

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Penelitian yang dilakukan oleh Salim dkk., (2019) dengan judul Sistem Informasi Pelayanan Kesehatan Masyarakat Berbasis Web Pada Rumah Sakit Rimbo Medika. Tujuan dari penelitian tersebut yaitu membuat Sistem informasi pengolahan data jaminan kesehatan pada masyarakat dapat membantu dalam menyelesaikan permasalahan yang dialami pada bidang pelayanan kesehatan. Hasil dari penelitian tersebut yaitu Penelitian ini menghasilkan sistem informasi pelayanan kesehatan masyarakat yang memiliki beberapa fitur diantaranya pendaftaran pasien untuk petugas pendaftaran pasien sehingga data pasien dapat tersimpan dengan baik, mengurangi kesalahan dalam pencatatan serta penumpukan berkas.

Penelitian yang dilakukan oleh Gunadhi & Indra (2017) dengan judul penelitian Rancang Bangun Sistem Informasi Kesehatan Berbasis Website Di Kabupaten Garut mempunyai tujuan yaitu memberikan informasi untuk masyarakat peserta BPJS mengakses informasi layanan dokter/ Poliklinik yang dirujuk BPJS di Kabupaten Garut. Hasil dari penelitian ini yaitu dengan adanya sistem informasi kesehatan ini sehingga dapat membantu dan mempermudah masyarakat peserta BPJS dalam layanan fasilitas kesehatan sesuai lokasi kepesertaan di Kabupaten Garut.

Penelitian yang dilakukan oleh Astuti dkk., (2022) dengan judul Perancangan Sistem Informasi Kelengkapan Persyaratan Klaim BPJS Rawat Jalan Di Rumah Sakit Muhammadiyah Bandung menyatakan bahwa dengan terciptanya suatu sistem informasi yang dapat mengetahui faktor kelengkapan berkas klaim BPJS rawat jalan yang di kembalikan oleh pihak BPJS kepada pihak Rumah Sakit.

Menurut Nurlina, (2018) dengan judul prosiding Perancangan Sistem Informasi Pelayanan Jaminan Pemeliharaan Kesehatan Pada Kantor PT. Jamsostek (Persero) Cabang Makassar menyatakan bahwa dengan adanya pengembangan sistem tersebut mampu meningkatkan mutu pengolahan data sehingga penyampaian informasi dapat berjalan lancar.

Penelitian dengan judul Sistem Informasi Pengajuan Klaim Peserta Jaminan Kesehatan (JKN) Berbasis Web Di Rumah Sakit Dr. M.M. Dunda Limboto yang dilakukan oleh Lasena & Abuniy (2020) menyatakan bahwa sistem informasi pengajuan klaim peserta Jaminan kesehatan dapat membantu Rumah Sakit dalam meningkatkan pelayanan kesehatan untuk peserta JKN.

Penelitian yang dilakukan oleh Widyastuti & Muttaqien, (2019) dengan judul Rancang Bangun Sistem Informasi Lowongan Kerja Berbasis Web Menggunakan Framework Codeigniter menyatakan bahwa penggunaan framework codeigniter yang berbasis php dengan penerapan konsep OOP dan MVC membuat pengembangan web menjadi lebih terstruktur, sehingga memudahkan developer dalam maintenance dan pengembangan website yang dibangun.

Tabel 2.1. Penelitian Terkait

No	Nama Peneliti	Judul	Metode	Hasil Penelitian	Perbedaan
1	(Salim dkk, 2019)	Sistem Informasi Pelayanan Kesehatan Masyarakat Berbasis Web Pada Rumah Sakit Rimbo Medika.	Metode perancangan sistem menggunakan <i>Unified Modelling Language</i> . Metode pengembangan sistem menggunakan waterfall	Hasil dari penelitian tersebut yaitu Penelitian ini menghasilkan sistem informasi pelayanan kesehatan masyarakat yang memiliki beberapa fitur diantaranya pendaftaran pasien untuk petugas pendaftaran pasien sehingga data pasien dapat tersimpan dengan baik, mengurangi kesalahan dalam pencatatan serta penumpukan berkas	Sistem yang akan dibangun penulis menggunakan framework dari <i>codigniter</i> . Metode pengembangan sistem menggunakan metode prototype.

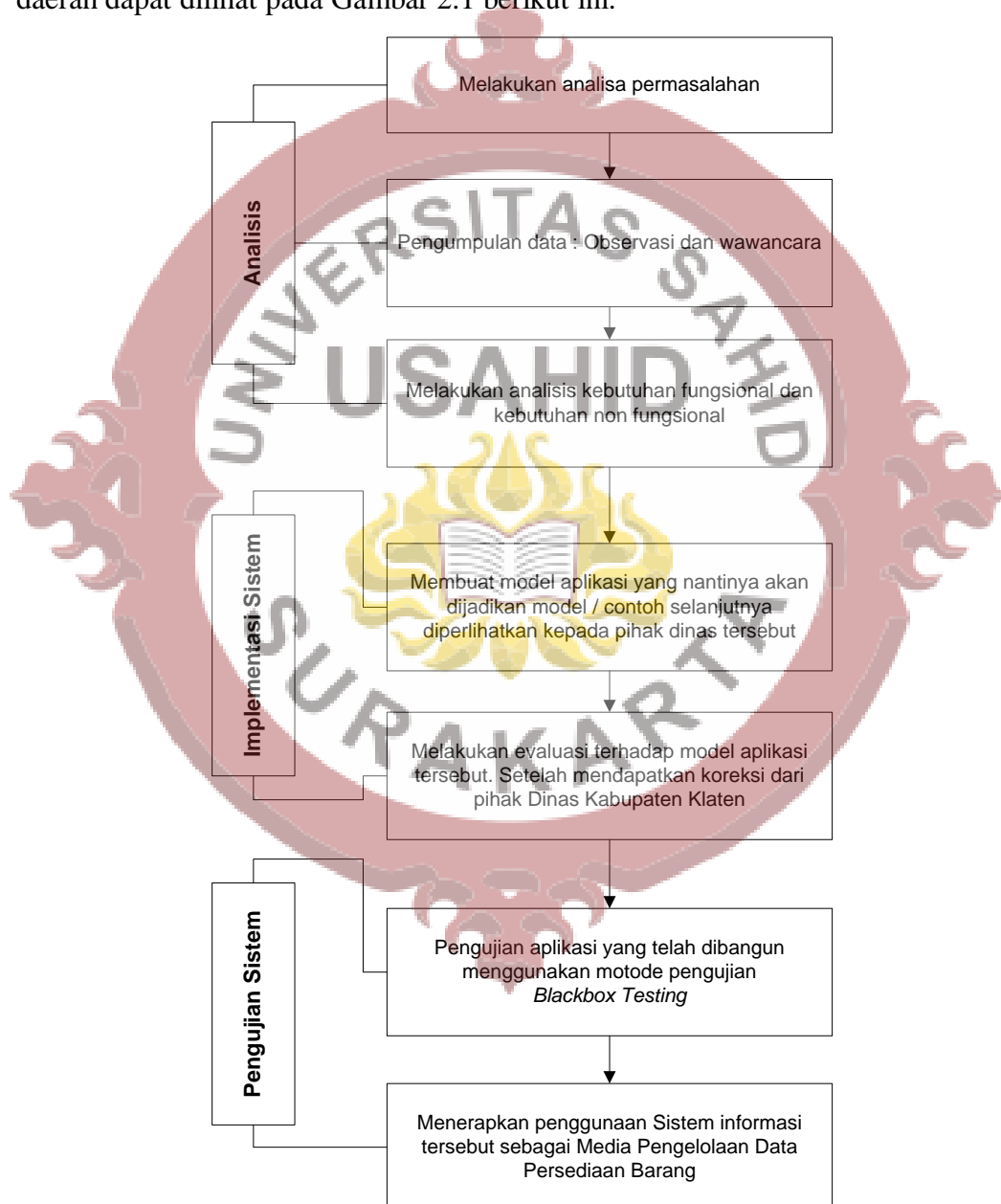
No	Nama Peneliti	Judul	Metode	Hasil Penelitian	Perbedaan
2	(Gunadhi & Indra, 2017)	Rancang Bangun Sistem Informasi Kesehatan Berbasis Website Di Kabupaten Garut	Metode model yang digunakan pada perancangan sistem ini adalah menggunakan metode <i>Evolutionary Web Development</i>	Dengan adanya sistem informasi kesehatan ini sehingga dapat membantu dan mempermudah masyarakat peserta BPJS dalam layanan fasilitas kesehatan sesuai lokasi kepesertaan di Ka-bupaten Garut	Metode model yang digunakan pada perancangan sistem ini adalah menggunakan metode <i>Unified Modelling Language</i>
3	(Astuti dkk., 2022)	Perancangan Sistem Informasi Kelengkapan Persyaratan Klaim BPJS Rawat Jalan Di Rumah Sakit Muhammadiyah Bandung	Metode penelitian yang digunakan adalah Kualitatif dengan pendekatan deskriptif. Teknik Pengumpulan data yang dilakukan yaitu dengan cara Studi Pustaka, Wawancara, dan Observasi.	Terciptanya suatu Sistem Informasi Kelengkapan Persyaratan Klaim BPJS Rawat Jalan Di Rumah Sakit Muhammadiyah Bandung dengan menggunakan Bahasa pemrograman visual basic	Pada penelitian ini tidak terdapat metode dalam mengembangkan sistem informasi, sedangkan pada penelitian milik penulis menggunakan metode pengembangan sistem informasi yaitu menggunakan <i>Prototype Model</i> .

No	Nama Peneliti	Judul	Metode	Hasil Penelitian	Perbedaan
4	(Nurlina, 2018)	Perancangan Sistem Informasi Pelayanan Jaminan Pemeliharaan Kesehatan Pada Kantor PT. Jamsostek (Persero) Cabang Makassar	Metode pengembangan sistem informasi menggunakan <i>model waterfall</i> . Model dalam perancangan sistem menggunakan <i>Context Diagram</i> dan <i>Data Flow Diagram</i> .	Dengan adanya pengembangan sistem pada PT. JAMSOSTEK (Persero) cabang Makassar, tersebut mampu meningkatkan mutu pengolahan data sehingga penyampaian informasi dapat berjalan lancar.	Metode pengembangan sistem informasi yang digunakan oleh penulis menggunakan Prototype Model. Model dalam perancangan sistem yang digunakan oleh Penulis menggunakan <i>Unified Modelling Language</i> .
5	(Lasena & Abuniy, 2020)	Sistem Informasi Pengajuan Klaim Peserta Jaminan Kesehatan (JKN) Berbasis Web Di Rumah Sakit Dr. M.M. Dunda Limboto	Metode dalam pengembangan aplikasi menggunakan <i>Research And Development</i> .	Sistem informasi pengajuan klaim peserta Jaminan kesehatan dapat membantu Rumah Sakit dalam meningkatkan pemberian pelayanan kesehatan untuk peserta JKN serta mempermudah dalam pengadministrasian berkas – berkas Klaim Rumah Sakit Umum Dr. M.M Dunda Limboto.	Metode pengembangan sistem informasi yang digunakan oleh penulis menggunakan Prototype Model. Pada penelitian tersebut tidak terdapat metode dalam pengujian aplikasi, sedangkan pada penelitian yang dilakukan oleh Penulis metode dalam melakukan pengujian sistem informasi menggunakan metode <i>Blackbox Testing</i> .

No	Nama Peneliti	Judul	Metode	Hasil Penelitian	Perbedaan
6	(Widyastuti & Muttaqien, 2019)	Rancang Bangun Sistem Informasi Lowongan Kerja Berbasis Web Menggunakan Framework Codeigniter	Metode pengembangan sistem menggunakan <i>Waterfall Model</i> Metode pengujian sistem menggunakan <i>Blackbox Testing</i>	Penggunaan framework codeigniter yang berbasis php dengan penerapan konsep OOP dan MVC membuat pengembangan web menjadi lebih terstruktur, sehingga memudahkan developer dalam maintenance dan pengembangan website yang dibangun	Studi kasus dalam sistem informasi yang akan dikembangkan berbeda dimana Penulis mengambil studi kasus mengenai pengelolaan data Jaminan Kesehatan Daerah  Metode pengembangan sistem yang akan digunakan oleh Penulis menggunakan <i>Prototype Model</i>

## 2.2 Kerangka Pemikiran

Kerangka pemikiran merupakan turunan dan spesifikasi dari masalah penelitian. Masalah penelitian adalah bagian dari kerangka pemikiran. Kerangka pemikiran dibangun sebagai acuan alur logika sistematis atau model konseptual yang menggambarkan kerangka pemikiran membantu dalam menguji masalah<sup>8</sup>. Adapun kerangka pemikiran dari pengembangan sistem informasi jaminan kesehatan daerah dapat dilihat pada Gambar 2.1 berikut ini.



**Gambar 2.1** Kerangka Pemikiran

## 2.3 Sistem

### 2.3.1 Pengertian Sistem

Menurut Mulyadi, (2016) sistem pada dasarnya adalah sekelompok unsur yang erat berhubungan satu dengan yang lainnya, yang berfungsi bersama – sama untuk mencapai tujuan tertentu.

Pendapat lain dari Romney, Marshall B (2015) bahwa sistem merupakan suatu rangkaian yang terdiri dari dua atau lebih komponen yang saling berhubungan dan saling berinteraksi satu sama lain untuk mencapai tujuan dimana sistem biasanya terbagi dalam sub sistem yang lebih kecil yang mendukung sistem yang lebih besar.

Selain itu menurut Hutahaean (2015) mengemukakan bahwa sistem adalah suatu jaringan kerja dari prosedur – prosedur yang saling berhubungan, berkumpul bersama – sama untuk melakukan kegiatan atau untuk melakukan sasaran tertentu.

Berdasarkan beberapa pendapat diatas maka dapat disimpulkan bahwa sistem merupakan kumpulan dari bagian – bagian atau beberapa sub sistem yang disatukan dan dirancang untuk mencapai tujuan tertentu.

### 2.3.2 Karakteristik Sistem

Suatu sistem memiliki ciri – ciri atau karakteristik yang terdapat pada sekumpulan elemen yang harus dipahami dalam mengidentifikasi pembuatan sistem. Adapun karakteristik sistem menurut Hutahaean (2015) adalah sebagai berikut :

1) **Komponen**

Sistem terdiri dari sejumlah komponen yang saling berinteraksi dan bekerja sama untuk membentuk satu kesatuan. Komponen sistem dapat berupa sub sistem atau bagian – bagian dari sistem.

2) **Batasan sistem (*boundary*)**

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem lainnya atau dengan lingkungan luar. Batasan sistem ini memungkinkan sistem dipandang sebagai satu kesatuan dan juga menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3) **Lingkungan luar sistem (*environment*)**



Apapun yang berada diluar batas dari sistem dan mempengaruhi sistem tersebut dinamakan dengan batasan sistem. Lingkungan luar yang bersifat menguntungkan wajib dipelihara dan yang merugikan harus dikendalikan agar tidak mengganggu kelangsungan sistem.

4) Penghubung sistem (*interface*)

Media penghubung diperlukan untuk mengalirkan sumber – sumber daya dari sub sistem ke sub sistem lainnya dinamakan dengan penghubung sistem.

5) Masukan sistem (*input*)

Energi yang dimasukkan kedalam sistem dinamakan dengan masukan sistem (*input*) dapat berupa perawatan dan masukan sinyal. Perawatan ini berfungsi agar sistem dapat beroperasi dan masukan sinyal adalah energi yang diproses untuk menghasilkan keluaran (*output*).

6) Keluaran sistem (*output*)

Hasil dari energi yang telah diolah dan diklasifikasikan menjadi keluaran yang berguna dinamakan dengan keluaran sistem (*output*). Informasi merupakan contoh keluaran sistem.

7) Pengolah sistem

Untuk mengolah masukan menjadi keluaran diperlukan suatu pengolah yang dinamakan dengan pengolah sistem.

8) Sasaran sistem

Sistem pasti memiliki tujuan atau sasaran yang sangat menentukan *input* yang dibutuhkan oleh sistem dan keluaran yang dihasilkan.

## 2.4 Informasi

Pengertian informasi menurut Krismiaji, ( 2015) bahwa informasi adalah data yang telah diorganisasi dan telah memiliki kegunaan dan manfaat. Definisi lain juga disampaikan oleh Romney, Marshall B, (2015) bahwa informasi (*information*) adalah data yang telah dikelola dan diproses untuk memberikan arti dan memperbaiki proses pengambilan keputusan. Sebagaimana perannya, pengguna membuat keputusan yang lebih baik sebagai kuantitas dari peningkatan informasi. Berdasarkan pengertian diatas maka dapat disimpulkan bahwa

pengertian informasi yaitu data yang diolah agar bermanfaat dalam pengambilan keputusan bagi penggunaannya.

## 2.5 Sistem Informasi

Pengertian sistem informasi menurut Krismiaji (2015) adalah cara-cara yang diorganisasi untuk mengumpulkan, memasukkan, dan mengolah serta menyimpan data, dan cara – cara yang diorganisasi untuk menyimpan, mengelola, mengendalikan, dan melaporkan informasi sedemikian rupa sehingga sebuah organisasi dapat mencapai tujuan yang telah ditetapkan.

Menurut Tarigan & Raharjo (2021) sistem informasi adalah kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi itu untuk mendukung operasi dan manajemen. Dalam arti yang sangat luas, istilah sistem informasi yang sering digunakan merujuk kepada interaksi antara orang proses – proses algoritmik, data, dan teknologi.

Berdasarkan definisi – definisi diatas maka dapat disimpulkan bahwa sistem informasi merupakan aktifitas pemanfaatan teknologi dalam mengolah data dan kemudian melaporkan hasil informasinya dengan tujuan untuk mendukung operasi serta manajemen yang ingin dicapai.

## 2.6 Unified Modeling Language (UML)

### 2.6.1 Pengertian UML

Menurut Rosa (2016) menyatakan “UML merupakan sebuah standar Bahasa yang digunakan untuk menganalisis dan merancang serta menggambarkan arsitektur program dalam pemrograman object oriented”.

UML (*Unified Modeling Language*) merupakan gabungan dari metode Booch, Rumbaugh (OMT) dan Jacobson. Tetapi UML mencakup lebih luas daripada OOAD. Pada pertengahan saat pengembangan UML, dilakukan standarisasi proses dengan OMG (Object Management Group) dengan harapan *UML* bakal menjadi bahasa standar pemodelan pada masa yang akan datang (yang sekarang sudah banyak dipakai oleh berbagai kalangan).

Tujuan *UML* diantaranya adalah :

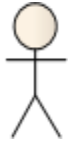

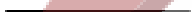



1. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
2. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan

### 2.6.2 *Use Case Diagram*

Menurut Tabrani & Aghniya (2019) menyimpulkan bahwa, “*use case* adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor”. *Use case diagram* merupakan diagram yang menjelaskan manfaat sistem dari sudut pandang orang yang berada di luar sistem atau aktor. *Use case diagram* adalah satu dari berbagai jenis diagram UML (Unified Modelling Language) yang menggambarkan hubungan interaksi antara sistem dan aktor. *Use Case* dapat mendeskripsikan tipe interaksi antara si pengguna sistem dengan sistemnya. *Use case class* digunakan untuk memodelkan dan menyatakan unit fungsi/layanan yang disediakan oleh sistem (or bagian sistem: subsistem atau class) ke pemakai. Diagram *use case* tidak menjelaskan secara detail tentang penggunaan *usecase*, namun hanya memberi gambaran singkat hubungan antara *usecase*, aktor, dan sistem.

*Use case diagram* adalah gambaran grafis dari beberapa atau semua actor, *use case*, dan interaksi diantaranya yang memperkenalkan suatu sistem. *Use case diagram* tidak menjelaskan secara detail tentang penggunaan *use case*, tetapi hanya memberi gambaran singkat hubungan antara *use case*, aktor, dan sistem. Di dalam *use case* ini akan diketahui fungsi-fungsi apa saja yang berada pada sistem yang dibuat”. Simbol-simbol *Use Case Diagram* dapat disajikan pada Tabel 2.2.




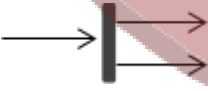

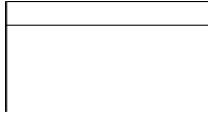
Tabel 2.2 Simbol-simbol *Use Case Diagram* (Tabrani & Aghniya, 2019)

Simbol	Keterangan
	<i>Actor</i> Orang Proses, Atau Sistem Lain Yang Berinteraksi Dengan Sistem Informasi Yang Akan Dibuat Di Luar Sistem Informasi Yang Akan Dibuat Itu Sendiri, Jadi Walaupun Simbol Dari Actor Adalah Gambar Orang, Biasanya Dinyatakan Menggunakan Kata Benda Di Awal Frase Nama <i>Actor</i> .
	<i>Use Case</i> Fungsionalitas Yang Disediakan Sistem Sebagai Unit-Unit Yang Saling Bertukar Pesar Antar Unit Atau Actor Biasanya Dinyatakan Dengan Menggunakan Kata Kerja Di Awal Frase Nama Use Case.
	<i>Asosiasi/Association</i> Komunikasi Antara <i>Actor</i> Dan Use Case Yang Berpartisipasi Pada Use Case Atau Use Case Memiliki Interraksi Dengan <i>Actor</i> .
	<i>Ekstensi/Extend</i> Relasi Use Case Tambahan Ke Sebuah Use Case Dimana Use Case Yang Ditambahkan Dapat Berdiri Sendiri Walau Tanpa Use Case Tambahan Memiliki Nama Depan Yang Sama Dengan Use Case Yang Di Tambahkan.
	<i>Generalisasi/Generalization</i> Hubungan Generalisasi Dan Spesialisasi (Umum-Khusus) Antara Dua Buah Use Case Dimana Fungsi Yang Satu Adalah Fungsi Yang Lebih Umum Dari Lainnya.
	<i>Menggunakan/Include</i> Relasi Use Case Tambahan Ke Sebuah Use Case Dimana Use Case Yang Ditambahkan Memerlukan Use Case Ini Untuk Menjalankan Fungsional Atau Sebagai Syarat Dijalankan Use Case Ini.

### 2.6.3 Activity Diagram

Menurut Menurut Tabrani & Aghniya (2019), mendefinisikan bahwa, diagram memodelkan proses bisnis dan urutan aktifitas dalam sebuah proses. Diagram ini sangat mirip dengan flowchart karena memodelkan *workflow* dari suatu aktifitas lainnya atau dari aktifitas ke status. Simbol-simbol *Activity Diagram* dapat disajikan pada Tabel 2.3.

**Tabel 2.3** Simbol-simbol *Activity Diagram* (Tabrani & Aghniya, 2019)

Simbol	Keterangan
	<i>Status Awal/Initial</i>  Status Awal Aktivitas Sistem, Sebuah Diagram Aktivitas Memiliki Sebuah Status Awal.
	<i>Aktivitas/ Activity</i>  Aktivitas Yang Dilakukan Sistem, Aktivitas Biasanya Diawali Dengan Kata Kerja.
	<i>Percabangan/ Decision</i>  Asosiasi Percabangan Dimana Lebih Dari Satu Aktivitas Digabungkan Menjadi Satu.
	<i>Penggabungan/ Join</i>  Asosiasi Penggabungan Dimana Lebih Dari Satu Aktivitas Lebih Dari Satu.
	<i>Status Akhir/ Final</i>  Status Akhir Yang Dilakukan Sistem, Sebuah Diagram Aktivitas Memiliki Sebuah Status Satu.
	<i>Swimline</i>  Memisahkan Organisasi Bisnis Yang Bertanggung Jawab Terhadap Aktivitas Yang Terjadi.

### 2.6.4 Sequence Diagram

Menurut Tabrani & Aghniya (2019), menyimpulkan bahwa, “*sequence diagram* menggambarkan interaksi antara sejumlah objek dalam urutan waktu”. Sequence Diagram adalah salah satu dari diagram-diagram yang ada pada UML, sequence diagram ini adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah object. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara object. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. Simbol-simbol *Sequence Diagram* dapat dilihat pada Tabel 2.4.

**Tabel 2.4** Simbol-simbol *Sequence Diagram* (Tabrani & Aghniya, 2019)

GAMBAR	NAMA	KETERANGAN
	<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

### 2.6.5 Class Diagram

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. Class diagram membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, class

diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.

Class diagram memiliki 3 area pokok (utama) yaitu: nama, atribut dan operasi. Didalam class diagram terdapat hubungan antar class secara konseptual, yang disebut relasi antar class (Tabrani & Aghniya, 2019) . Simbol-simbol *Class Diagram* dapat dilihat pada Tabel 2.5.


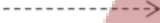

**Tabel 2.5** Simbol-simbol *Class Diagram* (Tabrani & Aghniya, 2019)

GAMBAR	NAMA	KETERANGAN
	<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

### 2.6.6 Deployment Diagram

*Deployment diagram* adalah salah satu diagram pada UML yang menunjukkan tata letak suatu sistem secara fisik, menampakkan bagian-bagian software yang berjalan pada bagian-bagian hardware. Berikut adalah simbol-simbol yang sering digunakan saat pembuatan deployment diagram dapat dilihat pada Tabel 2.6 berikut ini.

**Tabel 2.6** Simbol-simbol *Deployment Diagram* (Tabrani & Aghniya, 2019)



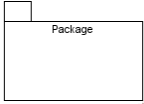
GAMBAR	NAMA	KETERANGAN
	<i>Node</i>	Merupakan sebuah alat-alat yang mendukung dalam penggunaan sistem.
	<i>Depedency</i>	Merupakan sebuah garis yang berfungsi menunjukkan operasi pada suatu component yang menggunakan component yang lain.
	<i>Link</i>	Merupakan suatu garis yang berfungsi menghubungkan obyek satu dengan obyek lain.

### 2.6.7 Component Diagram

*Componet diagram* adalah salah satu jenis diagram pada UML yang menggambarkan software pada suatu sistem. Komponen bisa berupa tabel, file data, file exe, dll (dynamic link library), dokumen dan lainlain. Berikut adalah simbol-simbol yang sering digunakan pada saat pembuatan component diagram dapat dilihat pada Tabel 2.7.



**Tabel 2.7** Simbol-simbol *Component Diagram* (Tabrani & Aghniya, 2019)

GAMBAR	NAMA	KETERANGAN
	<i>Component</i>	Merupakan sebuah gambaran yang melambangkan sebuah entitas software dalam sebuah sistem.
	<i>Depedency</i>	Merupakan sebuah garis yang berfungsi menunjukkan operasi pada suatu component yang menggunakan component yang lain.
	<i>Package</i>	Merupakan tempat komponen-komponen dimana komponen tersebut untuk dimodelkan sesuatu.

## 2.7 Fishbone Diagram

Diagram tulang ikan atau *fishbone* diagram adalah salah satu metode / tool di dalam meningkatkan kualitas. Sering juga diagram ini disebut dengan diagram Sebab-Akibat atau cause effect diagram. Penemunya adalah seorang ilmuwan jepang pada tahun 60-an. Bernama Dr. Kaoru Ishikawa, ilmuwan kelahiran 1915 di Tikyo Jepang yang juga alumni teknik kimia Universitas Tokyo. Sehingga sering juga disebut dengan diagram ishikawa. Metode tersebut awalnya lebih banyak digunakan untuk manajemen kualitas. Yang menggunakan data verbal (non-numerical) atau data kualitatif. Dr. Ishikawa juga ditengarai sebagai orang pertama yang memperkenalkan 7 alat atau metode pengendalian kualitas (7 tools). Yakni fishbone diagram, control chart, run chart, histogram, scatter diagram, pareto chart, dan flowchart (Munawar & Hafnani, 2015).

## 2.8 PHP (Hypertextprerosser)

Bahasa pemrograman PHP merupakan bahasa pemrograman untuk membuat website yang bersifat server-side scripting. PHP bersifat dinamis. PHP dapat dijalankan pada berbagai macam sistem operasi seperti Windows, Linux, dan Mac Os. Selain Apache, PHP juga mendukung beberapa web server lain, seperti Microsoft ISS, Caudium, dan PWS. PHP dapat memanfaatkan database

untuk menghasilkan halaman web yang dinamis. Sistem manajemen database yang sering digunakan bersama PHP adalah MYSQL. Namun, PHP juga mendukung sistem manajemen Database Oracle, Microsoft Access, Interbase, d-Base, dan PostgreSQL (Novendri dkk., 2019).

PHP adalah pemrograman interpreter yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. PHP disebut sebagai pemrograman *Server Side Programming*, hal ini dikarenakan seluruh prosesnya dijalankan pada server tidak dijalankan pada client. PHP merupakan suatu bahasa dengan hak cipta terbuka atau yang juga dikenal dengan istilah Open Source, yaitu pengguna dapat mengembangkan kode fungsi PHP dengan kebutuhannya (Hidayat dkk., 2017).

## 2.9 Website

*Website* merupakan kumpulan komponen yang terdiri dari teks, gambar, suara animasi sehingga merupakan media informasi yang menarik dan sangat dimininati untuk dipergunakan sebagai media berbagi informasi. Teknologi website mengolah data menjadi sebuah informasi dengan cara mengidentifikasi, mengumpulkan, mengelola dan menyediakan untuk dapat diakses secara bersama-sama (Widagdo dkk., 2018).

Website merupakan salah satu jenis layanan/fasilitas yang disediakan oleh internet yang paling banyak digunakan disamping layanan-layanan yang lainnya. Kelebihan website dibandingkan dengan fasilitas internet yang lainnya adalah karena website mampu menyajikan informasi berupa teks, gambar, suara ataupun video yang interaktif (Soejono dkk., 2018).

## 2.10 Basis Data

Basis Data adalah sebuah sistem perangkat lunak yang memungkinkan pengguna untuk dapat mendefinisikan, membuat, memelihara dan menyediakan akses kontrol ke dalam database. Teknologi Sistem Manajemen Basis Data sangat beragam, contohnya *Oracle RDBMS, MySQL, SQLite, SQL Server*, dan lainnya (Hamdallah dkk., 2020). Basis data merupakan komponen yang penting dalam

sebuah sistem informasi modern. Sebagian besar sistem informasi dewasa ini hampir semuanya menggunakan Relational Database Management System (RDBMS), Sistem Basis Data Relational (Pamungkas, 2018).

### 2.11 Codeigniter

*CodeIgniter* merupakan framework PHP yang dibuat berdasarkan *model view Controlleer (MVC)*. CI memiliki library yang lengkap untuk mengerjakan operasi-operasi yang umum dibutuhkan oleh aplikasi berbasis web misalnya mengakses database, memvalidasi form sehingga sistem yang dikembangkan mudah. CI juga menjadi satu-satunya Framework dengan dokumentasi yang lengkap dan jelas. Source code CI yang dilengkapi dengan comment didalamnya sehingga lebih memperjelas fungsi sebuah kode program dan CI yang dihasilkan sangat Bersih (*clean*) dan *search Engine Friendly (SEF)*. Codeigniter juga dapat memudahkan developer dalam membuat aplikasi web berbasis PHP, karena framework sudah memiliki kerangka kerja sehingga tidak perlu menulis semua kode program dari awal. Selain itu, struktur dan susunan logis dari codeigniter membuat aplikasi menjadi semakin teratur dan dapat fokus pada fitur-fitur apa yang akan dibutuhkan dalam pembuatan aplikasi tersebut (Sallaby & Kanedi, 2020).

### 2.12 Jaminan Kesehatan

Jaminan kesehatan adalah jaminan berupa perlindungan kesehatan agar para peserta memperoleh manfaat pemeliharaan kesehatan dan perlindungan dalam memenuhi kebutuhan dasar kesehatan yang diberikan kepada setiap orang yang telah membayar iuran Jaminan Kesehatan atau Iuran Jaminan Kesehatannya yang telah dibayar oleh Pemerintah Pusat atau Pemerintah Daerah (Peraturan Presiden Republik Indonesia, 2018).

### 2.13 Pengujian Sistem *Black Box*

*Black-Box Testing* merupakan Teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak. Blackbox Testing bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya difokuskan

pada informasi domain. Blackbox Testing memungkinkan pengembang software untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Keuntungan penggunaan metode Blackbox Testing adalah Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu; Pengujian dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan; Programmer dan tester keduanya saling bergantung satu sama lain. Kekurangan dari metode Blackbox Testing adalah Uji kasus sulit disain tanpa spesifikasi yang jelas; Kemungkinan memiliki pengulangan tes yang sudah dilakukan oleh programmer; Beberapa bagian back end tidak diuji sama sekali ( Jaya, 2018).

#### 2.14 Pengujian Sistem *Kuisisioner* (Beta)

Pengujian beta merupakan bagian dari pengujian non fungsional. Pengujian beta dilakukan dengan menyebarkan kuesioner yang diberikan kepada responden. Skala likert dirancang untuk menyakinkan responden menjawab dalam berbagai tingkatan dari setiap butir pertanyaan yang terdapat pada kuesioner. Setelah melakukan pengujian maka tahap selanjutnya melakukan analisa hasil pengujian. Untuk pengujian beta analisa dilakukan dengan melakukan perhitungan hasil survei dari pertanyaan yang sudah diisi oleh responden. Dimana perhitungan kuesioner diperoleh dengan rumus sebagai berikut (Khasanah & Murdowo, 2019).

$$Y = \frac{x}{\text{skor ideal}} \times 100 \%$$

$$X = \sum(N \times R)$$

Keterangan:

Y = nilai prosentase yang dicari

X = jumlah dari hasil perkalian nilai setiap jawaban dengan responden

N = nilai dari setiap jawaban

R = jumlah responden