

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian yang dilakukan oleh Eni Rohaini (2024), dalam jurnal yang berjudul Sistem Informasi Manajemen Keuangan Sekolah *Excellent Mandiri* School Kota Jambi menghasilkan sebuah sistem keuangan sekolah yang diharapkan nantinya dapat membantu meningkatkan kinerja bendahara lebih efisien dan efektif dalam mengelola keuangan sekolah dan laporan mudah dipahami. sistem manajemen keuangan sekolah ini menyediakan laporan keuangan secara visual pada menu beranda admin, yang mana dapat diketahui jumlah saldo yang tersedia, jumlah pengeluaran, jumlah setoran infak dan tabungan siswa. Pada sistem juga terdapat upgrade kelas siswa sehingga secara otomatis mengikuti pembayaran biaya sesuai kelas nya, hal ini membantu staff bagian keuangan memantau keuangan sekolah terkait pembayaran SPP siswa di setiap kelasnya[1].

Dirancang untuk membantu staf keuangan dalam mengawasi tugas-tugas administrasi sekolah seperti arus kas masuk, arus kas keluar, dan pelaporan, penelitian Dodi Hendarman (2022) berjudul "Pengembangan Sistem Informasi Keuangan Sekolah Berbasis Web Menggunakan Metodologi Prototipe di SMAN 2 Cikarang Pusat" merupakan usaha yang membuahkan hasil. Evaluasi sistem informasi menemukan bahwa sistem informasi keuangan yang baru diimplementasikan sangat menyederhanakan proses pengelolaan data untuk manajemen keuangan sekolah [2].

Fifin Lasma menerbitkan penelitian lanjutan berjudul "Sistem Informasi Akuntansi Pembayaran SPP di SDN Imelda Medan" yang membuat program yang menggantikan sistem manual dengan sistem berbasis komputer. Kebijakan dan persyaratan sekolah dalam menangani sistem pembayaran SPP menginformasikan pengembangan aplikasi ini.

"Model Aplikasi Pengolahan Data Keuangan di Sekolah Dasar" adalah judul penelitian selanjutnya Dwi Mulyani. Untuk mengefisiensikan pengelolaan data keuangan dan penyusunan laporan, penelitian ini bermaksud untuk membangun aplikasi untuk SD Islam Babul Jannah Banjarbaru. Hal ini didukung oleh temuan kuesioner survei. Berdasarkan survei, 80% pengguna menilai aplikasi ini dapat membantu pengelolaan data, 80% menilai dapat mempermudah dan mempercepat pengelolaan keuangan, 60% setuju, dan 40% sangat setuju dapat memperbaiki keterlambatan pelaporan.

Kelima, Evanita melakukan penelitian dengan judul "Rancangan Sistem Informasi Pencatatan Keuangan Berbasis Website di TK Lestari" dengan tujuan untuk memanfaatkan teknologi guna meningkatkan akurasi pelaporan keuangan, mengurangi human error, dan mempermudah pengelolaan keuangan. Dengan demikian, TK Lestari dapat menjalankan perannya sebagai PAUD dengan lebih baik, sehingga dapat memberikan manfaat bagi pendidikan dan tumbuh kembang anak di Kabupaten Pati.

2.2 Landasan Teori

2.2.1 Pengertian Sistem

Sistem adalah sekumpulan komponen atau jaringan kerja dari prosedur-prosedur yang saling berkaitan dan saling bekerja sama membentuk suatu jaringan kerja untuk mencapai sasaran atau tujuan tertentu[4].

2.2.2 Pengertian Informasi

informasi merupakan data yang sudah diolah yang ditujukan untuk seseorang, organisasi ataupun siapa saja yang membutuhkan. Informasi akan menjadi berguna apabila objek yang menerima informasi membutuhkan informasi tersebut[5].

2.2.3 Pengertian Sistem Informasi

Peranan sistem informasi dalam suatu organisasi sangatlah penting agar suatu organisasi dapat beroperasi secara baik dan memberikan nilai manfaat serta memahami lingkungan organisasi. Agar tujuan organisasi dapat tercapai diperlukan sistem informasi yang baik[6].

2.2.4 Pengertian Website

Sebuah situs *web* (sering pula disingkat menjadi situs, *website* atau *site*) adalah sebutan bagi sekelompok halaman *web* yang umumnya merupakan bagian dari suatu nama domain atau subdomain di WWW (*World Wide Web*) di internet[9]. WWW terdiri dari seluruh situs *web* yang tersedia kepada pengguna. Halaman sebuah situs web

diakses dari sebuah URL yang menjadi akar (*root*) yang disebut homepage (halaman induk), dan biasanya disimpan dalam server yang sama[7].

2.2.5 Pengertian Database MySQL

Database adalah suatu kumpulan data yang disusun sedemikian rupa sehingga membentuk informasi yang sangat berguna. *Database* berbentuk dari sekelompok data yang memiliki jenis/sifat yang sama. Ambil contoh, data berupa nama, kelas, dan alamat. Semua data tersebut dikumpulkan menjadi satu menjadi kelompok data baru, sebut saja sebagai data mahasiswa.



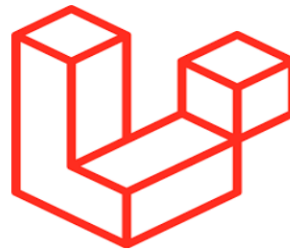
Gambar 2. 1 Logo MySQL

2.2.6 Pengertian Framework Laravel

Laravel merupakan sebuah kerangka kerja aplikasi berbasis *PHP* yang sumber terbuka dengan menggunakan konsep MVC. Laravel berada dibawah lisensi MIT dengan menggunakan GitHub sebagai tempat berbagi kode.

Laravel termasuk salah satu *Framework PHP* yang paling besar dan banyak digunakan, hal ini menjadikan laravel memiliki banyak tutorial dan komunitas yang dapat membantu bila mengalami kesulitan dalam membangun aplikasi dan laravel memiliki *third-party module* yang dapat unduh secara gratis.

Laravel dikembangkan oleh Taylor Otwell dan dirilis perdana pada Juni 2011 dan mulai dengan stabil pada 8 September 2020 dengan bahasa pemrograman yang digunakan *PHP 7*. Konsep MVC.



Gambar 2. 2 Logo Framework Laravel

2.2.7 Pengertian Visual Studio Code

Visual studio code merupakan kode editor sumber yang dikembangkan oleh Microsoft untuk Windows, Linux dan macOS.[16] Ini termasuk dukungan untuk *debugging*, kontrol git yang tertanam dan GitHub, penyorotan sintaks, penyelesaian kode cerdas, *snippet*, dan refactoring kode.

2.2.8 UML (*Unified Modeling Language*)


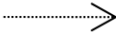

Unified Modeling Language merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah software yang berorientasikan pada objek. UML

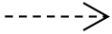






merupakan sebuah standar penulisan atau semacam blue print diamna didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik[8]. Terdapat beberapa diagram UML yang sering digunakan dalam pengembangan sebuah sistem. Terdapat beberapa diagram UML yang sering digunakan dalam pengembangan sebuah sistem yaitu:

1. Use Case

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah apa yang diperbuat sistem, dan bukan bagaimana. Sebuah use case mempresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang atau sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Tabel 2. 1 Simbol Use Case Diagram







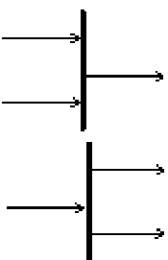

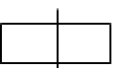
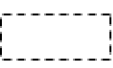
No	Gambar	Nama	Keterangan
1.		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2.		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya (dependent).
3.		Generalization	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data

No	Gambar	Nama	Keterangan
			dari objek yang ada di atasnya objek induk (ancertor).
4.		Include	Menspesifikasikan bahwa use case sumber secara eksplisit.
5.		Extend	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
6.		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7.		Sistem	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8.		Use Case	Deskripsi dari urutan aksi – aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
9.		Collaboration	Interaksi aturan – aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen – elemennya (sinergi).
10.		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

2. Activity Diagram

Sebuah diagram aktivitas UML menggambarkan perilaku dinamis dari suatu sistem atau bagian dari sistem melalui aliran kontrol antara aksi yang dilakukan sistem.

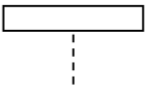
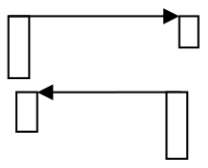

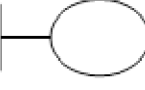

Tabel 2. 2 Simbol Activity Diagram




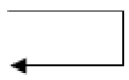
No	Gambar	Nama	Keterangan
1.		Activity	Memperlihatkan bagaimana masing - masing kelas antarmuka saling berinteraksi satu sama lain.
2.		Action	State dari sistem yang mencerminkan eksekusi suatu aksi.
3.		Initial Node	Bagaimana objek dibentuk atau diawali.
4.		Final Node	Pilihan untuk mengambil keputusan
5.		Fork Node	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.
6.		Decision	Pilihan untuk mengambil keputusan
7.		Fork/Join	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
8.		Rake	Menunjukkan adanya dekomposisi
9.		Time	Tanda waktu
10.		Send	Tanda pengiriman

3. Sequence Diagram

Sequence diagram menggambarkan interaksi antar di sekitar (pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek - objek yang terkait). *Sequence* diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah - langkah yang dilakukan sebagai respon dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Masing - masing objek, termasuk aktor, memiliki lifeline vertikal.

Tabel 2. 3 Simbol Sequence Diagram

No	Gambar	Nama	Keterangan
1.		LifeLine	Objek entity, antar muka yang saling berinteraksi.
2.		Message	Spesifikasi dari komunikasi antar objek yang memuat informasi – informasi tentang aktifitas yang terjadi.
3.		Actor	Menggambarkan orang yang sedang berinteraksi dengan sistem
4.		Boundary Class	Menggambarkan penggambaran dari form
5.		Entity Class	Menggambarkan hubungan kegiatan yang akan dilakukan

No	Gambar	Nama	Keterangan
6.		Control Class	Menggambarkan penghubung antara Boundary dengan tabel
7.		Activation	Sebagai sebuah objek yang akan melakukan sebuah aksi
8.		Message	Mengindikasikan komunikasi antara objek dengan objek
9.		Self Message	Menginndikasikan komunikasi kembali kedalam sebuah objek itu sendiri

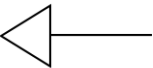
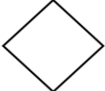
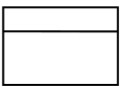


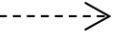

4. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok : nama (*stereotype*), atribut, dan metoda. Atribut dan metoda dapat memiliki salah satu sifat berikut:

- a. *Private*, tidak dapat dipanggil dari luar class yang bersangkutan.
- b. *Protected*, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya.

Tabel 2. 4

Tabel 2. 4 Simbol Class Diagram

No	Gambar	Nama	Keterangan
1.		Generalization	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>)
2.		Nary Association	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3.		Class	Himpunan dari objek - objek yang berbagi atribut serta operasi yang sama.
4.		Collaboration	Deskripsi dari urutan aksi - aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
5.		Dependency	Operasi yang benar - benar dilakukan oleh suatu objek.
6.		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
7.		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.