

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Sebelumnya

Penelitian sebelumnya dimaksudkan bahwa masalah yang hendak di teliti belum pernah di pecahkan oleh peneliti terlebih dahulu. Jika permasalahannya mirip, maka harus ditegaskan perbedaan penelitiannya dengan penelitian terlebih dahulu. Contoh nya peneliti dari *Farid dan Yulanda Yunus* di Tahun 2007 bahwa mereka telah meneliti perhitungan analisa pada rumah sakit dari hasil perhitungan tersebut maka jarak yang terdekat adalah rumah sakit umum islam ke rumah sakit planoma. Analisa Algoritma Haversine Formula Untuk Pencarian Lokasi Terdekat Rumah sakit dan Pukesmas Provinsi Di Gorontalo.

Sedangkan Penelitian di skripsi ini dimana penulis meneliti beberapa Taman Pendidikan Al-Qur'an yang ada di daerah Medan Denai dimana penulis menggunakan Metode Havarsine dan membangun sebuah aplikasi mencari rumah Al-Qur'an yang terdekat supaya mempermudah warga medan denai untuk mencari tahu informasi lebih mengenai Taman Pendidikan Al-Qur'an. Dimana aplikasi menggunakan GPS.

Dari jurnal di atas di jelaskan bahwa Metode Havarsine digunakan untuk mencari jarak terdekat dari user, sedang pada penelitian saya Metode Havarsine digunakan untuk mengetahui jarak terdekat dan di tambahkan dengan Fitur GPS.

II.2 Landasan Teori

II.2.1 Sistem Informasi (SI)

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 algoritmik, data, dan teknologi. Dalam pengertian ini, istilah ini digunakan untuk merujuk tidak hanya pada penggunaan organisasi teknologi informasi dan komunikasi (TIK), tetapi juga untuk cara di mana orang berinteraksi dengan teknologi ini dalam mendukung proses bisnis. Ada yang membuat perbedaan yang jelas antara sistem informasi, dan komputer sistem TIK, dan proses bisnis. Sistem informasi yang berbeda dari teknologi informasi dalam sistem informasi biasanya terlihat seperti memiliki komponen TIK. Hal ini terutama berkaitan dengan tujuan pemanfaatan teknologi informasi.

II.2.2 Sistem Informasi Geografis (SIG).

Sistem Informasi Geografis adalah informasi sistem komputerisasi yang memungkinkan penangkapan, pencontohan, pemanipulasian, penemuan kembali, penganalisan, dan presentasi data acuan geografis, sebagai fasilitas untuk menyiapkan, merepresentasikan, dan menginterpretasi fakta-fakta yang berkaitan dengan permukaan bumi. Nama lain dari sistem informasi geografis yaitu sistem informasi keruangan, sistem analisa data keruangan, dan sistem informasi sumber daya alam. Selain itu, sistem informasi geografis juga diartikan sebagai suatu sistem yang mengelola data geografis dengan

mengandalkan komputer. Pengolahan dan penyimpanan data pada sistem informasi geografi memanfaatkan komponen komputer yang terdiri dari perangkat keras dan perangkat lunak. **(Gintoro dkk,2010)**

II.2.3 Metode Haversine.

Metode Haversine merupakan sebuah cara yang digunakan untuk menghitung jarak antara titik permukaan bumi dengan menggunakan garis lintang (longitude) dan garis bujur (latitude) sebagai variable inputan (Chopde & Nichat, 2013), Haversine formula adalah sebuah persamaan penting pada navigasi yang dapat memberikan jarak berdasarkan bujur dan lintang (Chopde & Nichat, 2013). Dengan mengasumsikan bahwa bumi berbentuk sebuah lingkaran atau bulat sempurna dengan jari-jari $R = 6.367,45 \text{ km}$ dan lokasi dari 2 titik dikoordinat bola (lintang dan bujur) masing-masing dengan $lon1, lat1$ dan $lon2, lat2$. **(Chopde & Nichat, 2013).**

II.2.4 Global Positioning System (GPS).

Adalah sistem navigasi berbasis satelit yang terdiri dari setidaknya 24 satelit. GPS berfungsi dalam segala kondisi cuaca, di mana pun di dunia, 24 jam sehari, tanpa biaya berlangganan atau biaya penyiapan. Departemen Pertahanan AS (USDOD) awalnya menempatkan satelit ke orbit untuk penggunaan militer, tetapi mereka dibuat tersedia untuk digunakan sipil pada 1980-an.

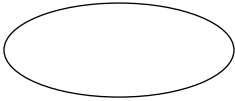
Cara kerja GPS Satelit GPS mengelilingi Bumi dua kali sehari dalam orbit yang tepat. Setiap satelit mengirimkan sinyal unik dan parameter orbital yang

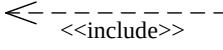
memungkinkan perangkat GPS untuk memecahkan kode dan menghitung lokasi tepat dari satelit. Penerima GPS menggunakan informasi dan trilaterasi ini untuk menghitung lokasi pasti pengguna. Pada dasarnya, penerima GPS mengukur jarak ke masing-masing satelit dengan jumlah waktu yang diperlukan untuk menerima sinyal yang dikirimkan. Dengan pengukuran jarak dari beberapa satelit lagi, penerima dapat menentukan posisi pengguna dan menampilkannya secara elektronik untuk mengukur rute lari Anda, memetakan lapangan golf, menemukan jalan pulang atau petualangan di mana saja. **(Gintoro dkk, 2010)**

II.2.5 *Use case diagram*

Dalam penyusunan suatu program diperlukan suatu model data yang berbentuk diagram yang dapat menjelaskan suatu alur proses sistem yang akan dibangun. Dalam penulisan laporan ini penulis menggunakan metode UML .

Tabel II.2. Simbol *Use Case Diagram*

Gambar	Nama	Keterangan
	<i>Use Case</i>	<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja.

	<i>Actor</i>	<i>Actor</i> adalah <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem.
	Asosiasi	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
	<i>Extends</i>	<i>Extends</i> merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.
	<i>Include</i>	<i>Include</i> merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.

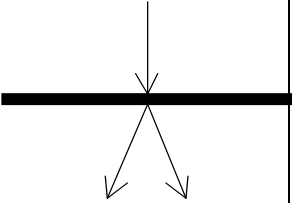
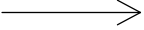
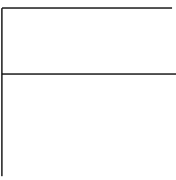
(Sumber: Ade Hendini; 2016)

II.2.6 Activity Diagram.

Activity diagram merupakan diagram yang menggambarkan aktivitas dari sebuah sistem atau proses bisnis. *Activity Diagram* menunjukkan bagaimana aktivitas-aktivitas tersebut bergantung satu sama lain, yang menggambarkan

proses-proses dan jalur-jalur aktivitas yang berjalan dan dilakukan oleh sistem.

Tabel II.3. Simbol Activity Diagram

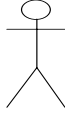
Gambar	Nama	Keterangan
	<i>Initial State</i> <i>/ Start State</i>	<i>Start Point</i> diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
	<i>Action State</i>	menggambarkan suatu proses atau kegiatan bisnis.
	<i>Transition</i>	digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Control Flow</i>	digunakan untuk menunjukkan kegiatan.
	<i>Decision</i>	Menggambarkan pilihan untuk pengambilan keputusan <i>true</i> atau <i>false</i> .
	<i>Final State</i>	Digunakan untuk mengakhiri aktivitas.
	<i>Swimlane</i>	Merupakan pembagian <i>Activity Diagram</i> untuk menunjukkan siapa melakukan apa.

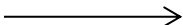
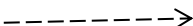
(Sumber: Ade Hendini; 2016)

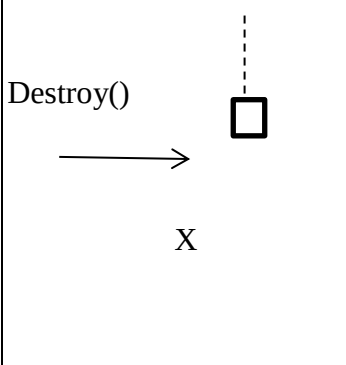
II.2.7 Sequence Diagram

Sequence Diagram merupakan suatu gambaran perilaku, diagram ini menunjukkan sejumlah contoh objek dan *message* (pesan) yang diletakkan diantara objek-objek yang ada di dalam *use case*.

Tabel II.4. Simbol Sequence Diagram

Gambar	Nama	Keterangan
	<i>aktor</i>	• Orang, proses, atau system lain yang berinteraksi dengan system informasi dan mendapatkan manfaat dari system. • Berpartisipasi secara berurutan dengan mengirimkan dan/atau menerima pesan • Ditempatkan di bagian diatas diagram.
	<i>objek</i>	• Berpartisipasin secara berurutan dengan mengirimkan dan / atau menerima pesan • Ditempatkan di bagian

		dia tas diagram
Objek sedang aktif berinteraksi 	<i>Objek sedang aktif berinteraksi</i>	• Menandakan kehidupan obyek selama urutan • Diakhiri tanda X pada titik dimana kelas tidak lagi berinteraksi
pesan() 	<i>pesan</i>	Objek mengirim satu pesan ke objek lainnya
<<create>> 	<i>create</i>	Menyatakan suatu objek membuat objek yang lain. Arah panah mengarah pada objek lainnya
1:masukkan 	<i>1:masuk</i>	Menyatakan bahwa suatu objek membuat objek yang lainnya, arah panah mengarah pada objek lainnya
1:keluaran 	<i>1:keluaran</i>	Objek/metode menghasilkan suatu kembalian masukkan ke objek lainnya arah panah mengarah pada objek yang menerima kembalian

	trov	nyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy
---	-------------	--

(Sumber : Ade Hendini ; 2016)

II.2.8 Class Diagram

Class Diagram merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan sistem. *Class Diagram* juga menunjuk kan atribut yang berhubungan dengan objek yang dikoneksi kan (Ade Hendini, 2016).

Class Diagram secara khas meliputi : Kelas (*Class*), *Relasi Assosiations*, *Generalitation* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operation / Method*) dan *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality* (Ade Hendini, 2016).

Tabel II.5. Multiplicity Class Diagram

<i>Multiplicity</i>	<i>Penjelasan</i>
1	Satu dan hanya satu.
0..*	Boleh tidak ada atau 1 atau lebih.
1..*	1 atau lebih.
0..1	Boleh tidak ada, maksimal 1.
n..n	Batasan antara. Contoh 2..4 mempunyai

	arti minimal 2 maksimal 4.
--	----------------------------

(Sumber: Ade Hendini ; 2016)