aplikasi melalui beberapa tahapan sebelum menghasilkan aplikasi jadi yang siap untuk digunakan. Tahapan-tahapan tersebut tidak hanya dijalani oleh seorang pengembang saja akan tetapi terdiri dari beberapa orang yang memiliki jabatan dan tugas masing-masing. Agar dapat bekerja sesuai dengan tugasnya secara efektif dan efisien maka diperlukan pengaturan tata cara dalam pengembangan aplikasi. Untuk itu digunakan methodology yang dipilih sesuai dengan kebutuhan dan menjadi pedoman dalam pengembangan aplikasi.

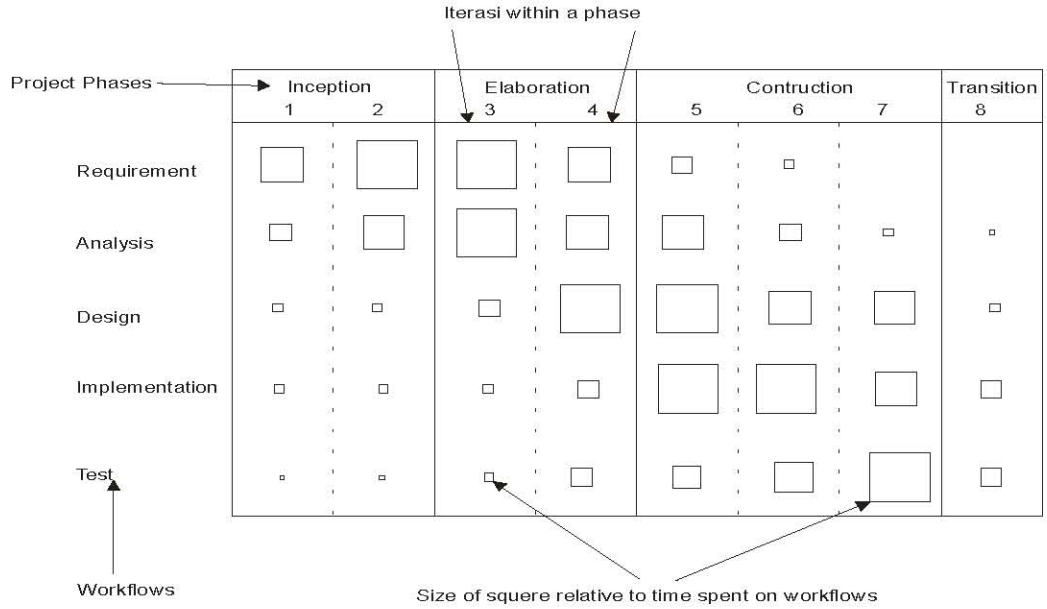
Metodologi pengembangan aplikasi adalah kumpulan tata cara, teknik dan langkah-langkah pengembangan yang ditujukan untuk menghasilkan jajaran *software* yang baik dan berkualitas. Dalam pengembangan aplikasi yang berorientasi objek terdapat beberapa metodologi yang dapat digunakan. Beberapa diantaranya telah disebutkan diatas yang merupakan unsur-unsur pembentuk *tools* UML.

Dalam pengembangan aplikasi "*Video On Demand* Melalui Web Dengan Menggunakan Proses Streaming (Studi kasus penyampaian kebutuhan informasi baik tulisan maupun audio visual melalui web)" ini digunakan sebuah metodologi yang disebut *Unified Software Development Process* (USDP). Metodologi USDP dikembangkan Jocobson dan team yang telah menciptakan UML pada tahun 1999. Mereka mengklaim bahwa USDP adalah tata cara dan teknik terbaik dalam pengembangan sistem informasi. Tata cara dan teknik pengembangan tersebut dapat dikelompokkan seperti berikut:

- *Iterative dan incremental development*
- *Component-based development*
- *Requirement Driven development*
- *Configurability*
- *Architecture Centrism*
- *Visual Modelling Techniques*

### 2.7.1 Tahap Pengembangan Metodologi USDP

Tidak seperti metodologi pengembangan aplikasi tradisional USDP memiliki 4 fase utama pengembangan. Fase-fase ini menggambarkan penekanan yang berbeda dalam urutan pengembangan sistem. Berikut adalah diagram yang menggambarkan proses pengembangan dalam USDP.




Gambar 2.3 Fase dan aliran kerja pada metodologi USDP.

*Workflows* atau aliran kerja pada metodologi USDP merupakan aktifitas-aktifitas pengembangan yang hampir sama dengan metodologi pengembangan aplikasi yang lainnya. Perbedaannya yaitu aktifitas-aktifitas tersebut dilakukan berulang sebanyak 4 fase dengan keseluruhan minimal 8 pengulangan yang harus dilakukan pada pengembangan. Fase pengembangan dilakukan secara berurutan dan project management menentukan apakah pengembangan dilakukan ke fase berikutnya atau tetap di fase tersebut. Untuk tiap fase secara umum memiliki aktifitas sebagai berikut:

a. Investigasi.
b. Pemodelan kebutuhan sistem.
c. Analisa model kebutuhan sistem.
d. Perancangan.
e. Pembuatan kode program.
f. Pengujian kode program.
g. Ulangi lagi semua aktifitas tersebut.

Tidak ada aturan yang jelas berapa kali pengulangan aktifitas-aktifitas tersebut dilakukan dalam tiap fase. Hal ini menjadi kebijakan project management apakah pengembangan dilanjutkan ke fase berikut atau mengulangi aktifitas-aktifitas pada fase tersebut.