

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian mengenai pengembangan fitur yang akan dilakukan oleh penulis sudah banyak dilaksanakan sebelumnya oleh penelitian lainnya mengenai studi kasus yang bervariasi.

Menurut Oktarina & Hajjah (2019) yang merancang sistem pendaftaran sidang skripsi dengan algoritma genetika menyimpulkan bahwa dengan melakukan penjadwalan secara prosedural sering terjadinya bentrok dan keterlambatan penjadwalan. dengan itu peneliti melakukan pengembangan fitur penjadwalan pada aplikasi agar dapat memaksimalkan penjadwalan secara keseluruhan berdasarkan tanggal hari, jam, dan ruangan pelaksanaan agar tidak ada lagi terjadi permasalahan tabrakan jadwal yang biasa ditemukan dengan melakukan cara prosedural.

Menurut Randicha, Amien, & Fatchur (2016) yang merancang sistem pendaftaran sidang skripsi yang berbasis web melalui SMS ini menyimpulkan bahwa program studi elektro pada Universitas Diponegoro masih bekerja dengan menggunakan kertas yang ditempelkan pada papan pengumuman, Ketika pengumuman muncul mahasiswa harus hadir di kampus. Dengan itu dirancangnya sebuah website yang berisi informasi tentang sistem penjadwalan tugas akhir yang berfungsi untuk mempermudah pihak *administrator* saat penyusunan jadwal sidang.

Menurut Nathan & Scobell (2017) yang merancang sistem untuk sidang skripsi menyimpulkan bahwa melakukan proses pendaftaran dan penilaian secara manual sering menciptakannya sebuah kendala untuk melakukan *monitoring* status pendaftaran dan penilaian sidang. Maka itu peneliti melakukan perancangan aplikasi yang memiliki fitur pendaftaran sidang hingga ke penilaian sidang untuk mengatasi masalah tersebut dengan aplikasi berbasis website yang dapat dilakukan secara efektif dan efisien.

Menurut Chandra & Irawan (2019) yang merancang sistem untuk pelaksanaan sidang skripsi yang berbasis *mobile* menyimpulkan bahwa Universitas Budi Luhur masih ditemukan kekurangan dalam pelaksanaan sidang dikarenakan masih menggunakan metode tradisional yaitu melakukan penilaian sidang melalui

pengisian form secara manual yang kemudian dilakukan pengimputan ke dalam sistem menggunakan *workstation* yang jumlah tertentu. Hasil dari penelitian ini adalah melakukan implementasi sistem yang memiliki fitur pengisian penilaian dan menyampaikan notifikasi kepada dosen untuk sedia sesuai jadwal sidang dalam sisi efisiensi waktu yang ditentukan.

Menurut Hanafi, Sukarsa, & Agung Cahyawan Wiranatha (2017) yang melakukan penelitian “Pertukaran Data Antar Database dengan Menggunakan Teknologi API” menyimpulkan bahwa dengan menggunakan API saat melakukan pertukaran data adalah ide bagus yang bisa digunakan sebagai *tools* yang mendukung koneksi *database* dalam transaksi pertukaran data. API merupakan solusi yang sangat praktis untuk perusahaan kategori kecil dalam pertukaran data, dikarenakan hanya menggunakan sedikit internet.

Menurut Adi (2015) yang melakukan penelitian “*Scrum Method Implementation in a Software Development Project Management*” menjelaskan bahwa *Scrum* pertama kali diciptakan oleh Jeff Sutherland pada tahun 1993 dan dimaksudkan untuk menjadi metode pengembangan yang memiliki prinsip mengenai cara kerja metode tangkas. *Scrum* merupakan kerangka kerja responsif tambahan untuk pengembangan aplikasi. Fokusnya adalah pada "strategi, pengembangan produk fleksibel, holistik di mana tim pengembangan bekerja sebagai unit untuk mencapai tujuan bersama" sebagai saingan dari "pendekatan sekuensial tradisional". *Scrum* memiliki beberapa proses kompleks di mana ada banyak faktor yang dapat mempengaruhi hasil akhir pengembangan sistem.

Berdasarkan tinjauan pustaka yang telah terorganisasi, dengan itu penulis memperlihatkan tabel tinjauan pustaka yang terdapat pada tabel 2.1 terdahulu terhadap aplikasi yang dikembangkan.

Tabel 2.1 Tabel Hasil Penelitian

Judul Penelitian	Peneliti	Tahun	Hasil Penelitian
Perancangan Sistem Penjadwalan Seminar Proposal dan Sidang Skripsi Dengan Metode Algoritma Genetika	Oktarina, Hajjah	2019	Pengembangan fitur penjadwalan pada aplikasi untuk mengoptimalkan penjadwalan secara keseluruhan dengan adanya beberapa kriteria seperti tanggal, hari, jam dan

			ruangan operasi sidang berlangsung agar tidak ada terjadi permasalahan tabrakan jadwal yang biasa ditemukan melalui cara prosedural.
Sistem Penjadwalan Sidang Tugas Akhir Berbasis Web dengan Pesan Peningat Melalui SMS dan Aplikasi Pada Perangkat Android di Jurusan Teknik Elektro Universitas Diponegoro	Randicha, Amien, & Fatchur	2016	Perancangan sistem penjadwalan sidang yang berfungsi untuk membantu pihak administrator dalam menyusun penjadwalan sidang akhir.
Rancang Bangun Sistem Informasi Sidang Proposal Skripsi di Universitas Nusantara PGRI Kediri	Nathan, Scobell	2017	Pengembangan aplikasi pendaftaran sidang hingga penilaian sidang berbasis aplikasi website yang dapat dilakukan secara efektif dan efisien.
Sistem Informasi Berbasis <i>Mobile</i> untuk Meningkatkan Efisiensi dan Kualitas Layanan Pelaksanaan Sidang Tugas Akhir Studi Kasus: Universitas Budi Luhur	Chandra, Irawan	2019	Implementasi fitur penilaian sidang serta memberikan pengingat kepada dosen untuk hadir sesuai jadwal sidang, Tanpa menggunakan cara tradisional dengan mengisi form sidang.
<i>Scrum Method Implementation in a Software Development Project Management</i>	Permana	2015	<i>Scrum</i> merupakan kerangka kerja tambahan yang responsif untuk pengembangan aplikasi untuk proyek aplikasi dan mengelola pengembangan produk atau aplikasi.

2.2 Landasan Teori

2.2.1 Metodologi *Scrum*

Menurut (Adi, 2015) dalam penelitiannya yang berjudul “*Scrum Method Implementation in a Software Development Project Management*” mendeskripsikan bahwa *Scrum* bermula diciptakan oleh Jeff Sutherland di 1993 yang memiliki tujuan sebagai salah satu metode pengembangan yang mengikuti pedoman pola *agile*. *Scrum* memiliki manfaat dalam pengembangan sistem. Lebih

ditekankan pada strategis pengembangan sistem supaya memiliki keluaran produk yang lebih fleksibilitas, disini tim *developer* dapat bekerja sama dalam mencapai target dengan pendekatan sekuensial tradisional. *Scrum* memiliki komponen yang berpengaruh terhadap nilai akhir pengembangan perangkat lunak. Terdapat jumlah tahapan pada *scrum* yang dijelaskan sebagai berikut:

1. *Product Backlog*
Peneliti yang melakukan pengembangan sistem akan merangkum kebutuhan yang diperlukan dan melakukan penyusunan semua kebutuhan sistem yang diperlukan oleh permintaan dari pihak yang bersangkutan. Selepas target telah disesuaikan, Semua keperluan dan anjuran akan dipisahkan dalam bentuk poin kecil yang dimana setiap poin memiliki kebutuhan yang penting saat melakukan perkembangan.
2. *Sprint Planning*
Sprint Planning adalah tahapan yang harus dilakukan di setiap saat eksekusinya *sprint* baru. Pada tahapan berikut peneliti akan melakukan penyusunan proses yang akan dikerjakan dan yang harus diselesaikan dalam periode 1 *sprint*.
3. *Task Breakout*
Pada tahap ini penelitian melakukan penjadwalan yang diperlukan untuk dikerjakan dan perlu diselesaikan dalam setiap *sprint*.
4. *Sprint*
Sprint dimaksud sebagai batasan waktu yang memiliki pengertian operasi yang perlu dieksekusi dan lebih tertuju pada waktu penyelesaian dan hasil yang dilihat berdasarkan *product backlog*. Menurut (Budi, Siswa, & Abijono, 2016), Metodologi *developing scrum* ini mempunyai kelebihan yang dijelaskan sebagai berikut:
 - a. Sangat praktis saat melakukan manajemen dikarenakan sebelum melakukan perancangan telah melakukan dokumentasi terlebih dahulu secara detail.
 - b. Waktu pengembangan aplikasi lebih cepat daripada metode *waterfall*, dikarenakan pengembangan dapat dipisahkan dengan *sub module*.

- c. Pekerjaan disampaikan di awal proyek mulai.
- d. Selalu melakukan uji coba dan evaluasi sehingga sistem dapat dikembangkan lebih baik.
- e. Kesalahan dapat diminimalisir karena selalu melakukan identifikasi.
- f. Sangat bagus untuk proyek yang memiliki *scope* yang luas dikarenakan pengembang dapat mengembang secara tahap ke tahap dan ditunjukkan pada pengguna.
- g. Metode ringan sesuai proyek ukuran kecil.
- h. Lebih ditekankan pada produk akhir.
- i. Jaminan kualitas karena melakukan uji coba pada tahap pertahap.

2.2.2 Sistem Informasi

Menurut (Hidayatullah & Arief, 2016) sistem informasi adalah sistem yang terorganisasi yang menyambungkan spesifikasi dalam transaksi pengolahan, membantu operasi, bersifat manajerial dan kegiatan yang strategis dari suatu organisasi dan mempersiapkan pihak luar tertentu dengan laporan yang diperlukan.

Menurut (Rahmawati & Bachtiar, 2018), Komponen sistem informasi terdiri dari 5 yaitu sebagai berikut:

1. *Hardware*, termasuk berbagai perangkat yang dapat dilihat secara langsung dan dapat disentuh secara fisik seperti komputer dan mesin *printer*.
2. *Software*, adalah perintah khusus yang dimaksudkan untuk menginstruksikan komponen untuk melaksanakan tugasnya.
3. *Data*, adalah komponen informasi yang paling penting atau mentah yang diproses sehingga bermakna dan dapat memberikan informasi.
4. *Prosedur*, adalah aturan yang menghubungkan berbagai jenis perintah dan data untuk menentukan desain dan penggunaan sistem informasi.
5. *Manusia*, adalah pelaksana, mereka yang terlibat dalam kegiatan sistem informasi, seperti operator, manajer, dan sebagainya.

Menurut (Rahayu, Ginantaka, & WP, 2017) Sistem informasi dibangun sebab memiliki keunggulan besar untuk komponen sistem dalam suatu organisasi atau perusahaan. Manfaat yang dirangkum dari sistem informasi dapat diklarifikasi sebagai berikut:

1. Pengurangan biaya.
2. Menghindar dari kesalahan.
3. Peningkatan kecepatan aktifitas.

Keuntungan sistem informasi dalam bentuk *tangible benefits* dan *intangible benefits* yaitu:

1. Benefit yang dapat dilihat dalam bentuk pengurangan biaya operasi dan pengurangan kesalahan telekomunikasi.
2. Manfaat tidak berwujud dalam bentuk peningkatan layanan yang lebih baik, kepuasan kerja yang lebih tinggi di antara karyawan dan pengambilan keputusan yang terbaik.

2.2.3 Website

Menurut (Nofyat, Ibrahim, & Ambarita, 2018) *website* merupakan sekumpulan halaman web yang telah dipublikasikan pada jaringan internet dan memiliki domain atau disebut juga dengan URL (*Uniform Resource Locator*) internet dapat juga disebut sebagai jaringan yang sangat luas. Seperti komputer lokal dan jaringan komputer area. Internet juga menggunakan komunikasi *protocol* yang sama yaitu TCP/IP (*Transmission Control Protocol/Internet Protocol*).

Menurut (Wijaya, Supriyanti, & Saefullah, 2017) aplikasi berbasis web dirancang dengan bahasa pemrograman HTML, CSS dan PHP dengan penjelasannya yaitu sebagai berikut:

1. HTML

Menurut (Ahmad, Rini, & Wiratama, 2016) HTML merupakan suatu kode pemrograman berbasis *layer* pada halaman *website* dengan menggunakan *markup* yang digunakan untuk mengembangi kerangka *website* yang dapat dilihat oleh pengguna, Memperllihatkan informasi tertentu pada *website* melalui *browser* dalam bentuk ASCII dengan *output* sebuah tampilan pada kerangka *website* yang dikunjungi. Penulisan *markup* yang diterjemahkan ke dalam perangkat lunak, kata-kata diterjemahkan menjadi ASCII sehingga terbentuknya sebuah halaman utama/home dengan HTML. Berawal dari penulisan kode yang sebelumnya banyak dipakai didunia penerbitan dan percetakan yang disebut SGML(*Standard Generalized Markup Language*). HTML merupakan suatu standar yang sudah

digunakan untuk merancang aplikasi berbasis *website* di dunia pengembangan *website*. Sekarang HTML adalah standar internet yang diartikan penggunaannya oleh *World Wide Web Consortium (W3C)*.

HTML dirancang oleh sebuah organisasi Caillau dengan Berners-lee Rebert saat mereka bekerja di CERN pada tahun 1989 (CERN merupakan sebuah Lembaga penelitian fisika energi tinggi di Jenewa).

2. CSS

CSS adalah penulisan kode yang digunakan untuk mempercantik kerangka *layer website*. CSS digunakan untuk mempercantik kerangka HTML dan XHTML yang digunakan oleh pengguna melalui pengaksesan *website*, saat ini CSS bisa diimplementasikan pada segala dokumen berbasis XML, SVG dan XML. CSS dikembangkan untuk memperpisahkan halaman utama dengan tampilan dokumen yang dapat meningkatkan daya akses situs yang dikunjungi, Menyediakan lebih banyak fleksibilitas dan pengontrolan dalam spesifikasi dari sebuah *formatting* dan mengurangi kerumitan saat pengembang menulis kode dan struktur dari konten (Nugroho, Riza, & Hariyani, 2016).

3. PHP

Menurut (Suhartini, Muhamad Sadali, 2020), PHP merupakan sebuah pemrograman *script server side* yang disusun untuk pengembang dalam menciptakan sebuah *website*, php juga dipakai sebagai bahasa pemrograman umum dikarenakan aliran proses PHP berjalan di ruang lingkup *backend* yaitu proses yang berjalan tanpa terlihatnya oleh pengguna. PHP telah dicetus sebagai pemrograman *backend*, Sehingga PHP tidak terlihat di saat pengguna melakukan *inspect/debugging* pada *browser web*.

Menurut (Dewi, Surya, & Jumansyah, 2020), peranan penting sebuah *website* terdiri dari:

1. *Website* yang merangkum informasi pribadi masing-masing pengguna (*Personal Website*).
2. *Website* yang digunakan oleh setiap perusahaan yang bergerak dalam bidang bisnis (*Commercial Website*).

3. *Website* yang digunakan oleh pemerintah, pendidikan yang bertujuan memberikan pelayanan kepada pengguna (*Government Website*).
4. *Website* yang digunakan oleh sebuah organisasi yang bersifat *non profit* atau tidak bersifat bisnis (*Non Profit Organization Website*).

2.2.4 *Database*

Database adalah kumpulan-kumpulan data yang saling berelasi dan terbentuk dalam kerangka tabel yang tersusun secara terstruktur (Gunawan, 2016).

Database biasanya tersimpan di perangkat *hardware* komputer dan yang canggih saat ini melalui *cloud*. *Database* dapat digunakan melalui *software* tergantung jenis *database* yang digunakan, dengan menggunakan aplikasi tersebut pengguna

Database dapat memamanajemenkan data-data yang tersimpan pada *database*.

Database adalah sebuah komponen penting pada sebuah sistem informasi,

Database merupakan peranan penting dari sebuah sistem dalam menyediakan informasi-informasi untuk para pengguna yang menggunakan sistem informasi.

Kerangka *database* bisa dilakukan dengan menuliskan sebuah bahasa pemrograman SQL pada sebuah aplikasi khusus atau juga dengan cara apapun yang bisa digunakan agar mempermudah pengguna dalam manipulasi dan menampilkan data, *database* terbagi menjadi dua jenis yaitu:

1. *Database* hirarki

Database hirarki adalah sebuah data yang tersusun dalam bentuk akar pohon, susunan tersebut terdiri atas atribut-atribut penting yang memiliki potensi pengaruh relasional dan tidak terpisah antara atribut-atribut lain, jenis ini merupakan hubungan relasional satu atribut dengan atribut yang *multiple*.

2. *Database* relasional

Database relasional adalah sebuah data yang tersusun dalam bentuk tabel yang terdiri atas dua rumusan dan susunan yang terstruktur. Struktur susunan berupa 2D yang terdiri dari total *row* dengan *field*, Setiap *field* memiliki *column record* yang tersusun secara horizontal maka dengan itu jika semakin banyak data yang tersimpan ke *database*, *row* yang berisi *field* data akan semakin bertambah dan dapat dilihat semakin banyak barisan yang menurun ke bawah. Setiap data yang saling berelasi atau

berhubungan antar satu sama lainnya maka itu setiap data yang tersimpan ke *database* digambarkan sebuah satu barisan yang terdiri dari kolom-kolom tergantung dari jumlah banyaknya kolom yang dideskripsikan.

Database terdiri atas atribut-atribut yaitu sebagai berikut:

1. Data dideskripsikan sebagai suatu informasi yang tersimpan pada tabel dalam bentuk barisan yang terdiri atas kolom-kolom dan setiap tabel yang membentuk *database*.
2. Perangkat keras komputer disebut dengan *hardware* berfungsi untuk mendukung proses dalam mengolah data yang terdiri dari *central processor unit*, *terminal*, *memory* dan *disk drive*.
3. Perangkat lunak atau disebut juga dengan aplikasi yang berfungsi untuk menjalani proses komputer dan pengaturan spesifikasi sesuai kebutuhan dari aplikasi tersebut.
4. Pengguna yang memiliki akses dalam melakukan manajemen *database* adalah *user administrator*.

Database Management System atau disingkat sebagai DBMS merupakan suatu aplikasi khusus berdasarkan jenis *database* yang digunakan oleh pengguna, aplikasi *database* tersebut dirancang bertujuan agar pengguna *administrator database* dapat dengan mudah melakukan manajemen data dan pelacakan data berdasarkan data yang terdapat pada *database*.

Structure Query Language atau biasa dikenal dengan *SQL* merupakan suatu bahasa *non procedural* yang biasa dipakai untuk mengakses data relasional *database*. *SQL* merupakan bahasa *scripting database* ini biasanya dipakai untuk memecahkan permasalahan yang ada pada *database*, dan juga memiliki kelebihan untuk mengelolakan kumpulan data yang ada di *database* (Raharjo, Tullah, & Setiana, 2019). *SQL* berawal dari *International Standarts Organization* dikenal juga dengan *ISO* dan *American National Standards Institute* yang juga dikenal dengan *ANSI* ini yang sebelumnya dipanggil dengan *SQL86*

Structure Query Language yang biasa diberi nama sebagai *SQL* merupakan sebuah bahasa *non procedural* dalam melakukan transaksi pengaksesan data pada relasional *database*. *SQL* merupakan sebuah bahasa *scripting database* yang digunakan untuk menyelesaikan masalah pada *database* serta mempunyai

kelebihan dalam pengolahan data-data pada *database* (Raharjo et al., 2019). Standar SQL bermula didefinisikan oleh ISO (*International Standards Organization*) dan ANSI (*American National Standards Institute*) yang dikenal dengan sebutan SQL86, penggunaan *SQL* ini dapat memperoleh hal yang dijelaskan dengan berikut:

1. Perubahan jalur dari sebuah *database*.
2. *Create, update, dan delete* manfaat dari *database*.
3. *Transferring* data dari *database* yang dipilih ke *database* yang ingin dituju.

SQL ini juga dikembangkan sehingga dapat digunakan juga oleh *Personal Computer* (PC) serta dikembangkan juga agar dapat digunakan oleh akomodasi *database* lainnya yang lebih maju dimana contoh lainnya bisa dijelaskan sebagai berikut:

1. MS. Access yang dapat digunakan di PC, dimana MS Access ini sangat bersahabat untuk dipakai, hanya dengan menjalankan *SQL*, MS. Access ini bisa langsung merespon data yang disediakan.
2. MS. Query adalah *SQL* yang produknya dari porposional yang disediakan oleh *microsoft windows*, yaitu MS. Visual Studio yang didalamnya terdapat *Visual Basic* dan *Visual C++* dimana dibutuhkan format ODBC untuk dapat dibaca oleh *database*.
3. *Oracle* dimana aplikasi ini hanya sering ditemui di *database* yang didalamnya terdapat data yang sangat besar dan biasanya sering dipakai di perusahaan.

2.2.5 *Unified Modeling Language (UML)*


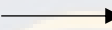
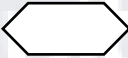







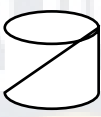
UML mendeskripsikan suatu istilah yang biasanya dipakai untuk menjelaskan alur dari pembuatan sebuah sistem atau istilah yang samanya yaitu seperti *flowchart* (Liana, Sutardi, & Muchlis, 2018).

Untuk menjelaskan bagaimana mendesain sebuah sistem ini diperlukan sebuah gambaran mudah yang menjelaskan dari tahap ke tahap merupakan pengertian dari *UML* ini. *UML* inilah yang membantu menjelaskan dengan alur yang mudah dipahami dimana didalamnya menjelaskan dengan bantuan gambar, pembangunan dan dokumentasi dari *software*.

1. *Flowchart*

Menurut Hidayat (2018) *Flowchart* dapat diartikan dengan kumpulan simbol, gambar atau kalimat yang masing masing mempunyai tujuan dan fungsi tersendiri bahasa pemrograman yang disambungkan dari satu dengan yang lain secara bertahap yang akhirnya menghasilkan suatu penjelasan langkah awal sampai pekerjaan suatu algoritma.


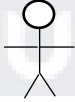


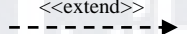
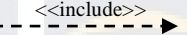
Tabel 2.2 Notasi *Flowchart*

Simbol	Nama	Keterangan
	Terminator	Fase proses pemulaian/pengakhiran Program
	Garis Alir (Flow Line)	alur berjalanya sistem
	Preparation	Proses memulai/kontribusi harga awal
	Process	Proses Kalkulasi/Proses pengolahan data
	Input/Output Data	Proses pemasukan/pengeluaran data, <i>argument</i> , informasi
	Predefined Process (Sub Program)	Tahap mulainya sub program/proses menjalani sub program
	Decision	Pemilihan data yang memberikan pilihan untuk alur selanjutnya
	On Page Connector	Penghubung bagian flowchart yang terletak didalam satu halaman
	Off Page Connector	Penghubung bagian flowchart yang terletak pada halaman berbeda
	Disk/On-Line Storage	Pemasukan berasaral dari disk atau pengeluaran yang disimpan pada disk.
	Magnetik Disk	Proses Pemasukan/Pengeluaran Menggunakan Disk Magnetik(Database).

2. Use Case Diagram

Use Case Diagram merupakan semacam model kelakuan (*behavior*) sistem informasi yang akan dikembangkan. *Use Case* menjelaskan interaksi antara satu dengan yang lainnya yang akan dirancang oleh pengembang. *Use case* ini biasanya dipakai untuk mengetahui fungsi dari sistem informasi dan juga mengetahui pihak yang dapat mengakses fungsi tersebut (Mimin, 2016). *Use case* memiliki bahasa sendiri yang dapat dilihat di tabel 3

Tabel 2.3 Notasi Use Case Diagram


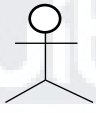



Simbol	Nama	Keterangan
	<i>User Case</i>	Tingkah laku yang diinginkan dari aplikasi atau komponen dari aplikasi.
	<i>Actor</i>	<i>Actor</i> adalah pemakai sistem. <i>Actor</i> tidak ada batas hanya manusia, jika sebuah sistem berkomunikasi yang menghasilkan pemasukkan dan pengeluaran bisa disebut actor.
	<i>Association</i>	Asosiasi dipakai untuk menghubungkan actor dengan use case.
	<i>Generalization</i>	<i>use case</i> bawahan yang bisa warisi perilaku dan nilai <i>parent use case</i>
	<i>Extend</i>	Dasar <i>Use Case</i> yang secara tersirat menggabungkan perilaku <i>use case</i> lainnya.
	<i>Include</i>	Dasar <i>Use Case</i> yang secara eksplisit menggabungkan perilaku <i>use case</i> lainnya.

3. Sequence Diagram

Sequence Diagram memiliki peran untuk menjelaskan tingkah laku suatu objek dari *use case* dimana menjelaskan kapan suatu objek tersebut dieksekusi dan menjelaskan pesan yang dikirim kepada objek dan diterima juga oleh objek tersebut. Sehingga agar dapat penjelasan suatu sekuen diagram ini harus diketahui oleh objek yang saling berhubungan dari *use case* dan juga dimana tahap yang memiliki tingkatan ini akan dimulai untuk menjadi objek itu.

Terdapat banyak sekuan diagram yang harus dijelaskan sehingga semakin banyak *use case* yang dijelaskan maka sekuen diagram yang harus dibuat juga makin banyak (Mimin, 2016).


Tabel 2.4 Notasi *Sequence Diagram*




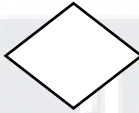
Simbol	Nama	Keterangan
	Objek	Object merupakan bawaan dari sebuah class dan dituliskan susunan secara horizontal.
	Actor	Actor juga bisa berkomunikasi dengan object, maka actor juga bisa diartikan sebagai kolom.
	Lifeline	Lifeline menunjukkan keberadaan pada sebuah object dalam basis waktu. Notasi untuk lifeline adalah garis yang putus putus vertical yang ditarik dari sebuah objek
	Activation	Activation dinotasikan sebagai sebuah kotak persegi empat yang digambarkan pada sebuah lifeline. Menunjukkan sebuah objek yang akan melakukan suatu aksi.
	Message	Message, digambarkan dengan anak anak panah berhorizontal diantara activation. Message menunjukkan komunikasi diantara objek.

4. *Activity Diagram*

Menurut Maimunah, Ilamsyah (2016) *Activity Diagram* adalah berbagai macam jalur aktivitas dari suatu sistem yang sedang beraktivitas, dimana setiap jalur tersebut akan menunjukkan jalur yang baru yang pada akhirnya menunjukan sebuah jalur penyelesaian.

Tabel 2.5 Notasi *Activity Diagram*

Simbol	Nama	Keterangan
	<i>Start Point</i>	Simbol diagram mulainya aliran aktivitas.

	<i>End Point</i>	Simbol diagram berakhirnya aliran pada sebuah aktivitas.
	<i>Control Flow</i>	Garis penyambung arah pada simbol notasi aktivitas diagram.
	<i>Action</i>	Simbol notasi yang digunakan untuk menjelaskan sebuah aksi yang dilakukan.
	<i>Decision</i>	Simbol dimana sebuah proses melakukan keputusan dalam pilihan jalur yang akan dituju.

2.2.6 Framework

Framework adalah sebuah *tools* yang dikembangkan oleh tim pengembang yang berisi bahasa pemrograman yang sudah didesain, bertujuan untuk membantu pengembang lain yang sedang merancang aplikasi dengan mudah dalam memecahkan masalah yang dihadapinya dikarenakan program yang didesain dapat digunakan langsung pada kegiatan hari-hari yang dilakukan oleh pengembang seperti pengaksesan *database*, penggunaan *variable* dan fungsi yang sudah tersedia dari *frameworknya*.

Framework dideskripsikan sebagai suatu komponen yang dapat dipakai oleh pengembang langsung tanpa harus menulis logika ulang dari awal yang tinggal hanya memanggil nama fungsi tersebut yang tersedia dengan mudah hingga menghemat waktu pengembang yang menggunakan *framework* tersebut. Contohnya saat pengembang membuat sebuah website yang memiliki fitur *asynchronous* setiap kali pengguna harus menggunakan XMLHttpRequest, Dengan menggunakan *framework* tools Xajax dapat menghemat waktu pengguna yang hanya tinggal dipanggil dan mudah dimengerti berbasis *asynchronous php*. Masing-masing *framework* memiliki kelebihan yang dapat dipilih sesuai keinginan pengembang (Destiningrum & Adrian, 2017). Poin tinggi yang dirangkum disaat pengembang menggunakan *framework* saat melakukan perancangan ialah:

1. *Framework* tersedia fungsi khusus yang siap dipanggil jika dibutuhkan saat mengembang.
2. *Framework* rata rata sudah menggunakan konsep *Model View Controller*.

3. Penulisan kode *framework* yang digunakan lebih singkat daripada menggunakan Bahasa program biasa.

Institut sains & teknologi AKPRIND Yogyakarta menjelaskan bahwa pemrograman dengan memakai *framework* akan lebih *user friendly* dan dapat mempermudah mahasiswa mengoperasikan data lebih cepat dari sebelumnya (Said, Setyaningsih, & Harmastuti, 2017).

2.2.7 Model View Controller (MVC)

Model View Controller biasanya disebut juga dengan MVC merupakan tips pemrograman yang ditemukan oleh Smalltalk (Trygve Reenskaug) digunakan untuk melakukan enkapsulasi data melalui bagian *Model*, Alur bisnis pada program prosesnya berjalan pada bagian *controller* dan tampilan yang dilihat oleh pengguna saat menggunakan aplikasi adalah *view* (Pastima & Kasnady, 2016).

Komponen MVC terbagi menjadi 3 bagian yaitu:

1. *Model*

Model memiliki fungsi dalam melakukan pengolahan data yang akan berkomunikasi dengan database dalam melakukan transaksi pengambilan data dan penyimpanan data.

2. *View*

View berperan sebagai lapisan interaksi dengan *user* dalam bentuk respon balik kepada *user* saat menggunakan sistem.

3. *Controller*

Controller berfungsi sebagai penyedia alur program dan transisi proses antar lapisan, yang menjadi penghubung antara komponen *model* dan *view* dalam penyelesaian alur pada suatu proses.

2.2.8 Web Service

Web Service ini merupakan suatu operasi yang mempunyai manfaat untuk supporting transaksi dari suatu objek dari satu ke yang lain melewati jaringan. *Web service* didalamnya terdapat yang namanya *interface* dimana berisi format yang bisa dibaca oleh aplikasi lainnya (Sibagariang, 2016). *Web Service* adalah kumpulan suatu data, *software* yang dapat di akseskan dengan *connection remote* diaman harus ada sebuah perantara yang sesuai. perbedaan antara *web service* dan

website adalah website diakses melalui URL dan mengembalikan tampilan halaman *browser* tetapi web *service* URL digunakan untuk mengembalikan sebuah data yang diinginkan atau menyimpan data dari mesin yang berbeda (Ariani & Endra, 2018).

2.3 Perancangan Sistem

2.3.1 Framework Bootstrap

Pencipta *Bootstrap* ini berawal pada tahun 2011 oleh dua orang yang bernama Mark Otto dan Jacob Thornton dimana kedua orang itu juga seorang developer media sosial yang terkenal bernama Twitter. Tujuan utama dari *Bootstrap* ini untuk membantu dan mempermudah pekerjaan para developer di Twitter, akan tetapi menjadi berkembang dan terkenal oleh para developer. *Bootstrap* ini juga digunakan untuk membuat sebuah aplikasi yang bersifat web yang sederhana, mudah dipahami dan juga bisa dipakai siapa saja tanpa dikenakan biaya.

Bootstrap ini dapat terbagi menjadi CSS dan HTML dimana kedua ini memiliki fungsi untuk membentuk suatu *layout, grid, form, table, typography*, dan *navigation*. *Bootstrap* juga terdapat *plugin JQuery* yang memiliki kelebihan untuk menciptakan suatu komponen komponen seperti *User Interface* yang menarik, kemudian *transitions, modal, dropdown, scrollspy, tooltip, tab, popover, alert, button* dan *carousel*. *Bootstrap* ini digunakan oleh para developer untuk menciptakan sebuah website secara responsif, fleksibel dan dapat mensupport beberapa *browser* seperti Chrome, Firefox, Safari, Opera, dan Internet Explorer (Wahyudi, Dewi, & Wibowo, 2017). *Bootstrap* ini ada banyak kelebihan terutama ketika kita ingin mendesain situs kita dimana kelebihan nya dijelaskan sebagai berikut:

1. *User Friendly*, didalam nya terdapat banyak *library* kode sehingga pengguna dapat membuat barang secara bervariasi yang akan digunakan pada website.
2. Fleksibel, terdapat izin pengguna yang akan menyesuaikannya dengan para pengguna sesuai dengan kebutuhan yang dibutuhkannya.

3. Desain, pada bagian design tampilan *website* dia memiliki berbagai tampilan yang sangat menarik dan dapat juga dibuat secara otomatis sesuai dengan keinginan pengguna.

2.3.2 Framework JQuery

Framework javascript ini merupakan suatu *Library* dari *javascript* yang digunakan untuk membentuk suatu rangkuman rangkuman kode yang dihasilkan di *JavaScript*, sehingga memberikan sebuah kemudahan dalam mengkodekan. Terdapat *quotes* “*write less, do more*” di aplikasi ini, *Jquery* pertama kali dipublikasikan oleh John Resig pada tahun 2006. Sebelumnya *Jquery* ini tidak cuma berbentuk *framework Javascript*, kemudian dikembangkan dan menjadi memiliki berbagai kelebihan yang tersedia (Wahyudi et al., 2017) dimana kelebihan yang dijelaskan dengan sebagai berikut::

1. *jQuery* ini didalamnya sudah menyediakan berbagai fungsi yang sudah tersedia didalamnya jadi tidak diperlukan menulis kembali pengkodekan secara berulang atau secara panjang sehingga menghemat waktu.
2. *jQuery* dilengkapi beberapa fitur fitur yang sangat bermanfaat untuk memanipulasikan *event* dari suatu elemen elemen yang berada di dokumen sebuah HTML.
3. Untuk designer atau bagian animasi sangat berguna karena mempermudah pembuatan animasi dan AJAX.

2.3.3 Xampp

XAMPP adalah sebuah aplikasi *web server apache* dimana didalamnya terdapat server MySQL dan kemudian juga mensupport bahasa mengkodekan dari PHP sehingga dapat menciptakan website yang lebih bagus. XAMPP ini biasanya dipakai pada sistem operasional secara fleksibel. Untuk linux memiliki cara penginstalan yang memakai *command line* dan kemudian untuk windows dan MacOS ini memiliki cara penginstalan yang lebih berbeda dengan linux, mereka menggunakan *interface free* sehingga disini lebih bersahabat dalam penggunaan XAMPP. Disini dijelaskan oleh peneliti (Erinton et al., (2017), ada terdapat 3 komponen komponen primer yang ada didalamnya yaitu *web server Apache*, PHP dan MySQL. Aplikasi Software ini terdapat banyak manfaat dan kemudahan ketika

kita sudah memilikinya karena didalamnya sudah disediakan apache yang dapat dijalankan pada web *server*, *scripting engine* PHP merupakan bahasa program yang digunakan untuk membuat *program* dan MySQL sebagai *database* untuk menampung dan memanajemen data berdasarkan aplikasi yang digunakan, jadi disini kita tidak perlu lagi menginstall masing masing komponen lagi.

2.3.4 MySQL

MySQL ini menjelaskan suatu *database* yang akan menghubungkan penulisan dengan bahasa PHP dengan bantuan SQL *query* serta *escape character* yang tentunya bahasanya sama dengan PHP. MySQL *client* ini lebih ke tampilan klien yang akan mempermudah pengguna tersebut untuk mengakses dengan bantuan *username* dan kata sandi, Sedangkan MySQLD merupakan sisi server dimana proses *service MySQL* dijalankan agar sisi *client* dapat digunakan untuk autentikasi pada server MySQL yang dijalankan. Kemudian terdapat juga kelebihan yang bisa kita ambil dari *database* MySQL ini salah satunya adalah mudah dan praktis dalam mentransfer data yang didalamnya dipakai jutaan pengguna dalam waktu bersamaan, kelebihan yang dapat diambil dari SQL adalah bahasa softwarenya karena bahasa *query* standarnya. MySQL bisa disupportkan dengan *open source software* seperti MySQL Workbench, *PHPMyAdmin*.

2.3.5 Framework Laravel

Laravel adalah suatu *Framework PHP* dimana semakin sederhana dan mudah digunakan untuk mendesain. Laravel dipublikasikan dengan lisensi MIT dimana yang disiapkan diGithub. Sama juga dengan *framework* PHP lainnya. Laravel ini diciptakan lebih tekankan pada metode MVC. Laravel menyediakan *tools command line* yang dikenal dengan *artisan* ini yang bisa dipakai untuk *bundle packaging* dan *installation bundle*. Perolehan survei dari sitepoint.com di Desember 2013 mengenai popularitas *framework* PHP , Laravel mendapat posisi kesatu (Erinton et al., 2017). Laravel ini juga memiliki fungsional seperti paket modular dan *dedicated dependency manager*. Terdapat juga berbagai manfaat menurut Faruqi et al., (2018) yang dijelaskan sebagai berikut:

1. Laravel memiliki peran penting untuk memberikan batasan untuk setiap objek yang dibantu *advanced mechanism query builder*.

2. Laravel memiliki fungsi *auto loading* jadi disini tidak lagi diperlukan waktu untuk perbaikan secara satu persatu alias *manually*
3. *Framework* laravel memiliki fungsi *migration database* ini yang berguna dalam mempermudah pekerjaan pengembangan aplikasi jika dalam pengembangan sebuah aplikasi tersebut dilakukan dan dikembangkan oleh lebih dari satu orang sehingga tidak berantakan dan juga terkontrol secara aman.