

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Terdapat beberapa penelitian terdahulu yang sudah meneliti tentang masalah yang serupa mengenai *Human Resource Information System (HRIS)*. Hal ini bisa menjadi salah satu acuan penulis dalam melakukan penelitian sehingga penulis dapat memperbanyak teori yang digunakan untuk mengkaji penelitian yang dilakukan. Berikut ini merupakan penelitian terdahulu yang berupa jurnal terkait dengan penelitian yang dilakukan penulis.

Penelitian yang pertama adalah penelitian ini mengenai tentang “Pengaruh Sistem Informasi Sumber Daya Manusia (HRIS) dan Kecerdasan Buatan Terhadap Kinerja Industri Pertahanan yang ditulis oleh Novi Manunggal, Imam Teguh Santoso dan Sigit Wicaksana (2022). Penelitian tersebut bertujuan untuk menganalisis pengaruh *Human Resource Information System (HRIS)* terhadap kinerja industri pertahanan dan pengaruh kecerdasan buatan terhadap kinerja industri pertahanan di era disrupsi. Dari hasil penelitian tersebut penulis menyimpulkan bahwa nilai signifikansi yang diperoleh membuktikan bahwa HRIS terbukti berhubungan signifikan dengan kinerja pegawai. Semakin akurat, tepat dan lengkap aplikasi HRIS maka semakin tinggi kinerja karyawan dalam menyelesaikan tugas pekerjaannya. Peningkat proses industri pertahanan dalam pengambilan keputusan sumber daya manusia melalui prosedur sistematis HRIS yang meliputi pengumpulan data, penyimpanan data, pemeliharaan data, penarikan data, dan validasi data (Manunggal, Santoso, & Wicaksana, 2022).

Penelitian yang kedua adalah mengenai “Sistem Informasi Manajemen Sumber Daya Manusia Berbasis Web (Studi Kasus : PT. Klik Teknologi Indonesia)” yang ditulis oleh “Nia Kusuma Wardhani dan Muhammad Thariq Abdul Aziz (2018)”. Penelitian tersebut menyimpulkan bahwa sistem informasi manajemen SDM yang dikembangkan membuat pengelolaan SDM di perusahaan menjadi lebih mudah, cepat dan akurat, sistem informasi manajemen SDM membuat karyawan-karyawan yang bekerja secara *remote* dapat melakukan absen dan bagian SDM dapat mengelola data absensi dari karyawan-karyawan tersebut(Wardhani & Thariq Abdul Aziz, 2018).

Penelitian yang terakhir adalah dengan judul penelitian “Pengembangan Aplikasi E-DUK Dalam Pengelolaan SDM Menggunakan Metode *Agile Development*” karya dari Pratasik dan Rianto yang dikerjakan pada tahun 2020. Penelitian tersebut bertujuan untuk menangani pendataan SDM di lingkungan Universitas Negeri Manado yang memiliki nilai validitas dan reabilitas data tinggi dengan menggunakan metode *Agile Development* dalam tahap pengembangan perangkat lunak. Hasil dari penelitian tersebut adalah perancangan sistem yang didalamnya memuat mengenai *usecase diagram* yang didalamnya melibatkan tiga aktor yaitu pengelola data, administrator, dan pimpinan, class diagram dan tampilan perancangan *Interface* dari sistem yang dirancang. Penelitian tersebut menyimpulkan bahwa melalui penelitian ini dapat dibuktikan bahwa metode *Agile Development* dapat digunakan dalam pengembangan aplikasi sistem informasi pada perguruan tinggi terlebih khusus dalam aplikasi e-DUK Universitas Negeri yang dalam pengembangannya hanya membutuhkan waktu yang singkat.

Penerapan metode *Agile Development* dalam proses pengembangan aplikasi e-DUK terdiri dari empat tahap yaitu perancangan, pengkodean, pengujian dan realease produk(Pratasik & Rianto, 2020).

Dari hasil penelitian diatas, dapat disimpulkan bahwa sistem informasi yang akan dibangun pada penelitian ini sebagian memiliki fitur yang sama. Persamaan fitur yang sangat menonjol ditunjukkan pada penelitian yang kedua, ditulis oleh Nia Kusuma Wardhani dan Muhammad Thariq Abdul Aziz (2018), dengan judul penelitian “Sistem Informasi Manajemen Sumber Daya Manusia Berbasis Web”. Namun yang membedakan antara penelitian diatas dengan penelitian yang akan dilaksanakan ini mempunyai perbedaan fitur seperti pengajuan jadwal interview yang dilakukan oleh HRD.

2.2 Sumber Daya Manusia

Sumber Daya Manusia (SDM) adalah individu produktif yang bekerja sebagai penggerak suatu organisasi, baik itu di dalam institusi maupun perusahaan yang memiliki fungsi sebagai aset sehingga harus dilatih dan dikembangkan kemampuannya. Pengertian sumber daya manusia makro secara umum terdiri dari dua yaitu SDM makro yaitu jumlah penduduk dalam usia produktif yang ada di sebuah wilayah dan SDM mikro dalam arti sempit yaitu individu yang bekerja pada sebuah institusi atau perusahaan.

Sumber Daya Manusia (SDM) merupakan suatu hal yang sangat penting dan harus memiliki dalam upaya mencapai tujuan organisasi atau perusahaan. Sumber daya manusia merupakan elemen utama organisasi dibandingkan dengan

elemen sumber daya yang lain seperti modal, teknologi, karena manusia itu sendiri yang mengendalikan faktor yang lain.

Sumber Daya Manusia (SDM) adalah salah satu faktor yang sangat penting bahkan tidak dapat dilepaskan dari sebuah organisasi, baik institusi maupun perusahaan. SDM juga merupakan kunci yang menentukan perkembangan perusahaan. Pada hakikatnya, SDM berupa manusia yang dipekerjakan di sebuah organisasi sebagai penggerak, pemikir dan perencana untuk mencapai tujuan organisasi itu (Susan, 2019).

2.2.1 Manajemen Personalia

Manajemen personalia adalah perencanaan, pengembangan, pembagian, kompensasi, penginterpretasian, dan pemeliharaan tenaga kerja dengan maksud untuk mencapai tujuan perusahaan, individu dan masyarakat (Ranupandojo & Husnan, 2022).

Menurut (Manullang, 2015) menyebutkan bahwa manajemen personalia adalah ilmu yang mempelajari bagaimana cara memberikan fasilitas untuk perkembangan, pekerjaan dan rasa partisipasi pekerjaan dalam suatu kegiatan.

Administrasi personalia adalah administrasi yang menangani masalah-masalah kepegawaian dalam suatu usaha atau lembaga (Sutopo, 1989). Sedangkan menurut (Arikunto, 1990) Administrasi Personal atau Administrasi Kepegawaian adalah segenap proses penataan yang bersangkutan paut dengan masalah memperoleh dan menggunakan tenaga kerja untuk suatu organisasi atau perusahaan dengan efisien, demi tercapainya tujuan perusahaan yang telah ditentukan.

Tujuan personalia berhubungan dengan tujuan perusahaan secara umum. Hal ini dikarenakan manajemen perusahaan berusaha untuk menimbulkan efisiensi dalam bidang tenaga kerja sebagai efisiensi keuntungan dan kontinuitas. Ada beberapa fungsi dari personalia yaitu terdiri dari :

1. Perencanaan, untuk menentukan program personalia yang akan membantu mencapai tujuan perusahaan yang telah ditetapkan. Tujuan ini memerlukan partisipasi aktif dari manajer personalia.
2. Pengorganisasian, jika perusahaan telah menentukan fungsi-fungsi yang harus dijalankan oleh pegawai, maka manajer personalia harus membentuk organisasi dengan merancang susunan dari berbagai hubungan antara jabatan personalia dan faktor-faktor fisik. Organisasi merupakan alat untuk mencapai tujuan dari perusahaan.
3. Pengarahan, apabila manajer sudah mempunyai rencana dan sudah mempunyai organisasi untuk melaksanakan rencana tersebut, fungsi selanjutnya adalah mengadakan pengarahan terhadap pekerjaan.
4. Pengawasan adalah mengamati dan membandingkan pelaksanaan dengan rencana dan mengoreksinya apabila terjadi penyimpangan. Dengan kata lain pengawasan adalah fungsi yang menyangkut masalah pengaturan berbagai kegiatan sesuai dengan rencana personalia yang dirumuskan sebagai dasar analisis dari tujuan organisasi fundamental.

Fungsi personalia secara operasionalnya terdiri dari :

1. Pengadaan adalah menyediakan sejumlah tertentu karyawan dan jenis keahlian yang diperlukan untuk mencapai tujuan perusahaan. Tujuan

tersebut menyangkut masalah pemenuhan kebutuhan tenaga kerja, proses seleksi dan penempatan kerja.

2. Pengembangan pegawai yang telah diperoleh dengan melalui pelatihan dengan tujuan yang untuk mengembangkan keterampilan.
3. Pemberian kompensasi adalah pemberian penghargaan yang adil dan layak terhadap para pegawai sesuai dengan sumbangan mereka dalam mencapai tujuan perusahaan.
4. Pemeliharaan adalah mempertahankan dan meningkatkan kondisi yang telah ada.

Tugas dari personalia adalah sebagai berikut :

1. Bertanggung jawab terhadap data karyawan, dan data organisasi perusahaan.
2. Melakukan sosialisasi dan koordinasi.
3. Menyiapkan perjanjian kerja dengan pegawai atau karyawan baru.
4. Penerimaan tenaga kerja.
5. Menambahkan, memperbaharui atau *update* dan menghapus data.

2.2.2 Perekrutan Calon Pegawai

Dalam sebuah perusahaan, penyusutan pegawai umumnya terjadi karena adanya pegawai yang memasuki masa pensiun, meninggal dunia atau dikeluarkan dari perusahaan dikarenakan melakukan pelanggaran tata tertib yang sudah ditetapkan oleh perusahaan. Oleh karena itu, pegawai baru yang akan menggantikannya memiliki masa kerja yang berbeda-beda. Rekrutmen juga dapat dilakukan untuk menambah pegawai baru kedalam suatu satuan kerja yang

kegiataannya menuntut aktivitas yang tinggi. Dalam proses rekrutmen juga memerlukan adanya proses seleksi yang efektif, hal ini dilakukan untuk melakukan pemerataan pegawai sehingga kekuatan SDM yang dimiliki menjadi lebih seimbang.

Rekrutmen dilakukan untuk kebutuhan tenaga kerja baik yang bersifat jangka pendek, menengah maupun panjang. Untuk dipekerjakan di perusahaan melalui sistem dan prosedur dari rekrutmen yang ditentukan perusahaan, sehingga pegawai baru tersebut dapat ditempatkan sesuai dengan kebutuhan jabatan yang telah dianalisis oleh manajemen sumber daya manusia (SDM). Pelaksanaan sistem rekrutmen terhadap calon pegawai atau karyawan dimaksudkan agar perusahaan dapat memperoleh pegawai yang berkualitas dan mampu merealisasikan tujuan perusahaan.

Setelah proses rekrutmen, untuk memperoleh calon karyawan yang berkualitas selesai maka akan dihasilkan sejumlah pelamar yang disaring melalui proses seleksi. Proses ini memilih orang yang memenuhi spesifikasi kebutuhan organisasi atau perusahaan. Proses ini sangat bervariasi untuk setiap perusahaan. Masing-masing menerapkan kebutuhan yang berbeda dalam perlakuan seleksi (Rachmawati, 2008).

2.3 *Human Resource Information System (HRIS)*

Human Resource Information System (HRIS) dapat didefinisikan sebagai sistem terintegrasi yang digunakan untuk mengumpulkan, menyimpan dan menganalisa informasi tentang sumber daya manusia dalam sebuah organisasi yang terdiri dari basis data (*database*), komputer aplikasi, perangkat keras dan

perangkat lunak yang diperlukan untuk mengumpulkan, merekam, menyimpan, mengelola, memberikan, menyajikan dan memanipulasi data untuk fungsi-fungsi sumber daya manusia (Jonni & Husain, 2016).

HRIS dapat membantu fungsi SDM untuk lebih efisien dan memberikan informasi yang lebih baik, hal ini dapat digunakan sebagai bahan pertimbangan dalam pengambilan keputusan (Jonni & Husain, 2016).

HRIS adalah sistem yang didesain agar *input* datanya berhubungan dengan sumber daya manusia dan outputnya dapat memberikan data yang efektif yang dapat digunakan untuk pengambilan keputusan (Jonni & Husain, 2016).

2.4 Sistem Informasi

Sistem adalah kumpulan orang yang saling bekerja sama dengan ketentuan-ketentuan aturan yang sistematis dan terstruktur untuk membentuk satu kesatuan yang melaksanakan suatu fungsi untuk mencapai tujuan. Sedangkan informasi adalah data yang diolah menjadi lebih berguna dan berarti bagi penerimanya, serta untuk mengurangi ketidakpastian dalam proses pengambilan keputusan mengenai suatu keadaan (Elisabet & Rita, 2017).

Sistem Informasi memiliki beberapa fungsi diantaranya sebagai berikut (Elisabet & Rita, 2017) :

1. Untuk meningkatkan aksesibilitas data yang ada secara efektif dan efisien kepada pengguna, tanpa dengan perantara sistem informasi.
2. Memperbaiki produktivitas aplikasi pengembangan dan pemeliharaan sistem.

3. Menjamin tersedianya kualitas dan keterampilan dalam memanfaatkan sistem informasi secara kritis
4. Mengidentifikasi kebutuhan mengenai keterampilan pendukung sistem informasi.
5. Mengantisipasi dan memahami akan konsekuensi ekonomi.
6. Menetapkan investasi yang akan diarahkan pada sistem informasi.
7. Mengembangkan proses perencanaan yang efektif.

Sistem bisa dikatakan sistem yang baik apabila memiliki beberapa karakteristik diantaranya (Hutahean, 2015) :

1. Komponen

Suatu sistem terdiri dari sejumlah komponen-komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem terdiri dari komponen yang berupa subsistem atau bagian-bagian dari sistem.

2. Batasan Sistem (*boundary*)

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar sistem (*environment*) adalah diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan dapat bersifat menguntungkan

yang harus tetap dijaga dan yang merugikan yang harus dijaga dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung Sistem (*Interface*)

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui *interface* ini memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem yang lain. *Output* dari subsistem akan menjadi *input* untuk subsistem lain melalui penghubung (*interface*).

5. Masukkan Sistem (*Input*)

Masukkan (*input*) adalah energi yang dimasukkan ke dalam sistem, yang dapat berupa perawatan (*maintenance input*), dan masukkan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Contoh dalam sistem komputer program adalah *maintenance input* sedangkan data adalah *signal input* untuk diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

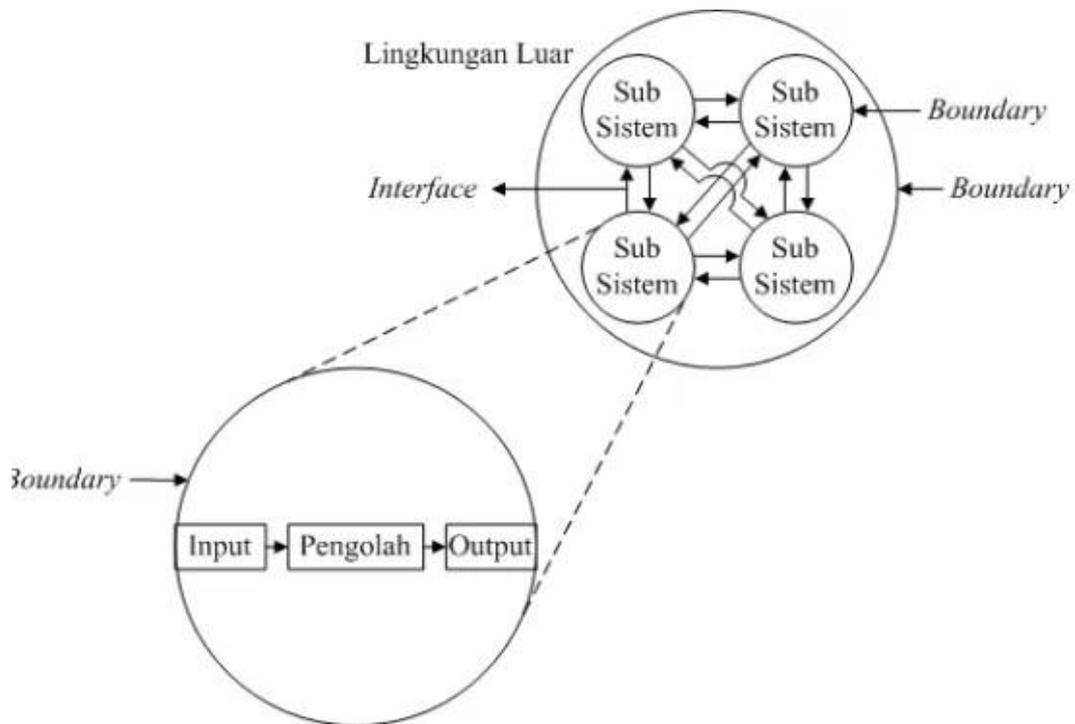
Keluaran sistem adalah hasil dari masukkan sistem yang sudah diolah oleh pengolah sistem.

7. Pengolah Sistem

Suatu sistem menjadi bagian pengolah yang akan merubah masukkan (*input*) menjadi keluaran (*output*).

8. Sasaran Sistem

Sistem mempunyai tujuan atau sasaran. Sasaran dari sistem sangat bergantung pada masukan (*input*) yang dibutuhkan sistem dan keluaran (*output*) yang akan dihasilkan oleh sistem.



Gambar 2.1 Karakteristik Sistem

2.5 Agile Software Development

Agile Software Development berasal dari kata *Agility/Agile* yang berarti cepat, ringan, bebas bergerak, efektif (cepat dan adaptif) dalam merespon perubahan. *Agile Software Development* adalah sekumpulan metodologi perancangan perangkat lunak yang berbasis pada pengembangan iterative, dimana persyaratan dan solusi berkembang melalui kolaborasi antar tim yang terorganisir (Hariman & Meilisa, 2020).

Agile Alliance's Manifesto, diantaranya adalah sebagai berikut :

1. Interaksi dan personel lebih penting daripada proses dan alat, di dalam *agile* interaksi antar anggota tim sangat penting, karena tanpa adanya interaktif yang baik maka proses pembuatan perangkat lunak tidak akan berjalan sesuai rencana.
2. Perangkat lunak yang berfungsi lebih penting daripada dokumentasi yang lengkap, saat melakukan proses demonstrasi kepada klien, perangkat lunak yang berfungsi dengan baik akan lebih berguna daripada dokumentasi yang lengkap.
3. Kolaborasi dengan klien lebih penting daripada negosiasi kontrak, salah satu ciri dari *agile* adalah klien menjadi bagian dari tim pengembangan perangkat lunak. Kolaborasi yang baik dengan klien saat proses pembuatan perangkat lunak sangatlah penting ketika menggunakan *agile*. Karena fungsi-fungsi disesuaikan dengan keinginan klien.
4. Respon terhadap perubahan lebih penting daripada mengikuti rencana, *Agile Development methods* berfokus terhadap kecepatan respon tim ketika klien menginginkan perubahan saat proses pembuatan perangkat lunak.



Gambar 2.2 *Agile Metodology* (Reddy, 2019)

2.6 Scrum

Metode perancangan sistem ini yang dapat digunakan untuk menghasilkan suatu produk. *Scrum* adalah salah satu metodologi manajemen perangkat lunak yang *responsive* serta berbasis metodologi *Agile Development* yang mampu memberikan nilai dan manfaat yang terbaik terhadap suatu perangkat lunak yang akan dikembangkan (Suharno, Gunantara, & Made, 2020). *Scrum* merupakan sebuah kerangka kerja dimana didalam proses perancangannya dapat memasukkan berbagai proses dan teknik. *Scrum* dapat menunjukkan hasil dari proses perancangan suatu produk sehingga dapat menghasilkan produk yang lebih baik. *Scrum* memiliki kerangka kerja yang dimana orang-orang atau tim yang terlibat dapat menyesuaikan permasalahan kompleks yang sewaktu-waktu berubah, di mana pada saat bersamaan dapat menghasilkan produk atau aplikasi

dengan nilai yang maksimal secara kreatif dan produktif (Schwaber & Sutherland, 2020). Adapun tahapan-tahapan dari *scrum* yang ditulis oleh (Pratama, 2018) :

2.6.1 *Product Backlog*

Product Backlog adalah daftar utama dari semua fungsi atau fitur yang diinginkan dalam produk atau aplikasi. *Product backlog* merupakan kumpulan dari beberapa *Product Backlog Item (PBI)* yang merepresentasikan *requirement* yang dibutuhkan oleh produk atau aplikasi. PBI umumnya merupakan *user story*, merupakan gambaran bagaimana pengguna (*user*) dapat melakukan sesuatu dan berinteraksi dengan produk melalui cerita. *Product owner* bertanggung jawab atas isi dari *Product backlog*.

2.6.2 *Sprint Planning*

Sprint Planning dibuat untuk merencanakan kolaborasi pekerjaan yang dapat dilakukan oleh *scrum team*. *Sprint Planning* membahas tentang hal yang akan dilakukan untuk meningkatkan hasil yang diperoleh dari *sprint* tersebut. *Product Owner* memilih prioritas tertinggi yang dikerjakan pada *sprint* berikutnya, yang kemudian dibagi menjadi beberapa *task* untuk dikerjakan oleh tim.

2.6.3 *Sprint Backlog*

Sprint Backlog berisi tentang target modul-modul aplikasi yang harus diselesaikan pada setiap *sprint*. Isi *sprint backlog* merupakan bagian dari *product backlog*. Selain itu pada *sprint backlog* ini dilakukan pembagian kerja dari setiap modul.

2.6.4 *Daily Scrum*

Daily Scrum merupakan pertemuan secara singkat yang diadakan setiap harinya untuk tim perancang mensinkronisasikan segala aktivitas dan rencana yang dikerjakan ke depannya. Hasil yang diperoleh tiap harinya dicatat pada *sprint backlog*.

2.6.5 *Sprint Review*

Sprint Review merupakan pembahasan tentang apa saja yang sudah dilakukan selama *Sprint*. *Sprint review* diadakan di akhir *scrum* untuk melihat *progress* yang telah dilakukan dan penyesuaian terhadap *product backlog* jika diperlukan.

2.6.6 *Sprint Retrospective*

Sprint Retrospective dilakukan setelah proses *sprint review* dan digunakan untuk inspeksi diri serta membahas apa saja yang akan dilakukan pada *Sprint* berikutnya. Setiap tim harus mampu meningkatkan produktivitasnya dan dapat diimplementasikan ke *sprint* selanjutnya.

2.7 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) merupakan metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah *software* yang berorientasikan pada objek. UML merupakan sebuah standar penulisan atau semacam *blue print* dimana didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik (Prihandoyo, 2018). Tujuan pembuatan *Unified Modeling Language (UML)* sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem. UML pertama kali di populerkan oleh Grady Booch dan James Rumbaugh pada tahun 1994 (Mulyani, 2016).

2.7.1 *Use Case*

Use Case Diagram digunakan untuk menggambarkan perilaku sistem, sub-sistem atau komponen dalam bentuk yang tampak kepada *user*. *Use Case Diagram* adalah gambaran dari beberapa atau seluruh aktor dan *use case* dengan tujuan mengenali interaksi mereka dalam suatu sistem (Booch, Rumbaugh, & Jacobson, 2005). Sedangkan menurut (Muslihudin & Oktafianto, 2016) *Use Case Diagram* adalah diagram yang menampilkan aktor, *use case* dan relasi antara keduanya didalam sistem. *Use Case Diagram* menggambarkan ruang lingkup dari

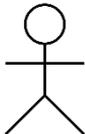
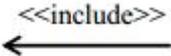
sistem yang akan dibangun dan merupakan sarana komunikasi dengan pengguna akhir untuk mendapatkan pemahaman yang lebih baik tentang sistem yang akan dibuat.

Komponen-komponen untuk membentuk *use case diagram* sebagai berikut (Muslihudin & Oktafianto, 2016) :

1. Aktor, menggambarkan sebagai pihak-pihak yang berperan dalam suatu sistem.
2. *Use case*, aktivitas atau sarana yang disediakan oleh sistem.
3. Hubungan (*link*), aktor mana saja yang terlibat dalam *use case* ini.

Berikut ini adalah kumpulan simbol yang digunakan untuk merancang *Use Case Diagram* (Dennis, Wixom, & Roth, 2012):

Tabel 2.1 Simbol-simbol *Use Case Diagram*

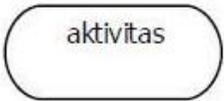
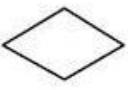
Simbol	Nama	Keterangan
	Aktor	Orang atau <i>user</i> yang berinteraksi dengan sistem dan memperoleh manfaat darinya. Digambarkan dalam figur seperti orang dan diberi nama sesuai dengan perannya.
	<i>Use Case</i>	Merupakan bagian utama dari fungsionalitas sistem yang diberi label kata kerja.
	<i>Association</i>	Merupakan penghubung antara aktor dengan <i>use case</i> yang saling berinteraksi dengannya.
	<i>Include</i>	Merupakan penyertaan fungsi dari satu <i>use case</i> di dalam yang lain. Digambarkan dengan yang ditarik dari <i>use case</i> ke pangkalan <i>use case</i> yang digunakan.
	<i>Extend</i>	Merupakan perluasan dari <i>use case</i> untuk memasukkan perilaku opsional.

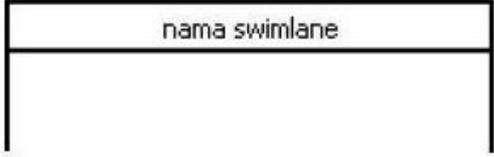
2.7.2 Activity Diagram

Activity diagram menggambarkan rangkaian alir aktivitas dalam sistem yang dirancang, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti *use case* atau interaksi. *Activity diagram* berupa *flow chart* yang digunakan untuk memperlihatkan alir kerja dari sistem (Sugiarti, 2013).

Berikut adalah simbol-simbol yang terdapat pada *activity diagram*.

Tabel 2.2 Simbol-simbol *Activity Diagram*

Simbol	Nama	Keterangan
	Status Awal	Status awal aktivitas sistem, bagaimana objek dibentuk atau diawali.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / <i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	Penggabungan / <i>join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
---	-----------------	--

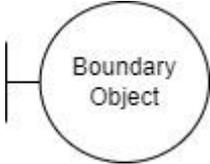
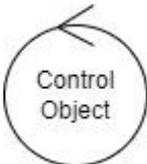
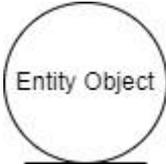
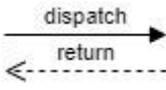
2.7.3 *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri dari atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktifitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan (Unhelkar, 2017).

Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi atau metode dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message* (Muslihudin & Oktafianto, 2016).

Berikut adalah simbol-simbol yang ada dalam *sequence diagram* (Dennis, Wixom, & Roth, 2012) :

Tabel 2.3 Simbol - simbol *Sequence Diagram*

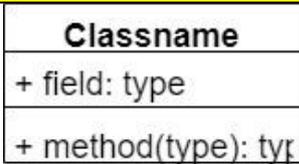
Simbol	Nama	Keterangan
 Actor	<i>Actor</i>	Seseorang yang berinteraksi dengan sistem dan memperoleh manfaat darinya. Digambarkan dalam figur seperti manusia dan diberi nama sesuai perannya.
 Boundary Object	<i>Boundary</i>	Dalam model MVC <i>boundary</i> sebagai objek <i>view</i> dalam pengembangan <i>website</i> untuk menampilkan tampilan yang di <i>request</i> oleh aktor.
 Control Object	<i>Control</i>	Dalam model MVC sebagai objek <i>controller</i> dalam pengembangan <i>website</i> untuk mengirim dan menerima pesan dari <i>view</i> dan <i>model</i> .
 Entity Object	<i>Entity</i>	Dalam model MVC sebagai objek <i>model</i> dalam pengembangan <i>website</i> untuk mengirim dan menerima pesan dari <i>controller</i> .
	<i>Lifeline</i>	Menunjukkan keberlangsungan suatu objek selama suatu urutan.
	<i>Activation</i>	Menunjukkan ketika suatu objek mengirim atau menerima pesan.
	<i>Message</i>	Menyampaikan informasi dari satu objek ke objek lainnya.

2.7.4 Class Diagram

Class Diagram adalah salah satu pemodelan yang cukup penting dalam UML, fungsinya untuk membuat sebuah *logical models* dari sebuah sistem (Wu, 2006). Sebuah *class diagram* akan menunjukkan bagaimana skema dari arsitektur sebuah sistem yang sedang dirancang (Kendal, 2009). *Class diagram* digambarkan dengan *class* yang berisi *atribut* dan *method*, setiap *class* akan dihubungkan dengan garis disebut Asosiasi.

Berikut adalah simbol-simbol yang terdapat pada *class diagram*.

Tabel 2.4 Simbol-simbol pada *Class Diagram*

Simbol		Nama	Keterangan	
		<i>Class</i>	Sebuah kelas yang terdiri dari nama kelas, atribut, dan <i>method</i> .	
		<i>Relation</i>	Sebuah gambar relasi untuk menghubungkan antar <i>class</i> .	

2.8 Basis Data (*Database*)

Basis data (*database*) merupakan gabungan data atau *file* yang dibentuk dengan hubungan/relasi yang logis dan dapat diungkapkan dengan catatan serta bersifat independen (Lubis, 2016). Adapun basis data adalah tempat berkumpulnya data yang saling berhubungan dalam suatu wadah (organisasi/perusahaan) bertujuan agar dapat mempermudah dan mempercepat untuk pemanggilan atau pemanfaatan kembali data tersebut. Arti lain dari sistem basis data adalah suatu sistem penyusunan dan pengelolaan *record-record* dengan menggunakan komputer, dengan tujuan untuk menyimpan atau merekam serta memelihara data secara lengkap pada sebuah organisasi/perusahaan, sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk kepentingan proses pengambilan keputusan (Lubis, 2016).

Sistem Basis Data menurut (Pamungkas, 2017) merupakan sekumpulan basis data dengan para pemakai yang menggunakan basis data secara bersama-sama, personil yang merancang dan mengelola basis data, teknik-teknik untuk merancang dan mengelola basis data, serta sistem komputer yang mendukungnya.

2.8.1 Database Planning (Perencanaan Basis Data)

Aktivitas manajemen untuk merealisasikan tahapan basis data secara efektif dan efisien. Perencanaan basis data terintegrasi dengan keseluruhan strategi sistem informasi organisasi. Terdapat 3 hal yang harus diperhatikan untuk menentukan strategi sistem informasi :

1. Identifikasi rencana dan sasaran dari organisasi termasuk mengenai sistem informasi yang dibutuhkan.
2. Evaluasi sistem informasi yang ada untuk menetapkan kelebihan dan kekurangan yang dimiliki oleh sistem tersebut.
3. Penaksiran kesempatan teknik informatika yang mungkin memberikan keuntungan kompetitif.

2.8.2 Database Design (Desain Basis Data)

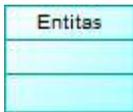
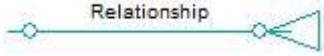
Desain basis data adalah sebuah proses membuat desain yang akan mendukung operasional dan tujuan perusahaan. Terdapat beberapa tujuan desain basis data adalah :

1. Menggambarkan relasi data, antara data yang dibutuhkan oleh aplikasi dan *user view*.
2. Menyediakan model data yang mendukung seluruh transaksi yang diperlukan.
3. Menspesifikasi desain dengan struktur yang sesuai dengan kebutuhan sistem.

2.8.3 *Conceptual Data Model (CDM)*

CDM (*Conceptual Data Model*) merupakan konsep yang berkaitan dengan pandangan pemakai terhadap data yang di simpan dalam basis data. CDM sudah di buat dalam bentuk tabel-tabel tanpa tipe data yang menggambarkan relasi antar tabel untuk keperluan implementasi ke basis data. CDM merupakan hasil penjabaran lebih lanjut dari ERD. Ada aturan-aturan yang harus diikuti dalam melakukan konversi ERD menjadi CDM (Sugiarti, 2013).

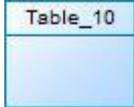
Tabel 2.5 *Conceptual Data Modeling (CDM)*

Simbol	Nama	Keterangan
	Entitas / Tabel	Entitas atau tabel yang menyimpan data dalam basis data.
	Relasi	Relasi antar tabel yang terdiri atas nama relasi dan <i>multiplicity</i> .

2.8.4 *Physical Data Model (PDM)*

Model relasional atau *Physical Data Model (PDM)* adalah model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antara data-data tersebut. Setiap tabel mempunyai sejumlah kolom di mana setiap kolom memiliki nama yang unik beserta tipe datanya. PDM merupakan konsep yang menerangkan detail dari bagaimana data di simpan di dalam basis data (Sugiarti, 2013).

Tabel 2.6 *Physical Datam Model (PDM)*

Simbol	Nama	Keterangan
	Tabel	Tabel yang menyimpan data dalam basis data
	Relasi	Relasi antar tabel yang terdiri dari persamaan antara primary key tabel yang di acu dengan kunci yang menjadi referensi acuan di tabel lain.

2.9 Website

Website merupakan kumpulan halaman digital yang berisi informasi berupa teks, animasi, gambar, suara dan video atau gabungan dari semuanya yang terkoneksi oleh internet, sehingga dapat dilihat oleh seluruh siapapun yang terkoneksi jaringan internet. Jenis kategori *website* (Sari, Abdilah, & Sunarti, 2019):

1. Web statis

Merupakan *website* yang mempunyai halaman yang tidak berubah. Perubahan suatu halaman dilakukan secara manual dengan mengedit *code* yang menjadi struktur dari *website* tersebut.

2. Web Dinamis

Merupakan *website* yang secara terstruktur diperuntukan untuk di *update* sesering mungkin. biasanya disediakan halaman *backend* untuk melakukan perubahan konten dari *website* tersebut.

Contoh : web portal, web berita, dll.

3. Web Interaktif

Merupakan *website* yang berinteraksi antara penggunanya. Biasanya, berupa forum diskusi maupun blog. Dimana adanya moderator sebagai pengatur alur diskusi.

2.10 *Hypertext Preprocessor* (PHP)

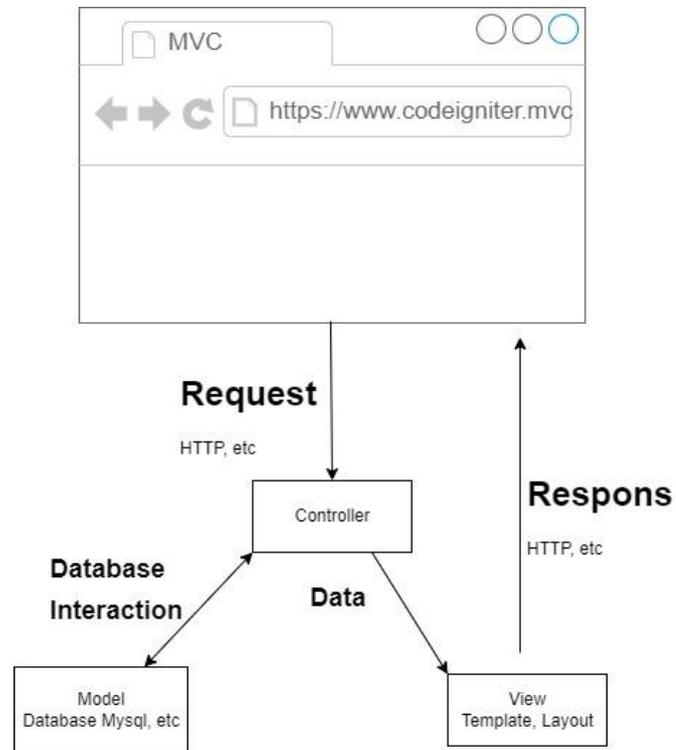
Hypertext Preprocessor (PHP) adalah sebuah bahasa *script* berbasis server (*server-side*) yang mampu mem-parsing kode php dari kode web dengan ekstensi .php, sehingga menghasilkan tampilan *website* yang dinamis di sisi *client* (*browser*).

PHP pertama kali dikembangkan oleh seorang programmer bernama Rasmus Lerdorf pada tahun 1995. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakan PHP/FI sehingga banyak pemrograman yang tertarik untuk ikut mengembangkan PHP. Dan selanjutnya pada tahun 1997 perusahaan bernama Zend, mengembangkan interpreter PHP tersebut menjadi lebih baik (Sari, Abdilah, & Sunarti, 2019).

2.11 *Codeigniter*

Framework adalah kerangka kerja atau sekumpulan file-file yang sudah terinclude, yang mana di dalam file tersebut terdapat perintah kode program dan fungsi dasar untuk melakukan tugas tertentu. Secara *visual*, *framework* merupakan potongan-potongan kode program atau fungsi yang siap pakai. Kita cukup tahu cara penerapan beserta penggunaannya (Purbadian, 2016).

Codeigniter adalah framework PHP yang memakai sistem *Model View Controller* (MVC), sehingga secara garis besar terdapat tiga kompone utama, yaitu model, view, dan controller (Basuki, 2016).



Gambar 2.3 Hubungan MVC di Codeigniter