

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Penelitian yang dilakukan oleh Hartono, dkk.(2020) ini telah dipublikasikan dalam sebuah makalah jurnal yang berjudul “Rancang Bangun Aplikasi *E-raport* disekolah SMP Negeri 29 Gresik Berbasis *WEB*”. Tujuannya adalah untuk mendeskripsikan perencanaan program *e-rapor* Pengumpulan data dilakukan melalui studi wawancara, observasi, dan dokumentasi, tujuannya agar setiap kegiatan penilaian dapat berjalan secara maksimal, sebab kegiatan-kegiatan tersebut akan berdampak positif bagi warga sekolah khususnya para bapak dan ibu guru dalam proses *penginputan* nilai ke dalam *e-rapor* [4].

Febriansah, dkk. (2023), penelitian mengenai “Perancangan Aplikasi Hasil Belajar Siswa (Rapor)”. sistem aplikasi berbasis *Web* yang di harapkan dapat merubah pola kerja guru daripola manual ke pola digital dan juga dapat mempermudah guru dalam melakukan penilaian ke siswa bahkan sampai ke cetak rapor dan evaluasi nilai hasil belajar siswa, Dalam penelitian ini, selain memaparkan kajian teori yang digunakan sebagai dasar penyusunan, juga akan dibahas mengenai perancangan sistem dan pembuatan aplikasinya. Sehingga dapat diharapkan dan diimplementasikan dalam bentuk sebuah produk yang akan memperbaiki segala kekurangan yang ada pada sistem yang lama [5].

Menurut penelitian yang dilakukan oleh Fajril, dkk. (2023).

Dalam jurnal yang berjudul "Implementasi penggunaan rapor digital madrasah (rdm) berbasis online dalam menyusun administrasi penilaian hasil belajar peserta didik." *Idarah: Jurnal Manajemen Pendidikan* 7.1 (2023): 177-189, bertujuan untuk memberikan deskripsi terkait dengan kemudahan dalam menggunakan rapor digital madrasah secara online. penelitian menunjukkan bahwa penggunaan rapor digital madrasah berbasis online lebih efektif dibandingkan dengan rapor konvensional dan rapor pada Microsoft Excel [6].

Penelitian lainnya menurut Rahmat, dkk. (2023), Dalam Skripsi yang berjudul tentang "*Analisis Penerapan Aplikasi Rapor Digital Madrasah Pada Madrasah Tsanawiyah Negeri 6 Montasik Aceh Besar*". Tujuan penelitian ini yaitu Untuk Mengetahui bagaimana penilaian guru Madrasah Tsanawiyah Negeri 6 Montasik terhadap penggunaan Aplikasi Rapor Digital Madrasah, dan untuk mengetahui bagaimana perubahan sisi penggunaan dan pengaruhnya terhadap penggunaan Aplikasi Rapor Digital Madrasah [7].

Dalam penelitian yang dilakukan oleh Hairony, dkk. (2023). "Sistem Informasi Penilaian Kinerja Guru Dalam meningkatkan Profesionalisme Guru di SDN Batu Ampar 13 Pagi Berbasis Web". *Journal of Informatics and Advanced Computing (JIAC)*, 4(1), 65-75. Bertujuan sebuah aplikasi penilaian kinerja guru yang dapat memudahkan proses pemberian penilaian, meminimalisir kesalahan perhitungan, dan terjaganya setiap data yang ada. Aplikasi ini berbasis Web dengan menggunakan Bahasa pemrograman Php

dan *database* MySQL, Oleh karena itu, dengan adanya aplikasi penilaian kinerja guru dapat memudahkan dalam memberi dan mengelola penilaian serta dalam penyajian laporan hasil penilaian kinerja guru yang dibutuhkan dapat diperoleh secara tepat, cepat, dan real time [8].

2.2. Landasan Teori

2.2.1. Sistem Informasi

Sistem informasi dapat dikatakan untuk menyediakan informasi yang tepat, Pada waktu yang tepat pula dan dalam format yang tepat kepada para penggunannya. Sistem informasi ini dapat diterapkan dalam berbagai konteks, seperti bisnis, pendidikan, kesehatan, pemerintahan, dan berbagai sector lainnya untuk membantu meningkatkan efisiensi, efektivitas, dan pengambilan keputusan yang lebih baik lagi, (Triatama, 2023) A. S., & Sintaro, S. (2023) [9].

2.2.2. Pengolahan Nilai Siswa

Pengolahan nilai siswa menurut (Maria & Muawanah, 2018) Bagian dari kegiatan belajar mengajar di sekolah, merupakan bagian penting dari belajar mengajar sebagai tolak ukur keberhasilan siswa. Dengan praktik yang baik, kurasi data akan mudah, cepat dan akurat. Ada tiga langkah utama dalam proses pengolahan data yaitu.

1) *Input* merupakan proses masukkan data ke dalam proses

komputer lewat alat (*input drive*). Processing merupakan pengolahan data masukan diolah dengan pengolahan yaitu komputasi penyimpanan, pilihan komparatif, sortir kontrol atau alur pencarian.

- 2) *Output* merupakan proses menghasilkan keluaran berupa informasi dari hasil pengolahan data ke alat keluaran (*output device*).

Pengolahan data nilai siswa adalah data siswa yang diolah menjadi nilai harian, rata-rata semester, dan nilai semester dengan menggunakan sistem komputerisasi yang memberikan data yang cepat, akurat, dan tepat pada saat [10].

2.2.3. Web (*wold wide Web*)

Web adalah salah satu layanan yang ada di internet yang memungkinkan pengguna untuk mengakses dan berbagi informasi melalui media digital. *Web* terdiri dari berbagai situs *Web* yang terhubung satu sama lain melalui jaringan internet. Situs *Web* adalah kumpulan halaman *Web* yang berisi teks, gambar, video, atau elemen multimedia lainnya yang dapat diakses oleh pengguna melalui peramban *Web* (*Web browser*), (Ramadhanti, 2024) [11].

Di *Web* pengguna dapat melakukan berbagai aktivitas, seperti mencari informasi, berkomunikasi dengan orang lain melalui email atau media sosial, berbelanja online, bermain game, menonton video, dan banyak lagi. *Web* telah menjadi salah satu aspek penting

dalam kehidupan sehari-hari banyak orang, dan memiliki dampak besar terhadap cara kita bekerja, belajar, berkomunikasi, dan berinteraksi dengan dunia di sekitar kita.

2.2.4. Visual Studio Code

Visual Studio Code (Vc Code) ini adalah sebuah teks editor ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi *Linux*, *Mac* dan *Windows*. Teks editor ini secara langsung mendukung bahasa pemrograman *JavaScript*, *Typescript* dan *Node.js*, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang melalui *marketplace VS Code* (seperti *C++*, *C#*, *Python*, *Go*, *Java* dan beberapa bahasa pemrograman lainnya).

Banyak sekali fitur-fitur yang disediakan oleh *VS Code*, diantaranya *Intellisense*, *Git Intregation*, *Debugging* dan fitur ekstensi yang menambah kemampuan teks editor. Fitur-fitur tersebut akan terus bertambah seiring dengan bertambahnya versi *VS Code*. Pembaruan versi *VS Code* ini juga dilakukan berkala setiap bulan, dan inilah yang membedakan *VS Code* dengan teks editor yang lain[12].

2.2.5. MYSQL

MySQL atau kepanjangan dari *My Structured Structured Query Language*. *MySQL* merupakan bahasa terstruktur yang khusus digunakan untuk mengolah *database*. *MySQL* pertama kali

didefinisikan oleh American National Standards Institute (ANSI) pada tahun 1986. MySQL adalah sebuah sistem manajemen *database* yang bersifat open source MySQL dapat digunakan untuk membuat dan mengola *database* beserta isinya[13].

2.2.6. Database

Basisdata (database) adalah suatu alat yang digunakan untuk menyimpan informasi, mengambil informasi kapanpun dibutuhkan, dan mengatur informasi yang tersimpan. Jika menggambarkan lemari file merupakan suatu *basisdata*.

DBMS (Database Management System) merupakan suatu perangkat lunak yang digunakan untuk membuat, memelihara, mengontrol dan mengakses *basisdata* secara praktis dan efisien. Sedangkan *RDBMS* merupakan salah satu *DBMS* yang mendukung adanya relasi atau hubungan (Gayatri, 2024) [14].

2.2.7. XAMPP

XAMPP dikembangkan dari sebuah tim proyek bernama Apache Friends, yang terdiri dari Tim Inti (Core Team), Tim Pengembang (Development Team) & Tim Dukungan (Support Team). XAMPP adalah singkatan yang masing-masing hurufnya adalah: X=Program ini dapat dijalankan dibanyak sistem operasi seperti: Windows, Linux, Mac Os, dan Solaris. A=Apache, merupakan aplikasi *Web* server. Tugas utama apache adalah menghasilkan halaman *Web* yang benar kepada user berdasarkan

kode PHP yang dituliskan oleh pembuat halaman *Web* [15].

2.2.8. PHP

PHP (Hypertext Preprocessor) Pertama kali ditemukan pada 1995 oleh seorang *Software Developer* bernama Rasmus Lerdorf. Ide awal *PHP* adalah ketika itu Rasmus ingin mengetahui jumlah pengunjung yang membaca *resume online*. *Script* yang dikembangkan baru dapat melakukan dua pekerjaan, yakni merekam informasi pengunjung dan menampilkan jumlah pengunjung dari suatu *website*. Sampai sekarang kedua tugas tersebut masih tetap populer digunakan di dunia web saat ini. Kemudian, dari situ banyak orang di milis mendiskusikan *script* buatan Rasmus Lerdorf, hingga akhirnya rasmus mulai membuat sebuah *tool/script* bernama *PHP*, *PHP* dibangun dari *scripts* yang ditulis secara *plaintext*. *PHP Interpreter* adalah bagian dari perangkat lunak yang ada pada *Web Server*, yang membaca file tersebut dan mengartikannya, memberikan keluaran *HTML* dan petunjuk mengenai bagaimana perilaku yang ada maupun menginterpretasikan masukan dari pengguna [16].

2.2.9. HTML

HTML (Hypertext Markup Language) adalah suatu cara memberikan tanda yang memberikan perintah kepada *browser* bagaimana suatu teks terstruktur. *HTML* memberikan perintah kepada *browser* bagaimana struktur dari dokumen, bagaimana

heading-nya, bagaimana paragrafnya, bagaimana suatu teks akan ditampilkan dan lainnya. Dengan informasi yang diberikan, *browser* dibangun dengan perintah dasar bagaimana menampilkan setiap elemen yang ada [17].

2.2.10. CSS

CSS (Cascading Style Sheets) adalah aturan untuk mengatur beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. *CSS* bukan merupakan bahasa pemrograman. Sama halnya *styles*, misalnya *heading*, *subbab*, *footer*, *images*, dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (file). Pada umumnya *CSS* dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa *HTML* dan *XHTML* [18].

2.2.11. UML


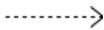
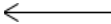

Unified Modeling Language atau lebih sering dikenal dengan sebutan *UML*, adalah salah satu metode dalam teknik rekayasa perangkat lunak yang digunakan untuk menggambarkan alur dan cara kerja sistem, fungsi, tujuan dan mekanisme kontrol sistem tersebut. Dalam teknik rekayasa perangkat lunak bidang analisis dan perancangan sistem informasi, saat ini lebih banyak menggunakan gabungan dari konsep pemrograman berorientasi objek dengan teknik pembuatan perangkat lunak, dimana suatu sistem dilihat sebagai objek tersendiri yang sudah mencakup data







dan proses atau dapat bekerja secara mandiri dalam satu set sistem (*package*).

1. Use Case Diagram

Use Case diagram digunakan untuk mengkomunikasikan interaksi manusia (*actor*) dengan apa yang bisa dilakukan oleh sistem. Sebuah *Use Case* dapat mewakili beberapa jalur interaksi manusia dengan sistem dan setiap jalur disebut sebagai skenario [19]. Simbol *Use Diagram* dapat dilihat seperti pada Tabel 2.1.

Tabel 2. 1 Use Case Diagram

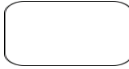




NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang dimainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.

NO	GAMBAR	NAMA	KETERANGAN
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-ak yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	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. Diagram Activity (Activity Diagram)

Activity diagram adalah diagram yang menggambarkan alur kerja atau aktivitas suatu sistem dalam perangkat lunak. Simbol *Activity Diagram* dapat dilihat seperti pada Tabel 2.2.

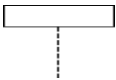

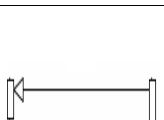
Tabel 2. 2 Diagram Activity

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actifty</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Actifty</i> <i>Final Node</i>	Bagaimana objek dibentuk dan dihancurkan.
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

3. Diagram Urutan (Sequence Diagram)

Sequence diagram mengilustrasikan objek-objek yang terdapat pada *Use Case* dan menggambarkan arus pesan antara satu sama lain pada Objek *Use Case*. *Sequence Diagram* bersifat dinamis dan lebih banyak menampilkan aktifitas objek berdasarkan urutan waktu[18]. Simbol *Sequence Diagram* dapat dilihat seperti pada Tabel 2.3.


Tabel 2. 3 Sequence Diagram


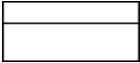





NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entit</i> antar muka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

4. Class Diagram

Class Diagram adalah model statis yang mendukung tampilan data dan informasi dari keseluruhan sistem. Penggunaan *Class Diagram* dikaitkan dengan struktur basis data sistem atau dapat menggantikan ERD pada proses penggambaran diagram rekayasa perangkat lunak yang konvensional. Simbol *Class Diagram* dapat dilihat seperti pada Tabel 2.4.

Tabel 2. 4 Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).

NO	GAMBAR	NAMA	KETERANGAN
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya
8		Delay Symbol	Adalah simbol yang digunakan untuk menunjukkan proses delay (menunggu) yang perlu dilakukan. Seperti menunggu surat untuk diarsipkan dll.