

BAB II TINJAUAN PUSTAKA

2.1. Studi Sebelumnya

Penelitian yang dilakukan oleh Mara Destiningrum berjudul *Sistem Informasi Penjadwalan Dokter Berbasis Web dengan Menggunakan Framework Codeigniter (Studi Kasus: Rumah Sakit Yukum Medical Centre)* mengangkat sebuah masalah yaitu sistem penjadwalan pada Rumah Sakit tersebut yang masih menggunakan kertas sehingga menyebabkan pencarian informasi menjadi lebih lambat dimana pasien terkadang membutuhkan layanan Kesehatan dengan cepat. Pengembangan sistem dilakukan dengan metodologi *System Development Life Cycle* dengan model *waterfall*. Analisis perancangan dilakukan dengan *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram*. Implementasi pengembangan dilakukan dengan menggunakan *framework Codeigniter* sebagai desain tampilan antarmuka, dan MySQL sebagai pengelola *database*. Dari penelitian tersebut, didapatkan bahwa penggunaan sistem penjadwalan berbasis *website* dapat memberikan informasi jadwal praktik dokter dengan cepat dan akurat dan menjadikan proses penjadwalan secara keseluruhan lebih efisien [5].

Penelitian yang dilakukan oleh Muhammad Susilo (2018) berjudul *Rancang Bangun Website Toko Online Menggunakan Metode Waterfall* membahas mengenai pengembangan *website* kepada toko-toko di daerah Bengkalis yang menjual aksesoris komputer dan laptop. Penjualan yang pada awalnya masih dilakukan secara tradisional dinilai tidak dapat mencapai jangkauan pasar yang lebih luas. Pengembangan *web* dilakukan dengan menggunakan metodologi *System Development Life Cycle* model

waterfall, dan untuk pemrograman menggunakan bahasa PHP, HTML, dan CSS. Analisis perancangan sistem dilakukan dengan menggunakan *Use Case Diagram*, serta *Entity Relationship Diagram* [6].

Penelitian yang dilakukan oleh Iman Hariman dan Christina Meilisa (2020) berjudul *Sistem Informasi Manajemen Training Menggunakan Metode Agile Software Development the Papandayan Hotel Bandung* melakukan perancangan sistem untuk melakukan pengembangan sumber daya manusia untuk meningkatkan kualitas dan kompetensi kinerjanya. Penelitian tersebut dilakukan karena terdapat permasalahan pada perusahaan yaitu adanya kesulitan dalam melakukan distribusi informasi *training*, serta penentuan calon peserta *training* tersebut. Analisis perancangan sistem yang digunakan dalam penelitian tersebut mencakup *use case diagram*, *activity diagram*, *class diagram*, dan *Sequence Diagram* [7].

Penelitian berjudul *Agile Development Methods dalam Pengembangan Sistem Informasi Pengajuan Kredit Berbasis Web (Studi Kasus: Bank BRI Unit Kolonel Sugiono)* oleh Irfan Mahendra dan Deny Tresno Eby Yanto (2018) mengangkat masalah pengajuan kredit pada bank yang masih dilakukan secara manual yang membutuhkan waktu yang cukup lama, sehingga dinilai dapat berpengaruh negatif terhadap perkembangan bisnis di tengah persaingan yang semakin meningkat dan ketat. Rangkaian kegiatan yang dilakukan dalam pembangunan sistem tersebut meliputi analisa kebutuhan sistem, desain, *code generation*, *testing*, dan *support* [8].

Penelitian yang dilakukan oleh Imtihan, dkk berjudul *Sistem Informasi Pengajian Guru Honorer Menggunakan Konsep Agile Software Development dengan Metodologi Extreme Programming (XP)*

pada SMK Bangun Bangsa membahas mengenai perancangan sistem penggajian yang sebelumnya masih belum terkomputerisasi. Sistem penggajian sebelumnya dinilai masih terlalu lambat, serta sering menimbulkan kesalahan, karena penghitungan gaji guru dilakukan dengan melakukan rekap absensi guru yang dilakukan setiap tiga bulan sekali. Perancangan sistem yang dilakukan mencakup perancangan *data flow diagram* (DFD), perancangan tabel, *entity relationship diagram* (ERD), perancangan interface, serta perancangan *code* pemrograman. Sistem dibangun dengan menggunakan Bahasa pemrograman PHP (*Hypertext Preprocessor*), serta menggunakan *database* MySQL. Sistem penggajian yang dirancang diharapkan dapat mempermudah pekerjaan bendahara dalam melakukan proses penghitungan kehadiran dan gaji guru, serta proses pelaporan penggajian sekolah [9].

Tabel 1. Studi Sebelumnya

No.	Peneliti	Judul	Tujuan	Metode	Hasil
1	Mara Destiningrum, Qadhli Jafar Adrian (2017)	Sistem Informasi Penjadwalan Dokter Berbasis Web Dengan Menggunakan Framework Codeigniter (Studi Kasus: Rumah Sakit Yukum Medical Centre)	Meningkatkan efisiensi pencarian informasi dimana sebelumnya sistem penjadwalan masih dicatat pada kertas.	Waterfall	Hasil dari perhitungan kualitas sistem dihasilkan dengan presentase tanggapan sebesar 87,87 % dengan kriteria perhitungan sistem sangat baik.

2	Muhammad Susilo (2018)	Rancang Bangun <i>Website</i> Toko Online Menggunakan Metode Waterfall	Meningkatkan penjualan menggunakan <i>website</i> , dimana sebelumnya penjualan masih dilakukan sepenuhnya secara tradisional.	Waterfall	<i>Website</i> sudah berhasil dijalankan dan diuji coba pada browser Google Chrome versi 60.0.3112.90, dan Mozilla Firefox versi 3.5.2.
3	Irman Hariman, Christina Meilisa (2020)	Sistem Informasi Manajemen Training Menggunakan Metode Agile Software Development the Papandayan Hotel Bandung	Meningkatkan efisiensi dalam distribusi informasi training, dan penentuan calon peserta training.	Agile	<i>Website</i> sudah berhasil diterapkan dan diuji coba. <i>Training Officer</i> dapat menyampaikan spesifikasi <i>training</i> lebih mudah dan dilakukan secara rutin setiap bulannya.
4	Irfan Mahendra, Deny Tresno Eby Yanto (2018)	Agile Development Methods dalam Pengembangan Sistem Informasi	Mempermudah proses pengajuan <i>kredit</i> yang sebelumnya masih dilakukan secara manual sehingga	Agile	Nasabah sudah dapat mengajukan permohonan <i>kredit</i> melalui sistem ini, dan

		Pengajuan <i>Kredit</i> Berbasis <i>Web</i> (Studi Kasus: Bank BRI Unit Kolonel Sugiono)	menghambat perkembangan perusahaan.		pihak bank dapat langsung menganalisa data yang tersedia pada sistem untuk dilanjutkan ke proses pengambilan keputusan persetujuan <i>kredit</i> .
5	Khairul Imtihan, Rabiatul Hadawiyah, Hasyim Asyari (2017)	Sistem Informasi Penggajian Guru Honorer Menggunakan Konsep Agile Software Development dengan Metodologi Extreme Programming (XP) pada SMK Bangun Bangsa	Mengkomputerisasi sistem penggajian guru honorer, yang sebelumnya masih dilakukan dengan cara rekap absensi guru setiap tiga bulan yang rentan akan kesalahan dan memakan waktu yang lama.	Agile	Sistem yang dibangun dapat mempermudah dan mengefisiensi bendahara sekolah dalam merekap absensi dan melakukan penghitungan gaji.

2.2. Dasar Teori

1. Sistem

Menurut Sihotang (2018), sistem merupakan suatu jaringan kerja di mana semua prosedur-prosedur yang berada di dalamnya saling berhubungan, serta berkumpul bersama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu [10].

2. Website

Menurut Yuhefizar (2013:2), pengertian *website* adalah “keseluruhan halaman-halaman *web* yang terdapat dari sebuah *domain* yang mengandung informasi” [11]. Hidayat (2010) mengartikan *domain* sebagai sebuah pengenal berbentuk alamat yang bersifat unik yang berfungsi untuk mengenali sebuah *website* pada internet, atau dengan kata lain *domain* adalah alamat yang digunakan untuk menemukan dan mencari sebuah *website* pada dunia internet [12].

3. Unified Modelling Language (UML)

Sulistyorini (2009) dalam penelitiannya berjudul Pemodelan Visual dengan Menggunakan UML dan Rational Rose mendefinisikan *Unified Modelling Language* (UML) merupakan suatu bahasa yang menjadi standar dalam industri yang digunakan untuk mempermudah kegiatan visualisasi, desain, dan dokumentasi perangkat lunak. UML memberikan standar untuk melakukan pemodelan aplikasi perangkat lunak untuk memberikan spesifikasi alur dan proses-proses yang dilakukan sistem. Standar UML dibangun secara universal, namun karena konsep dasar dari UML menggunakan kelas-kelas dan operasi, UML umumnya digunakan untuk pemrograman berorientasi objek [13].

4. Pemrograman Berorientasi Objek

Grady Booch dalam bukunya *Object-Oriented Analysis and Design with Applications* (2007:41) mengartikan *Object-Oriented Programming*

atau pemrograman berorientasi objek sebagai metode dalam implementasi di mana program-program terorganisasi sebagai kumpulan obyek-obyek yang bersifat kooperatif, di mana masing-masing obyek melambangkan instansi dari beberapa kelas, dan dimana masing-masing kelas merupakan bagian dari suatu hirarki kelas-kelas yang terhubung melalui hubungan pewarisan [14].

Menurut Ashu Kumar (2014:6), pemrograman berorientasi objek mempunyai beberapa prinsip, yaitu [15]:

1. Abstraksi data (*Data Abstraction*): Tindakan untuk merepresentasikan fitur-fitur yang bersifat esensial tanpa menyertakan semua detail-detail di belakangnya.
2. Modularitas (*Modularity*): Proses membagi program menjadi berbagai macam modul atau bagian.
3. Polimorfisme (*Polymorphism*): Proses untuk menggunakan segala sesuatu (*function, operator, constructor*) untuk lebih dari satu tujuan dengan menggunakan identitas/nama yang sama.
4. Pewarisan (*Inheritance*): Proses untuk mewariskan atau menghubungkan data dari suatu kelas kepada kelas lain, dibandingkan dengan membuat ulang data.
5. Enkapsulasi data (*Data Encapsulation*): Menggabungkan data-data dan fungsi-fungsi menjadi sebuah unit/struktur tunggal.

5. Basis Data

Pamungkas (2017) mendefinisikan basis data sebagai sebuah kumpulan data yang saling berhubungan, yang disimpan secara bersama pada sebuah sarana, dimana data tersebut diorganisasikan berdasarkan *schema* atau struktur tertentu, dan menggunakan software untuk melakukan manipulasi untuk kegiatan-kegiatan tertentu [16].

6. Framework

Menurut Betha Sidik (2012), *framework* adalah kumpulan dari petunjuk-petunjuk yang dikumpulkan dalam kelas-kelas dan fungsi-fungsi dengan kegunaan masing-masing dan berfungsi untuk mempermudah *developer* perangkat lunak dalam memanggilnya karena pengembang tidak perlu menuliskan ulang *syntax* program yang sama sehingga dapat mempercepat pekerjaan [17].

7. System Development Life Cycle (SDLC)

Martias (2018) mendefinisikan SDC sebagai kegiatan pengembangan sistem yang mempunyai tahapan-tahapan berurutan untuk dapat menerapkan kebutuhan bisnis ke dalam sebuah aplikasi/sistem [18].

Proses-proses yang dicakup oleh metode *System Development Life Cycle* yaitu [19]:

1. Perencanaan (*Planning*)

Pada fase ini, kegiatan yang dilakukan yaitu menilai kelayakan sistem yang akan dibangun, baik dari sisi biaya, teknis, maupun operasional. Beberapa aktivitas yang dilakukan antara lain:

- a) Mengidentifikasi masalah yang akan dipecahkan oleh sistem, serta obyek-obyek yang termasuk di dalamnya.
- b) Mengidentifikasi nilai bisnis, dalam bentuk pembuatan perkiraan pembangunan proyek dari segi biaya dan waktu yang dibutuhkan, serta pendapatan yang akan dihasilkan oleh sistem.
- c) Membuat rencana kerja, dengan mengalokasikan waktu kerja serta sumber daya yang digunakan.
- d) Mengorganisasi proyek dari tim.
- e) Melakukan kontrol dan memberikan arahan terkait proyek.

2. Analisis (*Analysis*)

Pada fase ini, dilakukan pemeriksaan terhadap sistem dengan tujuan untuk mendapatkan dokumentasi lengkap untuk mengetahui kebutuhan dari *user*, serta lingkup yang akan dijangkau oleh sistem, begitu juga dengan cara untuk membangun sistem. Beberapa aktivitas yang dilakukan antara lain:

- a) Menganalisis kebutuhan dari segi informasi, perangkat keras, dan perangkat lunak yang dibutuhkan untuk membuat sistem dan memasukkannya dalam sebuah dokumentasi yang disebut juga dengan Spesifikasi Kebutuhan Perangkat Lunak (SKPL).
- b) Melakukan analisis lingkup sistem.

3. Desain (*Design*)

Pada fase ini, alur kerja sistem mulai dirancang, bersamaan juga dengan model data, serta rancangan pemrograman yang didapatkan dari analisis kebutuhan. Beberapa aktivitas yang dilakukan, antara lain:

- a) Perancangan basis data.
- b) Perancangan grafis.
- c) Perancangan program.
- d) Perancangan keamanan sistem.

4. Implementasi (*Implementation*)

Pada fase ini, sistem yang sudah selesai dikembangkan dan diuji coba akan diluncurkan dan digunakan oleh pengguna pada perangkatnya masing-masing, dan menggunakan sistem sesuai kebutuhan pengguna tersebut.

5. Pemeliharaan (*Maintenance*)

Fase ini merupakan fase yang berkelanjutan yang dilakukan secara rutin. Aktivitas yang dilakukan pada fase ini yaitu pengawasan dan peningkatan kinerja sistem agar dapat mempertahankan kinerja sistem.

8. SDLC Model Hybrid

Penelitian yang dilakukan oleh tim Lucid Software menyatakan bahwa SDLC model *hybrid* merupakan model SDLC yang dikembangkan oleh Erick Bergmann dan Andy Hamilton yang menggabungkan cara pengembangan dari model *waterfall* dan *agile*. Model ini pada awalnya tercipta karena terdapat perbedaan metode antara pihak *software* yang melakukan pengembangan dengan metode *agile*, dan pihak *hardware* yang melakukan pengembangan dengan metode *waterfall*. Metode *Agile-Waterfall Hybrid* dapat mempunyai bentuk yang berbeda-beda berdasarkan pada pembagian proses pengembangan mana saja yang menggunakan metode *waterfall* dan *agile* [20]