



LAPORAN PRAKTEK KERJA LAPANGAN

**PERANCANGAN *PROTOTYPE*
PEMBELAJARAN *ONLINE* DENGAN *VIDEO*
STREAMING BERBASIS *WEB* DENGAN
NODE.JS PADA JURUSAN ILMU KOMPUTER
FMIPA UNUD**

Oleh:

IDA BAGUS RATHU EKA SURYA WIBAWA

NIM : 1308605045

Pembimbing:

I DEWA MADE BAYU ATMAJA DARMAWAN, S.KOM., M.Cs.

Program Studi Teknik Informatika

Jurusan Ilmu Komputer

Fakultas Matematika Dan Ilmu Pengetahuan Alam

Universitas Udayana

2016

HALAMAN PENGESAHAN

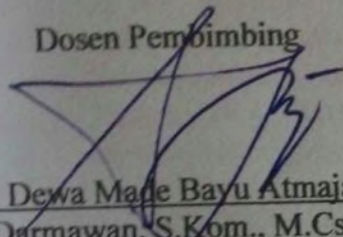
PERANCANGAN *PROTOTYPE* PEMBELAJARAN *ONLINE*
DENGAN *VIDEO STREAMING* BERBASIS *WEB* DENGAN
NODE.JS PADA JURUSAN ILMU KOMPUTER FMIPA UNUD

Oleh:

Ida Bagus Rathu Eka Surya Wibawa
NIM : 1308605045

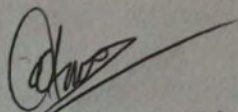
Bukit Jimbaran, 9 Januari 2017
Menyetujui,

Dosen Pembimbing



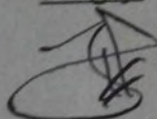
I Dewa Made Bayu Atmaja
Darmawan, S.Kom., M.Cs.
NIP. 19890127 201212 1 001

Pembimbing Lapangan



I Gede Oka Gartria Atitama,
S.Kom., M.Kom
NIP. 19910226 2016031 2 001

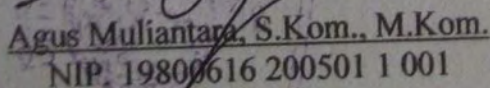
Penguji



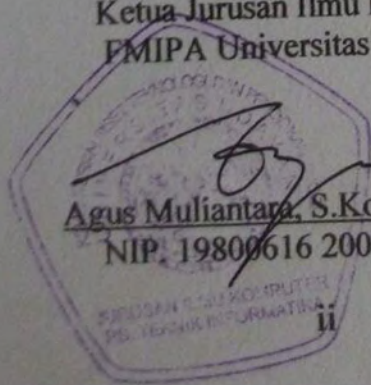
I Wayan Supriana, S.Si., M.Cs
NIP 19840829 2013012 2 002

Mengetahui,

Ketua Jurusan Ilmu Komputer
FMIPA Universitas Udayana



Agus Muliandara, S.Kom., M.Kom.
NIP. 19800616 200501 1 001



KATA PENGANTAR

Puji dan syukur dipanjatkan kehadiran Tuhan Yang Maha Esa karena atas segala berkat dan karunia-Nya sehingga dapat terselesaikannya laporan praktek kerja lapangan (PKL) dengan judul “Perancangan *Prototype* Pembelajaran *Online* dengan *Video Streaming* berbasis *Web* dengan *Node.js* pada Jurusan Ilmu Komputer FMIPA Unud”.

1. Bapak Agus Muliantara, S.Kom., M.Kom. selaku ketua Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Udayana
2. Bapak I Dewa Made Bayu Atmaja Darmawan, S.Kom., M.Cs. selaku pembimbing yang telah memberikan bimbingan, arahan, dan masukan selama penyusunan laporan ini.
3. Semua rekan – rekan Praktek Kerja Lapangan di lingkungan Jurusan Ilmu Komputer yang mendukung dan memberikan saran – saran kepada penulis selama melakukan Praktek Kerja Lapangan
4. Semua pihak yang telah membantu hingga laporan ini dapat terselesaikan.

Disebabkan keterbatasan pengetahuan dan kemampuan yang dimiliki, menyadari laporan ini jauh dari sempurna. Kritik dan saran yang bersifat membangun sangat diharapkan dari pembaca.

Akhir kata terima kasih dan mohon maaf apabila terdapat kesalahan baik yang disengaja maupun tidak disengaja.

Jimbaran, 30 November 2016

Penulis

DAFTAR ISI

	Halaman
HALAMAN PENGESAHAN	ii
KATA PENGANTAR	iii
DAFTAR ISI	iv
DAFTAR TABEL	vi
DAFTAR GAMBAR	vii
DAFTAR LAMPIRAN	viii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan	2
1.3 Manfaat	2
1.3.1 Manfaat Bagi Penulis	2
1.3.2 Manfaat Bagi Instansi PKL	2
1.4 Waktu Dan Tempat Pelaksanaan	2
BAB II GAMBARAN UMUM	4
2.1 Sejarah Jurusan Ilmu Komputer	4
2.2 Kegiatan Jurusan Ilmu Komputer	5
2.3 Struktur Kepengurusan Jurusan Ilmu Komputer	5
2.4 Tujuan dan Fungsi Jurusan Ilmu Komputer	6
2.4.1 Tujuan Jurusan Ilmu Komputer	6
BAB III KAJIAN PUSTAKA	7
3.1 <i>Streaming</i>	7
3.1.1 <i>Video Streaming</i>	7
3.2 Pengembangan Sistem dengan <i>Node.Js</i>	8
3.3 <i>Use Case Diagram</i>	9
3.5 <i>Flowchart</i>	13
3.6 <i>Entity Relationship Diagram</i>	14

3.7	<i>HTML (HyperText Markup Language)</i>	16
3.3.1	Fungsi <i>HTML (HyperText Markup Language)</i>	17
3.8	<i>MySQL</i>	17
BAB IV PEMBAHASAN		19
4.1	Gambaran Umum Sistem Pembelajaran <i>Online</i> dengan <i>Web Streaming</i> di Jurusan Ilmu Komputer FMIPA Unud	19
4.2	Perancangan Sistem.....	19
4.2.1	Analisis Kebutuhan Sistem.....	19
4.2.2	Perancangan Sistem.....	20
4.2.2.1	<i>Use Case Diagram</i>	21
4.2.2.2	<i>Activity Diagram</i>	22
4.2.2.3	Flowchart.....	27
4.2.2.4	Entity Relationship Diagram	28
4.2.3	Implementasi Rancangan Sistem.....	28
4.2.4	Pengujian Sistem	36
BAB V KESIMPULAN DAN SARAN		38
5.1	Kesimpulan.....	38
5.2	Saran.....	38
DAFTAR PUSTAKA.....		39

DAFTAR TABEL

	Halaman
Tabel 3.1 Tabel simbol use case diagram.....	9
Tabel 4.1 Tabel analisis kebutuhan sistem	20
Tabel 4.2 source code handle halaman awal	29
Tabel 4.3 source code pemutar video	29
Tabel 4.4 source code handle login	30
Tabel 4.5 source code koneksi mysql pada node.js	31
Tabel 4.6 source code query login pada node.js	32
Tabel 4.7 source code direct dashboard.....	33
Tabel 4.9 source code redirect menu upload	35
Tabel 4.10 Tabel pengujian fungsionalitas sistem.....	36

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Struktur organisasi Jurusan Ilmu Komputer	5
Gambar 4.1 <i>Use Case diagram sistem</i>	21
Gambar 4.2 <i>Activity diagram autentikasi user</i>	22
Gambar 4.3 <i>Activity diagram upload video</i>	23
Gambar 4.4 <i>Activity diagram pencarian video</i>	24
Gambar 4.5 <i>Activity diagram edit video</i>	25
Gambar 4.6 <i>Activity diagram hapus video</i>	26
Gambar 4.7 <i>Flowchart web streaming</i>	27
Gambar 4.8 <i>Entity relationship diagram</i>	28
Gambar 4.9 Tampilan awal <i>web</i>	30
Gambar 4.10 Tampilan Login Form.....	33
Gambar 4.11 Tampilan dashboard web.....	34
Gambar 4.12 Tampilan <i>form</i> unggah video.....	36

DAFTAR LAMPIRAN

	Halaman
LAMPIRAN	A-1
Lampiran 1. Form aktivitas harian	B-1
Lampiran 2. Dokumentasi Praktek Kerja Lapangan.....	C-1

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan ilmu pengetahuan yang semakin pesat mendorong lahirnya berbagai macam teknologi yang membawa dampak positif bagi kehidupan manusia. Perkembangan teknologi komputer yang begitu pesat membuat informasi yang didapatkan semakin aktual dan cepat. Salah satu permasalahan yang ada adalah teknologi yang digunakan untuk mengirimkan informasi tersebut secara cepat ke banyak orang melalui media komputer. Banyak informasi yang dapat diperoleh dan dikirimkan menggunakan jaringan komputer, antara lain berupa gambar, suara, atau gambar dan suara yang sering disebut *video*. Salah satu metode yang banyak digunakan untuk mengirimkan informasi adalah dengan *video*. Tetapi untuk mengirimkan *video* dalam jaringan tidak semudah mengirimkan gambar saja atau mengirimkan suara saja. Salah satu cara dalam mengirimkan *video* dalam jaringan adalah menggunakan streaming atau yang lebih dikenal dengan nama *video streaming*.

Video streaming merupakan teknologi *multimedia* dalam jaringan memungkinkan pengguna dapat memainkan *video* langsung dengan terkoneksi jaringan tentunya tanpa harus mengunduhnya terlebih dahulu, pada saat ini layanan *video streaming* saat ini sedikit mengangkat mengenai pendidikan/edukasi, meski ada beberapa namun masalah yang dihadapi seperti wadah diskusi mengenai kesulitan pelajaran yang dihadapi.

Pembelajaran *online* dengan *video streaming* dapat menjadi solusi untuk mengatasi masalah yang dihadapi mengenai pelajaran yang mungkin sulit dengan *format video streaming*, pembelajaran secara *visual* dapat meningkatkan minat mahasiswa untuk belajar, *sistem pembelajaran online* dengan *video streaming* berbasis *internet* dibangun dengan menggunakan *NodeJS* sebuah platform yang digunakan untuk membangun aplikasi *serverside* secara *real-time* disebuah jaringan *aplikasi*

1.2 Tujuan

Adapun beberapa tujuan yang ingin dicapai dalam Perancangan *Prototype Pembelajaran Online* dengan *Video Streaming* berbasis *Web* dengan *Node.js* pada Jurusan Ilmu Komputer FMIPA Unud adalah sebagai berikut :

1. Meningkatkan pemahaman materi kuliah yang diajarkan melalui media pembelajaran *visual*.

1.3 Manfaat

Adapun beberapa manfaat yang diperoleh dari Perancangan *Prototype Pembelajaran Online* dengan *Video Streaming* berbasis *Web* dengan *Node.js* pada Jurusan Ilmu Komputer FMIPA Unud adalah sebagai berikut :

1.3.1 Manfaat Bagi Penulis

Beberapa manfaat yang diperoleh penulis dalam melakukan Perancangan *Prototype Pembelajaran Online* dengan *Video Streaming berbasis Web* tersebut dan pelaksanaan praktek kerja lapangan, antara lain :

1. Mengetahui proses untuk melakukan perancangan dan implementasi *web streaming*
2. Menyesuaikan diri dalam menghadapi lingkungan kerja setelah menyelesaikan studi
3. Melihat secara langsung penggunaan / penerapan teknologi dan komunikasi di tempat praktek kerja

1.3.2 Manfaat Bagi Instansi PKL

Beberapa manfaat yang didapatkan bagi instansi dari adanya *Video Streaming* berbasis *Web* dan pelaksanaan praktek kerja lapangan, yaitu sebagai berikut :

1. Menjadi media pembelajaran yang diimplementasikan secara *visual*.
2. Dapat meningkatkan produktivitas pembelajaran melalui melalui *web streaming*.

1.4 Waktu Dan Tempat Pelaksanaan

Pelaksanaan praktek kerja lapangan bertempat di Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam,

Universitas Udayana, di Jalan Kampus Bukit Jimbaran. Dimulai pada tanggal 1 Maret 2016 sampai dengan 20 Mei 2016, yaitu selama dua bulan, 20 hari. Pelaksanaan Praktek Kerja Lapangan yang penulis laksanakan yaitu berlokasi di Laboratorium *Net Centric Computing* Kampus Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Udayana yang beralamat di Bukit Jimbaran, yaitu pada pukul 08.30 wita – 16.00 wita.

BAB II

GAMBARAN UMUM

2.1 Sejarah Jurusan Ilmu Komputer

Ilmu Komputer merupakan ilmu terapan dari ilmu – ilmu dasar yang mengalami perkembangan sangat pesat seiring dengan pesatnya perkembangan Ilmu Pengetahuan dan Teknologi (IPTEK). Penguasaan bidang ilmu komputer belakangan ini sangatlah dirasa perlu dalam meningkatkan sumber daya manusia sebagai tuntutan dari perkembangan teknologi. Khususnya dalam mendukung peningkatan kualitas Tri Dharma Perguruan Tinggi di dalam institusi dan untuk menunjang proses – proses pembangunan masyarakat (daerah dan nasional), bidang ilmu komputer sangat dirasa perlu dikembangkan di Universitas Udayana (Unud).

Gejala meningkatnya kebutuhan terhadap tenaga – tenaga terdidik, terampil, dan profesional di bidang ilmu komputer dan terapannya telah diantisipasi oleh pimpinan Unud sejak tahun 2005. Berawal dari persetujuan Senat Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Udayana (FMIPA Unud) tanggal 13 Agustus 2005 tentang Pembentukan Program Studi Ilmu komputer di Fakultas MIPA Unud yang kemudian dilanjutkan ketingkat Universitas melalui persetujuan Rapat Pimpinan Universitas Udayana tanggal 15 September 2005 yang menyetujui pendirian Jurusan Ilmu Komputer di Fakultas MIPA Unud.

Seiring dengan perjalanan waktu, akhirnya pada tanggal 12 April 2006 dikeluarkanlah Ijin Penyelenggaraan PS Ilmu komputer dari DIRJEN DIKTI dengan Surat Keputusan DIKTI No.1193/D/T/2006 yang berlaku selama 2 tahun terhitung dari tahun pertama akademik, maka Jurusan/PS Ilmu komputer FMIPA Unud secara resmi menyelenggarakan perkuliahan untuk mahasiswa angkatan I (tahun akademik 2006/2007) pada tanggal 3 September 2006 dengan jumlah mahasiswa terdaftar 100 (seratus) orang dari kapasitas sebenarnya yang hanya 50 (lima puluh) orang. Animo masyarakat untuk mendalami bidang ilmu komputer memang sangat tinggi, hal ini dapat dilihat dari banyaknya pendaftar pada angkatan pertama sebanyak 291 orang.

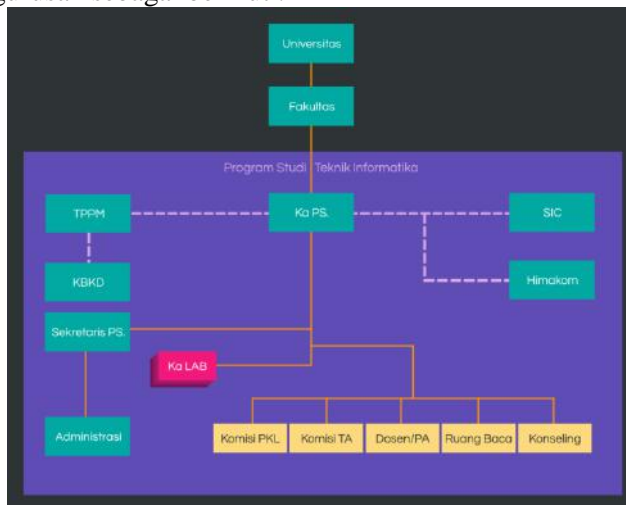
Begitu juga pada tahun ajaran 2007/2008 dimana Jurusan Ilmu Komputer sebagai jurusan baru sudah dapat mensejajarkan diri dengan jurusan - jurusan favorit lainnya dalam penerimaan mahasiswa dengan masuknya Jurusan Ilmu Komputer sebagai salah satu jurusan yang memperoleh mahasiswa sesuai dengan kuota penerimaan sehingga tidak ada bangku kosong.

2.2 Kegiatan Jurusan Ilmu Komputer

Jurusan Ilmu Komputer merupakan salah satu jurusan yang berada di bawah naungan Fakultas Matematika dan Ilmu Pengetahuan Alam, Unviersitas Udayana, yang memiliki beberpaa aktivtias – aktivitas akademik maupun non akademik, yaitu antara lain : belajar mengajar, seminar publikasi ilmiah, pengabdian masyarakat, kegiatan organisasi mahasiswa (Himakom dan SIC), dan lain sebagainya.

2.3 Struktur Kepengurusan Jurusan Ilmu Komputer

Jurusan Ilmu Komputer, FMIPA Unud memiliki struktur kepengurusan sebagai berikut :



Gambar 2.1 Struktur organisasi Jurusan Ilmu Komputer
sumber : (Computer Science Udayana University, 2016)

2.4 Tujuan dan Fungsi Jurusan Ilmu Komputer

Karakteristik Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana saat ini sangat dipengaruhi oleh kekuatan dan kelemahan internal jurusan serta peluang dan ancaman yang terdapat pada eksternal sistem. Berikut adalah tujuan dan fungsi dari Jurusan Ilmu Komputer.

2.4.1 Tujuan Jurusan Ilmu Komputer

Jurusan Ilmu Komputer FMIPA Unud memiliki beberapa tujuan yang ingin dicapai, yaitu sebagai berikut :

1. Menghasilkan lulusan yang berkualitas, mandiri, dan berbudaya serta memiliki wawasan luas dengan penguasaan bidang Ilmu Komputer/Teknik Informatika yang kompeten.
2. Menghasilkan lulusan yang memiliki kemampuan problem solving, kreatif, dan inovatif sehingga mampu berpartisipasi dalam pengembangan riset di bidang Ilmu Komputer/Tenik Informatika.
3. Meningkatkan kualitas dan kuantitas penelitian di bidang Ilmu Komputer/Tenik Informatika dalam mendukung pembangunan nasional.
4. Meningkatkan kuantitas dan kualitas pengabdian masyarakat untuk penyebarluasan perkembangan riset di bidang Ilmu Komputer/Tenik Informatika.

BAB III

KAJIAN PUSTAKA

3.1 *Streaming*

Multimedia streaming menggunakan *media video*, sebagai cara penyampaian konten informasi yang lebih unggul dibandingkan dengan media teks atau suara. Keunggulan dari media *video* bila dibandingkan dengan media teks atau suara yaitu informasi yang disampaikan lebih mudah dimengerti oleh *user*, karena informasi disajikan dalam bentuk *audio visual* (Smaldino, 2007:310). Sehingga *streaming* adalah sebuah teknologi untuk memainkan *file video* atau *audio* secara langsung ataupun dengan *pre-recorder* dari sebuah mesin *server* (*web server*) . Dengan kata lain, *file video* ataupun *audio* yang terletak dalam sebuah server dapat secara langsung pada *browser* saat proses *buffering* mulai berjalan. *File video* atau *audio* di *stream*, akan berbentuk sebuah *buffer* di komputer *client*, dan *data video – audio* tersebut akan mulai di *download* ke dalam *buffer* yang telah terbentuk pada mesin *client*. Dalam waktu sepersekian detik, *buffer* telah terisi penuh dan secara otomatis *file video-audio* langsung dijalankan oleh sistem. *Sistem* akan membaca informasi dari *buffer* dan tetap melakukan proses *download file*, sehingga proses *streaming* tetap berlangsung ke *computer client*.

3.1.1 *Video Streaming*

Video streaming merupakan istilah yang biasa digunakan saat melihat *video diinternet* melalui *browser* dimana pengguna tidak perlu *men-download file video* tersebut untuk dapat memutarinya. Istilah ini tersebut terdiri dari dua suku kata yaitu *video* dan *streaming*, secara istilah *video* berarti teknologi untuk menangkap, merekam, memproses, mentransmisikan dan menata ulang gambar bergerak, sedangkan *streaming* berarti proses penghantaran data dalam aliran berkelanjutan dan tetap yang memungkinkan pengguna mengakses dan menggunakan *file* sebelum *data* dihantar sepenuhnya. Sehingga *video streaming* dapat diartikan *transmisi file video* secara berkelanjutan yang

memungkinkan *video* tersebut diputar tanpa menunggu *file video* tersebut tersampaikan secara keseluruhan. *Video streaming* banyak diimplementasikan pada dunia pertelevisian untuk melakukan siaran dari *website* atau mengirimkan gambar siaran langsung melalui website atau disebut juga *live streaming*. Sehingga gambar yang didapatkan dari siaran langsung, sesegera mungkin ditransmisikan dan dapat diputar melalui *internet*

Pada saat ini teknologi *video streaming* dapat dikembangkan sebagai media pembelajaran *visual*, sehingga proses belajar dan pemahaman suatu materi yang di berikan di dalam kelas dapat di tuangkan ke dalam sebuah bentuk *video* pembelajaran yang dapat di akses melalui website yang dibangun.

3.2 Pengembangan Sistem dengan Node.js

Pengembangan *system* ini menggunakan *Node.js* sebagai *server-side system*. *Node.js* pertama kali diciptakan dan diperkenalkan untuk pengguna pada sistem Linux pada tahun 2009. *Node.js* dikembangkan oleh Ryan Dahl. *node.js* merupakan *platform* perangkat lunak yang berjalan pada *sisi-server* dan aplikasi jaringan. Platform ini terdiri dari 2 hal, yaitu *runtime environment* dan *script library*. *Node.js* memiliki pustaka *server HTTP* sendiri sehingga memungkinkan untuk menjalankan *webserver* tanpa menggunakan program *webserver* seperti *apache* atau *lighttpd*.

Teknologi pada sebagian besar *webserver* seperti *Apache HTTP Server* modelnya menggunakan *thread*. jadi untuk setiap *request* yang di terima maka akan di buat satu *thread* untuk melayaninya. Thread itu sendiri memakai sumber daya dari *sistem*, *Thread* bisa juga berjalan pada satu waktu secara bersamaan. Di aplikasi *realtime thread* harus dibuat hidup untuk mendukung koneksi yang telah dibuat oleh seorang *client*. dan mengakibatkan jumlah pada *thread* yang dapat hidup tadi terbatas tergantung sumber daya sistem yang tersedia. Intinya apabila terjadi jumlah *request* yang banyak melebihi kemampuan *server*, Maka *request* tersebut akan di tahan dulu sampai ada *thread* yang tersedia kembali karena ada *blocking* yang terjadi. Keuntungannya *Node.js* itu

sendiri menggunakan teknik *non-blocking* untuk mempercepat proses


3.3 Use Case Diagram

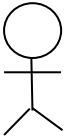




Use case diagram ialah model fungsional sebuah system yang menggunakan actor dan use case. Use case adalah layanan (services) atau fungsi-fungsi yang disediakan oleh system untuk penggunaanya (Henderi et al, 2008).

Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah system. Yang tekankan adalah “apa” yang dibuat system, dan bukan “bagaimana” sebuah use case menerangkan sebuah interaksi antar actor dengan system. Use case merupakan sebuah pekerjaan tertentu, misal :login ke system, meng-create sebuah daftar belanja, dsb. Seorang sebuah actor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan system untuk melakukan pekerjaan-pekerjaan tertentu.

Use Case Diagram dapat sangat membantu bila kita sedang menyusun requitment sebuah system, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua feature yang ada pada system. Sebuah use case yang meng-include dieksekusi secara normal. Sebuah use case dapat di-include oleh lebih dari satu use case lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsional yang common. Sebuah use case juga dapat meng-exted use case lain dengan behavior-nya sendiri. Sementara hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain.




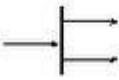
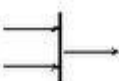

Tabel 3.1 Tabel simbol use case diagram

Simbol	Deskripsi
	Gambaran fungsionalitas dari suatu sistem, sehingga customer atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan

	dibangun.
<p>Aktor / <i>actor</i></p> 	<p>Mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. Actor hanya berinteraksi dengan use case tetapi tidak memiliki kontrol atas use case.</p>
<p>Asosiasi / <i>association</i></p> 	<p>Menghubungkan link antar element.</p>
<p>Ekstensi / <i>extend</i></p> <p><<extend>></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> lainnya, dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan tersebut, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek.</p>
<p>Generalisasi / <i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
<p><i>include</i></p> <p><<include>></p> 	<p>kelakuan yang harus terpenuhi agar sebuah event dapat terjadi, dimana pada kondisi ini sebuah use case adalah bagian dari use case lainnya.</p>

3.4 Activity Diagram

Activity diagram menurut Martin Fowler (2005 : 163) adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja. Dalam beberapa hal, activity diagram memainkan peran mirip diagram alir, tetapi perbedaan prinsip antara notasi diagram alir adalah activity diagram mendukung behavior paralel. Node pada sebuah activity diagram disebut sebagai action, sehingga diagram tersebut menampilkan sebuah activity yang tersusun dari action.

Simbol	Keterangan
	Start Point
	End Point
	Activities
	Fork (Percabangan)
	Join (Penggabungan)
	Decision
Swimlane	Sebuah cara untuk mengelompokkan activity berdasarkan Actor (mengelompokkan activity dalam sebuah urutan yang sama)

Terdapat beberapa hal penting yang harus diketahui, yaitu ;

1. Activity menggambarkan sebuah pekerjaan atau tugas dalam workflow
2. Pada UML, activity digambarkan dengan simbol kotak



3. Start state dengan tegas menunjukkan dimulainya suatu workflow pada sebuah activity diagram.
4. Hanya ada satu start state dalam sebuah workflow
5. Pada UML, start state digambarkan dengan simbol lingkaran yang solid



6. End state menggambarkan akhir atau terminal dari pada sebuah activity diagram

7. Bisa terdapat lebih dari satu end state pada sebuah activity diagram
8. Pada UML, end state digambarkan dengan simbol sebuah bull's eye



9. State transition menunjukkan kegiatan apa berikutnya setelah suatu kegiatan sebelumnya
10. Pada UML, state transition digambarkan oleh sebuah solid line dengan panah

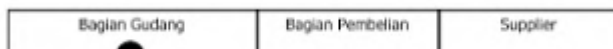


11. Decision adalah suatu titik atau point pada activity diagram yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi
12. Pada UML, decision digambarkan dengan sebuah simbol diamond



Swimlanes

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












1. Mulailah dengan node awal untuk titik awal.
2. Tambahkan partisi jika relevan untuk analisis yang dibuat.
3. Tambahkan aksi untuk setiap langkah utama dari use case.
4. Tambahkan alur dari setiap aksi ke aksi lain, keputusan atau node akhir. Setiap aksi hanya mendapat satu alur masuk dan satu alur keluar menuju ke forks, joins, decisions, dan merges.
5. Tambahkan decisions jika alur dipecah menjadi beberapa pilihan. Jangan lupa untuk menggabungkan kembali dengan merge.
6. Tambahkan forks dan joins jika aktivitas akan dilakukan secara paralel.

3.5 *Flowchart*

Menurut Sutanta (2011:91) “Entity Relationship Diagram (ERD) merupakan suatu model data yang dikembangkan berdasarkan objek.” Entity Relationship Diagram (ERD) digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara logis. Entity Relationship Diagram (ERD) didasarkan pada suatu persepsi bahwa real world terdiri atas obyek-obyek dasar tersebut. Penggunaan Entity Relationship Diagram (ERD) relatif mudah dipahami, bahkan oleh para pengguna yang awam. Bagi perancang atau analis sistem, Entity Relationship Diagram (ERD) berguna untuk memodelkan sistem yang nantinya, basis data akan di kembangkan. Model ini juga membantu perancang atau analis sistem pada saat melakukan analis dan perancangan basis data karena model ini dapat menunjukkan macam data yang dibutuhkan dan kerelasian antardata didalamnya.

Berikut ini adalah beberapa simbol yang digunakan dalam menggambar suatu flowchart :










SIMBOL	NAMA	FUNGSI
	TERMINATOR	Permulaan / akhir program
	GARIS ALIR (FLOW LINE)	Arah aliran program
	PREPARATION	Proses inisialisasi/pemberian harga awal
	PROSES	Proses perhitungan/proses pengolahan data
	INPUT/OUTPUT DATA	Proses input/output data, parameter, informasi
	PREDEFINED PROCESS (SUB PROGRAM)	Permulaan sub program/proses menjalankan sub program
	DECISION	Pertandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	ON PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	OFF PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

3.6 *Entity Relationship Diagram*

Menurut salah satu para ahli, Brady dan Loonam (2010), Entity Relationship Diagram (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh System Analys dalam tahap analisis persyaratan proyek pengembangan system. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain database relasional yang mendasari sistem informasi yang dikembangkan. ERD bersama-sama dengan detail pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk database.

Entity Relationship Diagram (ERD) adalah sebuah konsep yang mendeskripsikan hubungan antara penyimpanan (database) dan didasarkan pada persepsi dari sebuah dunia nyata yang terdiri dari sekumpulan objek yaitu disebut sebagai entity dan hubungan atau relasi antar objek- objek tersebut.

Berikut ini merupakan komponen – komponen pada Entity Relationship Diagram (ERD) :

<u>Notasi</u>	<u>Arti</u>
1. 	1. Entity
2. 	2. Weak Entity
3. 	3. Relationship
4. 	4. Identifying Relationship
5. 	5. Atribut
6. 	6. Atribut Primary Key
7. 	7. Atribut Multivalue
8. 	8. Atribut Composite
9. 	9. Atribut Derivatif

Dalam pembentukan ERD terdapat 3 komponen yang akan dibentuk yaitu :

1. **Entitas** : **Entitas** merupakan mengenai basis data yaitu suatu obyek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalam basis data. Pengertian lainnya menurut Brady dan Loonam (2010), entitas adalah objek yang menarik di bidang organisasi yang dimodelkan. Contoh : Mahasiswa, Kartu Anggota Perpustakaan (KAP), dan Buku.
2. **Hubungan (relasi/relationship)** : Suatu hubungan adalah hubungan antara dua jenis entitas dan direpresentasikan sebagai garis lurus yang menghubungkan dua entitas. Contoh : Mahasiswa mendaftar sebagai anggota perpustakaan (KAP), relasinya adalah mendaftar.

3. **Atribut : Atribut** memberikan informasi lebih rinci tentang jenis entitas. Atribut memiliki struktur internal berupa tipe data. Jenis-jenis atribut :
- a. **Atribut Key** = adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data (Row/Record) dalam tabel secara unik. Dikatakan unik jika pada atribut yang dijadikan key tidak boleh ada baris data dengan nilai yang sama.
Contoh : Nomor pokok mahasiswa (NPM), NIM dan nomor pokok lainnya
 - b. **Atribut simple** = Atribut yang bernilai atomic, tidak dapat dipecah/ dipilah lagi
Contoh : Alamat, penerbit, tahun terbit, judul buku.
 - c. **Atribut Multivalue** = Nilai dari suatu attribute yang mempunyai lebih dari satu (multivalue) nilai dari attribute yang bersangkutan.
Contoh : dari sebuah buku, yaitu terdapat beberapa pengarang.
 - d. **Atribut Composite** = Atribut composite adalah suatu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu yang masih bisah dipecah lagi atau mempunyai sub attribute.
Contoh : dari entitas nama yaitu nama depan, nama tengah, dan nama belakang
 - e. **Atribut Derivatif** = Atribut yang tidak harus disimpan dalam database Ex. Total. atau atribut yang dihasilkan dari atribut lain atau dari suatu relationship. Atribut ini dilambangkan dengan bentuk oval yang bergaris putus-putus

3.7 **HTML (HyperText Markup Language)**

(Menurut Yeni Kustiyahningsih dan Devie Rosa Anamisa, 2011) *HTML* kependekan dari *Hyper Text Markup Language*. Dokumen *HTML* adalah *file text* murni yang dapat dibuat dengan *editor text* sembarang. Dokumen ini dikenal sebagai *web page*. *File-file HTML* ini berisi instruksi-instruksi yang kemudian diterjemahkan oleh *browser* yang ada dikomputer *client (user)*

sehingga isi formasinya dapat ditampilkan secara *visual* dikomputer pengguna (*user*). *HTML* adalah singkatan dari *HyperText Markup Language* yaitu bahasa pemrograman standar yang digunakan untuk membuat sebuah halaman *web*, yang kemudian dapat diakses untuk menampilkan berbagai informasi di dalam sebuah penjelajah *web Internet (Browser)*.

HTML dapat juga digunakan sebagai *link – link* antara *file-file* dalam *situs* atau dalam komputer dengan menggunakan *localhost*, atau *link* yang menghubungkan antar *situs* dalam dunia *internet*. Supaya dapat menghasilkan tampilan wujud yang terintegrasi Pemformatan *hiperteks* sederhana ditulis dalam berkas format *ASCII* sehingga menjadi halaman *web* dengan perintah-perintah *HTML*. *HTML* merupakan sebuah bahasa yang bermula bahasa yang sebelumnya banyak dipakai di dunia percetakan dan penerbitan yang disebut *Standard Generalized Markup Language (SGML)*.

3.3.1 Fungsi *HTML (HyperText Markup Language)*

HTML (HyperText Markup Language) adalah suatu bahasa yang menggunakan tanda-tanda tertentu (*tag*) untuk menyatakan kode-kode yang harus ditafsirkan oleh *browser* agar halaman tersebut dapat ditampilkan secara benar.

Secara umum, fungsi *HTML* adalah untuk mengelola serangkaian data dan informasi sehingga suatu dokumen dapat diakses dan ditampilkan di Internet melalui layanan *web*.

Fungsi *HTML* yang lebih spesifik yaitu :

1. Membuat halaman *web*.
2. Menampilkan berbagai informasi di dalam sebuah *browser Internet*.
3. Membuat *link* menuju halaman *web* lain dengan kode tertentu (*hypertext*).

3.8 *MySQL*

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis.

Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basisdata (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basisdata transaksional maupun operasi basisdata non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Namun demikian pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web (wordpress), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional.

BAB IV PEMBAHASAN

4.1 Gambaran Umum Sistem Pembelajaran *Online* dengan *Web Streaming* di Jurusan Ilmu Komputer FMIPA Unud

Dalam perancangan *web* pembelajaran dengan video agar dapat menjadi solusi untuk mengatasi masalah yang dihadapi mengenai pelajaran yang mungkin sulit dengan *format video streaming*, pembelajaran secara *visual* dapat meningkatkan minat seseorang untuk belajar, *sistem video* edukasi berbasis *internet* dibangun dengan menggunakan *NodeJS* memiliki skalabilitas yang sangat tinggi, berbasis *event (event driven programming)*, dan memiliki konsep *asynchronous* yang sangat bagus sehingga membuatnya ringan dan *efisien*.

4.2 Perancangan Sistem

Perancangan sistem ini menggunakan *framework* yang *javascript* yaitu *node.js*. dipilihnya *node.js* dikarenakan pada *framework* ini memiliki pustaka *server HTTP* sendiri sehingga memungkinkan menjalankan *webserver* tanpa menggunakan program *webserver* seperti *apache*, atau *nginx*.

4.2.1 Analisis Kebutuhan Sistem

Dalam *web* pembelajaran *online* ini dilakukan analisa kebutuhan agar sistem dirancang kerkomputerisasi seperti yang diperlukan. Berikut ini analisa kebutuhan dari sistem *web streaming* untuk pembejaran adalah sebagai berikut :

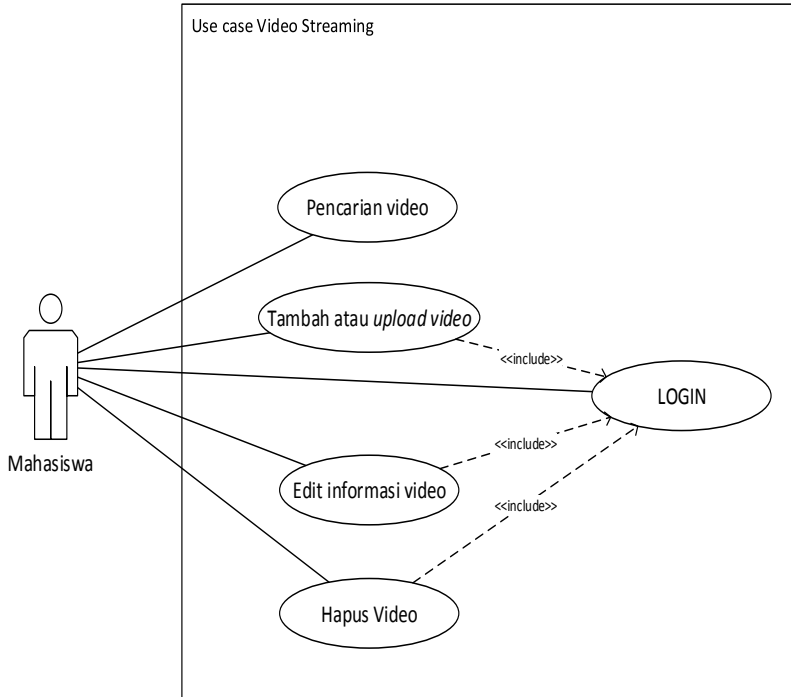
Tabel 4.1 Tabel analisis kebutuhan sistem

No	Pengguna	Kebutuhan	Deskripsi Kebutuhan
1	<i>User</i>	<i>User</i> dapat melihat <i>channel video</i> yang dikelola pada <i>dashboad</i>	<i>User</i> mengunggah <i>video</i> tutorial, dan menentukan <i>channel</i> yang di berikan. Tidak ada batasan untuk nama <i>channel</i> yang dimasukkan oleh <i>user</i> , sehingga dengan mudah <i>user</i> mengetahui <i>video</i> tersebut dan dikelola dengan baik.
2	<i>User</i>	Mengunggah <i>video</i> pembelajaran	<i>User</i> mengunggah <i>video</i> pada <i>form</i> yang telah disediakan, memberi judul, <i>channel</i> , <i>diskripsi</i> singkat <i>video</i> yang diunggah, sehingga setelah mengunggah <i>video</i> dapat dikelola dengan mencari <i>channel</i> yang telah di pilih.
3	<i>User</i>	<i>Maintenance video</i> yang <i>diupload</i> , mengubah <i>channel</i> , mengganti, dan menghapus	Melakukan <i>maintenance video</i> yang diunggah mahasiswa, yaitu sebagai berikut : <ol style="list-style-type: none"> 1. Memperbaharui <i>channel</i> 2. Menghapus <i>video</i> 3. Pencarian <i>video</i>

4.2.2 Perancangan Sistem

Perancangan sistem dilakukan berdasarkan hasil analisa kebutuhan sistem yang sudah dilakukan sebelumnya. Hal ini dilakukan agar, perancangan tidak keluar dari tujuan sistem yang dikembangkan.

4.2.2.1 Use Case Diagram

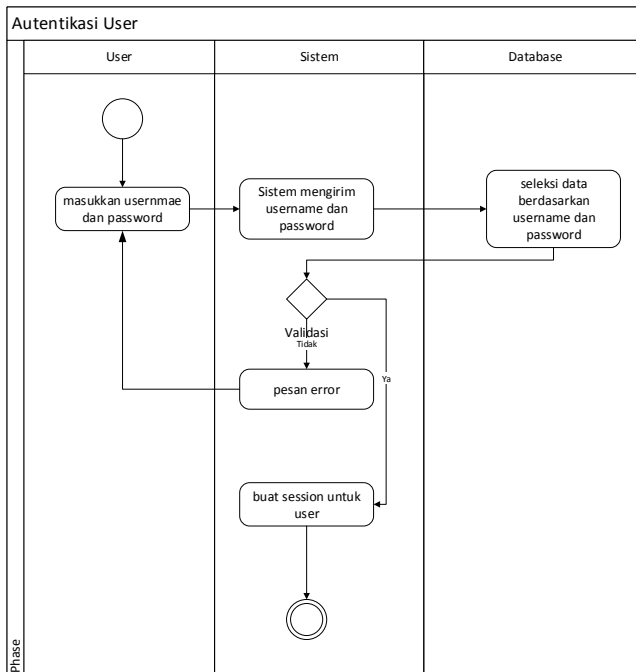


Gambar 4.1 Use Case diagram sistem

Pada gambar 4.1 menunjukkan *use case diagram* dari sistem. Diagram tersebut menunjukkan interaksi aktor mahasiswa dengan fungsionalitas sistem. Beberapa fungsi dapat dilakukan hanya jika aktor telah melakukan *login* terlebih dahulu.

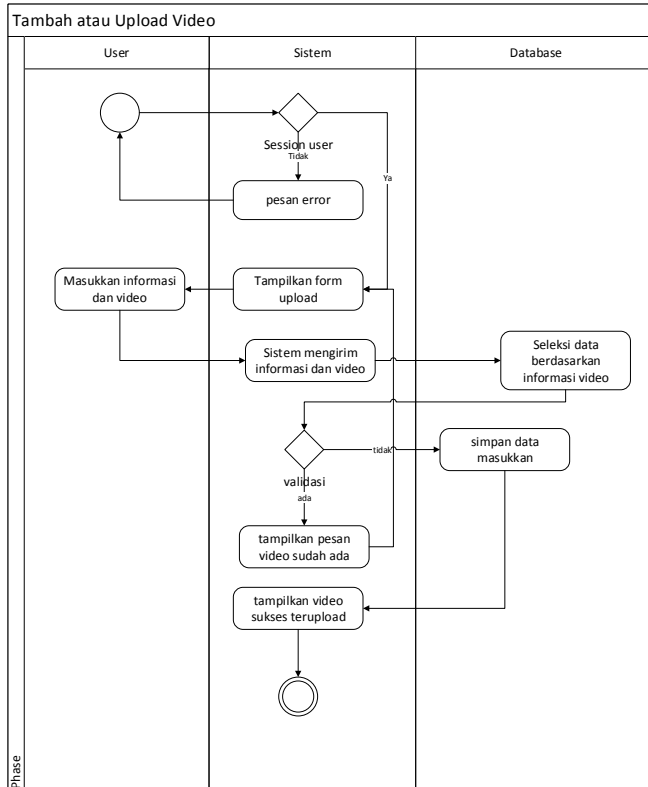
4.2.2.2 Activity Diagram

a) Autentikasi user



Gambar 4.2 Activity diagram autentikasi user

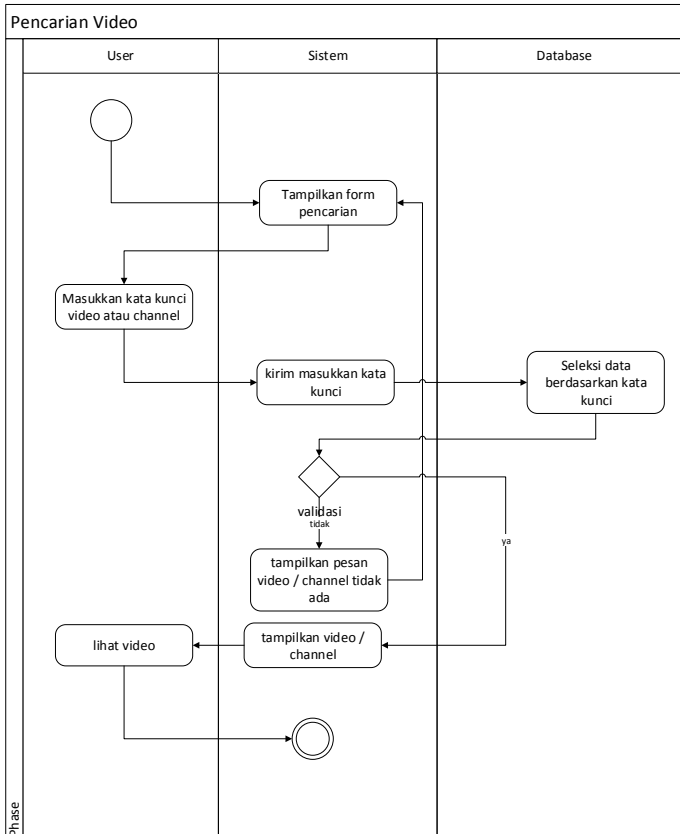
Pada gambar 4.2 menggambarkan aktivitas pada saat *autentikasi user* ketika login kedalam *sistem*. *User* memasukkan *username* dan *password* selanjutnya *sistem* akan mengirim masukan *user* untuk *diautentikasi* di dalam *databse*. Bila *data* yang dimasukkan *user valid* maka *user* dapat masuk kedalam *sistem*, dan bila jika tidak akan kembali ke *form login*.

b) Tambah atau *upload Video*

Gambar 4.3 Activity diagram upload video

Pada gambar 4.3 merupakan aktivitas untuk menambah/mengunggah *video* dengan memasukkna informasi pada *video* kemudian pada *sistem* akan *memvalidasi* informasi mengenai *video* yang diunggah oleh *user*, bila judul *video* sudah diunggah sebelumnya *sistem* akan memeberikan *video* sudah ada dengan naa yang digunakan, dan jika informasi *video* belum tercatat pda *database* maka *video* dapat diunggah.

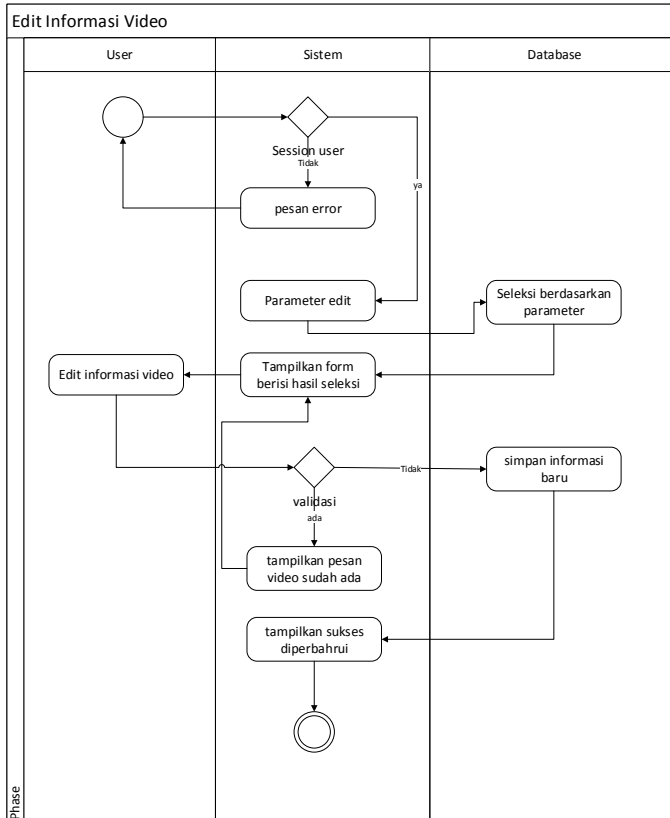
c) Pencarian video oleh user



Gambar 4.4 Activity diagram pencarian video

Pada gambar 4.4 merupakan aktivitas pencarian *video* oleh *user*. dengan memasukkan kata kunci seperti judul atau *channel video*, lalu sistem akan mencocokkan kata kunci dengan *database*, jika ditemukan ada pada *database* sistem akan menampilkan beberapa video yang dimaksud atau *channel* yang diinginkan akan ditampilkan.

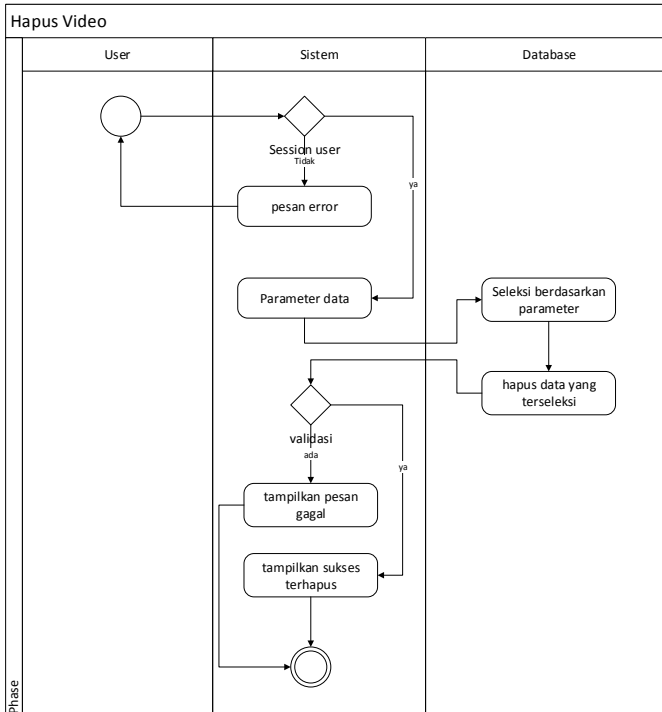
d) Edit informasi video oleh user



Gambar 4.5 Activity diagram edit video

Pada gambar 4.5 merupakan aktivitas diagram untuk melakukan perubahan pada informasi *video* yang telah diunggah. Dengan memilih *video* yang ingin dirubah informasinya. Bila perubahan informasi oleh user sudah *valid* maka perubahan tersebut akan disimpan pada *database*.

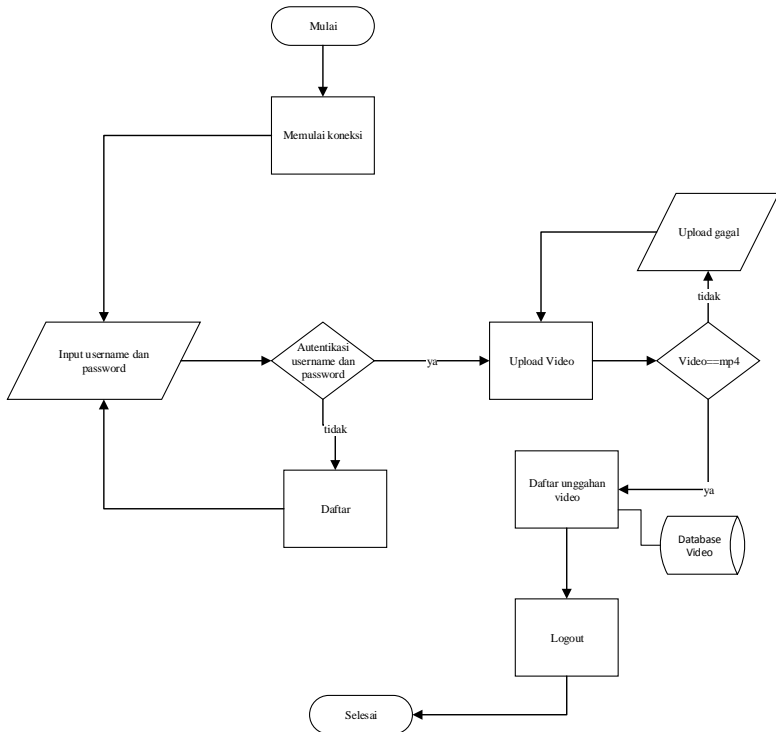
e) Hapus Video



Gambar 4.6 Activity diagram hapus video

Pada gambar 4.6 merupakan aktivitas diagram untuk melakukan penghapusan *video* yang ingin dihapus oleh *user*.

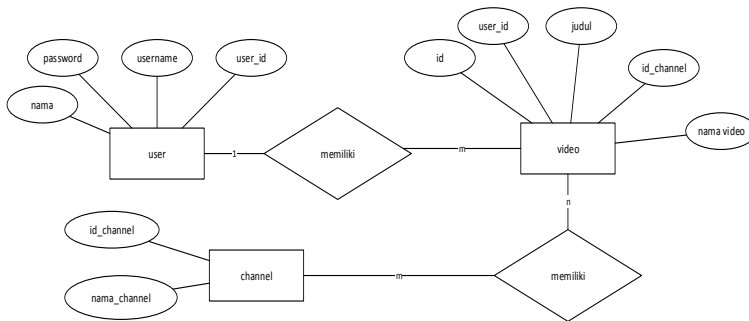
4.2.2.3 Flowchart



Gambar 4.7 *Flowchart web streaming*

Pada gambar 4.7 menggambarkan merupakan *flowchart* dari jalannya sistem pada untuk dapat melakukan *upload file* pertama harus melakukan *login username* dan *password* bila *username* dan *password* benar maka akan masuk ke *dashboard* dan dapat melakukan unggah *video* selanjutnya ketika selesai unggah kita dapat menyaksikan *video* yang sudah diunggah oleh user.

4.2.2.4 Entity Relationship Diagram



Gambar 4.8 *Entity relationship diagram*

Pada gambar 4.8 merupakan *Entity Relationship Diagram* *user* memiliki *username password* dan *nama*, dan satu *user* dapat mengunggagah banyak *video* dengan menyimpan *user_id* dari *user* sehingga ketika seleksi untuk menampilkan *video* akan menjadi lebih *efisien*.

4.2.3 Implementasi Rancangan Sistem

Implementasi perancangan *sistem* Perancangan *Prototype* Pembelajaran *Online* Dengan *Video Streaming* Berbasis Web Dengan *Node.js* Pada Jurusan Ilmu Komputer FMIPA Unud. Pada implementasi ini masih berupa *prototype* sehingga beberapa fitur belum terselesaikan dengan baik.

a) Tampilan Awal *Website*.

Untuk menampilkan tampilan awal pada web pada file index.js buat fungsi untuk handle halaman awal :

Tabel 4.2 source code handle halaman awal

```
router.get('/', function(req, res, next) {
  res.render('index', { title: 'MyVilear' });
  var fs = require('fs');
  fs.realpath(__dirname, function(err, path) {
    if (err) {
      console.log(err);
      return;
    }
    console.log('Path is : ' + path);
  });
  fs.readdir(__dirname, function(err, files) {
    if (err) return;
    files.forEach(function(f) {
      console.log('Files: ' + f);
    });
  });
});
```

Pada source code 4.2 merupakan source yang dibuat pada file index.js untuk menampilkan semua video yang ada dalam database. Pada fungsi readdir digunakan untuk mencari alamat video yang berada pada path: public/video, sehingga dapat menampilkan video.

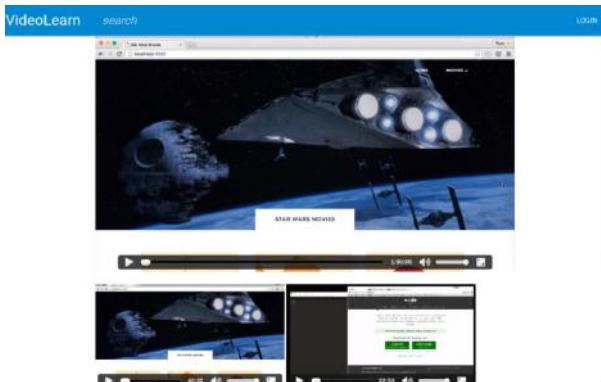
Selanjutnya pada tampilan HTMLnya untuk memutar video menggunakan fitur pemutar video yang telah disediakan oleh HTML5 sebagai berikut :

Tabel 4.3 source code pemutar video

```
<div class=" row container">
  <video      class="responsive-video"      width="1500px"
```

```
height="200px" controls>
  <source src="/video/video1.mp4" type="video/mp4">
</video>
</div>
```

Pada source code 4.3 digunakan untuk memutar video yang diunggah pada source code diatas masih memangging video secara manual sesuai alamat video, dikarenakan pada saat memanggil video secara otomatis belum berjalan dengan baik. Untuk tampilan halaman utama dapat dilihat pada gambar 4.9.



Gambar 4.9 Tampilan awal web

Pada gambar 4.9 gambaran semua *video* yang sudah *diupload* semua *user* yang telah terdaftar menjadi mahasiswa Unud. Sedangkan yang tidak memiliki user dapat menyaksikan dan mencari *video* yang diinginkan namun tidak dapat mengunggah *videonya*.

b) Tampilan Login

Pada tampilan login kita buat jalur yang akan mengarahkan pengguna ke halaman login pada `index.js`

Tabel 4.4 source code handle login

```
router.get('/login', function(req, res){
```

```
res.render('login');
});
```

Pada source code 4.4 digunakan untuk mendirect pengguna ketika memilih tombol login akan otomatis dibawa ke halaman login, karena hanya menampilkan login jadi menggunakan mode get saja.

Untuk mengeksekusi proses login pertama kita buat sebuah koneksi yang menghubungkan node.js pada mysql :

Tabel 4.5 source code koneksi mysql pada node.js

```
var connection = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: 'qwerty',
  database: 'belajar_db'
});

connection.connect(function (error) {
  // body...
  if (!error) {
    console.log('Error');
  } else {
    console.log('Connecting')
  }
});
```

Pada source code 4.5 digunakan untuk membangun koneksi ke mysql. Kita buat sebuah variable bernama connection untuk membuka koneksi ke database mysql. Host merupakan alamat database yang digunakan karena kita buat pada satu server yang sama digunakan localhost, masukkan user dan password dari mysql, lalu pilih database yang digunakan untuk mengelola username dan video yang dikelola oleh user.

Selanjutnya buat fungsi kembali untuk mengelola request user untuk dapat melakukan login

Tabel 4.6 source code query login pada node.js

```
router.post('/login', function(req, res){
  var Array = JSON.parse(JSON.stringify(req.body));
  console.log(Array);

  var username = req.body.username;
  var pass = req.body.password;
  var data = {
    "Data":""
  };
  connection.query("SELECT * from user WHERE username=?
and password=?", [username, pass], function(err, rows, fields){

    if(rows.length != 0){

      data["Data"] = "Successfully logged in..";
      res.json(data);
    }else{
      res.render('login');
      data["Data"] = "username or password is incorrect.";
      res.json(data);
      //res.render('login');
    }
  });
});
```

Pada source code 4.6 digunakan untuk mengambil informasi berupa username dan password yang dikirimkan oleh user dalam bentuk json. Kita buat parsing data yang diperlukan untuk mengambil username dan password. Selanjutnya kita lakukan pengujian jika username ditemukan dan tidak sama dengan 0 (nol) maka login berhasil dan jika tidak maka login akan gagal.

Gambar 4.10 Tampilan Login Form

Pada gambar 4.10 *form login* untuk dapat masuk ke dashboard dengan memasukkan *username* dan *password* yang di dapat dari kampus Ilmu Komputer.

c) Tampilan *Dashboard*

Pada tampilan dashboard sama seperti tampilan sebelumnya kita buat pada `index.js` yang berada pada folder `router` tambahkan kembali source code :

Tabel 4.7 source code direct dashboard

```
router.get('/dashboard', function(req, res){
    res.render('dashboard');
});
```

Pada source code 4.7 digunakan untuk menghandle setelah login berhasil `index.js` akan mendirect ke tampilan dashboard sehingga user dapat mengelola video yang akan diunggah serta melihat keperluan channel yang dimiliki.

Tabel 4.8 source code menampilkan channel

```
<div class="row">
  <div class="col s6">
    <div style="padding: 70px;" align="center" class="card">
      <div class="row">
        <div class="left card-title">
```

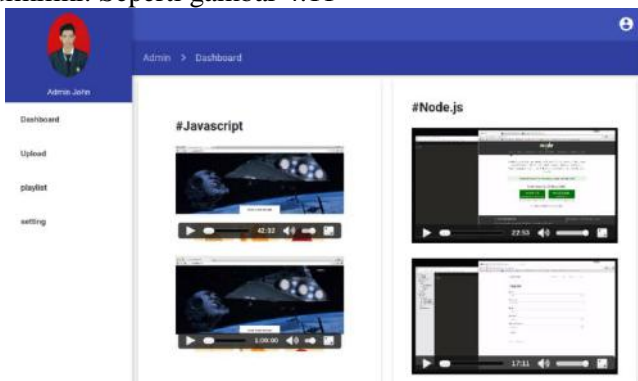
```

        <b>#Javascript</b>
    </div>
</div>

    <div class="row">
    <video class="responsive-video" width="400px"
height="200px" controls>
    <source src="/video/video2.mp4" type="video/mp4">
    </video>
    <div class="col s1">&nbsp;</div>
    <div class="col s1">&nbsp;</div>
    <video class="responsive-video" width="400px"
height="200px" controls>
    <source src="/video/video1.mp4" type="video/mp4">
    </video>
    </div>
</div>
</div>

```

Pada source code 4.8 digunakan untuk menampilkan setiap channel yang kita buat lalu menampilkan video sesuai list channel yang dimiliki. Seperti gambar 4.11



Gambar 4.11 Tampilan dashboard web

Pada gambar 4.11 merupakan tampilan *dashboard* berisi *file video* yang telah *diupload* dan dikategorikan sesuai *channel*.

d) Tampilan *Upload*

Pada tampilan upoad untuk dapat mengarahkan tombol menu upload tambahkan method get pada index.js

Tabel 4.9 source code redirect menu upload

```
router.get('/upload', function(req, res){
    res.render('upload');
});

router.post('/upload', function(req, res){
    //res.render('upload');

    var file = + "/video" + req.files.file.name;
    console.log(file);

});
```

Pada source code tabel 4.9 digunakan untuk mengarahkan menu upload ke form yang digunakan oleh pengguna untuk mengunggah video pembelajaran yang dibuat. Untuk tampilan antarmuka dapat dilihat pada gambar 4.12.

Gambar 4.12 Tampilan *form* unggah video

Pada gambar 4.12 merupakan tempat dimana *user* dapat mengunggah *video*, dengan menambah judul dan *channel* yang diinginkan.

4.2.4 Pengujian Sistem

Pada pengujian *sistem*, digunakan jenis pengujian fungsionalitas *sistem*. Pada pengujian fungsionalitas *sistem*, dilakukan pengujian untuk memastikan bahwa setiap fungsionalitas sistem utama sudah berjalan dengan baik. Berikut adalah tabel pengujian fungsionalitas sistem, yaitu sebagai berikut.

Tabel 4.10 Tabel pengujian fungsionalitas sistem

N o	Pengguna	Fungsional	Hasil Tes	Keterangan
1	User	Menampilkan semua <i>video</i>	Berhasil	Mampu untuk menampilkan dan memutar <i>video</i>
2	User	Kategori sesuai <i>channel</i>	Belum Berhasil	Belum berjalan
3	User	Mengunggah <i>video</i> dan dikategorikan sesuai	Belum Berhasil	Belum berjalan

		<i>channel</i> yang diinginkan		
4	<i>User</i>	Parsing data <i>video</i> melalui <i>mysql</i> ke <i>JSON</i>	Belum Berhasil	Belum berjalan

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil kegiatan yang dilakukan, dapat disimpulkan beberapa hal terkait dengan perancangan *prototype* Sistem Pembelajaran *Online* dengan *Web Streaming* di Jurusan Ilmu Komputer FMIPA Unud, adalah dengan dibangunnya *sistem* pembelajaran *online* melalui *video visual user* dapat lebih mudah mengerti gambaran materi – materi yang belum dipahami dalam kelas. Namun lebih ditekankan pada *parsing* database ke *JSON* sehingga dengan cepat mengambil informasi dari *video* yang dicari tersebut.

5.2 Saran

Adapun saran yang dapat disampaikan agar *sistem* ini dapat dikembangkan dengan baik penggunaan *database* yang sesuai dengan *framework node.js*, karna masih *static*, dapat dikembangkan secara *dinamis* dan dapat menambahkan *server red5* yang dapat mengatur resolusi *video* yang akan dilihat oleh pengguna (*user*) serta dalam proses parsing database ke json dengan node.js dapat menggunakan *mongodb*, dikarenakan untuk parsing dari *mysql* ke json dengan node.js masih mengalami masalah.

DAFTAR PUSTAKA

- Computer Science Udayana University. (2016, April 1). *Struktur Organisasi*. Diambil kembali dari Jurusan Ilmu Komputer FMIPA Universitas Udayana:
<http://www.cs.unud.ac.id/id/Struktur-Organisasi>
- Dirga, D. (2011, 09). *Tentang Activity Diagram*. Diambil kembali dari <http://dewadirga.blogspot.co.id/>:
<http://dewadirga.blogspot.co.id/2011/09/tentang-activity-diagram.html>
- Kampus, A. (2010, 10 1). *Teknologi Streaming*. Diambil kembali dari arsipkampus.files.wordpress.com:
<http://arsipkampus.files.wordpress.com/2010/10/teknologi-streaming.pdf>
- node.js. (t.thn.). *about node.js*. Diambil kembali dari <https://nodejs.org>: <https://nodejs.org>
- Sutanta, E. (2011). *Basis Data Dalam Tinjauan Konseptual*. Yogyakarta : Andi Offset.
- techno-evolutioner. (2016). *Belajar Javascript dengan Node Js*. Diambil kembali dari techno-evolutioner.blogspot.co.id:
<https://techno-evolutioner.blogspot.co.id/2016/10/belajar-javascript-dengan-nodejs.html>
- Universitas Brawijaya. (2011, 07). *Video Streaming.doc*. Diambil kembali dari blog.ub.ac.id:
<http://blog.ub.ac.id/yafemo/files/2011/07/VIDEO-STREAMING1.doc>

LAMPIRAN

Lampiran 1. Form aktivitas harian

AKTIVITAS HARIAN PKL

Nama : Ida Bagus Rathu Eka Surya Wibawa
 NIM : 1308605045
 Lokasi PKL : Lab. Jaringan, ILKOM, Univ. Udayana
 Waktu Pelaksanaan : 05 September 2016 - 25 November 2016

No.	Nama Penanggung Jawab/Jabatan	Pelaksanaan PKL			Keterangan
		Tanggal	Lokasi	Aktivitas	
1	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	05-09-2016	Lab NCC ILKOM Unud	Bimbingan ke pembimbing PKL untuk tugas PKL	
2	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	06-09-2016	Lab NCC ILKOM Unud	Pembagian tugas PKL dan pencarian tutorial terkait tugas yang diberikan	
3	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	07-09-2016	Lab NCC ILKOM Unud	Mempelajari topologi jaringan Ilmu Komputer	
4	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	08-09-2016	Lab NCC ILKOM Unud	Mempelajari topologi jaringan Ilmu Komputer	
5	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	09-09-2016	Lab NCC ILKOM Unud	Mempelajari topologi jaringan Ilmu Komputer	
6	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	10-09-2016	Lab NCC ILKOM Unud		
7	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	11-09-2016			

8	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	12-09-2016			
9	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	13-09-2016	Lab NCC ILKOM Unud	Mempelajari tutorial instalasi server VPN	
10	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	14-09-2016	Lab NCC ILKOM Unud	Mempelajari tutorial instalasi server VPN	
11	I Gede Oka Gatria Atitama, S.Kom., M.Kom.	15-09-2016	Lab NCC ILKOM Unud	Membuat server ujicoba VPN	
12	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	16-09-2016	Lab NCC ILKOM Unud	Membuat server ujicoba VPN	
13	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	17-09-2016			
14	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	18-09-2016			
15	I Gede Oka Gatria Atitama, S.Kom., M.Kom.	19-09-2016	Lab NCC ILKOM Unud	Membuat server ujicoba VPN	
16	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	20-09-2016	Lab NCC ILKOM Unud	Dokumentasi server VPN	
17	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	21-09-2016	Lab SI ILKOM Unud	Mempelajari materi untuk instalasi jaringan dan thin client di lab sistem informasi	
18	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	22-09-2016	Lab SI ILKOM Unud	Mempelajari materi untuk instalasi jaringan dan thin client di lab sistem informasi	

19	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	23-09-2016	Lab SI ILKOM Unud	Membeli alat-alat yang dibutuhkan untuk pembangunan lab sistem informasi	
20	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	24-09-2016			
21	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	25-09-2016			
22	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	26-09-2016	Lab SI ILKOM Unud	Instalasi jaringan dan thin client di lab sistem informasi	
23	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	27-09-2016	Lab SI ILKOM Unud	Instalasi jaringan dan thin client di lab sistem informasi	
24	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	28-09-2016	Lab SI ILKOM Unud	Instalasi jaringan dan thin client di lab sistem informasi	
25	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	29-09-2016	Lab SI ILKOM Unud	Instalasi jaringan dan thin client di lab sistem informasi	
26	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	30-09-2016	Lab SI ILKOM Unud	Instalasi jaringan dan thin client di lab sistem informasi	
27	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	01-10-2016			
28	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	02-10-2016			
29	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	03-10-2016	Lab SI ILKOM Unud	Instalasi jaringan dan thin client di lab sistem informasi	

30	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	04-10-2016	Lab SI ILKOM Unud	Instalasi jaringan dan thin client di lab sistem informasi	
31	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	05-10-2016	Lab SI ILKOM Unud	Pengujian jaringan dan thin client di lab sistem informasi	
32	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	06-10-2016	Lab SI ILKOM Unud	Pengujian jaringan dan thin client di lab sistem informasi	
33	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	07-10-2016	Lab SI ILKOM Unud	Pengujian jaringan dan thin client di lab sistem informasi	
34	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	08-10-2016			
35	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	09-10-2016			
36	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	10-10-2016	Lab SI ILKOM Unud	Dokumentasi instalasi jaringan dan thin client di lab sistem informasi	
37	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	11-10-2016	Lantai 2 Gedung BF ILKOM Unud	Survei pemansangan jaringan di lantai 2 Gedung BF	
38	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	12-10-2016	Lantai 2 Gedung BF ILKOM Unud	Instalasi jaringan di lantai 2 Gedung BF	
39	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	13-10-2016	Lantai 2 Gedung BF ILKOM Unud	Instalasi jaringan di lantai 2 Gedung BF	
40	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	14-10-2016	Lantai 2 Gedung BF ILKOM Unud	Instalasi jaringan di lantai 2 Gedung BF	

41	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	15-10-2016			
42	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	16-10-2016			
43	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	17-10-2016	Lantai 2 Gedung BF ILKOM Unud	Instalasi jaringan di Perpustakaan	
44	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	18-10-2016	Lantai 2 Gedung BF ILKOM Unud	Instalasi jaringan di Perpustakaan	
45	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	19-10-2016	Lantai 2 Gedung BF ILKOM Unud	Pengujian jaringan di Gedung Bf	
46	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	20-10-2016	Lantai 2 Gedung BF ILKOM Unud	Dokumentasi jaringan di Gedung Bf	
47	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	21-10-2016	Gedung BC ILKOM Unud	Survei pemasangan jaringan WiFi di Gedung BC	
48	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	22-10-2016			
49	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	23-10-2016			
50	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	24-10-2016	Gedung BC ILKOM Unud	Instalasi jaringan WiFi di Gedung BC	
51	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	25-10-2016	Gedung BC ILKOM Unud	Instalasi jaringan WiFi di Gedung BC	
52	I Gede Oka Gatria Atitama, S.Kom., M.Kom.	26-10-2016	Gedung BC ILKOM Unud	Dokumentasi jaringan di Gedung Bc	

53	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	27-10-2016	Lab NCC ILKOM Unud	Merapikan lab NCC	
54	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	28-10-2016	Lab NCC ILKOM Unud	Merapikan lab NCC	
55	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	29-10-2016			
56	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	30-10-2016			
57	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	31-10-2016	Lab NCC ILKOM Unud	Merapikan lab jaringan jaringan	
58	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	01-11-2016	Gedung BD ILKOM Unud	Survei pemasangan jaringan di lantai 2 Gedung BD	
59	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	02-11-2016	Gedung BD ILKOM Unud	Instalasi jaringan WiFi di Gedung Bd	
60	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	03-11-2016			
61	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	04-11-2016			
62	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	05-11-2016			
63	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	06-11-2016	Gedung BD ILKOM Unud	Instalasi jaringan WiFi di Gedung Bd	

64	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	07-11-2016	Gedung BD ILKOM Unud	Instalasi jaringan WiFi di Gedung Bd	
65	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	08-11-2016			
66	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	09-11-2016			
67	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	10-11-2016			
68	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	11-11-2016	Gedung BD ILKOM Unud	Instalasi jaringan WiFi di Gedung Bd	
69	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	12-11-2016	Gedung BD ILKOM Unud	Instalasi jaringan WiFi di Gedung Bd	
70	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	13-11-2016	Gedung BD ILKOM Unud	Instalasi jaringan WiFi di Gedung Bd	
71	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	14-11-2016	Gedung BD ILKOM Unud	Instalasi jaringan WiFi di Gedung Bd	
72	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	15-11-2016			
73	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	16-11-2016			
74	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	17-11-2016	Gedung BD ILKOM Unud	Pembuatan infrastuktur jaringan lan dan instalasi jaringan untuk lab di lantai 2 gedung bd	

75	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	18-11-2016	Gedung BD ILKOM Unud	Pembuatan infrastuktur jaringan lan dan instalasi jaringan untuk lab di lantai 2 gedung bd	
76	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	19-11-2016	Gedung BD ILKOM Unud	Pembuatan infrastuktur jaringan lan dan instalasi jaringan untuk lab di lantai 2 gedung bd	
77	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	20-11-2016	Gedung BD ILKOM Unud	Pembuatan infrastuktur jaringan lan dan instalasi jaringan untuk lab di lantai 2 gedung bd	
78	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	21-11-2016	Gedung BD ILKOM Unud	Pembuatan infrastuktur jaringan lan dan instalasi jaringan untuk lab di lantai 2 gedung bd	
79	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	22-11-2016			
80	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	23-11-2016			
81	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	24-11-2016	Gedung BD ILKOM Unud	Pengecekan jaringan di Gedung BD	
82	I Gede Oka Gartria Atitama, S.Kom., M.Kom.	25-11-2016	Gedung BD ILKOM Unud	Dokumentasi jaringan di Gedung BD	

..... ,

Pembimbing Lapangan,

I Gede Oka Gartria Atitama, S.Kom., M.Kom.
NIP. 1991022620160312001

.....

Lampiran 2. Dokumentasi Praktek Kerja Lapangan



Pembuatan Lab Timur Jurusan Ilmu Komputer FMIPA UNUD

1



Pembuatan Lab Timur Jurusan Ilmu Komputer FMIPA UNUD

2



Test Lab Timur dengan NCOMPUTING



**Merapikan Kabel Switch Lab Barat Jurusan Ilmu Komputer
FMIPA UNUD**



Membuat Infrastruktur Wifi Di Jurusan Ilmu Komputer



**Membuat 3 Lab Di Gedung BF Jurusan Ilmu Komputer
FMIPA UNUD**



**Membuat 3 Lab Di Gedung BF Jurusan Ilmu Komputer
FMIPA UNUD**