

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian oleh Ayu Nila Dewi [5] membahas implementasi Sistem Informasi Manajemen Event Pekan Olahraga Seni Desa (PORSENIDES) Desa Beraban menggunakan Framework Laravel. Diketahui bahwa proses registrasi dan pengelolaan data dalam PORSENIDES masih dilakukan secara manual, yang menghambat efisiensi. Oleh karena itu, peneliti mengembangkan sistem berbasis web untuk memfasilitasi registrasi dan pengelolaan data. Sistem ini dikembangkan menggunakan metode waterfall dengan PHP Laravel dan MySQL. Pengujian menggunakan Black Box Testing menunjukkan kepuasan pengguna sebesar 90%, menunjukkan keefektifan sistem tersebut.

Penelitian oleh Ziyad Rusdi [6] membahas terkait Desain Sistem Informasi dan Registrasi Berbasis Web untuk Kompetisi Akademik bertujuan untuk menciptakan sarana untuk menghubungkan penyelenggara kompetisi dengan peserta potensial yang ingin mengembangkan kemampuannya. Dalam pengembangan tampilan situs web ini menggunakan HTML, CSS, Javascript, dan menghubungkan basis data dengan MySQL. Dengan situs web ini, diharapkan akan lebih mudah bagi mahasiswa untuk berpartisipasi dalam atau mendapatkan informasi mengenai kompetisi yang sedang diselenggarakan, sehingga mereka dapat mengembangkan potensinya.

Penelitian oleh [7] bertujuan untuk mengembangkan sistem informasi pendaftaran kursus di Lembaga Kursus dan Pelatihan (LKP) Indo Jaya Kebumen. Menggunakan metode waterfall, website yang dibuat tidak hanya memberikan informasi lengkap dan cepat kepada pengguna atau calon pendaftar, tetapi juga mempermudah proses pendaftaran dan pengolahan data bagi administrator. Hasil uji coba menunjukkan bahwa website ini dapat berjalan dengan baik, menyediakan informasi pendaftaran kursus secara cepat dan akurat.

Penelitian oleh [8] membahas tantangan keamanan yang dihadapi oleh CV. Nyebar, sebuah start-up berbasis IT yang menyediakan layanan pengelolaan

data event melalui aplikasi web. Proses pendaftaran event di sistem mereka belum diamankan dengan baik, membuatnya rentan terhadap serangan siber seperti sniffing data dan Distributed Denial of Service (DDoS). Serangan DDoS dapat mengakibatkan overload server jaringan karena permintaan paket yang terus-menerus dari bot. Studi ini menyarankan penggunaan teknologi blockchain dengan metode hash dan proof-of-work sebagai alternatif untuk meningkatkan keamanan pendaftaran event, mengamankan informasi, menjaga konsistensi data, dan melindungi dari serangan DDoS

2.2 Landasan Teori

1) Sistem

Sistem didefinisikan sebagai kumpulan dari komponen yang saling berhubungan satu dengan yang lainnya membentuk satu kesatuan untuk mencapai tujuan tertentu (Jogiyanto dalam Oktaviani et al., 2019). Sistem (*System*) dapat didefinisikan dengan pendekatan prosedur dan dengan pendekatan komponen. Dengan pendekatan prosedur, sistem dapat didefinisikan sebagai kumpulan dari prosedur-prosedur yang mempunyai tujuan tertentu. Dengan pendekatan komponen, sistem dapat didefinisikan sebagai kumpulan komponen yang saling berhubungan satu dengan yang lainnya, membentuk satu kesatuan untuk mencapai tujuan tertentu [9].

2) Sistem Informasi

Sistem informasi merupakan gabungan dari 4 (empat) bagian utama, yaitu: Software, Hardware, Infrastruktur, dan Sumber Daya Manusia yang terlatih untuk menciptakan sebuah sistem yang dapat mengelola data menjadi informasi yang bermanfaat [9].

3) Website

Website adalah keseluruhan halaman halaman yang terdapat dari sebuah domain yang mengandung informasi atau yang berisikan dokumen-dokumen multimedia (teks, gambar, suara, animasi, dan video). Domain

adalah nama unik yang dimiliki oleh sebuah institusi sehingga bisa diakses melalui internet [5].

4) **Node.js**

platform perangkat lunak yang dirancang untuk menjalankan kode JavaScript di luar lingkungan browser web. Dengan Node.js, developer bisa mengembangkan aplikasi web full-stack menggunakan JavaScript di sisi server (backend) dan client (*frontend*) [10].

5) **CSS**

CSS adalah suatu cara untuk membuat format atau layout halaman web menjadi lebih menarik dan mudah dikelola. CSS muncul karena sulitnya mengatur layout tampilan dokumen yang dibuat dengan HTML murni meskipun telah menggunakan berbagai kombinasi format [11].

6) **MySQL**

MySQL adalah sebuah RDBMS (Relational Database Management System) yang sangat cepat dan kuat. Sebuah database mendukung kita untuk menyimpan, mencari, menyusun dan menerima data secara efisien. MySQL menggunakan SQL (*Structured Query Language*), bahasa query standar database. MySQL telah tersedia secara publik sejak tahun 1996 tetapi memiliki sejarah pengembangan dari tahun 1979. Ini adalah database open source yang amat populer di dunia dan telah memenangkan Linux Journal Readers' Choice Award pada beberapa kesempatan [9].

7) **Metode Agile**

Agile Methodology adalah metode pengembangan perangkat lunak yang berfokus pada pengembangan yang berulang, kolaborasi antar tim, dan adaptasi terhadap perubahan [12]. Model *Agile* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Alur metode *agile*

Adapun tahapan dari metode tersebut sebagai berikut.

- a. **Perencanaan:** Tim proyek dan pemangku kepentingan bersama-sama menetapkan tujuan proyek, menentukan ruang lingkup, dan merencanakan iterasi (sprint) pertama. Pada tahap ini, backlog produk juga dibuat, yang berisi daftar tugas dan fitur yang harus diselesaikan.
- b. **Analisis dan Desain:** Tim melakukan analisis kebutuhan lebih lanjut, mungkin dengan menggunakan teknik seperti user stories. Desain sistem atau fitur juga dibahas dan disusun, sering kali dengan pendekatan iteratif untuk memungkinkan perubahan seiring waktu.
- c. **Implementasi:** Tim mulai mengembangkan fitur atau fungsi yang telah direncanakan dalam sprint pertama. Kode ditulis, pengujian unit dilakukan, dan hasilnya diintegrasikan secara teratur ke dalam kode sumber.
- d. **Pengujian:** Setelah implementasi, tim melakukan pengujian terhadap fitur atau fungsi yang dikembangkan. Ini termasuk pengujian fungsional, pengujian integrasi, dan pengujian penerimaan pengguna. Setiap bug atau masalah yang ditemukan diperbaiki segera.
- e. **Evaluasi:** Setelah selesai satu iterasi atau sprint, hasilnya dievaluasi oleh tim dan pemangku kepentingan untuk mengevaluasi kinerja, mengidentifikasi pembelajaran, dan menentukan langkah selanjutnya. Evaluasi ini penting untuk meningkatkan proses dan hasil pada iterasi

berikutnya.

- f. **Iterasi:** Proses ini berulang secara terus-menerus melalui serangkaian sprint, biasanya dengan durasi dua hingga empat minggu. Setiap sprint berfokus pada pengembangan fitur atau fungsi tertentu dan menghasilkan deliverables yang dapat digunakan oleh pengguna.
- g. **Penyerahan:** Setelah serangkaian sprint, produk yang sudah selesai diserahkan kepada pemangku kepentingan untuk digunakan atau diluncurkan. Pengiriman ini dapat berupa rilis perangkat lunak, peningkatan produk, atau fitur tambahan.

8) **Payment Gateway**

Platform teknologi yang berperan sebagai perantara dalam transaksi keuangan elektronik. Fungsinya memfasilitasi bisnis menerima, memproses, dan mengelola berbagai metode pembayaran secara aman dan efisien. Adapun *payment gateway* yang digunakan pada penelitian ini ialah *platform payment gateway* Xendit.

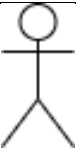


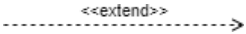

9) **Unified Modeling Language (UML)**

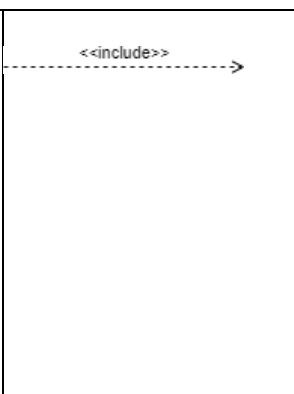
Unified Modeling Language (UML) memiliki arti permodelan standar. UML memiliki sintaks dan semantik dengan aturan yang harus diikuti. UML bukan hanya sekedar diagram tapi juga menceritakan konteksnya [13].

1) *Usecase Diagram*

Use case adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah actor [13]. Kegunaan use case sesungguhnya adalah untuk mendefinisikan suatu bagian perilaku sistem yang bersifat koheren tanpa perlu menyiapkan struktur internal sistem yang sedang dikembangkan.

Tabel 2.1 Simbol *usecase* diagram

No.	Simbol	Nama simbol	Keterangan
1		Actor	Menggambarkan peran orang atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri.
2		<i>Use case</i>	Fungsionalitas yang disediakan oleh sistem, sebagai unit untuk bertukar pesan antar unit atau actor.
3		Asosiasi / <i>association</i>	Komunikasi antara actor dan use case yang berpartisipasi pada use case dan memiliki interaksi dengan actor.
4		Ekstend / <i>extend</i>	Relasi use case tambahkesebuah use case, dimana use case yangditambahkan dapat berdiri sendiri walaupun tanpa use case tambahan.
5		Generalisasi / <i>generalization</i>	Hubungan generalisasi dengan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

6		Menggunakan / <i>include</i>	Relasi use case tambahkesebuah usecase,dimana usecase yang ditambahkan memerlukan use case iniuntuk menjalankan fungsinya.
---	---	---------------------------------	--

2) Activity Diagram

Activity diagram memodelkan *workflow* proses bisnis dan urutan aktifitas dalam sebuah proses. Diagram ini sangat mirip dengan flowchart karena memodelkan *workflow* dari suatu aktifitas lainnya atau dari aktifitas ke status [14].

Tabel 2.2 Simbol *activity diagram*

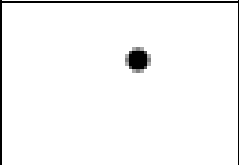
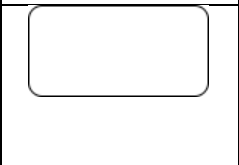
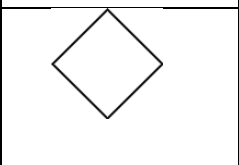
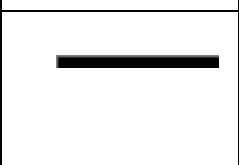
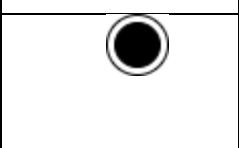
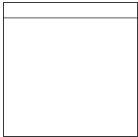






No.	Simbol	Nama symbol	Keterangan
1		Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
2		Aktivitas	Aktivitas yang dilakukan system, aktivitas biasanya diawali dengan kata kerja.
3		Percabangan / <i>decision</i>	Percabangan yang mana ada pilihan aktivitas lebih dari satu.
4		Penggabungan / <i>join</i>	Penggabungan yang mana lebih dari satu aktivitas digabungkan menjadi satu
5		Status akhir	Status akhir yang dilakukan oleh system, sebuah

			diagram aktivitas memiliki status akhir.
6		Swimlane	Swimlane Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

3) Class Diagram

Diagram kelas atau *class diagram* berfungsi menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi [14], simbol - simbol yang ada pada diagram kelas pada tabel *class diagram 2.3* dibawah ini:




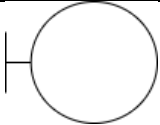

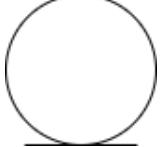
Tabel 2.3 Simbol class diagram


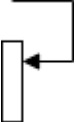
Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur system
<p>Antarmuka/<i>Interface</i></p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
<p>Asosiasi/<i>association</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p>Asosiasi berarah/<i>directed association</i></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i>
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
<p>Agregasi/<i>aggregation</i></p> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

4) *Sequence Diagram*

Sequence diagram menggambarkan interaksi antara sejumlah objek dalam urutan waktu [14].

Tabel 2.4. Simbol *sequence diagram*

No.	Simbol	Nama simbol	Keterangan
1		Actor	Menggambarkan peran orang atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri.
2		Garis Hidup <i>Lifeline</i>	Komponen yang digambarkan garis putus menghubungkan suatu objek.
3		<i>Activation</i>	Menyatakan objek dalam keadaan aktif dan berinteraksi.
4		<i>BoundaryClass</i>	Menangani komunikasi antar lingkungan system.
5		Control class	Bertanggung jawab terhadap kelas-kelas objek yang berisi logika.
6		Entity class	Gambaran system sebagai landasan Menyusun basis data

7		A message	Menggambarkan pengiriman pesan.
8		Recursive	Pesan untuk dirinya

b. Pengujian Sistem *Black Box*

Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. dengan demikian, peingujian *black-box* keimungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program (Heirdiansah eit al., 2020) pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

- a. Fungsi – fungsi yang tidak beinar atau hilang,
- b. Keisalahan *inteirfacei*
- c. Keisalahan dalam struktur data atau akseis eiksteirnal
- d. Keisalahan kineirja
- e. Inisialisasi dan keisalahan teirminasi