

## **BAB II**

### **KAJIAN PUSTAKA**

#### **2.1 Penelitian Relevan**

Hasil penelitian terkait yang akan dikaji untuk menyelesaikan masalah yang berkaitan dengan penelitian ini adalah sebagai berikut:

Pertama, hasil penelitian terkait dilakukan oleh Supianto, Atikah, Siti Martiah melakukan penelitian yang berjudul “Sistem Informasi administrasi sekolah pada SMK Kesuma 1 Depok”.. menjelaskan bahwa Saat ini, sistem pengelolaan administrasi sekolah pada SMK Kesuma Bangsa 1 Depok masih dilakukan secara manual, sehingga menyebabkan penumpukan kertas, terlalu lama pencarian data Ketika ingin berbagi data. Semua siswa melakukan pembayaran sekolah wajib membawa buku bayaran sekolah, jika tidak membawa buku bayaran tidak bisa melakukan pembayaran sekolah. Pelayanan informasi yang diberikan kepada semua siswa dan orang tua siswa tidak maksimal dikarenakan kemungkinan terdapat ketidaksamaan data. Sulitnya membuat laporan yang diperlukan per periode yang diinginkan karena sistem manual tidak dapat menghasilkan secara cepat dan akurat. Hal ini menjadi perhatian penulis untuk mengembangkan sistem pada SMK Kesuma Bangsa 1 Depok tempat dimana penulis melakukan riset yang selama ini masih menggunakan sistem manual dan belum menggunakan sistem yang terkomputerisasi dalam melakukan segala transaksinya. Oleh sebab itu penulis ingin merancang sebuah sistem informasi administrasi sekolah dalam bentuk aplikasi dengan Bahasa pemrograman java netbeans. (Maskurin, 2019)

Kedua, hasil penelitian terkait selanjutnya dilakukan oleh Dea Yuziztin, Dindon Fikri Aji, Dwi Ananto Pamungkas, dengan judul “Sistem Informasi Administrasi Siswa Berbasis Website Pada SMA Islam Putradana Bekasi”. menjelaskan bahwa masalah yang dihadapi ialah Teknologi yang berjalan pada SMA Islam Putradarma masih belum menggunakan teknologi berbasis website untuk pengolahan data administrasi dan absensi siswanya. Sehingga semua sistem masih menggunakan sistem manual

Menggunakan sistem manual menyebabkan banyak sekali kendala yang timbul karena dokumen berupa lembaran kertas sehingga sering terjadi redundansi data siswa, kurangnya efektifitas dalam pengelolaan data karena data yang dibutuhkan masih harus berinteraksi langsung dengan pihak yang bersangkutan, serta lambatnya dalam pembuatan. (Hidayatullah, Pradana, & Saputra, 2017).

Ketiga, hasil penelitian yang dilakukan oleh Imam Sutono, dengan judul “Perancangan Sistem Informasi Administrasi Akademik Berbasis Web”. menjelaskan kendala Dimana sistem informasi kendala yang dihadapi dalam menerapkan sistem seperti itu. Selain itu, penulis berharap pembaca dapat memahami penggunaan sistem aplikasi sistem informasi akademik. Selama penulis penelitian menemukan bahwa pada SDN Pondok Labu 12 ini tidak memiliki sistem informasi pengolahan data administrasi akademik terkomputerisasi sehingga dapat membantu mencari informasi cepat dan efektif. Hal ini karena kurangnya sumber daya manusia yang tahu tentang sistem informasi, setelah melakukan beberapa penelitian, penulis menyimpulkan bahwa SDN Pondok Labu 12 Pagi membutuhkan sistem komputerisasi sehingga dapat membantu guru dalam membuat administrasi, penulis mencoba untuk membuat sistem Administrasi Akademik menggunakan Bahasa pemrograman.(Putra & Novelan, 2020).

Dari ketiga uraian jurnal diatas dapat disimpulkan bahwa Persamaan dari penelitian ini dengan hasil penelitian terkait sebelumnya ialah sistem informasi yang digunakan berbasis web. Sedangkan untuk perbedaan terletak pada objek penelitian, judul dan bahasa pemrograman, model yang dipakai, pengujian sistem serta *user* yang berintraksi dengan sistem. Penelitian ini diharapkan dapat membantu staf dalam pelaksanaan tugasnya.

## **2.2 Landasan Teori**

Untuk mendukung pembuatan penelitian ini, maka perlu dikemukakan hal-hal atau teori yang berkaitan dengan permasalahan dan ruang lingkup pembahasan sebagai landasan dalam penelitian ini.

### 2.2.1 Sistem Informasi

Sistem informasi adalah suatu sistem dalam suatu organisasi yang memenuhi kebutuhan pemrosesan transaksi harian, mendukung operasi, merupakan kegiatan manajerial dan kebutuhan strategis suatu organisasi serta menyediakan pihak luar dengan laporan yang diperlukan (Maskurin, 2019).

### 2.2.2 Flowchart

*Flowchart* adalah deskripsi grafis dari urutan prosedur suatu program yang menunjukkan alur dalam suatu program atau prosedur sistem secara logis. *Flowchart* digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. *flowchart* merupakan langkah-langkah penyelesaian masalah yang dituliskan dalam simbol-simbol tertentu. Diagram ini akan menunjukkan alur program secara logika. Bagan alir (*Flowchart*) di antaranya ialah :

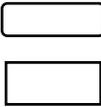
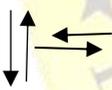
1. Diagram alir digambarkan dengan orientasi dari atas kebawah dan dari kiri kekanan.
2. Setiap kegiatan /proses dalam kegiatan alir harus harus dinyatakan secara eksplisit.
3. Setiap diagram alir harus dimulai dari satu start state dan berakhir pada satu atau lebih terminal akhir /terminator/haltstate.

Gunakan *connector* dan *off-page connector state* dengan label yang sama untuk menunjukkan keterhubungan antar path algoritma yang terputus/terpotong misalnya sebagai akibat pindah/ganti halaman. Tujuan dari *flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi, dan jelas menggunakan simbol-simbol yang standar. (Sari, 2020)

**Tabel 2.1.** Simbol *Flowchart*

Simbol	Nama Simbol	Makna
	Koneksi	Penghubung dapat diisi huruf/angka/huruf dan angka

**Tabel 2.1.** Simbol *Flowchart* (Lanjutan)

Simbol	Nama Simbol	Makna
	Dokumen	Penggunaan untuk dokumen/teks
	Mulai/Akhir	Digunakan untuk mulai awal dan mengakhiri atau selesai
	Process	Digunakan untuk memproses , hitungan dll.
	Data input, output	Masukan input atau keluaran output
	Penyimpanan data	Untuk penyimpanan data/file
	Magnetic Disk	Penyimpanan Permanen
	Panah Penghubung	Menunjukkan alur data

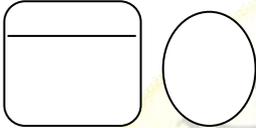
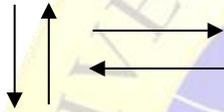
### 2.2.3 Context Diagram

Diagram konteks adalah diagram yang terdiri dari proses dan menggambarkan keseluruhan sistem yang akan dibangun. Diagram konteks adalah level paling tinggi dari *data flow diagram* (DFD) yang menggambarkan *input* ke dalam sistem atau *output* yang dihasilkan oleh sistem. Diagram konteks bertujuan untuk mengartikan, membangun dan memvisualisasikan model sistem. (Verawati & Liksha, 2018)

Secara uraian dapat dikatakan bahwa konteks diagram berisi siapa saja yang memberikan data (*inputan*) ke sistem serta kepada siapa data informasi itu diberikan (melalui sistem) dan apa saja isi atau jenis laporan yang harus

dihasilkan sistem. Adapun symbol-simbol *context diagram* dapat dilihat pada tabel berikut:

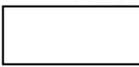
**Tabel 2.2.** *Simbol Context Diagram*

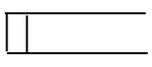
Simbol	Keterangan
	Entitas External Menggambarkan sistem yang akan memberikan input atau menerima output dari sistem.
	Proses Menggambarkan entitas atau proses aliran data yang masuk ke dalam proses untuk hasil keluaran (output).
	Aliran Data Menggambarkan aliran data atau informasi ke sistem.

#### 2.2.4 Data Flow Diagram (DFD)

*Data Flow Diagram* (DFD) adalah alat pembuatan model yang memungkinkan professional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dibagi menjadi beberapa tahapan dan dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi (Yurindra, 2017). Simbol-simbol DFD dapat dilihat pada tabel berikut:

**Tabel 2.3.** *Simbol Data Flow Diagram*

No	Nama	Simbol	keterangan
1	Entity		Merupakan sumber / tujuan dari aliran data ke sistem
2	Aliran data		menggambarkan aliran data dari satu proses ke proses lainnya.
3	Proses		proses/fungsi yang mentransformasikan data secara umum.

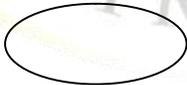
4	Berkas /tempat penyimpanan		merupakan komponen yang berfungsi untuk menyimpan data
---	----------------------------	---	--

### 2.2.5 ERD (*Entity Relational Diagram*)

*Entity Relationship Diagram* adalah suatu model yang menggambarkan hubungan antardata di databases berdasarkan data yang relasi/hubungan antar data. *Entity Relationship Diagram* dibagi menjadi dua, yaitu CDM (*Conceptual Data Model*) dan PDM (*Physical Data Model*). CDM (*Conceptual Data Model*) adalah data yang dapat merepresentasikan sebuah kenyataan, dimasukkan ke sebuah proses logika dan dapat menghasilkan informasi, sedangkan, ERD (*Physical Data Model*) adalah jenis ERD yang mana data disimpan pada media penyimpanan (*storage*) dalam suatu susunan secara fisik. (Sari, 2020)

Adapun simbol-simbol dari *Entity Relationship Diagram* (ERD) adalah sebagai berikut:

**Tabel 2.4.** Simbol-simbol Entity Relationship Diagram (ERD)

Nama	Simbol	Keterangan
<i>Entitas</i>		Suatu kumpulan objek atau sesuatu yang dapat dibedakan atau dapat didefinisikan secara unik.
<i>Relationship</i>		Hubungan yang terjadi antara satu entitas atau lebih.
<i>Atribut</i>		Karakteristik dari entitas atau <i>Relationship</i> yang menyediakan penjelasan detail entitas atau <i>relation</i> .
<i>Link</i>		Baris sebagai penghubung antara himpunan, relasi dan himpunan entitas dari atributnya.

### 2.2.6 MySQL

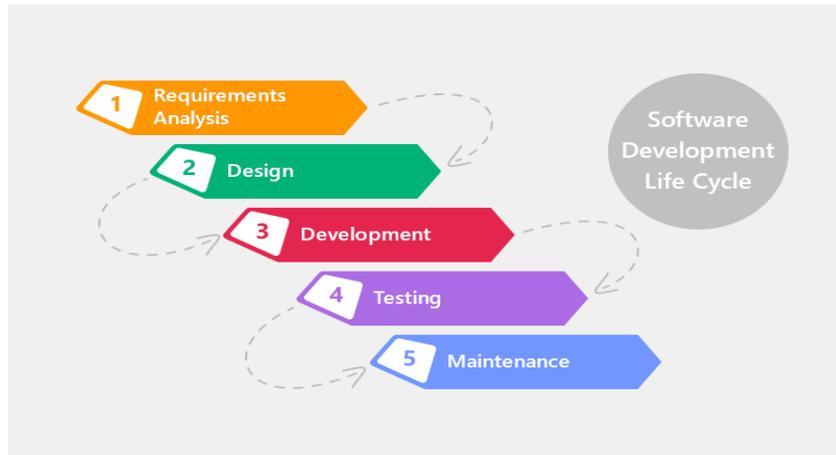
MySQL adalah suatu perangkat lunak database relasi ( *Relation Database Management Sistem/RDMS* ) MySQL merupakan sebuah perangkat lunak atau software sistem manajemen basis data SQL atau DBMS *Multithread* dan *multi user*. MySQL sebenarnya merupakan turunan dari salah satu konsep utama dalam database untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan secara mudah dan otomatis. (Verawati & Liksha, 2018)

### 2.2.7 PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang dirancang untuk membangun aplikasi *web*. Ketika dipanggil dari *web browser*, program yang ditulis dengan akan di-parsing di dalam *web server* oleh interpreter PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke *web browser*. Karena pemrosesan program PHP dilakukan di lingkungan *web server* maka dikatakan sebagai bahasa server (*server – side*). (Maskurin, 2019).

### 2.2.8 Model Waterfall

Model *waterfall* merupakan “metode yang melakukan pendekatan secara sistematis dan terurut mulai dari level kebutuhan sistem sampai menuju ke tahap analisis, desain, *coding*, *testing/verification*, dan *maintenance*”. Model *waterfall* juga merupakan model pengembangan perangkat lunak yang menurun ke bawah terdiri dari *planning*, *analysis*, *design*, *implementation*, dan *use* serta identik dengan SDLC (*system development life cycle*). (Muharto & Arisandy, 2016)



**Gambar 2.1.** Model *Waterfall*

