**PENGEMBANGAN SISTEM INFORMASI *TRACER STUDY* DI SEKOLAH TINGGI TEKNOLOGI GARUT**

**SKRIPSI**

Dibuat sebagai syarat pelaksanaan penelitian mahasiswa program sarjana

**Oleh :**

**Fikri Zakaria Rahman**

**1406054**



**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TINGGI TEKNOLOGI GARUT**

**2019**

# KATA PENGANTAR

*Alhamdulillahirobbil’alamiin*. Segala puji hanya milik Allah SWT yang telah melimpahkan nikmat serta rahmat kepada penulis sehingga penulis dapat menyelesaikan penyusunan skripsi ini. Shalawat serta salam semoga tetap terlimpah curahkan kepada Nabi Muhammad SAW, para keluarganya, sahabat-sahabatnya, *tabi’in atbaut tabi’in* dan mudah-mudahan sampai kepada kita selaku pengikut ajarannya.

Dalam penyusunan skripsi ini, tidak sedikit hambatan yang penulis hadapi. Namun penulis menyadari bahwa kelancaran dalam penyusunan skripsi ini tidak lain berkat bantuan, dorongan dan bimbingan orang tua serta dosen sehingga kendala – kendala yang penulis hadapi dapat teratasi.

Dalam penulisan skripsi ini penulis menyampaikan ucapan terima kasih yang tidak terhingga kepada yang terhormat :

1. Bapak Dr. H. Hilmi Aulawi, ST., MT., selaku Ketua Sekolah Tinggi Teknologi Garut;
2. Bapak Dede Kurniadi, M.Kom selaku Ketua Program Studi Informatika dan pembimbing yang memberikan dukungan dan bimbingan dengan keikhlasan dan kesabaran dalam penyusunan skripsi ini;
3. Ibu Leni Fitriani, M.Kom selaku pembimbing yang memberikan dukungan dan bimbingan dengan keikhlasan dan kesabaran dalam penyusunan skripsi ini;
4. Seluruh staff dan civitas academica di Sekolah Tinggi Teknologi Garut.
5. Ibu yang telah memberikan dorongan baik secara materi maupun spiritual kepada penulis;
6. Saudara dan seluruh keluarga serta orang–orang terdekat yang sudah memberikan doa, bantuan, dorongan dan motivasi bagi penulis selama melakukan penyusunan skripsi;
7. Rekan-rekan Informatika yang telah membantu dan memberikan dukungan;
8. Irna Restiani, S.Kom yang tidak pernah berhenti memotivasi penulis dalam segala hal;
9. Bapak/ Ibu Staf prodi Teknik Sipil, Teknik Informatika, Teknik Industri, Dan BAAK yang banyak membantu penulis dalam mendapatkan data alumni;
10. Rekan-rekan CV. Aksi Plus dan CV. Edusoft69, saudara Antonio Saepul Islam, M. Iqbal, Dan Mubariq Ahmad yang selalu membantu menyalurkan ide dan gagasan;
11. Sahabat-sahabati Pergerakan Mahasiswa Islam Indonesia (PMII) Kabupaten Garut yang selalu berbagi pikiran dalam segala aspek;
12. Seluruh jajaran Dewan Pengurus Cabang Partai Kebangkitan Bangsa Kabupaten Garut yang selalu memfasilitasi penulis dalam penyusunan skripsi ini.

Penulis berharap semoga Allah SWT memberikan imbalan yang setimpal pada mereka yang telah memberikan bantuan dan dapat menjadikan semua bantuan ini sebagai ibadah, *amiin yaa robbal ‘alamiin*.

Semoga skripsi ini dapat bermanfaat, umumnya bagi para pembaca dan khususnya bagi penulis selaku pembuat skripsi ini.

|  |
| --- |
| Garut, Juni 2019  Penulis |

# DAFTAR ISI

[KATA PENGANTAR i](#_Toc11735254)

[DAFTAR ISI iii](#_Toc11735255)

[DAFTAR GAMBAR iv](#_Toc11735256)

[DAFTAR TABEL v](#_Toc11735257)

[1. PENDAHULUAN 6](#_Toc11735258)

[1.1. Latar Belakang 6](#_Toc11735259)

[1.2. Masalah Penelitian 7](#_Toc11735260)

[1.3. Tujuan Penelitian 7](#_Toc11735261)

[1.4. Pertanyaan Penelitian 8](#_Toc11735262)

[1.5. Cakupan Penelitian 8](#_Toc11735263)

[1.6. Manfaat Penelitian 8](#_Toc11735264)

[1.7. Sistematika Penulisan 9](#_Toc11735265)

[2. TINJAUAN PUSTAKA 10](#_Toc11735266)

[2.1. Penelitian Sebelumnya 10](#_Toc11735267)

[2.2. Kesenjangan Penelitian Sebelumnya 10](#_Toc11735268)

[2.3. Sistem Informasi 11](#_Toc11735269)

[2.4. *Tracer Study* 11](#_Toc11735270)

[2.5. Alumni 13](#_Toc11735271)

[2.6. PHP ( *Hypertext Preprocessor* ) 13](#_Toc11735272)

[2.7. *MySql* ( *My Structured Query Language* ) 13](#_Toc11735273)

[2.8. *CodeIgniter* 14](#_Toc11735274)

[2.9. *Unified Modelling Language* 14](#_Toc11735275)

[2.9.1. Diagram *Unified Modelling Languag*e 14](#_Toc11735276)

[2.10. *Rational Unified Process* (RUP) 22](#_Toc11735277)

[2.11. *Black Box Testing* 24](#_Toc11735278)

[3. METODOLOGI PENELITIAN 26](#_Toc11735279)

[3.1. *Work Breakdown Structure* 26](#_Toc11735280)

[3.2. Diagram Alur Aktivitas 28](#_Toc11735281)

[4. PEMBAHASAN 31](#_Toc11735282)

# DAFTAR GAMBAR

[Gambar 2.1 Tujuan *Tracer Study* 7](#_Toc6476378)

[Gambar 2.2 Fase *Rational Unified Process* 18](#_Toc6476379)

[Gambar 3.1 *Work Breakdown Structure* 21](#_Toc6476380)

[Gambar 3.2 Diagram Alur Aktivitas 23](#_Toc6476381)

# DAFTAR TABEL

[Tabel 1.1 Penerimaan Dan Lulusan Sekolah Tinggi Teknologi Garut 6](#_Toc12443971)

[Tabel 2.1 Simbol Class Diagram 15](#_Toc12443972)

[Tabel 2.2 Simbol Deployment Diagram 16](#_Toc12443973)

[Tabel 2.3 Simbol Use Case Diagram 17](#_Toc12443974)

[Tabel 2.4 Simbol Use Case Diagram (Lanjutan) 18](#_Toc12443975)

[Tabel 2.5 Simbol Activity Diagram 19](#_Toc12443976)

[Tabel 2.6 Simbol Sequence Diagram 20](#_Toc12443977)

[Tabel 2.7 Simbol Sequence Diagram (Lanjutan) 21](#_Toc12443978)

[Tabel 2.8 Simbol Collaboration Diagram 22](#_Toc12443979)

[Tabel 3.1 Diagram Alur Aktivitas 28](#_Toc12443980)

[Tabel 3.2 Diagram Alur Aktivitas (Lanjutan) 29](#_Toc12443981)

[Tabel 4.1 Penjelasan Proses Bisnis Berjalan 33](#_Toc12443982)

# PENDAHULUAN

## Latar Belakang

Saat ini segala aspek kehidupan hampir tidak lepas dari sistem informasi. Mulai dari dunia pendidikan, perekonomian, perusahaan, serta berbagai area sektor lainnya. Sistem informasi sendiri merupakan kumpulan data yang saling terhubung satu sama lain dalam membentuk satu kesatuan guna memproses data sampai pendistribusian informasi. Dalam dunia Pendidikan, sistem informasi banyak digunakan dalam menunjang alur informasi. Sebagaimana yang dilakukan oleh sekolah tinggi teknologi garut yang dalam menjalankan alur informasi nya menggunakan sebuah sistem informasi yang bernama SIAM ( *sistem informasi akademik mahasiswa* ).

Sekolah Tinggi Teknologi Garut merupakan sebuah lembaga pendidikan di kabupaten garut yang berada dibawah naungan yayasan Al – Musaddadiyah. Sekolah tingi teknologi garut memiliki 3 program studi yaitu teknik informatika, teknik sipil, dan teknik industri. Berikut merupakan data penerimaan mahasiswa dan lulusan sekolah tinggi teknologi garut selama 2 tahun terakhir :

**Tabel 1.1 Penerimaan Dan Lulusan Sekolah Tinggi Teknologi Garut**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Jurusan | Penerimaan | | | Lulusan | | |
| 2016 | 2017 | 2018 | 2016 | 2017 | 2018 |
| 1 | Teknik Informatika | 116 | 137 | 113 | 76 | 108 | 87 |
| 2 | Teknik Sipil | 106 | 89 | 91 | 20 | 15 | 33 |
| 3 | Teknik Industri | 84 | 72 | 86 | 19 | 23 | 45 |

Diterimanya lulusan atau alumni suatu lembaga pendidikan di dunia kerja maupun masyarakat, merupakan salah satu indikator keberhasilan lembaga tersebut (fitriani, 2016). Dalam rangka mengetahui sebaran informasi alumni tersebut, dibentuklah suatu studi tentang penelusuran alumni atau yang sekarang dikenal dengan istilah *tracer study. Tracer study* dapat digunakan untuk melacak sejauh mana keberhasilan suatu lembaga dalam proses pendidikan bagi anak didiknya.

Dalam melacak informasi alumninya, sekolah tinggi teknologi garut sudah memiliki sistem informasi *tracer study* yang terintegrasi dengan *carieer development center* dibawah naungan bidang kemahasiswaan yang sebelumnya dibuat oleh mahasiswa sekolah tinggi teknologi garut. Namun sistem tersebut belum berjalan optimal dikarenakan hanya baru menjaring beberapa puluh alumni dari sekian banyak lulusan yang ada.

Berdasarkan penjelasan uraian diatas maka penulis berkeinginan untuk mengembangkan sistem yang ada dengan harapan dapat lebih optimal dalam menjaring informasi mengenai alumni. Penulis mengangkat judul ***“Pengembangan Sistem Informasi Tracer Study Di Sekolah Tinggi Teknologi Garut”*** dengan menggunakan bahasa pemograman PHP, jika pada sistem sebelumnya menggunakan *framework Laravel,* untuk pengembangan sistem ini penulis menggunakan *codeigniter* sebagai *framework* PHP, dan *MySql* sebagai basis data.

## Masalah Penelitian

Berdasarkan uraian latar belakang diatas maka dapat dirumuskan masalah sebagai berikut :

1. Belum adanya fitur penambahan *instrument* pertanyaan *tracer study*;
2. Belum berjalan optimalnya sistem yang ada dalam menjaring informasi alumni.

## Tujuan Penelitian

Setelah menemukan permasalahan yang ada maka dapat dihasilkan tujuan penelitian sebagai berikut :

1. Penambahan *instrument* pertanyaan *tracer study*;
2. Untuk pelacakan informasi alumni berbasis web Sekolah Tinggi Teknologi Garut.

## Pertanyaan Penelitian

Adapun pertanyaan penelitian ini adalah bagaimana mengembangkan sistem informasi *tracer study* yang ada di sekolah tinggi teknologi garut supaya dapat berjalan optimal dalam menjaring informasi alumni?

## Cakupan Penelitian

Untuk cakupan pada penelitian ini adalah sebagai berikut :

1. Data penelitian diambil dari lulusan mahasiswa sekolah tinggi teknologi garut selama 2 tahun terakhir serta bersumber dari penelitian sebelumnya;
2. Pengembangan sistem informasi *tracer study* ini dapat digunakan untuk melakukan pelacakan informasi alumni sekolah tinggi teknonologi garut;
3. Penelitian ini menggunakan metodologi *Rational Unified Process* (RUP) meliputi tahap *inception, elaboration* dan *construction*;
4. *Unified Modelling Language* (UML) adalah pemodelan yang digunakan pada penelitian ini meliputi *usecase diagram, activity diagram, sequence diagram,* dan *class diagram;*
5. Dalam merancang sistem informasi *tracer study* digunakan Bahasa pemrograman PHP dan untuk *database* menggunakan *Mysql*.

## Manfaat Penelitian

Diharapkan penelitian ini dapat memberikan manfaat sebagai berikut :

1. Sebagai tolak ukur keberhasilan lembaga sekolah tinggi teknologi garut dalam mencetak pribadi yang siap bersaing di dunia global dibuktikan dengan sebaran informasi yang didapat dari alumni;
2. Proses yang lebih optimal dalam pelacakan informasi alumni sekolah tinggi teknologi garut;
3. Mempermudah mahasiswa dan alumni dalam mengakses relasi dengan dunia pekerjaan.

## Sistematika Penulisan

Skripsi skripsi ini disusun dengan sistematika sebagai berikut :

1. **PENDAHULUAN**, berisi latar belakang, masalah, tujuan, pertanyaan, cakupan, dan manfaat penelitian, serta sistematika skripsi;
2. **TINJAUAN PUSTAKA**, berisi analisis rinci terhadap penelitian sebelumnya dan hubungannya dengan penelitian yang akan dilaksanakan;
3. **METODOLOGI PENELITIAN**, berisi tahapan penelitian berikut urutan aktivitas, waktu pelaksanaan, dan sumber daya yang akan digunakan untuk mencapai tujuan penelitian;

# TINJAUAN PUSTAKA

## Penelitian Sebelumnya

Penelitian ini tidak lepas dari berbagai sumber yang penulis jadikan sebagai refensi. Berikut merupakan jurnal atau penelitian sebelumnya yang menjadi dasar acuan dalam pelaksanaan penelitian ini :

1. Penelitian yang dilakukan oleh (Budi & kawan, 2017) yang berjudul “*Tracer Study Institute Teknologi Bandung 2016 Angkatan 2009*” yang bertujuan untuk mengetahui capaian karir mahasiswa ITB lulusan 2009.
2. Penelitian yang dilakukan oleh (fitriani, 2016) yang berjudul “ *Perencanaan Pengembangan Pusat Karir Sebagai Penunjang Sumber Daya Manusia Di Sekolah Tinggi Teknologi Garut*” merupakan penelitian awal yang menjadi gagasan dikembangkannya *Tracer Study* di sekolah tinggi teknologi garut.
3. Penelitian yang dilakukan oleh (sintiani, fitriani, & kurniawati, 2017) yang berjudul “*Pengembangan Aplikasi Tracer Study STT – Garut*” menjadi pengembangan pembuatan sistem informasi tracer study di sekolah tinggi teknologi garut. Metode yang digunakan dalam penelitian tersebut adalah metodologi berorientasi objek dengan pendekatan *Unified Software Development Process* dan pemodelan *Unified Modelling Language*.

## Kesenjangan Penelitian Sebelumnya

Pada penelian pertama yang dilakukan *career center* ITB, penelitian tersebut berfokus pada pengenalan *tracer study,* serta analisis dan statistik. Analisis yang dilakukan dibatasi pada lulusan dengan rentang waktu 1 sampai 3 tahun. Statistik yang digunakan pada ppenelitian tersebut yaitu statistik deskriptif, dan statistik inferensi.

Selanjutnya pada penelitian kedua, penelitian ini merupakan analisis kebutuhan *tracer study* di sekolah tinggi teknologi garut. Yang mana penelitian *tracer study* ini termasuk kedalam poin pertama dari salah satu bagian pengembangan pusat karir sekolah tinggi teknologi garut bidang *office*.

Kemudian pada penelitian yang ketiga, sudah dibuat sistem informasi yang bernama *tracer study* STTG. Sistem informasi ini sudah dapat beroperasi dan menampung beberapa data alumni. Sistem informasi ini masih terbatas pada bagian kuesioner yang hanya bersifat statis, serta cakupan alumni yang di dapat belum optimal.

Berdasarkan uraian penelitian – penelitian diatas, dilaksanakanlah pengembangan sistem informasi *tracer study* STTG dengan tujuan agar sistem ini nantinya mampu menjaring lebih banyak alumni, serta penambahan fitur dan informasi yang dirasa dibutuhkan dalam pengembangan sistem ini.

## Sistem Informasi

Sistem informasi adalah sebuah sarana pengolahan data yang menghasilkan informasi yang cepat dan murah untuk masyarakat industri. Sistem informasi juga merupakan sebuah sistem atau sarana yang dapat mengelolah dan menghasilkan informasi yang akan digunakan oleh pihak yang membutuhkan (Tohari, 2014).

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan – laporan diperlukan (Sutabri, 2016).

## *Tracer Study*

*Tracer Study* merupakan salah satu metode yang digunakan oleh beberapa perguruan tinggi, khususnya di Indonesia untuk memperoleh umpan balik dari alumni. *Tracer Study* dilaksanakan sebagai kebutuhan akan pentingnya umpan balik dari para alumni untuk perbaikan sistem dan pengelolaan pendidikan.. *Tracer Study* juga bermanfaat dalam memetakan dunia usaha dan industri sehingga jeda diantara kompetensi yang diperoleh alumni saat kuliah dengan tuntutan dunia kerja dapat diperkecil (Budi & kawan, 2017).

*Tracer study* merupakan suatu bentuk penelitian mengenai situasi alumni khususnya dalam hal pencarian kerja, situasi kerja, dan pemanfaatan pemerolehan kompetensi selama kuliah di Universitas atau Perguruan tinggi . Di negara-negara maju, studi pelacakan jejak alumni adalah studi utama yang telah dilaksanakan secara sistematis, institusional, dan terus menerus (Saiful, Sudianto, & Nurhidayati, 2019).

Pendekatan melalui lulusan, dan angkatan merupakan metode pndekatan yang umum dilakukan di Indonesia. Kedua pendekatan itu dilakukan supaya mendapatkan responden yang optimal sebagai pengisi data *tracer study*. Kedua pendekatan ini memerlukan diskusi yang lebih mendalam, mengningat kriteria dan syarat keduanya sangat berbeda dalam penerapan.

1. Tujuan *Tracer Study*

*Tracer Study* bertujuan untuk mengetahui hasil pendidikan dalam bentuk transisi dari dunia pendidikan tinggi ke dunia usaha dan industri, keluaran pendidikan berupa penilaian diri terhadap penguasaan dan pemerolehan kompetensi, proses pendidikan berupa evaluasi proses pembelajaran dan kontribusi pendidikan tinggi terhadap pemerolehan kompetensi serta input pendidikan berupa penggalian lebih lanjut terhadap informasi lulusan (Budi & kawan, 2017).



**Gambar 2.1 Tujuan Tracer Study**

1. Manfaat *Tracer Study*

Menurut ramadiani, putut,dan eko (Ramadiani, Widagdo, & Junirianto, 2016) *tracer study* memiliki manfaat sebagai berikut :

1. Memenuhi kebutuhan data dan informasi terkait lulusan;
2. Memenuhi kebutuhan data akreditasi perguruan tinggi;
3. Mempermudah perguruan tinggi dalam mengelola data;
4. Memudahkan perguruan tinggi dalam menunjang keputusan menentukan perencanaan strategis sebagai upaya peningkatan kualitas mutu lulusan.

## Alumni

Alumni perguruan tinggi merupakan komponen yang seharusnya saling memiliki sinergi baik. Eksistensi alumni ini sangat perlu untuk diketahui agar institusi mampu melakukan refleksi demi tercapainya perbaikan kualitas pembelajaran dan segala kegiatan akademik. Masukan dari alumni dinilai penting karena merupakan bagian yang tidak terpisahkan dengan lembaga (Nugroho & Arifudin, 2014).

## PHP ( *Hypertext Preprocessor* )

PHP atau kepanjangan dari *Hypertext Preprocessor* merupakan Bahasa pemrograman yang digunakan untuk membuat apliksi berbasis website. Sebagai sebuah aplikasi, website tersebut hendaknya memiliki sifat dinamis dan interaktif. Dinamis disini artinya website tersebut bias berubah tampilan kontennya sesuai kondisi tertentu ( misalnya, tampilan produk yang berbeda untuk setiap pengunjung). Kemudian interaktif dalam arti dapat memberikan respon atau *feedback* bagi pengguna ( misalnya, menampilkan hasil pencarian ) (Enterprise, 2017).

## *MySql* ( *My Structured Query Language* )

*MySQL* atau *My Structured Query Language* adalah salah satu aplikasi DBMS ( *Database Management System* ) yang sudah sangat banyak digunakan oleh para pemrogram aplikasi web. Dalam sistem database tak relasional, semua informasi disimpan pada satu bidang luas, yang kadangkala data di dalamnya sangat sulit dan melelahkan untuk diakses. Tetapi *MySQL* merupakan sebuah sistem database relasional, sehingga dapat mengelompokkan informasi ke dalam tabel-tabel atau grup-grup informasi yang berkaitan. Setiap tabel memuat bidang-bidang yang terpisah, yang mempresentasikan setiap bit informasi. *MySQL* menggunakan indeks untuk mempercepat proses pencarian terhadap baris informasi tertentu (Lutfi, 2017).

## *CodeIgniter*

*CodeIgniter* adalah sebuah *framework* bahasa pemrograman PHP. *CodeIgniter* bisa dibilang framework PHP paling popular di Indonesia berkat kemudahan yang ditawarkan dalam penggunaanya. Maka tak heran jika situs – situs besar juga mengadopsi sistem ini sebagai *core engine*  websitenya (Komputer, 2014).

## *Unified Modelling Language*

*Unified Modeling Language* adalah sebuah bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek serta merupakan bahasa visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak dengan menggunakan diagram dan teks-teks pendukung (Shalahuddin & Sukamto, 2015)

1. Diagram Unified Modelling Language

Pada pemodelan *Unified Modelling Language* tersedia diagram-diagram yang dikelompokan menjadi 3 kategori, yaitu *structure diagram, behavior diagram* dan *interaction diagram* (Shalahuddin & Sukamto, 2015)*.*

Penjelasan mengenai diagram – diagram *Unified Modelling Language* adalah sebagai berikut :

1. *Structure diagram* yaitu diagram-diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan (Shalahuddin & Sukamto, 2015). Diagram yang dikelompokan pada *structure diagram* diantaranya adalah :
2. *Class Diagram*

*Class diagram* digunakan untuk menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki atribut dan metode atau operasi yang dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem (Shalahuddin & Sukamto, 2015). Berikut adalah simbol-simbol yang ada pada *class diagram* :

**Tabel 2.1 Simbol Class Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Simbol | Nama | Deskripsi |
| 1. | + operasi ( )  + atribut  Nama\_kelas | Kelas / *Class* | Kelas pada struktur sistem. |
| 2. |  | Asosiasi / *Association* | Hubungan antar kelas dengan makna umum. |
| 3. |  | Asosiasi Berarah / *Directed Association* | Hubungan antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain. |
| 4. |  | Generalisasi / *Generalization* | Hubungan antar kelas dengan makna generalisasi-spesialisasi (umum khusus) |
| 5. |  | Ketergantungan / *Dependency* | Hubungan antar kelas dengan makna ketergantungan antar kelas |
| 6. |  | Agregasi / *Aggregation* | Hubungan antar kelas dengan makna semua bagian |

Sumber : (Shalahuddin & Sukamto, 2015)

1. *Deployment Diagram*

*Deployment diagram* merupakan diagram yang digunakan untuk menunjukan konfigurasi komponen dalam proses eksekusi aplikasi. *Deployment diagram* juga digunakan untuk memodelkan sistem tambahan rancangan, sistem *client/server*, sistem terdistribusi murni dan rekayasa ulang aplikasi (Shalahuddin & Sukamto, 2015). Berikut adalah simbol-simbol yang ada pada *deployment diagram* :

**Tabel 2.2 Simbol Deployment Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Simbol | Nama | Deskripsi |
| 1. | *Package* | *Package* | *Package* merupakan sebuah bungkusan dari satu atau lebih *node*. |
| 2. | Nama\_node | *Node* | Biasanya mengacu pada perangkat keras (*hardware*), perangkat lunak yang tidak dibuat sendiri, jika di dalam *node* disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen. |
| 3. |  | *Link* | Hubungan antar *node.* |
| 4. |  | Ketergantungan / *Dependency* | Ketergantungan antar *node* Arah panah mengarah pada *node* yang dipakai. |

Sumber : (Shalahuddin & Sukamto, 2015)

1. *Behavior diagram* merupakan kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian proses perubahan yang terjadi pada sebuah sistem (Shalahuddin & Sukamto, 2015). Diagram yang dikelompokan pada *behavior diagram* diantaranya adalah :
2. *Use Case Diagram*

*Use case diagram* merupakan pemodelan untuk menjelaskan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi. *Use case diagram* digunakan untuk mengetahui fungsi-fungsi yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Aktor merupakan orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor (Shalahuddin & Sukamto, 2015). Berikut adalah simbol-simbol yang ada pada *use case diagram* :

**Tabel 2.3 Simbol Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Simbol | Nama | Deskripsi |
| 1. |  | *Use Case* | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. |
| 2. |  | Aktor / *Actor* | Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat. |
| 3. |  | Asosiasi / *Association* | Komunikasi antara aktor dan *use case.* |
| 4. | <<extend>> | Ekstensi / *Extend* | Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walaupun tanpa *use case* tambahan itu. |

**Tabel 2.4 Simbol Use Case Diagram (Lanjutan)**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Simbol | Nama | Deskripsi |
| 5. |  | Generalisasi / *Generalization* | Relasi generalisasi-spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum. |
| 6. | <<include>> | Menggunakan / *Include* | Hubungan *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya. |
| 7. |  | Sistem | Menspesifikasikan paket yang menggambarkan sistem informasi secara terbatas. |
| 8. |  | *Note* | Elemen fisik yang eksis saat sistem informasi dijalankan dan mencerminkan sumber daya komputasi. |

Sumber : (Shalahuddin & Sukamto, 2015)

1. *Activity Diagram*

*Activity diagram* merupakan pemodelan untuk menggambarkan aliran kerja atau aktifitas dari sebuah sistem. *Activity diagram* inimenggambarkan aktifitas yang dapat dilakukan oleh sistem bukan aktifitas yang dilakukan oleh aktor (Shalahuddin & Sukamto, 2015). Berikut ini adalah simbol-simbol yang digunakan untuk membuat *activity diagram* :

**Tabel 2.5 Simbol Activity Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Simbol | Nama | Deskripsi |
| 1. |  | Status Awal | Menyatakan status awal aktifitas suatu sistem. |
| 2. |  | Aktifitas / *Activity* | Aktifitas yang dilakukan sistem. Biasanya diawali dengan kata kerja. |
| 3. |  | Percabangan / *Decision* | Asosiasi percabangan untuk digunakan saat ada pilihan aktifitas yang lebih dari satu. |
| 4. |  | Penggabungan / *Join* | Asosiasi penggabungan dimana lebih dari satu aktifitas yang digabungkan menjadi satu. |
| 5. |  | Status Akhir | Status akhir yang dilakukan oleh sistem. Sebuah diagram aktifitas memiliki sebuah status akhir. |
| 6. | Nama\_swimlane | *Swimlane* | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi. |

Sumber : (Shalahuddin & Sukamto, 2015)

1. *Interaction Diagram*

*Interaction diagram* merupakan kumpulan diagram yang digunakan untuk menggambarkan interaksi antar subsistem pada suatu sistem atau interaksi sistem dengan sistem lain (Shalahuddin & Sukamto, 2015). Diagram yang dikelompokan pada *interaction diagram* diantaranya adalah :

1. *Sequence Diagram*

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan menjelaskan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Dalam pembuatan *sequence diagram* dibutuhkan untuk melihat skenario *use case* (Shalahuddin & Sukamto, 2015). Berikut adalah simbol-simbol pada *sequence diagram* :

**Tabel 2.6 Simbol Sequence Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Simbol | Nama | Deskripsi |
| 1. |  | Aktor / *Actor* | Orang, proses atau sistem lain yang berintekrasi dengan sistem yang dibuat. |
| 2. |  | Garis Hidup / *Lifeline* | Menyatakan kehidupan suatu objek. |
| 3. | Nama\_objek | Objek / *Object* | Menyatakan objek yang berinteraksi menggunakan pesan. |
| 4. |  | Waktu Aktif / *Join* | Menyatakan objek dalam keadaan aktif dan berinteraksi. Semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. Aktor tidak memiliki waktu aktif. |
| 5. | <<create>> | Pesan Tipe *Create* | Menyatakan suatu objek membuat objek yang lain. Arah panah mengarah pada objek yang dibuat. |

**Tabel 2.7 Simbol Sequence Diagram (Lanjutan)**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Simbol | Nama | Deskripsi |
| 6. | 1 : nama\_metode () | Pesan Tipe *Call* | Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. |
| 7. | 1 : masukan | Pesan Tipe *Send* | Menyatakan bahwa suatu objek mengirimkan data ke objek lainnya. |
| 8. | 1 : keluaran | Pesan Tipe *Return* | Menyatakan suatu objek yang menjalankan suatu operasi atau metode yang menghasilkan suatu kembalian ke objek tertentu. |

Sumber : (Shalahuddin & Sukamto, 2015)

1. *Collaboration Diagram*

*Collaboration diagram* atau *communication diagram* merupakan diagram yang digunakan untuk menggambarkan interaksi antar objek dalam bentuk urutan pengiriman pesan. *Collaboration diagram* mempresentasikan informasi yang diperoleh dari *class diagram, sequence diagram* dan *use case diagram* untuk menjelaskan gabungan antara struktur statis dan tingkah laku dinamis dari suatu sistem (Shalahuddin & Sukamto, 2015). Berikut ini adalah simbol-simbol yang digunakan pada *collaboration* *diagram* :

**Tabel 2.8 Simbol Collaboration Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Simbol | Nama | Deskripsi |
| 1. | Nama\_objek | Objek / *Object* | Objek yang melakukan interaksi pesan. |
| 2. |  | *Link* | Relasi antar objek yang menghubungkan antara suatu objek dengan objek lainnya atau dengan dirinya sendiri*.* |
| 3. |  | Arah Pesan / *Stimulus* | Arah pesan yang terjadi. Jika pada *link* ada dua arah pesan yang berbeda maka arah pesan juga digambarkan menjadi dua arah pada dua sisi *link*. |

Sumber : (Shalahuddin & Sukamto, 2015)

## *Rational Unified Process* (RUP)

*Rational Uniefied Process* adalah metode pengembangan perangkat lunak dengan pendekatan *iterative* yang dikembangkan oleh *Rational Software* terstruktur dengan baik, jelas penggunaanya, penyelesaiannya, dan waktu penggunaannya serta menggunakan konsep *object oriented* dengan menggunakan pemodelan *Unified Modelling Language* dalam proses rekayasa perangkat lunak (Kructhen, 2003).



**Gambar 2.2 Fase Rational Unified Process**

(Kructhen, 2003)

*Rational* *Unified Process* adalah sutau pendekatan yang dilakukan berulang – ulang dalam pengembangan perangkat lunak ( *iterative* ), progres menaik yang bertahap ( *incremental* ), arsitektur yang terfokus ( *architecture – centric* ), dan lebih diarahkan berdasarkan penggunaan kasus ( *usexase driven* ) serta menyediakan struktur yang baik dalam pendefinisian untuk alur hidup proyek perangkat lunak (Shalahuddin & Sukamto, 2015).

Berikut merupakan empat fase dalam metode *Rational Unified Process* :

1. Fase *inception*

Tahap dimana kita mendefinisikan kebutuhan sistem yang akan dibuat ( *requirement* ) dan memodelkan proses bisnis yang dibutuhkan ( *business modeling* ). Tahap ini membangun *business case* untuk sistem dan membatasi ruang lingkupnya, untuk melakukan hal ini diharuskan untuk mengidentifikasi semua entitas eksternal yang akan berinteraksi dengan sistem, dan mendefinisikan interaksi pada level tertentu. Ini juga termasuk mengidentifikasi semua *use case* dan menjelaskan beberapa yang signifikan. *Business case* termasuk kriteria keberhasilan, perkiraan resiko, dan mengestimasikan sumber daya yang dibutuhkan.

1. Fase *elaboration*

Tujuan dari fase *elaboration* adalah menganalisis domain masalah, membuat sebuah dasar arsitektur, membangun rencana proyek, dan mengeliminasi resiko terbesar dari proyek. Untuk menjalankan objek-objek tersebut diperlukan melihat lebih luas dan lebih dalam terhadap sistem. Pada tahap ini merupakan tahap paling sulit karena pada tahap ini memastikan bahwa arsitektur, kebutuhan, dan perencanaan cukup stabil sehingga waktu dan biaya tidak berubah.

1. Fase *construction*

Dalam fase ini semua komponen dan fitur aplikasi yang dibuat dan di integrasikan kedalam *software*. Dalam fase ini juga dituntut untuk mengoptimalkan sumber daya, biaya, jadwal dan kualitas. Hal ini menjadi salah satu alasan mengapa pengembangan dari arsitektur ditekankan pada fase *elaboration*. Keluaran dari fase *construction* adalah sebuah *software*  yang sudah siap diserahkan kepada *client.*

1. Fase *transition*

*Transition* adalah fase dimana *software* diserahkan kepada *client*, dalam fase ini juga dilakukan pengujian terhadap *software*, dan apabila *software* telah diserahkan kepada *client*, *developer* masih memantau bagaimana kinerja dari *software* tersebut.

## *Black Box Testing*

*Black box testing* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program (Mustaqbal, Firdaus, & Rahmadi, 2016).

Metode pengujian *Black Box Testing* merupakan metode pengujian yang berfokus terhadap persoalan tentang apakah unit program memenuhi kebutuhan yang disebutkan dalam spesifikasi. Pada *black box testing,* cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi modul, kemudian diamati apakah hasil dari modul itu sesuai dengan proses bisnis yang diinginkan (Mustaqbal, Firdaus, & Rahmadi, 2015).

# METODOLOGI PENELITIAN

## *Work Breakdown Structure*

*Work breakdown structure* merupakan gambaran proses yang dilakukan selama penelitian dalam menjelaskan tujuan penelitian dan tahapan – tahapan penelitian yang disesuaikan dengan metodologi pengembangan sistem serta aktivitas yang dilakukan pada setiap tahapan – tahapan tersebut. Berikut merupakan gambaran *work breakdown structure* pada penelitian ini:



**Gambar 3.1 Work Breakdown Structure**

Berikut merupakan penjelasan dari tahapan aktivitas pada perancangan sistem berdasarkan *work breakdown structure:*

1. ***Study* Literatur**

Studi literatur adalah melakukan pencarian informasi dan data yang penting mengenai segala sesuatu yang berhubungan dengan penelitian (Syafruddin, Hakim, & Despa, 2014). Dalam memenuhi kebutuhan informasi penelitian ini, penulis melakukan observasi ke kantor setiap jurusan yang ada di sekolah tinggi teknologi garut untuk mendapatkan jumlah alumni pada 2 tahun terakhir.

1. ***Inception***

Tahap *inception* merupakan tahap awal dari RUP, guna menjelaskan hasil dari studi literatur yang diolah menjadi spesifikasi kebutuhan sistem dan proses bisnis baru yang dirancang untuk mendapatkan hasil akhir yang berupa spesifikasi sistem, aktor, *use* *case* diagram dan *activity* diagram yang akan digunakan dalam pembuatan diagram selanjutnya pada tahap *elaboration* (Sukamto & Shalahuddin, 2018)*.*

1. ***Ellaboration***

Pada tahap ini dilakukan analisis lebih lanjut dari tahap *inception* dengan mengembangkan proses bisnis yang dihasilkan pada tahap *inception*. dimana pada tahap *elaboration* iniaktor, *use* *case* *diagram* dan *activity* *diagram* akan digunakan untuk membuat gambaran awal dari sebuah sistem dan rincian sistem yang akan dibuat bersamaan dengan perancangan *layout*. Maka dari itu, hasil yang didapat pada tahap *elaboration* ini yaitu *sequence* *diagram*, *class* *diagram*, struktur menu dan rancangan *layout* yang akan diolah pada tahap *construction* guna mengimplementasikan rancangan kedalam Bahasa pemrograman (Sukamto & Shalahuddin, 2018).

1. ***Construction***

Setelah perancangan yang telah dilakukan pada tahap *elaboration* selesai, maka selanjutkan akan dilakukan implementasi hasil dari perancangan tersebut ke dalam bahasa pemrograman untuk mendapatkan hasil akhir sebuah sistem yang kemudian selanjutnya akan dilakukan pengujian perangkat lunak (Sukamto & Shalahuddin, 2018)

1. ***Transtition***

Tahap ini merupakan tahap yang dilakukan untuk uji coba sistem dengan menggunakan *black* *box* *testing*, dengan pengujian yang dilakukan hanya pada area fungsional saja. Setelah pengujian berhasil dilakukan maka sistem dapat digunakan, jika tidak berhasil/ tidak sesuai dengan analisis dan kebutuhan maka kembali lagi ke tahap sebelumnya untuk mengerjakan tahap *costruction* (Greenit, 2018).

1. ***Publish***

Tahap ini merupakan tahap akhir yang dilakukan setelah mendapatkan hasil dari tahapan sebelumnya, yaitu *transtition*.

## Diagram Alur Aktivitas

Berdasarkan dari *Work Breakdown Structure* yang telah dibuat, maka diagram alur aktivitas penelitian ini dapat disajikan sebagai berikut :



**Gambar 3.2 Diagram Alur Aktivitas**

Diagram alur aktivitas menggambarkan proses dari aktivitas – aktivitas pengembangan yang akan dilaksanakan dengan menggunakan metode *Rational Unified Process* dan pemodelan *Unified Modeling Language*. Adapun waktu untuk pelaksanaan penelitiannya disajikan dalam bentuk tabel sebagai berikut :

**Tabel 3.1 Diagram Alur Aktivitas**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Tahapan | Aktivitas | Mulai | Selesai | Durasi |
| 1 | Studi Literatur | Menentukan Topik Penelitian Dan Analisis | 12/02/2019 | 22/02/2019 | 10 Hari |

**Tabel 3.2 Diagram Alur Aktivitas (Lanjutan)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Tahapan | Aktivitas | Mulai | Selesai | Durasi |
|  |  | Mengembangkan Kerangka Berpikir | 24/02/2019 | 28/02/2019 | 4 Hari |
| Mengumpulkan Data Lulusan | 02/03/2019 | 03/03/2019 | 1 Hari |
| 2 | *Inception* | Menentukan Spesifikasi Sistem | 05/03/2019 | 07/03/2019 | 2 Hari |
| Menentukan Aktor | 10/03/2019 | 11/03/2019 | 2 Hari |
| Merancang *Use Case* Diagram | 13/03/2019 | 16/03/2019 | 4 Hari |
| Merancang *Activity* Diagram | 17/03/2019 | 20/03/2019 | 4 Hari |
| 3 | *Elaboration* | Merancang *Squence* Diagram | 21/03/2019 | 24/03/2019 | 4 Hari |
| Merancang *Class* Diagram | 25/03/2019 | 30/03/2019 | 4 Hari |
| 4 | *Contruction* | Merancang *Layout* | 03/04/2019 | 05/04/2019 | 3 Hari |
| Merealisasikan Rancangan Ke Dalam Bahasa Pemrograman | 12/04/2019 | 28/05/2019 | 47 Hari |
| 5 | *Trantition* | Pengujian *Balckbox* *Testing* | 30/05/2019 | 08/06/2019 | 10 Hari |
| 6 | *Publish* | Pembuatan Artikel |  |  |  |
|  |  | Unggah Jurnal STTG |  |  |  |
|  |  |  |  |  |  |

Pada Tabel 3.1 menjelaskan alokasi waktu pengerjaan yang tampak seperti alur aktivitas pada gambar 3.2, terhitung dari minggu kedua pada bulan februari, dengan aktivitas penelitian berdasarkan tahapan metode *Rational* *Unified* *Process* (RUP).

# PEMBAHASAN

## Hasil Penelitian

Proses pada penelitian ini dilaksanakan sesuai dengan tahapan-tahapan dan aktivitas-aktivitas yang tergambarkan pada *work breakdown structure*. Dalam pelaksanaannya, penelitian ini mendapatkan beberapa temuan seperti belum berjalan optimalnya sistem informasi *tracer study* yang sedang berjalan, dan belum tersedianya *instrument* pertanyaan pada sistem tersebut. Pengoptimalan dan *instrument* pertanyaan diperlukan agar supaya sebaran informasi yang didapat lebih akurat dan tepat sasaran. Pada sistem informasi yang sedang berjalan, alumni mendaftarkan email untuk mendapatkan akun tracer study. Kemudian alumni dipersilahkan untuk mengisi form *kuesioner* yang telah disediakan. Pada form *kuesioner* inilah terdapat kekurangan *instrument* pertanyaan, yang mana pada form tersebut pertanyaan masih bersifat statis. Belum tersedianya rekap data hasil pengisian form *kuesioner* tersebut. Untuk memenuhi kebutuhan data alumni, rekap data dibutuhkan agar dapat terindentifikasi setiap alumni yang sudah mendaftar serta mengisi form *kuesioner* tersebut.

Penjaringan alumni dilakukan oleh bidang kemahasiswaan sekolah tinggi teknologi garut, dalam hal ini CDC ( *carieer development center* ) yang di prakarsa oleh ibu Rina Kurniawati., M.Si Bersama ibu Leni Fitriani., S.T, M.Kom. Penjaringan alumni dilakukan guna mendapatkan sebaran informasi alumni yang sudah dan belum bekerja agar terciptanya sinegritas antara perguruan tinggi sebagai lembaga pendidikan dengan alumni yang mana nantinya dapat menjadi acuan bagi mahasiswa yang masih menimba ilmu disekolah tinggi teknologi garut.

## Pembahasan Hasil Penelitian

Penelitian ini dilakukan di sekolah tinggi teknologi garut yang bertujuan untuk memenuhi kebutuhan data sebaran alumni, serta sebagai acuan dalam mengetahui perkembangan peserta didik paska selesai menjalani pendidikan di sekolah tinggi teknologi garut.

1. Jawaban Penelitian

Jawaban atas permasalahan penelitian yang berkaitan dengan pengoptimalan sistem dan penambahan *instrument* pertanyaan pada sistem informasi *tracer study*, akan disajikan secara detail dan terperinci sesuai dengan tahapan metodologi *Rational Unifie Process*, sebagaimana tergambar pada *Work Breakdown Structure*.

1. *Inception*

*Inception*merupakan tahapan awal setelah bahan-bahan penelitian terkumpul dan dipersiapkan. Kemudian pemahaman akan definisi sistem yang akan dibuat berdasarkan aktifitas yang dilakukan, seperti menentukan spesifikasi sistem, menentukan actor, merancang *Use Case* diagram, dan merancang *Activity* diagram.

1. Proses Bisnis Berjalan

Analisis pada proses bisnis berjalan dilakukan guna mendapatkan laporan titik mana saja yang diperlukan pengembangan. Berikut merupakan proses bisnis berjalan dari sistem informasi *tracer study* yang digunakan di sekolah tinggi teknologi garut :



**Gambar 4.1 Proses Bisnis Berjalan**

Proses bisnis berjalan dari sistem informasi *tracer study* dapat diuraikan sebagaimana berikut :

**Tabel 4.1 Penjelasan Proses Bisnis Berjalan**

|  |  |  |
| --- | --- | --- |
| No | Aktivitas | Deskripsi |
| 1 | Registrasi | Alumni melakukan registrasi untuk dapat mengakses sistem informasi *tracer study* dengan mengisi form yang telah disediakan |
| 2 | Kondisi “Apakah Berhasil?” | Merupakan kondisi dimana ketika alumni mengisi form registrasi, semua kolom pada form tersebut harus terisi. Jika terdapat kolom yang masih kosong, sistem akan memberi tanda merah pada kolom tersebut dan alumni harus mengisi kembali kolom tersebut. Pada aktivitas terdapat 2 kondisi, yang mana jika “ya” alumni dapat melakukan *login* dengan *username* yang sudah di daftarkan, dan jika “tidak”, alumni harus kembali memeriksa form pendaftaran. |
| 3 | *Login* | Merupakan aktivitas yang dilakukan alumni setekah berhasil melakukan registrasi |
| 4 | Mengisi Kuesioner | Aktivitas yang dilakukan alumni setelah berhasil *login* kedalam sistem informasi *tracer study,* selajutnya mengisi kuesioner yang telah disediakan. |
| 5 | Laporan Hasil Kuesioner | Sistem melakukan penyimpanan data hasil pengisian kuesioner yang telah dilakukan oleh alumni. |

**Tabel 4.2 Penjelasan Proses Bisnis Berjalan Lanjutan**

|  |  |  |
| --- | --- | --- |
| No | Aktivitas | Deskripsi |
| 6 | Mengakses Laporan Hasil Kuesioner | Admin melakukan pengecekan terhadap laporan kuesioner yang tersimpan di dalam sistem |
| 7 | Membuat Laporan | Setelah melakukan pengecekan, admin membuat laporan rekapitulasi yang nantinya akan diserahkan ke kepala CDC (*Carieer Development Centre*) |
| 8 | Menerima Laporan | Kepala CDC (*Carieer Development Centre*) akan menerima laporan yang sebelumnya dibuat oleh admin. |

1. Spesifikasi Kebutuhan Sistem

Pada perancangan sistem, menentukan kebutuhan spesifikasi sistem menjadi penjelasan akan kebutuhan-kebutuhan bagi pembangunan sistem tersebut. Berdasarkan analisis pada proses bisnis yang sedang berjalan, maka kebutuhan sistem yang harus dikembangkan adalah sebagai berikut :

1. Syarat Tampilan
2. Tampilan menu yang lebih interaktif;
3. Tersedianya informasi lowongan pekerjaan baik dari informasi yang didapat dari alumni, ataupun dari CDC (*Carieer Development centre*) itu sendiri;
4. Penambahan *instrument* pertanyaan yang dapat dimodifikasi oleh admin.
5. Persyaratan Sistem

Untuk menyelesaikan permasalahan yang timbul pada sistem yang sedang berjalan, maka dibuatkanlah pengembangan sistem yang memiliki syarat sebagaimana hasil analisis pada data-data. Adapun persyaratan sistem yang dibutuhkan adalah sebagai berikut:

**Tabel 4.3 Spesifikasi Sistem**

|  |  |  |
| --- | --- | --- |
| No. | Spesifikasi | Rincian |
| 1. | Spesifikasi Fungsional Sistem | 1. Sistem menyediakan fasilitas input, edit, hapus dan menampilkan data alumni; 2. Sistem menyediakan fasilitas input dan menampilkan data hasil kuesioner; 3. Sistem menyediakan fasilitas untuk input, hapus, dan edit form kuesioner; 4. Sistem menyediakan fasilitas *page builder*; 5. Sistem menyediakan fitur untuk menampilkan laporan hasil dari pengisian kuesioner. |
| 2. | Spesifikasi Non Fungsional Sistem | 1. Sistem penggajian ini dirancang menggunakan bahasa Pemrogram-an PHP, *Framework CodeIgniter,* dan MySQL untuk *database*; 2. Syarat minimal perangkat komputer yang dibutuhkanyaitu: 3. *Processor*: intel dual core atau AMD; 4. Sistem Operasi: *Windows* 7; 5. *RAM*: 2 GB; 6. *Harddisk*: 512 GB; 7. *Web* *browser*: Google Chrome 64/84 bit. |
| 3. | Persyaratan Pengembangan | 1. Menggunakan *Use case* diagram untuk menggambarkan interkasi pengguna dengan sistem; 2. Penggunaan bahasa pemrograman PHP dan *Mysql* untuk database; 3. Metodologi dan *tool* yang tepat dalam pengembangan sistem informasi. |
| 4. | Batasan Sistem | 1. Admin dapat memverifikasi kebenaran pengisi kuesioner dengan melihat npm yang dimasukan oleh alumni; 2. Admin dapat menghapus dan menambahkan *user* sistem informasi *tracer study*; 3. Admin dapat menambahkan dan mengurangi informasi yang ditampilkan pada sistem informasi *tracer study*; |

**Tabel 4.4 Spesifikasi Sistem Lanjutan**

|  |  |  |
| --- | --- | --- |
| No. | Spesifikasi | Rincian |
|  |  | 1. Kepala CDC dapat melihat laporan hasil pengisian kuesioner; 2. Alumni dapat mengisi kuesioner yang terdapat pada sistem informasi *tracer study*;   Alumni dapat melihat informasi tentang lowongan pekerjaan yang terdapat pada sistem informasi *tracer study*; |

1. Target Pengguna

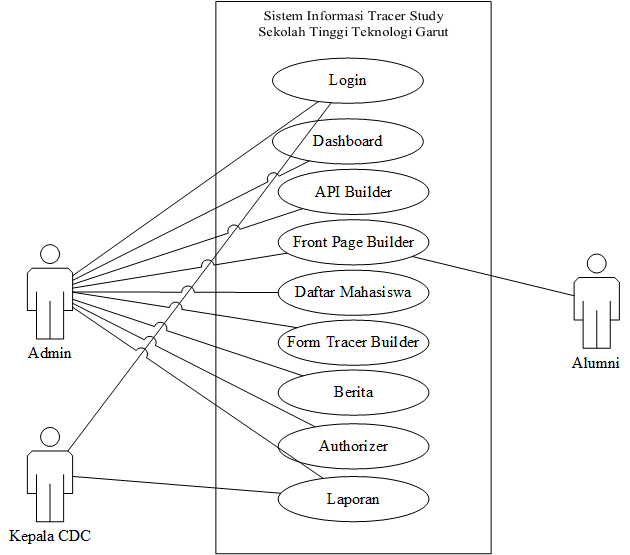
Identifikasi aktor diperlukan dalam pengembangan sistem informasi ini, yang mana aktor ini nantinya akan terlibat langsung dalam aktivitas sistem. Adapun target pengguna yang akn terlibat dalam sistem informasi *tracer study* ini yaitu admin, kepala CDC (*carieer development centre*), dan alumni. Berikut merupakan aktivitas yang dapat dilakukan oleh aktor :

**Tabel 4.5 Aktivitas Aktor**

|  |  |  |
| --- | --- | --- |
| No. | Aktor | Aktivitas |
| 1. | Admin | Melakukan *login*  Mengelola data pengguna (menambah, mengedit, dan menghapus data pengguna)  Mengelola *Crud Builder*  Mengelola *Page Builder*  Mengelola *Form* kuesioner atau *Form Builder* |
| 2. | Kepala CDC | Melakukan *Login*  Melihat laporan hasil pengisian kuesioner |
| 3. | Alumni | Melakukan *Login*  Mengisi kuesioner |

1. Merancang *Use Case*

Interaksi antara pengguna dengan sistem akan tergambar jelas dengan *Use case* diagram. *Use case*  sendiri berguna untuk menggambarkan alur aktivitas yang terjadi pada sistem, juga sebagai informasi mengenai fungsi-fungsi yagn terdapat pada sistem. Berikut merupakan *use case* diagram dan *scenario* pada sistem informasi *tracer study*:



**Gambar 4.2 Use Case Diagram**

Gambaran mengenai *use case diagram* sistem informasi *tracer study* dapat dideskripsikan kedalam table sebagai mana yang terdapat pada tabel 4.6 dan 4.7.

**Tabel 4.6 Deskripsi Use Case Diagram**

|  |  |  |
| --- | --- | --- |
| No. | Us*e Case* | Deskripsi |
| 1. | *Login* | Proses awal untuk menjalankan sistem informasi. |
| 2. | *Dashboard* | Halaman utama admin |
| 3. | *API Buiilder* | Fitur yang digunakan untuk menhasilkan API ( *Application Programing Interface* ) dari data table yang dipilih |
| 4. | *Front Page Builder* | Fitur yang digunakan oleh admin untuk membuat *page layou*t *tracer study* yang nantinya dapat diakses oleh alumni. |
| 5. | Data Mahasiswa | Fitur yang digunakan oleh admin untuk membuat dan memanipulasi *form* kuesioner yang nantinya diisi oleh alumni. |

**Tabel 4.7 Deskripsi Use Case Diagram Lanjutan**

|  |  |  |
| --- | --- | --- |
| No. | *Use Case* | Deskripsi |
| 6. | Berita | Adalah fitur yang menyediakan informasi seputar lowongan pekerjaan, dan informasi terkait perkembangan sekolah tinggi teknologi garut |
| 7. | *Authorizer* | Fitur yang digunakan oleh admin untuk menambah dan menghapus pengguna. |
| 8. | Kuesioner | Menu ini terdapat pada *page* yang dibuat oleh admin dan peruntukan untuk alumni. |
| 9. | Laporan | Menu ini diperuntukan khusus kepala CDC untuk mengetahui hasil dari pengisian kuesioner yang telah dilakukan oleh alumni. |

Jalannya masing-masing *use case* yang telah didefinisikan, dapat digambarkan dengan *scenario use case*. Berikut merupakan *Scenario-scenario use case* yang telah dibuat :

**Tabel 4.8 Scenario Use Case Login**

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Skenario Normal | |
| 1. Memasukan *Username* dan *password* |  |
|  | 1. Melakukan validasi pada tabel di *database* terhadap *username* dan *password* yang dimasukan |
|  | 1. Masuk kedalam sistem informasi |
| Skenario Alternatif | |
| 1. Memasukan *username* dan *password* |  |
|  | 1. Melakukan validasi pada tabel di *database* terhadap *username* dan *password* yang dimasukan |
|  | 1. Menampilkan pesan kesalahan ”*user does not exist*” |
| 1. Memasukan *username* dan *password* yang tersedia |  |

**Tabel 4.9 Scenario Use Case Login Lanjutan**

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
|  | 1. Melakukan validasi pada tabel di database terhadap username dan password yang dimasukan |
| 1. Masuk kedalam sistem informasi |  |

**Tabel 4.10 Scenario Use Case Dashboard**

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| 1. Admin memilih menu *dashboard*  pada sistem |  |
|  | 1. Menampilkan halaman utama |

**Tabel 4.11 Scenario Use Case API Builder**

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| 1. Admin memilih menu API *builder* pada sistem |  |
|  | 1. Menampilkan *rest* data yang tersedia |

**Tabel 4.12 Scenario Use Case Front Page Builder**

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| 1. Admin memilih menu *front page builder* pada sistem |  |
|  | 1. Menampilkan sub menu |
| 1. Memilih menu tambahkan *page* baru |  |
|  | 1. Menampilkan form untuk membuat *page* baru |

**Tabel 4.13 Scenario Use Case Data Mahasiswa**

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| 1. Admin memilih menu data mahasiswa |  |
|  | 1. Menampilkan daftar mahasiswa |

**Tabel 4.14 Scenario Use Case Form Tracer Builder**

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| 1. Admin memilih menu *form tracer builder* pada sistem |  |
|  | 1. Menampilkan sub menu |
| 1. Memilih menu tambahkan baru |  |
|  | 1. Menampilkan form untuk membuat *form* baru |
| 1. Memilih ikon tambah “+” |  |
|  | 1. Menampilkan *list* pilihan untuk setiap jenis *form* |

**Tabel 4.15 Scenario Use Case Authorizer**

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| 1. Admin memilih menu *authorizer* |  |
|  | 1. Menampilkan sub menu *user* |

**Tabel 4.16 Scenario Use Case Laporan**

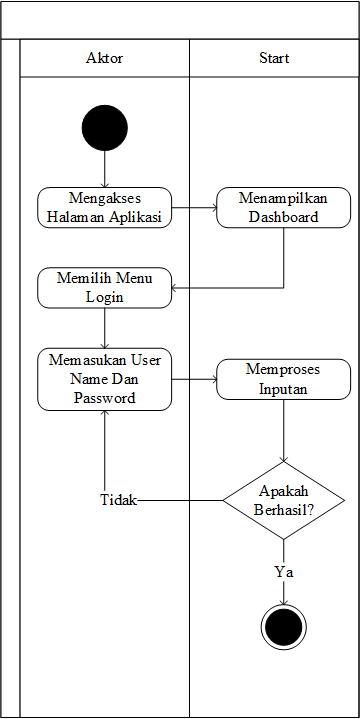
|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| 1. Admin memilih menu laporan |  |
|  | 1. Menampilkan sub menu |
| 1. Memilih laporan kuesioner |  |
|  | 1. Menampilkan list pengisi kuesioner |

1. Merancang *Activity Diagram*

Aktivitas selanjutnya setelah rancangan *use case* dihasilkan adalah merancang *activity diagram*. Aliran kerja atau aktivitas dari suatu sistem digambarkan oleh *activity diagram*. Rancangan *activity diagram* pengembangan sistem informasi *tracer study* digambarkan sebagaimana berikut :

1. *Activity Diagram Login*

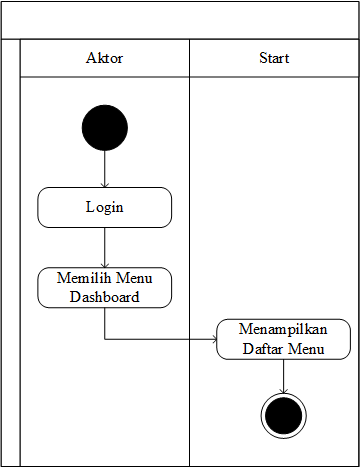
*Activity diagram login* merupakan penggambaran aktivitas yang terjadi antara aktor dan sistem. Dalam aktivitas tersebut terjadi interaksi dimana sistem melakukan proses terhadap inputan yang dilakukan oleh user. Adapun *activity diagram login* sebagai berikut :



**Gambar 4.3 Activity Diagram Login**

1. *Activity Diagram Dashboard*

*Activity diagram dashboard* merupakan kegiatan yang dilakukan oleh user untuk mengakses halaman utama sistem informasi. Kegiatan user tersebut digambarkan dalam *activity diagram* sebagaimana berikut :



**Gambar 4.4 Activity Diagram Dashboard**

Adapun untuk keseluruhan gambar *activity diagram*, dapat dilihat pada lampiran B.

1. *Elaboration*

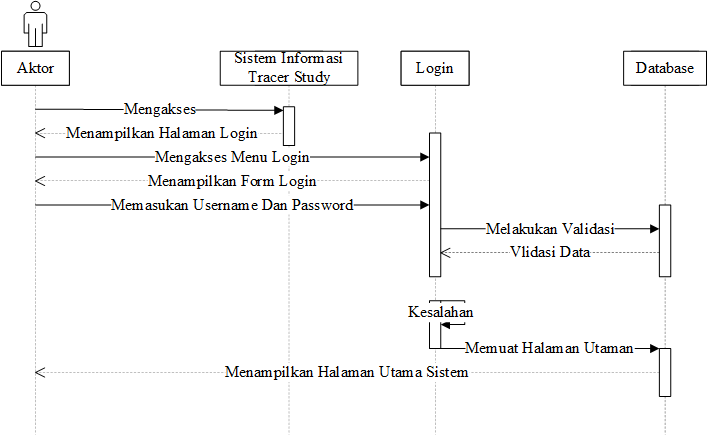
Tahapan perencanaan arsitektur sistem, dilakukan pada tahap *elaboration*. Pada tahapan ini dilakukan aktivitas perancangan *sequence diagram,* dan *class diagram.*

1. Merancang *Sequence Diagram*

*Sequence diagram* merupakan interaksi antar objek yang mengindentifikasikan komunikasi antar objek-objek tersebut yang digambarkan oleh diagram.

1. *Sequence Diagram Login*

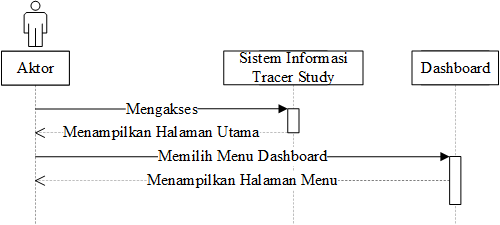
*Sequence diagram login* proses interaksi yagn terajdi antara aktor dengan sistem ketika pengoperasian awal sistem informasi *tracer study*. Adapun *sequence diagram login* sebagaimana berikut :



**Gambar 4.5 Sequence Diagram Login**

1. *Sequence Diagram Dashboard*

Aktivitas pertama aktor setelah masuk kedalam sistem informasi adalah mengakses menu *dashboard*. Menu dashboard merupakan halaman awal sistem informasi. Adapun aktivitas aktos dalam mengakses menu *dashboard* digambarkan pada *sequence* *diagram* sebagai berikut :

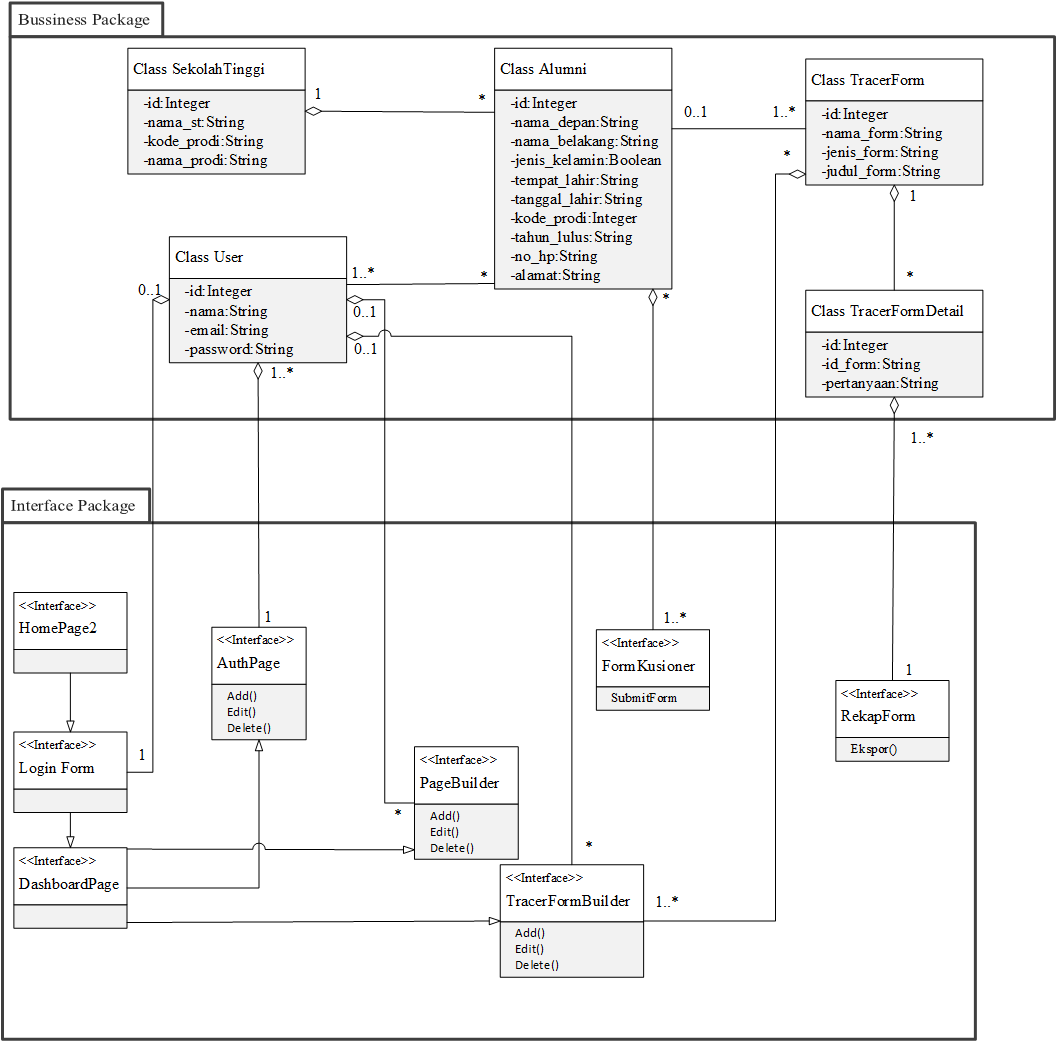


**Gambar 4.6 Sequence Diagram Dashboard**

Adapun untuk keseluruhan gambar *sequence diagram*, dapat dilihat pada lampiran C.

1. Merancang *Class Diagram*

*Class diagram* merupakan pendefinisian kelas-kelas yang akan dibuat dalam membangun sebuah sistem. *Class diagram* juga merupakan penggambaran struktur sistem yang dapat melakukan fungsi-fungsi sesuai kebutuhan sistem. *Class diagram* sistem informasi *tracer study* merupakan penggambaran dari kelas-kelas yang akan saling terhubung satu sama lain serta menjalankan fungsi-fungsi yang dibutuhkan saat akan memproses sistem tersebut. Berikut merupakan *class diagram*  sistem informasi *tracer study* :



**Gambar 4.7 Class Diagram Tracer Study**

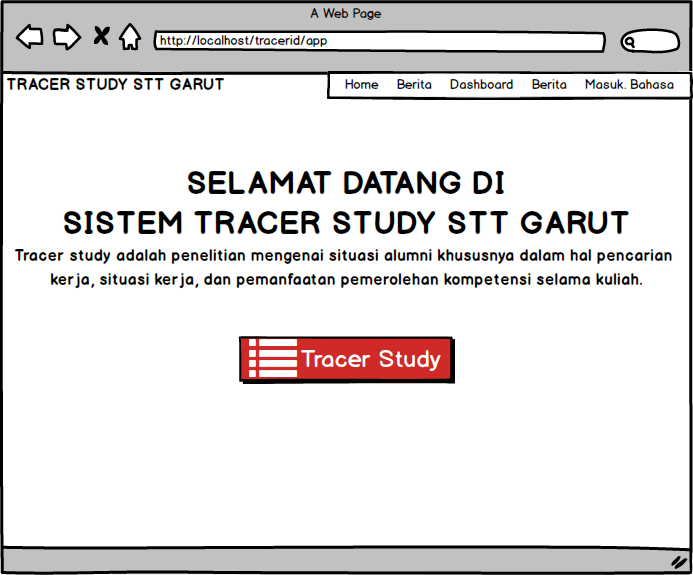
1. *Contruction*

*Contruction* merupakan tahapan awal dalam pengembangan hasil rancangan pada tahapan sebelumnya. Pada tahapan ini dilakukan perancangan *layout* serta merealisasikan rancangan kedalam Bahasa pemrograman.

1. Perancangan *Layout*

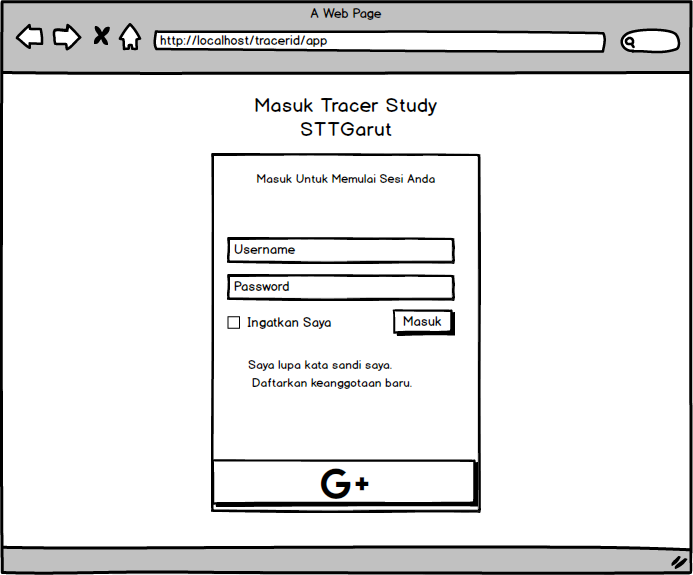
Perancangan *layout* merupakan aktivitas yang dilakukan dalam merancang tampilan suatu sistem. Adapun rancangan tampilan sistem informasi *tracer study* adalah sebagai berikut :

1. Tampilan Awal Sistem



**Gambar 4.8 Rancangan Tampilan Awal Sistem**

1. Halaman Login



**Gambar 4.9 Rancangan Tampilan Login**

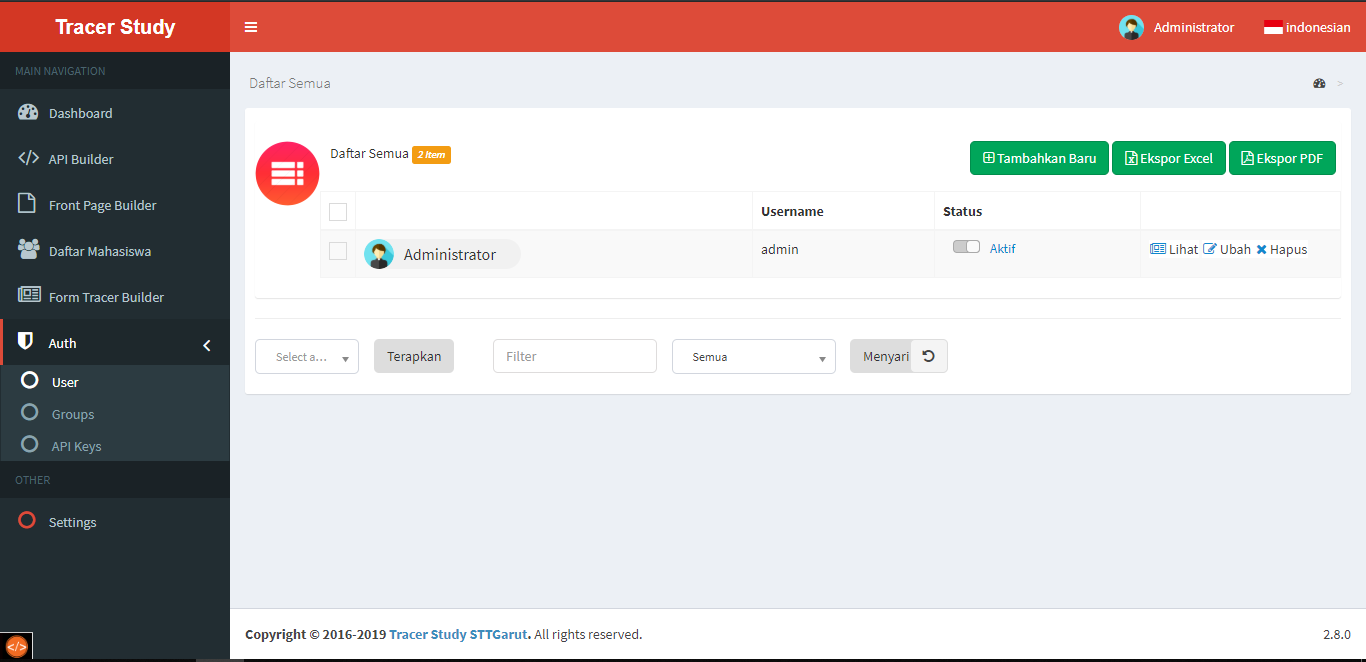
Adapun untuk keseluruhan gambar rancangan tampilan, dapat dilihat pada lampiran D.

1. Merealisasikan Rancangan Kedalam Bahasa Pemrograman

Pada tahapan ini dilakukan penerapan hasil rancangan kedalam pembuatan sistem secara langsung. Yang mana pada tahap ini sistem akan dibuat sebagaimana rancangan yang sebelumnya sudah dibuat. Dari mulai pembuatan *database*, sampai pada struktur menu sistem semuanya telah teralokasikan sebagaimana rancangan yang telah dibuat. Beriku adalah hasil realisasi rancangan kedalam Bahasa pemrograman :



**Gambar 4.10 Tampilan Awal Sistem**



**Gambar 4.11 Tampilan Menu Admin**

Adapun untuk keseluruhan gambar tampilan sistem, dapat dilihat pada lampiran E.

1. Pengujian

Tahap pengujian adalah tahap akhir dalam metodologi penelitian *rational unified process* ( RUP ). Pada tahap ini secara keseluruhan sistem diuji menu per menu. Adapun tahan pengujian ini sebagaimana tergambar pada tabel 4.17 :

**Tabel 4.17 Pengujian Sistem**