



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Perangkat Lunak

Menurut Kadir (2017:2), “Perangkat lunak adalah instruksi-instruksi yang ditujukan kepada komputer agar dapat melaksanakan tugas sesuai kehendak pemakai”.

Sedangkan menurut Krisbiantoro (2018:3), “Perangkat lunak merupakan program komputer yang berisi perintah-perintah untuk melakukan proses tertentu”.

Dapat disimpulkan bahwa perangkat lunak adalah serangkaian instruksi elektronik yang ditujukan kepada komputer untuk yang berisi perintah-perintah untuk melakukan proses tertentu sesuai kehendak pemakai.

2.1.2. Komputer

Krisbiantoro (2018:3) menyatakan bahwa Komputer merupakan sekumpulan alat elektronik yang bekerja sama, dapat menerima data (*input*), mengolah data (*process*) dan memberikan informasi (*output*) serta terkoordinasi dibawah control program yang tersimpan di memori komputer.

Menurut Kadir (2017:2), “Komputer merupakan peralatan elektronik yang bermanfaat untuk melaksanakan berbagai pekerjaan yang dilakukan oleh manusia”.

Dari pengertian diatas dapat ditarik kesimpulan bahwa komputer merupakan sekumpulan alat elektronik yang bermanfaat untuk melaksanakan berbagai pekerjaan yang dilakukan oleh manusia serta terkoordinasi dibawah control program yang tersimpan di memori komputer.



2.1.3. Internet

Menurut Sujatmiko (2012:138), “Internet adalah jaringan global yang menghubungkan berjuta-juta komputer di seluruh dunia melalui jalur telepon kabel maupun satelit”. Selain itu menurut Sutarman (2012:283), “Internet adalah kumpulan dari berbagai macam jenis komputer yang saling terhubung dengan menggunakan media telekomunikasi (telepon, *wireless*, satelit, dan sebagainya) dengan jangkauan seluruh dunia atau global”.

Sehingga berdasarkan pengertian diatas dapat disimpulkan bahwa internet adalah kumpulan jaringan global dari berbagai macam komputer di seluruh dunia yang saling terhubung melalui media telekomunikasi.

2.2. Teori Judul

2.2.1. Aplikasi

Menurut Kamus Besar Bahasa Indonesia, Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer.

Dari pengertian tersebut, dapat disimpulkan bahwa aplikasi merupakan suatu program untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu untuk dieksekusi oleh komputer.

2.2.2. Fasilitas Kesehatan

Menurut Fasilitas Kesehatan merupakan suatu tempat yang digunakan untuk menyelenggarakan upaya pelayanan kesehatan baik dari segi promotif, preventif, kuratif, serta rehabilitatif yang dilakukan oleh pemerintah, pemerintah daerah dan masyarakat (Pemenkes RI No. 75, 2014).

Jadi, Fasilitas kesehatan merupakan tempat untuk menyelenggarakan upaya pelayanan kesehatan bagi masyarakat yang membutuhkan.



2.2.3. Android

Menurut Hansun *et al.* (2018:1), “Android merupakan sistem operasi *mobile* yang saat ini paling banyak digunakan di dunia, berbagai perangkat *digital* mulai dari *smartphone*, televisi hingga kacamata digital dapat menggunakan android sebagai pusat operasinya”.

Selain itu menurut Firly (2019:3), “Android merupakan sistem operasi yang berfungsi sebagai penghubung (*device*) antara pengguna dan perangkat keras pada *smartphone* atau alat elektronik tertentu sehingga hal tersebut memungkinkan pengguna dapat berinteraksi dengan *device* dan menjalankan berbagai macam aplikasi *mobile*”.

Maka dapat ditarik kesimpulan bahwa android merupakan sebuah sistem operasi yang dijalankan di berbagai alat elektronik sehingga memungkinkan pengguna dapat berinteraksi dan menjalankan berbagai macam aplikasi *mobile*.

2.2.4. Algoritma

Menurut Kadir (2017:15), “Algoritma adalah sekumpulan langkah rinci yang ditujukan untuk komputer dalam menyelesaikan suatu masalah. Langkah-langkah yang dimaksud adalah agar bisa dituangkan ke dalam program, sehingga bisa dieksekusi oleh komputer. Walaupun demikian, untuk algoritma yang sederhana, langkah-langkah yang terdapat di algoritma diuji secara manual”.

Menurut pengertian diatas dapat disimpulkan bahwa algoritma merupakan sekumpulan langkah logis yang dituangkan ke dalam program, sehingga bisa dieksekusi oleh komputer untuk menyelesaikan suatu masalah.

2.2.5. Haversine Formula

Menurut Putra *et al.* (2015:2), “Haversine Formula merupakan metode untuk mengetahui jarak antar dua titik dengan memperhitungkan bahwa bumi bukanlah sebuah bidang datar namun adalah sebuah bidang yang memiliki derajat kelengkungan. Metode Haversine Formula menghitung jarak antara 2 titik dengan berdasarkan panjang garis lurus antara 2 titik pada garis bujur dan lintang”.

Berikut bentuk Rumus Haversine Formula :



$$\text{Haversine} \left(\frac{d}{r} \right) = \text{haversine} (\theta_1 - \theta_2) + \cos(\theta_1) \cdot \cos(\theta_2) \cdot \text{haversine} (\lambda_2 - \lambda_1)$$

Keterangan :

$$\text{Haversine} (\theta) = \sin^2 \left(\frac{\theta}{2} \right) = \frac{1 - \cos(\theta)}{2}$$

θ_1 = *latitude* dari titik 1 atau posisi user.

θ_2 = *latitude* dari titik 2 atau lokasi tujuan.

λ_1 = *longitude* dari titik 1 atau posisi user.

λ_2 = *longitude* dari titik 2 atau posisi lokasi tujuan.

d = Jarak antara dua titik.

R = Radius Bumi.

Sedangkan menurut Purnawan *et al.* (2018:10), “Haversine Formula ialah persamaan penting dalam sistem navigasi, nantinya formula haversine ini akan menghasilkan jarak terpendek antara dua titik, misalnya pada bola yang diambil dari garis bujur (*longitude*) dan garis lintang (*latitude*). haversine formula merupakan penerapan dari konsep trigonometri yang merupakan bagian dari geometri”.

Jadi, haversine formula merupakan metode yang digunakan untuk mengukur jarak terpendek dari dua titik yang diambil dari garis bujur dan garis lintang.

2.2.6. Pengertian Aplikasi Pencarian Fasilitas Kesehatan di Kota Palembang Berbasis Android Menggunakan Metode Algoritma Haversine Formula

Aplikasi Pencarian Fasilitas Kesehatan di Kota Palembang Berbasis Android Menggunakan Metode Algoritma Haversine Formula adalah suatu program aplikasi berbasis android yang digunakan untuk menemukan lokasi terdekat dari fasilitas kesehatan di kota Palembang beserta informasi terkait dengan menggunakan metode algoritma haversine formula.



2.3. Teori Khusus

2.3.1. UML (*Unified Modeling Language*)

Menurut Sukamto dan Shalahuddin (2013:133), “UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

UML adalah salah satu alat bantu yang sangat andal di dunia pengembangan sistem yang berorientasi objek dan merupakan bahasa pemodelan. UML yang terdiri dari serangkaian diagram memungkinkan bagi sistem analis untuk membuat cetak biru sistem yang komperhensif kepada klien, *programmer* dan setiap orang yang terlibat dalam proses pengembangan tersebut (Munawar, 2005:132).

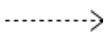
Berikut ini adalah tipe-tipe diagram UML:

2.3.1.1. *Usecase Diagram*

Usecase Diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang dibuat. *Usecase* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *usecase* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Sukamto dan Shalahuddin, 2013:155).

Adapun simbol-simbol yang ada didalam *usecase diagram* sebagai berikut:

Tabel 2.1 Simbol-simbol *usecase diagram*

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan di mana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri

Lanjutan Tabel 2.1 Simbol-simbol *usecase diagram*

			(<i>independent</i>).
3		<i>Generalization</i>	Hubungan di mana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>usecase</i> sumber secara eksplisit.
5		<i>Extend</i>	Menspesifikasikan bahwa <i>usecase</i> target memperluas perilaku dari <i>usecase</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i>
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Sumber: Sukamto dan Shalahuddin, 2013



2.3.1.2. Activity Diagram

Sukamto dan Shalahuddin (2018:161), “Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.

Adapun simbol-simbol yang ada didalam *activity diagram* sebagai berikut:

Tabel 2.2 Simbol-simbol *activity diagram*

No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

Sumber: Sukamto dan Shalahuddin, 2013

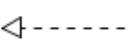
2.3.1.3. Class Diagram

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program atau programmer membuat kelas-kelas sesuai rancangan di dalam kelas agar antara dokumentasi perancangan dan perangkat lunak *sinkron* (Sukamto dan Shalahuddin, 2013:143).



Adapun simbol-simbol yang ada didalam *class diagram* sebagai berikut:

Tabel 2.3 Simbol-simbol *class diagram*

No	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Hubungan di mana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan di mana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Sumber: Sukamto dan Shalahuddin, 2013

2.3.1.4. *Sequence Diagram*

Sequence Diagram menggambarkan kelakuan objek pada *usecase* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena, itu untuk menggambarkan diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *usecase* beserta metode-metode yang dimiliki kelas yang di instansi menjadi objek itu. Membuat *sequence*



diagram juga dibutuhkan untuk melihat skenario yang ada pada *usecase* (Sukamto dan Shalahuddin, 2013:165).

Adapun simbol-simbol yang ada didalam *sequence diagram* sebagai berikut:

Tabel 2.4 Simbol-simbol *sequence diagram*

No	Gambar	Nama	Keterangan
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Sumber: Sukamto dan Shalahuddin, 2013

2.3.3. Metode Pengembangan *Rapid Application Development (RAD)*

Menurut Sarosa (2017:73), “*Rapid Application Development (RAD)* adalah salah satu metodologi yang pertama kali muncul untuk menjawab masalah kelambatan penyelesaian pengembangan sistem informasi pada SDLC konvensional”.

Menurut Whitten dan Bentley (2007:98), “*Rapid Application Development (RAD)* adalah sebuah strategi pengembangan sistem yang menekankan kecepatan dalam pengembangan melalui keterlibatan pengguna dalam pembangunan secara cepat, iteratif, dan incremental dari suatu serangkaian *prototype* dari suatu sistem yang dapat berkembang menjadi suatu sistem akhir atau versi tertentu”.

Sehingga dapat disimpulkan pengertian *Rapid Application Development (RAD)* adalah metodologi pengembangan sistem yang menekankan kecepatan dalam penyelesaian pengembangan sistem melalui keterlibatan pengguna.



2.4. Teori Program

2.4.1. Java

Java adalah bahasa pemrograman berorientasi objek dengan unsur-unsur seperti bahasa C++ dan bahasa lainnya yang memiliki *libraries* yang cocok dengan lingkungan internet. Java dapat melakukan banyak hal dalam pemrograman, seperti aplikasi interaktif, pembuatan animasi halaman web, serta pemrograman Java untuk ponsel. Java juga bisa digunakan pada internet, *handphone*, dan sebagainya (Nyura, 2010:18).

Menurut definisi Sun Microsystem, di dalam buku Shalahuddin dan Rosa (2010:1), “Java adalah nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer yang berdiri sendiri (*standalone*) ataupun pada lingkungan jaringan”.

2.4.2. PHP

PHP singkatan dari PHP: *Hypertext Preprocessor* merupakan bahasa pemrograman yang digunakan untuk membuat aplikasi berbasis *website*. Sebagai sebuah aplikasi, *website* tersebut hendaknya memiliki sifat dinamis dan interaktif. Memiliki sifat dinamis artinya, *website* tersebut bisa berubah tampilan kontennya sesuai kondisi tertentu. Interaktif artinya, *website* tersebut dapat memberi *feedback* bagi *user* misalnya, menampilkan hasil pencarian produk (Enterprise, 2017:1).

Selain itu, menurut Supono dan Virdiandry (2018:3), “PHP adalah suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat *server-side* yang dapat ditambahkan ke HTML”.

Dari pengertian di atas dapat disimpulkan bahwa PHP merupakan sebuah bahasa pemrograman yang digunakan untuk membuat aplikasi *web* yang dapat ditambahkan ke HTML.



2.4.3. MySQL

Menurut Buana (2014:2), “MySQL Merupakan *database server* yang paling sering digunakan dalam pemograman PHP. MySQL digunakan untuk menyimpan data dalam *database* dan memanipulasi data-data yang diperlukan. Manipulasi data tersebut berupa menambah, mengubah, dan menghapus data yang berada dalam *database*”.

Kadir (2017:2), “MySQL merupakan *software* yang tergolong sebagai DBMS (*Database Management System*) yang bersifat *open source*”. *Open Source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (kode yang dibuat untuk membuat MySQL), selain bentuk *executable* nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi”.

Dari pengertian di atas dapat disimpulkan bahwa MySQL merupakan suatu aplikasi yang digunakan untuk menyimpan dan memanipulasi data yang dapat dijalankan secara langsung dalam sistem operasi.

2.4.4. XAMPP

Menurut Enterprise (2017:93), XAMPP merupakan *tool* yang digunakan untuk membantu kita bekerja menggunakan MySQL *Server*, namun dengan tampilan antarmuka grafis yang lebih ramah untuk siapa pun, terlebih bagi kalangan pemula.

Menurut Ramadhan dan Uus Rusmawan (2018:3), XAMPP adalah *server* yang paling populer di kalangan *programmer*, selain fiturnya yang lengkap, XAMPP sangat mudah digunakan”.

Dari pengertian di atas dapat disimpulkan bahwa XAMPP merupakan suatu *server* untuk mengelola MySQL *Server* yang sangat mudah digunakan.

2.4.5. Android Studio

Menurut Purnomo *et al.* (2020:6), “Android studio adalah salah satu *software* untuk membuat aplikasi android. Android Studio ini banyak digunakan oleh para *start up* atau *developer* android untuk membuat sebuah aplikasi berbasis android”.



Android studio adalah IDE (*Integrated Development Environment*) resmi untuk pengembangan aplikasi Android dan bersifat *open source* atau gratis. Peluncuran Android Studio ini diumumkan oleh *google* pada 16 mei 2013 pada *event Google I/O Conference* untuk tahun 2013. Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android (Juansyah, 2015:2).

Jadi, android studio adalah *Integrated Development Environment* (IDE) utama *google* yang resmi untuk pengembangan aplikasi berbasis android.

2.4.6. Basis Data (*Database*)

Menurut Kristanto (2018:79), “Basis data adalah kumpulan data, yang dapat digambarkan sebagai aktifitas dari satu atau lebih organisasi yang berelasi”.

Sedangkan menurut Nugroho (2017:175), “*Database* atau Basis Data merupakan sekumpulan data yang sangat kompleks, kemudian data tersebut memiliki hubungan antara data yang satu dengan yang lainnya”.

Sehingga, kesimpulannya basis data ialah kumpulan data yang kompleks yang saling berhubungan satu sama lainnya.

2.4.7. Google Maps API

Google Maps API adalah sebuah layanan (*service*) yang diberikan oleh *google* kepada para pengguna untuk memanfaatkan *google maps* dalam mengembangkan aplikasi. *Google Maps* API menyediakan beberapa fitur untuk memanipulasi peta, dan menambah konten melalui berbagai jenis *services* yang dimiliki, serta mengizinkan kepada pengguna untuk membangun aplikasi *enterprise* di dalam websitenya (Mahdia dan Fiftin, 2013:164).

Menurut Sirenden (2012:47), “*Google Maps* API merupakan aplikasi antarmuka yang dapat diakses lewat javascript agar *google maps* dapat ditampilkan pada halaman *web* yang sedang kita bangun. *Google Maps* adalah layanan gratis yang diberikan oleh *google* dan sangat populer. *Google Maps* adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah.



Dengan kata lain, *google maps* merupakan suatu peta yang dapat dilihat dengan menggunakan suatu *browser*".

Berdasarkan pengertian diatas dapat ditarik kesimpulan bahwa *google maps* API ialah suatu aplikasi antarmuka yang disediakan oleh *google* untuk melakukan aktifitas tertentu yang berkaitan dengan *google maps*.

2.5. Penelitian Terdahulu

Berdasarkan penelitian yang telah dilakukan oleh Farid dan Yulanda Yunus dengan judul Analisa Algoritma Haversine Formula Untuk Pencarian Lokasi Terdekat Rumah Sakit Dan Puskesmas Provinsi Gorontalo pada tahun 2017, p-ISSN: 2087-1716 dan e-ISSN: 2548-7779. Merancang suatu aplikasi tentunya membutuhkan suatu metode yang khusus. Metode yang digunakan pada perancangan aplikasi ini adalah Haversine Formula yang merupakan metode persamaan yang penting pada navigasi, memberikan jarak lingkaran besar antara dua titik pada permukaan bola (Bumi) berdasarkan bujur dan lintang. Tujuan penelitian ini menghasilkan suatu analisa algoritma Haversine Formula lokasi pemetaan Rumah Sakit dan Puskesmas untuk pencarian lokasi terdekat ke rumah sakit satu ke rumah sakit lainnya dan puskesmas ke puskesmas lainnya.

Menurut penelitian yang dibuat oleh Aldy Putra Aldya *et al* pada tahun 2019 yang berjudul Haversine Formula Untuk Membatasi Jarak Pada Aplikasi Presensi Online, p-ISSN: 2541-1179 dan e-ISSN : 2581-1711. Implementasi metode haversine formula ini diharapkan dapat memberikan suatu solusi untuk menggantikan sistem presensi konvensional ke dalam bentuk sistem presensi digital dengan merancang dan membuat aplikasi sistem presensi online berbasis android menggunakan batasan radius GPS. Tujuan dari penelitian ini ialah untuk mengatasi permasalahan presensi tersebut dengan membuat aplikasi presensi online menggunakan *smartphone* dan menggunakan fitur GPS, untuk membatasi area lokasi proses presensi digunakan rumus harvesine. Hasil penelitian ini adalah sistem yang mampu mendeteksi lokasi dan menghitung jarak sebagai pembatas toleransi presensi.



Dalam jurnal penelitian yang dibuat oleh Yulianto *et al* pada tahun 2018 yang berjudul Penerapan Formula Haversine Pada Sistem Informasi Geografis Pencarian Jarak Terdekat Lokasi Lapangan Futsal e-ISSN : 2597-4963 dan p-ISSN 1858-4853. Pada penelitian ini akan membangun sebuah sistem berbasis *website* yang diharapkan dapat menyajikan informasi dan lokasi lapangan futsal serta ditunjang dengan pencarian jarak terdekat. Sistem Informasi Geografis dapat menjadi solusi dalam menampilkan informasi dan lokasi lapangan futsal yang disajikan dalam bentuk peta digital. Pencarian jarak terdekat memiliki banyak metode, persamaan, atau formula salah satunya formula Haversine. Hasil dari penelitian ini ialah sistem yang mampu menghasilkan pemetaan lokasi lapangan futsal di kota Samarinda dilengkapi dengan informasi mengenai lokasi tersebut.

Berdasarkan jurnal penelitian yang dibuat oleh Irmawati dan Galih Bayu Aji Listianto pada tahun 2018 dengan ISSN: 2338-8161 yang berjudul Metode *Rapid Application Development* (RAD) Pada Perancangan Website Inventory Pt. Sarana Abadi Makmur Bersama (S.A.M.B) Jakarta. Dimana metode *Rapid Application Development* (RAD) ini terdiri dari 4 tahapan utama yang biasanya digunakan dalam pengembangan sistem. Hasil dari penelitian ini ialah sebuah aplikasi berbasis *website* yang dapat berjalan disemua sistem operasi selama ada *browser* dan koneksi internet, yang bertujuan agar data inventori asset barang dapat digunakan relatif cepat, relatif tepat, dan relatif data lebih akurat