

**PENCARIAN USTADZ UNTUK WILAYAH KOTA
MAKASSAR MENGGUNAKAN ALGORITMA HAVERSINE
FORMULA BERBASIS WEBSITE**

TUGAS AKHIR

**Karya tulis sebagai salah satu syarat untuk
memperoleh gelar Sarjana dari
Universitas Fajar**

Oleh :

**RIZAL MANDALA PUTRA
1520221024**



**PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS FAJAR
MAKASSAR
2022**

HALAMAN PENGESAHAN

Pencarian Ustadz Untuk Wilayah Kota Makassar Menggunakan Algoritma
Haversite Formula Berbasis Website

Disusun Oleh :

Rizal Mandala Putra
1520221024

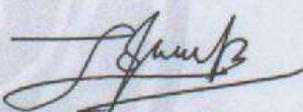
Telah diperiksa dan disetujui oleh Dosen Pembimbing

Makassar, 03 April 2023

Pembimbing I

Pembimbing II


Indah Purwitasari Ihsan, S.T., M.T
NIDN: 0922118603


Zagita Maria Putra, S.T., M.T
NIDN: 0922118603

Mengetahui,


Dekan Fakultas Teknik

UNIVERSITAS FAJAR
DEKAN FAKULTAS
Prof. Dr. Ir. Erniati, S.T., M.
NIDN: 0906107701

Ketua Program Studi


Safaruddin, S.Si., M.T
NIDN: 0909106901

PERNYATAAN ORISINAL

Penulis dengan ini menyatakan bahwa tugas akhir :

“Pencarian Ustadz Untuk Kota Makassar Menggunakan Algoritma Haversine Formula Berbasis Website” adalah karya orisinal saya dan setiap serta seluruh sumber acuan telah ditulis sesuai dengan panduan penulisan ilmiah yang berlaku di fakultas teknik Universitas Fajar.

Makassar, 03 April 2023

Yang :



Rizal Mandala Putra

KATA PENGANTAR

Syukur Alhamdulillah saya panjatkan kepada ALLAH SWT karena hanya dengan rahmat dan hidayah-Nya sehingga saya dapat menyelesaikan laporan inik yang berjudul **“Pencarian Ustadz Untuk Kota Makassar Menggunakan Algoritma Haversine Formula Berbasis Website”**. Laporan skripsi ini disusun sebagai salah satu syarat bagi mahasiswa Program Studi Teknik Elektro untuk dapat menyelesaikan studi Program Strata Satu (S1) pada Universitas Fajar Makassar. Pada saat penyusunan tugas akhir ini sangat banyak mendapat bantuan dari berbagai pihak, oleh karna itu dalam kesempatan ini pula ijinan saya untuk mengucapkan terima kasih serta penghargaan setinggi-tingginya kepada:

1. Allah SWT atas berkat dan rahmatnya yang tidak terputus kepada kami.
2. Ayah dan ibu serta keluarga besar yang senantiasa mendoakan agar kami selalu diberi kekuatan dan kesabaran.
3. Ibu Prof. Dr. Ir. Erniati, S.T., M.T. selaku Dekan Fakultas Teknik Universitas Fajar.
4. Ibu Asmawaty Azis, S.T., M.T. selaku Ketua Program Studi Teknik Elektro Universitas Fajar.
5. Ibu Zaryanti Zainuddin, S.T., M.T., selaku Penasehat Akademik yang telah membimbing dalam segala proses.
6. Ibu Indah Purwitasari Ihsan, S.T., M.T. selaku Dosen Pembimbing I.
7. Bapak Zagita Marna Putra, S.T., M.T. selaku Dosen Pembimbing II.
8. Dosen-dosen Prodi Teknik Elektro Universitas Fajar.
9. HME FT-UNIFA yang telah memberikan kesempatan untuk berlembaga dan mengembangkan diri.
10. Teman-teman Program Studi Teknik Elektro 2015. Terima kasih untuk segalanya selama ini atas dukungan dan nasehat kalian selama penulisan tugas akhir ini.
11. Rekan-rekan Mahasiswa Universitas Fajar, yang selalu memberikan bantuan dan masukan, serta semua pihak yang telah membantu sehingga laporan Skripsi ini dapat terselesaikan.

Akhir kata, penulis menyadari bahwa penelitian ini jauh dari kesempurnaan. Oleh sebab itu, kritik maupun saran selalu penulis harapkan demi tercapainya hal terbaik dari tugas akhir ini. Besar harapan penulis, semoga tugas akhir ini dapat memberikan manfaat sekaligus menambah pengetahuan bagi berbagai pihak. Amin.

Makassar, Agustus 2022

Penulis

Abstrak

Rizal Mandala Putra, Teknik Elektro, Pencarian Ustadz Untuk Wilayah Kota Makassar Menggunakan Algoritma Haversine Formula Berbasis Website. Di Kota Makassar sangat banyak ustad tetapi disaat bulan suci ramadhan beberapa ustad kesulitan untuk mengatur schedule dan masyarakat yang sedang berduka sering mendapatkan kesulitan untuk mencari ustad agar acara tausiyah yang bisa berjalan sesuai yang diharapkan. Menerapkan website pencarian lokasi rumah ustadz dengan memanfaatkan *Global Positioning System* (GPS) dan Google Map API dan Menerapkan metode *haversine formula* dalam menentukan jarak terdekat lokasi ustadz dengan masyarakat. *Haversine formula* merupakan sebuah persamaan yang penting dalam *navigasi*, dimana *formula* ini memberikan jarak di antara dua titik pada lingkaran bola dari setiap garis bujur (*longitude*) dan garis lintang (*latitude*). Ini adalah kasus khusus dari sebuah *formula* yang lebih umum dalam *trigometri* lingkaran bola, *law of haversine*, *haversine formula* menghubungkan sisi dan sudut dari sebuah segitiga bola. Untuk mengetahui bahwa rumus *Haversine* dapat menghitung jarak antara dua titik dipermukaan bumi maka perlu di lakukan uji coba berikut ini merupakan penjabaran rumus *Haversine* di dalam *query* MySQL : $R = 6371$ (km), $\Delta lat = latitude$ tujuan, $\Delta long = longitude$ tujuan, $\Delta lat user = latitude user$, $\Delta long user = longitude user$, 1 derajat = 0.0174532925 radian, Dari penjabaran diatas maka didapatkan rumus *query* MySQL di bawah ini : $(R * ACOS (SIN (RADIANS (\Delta lat)) * SIN (RADIANS (\Delta lat user)) + COS (RADIANS (\Delta long - \Delta long user)) * COS (RADIANS (\Delta lat)) * COS (RADIANS (\Delta lat user))))$

Kata Kunci : Ustad, GPS, Google Map API, *Haversine Formula*

Abstract

Rizal Mandala Putra, Electrical Engineering, Searching for Ustadz for Makassar City Area Using Website-Based Formula Haversine Algorithm. In Makassar, there are very many clerics, but during the holy month of Ramadan, some clerics find it difficult to arrange schedules and people who are grieving often find it difficult to find clerics so that the tausyah event can run as expected. Implementing a website to find the location of the ustadz's house by utilizing the Global Positioning System (GPS) and the Google Map API and applying the haversine formula method in determining the closest distance between the ustadz's location and the community. The haversine formula is an important equation in navigation, where this formula gives the distance between two points on a spherical circle of each longitude and latitude. This is a special case of a more general formula in spherical trigonometry, the law of haversine, the haversine formula connecting the sides and angles of a spherical triangle. To find out that the Haversine formula can calculate the distance between two points on the earth's surface, it is necessary to carry out the following experiments, which are the translation of the Haversine formula in a MySQL query: $R = 6371$ (km), $\Delta lat =$ destination latitude, $\Delta long =$ destination longitude, $\Delta lat user =$ latitude user, $\Delta long user =$ longitude user, $1 \text{ degree} = 0.0174532925 \text{ radians}$. From the above description, the following MySQL query formula is obtained: $(R * \text{ACOS}(\text{SIN}(\text{RADIANS}(\text{lat}))) * \text{SIN}(\text{RADIANS}(\text{lat user})) + \text{COS}(\text{RADIANS}(\text{long} - \text{long user})) * \text{COS}(\text{RADIANS}(\text{lat})) * \text{COS}(\text{RADIANS}(\text{lat user})))$

Keywords: Searching, Ustad, GPS, Google Map API, Haversine Formula

DAFTAR ISI

PERSETUJUAN HASIL	i
KATA PENGANTAR.....	ii
ABSTRAK	iv
DAFTAR ISI.....	vi
DAFTAR TABEL	ix
DAFTAR GAMBAR.....	x
BAB I PENDAHULUAN	
I.1 Latar Belakang	1
I.2 Rumusan Masalah	2
I.3 Tujuan Penelitian	2
I.4 Batasan Masalah.....	3
BAB II TINJAUAN PUSTAKA	
II.1 Tinjauan Teori	4
II.1.1 Ustad	4
II.1.2 Definisi Sistem.....	5
II.1.3 Definisi Mesin Pencarian (<i>Search Engine</i>).....	5
II.1.4 Haversine Formula.....	6
II.1.5 GPS (<i>Global Positioning System</i>)	8
II.1.6 Google Maps API.....	8
II.1.7 API (<i>Application Programming Interface</i>)	9
II.1.8 Website	11
II.1.9 Leaflet JavaScript (LeafletJS).....	11
II.1.10 Visual Studio Code (VSCode).....	11
II.1.11 XAMPP.....	12
II.1.12 Google Chrome	12
II.1.13 Bahasa Program	13
II.1.13.1 CSS	13
II.1.13.2 Java Script	14
II.1.13.3 <i>Hyper Text Markup Language</i> (HTML)	14
II.1.13.4 PHP	15

II.1.14	Basis Data (<i>Database</i>)	15
II.1.15	<i>Database Management System</i> (DBMS)	17
II.1.16	SQLite	18
II.1.17	UML (<i>Unified Modelling Language</i>)	19
II.1.17.1	<i>Use Case Diagram</i>	20
II.1.17.2	<i>Activity Diagram</i>	22
II.1.17.3	<i>Sequence Diagram</i>	25
II.1.17.4	<i>Class Diagram</i>	27
II.1.18	<i>Flowchart</i>	29
II.1.19	Pengujian Sistem	31
II.1.19.1	<i>Black Box Testing</i>	31
II.1.19.2	<i>White Box Testing</i>	35
II.2	Penelitian Terdahulu (<i>State of The Art</i>)	38
II.3	Kerangka Pikir	41

BAB III METODOLOGI PENELITIAN

III.1	Prosedur Penelitian	42
III.2	Rancangan Sistem	43
III.2.1	Sistem Berjalan	43
III.2.2	Sistem Direncanakan	44
III.2.2.1	<i>Use Case Diagram</i>	45
III.2.2.2	<i>Activity Diagram</i>	46
III.2.2.3	<i>Sequence Diagram</i>	48
III.2.2.4	<i>Class Diagram</i>	49
III.2.3	Perancangan Antarmuka	51
III.3	Waktu dan Tempat Penelitian	52
III.3.1	Waktu Penelitian	52
III.3.2	Tempat Penelitian	52
III.4	Alat dan Bahan Penelitian	52
III.5	Metode Pengumpulan Data	53
III.6	Tahap Analisis Data / Pengujian Sistem	53

BAB IV HASIL DAN PEMBAHASAN

IV.1 Hasil	54
IV.1.1 Tampilan <i>Login dan Register</i>	54
IV.1.2 Tampilan <i>Homepage</i>	56
IV.1.3 Tampilan <i>Menu Ustad Kami</i>	59
IV.1.4 Tampilan <i>Menu Kontak</i>	61
IV.1.5 Tampilan <i>Database</i>	64
IV.2 Pembahasan	65
IV.2.1 Penerapan Algoritma <i>Haversine</i>	65
IV.2.2 <i>Blackbox Testing</i>	72
IV.2.3 <i>Whitebox Testing</i>	73
IV.2.4 Pengukuran Jarak dan Selisih	78

BAB V PENUTUP

V.1 Kesimpulan	81
V.2 Saran	81

DAFTAR PUSTAKA

DAFTAR TABEL

Tabel 2.1 Simbol-simbol <i>Use Case Diagram</i>	21
Tabel 2.2 Simbol-simbol <i>Activity Diagram</i>	23
Tabel 2.3 Simbol-simbol <i>Sequence Diagram</i>	26
Tabel 2.4 Simbol-simbol <i>Class Diagram</i>	28
Tabel 2.5 Simbol-simbol dalam <i>Flowchart</i>	29
Tabel 2.6 Pengujian <i>Blackbox</i>	33
Tabel 2.7 Hasil Pengujian <i>Blackbox</i>	34
Tabel 2.8 Penelitian Terdahulu (<i>State of The Art</i>).....	38
Tabel 3.1 <i>Hardware</i> dan <i>Software</i>	52
Tabel 4.1 Perhitungan Jarak	72
Tabel 4.2 Instrumen Pengujian Fungsional Website	72
Tabel 4.3 <i>Test Case</i> Cari Ustad	75
Tabel 4.4 <i>Test Case</i> Form Pendaftaran.....	77
Tabel 4.5 Perhitungan Selisih dan <i>Percent Error</i>	80

DAFTAR GAMBAR

Gambar 2.1	Hasil Pengukuran di <i>Google Earth</i>	7
Gambar 2.2	Prinsip Kerja GPS	8
Gambar 2.3	<i>API Illustration</i>	10
Gambar 2.4	<i>Use Case Diagram</i> Bank ATM	22
Gambar 2.5	Contoh <i>Activity Diagram</i>	25
Gambar 2.6	Contoh <i>Sequence Diagram</i>	27
Gambar 2.7	Contoh <i>Class Diagram</i>	29
Gambar 2.8	Sistem Pengujian	32
Gambar 2.9	Pengujian Sistem	35
Gambar 2.10	Pengujian <i>Flowchart</i> Menampilkan Peta	36
Gambar 2.11	<i>Flowchart</i> Pengujian Kedua	37
Gambar 3.1	Prosedur Penelitian.....	42
Gambar 3.2	Sistem yang Berjalan.....	43
Gambar 3.3	<i>Flowchart</i> Penyelesaian Masalah	44
Gambar 3.4	<i>Use Case Diagram</i> Website.....	45
Gambar 3.5	<i>Activity Login</i>	46
Gambar 3.6	<i>Activity</i> Daftar Akun.....	46
Gambar 3.7	<i>Activity</i> Form Pendaftaran Akun	46
Gambar 3.8	<i>Activity</i> Validasi Profil	47
Gambar 3.9	<i>Activity</i> Form Usulan Ustad	47
Gambar 3.10	<i>Activity</i> Mencari Ustad yang Terdekat	47
Gambar 3.11	<i>Sequence Login</i>	48
Gambar 3.12	<i>Sequence</i> Daftar Akun.....	48
Gambar 3.13	<i>Sequence</i> Validasi Akun.....	48
Gambar 3.14	<i>Sequence</i> Validasi Profil	49
Gambar 3.15	<i>Sequence</i> Mencari Ustad yang Terdekat	49
Gambar 3.16	<i>Class Diagram</i>	49
Gambar 3.17	Tampilan Ustad	51
Gambar 3.18	Tampilan Tanya Ustad	51
Gambar 3.19	Tampilan Daftar Akun.....	51

Gambar 3.20	Tampilan Google Maps	52
Gambar 4.1	Tampilan <i>Login</i>	54
Gambar 4.2	<i>Script</i> Tampilan <i>Login</i>	55
Gambar 4.3	Tampilan <i>Register</i>	55
Gambar 4.4	<i>Script</i> Tampilan <i>Register</i>	56
Gambar 4.5	Tampilan <i>Homepage</i>	56
Gambar 4.6	<i>Script Homepage</i>	57
Gambar 4.7	<i>Menu</i> Cari Ustad.....	57
Gambar 4.8	<i>Script Menu</i> Cari Ustad	57
Gambar 4.9	Tampilan Google Maps	58
Gambar 4.10	<i>Script</i> Tampilan Google Maps	58
Gambar 4.11	<i>Menu</i> Tanya Ustad.....	59
Gambar 4.12	<i>Script Menu</i> Tanya Ustad	59
Gambar 4.13	<i>Menu</i> Ustad Kami.....	59
Gambar 4.14	<i>Script Menu</i> Ustad Kami	60
Gambar 4.15	Tampilan Profil Ustad	60
Gambar 4.16	<i>Script</i> Tampilan Profil Ustad.....	60
Gambar 4.17	Tampilan Kontak Ustad.....	61
Gambar 4.18	<i>Scripti</i> Kontak Ustad	61
Gambar 4.19	Tampilan Halaman Kontak	62
Gambar 4.20	<i>Script</i> Tampilan Halaman Kontak.....	62
Gambar 4.21	Tampilan Form Pendaftaran	62
Gambar 4.22	<i>Script</i> Tampilan Form Pendaftaran	63
Gambar 4.23	Tampilan Form Usulan Ustad	63
Gambar 4.24	<i>Script</i> Tampilan Form Usulan Ustad.....	64
Gambar 4.25	<i>Script</i> Koneksi	64
Gambar 4.26	Tampilan Database Form Pendaftaran	64
Gambar 4.27	<i>Script</i> Database Form Pendaftaran	64
Gambar 4.28	Tampilan Database Usulan Ustad	65
Gambar 4.29	<i>Script</i> Database Form Usulan Ustad	65
Gambar 4.30	Diagram <i>Flowchart White Box Testing</i> Cari Ustad.....	74

Gambar 4.31	Diagram <i>Flowgraph White Box Testing</i> Cari Ustad.....	74
Gambar 4.32	Diagram <i>Flowchart</i> Form Pendaftaran Ustad	76
Gambar 4.33	Diagram <i>Flowgraph</i> Form Pendaftaran Ustad	76

BAB I

PENDAHULUAN

I.1. Latar Belakang

Seiring dengan kemajuan teknologi, tingkat kebutuhan manusia akan teknologi dalam penggunaan sehari-hari semakin tinggi, salah satu contoh perkembangan teknologi yaitu sistem informasi yang terkomputerisasi. Sistem informasi mempunyai peran yang sangat penting karena berkaitan dengan pengolahan data yang digunakan diberbagai perusahaan dan instansi untuk mempermudah pekerjaan. Saat ini telepon genggam (*handphone*) yang sedianya sebagai alat komunikasi sudah memiliki kelebihan yang cukup banyak dari fungsi dasarnya. Hal ini bisa terjadi tak lepas karena adanya sistem operasi pada telepon genggam yang mendukung untuk menjadi telepon pintar (*smartphone*). Salah satu sistem operasi yang digunakan pada telepon pintar sekarang ini adalah Android. Sudah banyak website menggunakan android, sehingga lebih mudah digunakan oleh siapa saja, kapan saja dan dimana saja

Dakwah yang dipersepsikan oleh masyarakat pada umumnya adalah kegiatan penyampaian pesan verbal yang berisi nasehat-nasehat agama beserta dalil-dalil yang di jadikan rujukan. Tapi, Terkadang ada beberapa daerah yang hanya memprioritaskan penceramah tertentu untuk berceramah di tempatnya. Di Kota Makassar terdapat ratusan masjid dan lokasi ceramah yang masih minim informasi alamatnya sehingga Ustadz atau penceramah seringkali kesulitan menemukan lokasi yang akan ditempati ceramah. Kesulitan ini menyebabkan Ustadz atau penceramah kadang tidak melaksanakan tugasnya

di lokasi yang telah dijadwalkan. Alamat atau lokasi menjadi informasi Ustadz dan penceramah dapat menemukan lokasi yang ditempati ceramah.

Berdasarkan permasalahan diatas sehingga diperlukannya membuat sebuah website yang Berjudul “**Website Pencarian Ustadz Untuk Wilayah Makassar Menggunakan Algoritma Haversine Formula Berbasis Mobile**” agar dapat mempermudah masyarakat mencari ustadz untuk menghadiri tausyiah, syukuran, ceramah, dan bahkan menjadi guru pengajian.

Dengan adanya website ini diharapkan dapat membantu mencari ustadz karena terkadang ustadz memiliki kepentingan yang lain ataupun harus menghadiri kegiatan di tempat lain. Masyarakat juga dengan mudah mendapatkan informasi tentang ustadz favorit dan bisa *chatting* secara personal.

I.2. Rumusan Masalah

Sesuai dengan latar belakang masalah yang telah di kemukakan, makadapat di uraikan rumusan masalah, sebagai berikut :

1. Bagaimana menerapkan website pencarian lokasi rumah ustadz denganmemanfaatkan *Global Positioning System (GPS)* dan *Google Map API*?
2. Bagaimana menerapkan metode *haversine formula* dalam menentukanjarak terdekat lokasi ustadz dengan masyarakat?

I.3. Tujuan Penelitian

Sesuai dengan rumusan masalah yang telah di kemukakan, maka dapat diuraikan tujuan penelitian, sebagai berikut :

1. Menerapkan website pencarian lokasi rumah ustadz dengan memanfaatkan *Global Positioning System (GPS)* dan *Google Map API*
2. Menerapkan metode *haversine formula* dalam menentukan jarak terdekat lokasi ustadz dengan masyarakat

I.4. Batasan Masalah

Berdasarkan perumusan masalah yang telah disebutkan diatas, maka batasan masalah pada website ini adalah sebagai berikut :

1. Website yang akan dibuat yaitu berbasis *Mobile*
2. Website ini hanya di Kota Makassar

BAB II

TINJAUAN PUSTAKA

II.1 Tinjauan Teori

II.1.1 Ustadz

Menurut Ngainun Naim (Yogyakarta, 2016 : 4) Guru / ustadz / ustadzah adalah sosok yang rela mencurahkan sebagian besar waktunya untuk mengajar dan mendidik siswa. Guru / ustadz /ustadzah adalah sumber belajar yang utama, karena tanpa adanya guru maka proses pembelajaran tidak akan bisa berlangsung secara maksimal. Seseorang akan mungkin dapat belajar sendiri, namun tanpa adanya bimbingan dari guru maka hasilnya tidak akan bisa maksimal. Dengan begitu, untuk menjadi guru / ustadz / ustadzah seharusnya mempunyai banyak ilmu, mau mengamalkan dengan sungguh-sungguh ilmunya dalam proses pembelajaran, toleran, dan senantiasa berusaha menjadikan siswanya memiliki kehidupan yang lebih baik. Secara prinsip, orang yang disebut sebagai guru / ustadz /ustadzah bukan hanya orang yang memiliki kuallifikasi keguruan secara formal yang diperoleh melalui jenjang pendidikan di perguruan saja, namun jika ada orang yang mempunyai kompetensi keilmuan tertentu dan dapat menjadikan orang lain pandai dalam halkognitif, afektif dan psikomotorik maka mereka juga bisa disebut sebagai guru / ustadz / ustadzah.

II.1.2 Definisi Sistem

Menurut Abdul Kadir (2014:61) bahwa “Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan”.

Menurut Sutabri (2012:3) bahwa “Sistem adalah suatu kumpulan atau himpunan dari suatu unsur, komponen, atau variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu”.

Menurut Sutarman (2012:13) bahwa “Sistem adalah kumpulan elemen yang saling berhubungan dan berinteraksi dalam satu kesatuan untuk menjalankan suatu proses pencapaian suatu tujuan utama”.

Menurut Fathansyah (2015:11) bahwa “Sistem adalah sebuah tatanan (keterpaduan) yang terdiri atas sejumlah komponen fungsional (dengan satuan fungsi dan tugas khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses tertentu”.

Dari beberapa pengertian di atas penulis dapat menyimpulkan bahwa sistem merupakan sekumpulan elemen, himpunan dari suatu unsur, komponen fungsional yang saling berhubungan dan berinteraksi satu sama lain untuk mencapai tujuan yang diharapkan

II.1.3 Definisi Mesin Pencarian (*Search Engine*)

Menurut Rhoni Rhodin (2017:65) Mesin pencari (*search engine*) adalah salah satu program komputer yang dirancang khusus untuk membantu seseorang menemukan file-file yang disimpan dalam

komputer, misalnya dalam sebuah web server umum (www) atau komputer sendiri.

II.1.4 Haversine Formula

Haversine formula merupakan sebuah persamaan yang penting dalam navigasi, dimana *formula* ini memberikan jarak di antara dua titik pada lingkaran bola dari setiap garis bujur (*longitude*) dan garis lintang (*latitude*). Ini adalah kasus khusus dari sebuah *formula* yang lebih umum dalam trigometri lingkaran bola, *law of haversine*, *haversine formula* menghubungkan sisi dan sudut dari sebuah segitiga bola.

Haversine formula nantinya akan digunakan dalam perhitungan jarak antara dua titik. Dalam hal ini adalah titik posisi *user* dan titik posisi tujuan, sehingga dapat menjadi kunci utama dalam perbandingan jarak pada penentuan jarak terdekat. Adapun rumus dan contoh kasus *haversine formula* sebagai berikut :

Nilai koordinat bumi yang digunakan disini dalam bentuk *Decimal Deggre* seperti berikut :

Titik 1

Latitude : -7.275515

Longitude: 112.794895

Titik 2

Latitude : -7.276207

Longitude : 112.790843

Kenapa menggunakan bentuk *Decimal Degree*? karena untuk memudahkan dalam pengecekannya. Disini pengecekannya menggunakan website dari google yaitu Google Earth. Pengujian

mencari jarak dengan menggunakan metode Haversine. Rumus untuk metode Haversine seperti berikut :

$$\Delta\sigma = 2 \arcsin \left(\sqrt{\sin^2 \left(\frac{\Delta\phi}{2} \right) + \cos\phi_s \cos\phi_f \sin^2 \left(\frac{\Delta\lambda}{2} \right)} \right)$$

$\Delta\sigma$ Interior

$\Delta\phi$ Spherical Angel

ϕ_s Latitude1 –

ϕ_f Latitude2

$\Delta\lambda$ Longitude1 – Longitude2

Latitude2

Longitude1 – Longitude2

Penulisan program untuk pengukuran jarak dengan metode haversine seperti berikut :

$$=(6371.1*((2*ASIN(SQRT((SIN((RADIANS(B7)-RADIANS(B6))/2)^2)+COS(RADIANS(B7))*COS(RADIANS(B6))*(SIN((RADIANS(C7)-RADIANS(C6))/2)^2))))))$$

Brikut ini adalah contoh gambar pengujian dari hasil pengukuran di *google earth* :

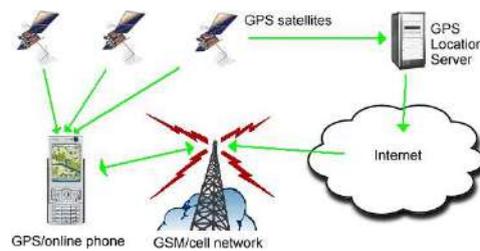


Gambar 2.1 Hasil Pengukuran di *Google Earth*

(Sumber : <https://bsierad.com/cara-mengukur-jarak-di-google-map-menggunakan-metode-euclidean-dan-haversine/>)

II.1.5 GPS (*Global Positioning System*)

GPS (*Global Positioning System*) adalah sistem satelit navigasi penentuan yang dimiliki dan dikelola oleh Amerika Serikat. Sistem ini didesain untuk memberikan posisi dan kecepatan tiga dimensi serta informasi mengenai waktu, secara kontinyu diseluruh dunia tanpa bergantung waktu dan cuaca, kepada banyak orang secara simultan. Pada saat ini, system GPS (*Global Positioning System*) sudah banyak yang digunakan orang di seluruh dunia. Di Indonesia pun, GPS (*Global Positioning System*) sudah banyak diwebsitekan terutama yang terkait dengan website-website yang menuntut informasi tentang posisi.



Gambar 2.2 Prinsip Kerja GPS

(Sumber : <http://idorastafara.blogspot.com/2012/10/gps.html>)

Dibandingkan dengan system dan metode metode penentuanposisi lainnya. GPS (*Global Positioning System*) mempunyai banyak kelebihan dan menawarkan lebih banyak keuntungan, baik dalam segi operasionalisasinya maupun kualitas posisi yang diberikan.

II.1.6 Google Maps API

Menurut Elian (2012) *Google Maps* API merupakan pengembangan teknologi dari google yang digunakan untuk menanamkan *Google Maps*

di suatu website yang tidak dibuat oleh *Google*. *Google Maps* API adalah suatu library yang berbentuk javascript yang berguna untuk memodifikasi peta yang ada di *GoogleMaps* sesuai kebutuhan.

Dalam perkembangan *Google Maps* API diberikan kemampuan untuk mengambil gambar peta statis. Melakukan geocoding, dan memberikan penuntun arah. *Google Maps* API bersifat gratis untuk publik. Penggunaan *Google Maps* API pada pengembangan website android dengan menggunakan Eclipse dan computer menggunakan sistem operasi windows.

Kekurangan yang ada pada *Google Maps* API yaitu jika ingin melakukan akses harus terdapat layanan internet pada perangkat yang digunakan. Sedangkan kelebihan yang ada pada *Google Maps* API yaitu:

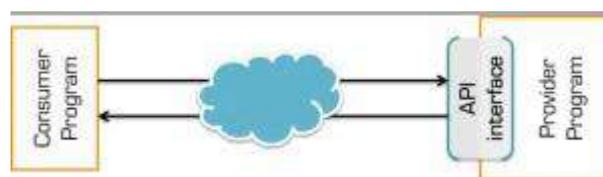
1. Dukungan penuh yang dilakukan *Google* sehingga terjamin dan bervariasi fitur yang ada pada *Google Maps* API.
2. Banyak pengembang yang menggunakan *Google Maps* API sehingga mudah dalam mencari referensi dalam pengembangan website

II.1.7 API (*Application Programming Interface*)

Manurut BA Pahlevy (2017) *Application Programming Interface* (API) adalah Sebuah bahasa dan pesan format yang digunakan oleh program website untuk berkomunikasi dengan sistem operasi atau program pengendalian lainnya seperti *Database Manajement System* (DBMS) atau protokol komunikasi. API diimplementasikan dengan menulis panggilan fungsi dalam program, yang menyediakan hubungan

ke subrutin yang diperlukan untuk eksekusi. Dengan demikian, API menyiratkan bahwa *driver* atau program modul tersedia dalam komputer untuk melakukan operasi atau perangkat lunak yang harus dihubungkan ke dalam program yang ada untuk melakukan tugas-tugas. API dapat menjelaskan cara sebuah tugas (*task*) tertentu dilakukan. Dalam pemrograman *procedural* seperti bahasa C, aksi biasanya dilakukan dengan media pemanggilan fungsi. Karena itu, API biasanya menyertakan penjelasan dari fungsi/rutin yang disediakan.

Application Programming Interface dibuat dengan maksud untuk memudahkan *programmer* dalam mengembangkan websitenya. Didalam API nantinya akan ada sekumpulan perintah, fungsi dan protokol yang dapat digunakan oleh *programmer* saat membangun perangkat lunak untuk sistem operasi tertentu. Dengan API tersebut memungkinkan *programmer* menggunakan perintah, fungsi dan protokol tersebut untuk berinteraksi dengan sistem operasi lain. Berikut ilustrasi atau cara kerja API :



Gambar 2.3 API Illustration

(Sumber : <https://asimetrissite.wordpress.com/apakah-application-programming-interface-api/>)

Application Programming Interface (API) dibuat dengan tujuan :

1. *System call* yang terdapat pada beberapa sistem operasimungkin saja

berbeda, dengan menggunakan API *programmer* dapat menjalankan programnya disistem mana saja, dengan syarat API tersebut sudah ter-*install* di sistem operasi yang digunakannya..

2. *System call* yang terdapat pada sistem operasi begitu banyak, dengan API *system call* tersebut dapat digunakan dengan bahasa yang lebih terstruktur dan mudah dimengerti oleh pembuat atau pengembangan program.

II.1.8 Website

Bahasa pemrograman berbasis web merupakan bahasa pemrograman yang digunakan untuk membuat suatu website yang dapat berinteraksi dengan semua *user* tanpa harus melakukan instalasi pada 1 pc atau laptop.

II.1.9 Leaflet JavaScript (LeafletJS)

Menurut Vladimir Agafonkin (2011), Leaflet JavaScript atau di singkat (LaefletJS) merupakan perpustakaan JavaScript yang bersifat *Open Source. Library* ini khusus digunakan untuk membangun website pemetaan berbasis web, memdukung sebagian besar *mobile* dan *desktop platform*.

II.1.10 Visual Studio Code (VSCode)

Visual Studio Code adalah kode editor sumber yang dikembangkan oleh *Microsoft* untuk *Windows*, *Linux* dan *macOS*. Ini termasuk dukungan untuk *debugging*, kontrol git yang tertanam dan GitHub, penyorotan sintaksis, penyelesaian kode cerdas, snippet,

dan refactoring kode. Ini sangat dapat disesuaikan, memungkinkan pengguna untuk mengubah tema, pintasan *keyboard*, preferensi, dan menginstal ekstensi yang menambah fungsionalitas tambahan.

II.1.11 XAMPP

Menurut Jogiyanto (2005:2) XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. XAMPP merupakan *tool* yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dengan menginstall XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi web server Apache, PHP dan MySQL secara manual. XAMPP akan menginstalasi dan mengkonfigurasikannya secara otomatis untuk anda atau *auto* konfigurasi.

II.1.12 Google Chrome

Menurut Bintara (2020), Google Chrome merupakan *browser web* yang di kembangkan oleh perusahaan Google. Chrome pertama kali dirilis pada tanggal 2 September 2008. Namun pada waktu pertama kali rilisnya Google Chrome ini hanya *software* yang digunakan untuk sistem Microsoft Windows serta kemudian *porting* Linux, macOS, iOS, dan juga android. Google Chrome ini merupakan *website browser* yang paling banyak digunakan oleh pengguna internet saat ini. Pada bulan November 2020, *StatCounter* telah memperkirakan bahwa Chrome ini memiliki 70% pangsa pasar *browser* di seluruh dunia (setelah memuncak pada 72.38% pada November 20018). Berikut adalah kelebihan dan

kekurangan dari Google Chrome :

- Kelebihan
 1. Tampilan *Modern*,
 2. Performa cepat,
 3. Aman dan nyaman,
 4. Integrasi dengan Google Apps,
 5. Fitur *inbuilt* yang kuat.
- Kekurangan
 1. Penggunaan Memori dan CPU tinggi,
 2. Pilihan kustomisasi dan opsi terbatas.

II.1.13 Bahasa Program

II.1.13.1 CSS

CSS yang merupakan singkatan dari *Cascading Style Sheet*. Jadi, dapat disimpulkan bahwa HTML5 digunakan untuk menentukan struktur dokumen *website* dan CSS bertugas untuk mempercantik *website* itu. Kira-kira dapat dianalogikan seperti seorang arsitek dan desainer interior. CSS adalah “kumpulan kode untuk mendefinisikan desain dari bahasa *markup*”. Karena ada kata bahasa *markup* pada definisi CSS, maka relasi antara CSS dan HTML sangatlah dekat. Dengan CSS sebuah desain *website* yang dibangun menggunakan HTML akan menjadi lebih menarik dan variatif.

II.1.13.2 Java Script

Javascript diperkenalkan pertama kali oleh Netscape pada tahun 1995. Pada awalnya bahasa ini dinamakan “LiveScript” yang berfungsi sebagai bahasa sederhana untuk browser Netscape Navigator 2. Javascript adalah bahasa yang berbentuk kumpulan skrip yang pada fungsinya berjalan pada suatu dokumen HTML, sepanjang sejarah internet bahasa ini adalah bahasa skrip pertama untuk *web*. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap bahasa HTML dengan mengizinkan pengeksekusian perintah perintah di sisi user, yang artinya di sisi *browser* bukan di sisi *server web*. Javascript bergantung kepada *browser (navigator)* yang memanggil halaman *web* yang berisi skrip-skrip dari Javascript dan tentu saja terselip di dalam dokumen HTML.

II.1.13.3 Hyper Text Markup Language (HTML)

HTML atau *Hyper Text Markup Language* merupakan sebuah bahasa pemrograman terstruktur yang dikembangkan untuk membuat halaman *website* yang dapat diakses atau ditampilkan menggunakan *Web Browser*. HTML sendiri secara resmi lahir pada tahun 1989 oleh Tim Barners Lee dan dikembangkan oleh *World Wide Web Consortium (W3C)*, yang kemudian pada tahun 2004 dibentuklah *Web Hypertext Application Tecnology Working Group (WHATG)* yang hingga kini bertanggung jawab akan perkembangan bahasa HTML

ini. Hingga kini telah mengembangkan HTML 5, sebuah versi terbaru dari HTML yang mendukung tidak hanya gambar dan teks, namun juga menu interaktif, audio, video dan lain sebagainya.

II.1.13.4 PHP

PHP merupakan bahasa pemrograman berbasis *web* yang memiliki kemampuan untuk memproses data dinamis. PHP dikatakan sebagai sebuah *server-side embedded script language* artinya sintaks-sintaks dan perintah yang kita berikan akan sepenuhnya dijalankan oleh *server* tetapi disertakan pada halaman HTML biasa. Website-website yang dibangun oleh PHP pada umumnya akan memberikan hasil pada *web browser* tetapi prosesnya secara keseluruhan dijalankan di *server*, pada prinsipnya *server* akan bekerja apabila ada permintaan dari *client*. Dalam hal ini *client* menggunakan kode-kode PHP untuk mengirimkan permintaan ke *server*.

II.1.14 Basis Data (Database)

Menurut Abdulloh (2018:103), Database atau basis data adalah kumpulan informasi yang disimpan dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi.

Menurut Enterprise (2017:1), Database adalah suatu website yang menyimpan sekumpulan data. Setiap database mempunyai perintah tertentu untuk membuat, mengakses, mengatur, mencari, dan menyalin

data yang ada di dalamnya.

Maka dari beberapa pendapat dapat disimpulkan bahwa database atau basis data adalah suatu website yang berisi sekumpulan informasi yang tersimpan secara sistematis di komputer.

Menurut Yanto (2016 : 13) Komponen penting dalam sistem basis data adalah :

1. Data

Merupakan informasi yang disimpan dalam suatu struktur tertentu yang terintegrasi.

2. *Hardware*

Merupakan perangkat keras berupa computer dengan media penyimpanan yang digunakan untuk menyimpan data karena pada umumnya basis data memiliki ukuran yang besar.

3. Sistem Operasi

Program yang mengaktifkan dan memfungsikan sistem komputer, mengendalikan seluruh sumber daya dalam komputer, dan melakukan operasi dasar dalam computer meliputi input, proses dan output.

4. Basis Data

Basis data sebagai inti dari sistem basis data. Basis data menyimpan data serta struktur sistem basis data baik entitas maupun objek-objek secara detail.

5. *Database Management System*

Merupakan perangkat lunak yang digunakan untuk melakukan pengelolaan basis data.

6. *User*

Merupakan penggunaan yang menggunakan data yang tersimpan dan dikelola. *User* dapat berupa seseorang yang mengelola basis data yang disebut *database administrator* (DBA), bisa juga disebut *end user*.

7. Website Lainnya

Program yang dibuat untuk memberikan *interface* kepada *user* sehingga lebih mudah dan terkontrol dalam mengakses basis data.

II.1.15 Database Management System (DBMS)

Menurut Abdulloh (2018:103), DBMS yaitu sistem perangkat lunak yang menyediakan layanan bagi user untuk membuat, mengontrol dan mengakses database. Sedangkan menurut Rosa (2016:44), DBMS (Database Management System) atau dalam bahasa Indonesia sering disebut sebagai sistem manajemen basis data adalah suatu sistem website yang digunakan untuk menyimpan, mengelola dan menampilkan data.

Berdasarkan penjelasan di atas penulis menyimpulkan bahwa DBMS (Database Management System) adalah suatu perangkat lunak yang digunakan untuk mengelola sebuah data.

II.1.16 SQLite

SQLite merupakan sebuah sistem manajemen basisdata relasional yang bersifat ACID-compliant dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. Tidak seperti paradigma client-server umumnya, inti SQLite bukanlah sebuah sistem yang mandiri yang berkomunikasi dengan sebuah program, melainkan sebagai bagian integral dari sebuah program secara keseluruhan. Sehingga protokol komunikasi utama yang digunakan adalah melalui pemanggilan API secara langsung melalui bahasa pemrograman. Mekanisme seperti ini tentunya membawa keuntungan karena dapat mereduksi overhead, latency times, dan secara keseluruhan lebih sederhana. Seluruh elemen basisdata (definisi data, tabel, indeks, dan data) disimpan sebagai sebuah file. Kesederhanaan dari sisi desain tersebut bisa diraih dengan cara mengunci keseluruhan file basis data pada saat sebuah transaksi dimulai.

Menurut Nazaruddin S (2012) Di Android juga memiliki fasilitas untuk membuat database yang dikenal dengan SQLite, SQLite adalah salah satu *software* yang embedded yang sangat populer, kombinasi SQL interface dan penggunaan *memory* yang sangat sedikit dengan kecepatan yang sangat cepat. SQLite di Android termasuk dalam Android *runtime*, sehingga setiap versi dari Android dapat membuat database dengan SQLite.

Dalam sistem Android memiliki beberapa teknik untuk melakukan penyimpanan data. Teknik yang umum digunakan adalah sebagai

berikut:

- a. *Shared Preferences* yaitu menyimpan data beberapa nilai (*value*) dalam bentuk *groups key* yang dikenal dengan *preference*.
- b. *Files* yaitu menyimpan data dalam *file*, dapat berupa menuliske *file* atau membaca dari *file*.
- c. *SQLite Database*, yaitu menyimpan data dalam bentuk *database*.
- d. *Content Providers*, yaitu menyimpan data dalam bentuk *content providers service*.

Pada bagian *database* ini tidak ada *database* yang disediakan oleh Android, jika menggunakan *SQLite* harus meng-*create database* sendiri, mendefinisikan tabelnya, *index* serta datanya. Untuk membuat dan membuka *database* yang paling baik adalah menggunakan *libraries*: `Import Android. database. sqlite. SQLite Open Helper`; Yang menyediakan tiga metode yaitu :

1. *Constructor*, menyediakan representasi versi dari *database* dan skema *database* yang kita gunakan.
2. *onCreate()*, menyediakan *SQLite Database object* yang tidak kita gunakan dalam definisi table dan inisialisasi data.
3. *onUpgrade()*, menyediakan fasilitas kovensi *database* dari *database* versi lama ke *database* versi baru atau sebaliknya.

II.1.17 UML (*Unified Modelling Language*)

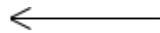
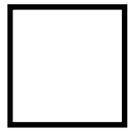
Menurut Rosa A.S dan M. Shalahudin (2014 : 133) *Unified Modeling Language* (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*,

membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. *Unified Modeling Language* (UML) muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. *Unified Modeling Language* (UML) hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan *Unified Modeling Language* (UML) tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya *Unified Modeling Language* (UML) paling banyak digunakan pada metodologi berorientasi objek.

II.1.17.1 Use Case Diagram

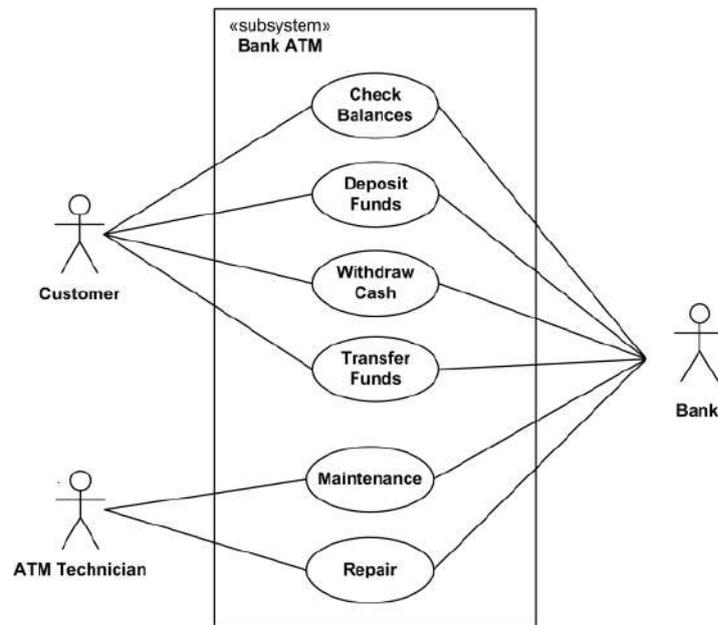
Rosa A.S dan M. Shalahudin (2014 : 155), *use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol yang ada pada diagram *use case*

Tabel 2.1 Simbol-simbol *Use Case* Diagram

NO.	GAMBAR	NAMA	KETERANGAN
1.		<i>Actor</i>	Menspesifikasikan himpunan peran yang dimainkan oleh pengguna ketika berinteraksi dengan <i>Use Case</i>
2.		<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>)
3.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit
4.		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
5.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas

6.		<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i> .
----	---	-----------------	---

Adapun berupa contoh gambar *UseCase* Diagram, sebagai berikut :



Gambar 2.4 *UseCase* Diagram Bank ATM

(Sumber : <https://www.dicoding.com/blog/wp-content/uploads/2020/04/use-case-diagram-atm.png>)

II.1.17.2 *Activity Diagram*

Menurut Rosa dan M. Shalahudin (2014 : 161), diagram aktifitas atau *activity* diagram menggambarkan *workflow* (aliran

kerja) atau aktifitas dan sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram *aktivitas* menggambarkan aktifitas sistem bukan yang dilakukan *actor*, jadi aktifitas yang dapat dilakukan oleh sistem.

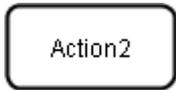
Activity diagram menggambarkan aliran fungsional sistem, pada tahap pemodelan bisnis, diagram aktivitas dapat digunakan untuk menunjukkan aliran kerja bisnis dan untuk menggambarkan aliran kejadian dalam use case. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan,
2. Urutan atau pengelompokan tampilan dari sistem dimana setiap aktivitas dianggap memiliki sebuah rancangan antar muka tampilan.
3. Rancangan pengujian dimana aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

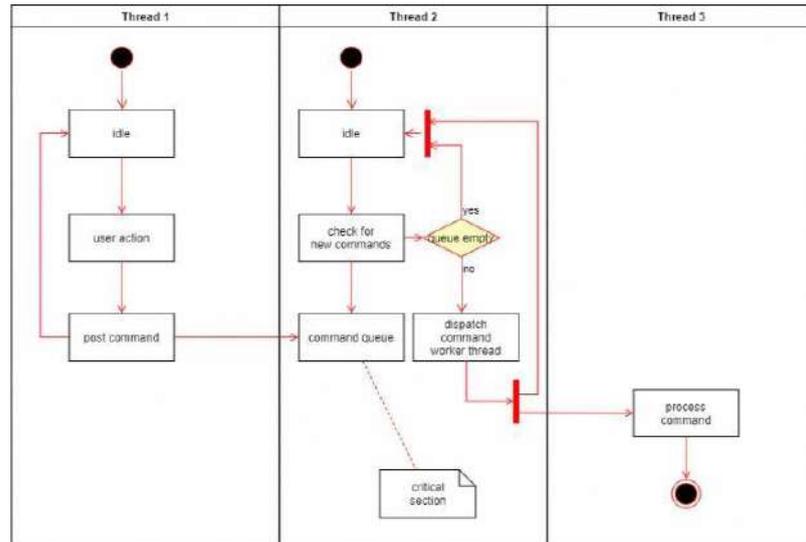
Berikut adalah simbol-simbol yang ada pada diagram kelas:

Tabel 2.2 Simbol-simbol *Activity Diagram*

NO.	GAMBAR	NAMA	KETERANGAN
1.		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali

2.		<i>Partition (vertical)</i>	Pengelompokkan aktifitas berdasarkan aktor dan sistem
3.		<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi
4.		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran
5.		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan diakhiri
6.		<i>Decision Node dan Merge Node</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 Objek

Adapun berupa contoh gambar *Activity Diagram*, sebagai berikut :



Gambar 2.5 Contoh Activity Diagram

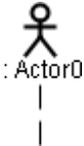
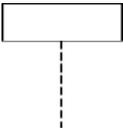
(Sumber : <https://www.dicoding.com/blog/wp-content/uploads/2020/04/internship-Rendi-activity-diagram-768x521.png>)

II.1.17.3 Sequence Diagram

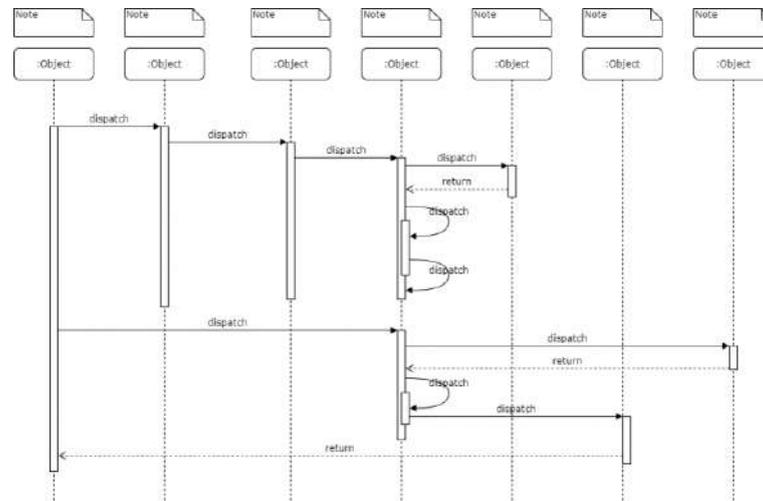
Menurut Rosa dan M. Shalahudin (2014 : 165), diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek- objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang di instansikan menjadi objek itu. Banyak diagram sekuen yang harus digambarkan adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak. Berikut

adalah symbol-simbol yang ada pada diagram sekuen :

Tabel 2.3 Simbol-simbol *Sequence Diagram*

NO.	GAMBAR	NAMA	KETERANGAN
1.		<i>LifeLine (Actor)</i>	Objek <i>actor</i> , antarmuka yang saling berinteraksi
2.		<i>LifeLine (Entity)</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi
3.		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
4.		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
5.		<i>Message</i>	Pesan yang dikirim untuk objek itu sendiri
6.		<i>Execution Occurance</i>	Menunjukkan fokus kontrol objek pada suatu waktu

Adapun berupa contoh gambar *Activity Diagram*, sebagai berikut :



Gambar 2.6 Contoh *Sequence Diagram*

(Sumber : <https://www.dicoding.com/blog/wp-content/uploads/2020/04/internship-Rendi-sequence-diagram-768x504.png>)

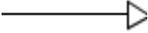
II.1.17.4 *Class Diagram*

Menurut Rosa dan M. Shalahudin (2014 : 141), diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan *method* atau operasi. Berikut penjelasan atribut dan *method* :

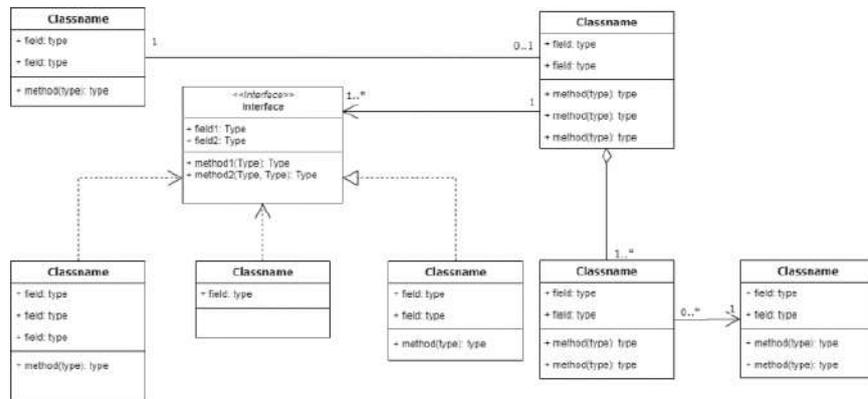
1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Berikut adalah simbol-simbol yang ada pada diagram kelas

Tabel 2.4 Simbol-simbol *Class Diagram*

NO.	GAMBAR	NAMA	KETERANGAN
1.		<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>)
2.	Class	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama
3.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit
4.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya
5.	1 , 1..*	<i>Multiplicity</i>	Jumlah banyaknya objek sebuah <i>class</i> yang berelasi dengan sebuah objek lain pada <i>class</i> lain yang berasosiasi dengan <i>class</i> Tersebut

Adapun berupa contoh gambar *Class Diagram*, sebagai berikut:



Gambar 2.7 Contoh *Class Diagram*

(Sumber : <https://www.dicoding.com/blog/wp-content/uploads/2020/04/internship-Rendi-class-diagram-768x352.png>)

II.1.18 *Flowchart*

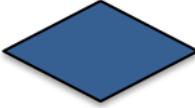
Menurut Iswandy (2015), *Flowchart* merupakan urutan-urutan langkah kerja suatu proses yang di gambarkan dengan menggunakan simbol-simbol yang disusun secara sistematis.

Sedangkan menurut Harumy dkk (2016), Secara formal, *flowchart* didefinisikan sebagai skema penggambaran dari algoritma atau proses. Table berikut menampilkan simbol-simbol yang digunakan dalam menyusun *flowchart*.

Berikut ini adalah simbol-simbol yang ada pada *flowchart* :

Tabel 2.5 Simbol-simbol dalam *Flowchart*

No.	Nama	Simbol	Fungsi / Keterangan
1.	<i>Terminator</i>		Sebagai simbol ' <i>START</i> ' atau ' <i>END</i> ' untuk memulai atau mengakhiri <i>flowchart</i>

2.	<i>Input/Output</i>		Digunakan untuk menuliskan proses menerima data atau mengeluarkan data.
3.	Proses		Digunakan untuk menuliskan proses yang diperlukan, misalnya operasi aritmatika.
4.	Conditional / Decision		Digunakan untuk menyatakan proses yang membutuhkan keputusan
5.	Preparation		Digunakan untuk memberikan nilai awal.
6.	Arrow		Sebagai penunjuk arah dan alur proses.
7.	Connector (On-page)		Digunakan untuk menyatukan arrow.
8.	Connector (Off-page)		Digunakan untuk menghubungkan <i>flowchart</i> yang harus digambarkan pada halaman yang berbeda. Biasanya pada simbol ini diberi nomor sebagai penanda, misalnya angka 1.

9.	Display		Digunakan untuk menampilkan data ke <i>monitor</i> .
----	---------	---	--

II.1.19 Pengujian Sistem

Pengujian sistem adalah pengujian program perangkat lunak yang lengkap dan terintegrasi. Perangkat lunak atau yang sering dikenal dengan sebuah *software* hanyalah satuan elemen dari sistem berbasis komputer yang lebih besar. Biasanya, perangkat lunak dihubungkan dengan perangkat lunak dan perangkat keras lainnya. Pengujian perangkat lunak dapat dibedakan menjadi dua yaitu *Black Box* dan *White Box Testing*.

II.1.19.1 *Black Box Testing*

Menurut Syafnidawaty (2020), *Black Box Testing* atau yang sering dikenal dengan sebutan pengujian fungsional merupakan metode pengujian Perangkat Lunak yang digunakan untuk menguji perangkat lunak tanpa mengetahui struktur internal kode atau Program. Dalam pengujian ini, tester menyadari apa yang harus dilakukan oleh program tetapi tidak memiliki pengetahuan tentang bagaimana melakukannya.

Pada *Black Box Testing* ini dilakukan pengujian yang didasarkan pada detail website seperti tampilan website, fungsi-fungsi yang ada pada website, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan oleh *customer*. *Black-box testing* ini lebih menguji ke

tampilan luar (*Interface*) dari suatu website agar mudah digunakan oleh pengguna. Pengujian ini tidak melihat dan menguji *source code* program. *Black-box testing* bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya hanya terfokus pada informasi domain.

Sedangkan menurut Rizky dikutip oleh Indriyani dalam jurnal PARADIGMA (2015 : 30) *Black box testing* adalah tipe *testing* yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para tester memandang perangkat lunak seperti layaknya sebuah “kotak hitam” yang tidak penting dilihat isinya, tapi cukup dikenali proses *testing* di bagian luar.



Gambar 2.8 Sistem Pengujian
(Sumber : <https://raharja.ac.id/2020/10/20/black-box-testing/>)

Berikut ini adalah keunggulan dan kelemahan *black box testing* :

1. Keunggulan *Black Box Testing*

Meskipun dalam pelaksanaan *testing* kita dapat menguji keseluruhan fungsional perangkat namun formal *black box testing* yang sebenarnya kita dapat memilih subset tes yang secara efektif dan efisien dapat menemukan cacat. Dengan cara ini *black box testing* dapat membantu memaksimalkan program.

2. Kelemahan *Black Box Testing*

Ketika tester melakukan *black box testing*, tester tidak akan pernah yakin apakah perangkat lunak yang diuji telah benar-benar lolos pengujian. Hal ini terjadi karena kemungkinan masih ada

beberapa jalur eksekusi yang belum pernah diuji. Untuk menemukan cacat perangkat lunak menggunakan *black box testing*, tester seharusnya membuat setiap kemungkinan kombinasi data *input* benar. Adapun contoh kasus pengujian *black box*, sebagai berikut :

Pengujian *Black Box* Metode yang digunakan dalam pengujian alpha adalah metode *black box* yang focus pada persyaratan fungsional dari perangkat lunak yang dibangun. Berdasarkan rencana pengujian, maka dapat dilakukan pengujian *black box* pada *prototype* website peta hama dan penyakit ikan yang dijelaskan pada tabel dibawah ini :

Tabel 2.6 Pengujian *Black Box* pada *prototype* website

Aktivitas Pengujian	Realisasi yang diharapkan	Hasil Pengujian	Kesimpulan
Masuk halaman awal	Terhubung dengan API Google Maps	Muncul Peta Kota Kediri.	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
Klik Dropdown Filter criteria	Muncul kriteria sesuai dengan label	Dropdown dapat berfungsi sesuai yang diharapkan.	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
Klik tombol "Cari Lokasi"	Muncul peta Lokasi sesuai kriteria yang dimasukkan	Tombol "Cari Lokasi" berfungsi sesuai yang diharapkan.	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
Klik tombol "Cari Lokasi"	Muncul peta lokasi dengan warna yang berbeda.	Ada beberapa peta lokasi yang memiliki warna yang sama	<input type="checkbox"/> Diterima <input checked="" type="checkbox"/> Ditolak
Klik pada peta lokasi	Muncul informasi detail lokasi pada setiap peta	Popup informasi berfungsi sesuai yang diharapkan	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
Akses Melalui perangkat <i>smart phone</i> layar ≥ 5 inch	Tampilan sesuai dengan ukuran layar smartphone	Belum mampu menampilkan tampilan sesuai dengan layar smartphone	<input type="checkbox"/> Diterima <input checked="" type="checkbox"/> Ditolak

Hasil dari pengujian dengan *black box* ada beberapa hasil pengujian yang belum sesuai dengan realisasi yang diharapkan sehingga dilakukan beberapa perbaikan sesuai dengan masalah yang didapatkan, berikut perbaikan yang dilakukan :

1. Perbaikan pertama dilakukan pengecekan warna peta yang ada kemunculan warna yang sama. Langkah yang dilakukan untuk memperbaiki bug tersebut dengan mengecek logika warna dan menambahkan field status warna pada tabel untuk memudahkan pemberian warna sehingga warna muncul sesuai yang diharapkan.
2. Perbaikan kedua dilakukan pengecekan fungsi tampilan yang tidak sesuai dengan perbedaan pada akses *device* layar yang berbeda. Langkah yang dilakukan dengan menambahkan *style* menggunakan “*Bootstrap*”. *Bootstrap* merupakan *framework* untuk membangun desain web secara responsive. Artinya, tampilan web yang dibuat oleh bootstrap akan menyesuaikan ukuran layar dari browser yang kita gunakan baik di desktop, tablet ataupun *mobile- device*.

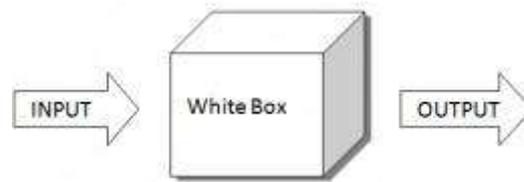
Setelah perbaikan dilakukan maka dilakukan pengujian ulang dengan *black box*. Pengujian hanya dilakukan pada fungsi yang belum sesuai yang diharapkan. Pengujian ulang *black box* pada *prototype* website pemetaan hama dan penyakit ikan yang dijelaskan pada tabel dibawah ini :

Tabel 2.7 Hasil dari Pengujian ulang *black box* pada *prototype* website

Aktivitas Pengujian	Realisasi yang diharapkan	Hasil Pengujian	Kesimpulan
Klik tombol “Cari Lokasi”	Muncul peta lokasi dengan warna yang berbeda.	Warna peta pada <i>prototype</i> aplikasi berfungsi sesuai yang diharapkan	[x] Diterima [] Ditolak
Akses Melalui perangkat <i>smart phone</i> layar ≥ 5 inch	Tampilan sesuai dengan ukuran layar <i>smartphone</i>	Tampilan <i>prototype</i> aplikasi berfungsi sesuai yang diharapkan	[x] Diterima [] Ditolak

II.1.19.2 *White Box Testing*

Menurut Rizky dikutip oleh Indriyani dalam jurnal PARADIGMA (2015 : 30), *White box testing* secara umum merupakan jenis *testing* yang lebih berkonsentrasi terhadap isi dari perangkat lunak itu sendiri. Jenis ini lebih banyak berkonsentrasi pada *source code* dari perangkat lunak yang dibuat sehingga membutuhkan proses *testing* yang jauh lebih lama dan lebih mahal dikarenakan membutuhkan ketelitian dari para tester serta kemampuan teknis pemrograman bagi para *testernya*.



Gambar 2.9 Pengujian Sistem

(Sumber : <https://www.erfaansetyadi.com/2019/01/08/pengertian-white-box-testing/>)

Berikut ini adalah Kelebihan *White Box Testing* :

1. Kesalahan logika Digunakan pada sintaks “*if*” dan pengulangan. Dimana *white box testing* akan mendeteksi kondisi-kondisi yang tidak sesuai dan mendeteksi kapan proses pengulangan akan berhenti.
2. Ketidakesuaian asumsi Menampilkan asumsi yang tidak sesuai dengan kenyataan, untuk dianalisa dan diperbaiki.
3. Kesalahan ketik Mendeteksi bahasa pemrograman yang bersifat *case sensitive*. Kelemahan *White Box Testing* Untuk perangkat

lunak yang tergolong besar, *White Box Testing* dianggap sebagai strategi yang tergolong boros karena akan melibatkan sumber daya yang besar untuk melakukannya.

Adapun contoh kasus pengujian *black box*, sebagai berikut :

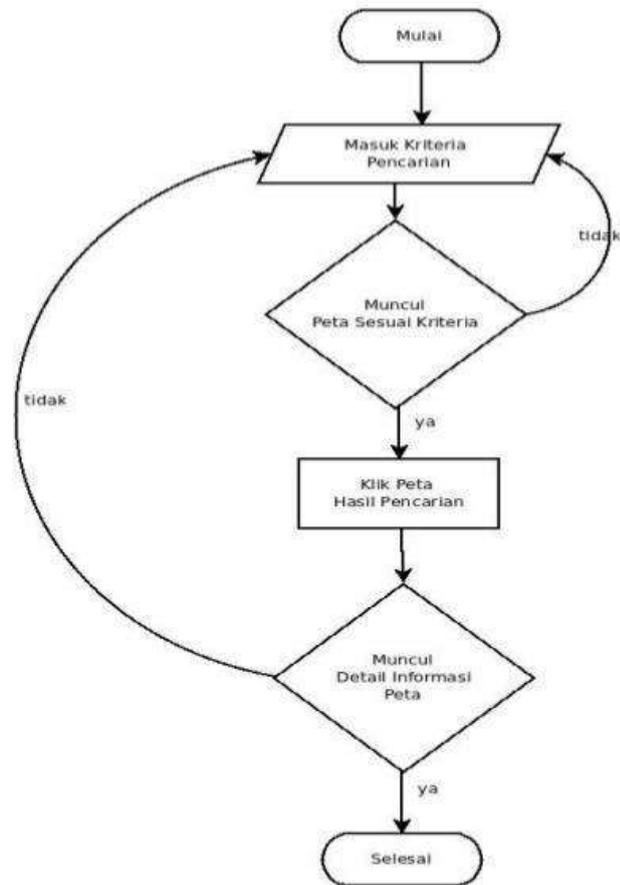
Pengujian white box merupakan metode perancangan test case yang menggunakan struktur kontrol dari perancangan prosedural dalam mendapatkan test case. Pengujian dilakukan dengan skema flowchart diagram gambar dibawah ini :



Gambar 2.10 Pengujian *Flowchart* Menampilkan Peta (Sumber : https://1.bp.blogspot.com/-4e4_xRSy558/W5HeVotLxuI/AAAAAAAAADEM/nGIWuo3KEnc5En4b91F5RHQfSuW2C6M7wCEwYBhgL/s1600/Selection_023.png)

Pengujian dilakukan untuk melihat apakah halaman awal ketika dibuka muncul peta seluruh wilayah Kota. Hasil dari flowchart menunjukkan wilayah kota muncul saat masuk halaman awal alamat prototype website. Pengujian Kedua dilakukan pengujian logika apakah alamat kriteria muncul saat dimasukkan filter sesuai kriteria.

Hasil dari pengujian hasil muncul sesuai dengan kriteria yang dimasukkan. Berikut ini flowchart pengujian kedua pada gambar dibawah :



Gambar 2.11 Flowchart pengujian kedua

(Sumber : [https://2.bp.blogspot.com/-](https://2.bp.blogspot.com/-mh6jKOWz5Mg/W5HetY572nI/AAAAAAAAADEU/g6zfAiqsSUMgalIb15u_abTsYc2xQ4FHLQCLcBGAs/s1600/Selection_024.png)

[mh6jKOWz5Mg/W5HetY572nI/AAAAAAAAADEU/g6zfAiqsSUMgal](https://2.bp.blogspot.com/-mh6jKOWz5Mg/W5HetY572nI/AAAAAAAAADEU/g6zfAiqsSUMgalIb15u_abTsYc2xQ4FHLQCLcBGAs/s1600/Selection_024.png)

[Ib15u_abTsYc2xQ4FHLQCLcBGAs/s1600/Selection_024.png](https://2.bp.blogspot.com/-mh6jKOWz5Mg/W5HetY572nI/AAAAAAAAADEU/g6zfAiqsSUMgalIb15u_abTsYc2xQ4FHLQCLcBGAs/s1600/Selection_024.png))

Analisa dari hasil pengujian white box menunjukkan website sudah sesuai harapan dari sisi ketepatan logika yang ada pada prototype.

II.2 Penelitian Terdahulu (*State of The Art*)

Tabel 2.8 Penelitian Terdahulu

No.	Nama Peneliti	Judul Penelitian	Tanggal & Waktu Penelitian	Metode	Objek
1.	Taufik Ridho As Siddiqi	Desain Website Sistem Informasi Kajian Keislaman Berbasis Android	2018	<i>SUS</i> <i>Questionnaire</i>	Masyarakat umur dan takmir masjid
2.	Reza Maulana & Muhamma d Irfan	Penerapan Progressive WEB APPS (PWA) Pada Sistem Layanan Konsultasi Kepada Pakar Syari'ah Islam	2019	<i>Progressive</i> <i>Web Apps</i> (PWA)	Ustadz dan <i>User</i> (Masyarakat)
3.	Endar Suprih Wihidayat	Info Ustadz Website Android	Semarang, 10 Oktober	Sinkronisasi Data	Masyarakat umum dan Ustadz

	& Bagus Dwi Cahyo	Pencarian Ustadz, Tausiyah, Dan Lokasi Pengajian Dengan Sistem	2015		
4.	Ryan Zulham Ramadhani	Rancang Bangun Website Jadwal Kajian Islam Berbasis Android Pada Perusahaan Sd Technologies	Tangerang , Oktober 2018	- <i>Extreme Programmin g (XP) - Object- Oriented Analysis (OOA)</i>	Tempat Kajian, Panitia Kajian, dan Peserta Kajian

5.	Harni Kusniyati & Haries Fadhillah	Website Pencarian Ustadz Untuk Wilayah Dki Jakarta Menggunakan Algoritma Haversine Formula BerbasisAndroid	September 2016	<i>Haversine</i> <i>Formula</i>	Lokasi, Ustadz, dan <i>User</i> (Masyarakat)
----	---	---	-------------------	------------------------------------	---

II.3 Kerangka Pikir

Berikut ini kerangka pikir dari website pencarian ustadz untuk wilayah makassar menggunakan algoritma *haversine formula* berbasis android.

Seiring dengan kemajuan teknologi, Saat ini telepon genggam (*handphone*) yang sedianya sebagai alat komunikasi sudah memiliki kelebihan yang cukup banyak dari fungsi dasarnya. Adapun peran ustadz/ustadzah adalah Berdakwah, tabligh, khutbah, mengayomi, mengajarkan, mendidik sekaligus membina dan membimbing dalam menanamkan nilai-nilai akhlak kepada masyarakat/santrinya agar menjadi generasi yang shaleh dan akram.



Di Kota Makassar terdapat ratusan masjid dan lokasi ceramah yang masih minim informasi alamatnya sehingga Ustadz atau penceramah seringkali kesulitan menemukan lokasi yang akan ditempati ceramah. Kesulitan ini menyebabkan Ustadz atau penceramah kadang tidak melaksanakan tugasnya di lokasi yang telah dijadwalkan.



Aplikasi ini menggunakan metode algoritma *haversine formula* untuk memudahkan para pengguna, *Haversine formula* nantinya akan digunakan dalam perhitungan jarak antara dua titik yaitu titik posisi *user* dan titik posisi tujuan.



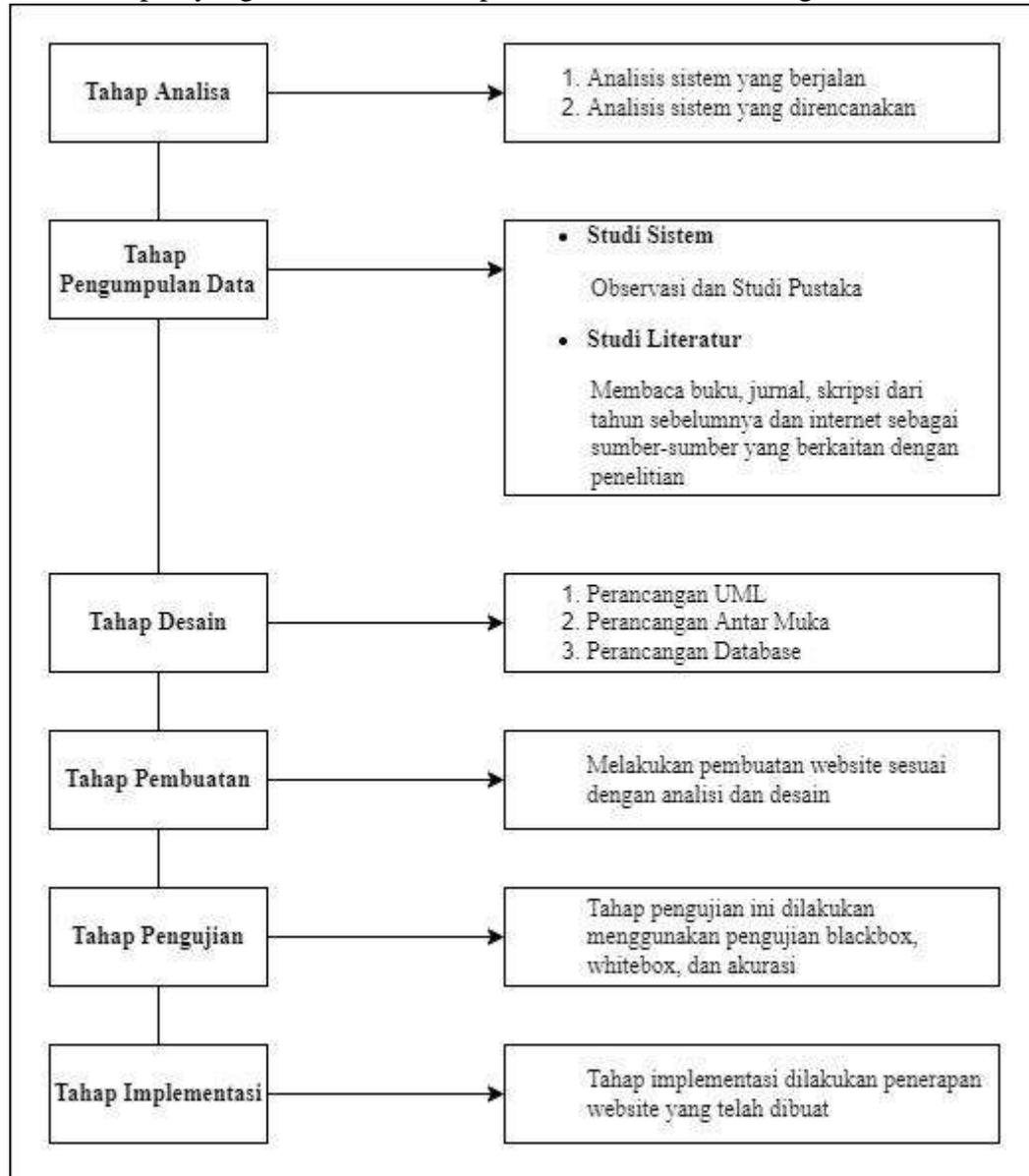
Dengan adanya aplikasi ini akan memudahkan masyarakat untuk menemukan ustadz, yang dimana kegiatan seperti tausiyah, ceramah, dan bahkan pengajian bisa dengan mudah berlangsung tanpa mengkhawatirkan tentang sulitnya untuk mendapatkan ustadz.

BAB III

METODOLOGI PENELITIAN

III.1 Prosedur Penelitian

Tahapan yang dilakukan dalam penelitian ini adalah sebagai berikut :

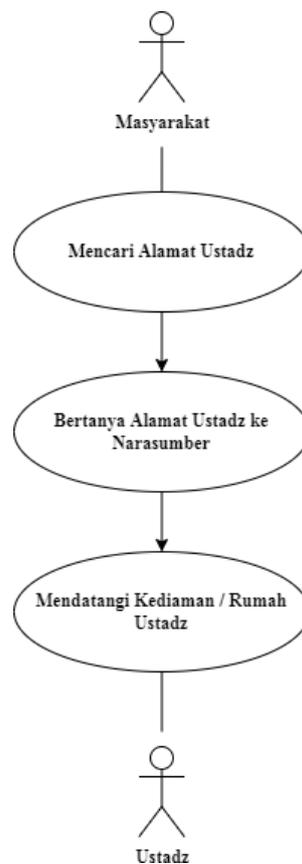


Gambar 3.1 Prosedur Penelitian

III.2 Rancangan Sistem

III.2.1 Sistem Berjalan

Sebelum dilakukan perancangan dan pembuatan website pencarian ustadz ini perlu dilakukan analisis terhadap permasalahan yang ada berkaitan dengan tujuan perancangan yang ditentukan sejak awal.



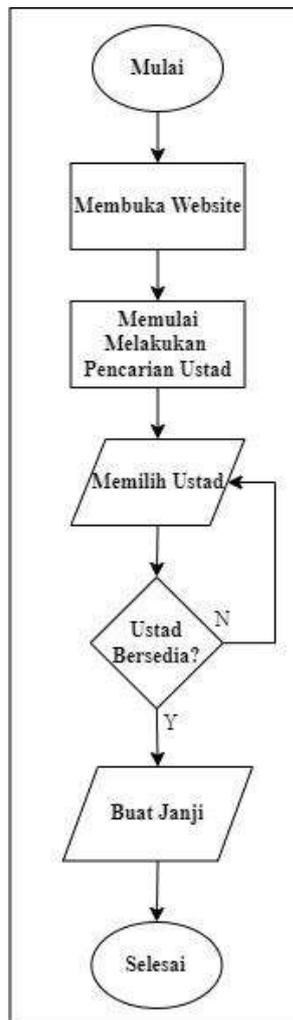
Gambar 3.2 Sistem yang Berjalan

Pada Diagram Flowchart diatas dapat terlihat bahwa Masyarakat harus mencari ustadz dengan cara bertanya ke Narasumber (Masyarakat lain) kemudian mengunjungi kediaman Ustadz atau menghubungi Ustadz tapi jikalau Ustadz tidak bisa menghadiri kegiatan maka Masyarakat harus mencari Narasumber lain untuk bisa menemukan

Ustadz yang bisa menghadiri kegiatan yang akan diadakan oleh Masyarakat.

III.2.2 Sistem Direncanakan

Sistem pencarian ustadz ini didukung oleh program website mobile sehingga Masyarakat tidak kesulitan untuk mencari Ustadz dan tidak membutuhkan waktu yang lama.

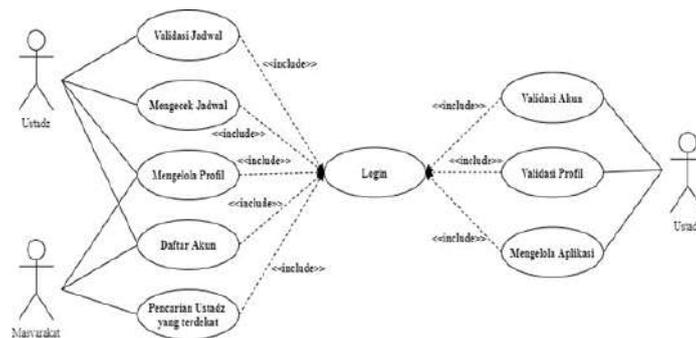


Gambar 3.3 Flowchart Penyelesaian Masalah

Pada Flowchart diatas dapat terlihat bahwa Masyarakat hanya membuka website dan melakukan *login* lalu membuka menu pencarian Ustadz terdekat dan memilih Ustadz terdekat. Setelah memilih Ustadz terdekat, Masyarakat hanya menunggu tanggapan dari Ustadz yang telah dipilih. Tapi jikalau Ustadz tidak bisa menghadiri kegiatan maka Masyarakat hanya kembali memilih Ustadz terdekat lainnya.

III.2.2.1 Use Case Diagram

Sebelum membuat website, kita perlu menggambarkan rancangan menu dan berapa aktor yang akan digunakan. Berikut ini adalah gambaran rancangan menu dan berapa aktor yang akan digunakan :

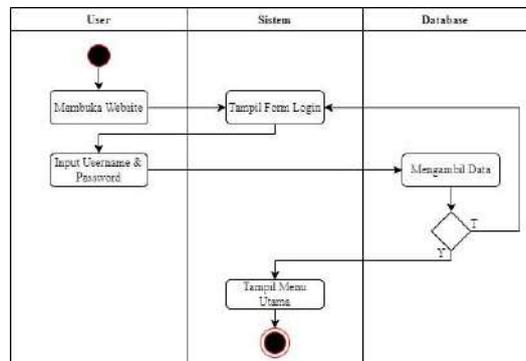


Gambar 3.4 Use Case Diagram Website

Use Case Diagram di atas menggambarkan bahwa ada 3 User yang terlibat pada sistem informasi ini yaitu admin, ustadz, dan masyarakat. Admin memiliki akses validasi profil dan kelola website. Sedangkan ustadz memiliki akses mengelola profil dan menjawab pertanyaan. Sedangkan masyarakat memiliki akses berupa mencari ustadz yang terdekat dan tanya ustadz.

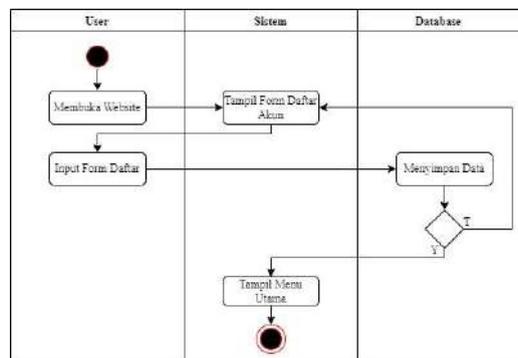
III.2.2.2 Activity Diagram

A. Activity Login



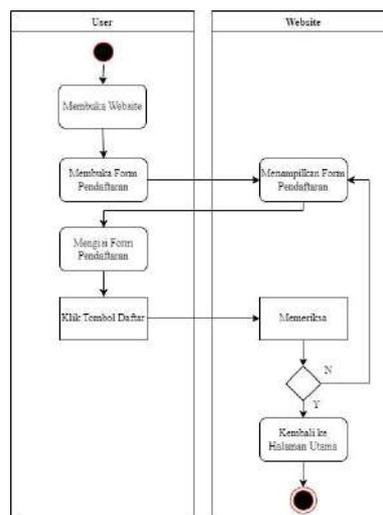
Gambar 3.5 Activity Login

B. Activity Daftar Akun



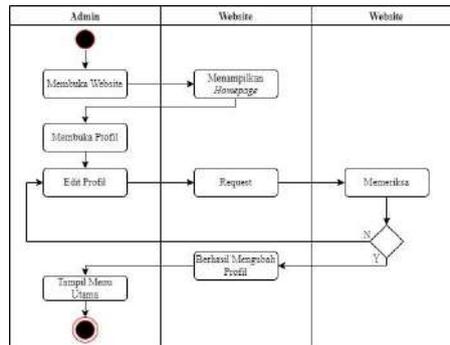
Gambar 3.6 Activity Daftar Akun

C. Activity Form Pendaftaran Ustad



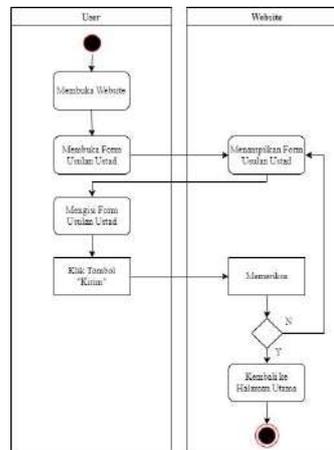
Gambar 3.7 Activity Form Pendaftaran Ustad

D. Activity Validasi Profil



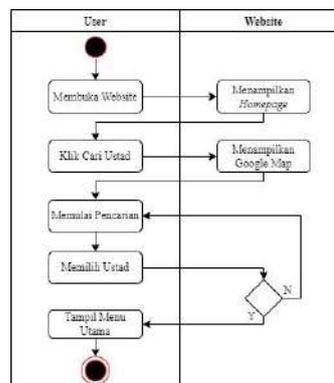
Gambar 3.8 Activity Diagram Validasi Profil

E. Activity Form Usulan Ustad



Gambar 3.9 Activity Form Usulan Ustad

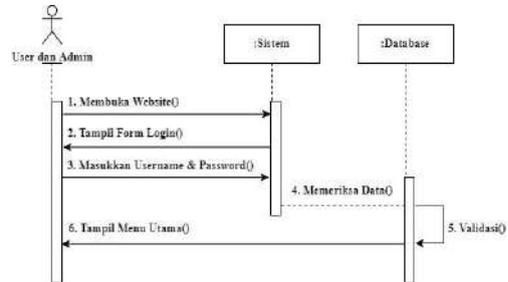
F. Activity Mencari Ustadz yang Terdekat



Gambar 3.10 Activity Diagram Mencari Ustadz yang Terdekat

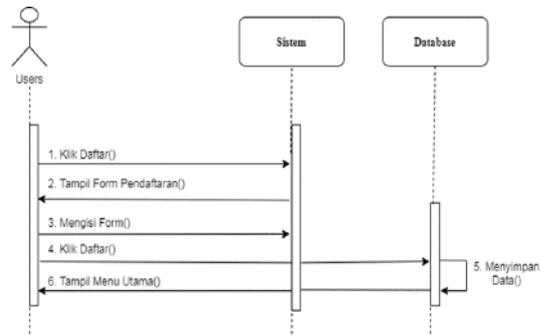
III.2.2.3 Sequence Diagram

A. Sequence Login



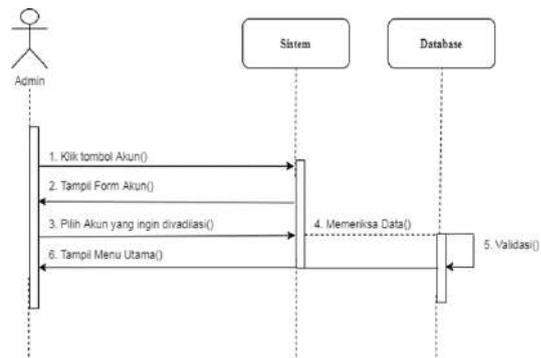
Gambar 3.11 Sequence Logn

B. Sequence Daftar Akun



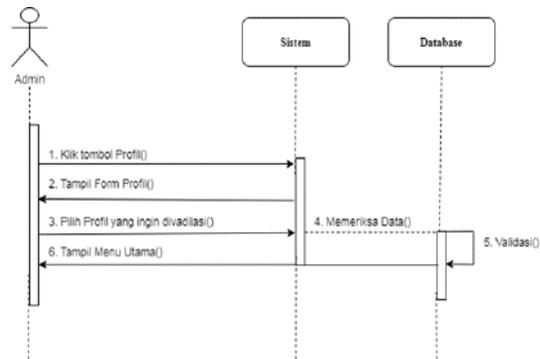
Gambar 3.12 Sequence Daftar Akun

C. Sequence Validasi Akun



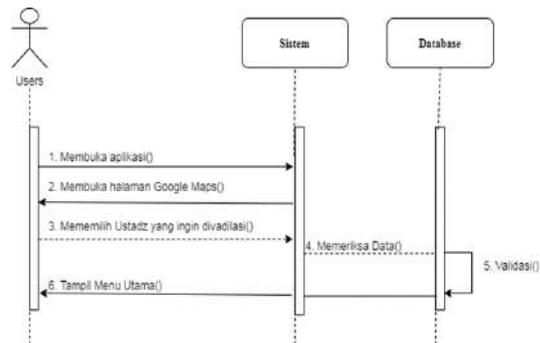
Gambar 3.13 Sequence Validasi Akun

D. Sequence Validasi Profil



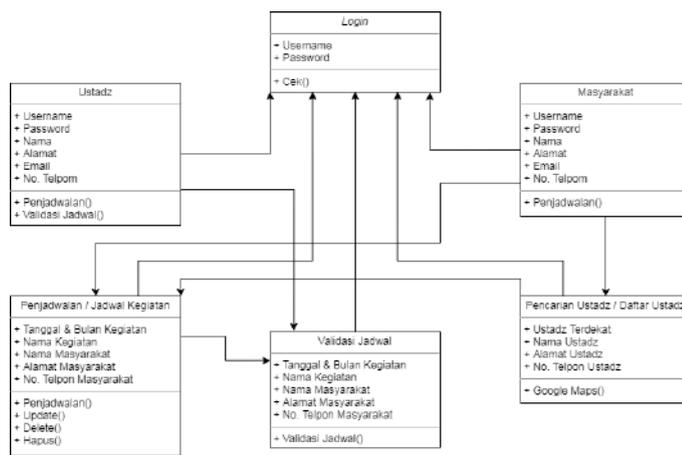
Gambar 3.14 Sequence Validasi Profil

E. Sequence Mencari Ustadz yang terdekat



Gambar 3.15 Sequence Mencari Ustadz yang Terdekat

III.2.2.4 Class Diagram



Gambar 3.16 Class Diagram

1. Perancangan Database

a. Users

No.	Kolom	Tipe	Keterangan
1.	Id	Int(11)	
2.	Nama	String	
3.	Password	String	
4.	Nama Admin	Varchar(25)	

b. Ustadz

No.	Kolom	Tipe	Keterangan
1.	User_id	Int(11)	
2.	Nama	Varchar(25)	
3.	Alamat	String	
4.	Password	String	
5.	Email	String	
6.	No. Telp	Int(13)	

c. Masyarakat

No.	Kolom	Tipe	Keterangan
1.	User_id	Int(11)	
2.	Nama	Varchar(25)	
3.	Password	String	
4.	Alamat	String	
5.	Email	String	
6.	No. Telp	Int(13)	

d. Pencarian Ustadz / Daftar Ustadz

No.	Kolom	Tipe	Keterangan
1.	Id_Ustadz	Int(11)	

2.	Nama_Ustadz	Varchar(25)	
3.	Alamat	String	
4.	No. Telp	Int(13)	

III.2.3 Perancangan Antarmuka



Gambar 3.17 Tampilan Cari Ustadz



Gambar 3.18 Tampilan Tanya Ustadz



Gambar 3.19 Tampilan Daftar Ustadz



Gambar 3.20 Tampilan Google Maps

III.3 Waktu dan Tempat Penelitian

III.3.1 Waktu Penelitian

Dalam Penelitian ini ditentukan waktu penelitian dilakukan pada bulan Januari – Maret 2022.

III.3.2 Tempat Penelitian

Penelitian dilaksanakan di sekitaran Kecamatan Manggala, Kota Makassar, Sulawesi Selatan 90234.

III.4 Alat dan Bahan Penelitian

Kebutuhan dalam perancangan website ini terdiri dari perangkat keras (hardware) dan perangkat lunak (*software*), antara lain :

Tabel 3.1 *Hardware dan Software*

Perangkat keras (<i>Hardware</i>)	Perangkat lunak (<i>Software</i>)
Laptop	Sistem operasi Windows 10
	Flutter
<i>Smartphone Android</i>	
Printer	

III.5 Metode Pengumpulan Data

Teknik pengumpulan data merupakan suatu cara yang digunakan oleh peneliti untuk mengumpulkan data terkait permasalahan penelitian yang diambilnya. Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam rangka mencapai tujuan penelitian. Dalam penelitian ini, penulis mengumpulkan data, melalui: observasi, mengamati data, membaca dan mengutip dari internet serta sumber- sumber lain yang berhubungan dengan penelitian ini.

III.6 Tahap Analisis Data / Pengujian Sistem

Metode pengujian yang digunakan pada penelitian ini yaitu metode *Black Box Testing* dan *White Box Testing*. Pengujian *Black-Box* berfokus pada persyaratan fungsional dan suatu sistem. *White box* testing adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara procedural.

BAB IV

HASIL DAN PEMBAHASAN

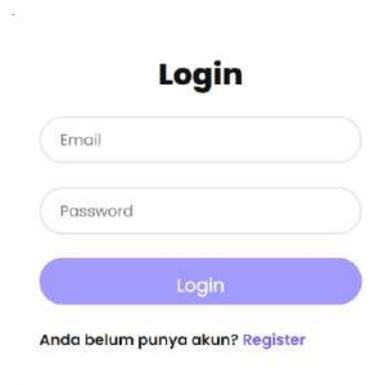
IV.1 Hasil

Perancangan website ini diberi nama Cari Ustadz Yuk (CUY) dengan menggunakan metode Haversine Formula, Berdasarkan penelitian yang telah dilakukan, berikut adalah hasil yang telah diperoleh:

IV.1.1 Tampilan *Login dan Register*

a. Tampilan *Login*

login ialah menu yang digunakan untuk masuk dalam suatu website atau web dan lain sebagainya. Apabila anda sudah melakukan proses *register*, anda bisa masuk ke website ataupun web dengan cara memasukan email ataupun username di kolom email. Selain itu, anda juga perlu menuliskan password di kolom password.



Login

Email

Password

Login

Anda belum punya akun? [Register](#)

Gambar 4.1 Tampilan *Login*

Gambar 4.1 menunjukkan tampilan yang menjadi gerbang awal untuk mengakses *beranda website*. Berikut merupakan *script* dari gambar 4.1 :

```

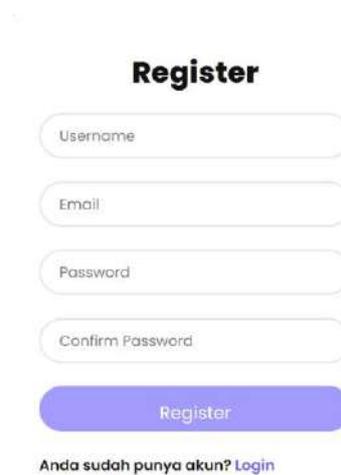
<div class="container">
  <form action="" method="POST" class="login-email">
    <p class="login-text" style="font-size: 2rem; font-weight: 800;">Login</p>
    <div class="input-group">
      <input type="email" placeholder="Email" name="email" value="<?php echo $email; ?>" required>
    </div>
    <div class="input-group">
      <input type="password" placeholder="Password" name="password" value="<?php echo $_POST
      ['password']; ?>" required>
    </div>
    <div class="input-group">
      <button name="submit" class="btn">Login</button>
    </div>
    <p class="login-register-text">Anda belum punya akun? <a href="Login/register.php">Register</a></p>
  </form>
</div>

```

Gambar 4.2 Script Tampilan Login

b. Tampilan Register

Sign up merupakan menu yang digunakan untuk mendaftar menjadi anggota dari sebuah web atau website. Dalam proses pendaftaran, anda diharuskan untuk mengisi sebuah formulir mengenai identitas anda secara tepat seperti halnya akun e-mail, nama depan, password.



Gambar 4.3 Tampilan Register

Gambar 4.3 menunjukkan tampilan yang menjadi gerbang awal untuk bisa melakukan *Login*. Berikut merupakan *script* dari gambar 4.3 :

```

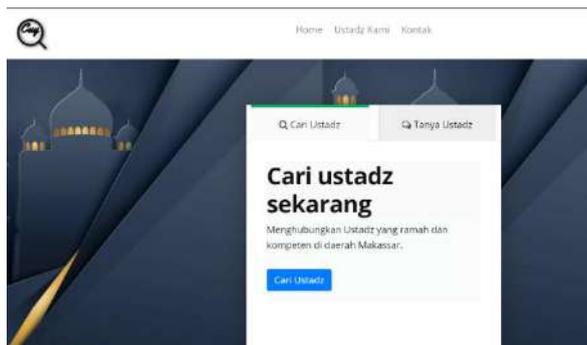
<div class="container">
  <div class="row">
    <div class="col-md-12">
      <div class="text-center">
        <h2>Register</h2>
      </div>
      <div class="form-group">
        <input type="text" class="form-control" value="" placeholder="Nama Lengkap" />
      </div>
      <div class="form-group">
        <input type="text" class="form-control" value="" placeholder="Email" />
      </div>
      <div class="form-group">
        <input type="password" class="form-control" value="" placeholder="Password" />
      </div>
      <div class="form-group">
        <input type="password" class="form-control" value="" placeholder="Konfirmasi Password" />
      </div>
      <div class="text-center">
        <button type="submit" class="btn btn-primary">Register</button>
      </div>
      <div class="text-center">
        <p>Anda sudah punya akun? <a href="#">Login</a></p>
      </div>
    </div>
  </div>
</div>

```

Gambar 4.4 Script Tampilan Register

IV.1.2 Tampilan Homepage

Homepage merupakan halaman awal atau halaman muka dari suatu website atau situs web, dimana pada alamat tersebut bisa mengakses dokumen atau konten website. Pengertian lainnya yaitu sebuah halaman default yang telah di setting untuk browser untuk ditayangkan saat mengakses melalui www (*World Wide Web*).



Gambar 4.5 Tampilan Homepage

Gambar 4.5 menunjukkan tampilan yang menjadi gerbang awal untuk mengakses *beranda website* dan juga menjadi tempat bagi *user* bisa memulai mencari ustadz. Berikut merupakan *script* dari gambar 4.5:

```

<headers>
<nav class="navbar navbar-expand-lg navbar-light fixed-top bg-light">
<div class="container">
<a class="navbar-brand" href="index.html">

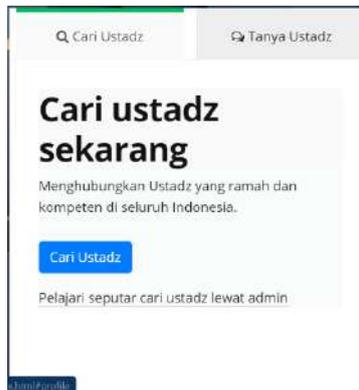
<button class="navbar-toggler navbar-toggler-right collapsed" type="button" data-toggle="collapse"
data-target="#navbarCollapse">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse justify-content-end" id="navbarCollapse">
<ul class="navbar-nav">
<li class="nav-item">
<a class="nav-link" href="index.html">Home</a>
</li>
<li class="nav-item">
<a class="nav-link" href="tentang-kami.html">Tentang Kami</a>
</li>
<li class="nav-item">
<a class="nav-link" href="kontak.html">Kontak</a>
</li>
</ul>
</div>
</div>
</nav>
</headers>

```

Gambar 4.6 Script Homepage

a. Menu Cari Ustadz

Menu cari ustadz merupakan isi dari halaman *home*, menu cari ustadz juga ialah *menu* sebelum membuka tampilan *google maps*.



Gambar 4.7 Menu Cari Ustadz

Gambar 4.7 menunjukkan tampilan yang akan digunakan oleh *User* untuk mengakses *Google Maps*. Berikut merupakan *script* dari gambar 4.7 :

```

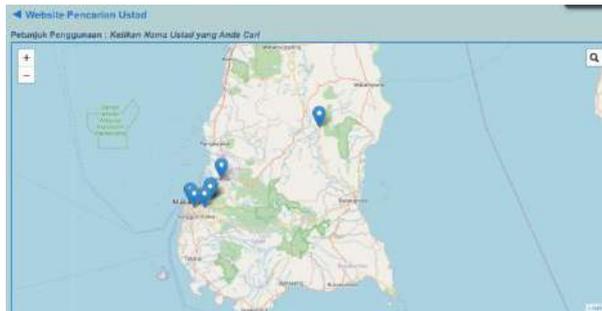
<div class="tab-content" id="myTabContent">
<div class="tab-pane fade active show" id="home" role="tabpanel" aria-labelledby="home-tab">
<h2 class="title-tab">Cari ustadz yuk sekarang</h2>
<h1 class="sub-title-tab">Menghubungkan Ustadz yang ramah dan kompeten di daerah Makassar.</h1>
<form class="mainform" method="get" action="/Rumah Ustad/examples/Ustad.html">
<div class="formgroup fg-1">
<div class="inner">
<input type="submit" class="btn btn-primary mb-3" value="Cari Ustadz">
<br>
</div>
</div>
</form>
</div>

```

Gambar 4.8 Script Menu Cari Ustadz

b. Tampilan *Google Maps*

Tampilan *google maps* merupakan tampilan yang menampilkan lokasi rumah ustadz yang dimana akan sangat mempermudah *user* untuk memilih ustadz terdekat.



Gambar 4.9 Tampilan *Google Maps*

Gambar 4.9 menunjukkan tampilan yang akan digunakan oleh *User* untuk memulai mencari ustadz. Berikut merupakan *script* dari gambar 4.9 :

```
<body>
<h3><a href= "../index.html" > <big></big> Website Pencarian Ustad</a></h3>

<h4>Petunjuk Penggunaan : <em>Ketikkan Nama Ustad yang Anda Cari</em></h4>
<div id="map"></div>

<script src="https://unpkg.com/jquery@3.3.1/dist/jquery.js"></script>
<script src="https://unpkg.com/leaflet@1.3.0/dist/leaflet.js"></script>
<script src="./src/leaflet-search.js"></script>
<script>

var data = [
  {"loc":[-5.129985, 119.519647], "title":"Dr. Hendra Kholid, MA"},
  {"loc":[-5.119315, 119.524519], "title":"Himan Muzzahin, SQ, MA"},
  {"loc":[-5.165862, 119.493978], "title":"Moh ali Habib, S.H."},
  {"loc":[-5.119315, 119.524519], "title":"Fathuliah S.Ag"},
  {"loc":[-5.165839, 119.441533], "title":"Dedi Jufri S.Ag"},
  {"loc":[-5.819228, 119.576572], "title":"Drs.H.Abd Rahman Sakka,Lc.,M.A"}
];
```

Gambar 4.10 *Script* Tampilan *Google Maps*

c. *Menu Tanya Ustadz*

Menu tanya ustadz merupakan menu yang membuat *user* bisa membuat pertanyaan seputar islam dan salah satu ustadz akan menjawab pertanyaan *user*.



Gambar 4.11 Menu Tanya Ustadz

Gambar 4.11 menunjukkan tampilan yang akan digunakan oleh *User* untuk memulai bertanya ke ustadz. Berikut merupakan *script* dari gambar 4.11 :

```

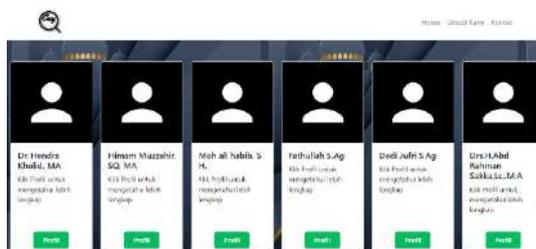
<div class="tab-pane fade" id="profile" role="tabpanel" aria-labelledby="profile-tab">
  <h2 class="title-tab">Punya Pertanyaan seputar agama Islam</h2>
  <h3 class="sub-title-tab">CUY akan membantu menjawab pertanyaan anda</h3>
  <p><strong style="color: black;">Klik "Mulai tanya", ikuti petunjuk widget "Tanya Ustadz" di bawah kanan layar.</strong></p>
  <form class="mainForm" method="get" action="/ustadz_kami.html">
    <div class="formgroup fg-1">
      <div class="inner">
        <input type="submit" class="btn btn-primary mb-3" value="Mulai tanya">
      </div>
    </div>
  </form>
</div>

```

Gambar 4.12 Script Menu Tanya Ustad

IV.1.3 Tampilan Menu Ustad Kami

Tampilan halaman ustad kami merupakan halaman yang menampilkan daftar ustad beserta informasi mengenai ustad. Halaman ustad kami juga memiliki fitur artikel dan halaman komentar.



Gambar 4.13 Menu Ustad Kami

Gambar 4.13 menunjukkan tampilan daftar ustadz dan *user* bisa



Gambar 4.19 Tampilan Halaman Kontak

Gambar 4.19 menunjukkan tampilan tentang kontak admin, ustadz juga bisa melakukan pendaftaran agar informasi mengenai dirinya bisa terinput ke dalam *website* dan *user* juga dapat memberikan usulan mengenai ustadz. Berikut merupakan *script* dari gambar 4.19 :

```

<section class="kontak">
  <div class="row post justify-content-center">
    <div class="col-lg-4 offset-lg-1">
      <div class="inner">
        <h1 class="title-page">Kontak</h1>
        <div class="content">
          <p><b>Kontak Kami</b></p>
          <p>Jl. Rongga Tangaya Dalam 5 Blok 5</p>
          <p>Telp. +6281 242 383 583</p>
          <p>email: rizalmandaleputra2@gmail.com</p>
          <a href="/php/form_pendaftaran.php" class="btn btn-success btn-register-ustadz">Pendaftaran Ustadz</a>
          <a href="/php/usulan_ustadz.php" class="btn btn-success btn-register-ustadz">Usulan Ustadz</a>
        </div>
      </div>
    </div>
  </div>
</section>

```

Gambar 4.20 Script Tampilan Halaman Kontak

a. Tampilan Form Pendaftaran

Tampilan form pendaftaran merupakan salah satu fitur dari halaman kontak yang dimana ustadz bisa melakukan pendaftaran.



Gambar 4.21 Tampilan Form Pendaftaran

Gambar 4.21 menunjukkan tampilan yang akan digunakan



Gambar 4.28 Tampilan Database Usulan Ustad

```

<?php
if (isset($_POST['kirim'])) {
$query=mysql_query("INSERT INTO uustad('nama_ustadz',
'no_hp_ustadz', 'nama', 'no_hp', 'email')
VALUES('".$_POST['nama_ustadz']."'.'".$_POST['no_hp_ustadz']."'.'.'.
$_POST['acara_ustadz']."'.'".$_POST['nama']."'.'".$_POST['email']'.
'')");
if ($query) {
}else{
mysql_error();
}
}
}
?>

```

Gambar 4.29 Script Database Form Usulan Ustad

IV.2 Pembahasan

Berdasarkan hasil yang telah didapatkan dalam Pencarian Ustadz Untuk Kota Makassar Menggunakan *Algoritma Haversine Formula* Berbasis *Website*, maka sistem ini telah dibuat sesuai dengan perancangan dan pengujian yang telah dilakukan sebelumnya.

Pengujian yang dilakukan berupa pengujian yang bertujuan untuk mengetahui hasil dari perancangan dan implementasi Pencarian Ustadz Untuk Kota Makassar Menggunakan *Algoritma Haversine Formula* Berbasis *Website*. Pengujian dilakukan dengan metode *blackbox testing* dan *whitebox testing* yang bertujuan untuk mengetahui apakah fungsi-fungsi pada sistem yang dibuat sudah berjalan sebagaimana mestinya.

IV.2.1 Penerapan Algoritma Haversine

Untuk mengetahui bahwa rumus *Haversine* dapat menghitung jarak antara dua titik dipermukaan bumi maka perlu di lakukan uji coba berikut

ini merupakan penjabaran rumus *Haversine* di dalam *query MySQL* :

$$\begin{aligned}R &= 6371 \text{ (km)} \\ \Delta\text{lat} &= \text{latitude tujuan} \\ \Delta\text{long} &= \text{longitude tujuan} \\ \Delta\text{lat user} &= \text{latitude user} \\ \Delta\text{long user} &= \text{longitude user} \\ 1 \text{ derajat} &= 0.0174532925 \text{ radian}\end{aligned}$$

Dari penjabaran diatas maka didapatkan rumus *query MySQL* di bawah ini :

$$\begin{aligned}&(\mathbf{R * ACOS (SIN (RADIANS (\Delta\text{lat})) * SIN (RADIANS (\Delta\text{lat user}))} \\ &+ \mathbf{COS (RADIANS (\Delta\text{long} - \Delta\text{long user})) * COS (RADIANS (\Delta\text{lat})) *} \\ &\mathbf{COS (RADIANS (\Delta\text{lat user})) })\end{aligned}$$

Perhitungan :

Lokasi user Jl. Prof. Abdurahman Basalamah No.101, Karampuang, Kec. Panakkukang, Kota Makassar, Sulawesi Selatan 90231

$$\begin{aligned}\text{lat1} &= -5.149241 \\ \text{long1} &= 119.450064\end{aligned}$$

A. Perhitungan 1

Jl. Keberkahan I, Paccerakkang, Kec. Biringkanaya, Kota Makassar, Sulawesi Selatan 90245

$$\begin{aligned}\text{lat2} &= -5.129905 \\ \text{long2} &= 119.519647\end{aligned}$$

$$\begin{aligned}\Delta\text{lat} &= \frac{\pi}{180} * (\text{latitude tujuan} - \text{latitude user}) \\ &= \frac{3.14}{180} * (-5.129905 - (-5.149241)) \\ &= -0,000337477\end{aligned}$$

$$\Delta\text{long} = \frac{\pi}{180} * (\text{longitude tujuan} - \text{longitude user})$$

$$= \frac{3.14}{180} * (119.519647 - 119.450064)$$

$$= -0,001214452$$

$$a = \sin\left(\frac{\Delta\text{lat}}{2}\right)^2 = \sin\left(\frac{-0,000337477}{2}\right)^2$$

$$= 8,67327\text{E}-12$$

$$c = \cos(\text{lat1}) * \cos(\text{lat2}) * \sin\left(\frac{\Delta\text{long}}{2}\right)$$

$$= \cos(-5.149241) * \cos(-5.129905) *$$

$$\sin\left(\frac{-0,001214452}{2}\right)^2$$

$$= 1,11418\text{E}-10$$

$$d = R * 2 * \text{asin}(\sqrt{a + c})$$

$$= 6371 * 2 *$$

$$\text{asin}(\sqrt{8,67327\text{E} - 12 + 1,11418\text{E} - 10})$$

$$= 6371 * 2 * \text{asin}(\sqrt{1,09586\text{E} - 05})$$

$$= 0,139634857$$

B. Perhitungan 2

Paccerrakkang, Kec. Biringkanaya, Kota Makassar,

Sulawesi Selatan 90562

$$\text{lat2} = -5.119315$$

$$\text{long2} = 119.524519$$

$$\Delta\text{lat} = \frac{\pi}{180} * (\text{latitude tujuan} - \text{latitude user})$$

$$= \frac{3.14}{180} * (-5.119315 - (-5.149241))$$

$$= -0,000522307$$

$$\Delta\text{long} = \frac{\pi}{180} * (\text{longitude tujuan} - \text{longitude user})$$

$$= \frac{3.14}{180} * (119.524519 - 119.450064)$$

$$= -0,001299485$$

$$a = \sin\left(\frac{\Delta\text{lat}}{2}\right)^2 = \sin\left(\frac{-0,000522307}{2}\right)^2$$

$$= 2,07753\text{E}-11$$

$$c = \cos(\text{lat1}) * \cos(\text{lat2}) * \sin\left(\frac{\Delta\text{long}}{2}\right)$$

$$= \cos(-5.149241) * \cos((-5.119315) * \sin\left(\frac{-0,001299485}{2}\right)^2$$

$$\sin\left(\frac{-0,001299485}{2}\right)^2$$

$$= 1,27569\text{E}-10$$

$$d = R * 2 * \text{asin}(\sqrt{a + c})$$

$$= 6371 * 2 * \text{asin}(\sqrt{2,07753\text{E} - 11 + 1,27569\text{E} - 10})$$

$$\text{asin}(\sqrt{2,07753\text{E} - 11 + 1,27569\text{E} - 10})$$

$$= 6371 * 2 * \text{asin}(\sqrt{1,48344\text{E} - 10})$$

$$= 0,155193309$$

C. Perhitungan 3

Jl. Nipa-nipa Dalam 4 No.26, RT.001/RW.01, Manggala,

Kec. Manggala, Kota Makassar, Sulawesi Selatan 90234

$$\text{lat2} = -5.165862$$

$$\text{long2} = 119.493971$$

$$\Delta\text{lat} = \frac{\pi}{180} * (\text{latitude tujuan} - \text{latitude user})$$

$$= \frac{3.14}{180} * (-5.165862 - (-5.149241))$$

$$= 0,000290091$$

$$\Delta\text{long} = \frac{\pi}{180} * (\text{longitude tujuan} - \text{longitude user})$$

$$= \frac{3.14}{180} * (119.493971 - 119.450064)$$

$$= -0,000766304$$

$$a = \sin\left(\frac{\Delta\text{lat}}{2}\right)^2 = \sin\left(\frac{0,000290091}{2}\right)^2$$

$$= 6,40861\text{E}-12$$

$$c = \cos(\text{lat1}) * \cos(\text{lat2}) * \sin\left(\frac{\Delta\text{long}}{2}\right)$$

$$= \cos(-5.149241) * \cos((-5.165862) *$$

$$\sin\left(\frac{-0,000766304}{2}\right)^2$$

$$= 4,43581\text{E}-11$$

$$d = R * 2 * \text{asin}(\sqrt{a + c})$$

$$= 6371 * 2 *$$

$$\text{asin}\left(\sqrt{6,40861\text{E} - 12 + 4,43581\text{E} - 11}\right)$$

$$= 6371 * 2 * \text{asin}\left(\sqrt{5,07668\text{E} - 11}\right)$$

$$= 0,090787761$$

D. Perhitungan 4

Sudiang, Kec. Biringkanaya, Kota Makassar, Sulawesi

Selatan 90552

$$\text{lat2} = -5.093618$$

$$\text{long2} = 119.533421$$

$$\begin{aligned}\Delta\text{lat} &= \frac{\pi}{180} * (\text{latitude tujuan} - \text{latitude user}) \\ &= \frac{3.14}{180} * (-5.093618 - (-5.149241)) \\ &= -0,000970804\end{aligned}$$

$$\begin{aligned}\Delta\text{long} &= \frac{\pi}{180} * (\text{longitudo tujuan} - \text{longitudo user}) \\ &= \frac{3.14}{180} * (119.533421 - 119.450064) \\ &= -0,001454854\end{aligned}$$

$$\begin{aligned}a &= \sin\left(\frac{\Delta\text{lat}}{2}\right)^2 = \sin\left(\frac{-0,000970804}{2}\right)^2 \\ &= 7,17725\text{E}-11\end{aligned}$$

$$\begin{aligned}c &= \cos(\text{lat1}) * \cos(\text{lat2}) * \sin\left(\frac{\Delta\text{long}}{2}\right) \\ &= \cos(-5.149241) * \cos((-5.093618)) * \\ &\quad \sin\left(\frac{-0,001454854}{2}\right)^2 \\ &= 1,59904\text{E}-10\end{aligned}$$

$$\begin{aligned}d &= R * 2 * \text{asin}(\sqrt{a + c}) \\ &= 6371 * 2 * \\ &\quad \text{asin}(\sqrt{7,17725\text{E} - 11 + 1,59904\text{E} - 10}) \\ &= 6371 * 2 * \text{asin}(\sqrt{2,31676\text{E} - 10}) \\ &= 0,193944961\end{aligned}$$

E. Perhitungan 5

Tidung, Kec. Rappocini, Kota Makassar, Sulawesi Selatan

90222

$$\text{lat2} = -5.165039$$

$$\text{long2} = 119.441533$$

$$\begin{aligned}\Delta\text{lat} &= \frac{\pi}{180} * (\text{latitude tujuan} - \text{latitude user}) \\ &= \frac{3.14}{180} * (-5.165039 - (-5.149241)) \\ &= 0,000275727\end{aligned}$$

$$\begin{aligned}\Delta\text{long} &= \frac{\pi}{180} * (\text{longitude tujuan} - \text{longitude user}) \\ &= \frac{3.14}{180} * (119.441533 - 119.450064) \\ &= 0,000148894\end{aligned}$$

$$\begin{aligned}a &= \sin\left(\frac{\Delta\text{lat}}{2}\right)^2 = \sin\left(\frac{0,000275727}{2}\right)^2 \\ &= 5,78967\text{E-}12\end{aligned}$$

$$\begin{aligned}c &= \cos(\text{lat1}) * \cos(\text{lat2}) * \sin\left(\frac{\Delta\text{long}}{2}\right) \\ &= \cos(-5.149241) * \cos((-5.165039)) * \\ &\quad \sin\left(\frac{0,000148894}{2}\right)^2 \\ &= 1,67466\text{E-}12\end{aligned}$$

$$\begin{aligned}d &= R * 2 * \text{asin}\left(\sqrt{a + c}\right) \\ &= 6371 * 2 * \\ &\quad \text{asin}\left(\sqrt{5,78967\text{E-}12 + 1,67466\text{E-}12}\right) \\ &= 6371 * 2 * \text{asin}\left(\sqrt{7,46433\text{E-}12}\right) \\ &= 0,034812316\end{aligned}$$

Tabel 4.1 Perhitungan Jarak

No.	Titik User			Titik Tujuan			Hasil Jarak dari	
	Nama	Latitude	Longitude	Nama	Latitude	Longitude	Haversine	Google Maps
1	Jl. Prof. Abdurahman Basalamah No.101, Karampung, Kec. Panakkukang, Kota Makassar, Sulawesi Selatan 90231	-5,14924	119,450064	Jl. Keberkahan I, Paccera-kkang, Kec. Biringkanaya	-5,129905	119,519647	13,9 KM	12,8 KM
2				Paccera-kkang, Kec. Biringkanaya	-5,119315	119,524519	15,5 KM	10,4 KM
3				Jl. Nipani-pa Dalam 4 No.26, RT.001/RW.01, Manggala, Kec. Manggala Sudiang, Kec. Biringkanaya	-5,165862	119,493971	9,0 KM	6,0 KM
4					-5,093618	119,533421	19,3 KM	16,0 KM
5				Tidung, Kec. Rappocini	-5,165039	119,441533	3,4 KM	4,1 KM
Hasil jarak yang terdekat							3,4 KM	4,1 KM

IV.2.2 Blackbox Testing

Tabel 4.2 Instrumen Pengujian Fungsional Website

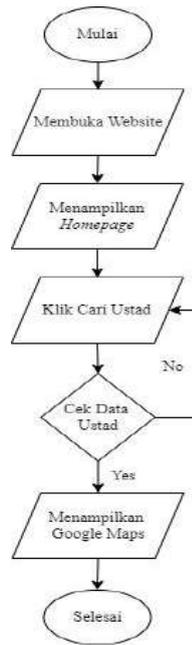
No.	Komponen yang diujikan	Skenario	Hasil yang diharapkan	Keterangan	
				Berfungsi	Tidak Berfungsi
1.	Membuka Website	Mengunjungi website	Website akan menampilkan <i>homepage</i>	√	
2.	Tampilan Cari Ustad	Klik homepage	Sistem akan merespon dan akan menampilkan cari ustad	√	
3.	Tampilan Tanya Ustad	Klik homepage	Sistem akan merespon dan akan menampilkan tanya ustad	√	

4.	Tampilan Google Maps	Klik tombol Cari Ustad	Sistem akan merespon dan akan menampilkan <i>google maps</i>	√	
5.	Menu Daftar Ustad	Klik menu ustad kami pada menu bar	Sistem akan merespon dan akan menampilkan daftar ustad pada <i>menu</i> ustad kami	√	
6.	Menu Kontak	Klik menu kontak pada menu bar	Sistem akan merespon dan akan menampilkan kontak pada <i>menu</i> kontak	√	
7.	Form Pendaftaran Ustad	Klik menu kontak pada menu bar	Sistem akan merespon dan akan menampilkan kontak lalu anda klik tombol pendaftaran ustad pada tampilan <i>menu</i> kontak	√	
8.	Form Usulan Ustad	Klik menu kontak pada menu bar	Sistem akan merespon dan akan menampilkan kontak lalu anda klik tombol usulan ustad pada tampilan <i>menu</i> kontak	√	

IV.2.3 Whitebox Testing

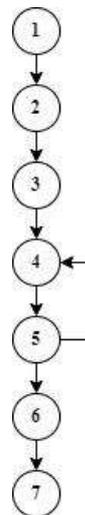
1. Cari Ustad

Berfungsi untuk mencari ustad, di sini ada 1 klarifikasi hak akses untuk mengakses website yaitu *user*.



Gambar 4.30 Diagram *Flowchart White Box Testing Cari*

Ustad



Gambar 4.31 Diagram *Flowgraph White Box Testing Cari*

Ustad

Flowgraph Cari Ustad kompleksitas siklomatis

dihitung menggunakan 3 (tiga) cara, yaitu :

1. Grafik alir mempunyai 2 region

2. $V(G) = 7 \text{ edge} - 7 \text{ node} + 2 = 2$

3. $V(G) = 1 \text{ simpul yang diperkirakan} + 1 = 2$

Dengan demikian kompleksitas siklomatis dari flowgraph yang dijelaskan pada Gambar 4.31 adalah 1.

Dengan jalur independennya adalah :

Jalur 1 : 1-2-3-4-5-6-7

Jalur 2 : 1-2-3-4-5-4-5-6-7

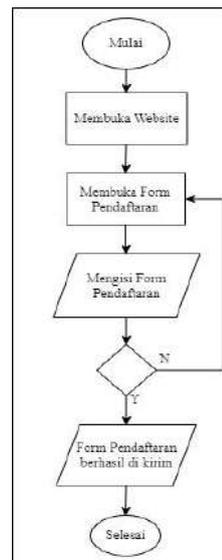
Tabel 4.3 Test Case Cari Ustad

Path	1
Jalur	1-2-3-4-5-6-7
Skenario	<ul style="list-style-type: none"> - Mulai - Membuka Website - Menampilkan <i>Homepage</i> - Klik Cari Ustad - Cek Data Ustad - Menampilkan Google Maps - Selesai
Hasil Pengujian	Berhasil
Path	2
Jalur	1-2-3-4-5-4-5-6-7
Skenario	<ul style="list-style-type: none"> - Mulai - Membuka Website - Menampilkan <i>Homepage</i> - Klik Cari Ustad - Cek Data Ustad - Klik Cari Ustad - Cek Data Ustad - Menampilkan Google Maps

	Selesai
Hasil Pengujian	Berhasil

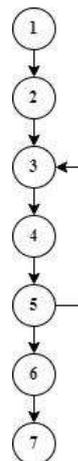
2. Form Pendaftaran Ustad

Berfungsi untuk mencari ustad, di sini ada 1 klarifikasi hak akses untuk mengakses website yaitu ustad.



Gambar 4.32 Diagram *Flowchart White Box Testing Form*

Pendaftaran Ustad



Gambar 4.33 Diagram *Flowgraph White Box Testing Form*

Pendaftaran Ustad

Flowgraph Cari Ustad kompleksitas siklomatis

dihitung menggunakan 3 (tiga) cara, yaitu :

4. Grafik alir mempunyai 2 region
5. $V(G) = 7 \text{ edge} - 7 \text{ node} + 2 = 2$
6. $V(G) = 1 \text{ simpul yang diperkirakan} + 1 = 2$

Dengan demikian kompleksitas siklomatis dari flowgraph yang dijelaskan pada Gambar 4.33 adalah 1.

Dengan jalur independennya adalah :

Jalur 1 : 1-2-3-4-5-6-7

Jalur 2 : 1-2-3-4-5-3-4-5-6-7

Tabel 4.4 *Test Case* Form Pendaftaran

Path	1
Jalur	1-2-3-4-5-6-7
Skenario	<ul style="list-style-type: none">- Mulai- Membuka Website- Menampilkan Form Pendaftaran- Mengisi Form Pendaftaran- Proses- Form Pendaftaran berhasil dikirim- Selesai
Hasil Pengujian	Berhasil
Path	2
Jalur	1-2-3-4-5-3-4-5-6-7
Skenario	<ul style="list-style-type: none">- Mulai- Membuka Website- Menampilkan Form Pendaftaran- Mengisi Form Pendaftaran

	<ul style="list-style-type: none"> - Proses - Menampilkan Form Pendaftaran - Mengisi Form Pendaftaran - Proses - Form Pendaftaran berhasil dikirim - Selesai
Hasil Pengujian	Berhasil

IV.2.5 Pengukuran Jarak dan Selisih

Berikut ini salah satu contoh perhitungan dari rumus untuk menghitung selisih dan *percent error* dari perhitungan *haversine* dan *google maps* secara matematis.

Diketahui :

a. Perhitungan 1

Jarak Google Maps = 12,8 Km

Jarak Haversine = 13,9 Km

Selisih = Jarak Haversine - Jarak Google Maps

= 13,9 Km – 12,8 Km

= 1,1 Km

ABS(Error) = $\frac{|jarak\ Haversine - jarak\ google\ maps|}{jarak\ haversine} * 100\%$

= (13,9 – 12,8) / 13,9 * 100%

= 7,91%

b. Perhitungan 2

Jarak Google Maps = 10,4 Km

Jarak Haversine = 15,5 Km

Selisih = Jarak Haversine - Jarak Google Maps

= 15,5 Km – 10,4 Km

= 5,1 Km

$$\begin{aligned} \text{ABS(Error)} &= \frac{|\text{jarak Haversine} - \text{jarak google maps}|}{\text{jarak haversine}} * 100\% \\ &= (15,5 - 10,4) / 15,5 * 100\% \\ &= 32,90\% \end{aligned}$$

c. Perhitungan 3

$$\text{Jarak Google Maps} = 6,0 \text{ Km}$$

$$\text{Jarak Haversine} = 9,0 \text{ Km}$$

$$\begin{aligned} \text{Selisih} &= \text{Jarak Haversine} - \text{Jarak Google Maps} \\ &= 9,0 \text{ Km} - 6,0 \text{ Km} \\ &= 3,0 \text{ Km} \end{aligned}$$

$$\begin{aligned} \text{ABS(Error)} &= \frac{|\text{jarak Haversine} - \text{jarak google maps}|}{\text{jarak haversine}} * 100\% \\ &= (9,0 - 6,0) / 9,0 * 100\% \\ &= 33,33\% \end{aligned}$$

d. Perhitungan 4

$$\text{Jarak Google Maps} = 16,0 \text{ Km}$$

$$\text{Jarak Haversine} = 19,3 \text{ Km}$$

$$\begin{aligned} \text{Selisih} &= \text{Jarak Haversine} - \text{Jarak Google Maps} \\ &= 19,3 \text{ Km} - 16,0 \text{ Km} \\ &= 3,3 \text{ Km} \end{aligned}$$

$$\begin{aligned} \text{ABS(Error)} &= \frac{|\text{jarak Haversine} - \text{jarak google maps}|}{\text{jarak haversine}} * 100\% \\ &= (19,3 - 16,0) / 19,3 * 100\% \\ &= 17,09\% \end{aligned}$$

e. Perhitungan 5

$$\text{Jarak Google Maps} = 4,1 \text{ Km}$$

$$\text{Jarak Haversine} = 3,4 \text{ Km}$$

$$\begin{aligned} \text{Selisih} &= \text{Jarak Haversine} - \text{Jarak Google Maps} \\ &= 3,4 \text{ Km} - 4,1 \text{ Km} \\ &= 1,3 \text{ Km} \end{aligned}$$

$$\text{ABS(Error)} = \frac{|\text{jarak Haversine} - \text{jarak google maps}|}{\text{jarak haversine}} * 100\%$$

$$= (3,4 - 4,1) / 3,4 * 100\%$$

Tabel 4.5 Perhitungan Selisih dan *Percent Error*

No.	Titik User			Titik Tujuan			Hasil Jarak dari		Selisih
	Nama	Latitude	Longitude	Nama	Latitude	Longitude	Haversine	Google Maps	
1	Jl. Prof. Abdurahman Basalamah No.101, Karampuang, Kec. Panakkukang, Kota Makassar, Sulawesi Selatan 90231	-5,14924	119,450064	Jl. Keberkahan I, Pacerakka ng, Kec. Biringkanaya	-5,129905	119,519647	13,9 KM	12,8 KM	1,1 KM
2				Pacerakka ng, Kec. Biringkanaya	-5,119315	119,524519	15,5 KM	10,4 KM	5,1 KM
3				Jl. Nip-nipa Dalam 4 No.26, RT.001/R W.01, Manggala, Kec. Manggala	-5,165862	119,493971	9,0 KM	6,0 KM	3,0 KM
4				Sudiang, Kec. Biringkanaya	-5,093618	119,533421	19,3 KM	16,0 KM	3,3 KM
5				Tidung, Kec. Rappocini	-5,165039	119,441533	3,4 KM	4,1 KM	1,3 KM

Dari hasil *Percent Error* perhitungan *haversine formula* dapat disimpulkan memiliki tingkat *error* yang cukup tinggi. Hal tersebut dikarenakan perhitungan *haversine formula* mengambil garis lurus tanpa mengikuti jalur rute jalan terpendek.

BAB V

PENUTUP

V.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dapat disimpulkan bahwa :

1. Sistem pencarian ustad ini dibuat berdasarkan kebutuhan masyarakat untuk mempermudah mencari penceramah/ustad. Sistem ini digunakan dua arah langsung oleh ustad dan yang mencari ustad. Sistem ini berjalan di sistem operasi berbasis website.
2. Website ini menggunakan metode pengujian *blackbox testing*. Berdasarkan pengujian fungsional dari metode *blackbox* yang beberapa kali melalui pengujian, bahwa fungsional website telah berfungsi dengan baik sesuai dengan rancangan. Berdasarkan hasil analisis pengujian efektifitas oleh pengguna, website ini telah efektif sesuai dengan yang diharapkan.

V.2 Saran

Website untuk melakukan pencarian ustadz ini masih jauh kesempurnaan. Oleh karena itu, perlu adanya pengembangan lebih lanjut yang mampu memperbaiki segala kekurangan yang ada pada sistem yang telah dibuat. Adapun saran-saran yang dapat diberikan, sebagai berikut :

1. Sebaiknya website ini bisa di onlinekan.
2. Sebaiknya selain membuat website dapat dikembangkan dengan cara membuat website yang akan membuat user dengan mudah menemukannya.

DAFTAR PUSTAKA

- Anugerah, G. R. (2020). *Website Gauslab berbasis CodeIgniter dan Flutter dengan Metode Waterfall* (Doctoral dissertation, universitas islam indonesia).
- Atmodjo, G. E., & Krisjanti, Adelpia, M. N. (2016). Preferensi Konsumen Terhadap Merek Smartphone Berdasarkan Sistem Operasi. Fakultas Ekonomi, Universitas Admajaya.
- Andarsyah, R., & Siherli, R. A. G. (2020). ANALISIS DAN IMPLEMENTASI WEBSITE PEMBAYARAN INSTAN MENGGUNAKAN VIRTUAL ACCOUNT BERBASIS ANDROID PADA PT. POS INDONESIA (PERSERO). *Jurnal Teknik Informatika*, 12(1), 15-21.
- Anugerah, G. R. (2020). *Website Gauslab berbasis CodeIgniter dan Flutter dengan Metode Waterfall* (Doctoral dissertation, universitas islam indonesia).
- Arifin, A., & Limbong, B. (2016). Sistem Informasi Perhitungan Suku Cadang (sparepart) Dalam Satu Mesin Produksi.” *JURNAL SISFOTEK GLOBAL*, 6(2).
- Abdurahman, M. (2017). Sistem Informasi Pengolahan Data Pembelian dan Penjualan pada Toko Koloncucu Ternate. *IJIS-Indonesian Journal On Information System*, 2(1).
- Farida, U, Tarmizi M. Jakfar. (2015). Diskursus sunnah sebagai sumber hukum Islam: perspektif Ushuliyin dan Muhadditsin. *Yudisia*, 6(1), 237-255.
- Ibrahim, K. A. B., & Dian, G. (2021). RANCANG BANGUN WEBSITE BERBASIS ANDROID UNTUK BRAND CLOTHING SAND BEACH DENGAN SKEMA DISKON MENGGUNAKAN HUNGARIAN ALGORITHM. *JSI (Jurnal sistem Informasi) Universitas Suryadarma*, 8(1), 47-56.
- Naim, N. (2016). Menjadi guru inspiratif: memberdayakan dan mengubah jalan hidup siswa.
- Nurafifah, I. M. Analisis Tingkat Kompetensi Dan Efisiensi Bank Umum Syariah Di Indonesia (Bachelor's thesis, Fakultas Ekonomi dan Bisnis uin jakarta).
- Permana, Angga Aditya, Sutabri T. 2017. “Rancangan Sistem Informasi Simpan Pinjam Pada Koperasi Guru Dan Pegawai Smp Negeri 45 Jakarta.” *JIKA (Jurnal Informatika)* 1(2): 79–87.

- Rasyid, S. Upaya meningkatkan hasil belajar baca tulis al-qur'an (BTQ) pada mata pelajaran Pendidikan Agama Islam & Budi Pekerti melalui metode peerlession di kelas IV SDN Harapan Baru II Kota Bekasi (Bachelor's thesis, Fakultas Ilmu Tarbiyah dan Keguruan (FITK) UIN Syarif Hidayatullah Jakarta).
- Rofiuddin, A. A, Ruswa Darsono. (2016). Penentuan Hari Dalam Sistem Kalender Hijriah. *Al-Ahkam*, 26(1), 117-136.
- Radliya, Verdi Yasin, Nizar Rabbi, and Erick Hermawan. 2015. "Website Pendukung Kinerja Karyawan Di Mitra Dinamika Konsultan." *Jurnal Manajemen Informatika (JAMIKA)* 5(2): 81–95.
- Syahputra, Fathansyah, Hakim Firman. 2019. "Website Penggunaan Nomor Antrian Di Tempat Pelayanan Terpadu Seksi Pelayanan Pada Kantor Pelayanan Pajak Pratama Palembang Seberang Ulu Berbasis Website". Phd Thesis. Politeknik Negeri Sriwijaya.
- Susheri, S, Muhyidin Khazin. (2012). *Analisis rumus trigonometri dalampenentuan arah kiblat* (Doctoral dissertation, IAIN Walisongo).
- Safaat, N, Akhmad Dharma Kasman. (2012). Android; Pemrograman Website Mobile Smartphone dan Tablet PC Berbasis Android.
- Utami, N. (2020). WEBSITE MANAJEMEN PEMESANAN DAN PENJUALAN BAJA INDUSTRI PADA PT. TIRA AUSTENITE TBK PALEMBANG MENGGUNAKAN METODE UP SELLING-CROSS SELLING (Doctoral dissertation, POLITEKNIK NEGERI SRIWIJAYA).
- Zhafarina, A. SISTEM PENDUKUNG KEPUTUSAN SELEKSI GURU PRIVAT BERBASIS WEB MENGGUNAKAN ALGORITMA SIMPLE ADDITIVE WEIGHTING.