

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian sebelumnya telah dilakukan oleh N. Handayani[5] pada tahun 2020. Pembuatan sistem ini dilakukan karena proses pengajuan anggaran kegiatan, laporan kegiatan, dan pengelolaan data unit kerja, pada Universitas Trilogi belum dilakukan secara konvensional alat bukti anggaran saat ini unit kerja mengajukan dan proses laporan kegiatan masih berupa kertas dan nota pembayaran saja. Penelitian tersebut menjelaskan bahwa sistem ini dibangun menggunakan metode *waterfall*, *mysql*, dan menggunakan metode perancangan sistem *Entity Relationship Diagram (ERD)* dan *Data Flow Diagram (DFD)*. Hasil penelitian ini berupa aplikasi yang terdapat beberapa *user level*, yaitu Unit Kerja, Atasan Langsung (Dekan), Wakil Rektor 1 dan, Rektor dan Bagian Keuangan. Masing-masing aktor mempunyai hak akses yang berbeda dalam melakukan aktivitas di aplikasi tersebut. Yang diharapkan pengguna jadi lebih mudah dalam proses pengajuan anggaran dan laporan kegiatan.

Pada penelitian lain yang telah dilakukan oleh A. Rasyid, F. Pradana, dan D. Rusdianto[6] pada tahun 2018. Permasalahan yang dihadapi dari pihak kemahasiswaan yaitu kesalahan penulisan format dari proposal mahasiswa, mahasiswa tidak mengetahui prosedur pengajuan, berkas-berkas yang dikumpulkan mahasiswa kurang lengkap, dan banyaknya mahasiswa yang bertanya tahapan proposalnya saat ini. Sedangkan permasalahan dari mahasiswa, alur dari pengajuan kegiatan terlalu panjang dan jika ada beberapa pihak bersangkutan tidak hadir atau tidak dapat ditemui, akan menambah waktu yang diperlukan untuk mengajukan kegiatan. Penelitian ini menggunakan *framework* Laravel, sistem informasi pengajuan kegiatan ini telah diuji menggunakan metode *white box testing* pada pengujian unit dengan hasil kompleksitas logika rata-rata adalah 3. Sedangkan, pada *black box testing* dengan pengujian validasi diperoleh 100% valid untuk seluruh kasus uji.

Penelitian lainnya juga dilakukan oleh N. Ayu, S. Informasi, dan S. Komputer[7] pada tahun 2017. Untuk kegiatan pengabdian masyarakat bersama mahasiswa saat ini masih secara manual proses pengajuannya. Dilihat dari proses pengajuan surat ke unit P2M yang harus diteruskan ke bagian PKI dan menunggu konfirmasi kesediaan dosen untuk mengikuti

kegiatan tersebut. Sistem ini dibangun diagram konteks, *data flow* diagram level 0, ERD, konseptual *database*, struktur tabel dan *interface* sistem untuk pengembangan sistem tahap berikutnya. Hasil dari penelitian ini yang telah dikerjakan adalah dalam penelitian ini telah menciptakan perancangan suatu sistem untuk perancangan sistem pengajuan kegiatan pengabdian mahasiswa.

Penelitian yang lain juga telah dilakukan oleh A. Negoro dan Y. Dewantara[8] pada tahun 2018. Berdasarkan penelitian yang telah dilakukan dapat diketahui bahwa proses penyelenggaraan *event* masih manual, yakni menggunakan proposal berbentuk *hardcopy* untuk diajukan perijinannya. Proposal berbentuk *hardcopy* membutuhkan banyak tempat untuk pengarsipan, proses pembukuan, pencatatan, serta penginformasian ruangan masih manual sehingga penyelenggaraan *event* atau Wakil Dekan III harus mengkonfirmasi Sub. Bagian Umum dan Perlengkapan terlebih dahulu untuk menyelenggarakan dan mengizinkan *event*, belum adanya koordinasi antara satu dan lainnya sehingga dapat terjadi *crash*. Penelitian ini menggunakan pendekatan kualitatif dan sumber data dari penelitian ini menggunakan Data Primer dan Data Sekunder. Hasil dari penelitian ini adalah penyelenggara *event* dapat mengetahui ruangan yang tersedia untuk direservasi secara langsung dan otomatis beserta koordinasi dengan penyelenggaraan yang lain tentang siapa saja yang menggunakan ruangan.

Pada penelitian lainnya juga dilakukan oleh H. Rahmani, M. Sanjaya, dan D. Ananda[9] pada tahun 2018. Saat ini pengajuan proposal dan laporan kegiatan yang akan dilaksanakan oleh warga masih tergolong masalah karena banyaknya data yang tercecer. Hasil dari penelitian ini, berupa aplikasi berbasis web yang bertujuan untuk membantu pihak Kantor Desa Cipagalo serta warga dalam melakukan proses pengajuan proposal dan laporan kegiatan. Aplikasi ini dibangun dengan berbasis web dengan menggunakan *framework* Codeigniter dan menggunakan MySQL sebagai *database server* serta menggunakan teknik *black-box testing* yang difokuskan kepada keluaran yang dihasilkan oleh aplikasi. Adapun metode yang digunakan yaitu menggunakan metode *waterfall*. Hasil dari penelitian ini adalah sebagai aplikasi yang dapat membantu pekerjaan dalam mengelola proposal dan laporan agar lebih efisien.

Perbedaan sistem yang dibuat dengan sistem yang sudah ada yaitu di sistem ini selain mempermudah dalam proses pengajuan kegiatan, sistem ini dapat mengolah data pengajuan proposal kegiatan, data pengajuan LPJ kegiatan dan mengelola anggaran. Pada penelitian ini,

akan dibuat sistem informasi pengajuan kegiatan di gereja. Aplikasi web ini dibuat menggunakan metode *waterfall*, *native*, bahasa pemrograman PHP, *Database Developer* MySQL, *Web Server* Apache dan *tools editor* Sublime Text. Terdapat format pengajuan proposal yang sudah disediakan dan setiap proposal kegiatan harus disertai dengan dokumentasi. Hasil dari penelitian ini adalah sebuah sistem yang memudahkan jemaat yang akan mengajukan kegiatan, memudahkan jemaat dalam memantau proposal yang telah diajukan sudah sampai pada tahap mana, memudahkan pimpinan dalam memproses data-data pengajuan serta memudahkan bendahara gereja dalam mengelola data anggaran.

2.2 Landasan Teori

Dalam penelitian ini perlu adanya teori-teori yang mendasar sebagai penunjang proses penelitian. Teori-teori tersebut adalah sebagai berikut :

2.2.1 Sistem Informasi

Pada dasarnya, sistem informasi merupakan suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. Sistem informasi di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. [10]

2.2.2 Rekayasa Perangkat Lunak

Telah dijelaskan sebelumnya, bahwa sistem informasi adalah suatu sistem yang terdiri dari berbagai macam komponen untuk menyajikan suatu informasi. Sedangkan, rekayasa perangkat lunak (*system engineering*) itu sendiri merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin.

Berikut merupakan penjelasan dari *flowchart*, desain berbasis UML, dan *Use Case Diagram* :

A. Flowchart

Flowchart adalah suatu bagian dengan simbol-simbol yang menggambarkan urutan proses secara mendetail dan hubungan antara

suatu proses (intruksi) dengan proses lainnya dalam suatu program. Simbol *Flowchart* dapat dilihat pada Tabel 2.1. [11]

Tabel 2. 1 Simbol Flowchart

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

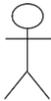
B. *Unified Modelling Language (UML)*

Unified Modelling Language (UML) adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam program berorientasi objek. Macam – macam dari *Unified Modelling Language (UML)* adalah sebagai berikut:

1. Use Case Diagram

Use case merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Simbol *Use Case* dapat dilihat pada Tabel 2.2

Tabel 2. 2 Simbol Use Case

No.	Simbol	Nama Simbol dan Keterangan
1.		<i>Actor</i> , mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> .
2.		<i>Dependency</i> , suatu proses ketika hubungan pada suatu elemen mandiri atau bisa disebut dengan independen, yang kemudian mempengaruhi elemen lain yang tidak bergantung pada elemen mandiri tersebut (<i>independent</i>).
3.		<i>Generalisation</i> , menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .
4.		<i>Include</i> , menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
5.		<i>Extend</i> , menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.
6.		<i>Association</i> , abstraksi dari penghubung antara aktor dengan <i>use case</i> .
7.		<i>System</i> , melakukan spesifikasi paket dengan menunjukkan sistem secara terbatas.

No.	Simbol	Nama Simbol dan Keterangan
8.		<i>Use case</i> , abstraksi dan interaksi antara sistem dan aktor.
9.		<i>Collaboration</i> , berbagai aturan dan elemen yang bekerja guna menyediakan <i>action</i> yang lebih besar dari jumlah elemennya.
10.		<i>Note</i> , elemen fisik yang terdapat ketika aplikasi mulai dijalankan dan menggambarkan suatu daya komputasi.

2. Sequence Diagram

Diagram *sequence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan di terima antar objek. Oleh karena itu, untuk menggambarkan diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek.[12] Simbol *Sequence* diagram dapat dilihat pada Tabel 2.3

Tabel 2. 3 Tabel *Sequence* Diagram

GAMBAR	NAMA	KETERANGAN
	<i>Actor</i>	Menggambarkan orang yang berinteraksi dengan sistem.
	<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan.
	<i>Boundary Class</i>	Menggambarkan sebuah penggambaran dari <i>form</i> .
	<i>Control Class</i>	Menggambarkan penghubung antara <i>boundary</i> dengan tabel.

GAMBAR	NAMA	KETERANGAN
	<i>A focus of Control & A life line</i>	Menggambarkan tempat mulai dan berakhirnya sebuah pesan.
	<i>A message</i>	Menggambarkan pengiriman pesan.

Pengkodean perangkat lunak dilakukan pada awal pengembangan perangkat lunak tanpa menggunakan tahapan pengembangan perangkat lunak dan menemui kendala dengan seiring perkembangan sistem perangkat yang semakin besar.

1. Metode Pengembangan Sistem

Metode yang digunakan yaitu metode *waterfall*. Pada tahap pengembangan sistem, peneliti menggunakan metode *waterfall*. Metode *waterfall* merupakan metode yang beruntun serta tidak berubah-ubah dan umum digunakan sebagai perancangan perangkat lunak[4]. Tahapan pengembangan sistem berdasarkan 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. Dari penelitian diatas, bahwa Gereja Bethel Tabernakel Menganti menginginkan sistem *online* pengajaran kegiatan di gereja berbasis *website*.

b) *Systems and software design*

Tahapan perancangan sistem mengalokasikan berbagai kebutuhan sistem, baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Dalam 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.

2. Metode *Black Box Testing*

Pengujian *blackbox* merupakan sebuah metode pengujian terhadap suatu kegunaan atau fungsionalitas sebuah sistem. Pengujian sistem dilakukan untuk melihat apakah sistem yang dibangun sudah sesuai dengan keinginan dari pelanggan dan layak untuk digunakan. Tujuan dari metode *blackbox* untuk mengetahui apakah sistem telah benar menampilkan kesalahan yang ada jika terjadi *human error*. Pada pengujian *blackbox* hanya mengambil hasil *output* melalui data uji dan mengecek fungsionalitas dari *software*.

2.2.3 Pemrograman Berorientasi Objek

Pemrograman berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Pemrograman berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis.

2.2.4 Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

A. ERD (*Entity Relationship Diagram*)

ERD (*Entity Relationship Diagram*) adalah suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.[13] Berikut komponen dari ERD :

1. Entitas (*Entity*)

Objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain.

2. Atribut

Berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain. Berikut macam-macam atribut :

a. Atribut *key*

Satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data (*Row/Record*) dalam tabel secara unik. Dikatakan unik jika pada atribut yang dijadikan *key* tidak ada baris data dengan nilai yang sama.

b. Atribut *simple*

Atribut yang bernilai atomic, tidak dapat dipecah/dipilah lagi.

c. Atribut *multivalued*

Nilai dari suatu attribute yang mempunyai lebih dari satu (*multivalued*) nilai dari atribut yang bersangkutan.

d. Atribut *composite*

Suatu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu yang masih bisa dipecah lagi atau mempunyai sub atribut.

e. Atribut *derivative*

Atribut yang tidak harus disimpan dalam *database* atau atribut yang dihasilkan dari atribut lain atau dari suatu *relationship*.

3. Hubungan/Relasi

Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Berikut macam-macam relasi :

a. *One to one*

Setiap anggota entitas A hanya boleh berhubungan dengan satu anggota entitas B begitu pula sebaliknya.

b. *One to many*

Setiap anggota entitas A dapat berhubungan dengan lebih dari satu anggota entitas B tetapi tidak sebaliknya.

c. *Many to many*

Setiap entitas A dapat berhubungan dengan banyak entitas himpunan entitas B dan demikian pula sebaliknya.

2.3 Agama

Agama adalah sebuah koleksi terorganisir dari kepercayaan, sistem budaya, dan pandangan dunia yang menghubungkan manusia dengan tatanan/perintah dari kehidupan. Agama di Indonesia memegang peranan penting dalam kehidupan masyarakat. Dalam UUD 1945

dinyatakan bahwa “tiap-tiap penduduk diberikan kebebasan untuk memilih dan mempraktikkan kepercayaannya” dan “menjamin semuanya akan kebebasan untuk menyembah, menurut agama dan kepercayaannya”. [14]

Gereja merupakan sekumpulan orang percaya yang bersekutu untuk beribadah kepada Tuhan. Dengan adanya perkembangan gereja yang semakin luas pada setiap jamannya, maka kemudian gereja dibagi kedalam wilayah-wilayah dan tempat yang tetap untuk beribadah, dari hal itu kemudian berkembanglah pengertian akan sebuah gereja, berbagai pengertian dan pemaknaan tersebut sebenarnya menuju kepada esensi yang sama secara non fisik mengenai arti gereja tersebut. Dalam perkembangannya, secara fisik orang mengenal gereja sebagai sebuah bangunan tempat umat Kristiani berkumpul untuk beribadah. Sebenarnya, bangunan gereja tersebut merupakan representasi makna dari gereja sebagai jemaat yang dinaunginya. [14]

Ada yang berpendapat bahwa kegiatan gereja tidak bisa dipersempit pada hal-hal yang abstrak atau teologis. Gereja justru harus menunjukkan peranan kepeduliannya pada isu-isu sosial yang sangat konkrit, misalnya bagaimana membangun pemerintahan yang baik. Sehingga gereja haruslah tetap memberikan warna dalam kehidupan bermasyarakat di Indonesia, dengan terus memberikan inspirasi di tengah-tengah masyarakat yang majemuk di bangsa ini. [15]