

TUGAS AKHIR - IF184802

PENERAPAN UNIVERSAL SENTENCE ENCODER (USE) UNTUK LOCATION-AWARE CHATBOT PADA APLIKASI LAYANAN WISATA KEBUN BINATANG

VANIA CIKANINDI
05111740000193

Dosen Pembimbing I
Dr. Tech. Ir. R.V. Hari Ginardi, M.Sc.

Dosen Pembimbing II
Ir. Adhatus Solichah Ahmadiyah, S.Kom, M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya
2021

TUGAS AKHIR - IF184802

PENERAPAN UNIVERSAL SENTENCE ENCODER (USE) UNTUK LOCATION-AWARE CHATBOT PADA APLIKASI LAYANAN WISATA KEBUN BINATANG

VANIA CIKANINDI
05111740000193

Dosen Pembimbing I
Dr. Tech. Ir. R.V. Hari Ginardi, M.Sc.

Dosen Pembimbing II
Ir. Adhatus Solichah Ahmadiyah, S.Kom, M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya
2021



TUGAS AKHIR - IF184802

PENERAPAN UNIVERSAL SENTENCE ENCODER (USE) UNTUK LOCATION-AWARE CHATBOT PADA APLIKASI LAYANAN WISATA KEBUN BINATANG

VANIA CIKANINDI
05111740000193

Dosen Pembimbing I
Dr. Tech. Ir. R.V. Hari Ginardi, M.Sc.

Dosen Pembimbing II
Ir. Adhatus Solichah Ahmadiyah, S.Kom, M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya
2021

[Halaman ini sengaja dikosongkan]



UNDERGRADUATE THESIS - IF184802

IMPLEMENTATION OF UNIVERSAL SENTENCE ENCODER (USE) FOR LOCATION-AWARE CHATBOT IN ZOO TOURISM SERVICES APPLICATION

VANIA CIKANINDI
05111740000193

Supervisor I
Dr. Tech. Ir. R.V. Hari Ginardi, M.Sc.

Supervisor II
Ir. Adhatus Solichah Ahmadiyah, S.Kom, M.Sc.

DEPARTMENT OF INFORMATICS
Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember
Surabaya
2021

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

PENERAPAN UNIVERSAL SENTENCE ENCODER (USE) UNTUK LOCATION-AWARE CHATBOT PADA APLIKASI LAYANAN WISATA KEBUN BINATANG

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Bidang Studi Manajemen Informasi Cerdas
Program Studi S-1 Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh:

Vania Cikanindi
NRP: 05111740000193

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Dr. Tech. Ir. R.V. Hari Ginardi, M.Sc.
NIP. 19650518 199203 1 003

Ir. Adhatus Solichah Ahmadiyah, S.Kom, M.Sc.
NIP. 19850826 201504 2 002



(Pembimbing 1)

(Pembimbing 2)

SURABAYA
JANUARI 2022

[Halaman ini sengaja dikosongkan]

PENERAPAN UNIVERSAL SENTENCE ENCODER (USE) UNTUK LOCATION-AWARE CHATBOT PADA APLIKASI LAYANAN WISATA KEBUN BINATANG

Nama Mahasiswa : Vania Cikanindi
NRP : 05111740000193
Departemen : Teknik Informatika, Fakultas Teknologi
Elektro dan Informatika Cerdas, ITS
Dosen Pembimbing 1 : Dr. Tech. Ir. R.V. Hari Ginardi, M.Sc.
Dosen Pembimbing 2 : Ir. Adhatus Solichah Ahmadiyah, S.Kom,
M.Sc.

Abstrak

Chatbot merupakan program yang mampu melakukan percakapan dengan penggunaanya baik melalui teks maupun suara sehingga menghasilkan informasi dan atau melakukan tugas tertentu. Kebun binatang merupakan salah satu objek wisata rekreasi edukasi yang terdampak pandemi Covid-19. Untuk menghindari penularan Covid-19, salah satu kebun binatang di Jawa Barat memberlakukan beberapa kebijakan, salah satunya adalah mengurangi jumlah pekerja. Dengan pengurangan pekerja tersebut, pelayanan pemberian informasi menjadi tidak optimal sehingga pengunjung kebun binatang tidak mampu mendapatkan pengalaman berwisata secara maksimal. Dalam penelitian ini, teknologi GPS dan Universal Sentence Encoder (USE) digunakan untuk membuat sebuah location-aware chatbot untuk aplikasi layanan wisata kebun binatang. Universal Sentence Encoder digunakan untuk membuat vektor embedding dari kalimat pertanyaan pengunjung dan data pada brainfile yang kemudian dicocokkan menggunakan perhitungan cosine similarity. Teknologi GPS dan geofencing digunakan untuk mengetahui lokasi pengunjung sehingga memungkinkan chatbot untuk memberikan informasi terkait koleksi kebun binatang serta mengirimkan informasi mengenai fasilitas terdekat. Aplikasi ini diuji menggunakan ponsel Android untuk mengetahui fungsionalitas,

kerelevanan jawaban, dan ketergunaan aplikasi. Hasil dari pengujian fungsionalitas sistem menunjukkan bahwa sistem dapat berjalan sesuai dengan rancangan sistem. Pengujian kerelevanan jawaban menunjukkan bahwa chatbot menghasilkan kerelevanan sebesar 78%, akurasi sebesar 80%, presisi 86.7%, serta recall 90.7%. Hasil dari pengujian System Usability Scale (SUS) menunjukkan bahwa aplikasi ini memiliki ketergunaan yang dapat diterima oleh pengguna.

Kata kunci: Chatbot, Geofencing, Location-Aware Applications, Universal Sentence Encode

IMPLEMENTATION OF UNIVERSAL SENTENCE ENCODER (USE) IN LOCATION-AWARE CHATBOT FOR ZOO TOURISM SERVICES APPLICATION

Student Name : Vania Cikanindi
Registration Number : 05111740000193
Department : Department of Informatics, Faculty of
Intelligent Electrical and Informatics
Technology, ITS
First Supervisor : Dr. Tech. Ir. R.V. Hari Ginardi, M.Sc.
Second Supervisor : Ir. Adhatus Solichah Ahmadiyah,
S.Kom, M.Sc.

Abstract

Chatbot is a program that able to do conversations with users either through text or voice to obtain information and or perform certain tasks. Zoo is one of the tourist recreational attractions that affected by the Covid-19 pandemic. To avoid the transmission of Covid-19, the zoo management has implemented several policies, one of which was to reduce the number of workers. With the reduction of workers, the service of providing information became not optimal so the zoo visitors are unable to get the full experience and the tourists' interest in visiting the zoo is reduced. In this study, GPS technology and Universal Sentence Encoder (USE) were used to create a location-based chatbot for the zoo tourism application. Universal Sentence Encoder were used to create embedding vectors from question sentence and data in the brainfile which is then matched by using cosine similarity calculations. GPS and geofencing technology are used to determine the location of the visitor so the chatbot could provides information related to the zoo collections and send information about nearby facilities. This application was tested using an Android phone to test the functionality, relevance of the answers,

and user ratings of the usability of the application. The results of the system functionality test indicate that the system could run according to the system design. The answer relevance test shows that the chatbot produces 78% relevance of answer, 80% of accuracy, 86.7% of precision, and 90.7% of recall. The results of System Usability Scale (SUS) show that this application is acceptable.

Keywords: Chatbot, Geofencing, Location-Aware Applications, Universal Sentence Encoder

KATA PENGANTAR

Puji syukur atas kehadiran Allah SWT karena atas rahmat dan ridho-Nya penulis dapat menyelesaikan tugas akhir dengan judul:

PENERAPAN *UNIVERSAL SENTENCE ENCODER* (USE) UNTUK *LOCATION-AWARE CHATBOT* PADA APLIKASI LAYANAN WISATA KEBUN BINATANG

Pengerjaan tugas akhir ini dapat terselesaikan dengan dukungan dari berbagai pihak. Penulis dalam kesempatan ini ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Allah SWT karena atas rahmat dan ridho-Nya penulis dapat mengerjakan tugas akhir dan menyelesaikannya.
2. Keluarga penulis yang senantiasa mendoakan, mendukung, dan menyemangati penulis dalam menyelesaikan tugas akhir ini.
3. Bapak Dr. Tech. Ir. R.V. Hari Ginardi, M.Sc. sebagai Dosen Pembimbing I dan Ibu Ir. Adhatus Solichah Ahmadiyah, S.Kom, M.Sc. sebagai Dosen Pembimbing II yang telah memberikan bimbingan, arahan, dan bantuan dalam pengerjaan tugas akhir yang dikerjakan ini.
4. Alifi, Shella, Miranda, dan Nitama yang telah menjadi teman dalam menempuh perkuliahan bersama-sama serta selalu memberikan semangat dan dukungan.
5. Teman-teman penulis lainnya yang telah membantu dan pengerjaan tugas akhir ini, yaitu Lia, Haura, Yuki, dan Andre.
6. Chiho yang selalu memberikan dukungan dan memotivasi penulis dalam mengerjakan tugas akhir.
7. Pihak lainnya yang tidak dapat saya sebutkan satu persatu.

Penulis juga menyadari bahwa masih terdapat kekurangan dalam pengerjaan dan pelaksanaan tugas akhir ini. Oleh karena itu, penulis menerima kritik dan saran melalui alamat surel penulis, yaitu vcikanindi@gmail.com.

Semoga tugas akhir ini dapat memberikan manfaat bagi pembaca.

Surabaya, Desember 2021

Vania Cikanindi

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
<i>Abstrak</i>	vii
<i>Abstract</i>	ix
KATA PENGANTAR	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xix
DAFTAR KODE SUMBER	xxi
DAFTAR PERSAMAAN.....	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Metodologi	4
1.6.1 Penyusunan Proposal Tugas Akhir	4
1.6.2 Pengumpulan Data	4
1.6.3 Analisis Kebutuhan.....	4
1.6.4 Tahap Desain	5
1.6.5 Implementasi Perangkat Lunak	5
1.6.6 Uji Coba dan Evaluasi	6
1.6.7 Penyusunan Buku Tugas Akhir	6
1.7 Sistematika Penulisan	7
BAB II DASAR TEORI.....	9
2.1 <i>Chatbot</i>	9
2.2 Wisata Kebun Binatang	9
2.3 <i>Location-Based Services</i> (LBS)	10
2.4 <i>Location-Aware Application</i> (LAA)	12
2.5 <i>Global Positioning System</i> (GPS)	13
2.6 Google Maps	13
2.7 <i>Geofencing</i>	14
2.8 <i>Universal Sentence Encoder</i> (USE).....	14

2.9	<i>Cosine Similarity</i>	16
2.10	Persamaan Haversine	17
2.11	Android Studio	17
2.12	Java.....	17
2.13	Python	18
2.14	<i>Tensorflow</i>	18
2.15	<i>Library Googletrans</i>	19
2.16	Penelitian Terkait	19
BAB III ANALISIS DAN PERANCANGAN SISTEM		21
3.1	Analisis.....	21
3.1.1	Analisis Permasalahan	21
3.1.2	Analisis Kebutuhan.....	22
3.1.3	Deskripsi Umum	23
3.2	Perancangan Basis Data	24
3.2.1	<i>Brainfile</i>	24
3.2.2	Koleksi	25
3.2.3	Fasilitas	26
3.3	Perancangan Sistem.....	27
3.3.1	Diagram Kasus Penggunaan	27
3.3.2	Deskripsi Kasus Penggunaan dan Diagram Aktivitas.....	28
3.3.2.1	Melihat Peta Kebun Binatang.....	28
3.3.2.2	Menanyakan Pertanyaan pada <i>Chatbot</i>	30
3.3.2.3	Mendapat Notifikasi informasi Satwa	33
3.4	Perancangan Aturan <i>Chatbot</i>	34
3.4.1	Menerjemahkan <i>input</i> pengunjung ke dalam bahasa Inggris	37
3.4.2	<i>Encoding</i> menggunakan <i>Universal Sentence Encoder</i>	37
3.4.3	Tahap Perolehan Jawaban.....	38
3.4.4	Pengecekan Fasilitas pada Basis Data	39
3.4.5	Menentukan Fasilitas Terdekat	40
3.4.6	Menampilkan Posisi dari Fasilitas Terdekat	40

3.5	Perancangan Fitur Pemandu.....	41
3.6	Perancangan Antarmuka	42
BAB IV IMPLEMENTASI.....		43
4.1	Lingkungan Implementasi.....	43
4.2	Pengambilan Data	43
4.3	Implementasi.....	44
4.3.1	Implementasi Basis Data	44
4.3.2	Implementasi Kelas Fasilitas	45
4.3.3	Implementasi Kelas Koleksi	47
4.3.4	Implementasi Pengecekan Input <i>Chatbot</i>	49
4.3.5	Implementasi Penerjemahan	49
4.3.6	Implementasi Inisialisasi Model	49
4.3.7	Implementasi Perolehan Jawaban.....	51
4.3.8	Implementasi Pengecekan Fasilitas pada Basis Data	53
4.3.9	Implementasi Penentuan Fasilitas Terdekat.....	55
4.3.10	Implementasi Menampilkan Fasilitas	57
4.3.11	Implementasi <i>Geofencing</i>	58
4.3.12	Implementasi Notifikasi <i>Geofencing</i>	59
4.3.13	Implementasi Antarmuka.....	60
4.3.13.1	Antarmuka <i>Splash Screen</i>	61
4.3.13.2	Antarmuka Menu <i>Chatbot</i>	62
4.3.13.3	Antar Muka Menu Peta	63
BAB V UJI COBA DAN EVALUASI.....		65
5.1	Lingkungan Uji Coba.....	65
5.2	Pengujian.....	65
5.2.1	Pengujian Fungsionalitas Sistem	65
5.2.2	Pengujian Kerelevanan Jawaban <i>Chatbot</i>	66
5.2.3	<i>System Usability Scale (SUS)</i>	66
5.3	Evaluasi	67
BAB VI KESIMPULAN		73
6.1.	Kesimpulan	73
6.2.	Saran	73

LAMPIRAN A : Hasil Pengujian Fungsionalitas Sistem.....	75
LAMPIRAN B : <i>Majority Vote</i> Kerelevanan Jawaban <i>Chatbot</i>	
.....	79
LAMPIRAN C : Dokumentasi Hasil Kuesioner	91
DAFTAR PUSTAKA	97
BIODATA PENULIS	101

DAFTAR GAMBAR

Gambar 2.1 Kategori dari LBS [6]	10
Gambar 2.2 Metode <i>Positioning</i> Beserta Akurasinya [6]	12
Gambar 2.3 Diagram <i>Deep Averaging Network</i> [14]	16
Gambar 3.1 Struktur Basis Data <i>Node Brainfile</i>	24
Gambar 3.2 Struktur Basis Data <i>Node Koleksi</i>	25
Gambar 3.3 Struktur Basis Data <i>Node Fasilitas</i>	26
Gambar 3.4 Diagram Kasus Penggunaan Aplikasi	28
Gambar 3.5 Diagram Aktivitas Melihat Peta Kebun Binatang ...	30
Gambar 3.6 Diagram Aktivitas Menanyakan Pertanyaan pada <i>Chatbot</i>	32
Gambar 3.7 Diagram Aktivitas Mendapat Notifikasi Informasi Satwa	34
Gambar 3.8 Diagram Alir Aturan <i>Chatbot</i>	35
Gambar 3.9 Diagram Alir Tahap Perolehan Jawaban	39
Gambar 3.10 Diagram Alir Fitur Pemandu	41
Gambar 3.11 Rancangan Antarmuka Menu <i>Chatbot</i>	42
Gambar 3.12 Rancangan Antarmuka Menu Peta	42
Gambar 4.1 Implementasi <i>Node Brainfile</i>	45
Gambar 4.2 Implementasi <i>Node Fasilitas</i>	45
Gambar 4.3 Implementasi <i>Node Koleksi</i>	45
Gambar 4.4 Contoh Hasil Perhitungan <i>Cosine Similarity</i>	52
Gambar 4.5 Antarmuka <i>Splash Screen</i>	61
Gambar 4.6 Antarmuka Menu <i>Chatbot</i>	62
Gambar 4.7 Antarmuka Menu Peta	63
Gambar 5.1 Peringatan Lisensi Chaquopy	68
Gambar 5.2 Peringkat Nilai Skor <i>System Usability Scale</i>	71
Gambar 7.1 Dokumentasi Kuesioner Kerelevanan Pertanyaan 1-10	91
Gambar 7.2 Dokumentasi Kuesioner Kerelevanan Pertanyaan 11-20	92
Gambar 7.3 Dokumentasi Kuesioner Kerelevanan Pertanyaan 21-30	93

Gambar 7.4 Dokumentasi Kuesioner Kerelevanan Pertanyaan 31-4094

Gambar 7.5 Dokumentasi Kuesioner Kerelevanan Pertanyaan 41-5095

Gambar 7.6 Dokumentasi Kuesioner *System Usability Scale*96

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait	20
Tabel 3.1 Kebutuhan Fungsional.....	22
Tabel 3.2 Kebutuhan Non-Fungsional	23
Tabel 3.3 Penjelasan Atribut <i>Node Brainfile</i>	25
Tabel 3.4 Penjelasan Atribut <i>Node Koleksi</i>	26
Tabel 3.5 Penjelasan Atribut <i>Node Fasilitas</i>	27
Tabel 3.6 Deskripsi Kasus Penggunaan Melihat Peta Kebun Binatang	29
Tabel 3.7 Deskripsi Kasus Penggunaan Menanyakan Pertanyaan pada <i>Chatbot</i>	31
Tabel 3.8 Deskripsi Kasus Penggunaan Mendapat Notifikasi Informasi Satwa.....	33
Tabel 4.1 Spesifikasi Lingkungan Implementasi	43
Tabel 5.1 Spesifikasi Lingkungan Uji Coba.....	65
Tabel 5.2 Perolehan Skor dari <i>System Usability Scale</i>	67
Tabel 5.3 <i>Confusion Matrix</i>	69
Tabel 5.4 Perhitungan Skor SUS.....	71
Tabel 7.1 Hasil Pengujian Fungsionalitas Sistem	75
Tabel 7.2 <i>Majority Vote</i> Kerelevanan Jawaban <i>Chatbot</i>	79

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Kelas Fasilitas	47
Kode Sumber 4.2 Kelas Koleksi	48
Kode Sumber 4.3 Fungsi <i>isAskingLocation()</i>	49
Kode Sumber 4.4 Implementasi Penerjemahan.....	49
Kode Sumber 4.5 Pengambilan Data dari Firebase.....	50
Kode Sumber 4.6 Inisialisasi Model <i>Chatbot</i>	50
Kode Sumber 4.7 Fungsi <i>getSparseTensorFormat()</i>	51
Kode Sumber 4.8 Fungsi <i>getAnswer()</i>	53
Kode Sumber 4.9 Fungsi <i>isExistinDatabase()</i>	54
Kode Sumber 4.10 Mendapatkan Daftar Fasilitas yang Ditanyakan	54
Kode Sumber 4.11 Fungsi <i>getClosestFacility()</i>	56
Kode Sumber 4.12 Fungsi <i>showFacilityinMap()</i>	57
Kode Sumber 4.13 Menampilkan Posisi dari Koordinat.....	58
Kode Sumber 4.14 Kelas <i>GeofenceHelper</i>	59
Kode Sumber 4.15 Kelas <i>GeofenceBot</i>	60

[Halaman ini sengaja dikosongkan]

DAFTAR PERSAMAAN

Persamaan 2.1 <i>Cosine Similarity</i>	16
Persamaan 2.2 Haversine	17
Persamaan 5.1 Perhitungan Akurasi.....	69
Persamaan 5.2 Perhitungan Presisi	Error! Bookmark not defined.
Persamaan 5.3 Perhitungan <i>Recall</i>	69

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Bab ini menjelaskan tentang garis besar dari tugas akhir yang dikerjakan. Bab ini meliputi latar belakang, tujuan, rumusan masalah, batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan buku tugas akhir.

1.1 Latar Belakang

Kebun binatang merupakan salah satu tempat wisata rekreasi edukasi yang berada di pusat kota. Dengan luas hampir 14 hektar dan koleksi sebanyak kurang lebih 800 satwa, salah satu kebun binatang di daerah Jawa Barat memiliki berbagai fasilitas yang menarik untuk belajar sambil berwisata. Beberapa fasilitas dari kebun binatang tersebut antara lain adalah taman bermain anak, teater satwa, perahu kayuh, gajah tunggang, berbagai macam koleksi satwa, dan lain-lain. Kebun binatang ini juga menyediakan *zoo educator* atau pemandu kebun binatang yang dapat membantu pengunjung berwisata.

Pasien positif Covid-19 di Indonesia masih terus bertambah. Beberapa usaha telah dilakukan oleh pemerintah, diantaranya yaitu memberlakukan pembatasan sosial dan *new normal*. Dengan diberlakukannya pembatasan sosial, protokol kesehatan, dan *new normal*, diharapkan masyarakat dapat mengurangi kontak sehingga dapat menekan angka penyebaran Covid-19. Namun, pemberlakuan pembatasan sosial dan *new normal* tersebut memberikan dampak negatif di berbagai sektor, tidak terkecuali di sektor pariwisata.

Saat Pembatasan Sosial Berskala Besar (PSBB) diterapkan oleh pemerintah, kebun binatang dalam penelitian ini ditutup untuk sementara mulai pada tanggal 23 Maret 2020 hingga diterapkannya *new normal*. Saat dibuka kembali, pihak kebun binatang membatasi jumlah pekerja. Jumlah edukator yang menjadi salah satu media pemberian informasi mengenai satwa juga dibatasi. Akibatnya, pengalaman dan informasi yang bisa didapatkan pengunjung menjadi berkurang. Jumlah pengunjung

yang datang juga berkurang secara drastis. Sebelum pandemi, pengunjung yang datang bisa mencapai 3000 orang pada hari Minggu dan saat ini pengunjung kebun binatang pada hari Minggu berjumlah kurang dari seribu pengunjung dan bahkan hanya 2 orang pengunjung dalam sehari. Keadaan ini sangat mengkhawatirkan mengingat sumber utama dana pembelian pakan satwa di kebun binatang berasal dari pembelian tiket masuk.

Saat ini, teknologi sudah banyak berkembang dan cara memperoleh informasi semakin mudah. Untuk mengakses informasi, *mobile device* menjadi perangkat yang menjadi favorit karena mudah digunakan dan mudah dibawa. Salah satu teknologi yang dapat digunakan untuk memperoleh informasi melalui *mobile device* adalah *chatbot*. *Chatbot* merupakan sebuah program yang mampu menyimulasikan percakapan dengan menggunakan teknologi kecerdasan buatan sehingga mampu memberikan respon.

Dalam penelitian ini, teknologi GPS dan metode *Universal Sentence Encoder* (USE) digunakan untuk membangun sebuah *chatbot* pada aplikasi layanan wisata kebun binatang. Diharapkan dengan adanya penelitian ini dapat menghasilkan sebuah *chatbot* untuk aplikasi pemandu wisata kebun binatang sehingga meningkatkan minat berpariwisata dan mengurangi interaksi pengunjung dengan dengan *zookeeper* untuk mencegah penularan Covid-19.

1.2 Rumusan Masalah

Rumusan masalah pada tugas akhir ini adalah bagaimana membangun aplikasi *location-aware chatbot* dengan menerapkan *Universal Sentence Encoder* (USE) pada aplikasi wisata kebun binatang berbasis Android.

1.3 Batasan Masalah

Batasan masalah dari tugas akhir ini yaitu:

1. Hasil akhir dari tugas akhir ini adalah aplikasi *mobile*.
2. Pemodelan *chatbot* menggunakan bahasa Inggris.

3. *Chatbot* dibuat menggunakan bahasa pemrograman Python.
4. Aplikasi ini menggunakan platform Android.
5. Aplikasi dibuat menggunakan bahasa pemrograman Java dan Python.
6. Dibangun menggunakan platform PC dengan sistem operasi Windows 10.
7. Menggunakan *location-based service* dengan memanfaatkan teknologi GPS.
8. Peta yang digunakan dalam aplikasi menggunakan Google Map API.
9. *Chatbot* mampu mengirimkan pesan kepada pengguna berupa informasi mengenai koleksi kebun binatang ketika pengguna memasuki radius tertentu dari suatu kandang binatang.
10. *Chatbot* dapat mengirimkan informasi berupa lokasi dari fasilitas terdekat.
11. Aplikasi ini digunakan untuk wisatawan kebun binatang.

1.4 Tujuan

Tujuan dari tugas akhir ini adalah membangun *location-aware chatbot* untuk aplikasi layanan wisata kebun binatang menggunakan *Universal Sentence Encoder*.

1.5 Manfaat

Manfaat dari tugas akhir ini adalah sebagai berikut.

1. Memudahkan pengunjung mendapatkan informasi dan mengakses fasilitas yang ada di kebun binatang.
2. Mengurangi interaksi antar pengunjung dengan pemandu atau *zoo educator* kebun binatang untuk mengurangi kontak.
3. Memudahkan pengunjung untuk melihat koleksi di kebun binatang.

1.6 Metodologi

Pengerjaan tugas akhir ini dilaksanakan dengan tahap-tahap yang ditunjukkan sebagai berikut.

1.6.1 Penyusunan Proposal Tugas Akhir

Proposal tugas akhir memuat deskripsi dan penjelasan dari tugas akhir yang akan dilakukan. Deskripsi dan penjelasan pengerjaan tugas akhir tersebut dijadikan beberapa poin, yaitu latar belakang, rumusan masalah, batasan masalah, tujuan tugas akhir, manfaat tugas akhir, tinjauan pustaka, ringkasan tugas akhir, metodologi dan jadwal pengerjaan.

Latar belakang masalah berisi uraian tentang permasalahan yang mendorong penulis untuk membuat suatu solusi melalui tugas akhir. Kemudian, permasalahan tersebut dituliskan kembali pada bagian rumusan masalah dalam bentuk pertanyaan. Batasan masalah berisi variabel dan parameter dari penelitian. Tinjauan pustaka berisi referensi yang akan digunakan dalam pengerjaan tugas akhir. Ringkasan tugas akhir berisi penjelasan mengenai tahapan dari tugas akhir yang akan dilakukan. Tahapan tersebut lalu diperjelas dalam bagian metodologi. Selanjutnya, pembagian waktu kegiatan yang akan dilakukan selama pelaksanaan tugas akhir ditunjukkan dalam bagian jadwal kegiatan.

1.6.2 Pengumpulan Data

Pada tahap pengumpulan data, dilakukan studi literatur serta elisitasi kebutuhan. Adapun teknik elisitasi kebutuhan yang dilakukan untuk mengumpulkan data adalah sebagai berikut.

1. Analisis dokumen.
2. Observasi.
3. Wawancara

1.6.3 Analisis Kebutuhan

Pada tahapan analisis kebutuhan dilakukan pemahaman mengenai permasalahan dari ruang lingkup hingga menentukan

kebutuhan fungsional dan non-fungsional. Analisis Kebutuhan dilakukan dengan strategi sebagai berikut.

1. *Benchmarking*
2. *Outcome analysis*

1.6.4 Tahap Desain

Tahapan ini dilakukan untuk merancang sistem yang akan dibangun. Tahapan desain meliputi desain *back-end* dan *front-end*. Hasil dari tahapan ini berupa diagram-diagram, bagan-bagan, dan artefak lainnya yang mendukung tahap implementasi perangkat lunak.

1.6.5 Implementasi Perangkat Lunak

Aplikasi dibangun menggunakan *tools* Android Studio menggunakan bahasa pemrograman Java. Pembuatan *chatbot* pada aplikasi memanfaatkan teknologi Google Map API dan teknologi GPS serta menggunakan *Universal Sentence Encoder* (USE). Adapun alur untuk *chatbot* yang dibangun adalah sebagai berikut.

1. Model *Deep Averaging Network* dari USE dimuat.
2. Data kalimat-kalimat pertanyaan dijadikan vektor *embedding* menggunakan model USE.
3. Apabila pengguna mengirimkan pertanyaan, pertanyaan tersebut diterjemahkan ke dalam bahasa Inggris. Hal tersebut dilakukan karena *input* dari model USE berupa teks bahasa Inggris.
4. Apabila pertanyaan tersebut mengandung kata “terdekat”, maka sistem akan mengirimkan informasi lokasi terdekat dari fasilitas yang ditanyakan.
5. Selain itu, pertanyaan dimasukkan ke dalam model sehingga menghasilkan vektor *embedding*.
 - a. Menghitung kemiripan vektor pertanyaan pengguna dan vektor data pertanyaan menggunakan *cosine similarity*.

- b. Respon pertanyaan pengguna adalah jawaban dari data pertanyaan yang paling mirip dengan pertanyaan pengguna.

Apabila pengguna memasuki *geofence* dari lokasi koleksi binatang, maka *chatbot* akan mengirimkan informasi mengenai koleksi tersebut.

1.6.6 Uji Coba dan Evaluasi

Pengujian aplikasi dilakukan menggunakan metode *black box*. Pengujian dilakukan dengan membandingkan hasil yang diharapkan dan *output* dari aplikasi yang telah dibangun. Pengujian *chatbot* dilakukan berdasarkan relevansi jawaban dengan pertanyaan pengguna dengan mengujikan 50 pertanyaan.

1.6.7 Penyusunan Buku Tugas Akhir

Penyusunan buku tugas akhir dilaksanakan selama pelaksanaan tugas akhir. Buku tugas akhir berisi mengenai dasar teori dan metode yang diterapkan, hasil implementasi, serta hasil uji coba dari aplikasi yang dibuat. Sistematika dari buku tugas akhir secara garis besar adalah sebagai berikut.

1. Pendahuluan
 - a. Latar Belakang
 - b. Rumusan Masalah
 - c. Batasan Tugas Akhir
 - d. Tujuan
 - e. Metodologi
 - f. Sistematika Penulisan
2. Dasar Teori
3. Analisis dan Perancangan Sistem
4. Implementasi
5. Pengujian dan Evaluasi
6. Kesimpulan dan Saran
7. Daftar Pustaka

1.7 Sistematika Penulisan

Buku tugas akhir ini disusun dengan sistematika yang secara garis besar ditunjukkan sebagai berikut.

BAB I : Pendahuluan

Bab ini menunjukkan latar belakang, rumusan masalah, batasan, tujuan, manfaat, metodologi, dan sistematika penulisan dari penyusunan tugas akhir.

BAB II : Dasar Teori

Bab ini berisi teori-teori yang digunakan dalam pengerjaan tugas akhir.

BAB III : Analisis dan Perancangan Sistem

Bab ini menjelaskan desain dan rancangan sistem yang dibangun untuk tugas akhir.

BAB IV : Implementasi

Bab ini menjelaskan implementasi sistem berdasarkan desain dan rancangan sistem yang telah didesain sebelumnya.

BAB V : Uji Coba dan Evaluasi

Bab ini menjelaskan pengujian serta evaluasi dari hasil pengujian tersebut.

BAB VI : Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran berdasarkan uji coba dan evaluasi yang telah dilakukan.

[Halaman ini sengaja dikosongkan]

BAB II

DASAR TEORI

Bab ini membahas teori-teori yang menjadi dasar dalam pembangunan aplikasi *chatbot* dengan mengimplementasikan *Universal-Sentence Encoder* untuk menentukan respon dari *chatbot*. Adapun teori yang dijadikan penunjang diantaranya adalah pembahasan mengenai *chatbot*, *Location-Based Services* (LBS), *Positioning Methods*, *Location-Aware Application* (LAA) GPS, *Universal Sentence Encoder* (USE), *Geofencing*, serta Android Studio.

2.1 Chatbot

Chatbot merupakan program yang dapat berinteraksi dengan penggunanya menggunakan bahasa natural, baik menggunakan teks maupun melalui suara [1]. *Chatbot* memiliki dua komponen, yaitu program utama dan data *brainfile*. Program utama alias bot, terdiri dari *scanner* dan *parser*. Komponen ini bertugas untuk mendapatkan input dari pengguna serta melakukan *parsing* untuk mendapatkan jawaban atau respon. Komponen kedua adalah *brainfile* yang berisi pengetahuan dan kosakata [2].

2.2 Wisata Kebun Binatang

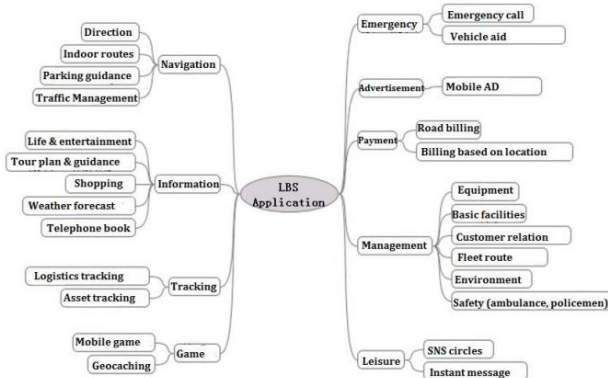
Kebun binatang merupakan tempat penangkaran, pengembangbiakan, dan tempat binatang-binatang ditampilkan sebagai atraksi yang dibuka untuk umum. Kebun binatang umumnya memiliki fungsi sebagai tempat edukasi, wisata, penelitian, dan penangkaran hewan. Menurut *South East Asian Zoos Association* (SEAZA) istilah kebun binatang (*zoo*) meliputi kebun binatang (*zoological garden*), taman safari, akuarium publik, terarium, dan berbagai koleksi hewan lainnya yang memiliki fungsi edukasi, penelitian, serta konservasi [3].

2.3 Location-Based Services (LBS)

Location-based services (LBS) atau layanan berbasis lokasi adalah sebuah konsep yang menggabungkan gagasan pelayanan dengan lokasi geografis yang berupa koordinat spasial. Ilmuwan dan peneliti mengklasifikasikan aplikasi LBS menjadi dua kelas, yaitu:

1. *Person-oriented LBS* adalah aplikasi yang berbasis pengguna. Pada kelas ini, aplikasi menggunakan posisi dari orang serta menggunakan posisi dari seseorang tersebut untuk meningkatkan pelayanan. Biasanya orang yang diposisikan dapat mengontrol pelayanan tersebut.
2. *Device-oriented LBS* adalah aplikasi yang tidak hanya berfokus pada posisi dari seseorang, tetapi juga dapat berfokus pada benda-benda, sekelompok orang, dan objek lainnya. Biasanya, benda atau orang yang diposisikan pada aplikasi ini, tidak dapat mengontrol pelayanan tersebut, sebagai contoh yaitu aplikasi pelacak mobil [4].

Teknologi LBS merupakan pertemuan dari 3 teknologi, yaitu *New Information and Communication Technologies* (NICTS) seperti perangkat seluler dan internet, *Geographic Information Systems* (GIS) dan *database* spasial [5]. Penggunaan LBS dibagi menjadi beberapa kategori seperti yang ditunjukkan oleh Gambar 2.1.



Gambar 2.1 Kategori dari LBS [6]

2.2.1 Positioning Methods

Positioning methods atau metode penentuan lokasi yang dapat dimanfaatkan dalam *Location-Based Services* adalah sebagai berikut [6].

1. *Terminal-based positioning*

Metode ini menggunakan perhitungan data GPS yang ditangkap oleh perangkat. Teknologi ini dinilai lebih aman karena proses penentuan posisi ditentukan di perangkat seluler. Namun, metode ini memiliki kekurangan, yaitu jika penerima GPS berada di dalam ruangan atau ketika cuaca sedang berawan, maka metode ini tidak akan bekerja dengan baik.

2. *Network-based positioning*

Lokasi pengguna dapat ditentukan saat terhubung dengan beberapa *base station* dengan menghitung posisi *base station* terdekat. Setiap *base station* pada jaringan seluler memiliki *device IDE* yang berbeda-beda dan data mengenai posisi *base station* disimpan di dalam *database* milik operator jaringan.

3. *Hybrid*

Metode ini menggabungkan metode *terminal-based positioning* dan *network-based positioning*. Contoh dari penentuan posisi ini adalah *Assisted GPS* (A-GPS).

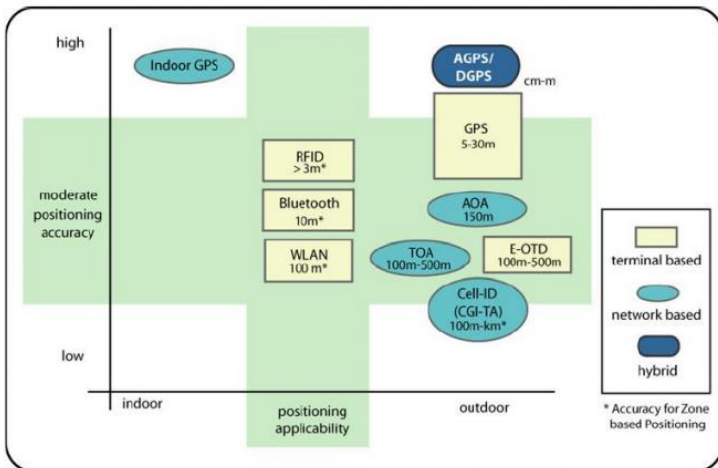
4. *Bluetooth-based positioning*

Setiap perangkat bluetooth memiliki alamat yang unik (BT_ADDR) yang apabila diaktifkan, perangkat tersebut akan dapat dicari dan dideteksi oleh perangkat lainnya. Metode pertama yang menggunakan teknologi bluetooth menggunakan *Radio Signal Strength Indicator* (RSSI) dari *access point* atau jalur akses pada bluetooth. Dengan menghitung jarak perangkat target dengan beberapa jalur akses, posisi perangkat target dapat diketahui. Metode lainnya yaitu membuat sebuah *database* yang menghubungkan BT_ADDR dari jalur akses suatu bluetooth dengan posisi perangkat bluetooth tersebut.

5. *WLAN-based positioning*

Penggunaan WLAN dalam penentuan posisi hampir sama dengan *bluetooth-based positioning*. Yang pertama yaitu mengukur RSSI dari jalur akses WLAN. Cara kedua yaitu dengan mendapatkan nama dan RSSI dari beberapa WLAN, kemudian dibandingkan dengan data yang dijadikan referensi serta informasi lokasi yang ada di *database*.

Metode penentuan posisi tentu memiliki kelebihan dan kekurangan. Tentunya, akurasi dari suatu metode sangat mempengaruhi performa dan kualitas dari suatu layanan yang berbasis lokasi. Akurasi dan penerapan metode *positioning* untuk dalam dan di luar ruangan ditunjukkan oleh Gambar 2.2.



Gambar 2.2 Metode Positioning Beserta Akurasinya [6]

2.4 Location-Aware Application (LAA)

LAA adalah aplikasi yang bekerja berdasarkan lokasi geografis pengguna. Informasi yang didapatkan akan disaring oleh aplikasi berdasarkan lokasi geografis pengguna dan digunakan

pada fitur dan pelayanan penggunaannya. Metode yang dapat digunakan untuk mendapatkan lokasi dari penggunaannya antara lain Wifi, GPS, GSM, dan bluetooth. Penggunaan GPS pada *Location-Aware Application* muncul pertama kali pada tahun 2002 [7].

2.5 *Global Positioning System (GPS)*

GPS dikembangkan oleh *United States Departement of Defence* untuk kepentingan militer. Prinsip dari teknologi ini adalah mengukur jarak antar penerima sinyal satelit dengan beberapa satelit. Posisi dari penerima sinyal dapat diketahui berdasarkan lokasi satelit dan jarak penerima sinyal. Penggunaan GPS biasanya digunakan untuk penentuan lokasi serta navigasi. Saat ini, GPS telah mampu diaplikasikan untuk berbagai kebutuhan seperti pelacakan dan penentuan lokasi [8]. GPS terdiri dari 3 segmen, yaitu [9]:

1) *Space segment*

Segmen ini merupakan jaringan dari satelit-satelit yang mengorbit bumi.

2) *User segment*

User segment adalah penerima sinyal GPS.

3) *Control segment (CS)*

Segmen ini bertugas untuk mengontrol dan memelihara satelit agar tetap dapat bekerja. Pemeliharaan yang dilakukan control segment antara lain adalah menjaga satelit agar tetap berada di jalur orbit, mengamati status dan kesehatan *subsystem*, tingkat daya baterai, dan tingkat propelan yang digunakan untuk bermanuver [10].

2.6 *Google Maps*

Google Maps menampilkan peta dunia berdasarkan satelit yang dapat digunakan untuk melihat suatu daerah. Terdapat 4 model peta yang disediakan oleh Google Maps, antara lain yaitu;

4) *Roadmap*

Roadmap atau peta jalan menampilkan peta dalam bentuk dua dimensi.

5) *Satellite*

Peta dengan mode *satellite* menampilkan peta hasil dari citra satelit.

6) *Terrain*

Model *terrain* menunjukkan *relief* atau bentuk permukaan dari bumi.

7) *Hybrid*

Model peta *hybrid* merupakan gabungan dari model lainnya sehingga memberikan tampilan satelit dengan informasi jalan dan nama daerah, sehingga memiliki informasi seperti peta *roadmap*.

Google Maps menyediakan layanan *Application Programming Interface* (API) sehingga peta dapat digunakan untuk membangun suatu aplikasi. Pada layanan tersebut, peta dapat diubah sesuai kebutuhan pengguna. Dengan menggunakan koordinat, peta tersebut dapat ditambahkan penanda, *polyline*, atau bentuk *polygon*[11].

2.7 Geofencing

Geofencing merupakan teknologi yang digunakan untuk memonitor suatu objek menggunakan GPS. Informasi koordinat dari objek tersebut secara berkala dikirimkan melalui jaringan telpon seluler menuju *control center*. Di sisi lain, beberapa data koordinat digunakan untuk membuat *virtual boundary* atau yang disebut juga dengan *geofence*. Sistem akan menentukan apakah posisi objek yang diamati berada di dalam atau di luar *geofence* tersebut [12].

2.8 Universal Sentence Encoder (USE)

Universal Sentence Encoder merupakan *encoder* kalimat yang menghasilkan vektor 512 dimensi. USE menyediakan dua *pre-trained model* yang dapat digunakan untuk berbagai kebutuhan

seperti analisis sentimen dan klasifikasi. Adapun tahapan yang dilakukan oleh USE, yaitu sebagai berikut:

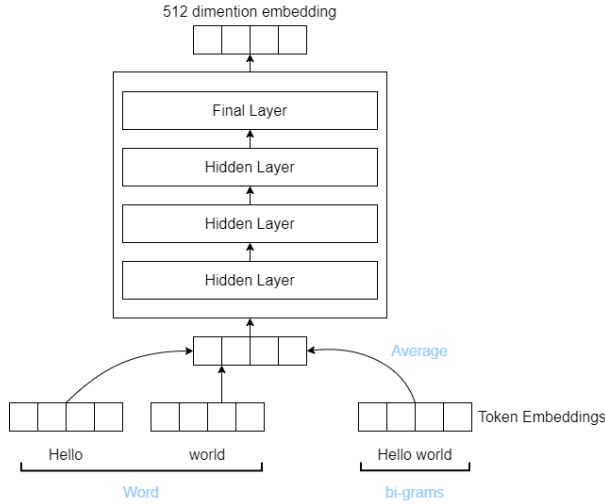
1. Mengubah kalimat menjadi huruf kecil dan tokenisasi menggunakan *Penn Tree Bank Tokenization* (PTB). Adapun proses tokenisasi yang dilakukan berupa:
 - a. Membagi kontraksi bahasa Inggris yang umum, seperti *I'll* menjadi *I* dan *'ll*.
 - b. Membagi sebagian besar tanda baca sebagai token terpisah.
 - c. Memisahkan tanda baca koma dan tanda kutip tunggal apabila diikuti dengan spasi.
 - d. Memisahkan tanda baca titik yang terdapat pada akhir kalimat.
2. Menggunakan model *encoder*. Kedua model tersebut dijelaskan sebagai berikut[13].

a. *Transformer*

Model pertama menggunakan bagian *encoder* dari arsitektur *transformer*. Terdapat 6 lapisan *transformer* yang setiap lapisannya memiliki *self-attention module* dan jaringan *feed-forward*. Model ini didesain untuk mendapatkan akurasi yang tinggi.

b. *Deep Averaging Network* (DAN)

Pada model ini, bi-grams dan embeddings untuk kata dari kalimat yang dimasukkan dirata-ratakan. Kemudian, hasil rata-rata tersebut dimasukkan kedalam 4 lapisan *feed-forward deep neural network* (DNN) hingga akhirnya menghasilkan vektor *embedding*. Model ini didesain untuk efisiensi dan kecepatan. Arsitektur ini dapat dilihat melalui Gambar 2.3.



Gambar 2.3 Diagram Deep Averaging Network [14]

2.9 Cosine Similarity

Kemiripan dua vektor bisa diukur menggunakan *cosine similarity*. *Cosine Similarity* menunjukkan kemiripan dua vektor melalui nilai cosine dari kedua vektor tersebut. Rumus dari *cosine similarity* ditunjukkan pada Persamaan 2.1 .

$$Sim(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|} = \frac{\sum_{i=1}^j w_{ai} \times w_{bi}}{\sqrt{\sum_{i=1}^j (w_{ai})^2} \cdot \sqrt{\sum_{i=1}^j (w_{bi})^2}} \quad (2.1)$$

\vec{a} merupakan vektor pertama dan \vec{b} adalah vektor kedua. Dilakukan perkalian *dot* antara \vec{a} dan \vec{b} lalu dari hasil perkalian tersebut dibagi dengan perkalian nilai norma dari vektor \vec{a} dan \vec{b} . Hasil dari perhitungan tersebut merupakan nilai cosine antara vektor \vec{a} dan \vec{b} dengan jangkauan nilai antara -1 hingga 1. Apabila nilai cosine yang didapatkan bernilai 1, maka kedua vektor tersebut

adalah vektor yang sama. Semakin nilai cosine mendekati nilai 1, semakin mirip kedua vektor tersebut.

2.10 Persamaan Haversine

Persamaan Haversine adalah salah satu metode yang dapat digunakan untuk menghitung jarak antara dua titik pada bumi. Metode Haversine menggunakan posisi pada garis bujur dan garis lintang bumi serta diameter bumi dalam mengukur jarak antar dua titik pada permukaan bumi. Persamaan Haversine ditunjukkan pada Persamaan 2.2.

$$d = 2r \sin^{-1} \left(\sqrt{\sin^2 \left(\frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left(\frac{\psi_2 - \psi_1}{2} \right)} \right) \quad (2.2)$$

d merupakan jarak antara dua titik dengan r adalah radius bumi. (ψ, φ) adalah titik pada garis bujur dan garis lintang.

2.11 Android Studio

Android studio adalah *Integrated Development Environment* (IDE) resmi untuk android. Beberapa fitur yang disediakan oleh Android studio yaitu [17]:

- 1) Menyediakan Emulator untuk simulasi.
- 2) Menggunakan *build tool* Gradle.
- 3) Terdapat fitur *Run Instant* yang digunakan untuk menyimulasikan aplikasi.
- 4) Dapat terintegrasi dengan Github.
- 5) Mendukung NDK dan C++.

2.12 Java

Bahasa pemrograman Java dikembangkan melalui sebuah proyek yang bernama “The GreenProject” pada tahun 1991 oleh

James Gosling di Sun Microsystem. Pemrograman Java memiliki beberapa karakteristik, yaitu [18]:

- 1) Bahasa pemrograman Java menggunakan sintak sehingga mudah dipahami.
- 2) Pemrograman Java berorientasi objek (*Object Oriented*) sehingga program dapat dibuat secara modular dan dapat digunakan kembali.
- 3) Memiliki reliabilitas yang tinggi.
- 4) Memiliki keamanan yang baik.
- 5) *Multithreaded*.
- 6) Bahasa pemrograman Java mampu berjalan pada lingkungan yang beragam.

2.13 Python

Python merupakan bahasa pemrograman tingkat tinggi yang berorientasi objek. Bahasa pemrograman Python dapat dijalankan di berbagai platform sistem operasi [19]. Python memiliki filosofi desain yang menekankan keterbacaan kode sehingga memiliki sintaks yang mudah dibaca dan dipahami [20].

2.14 Tensorflow

Tensorflow merupakan kerangka kerja *machine learning* berbentuk *open-source library* yang dapat bekerja di berbagai perangkat. Tensorflow pada mulanya dikembangkan dengan tujuan penelitian serta produksi dan kemudian dipublikasikan pada tahun 2015 dibawah lisensi Apache dan kemudian pada tahun 2017 versi stabil dari Tensorflow diluncurkan. Pada bulan September 2019, versi terbaru Tensorflow diluncurkan dengan nama Tensorflow 2.0. Tensorflow merepresentasikan algoritma *machine learning* dalam bentuk *dataflow graph* dan memodelkan data sebagai array n-dimensi (*tensor*).

2.15 *Library* Googletrans

Library googletrans merupakan *library* pada pemrograman Python yang mengimplementasikan Google Translate API. *Library* ini digunakan untuk melakukan penerjemahan bahasa yang bisa digunakan secara gratis. Googletrans menggunakan *server* yang sama dengan trans.google.com. Beberapa fitur dari googletrans diantaranya adalah penerjemahan melalui link, deteksi bahasa, dan penerjemahan teks.

2.16 Penelitian Terkait

Terdapat beberapa penelitian sebelumnya yang memiliki keterkaitan dengan tugas akhir ini. Penelitian-penelitian tersebut dijelaskan dalam Tabel 2.1.

Tabel 2.1 Penelitian Terkait

Nama Peneliti	Judul	Metode	Hasil
Mungki Astiningrum, Pramana Yoga Saputra, Maya Shoburu Rohmah (2018) [21]	Implementasi NLP dengan Konversi Kata pada Sistem <i>Chatbot</i> Konsultasi Laktasi	<i>Levenshtein Distance</i> , TFIDF, dan <i>Cosine Similarity</i>	Penerapan <i>Levenshtein distance</i> , TFIDF, dan <i>cosine similarity</i> menghasilkan akurasi sebesar 85% pada pertanyaan sembarang, nilai <i>recall</i> 85% dan nilai <i>precision</i> sebesar 94%.
Zhuowei Hu, Zhiheng Wang, Hongqi Liu, Hongxia Bie (2010) [22]	<i>Self-Service Folk Tourism Guiding Technology on Mobile Terminal with Multi-Mode: Application of GPS and Electronic Map</i>	GPS dan GIS	Aplikasi tur <i>self-guided</i> yang dibangun menggunakan GPS dan GIS memberikan kemudahan dalam penerapannya serta membutuhkan biaya yang rendah. Selain itu, wisatawan mampu mendapatkan pengalaman berwisata yang lebih fleksibel.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini akan dibahas mengenai tahapan analisis dan perancangan sistem aplikasi *chatbot* yang akan dibangun. Secara garis besar, pembahasan yang akan dijelaskan adalah analisis permasalahan, analisis kebutuhan, deskripsi umum, perancangan basis data, perancangan sistem, serta perancangan *chatbot*.

3.1 Analisis

3.1.1 Analisis Permasalahan

Kebun binatang merupakan salah satu destinasi wisata dan sarana edukasi. Kebun binatang yang menjadi objek pada penelitian ini menyediakan jasa pemandu yang disebut juga sebagai *educator* untuk memfasilitasi pengunjung. Sejak Covid-19 memasuki Indonesia, pihak kebun binatang melakukan beberapa upaya untuk mengurangi tingkat penyebaran virus, diantaranya yaitu dengan mengurangi jumlah pekerja yang bekerja dan menerapkan protokol kesehatan. Pengurangan pekerja tersebut tidak terkecuali berlaku pada jasa edukator dan menyebabkan pelayanan informasi menjadi tidak optimal. Akibatnya, pengalaman berwisata pengunjung berkurang karena kurangnya kemampuan memperoleh informasi dan minat wisatawan pun menjadi berkurang. Untuk itu, diperlukan suatu solusi agar pengunjung tetap mampu memperoleh informasi yang dibutuhkan.

Solusi yang diusulkan pada tugas akhir ini adalah membangun aplikasi *chatbot* dengan menerapkan model *Universal Sentence Encoder*. Diharapkan melalui aplikasi ini kebutuhan wisatawan untuk memperoleh berbagai informasi seputar kebun binatang bisa diperoleh dan mendapatkan informasi yang relevan melalui *chatbot* yang dibangun. Uraian dari aplikasi yang akan dibangun dijelaskan pada bagian deskripsi umum (3.1.3).

3.1.2 Analisis Kebutuhan

Elisitasi kebutuhan dilakukan dengan melakukan wawancara, analisis dokumen, serta observasi. Wawancara dilakukan secara informal dengan mewawancarai sukarelawan yang pernah mengunjungi kebun binatang untuk mengetahui informasi yang pengunjung butuhkan tetapi sulit untuk ditemukan. Analisis dokumen dilakukan pada pamflet kebun binatang. Kemudian observasi dilakukan dengan mengunjungi kebun binatang secara langsung. Analisis kebutuhan dilakukan melalui dua strategi, yaitu *benchmarking* dan *outcome analysis*. Metode *benchmarking* menganalisa aplikasi serupa yang pernah dikembangkan dan metode *outcome analysis* dilakukan dengan menganalisa kebutuhan berdasarkan tujuan yang ingin dicapai. *Benchmarking* dilakukan terhadap aplikasi kebun binatang lainnya, yaitu Ragunan Zoo dan Surabaya Zoo. *Outcome analysis* berfokus kepada kemampuan pengunjung untuk memperoleh informasi seputar kebun binatang. Kebutuhan fungsional aplikasi yang dibangun untuk aplikasi ini ditunjukkan pada Tabel 3.1.

Tabel 3.1 Kebutuhan Fungsional

Kode Kebutuhan	Aktor	Deskripsi Kebutuhan
F-001	Pengunjung	Melihat peta kebun binatang
F-002	Pengunjung	Melihat informasi Kebun binatang
F-003	Pengunjung	Mendapat informasi lokasi fasilitas terdekat
F-004	Pengunjung	Mendapat informasi koleksi satwa kebun binatang

Selain kebutuhan fungsional, terdapat kebutuhan non-fungsional yang diperoleh berdasarkan analisis kebutuhan yaitu *operational* dan *performance*. Pada kebutuhan non-fungsional *operational* diperoleh berdasarkan pertimbangan teknologi yang banyak digunakan saat ini, sedangkan *performance* didasari oleh penelitian yang dilakukan oleh Appel (2012) yang memberikan waktu jeda pada *chatbot* antara 15 hingga 30 detik dengan tujuan untuk memberikan pengalaman berkomunikasi layaknya berinteraksi dengan manusia [23]. Kebutuhan non-fungsional aplikasi ini ditunjukkan pada Tabel 3.2.

Tabel 3.2 Kebutuhan Non-Fungsional

Kode Kebutuhan	Deskripsi Kebutuhan
NF-001	Aplikasi dapat berjalan pada sistem operasi Android 10+
NF-002	Respon <i>chatbot</i> tidak melebihi 30 detik

3.1.3 Deskripsi Umum

Berdasarkan permasalahan yang telah disebutkan pada analisis permasalahan, solusi yang ditawarkan melalui tugas akhir ini adalah membangun sebuah aplikasi *chatbot* yang dapat mengganti peran edukator atau pemandu di kebun binatang. Aplikasi tersebut diharapkan mampu memberikan informasi aktual serta dapat berinteraksi dengan pengunjung layaknya edukator.

Aplikasi yang dibangun untuk tugas akhir ini dinamakan ZooSite. Zoosite memiliki dua fitur utama, yaitu *chatbot* dan fitur *guide*. Fitur *chatbot* dapat melakukan tanya-jawab dengan pengunjung serta menampilkan lokasi dari suatu fasilitas apabila pengunjung menanyakan fasilitas terdekat. Fitur *guide* memunculkan notifikasi mengenai satwa yang didekati oleh pengunjung melalui menu peta. Aplikasi ini dirancang untuk pengunjung kebun binatang dan dapat digunakan pada *platform*

Android dengan tujuan agar dapat digunakan dengan mudah dan dapat dikembangkan.

Pada perancangan fitur *chatbot*, digunakan *Universal Sentence Encoder* untuk memproses pertanyaan dari pengunjung serta menggunakan *cosine similarity* untuk menentukan kemiripan antara pertanyaan dari pengunjung dengan pertanyaan yang ada pada *brainfile* sehingga diperoleh jawaban yang relevan. Aplikasi yang dibangun juga menerapkan *geofencing* pada fitur *guide* agar sistem dapat mengirimkan informasi mengenai koleksi satwa yang didekati oleh pengunjung.

3.2 Perancangan Basis Data

Perancangan basis data untuk ZooSite mengimplementasikan *Realtime Database* pada Firebase. *Realtime Database Firebase* menyimpan dalam bentuk objek JSON. Setiap data yang dimasukkan disimpan dalam berbentuk *node* yang dapat berisi pasangan nilai dengan suatu *key* (*key-value*). Berikut penjelasan dari *node-node* yang digunakan pada aplikasi ini.

3.2.1 *Brainfile*

```
"Brainfile" : [{
  "jawaban" : "value",
  "pertanyaan" : "value"
},
...
, {
  "jawaban" : "value",
  "pertanyaan" : "value"
}]
```

Gambar 3.1 Struktur Basis Data Node Brainfile

Node Brainfile sesuai dengan namanya menyimpan data-data yang akan digunakan dalam model *chatbot*. *Node* ini berisi data *brainfile*. Penjelasan dari struktur *node Brainfile* adalah sebagai berikut.

1. *Brainfile* adalah *node* utama.

2. *Brainfile* terhubung dengan *node* data *brainfile*. *Node* data *brainfile* berisi data pertanyaan dan jawaban yang dijelaskan pada Tabel 3.3.

Tabel 3.3 Penjelasan Atribut *Node Brainfile*

Nama Atribut	Tipe Data	Keterangan
pertanyaan	String	Data prediksi pertanyaan
jawaban	String	Jawaban dari data pertanyaan

3.2.2 Koleksi

```

"Koleksi" : [{
  "id" : "value"
  "nama" : "value",
  "deskripsi" : "value",
  "latitude" : "value",
  "longitude" : "value"
},
...
]

```

Gambar 3.2 Struktur Basis Data Node Koleksi

Data dari koleksi satwa kebun binatang disimpan pada *node* ini. Struktur dari *node* ini dijelaskan sebagai berikut.

1. Koleksi adalah *node* utama.
2. *Node* Koleksi berisi *node* data koleksi yang memiliki atribut yang dijelaskan pada

3. Tabel 3.4.

Tabel 3.4 Penjelasan Atribut *Node Koleksi*

Nama Atribut	Tipe Data	Keterangan
id	Int	Id dari data koleksi satwa
nama	String	Nama dari satwa
deskripsi	String	Deskripsi satwa
latitude	String	Posisi kandang pada garis lintang
longitude	String	Posisi kandang pada garis bujur

3.2.3 Fasilitas

```

"Facilitas" : [{
  "id" : "value",
  "nama" : "value",
  "deskripsi" : "value",
  "jam_buka" : "value",
  "jam_tutup" : "value",
  "latitude" : "value",
  "longitude" : "value"
},
... ]
- {

```

Gambar 3.3 Struktur Basis Data Node Fasilitas

Node Fasilitas menyimpan data mengenai fasilitas yang ada di kebun binatang. Penjelasan dari *node* ini adalah sebagai berikut.

1. Fasilitas adalah *node* utama.
2. *Node* Fasilitas berisi *node-node* data fasilitas yang memiliki atribut yang ditunjukkan pada

3. Tabel 3.5.

Tabel 3.5 Penjelasan Atribut *Node Fasilitas*

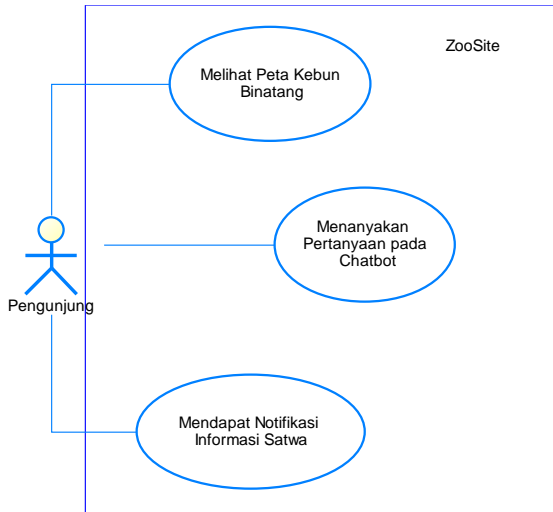
Nama Atribut	Tipe Data	Keterangan
id	Int	Id dari data fasilitas
nama	String	Nama fasilitas
deskripsi	String	Deskripsi fasilitas
latitude	String	Posisi fasilitas pada garis lintang
longitude	String	Posisi fasilitas pada garis bujur
jam_buka	String	Jam buka fasilitas
jam_tutup	String	Jam tutup fasilitas

3.3 Perancangan Sistem

Pada subbab ini, dijelaskan mengenai perancangan sistem dari ZooSite. Rancangan sistem untuk tugas akhir ini ditunjukkan oleh diagram kasus penggunaan, deskripsi kasus penggunaan, dan diagram aktivitas.

3.3.1 Diagram Kasus Penggunaan

Diagram kasus penggunaan menunjukkan hubungan antara pengguna dengan sistem. Diagram kasus penggunaan untuk sistem ini ditunjukkan pada Gambar 3.4.



Gambar 3.4 Diagram Kasus Penggunaan Aplikasi

3.3.2 Deskripsi Kasus Penggunaan dan Diagram Aktivitas

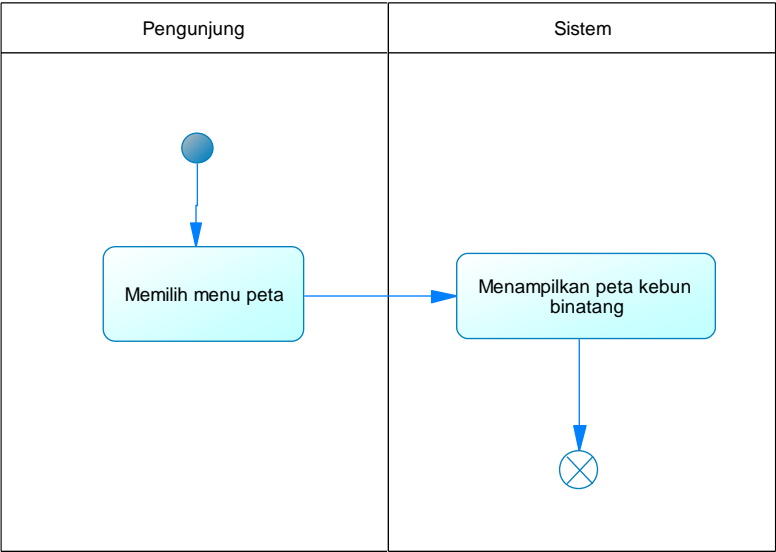
Berdasarkan Diagram Kasus Penggunaan yang telah dijelaskan sebelumnya, berikut merupakan deskripsi dan diagram aktivitas untuk setiap kasus penggunaan.

3.3.2.1 Melihat Peta Kebun Binatang

Deskripsi kasus penggunaan dan diagram aktivitas untuk melihat peta kebun binatang ditunjukkan pada Tabel 3.6 dan Gambar 3.5.

Tabel 3.6 Deskripsi Kasus Penggunaan Melihat Peta Kebun Binatang

Nama Use Case	Melihat Peta Kebun Binatang
Kode Use Case	UC01
Aktor	Pengunjung
Deskripsi	Pengunjung melihat peta kebun binatang
Use Case Relasi	-
Kondisi Awal	Pengunjung tidak mengetahui peta kebun binatang
Kondisi Akhir	Pengunjung mengetahui peta kebun binatang.
Alur Normal	
Pengunjung	Sistem
1. Pengunjung memilih menu peta.	1.1 Sistem menampilkan peta kebun binatang.
Alur Alternatif	
-	
Eksepsi	
-	



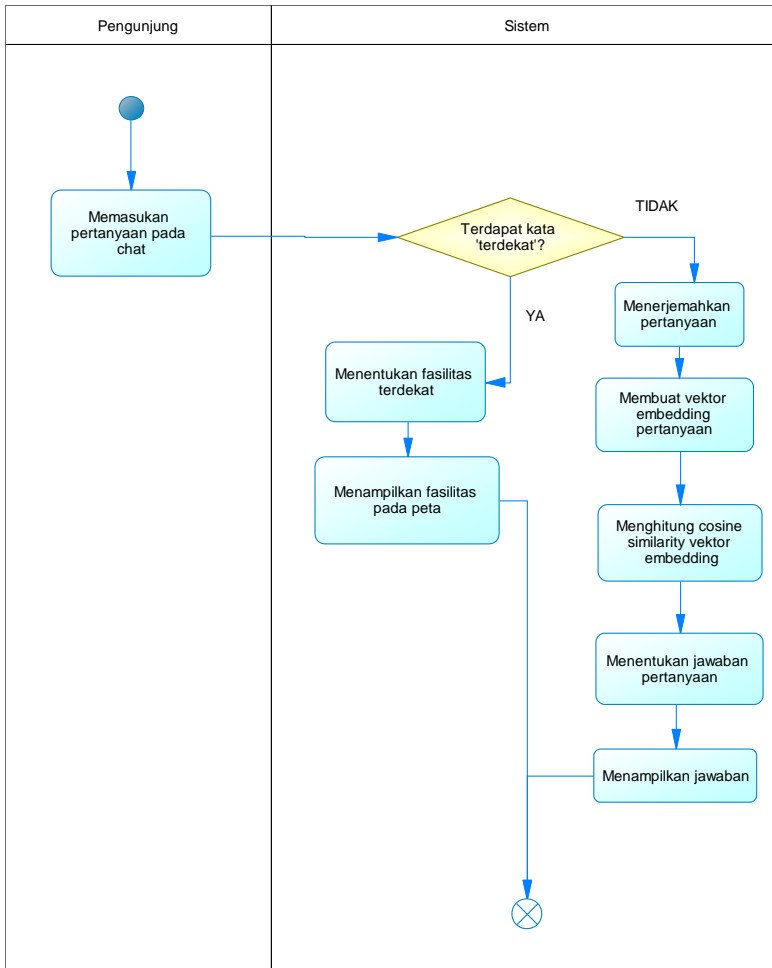
Gambar 3.5 Diagram Aktivitas Melihat Peta Kebun Binatang

3.3.2.2 Menanyakan Pertanyaan pada *Chatbot*

Deskripsi kasus penggunaan dan diagram aktivitas untuk menanyakan pertanyaan pada *chatbot* ditunjukkan pada Tabel 3.7 dan Gambar 3.6.

**Tabel 3.7 Deskripsi Kasus Penggunaan Menanyakan
Pertanyaan pada *Chatbot***

Nama Use Case	Menanyakan Pertanyaan pada <i>Chatbot</i>
Kode Use Case	UC02
Aktor	Pengunjung
Deskripsi	Pengunjung bertanya pada <i>chatbot</i>
Use Case Relasi	-
Kondisi Awal	Pengunjung tidak mengetahui suatu informasi kebun binatang
Kondisi Akhir	Pengunjung mengetahui suatu informasi kebun binatang.
Alur Normal	
Pengunjung	Sistem
1. Pengunjung memasukkan pertanyaan pada chat	1.1 Sistem menerjemahkan pertanyaan pengunjung 1.2 Sistem membuat <i>embedding</i> pertanyaan pengunjung 1.3 Sistem menghitung <i>cosine similarity</i> vektor <i>embedding</i> 1.4 Sistem menentukan jawaban pertanyaan 1.5 Sistem menampilkan jawaban
Alternatif	
1. 1 Pengunjung memasukkan pertanyaan yang mengandung kata 'terdekat' 1.1.1 Sistem menampilkan fasilitas pada peta	
Eksepsi	
-	



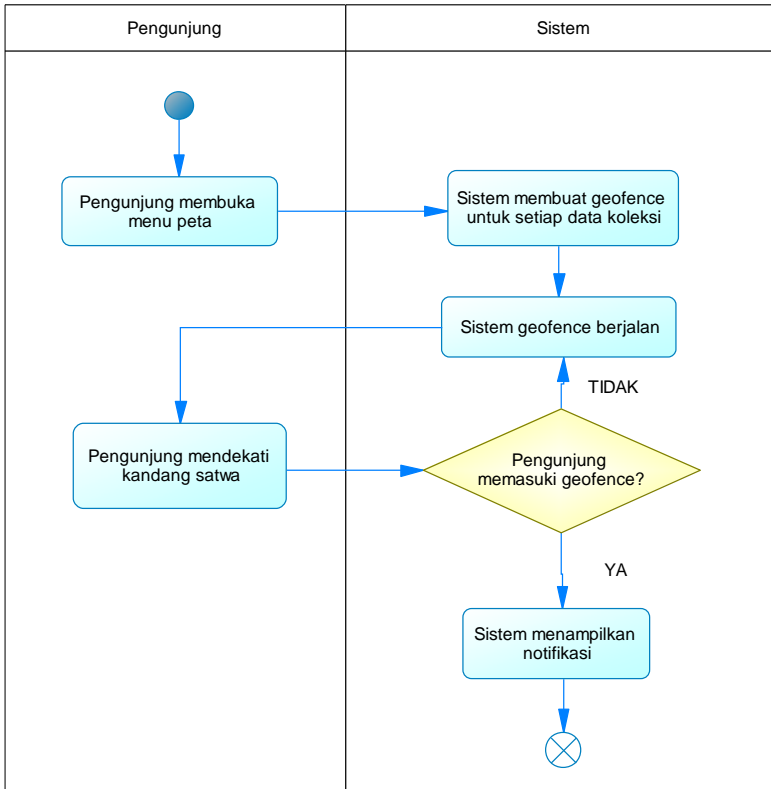
Gambar 3.6 Diagram Aktivitas Menanyakan Pertanyaan pada Chatbot

3.3.2.3 Mendapat Notifikasi informasi Satwa

Deskripsi kasus penggunaan dan diagram aktivitas untuk mendapat notifikasi informasi satwa ditunjukkan pada Tabel 3.8 dan Gambar 3.7.

Tabel 3.8 Deskripsi Kasus Penggunaan Mendapat Notifikasi Informasi Satwa

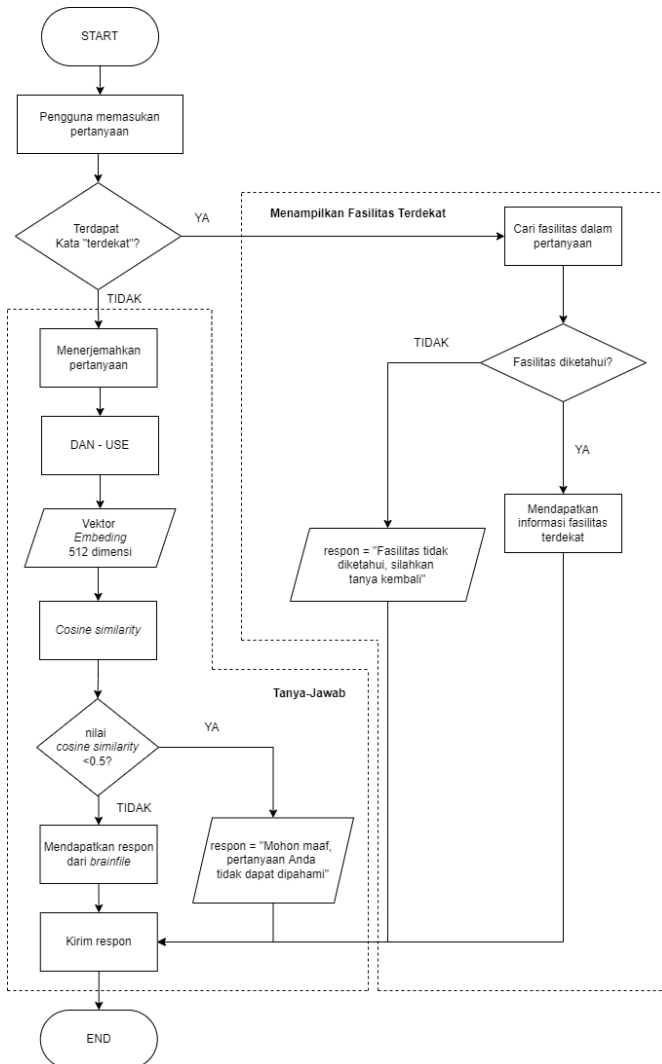
Nama Use Case	Mendapat Notifikasi Informasi Satwa
Kode Use Case	UC03
Aktor	Pengunjung
Deskripsi	Pengunjung mendapat informasi mengenai koleksi kebun binatang yang berada di dekat pengunjung.
Use Case Relasi	-
Trigger	Lokasi pengunjung memasuki area <i>geofence</i> .
Kondisi Awal	Pengunjung membuka halaman peta pada aplikasi.
Kondisi Akhir	Pengunjung mengetahui informasi dari satwa kebun binatang yang ada di dekatnya.
Alur Normal	
Pengunjung	Sistem
1. Pengunjung membuka menu peta. 2. Pengunjung mendekati kandang satwa.	1.1 Sistem membuat <i>geofence</i> untuk setiap data koleksi. 2.1 Sistem mendeteksi pengunjung memasuki suatu <i>geofence</i> . 2.2 Sistem menampilkan notifikasi.
Alur Alternatif	
-	
Eksepsi	
-	



Gambar 3.7 Diagram Aktivitas Mendapat Notifikasi Informasi Satwa

3.4 Perancangan Aturan *Chatbot*

Terdapat dua fungsi utama pada fitur *chatbot*, yaitu tanya-jawab dengan pengguna aplikasi dan menampilkan posisi fasilitas yang ditanyakan oleh pengunjung pada peta. Diagram alir dari kedua proses tersebut ditunjukkan oleh Gambar 3.8.



Gambar 3.8 Diagram Alir Aturan Chatbot

Penjelasan dari diagram alir aturan *chatbot* adalah sebagai berikut.

1. *Chatbot* akan melakukan pengecekan terlebih dahulu terhadap *input* pengunjung. Pengecekan *input* pengunjung dilakukan untuk mengetahui apakah pengunjung bertanya mengenai pertanyaan umum atau pertanyaan tentang lokasi fasilitas terdekat.
2. Apabila *input* pengunjung mengandung kata “terdekat” maka *chatbot* akan memproses *input* pengunjung sebagai berikut.
 - a. *Chatbot* melakukan pengecekan terhadap fasilitas yang ditanyakan.
 - b. Apabila salah satu fasilitas yang tersimpan pada basis data ditemukan pada *input* pengunjung, maka menu peta akan terbuka dan menunjukkan lokasi fasilitas yang ditanyakan.
 - c. Apabila tidak ditemukan fasilitas kebun binatang pada *input* pengunjung, maka *chatbot* akan mengirimkan pesan bahwa fasilitas yang ditanyakan tidak diketahui atau tidak sesuai dengan yang terdata pada basis data.
3. Apabila *input* pengunjung tidak mengandung kata “terdekat” maka *chatbot* akan masuk ke dalam alur tanya-jawab dan memproses *input* pengunjung sebagai berikut.
 - a. *Input* pengunjung diterjemahkan ke dalam bahasa Inggris.
 - b. Hasil terjemahan *input* pengunjung dimasukkan ke dalam model *Universal Sentence Encoder* untuk didapatkan vektor *embedding*-nya.
 - c. Model *chatbot* mengembalikan respon berupa jawaban.
 - d. *Chatbot* mengirimkan respon yang didapatkan sebagai jawaban.

Penjelasan lebih detail dari tiap tahapan aturan *chatbot* dijelaskan sebagai berikut.

3.4.1 Menerjemahkan *input* pengunjung ke dalam bahasa Inggris

Proses ini dilakukan apabila *input* pengunjung tidak mengandung kata “terdekat” dan masuk kedalam alur tanya-jawab. Tahapan pertama dari memperoleh jawaban melalui model *Universal Sentence Encoder* adalah menerjemahkan pertanyaan pengunjung ke dalam bahasa Inggris. Proses ini perlu dilakukan karena *Universal Sentence Encoder* adalah *pre-trained model*, yaitu model yang telah dilatih sebelumnya menggunakan data berbahasa Inggris. Pada proses penerjemahan pertanyaan pengunjung, digunakan bahasa pemrograman Python dan *library* *googletrans*. Dengan menggunakan *library googletrans*, pengunjung dapat menggunakan bahasa apapun yang didukung oleh *googletrans* sebagai *input*. Keluaran dari proses ini adalah variabel string berisi pertanyaan pengunjung yang sudah diterjemahkan.

3.4.2 *Encoding* menggunakan *Universal Sentence Encoder*

Setelah menerjemahkan *input* pengunjung, tahap selanjutnya adalah memperoleh vektor *embedding*. Pada tahapan ini, dilakukan perubahan pertanyaan pengunjung menjadi vektor *embedding* 512 dimensi menggunakan bahasa pemrograman Python. Agar dapat mengintegrasikan bahasa pemrograman Java dan Python, digunakan *Chaquopy SDK*. *Chaquopy* memungkinkan untuk menerapkan *library* dan kebutuhan untuk integrasi antara Python *machine learning* dengan aplikasi android. Masukan dari pengunjung yang telah diterjemahkan diubah menjadi vektor *embedding* dengan tujuan agar masukan pengunjung dapat dihitung kemiripannya dengan vektor *embedding* pertanyaan pada *brainfile* menggunakan *cosine similarity* sehingga didapatkan jawaban yang sesuai berdasarkan kemiripan tersebut. Tahap perolehan jawaban akan dijelaskan secara lengkap pada tahap selanjutnya. *Universal Sentence Encoder* digunakan dalam penelitian ini karena model ini dapat melakukan *embedding*

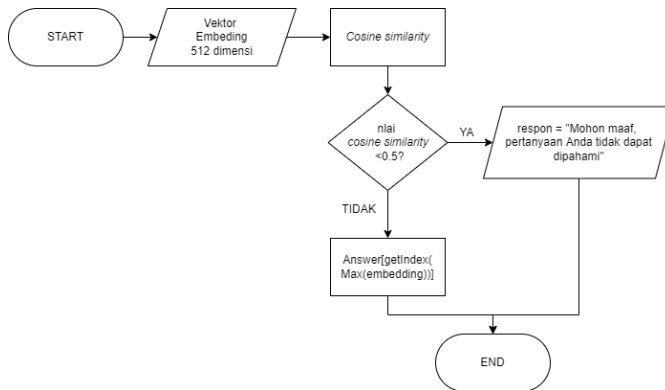
kalimat tanpa menghilangkan makna dan informasi dari kalimat yang dimasukkan. Selain itu model *Universal Sentence Encoder* sudah termasuk dengan tahapan *preprocessing* dengan mengubah seluruh huruf menjadi huruf kecil serta tokenisasi menggunakan *Penn TreeBank Tokenization* sehingga tidak perlu dilakukan *preprocessing* terlebih dahulu. Namun, pada versi ringan, perlu dilakukan perubahan format menjadi format yang mirip dengan format sparse tensor. Variasi *Universal Sentence Encoder* yang digunakan pada proses embedding ini adalah versi ringan dari *Deep Averaging Network* (DAN). Beberapa pertimbangan dari penggunaan versi ringan model *Universal Sentence Encoder* tersebut adalah sebagai berikut.

1. Model *Deep Averaging Network* dan *Transformer* dari *Universal Sentence Encoder* pada tensorflow memiliki ukuran yang besar. Apabila model yang digunakan memiliki ukuran yang besar maka akan membutuhkan waktu lebih lama.
2. Tidak tersedia modul-modul yang dibutuhkan untuk mengimplementasikan model *Universal Sentence Encoder* model *Transformer* dan DAN pada Chaquopy.

3.4.3 Tahap Perolehan Jawaban

Pada tahap ini, vektor *embedding* dari pertanyaan pengunjung dan pertanyaan pada *brainfile* dibandingkan dengan pertanyaan pengunjung menggunakan *cosine similarity* (Kemiripan dua vektor bisa diukur menggunakan *cosine similarity*. *Cosine Similarity* menunjukkan kemiripan dua vektor melalui nilai cosine dari kedua vektor tersebut. Rumus dari *cosine similarity* ditunjukkan pada Persamaan 2.1 .) dan menghasilkan vektor baru yang berisi nilai kemiripan berdasarkan *cosine similarity*. Apabila nilai terbesar dari *cosine similarity* pada vektor tersebut bernilai kurang dari 0.5, maka jawaban yang akan dikirimkan akan berupa pernyataan bahwa pertanyaan yang diajukan tidak dapat dipahami oleh *chatbot*. Apabila nilai tertinggi lebih dari atau sama dengan 0.5, maka didapatkan indeks dari vektor yang memiliki nilai

tertinggi tersebut dan kemudian berdasarkan indeks tersebut didapatkan jawaban dari *brainfile*. Nilai *cosine similarity* sebesar 0.5 didapatkan dengan melakukan percobaan dengan memasukkan pertanyaan yang tidak berhubungan dengan kebun binatang. Pembatasan nilai *cosine similarity* dilakukan agar menghindari pertanyaan yang tidak relevan dengan wisata kebun binatang tetapi tetap memungkinkan sistem untuk menghasilkan jawaban untuk pertanyaan yang berhubungan dengan kebun binatang. Diagram alir dari tahap memperoleh jawaban ditunjukkan pada Gambar 3.9.



Gambar 3.9 Diagram Alir Tahap Perolehan Jawaban

3.4.4 Pengecekan Fasilitas pada Basis Data

Apabila ditemukan kata ”terdekat” pada *input* pengunjung, maka proses selanjutnya adalah melakukan pengecekan terhadap *input* pengunjung untuk mengetahui apakah fasilitas yang ditanyakan pengunjung terdapat pada basis data. Proses ini dilanjutkan dengan mendapatkan data-data fasilitas yang ditanyakan oleh pengunjung. Proses tersebut dijelaskan sebagai berikut.

1. Mendapatkan data fasilitas dari database.
2. Mencari nama fasilitas di dalam *input* pengunjung yang. Kedua data tersebut diubah menjadi huruf kecil saat dibandingkan.

3. Apabila ditemukan string yang sama dengan nama fasilitas yang ada pada basis data, maka data fasilitas tersebut dimasukkan ke dalam array baru berisi objek fasilitas.
4. Apabila tidak ditemukan satu pun nama fasilitas pada *input* pengunjung, maka sistem akan mengirimkan pesan bahwa fasilitas tidak diketahui.

3.4.5 Menentukan Fasilitas Terdekat

Apabila fasilitas yang ditanyakan oleh pengunjung terdapat pada basis data, maka langkah selanjutnya adalah menentukan fasilitas yang terdekat dengan pengunjung. Pada tahap ini digunakan perhitungan Haversine (mengacu kepada Persamaan Haversine adalah salah satu metode yang dapat digunakan untuk menghitung jarak antara dua titik pada bumi. Metode Haversine menggunakan posisi pada garis bujur dan garis lintang bumi serta diameter bumi dalam mengukur jarak antar dua titik pada permukaan bumi. Persamaan Haversine ditunjukkan pada Persamaan 2.2.) untuk menentukan jarak terdekat antara lokasi fasilitas dengan pengunjung. Adapun langkah-langkah yang dilakukan ditunjukkan sebagai berikut.

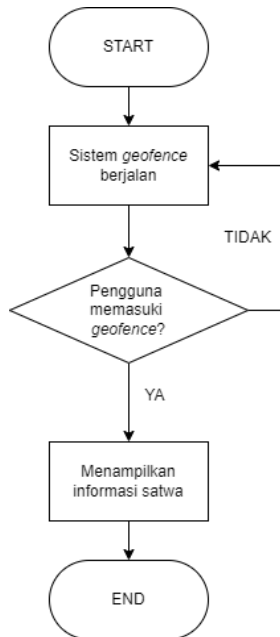
1. Apabila fasilitas yang ditanyakan oleh pengunjung hanya terdapat satu pada basis data, maka berlanjut ke tahap selanjutnya.
2. Apabila tidak, maka dilakukan inisial variabel jarak terdekat dengan nilai 9999.
3. Untuk setiap data fasilitas dilakukan perhitungan jarak menggunakan persamaan Haversine.
4. Apabila jarak yang diperoleh bernilai lebih kecil dibandingkan dengan nilai jarak terdekat, maka nilai jarak terdekat diubah menjadi jarak yang telah didapatkan dan nilai koordinat fasilitas tersebut disimpan untuk digunakan pada tahap selanjutnya.

3.4.6 Menampilkan Posisi dari Fasilitas Terdekat

Setelah didapatkan nilai koordinat dari fasilitas yang terdekat, tahap selanjutnya adalah membuka menu peta dan menampilkan posisi koordinat yang telah diperoleh. Pada tahap ini, data koordinat tersebut dibagi dengan menu peta. Kemudian pada menu peta, posisi kamera untuk peta diarahkan kepada koordinat yang telah didapatkan pada tahap sebelumnya sehingga menunjukkan posisi dari fasilitas terdekat.

3.5 Perancangan Fitur Pemandu

Selain dapat menjawab pertanyaan pengunjung, aplikasi ini juga memiliki fitur pemandu. Fitur pemandu akan memberikan informasi mengenai satwa yang didekati oleh pengunjung. Diagram alir dari rancangan fitur pemandu ditunjukkan pada Gambar 3.10.



Gambar 3.10 Diagram Alir Fitur Pemandu

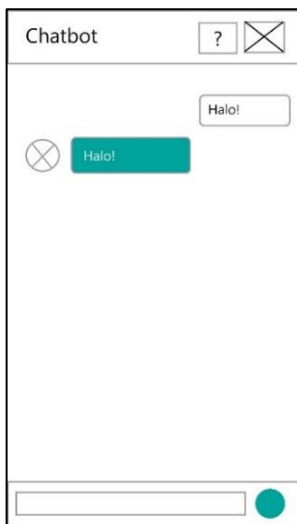
Agar sistem dapat menentukan posisi pengunjung dari kandang satwa kebun binatang, maka dibuat *geofence*. *Geofence* hanya dibuat pada koordinat posisi kandang satwa kebun binatang. Radius yang digunakan pada *geofence* ini adalah 50 meter.

3.6 Perancangan Antarmuka

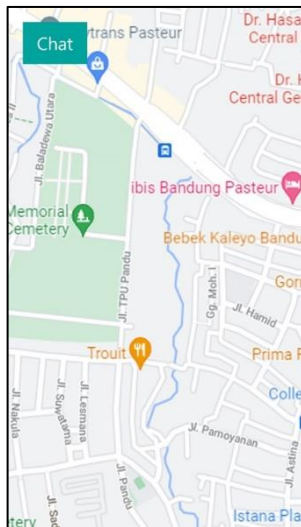
Antarmuka aplikasi dirancang dengan mempertimbangkan kemudahan pengunjung menggunakan aplikasi. Terdapat 2 menu dalam aplikasi, yaitu menu *chatbot* dan menu peta. Menu *chatbot* dijadikan sebagai menu utama dalam aplikasi. Di dalam menu *chatbot*, terdapat *text field* sebagai tempat pengunjung memasukkan pesan dan tombol untuk mengirimkan pesan tersebut. Selain itu, terdapat tombol dengan simbol tanda tanya untuk

membuka *dialog* bantuan. Desain antarmuka untuk menu *chatbot* ditunjukkan pada Gambar 3.11.

Pada rancangan ini, pengunjung dapat mengakses menu peta dengan menekan tombol peta pada menu *chatbot* yang terletak di pojok kanan atas *toolbar* menu. Pengunjung juga dapat kembali ke menu *chatbot* dengan menekan tombol kembali pada *handphone* atau menekan tombol ‘Chat’. Rancangan dari menu peta ditunjukkan pada Gambar 3.12.



Gambar 3.11 Rancangan Antarmuka Menu Chatbot



Gambar 3.12 Rancangan Antarmuka Menu Peta

BAB IV IMPLEMENTASI

Bab ini menjelaskan mengenai implementasi sistem berdasarkan rancangan sistem yang telah dibuat. Implementasi yang dijelaskan adalah lingkungan implementasi, pengambilan data, dan implementasi rancangan sistem.

4.1 Lingkungan Implementasi

Pada pelaksanaan tugas akhir ini, digunakan perangkat keras dan perangkat lunak untuk membangun aplikasi yang telah dirancang sebelumnya. Spesifikasi dari perangkat keras dan perangkat lunak yang digunakan ditunjukkan pada Tabel 4.1.

Tabel 4.1 Spesifikasi Lingkungan Implementasi

No.	Jenis Perangkat	Spesifikasi
1	Perangkat Keras	<ul style="list-style-type: none">• <i>Processor</i>: Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz• <i>Memory</i>: 8 GB
2	Perangkat Lunak	<ul style="list-style-type: none">• <i>Operating System</i>: Windows 10 Home• Android Studio Arctic Fox 2020.3.1

4.2 Pengambilan Data

Sebelum memasuki tahap implementasi, dilakukan pengumpulan data untuk dianalisa dan diolah sehingga dapat digunakan sebagai data *brainfile* dan data lainnya. Data-data yang digunakan antara lain yaitu:

1. Data satwa di kebun binatang.
2. Data koordinat lokasi fasilitas dan kandang satwa di kebun binatang.

3. Data mengenai kebun binatang yang dijadikan objek dalam tugas akhir ini.

Data-data tersebut diperoleh melalui beberapa metode berikut.

1. Observasi secara langsung di kebun binatang.
2. Wawancara informal terhadap pengunjung kebun binatang.
3. Website resmi dari kebun binatang .

Penyusunan data *brainfile* dilakukan dengan membuat daftar pertanyaan-pertanyaan yang berkaitan dengan konteks sebagai berikut.

1. Pertanyaan umum seperti sapaan dan informasi mengenai aplikasi.
2. Pertanyaan mengenai kebun binatang.
3. Pertanyaan mengenai satwa di kebun binatang.

Kemudian jawaban untuk pertanyaan tersebut dibuat berdasarkan informasi yang telah didapatkan dari pengambilan data. Jumlah data yang dihasilkan dari pengambilan data tersebut adalah sebagai berikut.

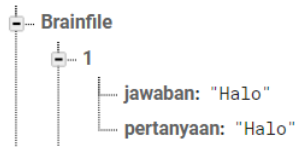
Data <i>Brainfile</i>	: 300 data
Data Koleksi	: 20 data
Data Fasilitas	: 25 data

4.3 Implementasi

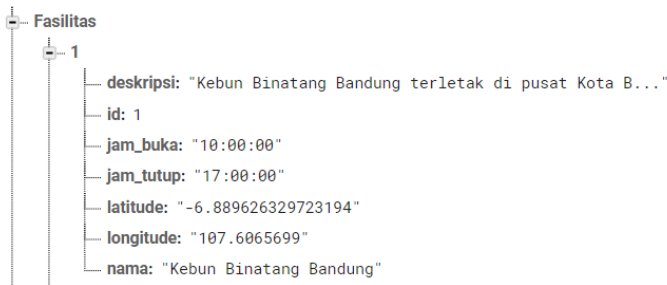
Bab ini menjelaskan implementasi sistem berdasarkan analisis dan perancangan sistem yang telah dijelaskan pada bab sebelumnya.

4.3.1 Implementasi Basis Data

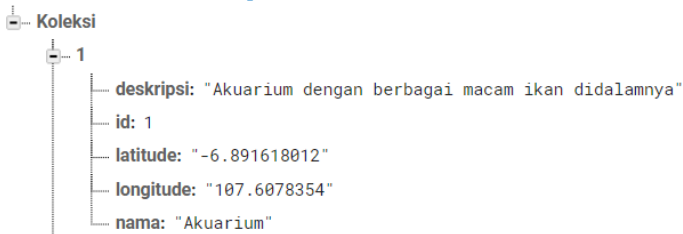
Sesuai dengan perancangan basis data, implementasi basis data untuk sistem mengimplementasikan *realtime database* milik Firebase. Contoh hasil dari implementasi basis data untuk aplikasi ini ditunjukkan oleh Gambar 4.1, Gambar 4.2, dan Gambar 4.3.



Gambar 4.1 Implementasi Node Brainfile



Gambar 4.2 Implementasi Node Fasilitas



Gambar 4.3 Implementasi Node Koleksi

4.3.2 Implementasi Kelas Fasilitas

Kelas Fasilitas digunakan untuk mengelola data fasilitas. Kelas ini memiliki atribut nama, latitude, longitude, deskripsi, jam_buka, dan jam_tutup. Atribut tersebut menyesuaikan dengan rancangan *node* Fasilitas pada basis data. *Method* dalam kelas ini terdiri dari *method set* dan *get* pada setiap atribut, *method* konstruktor, serta *method getCoordinate()* untuk mendapatkan nilai atribut latitude dan longitude sekaligus. Adapun implementasi dari kelas Fasilitas dengan menggunakan bahasa pemrograman Java ditunjukkan pada Kode Sumber 4.1.

```

public class Fasilitas {
    private String nama;
    private String latitude;
    private String longitude;
    private String deskripsi;
    private String jam_buka;
    private String jam_tutup;

    public Fasilitas(){}
    public Fasilitas(String nama, String latitude, String
longitude, String jam_buka, String jam_tutup){
        this.nama = nama;
        this.latitude = latitude;
        this.longitude = longitude;
        this.jam_buka = jam_buka;
        this.jam_tutup = jam_tutup;
    }
    public String getJam_buka() {
        return jam_buka;
    }
    public void setJam_buka(String jam_buka) {
        this.jam_buka = jam_buka;
    }
    public String getJam_tutup() {
        return jam_tutup;
    }
    public void setJam_tutup(String jam_tutup) {
        this.jam_tutup = jam_tutup;
    }
    public String getNama() {
        return nama;
    }
    public void setNama(String nama) {
        this.nama = nama;
    }
    public Double getLatitude() {
        return Double.valueOf(latitude);
    }
    public void setLatitude(String latitude) {
        this.latitude = latitude;
    }
    public Double getLongitude() {
        return Double.valueOf(longitude);
    }
    public String getDeskripsi() {
        return deskripsi;
    }
}

```

```
public void setDeskripsi(String deskripsi) {  
    this.deskripsi = deskripsi;  
}  
public void setLongitude(String longitude) {  
    this.longitude = longitude;  
}  
public String[] getCoordinate(){  
    String[] coordinate= {this.latitude,  
this.longitude};  
    return coordinate;  
}  
}
```

Kode Sumber 4.1 Kelas Fasilitas

4.3.3 Implementasi Kelas Koleksi

Kelas Koleksi digunakan untuk membantu mengelola data koleksi satwa pada basis data. Atribut pada kelas ini menyesuaikan dengan perancangan basis data. Atribut pada kelas ini terdiri dari nama, latitude, longitude, dan deskripsi. *Method* yang digunakan pada kelas ini meliputi *method set* dan *get* untuk setiap atribut, *method* konstruktor, dan *method getCoordinate()* untuk mendapatkan nilai titik koordinat. Implementasi untuk kelas ini ditunjukkan pada Kode Sumber 4.2.

```

public class Koleksi{
    private String nama;
    private String latitude;
    private String longitude;
    private String deskripsi;

    public Koleksi(){}
    public Koleksi(String nama, String latitude, String
longitude, Blob foto){
        this.nama = nama;
        this.latitude = latitude;
        this.longitude = longitude;
    }
    public String getNama() {
        return nama;
    }
    public void setNama(String nama) {
        this.nama = nama;
    }
    public Double getLatitude() {
        return Double.valueOf(latitude);
    }
    public void setLatitude(String latitude) {
        this.latitude = latitude;
    }
    public Double getLongitude() {
        return Double.valueOf(longitude);
    }
    public String getDeskripsi() {
        return deskripsi;
    }
    public void setDeskripsi(String deskripsi) {
        this.deskripsi = deskripsi;
    }
    public void setLongitude(String longitude) {
        this.longitude = longitude;
    }
    public String[] getCoordinate(){
        String[] coordinate= {this.latitude,
this.longitude};
        return coordinate;
    }
}

```

Kode Sumber 4.2 Kelas Koleksi

4.3.4 Implementasi Pengecekan Input *Chatbot*

Untuk menentukan apakah pengunjung menanyakan lokasi dari suatu fasilitas atau menanyakan pertanyaan umum, digunakan fungsi *isAskingLocation()*. Fungsi tersebut mengecek apabila *input* pengunjung mengandung kata “terdekat”. Implementasi dari fungsi *isAskingLocation()* ditunjukkan pada Kode Sumber 4.3.

```
private Boolean isAskingLocation(String message){
    return message.toLowerCase().contains("terdekat");
}
```

Kode Sumber 4.3 Fungsi *isAskingLocation()*

4.3.5 Implementasi Penerjemahan

Implementasi penerjemahan *input* pengunjung menggunakan bahasa pemrograman Python dan *library* *googletrans*. Untuk menerjemahkan *input* pengunjung, dibuat sebuah fungsi *trans_en()* yang mengembalikan string berupa hasil terjemahan *input* pengunjung. Kode untuk penerjemahan disimpan secara tersendiri dengan format file *.py*. Implementasi tersebut dapat dilihat dalam Kode Sumber 4.4.

```
from googletrans import Translator
translator = Translator()
def trans_en(query):
    result = translator.translate(query, dest='en')
    return str((result.text))
```

Kode Sumber 4.4 Implementasi Penerjemahan

4.3.6 Implementasi Inisialisasi Model

Tahapan pertama dari proses pengolahan terjemahan *input* pengunjung menjadi jawaban adalah mendapatkan data *brainfile* dari database. Proses ini menggunakan bahasa pemrograman Python serta *library* *Firestore*. Data pertanyaan dan jawaban dibagi menjadi 2 array yang berbeda. Proses pengambilan data ini ditunjukkan pada Kode Sumber 4.5.


```

From firebase import firebase

firebase =
firebase.FirebaseApplication("https://xxxxxxzoochatbot-
default-rtdb.firebaseio.com/", None)
brainfile_data = firebase.get('/Brainfile/', '')
pertanyaan = []
jawaban = []

for data in brainfile_data:
    qna = data
    if(type(qna) is dict):
        pertanyaan.append(qna.get('pertanyaan'))
        jawaban.append(qna.get('jawaban'))

```

Kode Sumber 4.5 Pengambilan Data dari Firebase

Sebelum aplikasi dapat digunakan, dilakukan inisialisasi untuk model SentencePiece. SentencePiece digunakan untuk memproses *input* pengguna sebelum dapat diolah menggunakan model USE. Proses ini ditunjukkan pada Kode Sumber 4.6.

```

NO_ANSWER = "Mohon maaf pertanyaan anda tidak dapat
dipahami, silahkan tanya kembali"

module = hub.Module("https://tfhub.dev/google/universal-
sentence-encoder-lite/2")

input_placeholder = tf.sparse_placeholder(tf.int64,
shape=[None, None])
encodings = module(
    inputs=dict(        values=input_placeholder.values,
                        indices=input_placeholder.indices,
                        dense_shape=input_placeholder.dense_shape))

with tf.Session() as sess:
    spm_path = sess.run(module(signature="spm_path"))

sp = spm.SentencePieceProcessor()
with tf.io.gfile.GFile(spm_path, mode="rb") as f:
    sp.LoadFromSerializedProto(f.read())

```

Kode Sumber 4.6 Inisialisasi Model Chatbot

Implementasi dari inisialisasi model hingga mendapatkan jawaban disimpan dalam satu file yang sama dengan format .py.

4.3.7 Implementasi Perolehan Jawaban

Untuk mendapatkan jawaban dari pertanyaan pengunjung yang tidak menanyakan lokasi dari suatu fasilitas, digunakan fungsi *getAnswer()*. Fungsi tersebut menghasilkan data jawaban dalam bentuk string dan digunakan ketika sistem menerima *input* pengunjung dari *chatbot* yang tidak menanyakan lokasi fasilitas. Fungsi ini menerima data terjemahan dari *input* pengunjung berbentuk string sebagai argumen.

Pada fungsi ini, daftar pertanyaan dari *brainfile* disalin ke array baru, yaitu *new_questions*. Selanjutnya, pertanyaan pengunjung yang sudah diterjemahkan juga dimasukkan ke dalam array tersebut. Array berisi gabungan dari pertanyaan dari *brainfile* dan pengunjung tersebut kemudian diubah menjadi format yang serupa dengan format sparse tensor melalui fungsi *getSparseTensorFormat()* sehingga dapat diproses oleh *Universal Sentence Encoder*. Hasil dari perubahan array *new_questions* tersebut disimpan pada variabel *values*, *indices*, dan *dense_shape*.

```
def getSparseTensorFormat(sp, sentences):
    ids = [sp.EncodeAsIds(x) for x in sentences]
    max_len = max(len(x) for x in ids)
    dense_shape=(len(ids), max_len)
    values=[item for sublist in ids for item in sublist]
    indices=[[row,col] for row in range(len(ids)) for col in
range(len(ids[row]))]
    return (values, indices, dense_shape)
```

Kode Sumber 4.7 Fungsi *getSparseTensorFormat()*

Selanjutnya, dengan menggunakan *encoder USE*, *embedding* dari kalimat-kalimat pertanyaan tersebut diubah menjadi array berukuran 512 sebanyak pertanyaan pada variabel *new_question*. Kemudian dilakukan perhitungan *cosine similarity* pada array *embedding*. Perhitungan *cosine similarity* tersebut dilakukan dengan membandingkan array *embedding* pertanyaan dari *brainfile* dengan *embedding* kalimat pertanyaan dari *input*

pengunjung. Proses tersebut menghasilkan array yang berisi nilai *cosine similarity* dari pertanyaan pengunjung dengan pertanyaan dari *brainfile*. Contoh hasil dari perhitungan tersebut dapat dilihat pada Gambar 4.4.

```
[[ 1.          -0.04581382  0.0476578  ...  0.12155358  0.12947093
  -0.04302795]
 [-0.04581382  1.0000001  0.28920168 ...  0.2510021  0.27444535
  0.25636193]
 [ 0.0476578  0.28920168  1.0000001  ...  0.39101467  0.2752145
  0.37214565]
 ...
 [ 0.12947093  0.27444535  0.2752145  ...  0.52975273  0.9999999
  0.24225992]
 [-0.04302795  0.25636193  0.37214565 ...  0.44445506  0.24225992
  1.          ]
 [ 0.1194644  0.2939394  0.58586466 ...  0.51150185  0.32965747
  0.37292677]]
```

Gambar 4.4 Contoh Hasil Perhitungan Cosine Similarity

Dapat dilihat pada gambar tersebut bahwa perhitungan tersebut tidak membandingkan data pertanyaan *input* pengunjung sebagai data terakhir dengan data *input* pengunjung sendiri (n, n). Selanjutnya dilakukan pengecekan nilai tertinggi pada data terakhir dari vektor yang merupakan perbandingan data *input* pengunjung dengan seluruh data pada *brainfile* ($\max(\text{cosine}[-I])$).

Apabila nilai tertinggi pada array cosine dibawah 0.5, maka fungsi `getAnswer()` akan mengembalikan data string NO_ANSWER. String tersebut berisi respon yang memberitahu pengunjung bahwa pertanyaan yang dimasukkan tidak dapat dipahami oleh *chatbot*. Hal tersebut dilakukan untuk menghindari jawaban yang tidak relevan akibat data yang kurang maupun pertanyaan yang tidak berkaitan dengan kebun binatang. Apabila nilai tertinggi dari *cosine similarity* lebih atau sama dengan 0.5, maka fungsi ini akan mengembalikan jawaban dengan cara mengambil data jawaban berdasarkan indeks dari data pada array cosine yang memiliki nilai tertinggi. Implementasi untuk fungsi ini ditunjukkan pada Kode Sumber 4.8.

```

def getAnswer(q_user):
    new_questions = pertanyaan.copy()
    new_questions.append(q_user)
    values, indices, dense_shape = getTensorSparseFormat(sp,
new_questions)
    with tf.Session() as session:
        session.run([tf.global_variables_initializer(),
tf.tables_initializer()])
        message_embeddings = session.run(
            encodings,
            feed_dict={input_placeholder.values: values,
                        input_placeholder.indices: indices,
                        input_placeholder.dense_shape:
dense_shape})
        cosine = cosine_similarity(message_embeddings,
message_embeddings[: -1])
        del message_embeddings
        del new_questions
        del values, indices, dense_shape
        if (max(cosine[-1]) < 0.5):
            return NO_ANSWER
        else:
            return jawaban[np.argmax((cosine[-1]))]

```

Kode Sumber 4.8 Fungsi *getAnswer()*

4.3.8 Implementasi Pengecekan Fasilitas pada Basis Data

Apabila pengunjung terdeteksi menanyakan lokasi dari suatu fasilitas, maka perlu dilakukan pengecekan fasilitas yang ditanyakan pada basis data. Untuk melakukan pengecekan tersebut, digunakan sebuah fungsi *isExistinDatabase()*. Fungsi ini melakukan pengecekan dengan membandingkan nama fasilitas pada basis data dengan query pengunjung. Implementasi tersebut ditunjukkan pada Kode Sumber 4.9.

```

private Boolean isExistinDatabase(List<Fasilitas>
fasilitas, String query){
    for(int i=0;i<fasilitas.size();i++){

if(query.toLowerCase().contains(fasilitas.get(i).getNama()
.toLowerCase())){
        Log.i("CEK_LOKASI",
fasilitas.get(i).getNama());
        return true;
    }
    }
    return false;
}

```

Kode Sumber 4.9 Fungsi *isExistinDatabase()*

Penggunaan fungsi *isExistinDatabase()* tersebut kemudian digunakan pada Kode Sumber 4.10 Mendapatkan Daftar Fasilitas yang Ditanyakan.

```

if (isAskingLocation(message)) {
if (isExistinDatabase(fasilitasList, message)) {
    if (!isGPSEnabled()) {showGPSAlert();}
else {
    checkLocationPermission();
    locationManager.requestLocationUpdates(
        locationManager.GPS_PROVIDER,
        1000, 0, new LocationListener() {
            @Override
            public void
onLocationChanged(Location location) {
                if(location == null){
                    showLocationRequestFailed();}
            else {
                LatLong[0] = location.getLatitude();
                LatLong[1] = location.getLongitude();
                ArrayList<Fasilitas> fasilitas1 = new ArrayList<>();
                for (int i = 0; i <
fasilitasList.size(); i++) {
                    if
(message.toLowerCase().contains(fasilitasList.get(i).getNa
ma().toLowerCase()))
{fasilitas1.add(fasilitasList.get(i));}}

```

Kode Sumber 4.10 Mendapatkan Daftar Fasilitas yang Ditanyakan

Apabila terdapat nama fasilitas dalam *input* pengunjung, maka selanjutnya sistem melakukan pengecekan GPS pengunjung. Apabila GPS pengunjung tidak aktif, maka sistem akan menampilkan peringatan berupa objek dialog. Apabila GPS pengunjung menyala, maka sistem akan mengirimkan permintaan lokasi pengunjung dan mendapatkan posisi pengunjung menggunakan LocationManager. Selanjutnya sistem akan menyalin data fasilitas yang namanya terdapat pada *input* pengunjung ke dalam variabel baru yang berupa ArrayList objek fasilitas.

4.3.9 Implementasi Penentuan Fasilitas Terdekat

Setelah didapatkan ArrayList objek fasilitas yang ditanyakan, proses selanjutnya adalah melakukan perhitungan untuk mendapatkan fasilitas terdekat dengan pengunjung. Proses perhitungan tersebut dibantu dengan fungsi *getClosestFacility()* yang ditunjukkan pada Kode Sumber 4.11.

```

private LatLng getClosestFacility(List<Fasilitas>
fasilitas, double[] latLong){
    LatLng latLongFacility = null;
    if(fasilitas.size()==1){
        return latLongFacility = new
LatLng(fasilitas.get(0).getLatitude(),
fasilitas.get(0).getLongitude());
    }
    Long id = null;
    double dLat;
    double dLon;
    double closestDistance = 9999;
    double distance;
    for(int i=0;i<fasilitas.size();i++) {
        dLat =
Math.toRadians(fasilitas.get(i).getLatitude() -
latLong[0]);
        dLon =
Math.toRadians(fasilitas.get(i).getLongitude() -
latLong[1]);
        distance = RADIUS * 2 *
            Math.asin(
                Math.sqrt(
                    Math.pow(Math.sin(dLat/2),2) +
Math.cos(Math.toRadians(latLong[0])) *
Math.cos(Math.toRadians(fasilitas.get(i).getLatitude())) *
Math.pow(Math.sin(dLon/2),2)));
        if(distance < closestDistance){
            latLongFacility = new
LatLng(fasilitas.get(i).getLatitude(),
fasilitas.get(i).getLongitude());
        }
    }
    return latLongFacility;
}

```

Kode Sumber 4.11 Fungsi *getClosestFacility()*

Sesuai dengan rancangan *chatbot* yang telah dijelaskan pada bab sebelumnya, penentuan lokasi terdekat ini menggunakan perhitungan Haversine. Fungsi *getClosestFacility()* menerima argumen berupa ArrayList objek Fasilitas yang berisi fasilitas yang ditanyakan serta array yang berisi koordinat pengunjung. Apabila ArrayList dari objek Fasilitas tidak berisi lebih dari 1 objek, maka fungsi ini akan mengembalikan objek berupa LatLng yang berisi

data koordinat dari fasilitas tersebut. Apabila terdapat lebih dari dua data, maka dilakukan perhitungan Haversine antara seluruh data koordinat fasilitas dengan koordinat pengunjung. Setiap iterasi perhitungan jarak tersebut, dilakukan perbandingan antara jarak fasilitas dengan nilai jarak terdekat sehingga setelah dilakukan perhitungan terhadap seluruh data fasilitas, didapatkan nilai jarak terdekat (*closestDistance*) dan koordinat dari fasilitas tersebut (*latLongFacility*).

4.3.10 Implementasi Menampilkan Fasilitas

Berdasarkan koordinat yang telah didapatkan melalui fungsi *getClosestFacility()*, menu peta dibuka dan menunjukkan lokasi dari koordinat tersebut. Nilai koordinat tersebut dikirim ke MapActivity untuk ditampilkan. Implementasi dari proses tersebut pada kelas *Chatbot* ditujukan oleh **Error! Reference source not**

```
private void showFacilityinMap(LatLng latLng) {
    Intent intent = new Intent(Chatbot.this,
    MapsActivity.class);
    intent.putExtra("COORDINATE_FACILITY", latLng);
    startActivity(intent);}
```

found..

Kode Sumber 4.12 Fungsi *showFacilityinMap()*

Data koordinat tersebut kemudian diproses pada MapsActivity. MapsActivity pada fungsi *onMapReady()* akan mengecek apabila terdapat *extras*. Apabila tidak NULL, maka sistem akan mendapatkan nilai *extras* berupa koordinat dengan kunci “COORDINATE_FACILITY”. Apabila tidak ditemukan *extras* maka sistem akan menyimpulkan bahwa pengunjung membuka peta melalui tombol peta. Dengan menggunakan nilai koordinat yang telah didapatkan, kamera pada peta akan diarahkan ke lokasi dari fasilitas tersebut. Implementasi menampilkan lokasi fasilitas pada kelas MapActivity ditunjukkan pada **Error! Reference source not found..**


```

Bundle extras = getIntent().getExtras();
if (extras != null) {
    LatLng latLngFacility =
    extras.getParcelable("COORDINATE_FACILITY");
    mMap.moveCamera(CameraUpdateFactory.newLatLng(latLngFacility));
    mMap.setMinZoomPreference(ZOOM_PREFERENCE);
}

```

Kode Sumber 4.13 Menampilkan Posisi dari Koordinat

4.3.11 Implementasi Geofencing

Untuk mengimplementasikan *geofence*, dibuat sebuah kelas bernama *GeofenceHelper* untuk memudahkan dalam pembuatan *Geofence*. Agar notifikasi dari *Geofence* menghasilkan pemberitahuan dengan text yang berbeda, maka *requestcode* diisi dengan id tiap data Koleksi. Implementasi kelas *GeofenceHelper* ditunjukkan pada Kode Sumber 4.14.

```

public class GeofenceHelper extends ContextWrapper {

    private float GEOFENCE_RADIUS = 50;
    PendingIntent pendingIntent;

    public GeofenceHelper(Context base) {
        super(base);
    }

    public GeofencingRequest setGeofencingRequest(Geofence
    Geofence) {
        return new GeofencingRequest.Builder()
            .addGeofence(Geofence)
            .setInitialTrigger(GeofencingRequest.INITI
            AL_TRIGGER_ENTER)
            .build();
    }

    public Geofence setGeofence(String id, LatLng latLng){
        return new Geofence.Builder()
            .setRequestId(id)
            .setExpirationDuration(Geofence.NEVER_EXPI
            RE)
            .setCircularRegion(latLng.latitude,
            latLng.longitude, GEOFENCE_RADIUS)
            .setTransitionTypes(Geofence.GEOFENCE_TRAN

```

```

        SITION_ENTER)
            .build();
    }

    public PendingIntent getGeofencePendingIntent (Koleksi
koleksi)
    {
        if (pendingIntent != null)
            return pendingIntent;
    }
    Intent intent = new Intent(this,
GeofenceBot.class);
    intent.putExtra("NAMA_HEWAN", koleksi.getNama());
    intent.putExtra("DESKRIPSI",
koleksi.getDeskripsi());
    return PendingIntent.getBroadcast(this,
koleksi.getId(), intent,
PendingIntent.FLAG_UPDATE_CURRENT);
}

    public String getError(Exception exception){
        if(exception instanceof ApiException){
            ApiException apiException = (ApiException)
exception;
            switch (apiException.getStatusCode()) {
                case GeofenceStatusCodes
                .API_NOT_CONNECTED:
                    return "API NOT CONNECTED";
                case GeofenceStatusCodes
                .GEOFENCE_NOT_AVAILABLE:
                    return "GEOFENCE NOT AVAILABLE";
                case GeofenceStatusCodes
                .GEOFENCE_REQUEST_TOO_FREQUENT:
                    return "GEOFENCE REQUEST TOO
FREQUENT";
            }
        }
        return exception.getMessage();
    }
}

```

Kode Sumber 4.14 Kelas GeofenceHelper

4.3.12 Implementasi Notifikasi Geofencing

Kelas *GeofenceBot* adalah sebuah kelas yang memperluas *BroadcastReceiver* yang memiliki fungsi untuk menampilkan pesan apabila pengunjung diketahui memasuki wilayah *geofence*. Notifikasi dari *event* pada *geofence* diimplementasikan dalam

berbentuk objek *Toast*. Implementasi dari kelas *GeofenceBot* ditunjukkan oleh Kode Sumber 4.15.

```
public class GeofenceBot extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent)
    {
        if(intent.getStringExtra("NAMA_HEWAN")!=null){
            Toast.makeText(context,
            intent.getStringExtra("NAMA_HEWAN")+"\n"+intent.getStrinGExtra("DESKRIPSI"), Toast.LENGTH_LONG).show();
        }
        else{
            Toast.makeText(context, "Failed to get data",
            Toast.LENGTH_SHORT).show();}
    }
}
```

Kode Sumber 4.15 Kelas GeofenceBot

4.3.13 Implementasi Antarmuka

Berikut merupakan hasil dari implementasi antarmuka aplikasi untuk tugas akhir ini. Terdapat tiga implementasi antarmuka, yaitu antarmuka *splash screen*, antarmuka menu chat, dan antarmuka menu peta.

4.3.13.1 Antarmuka *Splash Screen*

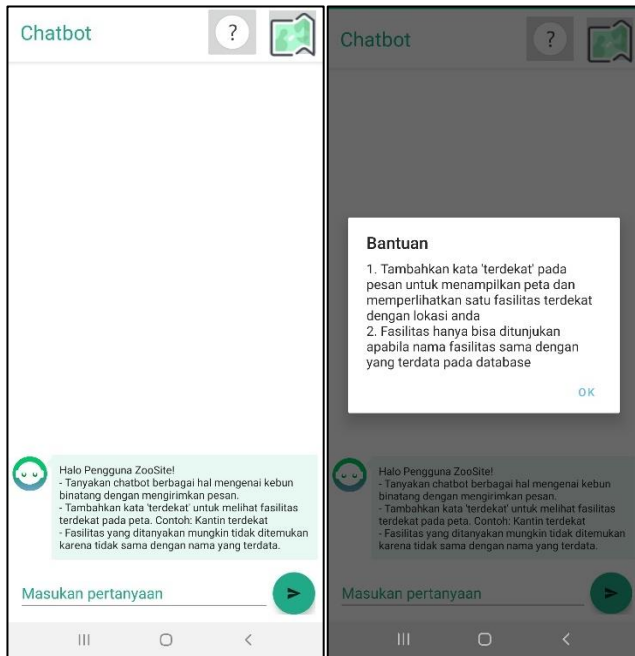
Splash screen muncul saat pengunjung membuka aplikasi. Tampilan dari *splash screen* tersebut ditunjukkan pada Gambar 4.5.



Gambar 4.5 Antarmuka Splash Screen

4.3.13.2 Antarmuka Menu *Chatbot*

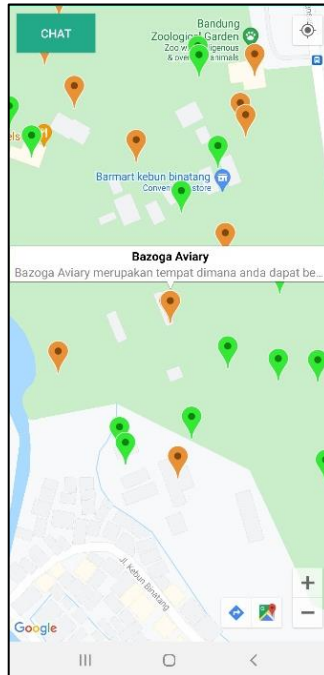
Tampilan dari implementasi dari menu *chatbot* ditunjukkan oleh Gambar 4.6.



Gambar 4.6 Antarmuka Menu *Chatbot*

4.3.13.3 Antar Muka Menu Peta

Menu peta memuat peta dan Hasil implementasi antarmuka dari menu peta ditunjukkan pada Gambar 4.7.



Gambar 4.7 Antarmuka Menu Peta

[Halaman ini sengaja dikosongkan]

BAB V

UJI COBA DAN EVALUASI

Bab ini menjelaskan uji coba dan evaluasi dari sistem yang telah dibangun untuk tugas akhir ini. Setelah dilakukan pengujian terhadap sistem, hasil dari uji coba tersebut kemudian dievaluasi untuk didapatkan kesimpulan.

5.1 Lingkungan Uji Coba

Pada tugas akhir ini, digunakan beberapa perangkat serta pustaka yang berfungsi untuk proses uji coba. Spesifikasi perangkat keras dan perangkat lunak yang digunakan dapat dilihat pada Tabel 5.1.

Tabel 5.1 Spesifikasi Lingkungan Uji Coba

No.	Jenis Perangkat	Spesifikasi
1.	Perangkat Keras	<ul style="list-style-type: none"> • Samsung Galaxy A30 • <i>Processor</i>: Exynos 7904 • <i>Memory</i>: 4GB
2.	Perangkat Lunak	<ul style="list-style-type: none"> • <i>Operating System</i>: Android 10

5.2 Pengujian

Subbab ini menjelaskan proses, skenario, dan hasil uji coba terhadap sistem yang telah dibangun. Pengujian yang dilakukan adalah uji coba fungsionalitas sistem, uji coba kerelevanan jawaban *chatbot*, dan *system usability scale*.

5.2.1 Pengujian Fungsionalitas Sistem

Pengujian fungsionalitas sistem dilakukan untuk menguji fungsionalitas dan kelayakan aplikasi yang dibangun. Pengujian ini dilakukan dengan metode *black box* berdasarkan alur sistem pada kasus penggunaan dari sistem ini. Skenario dan hasil pengujian

yang dilakukan ditunjukkan pada LAMPIRAN A : Hasil Pengujian Fungsionalitas Sistem Tabel 7.1.

5.2.2 Pengujian Kerelevanan Jawaban *Chatbot*

Pengujian kerelevanan jawaban *chatbot* bertujuan untuk mengetahui kerelevanan pertanyaan pengunjung dengan jawaban yang dihasilkan oleh *chatbot*. Pengujian ini dilakukan dengan meminta 12 sukarelawan yang pernah mengunjungi kebun binatang mengisi kuesioner. Sukarelawan menilai berdasarkan 50 pasangan pertanyaan dan jawaban apakah respon dari pertanyaan tersebut relevan atau tidak. 50 pertanyaan tersebut merupakan pertanyaan sembarang dan jawaban dari pertanyaan tersebut merupakan respon yang dihasilkan dari sistem yang dibangun pada tugas akhir ini. Pengujian ini dilakukan menggunakan data pada *brainfile* sebanyak 300 data. Hasil dari uji coba kerelevanan jawaban *chatbot* ditunjukkan pada LAMPIRAN B : *Majority Vote Kerelevanan Jawaban Chatbot* Tabel 8.1.

5.2.3 *System Usability Scale (SUS)*

Dalam pengujian aplikasi ini, dilakukan juga *System Usability Scale*. Pengujian ini dilakukan untuk mengetahui penilaian pengguna terhadap kemudahan penggunaan aplikasi aplikasi yang telah dibangun berdasarkan performa dari sistem yang telah dibuat. Pengujian ini dilakukan oleh 12 sukarelawan yang juga melakukan pengujian kerelevanan jawaban *chatbot*. Relawan diminta untuk menggunakan fitur ZooSite kemudian menilai pernyataan berdasarkan pendapat relawan dengan skala 1 hingga 5. Nilai 1 berarti sangat tidak setuju dan nilai 5 untuk sangat setuju. Adapun pernyataan yang diajukan pada kuesioner tersebut yaitu [24] :

1. Saya berpikir akan menggunakan sistem ini lagi.
2. Saya merasa sistem ini rumit untuk digunakan.
3. Saya merasa sistem ini mudah untuk digunakan.
4. Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.

5. Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.
6. Saya merasa ada banyak hal yang tidak konsisten (tidak serasi) pada sistem ini.
7. Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.
8. Saya merasa sistem ini membingungkan.
9. Saya merasa tidak ada hambatan dalam menggunakan sistem ini.
10. Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

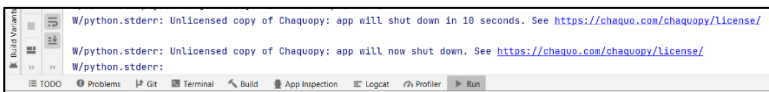
Hasil dari kuesioner *System Usability Scale* menghasilkan skor responden yang ditunjukkan oleh Tabel 5.2

Tabel 5.2 Perolehan Skor dari *System Usability Scale*

Responden	Skor Pertanyaan									
	1	2	3	4	5	6	7	8	9	10
1	4	1	5	2	5	1	5	1	4	4
2	4	1	4	2	5	1	4	1	4	3
3	4	2	5	1	5	3	5	1	5	2
4	5	1	5	1	5	1	5	1	5	3
5	5	2	5	1	5	1	4	2	4	3
6	5	2	5	2	5	2	4	1	5	5
7	2	3	4	2	4	5	2	4	2	5
8	4	1	5	1	4	1	5	2	5	2
9	3	1	4	1	5	2	3	2	5	1
10	4	1	5	2	5	2	4	1	5	3
11	4	1	4	2	4	2	5	2	4	4
12	5	1	5	1	5	1	5	1	5	5

5.3 Evaluasi

Pada subbab ini dijelaskan mengenai evaluasi dari hasil yang didapatkan pada pengujian. Pada uji coba fungsionalitas yang telah dilakukan, aplikasi dapat berjalan pada perangkat Android 10 dan seluruh *output* berdasarkan skenario-skenario yang telah ditentukan sudah sesuai dengan rancangan sistem. Dalam pelaksanaan uji coba, aplikasi tertutup beberapa kali akibat menggunakan Chaquopy yang belum berlisensi.



Gambar 5.1 Peringatan Lisensi Chaquopy

Pada uji coba kerelevanan jawaban model *chatbot*, dilakukan uji coba pada model *chatbot* dengan menentukan kerelevanan dari 50 pasangan pertanyaan dan jawaban. 5 dari 50 pasangan pertanyaan dan jawaban tersebut adalah pertanyaan yang tidak berkaitan dengan kebun binatang. Berdasarkan hasil dari uji coba tersebut, didapatkan sebanyak 39 jawaban *chatbot* dinilai relevan dengan pertanyaan pengunjung dan 11 jawaban tidak relevan dengan pertanyaan pengunjung. Dari 5 pertanyaan yang tidak berkaitan dengan kebun binatang, 4 jawaban dinilai tidak relevan. Hasil tersebut menunjukkan bahwa kerelevanan jawaban untuk pertanyaan bebas dari model ini adalah sebesar 78% dan untuk pasangan pertanyaan-jawaban dengan topik kebun binatang sebesar 84,4%. Jawaban yang tidak relevan tersebut disebabkan kurangnya data *brainfile* mengenai topik yang berkaitan, seperti jumlah satwa yang ada di kebun binatang, tentang kotak saran, serta data fasilitas yang lebih detail.

Berdasarkan jawaban yang telah didapat, dilakukan pula analisa untuk menentukan *confusion matrix* dengan menghitung banyak jawaban *true positive*, *true negative*, *true negative*, dan *false negative*. Analisa tersebut dilakukan dengan membandingkan

hasil tanya-jawab dengan data yang ada pada data *brainfile* berdasarkan kondisi sebagai berikut.

- TP (*true positive*) adalah banyak pertanyaan berkaitan dengan kebun binatang yang berhasil dijawab dan tepat.
- FP (*false positive*) adalah banyak pertanyaan berkaitan dengan kebun binatang yang mendapatkan jawaban yang tidak relevan.
- TN (*true negative*) merupakan banyak pertanyaan yang tidak berkaitan dengan kebun binatang dan tidak dipahami oleh sistem.
- FN (*false negative*) adalah banyak pertanyaan yang tidak berkaitan dengan kebun binatang, tetapi mendapatkan respon jawaban dari *brainfile*.

Jumlah data yang didapatkan berdasarkan analisa tersebut ditunjukkan dalam bentuk *confusion matrix* pada Tabel 5.3.

Tabel 5.3 Confusion Matrix

	<i>Positive</i>	<i>Negative</i>
<i>True</i>	39	1
<i>False</i>	6	4

Berdasarkan data tersebut, dilakukan perhitungan untuk mengetahui akurasi, presisi, dan *recall* dari jawaban yang telah didapatkan. Nilai presisi menunjukkan ketepatan *chatbot* dalam menentukan jawaban, nilai *recall* menunjukkan tingkat keberhasilan *chatbot* menemukan jawaban *chatbot*, dan nilai akurasi menunjukkan keberhasilan *chatbot* memprediksi jawaban. Rumus yang digunakan pada perhitungan akurasi, presisi, dan *recall* ditunjukkan pada Persamaan 5.1, Persamaan 5.2, dan Persamaan 5.3.

$$Akurasi = \frac{\sum TP + TN}{\sum TP + FP + TN + FN} \quad (5.1)$$

$$Presisi = \frac{\sum TP}{\sum TP + FP} \quad (5.2)$$

$$Recall = \frac{\sum TP}{\sum TP + FN} \quad (5.3)$$

Berdasarkan rumus tersebut, dilakukan perhitungan nilai akurasi, presisi, dan *recall* sebagai berikut.

$$Akurasi = \frac{39 + 1}{39 + 6 + 1 + 4} = \frac{40}{50} = 0,80 = 80\%$$

$$Presisi = \frac{39}{39 + 6} = \frac{39}{45} = 0.867 = 86.7\%$$

$$Recall = \frac{39}{39 + 4} = \frac{39}{43} = 0.907 = 90.7\%$$

Selain pengujian terhadap sistem, dilakukan pula penilaian *system usability scale* dari pengunjung melalui kuesioner untuk mengetahui performa dan kebermanfaatan sistem. Untuk mendapatkan kesimpulan berdasarkan *system usability scale*, dilakukan perhitungan berdasarkan jawaban setiap responden sebagai berikut.

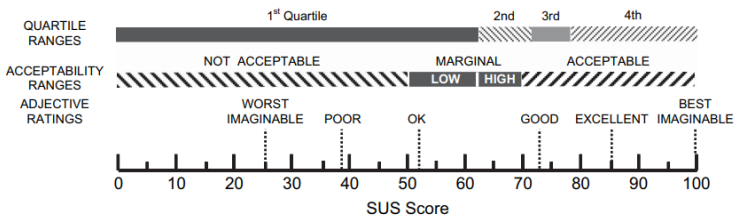
- a. Skor yang didapatkan pada pernyataan nomor ganjil dikurangi 1.
- b. Skor yang didapatkan pada pernyataan nomor genap, dilakukan dengan perhitungan dengan mengurangi nilai 5 dengan skor yang didapatkan tersebut.
- c. Seluruh nilai yang didapatkan pada poin a dan b dijumlahkan lalu kemudian dikalikan dengan 2,5 sehingga didapatkan nilai SUS.

Untuk mendapatkan nilai SUS akhir, dilakukan perhitungan nilai rata-rata skor SUS dengan menjumlahkan nilai SUS dari seluruh responden kemudian dibagi dengan jumlah responden. Perhitungan nilai SUS berdasarkan jawaban yang telah didapatkan ditunjukkan pada Tabel 5.4.

Tabel 5.4 Perhitungan Skor SUS

Responden	Perhitungan Nilai Skor Pertanyaan										Jumlah Nilai	Skor SUS
	1	2	3	4	5	6	7	8	9	10		
1	3	4	4	3	4	4	4	4	3	1	34	85
2	3	4	3	3	4	4	3	4	3	2	33	82.5
3	3	3	4	4	4	2	4	4	4	3	35	87.5
4	4	4	4	4	4	4	4	4	4	2	38	95
5	4	3	4	4	4	4	3	3	3	2	34	85
6	4	3	4	3	4	3	3	4	4	0	32	80
7	1	2	3	3	3	0	1	1	1	0	15	37.5
8	3	4	4	4	3	4	4	3	4	3	36	90
9	2	4	3	4	4	3	2	3	4	4	33	82.5
10	3	4	4	3	4	3	3	4	4	2	34	85
11	3	4	3	3	3	3	4	3	3	1	30	75
12	4	4	4	4	4	4	4	4	4	0	36	90
Skor SUS Akhir												81.25

Dengan rata-rata skor SUS sebesar 81,25 berdasarkan penilaian skor pada Gambar 5.2 dapat disimpulkan bahwa aplikasi memiliki ketergunaan yang dapat diterima dan mudah dipahami oleh pengguna.

*Gambar 5.2 Peringkat Nilai Skor System Usability Scale*

Ketika pengujian berlangsung saat responden membuka peta atau menggunakan *chatbot* dan kemudian menampilkan fasilitas terdekat yang ditanyakan, beberapa responden tidak mengerti

maksud dari tanda yang muncul pada peta. Selain itu, ketika responden mengirimkan pertanyaan, ada beberapa kasus responden tidak mengetahui bahwa pesan yang dikirim berhasil dimasukkan dan sedang diproses oleh model. Oleh karena itu, lebih baik jika dilakukan penambahan keterangan pada peta agar pengunjung lebih mudah mengerti tentang peta tersebut serta menambahkan indikator yang menunjukkan bahwa pesan yang dikirimkan sedang diproses.

BAB VI

KESIMPULAN

Bab ini membahas mengenai kesimpulan yang diperoleh dari pengerjaan tugas akhir dan saran terkait pengembangan dari tugas akhir ini yang dapat dilakukan di masa yang akan mendatang.

6.1. Kesimpulan

Berdasarkan uji coba dan evaluasi yang telah dilaksanakan, kesimpulan yang didapatkan adalah sebagai berikut.

1. Aplikasi ZooSite yang dikembangkan menggunakan *Universal Sentence Encoder* dan *cosine similarity* mampu menggantikan peran pemandu serta membantu pengunjung kebun binatang dengan menyediakan informasi yang dapat diperoleh dengan melakukan percakapan pada fitur *chatbot* dengan persentase korelevanan jawaban 78%, akurasi sebesar 80%, presisi 86.7%, dan *recall* 90.7%.
2. Dengan perolehan nilai skor SUS 81,25 menunjukkan bahwa ZooSite memiliki ketergunaan yang dapat diterima dan mudah dipahami oleh pengguna.
3. Penggunaan *Universal Sentence Encoder* pada *chatbot* mempermudah pengunjung untuk menggunakan kalimat secara bebas dalam melakukan tanya-jawab dengan *chatbot* serta menghasilkan jawaban yang cukup relevan.

6.2. Saran

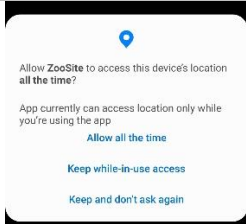
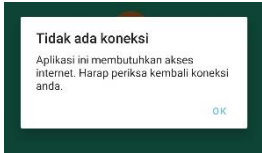

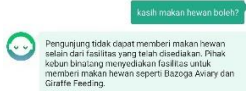
Saran yang diberikan dari hasil uji coba dan evaluasi pada pengerjaan tugas akhir ini adalah sebagai berikut:



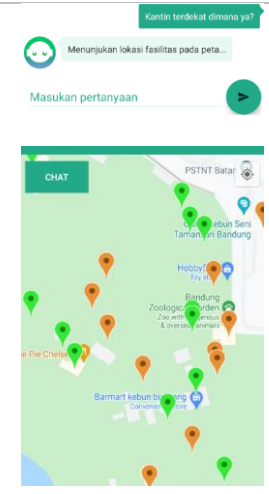
1. ZooSite tidak dipublikasikan dan belum mendapatkan lisensi resmi dari Chaquopy sehingga aplikasi dapat menutup apabila pengunjung menggunakan fitur *chatbot*. Oleh karena itu, agar aplikasi bisa digunakan sepenuhnya, diperlukan lisensi dari Chaquopy.


2. Menambahkan lebih banyak data *brainfile* karena relevansi jawaban bergantung pada kelengkapan data *brainfile*.
3. Memperbaiki tampilan agar lebih mudah dipahami oleh pengunjung seperti menambahkan legenda pada peta, menambahkan jalan, menampilkan indikator memproses jawaban untuk *chatbot*, dan mengubah penanda pada peta menjadi ikon yang melambangkan lokasi tersebut.

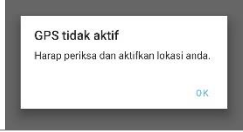
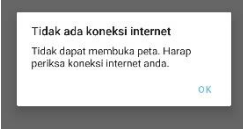
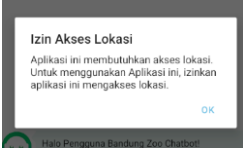

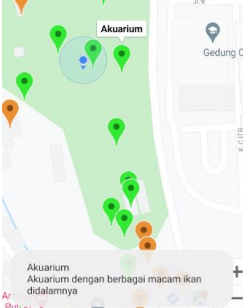
LAMPIRAN A : Hasil Pengujian Fungsionalitas Sistem

Tabel 7.1 Hasil Pengujian Fungsionalitas Sistem

No.	Skenario	Output yang diharapkan	Output yang diperoleh
1.	Aplikasi belum memiliki akses terhadap lokasi	Muncul peringatan izin mengakses lokasi	
2.	Aplikasi dibuka saat pengunjung tidak mengaktifkan internet	Muncul peringatan internet tidak menyala kemudian menutup aplikasi	
3.	Menggunakan menu <i>chatbot</i> saat tidak ada koneksi internet	Muncul peringatan internet tidak menyala melalui <i>chatbot</i>	
4.	Menanyakan pertanyaan umum (tidak mengandung kata “terdekat”) dan didapatkan nilai <i>cosine similarity</i> ≥ 0.5	<i>Chatbot</i> mengirimkan jawaban dari pertanyaan pengunjung	

No.	Skenario	Output yang diharapkan	Output yang diperoleh
5.	Menanyakan pertanyaan umum (tidak mengandung kata “terdekat”) dan didapatkan nilai <i>cosine similarity</i> < 0.5	Chatbot mengirimkan pesan bahwa pertanyaan pengunjung tidak dapat dipahami	
6.	Menanyakan pertanyaan umum (tidak mengandung kata “terdekat”) saat tidak terkoneksi dengan internet	Chatbot mengirimkan pesan bahwa aplikasi membutuhkan koneksi internet	
7.	Menanyakan lokasi fasilitas terdekat yang terdapat pada <i>database</i> : “Kantin terdekat ada di mana ya?”	Aplikasi membuka menu peta dan menunjukkan lokasi dari kantin terdekat	

No.	Skenario	Output yang diharapkan	Output yang diperoleh
			
8.	Menanyakan lokasi fasilitas terdekat yang tidak terdapat pada <i>database</i> : “Taman kota terdekat ada di mana ya?”	Aplikasi mengirimkan pesan bahwa fasilitas yang ditanyakan tidak terdata atau tidak sesuai dengan yang ada pada <i>database</i> .	
9.	Menanyakan lokasi fasilitas tanpa koneksi internet	<i>Chatbot</i> mengirimkan pesan bahwa aplikasi membutuhkan koneksi internet	
10.	Menanyakan lokasi fasilitas tanpa GPS menyala	<i>Chatbot</i> mengirimkan pesan bahwa GPS tidak menyala.	

No.	Skenario	Output yang diharapkan	Output yang diperoleh
11.	Membuka menu peta dengan GPS tidak aktif.	Muncul peringatan bahwa sistem membutuhkan koneksi GPS.	
12.	Membuka menu peta tanpa koneksi internet.	Muncul pesan bahwa aplikasi membutuhkan koneksi internet	
13.	Membuka menu peta tanpa izin akses lokasi pada latar belakang.	Muncul pesan bahwa beberapa fitur membutuhkan akses lokasi setiap saat.	
14.	Menekan penanda pada peta.	Muncul keterangan dari lokasi tersebut.	
15.	Memasuki Geofence dari kandang satwa.	Muncul Toast berisi deskripsi satwa tersebut.	

LAMPIRAN B : *Majority Vote* Kerelevanan Jawaban *Chatbot*

Tabel 8.1 Majority Vote Kerelevanan Jawaban Chatbot

No.	Pertanyaan	Jawaban yang diperoleh	Jumlah Pemilih		Suara Terbanyak
			Relevan	Tidak Relevan	
1.	Ada berapa harimau di sini?	Terdapat kurang lebih 800 individu hewan di kebun binatang ini.	4	8	Tidak Relevan
2.	Bagaimana memberi makan hewan?	Anda bisa memberi makan jerapah di fasilitas <i>giraffe feeding</i> , burung di <i>bazoga aviary</i> , dan rusa.	11	1	Relevan
3.	Berapa kilo harimau diberi makan?	Pemberian makan bergantung kepada spesies hewan yang diberi makan. Untuk keterangan lebih lanjut, kami tidak bisa memberi tahu lebih detail.	9	3	Relevan
4.	Spesies apa yang paling baru?	Hewan baru yang ada di kebun binatang kami adalah owa Jawa yang lahir di saat pandemik.	12	0	Relevan

No.	Pertanyaan	Jawaban yang diperoleh	Jumlah Pemilih		Suara Terbanyak
			Relevan	Tidak Relevan	
5.	Apakah ada tempat penangkaran?	Ya, sebagai tempat konservasi hewan, kebun binatang memiliki tempat penangkaran untuk hewan.	12	0	Relevan
6.	Apabila ada yang baru lahir, di kemanakan anaknya?	Bayi hewan yang lahir di kebun binatang ini akan dimasukkan ke penangkaran terlebih dahulu hingga umur tertentu.	12	0	Relevan
7.	Aman tidak kandangnya?	Kandang di kebun binatang bisa dipasikan aman untuk pengunjung dan hewan di dalamnya.	12	0	Relevan
8.	Boleh kasih makan hewan?	Pengunjung tidak dapat memberi makan hewan selain dari fasilitas yang telah disediakan. Pihak kebun binatang menyediakan fasilitas untuk memberi makan seperti <i>Bazoga Aviary</i> dan <i>Giraffe Feeding</i> .	12	0	Relevan

No.	Pertanyaan	Jawaban yang diperoleh	Jumlah Pemilih		Suara Terbanyak
			Relevan	Tidak Relevan	
9.	Fasilitas di sini ada apa aja?	Fasilitas dalam kebun binatang ini adalah koleksi satwa, pertunjukan satwa, kantin, wahana permainan seperti perahu kayuh dan taman bermain, serta sarana ibadah.	12	0	Relevan
10.	Ada kotak saran untuk pengunjung?	Pengunjung dapat menggunakan jasa <i>zoo educator</i> /pemandu sesuai dengan permintaan dengan jumlah terbatas.	1	11	Tidak Relevan
11.	Hewan yang baru apa?	Hewan baru yang ada di kebun binatang kami adalah owa Jawa yang lahir di saat pandemik.	9	3	Relevan
12.	Ada atraksi hewan?	Terdapat <i>Animal Attraction</i> atau Atraksi Hewan sebagai salah satu fasilitas dan hiburan bagi pengunjung.	12	0	Relevan
13.	Kebun binatang dapat donasi tidak?	Kami menerima donasi.	11	1	Relevan

No.	Pertanyaan	Jawaban yang diperoleh	Jumlah Pemilih		Suara Terbanyak
			Relevan	Tidak Relevan	
14.	Ada <i>tourguide</i> nya gak?	Kebun binatang ini menyediakan <i>zoo educator</i> (pemandu) sesuai permintaan pengunjung dengan kuota yang terbatas.	12	0	Relevan
15.	Harga tiket berapa?	Tiket apa yang Anda tanyakan? Silahkan tanya kembali.	9	3	Relevan
16.	Ada yang jual makanan?	Anda bisa membeli makanan di kantin kami yang tersedia di tengah kebun binatang atau restoran di dekat gerbang keluar 2. Untuk lokasi dari tempat tersebut bisa dilihat pada peta atau tanyakan tempat makan terdekat.	11	1	Relevan
17.	Hewan apa saja yang boleh dikasih makan?	Anda bisa memberi makan jerapah di fasilitas <i>giraffe feeding</i> , burung di <i>gazebo aviary</i> , dan rusa.	12	0	Relevan

No.	Pertanyaan	Jawaban yang diperoleh	Jumlah Pemilih		Suara Terbanyak
			Relevan	Tidak Relevan	
18.	Parkiran sebelah mana?	Parkiran mobil tersedia di pintu masuk kebun binatang dan parkiran motor di sebelah kiri arah masuk pintu kebun binatang.	12	0	Relevan
19.	<i>Food court</i> ada?	Kami menyediakan kantin di tengah kebun binatang, dekat dengan <i>Africa Zone</i> serta sebuah restoran dekat dengan gerbang keluar 2.	10	2	Relevan
20.	Mushala di mana?	Kami menyediakan mushaa dekat dengan pintu keluar dan gerbang keluar. Untuk letak fasilitas tersebut bisa dilihat pada peta atau mengirimkan pesan 'mushala terdekat'	12	0	Relevan
21.	Di mana beli minum?	Anda bisa membeli minum di kantin dan restoran yang tersedia di dekat <i>Zona Afrika</i> , di area tengah kebun binatang, atau restoran di	12	0	Relevan

No.	Pertanyaan	Jawaban yang diperoleh	Jumlah Pemilih		Suara Terbanyak
			Relevan	Tidak Relevan	
		dekat gerbang keluar 2.			
22.	Apa makanan gajah?	Gajah memakan dedaunan, rumput, serta buah-buahan.	12	0	Relevan
23.	Berapa buat kasih makan rusa	Untuk memberi makan jerapah, anda dapat membayar Rp 25.000, memberi makan burung di <i>Gazebo Aviary</i> dengan Rp. 10.000 dan rusa dengan membayar Rp 5.000 per wortel	10	2	Relevan
24.	Apa aja yang bisa dikasih makan wortel?	Merak hijau memakan biji-bijian, serangga, dan tunas muda.	0	12	Tidak Relevan
25.	Fotocopy di dekat sini di mana ya?	Anda dapat menemukan eland di <i>Zona Afrika</i> .	1	11	Tidak Relevan
26.	Di mana bisa naik gajah?	Anda bisa menaiki gajah di lokasi satwa tunggang dengan membayar Rp 30.000	12	0	Relevan

No.	Pertanyaan	Jawaban yang diperoleh	Jumlah Pemilih		Suara Terbanyak
			Relevan	Tidak Relevan	
27.	Kapan bazoga aviarynya buka?	Kebun binatang ini buka pada jam 10:00 hingga 16:00 saat pandemi.	3	9	Tidak Relevan
28.	Berapa tinggi gajah?	Gajah asia dapat tumbuh hingga 2,8 meter dan Gajah afrika bisa tumbuh hingga 3,2 meter.	12	0	Relevan
29.	Ada permainan apa aja?	Mohon maaf pertanyaan Anda tidak dapat dipahami, silahkan tanya kembali.	5	7	Tidak Relevan
30.	Karcis masuk bazoga aviary berapa?	Harga masuk tiket kebun binatang adalah 50.000 per orang dan anak-anak dengan tinggi lebih dari 80cm dikenakan biaya masuk. Bagi rombongan umum bisa membayar senilai 45.000 dan untuk rombongan sekolah 40.000 dengan minimal 30 orang. Tiket bagi rombongan tidak berlaku pada hari Minggu dan hari libur.	4	8	Tidak Relevan

No.	Pertanyaan	Jawaban yang diperoleh	Jumlah Pemilih		Suara Terbanyak
			Relevan	Tidak Relevan	
31.	Tempat foto di mana ya?	Pengunjung dapat menggunakan fasilitas kami seperti berfoto dengan burung parrot dan binturong di dekat pintu keluar 2.	11	1	Relevan
32.	Pintu keluar sebelah mana?	Parkiran mobil tersedia di pintu masuk kebun binatang dan parkiran motor berada di sebelah kiri arah masuk pintu kebun binatang.	1	11	Tidak Relevan
33.	Apa makanan binturong?	Binturong memakan mamalia kecil, anak burung, buah-buahan, telur dan terkadang pucuk daun.	12	0	Relevan
34.	Siamang asalnya dari mana?	Siamang tersebar luas di Indonesia, Malaysia, dan Thailand. Siamang tinggal di hutan.	10	2	Relevan
35.	Ada berapa macam tutulnya?	Terdapat kurang lebih 800 individu hewan di kebun binatang ini.	0	12	Tidak Relevan

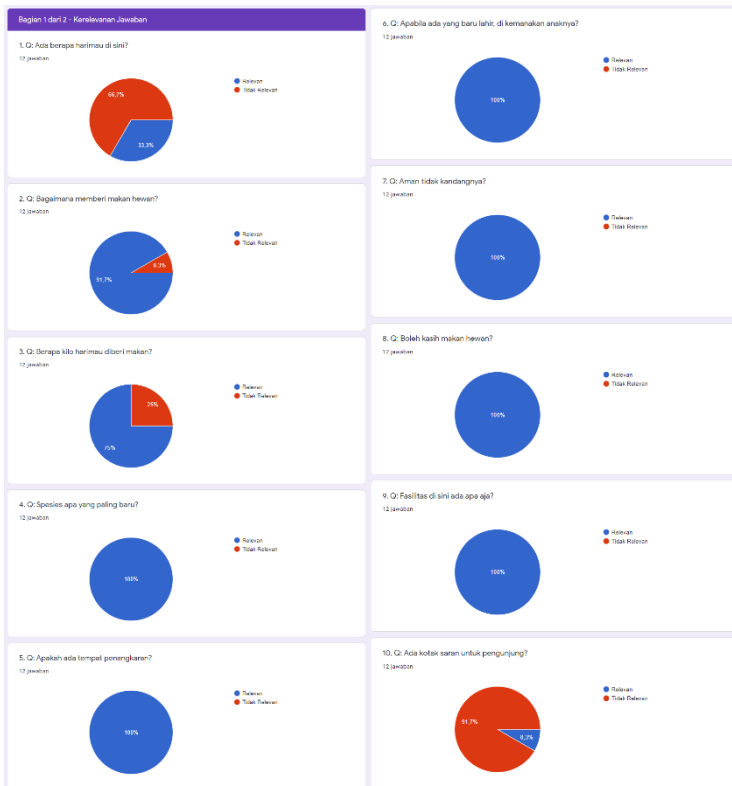
No.	Pertanyaan	Jawaban yang diperoleh	Jumlah Pemilih		Suara Terbanyak
			Relevan	Tidak Relevan	
36.	Medsos kebun binatangnya ada?	Situs web resmi Kebun binatang ini bisa diakses di www.-zoo.com . Facebook @officialzoo dan Twitter @_zoo	11	1	Relevan
37.	Di akuarium isinya apa?	Isi dari akuarium adalah ikan aligator.	0	12	Tidak Relevan
38.	Kapan kebun binatangnya tutup?	Kebun binatang ini buka pada jam 10:00 hingga 16:00 saat pandemi Covid-19.	12	0	Relevan
39.	Gajah apa itu yang di sana?	Spesies gajah di kebun binatang ini adalah gajah sumatra.	11	1	Relevan
40.	Harimau apa yang ada di sini?	Saat ini di kebun binatang ini terdapat 2 spesies harimau, yaitu harimau sumatra dan harimau benggala	12	0	Relevan
41.	Dimana letak macan?	Anda dapat menemukan harimau Benggala di bagian selatan kebun binatang, dekat dengan kandang singa serta harimau	9	3	Relevan

No.	Pertanyaan	Jawaban yang diperoleh	Jumlah Pemilih		Suara Terbanyak
			Relevan	Tidak Relevan	
		Sumatera di dekat akuarium dan beruang madu.			
42.	Kantin terdekat	Sistem memunculkan peringatan GPS kemudian setelah pengunjung memasukkan kembali pertanyaannya tersebut, sistem membuka peta dan menunjukkan kantin.	10	2	Relevan
43.	Kandang terdekat?	Fasilitas tidak ditemukan, silahkan tanya kembali.	5	7	Tidak Relevan
44.	Tutupnya jam berapa?	Kebun binatang buka pada jam 10:00 hingga 16:00 saat pandemi Covid-19.	12	0	Relevan
45.	Dimana kandang buaya?	Anda dapat melihat buaya di bagian belakang kebun binatang, dekat dengan kandang kura-kura dan Atraksi Hewan.	12	0	Relevan

No.	Pertanyaan	Jawaban yang diperoleh	Jumlah Pemilih		Suara Terbanyak
			Relevan	Tidak Relevan	
46.	Ada berapa hewan di kebun binatang?	Terdapat kurang lebih 800 individu hewan di kebun binatang.	11	1	
47.	Berapa harga tiket kebun binatang?	Harga masuk tiket kebun binatang adalah 50.000 per orang dan anak-anak dengan tinggi lebih dari 80cm dikenakan biaya masuk. Bagi rombongan umum bisa membayar senilai 45.000 dan untuk rombongan sekolah 40.000 dengan minimal 30 orang. Tiket bagi rombongan tidak berlaku pada hari Minggu dan hari libur.	12	0	Relevan
48.	Tempat istirahat	Mohon maaf pertanyaan Anda tidak dapat dipahami, silahkan tanya kembali.	9	3	Relevan
49.	Di mana bisa liat beruang?	Beruang madu dapat dilihat oleh pengunjung di dekat kandang macan dan tapir. Serta kandang lainnya yang	12	0	Relevan

No.	Pertanyaan	Jawaban yang diperoleh	Jumlah Pemilih		Suara Terbanyak
			Relevan	Tidak Relevan	
		terletak di bagian selatan kebun binatang, dekat dengan kandang singa dan macan.			
50.	Beruangnya makan apa?	Beruang madu memakan serangga, buah, dan mamalia kecil.	12	0	Relevan

LAMPIRAN C : Dokumentasi Hasil Kuesioner



Gambar 8.1 Dokumentasi Kuesioner Kerelevanan Pertanyaan 1-10



Gambar 9.2 Dokumentasi Kuesioner Kerelevanan Pertanyaan 11-20



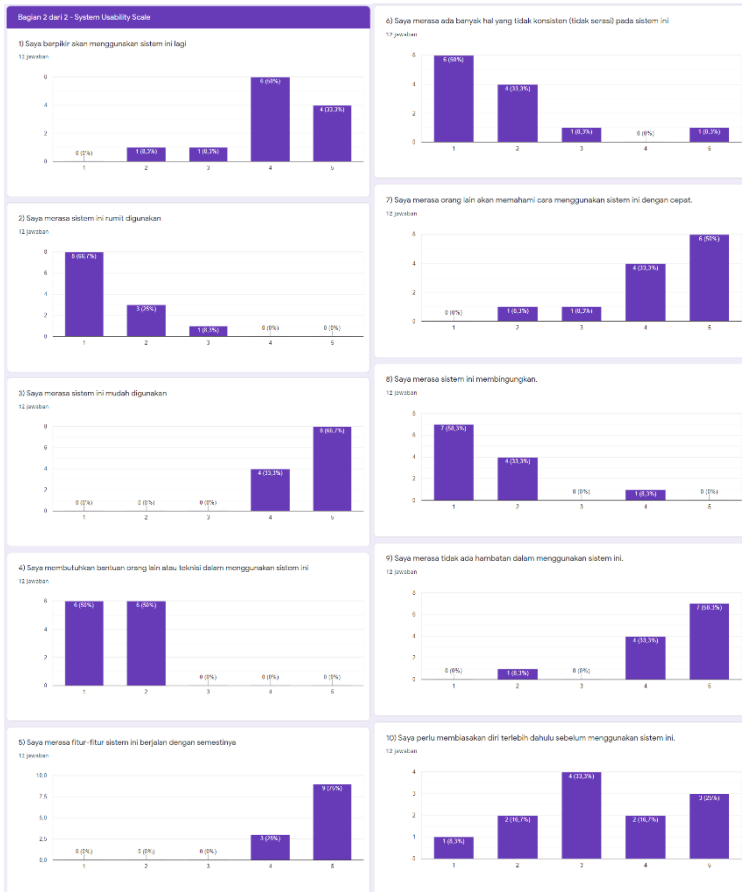
Gambar 9.3 Dokumentasi Kuesioner Kerelevanan Pertanyaan 21-30



Gambar 9.4 Dokumentasi Kuesioner Kerelevanan Pertanyaan 31-40



Gambar 9.5 Dokumentasi Kuesioner Kerelevanan Pertanyaan 41-50



Gambar 9.6 Dokumentasi Kuesioner System Usability Scale

DAFTAR PUSTAKA

- [1] Singh, R., Paste, M., Shinde, N., Patel, H., dan Mishra, N. (2018). “Chatbot using TensorFlow for small Businesses”. 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT). doi:10.1109/icicct.2018.8472998
- [2] Oktavia, Chaulina. 2020. “Implementasi Chatbot Menggunakan Dialogflow dan Messenger Untuk Layanan Customer Service Pada E-Commerce”. J I M P - Jurnal Informatika Merdeka Pasuruan. 4. 10.37438/jimp.v4i3.230.
- [3] Nekolný, Lukáš dan Fialová, Dana. 2018. Zoo Tourism: What Actually Is a Zoo?. Czech Journal of Tourism. 7. 153-166. 10.1515/cjot-2018-0008.
- [4] Schiller J. dan Voisard A. 2004. “Location-Based Services”. Elsevier.
- [5] S. Steiniger, M. Neun, dan A. Edwardes. 2012. “Foundation of LBS”, Swiss: CartouCHE.<URL:http://www.e-cartouche.ch/content_reg/cartouche/LBSbasics/en/text/LBSbasics.pdf>
- [6] J. Li. 2014. “Research and Implementation of LBS for Mobile Virtual Navigation System Based on Android”, Dissertation.
- [7] Greaney, C. 2011. “Location Aware Apps for Tourism”. United States: Lulu.com.
- [8] Xu, Guochang. 2007. “GPS: Theory, Algorithms and Applications.” GPS: Theory, Algorithms and Applications. 1-340. 10.1007/978-3-540-72715-6.
- [9] L. Gray. 2014. “How Does GPS Work?” U.S.A: Gareth Stevens Publishing.
- [10] Kaplan, E. dan Hegarty, C. 2006. “Understanding GPS: Principles and Applications”, 2nd ed, Norwood: Artech House, Inc.

- [11] S. Eko Budi. 2020. "Sistem Informasi Berbasis Web: Menggunakan Google Maps dan Mapbox API". Bandung: Informatika Bandung.
- [12] Reclus, F. dan Drouard, K. 2009. "Geofencing for fleet & freight management," 2009 9th International Conference on Intelligent Transport Systems Telecommunications, (ITST), Lille, pp. 353-356, doi: 10.1109/ITST.2009.5399328.
- [13] D. Cer et al. 2018. "Universal Sentence Encoder" arXiv:1803.11175v2 [cs.CL]
- [14] Chaudhary, A., 2020. "Universal Sentence Encoder Visually Explained" <URL <https://amitnss.com/2020/06/universal-sentence-encoder/>>(diakses: 6 November 2021 21:10)
- [15] F Rahutomo, T. Kitasuka, dan M.Aritsugi. 2012. "Semantic Cosine Similarity".
- [16] Kisanrao, N. M. 2013. "Landmark based shortest path detection by using A* Algorithm and Haversine Formula".
- [17] Hardiansah, and Suryono, S. 2020. "Panduan Praktis Membuat Aplikasi Android Dengan Android Studio (Kotlin)". Yogyakarta: PT Lauwba Techno Indonesia.
- [18] Yuniansyah. 2020. "Algoritma dan Pemrograman Menggunakan Bahasa Pemrograman Java (Teori dan Aplikasinya)". Bogor: Penerbit Lindan Bestari.
- [19] Perkasa, T. R., Widyantara H., dan Susanto, P. 2014. "Rancang Bangun Pendeteksi Gerak Menggunakan Image Subtraction pada Single Board Computer (SBC)", JCONES vol. 3, no. 2, 2014, pp. 90-97.
- [20] Z. Dobesova. 2011. "Programming language Python for data processing," 2011 International Conference on Electrical and Control Engineering, Yichang, pp. 4866-4869, doi: 10.1109/ICECENG.2011.6057428.
- [21] M. Astiningrum, M., Saputra, P. Y. , dan Rohmah, M. S.. 2018. "Implementasi NLP dengan Konversi Kata pada

- Sistem Chatbot Konsultasi Laktasi”, JIP, vol. 5, no. 1, pp. 46-52.
- [22] Z. Hu, Z. Wang, H. Liu and H. Bie. 2010. "Self-service folk tourism guiding technology on mobile terminal with multi-mode: Application of GPS and electronic map", 2010 IEEE International Geoscience and Remote Sensing Symposium, Honolulu, HI, pp. 4553-4556, doi: 10.1109/IGARSS.2010.5653882.
- [23] Gnewuch, U., Morana, S., Adam, M. T. P., dan Maedche, A. 2018. "Faster Is Not Always Better: Understanding the Effect of Dynamic Response Delays in Human-Chatbot Interaction."
- [24] Sharfina, Z., dan Santoso, H. B. 2016. "An Indonesian adaptation of the System Usability Scale (SUS)". 2016 International Conference on Advanced Computer Science and Information Systems (ICACISIS). doi:10.1109/icacsis.2016.7872776
- [25] Bangor, A., Kortum, P. T., dan Miller, J. T. 2008. "An Empirical Evaluation of the System Usability Scale". International Journal of Human-Computer Interaction, 24(6), 574–594. doi:10.1080/10447310802205776

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Vania Cikanindi lahir di Kota Bandung pada tanggal 26 Januari 1999. Penulis telah menempuh pendidikan formal di TK Darul Hikam (2003-2005), SD Darul Hikam (2005-2011), SMP Darul Hikam (2011-2014), SMA Negeri 3 Bandung (2014-2017), dan Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember (ITS) angkatan 2017. Penulis memilih bidang studi Manajemen Cerdas Informasi (MCI). Penulis aktif dalam organisasi sebagai anggota (2017-2018) dan staf ahli (2019) pada UKM Kendo serta Keluarga Muslim Informatika sebagai anggota (2019) dan Ketua Kemuslimahan (2020) pada Keluarga Muslim Informatika. Penulis juga mengikuti kegiatan kepanitiaan seperti Schematics sebagai staff ahli divisi NLC pada tahun 2019. Penulis dapat dihubungi melalui surel vcikanindi@gmail.com.