

BAB II TINJAUAN PUSTAKA

2.1 Sistem Informasi Administrasi

Menurut Isa, *et al.*, (2022), sistem informasi merupakan sekumpulan elemen yang saling terhubung atau berkaitan untuk tujuan tertentu, proses pengolahan data mentah menjadi informasi secara tersistem dapat menghasilkan informasi yang lebih sederhana dan mudah digunakan oleh pengguna.

Menurut Supriadi and Bachtiar (2018), Administrasi merupakan perencanaan, pengendalian dan pengorganisasian pekerjaan perkantoran serta penggerakan mereka yang melaksanakannya agar mencapai tujuan yang telah ditetapkan

Sehingga sistem informasi administrasi adalah kumpulan dari kompoenkomponen atau elemen-elemen yang saling berhubungan untuk melakukan proses pencatatan, pengaturan, pengalokasian suatu kegiatan untuk mencapai tujuan tertentu dengan *menggunakan sarana perlengkapan dan peralatan yang ada*.

2.2 Web Based

Menurut Urbieta *et al* (2019), *Web Based* adalah aplikasi yang dibuat berbasis *web* yang membutuhkan *web server* dan *browser* untuk menjalankannya.

Menurut Purwati, *et al* (2018) Dengan membuat sistem berbasis *web based* ada beberapa hal yang penting dan harus kita pikirkan sebelum membangun sistem tersebut, diantaranya:

1. Tidak membutuhkan *hardware* dengan spesifikasi yang tangguh untuk menjalankan aplikasinya.
2. Server yang dibutuhkan cukup diinstallkan *tools* pendukung saja agar klien mudah menjalankan aplikasi
3. Infrastruktur jaringan yang dibutuhkan juga cukup besar karena aplikasi yang dibuat dapat diakses dari jaringan luar (internet).

4. Aplikasi berbasis *web based* dapat diakses dari berbagai perangkat dengan syarat menggunakan *web browser* saja sudah dapat mengaksesnya.
5. Jika aplikasi yang sudah jadi ingin di *update*, sangat mudah untuk melakukannya karena tidak membutuhkan membuka keseluruhan aplikasi.

2.2.1 PHP

Menurut Subagja (2018), PHP adalah bahasa *server-side-scripting* yang menyatudengan HTML untuk membuat halaman *website* yang dinamis.

Menurut Oetomo and Maharginono (2020), berpendapat bahwa *PHP Hypertext Preprocessor* adalah bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server (server side HTML embedded scripting)*. PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua script PHP dieksekusi pada *server* dimana script tersebut dijalankan. Dengan menggunakan program PHP, sebuah *website* akan lebih interaktif dan dinamis.

Sehingga menurut Maldhan (2019) PHP merupakan bahasa pemrograman yang digunakan oleh pengembang untuk membuat sistem *website* dengan kumpulan bahasa HTML dan *script* lainnya.

2.2.2 MySQL

Menurut MySQL (2018), *MySQL* adalah singkatan dari *Structure Query Language* yang digunakan untuk mendefinisikan structure data, memodifikasi data pada basis data, menspesifikasi batasan keamanan (*security*), hingga pemeliharaan data.

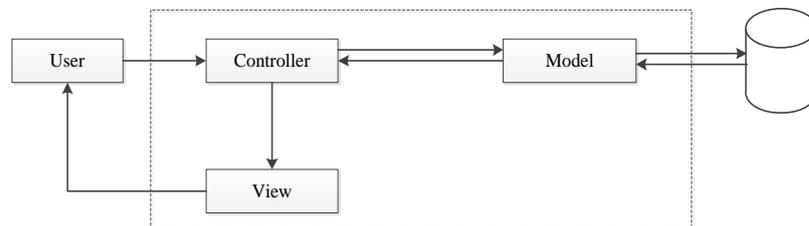
Menurut (Amin, 2018) mendefinisikan *mysql* adalah RDBMS yang cepat dan mudah digunakan, serta sudah banyak digunakan untuk berbagai kebutuhan.

MySQL merupakan bahasa standar yang paling banyak digunakan untuk mengakses *database* relasional dan merupakan aplikasi yang dapat dipergunakan secara bebas.

2.2.3 CodeIgniter

Menurut Raharjo (2018), *CodeIgniter* adalah *Framework* untuk bahasa pemrograman PHP, yang dibuat Rick Ellis pada tahun 2006. *CodeIgniter* memiliki banyak fitur yang membantu para pengembang PHP untuk dapat membuat aplikasi secara mudah dan cepat serta memiliki sifat yang fleksibel dapat mengembangkan dalam perangkat *website*, dekstop maupun *mobile*.

CodeIgniter memiliki konsep atau pola *Model-View-Controller* (MVC) sehingga kode-kode dapat di sederhanakan.

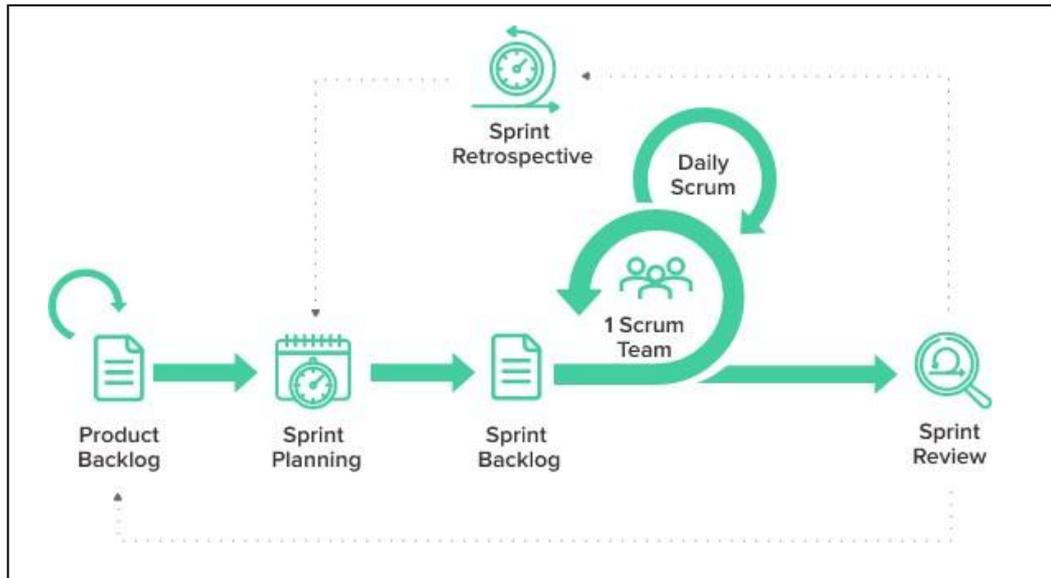


Gambar 2.1 Arsitektur MVC

2.3 Metode Scrum

“*Scrum Method Implementation in a Software Development Project Management*” dalam (Julianto, 2019) menyatakan bahwa *Scrum* pertama kali dikembangkan oleh Schwaber dan Sutherland pada tahun 1993 dan tujuannya adalah menjadi metodologi pengembangan yang mengikuti prinsip-prinsip metodologi *Agile*.

Scrum adalah suatu metodologi atau kerangka kerja yang terstruktur untuk mendukung pengembangan produk yang kompleks. *Scrum* terdiri dari sebuah tim yang memiliki peran dan tugas masing-masing. Setiap komponen dalam kerangka melayani tujuan tertentu dan sangat penting untuk kesuksesan penggunaan *Scrum* (Schwaber and Sutherland, 2017).



Gambar 2.2 Tahapan-Tahapan Metode Scrum

2.3.1 Tahapan-Tahapan Metode Scrum

Adapun tahapan-tahapan dalam *Scrum* oleh (Schwaber and Sutherland, 2017) adalah sebagai berikut ini:

1. Product Backlog

Peneliti sistem akan mengumpulkan dan menyusun semua kebutuhan sistem dan permintaan pengguna terhadap sistem, misalnya fitur-fitur yang dibutuhkan oleh pengguna terhadap sistem. *Product backlog* berada dalam tanggung jawab *product owner*. Setelah targetnya ditetapkan, semua kebutuhan dan permintaan akan dibagikan menjadi poin-poin kecil yang mana setiap poin tersebut mempunyai tingkat layak untuk dikembangkan.

2. Sprint Planning

Sprint Planning merupakan sebuah langkah yang wajib dilaksanakan setiap saat akan memulainya sprint baru. Pada langkah tersebut peneliti akan menyusun pekerjaan-pekerjaan apa saja yang harus diselesaikan dalam 1 *sprint*.

3. Sprint Backlog

Perencanaan *sprint* dilakukan dalam pertemuan/*meeting* antara pemilik produk dan tim developer, yang akan berkolaborasi untuk memilih produk yang akan dikembangkan *backlog* untuk dimasukkan kedalam proses *sprint*. Hasil dari pertemuan tersebut adalah *sprint backlog*.

4. Sprint

Dalam *Scrum*, *Sprint* adalah sebuah kerangka waktu yang berdurasi maksimal 1 bulan untuk mengembangkan produk yang berpotensi untuk dirilis. Dalam *Sprint* terdapat 2 bagian pekerjaan, yaitu:

a. Pertemuan Harian (*Daily Standup Meeting*)

Merupakan pertemuan dimana setiap 24 jam (1 hari), tim pengembang bertemu untuk membahas proses pengembangan produk.

b. Refleksi *Sprint*

Merupakan pertemuan yang dilakukan setiap bulannya, yang bertujuan untuk membahas hal dari *Sprint Backlog* yang telah berjalan dan telah berhasil dikerjakan, serta dapat memperbaiki dan meningkatkan kualitas produk pada *Sprint* yang berikutnya.

5. Working Increment (*Sprint Rivew*)

Increment merupakan hasil dari seluruh hal dalam *product backlog* yang telah selesai dikerjakan pada seluruh *sprint*.

2.4 Alat Pengembang Sistem

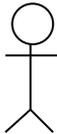
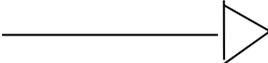
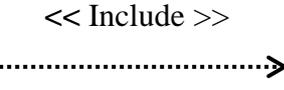
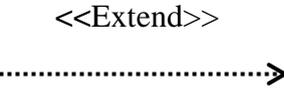
Alat pengembang sistem merupakan konsep desain yang digunakan untuk menggambarkan sistem dengan menggunakan diagram (Agarina and Karim, 2019). Penyesuaian alat yang digunakan harus sesuai dengan metode pengembangan yang dilakukan salah satunya adalah penerapan *Unified Modelling Language*. Menurut Rosa dan Salahuddin (2019), *Unified Modelling Language* adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada *Unified Modelling Language*.

2.4.1 Use Case Diagram

Menurut Rosa dan Salahuddin (2019) *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada Tabel 2.1.

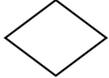
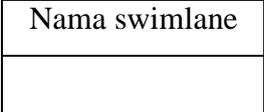
Tabel 2.1 Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1.		<i>Use case</i> : Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> .
2.		Aktor: seseorang/sesuatu yang berinteraksi dengan yang akan dibuat. diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda
3.		Asosiasi (<i>association</i>): merupakan komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.		Generalisasi (<i>generalization</i>): merupakan hubungan (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum
5.		Include berarti <i>use case</i> yang ditambahkan akan dipanggil saat <i>use case</i> tambahan dijalankan.
6.		Ekstensi (<i>extend</i>) merupakan <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.

2.4.2 Activity Diagram

Menurut Rosa dan Salahuddin (2019) *activity diagram* menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *activity diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol *Activity Diagram*

No.	Simbol	Keterangan
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan (<i>Decision</i>) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		Penggabungan (<i>Join</i>) merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.		Swimlane Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.
6.		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

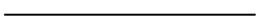
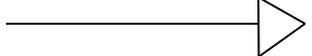
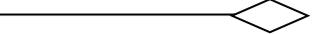
2.4.3 *Class Diagram*

Menurut Rosa dan Salahuddin (2019) *Class diagram* mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Class Diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Simbol *Class Diagram*

No.	Simbol	Deskripsi
1.		Kelas pada struktur sistem.

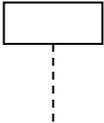
Tabel 2.3 Simbol *Class Diagram* (Lanjutan)

No.	Simbol	Deskripsi
2.	Antar Muka/Interface  Nama_Interface	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	Asosiasi / Association 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>symbol</i>
4.	Asosiasi Berarah / <i>Digunakan Association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>symbol</i> .
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	Ketergantungan / dependency 	Relasi antar kelas dengan makna ketergantungan antar kelas.
7.	Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

2.4.4 *Sequence Diagram*

Diagram rangkaian menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada eksekusi sebuah use-case atau operasi. Diagram ini mengilustrasikan bagaimana pesan terkirim dan diterima di antara objek dan dalam sekuensi pada Tabel 2.4.

Tabel 2.4 Simbol *Sequence Diagram*

No.	Simbol	Deskripsi
1.	<i>Object lifeline</i> 	Menggambarkan panjang kehidupan suatu objek selama scenario sedang di buat contohnya

Tabel 2.4 Simbol *Sequence Diagram* (Lanjutan)

No.	Simbol	Deskripsi
2.	<p style="text-align: center;"><i>Activation</i></p> 	Dimana proses sedang dilakukan oleh <i>object</i> atau <i>class</i> untuk memenuhi pesan atau perintah
3.	<p style="text-align: center;"><i>Message</i></p> 	Sebuah anak panah yang mengindikasikan pesan diantara objek. Dan objek dapat mengirimkan pesan ke dirinya sendiri

2.5 Pengujian *Black Box Testing*

Black box testing menurut Rosa dan Salahuddin (2019) yaitu pengujian perangkat lunak dari segi pendefinisian fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Pengujian yang dilakukan dengan membuat kasus yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji dilakukan harus dibuat dengan benar dan salah, seperti proses *login* “Jika user memasukan *username* dan *password* yang benar maka dapat *login* ?”.

2.6 Penelitian Terdahulu

Penelitian terdahulu bertujuan untuk mendapatkan bahan perbandingan dan acuan. Selain itu, untuk menghindari anggapan kesamaan dengan penelitian ini. Maka dalam kajian pustaka ini peneliti mencantumkan hasil-hasil penelitian terdahulu sebagai berikut:

1. Hasil Penelitian Oleh Nadiah, D Fitriati (2018)

Penelitian oleh Nadiah, D Fitriati dengan judul “Sistem Informasi Administrasi Sekolah Berbasis Web (Studi Kasus: SMK Negeri 16 Jakarta)”. Penelitian ini merupakan penelitian yang menggunakan Metode Waterfall. Hasil penelitian yaitu Perekapan data yang terkait dengan kegiatan administrasi di SMK Negeri 16 Jakarta menjadi lebih teorganisir dan semua data dapat terpusat dalam satu sistem, sehingga dapat meminimalisir terjadinya kerusakan atau kehilangan data

2. Hasil Penelitian Oleh Supianto, Atikah, Siti Marti’ah (2019)

Penelitian oleh Supianto, Atikah, Siti Marti’ah dengan judul “Perancangan Sistem Informasi Administrasi Sekolah Pada SMK Kesuma Bangsa 1 Depok”. Penelitian ini merupakan penelitian yang menggunakan metode grounded (grounded research). Hasil penelitian yaitu Sistem Informasi Administrasi Sekolah pada SMK Kesuma Bangsa 1 Depok, sehingga dapat mempermudah staff administrasi dalam melakukan pekerjaanya selain itu juga mempermudah siswa dalam proses pembayaran administrasi sekolah karena tidak perlu lagi membawa buku bayaran sekolah

3. Hasil Penelitian Oleh Dea Yuzistin, Dindon Fiqri Aji, Petrus Dwi Ananto Pamungkas (2016)

Penelitian oleh Dea Yuzistin, Dindon Fiqri Aji, Petrus Dwi Ananto Pamungkas dengan judul “Sistem Informasi Administrasi Siswa Berbasis *Website* Pada SMA Islam Putradarma Bekasi”. Penelitian ini merupakan penelitian yang menggunakan Metode Waterfall. Hasil penelitian yaitu Melalui perancangan sistem informasi administrasi berbasis *website* ini mampu meningkatkan efektifitas kinerja administrasi sekolah dan orang tua siswa dapat dengan mudah memantau kehadiran anaknya.