

BAB II

KAJIAN LITERATUR

2.1 Sistem Informasi Akademik

Sistem informasi merupakan sarana transformasi data dalam suatu organisasi yang mengelola kebutuhan pengolahan transaksi harian untuk mendukung operasional dan efisiensi organisasi [9]. Ketika diterapkan di lingkungan institusi pendidikan, sistem semacam ini lebih dikenal dengan istilah Sistem Informasi Akademik (SIA).

SIA merupakan sebuah platform digital komprehensif yang dirancang untuk mengelola berbagai entitas administrasi pendidikan secara terpusat. Transisi dari metode pencatatan data konvensional (seperti penggunaan arsip fisik atau lembar kerja yang tidak terintegrasi) ke dalam sistem basis data yang terpadu sangat esensial untuk mencegah terjadinya redudansi (penggandaan) dan inkonsistensi data[6].

Implementasi SIA bertujuan untuk meminimalisir tingkat kesalahan manusia (human error) dalam pengolahan data akademik komposit, seperti perhitungan nilai matematis atau rekapitulasi tingkat kehadiran. Selain itu, digitalisasi ini secara fungsional mempercepat alur kerja staf administrasi dan guru, serta mampu menghasilkan luaran informasi pelaporan seperti dokumen hasil belajar yang tervalidasi, akurat, dan dapat diakses secara seketika (*real-time*). Ruang lingkup pengelolaan pada sistem ini umumnya mencakup manajemen data induk pengguna (siswa dan pendidik), penjadwalan, pencatatan presensi, hingga pengolahan evaluasi akademik.

2.2 Sistem Penilaian dan Pelaporan Akademik (Rapor)

Penilaian hasil belajar dan pelaporan akademik (rapor) merupakan instrumen utama dalam sistem pendidikan untuk mengevaluasi tingkat ketercapaian kompetensi peserta didik. Pengelolaan data nilai di tingkat Sekolah Menengah Atas (SMA) membutuhkan tingkat akurasi yang tinggi. Oleh karena itu, digitalisasi sistem pengolahan nilai berbasis web menjadi solusi untuk menggantikan proses pencatatan konvensional yang rentan terhadap galat (*human error*)[10].

2.2.1 Pengolahan Nilai dan Regulasi Penilaian di SMA

Sistem E-Rapor SMA yang dikembangkan dalam penelitian ini berpijak pada Permendikbudristek Nomor 12 Tahun 2024, yang mengatur struktur kurikulum SMA dalam

dua fase: Fase E untuk kelas X dan Fase F untuk kelas XI dan XII [11]. Pembagian fase ini secara langsung memengaruhi cara pengelompokan mata pelajaran dalam perancangan basis data sistem.

Di sisi pengolahan nilai, acuan utamanya adalah Permendikbudristek Nomor 21 Tahun 2022 tentang Standar Penilaian Pendidikan. Beberapa ketentuan dalam regulasi ini membawa perubahan yang cukup signifikan terhadap logika perhitungan nilai dalam sistem rapor, antara lain [12]:

1. Peleburan Dimensi Penilaian, sistem penilaian tidak lagi memisahkan secara kaku antara nilai pengetahuan (kognitif) dan keterampilan (psikomotorik). Keduanya dilebur menjadi satu kesatuan Nilai Akhir Mata Pelajaran yang bersumber murni dari asesmen sumatif [13].
2. Otonomi Satuan Pendidikan, pemerintah tidak lagi menetapkan formula persentase baku secara nasional untuk ujian harian, tengah semester, maupun akhir semester. Sekolah diberikan otonomi penuh untuk menentukan bobot nilai secara mandiri, serta menetapkan batas minimal persentase kehadiran (presensi) sebagai syarat kenaikan kelas di dalam kurikulum operasionalnya [13].
3. Pemisahan Laporan Karakter (Rapor P5), penilaian sikap dikelola melalui dokumen terpisah dari rapor akademik intrakurikuler, yakni Proyek Penguatan Profil Pelajar Pancasila (P5) yang murni bersifat kualitatif dan deskriptif [14].

2.2.2 Komponen Rapor Akademik

Rapor merupakan dokumen resmi yang memuat rekam jejak capaian akademik siswa. Sebagai pangkalan data historis dan dokumen cetak formal sekolah, format penyusunan rapor menuntut standardisasi tata letak yang konsisten.

Pengembangan bentuk pelaporan hasil belajar ini berlandaskan pada Peraturan Menteri Pendidikan, Kebudayaan, Riset, dan Teknologi (Permendikbudristek) Nomor 21 Tahun 2022 tentang Standar Penilaian Pendidikan. Regulasi ini mengamanatkan bahwa pelaporan hasil belajar peserta didik harus disusun berdasarkan pengolahan data penilaian yang objektif, akurat, dan sistematis [12]. Selain itu, mengacu pada Permendikbudristek Nomor 16 Tahun 2022 tentang Standar Proses, pemantauan kehadiran (presensi) merupakan bagian tak terpisahkan dari pelaksanaan dan evaluasi pembelajaran, sehingga pencatatannya mutlak harus terintegrasi di dalam dokumen rapor [15].

Berdasarkan regulasi dan panduan teknis asesmen yang diterbitkan oleh pemerintah, satuan pendidikan memiliki keleluasaan dalam menentukan mekanisme pelaporan, namun

terdapat komponen informasi minimal yang wajib termuat di dalam rapor. Mengacu pada Panduan Pembelajaran dan Asesmen [16], sebuah dokumen rapor akademik tingkat Sekolah Menengah Atas (SMA) setidaknya harus memuat 12 informasi berikut:

1. Identitas murid: Memuat informasi dasar seperti Nama Lengkap dan Nomor Induk Siswa Nasional (NISN) / Nomor Induk Siswa (NIS)
2. Nama satuan pendidikan: Memuat identitas institusi sekolah yang menerbitkan rapor.
3. Kelas: Menunjukkan tingkatan rombongan belajar peserta didik.
4. Semester: Menunjukkan periode waktu pelaksanaan pembelajaran (Ganjil atau Genap).
5. Mata pelajaran: Memuat daftar mata pelajaran yang diampu oleh peserta didik sesuai dengan struktur kurikulum yang berlaku (Fase E atau Fase F).
6. Nilai: Berupa angka kuantitatif (skala 0-100) yang merupakan hasil pengolahan dari berbagai asesmen formatif dan sumatif.
7. Deskripsi Capaian Kompetensi: Penjelasan kualitatif yang menjabarkan kompetensi apa saja yang sudah dicapai dengan baik oleh peserta didik, serta kompetensi yang masih memerlukan bimbingan.
8. Deskripsi Capaian Kokurikuler: Penjelasan terkait perkembangan karakter dan keterlibatan peserta didik dalam kegiatan kokurikuler, seperti Proyek Penguatan Profil Pelajar Pancasila (P5).
9. Kegiatan ekstrakurikuler: Memuat informasi partisipasi dan predikat peserta didik dalam kegiatan pengembangan diri di luar jam pelajaran inti.
10. Ketidakhadiran: Rekapitulasi jumlah ketidakhadiran peserta didik selama satu semester (Sakit, Izin, dan Tanpa Keterangan).
11. Catatan Wali Kelas: Ruang untuk memberikan masukan, evaluasi formatif, atau motivasi dari wali kelas kepada peserta didik.
12. Tanggapan Orang Tua/Wali Murid: Kolom yang disediakan bagi orang tua atau wali untuk memberikan umpan balik dan membubuhkan tanda tangan sebagai bukti telah menerima laporan hasil belajar.

Format Laporan Hasil Belajar (Rapor) Jenjang SMA

Nama Murid : _____

NISN : _____

Sekolah : _____

Alamat : _____

Kelas : X

Fase : _____

Semester : _____

Tahun Ajaran : _____

No.	Mata Pelajaran	Nilai Akhir	Capaian Kompetensi
9	Informatika		
10	Seni Budaya, dan Prakarya		

Kokurikuler	
No.	Keterangan
1	Pramuka
2	PMR
dst.	

Ekstrakurikuler	
No.	Keterangan
1	Pramuka
2	PMR
dst.	

Ketidakhadiran	
Sakit	... hari
Izin	... hari
Tanpa Keterangan	... hari

Catatan Wali Kelas	
No.	Keterangan
1	
2	
dst.	

Tanggapan Orang Tua/Wali Murid		
Orang Tua Murid	Kepala Sekolah	Tempat, Tanggal rapor Wali Kelas
TTD	TTD	TTD

Gambar 2. 1 Format Laporan Hasil Belajar (Rapor) Jenjang SMA [16]

2.3 Framework Laravel

Kerangka kerja (*framework*) adalah sekumpulan pustaka (*library*) dan struktur kode yang telah disusun sedemikian rupa untuk mempermudah, mempercepat, serta menstandarisasi proses pengembangan perangkat lunak berbasis web [17]. Dalam ekosistem pengembangan web menggunakan bahasa pemrograman *PHP*, *Laravel* merupakan salah satu kerangka kerja terpopuler yang menawarkan sintaks ekspresif dan elegan.

Pada perancangan sistem informasi akademik ini, versi yang digunakan adalah *Laravel 12*. Penggunaan versi ini bertujuan untuk memastikan sistem E-Rapor mendapatkan pembaruan keamanan siber terkini, peningkatan kecepatan komputasi, serta efisiensi dalam pengelolaan direktori proyek yang lebih ramping dibandingkan versi pendahulunya [18].

Secara arsitektural, *Laravel* mengimplementasikan pola *Model-View-Controller* (MVC) secara terstruktur. Pola arsitektur MVC memisahkan kerangka aplikasi menjadi tiga komponen utama yang saling berinteraksi. Pemisahan ini bertujuan untuk memudahkan pengelolaan kode yang kompleks dengan membedakan antara logika bisnis, manipulasi

pangkalan data, dan tampilan antarmuka bagi pengguna [18]. Ketiga komponen tersebut meliputi:

1. *Model*: Komponen yang bertugas secara spesifik untuk menangani logika manipulasi, struktur, dan interaksi langsung dengan pangkalan data (*database*).
2. *View*: Komponen yang berfungsi untuk mengatur tata letak, presentasi data, dan visualisasi antarmuka pengguna (*User Interface*) di peramban web.
3. *Controller*: Komponen pengatur alur logika pusat yang menerima permintaan (*request*) dari pengguna, memprosesnya melalui *Model*, dan menentukan *View* mana yang akan dikembalikan sebagai respons akhir.

2.4 Metode Agile

Metode *Agile* merupakan pendekatan pengembangan perangkat lunak yang menekankan fleksibilitas, kolaborasi tim, iterasi cepat, dan respons terhadap perubahan. Berbeda dengan metode pengembangan tradisional yang bersifat sekuensial, *Agile* berfokus pada pengiriman nilai (*value*) fungsionalitas sistem kepada pengguna secara berkelanjutan [19].

Secara lebih spesifik, metode *Agile* memiliki karakteristik utama sebagai berikut [20]:

1. Iterasi pendek dan inkremental: Pekerjaan dibagi ke dalam unit-unit kecil yang dikembangkan dan dikirimkan secara berkala.
2. Kolaborasi intensif: Komunikasi yang aktif antar anggota tim pengembang, serta keterlibatan pemangku kepentingan (*stakeholder*) yang berkelanjutan di sepanjang proyek.
3. Fokus pada pengiriman nilai: Mengutamakan pengiriman perangkat lunak yang berfungsi nyata lebih cepat dibandingkan penyusunan dokumentasi yang berlebihan.
4. Adaptasi terhadap perubahan: Mampu mengakomodasi dan merespons perubahan persyaratan sistem meskipun sudah berada di fase akhir pengembangan.

Manfaat yang didapatkan dari metode ini termasuk peningkatan fleksibilitas, pengurangan risiko kegagalan, dan peningkatan kepuasan pengguna. Namun, tantangan teknis juga dapat muncul, seperti perlunya adaptasi budaya organisasi, tim yang belum terbiasa dengan otonomi tinggi, serta tuntutan kemampuan tim untuk bekerja dalam siklus yang cepat [19][20].

2.5 Metode Scrum

Scrum merupakan salah satu kerangka kerja (*framework*) taktis di dalam pendekatan *Agile* yang digunakan untuk mengelola dan mengembangkan perangkat lunak secara iteratif dan inkremental. *Scrum* berfokus pada kolaborasi tim, adaptasi terhadap dinamika perubahan kebutuhan sistem, serta penyampaian produk secara bertahap melalui siklus pengembangan yang disebut *Sprint* [21].

Dalam *Scrum*, proses pengembangan dibagi ke dalam periode waktu tetap (*time-boxed*) yang disebut *Sprint*, yang umumnya berlangsung antara 1 hingga 4 minggu. Setiap *Sprint* akan menghasilkan peningkatan (*increment*) produk berupa modul perangkat lunak (seperti antarmuka sistem presensi atau cetak nilai akhir) yang dapat diuji dan dievaluasi kelayakannya [21][22].

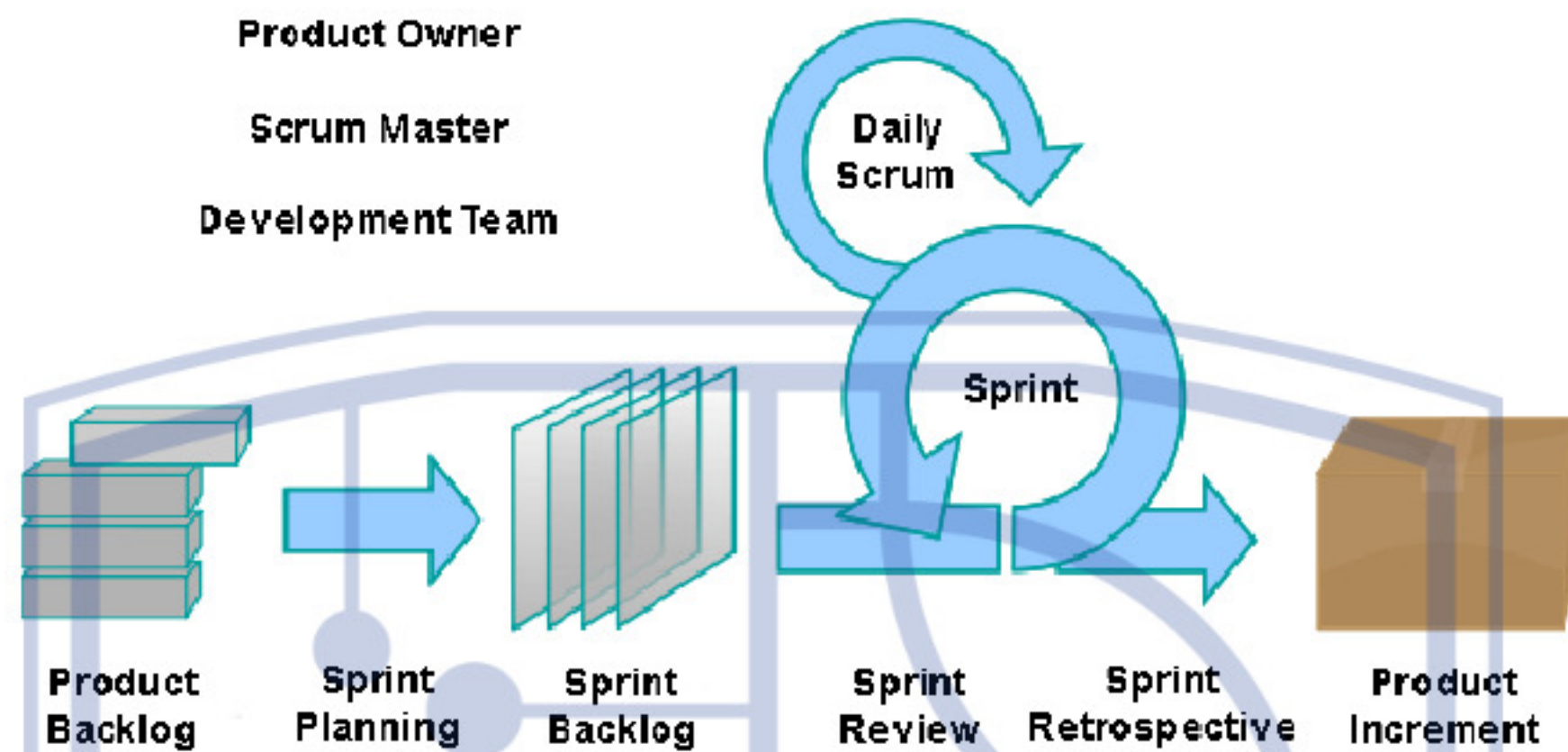
Untuk menjalankan siklus kerja secara efisien, kerangka *Scrum* membagi tanggung jawab ke dalam tiga peran utama yang sangat relevan dan terukur untuk diterapkan pada tim pengembang berskala kecil [21]:

1. **Pemilik Produk (*Product Owner*):** Bertanggung jawab dalam menentukan spesifikasi kebutuhan sistem dan mengelola daftar prioritas pekerjaan (*Product Backlog*). *Product Owner* memastikan fitur yang dikembangkan memiliki nilai guna dan prioritas penyelesaian yang jelas sesuai dengan regulasi yang dibutuhkan.
2. ***Scrum Master*:** Berperan sebagai fasilitator independen yang memastikan seluruh proses berjalan sesuai aturan kerangka kerja *Scrum*, serta membantu tim mengatasi hambatan (*impediment*) teknis selama masa pengerjaan berlangsung.
3. **Tim Pengembang (*Development Team*):** Kelompok pelaksana lintas fungsi (terdiri dari perancang sistem, analis basis data, dan pemrogram) yang bertugas merancang, membangun, dan menguji fungsionalitas website sesuai dengan *backlog* yang telah disepakati.

Dalam kerangka kerja *Scrum*, transparansi dan pengelolaan beban kerja didukung oleh tiga artefak utama sebagai bentuk rekam jejak sistematis [22]:

1. ***Product Backlog*:** Daftar prioritas menyeluruh yang berisi rincian fitur, fungsi, dan kebutuhan yang diperlukan oleh sistem yang dibuat. Daftar ini bersifat dinamis dan dapat terus diperbarui.
2. ***Sprint Backlog*:** Kumpulan item pekerjaan dari *Product Backlog* yang ditarik dan disepakati oleh tim untuk diselesaikan dalam satu siklus *Sprint* tertentu, lengkap dengan rencana implementasinya.

3. *Increment*: Hasil nyata dari modul perangkat lunak yang telah selesai dibangun dan diuji kelayakannya pada akhir sebuah *Sprint*. *Increment* harus dalam kondisi fungsional utuh dan siap diintegrasikan.



Gambar 2. 2 Tahapan Metode *Scrum* [21]

Proses pengembangan perangkat lunak menggunakan metode *Scrum* dijalankan melalui serangkaian siklus kegiatan yang terstruktur dan memiliki batasan waktu. Secara kronologis, tahapan pelaksanaannya diuraikan ke dalam 6 langkah operasional berikut [21][22]:

1. *Product Backlog* (Inisiasi): Tahap awal di mana *Product Owner* bersama tim mendefinisikan dan mengumpulkan seluruh kebutuhan fungsional sistem, kemudian menyusunnya menjadi daftar prioritas pekerjaan.
2. *Sprint Planning* (Perencanaan Sprint): Rapat perencanaan yang dilakukan pada awal setiap siklus. Tim pengembang memilih item prioritas tertinggi dari *Product Backlog* untuk ditarik ke dalam *Sprint Backlog*, serta merumuskan strategi teknis dan pembagian beban kerja.
3. *Sprint Execution* (Pelaksanaan Sprint): Tahap konstruksi fisik di mana tim pengembang mulai menerjemahkan rancangan dari *Sprint Backlog* menjadi instruksi kode program (*coding*). Tahap ini adalah inti dari pembuatan perangkat lunak.
4. *Daily Scrum* (Rapat Harian): Pertemuan sinkronisasi singkat yang diadakan setiap hari selama masa eksekusi *Sprint*. Setiap anggota tim melaporkan progres pekerjaan harian, rencana selanjutnya, dan mengidentifikasi kendala teknis.
5. *Sprint Review* (Tinjauan Sprint): Evaluasi fungsional yang dilaksanakan pada akhir *Sprint*. Tim pengembang mendemonstrasikan hasil modul perangkat lunak yang telah

berfungsi (*Increment*) kepada *Product Owner* dan pemangku kepentingan untuk mendapatkan umpan balik (*feedback*).

6. *Sprint Retrospective* (Retrospektif *Sprint*): Pertemuan internal penutup siklus bagi tim pengembang untuk mengevaluasi efektivitas alur kerja, komunikasi antar anggota, serta alat pemrograman yang digunakan guna merencanakan perbaikan pada *Sprint* berikutnya.

2.6 Analisis dan Perancangan Sistem

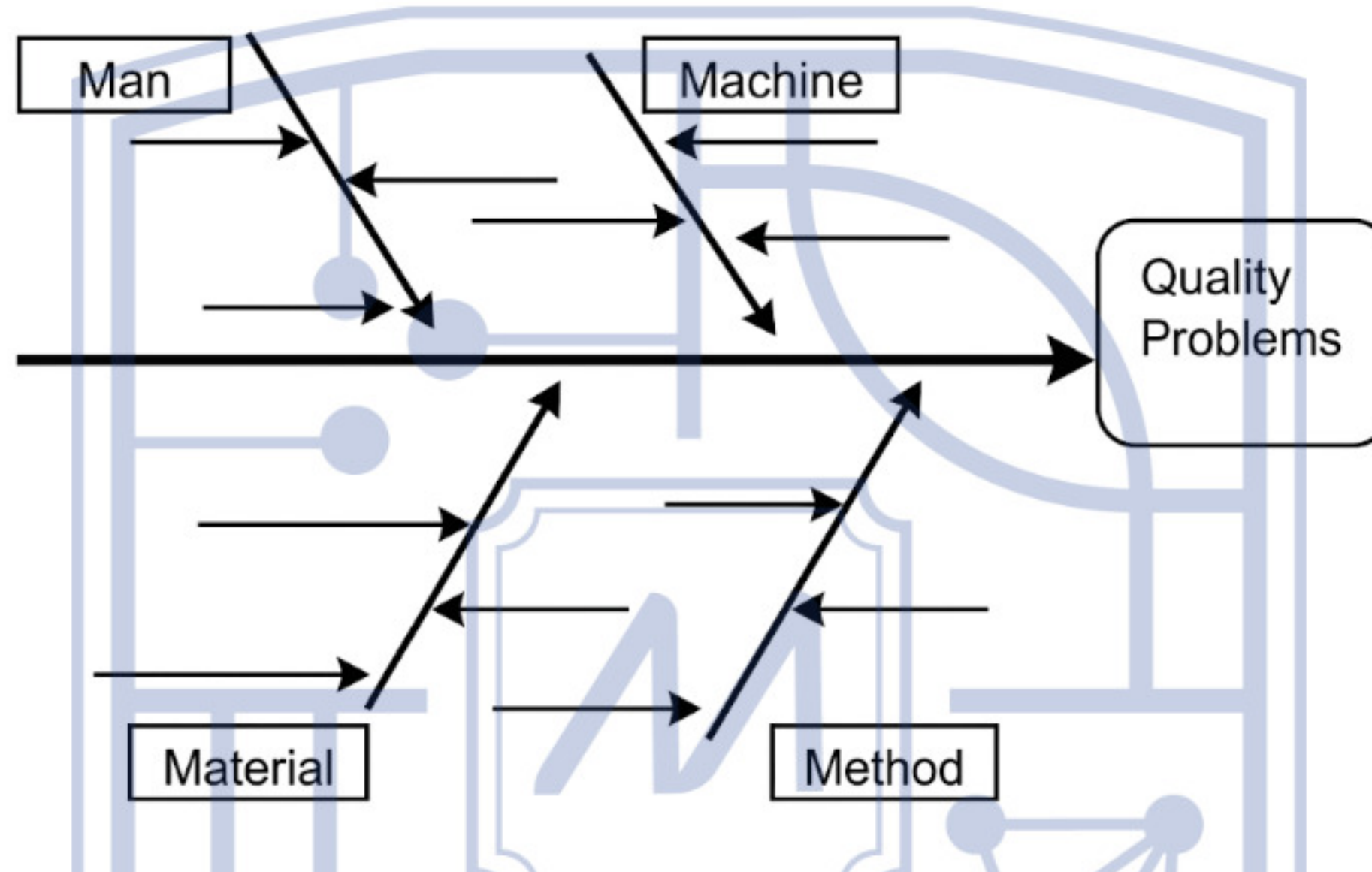
2.6.1 Metode Analisis PIECES

Metode *PIECES* merupakan sebuah kerangka kerja (*framework*) evaluasi yang dirancang untuk mengidentifikasi dan mengklasifikasikan permasalahan, peluang, serta arahan pengembangan teknis pada sebuah sistem informasi. Secara teoretis, kerangka kerja ini membedah kelayakan, keamanan, dan kinerja operasional sistem melalui enam domain utama [23], yaitu:

1. P - *Performance* (Kinerja): Mengevaluasi kemampuan sistem dalam memproses volume transaksi atau merespons instruksi pengguna dalam satuan waktu tertentu (*throughput* dan *response time*).
2. I - *Information* (Informasi): Menilai tingkat akurasi, kelengkapan, dan ketepatan waktu dari data keluaran (*output*) yang dihasilkan oleh sistem untuk mendukung pengambilan keputusan operasional.
3. E - *Economy* (Ekonomi): Menganalisis rasio efektivitas biaya operasional yang dikeluarkan dibandingkan dengan nilai manfaat atau penghematan komputasi yang diperoleh dari penerapan sistem.
4. C - *Control* (Kendali): Memastikan adanya mekanisme keamanan, validasi, dan pembatasan hak akses untuk melindungi integritas data dari manipulasi pihak yang tidak berwenang maupun galat sistem.
5. E - *Efficiency* (Efisiensi): Mengukur tingkat optimalisasi penggunaan sumber daya (kapasitas peladen, ruang memori, maupun waktu tenaga manusia) dalam menghasilkan sebuah keluaran.
6. S - *Service* (Layanan): Meninjau kualitas keandalan, fleksibilitas, dan kemudahan antarmuka sistem (*user-friendly*) saat dioperasikan oleh pengguna akhir.

2.6.2 Fishbone Diagram

Fishbone Diagram (Diagram Tulang Ikan) atau *Diagram Ishikawa* adalah instrumen kendali mutu visual yang dirancang untuk mengidentifikasi, mengeksplorasi, dan memetakan akar penyebab historis dari sebuah permasalahan spesifik secara terstruktur. Visualisasi diagram ini memisahkan akibat atau masalah utama yang diletakkan pada bagian kepala ikan, dengan faktor-faktor penyebabnya yang diilustrasikan sebagai tulang-tulang yang menyirip pada bagian badannya [24].



Gambar 2. 3 Ilustrasi *Fishbone Diagram* [25]

Secara konseptual, pemetaan akar masalah pada evaluasi dan perancangan sistem informasi umumnya diklasifikasikan ke dalam empat kategori kausalitas utama (konsep 4M) [25], yaitu:

1. Manusia (*Man*): Kategori yang menganalisis faktor kelalaian, keterbatasan pemahaman, tingkat kelelahan, atau kesalahan operasional (*human error*) yang dilakukan oleh pengguna akhir maupun Koordinator pengelola sistem.
2. Mesin (*Machine*): Kategori yang meninjau kegagalan fungsi dari instrumen teknologi, seperti limitasi kapasitas perangkat keras (*hardware*), ketidakstabilan jaringan peladen (*server*), atau bug pada perangkat lunak pendukung.
3. Metode (*Method*): Kategori yang mengevaluasi ketidakefektifan prosedur standar operasional, alur kerja komputasi, atau aturan birokrasi yang memicu terjadinya hambatan maupun anomali pada siklus sistem.
4. Material (*Material*): Kategori yang mengidentifikasi permasalahan pada kualitas komponen dasar. Dalam konteks sistem informasi, material merujuk pada ketidaklengkapan, redudansi, atau kerusakan arsitektur data (*input*) yang diolah.

Pemetaan visual yang komprehensif ini berfungsi untuk mencegah pengembang sistem mengambil kesimpulan yang prematur, sehingga solusi perangkat lunak yang dirancang nantinya benar-benar mengeliminasi akar permasalahan, bukan sekadar mengatasi gejalanya saja [24].

2.6.3 Use Case Diagram

Unified Modeling Language (UML) adalah bahasa pemodelan visual berstandar global yang digunakan untuk menspesifikasikan, menggambarkan, membangun, dan mendokumentasikan artefak dari sebuah sistem perangkat lunak yang berorientasi objek (*Object-Oriented*). UML memungkinkan pengembang untuk membuat cetak biru (*blueprint*) sistem yang mencakup aspek struktur statis maupun perilaku dinamis secara seragam [26].

Salah satu instrumen fundamental dalam pemodelan UML adalah *Use Case Diagram*. Diagram ini bersifat statis dan digunakan secara spesifik untuk memodelkan fungsionalitas sistem dari sudut pandang interaksi pengguna luar. *Use Case Diagram* memberikan gambaran tingkat tinggi mengenai "siapa" yang menggunakan sistem dan "apa saja" yang dapat dilakukannya, tanpa menjelaskan bagaimana komputasi teknis tersebut dieksekusi di balik layar [27].

Secara umum, anatomi penyusunan *Use Case Diagram* direpresentasikan melalui empat komponen struktural utama [26][27], yaitu:

1. **Aktor (*Actor*):** Disimbolkan dengan figur manusia (*stickman*), merepresentasikan peran entitas di luar sistem yang berinteraksi langsung dengan sistem. Aktor dapat berupa manusia, perangkat keras, maupun sistem eksternal lain.
2. **Sistem (*System Boundary*):** Disimbolkan dengan batas visual berupa bingkai persegi yang memisahkan antara aktor eksternal dengan kumpulan fungsionalitas internal yang disediakan oleh perangkat lunak.
3. **Kasus Penggunaan (*Use Case*):** Disimbolkan dengan bentuk elips (*oval*), mendeskripsikan unit fungsionalitas, layanan, atau tujuan spesifik yang disediakan oleh sistem untuk merespons instruksi.
4. **Relasi (*Relationship*):** Garis penghubung yang mendefinisikan bentuk komunikasi antar-komponen. Selain relasi asosiasi standar (garis lurus antara aktor dan *use case*), pemodelan logika fungsionalitas yang kompleks difasilitasi oleh dua relasi dependensi khusus, yaitu:

- a. Relasi <<include>>: Relasi dependensi wajib yang menunjukkan bahwa sebuah use case dasar harus memanggil dan mengeksekusi use case lainnya agar fungsionalitasnya dapat diselesaikan.
- b. Relasi <<extend>>: Relasi dependensi opsional yang menunjukkan fungsionalitas tambahan, di mana use case perluasan tersebut hanya akan dieksekusi jika syarat atau kondisi logika pada use case dasar terpenuhi.

2.6.4 Activity Diagram

Activity Diagram adalah salah satu instrumen pemodelan perilaku (*behavioral diagram*) di dalam kerangka *Unified Modeling Language (UML)* yang merepresentasikan alur kerja dinamis (*workflow*) atau urutan instruksi operasional di dalam sebuah sistem. Berbeda dengan *Use Case Diagram* yang mendeskripsikan sistem secara statis dariacamata pengguna, *Activity Diagram* membedah pergerakan data dan logika dari satu aktivitas komputasi ke aktivitas lainnya secara sekuensial [26].

Diagram ini sangat krusial untuk memodelkan struktur logika percabangan, *algoritma*, serta proses bisnis yang kompleks di balik layar (*backend*) [28]. Secara teoretis, struktur *Activity Diagram* dibangun menggunakan lima simbol arsitektural utama seperti pada Tabel 2.2, yaitu:

Tabel 2. 1 Simbol dalam *Activity Diagram* [26]

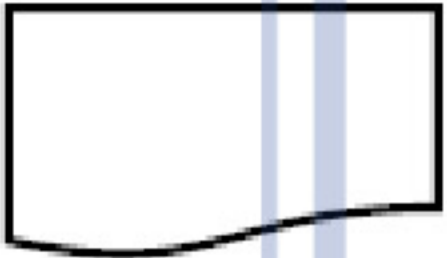
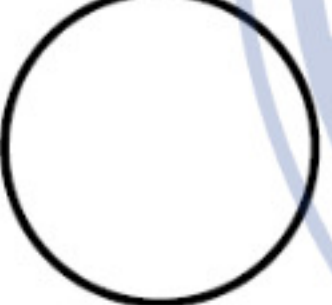
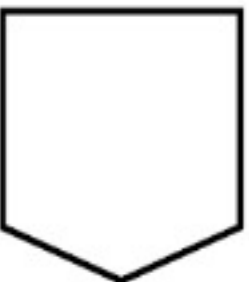
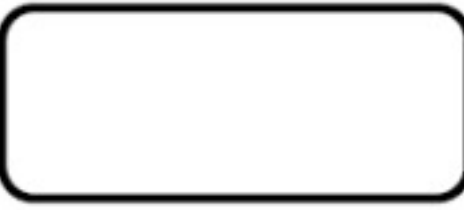
Simbol	Keterangan
Titik Awal (<i>Initial Node</i>) 	Menandai titik awal dimulainya suatu aliran proses dalam sistem.
Aktivitas (<i>Activity / Action</i>) 	Merepresentasikan unit instruksi atau tindakan yang dilakukan aktor atau sistem.
Percabangan (<i>Decision</i>) 	Menunjukkan simpul pilihan jalur berdasarkan kondisi tertentu (percabangan).
Penggabungan (<i>Join</i>) 	Menandakan sinkronisasi aliran paralel atau penggabungan beberapa alur kerja.

Titik Akhir (<i>Final Node</i>) 	Menandai akhir dari seluruh rangkaian aktivitas dalam satu proses.
--	--

2.6.5 Flowchart

Flowchart adalah representasi visual yang menggunakan simbol-simbol grafis untuk menggambarkan alur kerja atau proses. *Flowchart* digunakan untuk memodelkan urutan langkah-langkah dalam suatu sistem atau aktivitas dengan jelas dan terstruktur [18]. Terdapat beberapa simbol yang sering digunakan dalam *flowchart*, seperti pada Tabel 2.3 berikut :

Tabel 2. 2 Simbol dalam *Flowchart* [29]

Simbol	Arti	Keterangan
	<i>Document</i>	Simbol ini menyatakan bahwa input berasal dari dokumen dalam bentuk fisik, atau <i>output</i> yang perlu dicetak.
	<i>Flow</i>	Simbol ini digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain.
	<i>On-Page Reference</i>	Simbol yang menyatakan keluar-masuk atau penyambungan proses dalam lembar kerja yang sama.
	<i>Off-Page Reference</i>	Simbol yang menyatakan keluar-masuk atau penyambungan proses dalam lembar kerja yang berbeda.
	<i>Terminator</i>	Simbol yang menyatakan awal atau akhir dari sebuah kegiatan.
	<i>Process</i>	Simbol ini menyatakan suatu proses yang dilakukan oleh komputer.

	<i>Decision</i>	Simbol ini menunjukkan kondisi keputusan harus dibuat.
	<i>Input/Output</i>	Simbol yang menyatakan proses <i>input</i> atau <i>output</i> tanpa bergantung pada peralatan.
	<i>Manual Operation</i>	Simbol yang menyatakan proses dilakukan secara manual atau tidak dilakukan oleh komputer.
	<i>Predefine Proses</i>	Simbol yang menyatakan pelaksanaan suatu bagian (sub-program) atau prosedur.
	<i>Display</i>	Simbol yang menyatakan peralatan output yang digunakan.
	<i>Preparation</i>	Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal.

2.7 Pengujian Sistem

Pengujian adalah serangkaian aktivitas yang direncanakan secara sistematis untuk mengevaluasi sebuah sistem yang telah dibangun. Pengujian menggunakan metode *Black Box testing* bertujuan untuk mengetahui kelemahan sistem agar hasil akhirnya sesuai dengan rancangan awal, sehingga meminimalisasi kekurangan dan kesalahan pada aplikasi sebelum digunakan oleh pengguna akhir [30].

2.7.1 Black Box Testing

Metode *Black Box testing* merupakan pendekatan yang berfokus pada pengujian fungsional setiap fitur di dalam perangkat lunak. Metode ini memungkinkan penguji untuk mengevaluasi fungsionalitas sistem melalui berbagai skenario masukan (*input*) tertentu, guna memastikan apakah program yang dikembangkan telah memenuhi spesifikasi kebutuhan yang dirancang [31].

Fokus utama pada *Black Box* testing adalah verifikasi fungsionalitas sistem tanpa perlu meninjau arsitektur maupun struktur kode di dalamnya. Penguji menggunakan berbagai skenario untuk mengeksekusi beragam masukan dari pengguna guna mengidentifikasi galat atau perilaku sistem yang tidak diinginkan [31]. Secara terstruktur, tahapan dalam melakukan *Black Box testing* meliputi [32]:

1. Membuat *test case* (skenario uji) untuk menguji berbagai fungsi di dalam aplikasi.
2. Mengembangkan *test case* untuk mengevaluasi sejauh mana alur kerja sebuah fungsi telah selaras dengan kebutuhan dan ekspektasi pengguna.
3. Mengidentifikasi *bug* atau kesalahan operasional berdasarkan antarmuka aplikasi.

2.7.2 System Usability Scale (SUS)

Pengujian usabilitas (*usability testing*) merupakan tahapan esensial untuk mengukur sejauh mana sebuah sistem informasi dapat digunakan oleh pengguna secara efektif, efisien, dan memuaskan. Dalam proses pengembangan platform akademik digital, *System Usability Scale (SUS)* merupakan salah satu instrumen pengujian kuesioner yang paling komprehensif, valid, dan sering diimplementasikan [33].

Metode SUS dipilih karena kesederhanaannya dalam memberikan hasil metrik yang andal dan cepat (*quick and dirty*) melalui survei berskala *Likert*. Pendekatan ini sangat relevan diterapkan pada sistem informasi akademik untuk memastikan bahwa transisi dari proses administrasi konvensional (seperti rekapitulasi nilai dan cetak rapor manual) ke dalam sistem berbasis web tidak membebani pengguna akhir, yakni guru dan Koordinator sekolah [34].

2.7.2.1 Struktur Pertanyaan SUS

Instrumen pengujian SUS diukur menggunakan 10 butir pernyataan baku yang saling bersilangan. Struktur kuesioner ini terdiri dari 5 pernyataan bernada positif (terdapat pada nomor ganjil) dan 5 pernyataan bernada negatif (terdapat pada nomor genap). Pengguna akan memberikan penilaian dengan Skala Likert 1 hingga 5, yang merepresentasikan tingkat kepuasan dari "Sangat Tidak Setuju" hingga "Sangat Setuju" [35].

Tabel 2. 3 Skala Likert [35]

Jawaban	Skor
Sangat Tidak Setuju	1
Tidak Setuju	2

Netral	3
Setuju	4
Sangat Setuju	5

Adapun kesepuluh pernyataan baku SUS yang digunakan untuk mengukur usabilitas perangkat lunak adalah sebagai berikut [33]:

1. Saya berpikir akan sering menggunakan sistem ini.
2. Saya merasa sistem ini terlalu rumit padahal dapat dibuat lebih sederhana.
3. Saya merasa sistem ini mudah untuk digunakan.
4. Saya membutuhkan bantuan dari orang teknis untuk dapat menggunakan sistem ini.
5. Saya merasa fitur-fitur pada sistem ini berjalan dengan semestinya.
6. Saya merasa ada banyak hal yang tidak konsisten pada sistem ini.
7. Saya merasa mayoritas orang akan dapat mempelajari cara menggunakan sistem ini dengan cepat.
8. Saya merasa sistem ini sangat membingungkan untuk digunakan.
9. Saya merasa sangat yakin dapat menggunakan sistem ini.
10. Saya harus belajar banyak hal terlebih dahulu sebelum dapat menggunakan sistem ini.

2.7.2.2 Aturan Perhitungan Skor SUS

Untuk menentukan tingkat kemudahan sistem, SUS memiliki formula komputasi matematis yang terstandarisasi. Aturan perhitungan untuk setiap respons responden adalah sebagai berikut [36]:

1. Untuk setiap pertanyaan bernomor ganjil (pernyataan positif), skor respons pengguna dikurangi 1 (Skor - 1).
2. Untuk setiap pertanyaan bernomor genap (pernyataan negatif), nilai 5 dikurangi dengan skor respons pengguna (5 - Skor).
3. Seluruh hasil konversi dari 10 pertanyaan tersebut kemudian dijumlahkan.
4. Total penjumlahan akhir dikalikan dengan konstanta 2,5 untuk menghasilkan skor berskala 0 hingga 100.

Secara matematis, persamaan untuk mendapatkan skor akhir dari satu responden dapat dirumuskan sebagai berikut:

$$Skor\ SUS = \left(\sum_{i=1}^5 (R_{ganjil} - 1) + \sum_{i=1}^5 (5 - R_{genap}) \right) \times 2,5$$

(Keterangan: *R* adalah nilai bobot respons dari pengguna)

2.7.2.3 Interpretasi Penilaian (Acceptability Range)

Skor akhir rentang 0 hingga 100 yang dihasilkan oleh SUS merupakan persentil peringkat, bukan sebuah persentase mutlak. Untuk menentukan apakah sebuah sistem informasi akademik layak diimplementasikan, skor akhir diukur menggunakan pedoman rentang penerimaan (*Acceptability Range*) [34].

Tabel 2. 4 Tabel Acceptability Range [33][36]

Grade Scale	Nilai Skor	Adjective Ratings
A	91-100	Best Imaginable
B	81-90	Excellent
C	71-80	Good
D	61-70	Ok
F	0-60	Poor / Worst Imaginable

Berdasarkan standar industri usability, skor rata-rata (*benchmark*) kelayakan sebuah sistem adalah 68. Jika skor rata-rata SUS dari seluruh guru dan Koordinator berada di angka 68 atau lebih, maka sistem informasi tersebut dikategorikan *Acceptable* (Dapat Diterima) dengan kategori Grade minimum C, yang menandakan desain antarmuka sudah ramah pengguna (*user-friendly*). Sebaliknya, jika skor berada di bawah 68 (*Marginal* atau *Not Acceptable*), maka antarmuka sistem memerlukan evaluasi dan perbaikan desain lebih lanjut [33][36].

2.8 SMA W. R. Supratman 2 Medan

SMA W. R. Supratman 2 Medan adalah satuan pendidikan formal tingkat menengah atas berstatus swasta yang bernaung di bawah Yayasan Perguruan Wage Rudolf Supratman. Institusi ini berfokus pada peningkatan kualitas akademik serta pembentukan karakter siswa, dan secara spesifik menjadi objek studi kasus dalam penelitian dan implementasi perancangan sistem informasi akademik ini.



Gambar 2. 4 SMA W. R. Supratman 2 Medan



Gambar 2. 5 Struktur Organisasi Sekolah

Berdasarkan arsip pangkalan data referensi Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi (Kemendikbudristek) Republik Indonesia [37], identitas operasional dari instansi ini dijabarkan sebagai berikut:

1. Nama Satuan Pendidikan: SMA Swasta W. R. Supratman 2 Medan
2. Nomor Pokok Sekolah Nasional (NPSN): 10210832
3. Status Sekolah: Swasta

4. Bentuk Pendidikan: SMA
5. SK Pendirian Sekolah: 63 (Diterbitkan pada 28 Juni 1963)
6. Nomor SK Izin Operasional: 421.3/815
7. Alamat Lengkap: Jl. Brigjen Zein Hamid No. 33, Kelurahan Titi Kuning, Kecamatan Medan Johor, Kota Medan, Provinsi Sumatera Utara.

Dalam menjalankan fungsi operasional akademiknya, struktur birokrasi instansi ini melibatkan kolaborasi antara Koordinator (Wakil Kepala Sekolah), guru mata pelajaran, dan wali kelas. Proses administrasi yang berjalan mencakup pencatatan kehadiran siswa, rekapitulasi nilai evaluasi belajar dari masing-masing guru, hingga penyusunan laporan hasil akhir pembelajaran berupa pencetakan buku rapor oleh wali kelas. Pemahaman terhadap alur administrasi dan struktur entitas inilah yang menjadi landasan utama dalam memodelkan sistem informasi yang terintegrasi, guna memastikan perangkat lunak yang dibangun dapat memfasilitasi kebutuhan rekapitulasi dan pencetakan laporan akhir secara efisien dan akurat.

2.9 Penelitian Terdahulu

Sebagai bahan perbandingan dan penentuan celah penelitian (*research gap*), berikut diuraikan beberapa penelitian terdahulu yang membahas perancangan sistem informasi akademik berbasis web.

Penelitian pertama dilakukan oleh Setiawan, dkk. [5] dengan judul "Sistem Informasi Akademik Berbasis Web (Studi Kasus SMK Asshiddiqiyah Garut)". Penelitian ini menggunakan metode pengembangan *Rational Unified Process* (RUP) dengan empat tahapan, yaitu *inception*, *elaboration*, *construction*, dan *transition*. Sistem yang dihasilkan dilengkapi fitur pengelolaan nilai, absensi, jadwal pelajaran, tugas guru, serta *email gateway* yang secara otomatis mengirimkan ringkasan hasil pembelajaran kepada siswa dan orang tua. Pengujian dilakukan melalui *blackbox testing* dan kuesioner kepuasan pengguna berskala Likert, dengan hasil tingkat kepuasan siswa sebesar 95,7% (kategori sangat setuju) dan guru/staf sebesar 78,7% (kategori setuju). Meskipun sistem ini unggul dalam hal distribusi informasi otomatis kepada orang tua, penelitian tersebut belum menyediakan fitur pencetakan rapor dalam format baku sekolah, dan belum melibatkan peran khusus wali kelas yang terpisah dari guru mata pelajaran.

Penelitian kedua dilakukan oleh Sadewa, dkk. [6] dengan judul "Sistem Informasi Akademik Berbasis Web Pada SMK Al-Islam Surakarta". Penelitian ini menggunakan metode analisis PIECES untuk mengidentifikasi permasalahan sistem yang berjalan,

dilanjutkan dengan metode pengembangan *Waterfall*. Sistem yang dihasilkan mengintegrasikan proses pendaftaran siswa baru (PPDB), pencatatan absensi, pelaporan nilai, dan penyampaian informasi akademik dalam satu platform berbasis web dengan tiga aktor, yaitu admin, guru, dan siswa. Pengujian menggunakan *blackbox testing* terhadap tiga fitur utama, yakni absensi, pembayaran pendaftaran, dan pelaporan nilai, dengan hasil "Sesuai" pada seluruh skenario uji. Namun, sistem ini masih berfokus pada pengelolaan data internal sekolah dan belum dilengkapi fitur cetak rapor maupun rekapitulasi absensi bulanan otomatis per kelas.

Penelitian ketiga dilakukan oleh Amarulla, dkk. [7] dengan judul "Perancangan Sistem Informasi Akademik Pada SD Xaverius 1 Kota Jambi Berbasis Web". Penelitian ini menggunakan metode pengembangan *Waterfall* dengan pemodelan *Unified Modeling Language* (UML) berupa *use case diagram*, *activity diagram*, dan *class diagram*. Sistem yang dihasilkan mencakup pengelolaan data siswa, guru, mata pelajaran, kelas, tahun ajaran, dan nilai siswa, dengan tiga aktor yaitu admin, guru, dan siswa, yang bertujuan mempermudah tata usaha dalam pengolahan data serta penyampaian informasi akademik kepada orang tua dan siswa. Kekurangan pada penelitian ini terletak pada belum tersedianya fitur pencetakan laporan hasil belajar (rapor), sehingga proses pelaporan akhir akademik siswa masih harus dilakukan secara terpisah di luar sistem.

Berdasarkan uraian ketiga penelitian terdahulu tersebut, dapat disimpulkan bahwa masing-masing telah berhasil mendigitalisasi sebagian proses administrasi akademik sekolah, namun belum satu pun yang menyediakan fitur pencetakan rapor dalam format yang dapat disesuaikan dengan kebutuhan sekolah, serta belum mengintegrasikan secara menyeluruh antara data siswa, nilai, absensi, kelas, dan jadwal pelajaran dalam satu sistem terpadu dengan pembagian peran koordinator dan wali kelas secara eksplisit. Celah inilah yang menjadi dasar penelitian ini, yaitu merancang sebuah sistem informasi akademik berbasis web pada SMA Wage Rudolf Supratman 2 Medan yang mengintegrasikan seluruh komponen tersebut sekaligus menyediakan fitur cetak rapor sesuai format yang ditetapkan sekolah.