

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Sebelumnya/ Kajian Pustaka

Studi literatur dilakukan pada penelitian ini dengan permasalahan yang sedang dikaji baik melalui jurnal, buku, atau dokumen lain yang relevan. Pada bab ini, berisi rangkuman penelitian-penelitian sebelumnya yang berkaitan dengan topik penelitian yang akan dilaksanakan.

1. *Aplikasi Pendaftaran Seminar Menggunakan Metode MVC Berbasis Website Menggunakan Framework CodeIgniter 3.1.10 (Dermawan: 2020)*

Tujuan dari penelitian ini adalah membangun sebuah aplikasi pendaftaran seminar yang dapat membantu pengguna mendaftar dan memperoleh informasi berkaitan dengan seminar. Aplikasi yang digunakan adalah Visual Studio Code dan XAMPP. Metode di dalam pembangunan aplikasi web ini adalah metode MVC dengan *framework CodeIgniter 3.1.10*. Pengguna dengan level *user* dapat melakukan pendaftaran dan pencetakan sertifikat. Sedangkan pengguna dengan level admin dapat memperoleh laporan kehadiran peserta seminar[14].

2. *Learning Management System (LMS) pada E-Learning Menggunakan Metode Agile dan Waterfall berbasis Website (Yauma, Fitri, Ningsih: 2021)*

Penelitian ini menghasilkan *output* berupa *Learning Management System (LMS)* berbasis *website* pada sekolah MA Alwutsqo Depok. Layanan yang tersedia pada sistem ini adalah fitur *upload* dan *sharing* materi pembelajaran, sarana diskusi secara daring, fitur berbagi pesan, fitur *test* atau ujian untuk siswa, survei, laporan pembelajaran, dan lain-lainnya.

Terdapat dua metode yang digunakan pada penelitian ini. Yang pertama adalah metode *waterfall* karena dianggap cocok pada proses perancangan hingga pemeliharaan. Kemudian adalah metode *agile* yang diterapkan pada perubahan, perawatan, dan pengembangan sistem.

Pengambilan data yang diperlukan dilaksanakan secara langsung ke sekolah yang dimaksud dengan cara wawancara dan observasi terkait dengan kebutuhan fitur maupun sistem yang ingin dirancang ke dalam *e-learning*. Data dan informasi lainnya diambil dari jurnal maupun artikel yang terindeks SINTA sebagai bahan acuan untuk penelitian[15].

3. *Sistem Informasi Desa Berbasis Website (Studi Kasus Kelurahan Lalung) (Sholihin: 2021)*

Penelitian ini berfokus pada pembuatan sebuah sistem informasi untuk Kelurahan Lalung di Kabupaten Karanganyar. Sistem informasi ini terdiri dari pembukuan keuangan berbasis *website* yang dioperatori oleh perangkat desa. Masyarakat juga dapat mengaksesnya terkait dengan fungsi pengawasan demi terciptanya transparansi pada kelurahan tersebut. *Website* ini juga berfungsi untuk sarana promosi usaha yang dilakukan warga sekitar sebagai bukti dari pemberdayaan masyarakat.

Metode yang dipilih penulis pada penelitian ini adalah *SDLC* atau *Software Development Life Cycle*. Sedangkan modelnya adalah *waterfall*. Setelah proses pembangunan selesai, dilaksanakan pengujian sistem dengan metode *black-box-testing* dan *System Usability Scale (SUS)* [16].

4. *Pengembangan Sistem Informasi Pariwisata Berbasis Website Untuk Meningkatkan Kesadaran Masyarakat Terhadap Potensi Daerah Kabupaten Dompu (Rizaly, Rahman: 2021)*

Penelitian ini bertujuan untuk menaikkan tingkat kesadaran masyarakat akan berlimpahnya potensi pariwisata khususnya pada Kabupaten Dompu. *Output* pada penelitian ini adalah *website* pariwisata berupa sistem informasi mengenai pariwisata yang ada pada Kabupaten Dompu di Provinsi Nusa Tenggara Barat.

Pembangunan sistem informasi pariwisata Kabupaten Dompu ini menerapkan metode *waterfall*. Metode ini terdiri dari tahap *requirement*, *design*, *coding*, *testing*, *implementation*, dan *maintenance* [17].

5. *Perancangan Sistem Informasi Permohonan Perizinan Penelitian dengan Metode Agile dan Framework Laravel Berbasis Website (Hendrawan, Manuputty, Haryanto: 2020)*

Penelitian ini bertujuan untuk mendukung pekerjaan Pemerintah Kota Salatiga pada bagian manajemen data dan survei permohonan kerja praktik. Hasil penelitiannya berupa sebuah sistem informasi Berbasis *website* yang dapat membantu pekerjaan manajemen data dan laporan secara digital.

Metode perancangan aplikasi berbasis *website* ini menggunakan metode *agile* dengan *Laravel* sebagai *framework*-nya. Sedangkan pengujian hasil dari penelitian ini yang berupa sistem informasi berbasis *website* menggunakan metode *White Box Testing* [11].

6. *Sistem Informasi Pemasaran dengan Penerapan CRM (Customer Relationship Management) Berbasis Website menggunakan Metode Waterfall dan Agile (Saputra, Fitri, Nathasia: 2021)*

Tujuan dari penelitian ini adalah membangun sebuah sistem informasi yang dapat memudahkan pemilik Toko Bella Frame ART di dalam pemasaran, penjualan, dan pelayanan, sehingga berdampak terhadap pelanggan di dalam memperoleh informasi terkait dengan produk di Toko Bella Frame ART yang sedang dicari.

Pada penelitian ini, peneliti menggunakan dua metode di dalam pengembangan sistemnya. Metode pertama adalah metode *waterfall*. Metode ini digunakan dalam pengembangan aplikasi pada tahap pertama. Metode kedua adalah metode *agile* yang digunakan setelah metode *waterfall* selesai dilaksanakan[19].

Tabel 2. 1 Penelitian Sebelumnya

No.	Judul	Metode	Hasil	Perbedaan
1	Aplikasi Pendaftaran Seminar Menggunakan Metode MVC Berbasis Website Menggunakan Framework CodeIgniter 3.1.10	<i>MVC, Waterfall, Blackbox testing, Whitebox testing</i>	Penelitian ini menghasilkan <i>output</i> berupa sebuah aplikasi pendaftaran seminar. Pada pengujian <i>Blackbox</i> didapatkan bahwa semua skenario yang disusun pada testing ini berstatus valid. Sedangkan pada pengujian <i>Whitebox</i> didapatkan 15 <i>node</i> dan 31 <i>edge</i> dengan kompleksitas <i>cyclomatic</i> bernilai 11.	Metode <i>Waterfall, Whitebox testing</i>
2	<i>Learning Management System (LMS)</i> pada <i>E-Learning</i> Menggunakan Metode <i>Agile</i> dan <i>Waterfall</i> berbasis <i>Website</i>	<i>Agile, Waterfall, Blackbox</i>	Penelitian ini menghasilkan <i>output</i> berupa <i>Learning Management System (LMS)</i> berbasis <i>website</i> pada sekolah MA Alwutsqo Depok. Pada pengujian <i>Blackbox</i> didapatkan bahwa semua skenario pengujian berstatus valid.	Metode <i>Waterfall</i>
3	Sistem Informasi Desa Berbasis <i>Website</i> (Studi Kasus Kelurahan Lalung)	<i>Waterfall, Blackbox testing, System Usability Scale</i>	<i>Output</i> yang dihasilkan pada penelitian ini berupa sistem informasi desa berbasis <i>website</i> yang di dalamnya terdapat fitur pembukuan keuangan dan promosi usaha warga. Pengujian <i>Blackbox</i> didapatkan hasil yang valid pada setiap skenario yang dilakukan. Sedangkan pengujian <i>SUS</i> didapatkan nilai rata-rata	Metode <i>Waterfall, SUS</i>

No.	Judul	Metode	Hasil	Perbedaan
			78,2 sehingga masuk pada kategori <i>acceptable</i> .	
4	Pengembangan Sistem Informasi Pariwisata Berbasis <i>Website</i> Untuk Meningkatkan Kesadaran Masyarakat Terhadap Potensi Daerah Kabupaten Dompu	<i>Waterfall</i>	Penelitian ini menghasilkan keluaran berupa sistem informasi pariwisata berbasis <i>website</i> . Uji Coba Ahli Materi dan Desain menghasilkan nilai di atas 80% dengan kategori sangat layak.	Metode <i>Waterfall</i>
5	Perancangan Sistem Informasi Permohonan Perizinan Penelitian dengan Metode <i>Agile</i> dan <i>Laravel</i> Berbasis <i>Website</i>	<i>Agile, Whitebox testing</i>	Penelitian ini menghasilkan sistem informasi manajemen data dan survei berbasis <i>website</i> untuk Pemkab Salatiga. Semua proses pada pengujian dengan <i>Whitebox</i> dinyatakan berhasil.	Metode <i>Whitebox testing</i>
6	Sistem Informasi Pemasaran dengan Penerapan <i>CRM (Customer Relationship Management)</i> Berbasis <i>Website</i> menggunakan	<i>Agile, Waterfall, Blackbox</i>	<i>Output</i> pada penelitian ini berupa sistem informasi pemasaran, penjualan, dan pelayanan. Hasil dari pengujian <i>Blackbox</i> adalah setiap skenario yang diujikan berstatus valid.	Metode <i>Waterfall</i>

No.	Judul	Metode	Hasil	Perbedaan
	Metode <i>Waterfall</i> dan <i>Agile</i>			

Tabel 2.1 adalah data Penelitian Sebelumnya yang penulis jadikan referensi dan telah dirangkum. Penulis menjadikan penelitian yang berjudul “Aplikasi Pendaftaran Seminar Menggunakan Metode MVC Berbasis *Website* Menggunakan *Framework CodeIgniter 3.1.10*” sebagai rujukan utama dengan metode pembangunan sistem dan *testing website* sebagai pembeda. Informasi di dalamnya disusun dalam bentuk tabel agar penulis lebih mudah dalam membaca informasi di dalamnya.

2.2. Dasar Teori

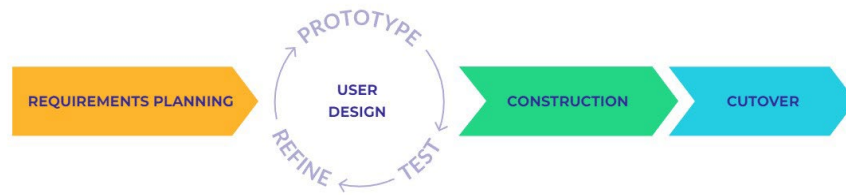
Pada subbab ini, dijelaskan istilah atau terminologi, konsep, maupun teori yang ada pada penelitian ini.

2.2.1. Sistem Informasi

Sistem merupakan kesatuan komponen maupun variabel dalam sebuah jaringan kerja yang berkaitan satu sama lain dalam menjalankan aktivitas atau aksi tertentu agar tercapai suatu tujuan yang dirancang sebelumnya. Informasi adalah sekelompok data atau fakta yang diproses dan dikelola sedemikian rupa agar dapat dimanfaatkan oleh penerimanya. Sedangkan sistem informasi merupakan rangkaian sub sistem yang menyeluruh dan terkoordinasi yang dapat mengolah data menjadi suatu informasi melalui serangkaian proses terpadu. [20][21][22]

2.2.2. *Rapid Application Development*

Rapid Application Development (RAD) merupakan salah satu model dari *System Development Life Cycle* (SDLC). Model pengembangan perangkat lunak ini memberikan penekanan pada siklus yang singkat di dalam pengembangannya dengan syarat kebutuhannya diketahui dengan baik.



Gambar 2.1 Model *Rapid Application Development* (RAD)[23]

Aktivitas di dalam metode RAD seperti yang tergambar pada Gambar 2.1 meliputi:

a. *Requirements Planning* (Rencana Kebutuhan)

Pada tahap ini, pengguna dan tim proyek mengidentifikasi objektif dari sistem yang akan digunakan. Fokusnya untuk mencapai tujuan bisnis dengan persyaratan atau kebutuhan yang agak longgar.

b. *User Design* (Desain Pengguna)

Tahapan ini merupakan bagian yang krusial pada metode RAD. *Developer* memulai pekerjaan untuk membuat prototipe dengan tujuan untuk menunjukkannya kepada klien secepat dan semurah mungkin. Setelah prototipe selesai, hasilnya didemonstrasikan kepada pengguna. Kemudian tim mengumpulkan semua *feedback* atas produk tersebut. Melalui umpan balik yang diperoleh sebelumnya, *developer* kembali ke tahap *prototyping* untuk membuat prototipe yang sesuai dengan pengguna.

c. *Construction* (Pembangunan Sistem)

Tahap ini adalah memulai pembangunan sistem yang telah dirancang pada tahap sebelumnya. Tahapan ini berfokus pada kualitas, skalabilitas, pemeliharaan, dll. Namun, pengguna terus berpartisipasi dalam memberikan *feedback* saat fitur diimplementasikan.

d. *Cutover* (Tahap Akhir)

Ini adalah fase terakhir yang meliputi pengujian dengan menggunakan *Black-Box Testing*. Tujuannya adalah mengurangi risiko cacat sistem. [11], [23], [24]

2.2.3. *CodeIgniter*

CodeIgniter merupakan sebuah *Application Development Framework* yang diperuntukkan untuk pembuatan *website* menggunakan *PHP*. Tujuannya adalah agar proses *development* dapat dilaksanakan dengan lebih cepat bila dibandingkan dengan menulis kode dari awal. *Framework* ini memiliki kumpulan *library* untuk *task* yang umum digunakan serta *interface* sederhana dan struktur logis untuk mengakses *library* tersebut. [13]

Framework ini dalam penggunaannya menggunakan konsep MVC (*Model – View – Controller*). *Model* merupakan kumpulan fungsi dan pengelola data di dalam domain aplikasi yang bertugas untuk merespons *request* atas suatu informasi maupun instruksi untuk mengubah suatu kondisi. *View* bertugas menerjemahkan informasi dari *Model* ke dalam bentuk yang sesuai agar dapat berinteraksi dengan penggunanya. *Controller* berfungsi menerima *input* dari pengguna kemudian memicu respons dengan memanggil objek yang telah didefinisikan sebelumnya. [25]

2.2.4. *MySQL*

MySQL merupakan *RDBMS (Relational Database Management System)* *opensource* yang paling banyak digunakan di dunia dengan lebih dari 100 juta pengguna. *MySQL* telah menjadi *RDBMS* yang direkomendasikan bagi banyak *software* dan *application developers* baik itu untuk *online* maupun *desktop platform* mengingat kecepatan dan kemudahan dalam penggunaannya. [26]

2.2.5. *Black-box Testing*

Black-box Testing atau *Behavioral Testing* adalah pengujian hasil *input* dan *output* dari sebuah *software* tanpa mengecek kode strukturalnya yang dilakukan pada masa akhir pembuatan *software* tersebut dengan tujuan mengetahui apakah perangkat lunak tersebut dapat berjalan dengan baik atau tidak. Dengan tanpa mengecek kode struktural *software* tersebut, maka penguji tidak perlu memiliki *skill coding*. [27], [28]