

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Sebagai acuan dalam penelitian ini, penulis mengkaji beberapa penelitian yang sejenis dan beberapa jurnal dengan pembahasan mengenai pengembangan sistem informasi pengelolaan tugas akhir. Serta, penerapan teori *Requirement Change Management* pada pengembangan sistem. Penelitian pertama yang dikaji dalam penelitian ini adalah penelitian mengenai, “Pengembangan Modul Pengelolaan Tugas Akhir Pada Sistem Informasi Akademik Perguruan Tinggi (Studi Kasus : STIKI Malang)”, yang dilakukan oleh Adji (2020). Penelitian ini membahas mengenai pengembangan proses pelaksanaan Tugas Akhir di tahun 2020, yang mana proses terjadi saat pelaksanaan tugas akhir meliputi proses pengajuan, bimbingan tugas akhir, hingga pengajuan seminar. Saat ini proses pelaksanaan tugas akhir mengalami beberapa perubahan diantaranya, terdapat adanya proses pengajuan judul tugas akhir oleh Koordinator laboratorium agar bisa disetujui dan bisa melanjutkan pada proses selanjutnya. Selain itu, adanya pengajuan ide judul tugas akhir yang ditawarkan oleh Koordinator laboratorium agar bisa digunakan oleh mahasiswa.

Pada penelitian selanjutnya yaitu mengenai, “Pengembangan Sistem Informasi Penyelesaian Tugas Akhir pada Program Studi PGPAUD UPI Kampus Tasikmalaya” yang telah dilakukan oleh Radliya & Alfaridzi (2017). Penelitian ini membahas mengenai pengembangan sistem informasi penyelesaian tugas akhir, yang mana pada penelitian ini bertujuan untuk dapat membantu dalam melakukan

validasi terhadap kelayakan tugas akhir mahasiswa untuk mengajukan skripsi, serta pendaftaran sidang skripsi. Namun, pada pengembangan sistem ini belum adanya fitur untuk pelayanan bimbingan, disamping itu, tidak adanya proses pengaturan jadwal seminar yang telah diajukan oleh mahasiswa.

Selanjutnya, penelitian mengenai “Aplikasi Pengontrolan Pelaksanaan Tugas Akhir” yang telah dilakukan oleh Yanto & Prasetyo (2015). Penelitian ini membahas mengenai aplikasi pengontrolan pelaksanaan tugas akhir, yang mana, akan dikembangkannya aplikasi yang dapat menghitung batas kuota mahasiswa bimbingan sesuai dengan peminatan. Selain itu, adanya proses penentuan dosen pembimbing berdasarkan pada peminatan tugas akhir mahasiswa, serta pencatatan hasil bimbingan yang telah dilakukan oleh dosen pembimbing dan mahasiswa. Namun, pada aplikasi ini belum adanya fitur catatan perbaikan dari dosen penguji setelah melakukan seminar, selain itu penelitian yang dilakukan oleh Yanto dan Prasetyo menggunakan metode waterfall.

Selain mengambil penelitian terdahulu mengenai penelitian tentang sistem informasi pengelolaan tugas akhir. Peneliti juga mengambil referensi terkait penelitian mengenai “*Effect of Project Management in Requirements Engineering and Requirements Change Management Processes for Global Software Development*” yang dilakukan oleh Shafiq et al., (2018) yang mana penelitian ini dilakukan bertujuan untuk menyediakan *Framework* atau kerangka kerja untuk proses *Requirement Change Management. Framework* yang diusulkan, dikategorikan menjadi tiga fase utama yaitu Fase Komunikasi (*Communication Phase*), Fase Analisa (*Analysis Phase*), dan Fase Pengembangan dan implementasi

(Development and Implementation Phase), sehingga, setiap fase memiliki tugas tertentu.

Requirement Change Management Framework pada penelitian yang dilakukan oleh Shafiq et al., (2018), dikutip kembali pada penelitian yang dilakukan oleh Institute of Electrical and Electronics Engineers, n.d. (2018) yang berjudul “*AZ-Model of software requirements change management in global software development*” penelitian ini juga mengusulkan framework pada penelitian Shafiq et al., (2018) menjadi Model AZ- Model RCM yang memiliki kemampuan untuk menangani Requirement Change dengan cara yang efisien. Model AZ-Model yang diusulkan memiliki kinerja terbaik untuk *Requirement Change Management*.

Dari hasil kajian diatas, dapat disimpulkan bahwa sistem informasi yang akan dibangun pada penelitian ini, sebagian besar memiliki persamaan fitur yaitu fitur dalam pengelolaan Tugas Akhir. Persamaan fitur yang paling menonjol ditunjukkan pada penelitian yang dilakukan oleh Adji (2020), mengenai “Pengembangan Modul Pengelolaan Tugas Akhir pada Sistem Informasi Akademik Perguruan Tinggi (Studi Kasus: STIKI Malang)” yang mana, fitur pengelolaan Tugas Akhir yang dikembangkan hampir sama. Disamping itu, Studi Kasus yang diambil dari Instansi yang sama yaitu STIKI Malang. Namun, dikarenakan adanya perubahan alur pengelolaan Tugas Akhir yang terjadi yang dimulai dari kurikulum 2019 sampai saat ini, perubahan tersebut memiliki dampak yang signifikan terhadap alur dari Pengajuan Pra Tugas Akhir yang ada di STIKI Malang.

Maka dari itu, pada penelitian ini akan dikembangkan aplikasi modul pengelolaan tugas akhir di STIKI Malang yang diharapkan, dapat memenuhi

perubahan dari *requirement* tersebut. Modul tugas akhir ini juga akan terintegrasi dengan sistem informasi akademik yang ada di STIKI Malang. Selain itu, peneliti mengimplementasikan *framework requirement change management* yang dilakukan oleh Shafiq et al., (2018) yang mana, perubahan *requirement* yang terjadi pada proses pelaksanaan Tugas Akhir di STIKI Malang dapat diakomodasi menggunakan *Framework* tersebut.

2.2 Teori Terkait

2.2.1 Requirement Change Management

Requirement Change Management merupakan proses untuk menafsirkan, mengelola, menganalisis, melacak, dan mendokumentasi perubahan pada kebutuhan proses tersebut meliputi identifikasi, evaluasi, dan penerapan perubahan terhadap *requirement* tersebut, serta memastikan bahwa perubahan tersebut dikomunikasikan, didokumentasikan dan diintegrasikan dengan baik ke dalam proses pengembangan sistem menurut Jayatilleke & Lai, (2018). Tujuan utama dari proses perubahan ini adalah untuk memenuhi kebutuhan client yang terus berkembang, serta kemajuan teknologi bersamaan dengan mempertahankan cakupan proyek pengembangan sistem, kualitas serta waktu pengerjaan.

Metode dari proses *Requirement Change management* dibagi menjadi 3 fase pengerjaan menurut Shafiq et al., (2018) yaitu :

Fase Pertama : Komunikasi (*Communication*)

Pada fase ini terjadinya proses komunikasi dengan *client* mengenai perubahan apa yang dibutuhkan, permintaan perubahan tersebut menjadi proses

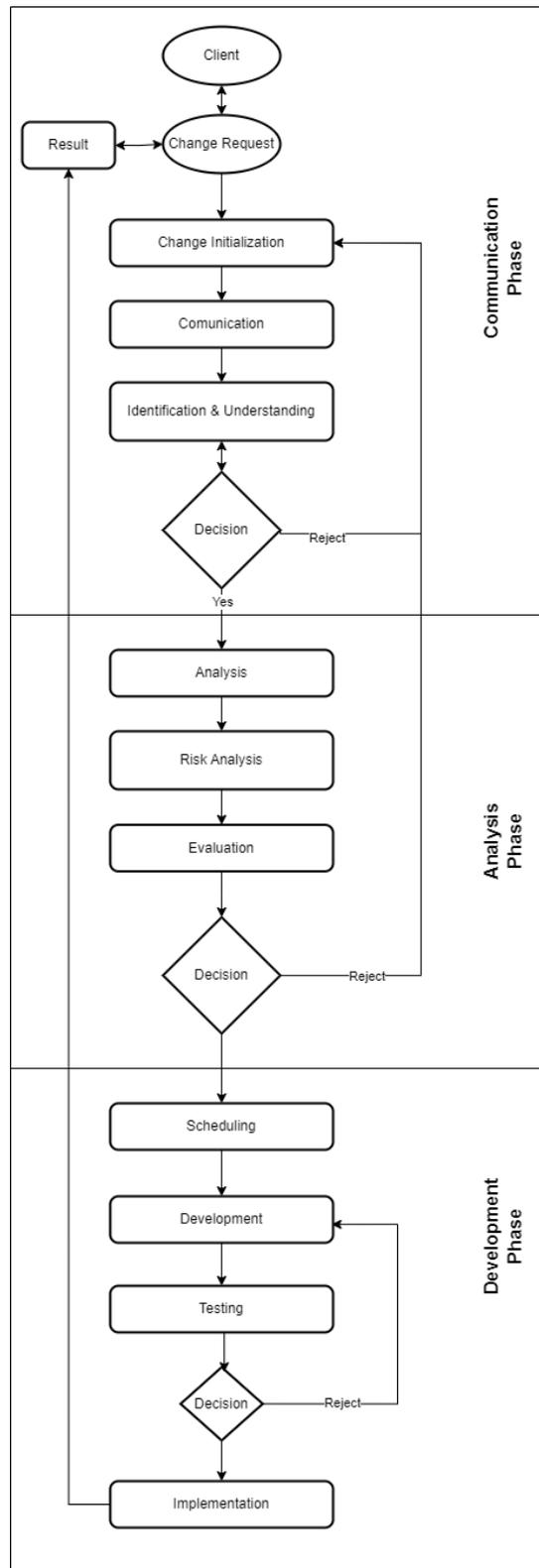
awal dari inisiasi perubahan *requirement*. Komunikasi juga dilakukan untuk mendapatkan data tentang perubahan yang terjadi, setelah proses pengumpulan data dilakukan selanjutnya yaitu, proses identifikasi dan memahami perubahan. Apabila perubahan tersebut bisa diidentifikasi maka akan dilanjutkan ke fase berikutnya dan jika tidak bisa diidentifikasi maka akan kembali ke proses pengumpulan data.

Fase Kedua : Analisa (*Analysis*)

Pada fase ini terjadinya proses analisa dan evaluasi terkait dengan perubahan yang telah diterima pada fase pertama. Pada proses ini dilakukan Analisa terkait perubahan yang terjadi dengan melakukan pengecekan berupa rincian perubahan *requirement* serta dampak dari perubahan tersebut dengan sistem yang ada sekarang, setelah dilakukan proses Analisa selanjutnya dievaluasi apakah dampak serta perubahan yang terjadi dapat dilanjutkan pada fase selanjutnya, apabila hasil dari evaluasi tersebut diterima maka dilakukan proses perancangan terkait perubahan dari *requirement* yang telah divalidasi. Perancangan sistem dilakukan agar dapat diterapkan ke dalam sistem

Fase Ketiga : Pengembangan dan Implementasi (*Development and Implementation*)

Pada fase ini, setelah fase kedua menghasilkan rancangan dari sistem yang akan dikembangkan, selanjutnya dilakukan proses penjadwalan untuk mengembangkan sistem tersebut agar sesuai dengan waktu pengerjaan sistem. Setelah proses penjadwalan dilakukan maka perubahan *requirement* tersebut dapat dimulai proses pengembangan sistemnya.



Gambar 2. 1 Alur Requirement Change Management menurut Shafiq et al., (2018)

2.2.2 Tugas Akhir

Menurut Machmsud (2016), Tugas Akhir merupakan salah satu dari beberapa jenis karya ilmiah yang disusun oleh mahasiswa dibawah bimbingan dosen pembimbing. Sehingga, memenuhi kaidah dan standar kualitas yang sesuai dengan keilmuannya. Pada penulisan laporan tugas akhir, harus mengikuti standar dan pedoman teknik baik dari segi penulisan. Hasil dari tugas akhir tersebut nantinya akan diterbitkan oleh masing-masing jurusan atau fakultas dan literatur yang berhubungan dengan penulisan karya ilmiah

Sedangkan menurut Havaluddin et al., (2021), Tugas Akhir merupakan salah satu syarat dalam penyelesaian jenjang pendidikan formal pada suatu perguruan tinggi. Mahasiswa harus menyelesaikan tugas akhir tersebut hingga dinyatakan sebagai Sarjana Strata 1 (S1). Berdasarkan dari penjabaran diatas mengenai pengertian dari tugas akhir, maka dapat disimpulkan bahwa tugas akhir merupakan bagian dari tugas yang harus diselesaikan mahasiswa sebagai persyaratan mahasiswa tersebut dinyatakan lulus sebagai Sarjana Strata 1 (S1). Hasil akhir dari Tugas Akhir tersebut harus memenuhi kaidah standar penulisan serta kualitas yang sesuai dengan rumpun keilmuannya.

Tugas Akhir yang dituliskan oleh mahasiswa dapat memberikan peluang untuk melatih diri dalam hal mengemukakan pendapat dan pemecahan masalah secara mandiri dan ilmiah, disamping itu dengan adanya tugas akhir mahasiswa dapat mengimplementasikan pengalam dan ketrampilan yang diperoleh selama berada di bangku kuliah.

2.2.3 CodeIgniter

CodeIgniter merupakan salah satu *framework* yang dibuat menggunakan bahasa pemrograman PHP yang bertujuan untuk memudahkan *programmer* untuk mengembangkan aplikasi berbasis *website*. CodeIgniter bersifat *open source* dan menggunakan model berbasis MVC (*Model View Controller*) yang mana merupakan konsep model modern yang saat ini banyak digunakan.

CodeIgniter juga menyediakan beberapa *library* dan *helper* yang berguna untuk mempercepat proses pengembangan sistem berbasis web. Dalam penggunaannya pun programmer cukup memilih komponen yang disediakan di *Library* CodeIgniter dan dikombinasikan hingga menjadi sebuah bentuk atau fungsi yang lebih kompleks. selain itu, CodeIgniter juga menyediakan komponen untuk membuat *library* sendiri.

2.2.4 Use Case Diagram

Menurut Musthofa & Adiguna, (2022) *Use Case Diagram* merupakan pemodelan yang dibuat untuk menggambarkan kelakuan (*behavior*) sistem informasi yang akan dibuat, biasanya *use case diagram* digunakan untuk mengetahui fungsi apa saja yang akan ada pada sistem informasi yang akan dikembangkan dan siapa saja yang berhak untuk mengakses dan menggunakan fungsi - fungsi tersebut.

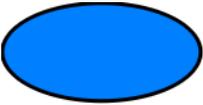
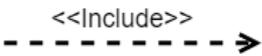
Bisa dibilang, *use case diagram* merupakan representasi secara grafis yang menggambarkan hubungan relasi antar aktor serta interaksi antara satu atau lebih aktor dengan sistem yang akan dikembangkan. Selain itu, *use case diagram* juga

bisa digunakan untuk mengetahui fungsi apa saja yang ada dalam didalam sebuah sistem.

Tujuan dari dibuatnya *use case diagram* yaitu untuk mempermudah *client* atau pihak yang memiliki kepentingan terhadap sistem tersebut untuk memahami dan berkomunikasi terkait rancangan dari alur sistem yang akan dibuat, selain itu juga untuk memudahkan pemahaman akan requirement atau kebutuhan sistem.

Berikut merupakan simbol yang ada pada *use case diagram* :

Tabel 2. 1 Simbol *use case diagram*

Simbol	Nama	Keterangan
 Actor	<i>Actor</i>	Entitas yang berinteraksi dengan sistem
	<i>Use Case</i>	Aktivitas yang dapat dilakukan <i>actor</i> pada sistem
	<i>Association</i>	Hubungan antara <i>actor</i> dan <i>use case</i>
	<i>System</i>	Sistem yang akan dikembangkan
	<i>Include</i>	Satu <i>use case</i> termasuk bagian dari <i>use case</i> lain
	<i>Extend</i>	Satu <i>use case</i> dapat diperluas dengan <i>use case</i> lain
	<i>Dependency</i>	Ketergantungan antara elemen - elemen diagram
	<i>Generalization</i>	Satu <i>actor</i> atau <i>use case</i> merupakan generalisasi dari yang lain

	<i>Realization</i>	Implementasi dari satu use case oleh yang lain
	<i>Collaboration</i>	Dua atau lebih actor dan use case yang terhubung
	<i>Note</i>	Penjelasan tambahan terkait elemen-elemen diagram
	<i>Anchor</i>	Hubungan <i>text note</i> dengan elemen - elemen diagram lain

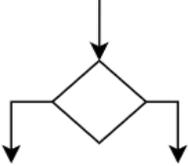
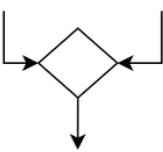
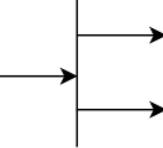
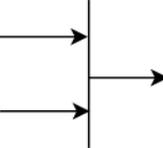
2.2.5 Activity Diagram

Menurut Musthofa & Adiguna, (2022) *Activity Diagram* merupakan penggambaran dari aliran kerja, atau aktivitas dari sebuah sistem informasi yang akan dikembangkan, bisa juga digunakan untuk menggambarkan proses bisnis atau menu yang ada pada sistem. *Activity diagram* biasanya digunakan untuk mendefinisikan atau mengelompokan alur dan tampilan dari sistem tersebut.

Fungsi dari *activity diagram* yaitu untuk memperlihatkan urutan dari aktivitas proses pada sistem, lalu untuk membantu *client* dalam memahami proses sistem secara keseluruhan. *Activity Diagram* dibuat berdasarkan pada beberapa *Use Case*.

Berikut merupakan komponen simbol yang ada pada *Activity diagram* :

Tabel 2. 2 Simbol *activity diagram*

Simbol	Nama	Keterangan
	<i>Initial State</i>	Awal dimulainya suatu aliran kerja pada <i>activity diagram</i> dan pada sebuah <i>activity diagram</i> hanya terdapat satu <i>initial state</i>
	<i>Final State</i>	Bagian akhir dari suatu aliran kerja pada sebuah <i>activity diagram</i> dan bisa terdapat lebih dari satu <i>final state</i>
	<i>Activity</i>	Aktivitas atau pekerjaan yang dilakukan dalam aliran kerja
	<i>Decision</i>	menggambarkan pilihan kondisi dimana ada kemungkinan perbedaan transisi, untuk memastikan bahwa aliran kerja dapat mengalir ke lebih dari satu jalur
	<i>Merge</i>	Berfungsi untuk menggabungkan kembali aliran kerja yang sebelumnya telah dipecah oleh <i>decision</i>
	<i>Transition / Association</i>	Menghubungkan aktivitas selanjutnya setelah aktivitas sebelumnya
	<i>Synchronization Fork</i>	memecah <i>behavior</i> menjadi aktivitas paralel
	<i>Synchronization Join</i>	Digunakan untuk menghubungkan kembali aktivitas yang paralel

2.2.6 Pemodelan Data (*Data Modeling*)

Menurut Fikry, (2019) Pemodelan Data merupakan sekumpulan konsep yang digunakan untuk menjelaskan struktur dari basis data (*database structure*) dan memberikan gambaran tingkat-tingkat abstraksi data (*data abstraction*).

Bisa disimpulkan pemodelan Data merupakan proses untuk merancang diagram deskriptif hubungan antara berbagai jenis informasi yang akan menghasilkan hasil akhir berupa *database*. Pemodelan Data juga bisa diartikan sebagai proses yang digunakan untuk menentukan dan menganalisa kebutuhan data yang diperlukan untuk mendukung proses bisnis dalam lingkup yang sesuai dengan sistem yang akan dikembangkan

Jenis model data yang digunakan dalam pemodelan data yaitu sebagai berikut :

1. Konseptual (*Conceptual Data Model*)

Conceptual Data Model (CDM) dapat didefinisikan yaitu tahapan dimana mengumpulkan informasi mengenai requirement yang dibutuhkan dalam suatu database dan membuat rancangan database yang masih berbentuk secara konsep secara general dimana hanya memikirkan tentang entitas – entitas apa saja yang akan ada pada *database*. Hasil dari tahapan *conceptual data model* adalah sebuah *Entity Relationship Diagram* (ERD) yang menggambarkan relasi antar *entity* serta atribut – atribut yang ada di dalam *entity*.

2. Logis (*Logical Data Model*)

Logical Data Model merupakan tahap dimana menganalisis *attribute*, *entity* dan relasi yang sudah ditentukan di tahap *conceptual data model* agar bisa diubah menjadi struktur *database* yang ada pada *relational database*. Hasil dari tahapan