



BAB 2 LANDASAN TEORI

2.1 Penelitian Terdahulu

Kajian terhadap penelitian terdahulu berfungsi untuk menunjukkan posisi dan relevansi penelitian yang akan dilakukan, sebagai pembanding, referensi, dan bahan pertimbangan dalam menyelesaikan penelitian. Bagian ini memaparkan tentang informasi hasil penelitian sebelumnya, yang relevan dengan penelitian yang akan dilakukan.

Penelitian terdahulu yang pertama berjudul “Sistem Informasi Manajemen legalisir Online Berbasis Website” yang dilakukan oleh Dairoh, Riszki Wijayatun Pratiwi, Dwi Intan Af'dah, Sharfina Febbi Handayani, dan Ferian Andhika Toarsi pada tahun 2024. Dalam penelitian ini, hasil yang diperoleh adalah sistem informasi manajemen legalisir ijazah online berbasis website yang telah berjalan sesuai fungsi setiap aktor yang terlibat serta telah dilakukan pengujian. Adapun hasil pengujian *black box* menunjukkan sistem telah berfungsi dengan baik sesuai peran masing-masing aktor. Sementara itu, hasil evaluasi usability UI/UX dari 38 pengguna menunjukkan nilai sebesar 82% yang termasuk kedalam kategori sangat baik. Studi ini menjadi pembanding, terutama dalam penggunaan metode Waterfall, namun memiliki perbedaan dalam framework yang digunakan, dimana framework Laravel memiliki fitur bawaan lebih lengkap dan ekosistem yang lebih besar, dibandingkan codeigniter.

Penelitian terdahulu yang kedua dilakukan oleh Wahyu Pramusinto, Sri Mulyati, dan Safrina Amini pada tahun 2021, dengan judul “Implementasi E-Services Untuk Legalisir Ijazah dan Transkrip Nilai Pada Universitas Budi Luhur.”. Penelitian ini menghasilkan sebuah aplikasi berbasis website yang digunakan oleh alumni untuk melegalisir transkrip nilai dan ijazah, sebagai bagian dari peran Universitas Budi Luhur dalam melayani alumninya, di mana pun alumninya berada.



Aplikasi yang dibangun, menggunakan bahasa pemrograman PHP dan database MariaDB. Studi ini relevan sebagai pembanding karena juga membuat sistem berbasis website, namun dalam penelitian ini, fitur pembayaran belum terintegrasi dengan payment gateway, sedangkan pembengan sistem legalisir online memerlukan fitur pembayaran digital yang terintegrasi.

Penelitian yang ketiga di lakukan oleh Dian Pramadhana, Kurnia Astrin Febrianti, Riyan Farismana, dan A. Lubis Ghozali pada tahun 2023, dengan penelitian yang berjudul “Perancangan Sistem Informasi Legalisir Ijazah Online di Politeknik Negeri Indramayu”. Dari penelitian ini sistem informasi legalisir ijazah online ini berjalan dengan baik, dengan adanya sistem informasi legalisir ijazah dan transkrip nilai yang telah dirancang dapat mempermudah serta mempercepat proses permohonan legalisir ijazah bagi para alumni. Studi ini dijadikan pembanding karena mengambil permasalahan yang sama yakni legalisir dokumen, namun pada penelitian tersebut tidak dijelaskan bagaimana alur alumni melakukan legalisir.

Penelitian terdahulu yang keempat dilakukan oleh Yudha Permana dan Herry Derajad Wijaya pada tahun 2020. Dengan judul “Implementasi E-Legalisir Untuk Legalisir Ijazah dan Transkrip Online pada Fakultas Ilmu Komputer Universitas Mercu Buana”. Hasil penelitian ini adalah sistem E-legalisir fakultas ilmu komputer Universitas Mercu Buana yang telah siap digunakan. Pengujian black box memberikan hasil valid dari setiap fitur yang telah dikembangkan serta System Usability Scale yang memberikan kategori baik. Studi ini relevan sebagai pembanding karena juga membuat sistem legalisir berbasis website, namun memiliki perbedaan yakni pada pilihan metode pembayaran yang hanya menyediakan transfer bank, dan belum terintegrasi dengan payment gateway.

Penelitian terdahulu yang kelima dengan hasil berupa aplikasi legalisir online berbasis web servis pada IAIN Salatiga. dengan memanfaatkan fasilitas web servis yang disediakan



sistem informasi akademik IAIN Salatiga mempermudah alumni dalam melakukan legalisir secara online, yang dilakukan oleh Mei Prabowo dan Edi Kuswanto pada tahun 2020, dengan judul “Metode Extreme Programming Dalam Pengembangan Aplikasi Legalisir Online Berbasis Web Service”. Penelitian ini dijadikan pembandingan karena mengangkat permasalahan yang sama yaitu legalisir online, akan tetapi dalam penelitian tidak dijelaskan metode pembayaran selain tunai. Seperti transfer bank ataupun payment gateway untuk memudahkan transaksi.



Tabel 2.1 Penelitian Terdahulu

No	Nama Peneliti dan Judul	Platform	Website	DBMS	Output
	(Dairoh dkk., 2024). Sistem Informasi Manajemen Legalisir Online Berbasis Website.	Website	Waterfall	(Tidak disebutkan database nya)	Sistem Informasi Manajemen Legalisir Ijazah.
	(Pramusinto dkk., 2021). Implementasi E-Services Untuk Legalisir	Aplikasi berbasis website	Waterfall	MariaDB	Aplikasi legalisir ijazah dan transkrip nilai berbasis website.
	(Pramadhana dkk., 2023). Perancangan Sistem Informasi Legalisir Ijazah Online di Politeknik Negeri Indramayu.	Website	Agile	MySQL	Platform website pengajuan permintaan legalisir online.
	(Permana dkk., 2020). Implementasi E-Legalisir Untuk Legalisir Ijazah dan Transkrip	Sistem berbasis website	Extreme Programming	MySQL	Aplikasi pendaftaran siswa baru berbasis Android

	Online pada Fakultas Ilmu Komputer Universitas Mercu Buana.				
5	(Mei Prabowo & Edi Kuswanto, 2020). Metode Extreme Programming Dalam Pengembangan Aplikasi Legalisir Online Berbasis Web Service.	Web service	Extreme Programming	Terintegrasi dengan database SIAKAD TIPD IAIN Salatiga (Tidak disebutkan databasenya)	Aplikasi legalisir online berbasis web servis.





2.2 Kajian Pustaka

2.2.1 Sistem

Sistem adalah suatu kesatuan yang terdiri dari elemen-elemen atau komponen-komponen yang saling berinteraksi dan bekerja sama untuk mencapai suatu tujuan tertentu. Komponen-komponen ini bisa berupa manusia, mesin, perangkat lunak, aturan, dan prosedur yang saling terkait. Sistem biasanya memiliki beberapa karakteristik utama, diantaranya:

- 1) Tujuan
Setiap sistem memiliki sasaran atau target yang ingin dicapai
- 2) Masukan (Input)
Sumber daya atau data yang masuk ke dalam suatu sistem untuk selanjutnya dilakukan pemrosesan.
- 3) Proses
Suatu operasi atau tindakan untuk mengubah suatu masukan menjadi sebuah keluaran.
- 4) Keluaran (Output)
Hasil pengolahan masukan yang telah diproses pada suatu sistem.
- 5) Batasan (Boundary)
Batasan yang membedakan sistem dengan lingkungan eksternal
- 6) Umpan Balik (Feedback)
Informasi yang dikembalikan ke sistem untuk dilakukan evaluasi dan perbaikan agar sistem menjadi lebih baik.

2.2.2 Legalisir

Legalisir merupakan suatu proses pemberian tanda legalisasi resmi atau pengesahan pada dokumen pendidikan, seperti ijazah, untuk memastikan keaslian. Legalisir kerap kali dibutuhkan saat seseorang ingin mengajukan atau



menggunakan dokumen tersebut di lembaga yang memerlukan verifikasi resmi. Keuntungan dalam melakukan legalisir adalah untuk memverifikasi keaslian dokumen oleh pihak berwenang.

2.2.3 Payment Gateway

Payment gateway ialah layanan yang menyediakan otorisasi untuk proses transaksi dalam bisnis yang dilakukan secara daring, payment gateway berfungsi sebagai penyedia layanan yang menghubungkan antara sistem dan lembaga keuangan untuk memproses transaksi secara daring. Metode pembayaran secara non-tunai lebih efektif dan efisien jika dibandingkan pembayaran tunai. Selain itu, metode ini dapat memberikan jaminan dalam kelancaran serta keamanan dalam sistem pembayarannya, yang didasarkan pada beberapa aspek, diantaranya, peningkatan efisiensi, keamanan yang lebih baik, perlindungan bagi konsumen, dan peningkatan ketersediaan akses dalam sistem pembayaran.

2.2.4 Jasa Ekspedisi

Jasa Ekspedisi merupakan layanan yang menyediakan pengiriman barang, dokumen, atau paket dari satu tempat ke tempat lain, baik dalam skala lokal, nasional, maupun internasional. Perusahaan ekspedisi biasanya memiliki berbagai metode pengiriman, seperti darat, laut, dan udara, tergantung pada jarak dan kebutuhan pelanggan. Fungsi utama jasa ekspedisi meliputi, pengambilan barang, distribusi, pelacakan, dan asuransi.

2.2.5 Biteship

Biteship adalah agregator logistik yang membantu pemilik bisnis dan perorangan untuk terhubung dengan lebih dari 30 layanan ekspedisi terbaik di Indonesia. Dengan menyediakan berbagai opsi pengiriman, Biteship mempermudah bisnis dalam mengelola proses pengiriman mereka. Melalui integrasi API yang sederhana, Biteship memungkinkan bisnis untuk menghitung tarif pengiriman,



malxak paket, dan mengelola pengiriman dengan lebih efisien. Pengguna juga dapat melakukan order kurir dari Biteship melalui Mobile App, Web Dashboard, atau integrasi API. Selain itu, Biteship menawarkan layanan pergudangan dan pemenuhan (fulfillment) yang terhubung langsung dengan berbagai platform e-commerce, serta plugin yang kompatibel dengan platform seperti WooCommerce dan Shopify. Dengan menggunakan Biteship, bisnis dapat menghemat waktu dan sumber daya yang biasanya diperlukan untuk mengintegrasikan berbagai layanan pengiriman secara individual, sehingga dapat fokus pada pengembangan bisnis mereka.

2.2.6 Unified Modeling Language

Unified Modeling Language atau UML adalah bahasa yang divisualisasikan dalam bentuk gambar atau grafik yang berfungsi untuk memberikan gambaran dan spesifikasi dalam pembangunan dan dokumentasi dari sebuah pengembangan sistem berorientasi objek. UML memberikan sebuah standar pembuatan blue print sistem, yang dapat terdiri dari konsep proses bisnis, pembuatan class yang dapat dituangkan pada bahasa pemrograman tertentu, rancangan basis data, serta komponen-komponen yang dibutuhkan dalam pengembangan sistem (Narulita dkk., 2024). Berikut ini macam-macam diagram pada UML beserta penjelasannya :

a. Use Case Diagram

Use Case Diagram digunakan untuk menganalisis kebutuhan sistem pada tingkat tinggi. yang diilustrasikan melalui berbagai skenario penggunaan. Berikut komponen yang ada pada Use Case Diagram, terdapat pada tabel 2.2 berikut.

Tabel 2.2 Tabel Use Case Diagram

Simbol	Nama	Keterangan
--------	------	------------

	Actor	Actor merupakan representasi dari pengguna dalam suatu sistem informasi. Namun juga dapat berupa sistem lain yang berinteraksi dengan sistem tersebut.
	Association	Association menggambarkan relasi yang terjalin antara actor dan use case.
	Use Case	Use case digambarkan dalam bentuk oval dan berfungsi sebagai abstraksi yang menghubungkan actor dengan tindakan atau fungsi sistem.
	Extend	Untuk menunjukkan bahwa suatu use case merepresentasikan tampilan fungsional dari sebuah skenario atau fungsi sistem.



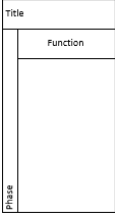


b. Activity Diagram

Activity Diagram adalah diagram yang menggambarkan alur kerja atau aktivitas dalam program yang sedang dirancang. Alur ini dapat mencakup menu-menu maupun proses bisnis yang ada didalam sistem tersebut. Berikut komponen yang ada pada Activity Diagram, terdapat pada tabel 2.3 berikut.

Tabel 2.3 Tabel Activity Diagram

Simbol	Nama	Keterangan
	Initial Node	Sebuah activity diagram memiliki sebuah initial node (status awal).
	Activity	Activity merepresentasikan aktivitas yang dilakukan sistem, activity biasanya diawali dengan kata kerja.
	Decision Node	Decision node menggambarkan percabangan, dimana ada pilihan aktivitas yang lebih dari satu.
	Final Node	Initial node (status akhir) yang dilakukan sistem.

	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
---	----------	---

c. Sequence Diagram

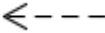
Sequence Diagram merupakan salah satu diagram UML yang menggambarkan interaksi antar objek sesuai urutan waktu terjadinya. Berikut komponen yang ada pada Sequence Diagram, terdapat pada tabel 2.4 berikut.

Tabel 2.4 Tabel Sequence Diagram

Simbol	Nama	Keterangan
	Actor	Actor pada Sequence Diagram adalah entitas yang berinteraksi dengan sistem.
	Object	Object adalah komponen yang berinteraksi dalam sistem.
	Lifeline	Menghubungkan objek selama sequence (message dikirim atau diterima dan aktivitasnya).





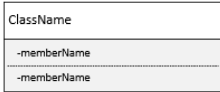
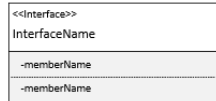
	Activation	Activation menggambarkan suatu objek mulai berpartisipasi didalam sebuah sequence. serta menandakan objek sedang menjalankan suatu aktivitas.
	Message Entry	Berfungsi untuk menggambarkan pesan/hubungan antar objek serta menunjukkan urutan kejadian yang terjadi.
	Message Self	Pesan yang dikirim oleh suatu objek kepada dirinya sendiri.
	Message Return	Menggambarkan pesan balasan dari suatu objek.

d. Class Diagram

Class Diagram merupakan salah satu diagram UML yang digunakan untuk menampilkan kelas-kelas yang berada dalam suatu sistem yang akan digunakan. Berikut komponen yang ada pada Class Diagram, terdapat pada tabel 2.5 berikut

Tabel 2.5 Tabel Class Diagram

Simbol	Nama	Keterangan
--------	------	------------

	Class	Menyatakan suatu relasi hubungan di antara model class dalam suatu sistem.
	Interface	Konsep interface dalam pemrograman berorientasi objek.
	Association	Menunjukkan hubungan atau keterkaitan umum antara dua kelas.
	Aggregation	Menunjukkan bahwa satu kelas terdiri dari kelas lain.

2.2.7 Laravel

Laravel adalah suatu *open-source framework* yang menyediakan sejumlah alat dan arsitektur aplikasi, termasuk bundle, migrasi, dan antarmuka baris perintah (CLI) yang lebih dikenal dengan Artisan. *Framework* ini menggabungkan fitur terbaik dari berbagai *framework* lain seperti Codeigniter, Yii, ASP.NET MVC, Ruby on Rails, Sinatra dan lainnya. Kelebihan Laravel terletak pada stabilitas reload data, performa yang lebih cepat, keamanan data, pemanfaatan fitur canggih seperti Blade dengan konsep *HMVC (Hierarchical Model View Controller)*,



fitur pengelolaan migrasi untuk pembuatan skema tabel dalam database, dan ketersediaan *library* yang siap digunakan. (Murod dkk., 2024).

2.2.8 Application Programming Interface

Application programming interface (API) adalah sekumpulan instruksi yang digunakan untuk membangun perangkat lunak yang mencakup fungsi, kelas, struktur, interface, serta elemen lainnya (Izza & Rajagede, 2022). API memungkinkan pengembang untuk membuka perangkat lunak sehingga bisa dikembangkan lebih lanjut atau diintegrasikan dengan perangkat lunak lainnya. Selain itu, API memungkinkan aplikasi untuk terhubung dengan aplikasi lain, memberikan kemampuan bagi pengembang untuk menggunakan sistem fungsional yang ada.

2.2.9 Metode Waterfall

Model Waterfall atau sering kali disebut sebagai *classic life cycle* adalah model pengembangan perangkat lunak yang menekankan fase-fase yang berurutan dan sistematis, dimulai dari spesifikasi kebutuhan konsumen dan berkembang melalui proses perencanaan (planning), pemodelan (modelling), pembangunan (construction), dan penyebaran (deployment), yang berujung pada dukungan terus menerus untuk sebuah perangkat lunak yang utuh. Model ini dapat digunakan pada saat kebutuhan untuk sebuah masalah telah dipahami dengan baik, dan pekerjaan dapat mengalir secara linear dari proses komunikasi hingga penyebaran (deployment). Situasi ini ditemui saat diadaptasi atau perpanjangan dari sisten yang ada sudah terdefinisi dengan baik. Adapun model ini juga dapat digunakan pada situasi di mana dibutuhkan usaha yang terbatas untuk pengembangan perangkat lunak, tetapi kebutuhan perangkat lunak sudah terdefinisi dengan baik dan cenderung stabil. Namun, dalam pengembangan perangkat





lunak, model ini cenderung menjadi salah satu pendekatan yang kurang iteratif dan fleksibel, karena proses mengalir satu arah (“ke bawah” seperti air terjun). (Pressman & Maxim, 2015).

2.2.10 Black Box Testing

Blackbox testing adalah metode yang memverifikasi hasil eksekusi aplikasi berdasarkan data inputan untuk memastikan fungsionalitas aplikasi berjalan sesuai ekspektasi (Mintarsih, 2023). Pengujian ini difokuskan pada kesesuaian alur kerja dengan kebutuhan user, fungsionalitas yang tersedia, dan antarmuka aplikasi. Metode ini tidak mencakup pemeriksaan kode sumber program. Tahapan dalam pengujian *blackbox* mencakup:

- a. Menyusun test case untuk menguji berbagai fungsi dalam aplikasi.
- b. Merancang test case guna menilai sejauh mana alur kerja suatu fungsi sesuai dengan kebutuhan dan permintaan pengguna.
- c. Mengidentifikasi bug atau kesalahan berdasarkan tampilan antarmuka aplikasi.

Proses pengujian harus menerapkan teknik yang tepat agar dapat menemukan kesalahan yang sebelumnya belum terdeteksi, sehingga meningkatkan kualitas perangkat lunak.