



BAB II
LANDASAN TEORI

BAB II

DASAR TEORI

2.1 Sistem Informasi

Sistem informasi adalah hasil dari pengolahan data menjadi bentuk yang lebih berguna dan bermakna bagi penerimanya. Tanpa adanya informasi, sistem tersebut tidak akan berfungsi dengan baik dan bahkan dapat mati. Data merupakan sumber informasi yang menggambarkan suatu kejadian yang sering terjadi. Namun, data perlu diproses dan diolah menjadi input yang berguna dalam suatu sistem. Oleh karena itu, diperlukan model untuk mengelompokkan dan memproses data agar dapat menghasilkan informasi yang bermanfaat[1].

2.2 Buku Induk Siswa

Buku induk siswa merupakan buku utama dan wajib di semua sekolah baik negeri maupun swasta, yang memberikan informasi yang paling lengkap tentang identitas dan keadaan siswa. Informasi tersebut meliputi identitas pribadi siswa dan informasi tentang nilai proses hasil belajar yang diperoleh selama di sekolah. Oleh karena itu pentingnya buku induk siswa untuk memudahkan penyimpanan data siswa dengan baik dan juga membantu alumni dalam memperoleh data diri dari pihak sekolah. Buku induk siswa adalah kumpulan nama siswa sepanjang masa sekolah di sebuah sekolah. Segera setelah menjadi siswa baru, peserta didik harus dicatat dalam buku induk siswa yang sering disebut sebagai buku pokok atau buku induk siswa di sekolah. Catatan dalam buku induk siswa ini dilengkapi dengan data peserta didik dan identitas siswa yang diambil dari formulir pendaftaran. Selain itu, identitas siswa dalam buku induk siswa berisi nomor kode sekolah, keterangan orangtua/wali siswa, nomor induk siswa, serta prestasi belajar siswa-siswi (daftar nilai) dari tahun ke tahun selama siswa tersebut belajar di sekolah tersebut. Pencatatan dalam buku induk harus jelas dan merupakan tanggung jawab Kepala Sekolah yang dapat diserahkan kepada Tenaga Administrasi Sekolah/TU[2].

2.3 Waterfall

Metode *Waterfall* menurut Pressman merupakan pendekatan yang memiliki ciri khas berupa tahapan yang terstruktur secara linier. Dalam metode ini, setiap fase harus diselesaikan secara menyeluruh sebelum melanjutkan ke fase berikutnya. Dengan fokus yang mendalam pada setiap fase, mulai dari analisis kebutuhan, desain, implementasi, hingga pengujian dan pemeliharaan, metode *Waterfall* memastikan bahwa setiap langkah dikerjakan secara maksimal sebelum beralih ke tahap berikutnya. Pendekatan ini memungkinkan pengelolaan yang lebih teratur dan dokumentasi yang rinci, meskipun kurang fleksibel dalam menghadapi perubahan di tengah proses pengembangan.

Berikut tahapan yang dilakukan sesuai dengan pengembangan sistem pada metode *waterfall*.

a. Requirement

Pada tahap ini, pengembang sistem perlu melakukan komunikasi dengan tujuan memahami kebutuhan perangkat lunak yang diinginkan oleh pengguna serta batasan yang ada pada perangkat lunak tersebut. Informasi tersebut bisa didapatkan melalui berbagai cara seperti wawancara, diskusi, atau survei langsung. Setelah itu, informasi yang diperoleh akan dianalisis untuk mendapatkan data yang sesuai dengan kebutuhan pengguna.

b. *Design*

Pada langkah ini, pengembang merancang sistem untuk membantu dalam menetapkan persyaratan perangkat keras (*hardware*) dan sistem, serta membantu dalam mendefinisikan struktur keseluruhan dari arsitektur sistem.

c. *Implementation*

Pada fase ini, sistem awalnya dibangun dalam bentuk program kecil yang disebut unit, yang kemudian tergabung dalam tahap berikutnya. Setiap unit dikembangkan dan diuji untuk memastikan fungsionalitasnya dalam apa yang disebut sebagai pengujian unit.

d. *Verification*

Pada langkah ini, sistem mengalami proses verifikasi dan pengujian untuk menilai apakah sistem secara keseluruhan atau sebagian memenuhi persyaratan sistem. Pengujian ini dapat dibagi menjadi tiga kategori: pengujian unit (dilakukan pada modul kode individu), pengujian sistem (untuk mengevaluasi respon sistem saat semua modul terintegrasi), dan pengujian penerimaan (dilakukan dengan atau atas nama pelanggan untuk menilai kepuasan terhadap semua kebutuhan pelanggan).

e. *Maintenance*

Tahap ini menandai akhir dari pendekatan waterfall. Pada tahap ini, perangkat lunak yang telah selesai dibangun diimplementasikan dan dilakukan pemeliharaan. Pemeliharaan mencakup tindakan memperbaiki kesalahan yang tidak terdeteksi pada tahap sebelumnya[4].

2.4 Laptop

Laptop adalah perangkat komputer portabel yang dirancang untuk mempermudah pekerjaan yang membutuhkan mobilitas tinggi. Laptop memiliki komponen dasar yang sama dengan komputer desktop, seperti CPU, RAM, dan hard disk, namun dikemas dalam bentuk yang lebih ringkas dan ringan sehingga mudah dibawa dan digunakan di mana saja. Dengan berkembangnya teknologi, laptop kini dilengkapi dengan fitur-fitur canggih yang mendukung berbagai kebutuhan mulai dari pekerjaan administratif hingga multimedia dan pengolahan data[5].

2.5 Balsamiq Mockup

Balsamiq Mockups adalah perangkat lunak yang dirancang untuk membuat desain atau prototipe tampilan antarmuka pengguna dalam sebuah aplikasi. Dengan *Balsamiq Mockups*, proses pembuatan antarmuka pengguna menjadi lebih mudah karena perangkat ini menyediakan berbagai alat yang memudahkan *prototyping* aplikasi yang sedang dikembangkan[6].

2.6 Visual Studio Code

Visual Studio Code (VS Code) adalah *editor* kode sumber yang ringan namun kuat, yang dirancang untuk mendukung berbagai bahasa pemrograman dan *framework*. *VS Code* menyediakan fitur-fitur seperti *debugging*, kontrol versi, dan penyorotan sintaks yang memudahkan pengembangan perangkat lunak. Jurnal ini menjelaskan bahwa *VS Code* mengintegrasikan berbagai ekstensi yang memungkinkan pengembangan yang lebih efisien dan fleksibel[7].

2.7 Web

Istilah *web* merujuk pada sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi digital seperti teks, gambar, video, *audio*, dan animasi yang dapat diakses melalui koneksi internet. Sementara itu, *website* atau situs adalah kumpulan halaman *web* yang dijalankan dari suatu alamat domain[8].

2.8 HTML

HTML merupakan bahasa markup yang digunakan untuk membuat dan memformat halaman *web*. HTML berfungsi untuk mengatur struktur dan tampilan konten pada halaman web dengan menggunakan elemen-elemen markup[9].

2.9 PHP

PHP (*Personal Home Page*) atau *HyperText Preprocessor*, adalah bahasa pemrograman yang digunakan untuk pengembangan website dengan sifat *server-side* dan diintegrasikan ke dalam HTML. Dalam dokumen HTML, kita dapat menyisipkan skrip PHP. Keunggulan PHP meliputi kemudahan dalam koneksi dan query database serta kompatibilitasnya dengan sistem operasi berbasis *Windows* maupun *UNIX*[9].

2.10 MySQL

MySQL adalah sebuah DBMS (*Database Management System*) yang mendukung multiuser dan multithreading. SQL (*Structured Query Language*) adalah bahasa standar yang digunakan untuk sistem manajemen database relasional, sehingga istilah seperti tabel, baris, dan kolom digunakan dalam MySQL. MySQL dikembangkan oleh TcX dan telah terbukti mampu mengelola sistem yang memiliki 40 database berisi 10.000 tabel, dengan 500 tabel di antaranya memiliki 7 juta baris (sekitar 100 gigabyte data). Dalam MySQL, sebuah database terdiri dari satu atau lebih tabel, setiap tabel terdiri dari beberapa baris, dan setiap baris mengandung satu atau lebih kolom[9].

2.11 Laravel

Laravel Framework adalah kerangka kerja pemrograman *open source* yang digunakan oleh banyak pengembang di seluruh dunia. Menurut Mulyadi, kemudahan penggunaan dan dokumentasi lengkap adalah beberapa faktor yang menjadikan Laravel sangat populer dalam beberapa tahun terakhir. Laravel menggunakan pola arsitektur *Model-View-Controller* (MVC), yang memisahkan aplikasi berdasarkan komponen-komponen seperti manipulasi data, controller, dan antarmuka pengguna. Keunggulan dari pendekatan ini adalah kemudahan dalam proses pemeliharaan dan skalabilitas aplikasi[10].

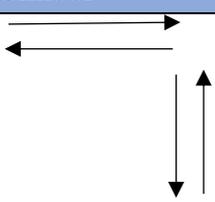
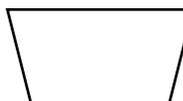
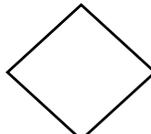
2.12 Model View Controller (MVC)

Model View Controller (MVC) adalah sebuah pola atau teknik pemrograman yang memisahkan pengembangan aplikasi menjadi beberapa komponen utama. Teknik ini membagi aplikasi menjadi tiga bagian: *model*, *view*, dan *controller*. Pembagian ini memungkinkan pengembang untuk fokus pada satu aspek dari aplikasi, sehingga mempermudah proses pengembangan dan pemeliharaan aplikasi secara keseluruhan. Dengan pemisahan ini, setiap bagian memiliki tanggung jawab yang jelas, seperti *model* untuk mengelola data, *view* untuk antarmuka pengguna, dan *controller* untuk mengatur aliran data di antara keduanya[11].

2.13 Flowchart

Flowchart adalah sebuah metode diagramatik yang digunakan untuk mempresentasikan algoritma dengan tujuan menghasilkan solusi dari suatu masalah. Diagram ini mempermudah pemahaman alur proses atau langkah-langkah yang diperlukan dalam menyelesaikan sebuah masalah, sehingga mempermudah komunikasi dan analisis proses tersebut. Dengan menggunakan flowchart, setiap tahapan dalam algoritma dapat digambarkan secara jelas dan sistematis, memungkinkan identifikasi potensi kesalahan atau area untuk perbaikan sebelum implementasi dilakukan, seperti pada tabel 2.1[12].

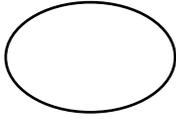
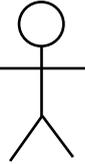
Tabel 2. 1 Simbol dan Fungsi Flowchart

Simbol	Nama	Fungsi
	Flow Direction Symbol / Connecting Line	Berfungsi untuk menghubungkan symbol yang satu dengan yang lainnya, menyatakan arus suatu proses.
	<i>Processing</i>	Digunakan untuk pengolahan yang akan dilakukan dalam komputer
	<i>Manual Operation</i>	Digunakan untuk menunjukkan pengolahan yang tidak dilakukan oleh komputer
	<i>Decision</i>	Digunakan untuk memilih proses yang akan dilakukan berdasarkan kondisi tertentu
	<i>Terminal</i>	Digunakan untuk memulai dan mengakhiri program
	<i>Document</i>	Mencetak keluaran dalam bentuk dokumen (melalui printer).
	Input / Output	Digunakan untuk menyatakan input dan output tanpa melihat jenisnya

2.14 Use Case Diagram

Use Case Diagram adalah diagram statis yang menunjukkan kumpulan use case dan aktor-aktor (jenis khusus dari kelas). Diagram ini memiliki dua fungsi utama: mendefinisikan fitur-fitur yang harus disediakan oleh sistem dan menggambarkan sifat sistem dari perspektif pengguna, seperti pada tabel 2.2[12].

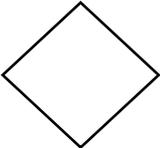
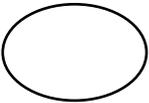
Tabel 2. 2 Simbol dan Fungsi Use Case

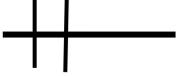
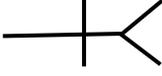
Simbol	Nama	Keterangan
	<i>Use Case</i>	Fungsionalitas yang disediakan oleh sistem terdiri dari unit-unit yang saling bertukar pesan antara unit atau aktor. Biasanya, ini dinyatakan dengan menggunakan kata kerja di awal frasa nama use case.
	Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, tetapi berada di luar sistem informasi tersebut. Meskipun simbol untuk aktor adalah gambar orang, aktor tidak selalu merupakan orang. Biasanya, aktor dinyatakan menggunakan kata benda di awal frasa nama aktor.
	<i>Association</i>	Komunikasi antara aktor dan use case terjadi ketika aktor berpartisipasi dalam use case atau ketika use case berinteraksi dengan aktor.
	<i>Include</i>	Relasi use case tambahan mengacu pada sebuah use case yang membutuhkan use case lain untuk dapat berfungsi atau sebagai syarat agar use case tersebut dapat dijalankan.

2.15 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah sebuah diagram yang digunakan untuk merancang hubungan antar tabel dalam basis data. ERD merupakan teknik yang digunakan untuk memodelkan kebutuhan data suatu organisasi, biasanya oleh Sistem Analis dalam tahap analisis persyaratan proyek pengembangan sistem. Dengan menggunakan ERD, hubungan antara berbagai entitas dalam basis data dapat dipetakan secara visual, sehingga memudahkan pemahaman dan pengelolaan struktur data yang kompleks, seperti pada tabel 2.3[12].

Tabel 2. 3 Simbol dan Fungsi ERD

Simbol	Nama	Keterangan
	<i>Relasi</i>	Hubungan yang terjadi antara satu entitas atau lebih yang tidak memiliki fisik tetapi hanya bersifat konseptual. Untuk mengetahui jenis hubungan yang ada antara dua entitas.
	Atribut	Atribut adalah karakteristik dari entitas atau relasi yang memberikan penjelasan detail tentang entitas atau relasi tersebut. Atribut berfungsi untuk menggambarkan

		sifat-sifat yang dimiliki oleh sebuah entitas. Biasanya, atribut direpresentasikan dalam bentuk elips atau lingkaran untuk memperjelas keterkaitannya dengan entitas.
	<i>One To One</i>	Alur berfungsi sebagai penghubung antara atribut dengan entitas, serta antara entitas dengan relasi. Alur direpresentasikan dalam bentuk garis untuk menunjukkan hubungan antara komponen-komponen dalam sebuah diagram.
	<i>Realization</i>	One to One, digunakan untuk menghubungkan antar entitas dengan hubungan satu ke satu
	<i>One To Many</i>	One to Many atau Many to One, digunakan untuk menghubungkan antar entitas dengan hubungan satu ke banyak atau sebaliknya
	<i>Entity</i>	Entitas adalah objek yang dapat dibedakan dengan objek lainnya, memberikan identitas dengan label dan nama, dan direpresentasikan dalam bentuk persegi panjang.