

BAB II TINJAUAN PUSTAKA

2.1 Studi Literatur

Pada penelitian terkait ini, mengkaji penelitian mengenai sistem aplikasi yang dilakukan oleh penelitian-penelitian sebelumnya yang dapat menjadi dasar dari penelitian yang dilakukan. Beberapa penelitian mengenai sistem pendukung keputusan diantaranya adalah sebagai berikut:

Roviana H. Dai, Lillyan Hadjaratie & Nuzran Firmansyah Bouti (2017) melakukan penelitian dengan judul “Rancang Bangun Aplikasi *e-report* Pengaduan Masyarakat *Design Public Complaint e-report Application*”. Dalam penelitian tersebut dibuat sebuah aplikasi sebagai *early warning system, reminder*/pengingat sekaligus membantu dalam monitoring laporan baik oleh internal Ombudsman maupun masyarakat sebagai Pelapor berbasis *web*. Pembuatan penelitian ini mengimplementasikan *SMS gateway*, sehingga informasi terkini dari penanganan kasus dari pelapor dapat mengecek perkembangan laporan. Aplikasi ini memanfaatkan fasilitas SMS yang di ponsel untuk mengirimkan perkembangan laporan. Dalam pemodelan sistemnya menggunakan Diagram Alir Data.

Wali Ja'far Shudiq (2017) melakukan penelitian dengan judul “Desain Program Sistem Informasi Pengaduan Masyarakat pada Kecamatan Kademangan Terintegrasi *SMS Gateway*”. Dalam penelitian ini membuat sebuah sistem yang dapat melakukan pengaduan dengan melakukan SMS dan memperoleh informasi pada kantor kecamatan pademangan. Aplikasi ini menggunakan *SMS Gateway*. Pemodelan sistem yang digunakan adalah model *waterfall*.

Ekalaya Bayu Putra, Dias Muhamad Ichsan, Elidjen (2013) melakukan penelitian dengan judul “Aplikasi dan Perancangan Sistem Pengaduan Masyarakat terhadap PEMKAB Merangin Berbasis *Web*”. Aplikasi menghasilkan aplikasi pengaduan masyarakat yang dapat memudahkan dalam melakukan pengaduan serta membantun pemerintah untuk mengawasi jalannya proses pengaduan tersebut. Aplikasi ini berbasis *website*. Pembangunan aplikasi menggunakan metode *waterfall*.

Yogi Isro' Mukti (2019) melakukan penelitian berjudul “Sistem Informasi Monitoring Kesehatan Masyarakat Berbasis *Web* Menggunakan Metode *Unified*

Modelling Language”. Aplikasi menghasilkan penyedia sistem informasi *monitoring* kesehatan masyarakat berbasis *web*, guna mengetahui sebaran penyakit yang sedang berkembang di masyarakat seperti *pneumonia* atau radang paru-paru. Pembangunan aplikasi menggunakan UML, dengan bahasa pemrograman PHP dan database *MySql*.

Berdasarkan dari empat penelitian diatas, munculah sebuah ide yang sama dengan penelitian tersebut yaitu dengan menerapkannya di Komisi Perlindungan dan Pengawasan Anak Daerah Kalimantan Barat untuk masyarakat Kalimantan Barat. Aplikasi yang di bangun dengan *framework Laravel*, dengan bahasa pemrograman PHP dan *database* menggunakan *MySql*. Pemilihan *Laravel* pada aplikasi *monitoring* agar proses pengaduan dan penerimaan data antar *client* dan *server* dapat berlangsung secara *update*, terus-menerus. Dengan adanya aplikasi ini pelapor dapat memperoleh informasi progres pengaduan yang dikirim oleh admin melalui *email* dan pelapor dapat meng-*upload* bukti setelah pengaduannya diverifikasi agar menjadi pertimbangan dari pihak KPPAD. Data yang dikirim berupa teks serta dilengkapi dengan keterangan.

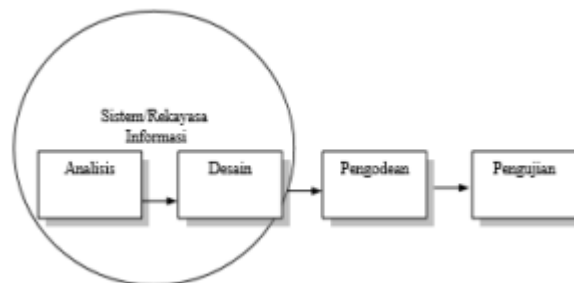
Tabel 2.1 Perbandingan Penelitian

No.	Penulis	Judul	Keterangan
1.	Roviana H. Dai, Lillyan Hadjaratie & Nuzran Firmansyah Bouti (2017)	Rancang Bangun Aplikasi e-report Pengaduan Masyarakat Design Public Complaint e-report Application	<ul style="list-style-type: none"> - Aplikasi ini menggunakan SMS Gateway, sehingga informasi terkini dari penanganan kasus dari pelapor dapat mengecek perkembangan laporan. - Memanfaatkan fasilitas sms yang di ponsel untuk mengirimkan perkembangan laporan.
2.	Wali Ja'far Shudiq (2017)	Desain Program Sistem Informasi Pengaduan Masyarakat pada Kecamatan Kademangan Terintegrasi SMS Gateway	<ul style="list-style-type: none"> - Membuat sebuah sistem yang dapat melakukan pengaduan dengan melakukan SMS dan memperoleh informasi pada kantor kecamatan pademangan.
3.	Ekalaya Bayu Putra, Dias	Aplikasi dan Perancangan Sistem Pengaduan	<ul style="list-style-type: none"> - Aplikasi menghasilkan aplikasi pengaduan masyarakat yang dapat memudahkan dalam

No.	Penulis	Judul	Keterangan
	Muhamad Ichsan, Elidjen (2013)	Masyarakat terhadap PEMKAB Merangin Berbasis <i>Web</i>	melakukan pengaduan serta membantun pemerintah untuk mengawasi jalannya proses pengaduan tersebut. Aplikasi ini berbasis <i>website</i> .
4.	Yogi Isro' Mukti (2019)	Sistem Informasi Monitoring Kesehatan Masyarakat Berbasis <i>Web</i> Menggunakan Metode <i>Unified Modelling Language</i>	- Aplikasi menghasilkan penyedia sistem informasi <i>monitoring</i> kesehatan masyarakat berbasis <i>web</i> , guna mengetahui sebaran penyakit yang sedang berkembang di masyarakat seperti <i>pneumonia</i> atau radang paru-paru.
5.	Dani Kusmeyadi (2022)	Aplikasi Monitoring Pengaduan Kasus Anak pada KPPAD	- Aplikasi yang di bangun dengan <i>framework Laravel</i> . - <i>Package library Laravel</i> pada aplikasi monitoring ini untuk proses komunikasi antar <i>client</i> dan <i>server</i> dapat berlangsung secara terus-menerus.

2.2 Waterfall

Model *Waterfall* adalah model menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, dan pengujian (Sukamto & Shalahuddin, 2018). Berikut gambar dari model *waterfall* dalam SDLC.



Sumber : Sukamto & Shalahuddin (2018)

Gambar 2.1 Ilustrasi model *waterfall*

1. Analisis

Melakukan analisis kebutuhan perangkat lunak, fungsi dan proses dari *web* yang dibuat, pengidentifikasian kendala dalam pembuatan *web*, menganalisis keandalan, kelemahan, dan teknologi yang dipakai.

2. Desain

Desain perangkat lunak adalah proses beberapa tahapan langkah pada rancangan pembuatan program perangkat lunak meliputi struktur data, arsitektur perangkat lunak, representasi antarmuka dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahapan analisis kebutuhan ke representasi rancangan agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Pada tahap ini, hasil dari desain perangkat lunak yang telah ada didokumentasikan.

3. Pengkodean

Desain harus ditranslasikan kedalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai desain yang telah dibuat pada tahap desain. Atau tahapan penulis membuat program dengan bahasa program seperti PHP, HTML, CSS dan lain-lain.

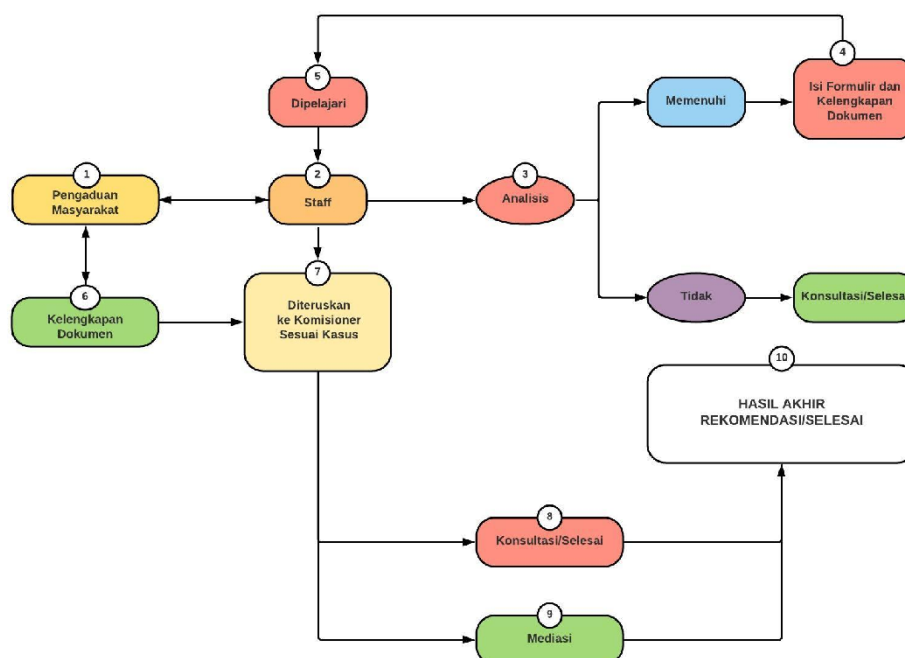
4. Pengujian

Tahapan ini penulis melakukan pengujian terhadap program yang telah dibuat untuk mengetahui kekurangan dari program tersebut. Seperti validasi halaman *login*, apakah sesuai dengan harapan.

2.3 Skema Pelaporan Kasus pada KPPAD

Laporan adalah suatu cara komunikasi di mana penulis menyampaikan informasi kepada seseorang atau suatu badan karena tanggung jawab yang dibebankan kepadanya (Keraf, 2004: 324).

Berikut adalah skema melapor di KPPAD Kalimantan Barat:



Sumber : Kantor KPPAD Kalimantan Barat

Gambar 2.2 Skema pelaporan KPPAD Kalimantan Barat

2.4 Monitoring

Monitoring sebagai suatu proses mengukur, mencatat, mengumpulkan, memproses dan mengkomunikasikan informasi untuk membantu pengambilan keputusan manajemen program/proyek (Clayton & Francoise, 1983).

Monitoring adalah suatu proses pengumpulan dan menganalisis informasi dari penerapan suatu program termasuk mengecek secara reguler untuk melihat apakah kegiatan/program itu berjalan sesuai rencana sehingga masalah yang dilihat/ditemui dapat diatasi (Nalahudin, 2010).

Monitoring merupakan penilaian secara terus menerus terhadap fungsi-fungsi program dalam hal *input*-an data dengan tujuan yang telah direncanakan.

2.5 Pengaduan

Pengaduan masyarakat merupakan suatu sumber informasi yang sangat penting bagi upaya-upaya pihak penyelenggara pelayanan untuk memperbaiki kesalahan yang mungkin terjadi, sekaligus secara konsisten menjaga dan meningkatkan pelayanan yang dihasilkan agar selalu sesuai dengan standar yang telah ditetapkan. Manajemen pengaduan adalah suatu proses kegiatan untuk menampung, mencatat, menelaah, menyalurkan, mengkonfirmasi, mengklarifikasi,

memberikan alternatif solusi kepada pengadu, mendokumentasi dan mensosialisasikan hasil pengelolaan pengaduan kepada masyarakat (Fanar Syukri, 2009).

Pengaduan, adalah laporan yang mengandung informasi atau indikasi terjadinya penyalahgunaan wewenang, penyimpangan atau pelanggaran perilaku yang dilakukan oleh aparat pengadilan, yang berasal dari masyarakat, anggota; instansi peradilan, instansi di luar pengadilan, maupun dari media massa dan sumber-sumber informasi lain yang relevan.

2.6 Anak

Anak adalah seorang yang dilahirkan dari perkawinan antara seorang perempuan dan laki-laki dengan tidak menyangkut bahwa seseorang yang dilahirkan oleh wanita meskipun tidak pernah melakukan pernikahan tetap dikatakan anak.

Anak menurut Komisi Perlindungan dan Pengawasan Anak Daerah merujuk pada Pasal 1 Ayat 1 Undang-Undang Nomor 35 Tahun 2014 berbunyi “Anak adalah seorang yang belum berusia 18 (delapan belas) tahun, termasuk anak yang masih dalam kandungan”.

2.7 Jenis-jenis Kasus Anak pada KPPAD

Arti kata kasus menurut KBBI, keadaan yang sebenarnya dari suatu urusan atau perkara; keadaan atau kondisi khusus yang berhubungan dengan seseorang atau suatu hal; soal; perkara.

Tindakan kekerasan pada anak merupakan sebuah perilaku tindakan penganiayaan yang mana dilakukan oleh orang tua, wali, maupun orang lain kepada anak-anak. Bentuk kekerasan yang terjadi dapat berupa fisik, psikologis, seksual, hingga secara sosial.

Jenis-jenis kasus anak yang terdapat di KPPAD.

- | | |
|---------------------------|--------------------------------|
| 1. Kekerasa Fisik. | 6. Hak Kuasa Asuh dan |
| 2. Kekerasa Psikis. | Penelantaran, |
| 3. Kejahatan Seksual. | 7. Perlindungan Anak. |
| 4. Anak Berhadapan Hukum. | 8. Penculikan dan Anak Hilang. |
| 5. Trafficking. | 9. Napza dan HIV/Aids. |

10. Pekerja Anak.

12. Hak Sipil.

11. Pornografi.

13. Anak Berkebutuhan Khusus

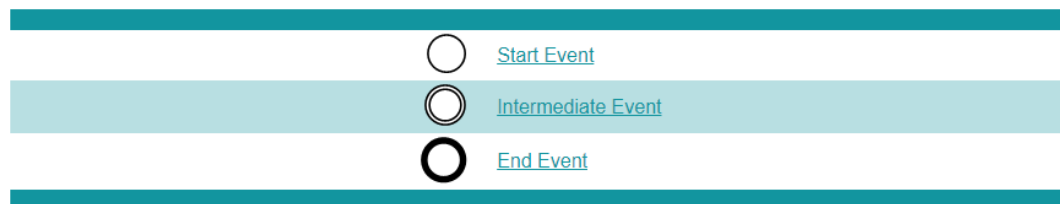
Sumber : Kantor KPPAD Kalimantan Barat

2.8 Business Process Modeling Notation (BPMN)

Business Process Modeling Notation (BPMN) adalah suatu notasi standar yang dapat berupa ikon atau gambar untuk digunakan di dalam pemodelan proses bisnis (Jeston et al., 2020).

Proses bisnis mengacu kepada sebuah metode di mana pekerjaan, dikelola, dikoordinasikan dan difokuskan untuk menghasilkan produk atau jasa yang bernilai. Proses bisnis juga merupakan arus kerja dari bahan baku, informasi dan pengetahuan yang kesemuanya itu adalah seperangkat aktivitas (Kenneth & Jane, 2015).

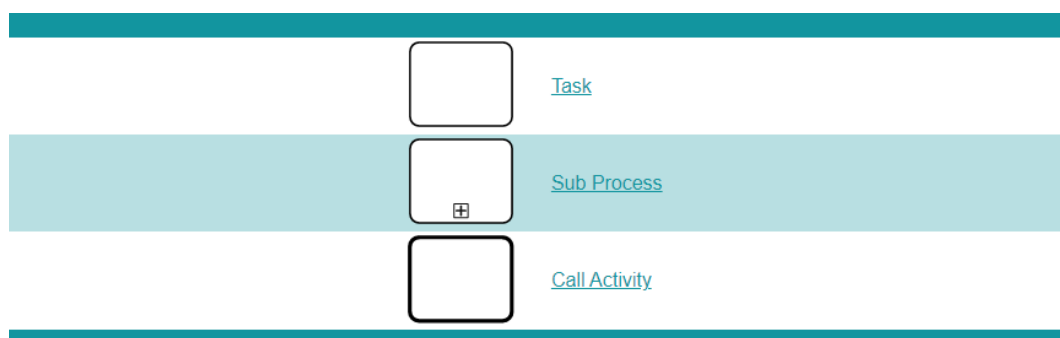
Deskripsi notasi-notasi BPMN dapat dilihat pada Gambar 2.3, dibawah ini.



Sumber : www.bpmn.org

Gambar 2.3 Notasi *event* pada BPMN

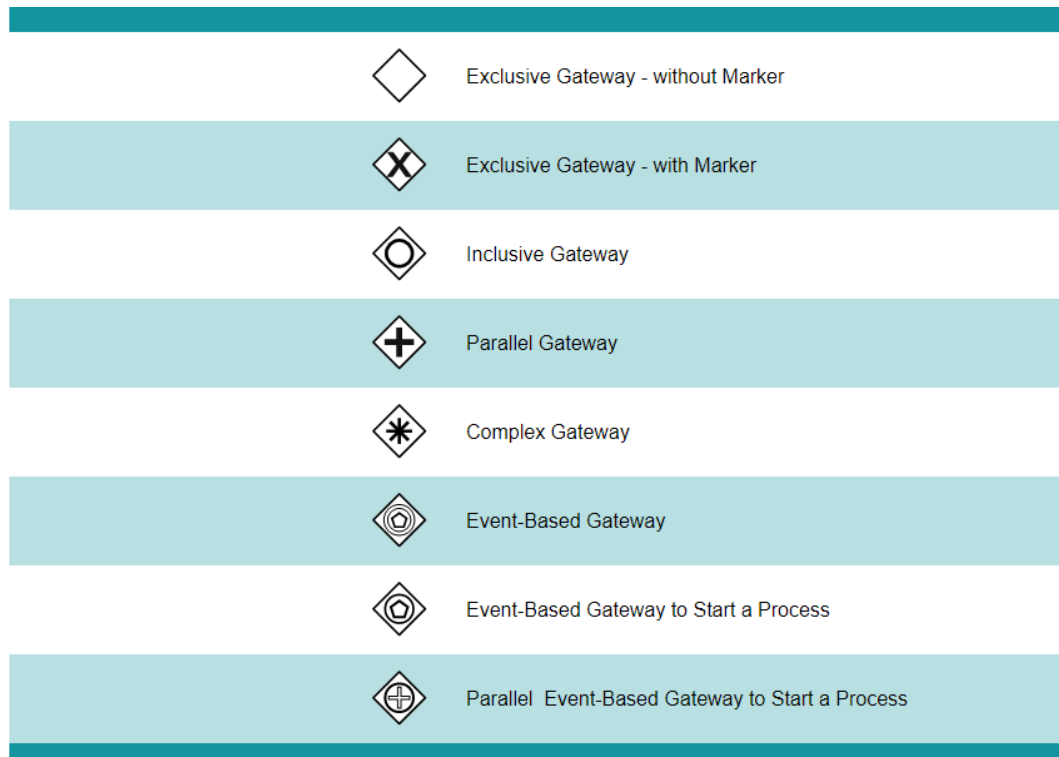
Sebuah Event disimbolkan dengan sebuah lingkaran yang menunjukkan terjadinya suatu proses. Sebuah Event mempengaruhi aliran proses dan biasanya mempunyai penyebab (trigger) atau sebuah dampak (result). Ada 3 jenis Event yaitu Start, Intermediate dan End.



Sumber : www.bpmn.org

Gambar 2.4 Notasi *activity* pada BPMN

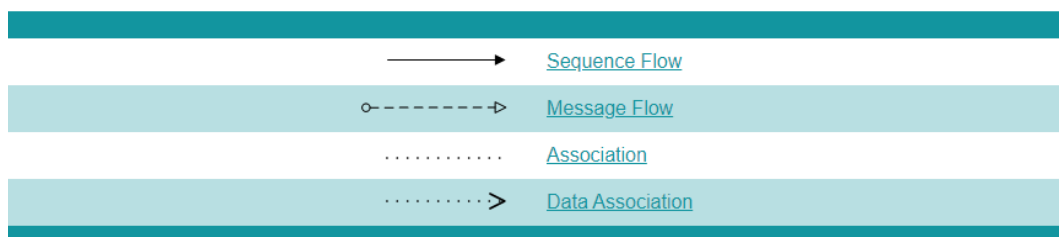
Sebuah Activity disimbolkan dengan sebuah persegi dengan sudut yang bulat. Pada Activity dapat berupa sebuah Task yang bersifat tunggal ataupun Sub-process yang terdiri dari beberapa aktivitas. Sebuah Sub-process disimbolkan dengan tanda tambah di bagian tengah bawah persegi.



Sumber : www.bpmn.org

Gambar 2.5 Notasi *gateway* pada BPMN

Gateway merupakan elemen yang disimbolkan dengan bentuk intan dan digunakan untuk mengatur pemisahan atau penyatuan dari beberapa aliran urutan. Tanda di dalam sebuah Gateway dapat mengindikasikan jenis dari sifat Gateway tersebut.



Sumber : www.bpmn.org

Gambar 2.6 Notasi *flow* pada BPMN

Objek aliran terhubung Bersama sama dalam sebuah diagram untuk

membuat struktur kerangka dasar dari proses bisnis seperti pada Gambar 2.6. Ada 3 jenis objek penghubung yang menyediakan fungsi ini, yaitu Sequence Flow, Message Flow dan Association. Sequence Flow digunakan untuk menunjukkan urutan aktivitas yang terjadi dalam sebuah proses. Message Flow digunakan untuk menunjukkan aliran pesan antara 2 partisipan proses yang terpisah, sedangkan Association digunakan untuk menghubungkan data, teks atau artifak lain dengan objek aliran.



Sumber : www.bpmn.org

Gambar 2.7 Notasi *swimlane* pada BPMN

Swimlane digunakan untuk mengatur aktivitas kedalam kategori visual untuk mengilustrasikan kemampuan fungsional atau tanggungjawab. Ada 2 jenis kategori yang ditunjukkan pada Gambar 2.7 termasuk Swimlanes yaitu Pool dan Lane. Pool menunjukkan sebuah partisipan dalam sebuah proses. Pool juga dapat bertindak sebagai wadah grafis untuk pemisah sebuah himpunan dari aktivitas dari pool lain. Sedangkan Lane merupakan subbagian dalam sebuah pool dan dapat diperluas sepanjang seluruh pool.

2.9 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma (berorientasi objek). Pemodelan (modeling) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami (Nugroho, A., 2010).

Unified Modeling Language (UML) adalah salah satu standar bahasa visual yang banyak digunakan di dunia industri untuk mengidentifikasi *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa & Shalahuddin, 2014: 133).

UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML hanya berfungsi untuk melakukan pemodelan, jadi penggunaan UML tidak terbatas pada metodogi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek

UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat.

Didalam UML ada 3 buah diagram yang dikelompokkan kedalam tiga kategori yaitu:

1. *Structure Diagrams*, yaitu kumpulan diagram-diagram yang menggambarkan struktur statis dari sistem yang dimodelkan
2. *Behavior Diagrams*, yaitu kumpulan diagram-diagram yang menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi di dalam sistem.
3. *Interaction Diagrams*, yaitu kumpulan diagram-diagram yang menggambarkan interaksi sistem dengan sistem lain ataupun interaksi antar subsistem dalam sebuah system.

Secara garis besar, beberapa diagram utama sudah dapat menggambarkan keseluruhan sistem. Diagram tersebut antara lain *use case diagram*, *class diagram*, *sequence diagram* dan *activity diagram*.


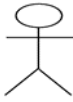

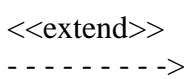
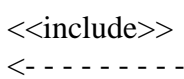
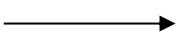
2.9.1 Use Case Diagram

Use Case Diagram menurut (Widodo, 2011: 10) Diagram *use case* bersifat statis, yang memperlihatkan himpunan *Use Case* dan aktor-aktor (suatu jenis khusus dari kelas) dan menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna yang menjadi persoalan itu apa yang

dilakukan bukan bagaimana melakukannya. *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sistem.

Deskripsi simbol-simbol yang digunakan pada *use case diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Deskripsi Notasi pada *Use Case Diagram*

No.	Notasi	Nama	Deskripsi
1.		<i>Use Case</i>	Menggambarkan fungsionalitas yang dimiliki sistem.
2.		<i>Actor</i>	Menggambarkan semua objek di luar sistem (bukan hanya pengguna sistem/perangkat lunak) yang berinteraksi dengan sistem yang dikembangkan.
3.		<i>Association</i>	Lintasan komunikasi antara <i>actor</i> dengan <i>user case</i> .
4.		<i>Extended</i> (Ekstensi)	Penambahan perilaku ke suatu <i>use case</i> dasar.
5.		<i>Include</i> (Menggunakan)	Penambahan perilaku ke suatu <i>use case</i> dasar yang secara <i>explicit</i> mendeskripsikan penambahan tersebut.
6.		Generalisasi / <i>generalization</i>	Hubungan generalisasi dan spesialisasi (umum - khusus) antara dua buah <i>use case</i> dimana fungsi yang satu merupakan fungsi yang lebih umum dari lainnya




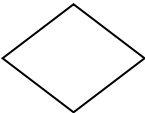

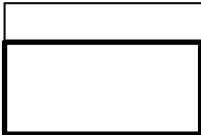
Sumber : Rosa dan M.Shalahuddin (2018:156)

2.9.2 Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) proses bisnis dan urutan aktivitas dalam sebuah proses. *Activity diagram* sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status. Membuat *activity diagram* pada awal pemodelan proses cukup menguntungkan untuk membantu memahami keseluruhan proses. *Activity diagram* juga bermanfaat untuk menggambarkan *parallel behavior* atau menggambarkan interaksi antara beberapa *use case*.

Berikut merupakan simbol notasi *activity diagram* pada Tabel 2.3.

Tabel 2.3 Deskripsi Notasi pada *Activity Diagram*

No.	Notasi	Nama	Deskripsi
1.		Status Awal (<i>initial node</i>)	Status awal aktivitas, sebuah diagram aktivitas memiliki status awal.
2.		Status Akhir (<i>final node</i>)	Status akhir yang dilakukan sistem.
3.		Aktivitas (<i>activity</i>)	Aktivitas yang dilakukan oleh sistem, biasanya diawali oleh kata kerja.
4.		Percabangan (<i>decision</i>)	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
5.		Penggabungan (<i>join</i>)	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
6.		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber: Sukamto dan M. Salahuddin (2018:162)

2.9.3 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun system (Sukamto & Shalahuddin, 2018).

Class diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. *Class diagram* juga menunjukkan properti dan operasi sebuah class dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut (Fowler, 2005: 53).

Class diagram digunakan untuk menggambarkan hubungan kelas – kelas antara satu dengan yang lain seta memiliki atribut dan operasi yang terdapat dalam

sistem yang akan dibuat. Atribut merupakan variabel – variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi – fungsi yang dimiliki oleh suatu kelas. Atribut dan metode dapat memiliki salah satu sifat sebagai berikut:

1. *Private* (-), hanya dapat digunakan oleh *class* yang memilikinya
2. *Public* (+), dapat digunakan oleh *class* lain.
3. *Protected* (#), hanya dapat dipanggil oleh *class* yang bersangkutan jumlah suatu anak yang mewarisinya.

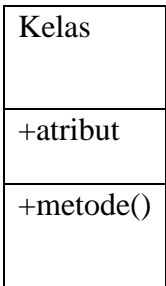

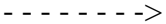
Nilai kardinalitas atau *multiplicity* sebuah *class* menunjukkan jumlah suatu objek yang dapat berhubungan dengan objek lain. Berikut nilai kardinalitas atau *multiplicity* pada Tabel 2.4 dan notasi *class diagram* pada Tabel 2.5.

Tabel 2.4 Jenis-Jenis *Multiplicity*

No.	Indikator	Keterangan
1.	0 .. 1	Nol atau satu
2.	1	Hanya satu
3.	0 .. *	Nol atau lebih
4.	1 .. *	Satu atau lebih

Sumber : Hamim Tohari (2014)

Tabel 2.5 Deskripsi Notasi pada *Class Diagram*

No.	Notasi	Nama	Deskripsi
1.		<i>Class</i>	Menggambarkan konsep dasar pemodelan sistem.
2.		Asosiasi (<i>Assosiation</i>)	Sebuah garis solid antara dua <i>class</i> , di tarik dari <i>class</i> sumber ke <i>class</i> target lebih spesifik, digunakan dalam struktur pewarisan.
3.		Ketergantungan (<i>Dependency</i>)	Relasi antara dua elemen jika perubahan definisi sebuah elemen (<i>supplier</i> atau sumber) dapat menyebabkan perubahan pada elemen lainnya (<i>Client</i> atau target).

Sumber : Sukamto dan M. Shalahuddin (2018:146)

2.10 Teknologi Pendukung

2.10.1 Database

Database yaitu kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil, dan dicari secara cepat (Raharjo, 2011). Selain berisi data, *database* juga berisi metadata yaitu data yang menjelaskan tentang struktur dari data itu sendiri, contoh: informasi yang diperoleh tentang nama-nama kolom dan tipe data yang ada pada sebuah tabel, data nama kolom dan tipe yang ditampilkan tersebut disebut metadata. Basis data juga disebut sebagai kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Istilah "basis data" berawal dari ilmu komputer.

Primary key yaitu suatu aturan yang berguna untuk memastikan bahwa setiap baris data di dalam suatu tabel bersifat unik/berbeda antara baris yang satu dengan yang lainnya sedangkan *foreign key* berguna untuk mendefinisikan kolom-kolom pada suatu tabel yang nilainya mengacu ke tabel lain.

2.10.2 MySql

MySql adalah *software* atau program aplikasi *Database Server* (Nugroho, B., 2013), yaitu *software* yang dapat kita pakai untuk menyimpan data berupa informasi teks dan juga angka. Semua informasi data yang dipakai di dalam satu program aplikasi disimpan dalam satu *software database*, yaitu *MySql*. Selain itu dengan menggunakan *database MySql*, maka data yang tersimpan di dalam *database* dapat diakses secara bersama menggunakan beberapa komputer/laptop yang berbeda, konsep ini sering disebut dengan sistem *multiuser*.

Database MySql juga dapat diinstal pada sebuah komputer pusat (*server*) yang tersimpan di dalam ruang *server*, kemudian datanya diakses melalui komputer/laptop yang terinstal dengan program *client* seperti program kasir toko atau program akademik siswa yang letaknya berbeda tempat, misalnya di lab ataupun di toko dan konsep tersebut disebut dengan sistem *client/server*.

2.10.3 Framework Laravel

Laravel merupakan *framework* PHP yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya. *Laravel* dirilis dibawah lisensi

MIT dengan sumber kode yang disediakan di *Github*. Sama seperti *framework* PHP lainnya, laravel dibangun dengan basis *Model View Controller* (MVC). Laravel dilengkapi command line tool yang bernama “*Artisan*” yang bisa digunakan untuk *packaging bundle* dan instalasi *bundle*. *Framework Laravel* dibuat oleh Taylor Otwell, proyek *laravel* dimulai pada April 2011. Awal mula proyek ini dibuat karena Otwell sendiri tidak menemukan *framework* yang up-to-date dengan versi PHP. Mengembangkan *framework* yang sudah ada juga bukan merupakan ide yang bagus karena keterbatasan sumber daya. Dikarenakan beberapa keterbatasan tersebut, Otwell membuat sendiri *framework* dengan nama laravel. Oleh karena itu *laravel* mensyaratkan PHP versi 5.3 keatas (Rohman, 2014: 1).

Laravel merupakan *framework* aplikasi web yang ekspresif dan sintaks yang elegan. Kami membangun *framework* ini dengan menyenangkan dan dari pengalaman-pengalaman kreatif agar menjadi produk yang memuaskan. *Laravel* berusaha mencoba menjadi *framework* yang mudah digunakan dengan mengurangi tugas-tugas umum yang sering digunakan dalam sebagian besar proyek-proyek *web* seperti otentikasi, *routing*, *session*, dan *caching*. *Laravel* membuat proses *development* yang menyenangkan bagi pengembang tanpa mengurangi fungsionalitas aplikasi. Dengan harapan, pengembang dapat membuat rangkaian kode-kode terbaik. *Laravel* berusaha untuk menggabungkan yang terbaik dari apa ada dalam *framework web* lain, termasuk *framework* yang menggunakan bahasa lain, seperti Ruby on Rails, *ASP.NET MVC*, dan Sinatra.

Laravel merupakan *framework* yang mudah diakses, powerful dan menyediakan *tools* yang diperlukan untuk skala aplikasi besar. *Laravel* juga merupakan sebuah aplikasi luar biasa dari sebuah kumpulan program kontrol, sistem migrasi yang ekspresif dan dukungan *tools* dibutuhkan dalam menguji aplikasi yang terintegrasi dengan beberapa aplikasi lainnya.

2.10.4 Laragon

Laragon adalah perangkat lunak bebas yang di dalamnya terdapat banyak sistem operasi sebagai *localhost* atau *server* mandiri. *Laragon* menyediakan banyak layanan, peralatan, dan fitur yang terdiri dari *Apache*, *PHP Server*, *PHPMyAdmin*, *MySQL*, *Memcached*, *Redis*, *Composer*, *Xdebug*, *Cmdr* dan *Laravel* (Putra et al.,

2019).

Laragon merupakan aplikasi yang dapat merubah komputer menjadi sebuah server maupun lokal atau dapat disebut sistem *web stack* untuk pengembangan *web*. *Laragon* berada di lingkungan yang terikat dengan sistem operasi *Windows* dan menawarkan program-program pendukung untuk pengembangan *web modern* seperti *Ruby on Rails*, *Laravel*, *Django*, *Flask*, *MEAN*, dan *Spring Boot* serta penggunaan *MySQL*, *PostgreSQL*, *MongoDB*, *Memcached*, *Redis*, *PHP*, *Ruby*, *Python*, *Node.js*, dan *Java* yang mudah. *Laragon* dapat bekerja dengan baik untuk mengembangkan sebuah *web* dan memiliki kecepatan yang luar biasa. *Laragon* memiliki beberapa fitur unggulan seperti mendukung *SSL*, memungkinkan pengembang mengubah versi program pendukung dengan mudah, mengatur database, membuat proyek aplikasi dengan cepat, dan tampilan yang *user friendly*.

Laragon digunakan sebagai *Universal Development Environment* untuk *PHP*, *Node.js*, *Python*, *Java*, *Go*, *Ruby*, yang portable, terisolasi, cepat, ringan, dan mudah dipakai. Perangkat lunak sama halnya dengan *XAMPP* dan *WAMP*, yang digunakan untuk membangun *local development environment* pada sistem operasi *windows*. Namun selain menyertakan *PHP*, *Apache Web Server*, dan database *MySQL* dalam paket instalasinya, *laragon* juga menyediakan segudang fitur menarik yang sangat relevan dengan kebutuhan pengembangan *web*. *Laragon* adalah perangkat lunak bebas yang mendukung banyak sistem operasi, berfungsi sebagai *server* diri sendiri atau *localhost*.

2.10.5 Email

Surat elektronik atau *electronic mail (e-mail)* adalah salah satu bentuk atau cara pengiriman surat, informasi, atau pesan (bisnis dan nonbisnis) yang dilakukan secara elektronik, tanpa kertas dan tanpa jasa pengirim (Purwanto, 2007: 170).

Electronic mail adalah layanan yang diberikan oleh internet yang berkembang sejak tahun 1960. Pada saat itu internet belum terbentuk, yang ada hanyalah kumpulan '*mainframe*' yang terbentuk sebagai jaringan. Mulai tahun 1980-an, *electronic mail* sudah bisa dinikmati oleh khalayak umum. *Electronic mail* adalah salah satu proses pengiriman surat melalui internet dengan menggunakan waktu yang sangat singkat dan cepat.

2.10.6 JavaScript

JavaScript adalah bahasa script yang disisipkan pada kode HTML dan diproses di sisi klien yaitu sering disebut *client side*. Bahasa ini menjadikan dokumen HTML menjadi semakin luas (Agusriandi, 2018)

2.10.7 Hyper Text Markup Language (HTML)

HTML merupakan halaman yang berada pada suatu situs internet atau web. HTML merupakan metode yang menautkan (*link*) satu dokumen ke dokumen lain melalui teks (Kuswayatno, 2006).

HTML adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web Internet dan pemformatan hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format ASCII normal sehingga menjadi halaman web dengan perintah-perintah HTML.

Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan *Standard Generalized Markup Language* (SGML), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium* (W3C). HTML dibuat oleh kolaborasi Caillau TIM dengan Berners-lee Robert ketika mereka bekerja di CERN pada tahun 1989 (CERN adalah lembaga penelitian fisika energi tinggi di Jenewa).

2.10.8 Hypertext Preprocessor (PHP)

PHP adalah singkatan dari *Hypertext Preprocessor* yang digunakan sebagai bahasa *script server-side* dalam pengembangan web yang disisipkan pada dokumen HTML. Penggunaan PHP memungkinkan web dapat dibuat dinamis sehingga *maintenance* situs *web* tersebut menjadi lebih mudah dan efisien. PHP ditulis menggunakan bahasa C. PHP merupakan *software open-source* yang disebar dan dilisensikan secara gratis (Peranginangin, 2006). Salah satu fitur yang dapat diandalkan oleh PHP adalah dukungannya terhadap banyak *database*,

yakni antara lain:

- | | |
|------------------|---------------|
| 1. Adabas D | 6. Frontbase |
| 2. dBase | 7. MySQL |
| 3. Direct MS-SQL | 8. Informix |
| 4. Empress | 9. IBM DB2 |
| 5. FilePro | 10. Hyperwave |

PHP atau yang memiliki kepanjangan PHP *Hypertext Preprocessor* merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu *website* dinamis. PHP menyatu dengan kode HTML, maksudnya adalah beda kondisi (Saputra, 2011: 1). HTML digunakan sebagai pembangun atau pondasi dari kerangka *layout web*, sedangkan PHP difungsikan sebagai prosesnya sehingga dengan adanya PHP tersebut, *web* akan sangat mudah di-*maintenance*.

PHP berjalan pada sisi *server* sehingga PHP disebut juga sebagai bahasa *Server Side Scripting*. Artinya bahwa dalam setiap/untuk menjalankan PHP, wajib adanya *web server*. PHP ini bersifat *open source* sehingga dapat dipakai secara cuma-cuma dan mampu lintas *platform*, yaitu dapat berjalan pada sistem operasi Windows maupun Linux. PHP juga dibangun sebagai modul pada *web server Apache* dan sebagai *binary* yang dapat berjalan sebagai CGI.

2.11 Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan suatu teknik yang digunakan menguji apakah sebuah perangkat lunak yang dihasilkan telah sesuai dengan yang diharapkan atau belum. Pengujian perangkat lunak merupakan salah satu elemen dari rekayasa perangkat lunak yang sering disebut dengan *verification and validating testing V&V* (Pressman, 2002: 572).

2.11.1 Pengujian *Scenario Test*

Dalam penelitian ini menggunakan teknik *test case* berdasarkan *scenario test*. *Test case* adalah pengujian yang dilakukan berdasarkan beberapa masukan seperti kondisi dan hasil yang telah ditentukan sebelumnya (Romeo, 2003).

Sebuah *Test Case* atau *Scenario Test* biasanya memiliki isi *pre-conditions*, *expected results* dan *post-conditions*. *Test Case* bertindak sebagai titik awal untuk

pelaksanaan tes, dan setelah mengaplikasikan sekumpulan nilai input, aplikasi memiliki hasil yang definitif dan meninggalkan sistem di beberapa titik akhir atau juga dikenal sebagai *post-condition* eksekusi.

Test case yang baik adalah *test case* yang mempunyai kemungkinan tinggi dalam menemukan kesalahan dan cacat yang sebelumnya belum ditemukan, bukan yang dapat memperlihatkan bahwa program telah bekerja dengan benar (Romeo, 2003).

Test case merupakan kunci untuk proses karena mengidentifikasi dan mengkomunikasikan kondisi yang akan diimplementasikan pada *test* dan yang diperlukan untuk memverifikasi keberhasilan.

Tabel 2.6 Tabel Pengujian *Scenario Test*

Kode	Feature	Test Case	Test Step	Expected Result	Result

Sumber : *medium.com*

Tabel pengujian terdiri dari 6 kolom, yaitu :

- a. Kode : ID dari *test case* itu sendiri.
- b. *Feature* : fitur atau fungsi yang akan diuji coba.
- c. *Test Case* : *test case* yang akan dibuat dengan masukan data valid dan tidak valid.
- d. *Test Step* : langkah-langkah untuk uji kasus.
- e. *Expected Result* : hasil yang diharapkan dari *scenario test* tersebut.
- f. *Result* : hasil/*output* yang dihasilkan berhasil/gagal.

2.11.2 Pengujian *User Acceptance Test (UAT)*

User Acceptance Test (UAT) atau Uji Penerimaan Pengguna adalah suatu proses pengujian oleh pengguna yang dimaksudkan untuk menghasilkan dokumen yang dijadikan bukti bahwa *software* yang telah dikembangkan telah dapat diterima oleh pengguna, apabila hasil pengujian (*testing*) sudah bisa dianggap memenuhi kebutuhan dari pengguna.

Proses pengujian UAT diawali dengan menyediakan dokumentasi persyaratan bisnis, kemudian dilanjutkan dengan proses bisnis (alur kerja) atau skenario dan yang terakhir yaitu pengujian menggunakan data (Branch, 2008).

Pengujian *User Acceptance Testing (UAT)* pada umumnya dilakukan

sebelum peluncuran sebuah fitur baru di dalam aplikasi. Dengan melakukan ini pengembang dapat memahami apakah rancangan yang dibuat sudah memenuhi harapan pengguna (Syafarwan, 2019).

Tabel 2.7 Pilihan Jawaban UAT

A	Sangat : Mudah/Bagus/Sesuai/Jelas
B	Mudah/Bagus/Sesuai/Jelas
C	Netral
D	Cukup : Sulit/ Bagus/Sesuai/Jelas
E	Sangat : Sulit/ Bagus/Sesuai/Jelas

Sumber : endangcahyapermana.wordpress.com

Tabel 2.8 Bobot Nilai Jawaban

Jawaban	Bobot
A. Sangat : Mudah/Bagus/Sesuai/Jelas	5
B. Mudah/Bagus/Sesuai/Jelas	4
C. Netral	3
D. Cukup : Sulit/ Bagus/Sesuai/Jelas	2
E. Sangat : Sulit/ Bagus/Sesuai/Jelas	1

Sumber : endangcahyapermana.wordpress.com

Tabel 2.9 Kriteria Interpretasi Skor

Jawaban	Bobot
81% - 100%	Sangat Baik
61% - 80%	Baik
41% - 60%	Cukup Baik
21% - 40%	Kurang Baik
0% - 20%	Sangat Kurang Baik

Sumber : Riduwan (2008: 89)

Tabel 2.10 Pertanyaan Kuesioner

No.	Pertanyaan	A	B	C	D	E
1.						
2.						
3.						
n.						

Sumber : endangcahyapermana.wordpress.com