

BAB II

LANDASAN TEORI

2.1 Definisi Model Pengembangan Perangkat Lunak *Prototype*

Menurut[10], *prototype* merupakan metode pengembangan perangkat lunak, yang berupa model fisik kerja sistem dan berfungsi sebagai versi awal dari sistem. Model *prototype* merupakan model pengembangan sebagai perantara pengembang dan pengguna agar dapat berinteraksi dalam proses kegiatan, model *prototype* Memiliki tahapan- tahapan sebagai berikut :

- a. Pengumpulan kebutuhan.
- b. Proses desain.
- c. Membangun *prototype*.
- d. Evaluasi *prototype*.

Jika semua langkah sudah terpenuhi maka selanjutnya melakukan pembuatan produk yang sesungguhnya.



Gambar 2. 1 Tahapan - Tahapan *Prototype*[10].

2.2 Definisi Metode Pengembangan Perangkat Lunak OOAD

Metode *Object Oriented Analysis and Design* (OOAD) adalah metode untuk menganalisis dan merancang sistem dengan pendekatan berorientasi *object*[11]. Dimana analisa identitas menjelaskan bagaimana *user* dapat dibedakan dengan

object melalui *event* yang dilakukan. Kelebihan OOAD yaitu menggunakan 2 metode menjadi 1 metode yaitu:

2.3.1 OOA (*Object Oriented Analysis*)

Tahap pertama dalam aktivitas OOA adalah pengidentifikasian objek dan kelas, yang meliputi orang, tempat, entitas, organisasi, konsep, ataupun kejadian tertentu. Pemodelan pada tahap ini dibuat dengan menggunakan *use case diagram* yang mana merupakan pemodelan dari perilaku (*behavior*) sistem[12].

2.3.2 OOD (*Object Oriented Design*)

Dalam tahap ini akan dilakukan pemodelan dengan menggunakan *activity diagram* dan *class diagram*. *Activity diagram* menunjukkan proses bisnis dari sistem, *class diagram* menggambarkan sistem berdasarkan kelas-kelas yang membentuk sistem. Dalam suatu kelas terdapat atribut (*attribute*) dan metode (*method*). Kelas memiliki *variabel-variabel* yang disebut atribut[12].

Kesimpulan penulis pada metode OOAD yaitu, perancangan sistem perangkat lunak yang berfokus pada desain arsitektur dari *activity diagram* dan *class diagram*, serta analisa yang berfokus pada entitas, konsep, kelas dari objek pada penggunaan *use case diagram*.

2.3 Algoritma Perhitungan Jarak *Haversine Formula*

Metode *haversine formula* merupakan sebuah cara yang digunakan untuk menghitung jarak antara titik permukaan bumi dengan menggunakan garis lintang (*longitude*) dan garis bujur (*latitute*) sebagai variable inputan, *haversine formula* adalah sebuah persamaan penting pada navigasi yang dapat memberikan jarak lingkaran besar antara dua titik pada permukaan bumi atau benda bulat berdasarkan bujur dan lintang. Dengan mengamsusikan bahwa bumi berbentuk sebuah lingkaran atau bulat sempurna dengan jari-jari R 6.367, 45 km dan lokasi dari 2 titik dikoordinat bola (lintang dan bujur) masing-masing dengan lat1, long1 dan lat2, long2[7].

2.4 Teori Pendukung

Ada beberapa definisi teori-teori pendukung dari judul penelitian yang akan digunakan dalam membangun aplikasi, sebagai berikut:

2.4.1 Definisi Aplikasi

Aplikasi merupakan penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan. Dengan kata lain aplikasi dapat diartikan juga sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu[5]. Saat ini perangkat lunak telah dilengkapi dengan *graphical user interface* (GUI) dengan perlengkapan ini komputer akan mengakses program menggunakan tulisan, grafik, dan gambar seperti ikon.

2.4.2 Definisi Teacher / Guru

Seorang guru adalah seseorang yang memberikan ilmu pengetahuan kepada anak didiknya dan bertanggung jawab untuk mendidik, mengajar, membimbing, mengarahkan, menilai dan mengevaluasi anak didiknya agar bermanfaat dimasa yang akan datang. Seorang guru harus mengetahui peran dan fungsinya yaitu:

- a. Guru Sebagai Pendidik
- b. Guru Sebagai Pengajar
- c. Guru Sebagai Pembimbing
- d. Guru Sebagai Pemimpin
- e. Guru Sebagai Penasehat
- f. Guru Sebagai Pendorong Kreatifitas

2.4.3 Definisi Android

Android adalah sistem operasi *mobile* berbasis linux yang merupakan sistem operasi terbuka. Sistem operasi android pada umumnya menggunakan 2 bahasa pemrograman yang resmi yaitu java dan kotlin. Pengembang memiliki akses penuh melalui API(*Application Programming Interface*) *Framework*. Android juga sebagai *platform mobile* pertama yang lengkap, terbuka dan gratis. keuntungan dari android ini ialah[3]:

1. Lengkap (*Complete Platform*): Adanya pendekatan yang komperhensif ketika sedang mengembangkan platform Android
2. Terbuka (*Open Source Platform*): Pengembang dapat dengan bebas mengembangkan aplikasinya.
3. Bebas/Gratis (*Free Platform*): Platform yang bebas untuk dikembangkan, tidak ada biaya lisensi ataupun biaya keanggotaan[3].

2.5 Tools Pendukung Pengembangan Perangkat Lunak

Dalam pembuatan aplikasi pada penelitian ini, penulis membahas beberapa *software* yang digunakan dalam perancangan pembagunan aplikasi android..

2.5.1 Definisi Android Studio

Perangkat yang bergerak pada sebuah sistem operasi untuk telepon seluler yang berbasis linux. Android merupakan sistem operasi untuk telepon seluler yang berbasis linux, android menyediakan *platform* yang terbuka untuk para pengembang atau *developer* untuk membuat aplikasi mereka sendiri agar dapat digunakan bermacam peranti bergerak. Android umum digunakan di *Smartphone* dan juga di tablet PC. Fungsinya sama seperti sistem operasi Symbian di Nokia, IOS di Apple dan BlackBerry OS[14].

Versi android yang digunakan untuk pembuatan aplikasi yaitu android versi 6.0 (Marshmallow) adalah versi pada generasi ke-6 setelah lolipop, versi ini diupgrade untuk meningkatkan cara kerja sistem keamanan android pada *smartphone* atau tablet. Versi ini dirilis pada tanggal 28 Mei 2015, yang dimana versi ini memiliki tingkat keamanan seperti *fingerpint*. Dan memiliki manajemen daya yang diberi nama *doze* dan membuat daya pada *smart phone* jadi irit[14], versi ini juga mendukung untuk library dari maps google yang dimana minimal android versi 6.0.

2.5.2 Definisi Website

Berikut definisi atau pengertian website menurut para ahli:

1. Menurut Yuhefizar[15], *Website* adalah suatu metode untuk menampilkan informasi di internet, baik berupa teks, gambar, suara maupun video yang interaktif dan mempunyai kelebihan untuk menghubungkan (link) satu dokumen dengan dokumen lainnya (*hypertext*) yang diakses melalui sebuah browser
2. Menurut Boone (Thomsom)[15], *Website* adalah koleksi sumber informasi kaya grafis yang saling berhubungan satu sama lain dalam internet yang lebih besar.
3. Menurut Haer Talib[15], *Website* adalah sebuah tempat di internet yang mempunyai nama dan alamat.

Dari pendapat para ahli penulis menyimpulkan, *website* adalah kumpulan informasi yang diakses dengan internet berupa nama dan alamat yang menghubungkan beberapa dokumen lain (*hypertext*) agar dapat diakses melalui browser yang terhubung secara online. *Page* ini hanya bisa diakses dengan menggunakan browser seperti Google Chrome, Opera, Firefox, Microsoft Edge, dan lain-lain.

2.5.3 Definisi Software Xampp

XAMPP adalah *software* web server apache yang di dalamnya sudah tersedia *database server* MySQL dan support PHP *Programming*. Jadi XAMPP merupakan paket instalasi Apache, PHP dan MySQL instant yang dapat digunakan untuk membantu proses instalasi ketiga produk tersebut. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia Apache *Web Server*, MySQL *Database Server*, PHP *Support* (PHP 4 dan PHP 5) dan beberapa modul lainnya[16].

2.5.4 Definisi JSON (*Java Script Object Notation*)

JavaScript Object Notation adalah format pertukaran data yang dibuat dari subset notasi objek literal dalam *JavaScript*. Sementara sintaks yang diterima oleh *JavaScript* untuk nilai literal sangat fleksibel, penting untuk dicatat bahwa JSON

memiliki aturan yang jauh lebih ketat. Menurut standar JSON, nama anggota objek harus berupa string JSON yang valid. String dalam JSON harus diapit dalam tanda kutip. Dalam *JavaScript*, elemen array dan nilai anggota objek dapat merujuk ke hampir semua ekspresi *JavaScript* yang *valid*, termasuk panggilan fungsi dan definisi[17]. JSON dibangun di atas dua struktur (JSON, 2018):

1. Kumpulan pasangan nama / nilai. Dalam berbagai bahasa, ini direalisasikan sebagai objek, catatan, struk, kamus, tabel hash, daftar kunci, atau array asosiatif.
2. Daftar nilai yang terurut. Di sebagian besar bahasa, ini direalisasikan sebagai array, vektor, daftar, atau urutan.

2.5.5 Definisi PHP (*Perl Hypertext Preprocessor*)

Menurut hidayatullah dan kawistara[18], PHP singkatan dari *Perl Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat open source. PHP sendiri adalah *script* untuk membuat halaman *web* menjadi dinamis, maksud dari dinamis berarti halaman yang akan ditampilkan dibuat menyesuaikan dengan permintaan *client*. Hal ini menyebabkan informasi yang diterima client selalu yang terbaru/*up to date* dan *script* PHP ini dijalankan/dieksekusi melalui *server*.

2.5.6 Definisi MySQL

MySQL merupakan salah satu basis data yang bersifat *free* (anda tidak perlu membayar) [18], atau aplikasi DBMS(*Database management system*) yang menjalankan fungsi pengolahan data untuk membangun sebuah aplikasi *web* yang dimana *server basisdata* dapat mengelola basis data dengan cepat dan menampung data dalam jumlah sangat besar serta dapat diakses oleh banyak *user*^[19].

2.5.7 Definisi JDK (*Java Development Kit*)

Sebuah produk dari *oracle corporation* yang ditujukan untuk pengembang bahasa *java*. JDK sebenarnya merupakan *software development kit* yang khusus

dikeluarkan oleh oracle. JDK menggunakan pemberian izin *Gnu Public License* (GPL) yang menjadikan *software* yang *free* dan *open source*[20].

2.5.8 Definisi *Black Box Testing*

Black box testing/pengujian kotak hitam adalah merupakan pengujian yang berfokus pada spesifikasi *funksional* dari perangkat lunak, *tester* dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi *funksional program*[21]. Proses *black box testing* dengan cara mencoba *program* yang telah dibuat dengan mencoba memasukkan data pada setiap *formnya*. Pengujian ini diperlukan untuk mengetahui *program* tersebut berjalan sesuai dengan yang dibutuhkan oleh pengguna.

2.5.9 Google Maps Api (*Application Programming Interface*)

Google *Maps* API adalah layanan gratis yang diberikan oleh Google dan sangat populer. Google *Maps* adalah suatu peta dunia yang dapat digunakan untuk melihat suatu daerah. Serta Google *Maps* API juga bisa dikatakan sebagai suatu *library* yang berbentuk *JavaScript*. Google *Maps* memanfaatkan teknologi digital imaging, seperti foto satelit sehingga anda bisa melihat bagaimana *landscape* planet bumi apabila dilihat dari luar angkasa[5].

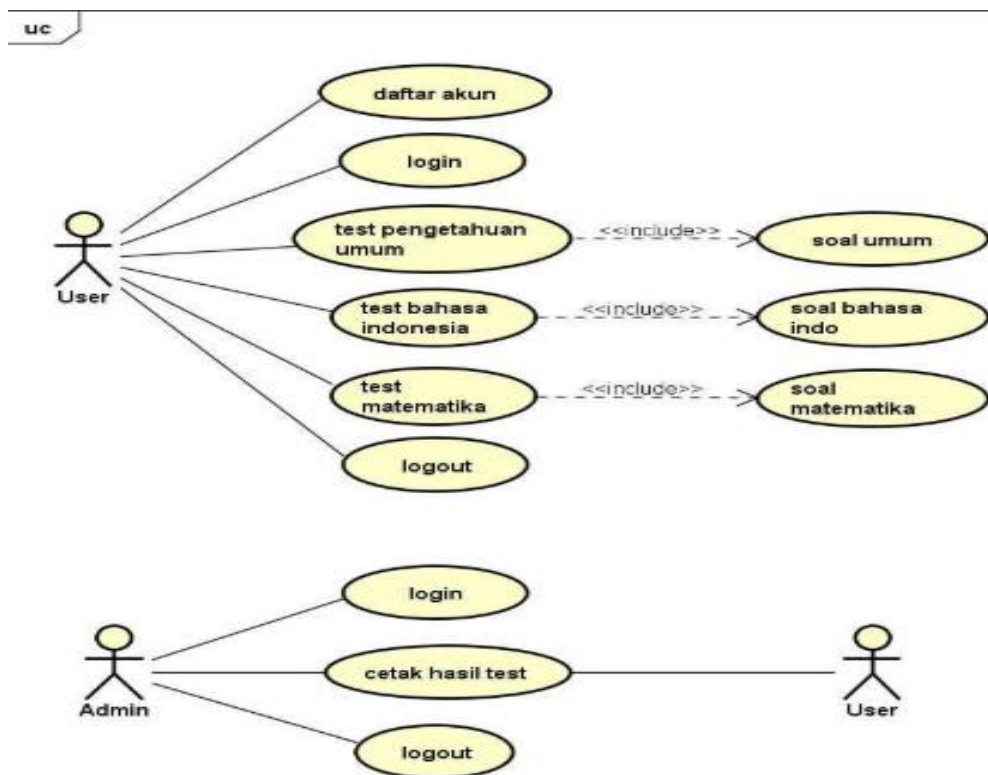
2.5.10 Definisi UML (*Unified Modeling Language*)

Perangkat lunak UML (*Unified Modeling Language*) adalah merupakan salah satu metode pemodelan *visual* yang digunakan dalam perancangan dan pembuatan sebuah *software* yang berorientasikan pada objek[22], UML sendiri menyediakan beberapa diagram *visual* yang menunjukkan berbagai aspek dalam sistem, ada beberapa diagram yang disediakan dalam UML, antara lain :

a. *Use Case Diagram*

Use case diagram adalah gambaran dari fungsionalitas yang diharapkan dari sebuah sistem, dan merepresentasikan sebuah interaksi antara aktor dan sistem. Didalam *use case* terdapat *actor* yang merupakan sebuah gambaran

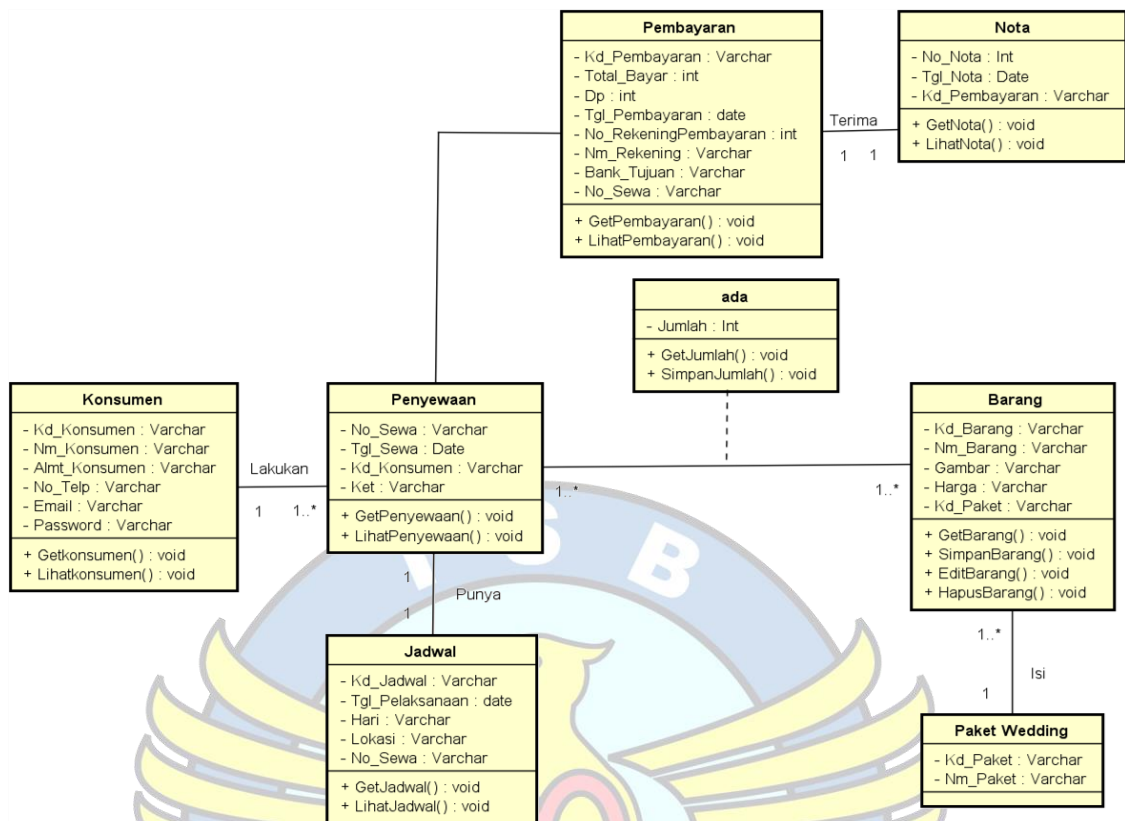
entitas dari manusia atau sebuah sistem yang melakukan pekerjaan di sistem[22]. *Use case* mempunyai tujuan umum pengguna. Ada empat aspek dalam diagram *use case*, antara lain adalah *actor*, *use case system/sub system*, *relationship* dan *boundary*.



Gambar 2. 2 Contoh *Use Case*[23].

b. Class Diagram

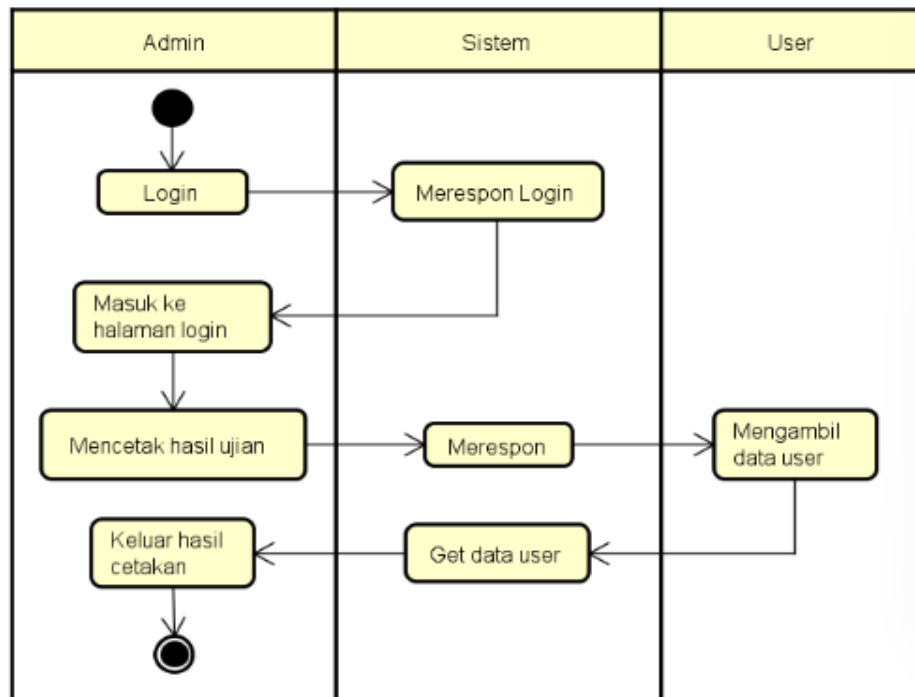
Merupakan gambaran struktur dan deskripsi dari *class*, *package*, dan objek yang saling berhubungan seperti diantaranya pewarisan, asosiasi dan lainnya[22]. *Class diagram* memiliki struktur dengan 3 area pokok (*Name*, *Attribute*, *Metoda*)[24].



Gambar 2. 3 Contoh *Class Diagram*[23].

c. Activity Diagram

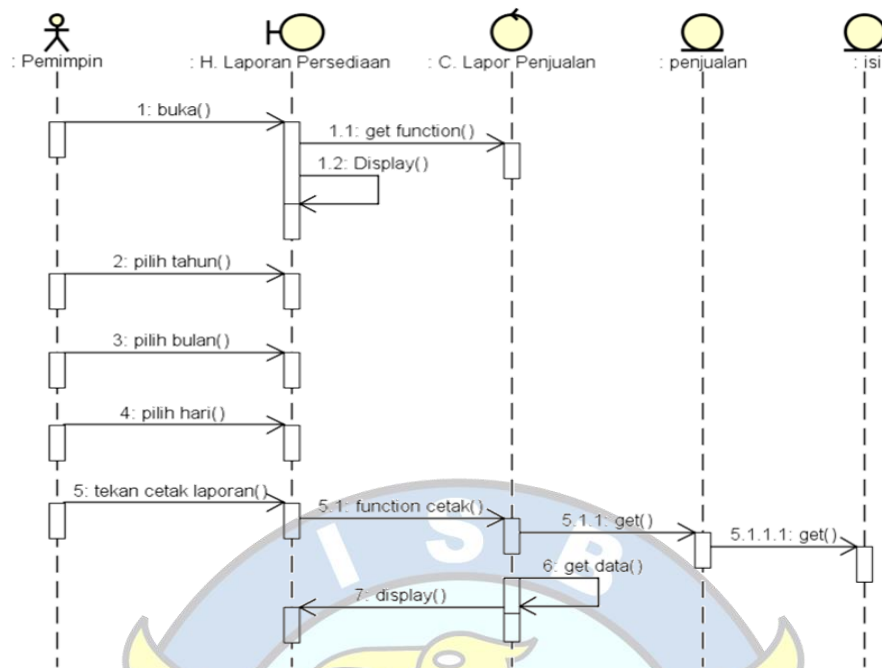
Merupakan teknik untuk menggambarkan aktifitas-aktifitas, objek, *state*, *transisi state* dan *event*[24]. Dengan kata lain kegiatan/gambaran alir dari aktifitas -aktifitas didalam sistem yang berjalan[22], seperti pengambilan keputusan dan sangat berguna untuk *operation* sebuah *object* dan proses bisnis.



Gambar 2. 4 Contoh *Activity Diagram*[23].

d. Sequence Diagram

Menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence diagram* adalah gambaran tahap demi tahap, termasuk kronologi (urutan)[24] atau interaksi antar objek didalam dan di sekitar sistem yang berupa *message* yang digambarkan terhadap waktu[22], yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram*.



Gambar 2. 5 Contoh Sequence Diagram[25].

2.6 Penelitian Terdahulu

Penelitian ini dilakukan dengan mengambil beberapa materi dari peneliti terdahulu :

Tabel 2. 1 Penelitian Terdahulu.

No	Penelitian (Tahun)	Judul	Masalah	Metode	Hasil
1	Lydia L., Adam S.W, Nico F. , Henny H. , Dwi H. B. (2019)	Yuk Les: Information Systems On Online Private Course Services Based On Mobile Application	Minimnya informasi yang tersebar mengenai penyedia jasa les private terutama di bidang non-akademik	System Development Life Cycle (SDLC)	Memudahkan siswa dalam mencari dan memesan jasa les private online
2	Marti Riastuti, Mei Raharja. (2021)	Penerapan Metode Haversine Formula pada Sistem Informasi Geografis	Sulitnya untuk menemukan lokasi kos-kosan secara detail di Kota Depok Jawa Barat bagi	Haversine Formula	Aplikasi web ini dapat memetakan lokasi tempat kos-kosan yang ada didaerah Kota dan Informasi juga dilengkapi dengan

No	Penelitian (Tahun)	Judul	Masalah	Metode	Hasil
		Tempat Kos Kota Depok Jawa Barat	yang baru menempuh pendidikan.		data-data secara detail
3	Ahmad Fauzi, Frengki Pernando, Mugi Raharjo. (2018)	Penerapan Metode Haversine Formula Pada Aplikasi Pencarian Lokasi Tempat Tambal Ban Kendaraan Bermotor Berbasis <i>Mobile Android</i>		<i>Haversain Formula</i>	Pada Penelitian ini menghasilkan aplikasi yang dapat melihat lokasi tambal ban yang sudah dimasukan dengan jarak yang paling pendek dari pengguna
4	Retno Palupi, Diyan Ayuk Yulianna, SM Santi Winarsih. (2021)	Analisa Perbandingan Rumus Haversine dan Rumus <i>Euclidean</i> Menggunakan Metode <i>Independent Sample t- Test</i>	Bagaimana mendapatkan probabilitas jarak yang paling signifikasi antara rumus haversine dan rumus euclidean	Rumus <i>Haversine</i> dan Rumus <i>Euclidean</i>	Hasil Penelitian ini menunjukkan rumus haversine dan rumus euclidean menunjukkan bahwa nilai probabilitas pada nilai signifikasi.
5	Sirait, Yulyanto, Sunaryo D.K., Jasmani. (2019)	Aplikasi Informasi Geografis Lokasi Fasilitas Sosial Berdasarkan Radius Berbasis <i>Android</i>	Bagaimana pembuatan aplikasi Sistem Informasi geografis untuk persebaran fasilitas sosial berdasarkan radius pada android.	<i>Haversine Formula</i>	Pada penelitian ini dibuktikan pemetaan geografis pada maps yang telah dibuat titik koordinat akan sesuai dengan jarak yang diinginkan pengguna menggunakan perhitungan haversain untuk mendapatkan lokasi fasilitas umum yang tersedia.