

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Selain kegiatan belajar-mengajar, dosen-dosen program studi Informatika UNPAR mengadakan berbagai kegiatan, baik kegiatan yang hanya melibatkan pihak internal program studi maupun pihak eksternal program studi. Contoh kegiatan tersebut adalah lokakarya, pemberian *workshop* ke sekolah-sekolah, perlombaan CHIPS, seminar, dan lain-lain.

Pada saat ini di lingkungan dosen program studi Informatika UNPAR terdapat beberapa dosen yang dipercaya untuk menempati jabatan-jabatan tertentu, sehingga pembagian beban kerja masing-masing dosen berbeda. Dosen pejabat maupun dosen non-pejabat harus dibagi dengan adil beban kerjanya.

Pelaksanaan kegiatan yang diadakan oleh dosen-dosen program studi Informatika UNPAR melibatkan pihak dosen Informatika UNPAR sebagai panitianya. Panitia yang akan ditugaskan untuk melaksanakan kegiatan tersebut dipilih oleh ketua program studi Informatika UNPAR. Panitia yang ditugaskan akan mengelola semua dokumen terkait kegiatan kemudian dokumen tersebut akan diserahkan kepada ketua prodi sebagai arsip kegiatan Informatika UNPAR.

Hingga saat ini, belum ada sistem informasi terkomputerisasi yang dapat digunakan untuk mengelola kegiatan-kegiatan yang ada di program studi Informatika UNPAR. Penyimpanan foto-foto dan dokumen terkait kegiatan dan pencatatan penugasan dosen masih dilakukan dengan cara manual. Beberapa kendala yang muncul sekarang adalah:

- Kesulitan ketua program studi Informatika Unpar untuk menyediakan atau mempresentasikan data yang akurat dengan cepat saat pelaporan ke dekan, akreditasi oleh DIKTI dan/atau visitasi.
- Kesulitan pencarian dokumen dari kegiatan tertentu.
- Pengambilan keputusan manajerial penugasan dosen.

Untuk meminimalkan masalah-masalah dalam pengelolaan kegiatan di program studi IF Unpar, akan dibangun sebuah sistem berbasis *web*. Sistem yang akan dibangun diharapkan dapat meminimalkan masalah-masalah pengelolaan kegiatan di program studi IF Unpar.

Pembangunan sistem informasi pengelolaan kegiatan ini menggunakan MySQL sebagai tempat penyimpanan. Sedangkan implementasi perangkat lunak yang dibuat menggunakan bahasa pemrograman PHP dengan *framework* CodeIgniter. *Framework* CodeIgniter merupakan salah satu *framework* dalam pembangunan *website* dengan metode MVC (*Model, View, Controller*), sehingga *framework* tersebut dapat memudahkan dalam implementasi hingga tahap pemeliharaan perangkat lunak.

### 1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, ditentukanlah rumusan masalah sebagai berikut :

- Bagaimana pengelolaan kegiatan-kegiatan di program studi Informatika Unpar?
- Bagaimana membangun sistem informasi yang dapat menyelesaikan permasalahan pengelolaan kegiatan di program studi IF Unpar?

### 1.3 Tujuan

Tujuan dari skripsi ini adalah sebagai berikut :

- Mempelajari pengelolaan kegiatan-kegiatan di program studi Informatika Unpar.
- Membangun sistem informasi, menganalisis kebutuhan-kebutuhan perangkat lunak, merancang, mengimplementasi dan menguji perangkat lunak untuk membantu mengelola kegiatan-kegiatan di program studi Informatika Unpar.

### 1.4 Batasan Masalah

Batasan masalah dari penelitian ini antara lain:

- Sistem informasi ini hanya menangani pengelolaan kegiatan-kegiatan yang diadakan oleh Program Studi Informatika Unpar dari proses pengelompokkan kegiatan hingga proses pengolahan data untuk penyusunan laporan.
- Sistem informasi ini hanya dapat digunakan oleh dosen-dosen Informatika Unpar, admin dan ketua program studi Informatika Unpar.

### 1.5 Metodologi

Bagian-bagian pekerjaan skripsi ini adalah sebagai berikut :

1. Melakukan survei ke Ketua program studi Informatika Unpar untuk mengetahui tata cara pengelolaan kegiatan-kegiatan di program studi IF Unpar.
2. Melakukan analisis pada hasil survei tata cara dan data yang diolah untuk mengelola kegiatan-kegiatan di program studi IF Unpar.
3. Melakukan studi literatur tentang pengembangan sistem informasi dan analisis sistem sejenis.
4. Mempelajari dan melakukan studi literatur tentang perancangan *database* dan bahasa pemrograman PHP dengan *framework* Code Igniter.
5. Melakukan eksperimen tentang perancangan *database* dan bahasa pemrograman PHP dengan *framework* Code Igniter.
6. Memodelkan sistem informasi tentang pengelolaan kegiatan-kegiatan di program studi IF Unpar.
7. Melakukan analisis kebutuhan perangkat lunak untuk pengelolaan kegiatan-kegiatan di program studi IF Unpar.
8. Mendesain perangkat lunak pengelolaan kegiatan-kegiatan di program studi IF Unpar dalam bentuk *prototype*.
9. Melakukan survei terhadap rancangan *prototype* perangkat lunak.
10. Melakukan implementasi perangkat lunak sesuai rancangan yang dibuat.

- 1 11. Melakukan pengujian perangkat lunak yang dibuat.
- 2 12. Menulis dokumen skripsi.

## 3 1.6 Sistematika Pembahasan

4 Sistematika pembahasan penelitian ini adalah:

- 5 1. Bab 1 Pendahuluan, berisi pembahasan latar belakang, rumusan masalah, tujuan, batasan  
6 masalah, metodologi penelitian dan sistematika pembahasan.
- 7 2. Bab 2 Dasar Teori, berisi dasar-dasar teori sistem informasi, aplikasi berbasis web, *Software*  
8 *Development Life Cycle*(SDLC), *framework CodeIgniter*.





## BAB 2

### LANDASAN TEORI

#### 2.1 Konsep Sistem Informasi

##### 2.1.1 Sistem

Sistem adalah sekumpulan komponen yang berinteraksi untuk mencapai tujuan. Hubungan antar elemen-elemen tersebut menentukan bagaimana sebuah sistem bekerja. Sedangkan sistem menurut Kamus Besar Bahasa Indonesia adalah perangkat unsur yang secara teratur saling berkaitan sehingga membentuk suatu totalitas.

##### 2.1.2 Struktur Sistem

Sebuah sistem dibagi menjadi 3 bagian, yaitu:

###### 1. Input

Input adalah semua elemen yang dibutuhkan untuk diproses oleh sistem.

###### 2. Proses

Proses adalah prosedur yang dilakukan sistem untuk mengkonversi input menjadi output.

###### 3. Output

Output mendeskripsikan hasil dari input setelah dilakukan proses.

##### 2.1.3 Data dan Informasi

###### 1. Data

Data adalah elemen-elemen yang diperoleh dari suatu aktivitas atau transaksi yang direkam berdasarkan fakta. Data belum dapat dikatakan sebagai informasi karena belum mempunyai arti bagi penerimanya.

###### 2. Informasi

Informasi adalah koleksi dari data yang sudah diorganisir dan diproses sehingga memiliki nilai lebih diatas nilai fakta individualnya sehingga lebih berguna dan lebih berarti bagi penerimanya

##### 2.1.4 Sistem Informasi

Sistem Informasi adalah sekumpulan komponen yang saling berhubungan yang mengumpulkan(input), memanipulasi(proses) dan menyebarkan (output) data dan informasi untuk mencapai tujuan tertentu. Sistem Informasi memiliki mekanisme untuk mengawasi dan mengontrol operasinya. Mekanisme ini sangat penting bagi sebuah instansi, perusahaan atau organisasi dalam mencapai tujuan dari instansi, perusahaan atau organisasi tersebut. Contohnya melihat kedisiplinan pegawai melalui sistem absen.

### 2.1.5 Sistem Informasi Berbasis Komputer

Sistem Informasi Berbasis Komputer adalah sekumpulan perangkat keras(*hardware*), perangkat lunak(*software*), *database*, jaringan, manusia dan prosedur yang dikonfigurasi untuk mengumpulkan, memanipulasi, menyimpan dan memproses data menjadi informasi.

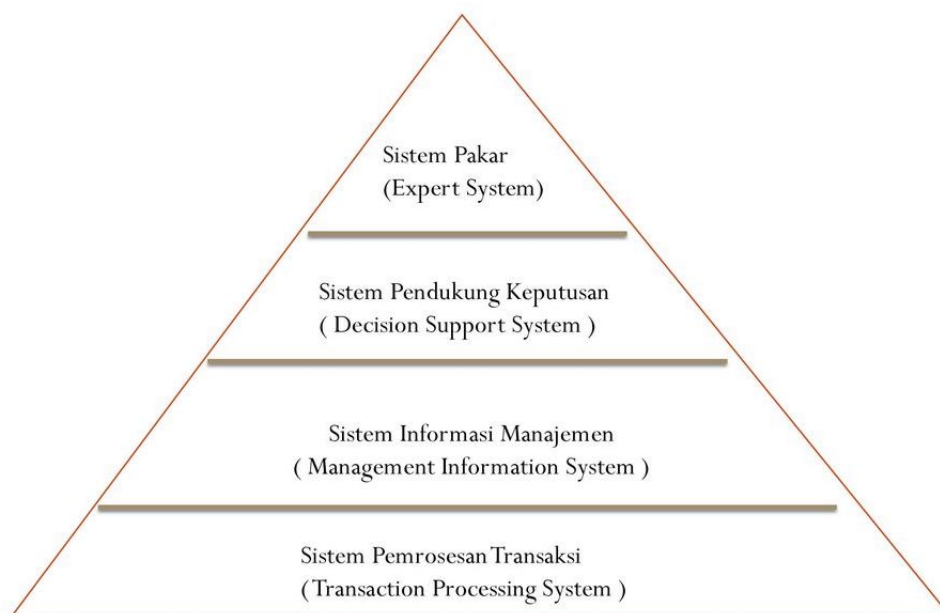
Sistem Informasi Berbasis Komputer terdiri atas beberapa komponen yang saling berintegrasi, yaitu:

1. *Hardware*(perangkat keras), peralatan komputer yang digunakan untuk melakukan aktivitas input, proses dan output.
2. *Software*(perangkat lunak), program komputer yang mengatur operasi pada komputer.
3. *Database*, tempat yang menyimpan dan mengorganisir data dan informasi.
4. Jaringan, peralatan komputer yang menghubungkan antar gawai.
5. Manusia, pihak yang berperan sebagai pengendali dan pelaksana.
6. Prosedur, tata cara kerja.

### 2.1.6 Jenis-jenis Sistem Informasi

Sistem Informasi di berbagai instansi, perusahaan atau organisasi memiliki fungsi yang bervariasi. Ada sistem informasi yang berfungsi untuk mencatat berbagai transaksi atau perubahan data yang terjadi, sistem informasi yang berguna untuk menghasilkan berbagai informasi yang digunakan sebagai bahan pengambilan keputusan dan sistem informasi yang bertugas memandu manajemen untuk mengambil keputusan. Masing-masing fungsi sistem tersebut dapat digunakan sebagai dasar untuk menggolongkan sistem informasi agar lebih mudah untuk diidentifikasi, dipelajari dan dikembangkan.

Sistem informasi dapat dikelompokkan berdasarkan beberapa cara, misalnya berdasarkan tingkatannya, kegunaannya dan berdasarkan prosesnya.



Gambar 2.1: Tingkatan sistem informasi

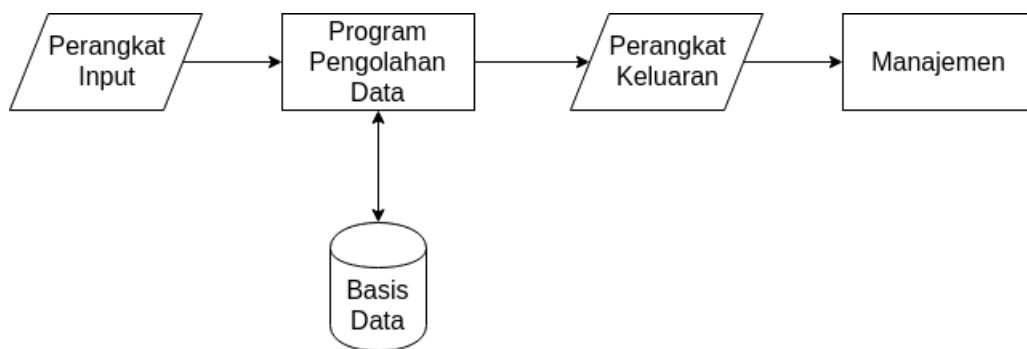
Dari gambar 2.1, sistem informasi dapat digolongkan ke dalam empat tingkatan, yaitu:

1. Sistem Pemrosesan Transaksi, fungsi utamanya adalah mencatat berbagai jenis transaksi.
2. Sistem Informasi Manajemen, fungsi utamanya adalah menyediakan berbagai bentuk informasi.
3. Sistem Pemandu Keputusan, fungsi utamanya adalah membantu manajemen dalam mengambil keputusan.
4. Sistem Pakar/Kepandaian Buatan, fungsi utamanya adalah menggantikan manusia dalam mengambil keputusan.

### 2.1.7 Sistem Pemrosesan Transaksi (TPS)

Sistem Pemrosesan Transaksi adalah jenis sistem informasi yang paling banyak dipakai karena fungsi utamanya yang merekam semua aktivitas atau transaksi yang ada di dalam instansi, perusahaan atau organisasi. Selain merekam semua aktivitas dan transaksi, TPS juga bertugas untuk menghasilkan informasi atau laporan dalam bentuk yang sudah baku, sifatnya rutin dan tidak banyak variasinya sehingga selalu mendapatkan laporan terbaru yang dapat disebarkan kapan saja. Manajemen pelaksana adalah manajemen yang banyak terlibat dalam penggunaan TPS. Manajemen pelaksana memerlukan laporan dari TPS untuk mengambil keputusan terstruktur. Keputusan terstruktur adalah keputusan yang masalahnya jelas dan solusi jelas, contoh: barang di toko habis, maka solusinya adalah ambil barang di gudang.

Sistem Pemrosesan Transaksi dapat dimodelkan seperti gambar 2.2.



Gambar 2.2: Model Sistem Pemrosesan Transaksi

Perangkat input terdiri atas berbagai alat, meliputi komputer, semua jenis telepon dan faksimile. Selain alat input juga dapat terdiri dari dokumen digital, seperti contohnya slip penjualan, faktur, dan lain-lain. Perangkat pemroses dapat terdiri atas program komputer yang dapat dipasang di dalam komputer, mesin ATM atau perangkat sejenis. Perangkat keluaran meliputi berbagai jenis monitor, printer, *speaker*, dan lain-lain.

### Langkah-langkah Pemrosesan Transaksi pada TPS

#### 1. *Data Entry*

Untuk dapat diproses, data transaksi pertama-tama harus dimasukkan ke dalam sistem.

#### 2. *Data Validation*

Validasi data merupakan langkah penting dalam pemrosesan transaksi untuk memastikan akurasi dan keandalan data. Untuk memastikan akurasi dan keandalan data tersebut, maka terlebih dahulu harus dibuat standar atau hasil yang telah diketahui.

#### 3. *Processing and revalidation*

Setelah akurasi dan reliabilitas data divalidasi, data siap untuk diproses. Ada dua cara untuk memproses transaksi, yaitu *online transaction processing (OLTP)* dan *batch processing*.

- *Online transaction processing (OLTP)*

Pemrosesan transaksi *online* adalah pemrosesan data yang hampir seketika. Istilah *online* berarti bahwa perangkat input data secara langsung terkait dengan TPS, oleh karena itu data diproses segera setelah dimasukkan ke dalam sistem. Perangkat input mungkin berada di lokasi berbeda dan dihubungkan ke sistem melalui jaringan atau melalui sistem telekomunikasi.

- *Batch processing*

Dalam *batch processing*, transaksi diakumulasikan dari waktu ke waktu dan diproses secara berkala. *Batch processing* dapat dilakukan secara harian, mingguan, atau bulanan, atau periode waktu lainnya yang sesuai dengan aplikasi yang diberikan.

#### 4. *Storage*

Data yang diproses harus disimpan dengan hati-hati dan tepat untuk digunakan di masa mendatang. Penyimpanan data adalah faktor penting bagi banyak organisasi karena nilai dan kegunaan data berkurang jika data tidak disimpan dengan benar. Jenis pengolahan dan jenis media penyimpanan, sampai taraf tertentu, merupakan masalah yang terkait.

#### 5. *Output generation*

Setelah data dimasukkan, divalidasi, diproses, divalidasi ulang, dan disimpan, hasilnya dapat dikomunikasikan kepada pengambil keputusan dalam bentuk dokumen, laporan dan formulir.

#### 6. *Query support*

Langkah terakhir dalam memproses transaksi adalah melakukan query pada sistem. Tahap ini memungkinkan pengguna untuk mengakses data dan informasi yang mungkin tidak tersedia secara langsung.

## 2.2 Konsep Aplikasi Berbasis Web

### 2.2.1 Asal Mula Web

Tim Berners-Lee menulis proposal yang ditujukan pada peneliti di CERN untuk membuat manajemen sistem informasi berbasis *hypertext* pada tahun 1989. Proposal tersebut kurang ditanggapi oleh peneliti-peneliti di CERN. Robert Cailliau adalah salah satu peneliti yang tertarik dengan proposal tersebut kemudian membantu Tim Berners-Lee mengajukan ulang proposal tersebut dan menamakan sistem tersebut *World Wide Web*.

Pada akhir 1990, Berners-Lee mengimplementasi *server* dan browser yang menggunakan *command-line* dengan menggunakan versi awal dari *Hypertext Trasfer Protocol*(HTTP) yang didesain untuk sistem ini. Pada pertengahan 1991, *server* dan browser telah dipakai untuk seluruh CERN. Akhirnya, perangkat lunak yang dikembangkan Berners-Lee digunakan untuk unduhan anonim di internet. HTTP dan *web* berkembang pesat karena peminatan banyak orang terhadap *web*. Sudah banyak perkembangan yang dapat dirasakan, seperti browser yang memiliki tampilan(GUI).

### 2.2.2 Definisi Web

World Wide Web (WWW) atau lebih dikenal dengan istilah web merupakan salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke internet. WWW adalah sejumlah besar dokumen, gambar dan sumber daya lainnya yang saling berkaitan yang dihubungkan oleh *hyperlink* dan *Universal Resource Locator (URL)*. *Hyperlink* dan URL memungkinkan *web server* menyimpan dokumen, gambar, atau sumber daya untuk kemudian dikirimkan sesuai permintaan menggunakan protokol HTTP.

Web pada awalnya adalah ruang informasi dalam internet yang menggunakan teknologi *hypertext*. Pengguna web dituntun untuk menemukan informasi dengan mengikuti tautan yang disediakan



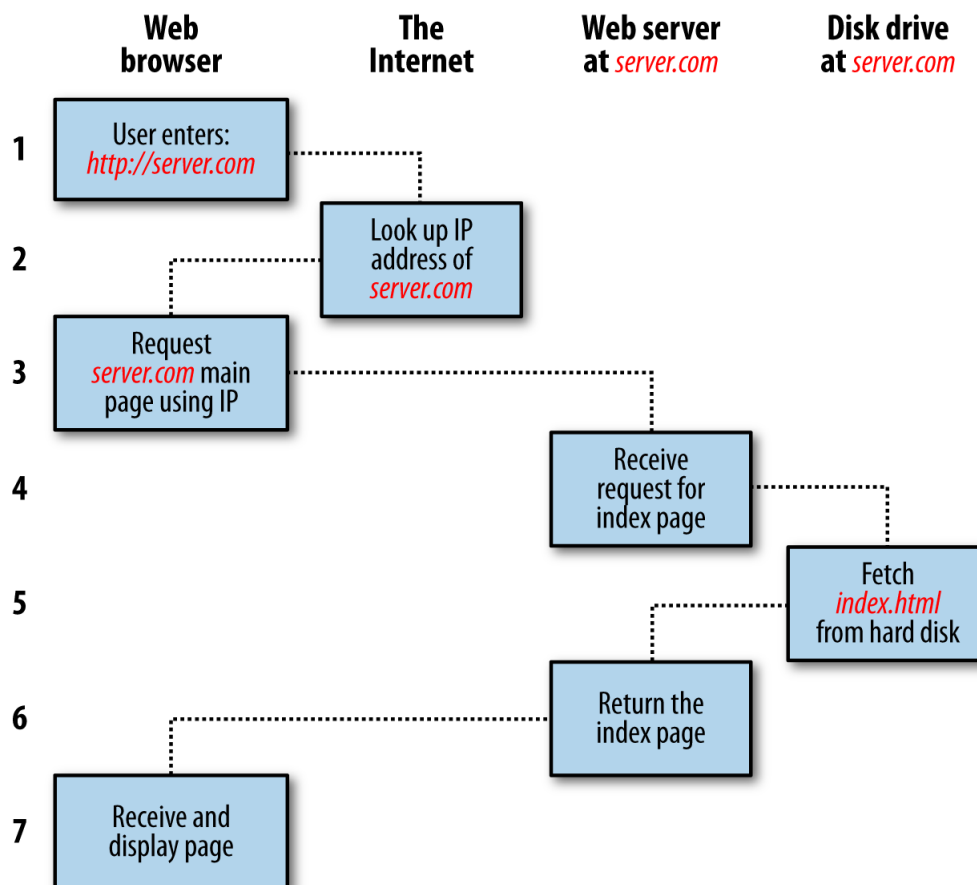
1 dalam dokumen web yang ditampilkan pada *browser* web. Kepopuleran web sebagai standar  
 2 *interface* pada layanan-layanan yang ada di internet menyebabkan pengembangan pemakaian web,  
 3 tidak hanya untuk penyedia informasi saja, tetapi juga untuk komunikasi (*e-mail*, *chatting*) sampai  
 4 dengan melakukan transaksi bisnis.

### 5 2.2.3 Arsitektur Aplikasi Berbasis Web

6 Aplikasi berbasis web berjalan atas dasar mekanisme *client-server*. Sisi *client* adalah perangkat  
 7 elektronik seperti komputer, *handphone*, *smartphone*, dll yang tertanam perangkat lunak *web*  
 8 *browser* di dalamnya. Sedangkan sisi server adalah perangkat elektronik yang dipasang *web server* di  
 9 dalamnya. *Client-server* untuk aplikasi berbasis web biasanya berkomunikasi menggunakan protokol  
 10 HTTP yang ditempatkan diantara *client* dan *server*. Komunikasi tersebut dapat menggunakan  
 11 jaringan dalam lingkup kecil seperti *local area network* atau lingkup besar seperti Internet.

12 Pada model *client-server*, Client dapat meminta *resources* dari *server* dan *server* berkewajiban  
 13 untuk memenuhi permintaan *client*. Mekanisme tersebut juga disebut mekanisme *request* dan  
 14 *response*.

15 Pada level sederhana, proses *request/response* terdiri dari *web browser* meminta *web server* untuk  
 16 mengirimkan halaman *web* dan server membalas dengan mengirim halaman *web* yang diminta.



Gambar 2.3: Prosedur *request* dan *response* pada level sederhana

17 Langkah-langkah pada *request* dan *responsenya* adalah sebagai berikut:

- 18 1. Memasukkan *http://server.com* pada *address bar* browser
- 19 2. Browser akan mencari IP *address* dari *server.com*.
- 20 3. Browser meminta *home page* pada *server.com*.

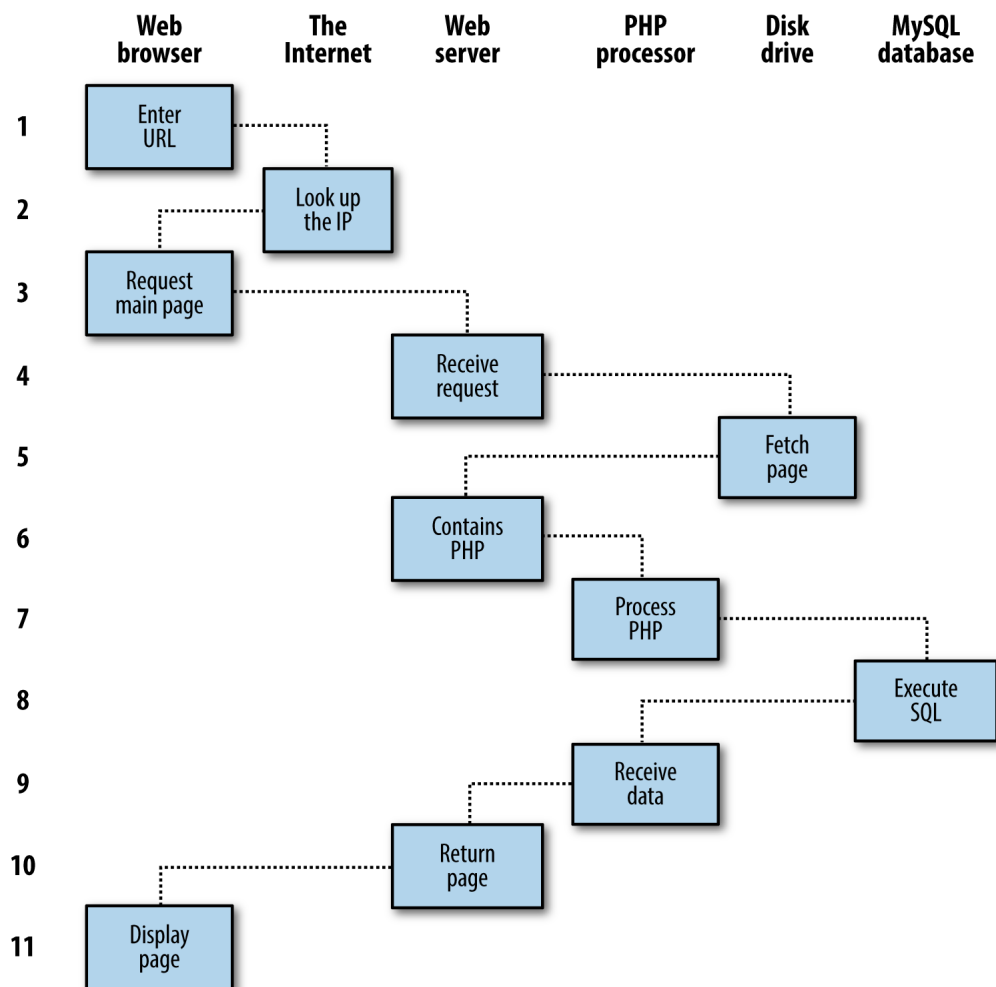
4. *Request* dari browser tadi akan melewati internet menuju *web server*.

5. *Web server* menerima *request* tersebut, kemudian mencari halaman *web* pada *hard disk server*.

6. Halaman *web* tersebut kemudian dikirimkan ke *web browser*.

7. *Browser* menampilkan halaman *web*.

Untuk *dynamic web page*, prosedurnya lebih rumit karena membawa *server side script*, pada penjelasan ini menggunakan bahasa pemrograman PHP dan MySQL *database*.



Gambar 2.4: Prosedur *request* dan *response* pada *dynamic web page*

Langkah-langkah pada *request* dan *responsenya* adalah sebagai berikut:

1. Memasukkan *http://server.com* pada *address bar* browser

2. Browser akan mencari *IP address* dari *server.com*.

3. Browser meminta *home page* pada *server.com*.

4. *Request* dari browser tadi akan melewati internet menuju *web server*.

5. *Web server* menerima *request* tersebut, kemudian mencari halaman *web* pada *hard disk server*.

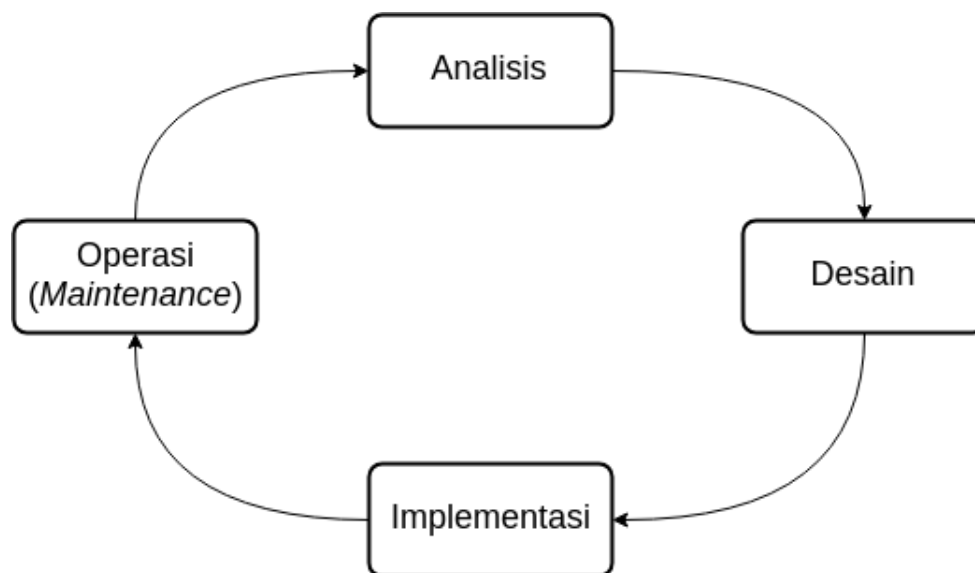
6. *Web server* melihat bahwa file yang dikirimkan dari *hard disk* mengandung *script* PHP dan mengirim halaman tersebut ke PHP *interpreter*.

7. PHP *interpreter* menjalankan kode PHP tersebut.
8. Beberapa *script* PHP mungkin mengandung *query* MySQL, oleh sebab itu PHP *interpreter* akan melewati *database* MySQL.
9. *Database* MySQL mengembalikan hasil dari *statement script* PHP ke PHP *interpreter*.
10. PHP *interpreter* mengembalikan hasil dari *script* yang telah dijalankan sebelumnya bersama dengan hasil dari *database* MySQL ke *web server*.
11. *Web server* mengembalikan halaman yang diminta oleh *client* kemudian akan ditampilkan oleh *web browser*.

## 2.3 Software Development Life Cycle

Dalam pembangunan sistem informasi, dibutuhkan langkah-langkah kerja yang terstruktur untuk dapat menghasilkan produk akhir yang sesuai tujuan. Terdapat dua kelompok cara untuk mengembangkan sistemnya, yaitu cara konvensional dan cara cepat. Cara konvensional sering disebut dengan SDLC (singkatan dari *Software Development Life Cycle* atau siklus pengembangan sistem). Cara kedua adalah metode cepat atau sering disebut juga dengan istilah RAD (singkatan dari *Rapid Application Development*).

SDLC adalah pendekatan bertahap untuk melakukan analisis dan desain terhadap sistem yang menyatakan pengembangan sistem paling baik dengan menggunakan siklus analisis tertentu. SDLC sendiri berfungsi sebagai paduan pengembangan sistem informasi dan menyediakan media yang fleksibel jika terjadi perubahan agar tetap konsisten. Pengembangan sistem dengan cara ini terdiri atas empat langkah besar seperti gambar 2.5.



Gambar 2.5: Pengembangan sistem dengan cara SDLC

Berikut penjelasan untuk setiap langkah dalam SDLC.

### 1. Analisis

Pada tahap analisis, manajemen perusahaan atau organisasi mengidentifikasi masalah yang ada atau menciptakan masalah baru. Masalah yang ada misalnya menurunnya penjualan dari bulan ke bulan, atau semakin seringnya terjadi kemacetan produksi karena kerusakan mesin. Identifikasi masalah ini akan menunjukkan masalah-masalah yang akan ditangani oleh sistem.

Selain masalah, analisis terhadap proses, rencana atau unit yang ada dalam perusahaan atau organisasi juga diperlukan.

Setelah berbagai analisis dilakukan terhadap masalah, proses, rencana atau unit yang ada di perusahaan atau organisasi, maka perlu ada usulan jalan keluarnya. Usulan ini dapat berupa proses bisnis baru atau lama yang dibuat lebih baik dengan memanfaatkan sistem terkomputerisasi.

## 2. Desain

Perancangan atau desain sistem adalah tahap untuk merealisasikan usulan yang dihasilkan pada tahap analisis. Pada tahap ini berbagai pekerjaan yang dilakukan antara lain:

- Merancang berbagai prosedur atau langkah-langkah baku untuk menangani suatu pekerjaan.
- Merancang struktur organisasi yang akan terkait dengan suatu transaksi atau pekerjaan.
- Merancang berbagai dokumen *input* maupun *output*. Dokumen ini dapat berupa dokumen tercetak maupun tidak tercetak (tampilan komputer).
- Merancang susunan jaringan komputer.
- Merancang basis data yang akan digunakan di dalam sistem.

Agar perancangan sistem berhasil dengan baik, perancang harus memperhatikan kebutuhan pemakai sistem. Tanpa pengujian perancangan, jika perancangan yang dibuat tidak memenuhi kebutuhan pemakai sistem, maka dapat dikatakan perancangan yang dilakukan gagal. Hasil tahap desain ini adalah rancangan sistem baru yang akan masuk ke tahap pengembangan selanjutnya, yaitu tahap implementasi.

## 3. Implementasi

Tahap implementasi adalah tahap menjalankan sistem yang baru saja dirancang. Dalam tahap ini, perancang sistem melakukan:

- Pemasangan berbagai perangkat (komputer, pesawat faks, *cash register*, dan alat-alat lain).
- Pelatihan kepada para pemakai sistem maupun para penerima informasi sistem.
- Jika diperlukan, perancang sistem akan mengadakan perubahan kecil atas rancangannya.

Untuk melakukan tahap implementasi pada SDLC, terdapat tiga metode yang dapat dipakai, yaitu metode langsung, metode bertahap dan metode paralel.

### (a) Metode langsung

Dalam metode ini sistem lama langsung digantikan oleh sistem yang baru. Keuntungannya adalah implementasi tidak membutuhkan waktu lama untuk menggunakan sistem yang baru. Kerugiannya adalah apabila penerapan sistem baru gagal dan sistem lama terlanjur dihentikan, maka perusahaan atau organisasi akan kesulitan beroperasi.

### (b) Metode bertahap

Dalam metode bertahap, sebagian sistem lama akan dihentikan dan digantikan dengan sistem baru. Apabila sistem baru berjalan dengan baik, maka akan dilanjutkan dengan penghentian sistem lama yang lain. Maka apabila suatu subsistem gagal diterapkan, tidak akan mempengaruhi sistem yang lain. Tetapi untuk melakukan metode ini memerlukan relatif lama untuk pengimplementasiannya.

### (c) Metode paralel

Dalam metode ini, sistem baru akan digunakan bersama-sama dengan sistem lama dalam kurun waktu tertentu. Apabila sistem baru dapat berfungsi dengan baik, sistem lama akan segera dihentikan.

#### 4. Operasi(*Maintenance*)

Tahap paling akhir dalam pengembangan sistem adalah penggunaan sistem. Apabila sudah diujicoba dan diterapkan ternyata sistem baru berjalan dengan baik, maka manajemen akan memutuskan untuk menggunakan sistem tersebut. Karena digunakan, sistem ini akan menghasilkan berbagai informasi yang diperlukan oleh pembuat keputusan.

Dalam tahap ini, dimungkinkan untuk diadakan perubahan kecil, misalnya merubah warna, menambah informasi tertentu, dll. Apabila perubahan yang dilakukan besar, maka manajemen akan kembali melakukan tahap analisis dan tahap-tahap selanjutnya, maka siklus pengembangan sistem akan berulang lagi dari awal. Apabila pengembangan sistem tidak disetujui, maka perusahaan atau organisasi akan kembali ke sistem lama yang sudah dimodifikasi.

## 2.4 Framework CodeIgniter

CodeIgniter adalah sebuah *framework* pengembangan perangkat lunak berbasis web menggunakan bahasa pemrograman PHP. Tujuan utama dari dibuatnya *framework* ini adalah untuk mendukung pengembangan perangkat lunak dengan waktu yang jauh lebih cepat dengan memanfaatkan *toolkit* yang sederhana dan elegan untuk membuat aplikasi web berfitur lengkap. CodeIgniter merupakan *framework* yang menggunakan arsitektur *Model-View-Controller* (MVC).

### 2.4.1 Arsitektur Model-View-Controller

CodeIgniter dibangun berdasarkan model pengembangan *Model-View-Controller* (MVC). MVC adalah pendekatan perangkat lunak yang memisahkan logik aplikasi dari bagian presentasi. Dalam penerapannya, model ini memungkinkan halaman web penggunanya mengandung *script* tampilan yang minimal karena terpisah dengan *script* PHP. MVC terbagi menjadi 3 komponen, yaitu:

#### 1. *Model*

*Model* merepresentasikan struktur data aplikasi. Pada umumnya, model mengandung fungsi untuk mengambil data, memasukkan data baru dan memperbarui data dalam basis data aplikasi.

#### 2. *View*

*View* adalah informasi yang ditampilkan ke pengguna. Pada umumnya, *view* akan berupa halaman web, tetapi pada CodeIgniter, *view* dapat berupa potongan halaman seperti *header* atau *footer*.

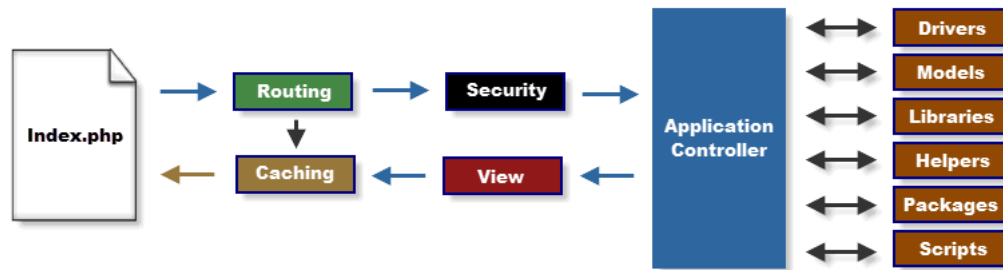
#### 3. *Controller*

Controller berperan sebagai jembatan penghubung antara *model*, *view*, dan *resource* lain yang dibutuhkan untuk memproses *request* HTTP dan membangkitkan sebuah halaman web.

CodeIgniter menggunakan pendekatan yang tidak terlalu mengikat kepada model MVC karena *model* pada *framework* ini bersifat opsional.

### 2.4.2 Flowchart Aplikasi

Untuk menggambarkan alur data pada sistem yang menggunakan *framework* CodeIgniter, berikut ini flowchart aplikasi berbasis CodeIgniter.



Gambar 2.6: Flowchart aplikasi berbasis CodeIgniter

Berdasarkan flowchart pada gambar 2.6, alur data pada sistem yang menggunakan CodeIgniter adalah:

1. *File* index.php berperan sebagai *controller* utama yang menginisialisasi *resource* dasar yang dibutuhkan untuk menjalankan CodeIgniter.
2. *Router* kemudian menganalisa *request* HTTP dari *client* dan menentukan tindakan yang harus diambil.
3. Apabila *cache*file masih tersedia, *cache* tersebut dikirimkan langsung ke browser dengan tidak melakukan eksekusi sistem normal.
4. Sebelum *controller* aplikasi dipanggil, *request* HTTP dan data yang dikirim dari *client* akan diperiksa terlebih dahulu untuk memastikan keamanan.
5. *Controller* memanggil semua *model*, *library*, *plugin*, *helper* dan *resource* lainnya yang dibutuhkan untuk memproses *request* tersebut.
6. *Rendering view* yang telah difinalisasi dilakukan kemudian dikirim ke browser untuk ditampilkan. Apabila *caching* dinyalakan, *view* tersebut disimpan dalam *cache* terlebih dahulu sehingga saat dibutuhkan kembali dapat langsung ditampilkan.

## 2.5 Kementerian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia

Kementerian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia adalah kementerian dalam Pemerintah Indonesia yang menyelenggarakan urusan di bidang riset, teknologi, dan pendidikan tinggi. Kementerian Riset, Teknologi, dan Pendidikan Tinggi berada di bawah dan bertanggung jawab kepada Presiden. Kemenristekdikti dipimpin oleh seorang Menteri Riset, Teknologi dan Pendidikan Tinggi (Menristekdikti) yang sejak tanggal 27 Oktober 2014 dijabat oleh Muhammad Nasir.

### 2.5.1 Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT)

Badan Akreditasi Nasional Perguruan Tinggi merupakan satu-satunya badan akreditasi yang memperoleh wewenang dari Kementerian Riset, Teknologi dan Pendidikan Tinggi Republik Indonesia dalam meningkatkan mutu pendidikan tinggi, memperkenalkan serta menyebarkan "Paradigma Baru dalam Pengelolaan Pendidikan Tinggi", dan meningkatkan relevansi, atmosfer akademik, pengelolaan institusi, efisiensi dan keberlanjutan pendidikan tinggi. BAN-PT berdiri pada tahun 1994, berlandaskan UU No. 2 tahun 1989 tentang Sistem Pendidikan Nasional, dan PP No. 60 tahun 1999 tentang Pendidikan Tinggi. Sebagai satu satunya badan akreditasi yang diakui oleh pemerintah BAN-PT memiliki wewenang untuk melaksanakan sistem akreditasi pada pendidikan

tinggi. Fungsi utama Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT) menurut peraturan perundangan yang ada (UU No. 20 tahun 2003, PP No. 60/1999, SK Menteri Pendidikan Nasional No. 118/U/2003), pada dasarnya adalah membantu Menteri Pendidikan Nasional dalam pelaksanaan salah satu kewajiban perundangannya, yaitu penilaian mutu perguruan tinggi, Kedinasan, Keagamaan, dan Swasta.

Proses akreditasi program studi dimulai dengan pelaksanaan evaluasi diri di program studi yang bersangkutan. Evaluasi diri tersebut mengacu pada pedoman evaluasi diri yang telah diterbitkan BAN-PT, namun, jika dianggap perlu, pihak pengelola program studi dapat menambahkan unsur-unsur yang akan dievaluasi sesuai dengan kepentingan program studi maupun institusi perguruan tinggi yang bersangkutan. Dari hasil pelaksanaan evaluasi diri tersebut, dibuat sebuah rangkuman eksekutif (*executive summary*), yang selanjutnya rangkuman eksekutif tersebut dilampirkan dalam surat permohonan untuk diakreditasi yang dikirimkan ke sekretariat BAN-PT. Sekretariat BAN-PT akan mengkaji ringkasan eksekutif dari program studi tersebut, dan jika telah memenuhi semua komponen yang diminta dalam pedoman evaluasi diri, sekretariat BAN-PT akan mengirimkan instrumen akreditasi yang sesuai dengan tingkat program studi. Setelah instrumen akreditasi diisi, program studi mengirimkan seluruh berkas (instrumen akreditasi yang telah diisi dan lampirannya, beserta salinannya) ke sekretariat BAN-PT. Jumlah copy yang harus disertakan untuk program studi tingkat Diploma dan Sarjana sebanyak 3 copy, sedangkan untuk program studi tingkat Magister dan Doktor sebanyak 4 copy. Penilaian dilakukan setelah seluruh berkas diterima secara lengkap oleh sekretariat BAN-PT.





## BAB 3

### ANALISIS

#### 3.1 Analisis Lingkungan Program Studi Informatika UNPAR

Terdapat perbedaan lingkungan kerja yang terdapat pada program studi Informatika UNPAR jika dibagi berdasarkan tugasnya.

##### 1. Dosen

Tugas utama seorang dosen adalah melaksanakan pendidikan, penelitian dan pengabdian kepada masyarakat serta merencanakan, melaksanakan proses pembelajaran, serta menilai dan mengevaluasi hasil pembelajaran. Namun terdapat beberapa dosen yang dipercaya untuk menduduki jabatan tertentu pada sebuah program studi. Dosen pejabat adalah dosen yang dipercaya untuk menduduki jabatan tertentu. Selain melakukan tugas utama sebagai dosen, dosen pejabat juga bertugas untuk mengurus hal-hal yang bersifat administratif. Sebaliknya dosen non-pejabat lebih banyak bertugas untuk melakukan tugas utama dosen.

##### 2. Tata Usaha

Pegawai tata usaha bertugas untuk mengurus semua hal-hal administrasi. Contohnya membagi jadwal mengawas ujian.

##### 3. Pekarya

Pekarya bertugas untuk menyiapkan ruang kelas yang akan dipakai untuk kegiatan belajar-mengajar.

#### 3.2 Analisis Prosedur Kini

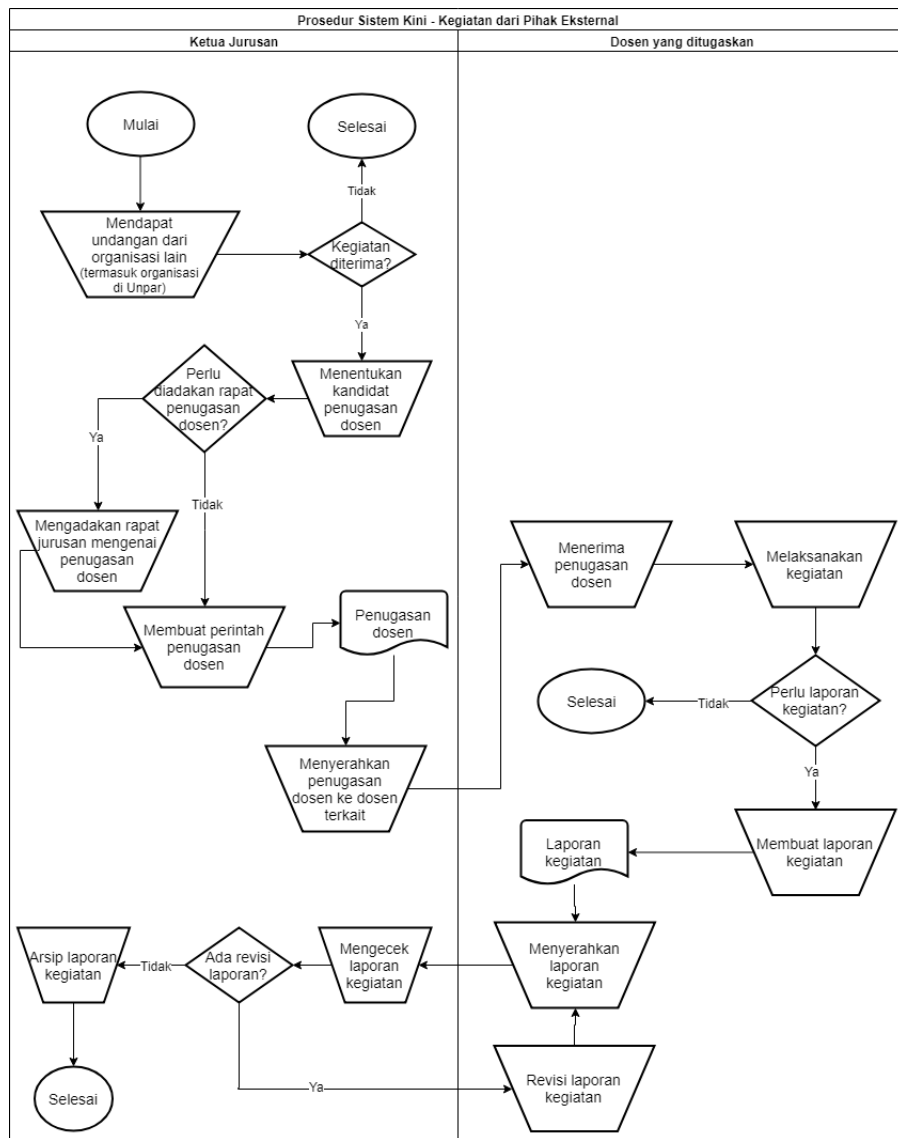
##### 3.2.1 Gambaran Prosedur Kini

Terdapat 2 jenis kegiatan yang dilakukan oleh Program Studi Informatika UNPAR yaitu kegiatan yang hanya melibatkan pihak internal prodi dan kegiatan yang melibatkan pihak eksternal. Kegiatan yang melibatkan pihak eksternal adalah kegiatan yang diajukan oleh pihak dosen Informatika UNPAR kepada instansi lain atau program studi Informatika UNPAR mendapat undangan yang berasal dari luar organisasi dosen-dosen Informatika UNPAR. Sedangkan kegiatan yang hanya melibatkan pihak internal prodi adalah kegiatan yang diajukan atau berasal dari organisasi dosen-dosen Informatika UNPAR. Prosedur pengajuan hingga pelaksanaan kegiatan dari setiap jenis kegiatan tersebut berbeda.

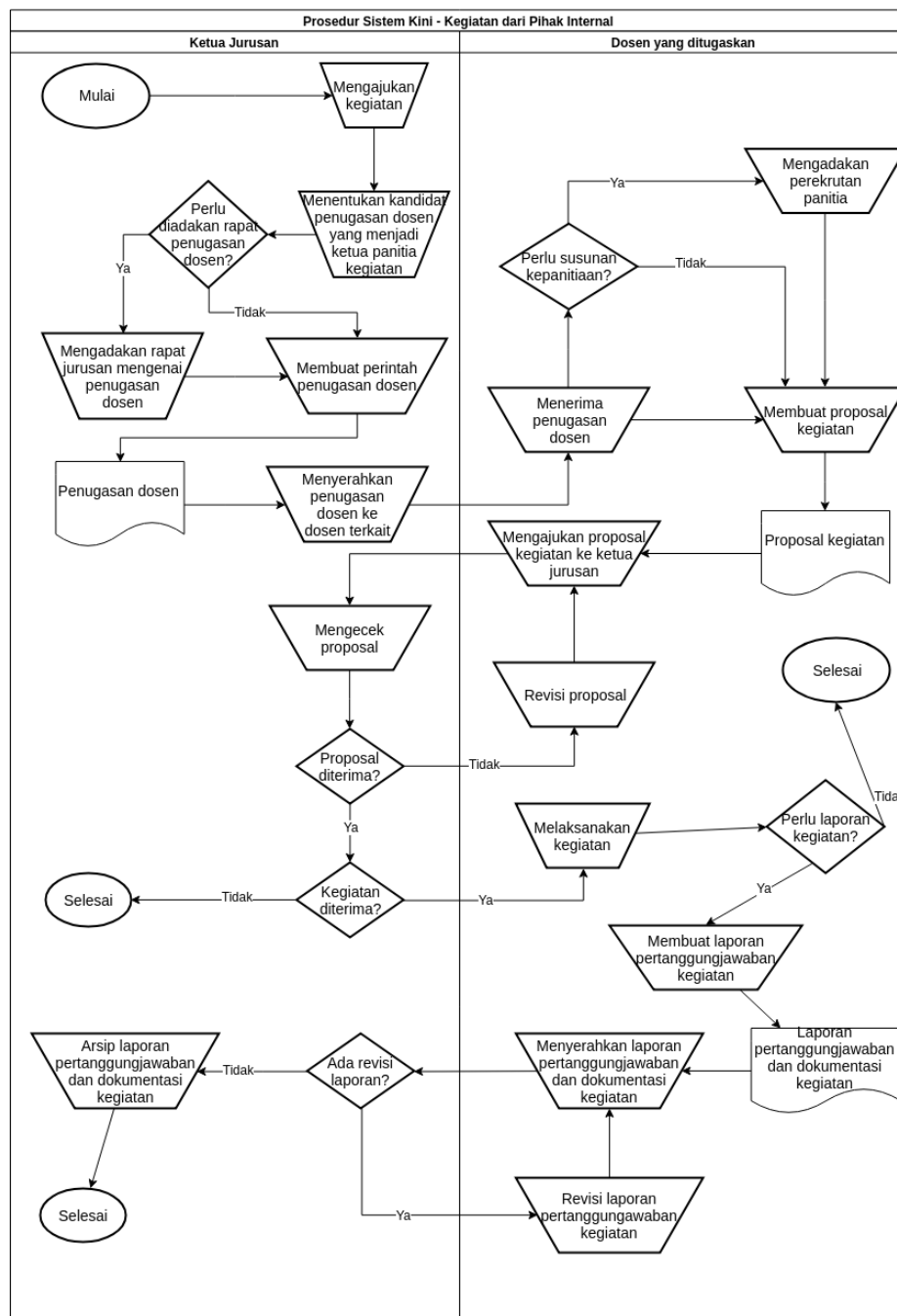
Prosedur yang melibatkan pihak eksternal dimulai dari ketua jurusan atau dosen Informatika UNPAR mendapat undangan dari instansi lain. Penerimaan undangan untuk mengikuti kegiatan dari instansi lain ditentukan oleh ketua jurusan. Ketua jurusan melihat jadwal dan kesibukan masing-masing dosen. Jika tidak memungkinkan untuk tidak mengikuti kegiatan, maka ketua jurusan akan menolak undangan tersebut. Sebaliknya jika kegiatan tersebut memungkinkan untuk diikuti, ketua jurusan akan menentukan kandidat dosen yang akan ditugaskan untuk mengikuti kegiatan tersebut. Pemilihan kandidat dosen yang akan ditugaskan dilihat dari kesibukan setiap

masing-masing dosen. Penetapan penugasan dosen didapatkan dari keputusan ketua jurusan atau rapat jurusan yang diadakan oleh ketua jurusan. Setelah menerima penugasan, dosen yang ditugaskan akan mengikuti kegiatan tersebut kemudian membuat laporan kegiatan jika diperlukan. Dokumen dan dokumentasi terkait kegiatan kemudian diserahkan ke ketua jurusan untuk diperiksa. Jika tidak terdapat revisi maka ketua jurusan akan menyimpan laporan dan dokumentasi kegiatan sebagai arsip kegiatan Program Studi Informatika UNPAR. Flowchart pelaksanaan kegiatan yang melibatkan pihak eksternal dapat dilihat pada gambar 3.5.

Sedangkan prosedur yang hanya melibatkan pihak internal Program Studi Informatika dimulai dari ketua jurusan Informatika mengusulkan kegiatan. Kemudian ketua jurusan menentukan kandidat yang cocok untuk ditugaskan menjadi ketua pelaksana kegiatan. Hampir sama seperti kegiatan yang melibatkan pihak eksternal, kandidat dosen yang akan ditugaskan untuk menjadi ketua pelaksana kegiatan juga dilihat dari kesibukan dari masing-masing dosen. Kategori selain melihat kesibukan masing-masing dosen untuk ditugaskan menjadi ketua pelaksana pada pelaksanaan kegiatan yang hanya melibatkan pihak internal program studi adalah dosen tersebut pernah menjadi bagian dari salah satu panitia dari kegiatan tersebut. Dari kandidat dosen yang telah ditentukan oleh ketua jurusan nantinya akan ditugaskan 1 dosen untuk dijadikan ketua pelaksana kegiatan dari rapat jurusan atau keputusan sepihak oleh ketua jurusan. Jika memerlukan susunan kepanitiaan, maka ketua pelaksana akan mengadakan perekrutan panitia. Kemudian ketua pelaksana kegiatan membuat proposal kegiatan yang nantinya diajukan ke ketua jurusan. Jika terdapat revisi pada proposal dari ketua jurusan, maka proposal akan direvisi oleh ketua pelaksana kemudian akan diajukan ulang. Jika proposal diterima dan kegiatan memungkinkan untuk dilaksanakan, ketua pelaksana akan melaksanakan kegiatan sesuai tanggal yang terdapat pada proposal. Kemudian jika kegiatan telah selesai dilaksanakan maka ketua pelaksana kegiatan akan menyusun laporan pertanggungjawaban dan dokumentasi kegiatan jika diperlukan. Laporan kegiatan tersebut diserahkan ke ketua jurusan Informatika UNPAR untuk diperiksa. Jika tidak terdapat revisi maka ketua jurusan akan menyimpan laporan dan dokumentasi kegiatan sebagai arsip kegiatan Program Studi Informatika UNPAR. Flowchart pelaksanaan kegiatan yang hanya melibatkan pihak internal dapat dilihat pada gambar 3.2.



Gambar 3.1: Flowchart kegiatan yang melibatkan pihak eksternal



Gambar 3.2: Flowchart kegiatan yang melibatkan pihak internal

### 3.2.2 Analisis Masalah Prosedur Kini

Ketua jurusan merupakan pihak yang bertugas untuk mengurus pengelolaan pelaksanaan kegiatan. Selama ini pengolahan dokumen dan dokumentasi terkait kegiatan oleh ketua jurusan masih dilakukan secara manual. Dokumen dan dokumentasi terkait kegiatan diarsip secara manual. Selain itu, untuk memilih kandidat penugasan dosen juga harus mencari dokumen kegiatan sebelumnya untuk melihat siapa saja dosen yang sudah pernah ditugaskan. Prosedur yang digunakan saat ini menimbulkan beberapa masalah, antara lain:

- Penyimpanan arsip data kegiatan yang pernah dilaksanakan program studi Informatika UNPAR memakan tempat.
- Risiko terjadinya *human error*. Contohnya dokumen terkait kegiatan hilang atau rusak.

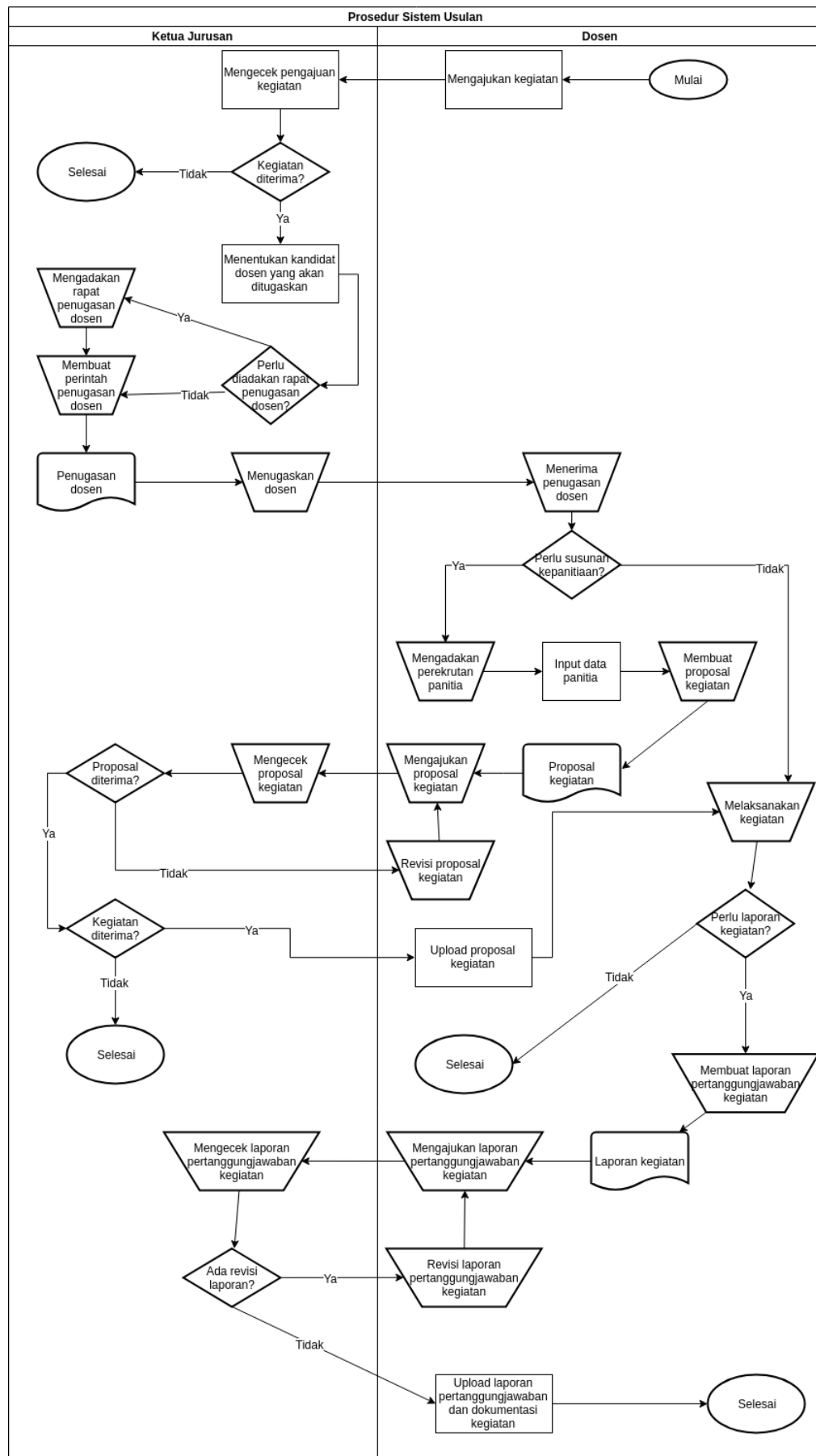
- Proses pencarian data yang pernah dilaksanakan saat dibutuhkan memakan waktu lama.

### 3.3 Deskripsi Prosedur Usulan

Untuk meminimalisir masalah-masalah yang ada selama pengelolaan kegiatan di program studi Informatika UNPAR, penulis mengusulkan prosedur baru. Prosedur baru yang diusulkan menggunakan sistem terkomputerisasi yang dirancang khusus untuk membantu pengelolaan kegiatan-kegiatan di program studi Informatika UNPAR.

Prosedur usulan dimulai dari dosen atau ketua jurusan mengusulkan kegiatan atau mendapatkan undangan dari pihak lain untuk mengikuti kegiatan. Jika kegiatan diterima maka ketua jurusan akan menentukan kandidat penugasan dosen. Untuk menentukan kandidat penugasan, ketua jurusan dapat melihat kesibukan masing-masing dosen yang telah direkam oleh sistem. Setelah menentukan kandidat, ketua jurusan akan mengadakan rapat atau melakukan penugasan sepihak untuk menentukan penugasan dosen dari kandidat tersebut. Jika kegiatan adalah kegiatan yang tidak memerlukan susunan kepanitiaan, maka dosen yang ditugaskan akan melaksanakan kegiatan sesuai tanggal yang ditentukan. Jika laporan pertanggungjawaban kegiatan diperlukan, maka dosen yang ditugaskan akan membuat laporan pertanggungjawaban kegiatan yang kemudian diserahkan ke ketua jurusan untuk diperiksa. Jika ada revisi maka dosen yang ditugaskan akan merevisi laporan dan mengajukan kembali laporan kegiatan. Jika tidak ada revisi, dosen yang ditugaskan akan mengunggah laporan dan dokumentasi kegiatan ke sistem. Jika kegiatan adalah kegiatan yang membutuhkan kepanitiaan, maka dosen yang ditugaskan akan mengadakan *open recruitment* panitia untuk kegiatan tersebut. Setelah mengadakan *open recruitment* panitia, dosen yang ditugaskan akan memasukkan data panitia ke sistem dan membuat proposal. Jika terdapat revisi pada proposal, dosen yang ditugaskan akan merevisi proposal dan akan diajukan lagi ke ketua jurusan. Jika proposal diterima dan kegiatan memungkinkan untuk dilaksanakan, dosen akan mengunggah proposal ke sistem. Kemudian dosen akan melaksanakan kegiatan sesuai tanggal yang ditentukan. Setelah kegiatan selesai dilakukan, dosen akan membuat laporan pertanggungjawaban kegiatan dari pelaksanaan kegiatan jika diperlukan. Laporan kegiatan tersebut diserahkan ke ketua jurusan Informatika UNPAR untuk diperiksa. Jika tidak terdapat revisi maka dosen akan mengunggah laporan pertanggungjawaban dan dokumentasi kegiatan ke sistem. Flowchart pelaksanaan kegiatan yang hanya melibatkan pihak internal dapat dilihat pada gambar 3.3.

Prosedur ini akan diterapkan dalam sebuah sistem informasi berjenis Sistem Pemrosesan Transaksi (TPS). Langkah-langkah pemrosesan transaksi yang terjadi pada TPS akan direalisasikan dalam bentuk sistem berbasis *web* dinamis. Sistem berbasis *web* tersebut diimplementasi menggunakan bahasa pemrograman PHP dengan *framework* CodeIgniter. *View* pada CodeIgniter digunakan untuk membuat tampilan *web* yang dapat menampilkan formulir *online* guna melakukan *input* serta menampilkan *output* setelah datanya diolah. *Controller* pada CodeIgniter digunakan untuk melakukan validasi data dari *input* serta menghubungkan *view* dengan *model*. *Model* pada CodeIgniter digunakan untuk menulis kueri yang mengakses data dan informasi kemudian ditampilkan pada halaman *web* atau menyimpan data dari *input* ke *database*.



Gambar 3.3: Flowchart prosedur usulan

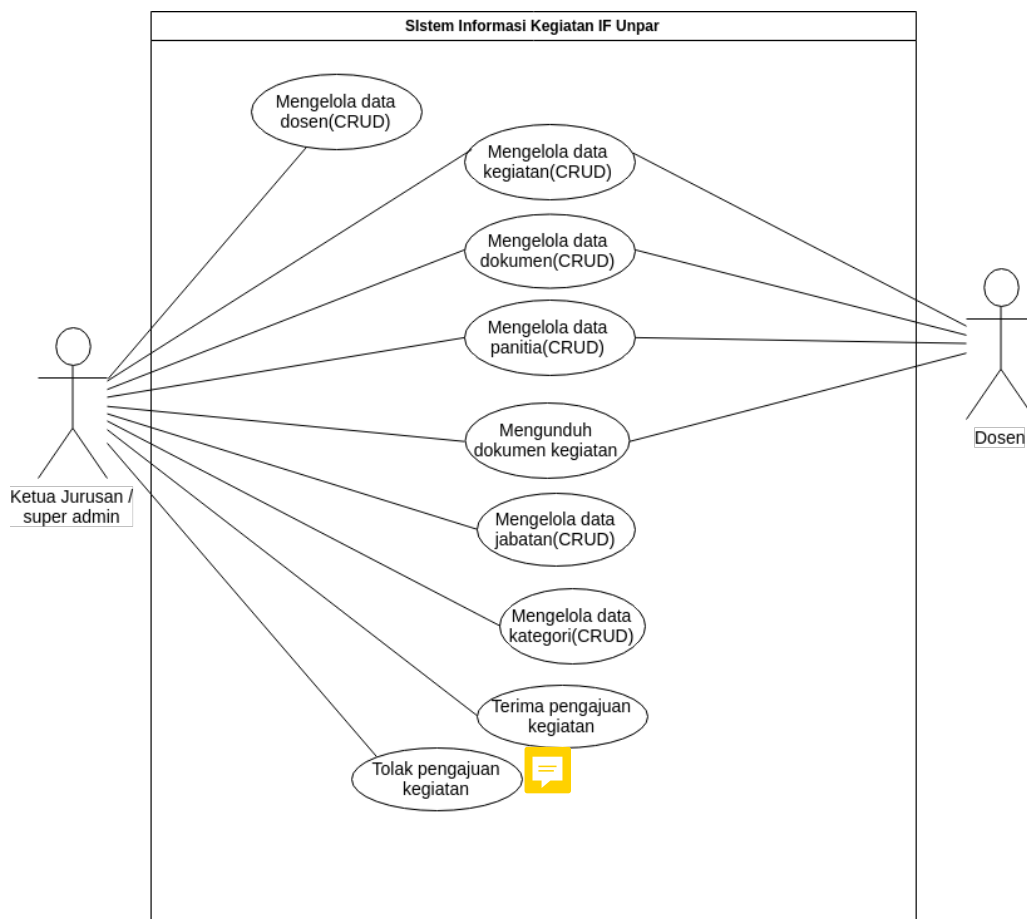
### 3.4 Spesifikasi Pengguna

Dalam sistem informasi ini, ada tiga jenis pengguna, yaitu :

1. Ketua Jurusan, yaitu ketua jurusan Informatika UNPAR sebagai pihak yang mengelola seluruh kegiatan di program studi Informatika UNPAR
2. Dosen, yaitu seluruh dosen yang tergabung pada organisasi dosen Informatika UNPAR sebagai pihak yang akan menjadi penanggung jawab atau panitia dari kegiatan program studi Informatika UNPAR
3. Super Admin, yaitu pegawai UNPAR (TU dan dosen) yang ditunjuk oleh ketua jurusan Informatika UNPAR untuk membantu mengelola kegiatan di program studi Informatika UNPAR

### 3.5 Usecase Diagram

Berdasarkan prosedur baru yang sudah diusulkan pada bagian prosedur usulan 3.3, perangkat lunak yang dibangun akan memiliki fitur yang digambarkan dalam bentuk Usecase diagram 3.4



Gambar 3.4: Usecase Diagram

Berikut ini skenario untuk setiap *usecase* dalam sistem informasi ini.

Nomor fungsi :	P-01
Nama usecase :	Mengelola data kegiatan
Deskripsi :	Fitur untuk melihat data kegiatan
Aktor :	Ketua Jurusan, Dosen
Pre-kondisi :	Ketua Jurusan atau Dosen sudah login ke dalam sistem
Pos-kondisi :	Data kegiatan tersimpan
<b>Skenario:</b>	
<b>Actor Actions</b>	<b>System Responses</b>
1. Masuk ke halaman data kegiatan	
	2. Menampilkan data Kegiatan
3. Menekan tombol tambah data kegiatan	
	4. Menampilkan form tambah kegiatan
5. Mengisi form tambah kegiatan	
6. Menekan tombol simpan	
<b>Skenario alternatif: mengubah data kegiatan</b>	
<b>Actor Actions</b>	<b>System Responses</b>
langkah 1-2 sama dengan skenario normal	
3. Menekan tombol edit pada kegiatan yang akan diubah	
	4. Menampilkan form data kegiatan yang dipilih
5. Mengubah data kegiatan	
6. Menekan tombol simpan	
<b>Skenario alternatif: hapus data kegiatan</b>	
<b>Actor Actions</b>	<b>System Responses</b>
langkah 1-2 sama dengan skenario normal	
3. Menekan tombol hapus pada kegiatan yang akan dihapus	
	4. Menampilkan halaman data kegiatan setelah dilakukan penghapusan

Tabel 3.1: Skenario mengelola data kegiatan



Nomor fungsi :		P-02	
Nama usecase :		Mengelola dokumen kegiatan	
Deskripsi :		Fitur untuk melihat data dokumen kegiatan	
Aktor :		Ketua Jurusan, Dosen	
Pre-kondisi :		Masuk ke halaman detail kegiatan yang dipilih	
Pos-kondisi :		Data dokumen kegiatan tersimpan	
Skenario:			
Actor Actions		System Responses	
1.	Menekan tombol tambah dokumen pada halaman detail kegiatan yan dipilih		
		2.	Menampilkan form dokumen kegiatan
3.	Mengisi form tambah dokumen kegiatan		
4.	Menekan tombol simpan		
Skenario alternatif: mengubah data dokumen kegiatan			
Actor Actions		System Responses	
1.	Menekan tombol edit pada dokumen kegiatan yang akan diubah		
		2.	Menampilkan form data dokumen kegiatan yang dipilih
3.	Mengubah data dokumen kegiatan		
4.	Menekan tombol simpan		
Skenario alternatif: hapus data dokumen kegiatan			
Actor Actions		System Responses	
1.	Menekan tombol hapus pada dokumen kegiatan yang akan dihapus		
		2.	Menampilkan halaman data dokumen kegiatan setelah dilakukan penghapusan

Tabel 3.2: Mengelola data dokumen kegiatan

Nomor fungsi :	P-03
Nama usecase :	Mengelola data penugasan dosen
Deskripsi :	Fitur untuk melihat data penugasan dosen
Aktor :	Ketua Jurusan, Dosen
Pre-kondisi :	Ketua Jurusan atau Dosen sudah login ke dalam sistem
Pos-kondisi :	Data penugasan dosen tersimpan
<b>Skenario:</b>	
<b>Actor Actions</b>	<b>System Responses</b>
1. Masuk ke halaman data penugasan dosen	
	2. Menampilkan data penugasan dosen
3. Menekan tombol tambah	
	4. Menampilkan form tambah penugasan dosen
5. Mengisi form tambah penugasan dosen	
6. Menekan tombol simpan	
<b>Skenario alternatif: mengubah data penugasan dosen</b>	
<b>Actor Actions</b>	<b>System Responses</b>
langkah 1-2 sama dengan skenario normal	
3. Menekan tombol edit pada penugasan dosen yang akan diubah	
	4. Menampilkan form data penugasan dosen yang akan diubah
5. Mengubah data penugasan dosen	
6. Menekan tombol simpan	
<b>Skenario alternatif: hapus data penugasan dosen</b>	
<b>Actor Actions</b>	<b>System Responses</b>
langkah 1-2 sama dengan skenario normal	
3. Menekan tombol hapus pada penugasan dosen yang akan dihapus	
	4. Menampilkan halaman data penugasan dosen setelah dilakukan penghapusan

Tabel 3.3: Skenario mengelola data penugasan dosen

Nomor fungsi :		P-04	
Nama usecase :		Mengelola data dosen	
Deskripsi :		Fitur untuk melihat data dosen	
Aktor :		Ketua Jurusan	
Pre-kondisi :		Ketua Jurusan sudah login ke dalam sistem	
Pos-kondisi :		Data dosen tersimpan	
Skenario:			
Actor Actions		System Responses	
1.	Masuk ke halaman data dosen		
		2.	Menampilkan data dosen
3.	Menekan tombol tambah		
		4.	Menampilkan form tambah dosen
5.	Mengisi form tambah dosen		
6.	Menekan tombol simpan		
Skenario alternatif: mengubah data dosen			
Actor Actions		System Responses	
langkah 1-2 sama dengan skenario normal			
3.	Menekan tombol edit pada data dosen yang akan diubah		
		4.	Menampilkan form data dosen yang akan diubah
5.	Mengubah data dosen		
6.	Menekan tombol simpan		
Skenario alternatif: hapus data dosen			
Actor Actions		System Responses	
langkah 1-2 sama dengan skenario normal			
3.	Menekan tombol hapus pada data dosen yang akan dihapus		
		4.	Menampilkan halaman data dosen setelah dilakukan penghapusan

Tabel 3.4: Skenario mengelola data dosen

Nomor fungsi :		P-05	
Nama usecase :		Mengelola data jabatan penugasan dosen	
Deskripsi :		Fitur untuk melihat data jabatan penugasan dosen	
Aktor :		Ketua Jurusan	
Pre-kondisi :		Ketua Jurusan sudah login ke dalam sistem	
Pos-kondisi :		Data jabatan penugasan dosen tersimpan	
<b>Skenario:</b>			
<b>Actor Actions</b>		<b>System Responses</b>	
1.	Masuk ke halaman data jabatan penugasan dosen		
		2.	Menampilkan data jabatan penugasan dosen
3.	Menekan tombol tambah		
		4.	Menampilkan form tambah jabatan penugasan dosen
5.	Mengisi form tambah jabatan penugasan dosen		
6.	Menekan tombol simpan		
<b>Skenario alternatif: mengubah data jabatan penugasan dosen</b>			
<b>Actor Actions</b>		<b>System Responses</b>	
langkah 1-2 sama dengan skenario normal			
3.	Menekan tombol edit pada data jabatan penugasan dosen yang akan diubah		
		4.	Menampilkan form data jabatan penugasan dosen yang akan diubah
5.	Mengubah data jabatan penugasan dosen		
6.	Menekan tombol simpan		
<b>Skenario alternatif: hapus data jabatan penugasan dosen</b>			
<b>Actor Actions</b>		<b>System Responses</b>	
langkah 1-2 sama dengan skenario normal			
3.	Menekan tombol hapus pada jabatan penugasan dosen yang akan dihapus		
		4.	Menampilkan halaman data jabatan penugasan dosen setelah dilakukan penghapusan

Tabel 3.5: Skenario mengelola data jabatan penugasan dosen

Nomor fungsi :	P-06
Nama usecase :	Mengelola data kategori kegiatan
Deskripsi :	Fitur untuk melihat data kategori kegiatan
Aktor :	Ketua Jurusan
Pre-kondisi :	Ketua Jurusan sudah login ke dalam sistem
Pos-kondisi :	Data kategori kegiatan tersimpan
<b>Skenario:</b>	
<b>Actor Actions</b>	<b>System Responses</b>
1. Masuk ke halaman data kategori kegiatan	
	2. Menampilkan data kategori kegiatan
3. Menekan tombol tambah	
	4. Menampilkan form tambah kategori kegiatan
5. Mengisi form tambah kategori kegiatan	
6. Menekan tombol simpan	
<b>Skenario alternatif: mengubah data kategori kegiatan</b>	
<b>Actor Actions</b>	<b>System Responses</b>
langkah 1-2 sama dengan skenario normal	
3. Menekan tombol edit pada data kategori kegiatan yang akan diubah	
	4. Menampilkan form data kategori kegiatan yang akan diubah
5. Mengubah data kategori kegiatan	
6. Menekan tombol simpan	
<b>Skenario alternatif: hapus data kategori kegiatan</b>	
<b>Actor Actions</b>	<b>System Responses</b>
langkah 1-2 sama dengan skenario normal	
3. Menekan tombol hapus pada kategori kegiatan yang akan dihapus	
	4. Menampilkan halaman data kategori kegiatan setelah dilakukan penghapusan

Tabel 3.6: Skenario mengelola data kategori kegiatan

Nomor fungsi :	P-07
Nama usecase :	Terima pengajuan kegiatan
Deskripsi :	Fitur untuk menerima pengajuan kegiatan
Aktor :	Ketua Jurusan
Pre-kondisi :	Ketua Jurusan sudah login ke dalam sistem
Pos-kondisi :	Pembaruan data kegiatan tersimpan
<b>Skenario:</b>	
<b>Actor Actions</b>	<b>System Responses</b>
1. Masuk ke halaman pengajuan kegiatan	
	2. Menampilkan halaman pengajuan kegiatan
3. Menekan tombol detail kegiatan yang dipilih	
	4. Menampilkan halaman detail pengajuan kegiatan yang dipilih
5. Menekan tombol terima	

Tabel 3.7: Skenario terima pengajuan kegiatan

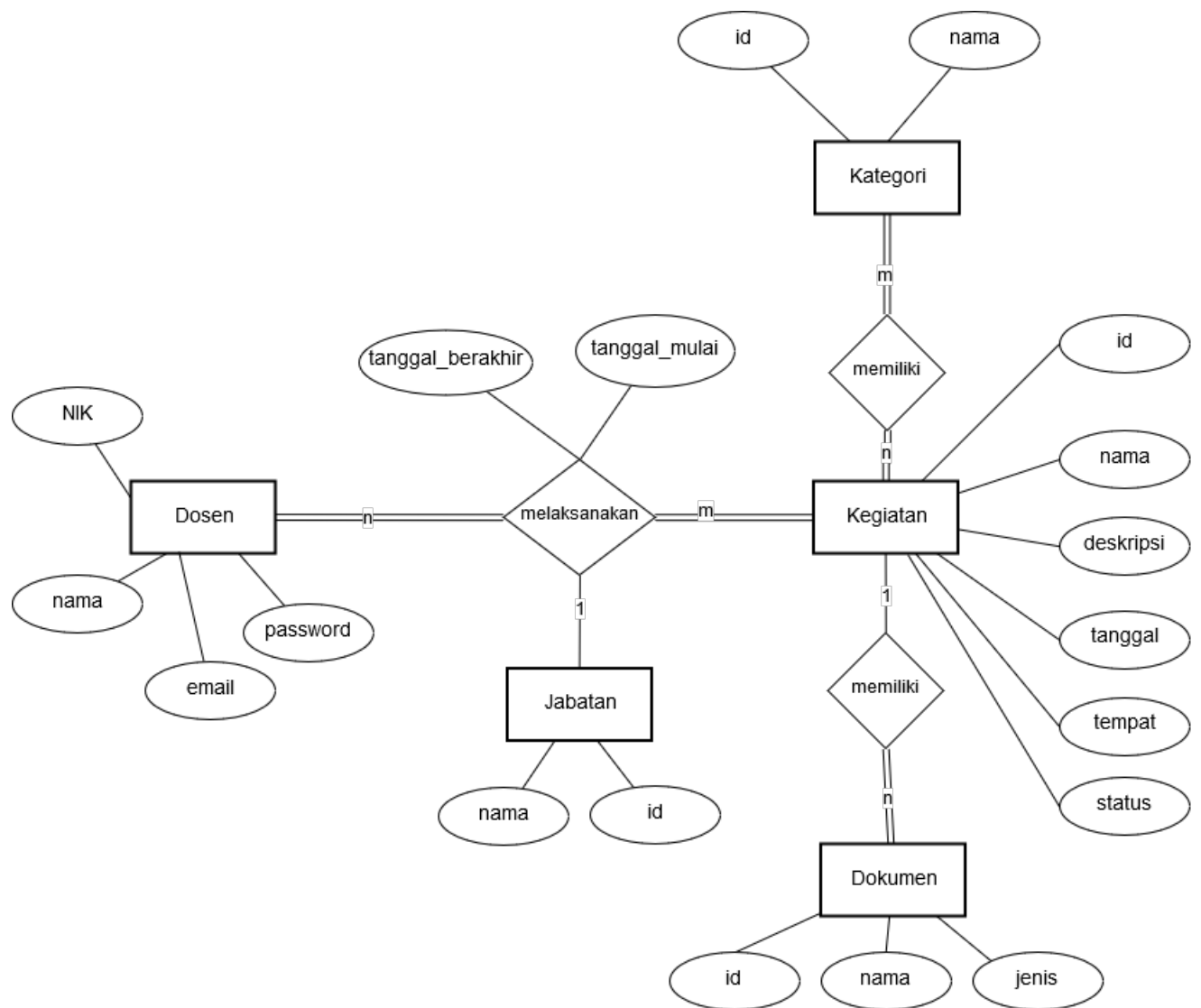
Nomor fungsi :	P-08
Nama usecase :	Tolak pengajuan kegiatan
Deskripsi :	Fitur untuk menolak pengajuan kegiatan
Aktor :	Ketua Jurusan
Pre-kondisi :	Ketua Jurusan sudah login ke dalam sistem
Pos-kondisi :	Pembaruan data kegiatan tersimpan
<b>Skenario:</b>	
<b>Actor Actions</b>	
<b>System Responses</b>	
1.	Masuk ke halaman pengajuan kegiatan
	2. Menampilkan halaman pengajuan kegiatan
3.	Menekan tombol detail kegiatan yang dipilih
	4. Menampilkan halaman detail pengajuan kegiatan yang dipilih
5.	Menekan tombol tolak

Tabel 3.8: Skenario tolak pengajuan kegiatan

Nomor fungsi :	P-09
Nama usecase :	Mengunduh dokumen kegiatan
Deskripsi :	Fitur untuk menampilkan dokumen suatu kegiatan
Aktor :	Ketua Jurusan dan Dosen
Pre-kondisi :	Masuk ke halaman detail suatu kegiatan
Pos-kondisi :	Terunduh dokumen suatu kegiatan
<b>Skenario:</b>	
<b>Actor Actions</b>	
<b>System Responses</b>	
1.	Menekan tombol unduk pada dokumen kegiatan yang akan diunduh
	2. Membangkitkan dokumen dalam bentuk gambar atau <i>file</i> pdf yang otomatis terunduh

Tabel 3.9: Skenario mengunduh dokumen kegiatan

### 1 3.6 Analisis Kebutuhan Basis Data



Gambar 3.5: ER Diagram





## DAFTAR REFERENSI



# LAMPIRAN A

## KODE PROGRAM

Listing A.1: MyCode.c

```

1
2 // This does not make algorithmic sense,
3 // but it shows off significant programming characters.
4
5 #include<stdio.h>
6
7 void myFunction( int input, float* output ) {
8     switch ( array[i] ) {
9         case 1: // This is silly code
10             if ( a >= 0 || b <= 3 && c != x )
11                 *output += 0.005 + 20050;
12             char = 'g';
13             b = 2^n + ~right_size - leftSize * MAX_SIZE;
14             c = (--aaa + &daa) / (bbb++ - ccc % 2 );
15             strcpy(a,"hello_$@?");
16         }
17         count = ~mask | 0x00FF00AA;
18     }
19
20 // Fonts for Displaying Program Code in LATEX
21 // Adrian P. Robson, nepsweb.co.uk
22 // 8 October 2012
23 // http://nepsweb.co.uk/docs/progfonts.pdf

```

Listing A.2: MyCode.java

```

1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.HashSet;
4
5 //class for set of vertices close to furthest edge
6 public class MyFurSet {
7     protected int id; //id of the set
8     protected MyEdge FurthestEdge; //the furthest edge
9     protected HashSet<MyVertex> set; //set of vertices close to furthest edge
10    protected ArrayList<ArrayList<Integer>> ordered; //list of all vertices in the set for each trajectory
11    protected ArrayList<Integer> closeID; //store the ID of all vertices
12    protected ArrayList<Double> closeDist; //store the distance of all vertices
13    protected int totaltrj; //total trajectories in the set
14
15    /*
16     * Constructor
17     * @param id : id of the set
18     * @param totaltrj : total number of trajectories in the set
19     * @param FurthestEdge : the furthest edge
20     */
21    public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
22        this.id = id;
23        this.totaltrj = totaltrj;
24        this.FurthestEdge = FurthestEdge;
25        set = new HashSet<MyVertex>();
26        ordered = new ArrayList<ArrayList<Integer>>();
27        for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
28        closeID = new ArrayList<Integer>(totaltrj);
29        closeDist = new ArrayList<Double>(totaltrj);
30        for (int i = 0;i <totaltrj;i++) {
31            closeID.add(-1);
32            closeDist.add(Double.MAX_VALUE);
33        }
34    }
35
36 }

```



## LAMPIRAN B

### HASIL EKSPERIMEN

Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4