**RANCANG BANGUN SISTEM INFORMASI PENERIMAAN MAHASISWA MAGANG BERBASIS WEB PADA RAJAWALI TELEVISI (RTV)**

Sebagai Syarat Untuk Melanjutkan Tugas Akhir

Pada Program Studi Sistem Informasi



Disusun Oleh :

Bayu Trianto (15108033)

Nabila Ratna Febriani (15108016)

**PROGRAM STUDI SISTEM INFORMASI**

**FAKULTAS INDUSTRI KREATIF DAN TELEMATIKA**

**UNIVERSITAS TRILOGI**

**2019**

**LEMBAR PERNYATAAN MAHASISWA**

Bersama ini Kami Menyatakan Bahwa isi yang terkandung dalam Kerja Praktek ini dengan judul:

**RANCANG BANGUN SISTEM INFORMASI PENERIMAAN MAHASISWA MAGANG BERBASIS WEB PADA RAJAWALI TELEVISI (RTV)**

Adalah murni merupakan hasil penelitian dan pemikiran kami sendiri.

Demikian pernyataan ini saya buat dan siap menerima konsekuensi apapun dimasa yang akan datang, bila ternyata Kerja Praktek ini merupakan Salinan ataupun contoh karya - karya yang telah dibuat/diterbitkan sebelum tanggal Laporan Kerja Praktek ini.

Jakarta, Januari 2019

Bayu Trianto Nabila Ratna Febriani

**LEMBAR PERSETUJUAN DOSEN PEMBIMBING**

**RANCANG BANGUN SISTEM INFORMASI PENERIMAAN MAHASISWA MAGANG BERBASIS WEB PADA RAJAWALI TELEVISI (RTV)**

Nama : Bayu Trianto , Nabila Ratna Febriani

Nim : 15108033 . 15108016

Jurusan : Sistem Informasi

Telah disetujui Pada Tanggal, Februari 2019

**Oleh:**

**Dosen Pembimbing**

**(Nina Sariana, S.Kom.,MMSI,)**

**Mengetahui,**

**Kepala Program Studi Sistem Informasi**

**(Rudi Setiawan, S.Kom., M.Cs)**

# LEMBAR PENGESAHAN JURUSAN

**RANCANG BANGUN SISTEM INFORMASI PENERIMAAN MAHASISWA MAGANG BERBASIS WEB PADA RAJAWALI TELEVISI (RTV)**

Nama : Bayu Trianto, Nabila Ratna Febriani

NIM : 15108033 , 15108016

Jurusan : Sistem Informasi

Telah Diuji Pada Tanggal, 1 Maret 2019

**Oleh Penguji :**

**1.**

**2.**

**LEMBAR ABSENSI BIMBINGAN KERJA PRAKTEK**

**“RANCANG BANGUN SISTEM INFORMASI PENERIMAAN MAHASISWA MAGANG BERBASIS *WEB* PADA RAJAWALI TELEVISI (RTV)”**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Nama | | : Bayu Trianto , Nabila Ratna Febriani | |
|  | NIM | | : 15108033 , 15108016 |  |
|  | Jurusan | | : Sistem Informasi |  |
|  |  |  | |  |
| **NO.** | **TANGGAL** | **CATATAN DOSEN PEMBIMBING** | | **PARAF** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Dosen Pembimbing**

**Nina Sariana S Kom., MM SI**

**KATA PENGANTAR**

Puji syukur kami panjatkan kehadirat Tuhan yang Maha Esa yang telah memberikan berkat dan rahmat-Nya kepada kami sehingga kami dapat menyelesaikan Laporan Kerja Praktek dengan judul **“RANCANG BANGUN SISTEM INFORMASI PENERIMAAN MAHASISWA MAGANG BERBASIS *WEB* PADA RAJAWALI TELEVISI (RTV)”.**

Pada kesempatan ini kami ingin menyampaikan rasa terima kasih kepada:

1. Tuhan Yang Maha Esa.
2. Bapak Homa P. Harahap, S.Si., M.M selaku Dekan Fakultas Industri Kreatif dan Telematika Universitas Trilogi.
3. Bapak Rudi Setiawan, S.Kom., M.Cs selaku Ketua Program Studi Sistem Informasi Universitas Trilogi.
4. Ibu Nina Sariana, S.Kom.,MMSI selaku Dosen Pembimbing Praktek Kerja Lapangan.
5. Ibu Elisabeth Soelandjani selaku pembimbing Praktek Kerja Lapangan Divisi/Dapartment : Programming/ Programming Research di Rajawali Televisi (RTV)
6. Orang tua, yang selalu memberikan semangat dan motivasi penulis.
7. Seluruh rekan-rekan yang turut membantu dalam penyelesaian baik program maupun penulisan laporan kerja praktek ini.

Kami menyadari bahwa Laporan Kerja Praktek ini masih jauh dari kata sempurna, baik dalam penyusunan, tata bahasa, maupun dalam hal penyajian data-data yang dilaporkan. Oleh karena itu, kritik dan saran yang bersifat membangun sangat kami harapkan. Atas semua perhatian dari segala pihak yang telah membantu kami dalam menyusun laporan kerja praktek ini, kami ucapkan terima kasih.

Jakarta, Januari 2019

Bayu Trianto Nabila Ratna Febriani **DAFTAR ISI**

**DAFTAR GAMBAR**

**DAFTAR TABEL**

**ABSTRAK**

Dalam penulisan laporan kerja praktek ini penulis akan membahas tentang Rancang Bangun Sistem Informasi Penerimaan Mahasiswa Magang Berbasis Web Pada Rajawali Televisi (RTV), pembuatan sistem ini di lakukan karena belum adanya format untuk penerimaan mahasiswa magang sehingga di buatlah sistem ini. Sistem laporan ini dibangun menggunakan bahasa pemograman PHP menggunakan *Framework* *Laravel* dan *Database* MySQL. Laporan yang penulis buat antara lain untuk user mengetahui tentang penerimaan mahasiswa magang di Rajawali Televisi (RTV).

**Kata Kunci:** *Web, Requirement*

**BAB I**

**PENDAHULUAN**

**1.1 Latar Belakang Masalah**

Kemajuan teknologi dan informasi beberapa tahun terakhir berkembang demikian pesat. Hal tersebut salah satunya ditandai dengan penerapan sistem yang serba terkomputerisasi di sejumlah instansi maupun perusahaan. Penerapan sistem yang serba terkomputerisasi memberikan kemudahan baik bagi perusahaan yang bersangkutan maupun bagi pengguna dalam mencari informasi yang diinginkan secara cepat dan tepat.

Sebuah sistem informasi berbasis *website* terus memberikan peran aktif terhadap suatu organisasi, kelompok usaha, maupun perusahaan dalam menajalankan kegiatan atau aktifitas bisnis mereka. Salah satu sistem yang kami buat adalah sistem penerimaan mahasiswa magang pada Rajawali Televisi atau lebih dikenal dengan nama RTV. Sistem ini dapat mempermudah dalam perimaan mahasiswa magang agar menjadi lebih efektif dan efisien.

**RTV** (singkatan dari **Rajawali Televisi**) adalah stasiun televisi nasional yang mengedepankan unsur pendidikan dalam setiap sajian programnya. Dengan semangat inovasi, ingin berbagi nilai positif melalui produksi hiburan dan informasi akurat dengan tujuan untuk membangun masyarakat Indonesia agar menjadi lebih maju. Hampir semua bidang sudah menggukanan sistem untuk memudahkan pekerjaannya mengolah data yang ada. Akan tetapi masih ada pula bidang yang mengelola data belum menggunakan sistem sehingga data yang diolah kurang kurang efesien dalam proses pengolahan datanya. Dengan melihat kekurangan pengolahan data tersebut, maka dibutuhkan sebuah sistem baru yang mampu melakukan pengolahan data secara cepat, akurat dan dapat melakukan pembaharuan *(up to date)* sesuai dengan perkembangan kebutuhan informasi.

Sistem penerimaan mahasiswa magang masih menggunakan sistem yang manual yakni dengan melakukan panggilan melalui telefon biasa. Sistem manual tersebut, tentu akan berdampak kepada pengolahan data yang kurang efisien, yakni terkait dengan kurang nya informasi kepada mahasiswa magang apabila mahasiswa tersebut sedang ada gangguan dengan telefon nya. Dengan melihat kekurangan yang dimiliki oleh RTV dalam hal pemanfaatan teknologi informasi, maka dibutuhkan satu sistem informasi yang mampu dijadikan sebagai sistem yang terkomputerisasi.

Berdasarkan latar belakang diatas, maka kami mengembangkan sistem berbasis web untuk mempermudah dalam penerimaan mahasiswa magang, mulai dari divisi yang sedang membutuhkan mahasiswa magang, jumlah kuota yang masih ada, hingga status mahasiswa magang tersebut.

## **1.2 Identifikasi Masalah**

Berdasarkan uraian pada latar belakang, dapat diketahui bahwa identifikasi masalah yang dapat diambil adalah belum terdapatnya sistem informasi terkomputerisasi dalam penerimaan mahasiswa magang di Rajawali Televisi (RTV).

## **1.3 Rumusan Masalah**

Berdasarkan identifikasi masalah diatas, dapat ditarik rumusan masalah yakni bagaimana merancang sistem informasi terkomputerisasi sebagai media penerimaan mahasiswa magang yang terkait dengan data divisi, kuota, dan status pendaftaran mahasiswa magang pada Rajawali Televisi (RTV).

## **1.4 Batasan Masalah**

Adapun Batasan masalah dalam penelitian ini adalah bagaimana merancang dan membangun sistem informasi terkomputerisasi yang mampu digunakan sebagai media media penerimaan mahasiswa magang yang terkait dengan data divisi, kuota, dan status mahasiswa magang pada Rajawali Televisi (RTV).

## **1.5 Tujuan Penelitian**

Tujuan dari penelitian ini adalah untuk merancang sistem informasi penerimaan mahasiswa magang yang terkait dengan data divisi, kuota, dan status mahasiswa magang pada Rajawali Televisi (RTV).

**1.6 Manfaat Penelitian**

Manfaat dari penelitian ini adalah sebagai berikut:

1. Bagi Penulis.

Dapat menambah pengetahuan dan wawasan bagi penulis dalam hal pembuatan sistem informasi.

1. Bagi Perusahaan

Dengan adanya sistem informasi, maka akan membantu dalam hal penerimaan mahasiswa magang yang terkait dengan data divisi, kuota, dan status mahasiswa magang pada Rajawali Televisi (RTV).

1. Bagi Universitas Trilogi.

Dengan hasil penelitan ini, semoga dapat bermanfaat dan dijadikan sebagai pedoman bagi mahasiswa/mahasiswi Universitas Trilogi yang membacanya.

## **1.7 Metodologi Penelitian**

Dalam penelitian ini, sumber data yang dikumpulkan berupa data primer yang diperoleh dengan cara melakukan observasi secara langsung pada perusahaan terkait. Sedangkan untuk data sekunder, data yang digunakan yakni dalam bentuk data yang sudah ada, seperti struktur organisasi, dan lain-lain.

Adapun tahapan-tahapan untuk mendapatkan data yang dibutuhkan penulis menggunkan beberapa metode, yakni sebagai berikut:

1. Metode Observasi.

Merupakan teknik pengumpulan data dimana penulis melakukan pengamatan secara langsung ke objek penelitian untuk mengetahui kegiatan yang dilakukan. Dalam pelaksanannya, penulis melakukan observasi secara langsung terhadap *Human Resources Development* Rajawali Televisi (RTV) dan juga terhadap situasi di tempat tersebut.

1. Metode Wawancara.

Wawancara merupakan Teknik pengumpulan data yang dilakukan melalui tatap muka dan tanya jawab langsung antara pengumpul data mamupun peneliti terhadap narasumber atau sumber data.

Pengumpulan data dilakukan dengancara bertanya langsung kepada pegawai *Human Resources Development* Rajawali Televisi (RTV) tentang bagaimana proses penerimaan mahasiswa magang yang telah dilakukan selama ini.

1. Metode Studi Pustaka

Melakukan studi pustaka dengan cara mengumpulkan sumber-sumber berupa buku, jurnal, maupun informasi dari intenet sebagai landasan teori dalam pelaksanaan kegiatan penyusunan laporan.

## **1.8 Sistematika Penulisan**

Secara garis besar, penulisan laporan kerja praktek ini terdiri dari beberapa bab, adapun sistematika serta susunan dari penulisan laporan ini adalah sebagai berikut:

**BAB I PENDAHULUAN**

Bab ini berisi tentang Latar Belakang Masalah, Identifikasi Masalah, Rumusan Masalah, Batasan Masalah, Tujuan Laporan, Manfaat Laporan, Metodologi Penelitian, dan Sistematika Penulisan.

**BAB II LANDASAN TEORI**

Bab ini menjelaskan tentang teori yang digunakan sebagai panduan dasar dan landasan ilmiah dalam pengembangan sistem ini.

**BAB III METODE PENELITIAN**

Dalam bab ini menguraikan metode penelitian yang digunakan penulis, baik itu metode pengumpulan data maupun metode pengembangan sistem dan kerangka pemikiran.

**BAB IV ANALISIS SISTEM**

Dalam bab ini menganalisis objek penelitian, analisis kebutuhan sistem, serta perancangan sistem yang akan dibuat. Dalam implementasi diuraikan tentang pemilihan bahasa pemrograman yang digunakan, perangkat keras yang diperlukan, pemrograman dan pengujian aplikasi.

**BAB V PENUTUP**

Bab ini berisi mengenai kesimpulan dari pokok-pokok pembahasan penelitian ini, serta saran yang berguna untuk pengembang peneliti berikutnya dan pada perusahaan.

**DAFTAR PUSTAKA**

Pada bagian ini berisikan tentang sumber-sumber yang mendukung dalam penulisan laporan kerja praktek ini dari beberapa buku, jurnal, *website* maupun sumber informasi lainnya.

# 

# BAB II LANDASAN TEORI

## **2.1 Teori Dasar**

### **2.1.1 Rancang Bangun**

Perancangan atau rancang bangun merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dan sebuah sistem kedalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem di implementasikan. Sedangkan pengertian pembangunan atau bangun sistem adalah kegiatan menciptakan sebuah sistem baru atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun hanya sebagian (Pressman, 2014).

Perancangan sistem adalah menentukan proses dan data yang diperlukan oleh sistem baru. Jika sistem tersebut berbasis komputer, rancangan dapat menyertakan spesifikasi jenis peralatan yang akan digunakan (Sutabri, 2014)

Dari penjelasan diatas dapat disimpulkan bahwa rancang bangun sistem merupakan sebuah kegiatan menerjemahkan hasil analisa kedalam bentuk paket perangkat lunak dengan menentukan proses dan data yang diperlukan untuk menciptakan sebuah sistem baru atau memperbaiki sistem yang telah ada.

## **2.2 Sistem Informasi**

Sistem informasi yaitu suatu sistem yang menyediakan informasi untuk manajemen dalam mengambil keputusan dan juga untuk menjalankan operasional perusahaan, di mana sistem tersebut merupakan kombinasi dari orang-orang, teknologi informasi dan prosedur-prosedur yang tergorganisasi. Biasanya suatu perusahan atau badan usaha menyediakan semacam informasi yang berguna bagi manajemen. Berikut penjelasan tentang sistem informasi:

### **2.2.1 Pengertian Sistem**

Sistem adalah suatu panduan yang terdiri dari beberapa unsur yang tergabung satu sama lain agar mempermudah laju aliran informasi, energi, maupun materi hingga dapat mencapai tujuan tertentu, definisi sistem pada berbagai bidang berbeda-beda, tetapi meskipun istilah sistem yang digunakan bervariasi, semua sistem pada bidang-bidang tersebut mempunyai beberapa persyaratan umum, yaitu sistem harus mempunyai elemen, lingkungan, interaksi antar elemen, interaksi antara elemen dengan lingkungannya, dan yang terpenting adalah sistem harus mempunyai tujuan yang akan dicapai.

### Sistem pada dasarnya adalah sekelompok unsur yang erat berhubungan satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu (Sutabri, 2014)

### **2.2.2 Pengertian Informasi**

Informasi adalah sejumlah data yang telah diolah melalui pengolahan data dalam rangka menguji tingkat kebenarannya dan ketercapaiannya sesuai kebutuhan, sedangkan data merupakan bahan mentahnya (Santoso, 2017).Sistem informasi adalah komponen-komponen yang saling berkaitan yang bekerja bersama-sama untuk mengumpulkan, mengolah, menyimpan, dan menampilkan informasi untuk mendukung pengambilan keputusan, koordinasi, pengaturan, analisa, dan visualisasi pada sebuah organisasi.

**2.2.3 Pengertian Sistem Informasi**

Sistem informasi yaitu suatu sistem yang menyediakan informasi untuk manajemen dalam mengambil keputusan dan juga untuk menjalankan operasional perusahaan, di mana sistem tersebut merupakan kombinasi dari orang-orang, teknologi informasi dan prosedur-prosedur yang tergorganisasi. Sistem informasi dapat didefinisikan sebagai suatu susunan dari orang, data, proses dan teknologi informasi yang saling berhubungan untuk mengumpulkan, memproses, menyimpan dan menyediakan keluaran informasi yang diperlukan untuk mendukung suatu organisasi (Yusdiardi, 2014).

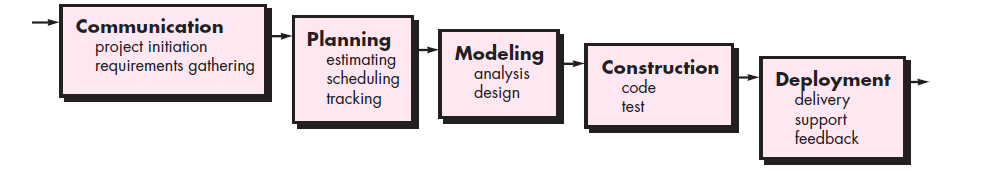
**2.3 Aplikasi Berbasis *Web***

Suatu Aplikasi Berbasis Web merupakan suatu [aplikasi](https://id.wikipedia.org/wiki/Aplikasi) yang diakses menggunakan [penjelajah web](https://id.wikipedia.org/wiki/Penjelajah_web) melalui suatu [jaringan](https://id.wikipedia.org/wiki/Jaringan_(komputer)) seperti [Internet](https://id.wikipedia.org/wiki/Internet) atau [intranet](https://id.wikipedia.org/wiki/Intranet). Dan juga merupakan suatu aplikasi [perangkat lunak](https://id.wikipedia.org/wiki/Perangkat_lunak) komputer yang dikodekan dalam [bahasa](https://id.wikipedia.org/wiki/Bahasa_pemrograman) pemrograman yang didukung penjelajah web (seperti [Java](https://id.wikipedia.org/wiki/Java), [Java Script](https://id.wikipedia.org/wiki/JavaScript), [PHP](https://id.wikipedia.org/wiki/PHP), [Python](https://id.wikipedia.org/wiki/Python_(bahasa_pemrograman)), dll) dan bergantung pada penjelajah tersebut untuk menampilkan aplikasi . Aplikasi *web* mempunyai sifat untuk dapat diakses dimana saja dan kapan saja, membuat banyak aplikasi-aplikasi bermunculan dan dimanfaatkan oleh orang banyak. Baik itu aplikasi yang berfungsi untuk membantu pekerjaan perkantoran atau aplikasi yang bersifat hiburan, jejaring sosial, dan lain-lain (Faisal, 2014).

### **2.4 Metode *Waterfall***

Metode *Waterfall* adalah merupakan model yang paling populer diantara model pendekatan dan bekerja secara linier dan berurutan di lain pihak model ini merupakan model yang paling awal terbentuk. Pendekatan ini mengikuti suatu paradigma yang dikenal dengan *documentation driver paradigm*, harapan nya adalah bahwa disetiap proses pengembangan sistem, setiap langkah yang dilakukan akan dikomentasi atau ditulis dalam bentuk spesifikasi dalam laporan yang detail. Pada model ini memiliki beberapa kelemahan antara lain:

1. Spesifikasi dalam perancangan sistem harus sudah pasti dan adanya model sistem yang berubah-rubah menjadi tidak stabil.
2. Pekerjaan merancang dan pembuatan kode sering tidak konsisten dengan spesifikasi, misalnya adanya komponen sistem yang hilang dan kebutuhan pengembangan yang tidak diharapkan
3. Performa *system* tidak dapat diuji sampai dengan proses coding benar-benar selesai dilakukan. Dapat dilihat pada Gambar 2.1 sebagai berikut :



Gambar 2.1 Metode Waterfall

Kelebihan menggunakan metode *waterfall* adalah prosesnya lebih terstruktur, hal ini membuat kualitas *software* menjadi baik dan tetap terjaga. Dari sisi pengguna juga lebih menguntungkan, karena dapat merencanakan dan menyiapkan kebutuhan data dan proses yang diperlukan sejak awal. Penjadwalan juga menjadi lebih menentu, karena jadwal setiap proses dapat ditentukan secara pasti. Sehingga dapat dilihat jelas target penyelesaian pengembangan program.

Dengan dilakukannya proses pengerjaan secara berurutan maka dapat dilihat pula perkembangan pada setiap tahap secara pasti. Dari sisi lain, metode ini merupakan jenis metode yang pendokumentasian pengerjaannya lengkap sehingga proses pemeliharaan sistem dapat dilakukan dengan mudah. Metode *waterfall l*ini mempunyai tahapan-tahapan sebagai berikut:

a. *Requirements analysis and definition*

Layanan sistem, kendala, dan tujuan ditetapkan oleh hasil konsultasi dengan pengguna yang kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

b. *System and software design*

Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.

c. *Implementation and unit testing*

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

d. *Integration and system testing*

Unit-unit individu program atau program digabung dan diuji sebagai sebuah sistem lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak. Setelah pengujian, perangkat lunak dapat dikirimkan ke *customer*.

e. *Operation and maintenance*

Biasanya (walaupun tidak selalu), tahapan ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata. *Maintenance* melibatkan pembetulan kesalahan yang tidak ditemukan pada tahapan-tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai kebutuhan baru.

Pada prinsipnya, setiap tahapan di metode *Waterfall* menghasilkan satu atau lebih dokumen yang sudah disetujui. Tahap berikutnya tidak dapat dimulai sebelum tahapan sebelumnya selesai. Dalam tataran praktis, tahapan-tahapan tersebut saling tumpang tindih (*overlap*) dan memberikan informasi satu sama lain.

## **2.5 *Unified Modelling Language* (UML)**

*Unified Modeling Language* (UML) adalah tujuan umum, perkembangan, bahasa pemodelan di bidang rekayasa perangkat lunak, yang dimaksudkan untuk menyediakan cara standar untuk memvisualisasikan desain sistem.

Menurut Sukamto dan Shalahuddin (2013:133), UML (Unified Modeling Language) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Contoh UML:

1. *Use case Diagram*
2. *Class Diagram*
3. *Activity Diagram*
4. *Sequence diagram*
   * 1. ***Use Case Diagram***

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merepresentasikan sebuah proses tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Salahuddin, 2016)**.** *use case diagram* berguna tiga hal:

1. Menjelaskan fasilitas yang ada (*requirement*).
2. Komunikasi dengan klien.
3. Membuat test dari kasus-kasus secara umum.

Simbol-simbol yang dipakai dalam *use case* *diagram* adalah yang tertera dalam Tabel 2.5.1 berikut:

Tabel 2. 5. 1 Simbol dan Deskripsi Use Case diagram

|  |  |
| --- | --- |
| Simbol | Deskripsi |
| *Use Case* | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. |
| Aktor/*actor­*  Nama Aktor | Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri. |
| Asosiasi | Komunikasi antar aktor & *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor. |
| Ekstensi/e*xtend*  <<extend>> | Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang di tambahkan dapat berdiri sendiri walaupun tanpa *use case* tambahan itu. ­ |
| *Include*/*uses*  <<includes>>  <<uses>> | Ada dua sudut pandang yang cukup besar mengenai *include* di *use case :*   * *Include* berarti *use case* yang ditambahkan akan selalu dipanggil saat *use case* tambahan dijalankan. * *Include* berarti *use case* yang tambahan akan selalu melakukan pengecekan apakah *use case* yang ditambahkan telah dijalankan sebelum *use case* tambahan dijalankan. |

**2.5.2  *Class Diagram***

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class* *diagram* menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok :

1. Nama (dan *stereotype*).
2. Atribut.
3. Metode.

Atribut dan metoda dapat memiliki salah satu sifat berikut :

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

Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem(Sukamto dan Shalahuddin 2013).

Terdapat relasi antar *class* dalam *class diagram,* yakni sebagai berikut:

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas”).
3. Pewarisan, yaitu hubungan hirarkis antar *class. Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence* *diagram* yang akan dijelaskan kemudian.
   * 1. ***Activity* *Diagram***

Diagram aktivitas atau *activity diagram* menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Sukamto dan Shalahuddin  2013:161)

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

* 1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
  2. Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
  3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
  4. Rancangan menu yang ditampilkan pada perangkat lunak.

Simbol-simbol yang dipakai dalam *Activity* *diagram* adalah yang tertera dalam Tabel 2.5.3 berikut:

Tabel 2. 5.3 Simbol dan Deskripsi Activity Diagram

|  |  |
| --- | --- |
| Simbol | Deskripsi |
| Status Awal | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| Aktor/*actor­*  Aktifitas | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| Percabangan/*decision* | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| Penggabungan*/join* | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| Status Akhir | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.­ |
| *Swimlane*  Nama Swimlane | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

* + 1. ***Sequence* *Diagram***

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

*Sequence* *diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari *trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada *fase* desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation* *bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*.

Simbol-simbol yang dipakai dalam *Sequence diagram* adalah yang tertera dalam Tabel 2.5.4 sebagai berikut:

Tabel 2. 5.4 Simbol dan Deskripsi Sequence

|  |  |
| --- | --- |
| Simbol | Deskripsi |
| Aktor/*actor­*  Nama Aktor | Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri. |
| *Object*  *AnObject*: *a Class* | Berpartisipasi secara berurutan dengan mengirim dan / atau menerima pesan yang ditempatkan diatas diagram. |
| *Execution*  *Occurrence* | Menyatakan objek dalam keadaan aktif dan berinteraksi pesan. |
| *Message* | Pesan yang menggambarkan komunikasi yang terjadi antar objek |
| *Message (return)* | Pesan yang dikirim untuk diri sendiri secara langsung |
| *Lifeline* | Menyatakan kehidupan suatu objek |

# BAB III

# METODE PENELITIAN

## **Metode Pengumpulan Data**

## Metode pengumpulan data adalah metode metode atau hal hal yang dilakukan penulis dalam mendapatkan data untuk keperluan dan kepentingan penulisan laporan ini. Berikut beberapa teknik yang dilakukan penulis dalam mendapatkan data:

### **Observasi (Pengamatan)**

Observasi merupakan teknik pengumpulan data yang dilakukan melalui pengamatan terhadap suatu objek penelitian secara langsung. Observasi dilakukan untuk memperoleh informasi yang terjadi secara nyata dengan melakukan pengamatan atau peninjauan langsung di Rajawali Televisi (RTV).

### **Wawancara (*interview*)**

Wawancara (*interview*) merupakan proses tanya jawab secara langsung dengan sumber data atau pihak-pihak yang berkepentingan yang berhubungan dengan penelitian. Teknik wawancara ini penulis gunakan untuk mendapatkan data dan informasi yang berkaitan dengan prosedur pengolahan nilai siswa di Rajawali Televisi (RTV) .

Hasil serangkaian wawancara pada bagian-bagian yang berhubungan dengan masalah yang terkait, penulis melakukan wawancara kepada Ibu Elisabeth Soelandjani selaku pembimbing Praktek Kerja Lapangan dan Ibu Sofi selaku HRD . Berdasarkan hasil observasi dan wawancara yang penulis lakukan, penulis mendapatkan informasi mengenai prosedur dan proses penerimaan mahasiswa magang di Rajawali Televisi (RTV) dan permasalahan-permasalahan yang berhubungan dengan proses tersebut. Hasil dari wawancara nantinya akan di lampirkan oleh penulis.

### **Studi Pustaka**

Pengumpulan data dari buku-buku yang sesuai dengan tema permasalahan. Misalnya, pengumpulan teori-teori mengenai perancangan sistem informasi pengolahan untuk penyusunan laporan. Dengan menggunakan studi pustaka maka penyusun mendapatkan data yang lebih akurat dan dapat dipertanggung jawabkan.

### **Studi Literatur**

Pengumpulan data dari literatur-literatur dan dokumentasi dari internet, diktat, atau sumber informasi lain yang sesuai dengan materi sistem informasi penerimaan mahasiswa magang di Rajawali Televisi (RTV)

## **Metode Penelitian**

Menurut Pressman (2015), model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Nama model ini sebenarnya adalah “*Linear Sequential Model*”. Model ini sering disebut juga dengan “*classic life cycle*” atau metode *waterfall*. Model ini termasuk ke dalam model *generic* pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh *Winston Royce* sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *Software Engineering (SE)*. Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Fase-fase dalam *Waterfall* Model menurut referensi Pressman :

### **Fase *Communication* (*Project Initiation & Requirements Gathering*)**

Pada fase ini sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan *customer* demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi *software*. Pengumpulan data-data tambahan bisa juga diambil dari jurnal, artikel, dan internet. Pada pengujian aplikasi ini penulis menggunakan pengujian *black-box* yaitu merupakan metode perancangan data uji yang didasarkan ada spesifikasi perangkat lunak. Data uji dibangkitkan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak diuji apakah telah sesuai dengan yang diharapkan.

### **Fase *Planning (Estimating, Scheduling, Tracking)***

Tahap berikutnya adalah tahapan perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resiko-resiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan *tracking* proses pengerjaan sistem.

### **Fase *Modeling (Analysis & Design)***

Tahapan ini adalah tahap perancangan dan permodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur *software*, tampilan *interface*, dan algoritma program. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan. Fase *modelling* dibagi menjadi beberapa bagian, yaitu:

1. Desain Proses (UML)

Perancangan proses-proses yang akan terjadi di dalam sistem dan disajikan dalam bentuk diagram UML yakni dengan membuat beberapa diagram, antara lain: *Use Case Diagram, Class Diagram, Sequence Diagram,* dan *Activity Diagram.*

1. Desain *Database*

Perancangan tabel-tabel yang digunakan untuk menyimpan data berupa daftar *user*, peminjaman, ruangan dan status peminjaman yang kemudian akan di implementasikan ke dalam program.

1. Desain *User Interface*

Perancangan *User Interface* dengan memberikan fasilitas komunikasi antara pemakai dengan aplikasi, memberikan berbagai informasi yang dibutuhkan dan berbagai keterangan yang bertujuan untuk membantu mengarahkan alur penelusuran masalah sampai ditemukannya solusi dari permasalahan tersebut.

### **Fase *Deployment (Delivery, Support & Feedback)***

Tahapan *deployment* merupakan tahapan implementasi *software* ke *customer*, pemeliharaan *software* secara berkala, perbaikan *software*, evaluasi *software* dan pengembangan *software* berdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya. (Pressman, 2015).

### **Fase *Construction (Code & Test)***

Tahapan *construction* ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk/bahasa yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.

# BAB IV

# ANALISIS SISTEM

# Analisis

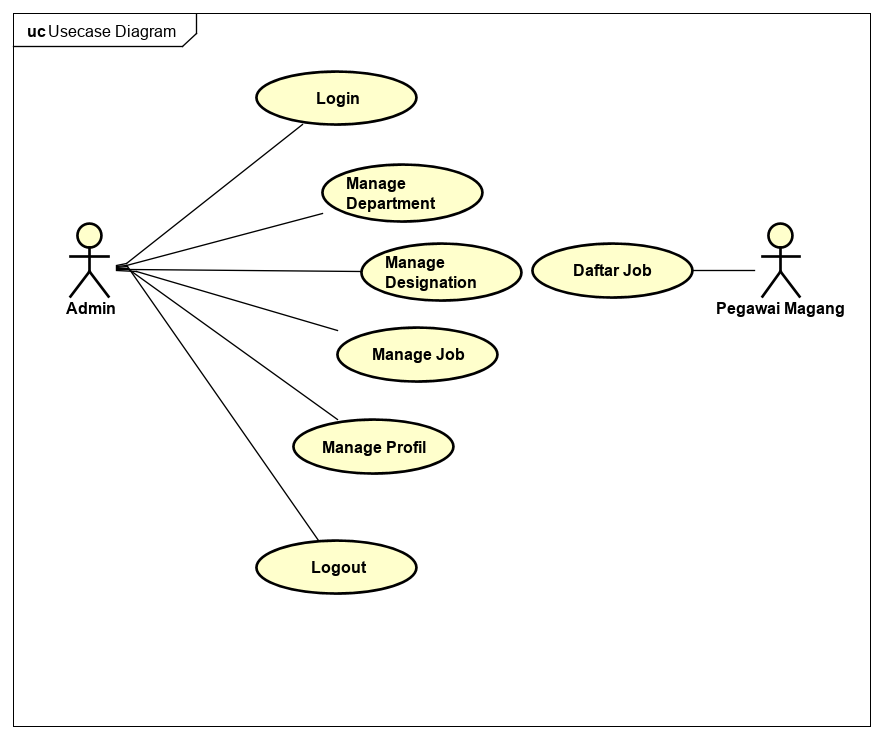
Analisa atau analisis adalah suatu cara untuk mengamati secara lebih mendalam sesuatu hal atau benda dengan cara menguraikan komponen-komponen pembentuknya atau penyusunnya untuk di kaji lebih lanjut. Analisa pada laporan ini dijelaskan lebih rinci sebagai berikut:

1. ***User Requirment***

Analisis kebutuhan *user* dilakukan untuk mengetahui seluruh kebutuhan yang diperlukan oleh *user* terhadap sistem. Analisis kebutuhan *user* dilakukan dengan melakukan diskusi dan wawancara terhadap pihak yang terkait dengan proses penerimaan mahasiswa magang pada Rjawali Televisi (RTV). Adapun daftar kebutuhan *user* terhadap sistem yaitu:

1. Sistem dapat digunakan oleh user dan admin.
2. Admin harus melakukan *login* kedalam sistem ketika akan melakukan pemberian informasi-informasi untuk calon mahasiswa magang.
3. Sistem harus dapat memberikan informasi mengenai data divisi yang membutuhkan mahasiswa magang.
4. Sistem harus dapat memberikan informasi mengenai kuota mahasiswa magang.
5. Sistem harus dapat memberikan informasi mengenai status pendaftran mahasiswa magang.
6. ***Use Case Diagram***

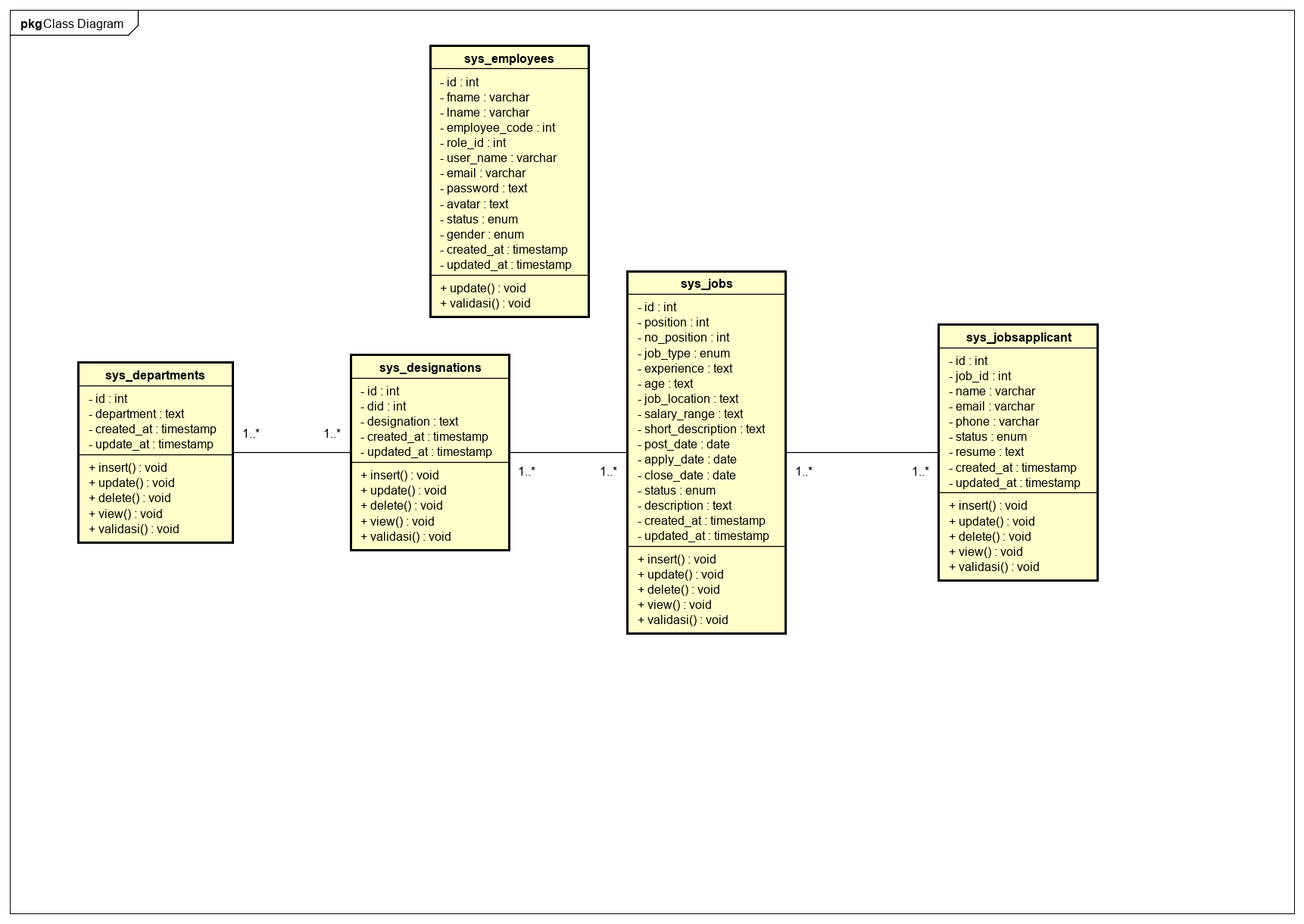
Untuk mengetahui fungsionalitas pada suatu sistem, digunakan penggambaran *use case* diagram. Pada *use case* diagram Sistem Penerimaan Mahasiswa Magangmemiliki 2 (dua) *actor*, yaitu Admin dan Pegawai Magang. *Actor* admin untuk bisa berinteraksi dengan sistem, admin harus memasukkan *username* dan *password* terlebih dahulu. Untuk dapat mengakses semua sistem maka mereka harus mengisi *form* yang terdapat di menu *Login* Dapat dilihat pada gambar 4.1.



Gambar 4.1 *Use Case Diagram* Sistem Penerimaan Mahasiswa Magang

1. ***Class Diagram***

Gambar 4.2 merupakan Class Diagram Sistem Penerimaan Mahasiswa Magang di Rajawali Televisi (RTV). Diagram Ini Digunakan untuk menampilkan beberapa kelas serta paket-paket yang digunakan dalam perancangan Sistem Penerimaan Mahasiswa Magang. Dapat lihat pada gambar 4.2.

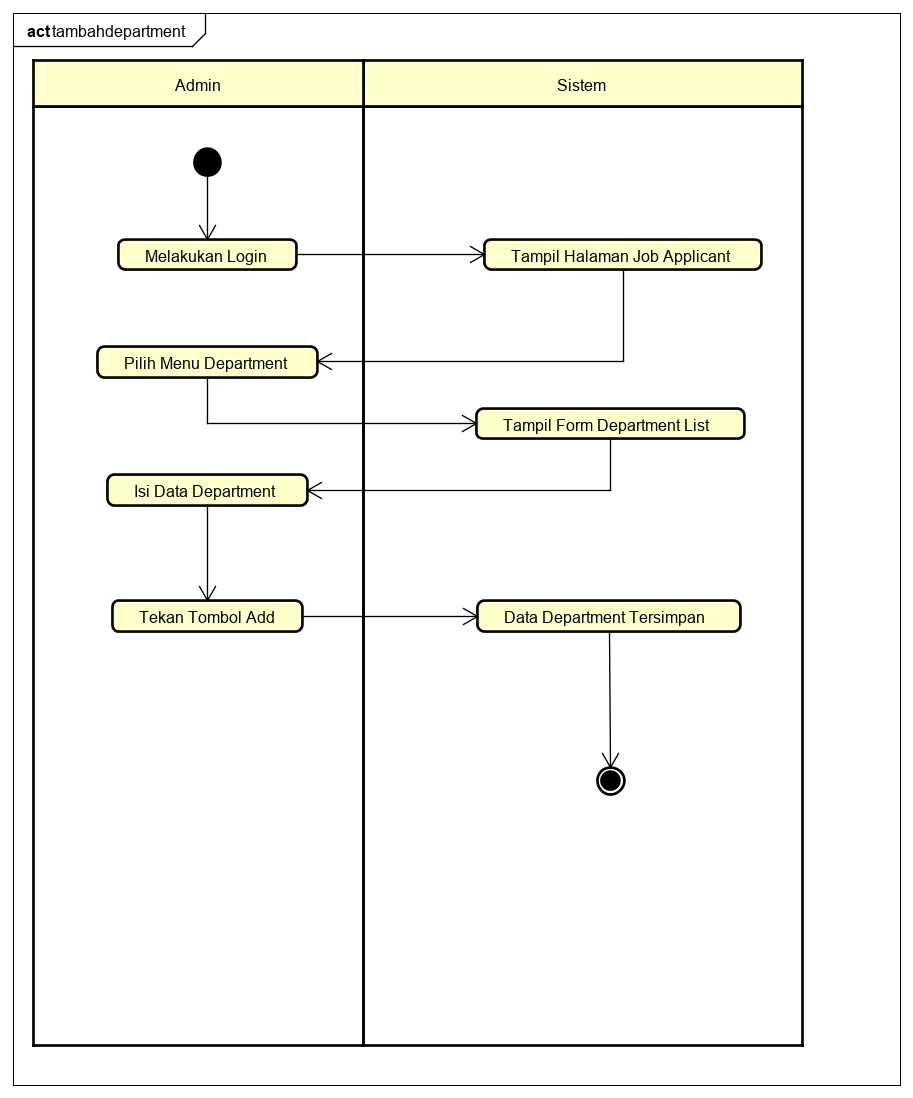


Gambar 4.2 *Class Diagram* SIstem Penerimaan Mahasiswa Magang

1. ***Activity Diagram***

*Activity* *diagram* menggambarkan berbagai alur dari suatu aktivitas dalam sistem yang sedang dirancang untuk mendeskripsikan aktivitas secara umum yang terjadi didalam operasi suatu sistem.

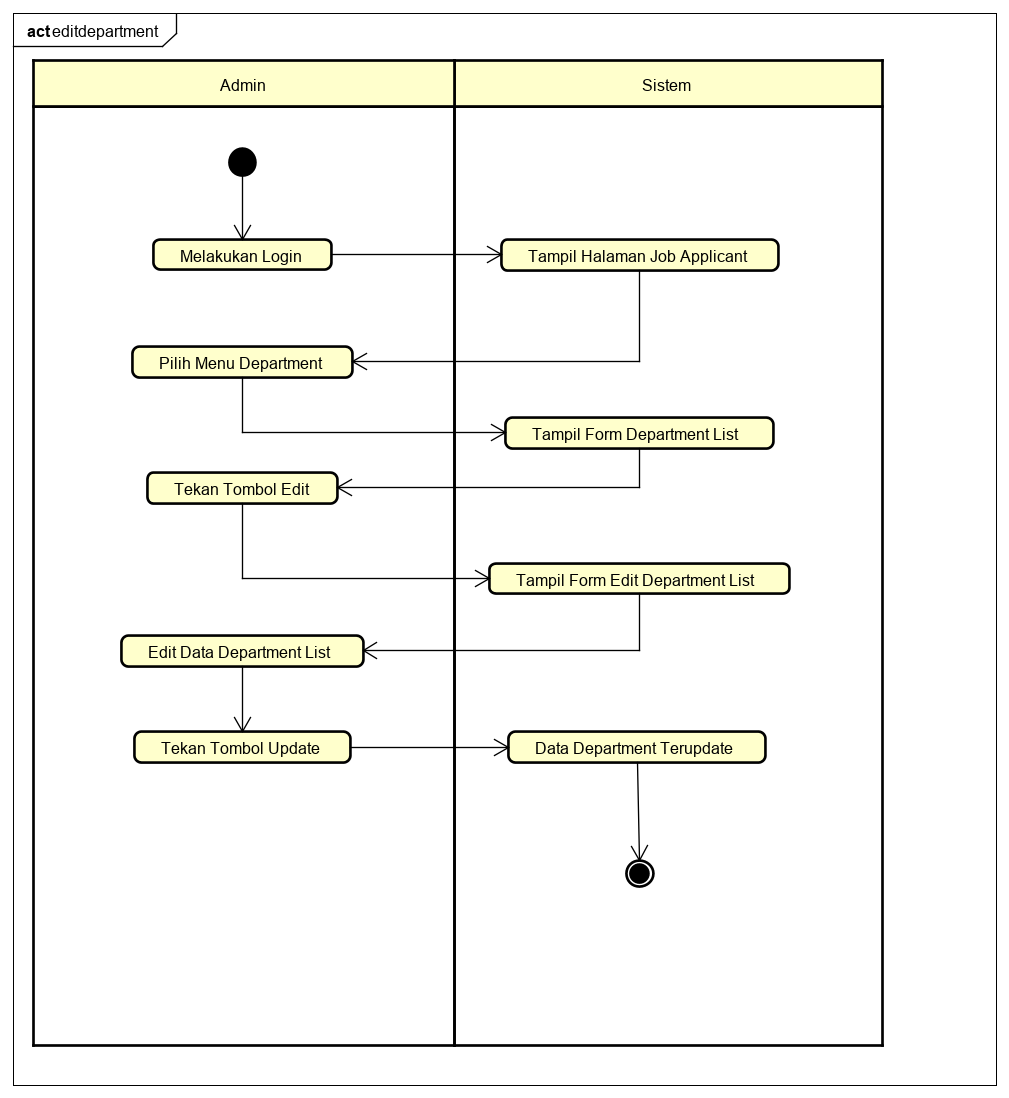
1. *Activity Diagram* Admin Tambah *Department*

**

Gambar 4.3 *Activity Diagram* Admin Tambah *Department*

Gambar 4.3 merupakan *Activity Diagram* admin untuk tambah *department*. Pertama admin harus melakukan *login*, apabila berhasil maka akan menampilkan halaman job, setelah itu pilih menu *department* untuk menampilkan form *department*, lalu mengisi data *department* dan klik tombol add untuk menyimpan data *department* yang baru diisi.

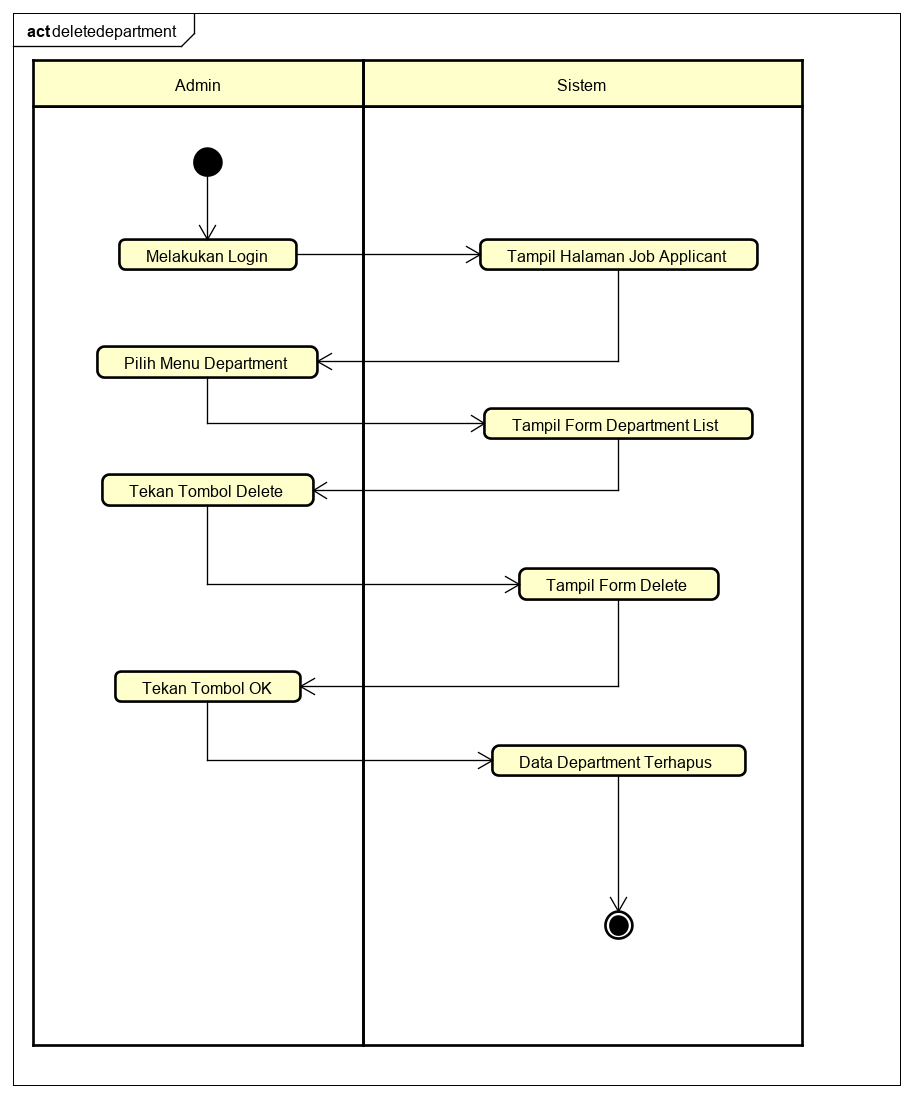
1. *Activity Diagram* Admin *Edit Department*



Gambar 4.4 *Activity Diagram* Admin *Edit* *Department*

Gambar 4.4 merupakan *Activity Diagram* admin untuk *edit department*. Pertama admin harus melakukan *login*, apabila berhasil maka akan menampilkan halaman *job*, setelah itu klik tombol edit untuk menampilkan *form* *edit department*, lalu merubah data *department* dan klik tombol *update* untuk menyimpan data *department* yang baru dirubah.

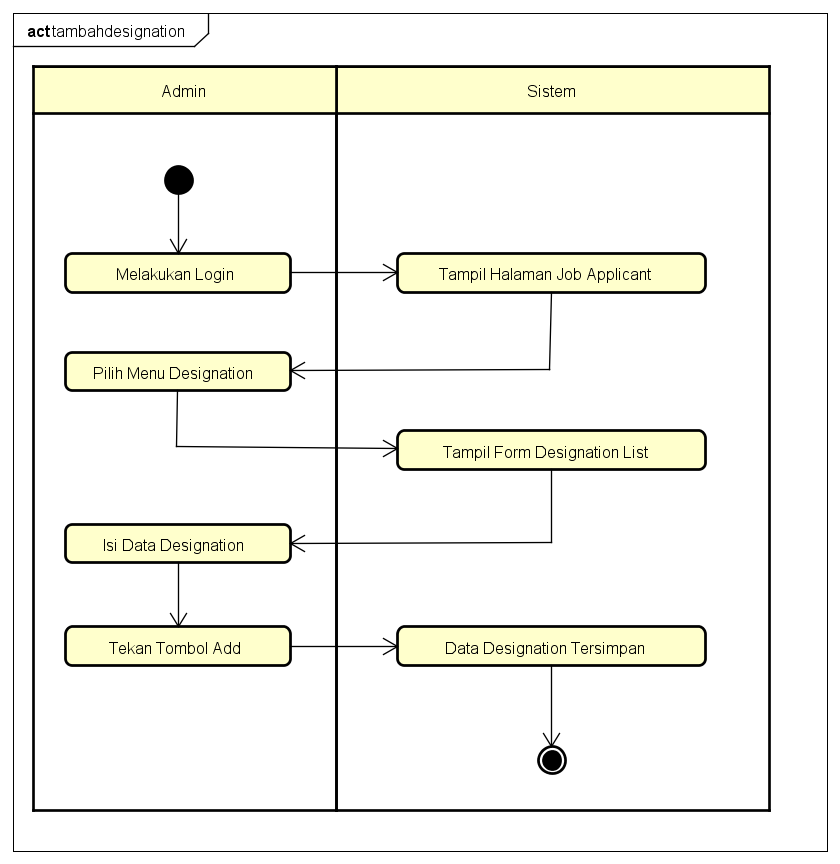
1. *Activity Diagram* Admin *Delete* *Department*



Gambar 4.5 *Activity Diagram* Admin *Delete* *Department*

Gambar 4.5 merupakan *Activity Diagram* admin untuk *delete department*. Pertama admin harus melakukan *login*, apabila berhasil maka akan menampilkan halaman *job*, setelah itu klik tombol *delete* untuk menampilkan *form* *Delete department*, lalu klik OK untuk menghapus data *department* yang ingin dihapus.

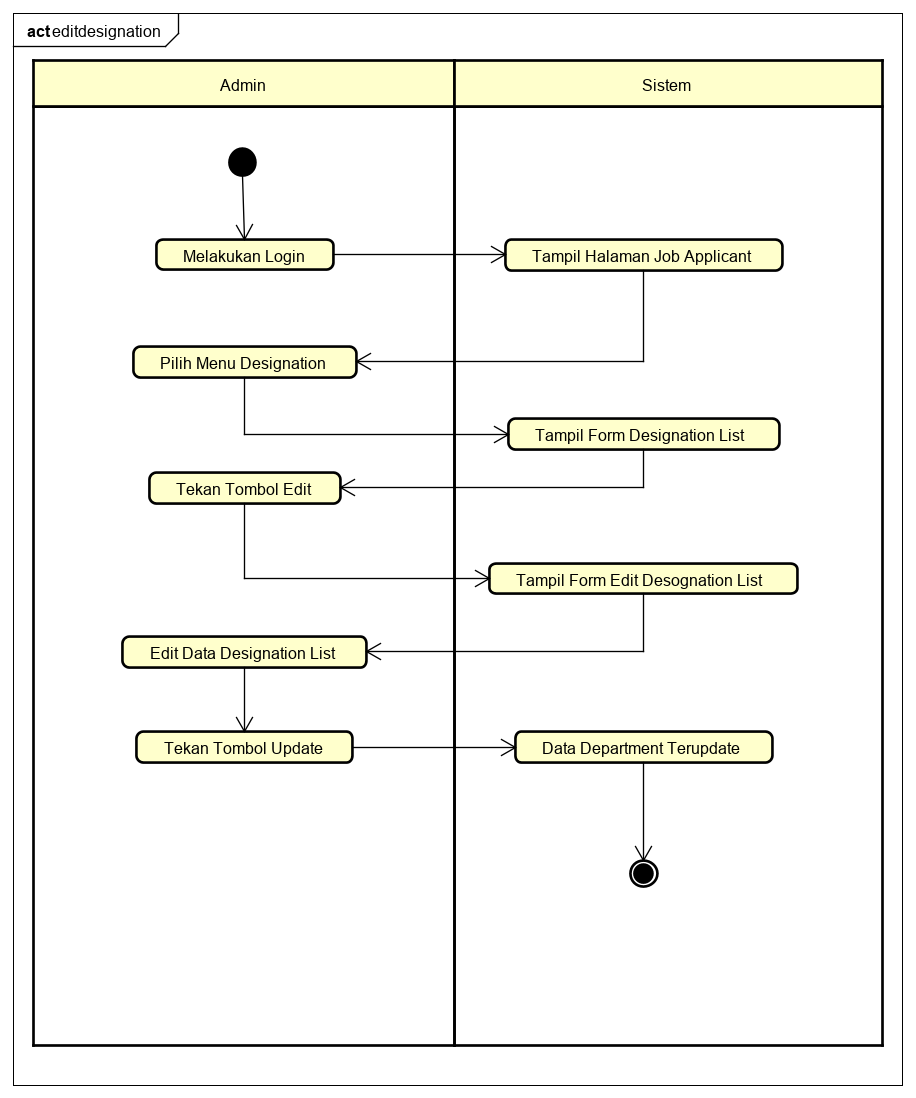
1. *Activity Diagram* Admin Tambah *Designation*

**

Gambar 4.6 *Activity Diagram* Admin Tambah *Designation*

Gambar 4.6 merupakan *Activity Diagram* admin untuk Tambah *Designation*. Pertama admin harus melakukan *login*, apabila berhasil maka akan menampilkan halaman *job*, setelah itu pilih menu *designation* untuk menampilkan form *designation*, lalu isi data dan klik *Add* untuk menyimpan data*.*

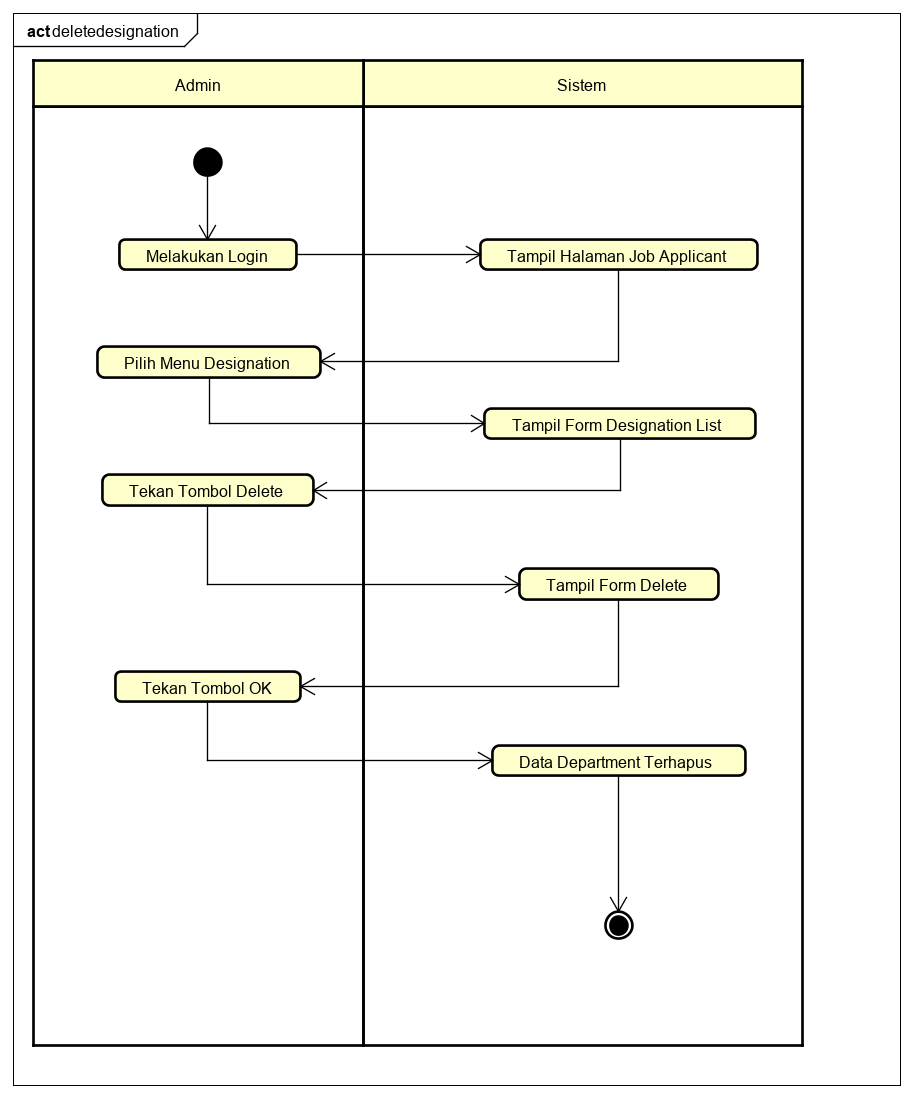
1. *Activity Diagram* Admin *Edit Designation*



Gambar 4.7 *Activity Diagram* Admin *Edit Designation*

Gambar 4.7 merupakan *Activity Diagram* admin untuk *Edit Designation*. Pertama admin harus melakukan *login*, apabila berhasil maka akan menampilkan halaman *job*, setelah itu pilih menu *designation* untuk menampilkan form *designation*, lalu *edit* data dan klik *update* untuk meng*update.*

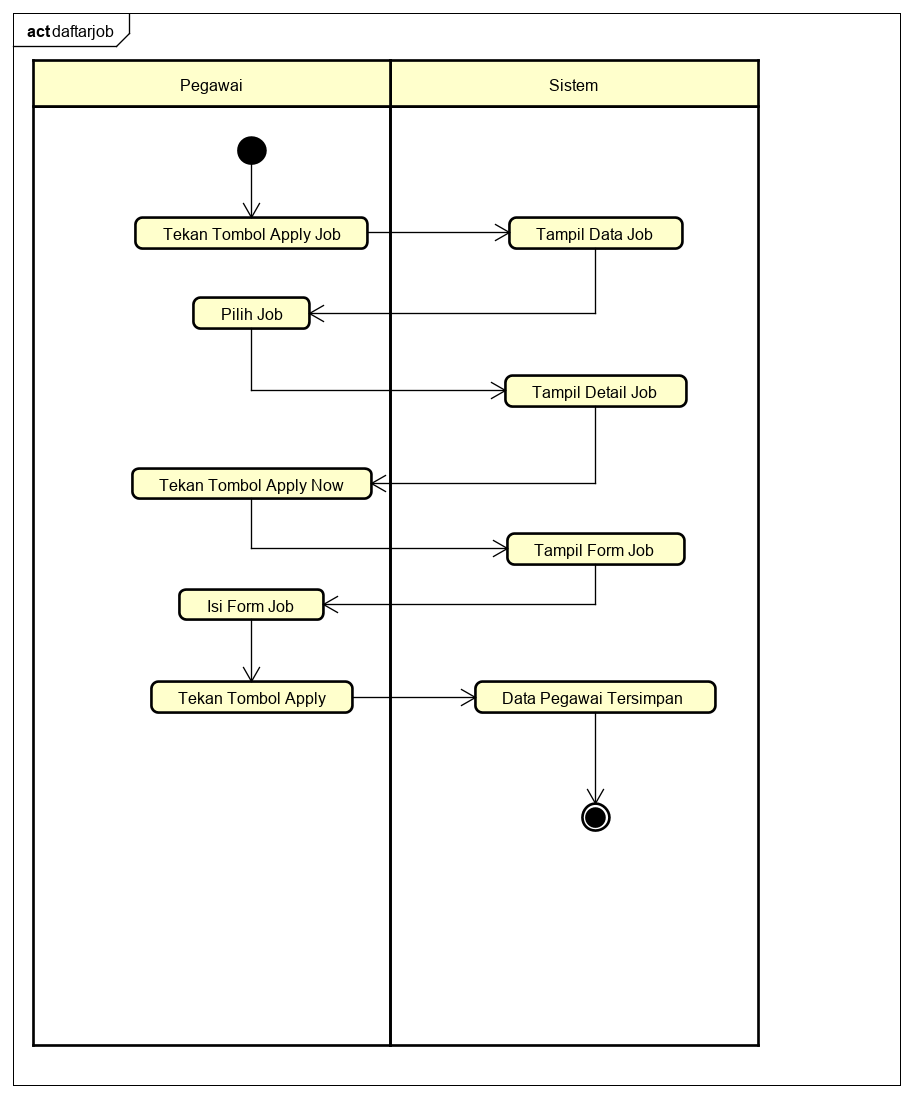
1. *Activity Diagram* Admin *Delete Designation*



Gambar 4.8 *Activity Diagram* Admin *Delete Designation*

Gambar 4.8 merupakan *Activity Diagram* admin untuk *Delete Designation*. Pertama admin harus melakukan *login*, apabila berhasil maka akan menampilkan halaman *job*, setelah itu pilih menu *designation* untuk menampilkan form *designation*, lalu *delete* data dan klik *OK* untuk menghapus data yang ingin di hapus*.*

1. *Activity Diagram* Pegawai Magang Daftar Job

**

Gambar 4.9 *Activity Diagram* Pegawai Magang Daftar Job

Gambar 4.9 merupakan *Activity Diagram* pegawai magang untukdaftar job. Pertama tekan tombol *apply* job untuk menampilkan data job, lalu pilih job untuk menampilkan detail job, setelah itu klik *aplly now* dan menuju tampilan form job, kemudian user akan mengisi *form* job dan menyimpannya dengan menekan tombol *aplly.*

1. **Rancangan *Interface***

Rancangan *Interface* diperlukan untuk menjembatani antara sistem yang dibagun dengan *user*. Berikut merupakan rancangan sistem *interface* penerimaan mahasiswa magang berbasis web pada Rjawali Televisi (RTV).

**BAB V**

**KESIMPULAN DAN SARAN**

## **5.1 Kesimpulan**

Dari rumusan masalah dan analisa permasalahan yang penulis lakukan, laporan kerja praktek ini dapat di simpulkan bahwa Pembuatan aplikasi penjadwalan pada PT. Hadidaya Sejahtera Pratama ini sudah sesuai dengan kebutuhan user/admin dan seluruh laporan perihal penjadwalan keberangkatan logistic perusahaan sudah satu pintu dengan aplikasi ini. Pembuatan aplikasi ini menggunakan Visual Studio 2010 dan dengan menggunakan database SQL Server 2012. Dengan adanya aplikasi ini user/admin dapat bekerja dengan lebih efektif dan efisien karena semua pencatatan jadwal sudah terdapat di aplikasi ini, juga lebih menghemat biaya dan tenaga.

Dari rumusan masalah dan analisa permasalahan yang telah dipaparkan pada bab-bab sebelumnya, maka penulis dapat menarik beberapa kesimpulan sebagai berikut:

1. Dengan adanya aplikasi ini, penjadwalan keberangkatan pengiriman mobil di PT. Hadidaya Sejahtera Pratama menjadi terintegrasi dengan rincian penagihan serta menjadi tidak parsial sehingga menjadikan proses penjadwalan keberangkatan pengiriman mobil dan pembuatan rincian penagihan menjadi lebih efektif dan efisien.
2. Dengan proses penjadwalan keberangkatan pengiriman mobil yang sudah tersistem ini, *user* atau admin dapat dengan mudah mencari data rincian penagihan atau mencari data keberangkatan dengan lebih efektif.

## **5.2 Saran**

Masih terdapat kekurangan dalam aplikasi penjadwalan ini, salah satunya detail transaksi yang belum tercatat secara otomatis dikarenakan setiap order kadang memiliki banyak jadwal. Adapun saran yang penulis berikan untuk pengembangan aplikasi lebih lanjut, yaitu:

* 1. Aplikasi ini dapat terintegrasi langsung dengan *database* berbasis *cloud*, sehingga jika ada kejadian komputer admin mengalami gangguan *harddisk* data di *database* tetap aman.
  2. Aplikasi ini dapat dikembangkan lagi menjadi sebuah aplikasi dengan tampilan *Graphical User Interface* (GUI) yang lebih menarik.
  3. Aplikasi ini dapat digunakan pada perangkat *mobile* atau berbasis android