DESAIN DAN IMPLEMENTASI SISTEM VIDEO ON DEMAND (VOD) BERBASIS WEB MENGGUNAKAN HTTP PSEUDOSTREAMING

SKRIPSI

TASI KOMPUTASI BERBASIS JAR

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun oleh:

PUTRI NORMA GUPITA NIM. 0810680051

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
PROGRAM STUDI TEKNIK INFORMATIKA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2013

LEMBAR PERSETUJUAN

DESAIN DAN IMPLEMENTASI SISTEM VIDEO ON DEMAND (VOD) BERBASIS WEB MENGGUNAKAN HTTP PSEUDOSTREAMING

SKRIPSI

ÓNSENTRASI KOLPITAS BEDAGIS JARINGAN

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun oleh:

PUTRI NORMA GUPITA NIM. 0810680051

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Himawat Aryadita S.T., M.Sc. NIP. 19801018 200801 1 003 <u>Sabriansyah R. Akbar S.T., M.Eng</u> NIK. 820809 06 1 1 0084

BRAWIJAYA

LEMBAR PENGESAHAN

DESAIN DAN IMPLEMENTASI SISTEM VIDEO ON DEMAND (VOD) BERBASIS WEB MENGGUNAKAN HTTP PSEUDOSTREAMING

SKRIPSI

MPÚTASI BERBASIS JARINGAN

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh:

PUTRI NORMA GUPITA NIM. 0810680051

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 17 Juli 2013

Penguji I

Penguji II

<u>Kasyful Amron, ST., M.Sc</u> NIP. 19750803 200312 1 003 Barlian Henryranu P, ST., MT NIK. 821024 06 11 0254

Penguji III

Adharul Muttaqin, ST., MT NIP. 19760121 200501 1 001

Mengetahui Ketua Program Studi Teknik Informatika

> <u>Drs. Marji, MT.</u> NIP. 19670801 199203 1 001

PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 26 Juli 2013 Mahasiswa,

Putri Norma Gupita
NIM 0810680051

PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Desain dan Implementasi Sistem Video on Demand (VOD) Berbasis Web Menggunakan HTTP Pseudostreaming" dengan baik. Skripsi ini diajukan untuk memenuhi persyaratan kelulusan dalam menyelesaikan pendidikan dan memperoleh gelar Sarjana Komputer konsentrasi Komputasi Berbasis Jaringan di Program Studi Teknik Informatika Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Malang.

Dalam menyelesaikan skripsi ini, penulis banyak mendapat bantuan dan dukungan dari berbagai pihak. Oleh karena itu penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada :

- 1. Ayahanda H. Hartoyo S. S.Pd. dan Ibunda Dra. Hj. Sulistyaningsih, beserta seluruh keluarga penulis yang selalu memberikan dukungan, nasehat dan doa.
- 2. Bapak Drs. Marji M.Si. selaku Ketua Program Studi Teknik Informatika Universitas Brawijaya dan Bapak Issa Arwani S.Kom., M.Sc. selaku Sekretaris Program Studi Teknik Informatika Universitas Brawijaya.
- 3. Bapak Himawat Aryadita S.T., M.Sc. selaku Dosen Pembimbing I sekaligus Dosen Pembimbing Akademik yang telah memberikan bimbingan, arahan dan motivasi hingga skripsi ini dapat terselesaikan.
- 4. Bapak Sabriansyah Rizqika Akbar S.T., M.Eng. selaku Dosen Pembimbing II yang telah memberikan bimbingan, arahan dan motivasi untuk penulisan skripsi ini.
- 5. Seluruh Dosen Jurusan Teknik Informatika yang telah bersedia membantu dalam membagi ilmunya kepada penulis.
- 6. Yuri Citra Pratama yang selalu memberikan motivasi dan dukungannya di setiap waktu, serta selalu memberikan saran dalam penyusunan skripsi ini.

- Sahabat sahabat penulis : Nanda Yustina, Dewi Rokhmah, Nur Hidayati, dan teman-teman Program Studi Teknologi Informasi angkatan 2008 yang selalu memberikan motivasi.
- 8. Mas Nyoman Wira atas bimbingannya dalam penyusunan skripsi ini.
- Teman-teman asisten dosen jaringan komputer atas dukungannya, beserta laboran Laboratorium Jaringan yang telah memberikan sarana dan prasarana yang memadai.
- 10. Seluruh Civitas Akademika Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Teknik Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.
- 11. Semua pihak yang tidak dapat disebutkan satu persatu yang telah banyak membantu dalam terselesaikannya skripsi ini.

Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan jauh dari sempurna, karena keterbatasan materi dan pengetahuan yang dimiliki penulis. Maka, saran dan kritik yang membangun dari semua pihak sangat diharapkan demi penyempurnaan selanjutnya. Semoga skripsi ini dapat bermanfaat dan berguna bagi semua pihak, baik penulis maupun pembaca, dan semoga Allah SWT meridhoi dan dicatat sebagai ibadah. Amin.

Malang, Juli 2013

Penulis

ABSTRAK

Putri Norma Gupita, 2013. Desain dan Implementasi Sistem Video On Demand (VOD) Berbasis Web Menggunakan HTTP Pseudostreaming. Skripsi Program Studi Teknik Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.

Dosen Pembimbing: Himawat Aryadita, S.T., M.Sc dan Sabriansyah R. Akbar, S.T., M.Eng.

Video merupakan komponen yang penting dalam teknologi multimedia. Video yang merepresentasikan gambar bergerak dan suara, layak untuk dijadikan sebagai alat bantu dalam proses belajar, ataupun karya berupa film, dan lain sebagainya. Diperlukan sebuah sistem khusus untuk menampung video karena kapasitasnya yang besar sehingga dapat meringankan beban platform *e-learning* seperti Moodle. Selain itu, pembangunan aplikasi yang dibuat dari awal dapat membuat pengembangan aplikasi menjadi lebih lama.

Oleh karena itu, pada penelitian "Desain dan Implementasi Sistem *Video On Demand* (VOD) Berbasis Web Menggunakan HTTP *Pseudostreaming*" akan dirancang sistem VOD berbasis web dengan menggunakan pendekatan *Component Based Software Engineering* (CBSE). Salah satu fitur utama yang dikembangkan dalam sistem ini adalah penggunaan HTTP *Pseudostreaming* yang memanfaatkan proses pemilihan acak pada video sehingga memudahkan pengguna dalam proses kontrol video.

Penelitian ini dilakukan dengan menguji fitur *pseudostreaming* serta menguji performa aplikasi terhadap *transfer rate* dan kualitas video pada sistem yang telah terhubung dengan Internet. Hasil menunjukkan bahwa sistem VOD yang dirancang dapat melakukan *pseudostreaming*, serta dapat mendeteksi *transfer rate* pengguna. Berdasarkan *transfer rate* tersebut, sistem akan memilih kualitas video yang direkomendasikan untuk pengguna sehingga mendapatkan kualitas *streaming* yang sesuai.

Kata Kunci: Video on Demand (VOD), pseudostreaming, metadata, FLV

ABSTRACT

Putri Norma Gupita, 2013. Design and Implementation System of Video On Demand (VOD) Web Based Using HTTP Pseudostreaming. Final Exam Informatic Engineering, Program of Information Technology and Computer Science, Brawijaya University.

Advisor: Himawat Aryadita, S.T., M.Sc and Sabriansyah R. Akbar, S.T., M.Eng.

Video is an important component in multimedia technology. Video representing moving image and sound deserves to be created as a tool in the learning process, or the creation in the form of movies, and so on. It requires a specific system to accommodate video because of its large capacity so can ease the burden of e-learning platforms such as Moodle. Besides, the development of application that was made from scratch can make application development becomes longer.

Therefore, in the study of "Design and Implementation System Video On Demand (VOD) Web Based Using HTTP Pseudostreaming" will be designed a web-based VOD system using the approach of Component Based Software Engineering (CBSE). One of the main features that were developed in this system is the use of HTTP pseudostreaming which utilize the random selection process in the video so it is easier for users to control the video.

The research was performed by testing pseudostreaming and application performance on the transfer rate and video quality on systems that are connected to the Internet. The results showed that the VOD system can perform pseudostreaming, also detect the user's transfer rate. Based on the transfer rate, the system will choose the video quality that is recommended for users to get the appropriate streaming quality.

Keywords: Video on Demand (VOD), pseudostreaming, metadata, FLV

DAFTAR ISI

	Halaman
PENGANTAR	
ABSTRAK	
ABSTRACT	iv
DAFTAR ISI	v
DAFTAR TABEL	ix
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	
BAB I PENDAHULUAN	4
1.1 Latar Belakang.	1
1.2 Rumusan Masalah	
1.3 Batasan Masalah	
1.4 Tujuan	
1.5 Manfaat	3
1.6 Sistematika Penulisan	4
BAB II DASAR TEORI	
2.1 Penelitian Terkait	5
2.2 Transmisi Multimedia	6
2.3 Video Streaming	6
2.4 Video on Demand	8
2.5 Arsitektur Video on Demand	10
2.6 Protokol Video on Demand	12
2.7 Metadata	19
2.8 Flash Video (FLV)	
2.9 Model Proses Perangkat Lunak	25
2.10 Component Based Software Engineeering (CBSE)	26
2.11 FFMPEG	
2.12 Yamdi	27

BAI	3 III	METO	DDE PENELITIAN	
				28
	3.2	Model	l Proses Perangkat Lunak	28
	3.3	Analis	sis Kebutuhan Fungsional	29
	3.4	Analis	sis Kebutuhan Non Fungsional	30
	3.5	Perano	cangan Sistem	30
	3.6	Implei	mentasi	31
	3.7	Pengu	ijian Sistem	31
	3.6	Analis	sis Hasil dan Pengambilan Kesimpulan	32
			En	
DAI) IX/	ANIATI	ISIS KEBUTUHAN DAN PERANCANGAN SISTEM	
DAI			aratan Sistem	
	7.1	4.1.1	Kebutuhan Antarmuka Eksternal	
		4.1.1	4.1.1.1 Antarmuka Pemakai	
			4.1.1.2 Antarmuka Perangkat Keras	
			4.1.1.3 Antarmuka Perangkat Lunak	
			4.1.1.4 Antarmuka Komunikasi	
		4.1.2	Analisis Basis Data	
		4.1.3	Analisis Kebutuhan Fungsional	
			4.1.3.1 Analisis Kebutuhan Data	
			4.1.3.2 Diagram Konteks	
		4.1.4	Analisis Kebutuhan Non Fungsional	
	4.2	Pencar	rian Komponen	
			Video Player	
		4.2.2	Encoder Video	39
		4.2.2	Injektor Metadata	40
	4.3	Modif	fikasi Komponen	40
	4.4	Perano	cangan Arsitektur	41
		4.4.1 I	Perancangan Arsitektur Sistem	41
		4.4.2 I	Diagram Konteks	
		4431	Data Flow Diagram	42

		4.4.4 Spesifikasi Proses	
		4.4.5 Perancangan Struktur Menu	63
		4.4.6 Perancangan Antar Muka	67
	4.5	Seleksi Komponen	75
	4.6	Perancangan Sistem	76
		4.6.1 Perancangan Basis Data	77
		4.6.1.1 Skema Relasi	77
		4.6.1.2 Perancangan Struktur Tabel	77
		4.6.1.2 Perancangan Struktur Tabel	81
2		MANY FIRM CO.	
BAH		MPLEMENTASI	
	5.1	Spesifikasi Sistem	
		5.1.1 Spesifikasi Perangkat Keras	
		5.1.2 Spesifikasi Perangkat Lunak	
		Konfigurasi Sistem	
	5.3	Implementasi Algoritma	
		5.3.1 Algoritma Create Filename	
		5.3.2 Algoritma Create Video Folder	94
		5.3.3 Algoritma FFmpeg Convert	96
		5.3.4 Algoritma Yamdi Inject	98
		5.3.5 Algoritma Speed Detection	
	5.4	Implementasi Antar Muka	101
		5.4.1 Implementasi Antar Muka Halaman <i>Home</i>	101
		5.4.2 Implementasi Antar Muka Halaman My Profile	102
		5.4.3 Implementasi Antar Muka Halaman Upload Video	102
		5.4.4 Implementasi Antar Muka Halaman Video	103
BAH		PENGUJIAN	Spr
	6.1	Pengujian Fungsional	105
		6.1.1 Pengujian Black Box	105

	6.1.2	Pengujian	Kualitas	Video	dan	Transfer	Rate	Melalui	Traffic
	Shapi	ng							108
6.2	Pengu	ijian Non F	ungsional						110
BAB VII	PENU	JTUP							
7.1	Kesin	npulan							114
7.2	Saran								115
DAFTAI LAMPII		ГАКА	51T	AS	I	BRA	14		DP-1 L-1

DAFTAR TABEL

No.	Judul	Halaman
Tabel 2.1	Perbandingan RTSP dan HTTP	14
Tabel 2.2	Tabel FLV Header	20
Tabel 2.3	Tabel FLV Body	21
Tabel 2.4	Tabel FLV Tag	21
Tabel 2.5	Tabel VideoTagHeader	22
Tabel 2.6	Tabel FLV <i>Tag</i> Tabel VideoTagHeader Tabel VideoTagBody Tabel OnMetaData Spesifikasi Proses	23
Tabel 2.7	Tabel OnMetaData	24
Tabel 4.1	Spesifikasi Proses	48
Tabel 4.2	Struktur Tabel Video	78
Tabel 4.3	Struktur Tabel User	78
Tabel 4.4	Struktur Tabel Category	79
Tabel 4.5	Struktur Tabel Course	79
Tabel 4.6	Struktur Tabel Video_comment	79
Tabel 4.7	Struktur Tabel Favourite	80
Tabel 4.8	Struktur Tabel user_log	80
Tabel 4.9	Struktur Tabel Login	80
Tabel 4.10	Struktur Tabel course_video_order	80
Tabel 5.1	Spesifikasi Perangkat Keras Server	90
Tabel 5.2	Spesifikasi Perangkat Keras Client	91
Tabel 5.3	Spesifikasi Perangkat Lunak Server	91
Tabel 5.4	Spesifikasi Perangkat Lunak Client	92
Tabel 5.5	Implementasi Algoritma Create Filename	93
Tabel 5.6	Implementasi Algoritma Create Video Folder	95
Tabel 5.7	Implementasi Algoritma Create FFmpeg Convert	97
Tabel 5.8	Implementasi Algoritma Yamdi Inject	98
Tabel 5.9	Implementasi Algoritma Speed Detection	99
Tabel 6.1	Skenario Uji Coba Sistem	105
Tabel 6.2	Hasil Pengujian Sistem	106

Tabel 6.3	Pembagian <i>Transfer Rate</i> terhadap Kualitas Video	109
Tabel 6.4	Pengujian Transfer Rate dan Kualitas Video	109
Tabel 6.5	Pengujian pada provider Tri (3)	110
Tabel 6.6	Pengujian pada provider Axis	111
Tabel 6.7	Pengujian pada provider Smartfren	112
Tabel 6.8	Pengujian pada provider Indosat	112



DAFTAR GAMBAR

No.	Judul	Halaman
Gambar 2.1	Framework Sistem VOD	5
Gambar 2.2	Cara Kerja Video Streaming	8
Gambar 2.3	On Demand Streaming	9
Gambar 2.4	Sistem VOD Tersentralisasi	11
Gamabr 2.5	Sistem VOD Tersentralisasi Sistem VOD Terdistribusi Client-Server Network Architecture Mekanisme Protokol RTSP Ilustrasi Progressive Download	11
Gambar 2.6	Client-Server Network Architecture	12
Gambar 2.7	Mekanisme Protokol RTSP	14
Gambar 2.8	Ilustrasi Progressive Download	15
Gambar 2.9	Ilustrasi Adaptive HTTP Streaming	17
Gambar 3.1	CBSE dengan Pemakaian Ulang	29
Gambar 3.2	Proses Identifikasi Komponen	29
Gambar 3.3	Sistem VOD yang akan dibangun	30
Gambar 3.4	Persyaratan Sistem VOD	31
Gambar 3.5	Flowchart Perancangan VOD	33
Gambar 4.1	Entity Relationship Diagram	37
Gambar 4.2	Arsitektur Sistem	41
Gambar 4.3	Diagram konteks	42
Gambar 4.4	DFD level 0 Aplikasi Video On Demand Berbasis Web	43
Gambar 4.5	DFD Level 1 Proses 1 Login	44
Gambar 4.6	DFD Level 1 Proses 2 Pengolahan Video	45
Gambar 4.7	DFD Level 1 Proses 3 Pengolahan User	46
Gambar 4.8	DFD Level 1 Proses 4 Upload Video	46
Gambar 4.9	DFD Level 1 Proses 6 Pencarian & Pemilihan Member	47
Gambar 4.10	DFD Level 1 Proses 7 Register	47
Gambar 4.11	DFD Level 1 Proses 8 Pengolahan Course	48
Gambar 4.12	Struktur Menu Member	64
Gambar 4.13	Struktur Menu Pengunjung	65
Gambar 4.14	Struktur Menu Admin	66

Gambar 4.15	Layout Menu Home	68
Gambar 4.16	Layout Menu Courses	68
Gambar 4.17	Layout Menu Videos	69
Gambar 4.18	Layout Menu My Profile	69
Gambar 4.19	Layout Menu Upload	70
Gambar 4.20	Layout Menu Create Course	70
Gambar 4.21	Layout Upload Video Course	71
Gambar 4.22	Layout Menu Signup	71
Gambar 4.23	Layout Menu My Videos	72
Gambar 4.24	Layout Menu Signup Layout Menu My Videos Layout Menu My Favorite Videos Layout Menu My Courses Layout Menu Account Edit	72
Gambar 4.25	Layout Menu My Courses	73
Gambar 4.26	Layout Menu Account Edit	_73
Gambar 4.27	Layout Menu Change Password	74
Gambar 4.28	Layout Menu Change Avatar	74
Gambar 4.29	Skema Relasi Database	77
Gambar 4.30	Flowchart Proses Login	81
Gambar 4.30	Flowchart Proses Pencarian Data	82
Gambar 4.31	Flowchart Proses Registrasi	83
Gambar 4.32	Flowchart Proses Menambah Data	84
Gambar 4.33	Flowchart Proses Mengedit Data	85
Gambar 4.34	Flowchart Proses Menghapus Data	86
Gambar 4.35	Flowchart Proses Upload Video	87
Gambar 4.36	Flowchart Proses Create Course	88
Gambar 4.37	Flowchart Proses Injeksi Metadata	89
Gambar 5.1	Tampilan Halaman <i>Home</i>	101
Gambar 5.2	Tampilan Halaman My Profile	102
Gambar 5.3	Tampilan Halaman Upload Video	103
Gambar 5.4	Tampilan Halaman Video	104

DAFTAR LAMPIRAN

No.	Judul	Halaman
Lampiran 1	Implementasi Basis Data	L-1
Lampiran 2	Konfigurasi Apache	L-6
Lampiran 3	Konfigurasi php.ini	L-7
Lampiran 4	Konfigurasi Flowplayer pada script html	L-8



BAB I PENDAHULUAN

1.1 Latar Belakang

Saat ini telah berkembang suatu platform yang khusus menangani masalah *e-learning*, yaitu CMS Moodle. Moodle (*Modular Object-Oriented Dynamic Learning Environment*) adalah sebuah nama untuk sebuah program aplikasi yang dapat merubah sebuah media pembelajaran ke dalam bentuk web. Dengan Moodle, dapat dibuat materi pembelajaran, kuis, jurnal elektronik dan lain-lain. Akan tetapi Moodle juga mempunyai kapasitas yang terbatas jika digunakan untuk menampung hasil pembelajaran berupa video. Maka dari itu, perlu adanya suatu tempat khusus untuk menampung hasil pembelajaran atau karya apapun berupa video khusus bagi para siswa ataupun pengajar. Untuk meringankan beban tersebut, maka dibuat aplikasi tersendiri yang khusus menampung data berupa video, yaitu *Video on Demand* (VOD).

Di sisi lain, seiring dengan pesatnya teknologi informasi, para pengembang aplikasi bersaing membuat aplikasi dan dituntut untuk menghadapi perubahan yang cepat. Berbagai permasalahan yang timbul seperti mempelajari bahasa pemrograman dan mencari bahasa yang cocok, membuat kode program, bahkan sering sebuah bagian dari program menjadi tidak terpakai karena menemukan lagi pendekatan baru yang lebih efektif, sehingga membuat pengembangan aplikasi menjadi lama. Untuk itu muncul sebuah pendekatan, yaitu *Component Based Software Engineering* (CBSE).

Pada CBSE, pihak pengembang tidak perlu membangun semua kebutuhan aplikasi dari awal, melainkan tinggal menggabungkan beberapa komponen yang tersedia menjadi satu aplikasi yang diinginkan. Kelebihan sebuah komponen adalah komponen dapat diganti atau dibuat menjadi lebih terkini tanpa harus mempengaruhi komponen lain dalam sebuah aplikasi, selama komponen baru tersebut masih mengikuti set *interface* yang sama.

Pendekatan CBSE merupakan salah satu alternatif untuk mendukung aplilkasi *Video on Demand* (VOD), terutama yang mempunyai fitur *pseudostreaming*. Keuntungan terbesar menggunakan *pseudostreaming* dari

BRAWIJAY

solusi *streaming* lainnya adalah kemampuan untuk mencari pada *timeline* ke posisi yang belum di-*download* oleh *player* [ROM-11]. Fitur ini diperlukan bagi sebagian besar pengguna video karena umumnya pengguna hanya ingin melihat sekilas video tersebut atau umumnya pengguna sudah penah melihat beberapa menit sebelumnya, sehingga pengguna dapat langsung melihat ke menit setelahnya. Untuk memenuhi fitur tersebut, diperlukan beberapa komponen dengan tingkat independensi yang cukup tinggi. Salah satu komponen utamanya adalah *video player*, serta komponen penunjang lainnya.

Dalam mengembangkan sistem VOD dituntut ketelitian dari pengembang aplikasi dalam mengimplementasikan komponen. Komponen yang dipakai harus dapat digunakan secara universal sehingga pihak lain dapat menggunakan kembali komponen tersebut dengan mudah.

Mengacu pada permasalahan yang ada, maka disini penulis akan mengangkat judul skripsi tentang "Desain dan Implementasi Sistem *Video On Demand* (VOD) Berbasis Web Menggunakan HTTP *Pseudostreaming*"

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, penulis merumuskan beberapa masalah yaitu sebagai berikut :

- 1 Bagaimana merancang aplikasi video *streaming* dengan karakteristik *Video on Demand (VOD)* menggunakan model pengembangan perangkat lunak *Component Based Software Engineering* (CBSE).
- 2 Bagaimana menerapkan VOD dengan model pseudostreaming.
- Bagaimana model pengujian fungsional dan non-fungsional pada aplikasi *VOD*.
- 4 Bagaimana menerapkan proses pendeteksian transfer rate sehingga dapat memperoleh kualitas video yang sesuai.

1.3 Batasan Masalah

Untuk menghindari adanya kemungkinan semakin berkembangnya masalah, maka penulisan laporan hanya menitikberatkan hal-hal berikut ini :

 Penelitian yang dilakukan hanya untuk streaming VOD berbasis web dan menggunakan HTTP Pseudostreaming sebagai teknik streaming videonya.

- Streaming dilakukan secara unicast dari server ke klien
- File video yang distreamingkan berekstensi FLV.

1.4 Tujuan

Tujuan dibuatnya skripsi ini adalah :

Untuk merancang aplikasi video *streaming* dengan karakteristik VOD menggunakan model pengembangan perangkat lunak *Component Based Software Engineering* (CBSE) dan menerapkan *VOD* dengan model *pseudostreaming*, serta mengetahui model pengujian fungsional dan non fungsional pada aplikasi *VOD*.

1.5 Manfaat

Manfaat dari pembuatan skripsi ini adalah:

1. Umum

Memberikan sebuah alternatif dalam pembangunan sebuah aplikasi VOD dengan biaya murah, cepat, serta tetap handal dengan pengelolaan yang mudah.

2. Khusus

- a. Mahasiswa
 - Sebagai salah satu referensi belajar bagi mahasiswa untuk meningkatkan pengetahuannya dalam bidang video *streaming* khususnya VOD.
 - Sebagai bahan acuan bagi mahasiswa lain yang akan melaksanakan penelitian tugas akhir dengan tema yang sama.

b. Penulis

Menambah pengetahuan serta pengalaman baru dalam membuat suatu aplikasi VOD dan dapat mengembangkan ke arah yang lebih baik.

1.6 Sistematika Penulisan

Sistematika penulisan dalam skripsi ini sebagai berikut:

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi pembahasan, dan sistematika penulisan.

BAB II Dasar teori

Berisikan dasar teori untuk menunjang penyelesaian masalah dan dijelaskan landasan teori dari poin – poin penting yang digunakan dalam proyek akhir ini.

BAB III Metode Penelitian

Membahas metode yang digunakan dalam penulisan yang terdiri dari studi literatur, analisis kebutuhan non fungsional, analisis kebutuhan fungsional, perancangan sistem, pengujian sistem, analisis hasil dan pengambilan kesimpulan

BAB IV Analisis Kebutuhan dan Perancangan Sistem

Membahas analisis kebutuhan dan perancangan "Desain dan Implementasi Sistem *Video On Demand* (VOD) Berbasis Web Menggunakan HTTP *Pseudostreaming*"

BAB V Implementasi

Membahas tentang implementasi dari sistem.

BAB VI Pengujian

Memuat proses dan hasil pengujian terhadap sistem yang telah direalisasikan.

BAB VII Penutup

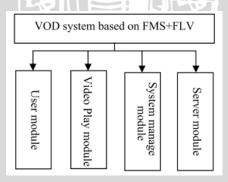
Memuat kesimpulan serta saran untuk pengembangan lebih lanjut.

BAB II DASAR TEORI

2.1 Penelitian Terkait

Penelitian mengenai sistem *video on demand* telah dilakukan oleh Xu Xiaotian dengan judul "*RTSP-Based Video on Demand System*" tahun 2002, dimana penelitian tersebut mengimplementasikan aplikasi server klien VOD berbasis RTSP. Aplikasi tersebut diimplementasikan dengan JMF (Java Media Framework), dikombinasikan dengan web server HTTP dan teknologi Java Web Start. Aplikasi klien-server yang dibuat dapat menjalankan 6 fungsi, yakni *describe, options, setup, play, pause* dan *teardown*.

Pada paper yang disusun oleh Fang Jin berjudul "Design and Implementation Video on Demand System based on FMS +FLV" tahun 2010, Fang Jin menerapkan sistem IVOD (Interactive Video on Demand) yang dikembangkan dengan ASP.NET dan SQL Server 2005, berbasis FLV (Flash Video) dan FMS (Flash Media Server).. Aplikasi yang dibuat mengijinkan pengguna untuk melakukan operasi interaktif seperti play, pause, fast forward, rewind, dan lain-lain. Fungsi tersebut diatas dibuat pada video play module. Berikut pada gambar 2.1 adalah gambaran sistem yang dibuat.



Gambar 2.1 Framework Sistem VOD

Sumber: (JIN-11)

Perbedaan penelitian yang dilakukan oleh Fang Jin dengan Xu Xiaotian adalah pada penelitian yang dilakukan Xu Xiaotian sistem VOD dikembangkan dengan menggunakan protokol RTSP sedangkan penelitian

yang dilakukan Fang Jin sudah dapat mengimplementasikan aplikasi VOD tersebut dengan berbasis protokol HTTP dengan menggunakan server FMS.

Pada tugas akhir ini saya akan menerapkan sistem VOD yang dikembangkan dengan menggunakan protokol HTTP yang hanya memerlukan web server saja tanpa menggunakan streaming server khusus dengan tujuan efisisen dan murah dalam penerapannya.

2.2 Transmisi Multimedia

Ada dua jenis cara data multimedia dapat ditransmisikan dalam Internet, yaitu : [KOZ-02]

- *Download mode*, klien dapat memainkan media setelah semua *file* media telah dilakukan proses *download* dari *server* ke komputernya. Sebagai contoh adalah jika melakukan *download* dari http://www.mp3.com.
- Streaming mode, klien dapat memainkan media secara langsung tanpa melakukan proses download. Bagian media yang diterima melalui proses transmisi dapat langsung dimainkan seketika itu juga. Sebagai contoh adalah jika melakukan streaming dari http://www.justin.tv

Ada juga metode ketiga, yaitu progressive download. Media dapat dimainkan beberapa detik setelah proses download dimulai atau klien dapat melihat media selagi media itu dalam proses download. Secara langsung terlihat seperti streaming tetapi kenyataannya adalah melakukan download. Istilah lainnya juga menyebutnya sebagai pseudostreaming, hanya saja perbedaannya pseudostreaming mempunyai fitur random seeking. Sebagai contoh adalah jika mengakses video pada http://www.youtube.com.

Pengertian *streaming* sendiri adalah sebuah teknologi untuk memainkan file video atau audio secara langsung ataupun dengan pre-recorder dari sebuah mesin server (web server). Dengan kata lain, file video ataupun audio yang terletak dalam sebuah server dapat secara langsung dijalankan pada komputer klien sesaat setelah ada permintaan dari pengguna, sehingga proses running aplikasi yang didownload berupa waktu yang lama dapat dihindari tanpa harus melakukan proses penyimpanan terlebih dahulu. [SEP-09]

Pada saat file video atau audio di *stream*, akan terbentuk sebuah *buffer* di komputer klien, dan data video audio tersebut akan mulai di *download* ke dalam *buffer* yang telah terbentuk pada mesin klien. Dalam waktu sepersekian detik, *buffer* telah terisi penuh dan secara otomatis file video audio dijalankan oleh sistem. Sistem akan membaca informasi dari *buffer* dan tetap melakukan proses *download* file, sehingga proses *streaming* tetap berlangsung ke mesin klien.

Perkembangan *streaming* dipengaruhi oleh tiga faktor, yakni: [KOZ-02]

- a. Kemajuan dalam algoritma kompresi untuk audio dan video
- b. Perkembangan dari streaming server
- c. Peningkatan pada jaringan broadband dan kabel modem.

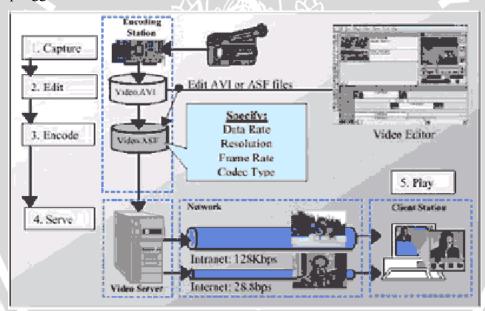
Berikut adalah fitur utama streaming: [KOZ-02]

- a. Dapat memberikan konten *live* seperti pertandingan sepak bola, konser, atau pidato politik.
- b. Dapat menyediakan akses acak untuk film berdurasi panjang. Server streaming dapat bertindak sebagai remote video player dan menyajikan beberapa fungsi VCR (misal *skip back* dan sebagainya, dan dapat menonton hanya pada beberapa bagian tanpa men*download* seluruh film).
- c. Tidak menempati ruang pada hardisk pengguna. Pengguna tidak mendapatkan salinan dari file media yang tinggal pada server *streaming*. Pengguna dapat menyimpan media, tetapi apa yang sebenarnya disimpan adalah URL dari stream tersebut, titik saat ini pada *timeline* media, dan pengaturan (seperti volume suara)
- d. Hanya menggunakan *bandwidth* jaringan yang dibutuhkan. Jika konten streaming melebihi kecepatan koneksi, beberapa paket data akan hilang dan konten akan rusak.
- e. Mendapatkan keuntungan ketika menggunakan pendekatan *broadcast* dan *multicast* (dimana satu aliran dapat dikirim ke banyak pengguna).

2.3 Video Streaming

Prinsip kerja video *streaming* pada awalnya adalah data dari *source* (audio atau video) akan di-*capture* dan disimpan pada sebuah *buffer* yang berada pada memori komputer (bukan media penyimpanan seperti harddisk) dan kemudian di-*encode* sesuai dengan format yang diinginkan [HAN-09].

Dalam proses *encode*, pengguna dapat mengkompresi data sehingga ukurannya tidak terlalu besar (bersifat optional). Namun pada aplikasi streaming yang menggunakan jaringan, biasanya data akan dikompresi terlebih dahulu sebelum dilakukan streaming, karena keterbatasan *bandwidth* jaringan. Setelah di-*encode*, data akan di-*stream* ke pengguna yang lain. Pengguna akan melakukan *decode* data dan menampilkan hasilnya ke layar pengguna.



Gambar 2.2 Cara Kerja Video Streaming

Sumber: [CAL-02]

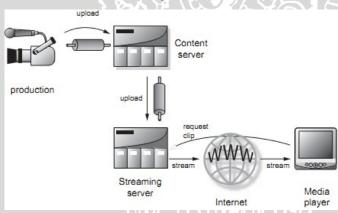
2.4 Video On Demand

VoD (*Video On Demand*) adalah aplikasi dimana klien melakukan request on-demand terhadap file audio atau video terkompresi yang telah tersimpan pada server. Umumnya pada aplikasi stored audio video, setelah delay beberapa detik klien baru bisa memulai untuk memainkan data audio

video yang dimintanya dan pada saat itu juga klien menerima data dari server secara terus-menerus (*continues*). Secara ideal, *delay* saat klien meminta file audio video hingga klien menerima data (dapat mendengarkan dan memainkan file audio video) seharusnya pada 1 sampai 10 detik. Persyaratan paket *delay* dan *jitter* tidak sebegitu ketat seperti aplikasi *real-time* seperti *internet telephony* dan *video conference*. [KUR-03]

Beberapa aplikasi *video on demand* menyediakan fitur interaktif seperti *fast forward, rewind,* dan lain-lain. Contoh aplikasi *video on demand* adalah Youtube, Vimeo, Hulu, BBC iPlayer, dan lain sebagainya.

Pada saat penonton menonton program TV dan film secara pasif, penonton tidak dapat berinteraksi dengan program dan tidak dapat mengontrol waktu ketika menonton program TV. Layanan video berdasarkan permintaan (VoD) telah dan sedang dikembangkan untuk memenuhi keterbatasan-keterbatasan yang disebutkan diatas.



Gambar 2.3 On Demand Streaming
Sumber: [AUS-05]

Pada VoD, sejumlah koleksi film disimpan pada server video. Pengguna mengakses video melalui jaringan. Beberapa kelebihan VoD adalah : [SYA-11]

- Aplikasi video yang disediakan *user-friendly*.
- Jika satu server video tidak memiliki video yang diinginkan, maka akan dibawa ke server lain.
- Dapat menonton video favorit kapanpun.

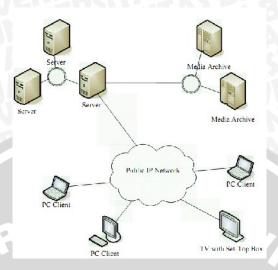
BRAWIJAYA

- Dapat memberhentikan sesaat (*pause*) dan mempercepat (*fast-forward*) dan mundur (*backward*) dan bisa juga mencari sebuah *scene* tertentu.
- Memiliki kualitas jaminan yang tinggi karena video disimpan secara digital. Kualitas tidak akan menurun sejalan dengan peningkatan jumlah pengguna yang menonton video.

Karena potensi pasar yang sangat besar, banyak perusahaan dan organisasi sedang melakukan penelitian dan percobaan mengenai sistem VoD.

2.5 Arsitektur Video on Demand

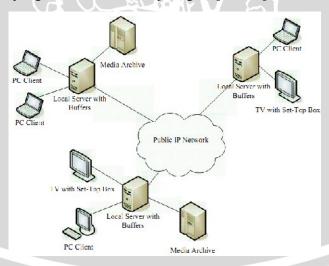
Seperti pada sistem jaringan lainnya, VOD dapat dirancang sebagai sistem multimedia tersentralisasi atau terdistribusi. Sistem VOD tersentralisasi menempatkan *processing* server dan media arsip berada dalam satu situs sebagai node pusat. Permintaan dari klien diproses pada node pusat, dan video yang diminta dikirimkan melalui jaringan ke situs klien. Gambar 2.4 menggambarkan sebuah arsitektur sistem tersentralisasi. VOD yang tersentralisasi mempunyai sistem yang sederhana untuk mengelola, tetapi skalabilitasnya kecil, *delay* jaringan lama, dan *throughput* rendah. Kinerja sistem tersentralisasi VOD dapat ditingkatkan jika server lokal ditambahkan. Server lokal ini memiliki buffer video, tetapi tidak mempunyai media *archive*. Film populer dapat disimpan dalam *buffer* video lokal sehingga mereka dapat dikirim ke klien lebih cepat. Video yang tidak berada di lokasi buffer lokal dapat dikirim ke klien dari arsip pusat ketika diminta. [KO-96]



Gambar 2.4 Sistem VOD Tersentralisasi

Sumber: [FLO-06]

Sebuah sistem terdistribusi VOD memiliki server pemrosesan lokal dan media arsip. Permintaan klien ditangani oleh server lokal (Gambar 2.5). Jika film yang diminta tidak di arsip lokal, server lokal dapat meminta film dari server terpencil yang terletak di seluruh jaringan. [KO-96]



Gambar 2.5 Sistem VOD Terdistribusi

Sumber: [FLO-06]

Arsitektur sistem dari sebuah *Video on Demand* pada dasarnya terdiri dari tiga bagian utama: klien, jaringan, dan server. Setiap bagian dapat dibagi menjadi komponen-komponen yang lebih kecil. Model ini disebut arsitektur "client-server". Arsitektur client-server adalah arsitektur yang paling luas digunakan dalam Internet [FLO-06]. Gambar 2.6 menggambarkan komunikasi antara klien dan server.



Gambar 2.6 Client-Server Network Architecture

Sumber: [FLO-06]

2.6 Protokol Video On Demand

Untuk membangun VOD ada beberapa protokol yang dapat digunakan. Dua diantaranya yakni RTSP (*Real Time Streaming Protocol*) dan HTTP (*Hypertext Transport Protocol*). Berikut penjelasan dari masing-masing protokol:

a. RTSP (Real Time Streaming Protocol)

RTSP adalah protokol level aplikasi untuk kontrol data *real-time streaming* yang menggunakan RTP sebagai protokol pengiriman data yang menawarkan kontrol seperti VCR untuk pengguna, yaitu *Play, Stop, Pause,* FF, dan REW, serta akses acak untuk bagian manapun dari media klip. [KOZ-02]

Fungsi penting dari RTSP adalah kemampuannya untuk memilih delivery channel yang optimal kepada klien. Misalnya, jika UDP tidak dapat digunakan, server streaming akan menawarkan pilihan protokol delivery – multicast UDP atau TCP untuk menyesuaikan klien yang berbeda-beda. RTSP juga membantu server untuk mengatur bandwidth media untuk kemacetan jaringan dalam rangka untuk memenuhi kapasitas yang tersedia.

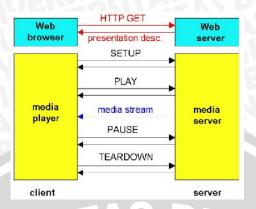
Berikut adalah hal-hal yang tidak dilakukan oleh protokol RTSP [KUR-03]:

- 1) RTSP tidak mendefinisikan skema kompresi untuk audio dan video
- 2) RTSP tidak mendefinisikan bagaimana audio dan video dienkapsulasi di paket untuk transmisi pada jaringan. Enkapsulasi untuk media streaming dapat disediakan oleh RTP atau oleh protokol khusus.
- 3) RTSP tidak membatasi bagaimana media distreamingkan, dapat menggunakan UDP atau TCP.
- 4) RTSP tidak membatasi bagaimana media *player* mem-*buffer* audio atau video. Audio atau video dapat dimainkan segera setelah sampai di klien, atau dapat dimainkan setelah delay beberapa detik, atau dapat didownload secara keseluruhan sebelum dimainkan.

RTSP mempunyai 4 perintah yang dikirim dari klien kepada server streaming. Keempat perintah tersebut adalah:

- 1) Setup, server mengalokasikan sumber daya kepada klien.
- 2) *Play*, server mengirim sebuah stream ke sesi klien 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.

Empat perintah RTSP dapat dilakukan setelah klien dan server sudah berada dalam session yang disepakati. Klien melakukan proses DESCRIBE dan server merespon dengan protokol SDP (Session Description Protocol). Untuk lebih jelasnya mekanisme protokol RTSP ada pada Gambar 2.7.



Gambar 2.7 Mekanisme Protokol RTSP

Sumber: [KUR-03]

b. HTTP (Hypertext Transport Protocol)

HTTP adalah sebuah protokol dimana file HTML bisa ditransfer antar klien dan server. Paket HTTP diproteksi dengan *checksum*. Server dan komputer klien memiliki koneksi dua arah, yang berarti bahwa ada umpan balik dari klien. Dengan demikian, paket yang rusak dapat ditransmisikan kembali, sehingga file yang diterima klien dalam keadaan utuh. [KOZ-02]

HTTP juga dapat digunakan untuk media *download*, terutama jika filenya kecil dan jumlah klien secara bersamaan terbatas. Jika kecepatan koneksi lebih rendah daripada data *rate* media, media masih dapat dimainkan tapi tidak berjalan dengan lancar. Waktu transfer file *download* tergantung pada ukuran file dan kecepatan koneksi.

Setelah mengetahui penjelasan dari protokol HTTP dan RTSP, dapat kita bandingkan kelebihan dan kekurangan masing-masing protokol pada tabel 2.1 berikut:

Tabel 2.1 Perbandingan RSTP dan HTTP

RTSP	HTTP
(Real Time Transfer Protocol)	(Hypertext Transport Protocol)
Interoperability yang sangat kurang	Memiliki interoperability yang luas
	baik pada sisi server maupun klien.
Mempunyai command play/pause	Ada cara khusus agar HTTP dapat
secara built-in	menghentikan streaming

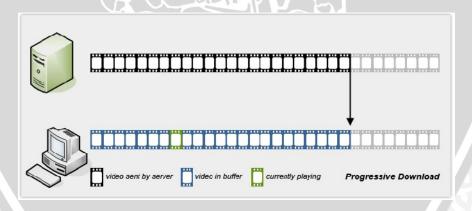
Membutuhkan streaming server	Hanya membutuhkan standart web
	server
Menggunakan UDP (paket dapat	Menggunakan TCP (high quality
hilang/terganggu)	delivery)
Mempertimbangkan packet loss	Tidak ada packet loss
Mempertimbangkan bandwidth user	Pengiriman data tidak tergantung
	pada bandwidth user
Terkendala oleh firewall (akses	Tidak terkendala oleh firewall (port
diblok)	80)

Sumber: [COU-09]

Untuk pengiriman video melalui HTTP, terdapat 3 teknik yang dapat digunakan pada video *streaming*, antara lain :

• Progressive Download

Progressive Download adalah sebuah metode yang menggabungkan metode download-and-play dan streaming. Ide dasarnya adalah file dapat dibuka sementara sisanya masih mendownload, sehingga tidak perlu menunggu file didownload lengkap. [AUS-05]



Gambar 2.8 Ilustrasi Progressive Download

Sumber: [WIJ-11]

Kelebihan menggunakan metode progressive dwnload adalah:

- Mudah dikonfigurasi (dari sudut pandang penyedia).
- Video yang sedang berjalan dapat dihentikan dan dibuffer. (dari sudut pandang pengakses).

- Lebih baik digunakan jika file videonya mempunyai durasi yang pendek.

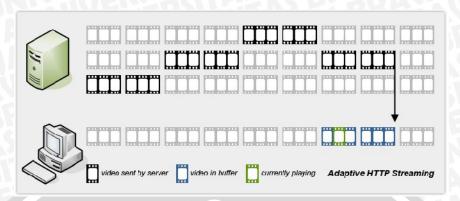
Kekurangan menggunakan metode ini adalah:

- Penggunaan bandwidth yang sia-sia jika pengguna memutuskan untuk berhenti menonton konten video setelah *progressive download* telah dimulai (misalnya, beralih ke konten lain).
- Seluruh video yang telah di*download* oleh klien, isinya tidak terproteksi jika dilihat dari sudut pandang penyedia.
- Video hanya dapat dicari hanya pada bagian yang ter*download*. Dengan kata lain, jika seorang pengakses ingin melihat 40-45 menit pada video berdurasi 1 jam, ia perlu menunggu video untuk men*download* 40 menit pertama.
- Tidak dapat merubah kualitas video ketika sudah dimainkan. Saat download telah dimulai, kualitas video sudah dikunci (sudah fix). Ketika mengganti *player* ke *fullscreen*, maka yang terlihat adalah video yang buram, karena telah ditentukan untuk dilihat pada ukuran yang lebih kecil.

Jika anda melihat *unseekable* video pada website, dan video dapat ditemukan pada cache browser anda, maka website tersebut menggunakan HTTP *Progressive download*.

• Dynamic Adaptive Streaming HTTP (DASH)

Metode ini merupakan teknologi terbaru. Metode ini menggabungkan manfaat dari RTMP / RTSP *Streaming* (efisiensi bandwidth, berganti kualitas video) dengan *progresif download* (tidak ada *streaming* server khusus atau protokol yang diperlukan). *Adaptive HTTP Streaming* bekerja dengan menyimpan video pada server dalam bagian-bagian kecil dan mengirimkannya melalui HTTP. Bagian-bagian kecil tersebut lalu kemudian digabung oleh video *player*. [WIJ-11]



Gambar 2.9 Ilustrasi Adaptive HTTP Streaming
Sumber: [WIJ-11]

Implementasi yang menggunakan metode *Adaptive Streaming* bermacam-macam, seperti HTTP *Live Streaming* (Apple), *Smooth Streaming* (Microsoft), HTTP *Dynamic Streaming* (Adobe), *Adaptive Bitrate* (Octoshape). [ROM-11]

Kelebihan dari DASH adalah:

- Tidak ada *streaming* server khusus atau protokol yang diperlukan.
- Proteksi yang lebih baik untuk isi dari streaming karena video telah dibagi menjadi bagian-bagian kecil.
- Efisiensi bandwidth.

Kekurangan menggunakan DASH adalah:

- Membutuhkan dukungan dari sisi klien, seperti Adobe Flash Player yang support HTTP *Dynamic Streaming* setelah 10.1.
- Kurangnya standarisasi dan kompatibilitas karena masih teknologi baru dan belum luas digunakan,
- Untuk *live streaming* terdapat *delay* yang lama dibandingkan dengan teknologi *live streaming* yang biasa (RTP, RTMP).
- Terkendala oleh software yang berbayar (*trial* 30 hari).
- Tidak ada implementasi dari *Adaptive Streaming* yang dapat bekerja dengan file MP4 biasa. File mp4 harus dikonversi dahulu ke dalam format terfragmentasi tertentu,

BRAWIJAYA

• Pseudo Streaming

HTTP *pseudostreaming*, sesuai namanya, bukanlah *streaming* yang sebenarnya, tetapi berbentuk *progressive download* yang memungkinkan pengguna untuk mencari bagian pada video yang belum di*download*. Protokol yang digunakan adalah HTTP, yang dikombinasikan dengan sebuah mekanisme pada *player* sehingga pengguna dapat melakukan pemilihan acak pada setiap bagian timeline setiap saat. [RAM-11]

HTTP *Pseudostreaming* memerlukan dukungan dari sisi klien dan sisi server. Untuk sisi server, plugin tersedia untuk Apache, Lighttpd, dan lainnya. Pada sisi klien, beberapa *player* perlu mensinkronisasikan video, membaca metadata, dan sebagainya. Dua contoh player yang support HTTP *Pseudostreaming* adalah JWPlayer dan Flowplayer. [ROM-11]

Keuntungan menggunakan metode pseudostreaming adalah:

- Kemampuan untuk mencari pada *timeline* pada posisi yang belum di*download* oleh *player* (*random seeking*).
- Memutar kembali video pada saat sedang berada di tengah-tengah timeline.
- Menyediakan integrasi untuk *streaming* server dan *server-side script*.
- Mendukung file FLV dan H.264.
- Tidak terlalu repot dengan firewall dan proxy server (lingkungan perusahaan).
- Waktu loading awal biasanya lebih pendek.
- Menghasilkan pengalaman menonton yang lebih baik jika koneksi lokal dibawah rata-rata.

Untuk menyediakan fitur *pseudostreaming* pada klien untuk file video berupa FLV, maka harus menambahkan konfigurasi pada server, menggunakan *flash player* yang mendukung *pseudostreaming*, dan menyiapkan file FLV. Konsep dasarnya adalah server menerima sebuah

parameter bersamaan dengan file videonya, yang mengindikasikan point awal ketika klien memilih pada *timeline* dan mengembalikan data videonya.

Berdasarkan perbandingan-perbandingan diatas, penulis akan membangun layanan *Video On Demand* menggunakan protokol HTTP dengan metode Pseudostreaming. Penulis tidak menggunakan metode DASH karena masih terkendala oleh software yang masih terbatas. *Pseudostreaming* dianggap lebih cocok karena efisien, mudah, dan murah untuk diimplementasikan. Untuk penjelasan mendetail mengenai *pseudostreaming* akan dijelaskan pada bab 2.6.

2.7 Metadata

Metadata adalah informasi terstruktur yang menggambarkan, menjelaskan, melokasikan suatu data sehingga mempermudah pencarian, penggunaan, atau mengatur sumber informasi. Metadata sering disebut data tentang data atau informasi tentang informasi. Misalnya, sebuah gambar memiliki metadata yang menginformasikan seberapa besar ukuran file gambar, kedalaman warnanya, resolusinya, kapan dibuat, dan sebagainya. Metadata direkam komputer secara otomatis saat sebuah file dibuat, sehingga bisa diketahui kapan file dibuat, siapa pembuatnya, berapa ukuran filenya, demikian juga ekstensinya. Namun demikian, metadata juga dapat disusun secara manual. [NIS-01]

Terdapat tiga macam tipe metadata, yaitu : [NIS-01]

- Metadata deskriptif, menggambarkan sebuah sumber daya yang bertujuan menemukan dan mengidentifikasi beberapa elemen seperti judul, abstrak, pengarang, dan kata kunci.
- 2. Metadata struktural, menunjukkan bagaimana memadukan beberapa kumpulan objek secara bersama-sama, seperti contoh, bagaimana halaman-halaman diurutkan untuk membentuk beberapa bab.
- 3. Metadata administratif, menyediakan informasi untuk membantu mengatur sumber daya, seperti kapan dan bagaimana ia diciptakan, tipe file dan informasi teknis lainnya, dan siapa saja yang dapat mengaksesnya.

2.8 Flash Video (FLV)

Flash adalah standar de facto untuk media dinamis pada web, mendukung beberapa format media, termasuk dua format penting untuk menampilkan stream video dan audio [ADO-10]:

- F4V, untuk H.264/AAC
- FLV, untuk codec seperti Sorenson Spark dan On2 VP6

Data audio dan video dalam file FLV dikodekan dengan cara yang sama seperti dalam file SWF. Sementara itu, format file F4V berbasis format media file ISO dan support dengan Flash Player 9 update 3.

Flash Video atau FLV merupakan format file video yang dikembangkan oleh Macromedia Corporation. Lebih sering digunakan dalam jaringan karena mempunya kapasitas yang kecil, kecepatan *loading* yang ttinggi dan kualitas yang lebih baik. [JIN-11]

Untuk memainkan video FLV, tidak perlu menginstall *player* khusus, Microsoft's IE (Internet Explorer) dapat memainkannya dengan baik. Sekarang ini, terdapat beberapa video *sharing* yang dikenal luas yang mengadopsi format video FLV, seperti Sina, Youku, Cool 6, Youtube dan lain sebagainya. File FLV mejadi format umum yang digunakan di Internet.

FLV Header

Sebuah file FLV selalu dimulai dengan FLV header. Tabel 2.2 menunjukkan isi dari FLV header, yaitu :

Tabel 2.2 Tabel FLV Header

Field	Tipe	Keterangan Keterangan
Signature	UI8	Byte signature selalu F (0x46)
Signature	UI8	Byte signature selalu L (0x4c)
Signature	UI8	Byte signature selalu V (0x56)
Version	UI8	File version (contoh, 0x01 untuk FLV versi 1)
TypeFlagsReserved	UB[5]	Selalu 0
TypeFlagsAudio	UB[1]	1 = tag audio tersedia
TypeFlagsReserved	UB[1]	Selalu 0
TypeFlagsVideo	UB[1]	1 = tag video tersedia
DataOffset	UI32	Panjang dari header dalam byte

Sumber: [ADO-10]

BRAWIJAY

Field DataOffset biasanya mempunyai nilai 9 untuk FLV versi 1. Field ini tersedia untuk mengakomodasi header yang lebih besar di versi yang akan datang.

FLV Body

Setelah FLV header, file FLV selanjutnya yaitu FLV body harus terdiri dari back-pointer dan tag alternative, seperti yang ditunjukkan dalam tabel 2.3 berikut:

Tabel 2.3 Tabel FLV Body

Field	Tipe	Keterangan
PreviousTagSize0	UI32	Selalu bernilai 0
Tag1	FLVTAG	Tags pertama
PreviousTagSize1	UI8	Ukuran tag sebelumnya termasuk header, dalam byte. Untuk FLV versi 1, nilainya 11 ditambah DataSize dari tag sebelumnya.
Tag2	FLVTAG	Tags kedua
🔏		
PreviousTagSizeN-1	UI32	Ukuran dari dua tag terakhir, termasuk header, dalam byte
TagN	FLVTAG	Tags terakhir
PreviousTagSizeN	Ui32	Ukuran dari tag terakhir, termasuk header, dalam byte.

Sumber: [ADO-10]

FLV Tag

FLV tag berisi metadata untuk audio, video, atau script, metadata enkripsi, dan payload. Tabel 2.4 berikut adalah isi dari FLV tag.

Tabel 2.4 Tabel FLV Tag

Field	Tipe	Keterangan
Reserved	UB[2]	Selalu bernilai 0
Filter	UB[1]	Sebagai indikasi apakah paket terpilih 0 = tidak 1 = ya
Tagtype	UB[5]	Tipe konten tags

		8 = audio
YE JAUNE	MINATER	9 = video
		18 = script data
DataSize	UI24	Ukuran pesan
TimeStamp	UI24	Waktu dalam hitungan
		millisecond yang terkait
NS DROP		dengan tags pertama
TimeStampExtended	UI8	Bentuk rinci timestamp
StreamID	UI24	Selalu bernilai 0
AudioTagHeader	Jika Tagtype ==8	Informasi audio
C	AudioTagHeader	31
VideoTagHeader	Jika Tagtype ==9	Informasi video
	VideoTagHeader	
EncryptionHeader	Jika Filter ==1	Informasi proteksi header
2	EncryptionTagHeader	
FilterParams	Jika Filter == 1	Filter untuk proteksi
	FilterParams	header
Data	IF TagType == 8	Spesifik data untuk setiap
1	AUDIODATA	tipe media
	IF TagType == 9	
X	VIDEODATA	4 3
(3	IF TagType == 18	
4	SCRIPTDATA	

Sumber: [ADO-10]

Saat playback, urutan waktu dari tag FLV tergantung dari timestamp FLV saja. Setiap mekanisme waktu yang dibangun ke dalam format data payload akan diabaikan.

VideoTagHeader

VideoTagHeader berisi metadata video. Tabel 2.5 berikut adalah isi dari VideoTagHeader.

 Tabel 2.5
 Tabel VideoTagHeader

Field	Type	Comment
Frame Type	UB[4]	Tipe dari video frame.
WATTAN		1=keyframe (untuk AVC, frame yang
		bisa dicari)
		2=inter frame (Untuk AVC, frame

	TIVIELE	yang tidak bisa dicari)
YATAUI		3= disposable interframe (untuk
		H.263 saja)
KINIVARI		4= generated key frame (disediakan
		hanya untuk server saja)
BRADAV		5 = video info/command frame
Codec ID	UB[4]	Identitas Codec.
		2= Sorenson H.263
		3= Screen Video
		4= On2 VP6
	CITA	5=On2 VP6 dengan alpha channel
		6= Screen video versi 2
		7 = AVC
<i>AVCPacketType</i>	IF CodecID	0 = AVC sequence header
	== 7	1 = AVC NALU
	UI8	2= AVC end of Sequence
CompositionTime	IF CodecID	IF AVCPAcketType ==1
	 7\\	Composition Time offset
	SI24	ELSE
		0

Sumber : [ADO-10]

VideoTagBody

VideoTagBody berisi informasi frame video. Tabel 2.6 berikut adalah isi dari VideoTagBody.

Tabel 2.6 Tabel VideoTagBody

Field	Type	Comment
VideoTagBody	If FrameType == 5	Video Frame
	UI8	Payload
	ELSE {	atau Frame Info
	If CodecID == 2	
	H263VIDEOPACKET	If FrameType=5,
	IF CodecID == 3	video
	SCREENVIDEOPACKET	data body berisi
N/A-JA	IF CodecID == 4	UI8
	VP6FLVALPHAVIDEOPACKET	dengan nilai:
	IF CodecID == 6	0 = awal dari
	SCREENV2VIDEOPACKET	pencarian
Brack	IF CodecID == 7	klien-side video

	AVCVIDEOPACKET	frame
	3+KTV=H=R)LK	sequence.
	AUTINIYHTER	SCITALA
411116	AVAY: IINIX TUI	1 = akhir dari
		klien-side
BRAN		video frame
AS BINS		sequence.

Sumber : [ADO-10]

OnMetaData

Berikut adalah isi dari metadata FLV yang ditunjukkan pada tabel 2.7:

Tabel 2.7 Tabel OnMetaData

D (TE.	T7. /
Parameter	Type	Keterangan
Audiocodecid	Number	Nomor ID codec audio yang digunakan
		dalam FLV. (Tak terkompresi = 0,
	MIL	ADPCM = 1, MP3 =
	いかが	2, NellyMoser = 5 dan 6).
Audiodatarate /	Number	Besar rate video
Audiodelay 🥎	Number	Jeda audio per detik
Audiosamplerate	Number	Frekuensi audio rate
Audiosamplesize	Number	Resolusi dari single audio sample
canSeekToEnd	Boolean	True jika tag video terakhir adalah frame
	以	kunci dan maka dapat dilakukan
	474	pencarian.
Duration	String	Panjang FLV dalam hitungan detik
Filesize	Number	Besar ukuran file dalam byte
Framerate	Number	Banyaknya frame per detik
Height	Number	Ukuran panjang video dalam pixel
Stereo	Boolean	Mengindikasikan stereo audio
Videocodecid	number	Nomor ID codec video yang digunakan
		dalam FLV. (Sorenson H.263 = 2, Screen
		Video = 3, On2 VP6 = 4 and 5, Screen
		Video $V2 = 6$).
Videodatarate	Number	Bit rate video dalam kilobits per detik
Width	number	Lebar video dalam pixel

Sumber: [ADO-10]

BRAWIJAY

2.9 Model Proses Perangkat Lunak

Model proses perangkat lunak merupakan deskripsi yang disederhanakan dari proses perangkat lunak yang dipresentasikan dengan sudut pandang tertentu. Proses perangkat lunak sendiri adalah serangkaian kegiatan dan hasil-hasil relevannya yang menghasilkan perangkat lunak [SOM-11]. Model proses perangkat lunak merupakan abstraksi dari proses sebenarnya yang dideskripsikan. Walaupun ada banyak proses perangkat lunak, ada kegiatan-kegiatan mendasar yang umum bagi semua proses perangkat lunak. Kegiatan-kegiatan tersebut adalah:

1. Perencanaan

Sebuah proses dasar untuk memahami mengapa sebuah sistem harus dibangun. Pada fase ini diperlukan analisa kelayakan dengan mencari data atau melakukan proses pengumpulan informasi kepada pengguna.

2. Analisa

Sebuah proses investigasi terhadap sistem yang sedang berjalan dengan tujuan untuk mendapatkan jawaban mengenai pengguna sistem, cara kerja sistem dan waktu penggunaan sistem. Dari proses analisa ini akan didapatkan cara untuk membangun sistem baru.

3. Rancangan

Proses penentuan cara kerja sistem dalam hal desain arsitektur, desain antar muka, *database* dan spesifikasi file, dan desain program. Hasil dari proses perancangan ini akan didapatkan spesifikasi sistem.

4. Implementasi

Proses pembangunan dan pengujian sistem, instalasi sistem, dan rencana dukungan sistem.

Beberapa contoh model atau paradigma umum pada perangkat lunak, diantaranya:

1. Model air terjun (waterfall).

Tahap-tahap utama dari model ini memetakan kegiatan-kegiatan pengembangan dasar, yaitu : analisis dan definisi persyaratan, perancangan sistem dan perangkat lunak, implementasi dan

pengujian unit, integrasi dan pengujian sistem, operasi dan pemeliharaan.

2. Pengembangan evolusioner

Pengembangan ini berdasarkan pada ide untuk mengembangkan implementasi awal, memperlihatkannya pada pengguna untuk dikomentari, dan memeperbaikinya versi demi versi sampai sistem yang memenuhi persyaratan diperoleh.

3. Pengembangan sistem formal

Pengembangan ini memiliki kesamaan dengan model air terjun, tetapi proses pengembangannya didasarkan pada transformasi matematis dari spesifikasi sistem menjadi program yang dapat dijalankan.

4. Pengembangan berdasarkan pemakaian ulang

Pendekatan ini didasarkan atas adanya komponen yang dapat dipakai ulang dalam jumlah yang signifikan. Proses pengembangan sistem fokus pada integrasi komponen-komponen ini ke dalam suatu sistem, dan bukan mengembangkannya dari awal.

Pada skripsi ini akan dikembangkan suatu sistem yang model proses perangkat lunaknya memakai pengembangan berdasarkan pemakaian ulang. Unit perangkat lunak yang dipakai ulang adalah pemakaian ulang berbasis komponen atau *Component-Based Software Engineering* (CBSE).

2.10 Component-Based Software Engineering (CBSE)

Rekayasa perangkat lunak berbasis komponen adalah pendekatan berbasis pemakaian ulang untuk mendefinisikan, melaksanakan, dan menyusun beberapa komponen independen dan ditambahkan ke dalam sistem [SOM-11]. Jadi, dalam membangun sebuah aplikasi, pihak pengembang tidak perlu membangun semua kebutuhan aplikasi tersebut dari awal, melainkan hanya tinggal menggabungkan bagian-bagian yang tersedia menjadi satu aplikasi yang diinginkan.

Sistem Video on Demand (VOD) yang dibangun menerapkan CBSE karena dalam membangun sistem diperlukan komponen-komponen

penyusun sehingga sistem dapat menjalankan fungsi *pseudostreaming*. Tanpa salah satu komponen maka sistem tidak dapat menjalankan fungsinya. Komponen-komponen penyusun sistem antara lain adalah modul Apache, FFMPEG, Yamdi, dan Flowplayer.

2.11 FFMPEG

FFmpeg adalah *framework* multimedia terkemuka yang dapat melakukan *decode, encode, transcode, mux, demux, stream, filter* dan memainkan berbagai macam format multimedia. FFmpeg mempunyai beberapa *library* seperti libavcodec, libavutil, libavformat, libavdevice, libswscale and libswresample. Selain itu, juga terdapat beberapa *tool* seperti ffserver, ffplay, dan ffprobe. [BEL-09]

FFmpeg dikembangkan pada Linux, tetapi dapat dikompilasi di banyak sistem operasi, termasuk Apple Inc Mac OS X, Microsoft Windows dan AmigaOS. Sebagian besar platform komputasi dan set instruksi arsitektur microprocessor juga di dukung, seperti x86 (IA-32 dan x86-64), PPC (PowerPC), ARM, DEC Alpha, SPARC, dan arsitektur MIPS.

2.12 Yamdi

Yamdi kepanjangan dari *Yet Another MetaData Injector* dan merupakan *injector* metadata untuk file FLV. Yamdi menambahkan *event* onMetaData ke dalam file FLV. Yamdi dapat berjalan di BSD, Linux, dan Windows dan dipublikasikan dibawah lisensi BSD. [OPP-11]

Beberapa fitur dari Yamdi adalah:

- Support file yang besar
- Support H.264
- Event onMetaData, onLastSecond, onLastKeyframe
- Hanya membutuhkan memori yang kecil
- Menghasilkan keluaran metadata dalam XML.

BAB III METODE PENELITIAN

Metodologi penelitian yang digunakan dalam pembuatan tugas akhir ini adalah sebagai berikut :

3.1 Studi Literatur

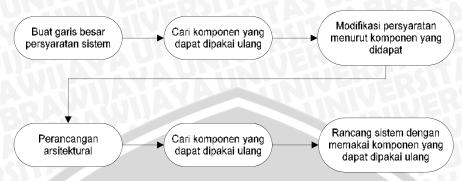
Sebelum sistem didesain dan dirancang akan dilakukan studi pustaka atau literatur terlebih dahulu untuk memahami konsep-konsep sistem yang harus dipelajari agar dalam perancangan sistem tidak mengalami kendala yang berarti. Dalam pengerjaan skripsi ini, literatur yang harus dicari adalah yang berhubungan dengan :

- 1. Video On Demand
- 2. Video Streaming
- 3. Protokol RTSP dan HTTP Pseudostreaming
- 4. PHP dan MySQL
- 5. Component Based Software Engineering (CBSE)

3.2 Model Proses Perangkat Lunak

Model proses perangkat lunak yang dikembangkan dalam sistem VOD memakai pengembangan berbasis komponen atau *Component Based Software Engineering (CBSE)* karena dalam penyusunannya memerlukan beberapa komponen agar dapat menjalankan beberapa fungsi. Keuntungan dari pengembangan berbasis komponen adalah diperkecilnya biaya pengembangan secara keseluruhan. Selain itu keuntungannya adalah keandalan bertambah, resiko proses diperkecil, dan proses pengembangan yang cepat.

Pengembangan berorientasi komponen dapat diintegrasikan ke dalam proses pengembangan sistem dengan menggunakan kegiatan pemakaian ulang melalui proses yang ditunjukkan pada gambar 3.1. Sedangkan untuk mengidentifikasi komponen yang akan digunakan ke dalam sistem, proses yang dilalui ditunjukkan pada gambar 3.2.



Gambar 3.1 CBSE dengan Pemakaian Ulang

Sumber: [SOM-11]



Gambar 3.2 Proses Identifikasi Komponen Sumber: [SOM-11]

Beberapa komponen yang dipakai dalam sistem ini diantaranya adalah Flowplayer sebagai video *player* sekaligus penyedia *plugin* untuk *pseudostreaming*. Selain itu FFMPEG yang digunakan sebagai konverter video, dan Yamdi yang digunakan sebagai injektor metadata pada file FLV. Alasan pemakaian ketiga komponen tersebut lebih lanjut dibahas pada bab IV.

3.3 Analisis Kebutuhan Fungsional

Kebutuhan data fungsional yang digunakan untuk perancangan sistem VOD ini meliputi analisis kebutuhan perangkat, kebutuhan data dan pemodelan sistem yang terdiri dari pembuatan Diagram Konteks dan *Data Flow Diagram* (DFD). Diagram Konteks yang menggambarkan hubungan sistem dengan lingkungan luar, sedangkan DFD untuk menggambar sistem sebagai suatu jaringan fungsional yang dihubungkan satu sama lain dengan alur data baik secara manual maupun komputerisasi. Analisis kebutuhan fungsional pada sistem ini mengacu pada standar IEEE 830.

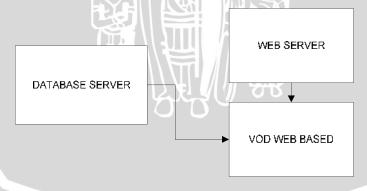
BRAWIJAYA

3.4. Analisis Kebutuhan Non Fungsional

Kebutuhan non fungsional, seperti terlihat dari namanya, adalah persyaratan yang tidak langsung berhubungan dengan fungsi spesifik yang disediakan oleh sistem. Persyaratan-persyaratan ini mungkin berhubungan dengan properti sistem yang muncul belakangan, seperti keandalan, waktu tanggap, dan penempatan pada media penyimpanan. Alternatifnya, persyaratan ini dapat mendefinisikan batasan pada sistem seperti kemampuan piranti I/O dan representasi data yang dipakai pada *interface* sistem. Parameter yang diukur dalam kebutuhan non fungsional ini yaitu *reliability*.

3.5 Perancangan Sistem

Perancangan yang dilakukan antara lain pembuatan database, perancangan sistem *video on demand* yang terdiri dari penerapan *streaming* yang menggunakan protokol HTTP dan metode *pseudostreaming* serta membuat aplikasi *video on demand* dengan konten dan fitur-fiturnya. Untuk menerapkan hal tersebut, maka dibuatlah blok diagram perancangan yang ditunjukkan pada gambar 3.3. Sedangkan untuk pengembangan aplikasi *video on demand* minimal harus mempunyai persyaratan yang ditunjukkan pada gambar 3.4.



Gambar 3.3 Sistem VOD yang akan dibangun

Sumber : [Metode Penelitian]



Gambar 3.4 Persyaratan Sistem VOD

Sumber : [Metode Penelitian]

3.6 Implementasi

Setelah melakukan perancangan, maka sistem VOD akan dibuat dengan mengacu pada perancangan sistem VOD tersebut. Implementasi diawali spesifikasi lingkungan dengan penjabaran pengembangan sistem implementasi basis dilanjutkan dengan data dengan menggunakan MySQL. Implementasi sistem dilakukan dengan menggunakan bahasa pemrograman PHP.

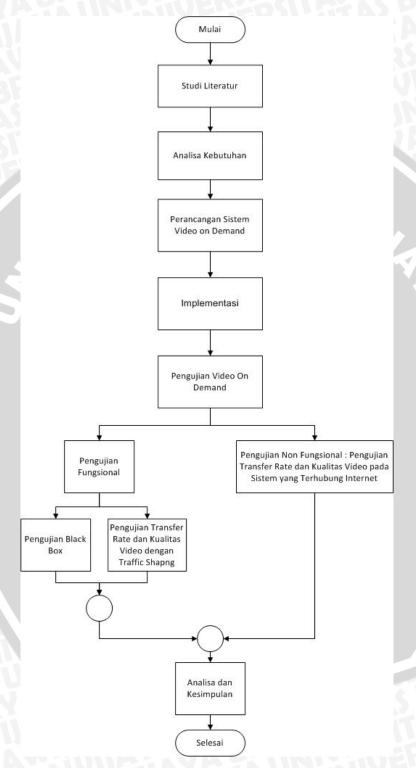
3.7 Pengujian Sistem

Pengujian sistem bertujuan untuk menemukan kesalahan atau kekurangan pada perangkat lunak yang diuji. Pengujian dimaksudkan untuk mengetahui perangkat lunak yang dibuat telah memenuhi kriteria yang sesuai dengan tujuan perancangan. Pada skripsi ini digunakan dua metode pengujian, yaitu pengujian terhadap setiap fungsi yang terdapat dalam sistem dan pengujian kualitas video terhadap *transfer rate user*. Pengujian ditanamkan

dengan mengunggah sistem VOD ke web hosting agar dalam pengujian kualitas video terhadap *transfer rate user* dapat dikoneksikan dengan jaringan yang ada di Internet.

3.8 Analisis Hasil dan Pengambilan Kesimpulan

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi dan pengujian sistem telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem yang telah dibuat. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan - kesalahan yang terjadi dan menyempurnakan penulisan serta untuk memberikan pertimbangan terhadap pengembangan sistem selanjutnya.



Gambar 3.5 Flowchart Perancangan VOD

BAB IV

ANALISIS KEBUTUHAN DAN PERANCANGAN SISTEM

Pada bab ini, akan dijelaskan garis besar persyaratan sistem yang penyusunannya mengacu pada IEEE 830. Setelah menganalisis kebutuhan, selanjutnya adalah perancangan sistem yang dimulai dari identifikasi komponen sampai dengan merancang sistem dengan komponen yang telah ditentukan.

4.1 Persyaratan Sistem

4.1.1 Kebutuhan Antarmuka Eksternal

Sistem *Video on Demand* dibangun di atas aplikasi internet dengan berbasiskan web. Dalam deskripsi kebutuhan antarmuka ekstenal akan dijelaskan kebutuhan antarmuka sistem dengan perangkat lain yang berada diluar cakupan sistem yang akan dikembangkan tetapi mempunyai keterkaitan dalam proses yang dilakukannya.

Dalam operasionalnya, sistem yang akan dikembangkan memerlukan adanya interaksi dengan komponen-komponen lain diluar sistem itu seperti: pengguna sistem, perangkat keras, perangkat lunak, dan perangkat komunikasi dimana sistem ini akan menggunakan jaringan internet.

4.1.1.1 Antarmuka Pemakai

Sistem yang akan dikembangkan membutuhkan interaksi dengan pemakai sistem. Dalam melakukan interaksi dengan pemakai sistem VOD, membutuhkan perangkat untuk melakukan proses *input* dan *output* dari dan ke pemakai. Perangkat tersebut adalah sebagai berikut:

1. Perangkat Keyboard

Keyboard diperlukan sebagai sarana bagi pemakai untuk mengetikkan data masukan yang akan diproses sistem.

2. Perangkat Mouse

Perangkat *mouse* digunakan sebagai sarana bagi pemakai untuk memasukkan data input bagi sistem.

BRAWIJAYA

3. Perangkat Monitor

Monitor sebagai sarana untuk menampilkan aplikasi kepada pemakai mempunyai spesifikasi diantaranya: monitor mampu menampilkan grafis dengan kualitas warna yang baik untuk menampilkan video.

4.1.1.2 Antarmuka Perangkat Keras

Antarmuka perangkat keras yang dibutuhkan dalam perangkat lunak ini meliputi:

1. Kartu Grafik (VGA Card)

Kartu grafik yang dibutuhkan dalam mengoperasikan sistem VOD diharapkan minimal sebesar 128 MB. Apabila menggunakan kartu grafik yang punya kemampuan dibawahnya, sistem masih bisa berjalan akan tetapi tampilan yang diberikan kurang baik.

2. Memori

Peryaratan minimum memori yang dibutuhkan dalam mengoperasikan sistem ini diharapkan sebesar 512 MB.

3. Prosesor

Prosesor yang digunakan untuk menjalankan sistem ini minimal adalah Intel Pentium 4 atau sekelasnya.

4. Ethernet Card dan Modem

Ethernet card dan modem dibutuhkan untuk menghubungkan komputer yang menjalankan perangkat lunak dengan jaringan internet. Spesifikasi ethernet card dan modem yang dibutuhkan adalah yang compatible dengan sistem operasi yang digunakan.

BRAWIJAY

4.1.1.3 Antarmuka Perangkat Lunak

Untuk menjalankan sistem *Video on Demand* di sisi pengguna, maka perangkat lunak yang dibutuhkan adalah :

1. Web Browser

Web browser yang digunakan dapat berupa Mozilla Firefox, Google Chrome, atau Opera. Tidak disarankan menggunakan Internet Explorer.

2. Flash Plugin Video

Flash Plugin sangat dibutuhkan untuk memainkan file video FLV pada sistem Video on Demand.

4.1.1.4 Antarmuka Komunikasi

Antarmuka komunikasi yang dibutuhkan dalam sistem VOD merupakan antarmuka untuk melakukan koneksi dalam jaringan internet meliputi:

1. Antarmuka komunikasi pada sisi Server

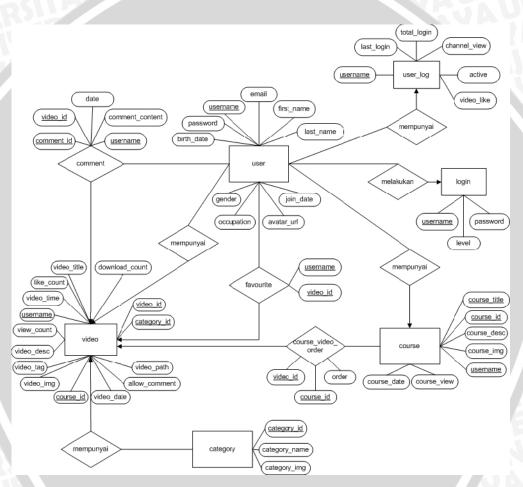
Aplikasi pada sisi *server* merupakan aplikasi yang melayani semua *request* yang dikirimkan oleh klien yang meminta layanan dengan protocol HTTP. Oleh karena itu pada sisi s*erver* dibutuhkan adanya sebuah *web server* yang terhubung dalam jaringan internet. *Web server* tersebut harus mempunyai alamat IP dan *domain* yang dapat dikenali dalam jaringan internet.

2. Antarmuka komunikasi pada sisi Klien

Pada sisi klien proses yang dilakukan adalah melakukan *request* kepada *server* untuk meminta layanan HTTP. Oleh karena itu antarmuka yang dibutuhkan pada sisi klien adalah perangkat komunikasi yang memungkinkan komputer klien terhubung dengan jaringan internet.

4.1.2 Analisis Basis Data

Data-data yang diperoleh pada tahap analisis sistem akan digunakan untuk mendesain basis data disertai dengan atribut-atribut entitasnya. Dalam merancang basis data, model data yang akan digunakan adalah *Entity Relationship Diagram* (ERD). Berikut ini adalah ERD hasil analisis yang telah dilakukan:



Gambar 4.1 Entity Relationship Diagram

Sumber : [Analisis Perancangan]

BRAWIJAY/

4.1.3 Analisis Kebutuhan Fungsional

Kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik serta sesuai dengan kebutuhan. Analisis kebutuhan fungsional ini meliputi analisis kebutuhan data dan pemodelan sistem. Pemodelan sistem dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Tahap-tahap pemodelan dalam analisis tersebut antara lain meliputi pembuatan Diagram Konteks dan *Data Flow Diagram* (DFD).

4.1.3.1 Analisis Kebutuhan Data

Untuk menjalankan sistem ini dibutuhkan masukan data yang dijadikan sebagai pendukung pembuatan sistem VOD ini. Adapun data yang diperlukan oleh sistem ini adalah :

- 1. Video. Video dapat berupa video pembelajaran siswa / mahasiswa, video dokumenter, film, dan lain-lain.
- 2. Data tekstual berupa nama video, kategori video dan deskripsi mengenai informasi video.

4.1.4 Analisis Kebutuhan Data Non Fungsional

Analisis kebutuhan non fungsional adalah persyaratan yang tidak langsung berhubungan dengan fungsi spesifik yang disediakan oleh sistem. Berikut identifikasi kebutuhan non fungsional pada aplikasi VOD dengan parameter *reliability*:

- Sistem dapat me-*load* halaman yang berisi database yang banyak, dan sistem jalannya halaman satu ke halaman lain juga cepat. serta sistem dapat menolak input data yang salah.
- Sistem dapat mendeteksi transfer rate user agar sistem dapat memilih kualitas video yang sesuai dengan transfer rate tersebut terutama pada saat sistem telah terhubung ke internet.

4.2 Pencarian Komponen

Dalam membuat sistem VOD menggunakan *pseudostreaming*, dibutuhkan beberapa komponen. Dalam membuat sistem *pseudostreaming* berbasis FLV streaming, ada 3 komponen penting, yaitu video *player*, encoder video, dan injektor metadata. Dalam menentukan komponen apa saja yang tepat, maka perlu dilakukan identifikasi komponen. Berikut akan dibahas identifikasi masing-masing komponen.

4.2.1 Video Player

Untuk membangun suatu aplikasi video *streaming*, maka komponen yang sangat dibutuhkan adalah video *player*. Terdapat bermacam-macam video player yang ditawarkan di Internet. Untuk kebutuhan *pseudostreaming*, maka dibutuhkan video player yang *open source* agar dapat dilakukan modifikasi. Ada dua video player yang dapat dijadikan pertimbangan jika ingin membuat fitur *pseudostreaming*, yaitu Flowplayer dan JWPlayer.

4.2.2 Encoder Video

Encoder video dalam aplikasi VOD ini digunakan untuk mengubah format video menjadi FLV karena yang digunakan dalam aplikasi ini adalah streaming berbasis FLV. Komponen ini diharapkan tidak hanya sebagai konverter saja melainkan juga dapat menambah fitur lain seperti membuat thumbnail, dan sebagainya. Encoder yang dapat digunakan antara lain adalah FFMPEG dan Mencoder.

FFmpeg adalah proyek *open source* yang membuat berbagai macam library dan program yang menangani data multimedia. FFmpeg juga mempunyai banyak komponen termasuk ffserver, libavfilter, ffplay, dan lainlain. Mencoder adalah *command line tool* yang men*decode*, *encode*, dan filter file. Mencoder dibangun untuk mensupport MPlayer dan juga termasuk dalam distribusi MPlayer.

BRAWIJAYA

4.2.3 Injektor Metadata

Video yang telah diubah format menjadi FLV, biasanya berisi metadata yang masih minim. Untuk melakukan proses *streaming* dengan *pseudostreaming* pada file flv maka harus memiliki event metadata seperti *array filepositions* dan *times* agar dapat melakukan proses *random seeking*. Maka dari itu file tersebut harus melakukan injeksi metadata terlebih dahulu. Injeksi metadata dapat dilakukan dengan menggunakan tool *injector* seperti Flytool2 dan Yamdi.

Flvtool2 adalah alat manipulasi untuk file FLV. Flvtool2 dapat menambahkan tag onMetaData, memotong file FLV, menambahkan cue point, menampilkan struktur FLV dan mencetak informasi metadata dalam XML atau YAML. Sedangkan Yamdi adalah metadata *injector* untuk FLV yang mempunyai fitur seperti: mendukung file berukuran besar, support H.264, dapat menambahkan tag onMetaData, onLastSecond, dan onLastKeyframe, penggunaan memori yang kecil, dan mencetak informasi metadata dalam XML.

4.3 Modifikasi Komponen

Pada tahap ini akan dilakukan modifikasi komponen berdasarkan persyaratan yang ada. Untuk video *player*, diperlukan beberapa persyaratan seperti: player mendukung fitur *pseudostreaming* dan *player* mempunyai fitur untuk pemilihan kualitas video (dalam hal ini *player* harus dapat dimodifikasi agar dapat memilih kualitas video berdasarkan *transfer rate user*).

Untuk *encoder*, persyaratan yang diperlukan untuk membuat sistem VOD adalah: *encoder* harus dapat mengkonversi video ke dalam format FLV dengan kualitas yang cukup baik setelah dikonversi dan encoder dapat membuat *thumbnail* yang digunakan untuk membuat *thumbnail preview* video.

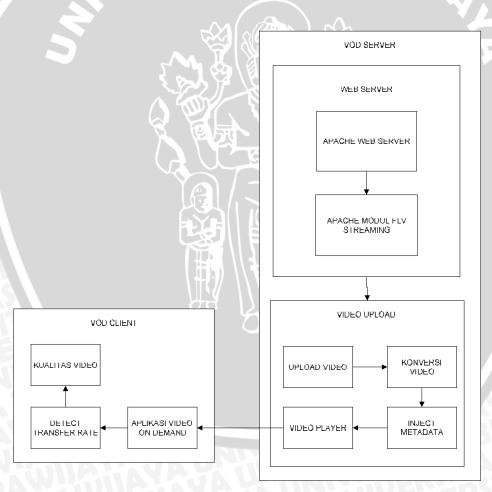
Untuk injektor metadata, yang diperlukan hanyalah untuk menginjeksi file video yang sudah berupa FLV dengan metadata yang utamanya berisi event *array filepositions* dan *times*. Tidak dilakukan adanya modifikasi pada persyaratan komponen ini.

4.4 Perancangan Arsitektur

Perancangan arsitektur bertujuan untuk mempermudah proses pembuatan perangkat lunak. Perancangan arsitektur ini terdiri dari perancangan arsitektur sistem, perancangan diagram konteks, DFD dan spesifikasi proses, perancangan struktur menu dan perancangan antar muka.

4.4.1 Perancangan Arsitektur Sistem

Setelah menganalisis kebutuhan sistem dan mengidentifkasi komponen, maka dibuatlah perancangan arsitektur sistem yang dijelaskan dengan blok diagram. Berikut adalah blok diagram arsitektur sistem yang ditunjukkan pada gambar 4.11.



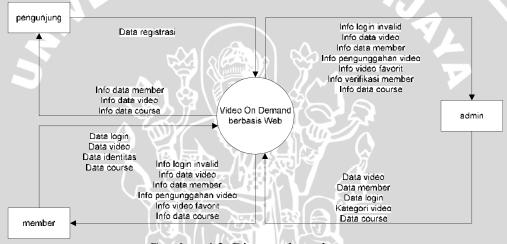
Gambar 4.2 Arsitektur Sistem

Sumber : [Analisis Perancangan]

BRAWIJAYA

4.4.2 Diagram Konteks

Diagram Konteks adalah suatu diagram alir yang tingkat tinggi yang menggambarkan seluruh jaringan, masukan dan keluaran. Sistem yang dimaksud adalah untuk menggambarkan sistem yang sedang berjalan, mengidentifikasikan data awal dan akhir yang masuk, dan keluaran sistem. Diagram ini merupakan gambaran umum sistem yang nantinya akan kita buat. Secara uraian dapat dikatakan bahwa diagram konteks itu berisi siapa saja yang memberikan data (inputan) ke sistem serta kepada siapa data informasi yang harus dihasilkan sistem.

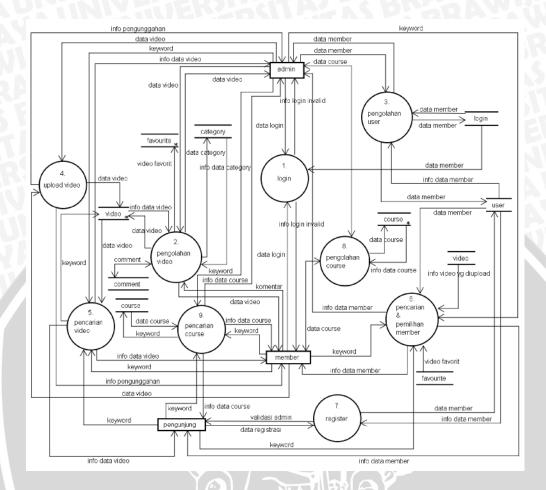


Gambar 4.3 Diagram konteks

Sumber : [Analisis Perancangan]

4.4.3 Data Flow Diagram

Data Flow Diagram (DFD) adalah representasi grafik dari sebuah sistem. DFD menggambarkan komponen-komponen sebuah sistem, aliran-aliran data di mana komponen-komponen tersebut, dan asal, tujuan, dan penyimpanan dari data tersebut.



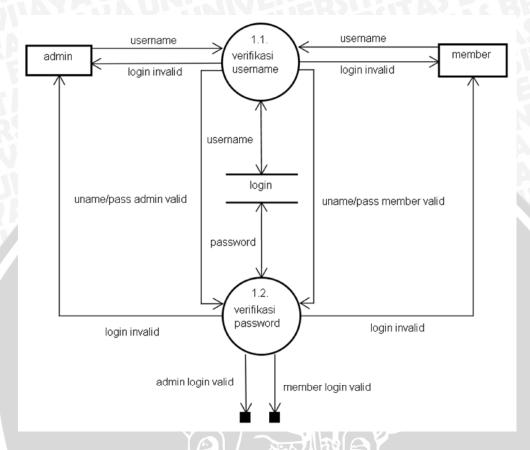
Gambar 4.4 DFD level 0 Aplikasi Video On Demand Berbasis Web

Sumber: [Analisis Perancangan]

DFD Level 1 Aplikasi Video On Demand Berbasis Web

Untuk menjelaskan lebih terperinci tentang proses-proses yang ada di dalam DFD level 0 maka dibawah ini akan digambarkan proses-prosesnya.

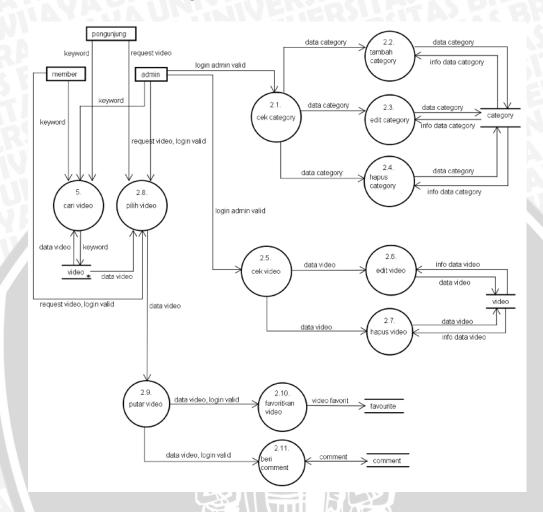
DFD level 1 Proses 1 Login



Gambar 4.5 DFD Level 1 Proses 1 Login

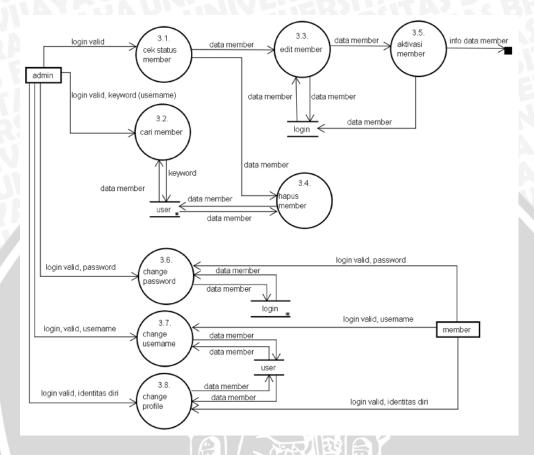
Sumber : [Analisis Perancangan]

DFD level 1 Proses 2 Pengolahan Video



Gambar 4.6 DFD Level 1 Proses 2 Pengolahan Video Sumber : [Analisis Perancangan]

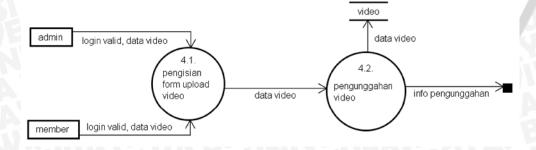
DFD level 1 Proses 3 Pengolahan User



Gambar 4.7 DFD Level 1 Proses 3 Pengolahan User

Sumber: [Analisis Perancangan]

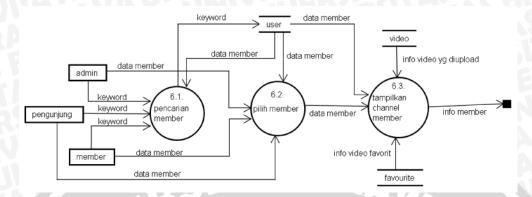
DFD Level 1 Proses 4 Upload Video



Gambar 4.8 DFD Level 1 Proses 4 Upload Video

Sumber: [Analisis Perancangan]

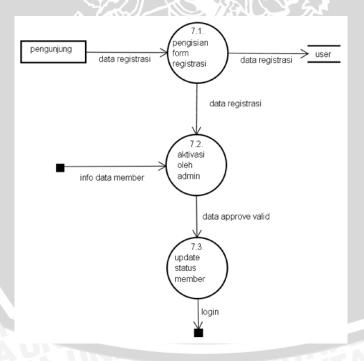
BRAWIJAYA



Gambar 4.9 DFD Level 1 Proses 6 Pencarian & Pemilihan Member

Sumber: [Analisis Perancangan]

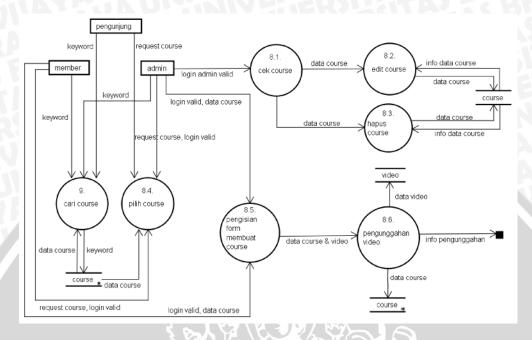
DFD Level 1 Proses 7 Register



Gambar 4.10 DFD Level 1 Proses 7 Register

Sumber: [Analisis Perancangan]

DFD Level 1 Proses 8 Pengolahan Course



Gambar 4.11 DFD Level 1 Proses 8 Pengolahan Course

Sumber : [Analisis Perancangan]

4.4.4 Spesifikasi Proses

Dalam spesfikasi proses digambarkan proses model aliran yang terdapat pada *Data Flow Diagram* (DFD). Spesifikasi proses dari DFD yang telah digambarkan diatas adalah sebagai berikut:

Tabel 4.1 Spesifikasi Proses

NO	Proses	Keterangan
1	Nomor Proses	1.
	Nama Proses	Proses Login
	Deskripsi	Proses untuk mendapatkan akses tambahan ke
	dalam sistem	
	Source	Pengunjung
	Input	Username admin / member
	Input	Password admin / member
	Output	Info login (valid / tidak valid)
	Destination	• Admin
l'a	Destination	Member

YA AV AV	Logika Proses	 Pengunjung memasukkan username dan password pada form login Sistem melakukan pengecekan pada database user Jika username dan password sesuai dengan database user, maka sistem akan membuat session ID dan memberikan akses lebih pada user, tetapi jika tidak sesuai maka sistem akan kembali menampilkan halaman / form login
2	Nomor Proses	2.
HTA	Nama Proses	Proses Pengolahan Video
	Deskripsi	 Menampilkan menu untuk pengolahan category oleh admin Menampilkan video hasil pencarian berdasarkan keyword Menampilkan informasi tentang komentar dan favorite pada video
5	Source	Member Admin
	Input	Data media (komentar, kategori, favorite) dan status login
	Output	Info data video
	Destination	Member Admin
	Logika Proses	 Admin/member login Admin/member memberi komentar pada video yang dipilih Admin/member memfavoritkan video yang disukai Admin/member dapat menambah dan menghapus video
3	Nomor Proses	3.
	Nama Proses	Proses Pengolahan User
	Deskripsi	Menampilkan menu untuk pengolahan user
	Source	Admin
	Input	Status login dan data member
	Output	Info data member
	Destination	Admin
	Logika Proses	 Admin login Admin memilih menu pengolahan data member Admin memasukkan keyword untuk melakukan pencarian member Admin melakukan proses edit / hapus member

4	Nomor Proses	4
VB	Nama Proses	Proses Upload Video
	Deskripsi	Proses untuk mengunggah video ke dalam sistem
AVL	Безкиры	Admin
	Source	Member
		Status login, video dan data video (judul,
	Input	deskripsi, kategori, keyword)
	Output	Info pengunggahan video
	Output	Admin
	Destination	Member
17-		
		Admin/member login Admin/member memilih many yaplas d
	25	Admin/member memilih menu upload Admin/member memilih data vidas (indul
	Logika Proses	Admin/member mengisi data video (judul, deskripsi, kategori, keyword) dan
	Logika 1 105C5	mengupload video
		Admin/member menunggu hingga proses
		upload selesai
5	Nomor Proses	5.
3	Nama Proses	Proses Pencarian Video
	Ivaliia i ioses	Proses untuk mendapatkan video yang diinginkan
	Deskripsi	berdasarkan keyword
		Admin
	Source	Member
	Source	Pengunjung
		User menginputkan keyword pada field yang
	Input	disediakan
	Output	Video dan data video
	Output	• Admin
	Destination	Member
		Pengunjung
		User memasukkan keyword pada field yang
	Logika Proses	disediakan untuk melakukan pencarian video
6	Nomor Proses	6.
	Nama Proses	Proses Pencarian & Pemilihan Member
	Ivaliia i ioses	Untuk mengakses halaman channel (profile)
	Deskripsi	member, melihat video member, video favorit,
	Безкітрзі	dan melihat komentar dari member lain
		Admin
	Source	Member
	Source	Pengunjung
	Input	Status login, data member, dan keyword
	Input	Info data member, info video member, dan info
	Output	komentar dari member lain
	SAW!	Admin
	Destination	
	ABREA	Member

		- II Cl 1
VA		 User mengakses menu Channel User memilih salah satu nama member
	Logika Proses	Sistem akan menampilkan halaman channel
		dari member yang dipilih
7	Nomor Proses	7.
	Nama Proses	Proses Register
	Deskripsi	Proses untuk mendaftar menjadi member
	Source	Pengunjung
471	Input	Data registrasi
24	Output	Validasi dari admin
	Destination	Member
	Logika Proses	 Pengunjung memilih menu Signup untuk mendaftar Pengunjung mengisikan data diri (username, password, email, gender, birth, dan occupation) pada form register Pengunjung menunggu akunnya diapprove
3		oleh admin Setelah diapprove oleh admin maka pengunjung telah menjadi member dan dapat melakukan login
8.	Nomor Proses	8.
	Nama Proses	Proses Pengolahan Course
	Deskripsi	 Menampilkan menu untuk mengolah (edit dan hapus) course Menampilkan menu untuk membuat course
	Source	Admin Member
	Input	Data course dan status login
	Output	Info data course, info pengunggahan
	Destination	Admin Member
	Logika Proses	 Admin/member login Admin/member memilih menu pengolahan data member Admin/member melakukan proses edit / tambah / hapus course
9.	Nomor Proses	9.
	Nama Proses	Proses Pencarian Course
	Deskripsi	Proses untuk mendapatkan course yang diinginkan berdasarkan keyword
	Source	Admin Member

	HINIYAU	Pengunjung
	Input	Keyword
		Data video dan data course
45	Output	
	Destination	• Admin
	Destination	• Member
		Pengunjung Licer memogukkan kayayard nada fiald yang
	Logika Proses	User memasukkan keyword pada field yang disediakan untuk melakukan pencarian course
10.	Nomor Proses	1.1, 1.2
	Nama Proses	Verifikasi username dan verifikasi password
	Deskripsi	Proses untuk verifikasi login
	Source	Pengunjung
	Input	Username dan password
-	Output	Info login (valid/tidak valid)
4	Destination	Admin Member
7	Logika Proses	 Pengunjung memasukkan username dan password pada form login Sistem melakukan pengecekan pada database login Jika username dan password sesuai dengan database login, maka sistem akan membuat session ID dan memberikan akses lebih pada user, tetapi jika tidak sesuai maka sistem akan kembali menampilkan halaman / form login
11.	Nomor Proses	2.1
	Nama Proses	Cek category
	Deskripsi	Menu untuk mengecek kategori video
	Source	Admin
	Input	Data category, login valid
	Output	Info data category
	Destination	Admin
	Logika Proses	 Admin login Admin memilih menu manage category untuk mengecek category yang ada
12.	Nomor Proses	2.2
	Nama Proses	Tambah category
	Deskripsi	Menu untuk menambah kategori video
	Source	Admin
	Input	Data category, login valid
	Output	Info data category
	Destination	Admin
	Logika Proses	Admin login

13.	Nomor Proses Nama Proses Deskripsi Source	 Admin memilih menu manage category untuk mengecek kategori yang ada Admin memilih link 'tambah' untuk menambah kategori yang diinginkan Admin mengisikan nama kategori baru lalu disimpan 2.3 Edit category Proses untuk mengedit category yang ada Admin
19	Input	Data category, login valid
	Output	Info data category
	Destination	Admin
UA	Logika Proses	 Admin login Admin memilih menu manage category untuk mengecek kategori Admin memilih link 'edit' untuk mengedit kategori yang diinginkan Admin melakukan pengubahan pada data yang dipilih
14.	Nomor Proses	2.4
Ī	Nama Proses	Hapus category
	Deskripsi	Proses untuk menghapus category yang ada
Ī	Source	Admin
Ī	Input	Data category, login valid
	Output	Info data category
	Destination	Admin
	Logika Proses	 Admin login Admin memilih menu manage category untuk mengecek kategori Admin memilih link 'hapus' untuk menhapus kategori yang diinginkan Admin melakukan konfirmasi penghapusan
15.	Nomor Proses	2.5
	Nama Proses	Cek video
	Deskripsi	Melakukan pengecekan video yang ada
	Source	Admin
	Input	Data video, login valid
NY	Output	Info data video
	Destination	Admin
	Logika Proses	 Admin login Admin memilih menu manage video untuk mengecek video yang ada dalam sistem
16.	Nomor Proses	2.6
8 6	Nama Proses	Edit video

	Deskripsi	Melakukan pengeditan video yang ada di sistem
	Source	Admin
	Input	Data video, login valid
	Output	Info data video
	Destination	Admin
	Logika Proses	 Admin login Admin memilih menu manage video untuk mengecek video Admin memilih link 'edit' untuk mengedit video yang diinginkan Admin melakukan pengubahan pada data yang dipilih
17.	Nomor Proses	2.7
	Nama Proses	Hapus video
	Deskripsi	Proses untuk menghapus video yang diinginkan
	Source	Admin
	Input	Data video, login valid
	Output	Info data video
	Destination	Admin = -
	Logika Proses	 Admin login Admin memilih menu manage video untuk mengecek video Admin memilih link 'hapus' untuk menghapus video yang diinginkan Admin melakukan konfirmasi penghapusan
18.	Nomor Proses	2.8
	Nama Proses	Pilih video
	Deskripsi	Memilih video yang akan diputar
	Source	AdminMemberPengunjung
	Input	Request video
	Output	Data video
	Destination	AdminMemberPengunjung
	Logika Proses	 Admin / member / pengunjung memasukkan keyword video yang diinginkan pada kolom pencarian Search Sistem akan menampilkan video yang sesuai dengan keyword tersebut Admin / member / pengunjung memilih video yang diinginkan
19	Nomor Proses	2.9
519	Nama Proses	Putar video

	Deskripsi	Memutar video yang akan dipilih
W B	Deskiihsi	Admin
$\{\lambda_i\}_{i=1}^n$	Source	• Member
	MELAY	Pengunjung
	Input	Data video
1313	Output	Info data video
	Bhas	• Admin
	Destination	• Member
5S_		Pengunjung
0		Admin / member / pengunjung memilih video yang diinginkan
	6	Sistem akan memutar video tersebut sesaat
	Logika Proses	setelah proses request dilakukan
	ME	Admin / member / pengunjung melakukan interaksi seperti start, pause, dan random seeking pada video yang diputar
20.	Nomor Proses	2.10
	Nama Proses	Favoritkan video
	Deskripsi	Fitur yang dapat memasukkan video yang dipilih menjadi video yang difavoritkan
	Source	Admin Member
	Input	Data video, login valid
	Output	Info data favourite
	Destination	• Admin
	Destination	Member
	E	Admin / member memilih video yang diinginkan Sistem akan menampilkan informasi video
	Logika Proses	 Sistem akan menampilkan informasi video Admin / member memasukkan video tersebut sebagai video yang difavoritkan dengan memilih tombol favorite
21.	Nomor Proses	2.11
	Nama Proses	Beri comment
	Deskripsi	Memberikan comment pada video yang dipilih
	Source	Admin Member
	Input	Data media, login valid
	Output	Info comment
	Destination	Admin Member
RA	Logika Proses	 Admin / member login Admin / member memilih video yang diinginkan

	UNIV	Admin / member memberikan comment pada kolom yang disediakan
22.	Nomor Proses	3.1
ALF	Nama Proses	Cek status member
	Deskripsi	Melakukan pengecekan member yang dudah terdaftar dalam sistem
	Source	Admin
	Input	Login valid
	Output	Info data member
	Destination	Admin
	Logika Proses	Admin login Admin memilih menu manage user untuk melakukan pengecekan member
23.	Nomor Proses	3.2
	Nama Proses	Cari member
4	Deskripsi	Melakukan pencarian member sesuai keyword yang dimasukkan
	Source	Admin
	Input	Login valid, keyword
-	Output	Info data member
	Destination	Admin
	Logika Proses	 Admin login Admin memasukkan keyword pada kolom pencarian untuk mencari member
24.	Nomor Proses	3.3
	Nama Proses	Edit member
	Deskripsi	Melakukan pengeditan pada member yang dipilih
	Source	Admin
	Input	Data member
	Output	Data member
	Destination	Admin
	Logika Proses	 Admin login Admin memilih menu manage user untuk melihat member yang mtelah mendaftar Admin memilih member lalu memilih link 'edit' untuk melakukan pengeditan data
25.	Nomor Proses	3.4
	Nama Proses	Hapus member
	Deskripsi	Melakukan penghapusan member yang dipiih
	Source	Admin
	Input	Data member
	Output	Data member
SA	Destination	Admin
	Logika Proses	Admin loginAdmin memilih menu manage user untuk

		SOCIETAS PIBOAMI
		melakukan pengecekan member
		Admin memilih user yang dituju dan memilih link 'hapus' untuk melakukan penghapusan member tersebut
26		
26	Nomor Proses	3.5
	Nama Proses	Aktivasi member
	BRA	Melakukan pengaktifan member pada member
	Deskripsi	ysang baru mendaftar agar statusnya menjadi aktif
124	Source	Admin
	Input	Data member
	Output	Info data member
	Destination	Admin
		Admin login
		Admin memilih menu manage user untuk melakukan pengecekan member
	Logika Proses	Admin memilih user yang dituju dan memilih
	Logika i roses	link 'edit' untuk melakukan pengeditan data
		approve dengan mengganti nilai 0 menjadi 1,
	7	maka status menjadi aktif
27.	Nomor Proses	3.6
	Nama Proses	Change password
	Deskripsi	Melakukan perubahan password pada akun
	Source	Admin Member
	Input	Login valid, password
	Output	Info data member
	Destination	Admin Member
	1	Admin / member login
	Logika Proses	 Admin / member memilih menu my account Admin / member memilih menu change password
		Admin / member memasukkan password yang baru lalu disimpan.
28.	Nomor Proses	3.7
	Nama Proses	Change username
	Deskripsi	Melakukan perubahan username pada akun
	Source	Admin
	Source	Member
	Input	Login valid, username
	Output	Info data member
SA	VILLATIVE S	Admin
N.	Destination	Member
	Logika Proses	Admin / member login
	2081111 110000	1 Islimi / memori iogin

29.	Nomor Proses Nama Proses Deskripsi	 Admin / member memilih menu my account Admin / member memilih menu change username Admin / member memasukkan username yang baru lalu disimpan. 3.8 Change profile Melakukan perubahan profile pada akun
	Source	AdminMember
	Input	Login valid, identitas diri
	Output	Info data member
	Destination	AdminMember
3	Logika Proses	 Admin / member login Admin / member memilih menu my account Admin / member memilih menu account settings Admin / member memasukkan identitas diri yang baru meliputi birth, email, gender, occupation lalu disimpan.
30.	Nomor Proses	4.1
	Nama Proses	Pengisian form upload video
	Deskripsi	Proses memasukkan video dalam sistem dengan memasukkan identitas video seperti judul, deskripsi, category.
	Source	Admin Member
	Input	Login valid, judul, deskripsi, category, dan video
	Output	Data video
	Destination	Admin Member
	Logika Proses	 Admin / member login Admin / member memilih menu upload Admin / member memasukkan identitas diri dan video yang ingin diupload
31.	Nomor Proses	4.2
	Nama Proses	Pengunggahan video
	Deskripsi	Proses pengunggahan video dan melakukan konversi video
	Source	Admin Member
	Input	Data video
	Output	Info pengunggahan

444		
W.	Destination	• Admin
		Member
		Admin / member login
		Admin / member memilih menu upload
		Admin / member memasukkan identitas diri
	Logika Proses	dan video yang ingin diupload
	ERAL P	Sistem akan melakukan proses pengunggahan
		dan pengkonversian secara otomatis selama
		proses upload
32.	Nomor Proses	6.1
	Nama Proses	Pencarian member
	Deskripsi	Melakukan pencarian member yang diinginkan
	193	• Admin
	Source	• Member
		Pengunjung
	Input	Keyword
	Output	Data member
	Output	Admin
	Destination	
	Destination	• Member
	1.67	• Pengunjung
		Admin / member / pengunjung memasukkan
	Logika Proses	keyword pada kolom pencarian untuk mencari
	7 6	member // // // // // // // // // // // // //
33.		6.2
	Nama Proses	Pilih member
	Deskripsi	Memilih suatu member lalu menampilkan
	Deskripsi	identitas member tersebut
	1	• Admin
	Source	Member
		Pengunjung
	Input	Data member
	Output	Data member
	1	• Admin
	Destination	Member
	Destination	Pengunjung
17		Admin / member / pengunjung memilih nama mambar yang diinginkan
Air	Logika Proses	member yang diinginkan
		Sistem akan menampilkan informasi member
2.1		tersebut
34.		6.3
	Nama Proses	Tampilkan channel member
IA F		Memilih suatu member lalu menampilkan
		channel dari member tersebut yang berisi tentang
	Deskrinsi	
	Deskripsi	video yang diunggah dan difavoritkan oleh member

		LOSING AND PROPERTY
UIA		• Admin
	Source	Member
	MATURE	Pengunjung
Att	Input	Data member
	Output	Data member
		• Admin
	Destination	• Member
		Pengunjung
		Admin / member / pengunjung memilih nama
24		member yang diinginkan
101-	Logika Proses	Sistem akan menampilkan informasi member
		tersebut beserta video apa saja yang diunggah
	43	oleh member tersebut
35.	Nomor Proses	7.1
	Nama Proses	Pengisian form registrasi
		Pengunjung yang ingin menjadi member terlebih
	Deskripsi	dulu mengisi form registrasi member
	Source	Pengunjung
	- ^_	Data registrasi (username, email, password,
	Input	gender, birth, occupation)
	Output	Data registrasi
	Destination	Member
	N E	Pengunjung memilih menu signup
	Logika Proses	Pengunjung mengisikan data diri pada form
		registrasi
36.	Nomor Proses	7.2
	Nama Proses	Aktivasi oleh admin
	-	Admin melakukan aktivasi member melalui
	Deskripsi	menu manage user
	Source	Admin
	Input	Info data member, data registrasi
	Output	Data approve valid
-	Destination	Member Member
	Destination	Admin login
		Admin logili Admin memilih menu manage user untuk
		melakukan pengecekan member
	Logika Proses	
	Logika i ioses	Admin memilih user yang dituju dan memilih link 'edit' untuk melakukan pengeditan data
		approve dengan mengganti nilai 0 menjadi 1,
		maka status menjadi aktif
37.	Nomor Proses	7.3
37.	Nama Proses	Update status member
	Ivallia Fluses	Status member menjadi aktif setelah diaktivasi
RA	Deskripsi	oleh admin
	Source	Pengunjung
	Source	1 Ongunjung

	Input	Data approve valid
V	Output	Login, data member
	Destination	Member
	MAYEU	Pengunjung mengisi form registrasi
	Logika Proses	Pengunjung menunggu statusnya diaktifkan oleh admin
	Photo P	 Setelah aktif, member sudah dapat melakukan login
38.	Nomor Proses	8.1
	Nama Proses	Cek course
	Deskripsi	Menu untuk mengecek course
	Source	Admin Member
	Input	Data course, login valid
	Output	Info data course
	Destination	Admin Member
3	Logika Proses	 Admin/member login Admin/member memilih menu manage course untuk mengecek course yang ada
39.	Nomor Proses	8.2
	Nama Proses	Edit course
	Deskripsi	Proses untuk mengedit course yang ada
	Source	Admin Member
	Input	Data course, login valid
	Output	Info data course
	Destination	Admin Member
	Logika Proses	 Admin/member login Admin/member memilih menu manage course untuk mengecek course Admin/member memilih link 'edit' untuk mengedit course yang diinginkan Admin/member melakukan pengubahan pada data yang dipilih
40.	Nomor Proses	8.3
	Nama Proses	Hapus course
	Deskripsi	Proses untuk menghapus course yang ada
	Source	AdminMember
AA	Input	Data course, login valid
RA	Output	Info data course
	Destination	• Admin

		Member
VI		
	VAY	Admin login
	Logilza Progag	Admin memilih menu manage course untuk manageaele apurse
	Logika Proses	mengecek course
		Admin memilih link 'hapus' untuk
41	N. D.	menghapus course yang diinginkan
41.	Nomor Proses	8.4
	Nama Proses	Pilih course
391	Deskripsi	Memilih course yang ingin dilihat
HS		• Admin
	Source	Member
	G	Pengunjung
	Input	Request course
	Output	Data course
		Admin
	Destination	Member
		Pengunjung
		Admin / member / pengunjung memasukkan
	~	keyword course yang diinginkan pada kolom
	\$ 83	pencarian Search
	Logika Proses	• Sistem akan menampilkan course yang sesuai
		dengan keyword tersebut
		Admin / member / pengunjung memilih
		course yang diinginkan
42.	Nomor Proses	8.5
	Nama Proses	Pengisian form membuat course
	D 1	Proses memasukkan beberapa video yang
	Deskripsi	terkumpul dalam course ke dalam sistem
	G	Admin
	Source	Member
	Input	Login valid, data course,data video
	Output	Data course, data video
		Admin
	Destination	Member
		Admin / member login
		Admin / member login Admin / member memilih menu create course
	Logika Proses	
		Admin / member memasukkan penjelasan agursa dan yidas yang ingin dipulasah
43.	Marray Duagas	course dan video yang ingin diupload 8.6
43.	Nomor Proses Nama Proses	
	Ivailia Proses	Pengunggahan video
	Deskripsi	Proses pengunggahan video yang tergabung ke
	WHITTAN	dalam course
	Source	• Admin
	KEZKUM	Member

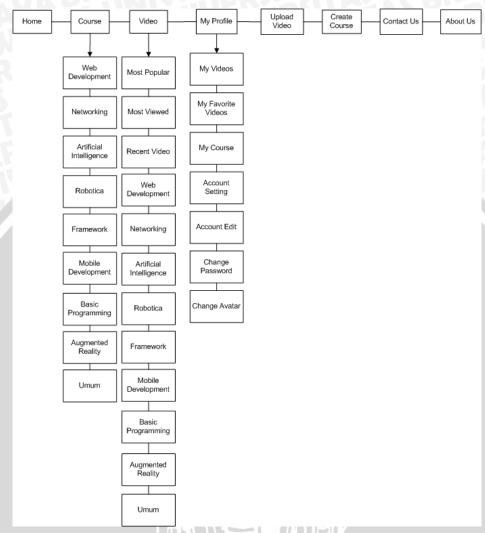
Input	Data video, data course
Output	Info pengunggahan
Destination	AdminMember
Logika Proses	Sistem akan melakukan proses pengunggahan dan pengkonversian secara otomatis selama proses upload setelah admin/member mengisi form create course

Sumber : [Analisis Perancangan]

4.4.5 Perancangan Struktur Menu

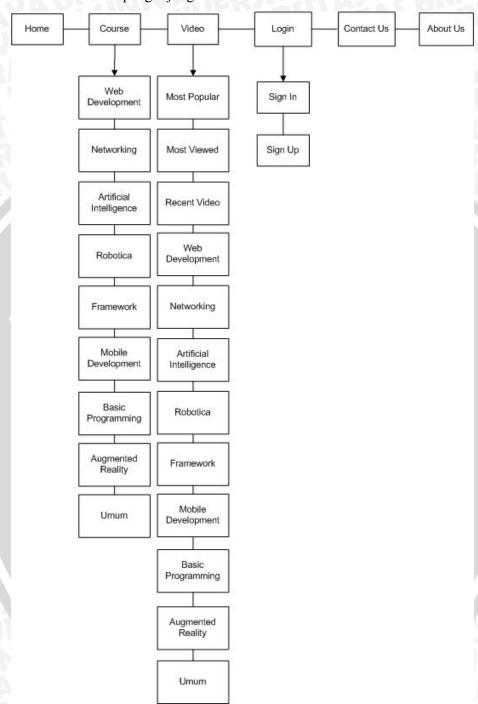
Untuk memperjelas mengenai aplikasi yang akan dibangun berikut ini struktur program yang akan dibangun agar lebih mengetahui proses yang dapat dilakukan ataupun dijalankan. Berikut adalah struktur menu dari sistem informasi berbasis website ini :

a. Struktur menu member



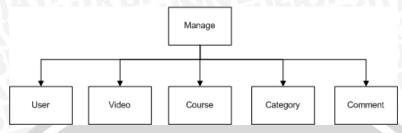
Gambar 4.12 Struktur Menu Member

b. Struktur menu pengunjung



Gambar 4.13 Struktur Menu pengunjung

b. Struktur menu admin



Gambar 4.14 Struktur Menu Admin

Sumber: [Analisis Perancangan]

Detail menu:

1. Signup

Menu ini digunakan oleh pengunjung yang ingin menjadi member dan member akan mengisi data registrasi

2. Login / Sign In

Menu ini digunakan oleh user yang ingin masuk ke dalam sistem dengan memasukkan username dan password

3. Home

Menu ini merupakan tampilan awal web

4. Video

Menu ini berisi video-video dengan disampingnya terdapat kategorikategori. Video-video tersebut dapat diurutkan bedasarkan recent, viewed, dan top rated.

5. Courses

Menu ini berisikan list courses dengan disampingnya terdapat kategorikategori.

6. Upload

Menu ini digunakan untuk mengunggah video ke sistem

7. My Profile

Menu ini digunakan untuk mengatur account, channel, profile, dan video. Contohnya seperti change password, change username, change profile, dan lain-lain.

BRAWIJAYA

8. Create Course

Menu ini digunakan untuk membuat course (video tutorial berangkai)

9. My Videos

Menu ini digunakan member untuk mengatur videonya. Video tersebut bisa dilihat, diedit, dan dihapus.

10. My Favorites Videos

Menu ini digunakan member untuk mengatur video yang difavoritkan. Video tersebut bisa dilihat atau didelete (mengembalikan favorite).

11. My Courses

Menu ini digunakan member untuk mengatur coursenya. Course tersebut bisa dilihat, diedit, dan dihapus.

12. Account Edit

Menu ini digunakan member untuk mengedit informasi profilenya.

13. Change Password

Menu ini digunakan member mengganti passwordnya.

14. Change Avatar

Menu ini digunakan member untuk mengganti avatar..

15. Contact Us

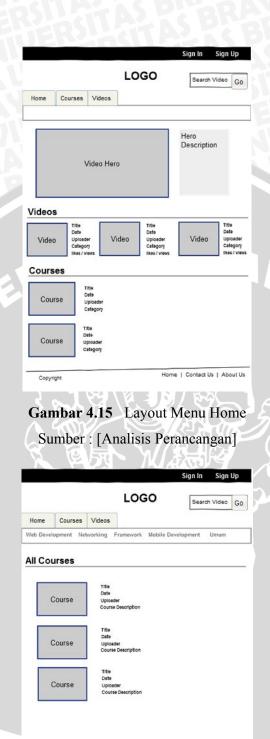
Menu ini digunakan member / pengunjung untuk memberikan saran atau pesan atau permasalahan lainnya ke admin.

16. About Us

Menu ini berisi informasi tentang profile website

4.4.3 Perancangan Antar Muka

Perancangan antarmuka pemakai atau user *interface* akan sangat menentukan kemudahan bagi pengguna. Berikut ini adalah rancangan antarmuka yang akan digunakan nantinya :



Gambar 4.16 Layout Menu Courses

Home | Contact Us | About Us

Copyright



Gambar 4.17 Layout Menu Videos

Sumber: [Analisis Perancangan]



Gambar 4.18 Layout Menu My Profile

			LOGO	Search Video	3(
Home	Courses	Videos			
Uplo	ad Your	Video			
Title					
text					
Descr	iption				
Choose Categ	se File	Choose	è		
All	lory	~			
Tags					
text					
S	aved	_			
Gan	nbar	4.19	Layout N	Menu Uplo	ล



Gambar 4.20 Layout Menu Create Course

			100	_			_
			LOG	U	Search	Video	G
Home	Courses	Videos					
Uplo	ad Your	Video	Course				
Cours	e Title						
text							
Curre	nt video i	n this co	urse				
No Vi	deo						
1 Ad	d Video detai	Delete		Browse			
2 Adi	d Video detai	Delete		Browse			
Add M	ore Video to	this Course					
	is Step	Uploa	d Video				
Skip th	no o cup						
Skip th	Воскр						
Skip ti	З						

Gambar 4.21 Layout Upload Video Course

Sumber : [Analisis Perancangan]

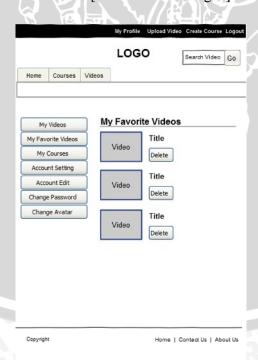
	LOGO	Search Video	Go
Home Courses \	/ideos		
Sign Up			_
First Name	text		
Last Name	text		
Username	text		
Password	text		
Confirm Password	text		
Email	text		
Date of Birth	text		
Gender	Male Female		
Occupation	Student Profesor Staff		
Sign Up			

Gambar 4.22 Layout Menu Signup



Gambar 4.23 Layout Menu My Videos

Sumber: [Analisis Perancangan]



Gambar 4.24 Layout Menu My Favorite Videos



Gambar 4.25 Layout Menu My Courses

Sumber: [Analisis Perancangan]



Gambar 4.26 Layout Menu Account Edit

My Videos My Favorite Videos My Courses Account Setting Account Edit Change Password Change Avatar LOGO Search Video G Change Your Password Exxt New Password Exxt Exxt		My Profile Upload	Video Create Course Lo
My Videos My Favorite Videos My Courses Account Setting Account Edit Change Your Password Lext New Password Ext Save Password		LOGO	Search Video G
My Favorite Videos My Courses Account Setting Account Edit Change Password Current Password Lext text text Lext Le	Home Courses Vid	leos	
My Favorite Videos My Courses New Password Account Setting Account Edit Change Password Current Password text Ext Ext	My Videos	Change Your F	assword
Account Setting Account Edit Change Password Save Password Save Password		Current Password	text
Account Edit Save Password Change Password	My Courses	New Password	text
Change Password			
		Save Password	
•	- Control Terror Control		

Gambar 4.27 Layout Menu Change Password

Sumber : [Analisis Perancangan]



Gambar 4.28 Layout Menu Change Avatar

BRAWIJAY/

4.5 Seleksi Komponen

Setelah mengetahui beberapa persyaratan yang diperlukan oleh sistem, maka akan dilakukan penyeleksian komponen yang nantinya akan digunakan dalam sistem. Penyeleksian dilakukan setelah melihat beberapa kekurangan dan kelebihan yang dimiliki oleh komponen. Berikut adalah komponen-komponen yang dipilih:

• Video Player

Untuk komponen video player, sistem akan menggunakan Flowplayer. Flowplayer lebih dipilih daripada JWPlayer dengan pertimbangan yaitu, **JWPlayer** dari segi teknisi lebih mudah mengkonfigurasi pseudostreaming tetapi kurang lengkap akan ketersediaan plugin seperti yang dibutuhkan sistem, seperti bitrate selection dan lain-lain. Sedangkan Flowplayer mempunyai ketersediaan yang sangat lengkap akan pluginplugin tersebut. Sistem akan menggunakan Flowplayer dengan plugin bitrate selection (plugin yang berfungsi untuk memilih kualitas video berdasarkan bitrate) dan dimodifikasi agar pemilihan video dilakukan berdasarkan transfer rate user.

Encoder

Untuk *encoder*, sistem akan menggunakan FFmpeg dibandingkan menggunakan Mencoder. FFmpeg mempunyai *library* yang lebih lengkap dan FFmpeg dapat digunakan untuk membuat *thumbnail* video. Selain itu, dokumentasi dari web FFmpeg lebih lengkap daripada Mencoder. FFmpeg juga lebih dikenal luas dan banyak dipakai sebagai konversi video (banyak juga disupport oleh web *hosting*) dibandingkan dengan Mencoder.

• Metadata Injector

Untuk komponen metadata *injector*, sistem memilih menggunakan Yamdi. Hal itu dikarenakan dalam segi performansi, Yamdi jauh lebih cepat daripada Flvtool2. Flvtool2 kurang cocok digunakan untuk

menginject metadata pada ukuran file yang besar karena Flvtool2 membaca keseluruhan file ke dalam memori. Selain itu, beberapa format file tidak disupport oleh Flvtool2. Yamdi lebih mensupport banyak format file. Dalam hal pemakaian memori, Yamdi juga menggunakan memori lebih kecil daripada Flvtool2.

4.6 Perancangan Sistem

Setelah tahap analisis sistem, sebagai tindak lanjut untuk menyelesaikan masalah, maka dibuat suatu rancangan sistem. Perancangan sistem adalah tahapan yang berguna untuk memperbaiki efisiensi kerja suatu sistem yang telah ada. Setiap perancangan sistem dapat digambarkan sebagai tahap untuk membangun suatu sistem dan mengkonfigurasikan komponen-komponen perangkat lunak dan perangkat kerasnya, sehingga menghasilkan sistem yang lebih baik.

Sistem *video on demand* berbasis web terdiri dari dua bagian, yaitu bagian administrator dan bagian klien. Administrator bertugas mengatur jalannya sistem seperti pengolahan video dan member, sedangkan klien yang mengakses informasi didalamnya. Klien dapat menjadi member dan mengakses video didalamnya, serta dapat mengunggah video yang diinginkan.

Data masukan yang dibutuhkan sebagai sumber informasi pada proses *streaming* adalah video. Video dapat berupa *pre-recorded* (yang sudah melalui proses *encoding*, sudah mempunyai metadata, dan tersimpan dalam *database*), maupun video yang diunggah ke dalam sistem oleh member.

Perancangan sistem terdiri atas perancangan basis data dan perancangan procedural. Berikut adalah penjelasan untuk perancangan basis data dan perancangan prosedural.

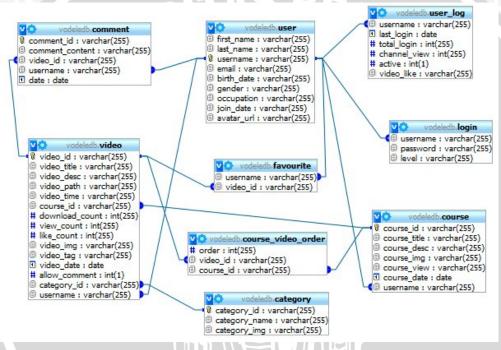
BRAWIJAY

4.6.1 Perancangan Basis Data

Pada tahapan perancangan basis data, akan dilakukan pemetaan model konseptual ke model basis data yang akan dicapai. Dalam perancangan basis data terdapat 2 model yaitu skema relasi dan perancangan struktur tabel.

4.6.1.1 Skema Relasi

Rancangan hubungan antar tabel dalam aplikasi yang akan dibangun dapat dilihat pada gambar berikut :



Gambar 4.29 Skema Relasi Database

Sumber : [Analisis Perancangan]

4.6.1.2 Perancangan Struktur Tabel

Perancangan ini adalah berupa perancangan struktur file, struktur file dirancang dalam bentuk tabel dimana setiap relasi akan dibuat menjadi suatu tabel khusus. Tabel 4.2 sampai 4.10 menunjukkan struktur tabel pada database.

Tabel 4.2 Struktur Tabel Video

Field	Type	Key	Keterangan
video_id	varchar(255)	primary	Nomor id dari video
video_title	varchar(255)		Judul video
video_desc	varchar(255)	-	Deskripsi tentang video
video_path	varchar(255)	-	Lokasi video di database
video_time	varchar(255)	-	Durasi video
course_id	varchar(255)	foreign	Nomor id dari course
download_count	int(255)	-	Banyaknya download
view_count	int(255)	-	Banyaknya view video
like_count	int(255)		Banyaknya yang menyukai video
video_img	varchar(255)		Gambar thumbnail video
video_tag	varchar(255)		Tag video untuk mempermudah search
video_date	Date		Tanggal upload video
category_id	varchar(255)	foreign	Nomor id dari kategori
username	varchar(255)	foreign	Nama user yang mengupload video
allow_comment	int(1)		Mengaktifkan comment

Tabel 4.3 Struktur Tabel User

Field	Type	Key	Keterangan
username	varchar(255)	primary	Username dari member
first_name	varchar(255)	-	Nama awal member
last_name	varchar(255)	-	Nama terakhir member
gender	varchar(255)	-	Jenis kelamin member
birth_date	varchar(255)	11-47	Tanggal lahir member
occupation	varchar(255)	U	Status member
avatar_url	varchar(255)	14-10	Lokasi avatar member

join_date	varchar(255)	43.6	Tanggal registrasi
email	varchar(255)	- 1	Email member

Sumber: [Analisis Perancangan]

Tabel 4.4 Struktur Tabel Category

Field	Type	Key	Keterangan
category_id	varchar(255)	primary	Id dari kategori
category_name	varchar(255)	A3	Nama kategori
category_img	varchar(255)	-	Deskripsi kategori

Sumber : [Analisis Perancangan]

Tabel 4.5 Struktur Tabel Course

Field	Type	Key	Keterangan
course_id	varchar(255)	primary	Id dari course
course_title	varchar(255)		Judul course
course_desc	varchar(255)	} \	Deskripsi course
course_img	varchar(255)	大學	Image course
course_view	varchar(255)		Banyaknya view course
course_date	Date		Tanggal pembuatan course
username	varchar(255)	foreign	Nama pemilik course

Sumber : [Analisis Perancangan]

Tabel 4.6 Struktur Tabel Video_comment

Field	Type	Key	Keterangan
comment_id	varchar(255)	primary	Id dari komentar suatu video
comment_content	varchar(255)	-	Isi komentar
video_id	varchar(255)	foreign	Id video yang dikomentari
username	varchar(255)	foreign	Username pengkomentar
date	varchar(255)		Tanggal pemberian komentar

Tabel 4.7 Struktur Tabel Favourite

Field	Type	Key	Keterangan
username	varchar(255)	foreign	Username member
video_id	varchar(255)	foreign	Id dari video

Sumber : [Analisis Perancangan]

Tabel 4.8 Struktur Tabel user_log

Field	Type	Key	Keterangan
username	varchar(255)	foreign	Username member
last_login	varchar(255)	-	Terakhir kali login
total_login	int(255)	-	Banyaknya login
channel_view	int(255)		Banyaknya view pada channel
active	int(1)		Status user
video_like	Text		Id video yang di-like member

Sumber : [Analisis Perancangan]

Tabel 4.9 Struktur Tabel login

Field	Type	Key	Keterangan
username	varchar(255)	Foreign	Username member
password	varchar(255)	1251	Password member
Level	varchar(255)		Level user

Sumber: [Analisis Perancangan]

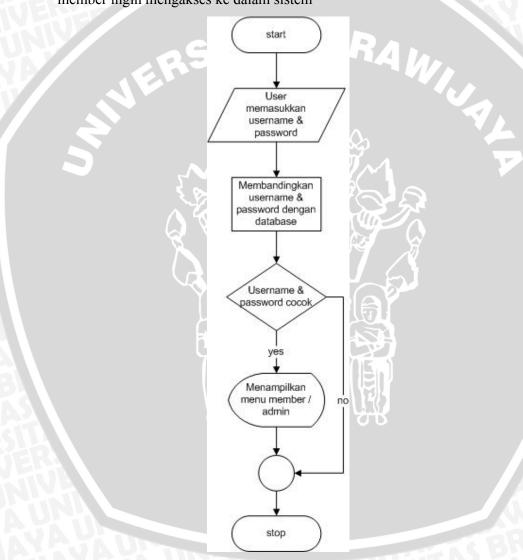
Tabel 4.10 Struktur Tabel course_video_order

Field	Type	Key	Keterangan
video_id	varchar(255)	foreign	Nomor video
course_id	varchar(255)	foreign	Nomor course
order	int(255)		Urutan video pada course

4.6.2 Perancangan Prosedural

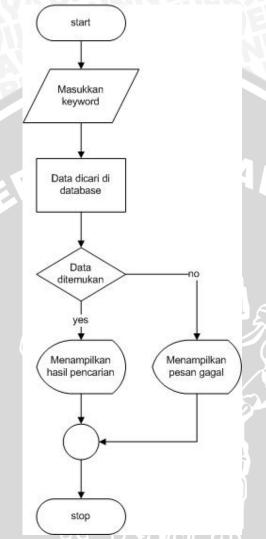
Perancangan prosedural mengubah elemen-elemen struktural dari arsitektur program ke dalam suatu deskripsi prosedural dari komponen-komponen perangkat lunak. Adapun perancangan prosedural untuk sistem yang akan dibangun adalah sebagai berikut:

1. Prosedural login, merupakan prosedur yang terjadi ketika admin atau member ingin mengakses ke dalam sistem



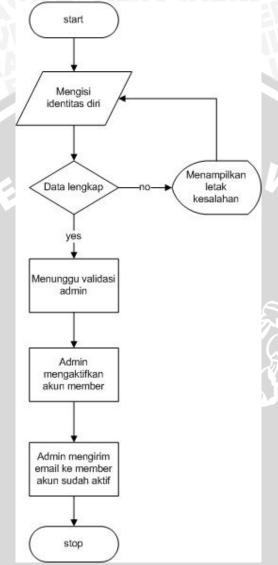
Gambar 4.30 Flowchart Proses Login

2. Prosedural pencarian, prosedur ini dilakukan ketika user akan melakukan pencarian data, baik berupa video maupun course.



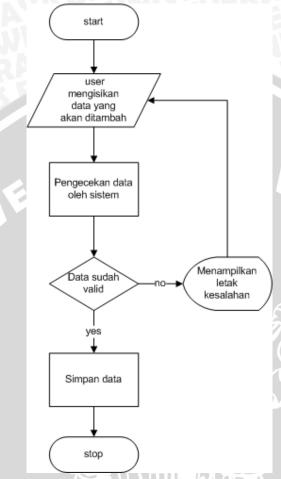
Gambar 4.31 Flowchart Proses Pencarian Data

3. Prosedural register, prosedur ini dilakukan ketika pengunjung akan melakukan pendaftaran menjadi member.



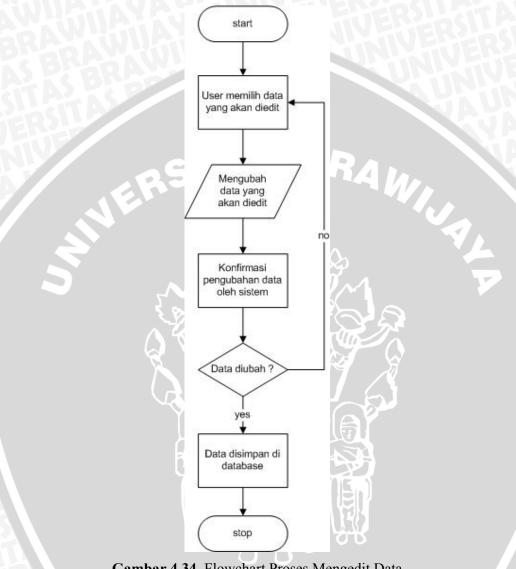
Gambar 4.32 Flowchart Proses Registrasi

4. Prosedural menambah data, adalah prosedur ketika admin atau member ingin menambahkan sebuah data.



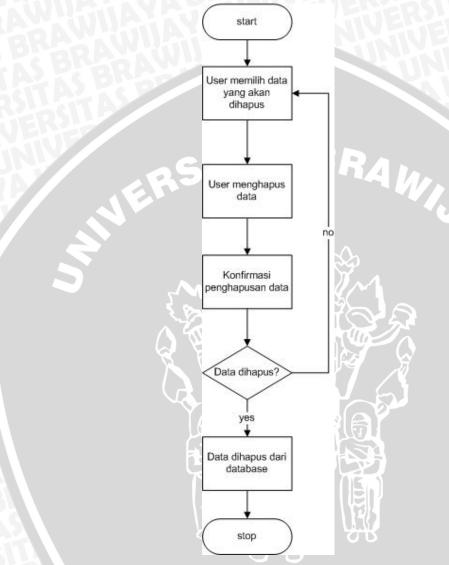
Gambar 4.33 Flowchart Proses Menambah Data

5. Prosedural mengedit data, adalah prosedur ketika admin atau member ingin mengedit data.



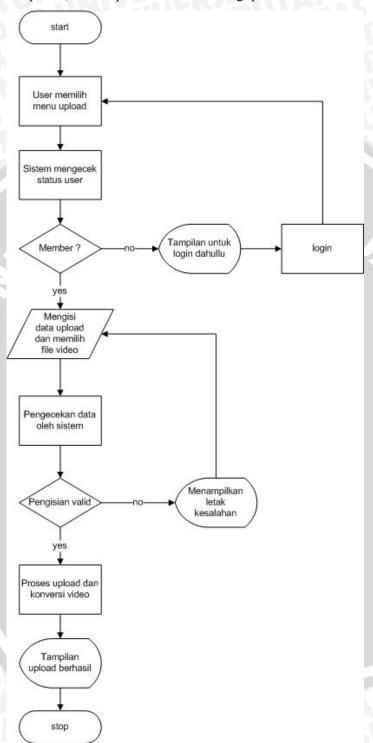
Gambar 4.34 Flowchart Proses Mengedit Data

6. Prosedur menghapus data, adalah prosedur ketika admin atau member ingin menghapus sebuah data.



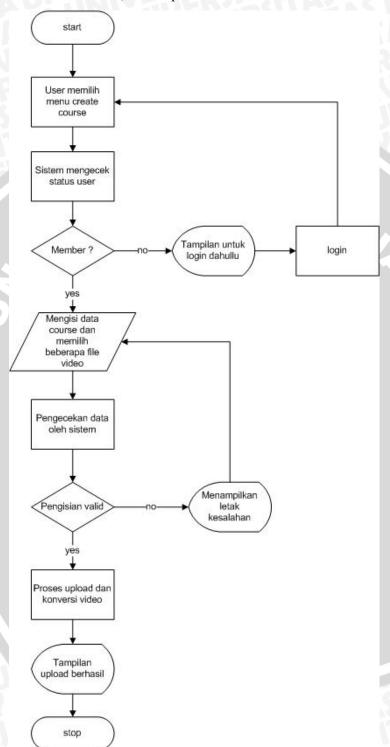
Gambar 4.35 Flowchart Proses Menghapus Data

7. Prosedural upload, adalah prosedur untuk mengupload video



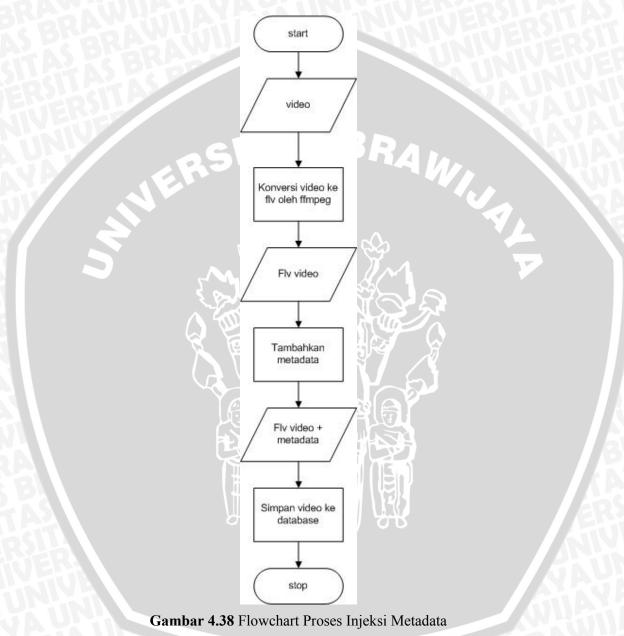
Gambar 4.36 Flowchart Proses Upload Video

8. Prosedural Create Course, adalah prosedur untuk membuat course



Gambar 4.37 Flowchart Proses Create Course

9. Prosedural *encoding* dan injeksi metadata, adalah prosedur yang dilakukan ketika proses upload



BAB V IMPLEMENTASI

Pada bab ini akan dilakukan implementasi terhadap sistem. Setelah sistem dianalisis dan didesain secara rinci, maka akan menuju tahap implementasi. Implementasi sistem merupakan tahap meletakkan sistem sehingga siap untuk dioperasikan. Implementasi bertujuan untuk mengkonfirmasi modul-modul perancangan, sehingga pengguna dapat memberi masukan kepada pengembangan sistem.

5.1 Spesifikasi Sistem

Hasil analisis kebutuhan dan perancangan perangkat lunak yang telah diuraikan pada Bab 4 menjadi acuan untuk melakukan implementasi menjadi sistem yang dapat berfungsi sesuai dengan kebutuhan. Spesifikasi sistem diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak.

5.1.1 Spesifikasi Perangkat Keras

Layanan *Video On Demand* dengan menggunakan HTTP *Pseudostreaming* ini menggunakan sebuah komputer dengan spesifikasi perangkat keras yang dijelaskan pada Tabel 5.1 dan 5.2

a. Spesifikasi Perangkat Keras Server

Tabel 5.1 Spesifikasi Perangkat Keras Server

No	Perangkat Keras	Keterangan		
1	Prosesor	Intel Core i3 CPU M350 @2.27 Ghz		
2	Hardisk	320 Gb		
3	RAM	2048MB		
4	VGA	726MB		
5	Monitor	14" LCD		

Sumber : [Implementasi]

BRAWIJAY

b. Spesifikasi Perangkat Keras Client

Tabel 5.2 Spesifikasi Perangkat Keras Client

No	Perangkat Keras	Keterangan
1	Prosesor	Intel Core i3 CPU M350 @2.27 Ghz
2	Hardisk	320 Gb
3	RAM	2048MB
4	VGA	726MB
5	Monitor	14" LCD

Sumber : [Implementasi]

5.1.2 Spesifikasi Perangkat Lunak

Layanan *Video On Demand* dengan menggunakan HTTP *pseudostreaming* ini menggunakan sebuah komputer dengan spesifikasi perangkat lunak yang dijelaskan pada Tabel 5.3 dan 5.4

a. Spesifikasi Perangkat Lunak Server

Tabel 5.3 Spesifikasi Perangkat Lunak Server

No	Perangkat Lunak	Keterangan
1	Sistem Operasi	LINUX UBUNTU 12.04
2	Bahasa Pemrograman	PHP
3	DBMS	MySQL
4	Manajemen DBMS	PHPMyAdmin
5	Web Server	Apache
6	Converter Video	FFMPEG
7	Injector Metadata	Yamdi
8	Flash Player	Flowplayer
9	Browser	Mozilla Firefox
10	Code Editor	Sublime Text 2
11	Plugin Browser	Adobe Flash Player Plugin

Sumber : [Implementasi]

b. Spesifikasi Perangkat Lunak Client

Tabel 5.4 Spesifikasi Perangkat Lunak Client

No	Perangkat Keras	Keterangan
1	Sistem Operasi	Windows / Linux
2	Browser	Mozilla Firefox / Google Chrome
3	Plugin Browser	Adobe Flash Player

Sumber: [Implementasi]

5.2 Konfigurasi Sistem

Agar rancangan sistem dapat diimplementasikan, maka ada beberapa kebutuhan yang perlu dikonfigurasi terlebih dahulu. Berikut adalah langkahlangkah yang harus ditempuh:

- Yang pertama kali dilakukan setelah menginstall server-server yang diperlukan seperti web server dan database server, adalah membuat basis data terlebih dahulu. Untuk implementasi basis data dapat dilihat di Lampiran 1.
- Setelah membuat basis data, maka dilakukan instalasi perangkat lunak yang diperlukan, yakni FFMPEG, FFMPEG-PHP, Yamdi, dan Flowplayer.
- 3) Setelah semua penginstalasian selesai, maka dilakukan konfigurasi terhadap Apache web server karena jika ingin membuat FLV streaming, harus dilakukan penginstalan modul flv terlebih dahulu. Cara konfigurasi Apache dapat dilihat di Lampiran 2.
- 4) Agar proses download video dapat berjalan dengan lancar dengan adanya file video yang berukuran besar, maka harus dikonfigurasi terlebih dahulu file php.ini yang terletak di /etc/php5/apache2/php.ini. Cara konfigurasi php.ini bisa dilihat di Lampiran 3.
- 5) Konfigurasi flowplayer yang dimasukkan dalam script program dapat dilihat pada Lampiran 4.

BRAWIJAY

Setelah semua langkah diatas sudah ditempuh maka kita dapat membuat tampilan web yang akan digunakan sebagai layanan *video on demand* ini.

5.3 Implementasi Algoritma

Sistem VOD ini memiliki beberapa proses atau *method* yang digunakan. Pada penulisan skripsi ini hanya dicantumkan algoritma dari beberapa proses (operasi) utama saja sehingga tidak semua *method* akan dicantumkan. Proses utama yang akan dibahas adalah proses *upload* dan proses deteksi kecepatan. Proses *upload* dibagi menjadi 4 proses kecil, yakni *create filename, create video folder, ffmpeg convert,* dan *yamdi inject*. Implementasi algoritma ini akan direpresentasikan dalam bentuk *pseudocode* dari algoritma – algoritma tersebut.

5.3.1 Algoritma Create Filename

Create filename merupakan proses yang dilakukan pertama pada saat mengupload video. Nama file video tidak akan pernah sama karena nama file dienkripsi menjadi md5. Implementasi algoritma create filename ditunjukkan pada tabel 5.5.

Tabel 5.5 Implementasi Algoritma Create Filename

```
Pseudocode Create Filename
Method : create filename
PARAMETER uid, title, tmp
DECLARATION
uid IS String
title IS String
tmp IS String
enc uid IS String
enc title IS String
tmp arr IS Array
enc_tmp IS String
filename IS String
DESCRIPTION
1. uid <- md5(uid)
2. enc uid <- substr(uid,0,6)</pre>
3. title <- md5(title)</pre>
4. enc title <- substr(title,0,6)
```

```
5. tmp_arr <- explode("/", tmp)
6. enc_tmp = substr(tmp_arr[2],3,15)
7. filename <- enc_uid.enc_title.enc_tmp
8. RETURN filename
```

Sumber: Implementasi

Penjelasan dari algoritma create filename pada tabel 5.5., yaitu :

- 1. baris 1 dan baris 3 menunjukkan proses enkripsi oleh md5 kepada variabel uid dan title.
- 2. baris 2 dan 4 menunjukkan hasil dari enkripsi oleh md5 kepada variabel uid dan title diolah oleh fungsi substr sehingga menjadi 6 karakter dan kemudian disimpan dalam variabel yang diberi prefix enc_. Fungsi substr adalah fungsi native php yang digunakan untuk memotong karakter string.
- 3. pada baris 5 menunjukkan variable tmp dipecah dengan fungsi explode sehingga menjadi variabel array yang berisi pecahan sesuai parameter yang diberikan kepada fungsi explode dan hasilnya diberikan kepada variabel tmp_arr. Fungsi explode adalah berfungsi untuk memecah variabel berdasar separator yang diberikan.
- 4. pada baris 6 diambil array ke 2 dari variabel tmp yang sebelumnya sudah dipecah untuk mendapatkan id dari upload session, dan kemudian dipotong oleh substr sehingga mendapatkan id dari variabel tmp yang kemudian disimpan dalam variabel enc tmp
- 5. pada baris 7, semua isi dari variabel ber prefix enc_ disatukan ke dalam sebuah variabel filename.
- 6. pada baris 8, fungsi ini mengembalikan value dari filename.

5.3.2 Algoritma Create Video Folder

Create video folder merupakan proses yang dilakukan setelah proses create filename pada saat mengupload video. Proses ini melakukan proses membuat folder dari setiap video yang diupload. Setiap member mempunyai folder sendiri yang terdiri dari folder course dan folder video. Isi dari folder video adalah video yang sudah dikonvert ke video dengan kualitas yang berbeda. Implementasi algoritma create video folder ditunjukkan tabel 5.6.

Tabel 5.6 Implementasi Algoritma Create Video Folder

```
Pseudocode Create Video Folder
Method : create video folder
PARAMETER username, filename, tmp file
DECLARATION
username IS String
filename IS String
tmp file IS String
user video path IS String
each video path IS String
video file path IS String
video thumb path IS String
USER_FOLDER IS Constant | define(USER FOLDER, '../user/');
DESCRIPTION
1. user video path <- USER FOLDER.username ."/video/"
2. each video path <- user video path . filename
3. video file path <- each video path . "/" . filename
4. video thumb path <- each video path ."/thumb/"
5. mkdir(each video path, 0777)
6. mkdir(video thumb path, 0777)
7. chmod(each video path, 0777)
8. chmod(video_thumb_path, 0777)
9. move uploaded file (tmp file, video file path)
10. chmod(video file path, 0777)
11. return each video path
```

Sumber: Implementasi

Penjelasan dari algoritma *create filename* pada tabel 5.6., yaitu :

- 1. pada baris 1 dilakukan konkatenasi antara USER_FOLDER dengan variabel username dan ditambahkan /video sehingga menghasilkan value yang menunjukkan lokasi menuju folder video oleh user yang kemudian disimpan dalam variabel user_video_pat.h
- pada baris 2 dilakukan konkatenasi juga variabel user_video_path dengan variable filename sehingga menghasilkan value yang mengarah ke url masing - masing folder video yang disimpan dalam variabel each_video_path.
- 3. pada baris 3 juga dilakukan konkatenasi antara variabel each_video_path dan variabel filename sehingga menghasilkan value yang mengarah ke url masing masing file video dan disimpan dalam variabel video file path.
- 4. pada baris 4 dilakukan konkatenasi juga variabel each video path

ditambah string /thumb sehingga menghasilkan value folder yang menyimpan thumbnail video yang kemudian disimpan dalam variabel video thumb path.

- 5. pada baris 5 dan 6 dilakukan mkdir utuk url each video path dan video thumb path. Fungsi mkdir adalah fungsi native php untuk membuat folder.
- 6. pada baris 7 dan 8 adalah kelanjutan dari baris 5 dan 6 dimana pada baris ini chmod digunakan untuk mengganti permissions menjadi 0777 kapada folder yang bersangkutan.
- 7. pada baris 9 dilakukan move file dari url tmp upload session ke video file path oleh fungsi native php yaitu move uploaded file.
- 8. setelah itu pada baris 10, url video file path diganti permissionnya sehingga menjadi 0777 atau public
- 9. pada baris terakhir fungsi ini mengembalikan variabel each video path

Algoritma FFmpeg Convert

FFmpeg Convert merupakan proses yang dilakukan setelah proses create video folder pada saat mengupload video. Proses ini melakukan proses mengkonvert video menjadi FLV sekaligus dikonvert ke tiga video yang berbeda kualitasnya sehingga dapat mendukung fitur rekomendasi kualitas video berdasarkan transfer rate user. Implementasi algoritma FFmpeg convert ditunjukkan tabel 5.7.

Tabel 5.7 Implementasi Algoritma FFmpeg Convert

```
Pseudocode FFmpeg Convert
METHOD : ffmpeg convert
PARAMETER each video path, filename
DECLARATION
each video path IS String
filename IS String
video source IS String
ffmpeg result IS Array
                               BRAWA
size IS Array
video result IS Array
video IS Array
video path IS String
video size IS String
DESCRIPTION
1. video source <- each video path."/".filename
2. ffmpeg result[0] = each video path . "/360sd " .
filename.".flv size >> 640x360 "
3. ffmpeg result[1] = each video path . "/480sd "
filename.".flv size >> 854x480"
4. ffmpeg result[2] = each video path . "/720hd " .
filename.".flv size >> 1280x720"
5. foreach ffmpeg result as key => value
6. video = explode("size >>", value)
7. video path = video[0]
8. video size = video[1]
9. exec("ffmpeg -i " . video_source . " -ar 44100 -s ".
video_size ." ". video path)
10. array_push(video_result, $video_path)
11. chmod(video path, 0777)
12. end foreach
13. return video result
```

Sumber: Implementasi

Penjelasan dari algoritma FFmpeg Convert pada tabel 5.7., yaitu :

- pada baris 1 dilakukan konkatenasi variabel each_video_path dan filename sehingga menghasilkan value yang mengarah ke folder video dan disimpan dalam variabel video_source.
- 2. pada baris 2 sampai 4 dilakukan konkatenasi variabel each_video_path dan di definisikan masing-masing kualitas videonya, serta didefinisikan value resolusinya, dan kemudian masing masing dimasukkan dalam variabel array ffmpeg result sebagai string yang nantinya akan diolah.
- 3. pada baris 5 sampai baris 12 dilakukan proses foreach terhadap ketiga item

BRAWIJAY/

di dalam variabel ffmpeg result, yang berfungsi untuk melakukan proses converting ke file flv oleh ffmpeg yang ditunjukkan pada baris 9. Fungsi exec digunakan untuk mengeksekusi perintah terminal.

4. pada baris 13 dikembalikan value video result.

5.3.4 Algoritma Yamdi Inject

Yamdi Inject merupakan proses yang dilakukan setelah proses FFmpeg convert pada saat mengupload video. Proses ini melakukan proses injeksi metadata ke file FLV. Metadata diperlukan agar dapat mengetahui informasi dari file video serta mendukung adanya fitur pseudostreaming. Implementasi algoritma Yamdi inject ditunjukkan tabel 5.8.

Tabel 5.8 Implementasi Algoritma Yamdi Inject

```
Pseudocode Yamdi Inject
METHOD : yamdi inject
PARAMETER ffmpeg result, each video path, filename
DECLARATION
each_video_path IS String
filename IS String
ffmpeg result IS String
video source IS String
yamdi result IS Array
video result IS Array
DESCRIPTION
1. video source = ffmpeg result
2. yamdi_result[0] = each_video_path ."/v_". filename .
" 360sd.flv"
3. yamdi result[1] = each video path ."/v ". filename .
" 480sd.flv"
4. yamdi_result[2] = each_video_path ."/v_". filename .
" 720hd.flv"
5. foreach yamdi result as key => value
6. exec("yamdi -i " . video source[key] . " -o " . value .
'' - 1''
7. chmod(value, 0777)
8. array push (video result, value)
9. endforeach
10. return video result
```

Sumber: Implementasi

BRAWIJAYA

Penjelasan dari algoritma FFmpeg Convert pada tabel 5.8., yaitu :

- pada baris 1 proses memasukkan value dari ffmpeg_result yang merupakan path dari proses converting sebelumya ke variabel video source.
- 2. pada baris 2 sampai 4 dilakukan konkatenasi variabel each_video_path untuk mempersiapkan path result dari inject yamdi, dan kemudian value ini dimasukkan ke dalam masing masing index variabel array yamdi result[]
- 3. pada baris 5 sampai 9 dilakukan proses foreach untuk mengekstrak array yamdi_result yang kemudian diproses untuk diinject metadatanya oleh baris 6.
- 4. pada akhirnya dikembalikan variabel video result.

5.3.5 Algoritma Speed Detection

Speed detection merupakan proses yang dilakukan agar video player dapat melakukan deteksi transfer rate yang nantinya digunakan untuk rekomendasi kualitas video mana yang akan diputar. Implementasi algoritma speed detection ditunjukkan tabel 5.9.

Tabel 5.9 Implementasi Algoritma Speed Detection

```
Pseudocode Speed Detection
DECLARATION
imageAddr IS String
startTime IS Integer
endTime IS Integer
downloadSize IS Integer
download IS Object
startTime IS Object
duration IS Float
bitsLoaded IS Float
speedBps IS Float
speedMbps IS Float
speed IS Integer
v360 IS Boolean
v480 IS Boolean
v720 IS Boolean
```

```
DESCRIPTION
_____
1. var imageAddr <-
"http://10.42.0.36/vodele/assets/images/muse.JPG" + "?n=" +
Math.random()
2. var downloadSize <- 3065328;</pre>
3. var download <- new Image();</pre>
4. startTime <- (new Date()).getTime();</pre>
5. download.src <- imageAddr;</pre>
6. download.onload
7. endTime <- (new Date()).getTime();</pre>
8. var duration <- (endTime - startTime) / 1000;</pre>
//Math.round()
9. var bitsLoaded <- downloadSize * 8;</pre>
10. var speedBps <- (bitsLoaded / duration).toFixed(2);</pre>
11. speedKbps <- (speedBps / 1024).toFixed(2);</pre>
12. var speedMbps <- (speedKbps / 1024).toFixed(2);
13. speed <- speedKbps;
14. if speed>0 and speed <500
15. v360 <- true
16. elseif speed>=500 and speed <1000
17. v480 <- true
18. elseif speed>=1000
19. v720 <- true
20. Endif
```

Sumber: Implementasi

Penjelasan dari algoritma Detection Speed pada tabel 5.9., yaitu :

- 1. pada baris 1 didefinisikan url *image* yang akan di*download* sebagai test.
- 2. pada baris 2 juga didefinisikan berapa besar file yang akan di download.
- 3. pada baris 3 didefinisikan variabel download sebagai sebuah object
- 4. pada baris 4 di inisiasi start time sebelum file di download
- 5. pada baris ke 5 url yang sudah didefinisikan sebelumnya dimasukkan ke dalam *object download* dan pada baris 6 dilakukan proses *download image*
- 6. pada baris 7 diperoleh *endtime* setelah *download* selesai
- 7. pada baris 8 sampai 12 dilakukan penghitungan kecepatan *download speed user* dengan memperhitungkan antara besar file dan lama file ketika di *download* sampai selesai
- 8. pada baris 13, variabel speed menyimpan hasil kecepatan user download dalam Kbps
- 9. pada baris 14 sampai 20 dilakukan seleksi untuk menetukan pemilihan kualitas terhadap download speed yang sudah dihitung.

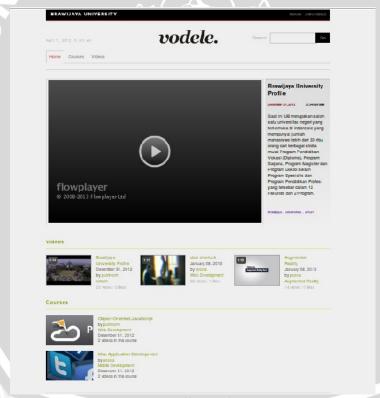
BRAWIJAY

5.4 Implementasi Antar Muka

Implementasi antarmuka dilakukan berdasarkan proses perancangan yang telah dilakukan pada bab 4. Sistem *video on demand* ini dibuat berbasis *web* sehingga implementasi antarmuka dibangun dengan bahasa pemrograman php dan html. Berikut ini akan ditunjukkan beberapa halaman antarmuka yang digunakan sistem.

5.4.1 Implementasi Antarmuka Halaman Home

Implementasi halaman *Home* adalah halaman antarmuka yang ditampilkan pertama kali saat memasuki web. Berikut gambar 5.1 menunjukkan antarmuka halaman *home*.



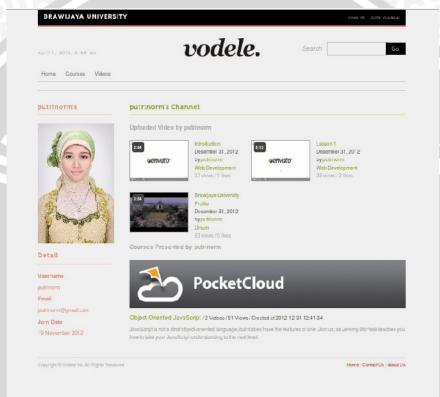
Gambar 5.1 Tampilan Halaman Home

Sumber : [Implementasi]

BRAWIJAY

5.4.2 Implementasi Antarmuka Halaman My Profile

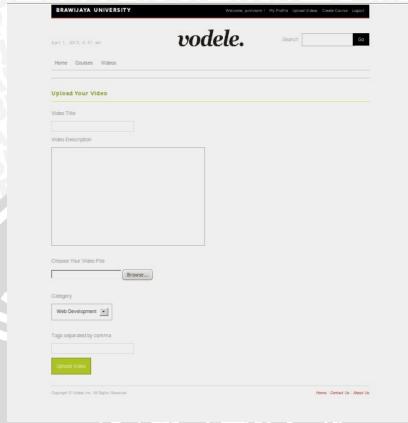
Implementasi halaman *My Profile* adalah halaman antarmuka yang menampilkan *dashboard* profil dari user yang sudah menjadi member. Halaman ini berisi daftar video dan *course* yang dimiliki oleh member yang bersangkutan. Berikut gambar 5.2 menunjukkan antarmuka halaman *my profile*.



Gambar 5.2 Tampilan Halaman *My Profile*Sumber : [Implementasi]

5.4.3 Implementasi Antarmuka Halaman Upload Video

Implementasi halaman *Upload* Video adalah halaman antarmuka yang menampilkan form yang digunakan member untuk mengupload videonya. Berikut gambar 5.3 menunjukkan antarmuka halaman *upload* video.

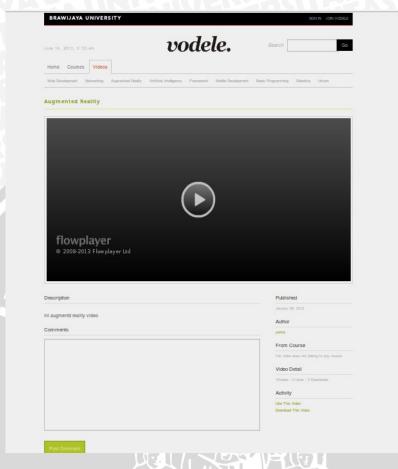


Gambar 5.3 Tampilan Halaman Upload Video

Sumber : [Implementasi]

5.4.4 Implementasi Antarmuka Halaman Video

Implementasi halaman Video adalah halaman antarmuka yang menampilkan video ketika ingin diputar. Sebelum video diputar sistem akan mendeteksi kecepatan user lalu memilih kualitas video yang sesuai dengan kecepatan tersebut.. Berikut gambar 5.4 menunjukkan antarmuka halaman video.



Gambar 5.4 Tampilan Halaman *Video*Sumber: [Implementasi]

BAB VI PENGUJIAN

Pengujian sistem merupakan tahap selanjutnya setelah program atau aplikasi perangkat lunak selesai dalam pembuatannya. Pengujian sistem yang dilakukan meliputi dua tahapan, yaitu pengujian fungsional dan non-fungsional. Pengujian tersebut dilakukan untuk mengevaluasi hasil sistem yang dibuat.

6.1 Pengujian Fungsional

Pengujian fungsional yang digunakan untuk menguji sistem yang telah dibuat terdiri atas metode pengujian *black box* dan pengujian kualitas video berdasarkan *transfer rate* menggunakan *traffic shaping*.

6.1.1 Pengujian Black Box

Pengujian *black box* terfokus pada persyaratan fungsional perangkat lunak. Pengujian ini berusaha menemukan kesalahan antara lain :

- 1. Fungsi-fungsi yang tidak benar atau hilang
- 2. Kesalahan interface
- 3. Kesalahan dalam struktur data
- 4. Kesalahan kinerja

Adapun pengujian-pengujian yang telah dilakukan, antara lain dapat di lihat dalam tabel 6.1 berikut ini

Tabel 6.1 Skenario Uji Coba Sistem

Kelas Uji	Butir Uji	Jenis Pengujian
Login	Login Username & Password	Black Box
	Member / Admin	
Pencarian Data	Pencarian Video	Black Box
	Pencarian Course	Black Box
	Pencarian Member	Black Box
Registrasi	Informasi Registrasi	Black Box
Penambahan Data	Tambah Kategori	Black Box
Pengeditan Data	Edit Kategori	Black Box
	Edit Video	Black Box
	Edit Member / Profile	Black Box

	Edit Course	Black Box
Penghapusan Data	Hapus Kategori	Black Box
	Hapus Video	Black Box
	Hapus Member	Black Box
	Hapus Course	Black Box
Unggah Video	Data Video	Black Box
Pembuatan Course	Data Course	Black Box
Pengujian Pseudostreaming	Timeline Video	Black Box
Deteksi Transfer Rate User	Video	Black Box

Sumber : [Pengujian]

Berikut merupakan hasil pengujian sistem yang telah dilakukan sesuai dengan skenario yang telah direncanakan ditunjukkan pada tabel 6.2

Tabel 6.2 Hasil Pengujian Sistem

ſ	No	Uji Coba Sistem	Hal yang Diharapkan	Kasus Uji Coba	Diterima	Ditolak	Ket.
	1	Login member / admin	Data valid	Username dan password benar		75 CAR	Login berhasil
				Username benar dan password salah		1	Login gagal
				Username salah dan password benar		\	Login gagal
	2	Pencarian data	Data yang dicari ada	Pencarian video	1		Data ditemukan
				Pencarian member	V		Data ditemukan
				Pencarian course	V		Data ditemukan
	3	Registrasi	Data masuk database	Data normal	- T-		Registrasi berhasil
	AB	NAWI RAWI BRA		Username sudah terdaftar sebelumnya		V	Registrasi gagal

4	Tambah kategori	Penambahan kategori	Data normal	1		Penambahan berhasil
AF		berhasil	Data kosong			Penambahan gagal
5	Edit data	Data berubah sesuai dengan	Edit Kategori	1	打具	Data ter <i>update</i>
	BKBF	yang diinginkan	Edit info video	1	TAP	Data ter <i>update</i>
	STILL	umgmkun	Edit profile	V		Data ter <i>update</i>
H	1)-		Edit info	V		Data ter <i>update</i>
6	Hapus data	Dapat menghapus	Hapus kategori		11,	Data terhapus
		data yang diinginkan	Hapus video	V		Data terhapus
	3	₽ \$\$	Hapus member			Data terhapus
		M	Hapus course			Data terhapus
7	Unggah video	Data masuk database	Data normal		3	<i>Upload</i> berhasil
		R FE	Data ada yang kosong	5	7	Upload gagal
8	Pembuatan course	Data masuk database	Data normal	N A)	Pembuatan course berhasil
			Data ada yang kosong		V	Pembuatan course gagal
9	Pseudo streaming	Sistem dapat menampilkan bagian video pada posisi timeline yang dipilih	Posisi timeline yang dipilih yaitu pada detik ke-68			Pseudo streaming berhasil
10	Deteksi transfer rate	Sistem dapat memilih kualitas video yang sesuai dengan transfer rate user	Putar sembarang video	1	HT A	Deteksi transfer rate berhasil

Sumber : [Pengujian]

BRAWIJAYA

Berdasarkan hasil pengujian fungsional dengan kasus uji diatas maka dapat ditarik kesimpulan bahwa pembangunan sistem VoD secara keseluruhan sistem sudah dapat menampilkan hasil yang sesuai dengan yang diharapkan dan proses *random seeking* pada *timeline* video serta pemilihan kualitas video berdasarkan *transfer rate* berhasil dilakukan.

6.1.2 Pengujian Kualitas Video dan Transfer Rate Melalui Traffic Shaping

Untuk menguji kualitas performansi dari sistem layanan VoD yang dibangun maka dilakukanlah pengukuran kualitas layanan secara objektif dengan menggunakan perantara *traffic shaping* Adapun metode pengambilan data samplenya yaitu :

- Setiap video mempunyai 3 kualitas video yang berbeda, yakni 360sd,
 480sd, dan 720hd masing-masing dengan *bitrate* yang berbeda pula.
- Pada saat klien memilih video, flowplayer ditambah dengan program buatan akan mendeteksi kecepatan user, lalu player akan memilih kualitas video yang sesuai dengan kecepatan tersebut berdasarkan *range* yang telah ditentukan.
- Pengujian dilakukan pada jaringan *peer to peer*.
- Pengujian dilakukan dengan batasan transfer rate sebesar 128 kbps, 256 kbps, 512 kbps, 1024 kbps, dan 2048 kbps menggunakan traffic shaping Wondershaper.
- Setiap transfer rate diatas akan diukur sebanyak 4 kali
- Tujuan pengujian yang dilakukan adalah untuk mengetahui bahwa kecepatan yang didapatkan di aplikasi berada dalam batasan transfer rate yang dilakukan oleh traffic shaping dan kualitas video yang diputar sesuai dengan transfer rate yang didapat.

Sebelumnya telah ditentukan pembagian *range* kualitas video yang direkomendasikan oleh sistem. Tabel 6.3 berikut adalah tabel pembagian *transfer rate* terhadap kualitas video.

Tabel 6.3 Pembagian Transfer Rate terhadap Kualitas Video

147	AUP	360p	480p	720p
	Resolusi	640 x 360	854 x 480	1280 x 720
4111	Maximum	1000 kbps	2000 kbps	4000 kbps
Video	Recommended	750 kbps	1000 kbps	2500 kbps
Bitrate	Minimum	400 kbps	500 kbps	1500 kbps

Sumber: [YOU-13]

Untuk aplikasi VOD yang dibangun, maka ditentukan batasan yang jelas untuk *transfer rate* disesuaikan dengan kemampuan sistem. Kecepatan di bawah 500 kbps akan mendapatkan video kualitas 360SD. Kecepatan di antara 500 kbps – 1000 kbps akan mendapat kualitas video 480SD. Kecepatan di atas 1000 kbps akan mendapat kualitas video 720HD.

Berikut pada tabel 6.4 adalah hasil pengujian *transfer rate* dan kualitas video berdasarkan batasan *transfer rate* pada *traffic shaping*.

Tabel 6.4 Pengujian Transfer Rate dan Kualitas Video

Kecepatan	Kecepatan yang Dideteksi & Kualitas Video yang Didapat			
	I		III—	IV
128 kbps	112.17 kbps	111.93 kbps	111.73 kbps	112.94 kbps
	0.11 Mbps	0.11 Mbps	0.11 Mbps	0.11 Mbps
	360 SD	360 SD	360 SD	360 SD
256 kbps	239.19 kbps	238.44 kbps	239.26 kbps	239.13 kbps
	0.23 Mbps	0.23 Mbps	0.23 Mbps	0.23 Mbps
	360 SD	360 SD	360 SD	360 SD
512 kbps	501.90 kbps	498.29 kbps	498.31 kbps	502.77 kbps
	0.49 Mbps	0.49 Mbps	0.49 Mbps	0.49 Mbps
	480 SD	360 SD	360 SD	480 SD
1024 kbps	1071.28 kbps	1069.62 kbps	1060.56 kbps	1059.38 kbps
THE T	1.05 Mbps	1.04 Mbps	1.04 Mbps	1.03 Mbps
	720 HD	720 HD	720 HD	720 HD
2048 kbps	2398.98 kbps	2397.12 kbps	2402.23 kbps	2382.38 kbps
	2.34 Mbps	2.34 Mbps	2.35 Mbps	2.33 Mbps
	720 HD	720 HD	720 HD	720 HD

Sumber : [Pengujian]

BRAWIJAY

Dari hasil pengujian pada tabel 6.4 didapatkan bahwa sistem mampu mendeteksi *transfer rate* sesuai dengan batasan *rate* yang ditentukan oleh *traffic shapping* dan sistem juga dapat menampilkan kualitas video yang sesuai dengan *transfer rate* yang didapat.

6.2 Pengujian Non Fungsional

Untuk pengujian non fungsional dari sistem layanan VoD yang dibangun maka dilakukan pengukuran pada sistem yang sudah mempunyai akses internet atau sudah dihosting. Berikut merupakan skenario ujinya :

- Pengujian dilakukan pada aplikasi VOD yang sudah terhosting ke internet (sistem menggunakan VPS (*Virtual Private Server*)).
- Pengujian dilakukan dengan mengukurnya pada beberapa provider.
 Provider yang digunakan antara lain Tri (3), Axis, Smartfren, dan Indosat.
- Setiap provider melakukan 20 kali pengukuran pada waktu yang berbeda-beda.
- Pengujian dilakukan dengan tujuan untuk menguji keandalan pada sistem yang dihosting ke internet serta kualitas video yang diputar sesuai dengan transfer rate yang didapat.

Berikut hasil pengujian yang dilakukan ditunjukkan pada tabel 6.5 sampai 6.8.

a. Provider Tri (3)

Tabel 6.5 Pengujian pada Provider Tri (3)

No	Jam	Kecepatan yang dideteksi	Kualitas Video
1	12.35	447.47 kbps	360 SD
2	12.41	280.71 kbps	360 SD
3	12.55	298.07 kbps	360 SD
4	13.12	375.74 kbps	360 SD
5	13.20	616.29 kbps	480 SD
6	13.26	502.14 kbps	480 SD
7	13.37	423.76 kbps	360 SD
8	13.43	390.50 kbps	360 SD
9	13.56	342.58 kbps	360 SD

10	14.09	750.34 kbps	480 SD
11	14.20	692.31 kbps	480 SD
12	14.27	627.71 kbps	480 SD
13	14.49	612.15 kbps	480 SD
14	18.24	1037.22 kbps	720 HD
15	18.30	474.69 kbps	360 SD
16	20.03	206.06 kbps	360 SD
17	20.15	202.80 kbps	360 SD
18	20.29	180.98 kbps	360 SD
19	20.35	184.25 kbps	360 SD
20	20.44	167.31 kbps	360 SD

Sumber [Pengujian]

b. Provider Axis

 Tabel 6.6 Pengujian pada Provider Axis

No	Jam	Kecepatan yang dideteksi	Kualitas Video	
1	20.58	429.63 kbps	360 SD	
2	21.12	468.13 kbps	360 SD	
3	21.36	354.00 kbps	360 SD	
4	21.53	656.34 kbps	480 SD	
5	21.59	864.93 kbps	480 SD	
6	22.06	786.92 kbps	480 SD	
7	22.18	481.01 kbps	360 SD	
8	22.24	276.60 kbps	360 SD	
9	22.32	552.69 kbps	480 SD	
10	00.31	796.63 kbps	480 SD	
11	00.42	785.82 kbps	480 SD	
12	00.52	740.75 kbps	480 SD	
13	01.07	219.22 kbps	360 SD	
14	01.12	228.95 kbps	360 SD	
15	01.18	218.44 kbps	360 SD	
16	01.23	174.28 kbps	360 SD	
17	01.32	185.48 kbps	360 SD	
18	01.42	228.55 kbps	360 SD	
19	01.59	237.74 kbps	360 SD	
20	02.05	146.28 kbps	360 SD	
C 1 [D "]				

Sumber: [Pengujian]

c. Provider Smartfren

Tabel 6.7 Pengujian pada Provider Smartfren

No	Jam	Kecepatan yang dideteksi	Kualitas Video
1	07.59	261.58 kbps	360 SD
2	08.06	196.09 kbps	360 SD
3	08.15	123.52 kbps	360 SD
4	08.35	81.51 kbps	360 SD
5	08.39	396.90 kbps	360 SD
6	08.49	520.35 kbps	480 SD
7	08.55	269.23 kbps	360 SD
8	09.05	218.83 kbps	360 SD
9	09.11	290.85 kbps	360 SD
10	09.39	144.20 kbps	360 SD
11	10.01	61.03 kbps	360 SD
12	10.12	118.25 kbps	360 SD
13	10.22	379.05 kbps	360 SD
14	10.48	212.76 kbps	360 SD
15	11.13	139.61 kbps	360 SD
16	11.23	109.25 kbps	360 SD
17	11.27	156.89 kbps	360 SD
18	15.47	92.88 kbps	360 SD
19	16.01	151.48 kbps	360 SD
20	16.10	90.00 kbps	360 SD

Sumber : [Pengujian]

d. Provider Indosat

Tabel 6.8 Pengujian pada Provider Indosat

No	Jam	Kecepatan yang dideteksi	Kualitas Video
1	08.26	1821.62 kbps	720 HD
2	08.38	1385.71 kbps	720 HD
3	08.46	382.26 kbps	360 SD
4	08.56	493.55 kbps	360 SD
5	09.06	646.43 kbps	480 SD
6	09.16	3291.19 kbps	720 HD
7	09.33	167.83 kbps	360 SD
8	09.43	1958.91 kbps	720 HD
9	09.53	806.47 kbps	480 SD
10	10.10	691.93 kbps	480 SD

BRAWIJAYA

11	10.21	118.34 kbps	360 SD
12	10.31	1269.76 kbps	720 HD
13	10.41	88.35 kbps	360 SD
14	10.55	230.84 kbps	360 SD
15	11.03	98.39 kbps	360 SD
16	11.07	260.26 kbps	360 SD
17	11.16	80.42 kbps	360 SD
18	11.24	63.26 kbps	360 SD
19	11.28	207.20 kbps	360 SD
20	11.32	142.37 kbps	360 SD

Sumber: [Pengujian]

Hasil analisis pengujian kinerja aplikasi terhadap transfer rate dan kualitas video pada sistem yang sudah dihosting dapat disimpulkan dalam beberapa poin, yaitu:

- a. Sistem layanan VoD yang dibuat sudah dapat mendeteksi kecepatan transfer rate user dengan cukup baik mengingat kondisi jaringan juga selalu berubahubah (unreliable).
- b. Sistem layanan VoD ini sudah dapat merekomendasikan kepada user kualitas video yang dipakai berdasarkan kecepatan transfer rate user. Untuk transfer rate diatas 1000 kbps akan mendapat video dengan kualitas baik (720 HD). Di antara 500 - 999 kbps akan mendapat video dengan kualitas medium (480 SD), dan dibawah 499 kbps akan mendapatkan video dengan kualitas buruk (360 SD).

BAB VII PENUTUP

7.1 Kesimpulan

Dari hasil pengujian dan pembahasan yang telah dibuat maka dapat ditarik kesimpulan sebagai berikut :

- 1) Cara merancang aplikasi video streaming dengan karakteristik video on demand dengan model pengembangan perangkat lunak Component Based Software Engineering (CBSE) adalah dengan membuat persyaratan sistem, identifikasi komponen, modifikasi komponen, perancangan arsitektur, identifikasi komponen kembali, dan merancang sistem dengan komponen yang dapat dipakai ulang.
- 2) Untuk menerapkan video on demand dengan model pseudostreaming adalah dengan mengetahui mekanisme pengaturan seperti pengecekan metadata, konversi video, dan konfigurasi sistem pseudostreaming dan mengintegrasikannya dengan konsep video on demand. Konsep video on demand semakin terdukung dengan adanya pseudostreaming karena user dapat mencari / seek timeline sesuai kebutuhan
- 3) Pengujian yang dilakukan adalah pengujian fungsional yaitu pengujian *black* box dan pengujian *transfer rate* dan kualitas video dengan *traffic shaping*, serta pengujian non fungsional yaitu pengujian *transfer rate* dan kualitas video pada sistem yang terhubung ke internet.
- 4) Pada pengujian fungsional, sistem sudah dapat menjalankan semua prosedur black box dan pseudostreaming juga berjalan dengan baik. Selain itu, pada pengujian transfer rate dengan traffic shaping, didapatkan bahwa sistem mampu mendeteksi transfer rate sesuai dengan batasan rate yang ditentukan oleh traffic shapping dan sistem juga dapat menampilkan kualitas video yang sesuai dengan transfer rate yang didapat.
- 5) Pada pengujian non fungsional, sistem layanan VoD yang dibuat sudah dapat mendeteksi kecepatan *transfer rate user* dengan cukup baik mengingat kondisi jaringan juga selalu berubah-ubah (*unreliable*), serta sistem sudah dapat

BRAWIIAYA

merekomendasikan kepada *user* kualitas video yang dipakai berdasarkan kecepatan *transfer rate user*.

7.2 Saran

Saran yang dapat menyempurnakan tugas akhir ini adalah :

- 1) VOD pseudostreaming ini dapat dikembangkan pada lingkungan mobile.
- 2) Video yang distreamingkan nantinya tidak hanya berformat flv saja, tetapi dapat juga menstreamingkan video berformat Mp4.
- 3) Dapat men*download* video dengan pilihan ekstensi berbeda seperti, mp3, mp4, avi, dan lain-lain.



BRAWIJAY

DAFTAR PUSTAKA

- [ADO-10] Adobe, 2010. "Adobe Flash Video File Format Specification Version 10.1". California: Adobe System Incorporated.
- [AUS-05] Austerberry, David. 2005. "The Technology of Video and Audio Streaming". USA: Focal Press.
- [BEL-09] Bellard, Fabrice. 2009. "About FFMPEG". (online). (http://ffmpeg.org, diakses tanggal 7 Agustus 2012).
- [CAL-05] California Software Labs. 2002. "Basic Streaming Technology and RTSP Protocol". California: Calsoft Labs.
- [COU-09] Courmont, Remi Denis. 2009. "Video On Demand: RTSP vs HTTP". (online). (http://www.remlab.net/op/vod.shtml, diakses tanggal 9 Februari 2012).
- [FLO-06] Flo, Arnfinn. 2006. "User-Perceived Quality of Service in Video on Demand Services". Thesis: Department of Telematics, Norwegian University of Science and Technology
- [HAN-09] Handojo, Andreas; (dkk.). 2009. "Aplikasi Video Conference dengan Kemampuan Beroperasi pada IPV4 dan IPV6". Yogyakarta: Seminar Nasional Aplikasi Teknologi Informasi (SNATI).
- [JIN-10] Jin, Fang. 2010. "Design and Implementation Video on Demand System based on FMS +FLV". China: Fourth International Conference on Genetic and Evolutionary Computing.
- [KO-96] Ko, Miranda; Irene Koo. 1996. "An Overview of Interactive Video On Demand System". Technical report: The University of British Columbia.

- [KOZ-02] Kozamernik, Franc. 2002. "Media Streaming Over the Internet an Overview of Delivery Technologies". EBU Technical Review.
- [KUR-03] Kurose, James; Ross, Keith. 2003. "Computer Networking: A Top-Down Approach Featuring the Internet". San Francisco: Addison Wesley.
- [NIS-01] Niso Press. 2001. "Understanding Metadata". Bethesda: Niso Press.
- [OPP-11] Oppermann, Ingo. 2011. "Yamdi Yet Another Metadata Injection for FLV". (online). (http://yamdi.sourceforge.net, diakses tanggal 10 Oktober 2012).
- [RAM-11] Rambla. 2011. "State of Play: An Overview of Streaming Protocols".

 (online). (http://www.rambla.be/state-play-overview-streaming-protocols, diakses tanggal 10 Mei 2013).
- [ROM-11] Roman10. 2011. "Video Delivery in HTTP". (online).
 (http://www.roman10.net/video-delivery-in-http/, diakses tanggal 9
 Februari 2012)
- [SEP-09] Septima, Uzma. 2009. "Video Streaming dengan Videolan Project". dalam Percikan Vol. 98 Maret 2009 Hal. 73-84.
- [SOM-11] Sommerville, Ian. 2011. "Software Engineering 9th ed". Boston: Addison-Wesley
- [SYA-11] Syarif, Abdusyi. 2011. "Sistem Multimedia". Pusat Pengembangan Bahan Ajar: Universitas Mercubuana
- [WIJ-11] Wijering, Jeroen. 2011. "What is Video Streaming". (online). (http://www.longtailvideo.com/blog/19578/what-is-video-streaming, diakses tanggal 7 Februari 2012).

[XIA-02] Xiaotian, Xu. 2002. "RTSP-Based Video-On-Demand System". Singapore: School of Electrical and Electronic Engineering, Nanyang Technology University.

[YOU-13] Youtube. 2013. "Live Encoder Settings, Bitrates, and Resolutions". (online).

(https://support.google.com/youtube/answer/2853702?topic=2853713&ctx =topic&hl=en, diakses tanggal 24 Juli 2013).



LAMPIRAN 1: Implementasi Basis Data

Pembuatan basis data

```
CREATE DATABASE 'vodele_db' ;
```

2. Tabel category

```
CREATE TABLE IF NOT EXISTS `category` (
  `category_id` varchar(255) NOT NULL,
  `category_name` varchar(255) NOT NULL,
  `category_img` varchar(255) NOT NULL,
  PRIMARY KEY (`category_id`)
);
```

3. Tabel user

```
CREATE TABLE IF NOT EXISTS `user` (
  `first_name` varchar(255) NOT NULL,
  `last_name` varchar(255) NOT NULL,
  `username` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `birth_date` varchar(255) NOT NULL,
  `gender` varchar(255) NOT NULL,
  `occupation` varchar(255) NOT NULL,
  `join_date` varchar(255) NOT NULL,
  `avatar_url` varchar(255) NOT NULL,
  PRIMARY KEY (`username`)
);
```

4. Tabel user_log

```
CREATE TABLE IF NOT EXISTS `user_log` (
  `username` varchar(255) NOT NULL,
  `last_login` date NOT NULL,
  `total_login` int(255) NOT NULL,
  `channel_view` int(255) NOT NULL,
  `active` int(1) NOT NULL,
  `video_like` text NOT NULL,
  KEY `username` (`username`)
);
```

BRAWIJAYA

BRAWIJAY/

Foreign key untuk tabel user log:

```
ALTER TABLE `user_log`

ADD CONSTRAINT `username` FOREIGN KEY (`username`)

REFERENCES `user` (`username`) ON DELETE CASCADE;
```

5. Tabel login

```
CREATE TABLE IF NOT EXISTS `login` (
  `username` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `level` varchar(255) NOT NULL,
  KEY `usernamefk` (`username`)
);
```

Foreign key untuk tabel login:

```
ALTER TABLE `login`

ADD CONSTRAINT `usernamefk` FOREIGN KEY
(`username`)

REFERENCES `user` (`username`) ON DELETE CASCADE;
```

6. Tabel course

```
CREATE TABLE IF NOT EXISTS `course` (
  `course_id` varchar(255) NOT NULL,
  `course_title` varchar(255) NOT NULL,
  `course_desc` varchar(255) NOT NULL,
  `course_img` varchar(255) NOT NULL,
  `course_view` varchar(255) NOT NULL,
  `course_date` date NOT NULL,
  `username` varchar(255) NOT NULL,
  PRIMARY KEY (`course_id`),
  KEY `fk_username` (`username`)
);
```

BRAWIJAY

Foreign key untuk tabel course:

```
ALTER TABLE `course`

ADD CONSTRAINT `fk_username` FOREIGN KEY
(`username`)
REFERENCES `user` (`username`) ON DELETE CASCADE;
```

7. Tabel video

```
CREATE TABLE IF NOT EXISTS `video` (
  `video id` varchar(255) NOT NULL,
  `video title` varchar(255) NOT NULL,
  `video_desc` varchar(255) NOT NULL,
  `video path` varchar(255) NOT NULL,
  `video time` varchar(255) NOT NULL,
  `course id` varchar(255) NOT NULL,
  `download count` int(255) NOT NULL,
 `view count` int(255) NOT NULL,
  `like count` int(255) NOT NULL,
  `video_img` varchar(255) NOT NULL,
  `video tag` varchar(255) NOT NULL,
  `video date` date NOT NULL,
  `allow comment` int(1) NOT NULL,
  `category id` varchar(255) NOT NULL,
  `username` varchar(255) NOT NULL,
 PRIMARY KEY ('video id'),
 KEY `fkusername` (`username`),
 KEY `course id` (`course id`),
  KEY `category id` (`category_id`)
);
```

Foreign key untuk tabel video:

```
ALTER TABLE `video`
 ADD
        CONSTRAINT
                      `fkusername`
                                     FOREIGN
                                                KEY
(`username`)
REFERENCES `user` (`username`) ON DELETE CASCADE,
 ADD CONSTRAINT
                      `course id`
                                     FOREIGN
                                                KEY
(`course id`)
REFERENCES `course` (`course id`)
                                        ON
                                             DELETE
CASCADE,
      CONSTRAINT `category id`
                                     FOREIGN
                                                KEY
```

```
BRAWIJAYA
```

```
(`category_id`)
REFERENCES `category` (`category_id`) ON DELETE
CASCADE;
```

8. Tabel comment

```
CREATE TABLE IF NOT EXISTS `comment` (
  `comment_id` varchar(255) NOT NULL,
  `comment_content` varchar(255) NOT NULL,
  `video_id` varchar(255) NOT NULL,
  `username` varchar(255) NOT NULL,
  `date` date NOT NULL,
  PRIMARY KEY (`comment_id`),
  KEY `fk_user` (`username`),
  KEY `video_id` (`video_id`)
);
```

Foreign key untuk tabel comment:

```
ALTER TABLE `comment`

ADD CONSTRAINT `fk_user` FOREIGN KEY (`username`)

REFERENCES `user` (`username`) ON DELETE CASCADE,

ADD CONSTRAINT `video_id` FOREIGN KEY
(`video_id`)

REFERENCES `video` (`video_id`) ON DELETE CASCADE;
```

9. Tabel favourite

```
CREATE TABLE IF NOT EXISTS `favourite` (
   `video_id` varchar(255) NOT NULL,
   `username` varchar(255) NOT NULL,
   KEY `fkuser` (`username`),
   KEY `fkvideo_id` (`video_id`)
);
```

BRAWIJAY/

Foreign key untuk tabel favourite:

```
ALTER TABLE `favourite`

ADD CONSTRAINT `fkuser` FOREIGN KEY (`username`)

REFERENCES `user` (`username`) ON DELETE CASCADE,

ADD CONSTRAINT `fkvideo_id` FOREIGN KEY
(`video_id`)

REFERENCES `video` (`video_id`) ON DELETE CASCADE;
```

10. Tabel course video order

```
CREATE TABLE IF NOT EXISTS `course_video_order` (
  `video_id` varchar(255) NOT NULL,
  `username` varchar(255) NOT NULL,
  `order` int(255) NOT NULL,
  KEY `course_id` (`course_id`),
  KEY `fk_videoid` (`video_id`)
);
```

Foreign key untuk tabel course_video_order:

```
ALTER TABLE `course_video_order `
ADD CONSTRAINT `course_id` FOREIGN KEY
(`course_id`)
REFERENCES `user` (`username`) ON DELETE CASCADE,
ADD CONSTRAINT `fk_videoid` FOREIGN KEY
(`video_id`)
REFERENCES `video` (`video_id`) ON DELETE CASCADE;
```

BRAWIJAYA

LAMPIRAN 2: Konfigurasi Apache

Pertama, download modul apache untuk flv dengan mengetikkan:

```
#wget http://people.apache.org/~pquerna/modules/mod flvx.c
```

Kedua, install apache prefork untuk mengcompile modul lalu konfigurasi dengan cara seperti berikut :

```
#apt-get install apache2-prefork-dev
#apxs2 -c -i ./mod_flvx.c
#chmod 644 /usr/lib/apache2/modules/mod_flvx.so
```

Ketiga, buat file /etc/apache2/mods-available/mod_flvx.conf dengan isi sebagai berikut :

```
LoadModule flvx_module
/usr/lib/apache2/modules/mod_flvx.so
AddHandler flv-stream .flv
```

Keempat, buat link simbolik ke / etc/apache2/mods-enable dengan perintah ln.

```
#In -s /etc/apache2/mods-available/mod_flvx.conf
/etc/apache2/mods-enable
```

Kelima, restart apache

```
#/etc/init.d/apache2 restart
```

LAMPIRAN 3: Konfigurasi php.ini

Edit beberapa bagian yang awalnya seperti ini :

```
post_max_size = 8M
upload_max_filesize = 2M
max_execution_time = 30
max_input_time = 60
memory_limit = 8M
```

Ubahlah menjadi seperti ini:

```
post_max_size =750M
upload_max_filesize =750M
max_execution_time =5000
max_input_time =5000
memory_limit =1000M
```

Setelah itu restart apache. Dengan ini maka server dapat mengupload video yang berukuran besar dengan ketentuan angka diatas.

LAMPIRAN 4: Konfigurasi Flowplayer pada script html

```
<script type="text/javascript"</pre>
src="assets/js/flowplayer/flowplayer-3.2.11.min.js"></script>
<script type="text/javascript"</pre>
src="assets/js/flowplayer/flowplayer.controls-
3.2.10.min.js"></script>
<script type="text/javascript"</pre>
src="assets/js/flowplayer/flowplayer.bitrateselect-
3.2.10.js"></script>
<script>
var download_url = "";
flowplayer("player", "assets/js/flowplayer/flowplayer
3.2.14.swf", {
    clip: {
       provider: 'pseudo',
       autoPlay: false,
       urlResolvers: 'bitrateselect'
       scaling: 'fit',
       <?php
       $base video = explode(" ", substr($row[video path], 3,
100));
       $video_path_360sd = $base_video[0] . " " . $base_video[1]
  " 360sd.flv";
       $video path 480sd = $base video[0] . " " . $base video[1]
  " 480sd.flv";
       $video path 720hd = $base video[0] . "_" . $base_video[1]
  " 720hd.flv";
       ?>
       bitrates: [
         { url: "<?php echo $video path 360sd ?>",
                width: 360, bitrate: 300, label: "360 SD" },
         { url: "<?php echo $video path 480sd ?>"
                width: 480, bitrate: 500, label: "480 SD" },
         { url: "<?php echo $video_path_720hd ?>",
                width: 720, bitrate: 1100, label: "720 HD" }
    },
    plugins: {
      bitrateselect: {
            url: "assets/js/flowplayer/flowplayer.bwcheck-
3.2.11.swf",
            checkOnStart: true,
            rememberBitrate: true,
```

BRAWIJAY

```
hdButton: {
              place: 'both',
           netConnectionUrl: "assets/js/flowplayer/flowplayer-
3.2.14.swf",
            onStreamSwitchBegin: function(newItem, currentItem)
                var download url = newItem.streamName;
                $("#download").attr("url", download_url);
            onBwDone: function(info, bitrate) {
             var bw = "Available bandwidth: " + bitrate + "
kbps. Video bitrate: " +
                    info.bitrate + ". " +
                    "Video width: " + info.width;
                    alert(bw);
       pseudo: {
"assets/js/flowplayer/flowplayer.pseudostreaming-3.2.10.swf"
}).bitrateselect('#options').controls("hulu");
</script>
```