2. LANDASAN TEORI

2.1 Tinjauan Pustaka

Pada sub bab ini akan dijelaskan landasan teori yang digunakan dalam proses pembuatan skripsi ini.

2.1.1 Sistem Informasi Manajemen

Menurut Pahlephi (2022), sistem informasi manajemen (SIM) dirancang untuk menyediakan informasi yang mendukung pengambilan keputusan, pengelolaan, dan operasi dalam suatu organisasi atau perusahaan. SIM terbukti bermanfaat dalam meningkatkan efektivitas dan efisiensi pengolahan data, serta dapat meminimalisasi biaya dan meningkatkan produktivitas perusahaan atau organisasi. Dalam konteks tugas akhir, sistem informasi manajemen digunakan untuk mendukung proses penyusunan tugas akhir oleh mahasiswa, mulai dari pendaftaran topik hingga kelulusan atau yudisium. Dengan adanya SIM, diharapkan dapat mempermudah semua pihak yang terlibat dalam manajemen tugas akhir, seperti penyimpanan dokumen persyaratan tugas akhir, penyebaran informasi terkait tugas akhir, dan aspek lainnya.

2.1.2 Proses Tugas Akhir di UK Petra

Tugas akhir adalah istilah umum yang mencakup berbagai bentuk proyek atau penelitian yang harus diselesaikan oleh mahasiswa untuk menyelesaikan program studi mereka. Bentuk tugas akhir bisa bervariasi tergantung pada jenjang pendidikan dan program studi yang ditempuh. Contoh bentuk tugas akhir termasuk skripsi, tesis, disertasi, proyek desain, atau laporan kerja praktik (Mahalia, 2023). Melalui tugas akhir, mahasiswa diharapkan dapat menyajikan temuan-temuan atau kontribusi ilmiah mereka terhadap topik yang dipilih. Dalam pengerjaannya, tugas akhir melibatkan beberapa aktivitas yang harus dilakukan secara berurutan, mulai dari pembuatan proposal hingga sidang akhir. Menurut Handayani et al. (2021), proses pengerjaan tugas akhir melibatkan beberapa tahap, antara lain pengajuan judul, pengajuan proposal, seminar proposal, penelitian, sidang akhir, dan proses revisi apabila hasil ujian sidang diterima dengan revisi. Setiap perguruan tinggi atau program studi mungkin memiliki format dan alur tugas akhir yang berbeda-beda. Di UK Petra, setiap program studi memiliki alur tugas akhir yang berbeda-beda. Berdasarkan hasil survei ke beberapa perwakilan

program studi di UK Petra dan hasil diskusi dengan PPSI, berikut ini adalah alur tugas akhir secara umum di UK Petra:

1. Pemilihan Topik → Mahasiswa mencari topik dan dosen pembimbing secara mandiri sesuai bidang studinya. Selain mencari topik sendiri, mahasiswa juga bisa mengambil tawaran topik dosen bila ada dosen yang mengajukan tawaran. Setelah mendapatkan topik dan dosen pembimbing, mahasiswa konsultasi terkait topik tugas akhirnya. Proses ini dilakukan di luar sistem.
2. Pengajuan Topik → Mahasiswa mendaftarkan topik tugas akhir di *website* SIM-TA. Dosen pembimbing menerima dan menyetujui topik tugas akhir mahasiswa lewat *website* SIM-TA.
3. Pengerjaan dan Bimbingan Proposal → Setelah disetujui oleh dosen pembimbing, mahasiswa bisa mulai melakukan bimbingan dan pembuatan proposal tugas akhir. Mahasiswa bisa memasukkan data bimbingan di *website* SIM-TA.
4. Pengumpulan Proposal untuk Maju Sidang Proposal → Setelah selesai membuat proposal dan sudah dikonsultasikan dengan dosen pembimbing lewat bimbingan. Mahasiswa mengumpulkan proposal dengan cara mengunggah *file* proposal di *website* SIM-TA. Dengan mengumpulkan proposal, mahasiswa siap maju sidang proposal dan koordinator tugas akhir bisa menambahkan jadwal sidang di *website* SIM-TA.
5. Sidang Proposal → Mahasiswa mengikuti sidang proposal sesuai jadwal yang dibuat oleh koordinator tugas akhir.
6. Pengerjaan Revisi Proposal → Setelah melaksanakan sidang proposal, bagi mahasiswa yang lulus dengan revisi akan mengerjakan revisi proposal sesuai masukan revisi dari dosen penguji dan pembimbing pada bagian berita acara revisi di *website* SIM-TA.
7. Pengumpulan Proposal Final → Mahasiswa mengumpulkan *file* revisi proposal di *website* SIM-TA. Apabila semua dosen penguji dan pembimbing sudah menyetujui revisi di berita acara revisi maka revisi proposal mahasiswa diterima.
8. Pengerjaan dan Bimbingan Tugas Akhir → Mahasiswa mulai mengerjakan tugas akhir dan melakukan bimbingan dengan dosen pembimbing secara berkala sesuai ketentuan jumlah minimal bimbingan untuk maju sidang akhir.
9. Pengajuan Maju Sidang Akhir → Apabila jumlah bimbingan mahasiswa sudah memenuhi ketentuan, dosen pembimbing pertama dapat mengajukan mahasiswa untuk maju sidang akhir lewat *website* SIM-TA. Setelah mahasiswa diajukan untuk maju sidang akhir, koordinator tugas akhir bisa menambahkan jadwal sidang akhir di *website* SIM-TA.

10. Pengumpulan Buku Skripsi → Mahasiswa mengumpulkan buku skripsi dan berkas lainnya di *website* SIM-TA.
11. Sidang Akhir → Mahasiswa mengikuti sidang akhir sesuai jadwal yang dibuat oleh koordinator tugas akhir.
12. Pengerjaan Revisi Buku Skripsi → Setelah melaksanakan sidang akhir, bagi mahasiswa yang lulus dengan revisi akan mengerjakan revisi buku skripsi sesuai masukan revisi dari dosen penguji dan pembimbing pada bagian berita acara revisi di *website* SIM-TA.
13. Pengumpulan Buku Skripsi Setelah Revisi dan Berkas Keperluan Tugas Akhir → Mahasiswa mengumpulkan *file* revisi buku skripsi di *website* SIM-TA. Apabila semua dosen penguji dan pembimbing sudah menyetujui revisi di berita acara revisi maka revisi buku skripsi mahasiswa diterima. Mahasiswa juga mengumpulkan berkas keperluan tugas akhir sesuai tanggal yang ditentukan.
14. Pengunggahan Buku Skripsi di Perpustakaan → Mahasiswa mengunggah buku skripsi secara mandiri di perpustakaan dengan mengikuti ketentuan yang berlaku.

2.1.3 Laravel

Laravel adalah sebuah *framework* pengembangan aplikasi web berbasis PHP yang bersifat *open-source*. Diciptakan oleh Taylor Otwell, Laravel awalnya dikembangkan untuk mengatasi kekurangan beberapa *framework* PHP lainnya seperti CodeIgniter, terutama dalam hal fitur-fitur seperti otentikasi dan otorisasi pengguna. *Framework* ini menawarkan fitur-fitur yang memudahkan pengembangan aplikasi web, termasuk sistem *routing*, manajemen *session* dan *cache*, ORM (*Object-Relational Mapping*) dengan *Eloquent*, *templating* menggunakan *Blade*, serta berbagai opsi untuk otentikasi, keamanan, dan manajemen basis data (Jalli, 2022). Dengan Laravel, pengembang dapat membangun aplikasi web kompleks dengan cepat dan efisien.

2.1.4 *Design Pattern* MVC *(Model-View-Controller)*

MVC adalah singkatan dari *Model-View-Controller*, salah satu *Design Pattern* yang digunakan untuk memisahkan logika aplikasi menjadi tiga komponen utama yang saling terhubung yakni *Model*, *View*, dan *Controller*. Masing-masing komponen memiliki tugas yang berbeda (Aprilia, 2021). *Model* merupakan bagian yang mengelola data dan berhubungan langsung dengan *database*. *View* merupakan bagian yang akan menyajikan tampilan informasi kepada pengguna. *Controller* adalah bagian yang menghubungkan *model* dan *view* dalam setiap proses *request* dari pengguna. Alur komunikasi dalam *design pattern* MVC dimulai dari *view*

(pengguna berinteraksi dengan layar), kemudian *controller* merespon dan meneruskan ke *model*, dan kembali lagi ke *controller* hingga ke *view* (Firdaus, 2019). *Design pattern* MVC membantu dalam mengorganisir kode sehingga lebih mudah untuk dipahami, dipelihara dan dimodifikasi.

2.1.5 *Eloquent* ORM *(Object-Relational Mapping)*

Eloquent ORM adalah salah satu fitur utama dari *framework* Laravel yang dirancang untuk menyederhanakan interaksi dengan *database*. Laravel *Eloquent* ORM memungkinkan *developer* untuk melakukan operasi *database* seperti membuat, membaca, memperbarui, dan menghapus (CRUD) tanpa menulis *query* SQL secara eksplisit (Shukla, 2023). Setiap tabel *database* memiliki "Model" terkait yang digunakan untuk berinteraksi dengan tabel tersebut. Model ini digunakan untuk mendefinisikan struktur dari tabel seperti nama tabel, atribut kolom dan metode untuk menggambarkan relasi *database*. Selain itu, Laravel *Eloquent* ORM juga menyediakan fitur-fitur seperti otomatisasi *timestamp* untuk mengelola *timestamp* (*created_at* dan *updated_at*) dari *record* data serta validasi data yang masuk ke dalam *database*.

2.1.6 Blade *Templating Engine*

Blade *Templating Engine* adalah fitur kunci dalam *framework* Laravel yang memungkinkan *developer* untuk dengan mudah membuat tampilan web yang dinamis dan efisien. Blade menyediakan sintaks yang dapat memungkinkan penggabungan logika PHP dalam tampilan HTML. Dengan Blade, *developer* dapat menggunakan perulangan, pengkondisian, dan pemanggilan komponen tampilan secara langsung dalam *file-template* Blade (Ramos, 2024). Dalam Laravel Blade, *file* tampilan harus disimpan dengan ekstensi .blade.php di dalam *folder /resources/views*. Hal ini memungkinkan Laravel untuk mengenali dan mengurai *template* Blade dengan benar saat aplikasi web dijalankan. Keunggulan utama dalam menggunakan *template* Blade adalah developer dapat membuat *template* master yang kemudian dapat diperluas dengan *file template* yang lain.

2.1.7 Amazon S3 (*Simple Storage Service*)

Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan *cloud* yang populer yang disediakan oleh Amazon Web Services (AWS), menawarkan solusi penyimpanan yang aman, skalabilitas dan ketersediaan data. S3 memungkinkan *developer* untuk menyimpan dan mengambil data dari mana saja secara daring dengan keandalan yang tinggi dan *latency*

yang rendah. Fitur utama S3 termasuk kemampuan untuk menyimpan berkas besar, mengatur akses berkas secara fleksibel melalui kebijakan akses, dan replikasi data otomatis untuk meningkatkan ketahanan (What is Amazon S3?, n.d.). Dalam *framework* Laravel, *developer* dapat mengintegrasikan Amazon S3 sebagai penyimpanan berkas untuk mengelola berkas-berkas statis atau media seperti gambar, video, dan *file* audio (Rosenthal, 2023). Dengan menggunakan *library* resmi "AWS SDK *for* PHP" atau paket pihak ketiga seperti "league/flysystem-aws-s3-v3", *developer* dapat dengan mudah mengunggah, mengunduh, dan mengelola berkas di Amazon S3 secara langsung dari aplikasi Laravel.

2.2 Tinjauan Studi

Berikut ini adalah penelitian yang telah dilakukan dalam pembuatan sistem informasi manajemen tugas akhir.

2.2.1 Sistem Informasi Skripsi STMIK Primakara berbasis *Website* menggunakan *Framework* Laravel (Satwika et al., 2020)

Masalah yang diangkat dalam penelitian ini adalah pengelolaan kegiatan proposal dan skripsi, pencarian, rekap data, dan berkas-berkas yang dijadikan sebagai arsip masih dilakukan secara manual. Hal ini menyebabkan komisi skripsi harus merekap mahasiswa pendaftar secara manual sehingga proses pendaftaran memakan waktu yang cukup lama. Untuk mengatasi masalah tersebut, dibuatlah sebuah sistem informasi manajemen skripsi berbasis web menggunakan *framework* Laravel untuk *backend* dan *framework* Vue.js untuk *frontend*. Metodologi pengembangan perangkat lunak pada penelitian ini menggunakan *Software Development Life Cycle* (SDLC) dengan model *waterfall*. Kemudian, untuk perancangan sistem digunakan *Use Case Diagram*, *Sequence Diagram*, dan *Entity Relationship Diagram* (ERD).

Hasil penelitian ini berhasil merancang dan membangun sistem informasi manajemen skripsi yang digunakan dalam proses pendaftaran dan pengelolaan mahasiswa yang mengambil skripsi. Hasil wawancara dengan komisi skripsi dan mahasiswa menunjukkan bahwa sistem ini mampu mempermudah proses pendaftaran dan membantu komisi skripsi dalam mengumpulkan data mahasiswa pendaftar. Namun, kekurangan dalam penelitian ini adalah bagian *frontend* dan *backend* dipisah menggunakan *framework* yang berbeda. Total ada tiga *frontend* dan satu *backend* yang dibuat dalam penelitian ini. Hal ini mengakibatkan pengembangan sistem memerlukan waktu dan tenaga yang cukup banyak.

2.2.2 Sistem Informasi Manajemen Skripsi Berbasis Web Di Universitas Muhammadiyah Gorontalo (Handayani et al., 2021)

Sama seperti penelitian pada butir 2.2.1, pada penelitian ini proses administrasi seperti pengelolaan jadwal ujian, penentuan dosen pembimbing dan penguji, serta pendataan mahasiswa yang melakukan skripsi masih dilakukan secara manual. Hal ini disebabkan oleh belum adanya *website* sistem informasi yang dapat mengintegrasikan proses administrasi ke dalam satu sistem. Penelitian ini membuat sebuah *website* sistem informasi menggunakan bahasa pemrograman PHP untuk tampilan serta MySQL untuk *database*. Metodologi pengembangan perangkat lunak pada penelitian ini menggunakan *Software Development Life Cycle* (SDLC) dengan model *waterfall*. Kemudian, untuk perancangan sistem digunakan Context Diagram, *Data-flow Diagram*, dan *Entity Relationship Diagram* (ERD).

Penelitian ini berhasil menghasilkan sebuah sistem berbasis web yang dapat mengintegrasikan seluruh proses administrasi dalam pengurusan skripsi dalam suatu sistem berbasis web. Proses administrasi tersebut antara lain penyebaran informasi atau pengumuman, penetapan jadwal ujian, penilaian hasil ujian, serta sarana bimbingan *online* melalui fitur *chat* yang terkomputerisasi. Keunggulan dalam penelitian ini adalah fitur bimbingan *online* melalui fitur *chat* yang terkomputerisasi. Selain keunggulan yang ada, terdapat kekurangan di mana tidak ada fitur untuk mengelola periode yang tersedia untuk mengambil skripsi sehingga data skripsi mahasiswa tidak bisa dipisahkan berdasarkan periode pengerjaan.