

BAB II

KAJIAN LITERATUR

2.1 Sistem Informasi

Sistem informasi merupakan suatu sistem yang terdiri dari berbagai komponen yang saling berinteraksi untuk mengumpulkan, mengolah, menyimpan, dan mendistribusikan informasi guna mendukung proses pengambilan keputusan dalam suatu organisasi. Sistem informasi berperan dalam mengubah data mentah menjadi informasi yang memiliki nilai guna bagi pengguna dalam menjalankan aktivitas operasional maupun manajerial. Sistem informasi merupakan kombinasi antara teknologi informasi dan aktivitas manusia yang menggunakan teknologi tersebut untuk mendukung operasi dan manajemen organisasi [4].

Sistem informasi membantu organisasi dalam mengelola data secara terstruktur sehingga informasi yang dihasilkan dapat digunakan secara efektif untuk pengambilan keputusan. Dalam lingkungan pendidikan, sistem informasi banyak digunakan untuk mengelola berbagai aktivitas akademik seperti pengolahan data siswa, pengelolaan nilai, absensi, jadwal pelajaran, serta laporan akademik. Penerapan sistem informasi manajemen di sekolah mampu meningkatkan efisiensi administrasi serta meminimalkan kesalahan pencatatan data karena seluruh informasi tersimpan dalam satu sistem yang terintegrasi [5].

Penerapan sistem informasi juga mempermudah penyimpanan, pencarian, serta penyajian informasi karena seluruh data tersimpan dalam suatu sistem yang terintegrasi.

2.1.1 Komponen Sistem Informasi

Sistem informasi terdiri dari beberapa komponen utama yang saling berinteraksi untuk menghasilkan informasi yang bermanfaat. Komponen-komponen tersebut meliputi perangkat keras, perangkat lunak, basis data, prosedur, serta sumber daya manusia. Seluruh komponen tersebut harus bekerja secara terintegrasi agar sistem informasi dapat menghasilkan informasi yang akurat, relevan, dan tepat waktu bagi organisasi [6].

Adapun komponen-komponen sistem informasi dijelaskan sebagai berikut:

1. Perangkat Keras (*Hardware*)

Perangkat keras merupakan komponen fisik yang digunakan dalam sistem informasi, seperti komputer, *server*, perangkat jaringan, serta perangkat *input* dan *output* yang berfungsi untuk mendukung proses pengolahan data.

2. Perangkat Lunak (*Software*)

Perangkat lunak adalah sekumpulan program atau instruksi yang digunakan untuk mengelola, memproses, dan mengolah data menjadi informasi yang berguna. Contohnya meliputi sistem operasi, aplikasi *database*, dan aplikasi berbasis *web*.

3. Basis Data (*Database*)

Basis data merupakan kumpulan data yang tersusun secara sistematis dan terorganisir sehingga dapat diakses, dikelola, serta diperbarui dengan mudah sesuai kebutuhan sistem informasi.

4. Prosedur (*Procedure*)

Prosedur merupakan serangkaian aturan atau langkah-langkah yang digunakan dalam menjalankan sistem informasi agar dapat beroperasi secara efektif dan sesuai dengan tujuan yang telah ditetapkan.

5. Sumber Daya Manusia (*Brainware*)

Sumber daya manusia atau *brainware* adalah individu yang terlibat dalam penggunaan dan pengelolaan sistem informasi, seperti administrator, operator, serta pengguna akhir yang berperan dalam menjalankan sistem tersebut.

Kelima komponen tersebut ersebut saling melengkapi dalam mendukung proses pengelolaan data sehingga sistem informasi dapat menghasilkan informasi yang akurat, relevan, dan mudah dimanfaatkan oleh pengguna.

2.1.2 Jenis-Jenis Sistem Informasi

Sistem informasi dalam organisasi diklasifikasikan berdasarkan fungsi dan tingkat penggunaannya dalam mendukung aktivitas operasional, manajerial, hingga strategis. Setiap jenis sistem informasi memiliki karakteristik dan tujuan yang berbeda, namun saling terintegrasi dalam mendukung proses bisnis dan pengambilan keputusan berbasis data. Seiring berkembangnya teknologi informasi, berbagai jenis sistem informasi juga mengalami perkembangan, termasuk pemanfaatan teknologi berbasis web untuk meningkatkan kemudahan akses, integrasi data, dan efisiensi pengelolaan informasi [7][8].

Sistem informasi dapat diklasifikasikan ke dalam berbagai jenis, di antaranya Transaction Processing System (TPS), Management Information System (MIS), dan Decision Support System (DSS). Salah satu penerapan Management Information System (MIS) adalah sistem informasi akademik yang digunakan untuk mengelola data siswa, nilai, presensi, dan informasi akademik lainnya di lingkungan pendidikan. Sistem ini menghasilkan informasi akademik serta data kedisiplinan siswa yang dibutuhkan oleh

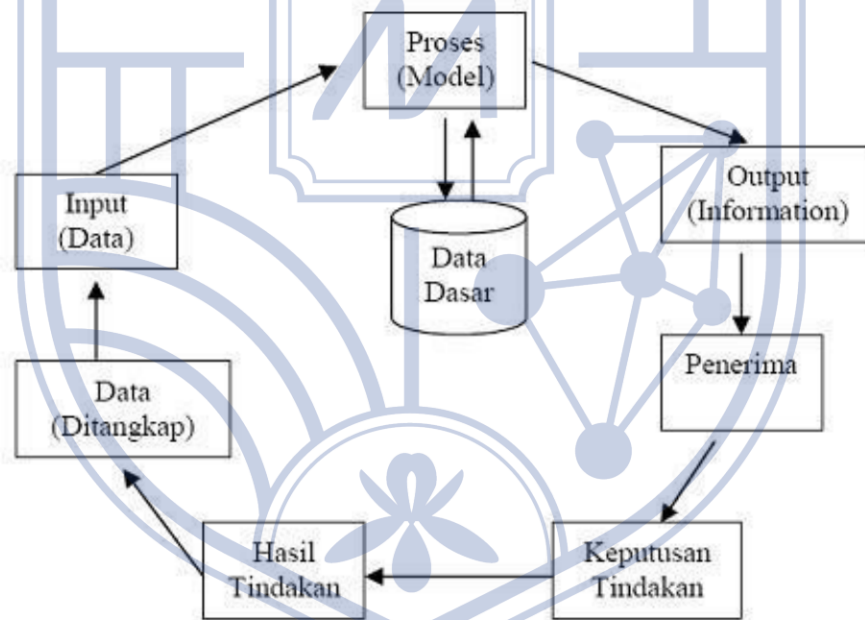
admin, guru, dan siswa untuk mendukung pengambilan keputusan operasional maupun manajerial di tingkat sekolah.

2.1.3 Fungsi Sistem Informasi

Sistem informasi memiliki beberapa fungsi utama dalam mendukung kegiatan organisasi, antara lain [9]:

1. Mengumpulkan data dari berbagai sumber.
2. Mengolah data menjadi informasi yang bermanfaat.
3. Menyimpan data dan informasi secara terstruktur.
4. Menyediakan informasi yang akurat untuk pengambilan keputusan.
5. Meningkatkan efisiensi dan efektivitas proses kerja.

Penerapan sistem informasi membantu organisasi memperoleh informasi secara lebih cepat dan akurat sehingga dapat mendukung proses operasional maupun pengambilan keputusan [9].



Gambar 2.1 Konsep Sistem Informasi

2.2 Sistem Manajemen Akademik

Sistem Manajemen Akademik merupakan implementasi sistem informasi yang digunakan untuk mengelola seluruh aktivitas akademik secara terintegrasi dalam institusi pendidikan. Sistem ini berfungsi untuk mengolah data akademik seperti data siswa, absensi,

nilai, jadwal, serta laporan akademik secara sistematis, sehingga mampu meningkatkan efisiensi dan akurasi pengelolaan data dibandingkan dengan sistem manual.

Penggunaan sistem informasi akademik berbasis web mempermudah proses pengolahan data karena seluruh informasi tersimpan dalam basis data yang terintegrasi. Selain mempercepat akses informasi, sistem ini juga membantu mengurangi kesalahan pencatatan serta mendukung proses administrasi akademik secara lebih efektif [10]. Selain itu, penggunaan sistem terintegrasi juga membantu mengurangi kesalahan pencatatan dan mempercepat proses administrasi di lingkungan pendidikan [11].

Saat ini, sistem manajemen akademik umumnya memanfaatkan basis data terpusat dan teknologi berbasis web sehingga data antarbagian dapat saling terhubung. Kondisi tersebut memungkinkan informasi akademik diperbarui secara real-time dan dapat diakses sesuai hak pengguna [12].

Sistem manajemen akademik memiliki beberapa karakteristik utama sebagai berikut:

1. Terintegrasi dalam satu basis data terpusat
Seluruh data akademik disimpan dalam satu sistem *database* sehingga memudahkan pengelolaan data dan menjaga konsistensi informasi.
2. Mendukung otentikasi dan otorisasi pengguna
Sistem menyediakan fitur keamanan berupa *login* dan pembagian hak akses berdasarkan peran pengguna, seperti admin, guru, dan siswa, sehingga keamanan data lebih terjamin.
3. Memberikan laporan akademik secara otomatis
Sistem mampu menghasilkan laporan nilai, absensi, dan rekapitulasi akademik secara cepat dan akurat tanpa proses manual.
4. Dapat diakses melalui jaringan (*web-based*)
Sistem berbasis *web* memungkinkan pengguna mengakses informasi kapan saja dan di mana saja melalui jaringan internet.
5. Mendukung pengolahan data secara *real-time*
Data yang di-*input* dapat langsung diproses dan ditampilkan sehingga informasi selalu terbaru dan relevan.
6. Meningkatkan efisiensi dan efektivitas proses akademik
Sistem membantu mengurangi pekerjaan manual, mempercepat proses administrasi, serta meningkatkan kualitas layanan pendidikan

2.3 Portal Akademik

Portal akademik berbasis *web* merupakan sistem aplikasi yang dirancang untuk menyediakan layanan informasi akademik secara terintegrasi melalui media internet atau intranet sekolah [13]. Portal ini berfungsi sebagai pintu gerbang (*gateway*) bagi seluruh pengguna dalam mengakses berbagai informasi dan layanan akademik dalam satu platform terpadu.

Secara konseptual, portal akademik merupakan implementasi dari sistem manajemen akademik yang dikembangkan menggunakan teknologi *web* [14]. Teknologi *web* memungkinkan sistem dapat diakses melalui *browser* tanpa memerlukan instalasi perangkat lunak tambahan pada sisi pengguna (*client-side*). Dengan demikian, sistem menjadi lebih fleksibel, mudah diakses, dan efisien dalam pengelolaannya [15].

1. Karakteristik Portal Akademik

Portal akademik memiliki beberapa karakteristik utama, yaitu sistem yang terintegrasi dalam satu basis data, mendukung *multi-user* dengan hak akses berbeda, menyediakan informasi *real-time*, serta dapat diakses melalui jaringan internet [15].

2. Arsitektur Portal Akademik

Secara umum, portal akademik berbasis *web* menggunakan arsitektur tiga lapis (*three-tier architecture*) yang terdiri dari *presentation layer*, *application layer*, dan *data layer* [16]. Arsitektur ini memungkinkan sistem lebih terstruktur dan mendukung keamanan data secara lebih baik.

3. Manfaat Portal Akademik

Penerapan portal akademik berbasis *web* mampu meningkatkan efisiensi administrasi, mengurangi redundansi data, serta mempercepat proses penyampaian informasi akademik.

4. Tantangan Implementasi Portal Akademik

Walaupun memiliki banyak keunggulan, implementasi portal akademik juga memiliki beberapa tantangan, antara lain:

- a. Kesiapan infrastruktur jaringan
- b. Kemampuan pengguna dalam mengoperasikan sistem
- c. Keamanan dan perlindungan data
- d. Konsistensi *input* data oleh pengguna

Oleh karena itu, pengembangan sistem harus memperhatikan aspek keamanan (*authentication* dan *authorization*), kemudahan penggunaan (*usability*), serta integritas data. Integrasi data akademik dan kedisiplinan dalam satu portal memungkinkan informasi siswa

tersimpan secara terpusat sehingga memudahkan proses pengelolaan, pemantauan, dan penyajian informasi. Dengan sistem yang terintegrasi, proses administrasi sekolah dapat dilakukan secara lebih efektif dan efisien.

2.4 SMA Budi Murni 1 Medan

SMA Swasta Katolik Budi Murni 1 Medan merupakan salah satu sekolah menengah atas swasta di Kota Medan yang berdiri sejak tahun 1957 dan berlokasi di Jalan Timor No. 34. Sekolah ini memiliki visi “Peserta didik berdedikasi, bersahabat, dan unggul serta Merdeka berlandaskan nilai kekatolikan” serta menerapkan Kurikulum Nasional dengan pendekatan Deep Learning yang menekankan pengembangan soft skill dan pembentukan karakter[17].

Dalam pengelolaan akademik dan kedisiplinan siswa, SMA Budi Murni 1 Medan telah menerapkan sistem kredit poin sebagai instrumen pencatatan dan evaluasi perilaku siswa. Proses pencatatan kredit poin masih dilakukan secara semi-manual melalui Google Form yang diisi oleh guru. Pencatatan absensi siswa masih menggunakan buku absensi manual yang dikelola oleh guru wali kelas dan guru mata pelajaran. Sementara itu, pengolahan nilai akademik dimulai dari guru mata pelajaran yang menginput nilai ke dalam file Excel menggunakan template yang disediakan sekolah. Template tersebut berisi tabel yang menampung data nilai setiap siswa pada mata pelajaran yang diampu. Selanjutnya, wali kelas merekap seluruh data nilai dari masing-masing guru mata pelajaran, kemudian menyerahkan rekapitulasi tersebut kepada admin atau staf tata usaha untuk dicetak menjadi rapor dalam bentuk fisik. Pengelolaan seluruh data akademik dan kedisiplinan tersebut menjadi tanggung jawab utama guru wali kelas, guru mata pelajaran, serta staf tata usaha sekolah. Kondisi ini menyebabkan proses penginputan data yang berulang, kesulitan pemantauan secara real-time, serta potensi inkonsistensi data antara kredit poin, absensi, dan nilai akademik karena belum adanya sistem informasi yang terintegrasi.

Dengan kondisi tersebut, pengembangan portal akademik berbasis web menjadi sangat relevan untuk mendukung transformasi digital pengelolaan data akademik di SMA Budi Murni 1 Medan.



Gambar 2.2 SMA Budi Murni 1 Medan

2.5 Basis Data

Basis data (*database*) merupakan kumpulan data yang saling berhubungan dan disusun secara sistematis sehingga dapat dikelola, diakses, serta diperbarui secara efisien untuk mendukung kebutuhan informasi suatu organisasi. Dalam sistem informasi berbasis *web*, basis data berperan sebagai pusat penyimpanan data yang memungkinkan integrasi antarmodul sistem dalam satu struktur yang terorganisir.

Perancangan basis data yang baik sangat berpengaruh terhadap kinerja sistem secara keseluruhan. Basis data yang terstruktur mampu meningkatkan konsistensi informasi, mengurangi redundansi data, serta mempermudah proses pencarian dan pengolahan data. Desain basis data yang sistematis mendukung efisiensi pengelolaan data serta meningkatkan kualitas penyajian informasi pada sistem berbasis *web* [18].

Dalam konteks sistem informasi akademik, basis data digunakan untuk menyimpan dan mengelola data siswa, guru, kelas, absensi, nilai, serta data kedisiplinan secara terintegrasi. Penggunaan basis data terpusat memungkinkan pengolahan data menjadi lebih cepat dan akurat, sehingga mendukung proses administrasi akademik yang lebih efektif.

2.5.1 Database Management System (DBMS)

Database Management System (DBMS) merupakan perangkat lunak yang digunakan untuk membuat, mengelola, serta mengontrol akses terhadap basis data. DBMS memungkinkan pengguna melakukan berbagai operasi manipulasi data seperti penyimpanan (*insert*), pembaruan (*update*), penghapusan (*delete*), dan pengambilan data (*select*) secara terstruktur.

Dalam sistem informasi berbasis *web*, DBMS berfungsi untuk menjaga integritas dan keamanan data melalui mekanisme autentikasi dan otorisasi pengguna. Selain itu, DBMS juga menyediakan fitur pengelolaan transaksi, pencadangan (*backup*), dan pemulihan data (*recovery*) guna menjaga keberlangsungan sistem. Penggunaan DBMS yang tepat mampu mendukung pengelolaan data secara terpusat serta meningkatkan stabilitas dan keandalan sistem informasi berbasis *web* [19].

Fungsi utama DBMS meliputi:

1. Pengelolaan struktur tabel dan relasi data
2. Pengaturan hak akses pengguna
3. Menjaga konsistensi dan integritas data
4. Pengamanan data melalui kontrol akses
5. Penyediaan fasilitas pencadangan dan pemulihan data

Dalam pengembangan portal akademik, DBMS berperan dalam mengintegrasikan data kredit poin, absensi, dan nilai ke dalam satu basis data yang terorganisir sehingga proses pelaporan dapat dilakukan secara cepat dan akurat.

2.5.2 Model Basis Data Non Relasional

Dalam penelitian ini, basis data yang digunakan adalah MongoDB, yaitu sistem manajemen basis data NoSQL berbasis dokumen. Berbeda dengan model relasional yang menggunakan tabel dan relasi antar tabel, MongoDB menyimpan data dalam bentuk dokumen fleksibel berbasis JSON (BSON)[20]. Pendekatan ini sangat sesuai dengan teknologi MERN Stack yang digunakan, karena memungkinkan pengembangan yang lebih cepat, skalabilitas yang tinggi, dan fleksibilitas skema data.

Keunggulan penggunaan MongoDB dalam penelitian ini antara lain:

1. Skema data yang fleksibel (*schema-less*), sehingga memudahkan penambahan field baru di kemudian hari.
2. Dukungan penuh terhadap JavaScript dan JSON, sehingga integrasi dengan Node.js dan Express.js menjadi sangat lancar.
3. Kemampuan menangani data yang bersifat hierarkis dan tidak terstruktur dengan baik.
4. Performa yang baik untuk aplikasi web modern.

2.5.3 Peran Basis Data dalam Portal Akademik

Basis data dalam portal akademik berbasis *web* berfungsi sebagai pusat integrasi seluruh data akademik dan kedisiplinan siswa. Sistem basis data terpusat memungkinkan pengelolaan data menjadi lebih terstruktur, aman, dan mudah dikontrol. Perancangan basis data yang baik memungkinkan sistem menghasilkan laporan secara otomatis serta mendukung proses pencarian data secara cepat dan akurat. Sistem informasi berbasis basis data terintegrasi mampu meningkatkan efektivitas pengelolaan data serta mempercepat penyajian informasi kepada pengguna [21].

Dalam penelitian ini, basis data berperan untuk:

1. Mengintegrasikan data kredit poin, absensi, dan nilai
2. Mendukung proses pencarian dan penyaringan data
3. Menyediakan laporan akademik secara otomatis
4. Menjaga keamanan dan konsistensi data siswa
5. Mendukung pengambilan keputusan berbasis data

Dengan demikian, penerapan basis data relasional yang terintegrasi dalam portal akademik menjadi komponen utama dalam meningkatkan efisiensi dan efektivitas pengelolaan data akademik.

2.5.4 Metode Pengembangan Sistem *Waterfall*

Waterfall merupakan metode pengembangan perangkat lunak yang dilakukan secara sistematis dan berurutan, dimana setiap tahapan harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya. Model ini menekankan proses dokumentasi yang jelas pada setiap fase pengembangan. Model *Waterfall* pertama kali diperkenalkan oleh Winston W. Royce dan hingga saat ini masih banyak digunakan dalam pengembangan sistem informasi, khususnya pada proyek yang memiliki kebutuhan sistem yang telah terdefinisi dengan baik. Beberapa penelitian terbaru menunjukkan bahwa model *Waterfall* masih relevan digunakan dalam pengembangan sistem berbasis *web* karena alur kerjanya yang terstruktur dan terdokumentasi dengan baik [22], [23]. Pemilihan metode ini dalam penelitian dilakukan karena sistem informasi akademik yang dikembangkan memiliki kebutuhan yang jelas, ruang lingkup yang terdefinisi, serta proses pengembangan yang membutuhkan dokumentasi sistematis.

Adapun tahapan model *Waterfall* yang diterapkan dalam penelitian ini adalah sebagai berikut:

1. Analisis Kebutuhan (*Requirement Analysis*)

Tahap analisis kebutuhan dilakukan untuk mengidentifikasi kebutuhan sistem melalui observasi, wawancara, dan studi literatur. Pada tahap ini ditentukan kebutuhan fungsional dan nonfungsional sistem, seperti pengelolaan data siswa, absensi, nilai, dan kredit poin. Tahap analisis kebutuhan merupakan fondasi utama dalam model *Waterfall* karena menentukan keberhasilan tahap selanjutnya [23].

2. Perancangan Sistem (*Design*)

Tahap perancangan dilakukan berdasarkan hasil analisis kebutuhan. Perancangan meliputi:

- a. Perancangan *database*
- b. Perancangan antarmuka pengguna (*user interface*)
- c. Perancangan diagram UML (*Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*)
- d. Perancangan arsitektur sistem

Perancangan yang sistematis bertujuan untuk menghasilkan sistem yang terstruktur dan mudah dikembangkan [22].

3. Implementasi (*Development*)

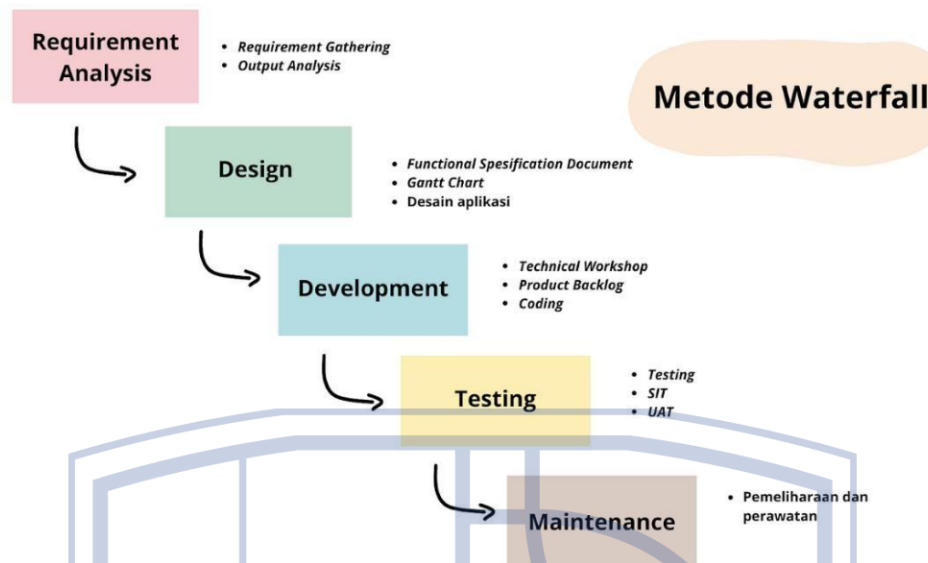
Tahap implementasi merupakan proses penerjemahan desain sistem ke dalam bahasa pemrograman. Sistem dikembangkan berbasis *web* sehingga dapat diakses melalui *browser*. Penerapan model *Waterfall* dalam pengembangan sistem berbasis *web* mampu menghasilkan sistem yang stabil dan sesuai dengan kebutuhan pengguna [24].

4. Pengujian (*Testing*)

Setelah proses implementasi selesai, dilakukan pengujian sistem untuk memastikan seluruh fungsi berjalan sesuai kebutuhan. Pengujian dilakukan menggunakan metode *Black Box Testing* untuk menguji fungsionalitas sistem. Tahap ini bertujuan untuk meminimalkan kesalahan sistem sebelum diimplementasikan kepada pengguna.

5. Pemeliharaan (*Maintenance*)

Tahap pemeliharaan dilakukan setelah sistem diimplementasikan. Pemeliharaan mencakup perbaikan kesalahan (*bug fixing*), pembaruan sistem, dan penyesuaian terhadap kebutuhan baru.



Gambar 2.3 Metode Waterfall

2.6 Tool Pengembangan Sistem

2.6.1 Metode Waterfall

Metode Waterfall merupakan salah satu model pengembangan perangkat lunak yang paling klasik dan banyak digunakan. Model ini bersifat linier dan sequensial, artinya setiap tahapan harus diselesaikan secara berturut-turut dan tidak boleh kembali ke tahapan sebelumnya. Tahapan utama dalam metode Waterfall meliputi analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan[25].

Metode ini dipilih dalam penelitian ini karena kebutuhan sistem sudah cukup jelas dan stabil sejak awal, serta memerlukan dokumentasi yang lengkap dan terstruktur pada setiap tahapannya. Waterfall sangat sesuai untuk pengembangan sistem informasi akademik yang bersifat terstruktur dan membutuhkan proses yang sistematis.

Adapun urutan tahapan Metode Waterfall yang digunakan dalam penelitian ini adalah sebagai berikut:

1. **Analisis Kebutuhan** Melakukan observasi, wawancara, dan analisis dokumen untuk mengidentifikasi kebutuhan fungsional dan nonfungsional sistem.
2. **Perancangan Sistem** Merancang tampilan antarmuka (user interface), basis data (ERD), serta model sistem menggunakan Use Case Diagram dan Activity Diagram.
3. **Implementasi** Membangun sistem secara aktual menggunakan teknologi MERN Stack (MongoDB, Express.js, React.js, dan Node.js).
4. **Pengujian** Melakukan pengujian menggunakan Black Box Testing dan User Acceptance Testing (UAT).

5. **Pemeliharaan** Melakukan perbaikan dan penyempurnaan sistem setelah diimplementasikan (opsional pada tahap skripsi).

Penerapan metode Waterfall memastikan proses pengembangan berjalan secara terstruktur dan terdokumentasi dengan baik.

2.6.2 *Use Case Diagram*

Use Case Diagram merupakan salah satu diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan interaksi antara aktor dengan sistem. Diagram ini berfungsi untuk merepresentasikan kebutuhan fungsional sistem secara visual sehingga memudahkan pemahaman terhadap proses dan layanan yang disediakan oleh sistem. Dalam proses pengembangan sistem informasi, *Use Case Diagram* digunakan pada tahap analisis kebutuhan untuk mengidentifikasi fungsi-fungsi utama sistem serta hubungan antara pengguna dengan sistem. Diagram ini membantu menggambarkan interaksi antara aktor dengan sistem secara terstruktur, sehingga memudahkan pengembang dalam memahami kebutuhan pengguna sebelum sistem diimplementasikan [26].

Use Case Diagram terdiri dari beberapa komponen utama, yaitu aktor, *use case*, dan relasi. Aktor merupakan entitas yang berinteraksi dengan sistem, sedangkan *use case* menggambarkan fungsi atau layanan yang disediakan oleh sistem. Relasi menunjukkan hubungan antara aktor dan *use case*, maupun antar *use case* seperti *include* dan *extend*. Struktur ini memungkinkan sistem dapat dimodelkan secara sistematis dan menggambarkan alur interaksi secara jelas.

Penggunaan *Use Case Diagram* memberikan manfaat yang signifikan dalam pengembangan sistem, antara lain membantu dalam memvisualisasikan kebutuhan sistem, meningkatkan komunikasi antara pengembang dan pengguna, serta menjadi dasar dalam perancangan sistem lanjutan. Selain itu, diagram ini juga dapat digunakan sebagai acuan dalam proses pengembangan perangkat lunak karena mampu menggambarkan *blueprint* sistem secara menyeluruh [27].

Seiring dengan perkembangan teknologi, *Use Case Diagram* tetap menjadi salah satu alat pemodelan yang penting dalam rekayasa perangkat lunak. Diagram ini banyak digunakan dalam berbagai metode pengembangan sistem karena kemampuannya dalam menyederhanakan kompleksitas sistem menjadi bentuk visual yang mudah dipahami dan terstruktur [28].

2.6.3 Activity Diagram

Activity Diagram merupakan salah satu diagram dalam Unified Modeling Language (UML) yang digunakan untuk menggambarkan alur aktivitas atau proses bisnis dalam suatu sistem secara berurutan. Diagram ini menunjukkan bagaimana suatu proses dimulai, dijalankan, hingga selesai, serta menggambarkan aliran kontrol antar aktivitas dalam sistem. Activity Diagram banyak digunakan dalam tahap analisis dan perancangan sistem untuk memodelkan proses bisnis secara lebih detail dan terstruktur[29].

Activity Diagram mampu merepresentasikan aliran kerja sistem secara grafis sehingga memudahkan pengembang dalam memahami proses yang terjadi di dalam sistem. Diagram ini biasanya digunakan untuk menggambarkan proses operasional, logika bisnis, serta interaksi antar komponen dalam sistem. Dengan adanya visualisasi tersebut, proses analisis menjadi lebih mudah dan sistematis.

Komponen utama dalam Activity Diagram meliputi *initial node* sebagai titik awal, *activity* sebagai aktivitas atau proses yang dilakukan, *decision node* sebagai percabangan kondisi, *control flow* sebagai alur proses, serta *final node* sebagai titik akhir. Dengan struktur tersebut, Activity Diagram mampu menggambarkan alur proses secara lengkap mulai dari input hingga output sistem.

Penggunaan Activity Diagram dalam pengembangan sistem informasi memberikan berbagai manfaat, antara lain membantu dalam memahami proses bisnis yang kompleks, mengidentifikasi potensi kesalahan dalam alur sistem, serta meningkatkan efisiensi dalam perancangan sistem. Selain itu, Activity Diagram juga dapat digunakan sebagai dasar dalam pengembangan sistem karena mampu menggambarkan kebutuhan proses secara rinci[29].

Dalam penelitian terbaru, Activity Diagram digunakan sebagai alat pemodelan untuk mendukung pengembangan sistem berbasis digital, seperti sistem pemesanan online dan sistem informasi berbasis web. Diagram ini membantu dalam mendefinisikan alur kerja sistem secara jelas sehingga sistem yang dikembangkan dapat berjalan sesuai dengan kebutuhan pengguna[29].

2.7 Teknologi Pengembangan Aplikasi Web

Dalam proses pengembangan sistem, diperlukan beberapa *tools* atau teknologi yang mendukung pembangunan aplikasi agar dapat berjalan dengan baik. Sistem ini dikembangkan menggunakan MERN *Stack*, yaitu kombinasi teknologi MongoDB, Express.js, React.js, dan Node.js yang memungkinkan pengembangan aplikasi *web* secara

full-stack menggunakan bahasa pemrograman JavaScript. *MERN Stack* banyak digunakan karena mendukung pengembangan aplikasi *web* yang dinamis, *scalable*, dan efisien [30].

2.7.1 MongoDB

MongoDB merupakan sistem manajemen basis data NoSQL yang menyimpan data dalam bentuk dokumen berbasis JSON (*JavaScript Object Notation*) sehingga memudahkan integrasi dengan aplikasi berbasis JavaScript. MongoDB memungkinkan penyimpanan data yang fleksibel, *scalable*, serta mampu menangani data dalam jumlah besar [30].

Dalam penelitian pengembangan aplikasi berbasis *MERN Stack*, MongoDB digunakan sebagai *database* utama untuk menyimpan data pengguna, data transaksi, maupun data lainnya yang dibutuhkan oleh sistem. Contoh penggunaan MongoDB dalam sistem:

1. Menyimpan data akun pengguna
2. Menyimpan data transaksi atau aktivitas sistem
3. Menyimpan data informasi yang akan ditampilkan pada halaman *web*

Contoh struktur data MongoDB:

```
{
  "nama": "Dimas",
  "email": "dimas@gmail.com",
  "role": "admin"
}
```

Gambar 2.4 Struktur Data MongoDB

2.7.2 Express.js

Express.js merupakan *framework backend* yang berjalan di atas Node.js dan digunakan untuk membangun REST API serta mengelola *request* dan *response* pada aplikasi *web*. *Framework* ini bersifat ringan dan fleksibel sehingga banyak digunakan dalam pengembangan aplikasi berbasis JavaScript [30]. Express.js berfungsi sebagai penghubung antara *frontend* dan *database* dengan menangani berbagai permintaan dari *client*, memproses data, kemudian mengirimkan respons kembali ke pengguna. Contoh penggunaan Express.js dalam sistem:

1. Membuat *endpoint API*
2. Mengelola *routing* aplikasi
3. Menghubungkan *server* dengan *database* MongoDB

Contoh kode Express.js:

```
const express = require('express')
const app = express()

app.get('/api/users', (req, res) => {
  res.json({message: "Data user berhasil diambil"})
})

app.listen(5000, () => {
  console.log("Server running on port 5000")
})
```

Gambar 2.5 Kode Express.js

2.7.3 React.js

React.js merupakan *library* JavaScript yang digunakan untuk membangun antarmuka pengguna (*user interface*) yang interaktif dan responsif. React menggunakan konsep *component-based architecture*, sehingga memudahkan pengembangan dan pemeliharaan kode aplikasi [31]. React memungkinkan pengembang membuat *Single Page Application* (SPA) yang dapat meningkatkan performa aplikasi karena halaman tidak perlu dimuat ulang secara keseluruhan. Contoh penggunaan React.js dalam sistem:

1. Menampilkan *dashboard* sistem
2. Menampilkan halaman *login* dan registrasi
3. Menampilkan data dari API

Contoh kode React.js:

```
function Dashboard() {
  return (
    <div>
      <h1>Dashboard Sistem</h1>
    </div>
  )
}

export default Dashboard
```

Gambar 2.6 Kode React.js

2.7.4 Node.js

Node.js merupakan *runtime environment* yang memungkinkan JavaScript dijalankan di sisi *server*. Node.js menggunakan arsitektur *event-driven* dan *non-blocking I/O*, sehingga mampu menangani banyak koneksi secara efisien dan meningkatkan performa aplikasi *web* [30]. Node.js digunakan untuk menjalankan *server backend* dan mengelola komunikasi antara *frontend* dan *database*. Contoh penggunaan Node.js dalam sistem:

1. Menjalankan *server* aplikasi
2. Mengelola *API backend*
3. Menghubungkan aplikasi dengan *database* MongoDB

Contoh kode Node.js:

```
const http = require('http')

const server = http.createServer((req, res) => {
  res.write("Server berjalan")
  res.end()
})

server.listen(3000)
```

Gambar 2.7 Kode Node.js

2.8 Metode Pengujian Sistem

Pengujian sistem merupakan tahap penting dalam pengembangan perangkat lunak untuk memastikan bahwa sistem yang dibangun telah berjalan sesuai dengan kebutuhan pengguna dan bebas dari kesalahan fungsional. Pengujian dilakukan setelah proses implementasi sistem selesai dilakukan. Pada penelitian ini metode pengujian yang digunakan adalah *Black Box Testing* dan *User Acceptance Testing* (UAT). Kedua metode tersebut digunakan untuk memastikan bahwa fungsi sistem berjalan dengan baik serta sistem dapat diterima oleh pengguna.

1. *Black Box Testing*

Black Box Testing merupakan metode pengujian perangkat lunak yang berfokus pada pengujian fungsi sistem tanpa melihat struktur internal kode program. Pengujian ini dilakukan dengan cara memberikan *input* pada sistem, kemudian mengamati *output* yang dihasilkan untuk mengetahui apakah hasil tersebut sudah sesuai dengan yang diharapkan.

Metode ini disebut sebagai pengujian kotak hitam karena penguji hanya melihat dari sisi *input* dan *output* sistem tanpa mengetahui proses internal yang terjadi di dalam program [32].

Dalam pengujian *Black Box Testing*, penguji akan membuat beberapa skenario pengujian untuk memastikan bahwa setiap fitur dalam sistem berjalan sesuai dengan spesifikasi yang telah dirancang sebelumnya. Pengujian ini biasanya dilakukan pada fitur-fitur utama seperti proses *login*, pengolahan data, penyimpanan data, serta penampilan data pada sistem.

Contoh penerapan metode *Black Box Testing* dapat dilihat pada penelitian yang menguji sistem informasi menggunakan teknik pengujian ini untuk memastikan bahwa fungsi sistem berjalan dengan baik dan sesuai dengan kebutuhan pengguna [33].

Tahapan dalam melakukan pengujian Black Box Testing pada sistem adalah sebagai berikut:

1. Identifikasi Fitur yang Diuji

Menentukan fitur atau fungsi sistem yang akan diuji, seperti login, input data, dan pengolahan data.

2. Menentukan Skenario Pengujian

Menyusun skenario pengujian berdasarkan kebutuhan sistem, baik untuk kondisi normal maupun kondisi kesalahan (error).

3. Menentukan Data Input

Menyiapkan data input yang akan digunakan dalam pengujian, baik data yang valid maupun tidak valid.

4. Menentukan Output yang Diharapkan

Menentukan hasil yang seharusnya muncul dari setiap skenario pengujian yang dilakukan.

5. Melakukan Pengujian

Menjalankan sistem dengan memasukkan data sesuai skenario yang telah dibuat.

6. Mencatat Hasil Pengujian

Mencatat hasil yang diperoleh dari sistem dan membandingkannya dengan output yang diharapkan.

7. Evaluasi Hasil Pengujian

Menentukan apakah sistem sudah berjalan sesuai dengan kebutuhan atau masih terdapat kesalahan yang perlu diperbaiki.

Tabel 2.1 Contoh *Black Box Testing*

No.	Fitur yang Diuji	<i>Input</i>	<i>Output</i> yang Diharapkan	Hasil
1	<i>Login</i>	<i>Email</i> dan <i>password</i> valid	Sistem berhasil masuk ke <i>dashboard</i>	Berhasil
2	<i>Login</i>	<i>Email</i> dan <i>password</i> salah	Sistem menampilkan pesan <i>error</i>	Berhasil
3	Tambah data	Mengisi <i>form</i> data	Data tersimpan di <i>database</i>	Berhasil

No.	Fitur yang Diuji	Input	Output yang Diharapkan	Hasil
4	Hapus data	Klik tombol hapus	Data berhasil dihapus	Berhasil

Melalui pengujian ini, dapat diketahui apakah seluruh fungsi sistem telah berjalan sesuai dengan kebutuhan yang telah ditentukan sebelumnya.

2. User Acceptance Testing (UAT)

User Acceptance Testing (UAT) merupakan metode pengujian sistem yang dilakukan oleh pengguna akhir (*end user*) untuk memastikan bahwa sistem yang telah dikembangkan sesuai dengan kebutuhan dan dapat digunakan dengan baik oleh pengguna. UAT bertujuan untuk mengevaluasi apakah sistem yang dikembangkan telah memenuhi kebutuhan operasional serta memberikan pengalaman penggunaan yang baik bagi pengguna [34].

Pada tahap ini, pengguna mencoba secara langsung fitur-fitur yang tersedia pada sistem dan memberikan penilaian terhadap kemudahan penggunaan, tampilan sistem, serta fungsionalitas sistem yang tersedia. Melalui proses ini dapat diketahui apakah sistem telah sesuai dengan kebutuhan pengguna dan layak untuk diimplementasikan [35].

UAT umumnya dilakukan dengan menggunakan kuesioner atau angket yang diberikan kepada responden setelah mereka menggunakan sistem. Kuesioner tersebut biasanya menggunakan skala penilaian seperti skala Likert untuk mengukur tingkat kepuasan dan penerimaan pengguna terhadap sistem. Hasil dari kuesioner kemudian dianalisis untuk menentukan tingkat kelayakan sistem yang dikembangkan [36].

Tahapan dalam melakukan pengujian User Acceptance Testing (UAT) adalah sebagai berikut:

1. Menentukan Responden

Menentukan pengguna yang akan terlibat dalam pengujian, seperti admin, guru, dan siswa sebagai pengguna sistem.

2. Menentukan Skenario Pengujian

Menyusun skenario penggunaan sistem yang akan diuji, seperti login, input nilai, melihat absensi, dan fitur lainnya.

3. Pelaksanaan Pengujian oleh Pengguna

Pengguna mencoba langsung sistem sesuai dengan skenario yang telah ditentukan.

4. Penyebaran Kuesioner

Setelah menggunakan sistem, responden diberikan kuesioner untuk menilai sistem.

5. Pengumpulan Data Hasil Pengujian

Mengumpulkan hasil kuesioner dari seluruh responden.

6. Analisis Data

Menganalisis hasil kuesioner menggunakan skala Likert untuk mengetahui tingkat kepuasan pengguna.

7. Kesimpulan dan Evaluasi

Menentukan apakah sistem telah memenuhi kebutuhan pengguna dan layak untuk digunakan atau masih perlu perbaikan.

Tabel 2.2 Contoh Kuesioner UAT

No.	Pernyataan	Sangat Setuju	Setuju	Netral	Tidak Setuju
1	Sistem mudah digunakan	✓			
2	Tampilan sistem menarik		✓		
3	Fitur sistem berjalan dengan baik	✓			
4	Sistem membantu pekerjaan pengguna	✓			

Hasil pengujian digunakan sebagai dasar untuk menilai tingkat kesesuaian sistem dengan kebutuhan pengguna serta menentukan kelayakan sistem sebelum diimplementasikan.