

BAB II

TINJAUAN PUSTAKA

2.1. Marketplace

E-marketplace adalah ruang elektronik yang digunakan sebagai sarana penjual dan pembeli bertemu dan melakukan berbagai jenis transaksi, dengan adanya *e-marketplace* penjual dapat menawarkan barang atau jasa tanpa memiliki tempat usaha, penjual dan pembeli dapat melakukan interaksi kapan saja dan dimana saja selama terhubung dengan internet.

2.2. Android

Android adalah sistem operasi yang dikeluarkan oleh Google. Android dibuat khusus untuk smartphone dan tablet dengan sistem operasi untuk perangkat mobile berbasis Linux yang awalnya dikembangkan oleh Android Inc. Android terdiri dari system operasi, *middeleware* dan aplikasi. Android merupakan istilah dalam Bahasa Inggris yang berarti robot yang menyerupai manusia. Pada tahun 2005, Google secara resmi telah membeli android, sehingga pengembangan Android sepenuhnya berada di tangan Google. Dalam proses pengembangan sistem operasi Android, dibentuklah organisasi *Open Handset Alliance*. Google merilis *software open source* untuk Android, sehingga dapat berkontribusi untuk mengembangkan Android, karena *lisensi open source* sangat mendukung untuk mengembangkan produknya dengan aman [4].

2.3. LBS

Layanan Berbasis lokasi adalah layanan informasi yang dapat diakses melalui *mobile device* dengan menggunakan *mobile network*, yang dilengkapi kemampuan untuk memanfaatkan lokasi dari *mobile device* tersebut. LBS memberikan kemungkinan komunikasi dan interaksi dua arah. Oleh karena itu pengguna

memberitahu penyedia layanan untuk mendapatkan informasi yang dia butuhkan, dengan referensi posisi pengguna tersebut. Layanan berbasis lokasi dapat digambarkan sebagai suatu layanan yang berada pada pertemuan tiga teknologi yaitu: *Geographic Information System, Internet Service, dan Mobile Devices* [5]

2.4. Rational Unified Process (RUP)

Perancangan pada aplikasi ini akan menggunakan *Rational Unified Process* (RUP) yang memiliki aktifitas yang berfokus pada pengemabangan model menggunakan UML dan metode ini sangat cocok dalam pengembangan perangkat lunak *Object Oriented Programming* (OOP) karena dapat memisahkan operasi-operasi atau komponen yang terdapat didalam sebuah objek-objek yang berbeda. Dalam metode ini terdapat 4 tahapan dalam pengembangannya [6], yaitu ;

a. Fase Inception

pada fase ini menentukan ruang lingkup pengembangan sistem dari hasil observasi dan wawancara dengan melakukan analisis kebutuhan dan pemodelan *use-case* diagram dan activity diagram

b. Fase Elaboration

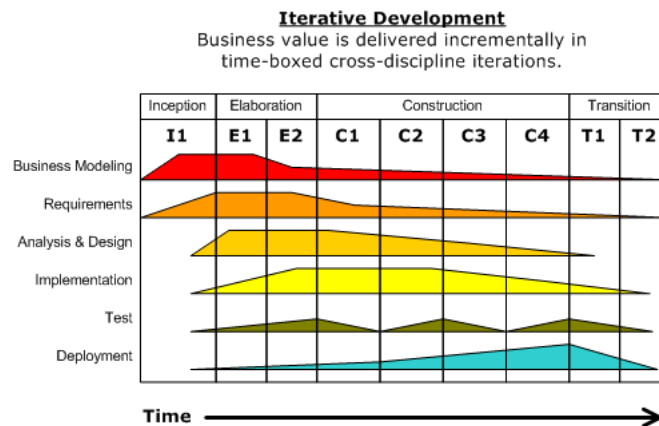
Hasil dari fase ini nantinya berupa class diagram, perancangan antarmuka, perancangan database dan perancangan algoritma.

c. Fase Construction

Fase ini menghasilkan implementasi sistem informasi berbasis aplikasi dan kode program

d. Fase Transition

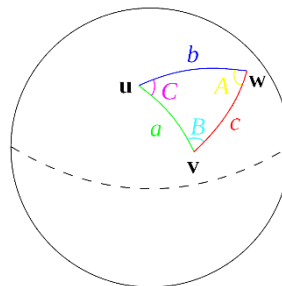
Pada fase transisi ini adalah fase deployment atau instalasi sistem. Pada fase ini dilakukan pengujian *User Acceptance Testing* (UAT)



Gambar 2.1 *Rational unified process architecture*

2.5. *Algoritma Haversine*

Haversine ini pertama kali ditemukan oleh James Andrew pada tahun 1805, dan pertama kali digunakan oleh Josef de Mendoza y Ríos pada tahun 1801. Istilah *Haversine* sendiri diciptakan pada tahun 1835 oleh Prof. James Inman. Josef de Mendoza y Ríos menggunakan *Haversine* untuk pertama kalinya dalam penelitiannya tentang "Masalah Utama Astronomical Nautical", Proc. Royal Soc, 22 Desember 1796. *Haversine* digunakan untuk mencari jarak antara bintang-bintang di tempat pertama. Rumus *Haversine* memiliki hukumnya sendiri yaitu semua persamaan yang digunakan berdasarkan bentuk bola bumi dengan menghilangkan faktor bumi yang agak elips (faktor ellipsoidal). Ini special kasus rumus umum dalam trigonometri bola yang berhubungan dengan sisi dan sudut dari bola segi tiga. Perhitungan jarak dari satu titik ke titik lain di permukaan bumi dipengaruhi oleh a derajat kelengkungan tertentu [3].



Gambar 2.2 *Haversine's law*

2.6. Kotlin

Kotlin merupakan Bahasa pemrograman yang dapat digunakan untuk berbagai macam pengembangan aplikasi, baik itu server atau backend, website, maupun mobile Android. Bahkan saat ini tengah dikembangkan Kotlin/Native. Kotlin/Native memungkinkan developer untuk menggunakannya sebagai bahasa pemrograman dalam pengembangan aplikasi di platform lain seperti embedded system, desktop, macOS, dan iOS. Berikut ini adalah beberapa kelebihan mengembangkan aplikasi Android dengan Kotlin :

a. Compatibility

Kotlin sepenuhnya kompatibel dengan JDK 6. Ini memastikan bahwa aplikasi yang dibangun dengan Kotlin dapat berjalan pada perangkat Android yang lebih lama tanpa ada masalah. Android Studio pun mendukung penuh pengembangan dengan bahasa Kotlin.

b. Performance

Dengan struktur bytecode yang sama dengan Java, aplikasi yang dibangun dengan Kotlin dapat berjalan setara dengan aplikasi yang dibangun dengan Java. Terdapat juga fitur seperti inline function pada Kotlin yang membuat kode yang dituliskan dengan lambda bisa berjalan lebih cepat dibandingkan kode yang sama dan dituliskan dengan Java.

c. Interoperability

Anda dapat menggunakan bahasa Kotlin bersamaan dengan bahasa Java, tanpa harus memigrasikan semua kode lama Anda ke Java. Sehingga Anda dapat memanggil kode Java dari Kotlin dan sebaliknya. Inilah alasan yang menyebabkan Kotlin menjadi cepat diterima oleh developer.

d. Compilation Time

Kotlin mendukung kompilasi inkremental yang efisien. Oleh karena itu, proses build biasanya sama atau lebih cepat dibandingkan dengan Java.

2.7. MYSQL

MYSQL merupakan proyek open-source yang dimiliki oleh Oracle yang dapat menyimpan data berkapasitas sangat besar hingga berukuran Giga Byte. MySQL adalah sebuah database server yang *free*, artinya bebas dalam menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya. MySQL Selain database server, MySQL juga merupakan program yang dapat mengakses suatu database MySQL yang berposisi sebagai server, yang berarti program kita berposisi sebagai *Client*. Jadi MySQL adalah sebuah database yang dapat digunakan sebagai *Client* maupun server [7]

2.8. Unified Modelling Language (UML)

UML (*Unified Modelling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti. UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, *Object Modeling Technique* (OMT) dan *Object Oriented Software Engineering* (OOSE). Metode ini menjadikan proses analisis dan desain ke dalam empat tahapan iteratif, yaitu : identifikasi kelas-kelas dan objek-objek, *identifikasi semantic* dari hubungan objek dan kelas tersebut, perincian *interface* dan implementasi. UML dibangun atas model 4+1 view. Yaitu *LogicalView*, *Development View*, *Process View*, *Physical View* dan *Scenario*. [8]

2.9. Use Case Diagram

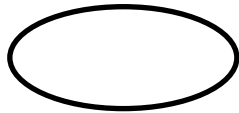
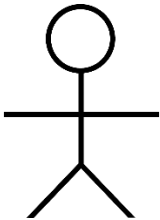
Use Case merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Syarat penamaan pada *use*



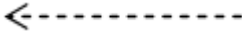
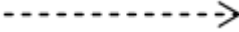

case adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut actor dan *use case*.

- Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, jadi walaupun symbol dari aktor adalah gambar orang tapi aktor belum tentu merukan orang.
- Use case merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor [8]

Tabel 2.1 berikut ini adalah Simbol-simbol yang digunakan dalam *Use Case Diagram* :

Tabel 2.1 Simbol *Use case Diagram*.



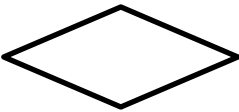
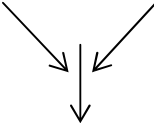
No	Simbol	Keterangan
1	Nama <i>UseCase</i> 	Fungsionalitas yang disediakan sistem sebagai unit yang saling bertukar pesan antara unit atau <i>factor</i> , biasanya dinyatakan dengan menggunakan kata kerja awal frase nama <i>Use Case</i>
2	Aktor 	Orang, proses atau sistem lain yang berinteraksi dengan sistem yang akan dibuat di luar sistem yang akan dibuat itu sendiri. Jadi, walaupun symbol dari aktor adalah gambar orang, tapi aktor belum tentu orang. Biasanya. dinyatakan menggunakan kata benda di awal frase nama aktor.


3	Asosiasi 	Komunikasi antara aktor dan <i>Use Case</i> , atau <i>Use Case</i> dan Aktor
4	Generalisasi 	Hubungan <i>Generalisasi</i> dan <i>Spesialisasi</i> (Umum - Khusus) antara dua buah <i>Use Case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
5	Ekstensi / <i>Extend</i> 	Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>Use Case</i> tambahan.
6	<i>Include</i>  <i>Uses</i> 	Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> , dimana <i>Use Case</i> yang ditambahkan memerlukan <i>Use Case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>Use Case</i> ini.

2.10. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktifitas yang dilakukan sistem[8]. Tabel 2.2 berikut ini adalah Simbol-simbol yang digunakan dalam *Activity Diagram*:

Tabel 2.2 Simbol *Activity Diagram*.

No	Simbol	Keterangan
1	Status Awal 	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
2	Aktivitas 	Aktivitas yang dilakujan suatu sistem.
3	Percabangan / <i>Decision</i> 	Simbol ini digunakan jika ada pilihan aktivitas lebih dari satu atau menggabungkan dua kegiatan pararel menjadi satu.
4	Penggabungan / <i>Join</i> 	Penggabungan / <i>Join</i> digunakan untuk menunjukkan adanya kegiatan yang digabungkan.

5	Status Akhir 	Status Akhir, adalah akhir dari aktifitas sebuah sistem.
---	---	--

2.11. Penelitian Terkait

Dalam penyusunan penelitian yang dilakukan, penulis mengumpulkan penelitian yang berkaitan dengan latar belakang penelitian dari penelitian-penelitian terdahulu sebagai referensi dan acuan dalam membuat penelitian dengan menggunakan *Haversine Formula*. Daftar penelitian terdahulu yang berkaitan dengan penelitian yang akan dikerjakan sebagai berikut :

1. Implementation of *Haversine* formula for school location tracking

Pada penelitian yang dilakukan oleh P. Dauni, M. D. Firdaus, R. Asfariani, M. I. N. Saputra, A. A. Hidayat, and W. B. Zulfikar (2019), membahas mengenai Implementasi formula *Haversine* untuk sekolah pelacakan lokasi, Aplikasi Informasi Lokasi Sekolah adalah aplikasi yang dikembangkan untuk menghasilkan berbagai informasi dan dapat dilihat dan disebarluaskan guna mendukung pelacakan sekolah dan lokasi mereka. Aplikasi Informasi Lokasi Sekolah dilakukan dengan mengumpulkan data analisis, tahap desain, dan implementasi. Aplikasi ini dibuat oleh basis Android untuk klien yang menggunakan Bahasa pemrograman Java dan server berbasis web untuk mengolah data menggunakan HTML dan PHP disertai dengan SQL sebagai pengelola database. Namun, metode Formula *Haversine* adalah digunakan untuk menentukan jarak antara lokasi pengguna dengan lokasi sekolah yang diinginkan. Berdasarkan pembahasan penelitian dalam pengembangan aplikasi ini, dapat disimpulkan bahwa Sekolah Aplikasi Informasi Lokasi dapat memberikan beberapa informasi dan lokasi sekolah yaitu terdekat dengan pengguna. Aplikasi ini juga dapat menghitung jarak dari lokasi pengguna ke tujuan menggunakan Metode Formula *Haversine*. Sehingga, pengguna yang menggunakan aplikasi ini akan mendapatkan kemudahan untuk menemukan sekolah terdekat pilihan mereka [3].

2. Penerapan Metode *Haversine* Formula Untuk Penentuan Titik Kumpul pada Aplikasi Tanggap Bencana

Pada penelitian yang dilakukan oleh Abadi Nugroho, Rio Jumardi, dan Nur Fajariah Ramadhania (2020), membahas mengenai kurangnya berita atau informasi yang sulit dipercaya sumbernya dan lambatnya penanganan kebencanaan karena harus mengecek keakuratan informasi kejadian bencana terlebih dahulu. Tujuan dari penelitian ini adalah membuat aplikasi yang dapat memberikan informasi bencana dan mencari titik kumpul dengan jarak terdekat menggunakan metode *Haversine* Formula serta teknologi Google Maps. dengan Metodologi pengembangan sistem yang digunakan adalah GRAPPLE (Guidelines for Rapid APPLICATION Engineering). Dari hasil penelitian ini, tercipta aplikasi informasi bencana dan titik kumpul untuk menentukan jarak terdekat dengan lokasi pengguna. Berdasarkan dari pengujian kelayakan aplikasi dari 10 responden menyatakan aplikasi ini layak digunakan dan dipublikasikan ke masyarakat [9].

3. Aplikasi Pencarian Pekerja Jasa Rumah Tangga di Sekitar dengan Metode Radius dan Rating Berbasis Android

Pada penelitian yang dilakukan oleh Jerry Erlangga, Eliyani (2021), membahas mengenai Berkurangnya lapangan pekerjaan akibat penurunan beberapa sektor usaha dikarenakan masa krisis akibat pandemi, membuat banyak pekerja memilih beralih menjadi pekerja rumah tangga untuk memperoleh penghasilan. Penelitian ini bertujuan untuk membuat aplikasi yang dapat mempertemukan pencari kerja dengan pemberi kerja. Digunakan Metode Radius yang memanfaatkan Google Maps API dalam mengakses lokasi pemberi kerja dan pencari kerja sehingga pencari kerja dapat dengan mudah mencari pekerjaan yang berada disekitarnya. Pekerjaan yang telah diselesaikan akan diberikan rating oleh pemberi kerja sesuai tingkat kepuasan terhadap hasil pekerjaan pencari kerja. Aplikasi yang dibangun sudah memiliki peningkatan keamanan seperti enkripsi koneksi web service ke *server Backend*, verifikasi OTP nomor telepon, dan verifikasi data pengguna

dengan foto kartu identitas. Menggunakan metode *black box testing*, semua fitur dapat berjalan secara optimal. Pengujian dilakukan terhadap 20 pengguna dengan menggunakan mobile phone berbasis android dengan beragam versi. Hasil ujicoba menunjukkan bahwa semua fitur dapat berjalan secara optimal pada Android versi 7 hingga 11 dengan akurasi 95% [10].

4. Aplikasi *Marketplace* Jasa Reparasi Barang Elektronik Berbasis Android

Pada penelitian ini dilakukan pembuatan Aplikasi *marketplace* yang dibuat berfokus pada jasa reparasi elektronik sebagai salah satu jasa yang paling banyak dicari semua orang. Pembuatan aplikasi mengikuti metode pengembangan perangkat lunak waterfall. Untuk mengetahui tingkat kemudahan aplikasi, sebuah evaluasi menggunakan kuesioner juga dilakukan pada penelitian ini. Berdasarkan hasil evaluasi, 56% jawaban “sangat setuju”, dan 38% jawaban “setuju” membuktikan bahwa aplikasi yang dibuat mampu membantu seseorang untuk memenuhi kebutuhan jasa reparasi barang elektronik dengan cara yang lebih mudah[11]

5. Penerapan Metode *Haversine* Pada Sistem Informasi Geografis Pencarian Lokasi Laundry Terdekat di Kota Makassar

Pada penelitian yang dilakukan oleh Muh. Iqbal , Poetri Lestari L.B, dan Nia Kurniati (2021), membahas mengenai jasa pencucian dengan jumlah kilogram serta jumlah satuan baju yang di laundry. Tujuan Penelitian ini dapat memberikan rekomendasi jasa lokasi laundry terdekat dengan posisi *user* dan memberi petunjuk arah menuju lokasi tempat laundry sehingga meminimalisir tenaga waktu dan biaya. Metode yang digunakan dalam penelitian ini adalah *Haversine* Formulamenghitung jarak antara dua titik di bumi berdasarkan panjang garis lurus antara dua titik tanpa mengabaikan kelengkungan yang di miliki bumi, dengan menerapkan teknologi tersebut dalam sebuah SIG pengguna sistem dapat mengetahui posisinya terhadap lokasi tertentu walupun masih mengabaikan relief bumi yang sebenarnya. Berdasarkan hasil penelitian setelah melakukan uji coba di beberapa

smartphone/mobile device, aplikasi dapat berjalan dengan baik operating sistem yang berbeda. Berdasarkan uji coba kecepatan dalam menentukan lokasi berdasarkan jarak tempuh terdekat, daerah (lingkungan) dan cuaca dapat mempengaruhi kecepatan dalam menentukan lokasi *user* berdasarkan jarak terdekat.

Tabel 2. 3 *Perbandingan Penelitian Terkait.*

No	Peneliti	Judul Penelitian	Metode	Hasil
1	P Dauni, M D Firdaus, R Asfariani, M I N Saputra, A A Hidayat dan W B Zulfikar (2019)	Implementation of <i>Haversine</i> formula for school location tracking	<i>Haversine</i>	Aplikasi Informasi Lokasi dapat memberikan beberapa informasi dan lokasi sekolah yaitu terdekat dengan pengguna. Aplikasi ini juga dapat menghitung jarak dari lokasi pengguna ke tujuan menggunakan Metode Formula <i>Haversine</i> . Sehingga, pengguna yang menggunakan aplikasi ini akan mendapatkan kemudahan untuk menemukan sekolah terdekat pilihan mereka
2	Abadi Nugroho, Rio Jumardi,	Penerapan Metode	<i>Haversine</i>	Aplikasi informasi bencana dan titik

	Nur Fajariah Ramadhania (2021)	<i>Haversine</i> Formula Untuk Penentuan Titik Kumpul pada Aplikasi Tanggap Bencana		kumpul ini menghasilkan jarak tempuh terdekat Berdasarkan hasil dari analisis kuesioner uji kelayakan aplikasi, dari 10 responden menyatakan bahwa aplikasi informasi bencana dan titik kumpul ini layak digunakan dan dipublikasikan untuk masyarakat umum.
3	Jerry Erlangga, Eliyani (2018)	Aplikasi Pencarian Pekerja Jasa Rumah Tangga di Sekitar dengan Metode Radius dan Rating Berbasis Android	<i>Haversine</i>	Aplikasi ini dapat berjalan dengan baik dengan akurasi 95% dari hasil uji coba terhadap 20 pengguna dimana 19 pengguna dapat memasang dan menjalankan aplikasi dengan baik sedangkan 1 pengguna tidak berhasil melakukan pemasangan dikarenakan versi Android dibawah versi 7
4	Asriana Dyah Kusumastuti, Donny Avianto,	Aplikasi <i>Marketplace</i> Jasa Reparasi		Penelitian kali ini telah berhasil

	Adityo Permana Wibowo (2020)	Barang Elektronik Berbasis Android		menghasilkan aplikasi <i>marketplace</i> di bidang jasa reparasi elektronik. Dalam rangka memberikan akses yang mudah bagi penggunaannya
5	Muh. Iqbal, Poetri Lestari L.B., Nia Kurniati (2021)	Penerapan Metode <i>Haversine</i> Pada Sistem Informasi Geografis Pencarian Lokasi Laundry Terdekat di Kota Makassar	<i>Haversine</i>	n setelah melakukan uji coba di beberapa smartphone/mobile device, aplikasi dapat berjalan dengan baik operating sistem yang berbeda. Aplikasi ini dapat membantu <i>user</i> dalam mencari tempat laundry. Berdasarkan uji coba kecepatan dalam menentukan lokasi berdasarkan jarak tempuh terdekat, daerah (lingkungan) dan cuaca dapat mempengaruhi kecepatan dalam menentukan lokasi <i>user</i> berdasarkan jarak terdekat

Dari 4 hasil penelitian yang relevan memberikan informasi mengenai metode *Haversine* yang akan digunakan seperti beberapa contoh tools yang digunakan dalam peneltian pada pencarian jasa servis terdekat. Dan dari 4 penelitian terkait diatas membahas mengenai pengukuran jarak dalam menentukan pencarian lokasi terdekat yang dapat menjadi acuan bagi penulis sebagai saran dalam pembuatan penelitian pencarian lokasi jasa servis terdekat menggunakan metode *Haversine* di wilayah Natar, Lampung Selatan - Bandar Lampung