

BAB III

PERANCANGAN SISTEM

3.1 Inialisasi Proyek

Tahap awal dari pengembangan sebuah sistem pakar, jika merujuk pada metode *Expert System Development Life Cycle* (ESDLC), adalah tahap inialisasi proyek. Tahap insialisasi dalam penelitian ini meliputi beberapa aspek yaitu, pendefinisian masalah, pemberian solusi alternatif, penentuan pakar, dan verifikasi metode yang digunakan dalam pembuatan sistem pakar.

3.1.1 Pendefinisian Masalah

Masalah yang sering dijumpai oleh seorang pembelajar bahasa Arab diantaranya sebagai berikut:

1. Kesulitan dalam menentukan jenis kata pada sebuah kalimat.
2. Keberadaan pakar atau orang yang mahir bahasa Arab yang tidak mudah ditemui setiap saat.
3. Kurangnya media belajar yang efektif dan interaktif.

3.1.2 Solusi Alternatif

Penulis dalam penelitian ini mengembangkan sebuah sistem yang dapat membantu seseorang untuk menentukan jenis kata dalam bahasa Arab. Sistem ini dapat menjawab pertanyaan pengguna terkait jenis dari sebuah kata dalam bahasa Arab. Sistem ini berupa aplikasi *chatbot* yang dapat berinteraksi dengan pengguna layaknya seorang pakar. Aplikasi *chatbot* dipilih guna memudahkan pengguna dalam pemakaian sistem tanpa harus mengunduh aplikasi tambahan, dengan asumsi setiap calon pengguna sudah memasang aplikasi perpesanan Telegram. Disamping itu, dengan menerapkan konsep aplikasi *chatbot* pada Telegram ini diharapkan dapat meraih lebih banyak pengguna mengingat banyaknya pengguna aplikasi Telegram ini.

Sistem ini diharapkan dapat membantu pengguna dalam mempelajari bahasa Arab dan mengidentifikasi jenis kata berbahasa Arab kapan dan dimana saja selama terhubung ke jaringan internet.

3.1.3 Pakar

Kepakaran yang diterapkan pada sistem ini mengacu kepada sebuah buku karya Prof. Dr. Ahmad Thib Raya, M.A. yang berjudul *Pangkal Penguasaan Bahasa Arab* yang diterbitkan oleh penerbit Marja pada tahun 2015.

3.1.4 Metode

Sistem ini mengatasi masalah dengan melakukan diagnosa terhadap suatu kata terlebih dahulu. Kata yang dimasukkan pengguna ke dalam sistem akan diidentifikasi dan menghasilkan sebuah konklusi berupa jenis dari kata tersebut. Berdasarkan hal tersebut, metode inferensi yang efektif bagi sistem ini adalah metode *Backward Chaining*. Metode ini dipilih karena lebih cocok untuk menyelesaikan masalah klasifikasi dan diagnosa [14].

3.2 Proses Rekayasa Pengetahuan

Tahap ini bertujuan untuk merepresentasikan pengetahuan yang telah dikumpulkan ke dalam basis pengetahuan. Basis pengetahuan sistem ini berisi tentang berbagai jenis kata dan turunannya (subjenis kata) beserta masing-masing cirinya. Setiap jenis kata utama (*isim*, *fi'il*, dan *harf*) memiliki ciri yang berbeda. Ciri-ciri kata dan jenis-jenis kata beserta relasinya yang digunakan pada penelitian ini disajikan pada tabel III-1 dan tabel III-2.

Tabel III-1 Ciri jenis kata

ID_CiriKata	Nama Ciri-Ciri Kata
C01	Termasuk kedalam kelompok huruf jar

ID_CiriKata	Nama Ciri-Ciri Kata
C02	Termasuk kedalam kelompok huruf nashab
C03	Termasuk kedalam kelompok huruf Jazm
C04	Termasuk ke dalam kelompok <i>isim jamak taksir</i>
C05	Dilekati <i>alif lam</i>
C06	Diakhiri dengan <i>tanwin</i>
C07	Didahului huruf <i>jar</i>
C08	Diakhiri dengan <i>ta marbutah</i> "ة"
C09	Diakhiri dengan uuni "ون"
C10	Diakhiri dengan <i>iina</i> "ين"
C11	Diakhiri dengan <i>aatun</i> "ات"
C12	Diakhiri dengan <i>aani</i> "ان"
C13	Diakhiri dengan <i>aini</i> "ين"
C14	Termasuk kedalam kelompok <i>isim isyarah</i>
C15	Termasuk kedalam kelompok <i>isim maushul</i>
C16	Termasuk kedalam kelompok <i>isim dhamir</i>
C17	Didahului huruf <i>qad</i> "قد"
C18	Didahului huruf <i>sin</i> "س"
C19	Didahului kata <i>saufa</i> "سوف"
C20	Didahului salah satu huruf <i>aniita</i> "ان, ي, ت"
C21	Didahului huruf <i>nashab</i>
C22	Didahului huruf <i>jazm</i>

Tabel III-1 berisi informasi tentang ciri-ciri jenis kata. Setiap ciri akan mendapatkan kode unik berupa ID_CiriKata dengan format Cxx. Dalam hal ini C merupakan singkatan dari Ciri dan xx merupakan nomor urut ciri jenis kata pada tabel.

Tabel III-2 Hubungan antara jenis kata dan ciri kata

ID_JenisKata	Jenis Kata	Ciri-ciri
JK01	Huruf Jar	C01
JK02	Huruf Nashab	C02
JK03	Huruf Jazm	C03
JK04	Isim Jamak Taksir	C13
JK05	Isim Jamak Mudzakar Salim	C05, C07, C09, C10
JK06	Isim Jamak Muannats Salim	C05, C07, C11
JK07	Isim Tatsniyah	C05, C07, C12, C13
JK08	Isim Mufrad Mudzakar	C05, C06, C07, !C08
JK09	Isim Mufraad Muannats	C05, C06, C07, C08
JK10	Isim Ma'rifat	C05, !C06, C14, C15, C16
JK11	Isim Nakirah	!C05, C06
JK12	Isim	C05, C06, C07
JK13	Fi'il	C17, C18, C19, C20, C21, C22

Tabel III-2 ini terdiri dari tiga kolom, ID_JenisKata, jenis kata, dan ciri kata. Kolom jenis kata berisi seluruh jenis kata yang digunakan pada sistem mengacu pada literatur bahasa Arab, baik itu jenis kata utama (*isim*, *fi'il*, dan *harf*) maupun subjenis kata yang merupakan turunan dari ketiga jenis kata utama. Adapun kolom ID_JenisKata terdiri dari kode unik yang dimiliki masing-masing jenis kata. Kode unik tersebut memiliki format JKxx dimana JK merupakan singkatan dari Jenis

Kata dan xx merupakan kode urut masing-masing jenis kata. Sebagai contoh, jenis kata *Huruf Jar* memiliki ID JK01. Kode 01 menunjukkan bahwa jenis kata *Huruf Jar* berada pada urutan pertama pada kolom Jenis Kata. Terdapat 13 jenis kata yang digunakan dalam sistem pakar ini.

Tabel III-2 menerangkan hubungan antara ciri dan jenis katanya. Jenis kata *isim jamak muannats salim* memiliki ciri-ciri diantaranya: dilekati *alif lam* atau diakhiri dengan tanwin dan/atau didahului huruf jar, dan diakhiri dengan *aani* "ان" atau diakhiri dengan *aini* "ين".

Pada dasarnya ketiga jenis kata utama tidak dapat memiliki ciri yang sama. Namun ketika jenis kata tersebut kita bagi lagi ke dalam sub-sub jenis kata, maka akan didapati bahwa satu ciri dimiliki oleh beberapa jenis kata (subjenis kata). Sebagai misal, *huruf jar*, *huruf nashab*, dan *huruf jazm* merupakan subjenis kata dari jenis kata *huruf*. Kendati begitu, dalam proses rekayasa kebutuhan ini penulis menyatukan jenis dan subjenis kata tersebut dalam tabel III-2 dengan judul kolom "Jenis Kata" guna mempermudah proses perancangan.

3.3 Implementasi

Pada tahap ini dilakukan analisis beserta pemodelannya. Analisis terdiri dari analisis kebutuhan non-fungsional yang terdiri dari analisis pengguna dan analisis perangkat lunak, kebutuhan fungsional, kaidah produksi, dan perancangan antarmuka.

Analisis kebutuhan fungsional pada aplikasi *mobile apps*, dalam hal ini adalah *chatbot*, dilakukan menggunakan *tools* STAR UML. Adapun tahapan analisis sistem pada penelitian ini menggunakan metode *Unified Modeling Language* (UML) meliputi *Use Case Diagram*, *Use Case Scenario*, *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram*.

3.3.1 Analisis Pengguna

Analisis pengguna sistem dimaksudkan untuk mengetahui siapa saja pengguna yang terlibat dalam sistem. Pengguna *Sistem Identifikasi Jenis Kata Dalam Bahasa Arab Menggunakan Metode Backward Chaining Berbasis Aplikasi Chatbot* terdiri dari *knowledge engineer* dan *user/pengguna* umum.

Setiap pengguna harus memenuhi kriteria yang telah ditetapkan. Seorang *knowledge engineer* harus mengerti penggunaan komputer, mampu mengoperasikan *smartphone*, terbiasa menggunakan media sosial Telegram karena sistem ini dijalankan pada layanan aplikasi tersebut, mengerti penggunaan DBMS MySQL guna melakukan manajemen data pengetahuan yang ada pada sistem, dan tentunya terbiasa dengan jenis huruf bahasa Arab berikut penggunaannya pada media elektronik.

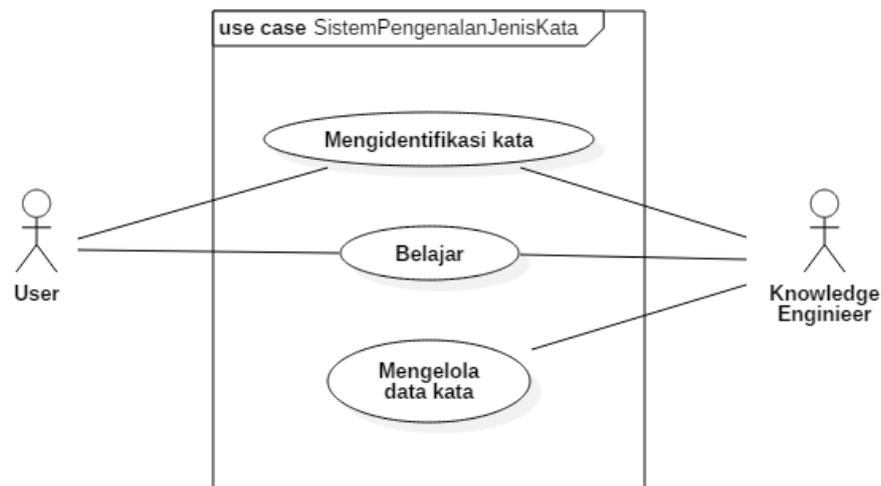
Adapun *user/pengguna* umum harus mampu menggunakan *smartphone* beserta konfigurasinya mengingat sistem ini akan menjalankan tipe *keyboard* bahasa Arab, terbiasa menggunakan media sosial Telegram dimana sistem ini dijalankan, dan ia merupakan seseorang yang sedang atau akan mempelajari bahasa arab.

3.3.2 Analisis Perangkat Lunak

Sistem ini dapat bekerja pada berbagai jenis perangkat lunak dimana aplikasi perpesanan Telegram dapat dijalankan. Berdasarkan situs resmi Telegram, aplikasi ini dapat dijalankan pada *platform* iOS (versi 8.0 dan setelahnya), Android (versi 4.1 dan setelahnya), dan Windows Phone. Bahkan terdapat versi web dan aplikasi desktop pada sistem operasi Windows, macOS, dan Linux.

3.3.3 Use Case Diagram

Use Case Diagram menggambarkan ada berapa banyak aktor dan *Use Case* pada sistem beserta interaksi antara keduanya. *Use Case Diagram* pada sistem pakar ini tercantum pada gambar III-1 berikut.



Gambar III.1 Rancangan *Use Case* Diagram

Pada gambar III-1 terdapat dua actor dan tiga *Use Case* yang dibutuhkan dalam sistem. Dua aktor tersebut adalah *User* dan *Knowledge Engineer*, keduanya akan dijelaskan pada subbab Definisi Aktor setelah ini. Sementara *Use Case* pada sistem ini terdiri dari *Use Case* Mengidentifikasi Kata, Belajar, dan Mengelola Data Kata. Seluruh *Use Case* akan dijelaskan pada subbab Definisi *Use Case*.

3.3.3.1 Definisi Aktor

Pada bagian ini setiap aktor yang terdapat pada sistem akan dideskripsikan berdasarkan aktifitas yang dapat dilakukannya dalam sistem. Aktor *Knowledge Engineer* dan *User* yang terdapat pada gambar III-1 akan dideskripsikan pada tabel III-4.

Tabel III-3 Definisi Aktor

No	Aktor	Deskripsi
1	<i>Knowledge Engineer</i>	Pengguna sistem sekaligus sebagai pengelola sistem yang dapat menambah,

No	Aktor	Deskripsi
		mengubah, dan menghapus basis pengetahuan dari sistem pakar.
2	<i>User</i>	Pengguna sistem yang dapat melakukan aktifitas identifikasi jenis kata pada sistem.

3.3.3.2 Definisi *Use Case*

Setiap *Use Case* yang terdapat pada gambar III-1 akan dideskripsikan pada subbab ini. Deskripsi yang dimaksud berupa penjabaran aktifitas yang terjadi pada sistem beserta actor yang terlibat. Definisi *Use Case* pada sistem ini tercantum pada tabel III-5 berikut.

Tabel III-4 Definisi *Use Case*

No	<i>Use Case</i>	Deskripsi
1	Pengidentifikasi Jenis Kata	Sistem melakukan identifikasi atau pengenalan pada kata yang dimasukkan pengguna
2	Belajar	Pengguna membuka materi pembelajaran
3	Pengelolaan Data Kata	<i>Knowledge Engineer</i> menambah, mengubah, dan menghapus data kata ke/di basis data

3.3.3.3 Use Case Scenario

Setiap *Use Case* yang tercantum pada gambar III-1 memiliki skenario masing-masing. Skenario ini merupakan alur proses yang terjadi pada *Use Case*. Kemudian *Use Case Scenario* pada sistem ini akan dijelaskan pada tabel III-5, III-6, dan III-7.

1. *Use Case* Pengidentifikasian Jenis Kata

Nama *Use Case*: Pengidentifikasian Jenis Kata

Skenario:

Tabel III-5 *Use Case Scenario* Pengidentifikasian Jenis Kata

<i>Aksi Aktor</i>	<i>Reaksi Sistem</i>
<i>Skenario Normal</i>	
1. Pengguna memilih tombol “/identifikasi” atau menuliskan command “/identifikasi”	
	2. Sistem menampilkan pesan tata cara identifikasi
3. Pengguna memasukkan kata/kalimat yang akan diidentifikasi	
	4. Sistem menampilkan hasil identifikasi
<i>Skenario Lain</i>	
1. Pengguna memasukkan kata/kalimat yang akan diidentifikasi	
	2. Sistem gagal mengidentifikasi dan menampilkan pesan gagal

2. *Use Case* BelajarNama *Use Case*: Belajar

Skenario:

Tabel III-6 *Use Case* Scenario Belajar

<i>Aksi Aktor</i>	<i>Reaksi Sistem</i>
<i>Skenario Normal</i>	
1. Pengguna memilih tombol “/belajar” atau menuliskan command “/belajar”	
	2. Sistem menampilkan menu pilihan materi belajar
3. Pengguna memilih materi belajar	
	4. Sistem menampilkan materi belajar yang dipilih
<i>Skenario Lain</i>	
-	-

3. Use Case Pengelolaan Data Kata

Nama Use Case: Pengelolaan Data Kata

Skenario:

Tabel III-7 Use Case Scenario Pengelolaan Data Kata

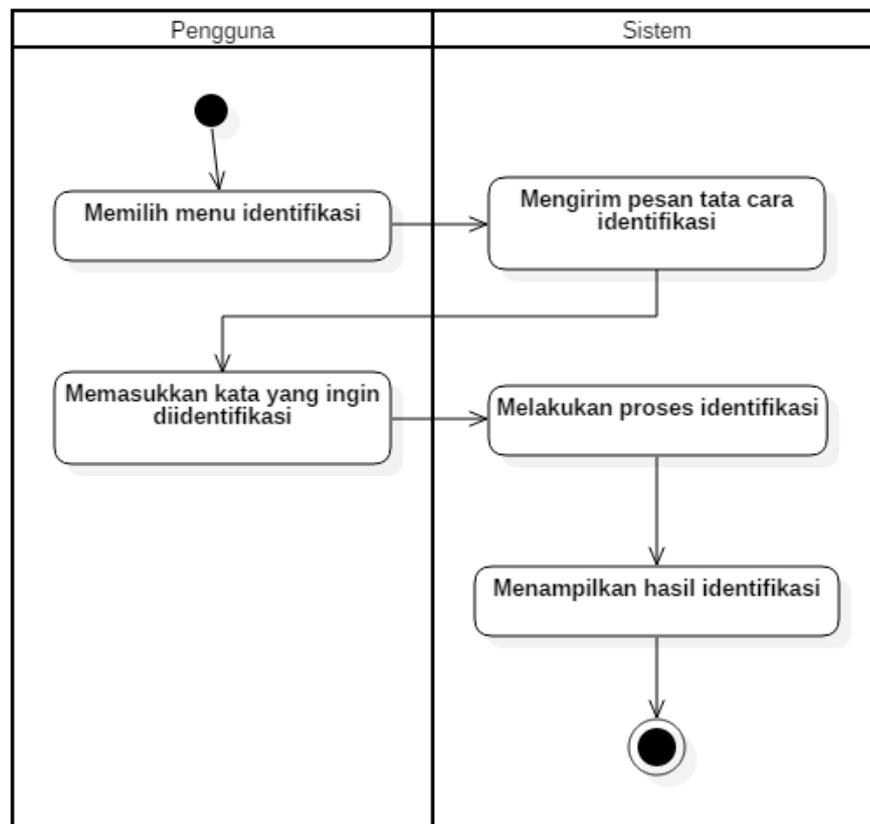
<i>Aksi Aktor</i>	<i>Reaksi Sistem</i>
<i>Skenario Normal</i>	
<i>1. Pengguna memilih tombol “/kelola” atau menuliskan command “/kelola”</i>	
	<i>2. Sistem menampilkan 3 menu: menu tambah, menu ubah, menu hapus.</i>
<i>3. Knowledge Engineer memilih salah satu menu</i>	
	<i>4. Sistem menampilkan pesan tentang tata cara kelola data</i>
<i>5. Knowledge Engineer memasukkan kata beserta makna yang ingin ditambah, diubah, atau dihapus</i>	
	<i>6. Sistem menampilkan pesan bahwa aktifitas pengelolaan data kata telah berhasil dilakukan</i>
<i>Skenario Lain</i>	
<i>1. Knowledge Engineer memasukkan kata beserta makna yang ingin ditambah, diubah, atau dihapus</i>	

<i>Aksi Aktor</i>	<i>Reaksi Sistem</i>
	2. Sistem menampilkan pesan bahwa aktifitas pengelolaan data kata gagal dilakukan

3.3.4 Activity Diagram

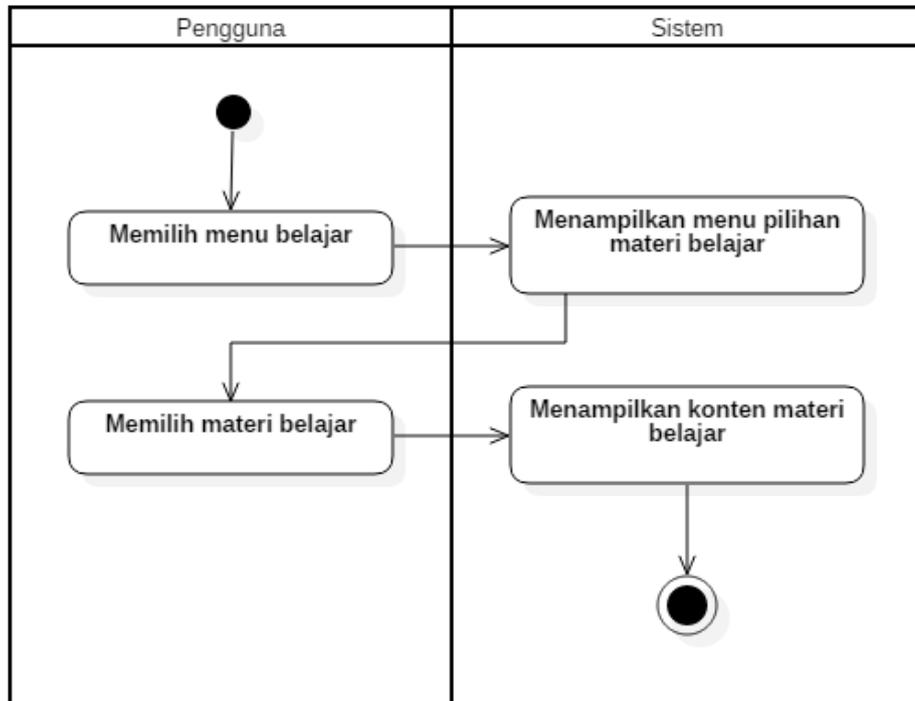
Activity Diagram memodelkan aliran kerja dari urutan aktifitas pada suatu proses yang mengacu pada *Use Case diagram* yang ada. Setiap *activity diagram* pada aplikasi *Sistem Identifikasi Jenis Kata Dalam Bahasa Arab Dengan Metode Backward Chaining Berbasis Aplikasi Chatbot* akan dijelaskan pada gambar III-2, III-3, dan III-4.

1. Activity Diagram Mengidentifikasi Jenis Kata



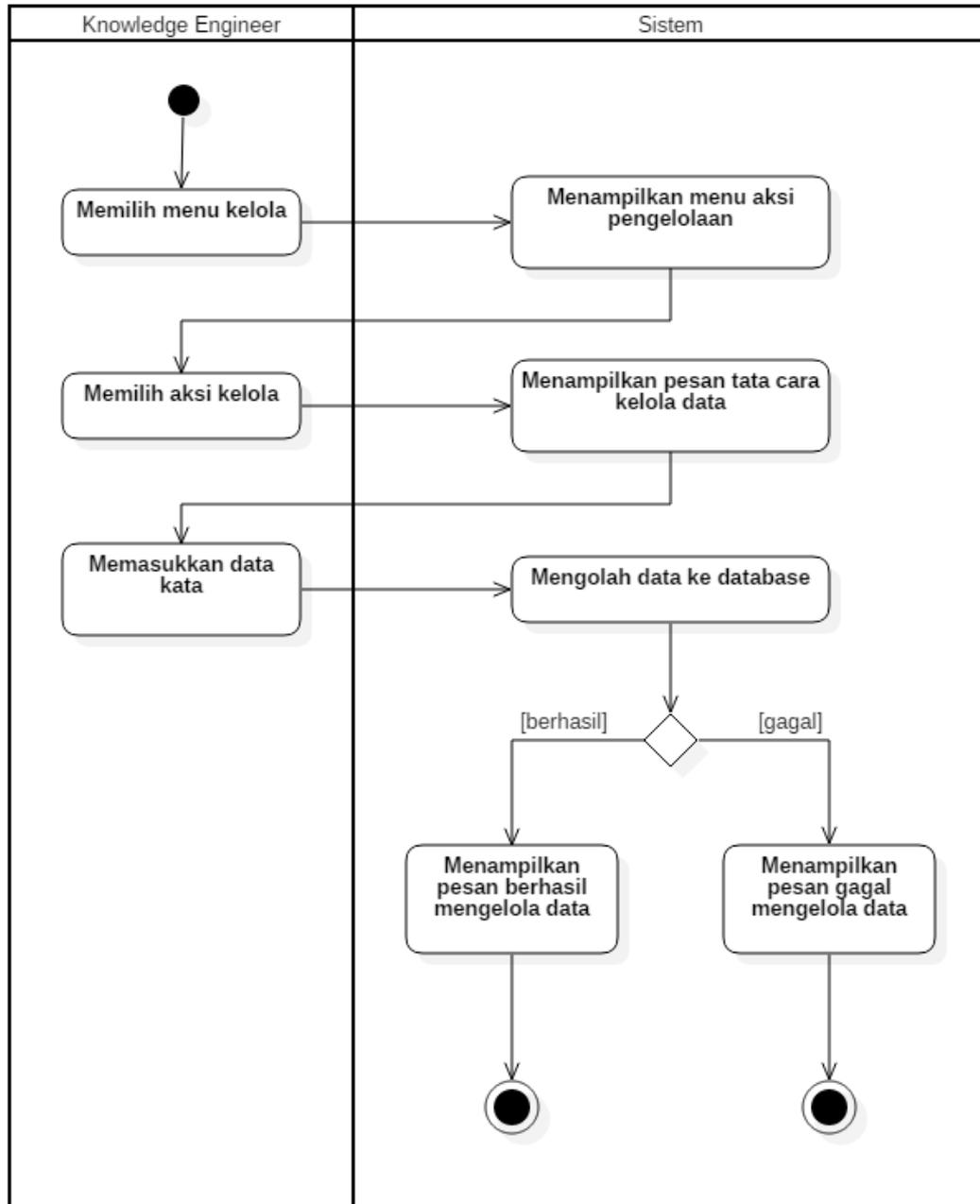
Gambar III.2 Activity Mengidentifikasi jenis kata

2. Activity Diagram Belajar



Gambar III.3 Activity Belajar

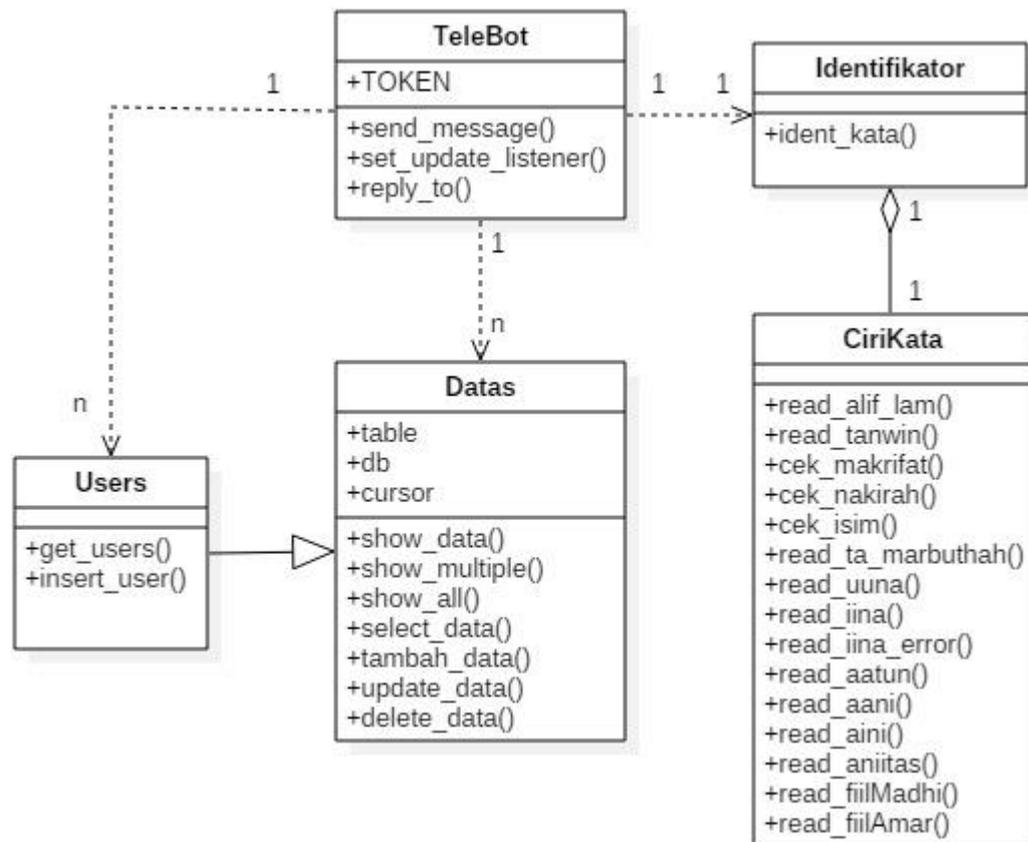
3. Activity Diagram Pengelolaan Data Kata



Gambar III.4 Activity Pengelolaan data kata

3.3.5 Class Diagram

Class Diagram menggambarkan model statis dari *class* yang ada pada sistem berisi struktur dan deskripsi *class* serta relasi antar *class*. *Class Diagram* pada sistem ini dapat dilihat pada gambar III.5 berikut.



Gambar III.5 class diagram

Nama Kelas: TeleBot

Tabel III-8 Rincian operasi dan atribut kelas TeleBot

Nama Operasi	Visibility (private, public)
+ send_message(chat_id: int, text: String, parse_mode: Boolean, reply_markup: Dictionary)	Public
+ set_update_listener()	Public
+ reply_to(message: String, text: String)	Public
Nama Atribut	Visibility (private, public)
+ TOKEN: String	Public

Nama Kelas: Datas

Tabel III-9 Rincian operasi dan atribut kelas Datas

Nama Operasi	Visibility (private, public)
+ show_data()	Public
+ show_multiple()	Public
+ show_all()	Public
+ select_data(field: String, name: String, index: Integer)	Public
+ tambah_data(f1: String, f2: String, m1: String, m2: String)	Public
+ update_data(val1: String, col: String, val2: String)	Public
+ delete_data(col: String, val1: String)	Public
Nama Atribut	Visibility (private, public)
+ table: String	Public
+ db: Dictionary	Public
+ cursor: Query	Public

Nama Kelas: Users

Tabel III-10 Rincian operasi dan atribut kelas Users

Nama Operasi	Visibility (private, public)
+ get_users() + insert_user(uid: Integer)	Public Public
Nama Atribut	Visibility (private, public)

Nama Kelas: Identifikator

Tabel III-11 Rincian operasi dan atribut kelas Identifikator

Nama Operasi	Visibility (private, public)
+ ident_kata()	Public
Nama Atribut	Visibility (private, public)

Nama Kelas: CiriKata

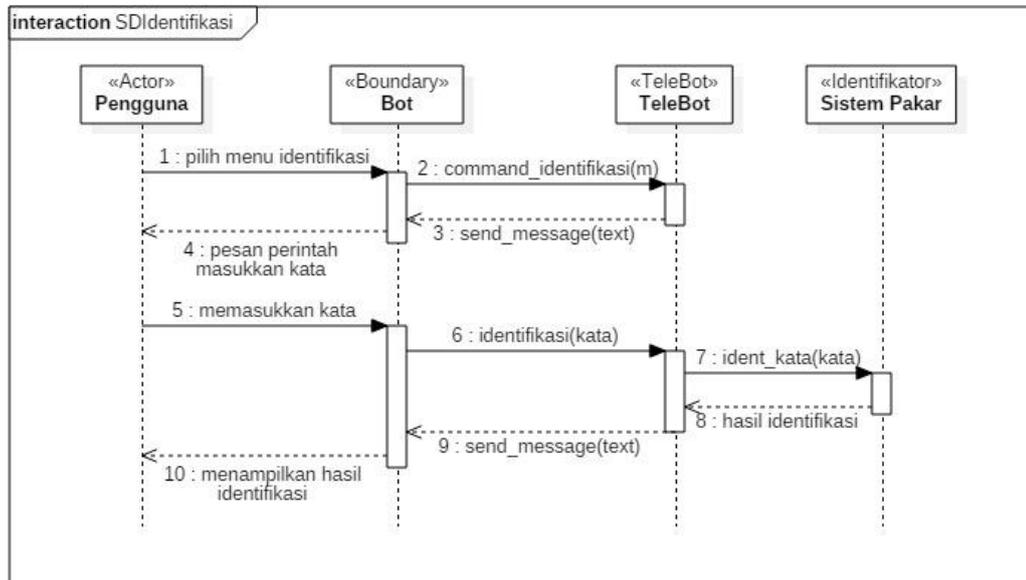
Tabel III-12 Rincian operasi dan atribut kelas CiriKata

Nama Operasi	Visibility (private, public)
+ read_alif_lam(msg: String)	Public
+ read_tanwin(msg: String)	Public
+ cek_makrifat(msg: String)	Public
+ cek_nakirah(msg: String)	Public
+ cek_isim(msg: String)	Public
+ read_ta_marbuthah(msg: String)	Public
+ read_uuna(msg: String)	Public
+ read_iina(msg: String)	Public
+ read_iina_error(msg: String)	Public
+ read_aatun(msg: String)	Public
+ read_aani(msg: String)	Public
+ read_aini(msg: String)	Public
+ read_aniitas(msg: String)	Public
+ read_fiilMadhi(msg: String)	Public
+ read_fiilAmar(msg: String)	Public
Nama Atribut	Visibility (private, public)

3.3.6 Sequence Diagram

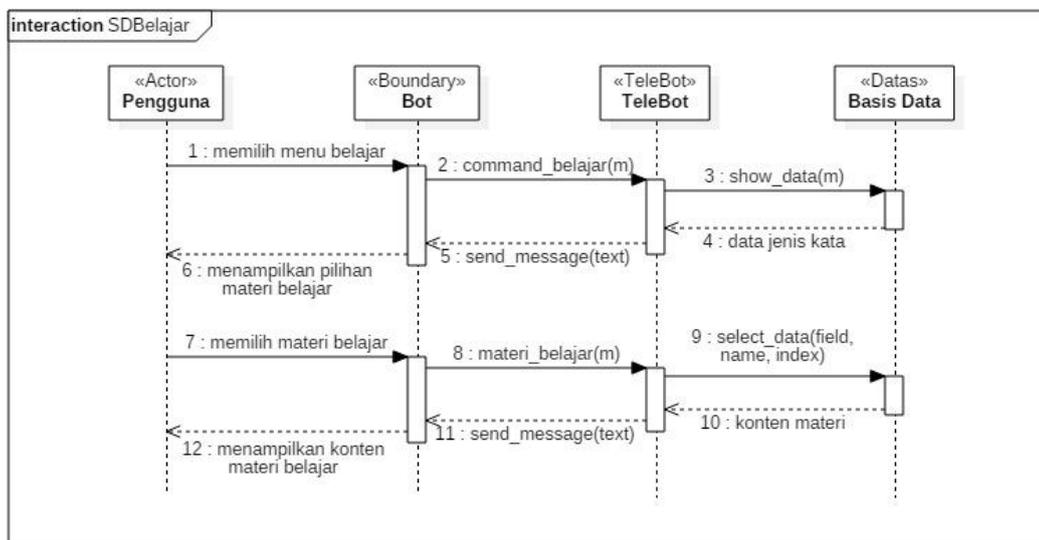
Kolaborasi dinamis antar sejumlah objek yang bertujuan untuk menunjukkan rangkaian pesan yang dikirim antar objek beserta interaksinya. *Sequence Diagram* pada sistem ini termuat dalam subbab ini pada tabel III-6, III-7, dan III-8.

1. Sequence Mengidentifikasi Kata



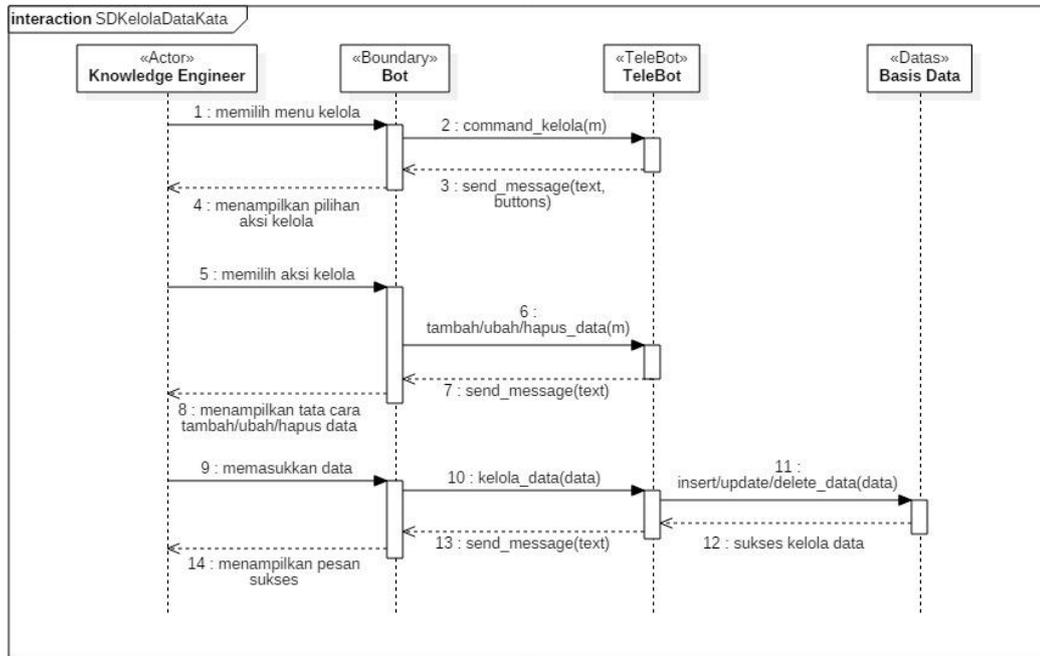
Gambar III.6 Sequence mengidentifikasi kata

2. Sequence Belajar



Gambar III.7 Sequence Belajar

3. Sequence Mengelola Data Kata



Gambar III.8 Sequence mengelola data kata

3.3.7 Kaidah Produksi

Pengetahuan yang telah dikumpulkan pada subbab 3.2 akan direpresentasikan ke dalam kode. Metode yang digunakan pada penelitian ini adalah metode kaidah produksi. Metode ini menghubungkan antara premis dan konklusi menggunakan bentuk aturan JIKA ... MAKA Adapun representasi pengetahuan yang dimaksud berupa sekumpulan aturan sebagaimana tercantum pada tabel III-16 berikut.

Tabel III-13 Kaidah Produksi

Nama Aturan	Bunyi Aturan
Aturan 1	JIKA termasuk ke dalam kelompok <i>huruf Jar</i> MAKA jenis katanya adalah <i>huruf Jar</i>
Aturan 2	JIKA termasuk ke dalam kelompok <i>huruf Nashab</i> MAKA jenis katanya adalah <i>huruf Nashab</i>

Nama Aturan	Bunyi Aturan
Aturan 3	JIKA termasuk ke dalam kelompok <i>huruf Jazm</i> MAKA jenis katanya adalah <i>huruf Jazm</i>
Aturan 4	JIKA termasuk ke dalam kelompok <i>isim jamak taksir</i> MAKA jenis katanya adalah <i>Isim Jamak Taksir</i>
Aturan 5	JIKA termasuk ke dalam kelompok <i>isim isyarah</i> ATAU termasuk ke dalam kelompok <i>isim maushul</i> ATAU termasuk ke dalam kelompok <i>isim dhamir</i> MAKA jenis katanya adalah <i>Isim Ma'rifat</i>
Aturan 6.1	JIKA dilekati <i>alif lam</i> DAN diakhiri dengan <i>uuni</i> "ون" ATAU Diakhiri dengan <i>iina</i> "ين" MAKA jenis hurufnya adalah <i>Isim Jamak Mudzakar Salim</i> dan <i>Ma'rifat</i>
Aturan 6.2	JIKA TIDAK dilekati <i>alif lam</i> DAN diakhiri dengan <i>uuni</i> "ون" ATAU Diakhiri dengan <i>iina</i> "ين" MAKA jenis hurufnya adalah <i>Isim Jamak Mudzakar Salim</i> dan <i>Nakirah</i>
Aturan 7.1	JIKA dilekati <i>alif lam</i> DAN diakhiri dengan <i>aatun</i> "ات" MAKA jenis hurufnya adalah <i>Isim Jamak Muannats Salim</i> dan <i>Ma'rifat</i>
Aturan 7.2	JIKA TIDAK dilekati <i>alif lam</i> DAN diakhiri dengan <i>aatun</i> "ات" MAKA jenis hurufnya adalah <i>Isim Jamak Muannats Salim</i> dan <i>Nakirah</i>
Aturan 8.1	JIKA dilekati <i>alif lam</i> DAN diakhiri dengan <i>aani</i> "ان" ATAU diakhiri dengan <i>aini</i> "ين" MAKA jenis hurufnya adalah <i>Isim Tatsniyah</i> dan <i>Ma'rifat</i>
Aturan 8.2	JIKA TIDAK dilekati <i>alif lam</i> DAN diakhiri dengan <i>aani</i> "ان" ATAU diakhiri dengan <i>aini</i> "ين" MAKA jenis hurufnya adalah <i>Isim Tatsniyah</i> dan <i>Nakirah</i>
Aturan 9.1	JIKA dilekati <i>alif lam</i>

Nama Aturan	Bunyi Aturan
	DAN TIDAK diakhiri dengan <i>tanwin</i> DAN TIDAK diakhiri dengan <i>ta marbutah</i> "ة" MAKA jenis katanya adalah <i>Isim Mufrad Mudzakar</i> dan <i>Ma'rifat</i>
Aturan 9.2	JIKA TIDAK dilekati <i>alif lam</i> DAN diakhiri dengan <i>tanwin</i> DAN TIDAK diakhiri dengan <i>ta marbutah</i> "ة" MAKA jenis katanya adalah <i>Isim Mufrad Mudzakar</i> dan <i>Nakirah</i>
Aturan 10.1	JIKA dilekati <i>alif lam</i> DAN TIDAK diakhiri dengan <i>tanwin</i> DAN diakhiri dengan <i>ta marbutah</i> "ة" MAKA jenis katanya adalah <i>Isim Mufrad Muannats</i> dan <i>Ma'rifat</i>
Aturan 10.2	JIKA TIDAK dilekati <i>alif lam</i> DAN diakhiri dengan <i>tanwin</i> DAN diakhiri dengan <i>ta marbutah</i> "ة" MAKA jenis katanya adalah <i>Isim Mufrad Muannats</i> dan <i>Nakirah</i>
Aturan 11	JIKA dilekati <i>alif lam</i> ATAU diakhiri dengan <i>tanwin</i> ATAU didahului <i>huruf jar</i> MAKA jenis hurufnya termasuk <i>Isim</i>
Aturan 12	JIKA didahului <i>huruf qad</i> "قد" ATAU didahului <i>huruf sin</i> "س" ATAU didahului kata <i>saufa</i> "سوف" ATAU didahului salah satu huruf <i>aniita</i> "ا, ن, ي, ت" ATAU didahului <i>huruf nashab</i> ATAU didahului <i>huruf jazm</i> MAKA jenis hurufnya termasuk <i>Fiil</i>

Pada tabel III-13 diatas dapat dilihat bahwa terdapat tujuh belas kaidah atau aturan yang digunakan pada sistem pakar ini. Seluruh kaidah tersebut mengacu pada tabel III-2. Kaidah tersebut terdiri dari dua jenis kaidah, yakni kaidah derajat pertama (*first order rule*) sebagaimana diterapkan pada aturan 1 s.d. aturan 4, dan

kaidah meta (*meta rule*) yang penerapannya dapat kita jumpai pada aturan 5 s.d. aturan 17.

Urutan aturan tersebut dirancang sedemikian rupa mengacu kepada tingkat kekhasan ciri kata. Ciri kata yang tingkat kekhasannya paling tinggi akan ditempatkan paling awal. Adapun jenis kata yang memiliki ciri paling umum atau tingkat kekhasan paling rendah ditempatkan pada aturan paling akhir.

3.3.8 Analisis API Bot Telegram

Jacobson menuturkan bahwa *Application Programming Interface* (API) adalah antarmuka yang dapat dibaca oleh mesin yang menghubungkan beberapa aplikasi, menyediakan metode untuk mengatur interaksi aplikasi, dan menjalin komunikasi antar aplikasi tanpa perlu mengetahui cara kerja internal tentang bagaimana fungsionalitas API disediakan [15]. Laman resmi Telegram menjelaskan bahwa API bot Telegram adalah sebuah antarmuka berbasis *Hypertext Transfer Protocol* (HTTP) yang dibuat untuk pengembang yang ingin menciptakan bot pada Telegram [16].

Penelitian ini memanfaatkan API Telegram dalam pembuatan bot. API digunakan untuk mendapatkan data-data berupa informasi pengguna dan isi pesan pengguna. Data-data tersebut diolah oleh program kemudian hasilnya akan dikirimkan kembali kepada pengguna melalui API tersebut.

Pengaksesan API ini dilakukan dengan cara mengirimkan *request* berupa *query*. Setiap *query* yang ditujukan kepada API harus dikirimkan melalui *Hypertext Transfer Protocol Secure* (HTTPS) dengan format “https://api.telegram.org/bot<token>/METHOD_NAME”. Token didapatkan dari bot yang telah dibuat pada Telegram. Terdapat banyak sekali metode (*methods*) yang disediakan oleh Telegram. Namun pada penelitian ini hanya satu metode yang digunakan yakni metode “sendMessage”.

Metode tersebut digunakan untuk menampilkan menu utama, pilihan materi belajar, menu kelola data, hasil identifikasi, materi pembelajaran, dan data hasil kelola, serta seluruh pesan yang disampaikan sistem kepada pengguna.

Penulis menggunakan modul `pyTelegramBotAPI` untuk memudahkan pengaksesan API *bot* Telegram. Modul ini merupakan salah satu modul pada bahasa pemrograman Python yang dikembangkan untuk memudahkan para pengembang yang ingin mengakses API *bot* Telegram.

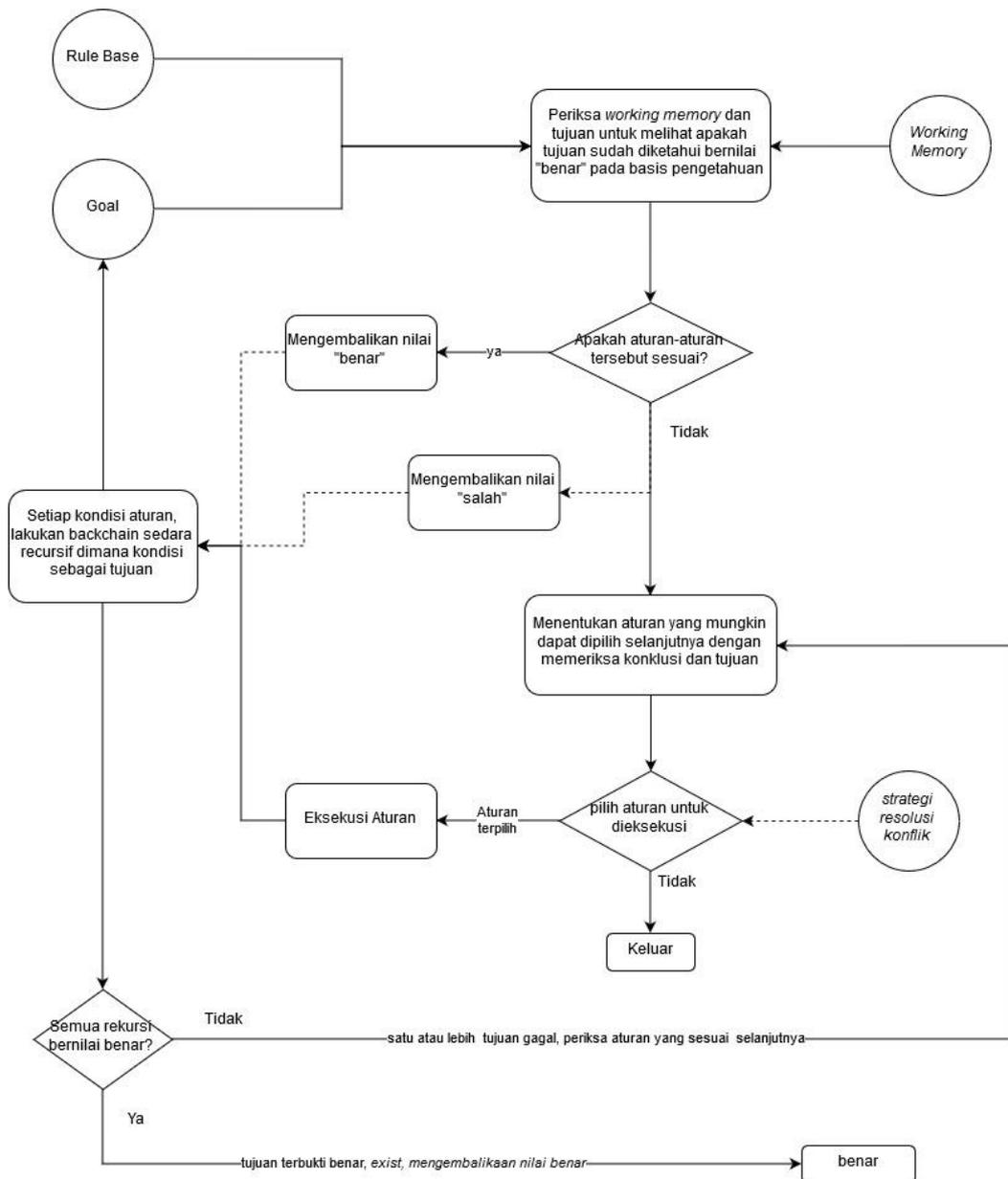
3.3.9 Analisis *Backward Chaining*

Backward chaining adalah sebuah bentuk pemikiran yang dikendalikan oleh tujuan atau *goal* [11]. Maksudnya adalah ketika proses penalaran ini dimulai, maka sistem akan membuat sebuah hipotesis konklusi. Hipotesis konklusi ini akan menjadi patokan penelusuran premis-premis. Premis-premis ini akan dicocokkan dengan fakta-fakta yang diberikan oleh pengguna. Apabila didapatkan kecocokkan, maka proses penelusuran akan dihentikan dan hipotesis konklusi akan dideklarasikan sebagai konklusi yang sebenarnya. Namun apabila tidak ditemukan kecocokan antara keduanya, maka sistem akan membuat hipotesis konklusi selanjutnya dan melakukan proses penelusuran yang lebih lanjut.

Adapun siklus pengontrolan *backward chaining* pada umumnya sebagaimana tercantum pada gambar III.9 adalah sebagai berikut [17]:

1. Mengecek konklusi dari aturan-aturan untuk menemukan setiap aturan yang dapat memenuhi tujuan teratas dari *stack*.
2. Proses aturan-aturan ini satu per satu:
 - a. Mengevaluasi kondisi dalam aturan If satu demi satu:
 - i. Apabila kondisi tidak diketahui oleh sistem (tidak terdapat pada basis data), *push* tujuan ke dalam *stack* lalu lakukan pemanggilan sistem secara rekursif.
 - ii. Apabila kondisi diketahui tetapi tidak sesuai dengan fakta, maka lanjutkan dengan melakukan pengulangan pada langkah 2.

- iii. Apabila tidak dapat menentukan apakah kondisi terpenuhi atau tidak, maka lanjutkan dengan melakukan pengulangan pada langkah 2.
- b. Jika semua kondisi dalam aturan yang dipilih terpenuhi, tambahkan fakta-fakta tersebut ke *working memory* pada bagian aturan *Then*. Lalu *pop* tujuan dari *stack* kemudian lakukan pemanggilan sistem.



Gambar III.9 Diagram alir metode *backward chaining* [17]

Sederhananya, urutan pengambilan hipotesis konklusi pada sistem ini mengacu kepada kaidah produksi yang tercantum pada tabel III-13. Sesuai tabel tersebut, proses penalaran akan dimulai dengan mengambil jenis kata huruf *Jar* sebagai hipotesis konklusi. Kemudian sistem akan mengecek premis yang berhubungan dengan jenis kata huruf *Jar*, dalam hal ini adalah ciri C01 yang berbunyi “termasuk ke dalam huruf *jar*”. Sehingga apabila kata yang dimasukkan pengguna termasuk ke dalam huruf *jar*, maka sistem akan menghentikan proses penelusuran dan mendeklarasikan jenis kata huruf *jar* sebagai konklusi. Pada akhirnya konklusi tersebut diteruskan kepada pengguna. Namun apabila kata tersebut tidak termasuk ke dalam jenis kata huruf *jar*, maka sistem akan mengambil aturan 2 dimana huruf *nashab* menjadi hipotesis konklusi. Menelusuri premis-premis yang berkaitan kemudian mencocokkannya dengan fakta-fakta dan begitu selanjutnya.

3.3.10 Perancangan Basis Data

Pada penelitian ini, penulis membuat sebuah basis data untuk mendukung sistem yang dibuat. Sebagian besar basis data ini dibuat untuk menyimpan kata-kata bahasa Arab yang tidak memiliki ciri khusus sehingga tidak dapat dikenali hanya dengan melihat struktur katanya. Kata-kata tersebut merupakan bagian dari jenis kata *huruf* dan sebagian dari jenis kata *isim*. Adapun *isim* yang dimaksud adalah *isim jamak taksir*, *isim dhamir*, *isim maushul*, dan *isim isyarah*.

Basis data ini juga berfungsi untuk menyimpan data jenis kata berikut dengan deskripsi dan contohnya serta data *user* yang telah menggunakan aplikasi *chatbot* yang dibuat. Data-data tersebut disimpan dalam tabel yang terpisah. Tabel yang terdapat pada basis data sistem ini berjumlah sepuluh buah. Berikut rincian tabel-tabel yang terdapat pada basis data:

1. Tabel *users*

Rincian tabel *users* dapat dilihat pada tabel III-14. Terdapat 6 *field* pada tabel ini, yaitu “uid”, “step”, ”level”, ”username”, “firstname”, dan ”lastname”. *Field* “uid” merupakan primary key pada tabel ini dimana nilainya berupa *chat_id* dari

pengguna. “Step” berupa nilai *integer* 0, 100, 110, 120, 130, atau 140. Nilai 0 menandakan pengguna baru membuka aplikasi, 100 menandakan ia berada pada menu identifikasi, 110 untuk menu belajar, 120 menu kelola bagian tambah data, 130 menu kelola bagian hapus data, dan 140 menu kelola bagian hapus data. “Level” menunjukkan tingkatan pengguna apakah “admin” atau “user”. “Username”, “firstname”, dan “lastname” merupakan data nama pengguna yang didapatkan dari layanan API *bot* Telegram.

Tabel III-14 Rincian tabel data pengguna

No	Nama Kolom	Tipe Data	Keterangan
1.	uid	int(10)	Primary Key
2.	step	int(3)	
3.	level	enum(‘user’, ‘admin’)	
4.	username	varchar(25)	
5.	firstname	varchar(25)	
6.	Lastname	varchar(25)	

2. Tabel jenis_kata

Tabel ini terdiri dari empat *field*: “id_jenis_kata” yang merupakan *primary key*; “jenis_kata” berisi kumpulan jenis kata bahasa Arab; “deskripsi_lengkap” merupakan deskripsi dari setiap jenis kata beserta contoh kata-katanya. *Field* inilah yang dipanggil ketika pengguna memilih materi pembelajaran; Dan terakhir adalah *field* “deskripsi_singkat” berisi deskripsi jenis kata yang lebih ringkas. Data dari *field* akan dimunculkan bersama hasil identifikasi. Rinciannya dapat dilihat pada tabel III-15.

Tabel III-15 Rincian tabel informasi jenis kata

No	Nama Kolom	Tipe Data	Keterangan
1.	id_jenis_kata	int(2)	Primary Key
2.	jenis_kata	varchar(25)	
3.	deskripsi_lengkap	longtext	
4.	deskripsi_singkat	mediumtext	

3. Tabel hurufjar

Tabel III-16 s.d. tabel III-23 merupakan tabel kumpulan kata-kata dan huruf-huruf beserta maknanya. Tabel-tabel tersebut terdiri dari tiga *field*, yakni id, kata, dan makna. *Field* “id” terdiri dari dua angka bertipe integer, “kata” berisi maksimal 25 karakter tipe varchar yang ditulis dengan huruf Arab, dan “makna” bertipe data varchar dengan isi maksimum 30 karakter dengan huruf latin.

Pada setiap tabel, judul *field* “id” berbeda-beda menyesuaikan dengan penamaan tabelnya. Misal pada tabel huruf *jar*, judul *field* id nya adalah “jarr_id” sebagaimana tercantum pada tabel III-16. Begitupun pada tabel-tabel yang lain. Berikut adalah tabel huruf *jar* yang terdiri dari tiga *field*, yakni “jar_id”, “kata”, dan “makna”. Tabel ini digunakan untuk menyimpan berbagai data kata huruf *jar* beserta maknanya.

Tabel III-16 Rincian tabel huruf *jar*

No	Nama Kolom	Tipe Data	Keterangan
1.	jarr_id	int(2)	Primary Key
2.	Kata	varchar(25)	
3.	Makna	varchar(30)	

4. Tabel hurufjazm

Tabel hurufjazm digunakan untuk menyimpan berbagai data kata huruf *jazm* beserta maknanya. Tabel ini terdiri dari *field* “jazm_id”, “kata”, dan “makna”. Adapun rinciannya dapat dilihat pada tabel III-17.

Tabel III-17 Rincian tabel huruf *jazm*

No	Nama Kolom	Tipe Data	Keterangan
1.	jazm_id	int(2)	Primary Key
2.	kata	varchar(25)	
3.	makna	varchar(30)	

5. Tabel hurufnashab_is

Tabel hurufnashab_is digunakan untuk menyimpan berbagai data kata huruf *nashab* pada *isim* beserta maknanya. Tabel ini terdiri dari *field* “nashis_id”, “kata”, dan “makna”. Adapun rinciannya dapat dilihat pada tabel III-18.

Tabel III-18 Rincian tabel huruf *nashab* pada *isim*

No	Nama Kolom	Tipe Data	Keterangan
1.	nashis_id	int(2)	Primary Key
2.	kata	varchar(25)	
3.	makna	varchar(30)	

6. Tabel hurufnashab_fi

Tabel hurufnashab_fi digunakan untuk menyimpan berbagai data kata huruf *nashab* pada *fi'il* beserta maknanya. Tabel ini terdiri dari *field* “nashfi_id”, “kata”, dan “makna”. Adapun rinciannya dapat dilihat pada tabel III-19.

Tabel III-19 Rincian tabel huruf *nashab* pada *fi'il*

No	Nama Kolom	Tipe Data	Keterangan
1.	nashfi_id	int(2)	Primary Key
2.	kata	varchar(25)	
3.	makna	varchar(30)	

7. Tabel isim_dhamir

Tabel isim_dhamir digunakan untuk menyimpan berbagai data kata *isim dhamir* beserta maknanya. Tabel ini terdiri dari *field* “dhamir_id”, “kata”, dan “makna”. Adapun rinciannya dapat dilihat pada tabel III-20.

Tabel III-20 Rincian tabel *isim dhamir*

No	Nama Kolom	Tipe Data	Keterangan
1.	dhamir_id	int(2)	Primary Key
2.	kata	varchar(25)	
3.	makna	varchar(30)	

8. Tabel isim_isyarah

Tabel isim_isyarah digunakan untuk menyimpan berbagai data kata *isim isyarah* beserta maknanya. Tabel ini terdiri dari *field* “isy_id”, “kata”, dan “makna”. Adapun rinciannya dapat dilihat pada tabel III-21.

Tabel III-21 Rincian tabel *isim isyarah*

No	Nama Kolom	Tipe Data	Keterangan
1.	isy_id	int(2)	Primary Key
2.	kata	varchar(25)	

No	Nama Kolom	Tipe Data	Keterangan
3.	makna	varchar(30)	

9. Tabel isim_maushul

Tabel isim_maushul digunakan untuk menyimpan berbagai data kata *isim maushul* beserta maknanya. Tabel ini terdiri dari *field* “maush_id”, “kata”, dan “makna”. Adapun rinciannya dapat dilihat pada tabel III-22.

Tabel III-22 Rincian tabel *isim maushul*

No	Nama Kolom	Tipe Data	Keterangan
1.	maush_id	int(2)	Primary Key
2.	kata	varchar(25)	
3.	makna	varchar(30)	

10. Tabel jamak_taksir

Tabel jamak_taksir digunakan untuk menyimpan berbagai data kata *isim jamak taksir* beserta maknanya. Tabel ini terdiri dari *field* “jt_id”, “kata”, dan “makna”. Adapun rinciannya dapat dilihat pada tabel III-23.

Tabel III-23 Rincian tabel *isim jamak taksir*

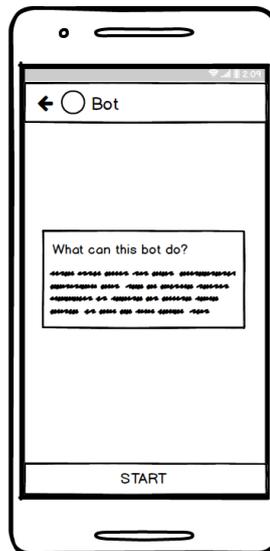
No	Nama Kolom	Tipe Data	Keterangan
1.	jt_id	int(11)	Primary Key
2.	kata	varchar(25)	
3.	makna	varchar(30)	

3.3.11 Perancangan Antarmuka

Antarmuka pengguna (*user interface*) adalah bagian dari sistem yang bertindak sebagai perantara antara pengguna dan sistem yang memfasilitasi pengguna untuk berinteraksi dengan sistem secara efisien [18]. Antarmuka pada sistem pakar identifikasi ini berupa tampilan *chatting* Telegram yang berfungsi sebagai pengendali *input-output*. Tampilan antarmuka dapat dilihat pada gambar III.10-III.16.

1. Tampilan awal

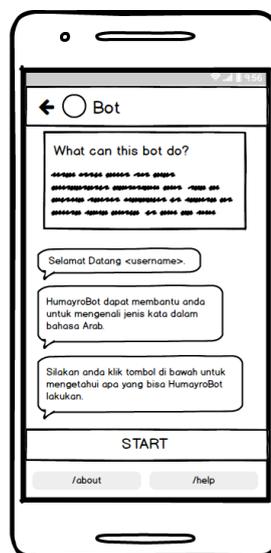
Gambar III.10 merupakan tampilan awal ketika pengguna pertama kali masuk ke dalam aplikasi. Pengguna akan disuguhkan tulisan tentang apa yang dapat dilakukan oleh aplikasi *chatbot* ini. Tulisannya berbunyi “What can this bot do? Humayro merupakan aplikasi BOT yang diciptakan sebagai media pembelajaran bahasa Arab.”



Gambar III.10 Tampilan awal *chatbot*

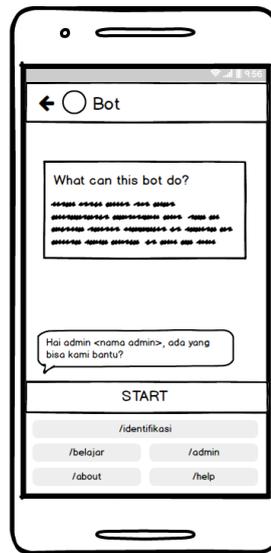
Ketika tombol “START” ditekan, sistem akan merespon sesuai dengan kondisi pengguna. Terdapat tiga kondisi dengan responnya masing-masing sebagai berikut:

- a. Apabila pengguna baru pertama kali menggunakan aplikasi ini dan belum terdaftar pada basis data, pesan yang ditampilkan adalah “Selamat datang <username>. HumayroBot dapat membantu anda untuk mengenali jenis kata dalam bahasa Arab. Silakan anda klik tombol di bawah untuk mengetahui apa yang bisa HumayroBot lakukan.” Bersamaan dengan pesan tersebut sistem menampilkan tombol menu “/about” dan “/help”. Gambar III.11 memperlihatkan tampilan sistem ketika merespon kondisi “a”.



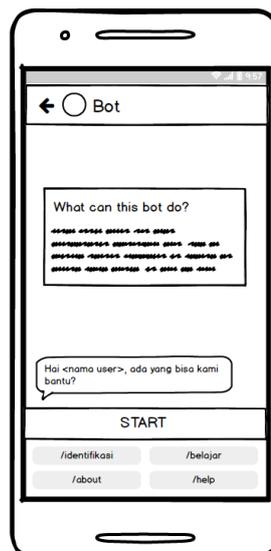
Gambar III.11 Tampilan respon sistem pada kondisi "a"

- b. Apabila pengguna sudah pernah menggunakan aplikasi, terdaftar pada sistem, dan memiliki predikat “admin”, maka pesan yang ditampilkan adalah “Hai admin <nama admin>, ada yang bisa kami bantu?” Bersamaan dengan pesan tersebut sistem menampilkan tombol menu “/identifikasi”, “/belajar”, “/admin”, “/about”, dan “/help”. Gambar III.12 memperlihatkan tampilan sistem ketika merespon kondisi “b”.



Gambar III.12 Tampilan respon sistem pada kondisi "b"

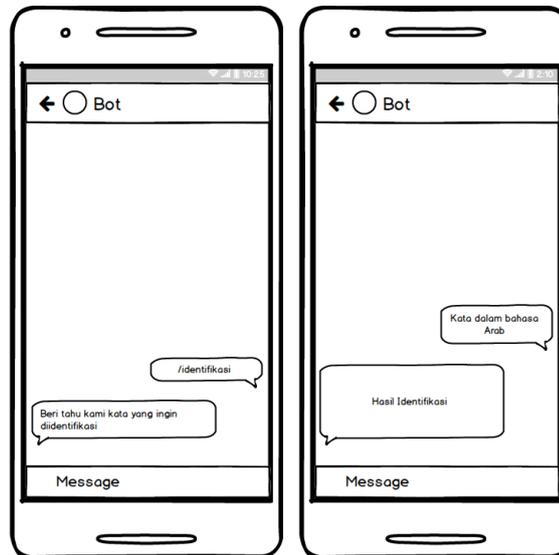
- c. Apabila pengguna sudah pernah menggunakan aplikasi, terdaftar pada sistem, dan memiliki predikat “user”, maka pesan yang ditampilkan adalah “Hai <nama user>, ada yang bisa kami bantu?”. Bersamaan dengan pesan tersebut sistem menampilkan tombol menu layaknya pada kondisi “b” tanpa menampilkan tombol “/admin”. Gambar III.13 memperlihatkan tampilan sistem ketika merespon kondisi “c”.



Gambar III.13 Tampilan respon sistem pada kondisi "c"

2. Tampilan fungsi identifikasi

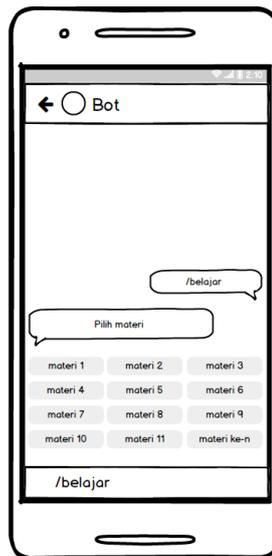
Setelah memilih menu “/identifikasi”, sistem akan mempersilakan pengguna untuk memasukkan kata yang ingin diidentifikasi seperti pada gambar III.14 sebelah kiri. Pengguna dapat memasukkan kata dalam bahasa Arab dengan tipe *keyboard* Arabic tentunya, kemudian sistem akan mengirimkan hasil identifikasi berupa pesan balasan seperti pada gambar III.14 sebelah kanan. Perlu diperhatikan bahwa pada keadaan fungsi identifikasi ini sistem hanya akan menerima huruf *hijaiyah*. Sehingga apabila pengguna memasukan huruf alfabet sistem akan mengirim pesan peringatan yang berbunyi “Mohon maaf, kami hanya bisa mengidentifikasi bahasa Arab :)”.



Gambar III.14 Tampilan fungsi identifikasi: tampilan saat pengguna memilih menu “/identifikasi” (kiri) dan tampilan saat pengguna melakukan kegiatan identifikasi (kanan).

3. Tampilan fungsi belajar

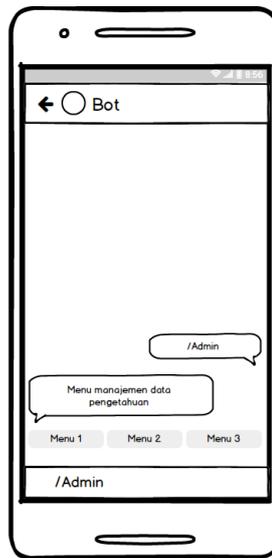
Menu pilihan materi belajar akan muncul ketika pengguna memilih menu “/belajar” dengan menekan tombol “/belajar” atau menuliskan *command* “/belajar” seperti pada gambar III.15. Materi belajar akan tampil setelah materi pembelajaran dipilih.



Gambar III.15 Tampilan fungsi belajar

4. Tampilan fungsi kelola data

Fungsi kelola data yang terdiri dari: tambah data, ubah data, dan hapus data hanya dapat diakses oleh “admin”. Subfungsi tersebut akan muncul setelah “admin” menekan tombol “/admin” atau menulis *command* “/admin”. Kemunculannya berupa tombol menu “Tambah”, “Ubah”, dan “Hapus” sebagaimana gambar III.16.



Gambar III.16 Tampilan fungsi kelola data

Terdapat tiga kondisi respon sistem terhadap masing-masing subfungsi. Setiap kondisi memiliki respon yang berbeda. Namun terdapat satu kesamaan pada ketiganya yakni pada setiap pemilihan subfungsi sistem akan menampilkan daftar tabel yang tersedia pada basis data dalam bentuk sekumpulan tombol. Adapun perbedaan dari ketiga kondisi tersebut adalah sebagai berikut:

a. Subfungsi tambah data

Setelah admin memilih salah satu tabel, maka akan muncul pesan yang mempersilakan admin untuk menambahkan data beserta format yang harus diikuti admin ketika menambah data. Hanya dengan format tersebutlah sistem dapat mendeteksi data yang hendak ditambahkan. Lalu admin akan menampilkan daftar data terkini yang ada pada tabel tersebut. Gambar III.17 memperlihatkan respon sistem pada subfungsi tambah data.

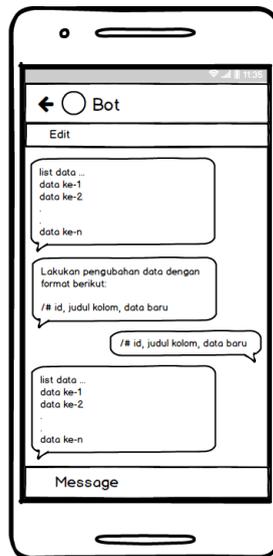


Gambar III.17 Tampilan respon sistem pada subfungsi tambah data

b. Subfungsi edit data

Berbeda dengan subfungsi tambah data, sistem akan menampilkan data tabel yang dipilih yang tersedia pada basis data beserta format pesan untuk melakukan perubahan data. Dalam format tersebut admin dapat memilih *field* mana yang ingin diubah nilainya dengan mencantumkan “id” dari baris yang ingin diubah.

Setelah admin mengirimkan data baru, sistem akan menampilkan kembali data terkini yang terdapat pada tabel tersebut. Pada gambar III.18 diperlihatkan bagaimana respon sistem pada subfungsi edit data.



Gambar III.18 Tampilan respon sistem pada subfungsi edit data

c. Subfungsi hapus data

Respon sistem pada subfungsi ini sama dengan respon yang terjadi pada subfungsi edit data. Perbedaan hanya terletak pada format pesan yang diizinkan sistem untuk melakukan pengelolaan data, dalam hal ini penghapusan data. Perbedaan tersebut dapat dilihat pada gambar III.19.



Gambar III.19 Tampilan respon sistem pada subfungsi hapus data