

BAB I

PENDAHULUAN

I.1 Latar Belakang

Perkembangan teknologi sistem informasi, dan *entertainment* di dunia *smartphone* berkembang sangat pesat belakangan ini, menyebabkan kebutuhan informasi dalam dunia *entertainment* khususnya film sangat dibutuhkan. Informasi tersebut bisa dalam bentuk *trailer* atau cuplikan. *Trailer* film atau cuplikan film merupakan sebuah promosi dari sebuah film yang akan ditayangkan, cuplikan biasanya berdurasi singkat, hanya berisi garis besar dari film tersebut. *Trailer* film sangatlah penting bagi pecinta film untuk melihat secara sekilas film yang akan ditontonnya.

YouTube merupakan sebuah situs web video sharing gratis paling populer saat ini dan menjadi media untuk promosi dan mencari cuplikan-cuplikan film karena dengan cepat dapat diakses oleh jutaan penggunanya. Perkembangan internet yang pesat dan sudah didukungnya berbagai format video pada *smartphone* dewasa ini membuat akses ke *YouTube* semakin mudah namun bagi sebagian orang masih terkendala dengan ukuran layar. Banyak pecinta film merasa kesulitan untuk mencari cuplikan film yang sedang tayang atau yang akan datang karena dianggap terlalu rumit, memakan waktu cukup lama dan bahkan cuplikan film tercampur dengan cuplikan film yang sudah lama.

Oleh karena itu untuk mempercepat akses ke

cuplikan film-film baru diperlukan sebuah aplikasi yang langsung menampilkan cuplikan-cuplikan film yang baru atau akan tayang. Platform Symbian menjadi target pembuatan aplikasi dikarenakan minimnya aplikasi sejenis yang tersedia di platform tersebut, dan selain itu perangkat *smartphone* yang berbasis *Symbian Belle* masih banyak dipasaran dan penggunaanya sangat banyak di berbagai penjuru dunia.

Maka dari itu penulis tertarik menciptakan sebuah aplikasi *YouTube Movie Trailers* untuk perangkat berbasis *Symbian Belle OS* dengan menggunakan *Qt framework*. Aplikasi ini dapat di instal ke semua *smartphone* yang menggunakan platform *Symbian Belle OS* dengan tujuan dapat memberikan informasi berupa cuplikan mengenai film-film yang sedang atau akan tayang.

I.2 Rumusan Masalah

Masalah *YouTube Movie Trailers* dirumuskan sebagai berikut "bagaimana membangun aplikasi *YouTube Movie Trailers* pada *smartphone* yang berbasis *Symbian Belle* dengan menggunakan *Qt* dan dapat memberikan informasi mengenai *trailer* film terbaru, memutar video *trailer*, melihat berapa banyak yang melihat, *like* dan *unlike* video *trailer* tersebut."

I.3 Batasan Masalah

Pada *YouTube Movie Trailers* ini diberikan pembatasan masalah sebagai berikut:

1. Hasil pencarian tidak dikategorikan menurut *genre* atau jenis film.
2. Aplikasi yang dibangun difokuskan pada bagaimana menampilkan *trailer*, sehingga tidak dilengkapi dengan fasilitas *like* dan *unlike* video.
3. Sistem informasi ini dibuat berdasarkan *channel* pengguna *YouTube*.

I.4 Tujuan Penelitian

Penelitian bertujuan untuk membangun suatu aplikasi pemutar video dari cuplikan film yang tersedia di *YouTube* yang berjalan pada *smartphone* berbasis *Symbian Belle*.

I.5 Metode Penelitian

Metode yang akan digunakan dalam penelitian ini terdiri dari langkah-langkah berikut:

1. Melakukan studi kepustakaan terhadap berbagai referensi yang berkaitan dengan penelitian yang dilakukan.
2. Metode Pembangunan Perangkat Lunak yang meliputi Analisis Kebutuhan Perangkat Lunak, Perancangan Perangkat Lunak, Implementasi Perangkat Lunak, Pengujian Perangkat Lunak.

I.6. Sistematika Penulisan Laporan

Penulisan laporan tugas akhir akan dibagi menjadi enam (6) bab yang meliputi:

Bab I : Pendahuluan

Bab ini berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan, metode yang digunakan, dan sistematika penulisan laporan.

Bab II : Tinjauan Pustaka

Bab ini berisi tentang uraian singkat hasil-hasil penelitian atau analisis terdahulu yang ada hubungannya dengan permasalahan yang akan ditinjau dalam Tugas Akhir.

Bab III : Landasan Teori

Bab ini berisi penjelasan teori-teori, pendapat, prinsip dan sumber-sumber lain yang dapat dipertanggungjawabkan secara ilmiah dan dapat dipergunakan sebagai pembanding atau acuan di dalam pembahasan masalah.

Bab IV : Analisis dan Perancangan Sistem

Bab ini akan menjelaskan tentang analisis perancangan aplikasi *YouTube Movie Trailers*.

Bab V : Implementasi dan Pengujian Perangkat Lunak

Pada bab ini akan dibahas gambaran dan langkah mengenai penggunaan aplikasi YMT (*YouTube Movie Trailers*), meliputi pengkodean,

pengujian aplikasi, dan pengujian kelayakan aplikasi yang akan dibuat.

Bab VI : Kesimpulan dan Saran

Bab ini berisi kesimpulan dari pembahasan tugas akhir secara keseluruhan dan saran untuk pertimbangan lebih lanjut.



BAB II

TINJAUAN PUSTAKA

2.1. Pengantar

Dalam bab ini akan dijelaskan tentang uraian singkat hasil-hasil penelitian atau analisis terdahulu yang ada hubungannya dengan permasalahan yang akan ditinjau dalam Tugas Akhir. Uraian disusun secara runtut dari yang umum sampai yang khusus (paling dekat dengan permasalahan yang akan ditinjau). Pada akhir tinjauan pustaka ditunjukkan perbedaan atau sifat khusus dari penelitian yang akan dilakukan, yang membedakan dengan penelitian-penelitian terdahulu.

2.2. Tinjauan Pustaka

Menurut Satwika (2011) *Proses Video Streaming Dengan Protocol Real Time Streaming Protocol* teknologi *streaming* adalah proses pengiriman data kontinyu alias terus-menerus yang dilakukan secara *broadcast* melalui Internet untuk ditampilkan oleh aplikasi *streaming* pada PC (klien). Paket-paket data yang dikirimkan telah dikompresi untuk memudahkan pengirimannya melalui Internet. Dalam pengolahan *video*, *video streaming* adalah sebuah teknologi untuk memainkan *file video* atau *audio* secara langsung ataupun dengan *pre-recorder* dari sebuah mesin *server (web server)*. Dengan kata lain *video* yang terdapat dalam komputer *server* dapat langsung dijalankan dan dilihat oleh klien dengan menggunakan sebuah media *player*. *Real Time Streaming Protocol* (RTSP) adalah sebuah protokol level aplikasi untuk kontrol atas pengiriman data dengan sifat *real-*

time. RTSP menyediakan kerangka *extensible* untuk mengaktifkan kendali pada pengiriman data *real-time*, seperti *audio* dan *video*, dengan menggunakan *Transmission Control Protocol* (TCP) atau *User Data Protocol* (UDP).

Sistem yang baik menurut Amalya (2012) adalah sistem yang selalu dapat menyesuaikan dengan perubahan lingkungan yang terjadi di sekitarnya. Atau, pengembangan sistem tidak harus membuat sistem yang baru, tetapi apabila sistem yang lama masih dapat digunakan dapat dilakukan perbaikan terhadap sistem yang lama tersebut. Metodologi penelitian dalam ilmu komputer / sistem informasi / teknologi informasi adalah kumpulan dari metode, prosedur, teknik, *tool* serta pendokumentasian yang membantu peneliti dalam melaksanakan sebuah penelitian dalam bidang ilmu komputer / sistem informasi / teknologi informasi.

Menurut Hardianto (2011) Aplikasi *Video* dan *Audio On Demand*, sebelum menggunakan teknologi *streaming*, layanan *video* di internet harus diunduh terlebih dahulu dan baru dapat disaksikan secara utuh yang disebut teknik konvensional. Dengan teknologi berbasis *streaming* ini diharapkan lebih dapat memberikan kecepatan dalam pengolahan penampilan *video* tersebut. Aplikasi dibuat menggunakan *Helix Streaming Server Real Player* karena merupakan solusi perangkat lunak yang digunakan untuk pengiriman *audio* dan *video streaming* secara *on-demand*.

Aplikasi *video streaming* yang dibuat oleh Pratama (2010) menggunakan koneksi *Wireless Local Area Network* (WLAN) untuk melakukan *streaming*. Aplikasi dibuat

dengan menggunakan teknologi *JSR 135* atau lebih dikenal dengan *Mobile Media API (MMAPI)*. *MMAPI* digunakan untuk mengontrol proses *video streaming* dan fitur-fitur pendukungnya, aplikasi yang dibuat menggunakan 2 pilihan protokol yaitu *RSTP* dan *HTTP*. Aplikasi yang dibuat berjalan pada telepon seluler berbasis *Symbian 60*.

Berdasarkan dari aplikasi-aplikasi yang telah dikembangkan sebelumnya, maka penulis berniat untuk mengembangkan aplikasi *video streaming* baru yang dapat diakses melalui *smartphone* yang berbasis *Symbian Belle OS*. Aplikasi yang akan dikembangkan ditujukan untuk menampilkan dan memutar video cuplikan-cuplikan film terbaru dari situs *YouTube*, dimana *channel* pengunggah menjadi sumber dari video-video yang diambil.

Penulis mengalami bahwa kebanyakan kecepatan akses internet dari beberapa *provider* tidak mumpuni untuk *streaming* video secara cepat. Oleh karena itu penulis menggunakan *YouTube API mobile* dengan protokol *RSTP* sebagai pengambil dan transfer data dari *YouTube*. Aplikasi ini akan dibangun menggunakan *tool Qt framework*.

BAB III

LANDASAN TEORI

3.1 Sistem Informasi

Sistem Informasi adalah kombinasi dari manusia, fasilitas atau alat teknologi, media, prosedur dan pengendalian yang bermaksud menata jaringan komunikasi yang penting, proses atas transaksi-transaksi tertentu dan rutin, membantu manajemen dan pemakai intern dan ekstern dan menyediakan dasar pengambilan keputusan yang tepat (Midjan, 2010).

Sistem informasi juga dapat diartikan sebagai suatu pengaturan orang, data, proses, presentasi informasi dan teknologi informasi yang saling berinteraksi untuk mendukung dan meningkatkan operasi sehari-hari dalam bisnis dan mendukung kebutuhan pemecahan masalah dan pembuatan keputusan dari manajer dan pemakai. Sistem informasi dan teknologi memegang peranan yang penting dalam proses bisnis. Sistem informasi dan teknologi informasi digunakan semaksimal mungkin untuk mendukung proses bisnis sehingga menjadi lebih efektif dan efisien (Shofawaty, 2010).

Ada tiga tujuan diterapkannya sistem informasi dalam perusahaan yaitu:

1. Untuk mengambil dan menyimpan data tentang aktivitas bisnis dan transaksi perusahaan dengan efektif dan efisien.
2. Untuk menyediakan informasi yang berguna untuk pengambilan keputusan.

3. Untuk melakukan kontrol agar data-data disimpan dan diproses dengan akurat.

3.1.1 Model Dasar Kesuksesan Teknologi Informasi

DeLone dan McLean (1992) mengembangkan suatu model parsimoni (model yang lengkap tetapi sederhana) yang mereka sebut dengan nama Model Kesuksesan Sistem Informasi DeLone dan McLean (*D&M IS Success Model*). Dari kontribusi beberapa penelitian sebelumnya dan akibat perubahan peran dan penanganan sistem informasi yang telah berkembang DeLone dan Mclean (2003) memperbarui modelnya dan menyebutnya sebagai Model Kesuksesan Sistem Informasi D&M Diperbarui (*Updated D&M IS Success Model*). Model yang diusulkan ini merefleksikan ketergantungan dari enam pengukuran kesuksesan sistem informasi. Keenam elemen atau faktor atau komponen atau pengukuran dari model ini adalah sebagai berikut (Arie, 2011) :

1. Kualitas Sistem (*System Quality*) digunakan untuk mengukur kualitas sistem teknologi informasinya sendiri.
2. Kualitas Informasi (*Information Quality*) mengukur kualitas keluaran dari sistem informasi.
3. Kualitas Pelayanan (*Service Quality*) awalnya digunakan di penelitian pemasaran (*marketing*). Penelitian-penelitian sistem informasi yang memasukkan pengukuran kualitas pelayanan (*service quality*) ke dalam model D&M meminjamnya dari penelitian pemasaran.
4. Penggunaan Informasi (*Use*) adalah penggunaan keluaran suatu sistem informasi oleh penerima.

5. Kepuasan Pemakai (*User Satisfaction*) adalah respon pemakai terhadap penggunaan keluaran sistem informasi.
6. Manfaat Bersih (*Net Benefits*) merupakan penggabungan dampak individual (*individual impact*) dan dampak organisasional (*organizational impact*). Dampak individual (*individual impact*) merupakan efek dari informasi terhadap perilaku pemakai. Sedangkan dampak organisasi (*organizational impact*) merupakan dampak dari informasi terhadap kinerja organisasi.

3.2 Sistem Informasi Berbasis Komputer

Sebuah sistem adalah sekelompok dua atau lebih komponen-komponen yang saling berkaitan (*interrelated*) atau subsistem-subsistem yang bersatu untuk mencapai tujuan yang sama (*common purpose*) (Hall, 2001 :5). Menurut Mulyadi (1999) didefinisikan sebagai kelompok unsur yang erat berhubungan satu sama lainnya, yang berfungsi bersama-sama untuk mencapai tujuan.

Burch dan Grudnitsky (1986) mendefinisikan informasi adalah data yang telah diletakkan dalam konteks yang lebih berarti dan berguna yang dikomunikasikan kepada penerima untuk digunakan dalam pembuatan keputusan. Menurut Davis dan Olson (1985) Informasi adalah data yang diproses lebih jauh sehingga mempunyai arti bagi si penerima dan mempunyai "nilai pengaruh" atas tindakan-tindakan, keputusan-keputusan sekarang atau masa yang akan datang.

3.3 Analisis Sistem

Untuk mengelola sumber informasi, setiap bagian dalam perusahaan harus ikut berpartisipasi untuk membangun sistemnya. Dalam proses ini diperlukan analisis terhadap kebutuhan sistem untuk mencari penyelesaian dari permasalahan yang dihadapi dengan membuat ataupun memperbaiki sistem dengan sistem yang baru. Analisis sistem adalah suatu proses mengumpulkan dan menginterpretasikan kenyataan-kenyataan yang ada, mendiagnosa persoalan dan menggunakan keduanya untuk memperbaiki sistem. Sedangkan analisis sistem adalah orang yang mempunyai kemampuan untuk menganalisis sebuah sistem, memilih alternatif pemecahan masalah dan menyelesaikan masalah tersebut dengan menggunakan komputer (Kristanto, 2004).

Siklus pembangunan sistem (*System Development Life Cycle*) merupakan suatu pendekatan yang sistematis untuk menyelesaikan permasalahan sistem. Pola-pola generik yang digunakan antara lain:

a. *Planning*

Pada tahap ini dilakukan pengenalan terhadap masalah yang dihadapi, lingkup maupun batasannya, serta merencanakan strategi dan tujuan pengembangan.

b. *Analysis*

Mempelajari dan menganalisis masalah, penyebab dan pengaruhnya. Kemudian mengidentifikasi dan menganalisis kebutuhan yang harus dipenuhi demi kesuksesan solusi.

c. *Design*

Merancang solusi, dalam hal ini perlu diingat bahwa tidak semua solusi memerlukan perancangan.

d. *Implementation*

Implementasi dari solusi permasalahan yang telah dirancang beserta *testing* hasil implementasi.

e. *Support*

Menganalisis solusi yang telah diimplementasikan, meninjau ulang rancangan dan memperbaiki implementasi.

Pada tahap *design* untuk mempermudah proses pemahaman sering digunakan suatu pemodelan yang dibedakan berdasarkan pendekatan terstruktur dan berorientasi objek.

3.4 Mobile Multimedia

Multimedia on Mobile adalah teknologi data yang berupa *file* multimedia, dimana *file* tersebut dapat dijalankan pada sebuah perangkat *mobile*, baik berupa video, audio, dan animasi. Data *file* bisa dijalankan langsung dengan menunggu proses akses *download* selesai terlebih dahulu (Santoso, 2011)

3.5 Qt Creator

Qt merupakan *framework* C++ yang komperhensif untuk pengembang aplikasi GUI (*Graphical User Interface*) lintas *platform* dengan menggunakan pendekatan "write once, compile anywhere". Qt memungkinkan para *programmer* menggunakan satu sumber pohon untuk aplikasi yang akan berjalan pada berbagai jenis *platform*. Qt *library* dan *tools*-nya juga merupakan

bagian dari *Qtopia Core*, sebuah produk yang menyediakan jendela sistemnya sendiri diatas *embedded Linux*.

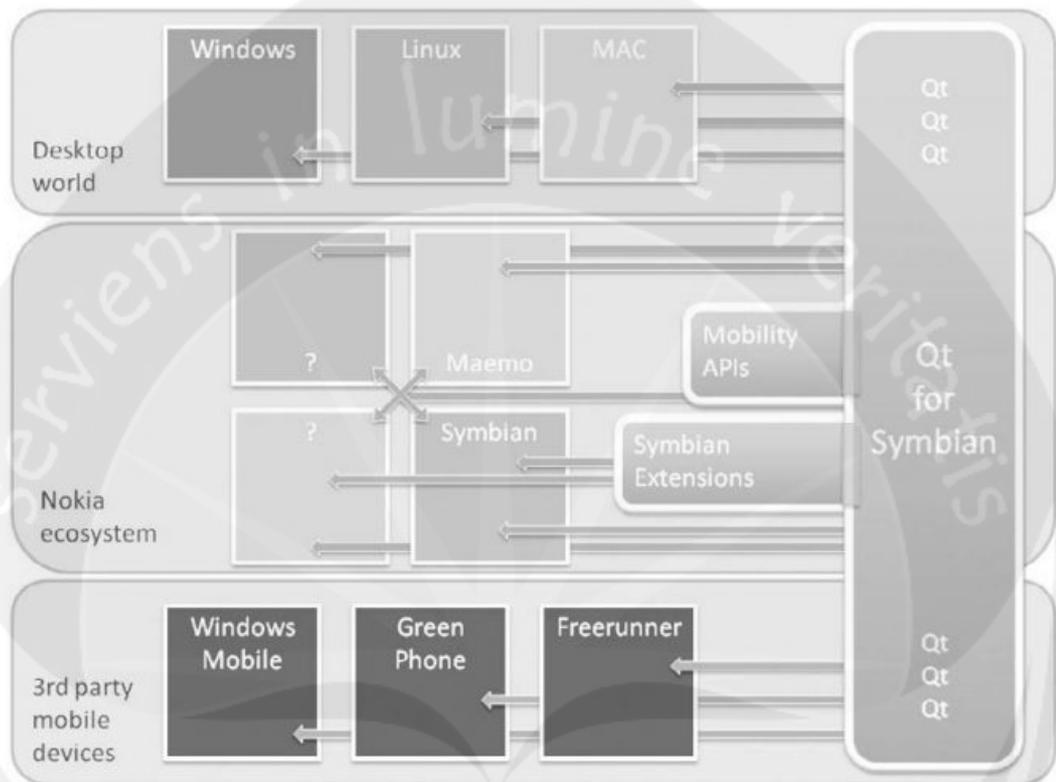
Qt mengawali reputasinya sebagai *framework* lintas platform, tetapi karena kekuatan API (*Application Programming Interface*) dan intuisinya banyak organisasi menggunakan Qt untuk membangun aplikasi *single-platform*. *Adobe Photoshop Album* merupakan salah satu contoh dari aplikasi *Windows* yang ditulis dengan Qt.

Qt tersedia dalam berbagai macam lisensi. Jika ingin membangun aplikasi komersial, pengembang harus membeli Qt dengan lisensi komersial; Jika ingin membangun aplikasi *open source*, maka pengembang dapat menggunakan Qt edisi *open source* (GPL). Qt merupakan pondasi dimana *K Desktop Environment* (KDE) dan berbagai macam aplikasi *open source* lainnya di bangun (Blanchette & Summerfield, 2006).

Qt pada mulanya berasal dari ide pembuatnya yaitu Haavard Nord dan Eirik Chambe-Eng yang berkeinginan untuk membuat sebuah "*objectoriented display system*", yang akhirnya mereka mengimplementasikannya menjadi sebuah produk pengembangan *software* dengan performansi tinggi dan bersifat *cross platform* yang merupakan sebuah *framework C++ GUI* (*Graphical User Interface*) yang banyak digunakan saat ini baik didunia komersil maupun *opensource software development*.

Produk Qt ini sekarang dihandle oleh *Qt Software* yang awalnya bernama *Trolltech Inc.* sebelum dibeli oleh Nokia untuk menjadi bagian divisi *software*-nya pada perangkat produk *embedded mobile*, (Hasan, 2009). Dan Qt mempunyai kemampuan untuk dapat digunakan

pada sistem operasi *Windows*, *Mac*, *Linux*, dan sistem operasi berbasis *Unix* lainnya dengan menggunakan pendekatan “*write once, compile anywhere*”, (Trolltech Developer, 2005).



Gambar 3.1. Qt Cross Platform Diagram

Built Tools dari Qt ialah *configure*, *Qmake*, *Moc*, dan *RCC* dan *Development Tools* dari Qt ialah *Qt Creator*, *Qt Designer*, *Qt Assistant*, *Qt Linguist*.

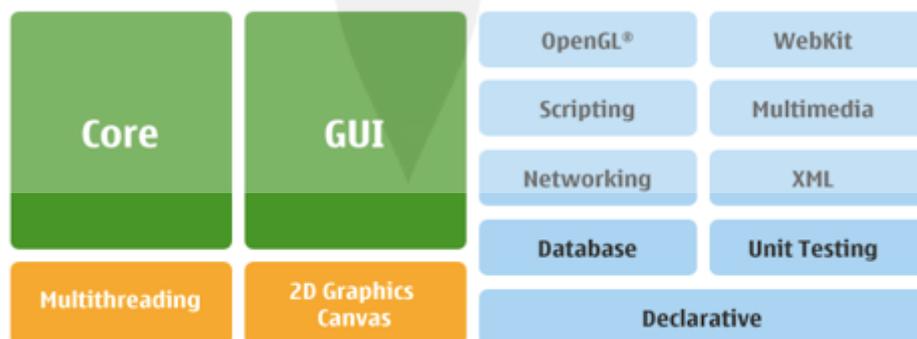
3.5.1 Qt Quick

Qt Quick singkatan dari *Qt User Interface Creation Kit* adalah teknologi antarmuka tingkat atas (*high-level user interface technology*) yang diklaim secara dramatis memudahkan para pengembang untuk mendisain antarmuka (*UI*) yang cantik, '*pixel perfect*' dan ringan disamping '*touch-enabled*' menggunakan basis pemrograman Qt, tanpa perlu pengetahuan tentang pemrograman C++.

Bagi para pengembang, *Qt Quick* menawarkan kelebihan-kelebihan antara lain:

1. Deklaratif yang mudah digunakan dengan bahasa mirip dengan skrip QML (*Qt Meta-Object Language*).
2. Tools didalam *Qt Creator IDE (Integrated Development Environment)* termasuk sebuah *visual editor* yang membantu para pengembang dan disainer menggabungkan elemen-elemen sederhana menjadi sebuah antarmuka animasi.
3. Kelebihan lainnya adalah pendekatan pemrograman deklaratif yang membantu pembuatan antarmuka-antarmuka menjadi gampang, hanya dengan memerintahkan apa yang harus dibuat dan tidak harus menuliskan bagaimana cara membuatnya.

Sementara Qt saat ini masih melayani pengembang C++, *Qt Quick* telah memperluas jangkauan pengguna Qt dengan kemampuan antarmuka. Hal ini membuka peluang emas untuk para disainer antarmuka termasuk pengembang yang terbiasa dengan bahasa skrip untuk memanfaatkan Qt dalam membuat antarmuka yang keren dan aplikasi-aplikasi untuk perangkat atau komputer yang mendukung *platform Qt* seperti halnya perangkat Nokia.



Gambar 3.2. Qt GUI Diagram

3.5.2 Qt QML

QML (Qt Meta Language atau Qt Modeling Language) adalah bahasa deklaratif berbasis *JavaScript* untuk merancang antar muka pengguna. *QML* merupakan bagian dari *Qt Quick*.

Qt quick dikembangkan oleh Nokia dalam *framework Qt*. *QML* digunakan untuk aplikasi mobile dimana *touch input*, *fluid animation (60 FPS)* dan *user experience* sangat penting.

Dalam *QML*, sebuah *user interface* dispesifikasikan sebagai sebuah pohon (*tree*) dari *objects* dengan properti-properti.

JavaScript digunakan sebagai bahasa *scripting* di *QML* (Qt Digia, 2011)

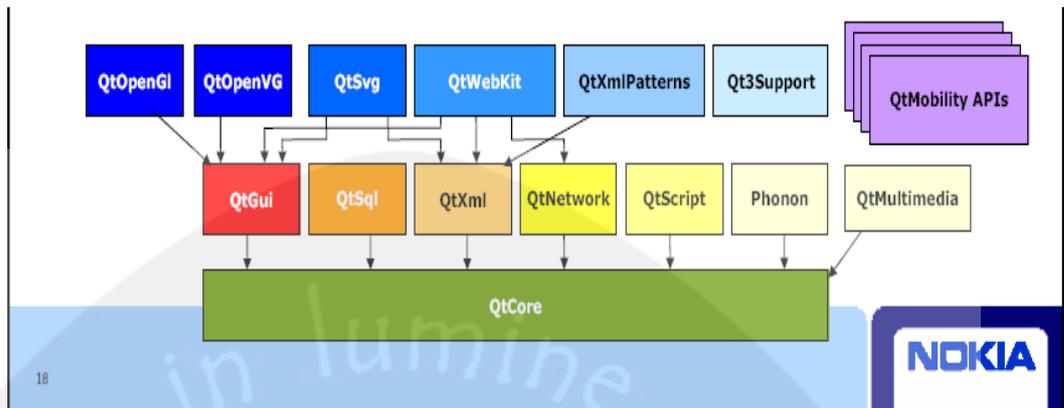
```
import QtQuick 1.0

Rectangle {
    width: 200
    height: 200
    color: "blue"

    Image {
        source: "pics/logo.png"
        anchors.centerIn: parent
    }
}
```

Gambar 3.3. Syntax dasar QML

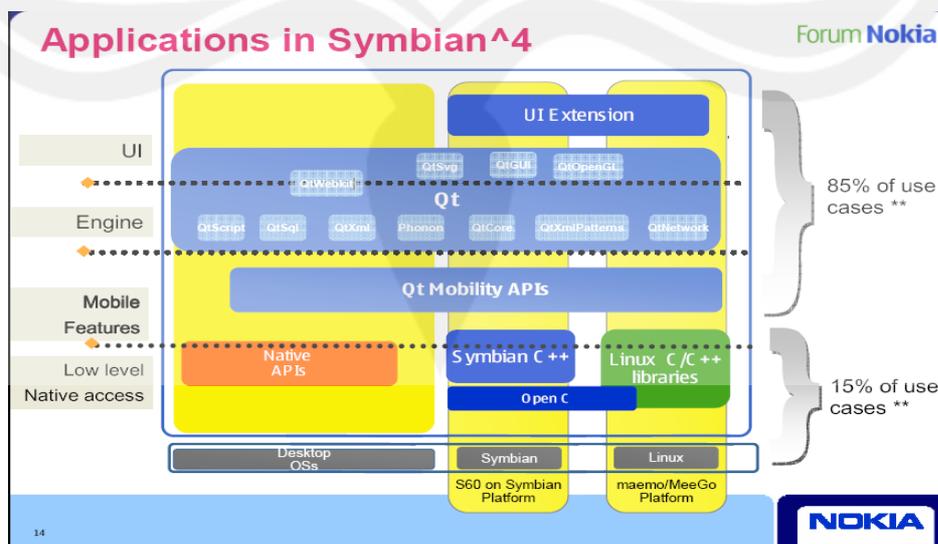
Qt lintas-platform terdiri dari kurang lebih 15 modul dan 700 class API, semua modul tergantung dari *Qt Core*. Terdapat 3 modul spesifik *platform* yaitu *QAx Container*, *QAx Server*, dan *QtD Bus*.



Gambar 3.4. Qt Modules

Jenis-jenis QML berdasarkan modulnya terbagi menjadi 8, yaitu:

1. *Qt Graphical Effect QML Types.*
2. *Qt Multimedia QML Types.*
3. *Qt Labs FolderListModel QML Types.*
4. *Qt Quick Local Storage QML Types.*
5. *Qt Quick Window QML Types.*
6. *Qt Quick XmlListModel QML Types.*
7. *Qt Quick Particles QML Types.*
8. *Qt Quick QML Types.*

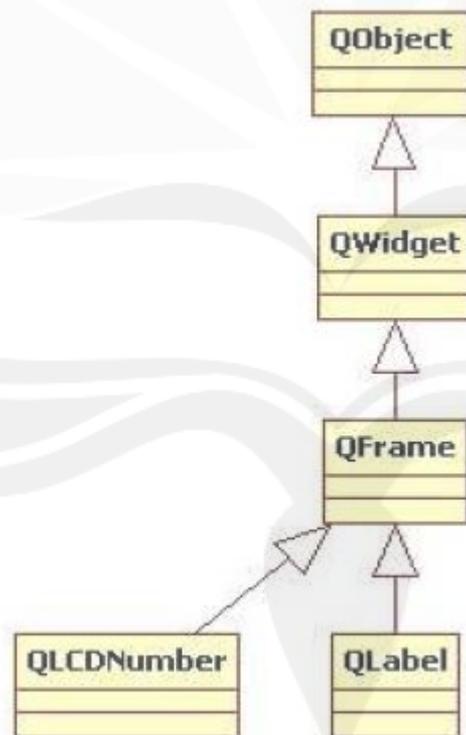


Gambar 3.5. Qt for Symbian

Tipe data dasar Qt sama dengan tipe data pada C++, yaitu *int*, *char*, *bool*, *double*. Untuk tipe data *string*, Qt mempunyai tipe data sendiri yaitu *QString*.

QObject Class Role, inti dari model obyek Qt, *class* dasar untuk semua *class* obyek termasuk semua *QWidget*s. Menyediakan pohon dan kepemilikan obyek. *QObject* mempunyai tanggung jawab untuk menyediakan titik tengah kepada konsep yang paling penting dalam Qt.

QObject mempunyai tiga tanggung jawab utama, yaitu, *memory management*, *introspection*, dan *event handling* (Blanchette, Summerfield, 2006).



Gambar 5.6. *QObject* Role

3.6 YouTube

YouTube adalah sebuah situs web *video sharing* (berbagi video) populer dimana para pengguna dapat memuat, menonton, dan berbagi klip video secara gratis. Umumnya video-video di YouTube adalah klip musik (video klip), *film*, TV, serta video buatan para penggunanya sendiri. Format yang digunakan video-video di YouTube adalah .flv yang dapat diputar di penjelajah web yang memiliki *plugin Flash Player*. (Rawasari, 2011).

YouTube awalnya menawarkan video dengan satu level kualitas, yaitu resolusi 320x240 piksel dengan *codec Sorenson Spark* (varian dari H.263), dengan mono MP3 audio. Pada bulan Juni 2007, YouTube menambahkan pilihan menonton video berformat 3GP di telepon genggam. Bulan Maret 2008, mode kualitas tinggi ditambahkan hingga resolusi 480x360 piksel. Bulan November 2008, mode HD 720p ditambahkan. Saat peluncuran 720p, pemutar YouTube diganti rasio aspeknya dari 4:3 menjadi layar lebar 16:9. Dengan fitur baru ini, YouTube mulai mengalihkan format kompresi video bakunya ke H.264/MPEG-4 AVC. Pada bulan November 2009, mode HD 1080p ditambahkan. Bulan Juli 2010, YouTube mengumumkan bahwa mereka telah meluncurkan serangkaian video berformat 4K, sehingga batas resolusinya naik hingga 4096x3072 piksel. Akan tetapi, per 2012 batas ini diturunkan menjadi 2048 x 1536.

Beberapa telepon pintar dapat mengakses video YouTube, tergantung penyedia dan data *plan*-nya. YouTube *Mobile* diluncurkan pada bulan Juni 2007 dengan protokol RTSP. Tidak semua video YouTube tersedia di situs versi telepon genggam.

3.6.1 YouTube Channel

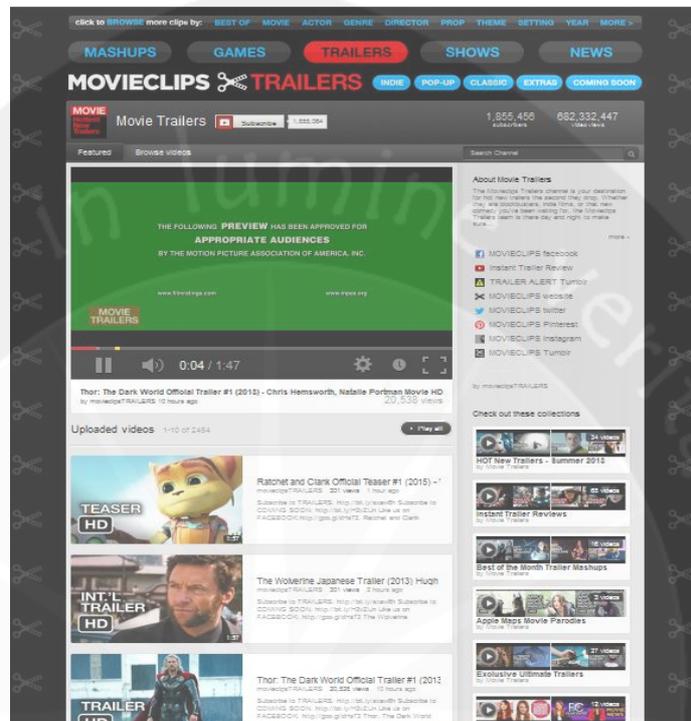
YouTube *Channel* adalah halaman pada YouTube dimana pengguna lain dapat saling terhubung. Pengguna lain dapat mengirim pesan, berlangganan video, dan mengetahui informasi pengguna lain atau perusahaan. *YouTube channel* juga merupakan tempat untuk melihat *statistic, channel views* untuk melihat sudah berapa lama pengguna aktif pada *website YouTube*, berapa video yang telah di unggah, berapa orang yang menyukai dan tidak menyukai video yang diunggah, serta sebagai sebuah tempat dimana pengguna mempunyai *control* pada halamannya sendiri. Tidak ada video yang tampil pada *channel* pengguna jika pengguna tidak menaruhnya disana (YouTube Confidential, 2011).

Sebuah *channel* pada YouTube merupakan *home page* untuk sebuah *account*. Pada *channel* ditampilkan nama, tipe *account*, video publik yang pengguna unggah, dan berbagai macam informasi yang dimasukkan oleh pengguna. Pengguna dapat menyesuaikan *background* dan skema warna dengan beberapa informasi yang diinginkan.

YouTube channel juga dapat menampilkan video favorit dari pengguna lain, aliran aktivitas, komentar, pelanggan, dan fitur jejaring sosial (Karch, 2011).

Ketika anda menjadi anggota YouTube, YouTube memberikan *channel* pribadi untuk anda. *Channel* ini memiliki divisi yang dirancang untuk menampilkan deskripsi singkat pribadi, *thumbnail* video yang telah diunggah, anggota dimana anda berlangganan videonya, video dari anggota lain yang telah anda pilih sebagai favaorit, *list* dari teman, pelanggan video anda, dan

tepat dimana pelanggan lain dapat memberikan komentar pada *channel* anda (Strickland, 2010).



Gambar 3.7. Channel dari movieClipsTrailers

3.7 Video Streaming

Video merupakan suatu media yang sangat penting untuk komunikasi dan hiburan selama puluhan tahun ini. Pertama kali, video diolah dan ditransmisikan dalam bentuk *analog*. Perkembangan di bidang komputer telah membantu terbentuknya video *digital*. Salah satu penerapan video *digital* yang digunakan dalam transmisi data adalah *video streaming*.

Video streaming adalah teknologi pengiriman data, video atau audio dalam bentuk yang telah dikompresi melalui jaringan *internet* yang ditampilkan oleh suatu *player* secara *realtime*. Pengguna memerlukan *player* yang merupakan aplikasi khusus untuk melakukan dekompresi

dan mengirimkan data berupa video ke tampilan layar monitor dan data berupa suara ke *speaker*. Sebuah *player* dapat berupa suatu bagian dari *browser* atau sebuah perangkat lunak. Inti dari *streaming* adalah membagi data dan *encoding*, kemudian mengirimkannya melalui jaringan dan pada saat data sampai pada pengguna maka akan dilakukan *decoding* serta pembacaan data. Ciri-ciri aplikasi *streaming* yaitu distribusi audio, video dan multimedia pada jaringan secara *realtime* atau *on demand*, *transfer* media data *digital* dari *server* dan diterima oleh pengguna sebagai *realtime stream* simultan sehingga pengguna tidak perlu menunggu keseluruhan data di *download* karena *server* mengirimkan data yang diperlukan setiap selang waktu tertentu. Hal ini memungkinkan pengguna untuk menjalankan *file content* seketika dengan periode *buffer* pendek.

Ada beberapa tipe *video streaming* antara lain *webcast*, dimana tayangan yang ditampilkan merupakan siaran langsung dan *Video on Demand (VOD)*, dimana tayangan yang akan ditampilkan sudah terlebih dahulu disimpan dalam *server*. Faktor-faktor yang mempengaruhi distribusi *video streaming* melalui jaringan antara lain: besarnya *bandwidth*, waktu tunda (*delay*), *lost packet*, dan juga teknik mendistribusikan video tersebut ke beberapa tujuan secara merata dan efisien (Apostolopoulos, 2002,p1).

3.8 Film

Film lahir sebagai penemuan teknologi. Dua bersaudara Lumiere, di Perancis menampilkan teknologi proyeksi gambar bergerak pada tahun 1895 dan sejak itu

dunia mengenal media baru yang bernama *film*.

Film merupakan karya seni budaya karena mencerminkan pola pikir dan gaya hidup dari masyarakat pembuatnya. Dengan kata lain, *film* merupakan medium ekspresi yang dikemas dengan selera seni yang indah dan memikat.

Film tidak termasuk cabang seni murni, *film* digolongkan sebagai seni aplikasi karena dominasi unsur teknik sangat kuat.

Picture adalah gambar sedangkan *image* merupakan gambaran. Gambar merupakan hasil rekaman dari sekedar apa yang kita lihat secara fisik, obyek sebagai obyek. Gambaran merupakan hasil rekaman dari hasil pemikiran tentang benda atau obyek yang kita lihat, bentuk notatif menjadi pengertian konotatif. Segala yang terlihat dan terdengar dalam layar *film* harus memiliki nilai gambaran atau *image* (Djarot, 2010).

3.9 Trailers

Trailer adalah bentuk unik dari pameran *film*, dimana wacana promosi dan kesenangan narasi disatukan (Kernan, 2004). Pada tahun 1989 Tom Gunning dalam diskusinya yang terkenal "*Cinema of Attractions*", ia menyebutkan bahwa *trailer* adalah sebuah bioskop dari atraksi yang akan datang, *trailer* merupakan iklan, tetapi juga memberikan rasa dari *film* yang akan tayang, menawarkan beberapa kesenangan pada saat pandangan pertama, makna, dan ide-ide. *Trailers* telah menjadi bagian yang penting dari sebuah *film*. *Trailer* menyampaikan dan memperkenalkan sebuah *film* dengan keajaiban media (Gray, 2006).

3.10 Symbian OS

Symbian adalah sistem operasi mobile (OS) dan platform komputasi yang dirancang untuk *smartphone* dan saat ini dikelola oleh Accenture. *Symbian* pada awalnya dikembangkan oleh *Symbian Ltd*, sebagai keturunan Psion EPOC dan berjalan secara eksklusif pada prosesor ARM, meskipun port x86 belum pernah dirilis (Vanesha, 2006). Status *Symbian* saat ini adalah platform open-source yang dikembangkan oleh *Symbian Foundation* pada tahun 2009, sebagai penerus dari *Symbian OS* asli. .

Versi *Symbian* yang terbaru adalah *Symbian ^3* dengan PR 02 yang disebut *Symbian belle*. *Symbian ^3* mendapat pembaruan pertama yang disebut *Symbian anna*. Digunakan di ponsel nokia x7-00 dan e6-00. sedangkan pembaruan terbaru disebut *Symbian belle*. semua ponsel *Symbian ^3* bisa diupgrade ke *Symbian belle* (Killip, 2012).

3.10.1 Arsitektur Sistem Operasi

Secara umum arsitektur *Symbian OS* sendiri dapat gambarkan menjadi empat lapisan berdasarkan penggunaan API (*Application Programming Interface*) yang tersedia (*Symbian Developer*, 2006), yaitu :

1. Lapisan pendukung aplikasi (*Application Utility Layer*)

Lapisan ini terdiri dari berbagai pendukung yang berorientasi pada aplikasi. Hal ini memungkinkan aplikasi lain (diluar sistem operasi) untuk berintegrasi dengan aplikasi dasar yang tersedia pada sistem operasi. Bentuk layanan lain termasuk proses pertukaran data dan manajemen data.

2. Lapisan layanan dan *framework* antarmuka grafis (*GUI Framework*)

Lapisan ini merupakan *framework* API yang tersedia untuk memberi dukungan terhadap penanganan *input user* secara grafis maupun suara yang dapat digunakan oleh aplikasi lain.

3. Lapisan komunikasi

Lapisan ini berfungsi sebagai sistem operasi yang fokus diimplementasi pada peralatan komunikasi *mobile*, *Symbian OS* memiliki kumpulan API yang fokus pada lapisan komunikasi. Bagian teratas pada lapisan ini terdapat dukungan pencarian dan pengiriman pesan teks. Berikutnya adalah antarmuka yang memberi dukungan komunikasi seperti *Bluetooth* dan *infrared (IrDA)* serta *USB*. Yang terakhir pada lapisan ini adalah protokol komunikasi berupa *TCP/IP*, *HTTP*, *WAP* dan layanan telepon.

4. Lapisan sistem API dasar

Lapisan ini merupakan kumpulan API yang mendukung pengaksesan data memori, tanggal dan waktu, serta sistem dasar lainnya.

3.10.2 Klasifikasi Sistem Operasi

Klasifikasi ini berdasar fungsionalitas dan hak akses dari API (*Application Programming Interface*) tertentu. Tujuan dari pendefinisian sistem ini selain untuk membedakan API mana saja yang bisa diakses oleh aplikasi yang dibuat oleh pihak pengembang aplikasi, juga tetap memelihara integrasi dari layanan yang disediakan bagi pihak pengembang aplikasi dengan API

yang umum digunakan. Hal ini juga dilakukan untuk memaksimalkan interoperabilitas antara berbagai produk yang menggunakan *Symbian OS*.

Terdapat empat kategori dalam klasifikasi API yang tersedia (*Symbian Developer, 2006*), yaitu:

1. (API) *Symbian* Umum

Komponen ini merupakan komponen (API) inti dari *Symbian OS*. Setiap pengembang aplikasi dapat berasumsi bahwa komponen ini terdapat pada setiap versi *Symbian OS* sehingga dapat digunakan pada setiap perangkat telepon bergerak yang menggunakan *Symbian OS* sebagai sistem operasinya. Dengan kata lain setiap kode program yang hanya menggunakan API pada kategori ini dapat dikompail dan dijalankan tanpa kesalahan pada setiap telepon yang menggunakan *Symbian OS*. Dengan adanya lisensi kerjasama, pengembang aplikasi dapat menambahkan dengan syarat tidak mengganti ataupun mengubah fungsi API standar yang dikategorikan pada bagian ini.

2. (API) *Symbian* Umum Tergantikan

Komponen yang memerlukan kostumisasi dari komponen *Symbian* Umum yang diperlukan untuk bekerja dengan ROM dari sistem dimana ia diinstal. Komponen ini merupakan komponen yang bekerja pada *low-level* dari *hardware* tertentu. Untuk mendapatkan komponen ini pihak pengembang aplikasi memerlukan lisensi dengan pihak *Symbian* karena versi komponen ini disediakan oleh pihak *Symbian*. Namun pada dasarnya komponen ini merupakan komponen standar (umum)

yang tersedia pada semua versi *Symbian* OS.

3. (API) *Symbian* Optional

Komponen-komponen ini sifatnya opsional (tidak selalu ada) pada semua versi *Symbian* OS. Namun jika tersedia, maka pengembang aplikasi mendapat jaminan bahwa aplikasinya dapat menggunakan API pada kategori ini pada versi *Symbian* OS yang sama.

4. (API) *Symbian* Opsional Tergantikan

Bentuk kategori ini mirip dengan kategori *Symbian* Opsional adalah kumpulan API yang tidak terikat dengan API umum yang ada pada versi *Symbian* OS dan dapat ditambahkan oleh pihak pengembang dengan suatu lisensi dari pihak *Symbian*.