



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Beberapa metode penelitian yang digunakan dalam penelitian ini antara lain.

a. Telaah Literatur

Dalam tahap ini, kegiatan yang dilakukan adalah pengumpulan informasi yang berhubungan dengan *voice chatbot* JACOB, kecerdasan buatan (*chatbot*, jaringan saraf tiruan), parafrasa kalimat, *Long Short-Term Memory* (Stacked Residual LSTM), *text summarization* (algoritma TextRank dan metode *Cosine Similarity*), dan *Technology Acceptance Model* terhadap sistem atau modul. Informasi yang dicari berasal dari buku, *website*, jurnal, dan *conference proceeding*. Dengan dilakukannya telaah literatur ini, ditentukan model jaringan saraf tiruan LSTM untuk menghasilkan parafrasa jawaban mengenai program *dual degree* Informatika sehingga jawaban yang diberikan oleh JACOB dapat bervariasi. Dengan dilakukan pembelajaran terhadap kalimat tersebut, CLEVEREE dapat memberikan respon sesuai kalimat yang telah dipelajari dan dapat memberikan kalimat yang bervariasi.

b. Analisis dan Perancangan

Dalam tahap analisis, dilakukan analisis terhadap metode, data, proses, dan spesifikasi yang digunakan pada modul CLEVEREE. Proses analisis kebutuhan modul CLEVEREE diamati dari aplikasi *voice chatbot* JACOB dan pada sistem admin JACOB. Dimulai dari bahasa yang digunakan, fitur yang dimiliki, dan lain-lain. Untuk hasil analisisnya dibahas pada Subbab 3.2.

Setelah mengumpulkan literatur dan analisis kebutuhan, perancangan modul CLEVEREE dibuat gambaran umum modul, *flowchart*, dan diagram *web service model* untuk mendeskripsikan proses dari modul yang dikembangkan. Selain itu, dirancang juga penambahan fitur baru pada sistem admin *voice chatbot* JACOB dan dilakukan perancangan dengan membuat struktur tabel dan rancangan antarmuka pada sistem admin. Hasil perancangan diagram, struktur tabel, dan rancangan antarmuka dapat dilihat pada Subbab 3.3.

c. Implementasi

Dalam pengimplementasian, berupa integrasi antara modul CLEVEREE dan aplikasi JACOB, modul CLEVEREE dikembangkan menjadi sebuah *web service* dengan *Flask framework* dengan bahasa pemrograman Python 3.5. Modul CLEVEREE merupakan *web service* yang dapat diakses pada aplikasi *voice chatbot* JACOB yang berbasis web. Dalam pembuatan *web service* menggunakan *framework web service* Python Flask 1.0.2. Tahap implementasi dilakukan dengan menuliskan baris-baris kode berupa pengimplementasian metode *Cosine Similarity*, algoritma TextRank, dan *pretrained model Stacked Residual LSTM*. Metode *Cosine Similarity* digunakan untuk menghitung nilai kemiripan antar dua kalimat. Kemudian dilanjutkan dengan implementasi algoritma TextRank untuk mendapatkan peringkat kalimat. Selain itu, dalam penambahan halaman ringkasan dan hasil parafrasa jawaban pada sistem admin JACOB menggunakan bahasa pemrograman PHP dengan *framework* Laravel.

Semua *library* atau *package* yang digunakan dalam pengembangan modul CLEVEREE adalah Tensorflow, networkx, NLTK, pandas, dan numpy. Tensorflow digunakan dalam proses pembelajaran menggunakan model jaringan *Stacked Residual LSTM* yang tersedia pada GitHub seperti yang telah dituliskan pada Subbab 2.5.1. *Library* networkx digunakan untuk mengimplementasikan algoritma TextRank. *Library* NLTK, pandas, dan numpy digunakan dalam proses *preprocessing* data.

d. Pengujian

Uji coba dilakukan untuk mengetahui jika modul masih memiliki kesalahan atau tidak. Dalam penelitian ini menggunakan uji coba dengan pendekatan *whitebox testing* untuk melakukan uji coba pada implementasi metode *Cosine Similarity*. Selain itu, uji coba dilakukan dengan memeriksa fungsionalitas modul telah berjalan dengan benar sesuai dengan perancangan.

e. Evaluasi

Evaluasi dilakukan untuk menentukan apakah modul CLEVEREE yang telah diimplementasikan ke dalam *voice chatbot* telah memenuhi tujuan pada penelitian ini. Selain itu, evaluasi yang mengukur kemudahan penggunaan dan manfaat yang dirasakan terhadap modul CLEVEREE menggunakan *Technology Acceptance Model (TAM)*. Dalam penelitian ini tidak melibatkan *external variables* dikarenakan jumlah sampel yang diambil menggunakan teknik *purposive sampling* kepada tiga orang admin JACOB. Dalam melakukan evaluasi, setiap persepsi dalam TAM dibuat beberapa pertanyaan dalam bentuk kuesioner. Kuesioner dibuat dengan

menggunakan skala Likert yang terdiri dari enam poin (Sangat Setuju sampai Sangat Tidak Setuju).

f. **Konsultasi dan Penulisan**

Pada tahapan ini, dilakukan penulisan laporan dengan tujuan mendokumentasikan proses penelitian yang telah dilakukan dan menyimpulkan hasil akhir yang didapat. Selain itu, dengan adanya penulisan laporan dapat dijadikan sebagai bukti bahwa penelitian telah dilakukan dan diselesaikan.

3.2 Analisis Kebutuhan Modul CLEVEREE

Berdasarkan analisis terhadap aplikasi *voice chatbot* JACOB diketahui bahwa:

- a. bahasa yang digunakan adalah bahasa Inggris,
- b. aplikasi JACOB menggunakan platform Wit.ai untuk mendapatkan nilai *intent* dan *entities* dari sebuah kalimat,
- c. basis pengetahuan JACOB disimpan dalam *database* MySQL,
- d. belum terdapat pengimplementasian kecerdasan buatan di dalam aplikasi JACOB selain penggunaan platform Wit.ai,
- e. aplikasi JACOB hanya dapat mengembalikan respon sesuai dengan yang ada di dalam *database* atau basis pengetahuan dan untuk masing-masing pertanyaan hanya memiliki satu jawaban,
- f. dalam sistem admin JACOB belum terdapat ringkasan untuk pertanyaan yang banyak ditanyakan pengguna sehingga untuk memasukkan kemungkinan pertanyaan masih dilakukan secara manual oleh admin,

- g. rancangan antarmuka dibuat untuk menampilkan hasil dari ringkasan pertanyaan dan parafrasa jawaban sehingga admin dapat melakukan validasi agar admin tidak perlu membuat kalimat sendiri, atau menghapus ringkasan pertanyaan atau parafrasa jawaban ketika tidak diperlukan sebagai pengetahuan aplikasi JACOB,
- h. *pretrained model* LSTM yang digunakan menggunakan *package* dari Tensorflow dengan bahasa pemrograman Python, sehingga modul CLEVEREE menggunakan bahasa pemrograman Python,
- i. bahasa pemrograman yang digunakan dalam aplikasi JACOB adalah PHP, HTML, dan JavaScript dengan *framework* Laravel sehingga modul CLEVEREE dirancang dan dikembangkan sebagai *web service* menggunakan bahasa pemrograman Python dengan *framework* Flask,
- j. penggunaan *POST request method* dan *GET request method* dipertimbangkan dengan ada atau tidaknya data yang dikirimkan ke *web service* pada modul CLEVEREE, dan
- k. tabel pada *database* aplikasi JACOB belum terdapat tabel untuk menyimpan hasil ringkasan pertanyaan dan parafrasa jawaban sehingga ditambahkan tabel untuk menyimpan data ringkasan pertanyaan berdasarkan satu sesi pengguna dan berdasarkan satu bulan, tabel untuk menyimpan hasil parafrasa jawaban, dan pengubahan struktur tabel logs yang digunakan sebagai tanda bahwa data *logs* telah diringkas atau belum.

3.3 Perancangan Sistem

Dari analisis yang telah dilakukan, modul CLEVEREE dirancang untuk penambahan kecerdasan buatan. Kecerdasan buatan tersebut dikembangkan

sebagai dua fitur baru dengan tahap perancangan yang dibuat adalah gambaran umum modul (lihat Subbab 3.3.1), *flowchart* (lihat Subbab 3.3.2), dan diagram perancangan *web service* (lihat Subbab 3.3.3). Selain itu, perancangan untuk pembaharuan sistem admin JACOB juga dilakukan dalam penelitian ini. Perancangan dilakukan dengan membuat struktur tabel untuk tabel yang diperbaharui atau ditambah (lihat Subbab 3.3.4) dan rancangan antarmuka pada sistem admin JACOB (lihat Subbab 3.3.5). Hasil dari perancangan dua fitur ini ditujukan untuk admin JACOB dengan penjelasan sebagai berikut.

a. Parafrasa jawaban

Dalam menghasilkan jawaban yang bervariasi digunakan model jaringan saraf tiruan LSTM untuk melakukan *paraphrase generation*. Dalam tugas ini, modul CLEVEREE mengimplementasikan *pretrained model Stacked Residual LSTM* untuk menghasilkan parafrasa. Hal yang perlu dilakukan dalam pembelajaran model ini adalah dengan menambahkan data kalimat jawaban yang diambil dari basis pengetahuan JACOB dan parafrasanya.

Kalimat ditokenisasi menjadi kumpulan kata dan kata diubah menjadi representasi vektor dengan dimensi sebesar 256 sehingga banyak unit LSTM yang digunakan pada model sebanyak 256. Pengubahan kata menjadi nilai vektor dilakukan dengan menggunakan *embedding layer* yang tersedia pada *package* Tensorflow. Jika kata tidak terdapat dalam kamus pelatihan

maka kata diganti menjadi *special word* yang berbeda dengan kata yang umum terdapat dalam kamus bahasa Inggris. Lalu *pretrained model* mempelajari data tersebut. Variabel yang digunakan dalam pelatihan adalah *learning rate* sebesar 0.001, *batch size* sebesar 32, *optimizer* yang

digunakan adalah Adam, jumlah *epoch* sebesar 1.500, dan menambahkan residu pada tiap dua *layer* LSTM. Pemilihan jumlah *epoch* sebesar 1.500 dilakukan dengan pertimbangan agar *error rate* yang dihasilkan tidak terlalu kecil sehingga model tidak menghasilkan kalimat yang sama persis dengan data latih. Selain itu, pertimbangan jumlah *epoch* juga dipengaruhi dari jumlah data latih sebanyak 416 kalimat tentang program dual degree Informatika dengan tambahan 102 kalimat di luar program dual degree Informatika dengan tujuan memperbanyak kosa kata atau kata unik data latih. Jumlah kata unik yang digunakan dalam pelatihan sebanyak 901 kata, Namun, jumlah kalimat dan kata unik dapat bertambah seiring bertambahnya pengetahuan JACOB. Selain itu, bobot yang digunakan untuk setiap melakukan proses pelatihan selalu diinisialisasi ulang dan tidak menggunakan bobot terakhir pada proses pelatihan sebelumnya. Setelah dipelajari, model dapat menghasilkan variasi jawaban. Sebelum variasi jawaban dimasukkan ke dalam basis pengetahuan JACOB, hasil *generate* yang dilakukan oleh model harus diperiksa oleh admin JACOB. Admin JACOB dapat memperbaiki, menerima, dan menolak variasi jawaban tersebut. Jika diterima, hasil variasi jawaban ditambahkan ke dalam basis pengetahuan.

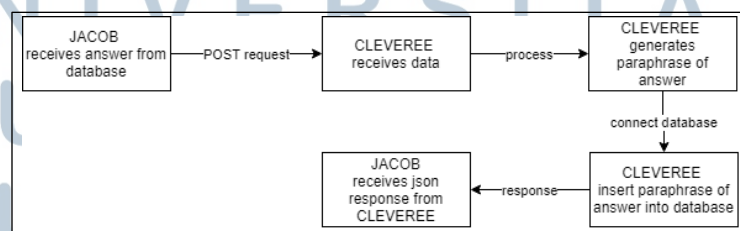
b. Ringkasan pertanyaan

Gambaran umum untuk fitur ini adalah data pertanyaan yang disimpan dalam pada berkas *logs conversation* yang dipisahkan berdasarkan satu sesi pengguna dilakukan penyaringan kalimat. Kalimat yang digunakan adalah kalimat seputar program *dual degree* Informatika yang berupa kalimat

pertanyaan atau permintaan. Kemudian dilakukan proses tokenisasi menjadi kumpulan kata. Selanjutnya, hapus karakter atau kata selain alfabet, ubah semua huruf menjadi huruf kecil, dan hapus semua kata *stopwords*. Selanjutnya, kata diubah menjadi nilai representasi vektor berdasarkan *pretrained word vector*, dan setiap vektor kata dalam satu kalimat dihitung rata-ratanya. *Pretrained word vector* yang digunakan adalah Word2Vec dengan dimensi sebesar 300. Setelah itu, metode *Cosine Similarity* digunakan untuk mencari nilai kemiripan antara dua kalimat. Kemudian dilakukan perhitungan untuk mendapatkan kalimat yang paling penting dan nilainya menggunakan algoritma TextRank. Berdasarkan telaah literatur yang telah dilakukan pada Subbab 2.5, jumlah kalimat yang digunakan dalam ringkasan daftar pertanyaan yang telah diurutkan diambil sebanyak 25% dari keseluruhan kalimat pertanyaan. Admin dapat melihat hasil peringkasan pada sistem admin JACOB dan dapat menambahkan atau membuang pertanyaan dari hasil ringkasan.

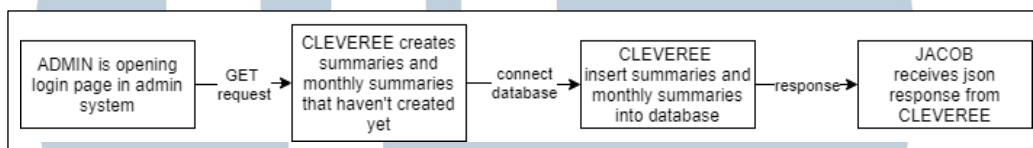
3.3.1 Gambaran Umum Modul CLEVEREE

Gambaran umum modul CLEVEREE digunakan untuk menjelaskan cara kerja modul yang terdiri dari dua fitur yaitu untuk parafrasa jawaban dan membuat ringkasan pertanyaan. Gambaran umum untuk fitur parafrasa jawaban dapat dilihat pada Gambar 3.1.



Gambar 3.1 Gambaran umum modul CLEVEREE untuk fitur parafrasa jawaban

Berdasarkan Gambar 3.1, gambaran kerja modul CLEVEREE untuk fitur parafrasa jawaban dimulai dengan JACOB mengirimkan POST *request* kepada modul CLEVEREE yang berupa *web service*. Setelah menerima data dari JACOB, CLEVEREE menghasilkan parafrasa kalimat jawab. Kemudian, CLEVEREE memasukan data hasil parafrasa ke dalam *database* dan memberitahukan kepada JACOB bahwa proses telah selesai.



Gambar 3.2 Gambaran umum modul CLEVEREE untuk fitur ringkasan pertanyaan

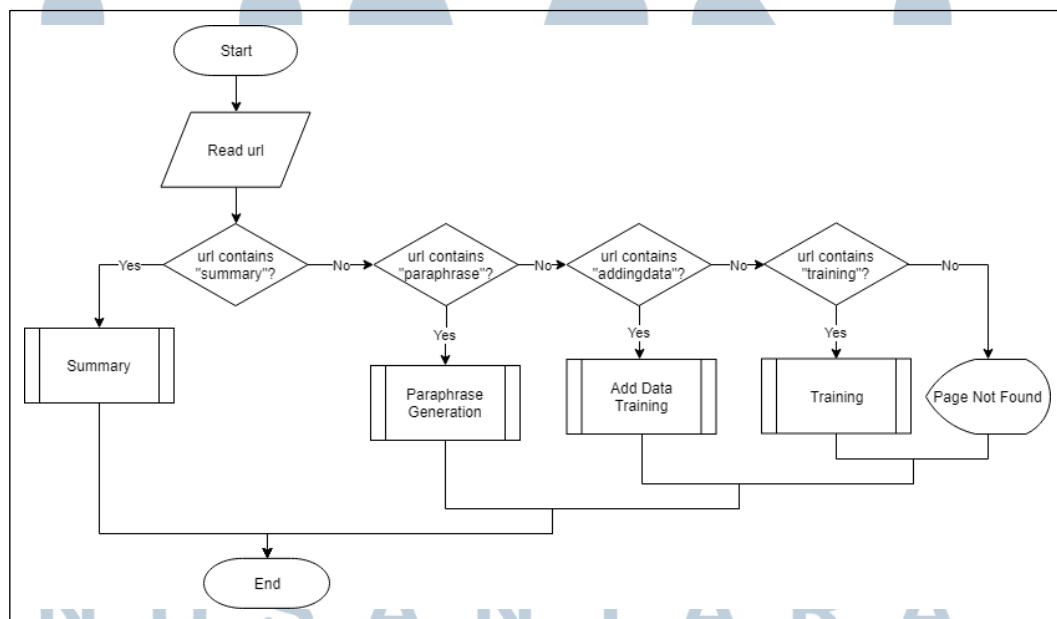
Berdasarkan Gambar 3.2, gambaran kerja modul CLEVEREE untuk fitur ringkasan pertanyaan dimulai dengan admin membuka halaman *login* pada sistem admin. Pada saat akan menampilkan halaman *login* dilakukan GET *request* ke CLEVEREE. Kemudian CLEVEREE membuat ringkasan untuk *logs* yang belum terdapat ringkasannya. Data peringkasan ditambahkan ke *database*. Lalu, CLEVEREE memberitahukan kepada JACOB bahwa proses telah selesai.

3.3.2 Flowchart Modul CLEVEREE

Flowchart digunakan untuk menggambarkan alur dari sistem yang dibuat. Alur modul terdiri dari *flowchart* utama, proses pengambilan data, proses *preprocessing* data, pembuatan *similarity matrix*, metode *Cosine Similarity*, peringkasan pertanyaan, *generate* parafrasa jawaban, penambahan data latih, dan proses pelatihan dijabarkan sebagai berikut.

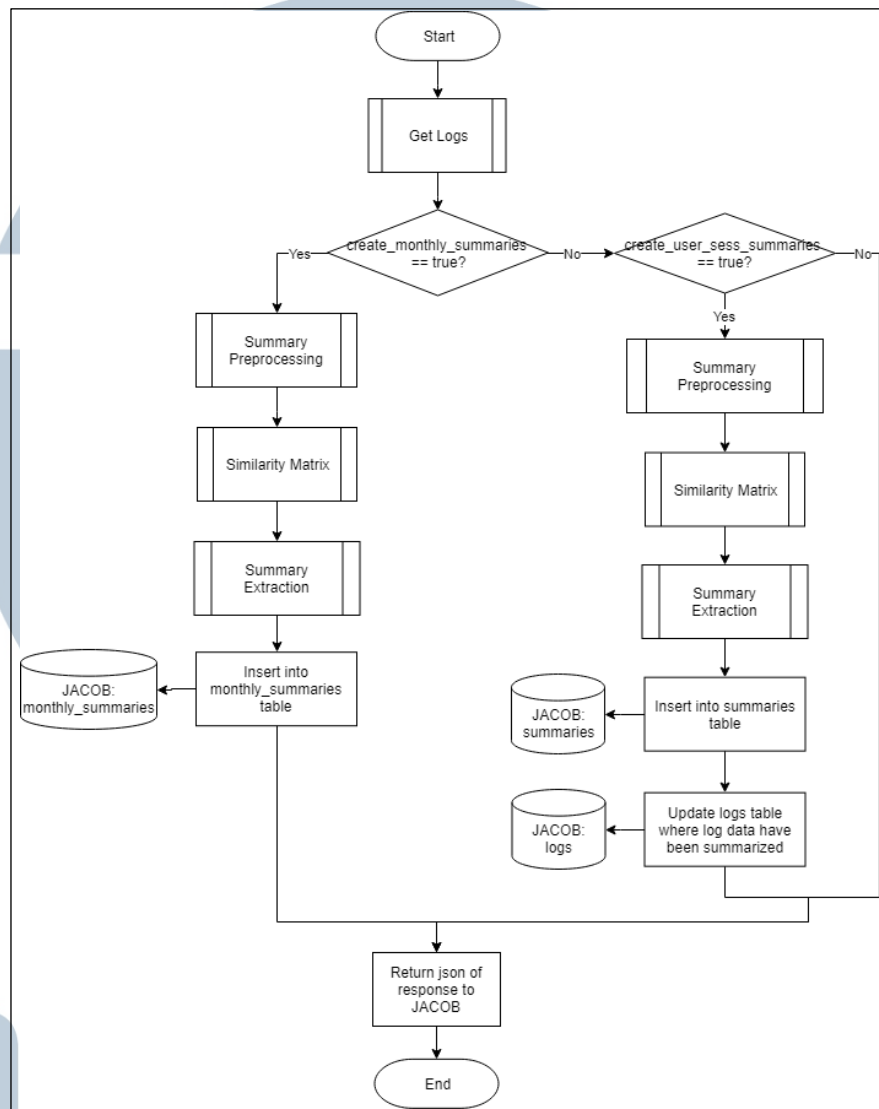
A. Flowchart Utama

Flowchart Utama menggambarkan keseluruhan alur modul secara umum pada modul CLEVEREE yang ditunjukkan pada Gambar 3.3. Berdasarkan Gambar 3.3 dapat dilihat bahwa modul CLEVEREE memiliki dua fitur utama, yaitu menghasilkan parafrasa jawaban dan ringkasan pertanyaan yang ditanyakan oleh pengguna. Pada fitur parafrasa jawaban dapat juga dilakukan penambahan data latih dan pelatihan kembali. Ketika JACOB melakukan *request* ke CLEVEREE, CLEVEREE melakukan pengecekan terhadap url. Jika url mengandung kata “paraphrase” maka dilakukan *generate* parafrasa yang ditunjukkan dalam Gambar 3.10. Jika url mengandung kata “summary” maka dilakukan peringkasan pertanyaan yang dapat dilihat pada Gambar 3.4. Jika url mengandung kata “addingdata” maka dilakukan penambahan kalimat ke data latih dengan *flowchart* dapat dilihat pada Gambar 3.11. Jika url mengandung kata “training” maka modul CLEVEREE melakukan pelatihan kembali dan *flowchart* untuk proses *training* ditunjukkan pada Gambar 3.12.



Gambar 3.3 *Flowchart* Utama

B. Flowchart Summary

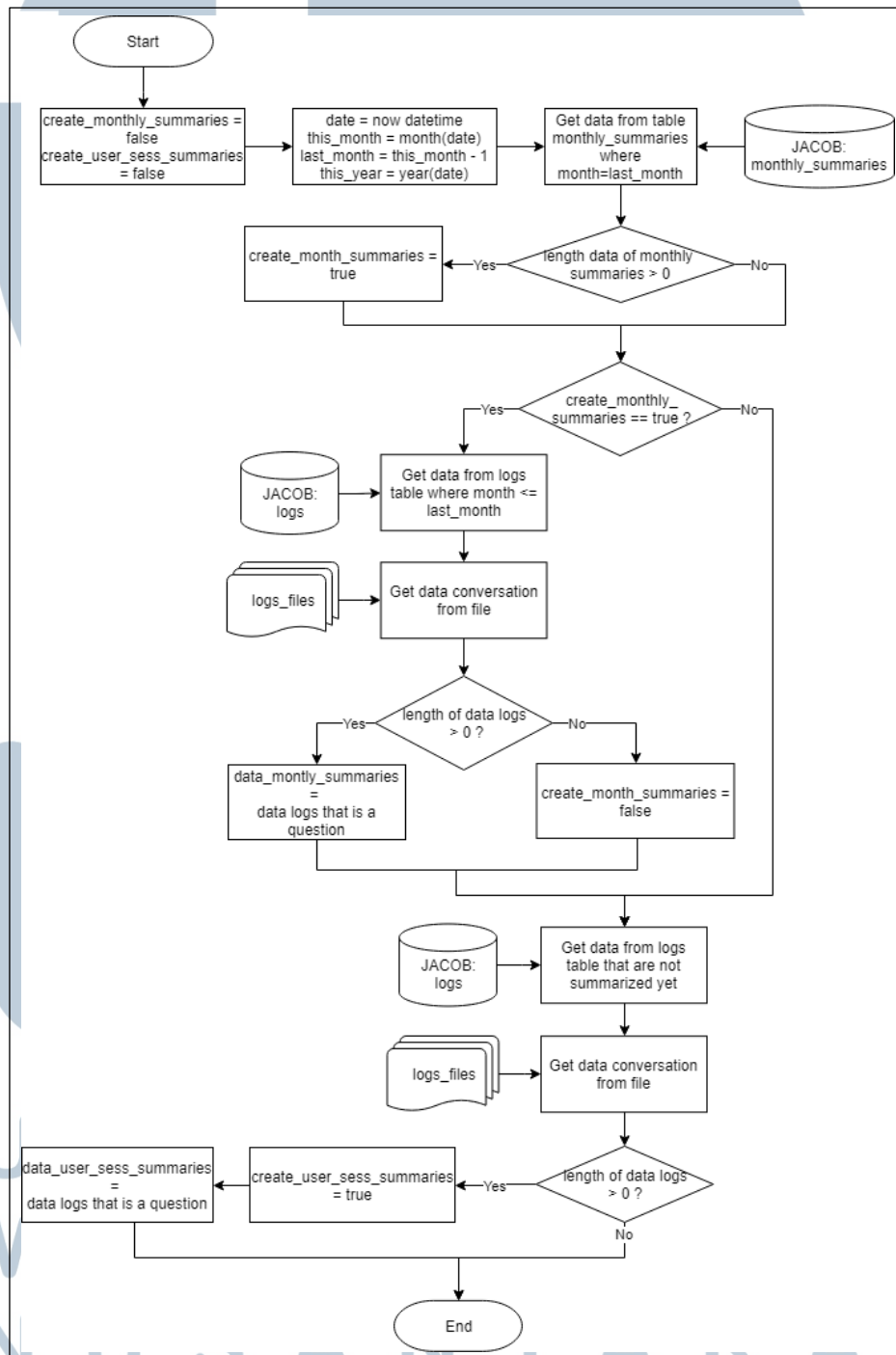


Gambar 3.4 *Flowchart Summary*

Gambar 3.4 merupakan *Flowchart* modul Summary dibuat untuk menggambarkan alur modul dalam fitur ringkasan. Berdasarkan *flowchart* ini, mulanya data pertanyaan diambil dan dijelaskan dalam *flowchart* pada Gambar 3.5. Setelah data didapat, dilakukan pengolahan kalimat pada modul Summary Preprocessing yang dijelaskan dalam Gambar 3.6. Setelah itu, dibuat *similarity matrix* untuk setiap dua kalimat pada kolom matriks dan alur perhitungannya dapat dilihat pada Gambar 3.7. Kemudian, didapat hasil peringkat kalimat yang

dilakukan pada modul Summary Extraction yang dijelaskan alurnya dalam Gambar 3.9. Terakhir, data hasil peringkasan ditambahkan ke dalam *database* dan modul CLEVEREE mengembalikan *response* ke *voice chatbot* JACOB.

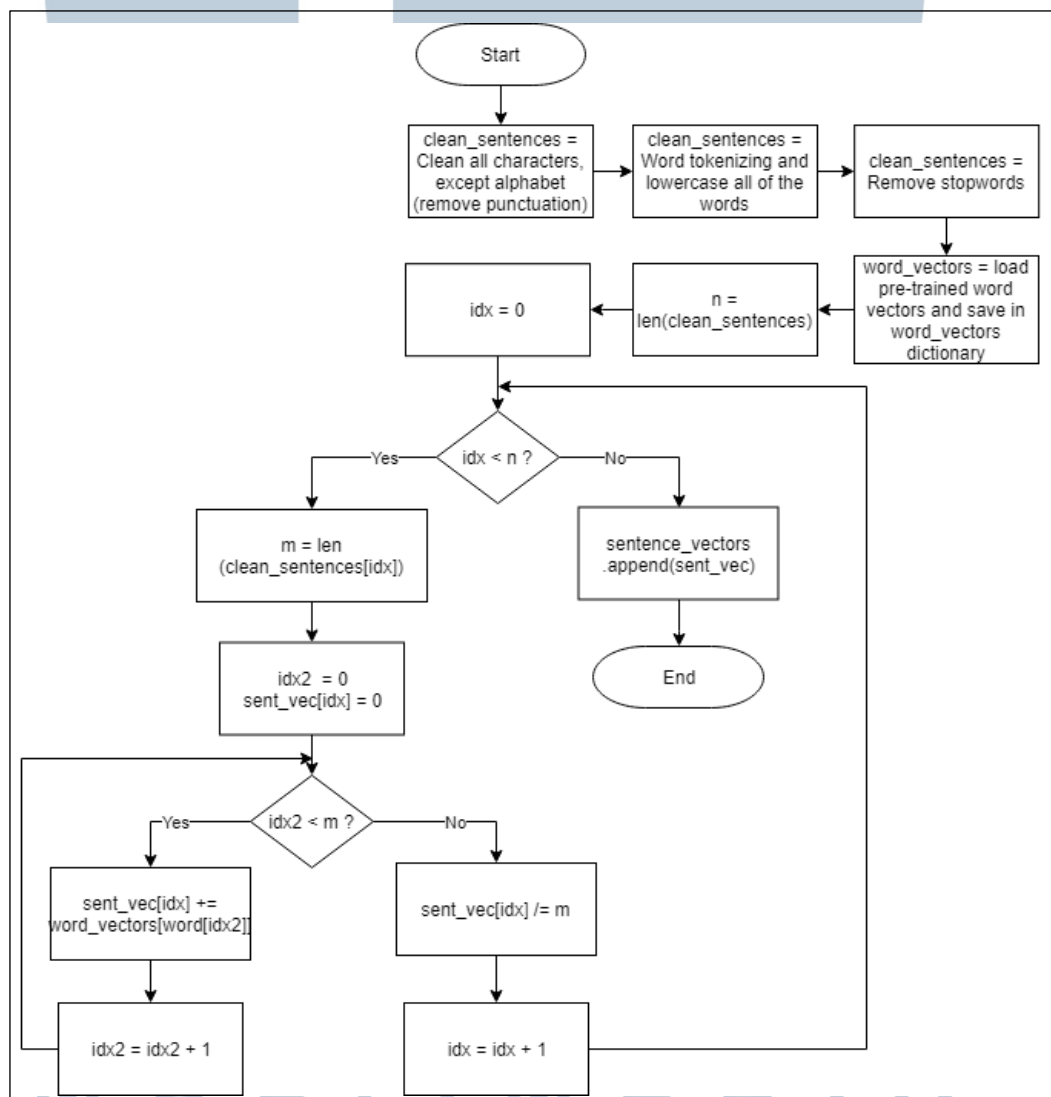
C. Flowchart Get Logs



Gambar 3.5 Flowchart Get Logs

Flowchart pada Gambar 3.5 menggambarkan proses pengambilan data *logs* percakapan antara JACOB dan pengguna. Pertama, dilakukan pengecekan data ringkasan pada *database*, untuk menentukan perlu dilakukan proses peringkasan pertanyaan atau tidak. Jika diperlukan maka lakukan pengambilan data pada tabel *logs* dan baca setiap dokumen percakapan setiap satu sesi pengguna. Nama dokumen diketahui berdasarkan nama yang tersimpan di dalam *database*.

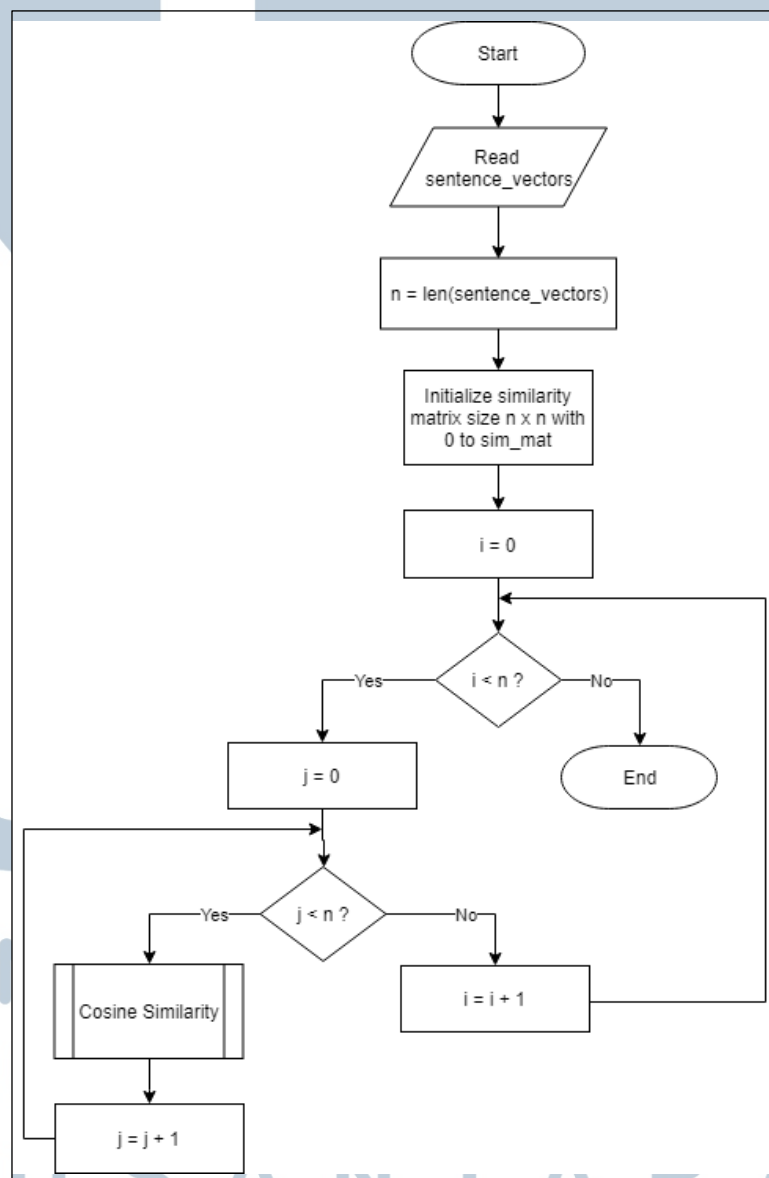
D. Flowchart Summary Preprocessing



Gambar 3.6 Flowchart Summary Preprocessing

Flowchart Summary Preprocessing pada Gambar 3.6 menunjukkan proses pengolahan kalimat, yaitu dengan melakukan penghapusan karakter selain alfabet, tokenisasi kata, membuat semua huruf menjadi kecil, dan menghapus *stopwords*. Kata kemudian diubah menjadi representasi vektor menggunakan *pretrained word vector*. Selanjutnya, untuk mencari nilai vektor pada satu kalimat dilakukan perhitungan rata-rata dari setiap kata yang terdapat pada variabel *clean_sentences*.

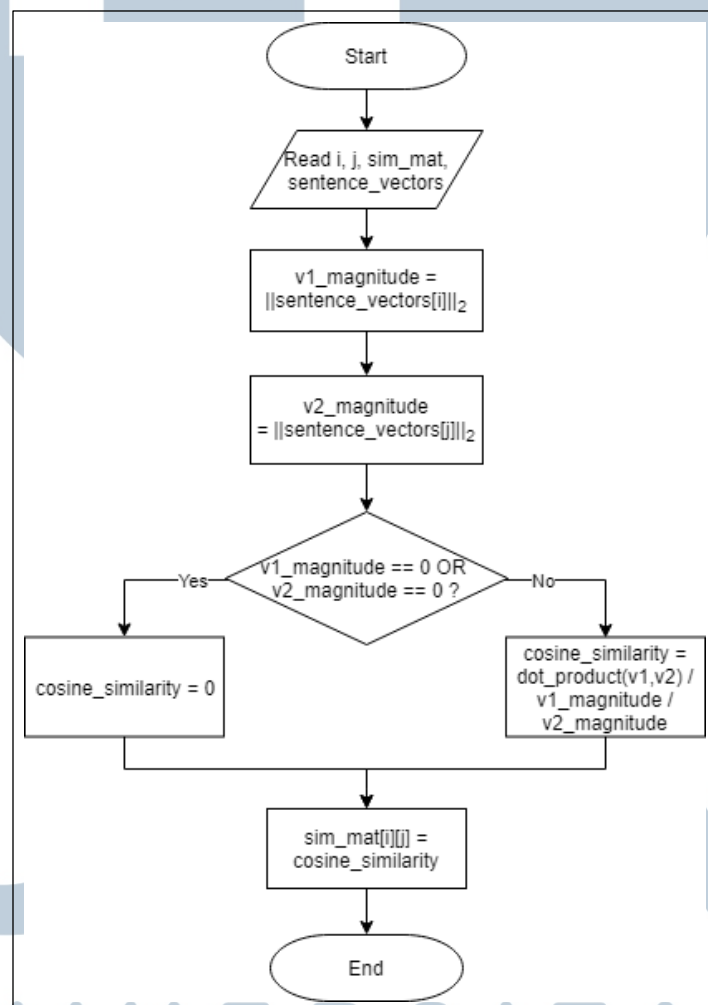
E. Flowchart Similarity Matrix



Gambar 3.7 *Flowchart Similarity Matrix*

Flowchart pada Gambar 3.7 menggambarkan alur pembuatan *similarity matrix* untuk tiap dua kalimat. Jadi, ukuran matriks yang dibutuhkan adalah n kali n dengan n merupakan banyak kalimat. Untuk mendapatkan nilai kemiripan antara dua kalimat, dilakukan perhitungan menggunakan metode *Cosine Similarity*. Alur metode *Cosine Similarity* dapat dilihat pada Gambar 3.8.

F. Flowchart Cosine Similarity

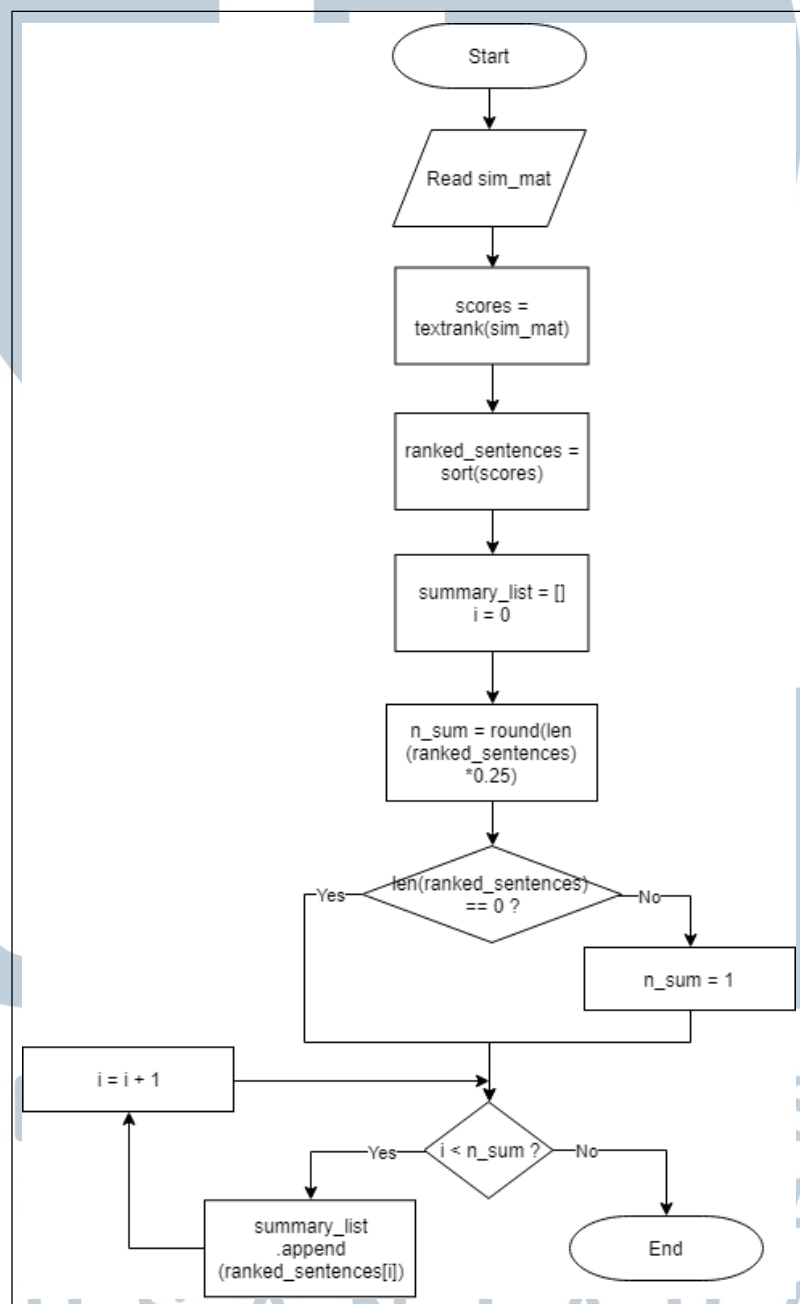


Gambar 3.8 Flowchart Cosine Similarity

Pada Gambar 3.8 menunjukkan *flowchart* perhitungan nilai kemiripan antara dua kalimat menggunakan metode *Cosine Similarity*. Dimulai dengan melakukan perhitungan panjang vektor kalimat pertama dan panjang vektor

kalimat kedua. Kemudian dilakukan pengecekan nilai panjang vektor kedua kalimat itu. Jika salah satu nilai panjang vektor kalimat terdapat nilai nol maka nilai kemiripan antara dua kalimat tersebut adalah nol. Sebaliknya, nilai kemiripan antara dua kalimat dihitung menggunakan metode *Cosine Similarity*.

G. Flowchart summary_extraction

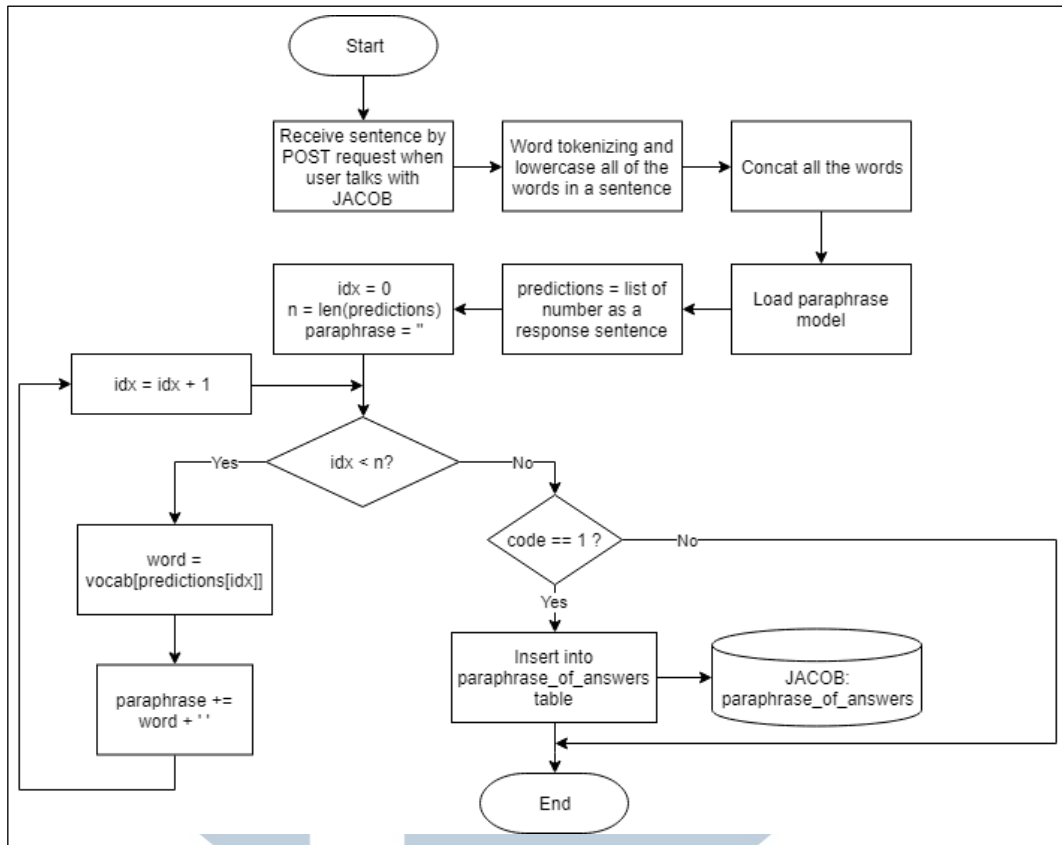


Gambar 3.9 Flowchart Summary Extraction

Pada Gambar 3.9 ditunjukkan *Flowchart* Summary Extraction yang merupakan alur untuk menghasilkan dan memilih kalimat yang dijadikan ringkasan. Alur yang dilakukan dimulai dari proses pemilihan kalimat yang digunakan sebagai ringkasan. Nilai dari masing-masing kalimat dihitung menggunakan algoritma TextRank dengan masukan berupa *similarity matrix* yang telah dibuat sebelumnya. Hasil dari algoritma TextRank berupa nilai probabilitas untuk setiap kalimat dengan nilai tertinggi sebagai kalimat yang paling penting. Lalu dilakukan pengurutan kalimat berdasarkan nilai tertinggi dan diambil 25% kalimat dari keseluruhan pertanyaan untuk dijadikan ringkasan pertanyaan.

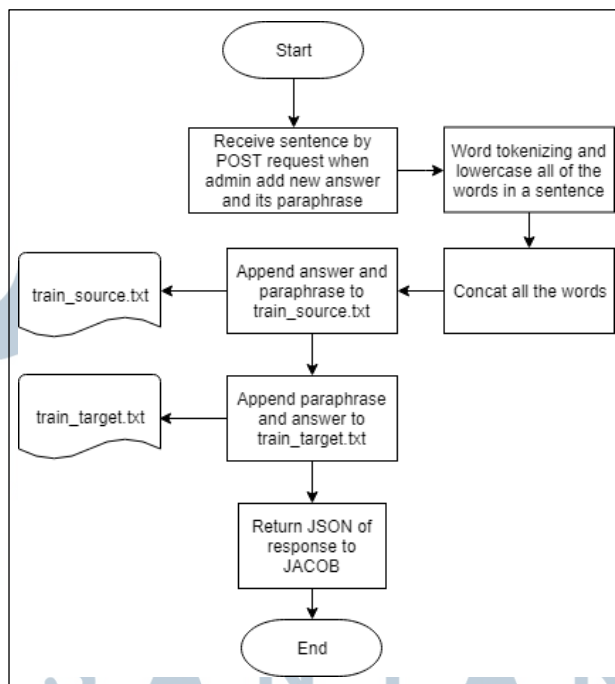
H. Flowchart Paraphrase Generation

Flowchart Paraphrase Generation menggambarkan alur dihasilkannya parafrasa jawaban pada modul CLEVEREE yang ditunjukkan pada Gambar 3.10. Alur dimulai dari modul CLEVEREE menerima kalimat dan kalimat dilakukan tokenisasi kalimat jadi kata. Setelah itu, semua huruf diubah menjadi kecil semua (*lowercase*) dan digabungkan kembali menjadi kalimat. Kemudian, kalimat tersebut dimasukkan ke dalam model sehingga dihasilkan kalimat baru dan berbeda dengan kalimat masukan. Sebelum diubah jadi kalimat, hasil yang dikeluarkan oleh model berupa kumpulan angka yang menyatakan nilai suatu kata. Oleh karena itu, dilakukan pengecekan terhadap *vocabulary* kata dan representasi angkanya untuk mendapatkan kata dari angka tersebut. Setelah proses selesai dilakukan, *response* dikembalikan kepada JACOB.



Gambar 3.10 Flowchart Paraphrase Generation

I. Flowchart Add Data Training



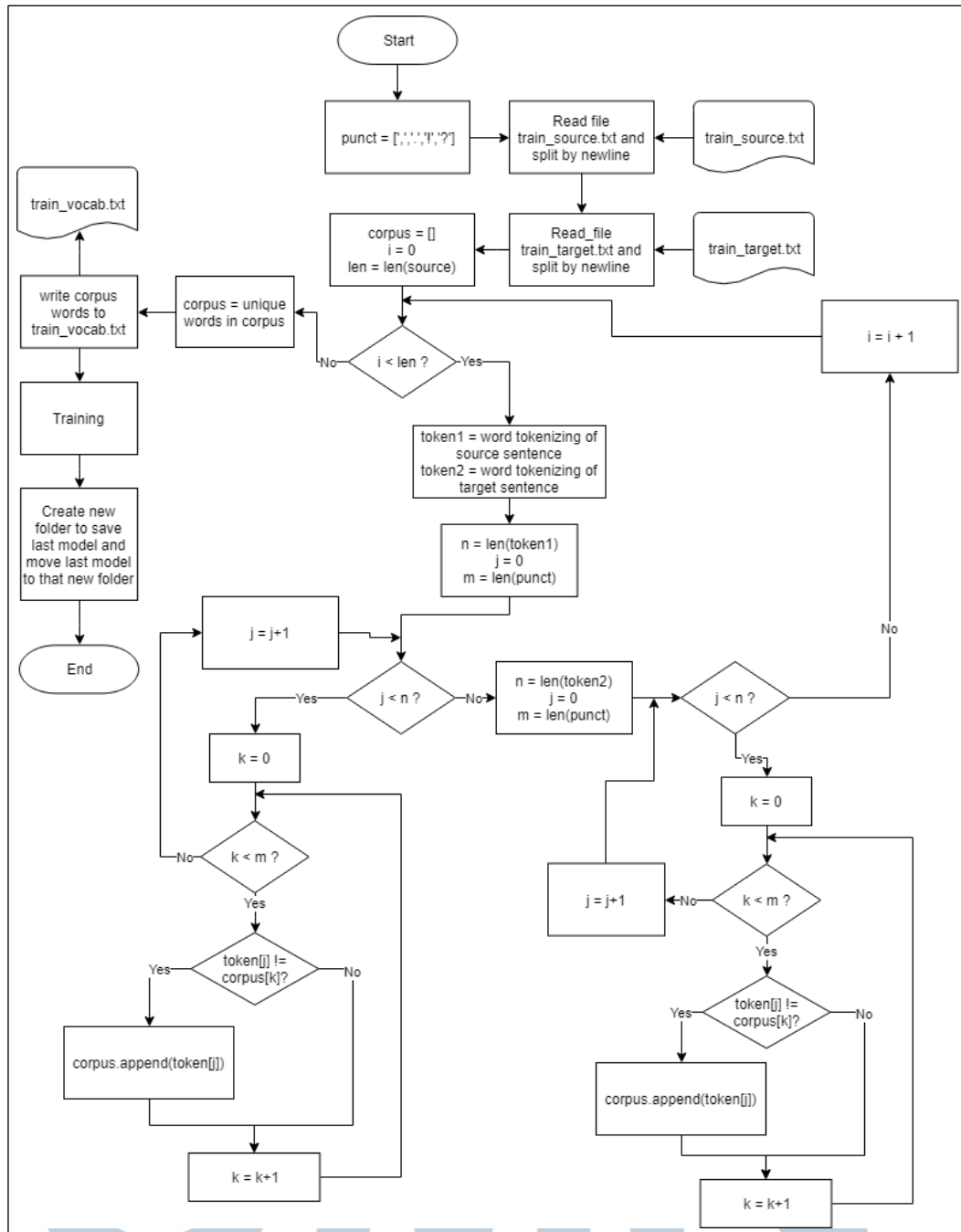
Gambar 3.11 Flowchart Add Data Training

Flowchart pada Gambar 3.11 merupakan alur CLEVEREE untuk melakukan penambahan data pelatihan ke dalam berkas `train_source.txt` dan `train_target.txt`. Alur dimulai dari penerimaan kalimat jawab baru yang ditambahkan oleh admin dan kalimat parafrasanya. Kemudian kalimat dipisah-pisah menjadi kumpulan kata dan dijadikan *lowercase*. Lalu gabungkan kembali kata-kata menjadi kalimat. Selanjutnya, dilakukan penambahan kalimat jawaban dan kalimat parafrasa ke berkas.

J. Flowchart Training

Pada Subbab ini dijelaskan alur dari *Flowchart Training* yang ditunjukkan dalam Gambar 3.12. Inti dari *flowchart* ini adalah menjelaskan alur *preprocessing* kalimat sebelum digunakan sebagai masukan model *Stacked Residual LSTM*. Hal yang dilakukan pertama adalah membaca data latih dan membuang tanda baca seperti koma, titik, dan lain lain. Lalu ambil kata unik sebagai *vocabulary* yang dapat digunakan dan dituliskan ke dalam berkas `train_vocab.txt`. Proses selanjutnya adalah melakukan proses pelatihan dengan menggunakan model *Stacked Residual LSTM* untuk memperbaharui pengetahuan. Setelah melakukan *training*, buat *folder* baru dan pindahkan model lama ke dalam *folder* baru tersebut.

U M N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

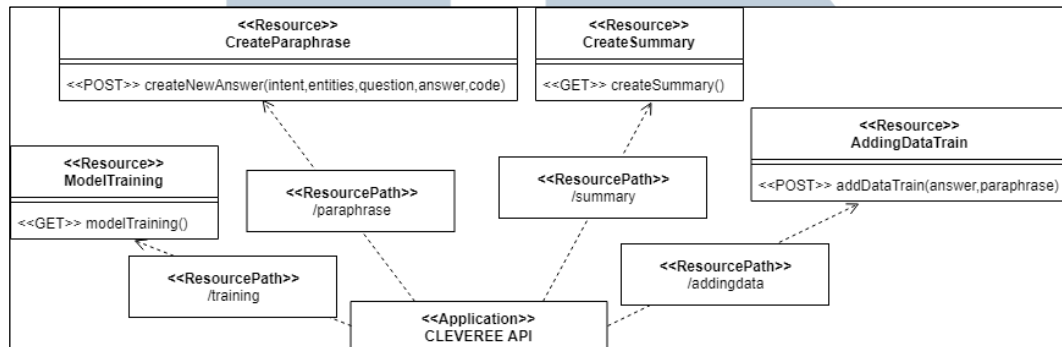


Gambar 3.12 Flowchart Training

3.3.3 Web Service Model CLEVEREE

Diagram *web service model* dirancang untuk modul CLEVEREE dan dapat dilihat pada Gambar 3.13. Modul CLEVEREE memiliki dua *ResourcePath* yang masing-masing menjalankan satu fitur, yaitu fitur ringkasan pertanyaan dan fitur parafrasa jawaban. Fitur ringkasan pertanyaan yang ditanyakan oleh pengguna

dapat diakses melalui *ResourcePath* berupa “/summary”. Dalam melakukan peringkasan dilakukan dengan melakukan GET *request* tanpa mengirimkan data ke modul CLEVEREE.



Gambar 3.13 Web service model API CLEVEREE

ResourcePath untuk fitur parafrasa jawaban adalah “/paraphrase”. Fitur parafrasa dilakukan ketika modul CLEVEREE menerima POST *request* dengan data berupa *intent*, *entities*, *question*, *answer*, dan *code*. Jika *code* sama dengan satu maka *web service* dipanggil saat JACOB berkomunikasi dengan pengguna.

Modul CLEVEREE dirancang juga agar dapat menambahkan data latih dan dapat melakukan *training*. Penambahan data latih dijalankan ketika ada yang melakukan POST *request* ke *ResourcePath* “/addingdata”, dengan data yang dikirimkan adalah kalimat jawaban dan kalimat parafrasanya. Proses *training* dijalankan ketika ada yang melakukan GET *request* tanpa mengirimkan data ke *ResourcePath* “/training” pada modul CLEVEREE. Proses *training* telah dijelaskan sebelumnya pada Perancangan Sistem dengan Subbab 3.3.

3.3.4 Struktur Tabel Pembaharuan Sistem Admin JACOB

Pada penelitian sebelumnya telah menggunakan *database* untuk menyimpan pertanyaan, *intent*, *entities*, *logs*, admin, dll. Namun, di dalam penelitian ini dilakukan penyesuaian dengan *database* JACOB sebelumnya. Penyesuaian

tersebut dilakukan dengan perubahan struktur tabel dan penambahan tabel. Perubahan struktur tabel dilakukan pada satu tabel, yaitu tabel logs. Dalam penelitian ini terdapat penambahan tiga buah tabel sebagai berikut.

- a. Tabel summaries, untuk menyimpan data ringkasan pertanyaan berdasarkan satu sesi pengguna.
- b. Tabel monthly_summaries, untuk menyimpan data ringkasan pertanyaan selama satu bulan.
- c. Tabel paraphrase_of_answers, untuk menyimpan data parafrasa jawaban.

Kegunaan tabel logs dalam penelitian ini masih sama dengan penelitian sebelumnya, yaitu untuk menyimpan data percakapan antara JACOB dan pengguna. Dalam penelitian ini, tabel logs dilakukan perubahan dengan menambah dua kolom, yaitu kolom is_summarized dan kolom user. Struktur tabel logs ditunjukkan pada Tabel 3.1.

Tabel 3.1 Struktur tabel logs

Field	Type	Null	Key	Default	Description
id	int (10)	NO	PK		<i>auto increment</i> dan <i>unsigned integer</i>
name	varchar(191)	NO			nama <i>logs file</i>
user	varchar(30)	NO		NULL	
is_summarized	boolean	NO		false	menyatakan sudah atau belum <i>logs</i> tersebut dibuat ringkasannya
created_at	timestamp	NO		<i>current timestamp</i>	
updated_at	timestamp	NO		<i>current timestamp</i>	

Berdasarkan Tabel 3.1, kolom is_summarized dirancang dengan tipe data boolean untuk menyatakan sudah atau belum data logs tersebut dibuat ringkasannya. Selain itu, kolom user dirancang dengan tipe data varchar dengan

maksimum panjang karakter sebanyak 30. Kolom user pada tabel logs ini digunakan untuk menyimpan nama pengguna pada satu sesi.

Selain pembaharuan yang dilakukan pada tabel logs, ditambahkan juga tabel summaries yang struktur tabel summaries ditunjukkan pada Tabel 3.2. Tabel summaries digunakan untuk menyimpan data ringkasan pertanyaan berdasarkan satu sesi pengguna. Kolom logs_id digunakan untuk menyimpan id dari sesi pengguna berdasarkan ringkasan pertanyaan yang telah terpilih. Setiap pertanyaan terdapat nilai *intent*, *entity*, dan *entity value* yang menyatakan konteks dari sebuah kalimat sehingga nilai tersebut disimpan di dalam tabel ini pada kolom intent, entity, dan entity_value. Kolom summary digunakan untuk menyimpan pertanyaan yang telah dipilih sebagai ringkasan pertanyaan penting yang ditanyakan selama satu sesi pengguna.

Tabel 3.2 Struktur tabel summaries

Field	Type	Null	Key	Default	Description
id	int (11)	NO	PK		<i>auto increment dan unsigned integer</i>
logs_id	int(10)	NO	FK		
date	datetime	NO			tanggal logs dibuat
intent	varchar(50)	YES			
entity	varchar(100)	YES			
entity_value	varchar(100)	YES			
summary	text	NO			kalimat pertanyaan
created_at	timestamp	NO		<i>current timestamp</i>	
updated_at	timestamp	NO		<i>current timestamp</i>	

Peringkasan pertanyaan juga dilakukan tiap bulan dan diambil semua pertanyaan yang ditanyakan selama satu bulan tersebut. Data ringkasan tersebut disimpan pada tabel monthly_summaries dan struktur tabelnya dapat dilihat pada

Tabel 3.3. Struktur tabel *monthly_summaries* terdiri dari kolom *id*, *logs_id*, *month*, *summary*, *created_at*, dan *updated_at*. Kolom *logs_id* digunakan untuk menyimpan id dari sesi pengguna berdasarkan ringkasan pertanyaan yang telah terpilih. Kolom *month* pada struktur tabel menyatakan tanggal dari pertanyaan tersebut ditanyakan. Sama seperti struktur tabel *summaries*, kolom *summary* digunakan untuk menyimpan kalimat pertanyaan yang dianggap penting selama satu bulan terakhir.

Tabel 3.3 Struktur tabel *monthly_summaries*

Field	Type	Null	Key	Default	Description
<i>id</i>	int (11)	NO	PK		<i>auto increment</i> dan <i>unsigned integer</i>
<i>logs_id</i>	int(10)	NO	FK		
<i>month</i>	datetime	NO			
<i>summary</i>	text	NO			
<i>created_at</i>	timestamp	NO		<i>current timestamp</i>	
<i>updated_at</i>	timestamp	NO		<i>current timestamp</i>	

Tabel *paraphrase_of_answers* digunakan untuk menyimpan hasil *generate* yang telah dilakukan oleh *pretrained model stack residual LSTM*. Berdasarkan hasil *generate* tersebut didapatkan parafrasa jawaban terhadap kalimat jawab yang diterima oleh modul CLEVEREE. Struktur tabel *paraphrase_of_answers* ditunjukkan pada Tabel 3.4 dan terdiri dari kolom *id*, *re_id*, *question*, *answer*, *created_at*, dan *updated_at*. Kolom *re_id* merupakan *foreign key* dari tabel *required_entities*. Kolom *answer* merupakan kolom untuk menyimpan hasil parafrasa jawaban.

Tabel 3.4 Struktur tabel *paraphrase_of_answers*

Field	Type	Null	Key	Default	Description
id	int (11)	NO	PK		<i>auto increment dan unsigned integer</i>
re_id	int(10)	NO	FK		id dari tabel <i>required_entities</i>
question	varchar(100)	NO			
answer	text				
created_at	timestamp	NO		<i>current timestamp</i>	
updated_at	timestamp	NO		<i>current timestamp</i>	

3.3.5 Rancangan Antarmuka Pembaharuan Sistem Admin JACOB

Berdasarkan analisis yang telah dilakukan pada Subbab 3.2, rancangan antarmuka untuk sistem admin JACOB dibuat untuk dapat memperlihatkan hasil dan dapat menambahkan pengetahuan kepada JACOB berdasarkan hasil yang dikerjakan oleh modul CLEVEREE. Perubahan pada sistem admin JACOB menambahkan tiga *tab*, yaitu *tab summary of questions* yang dapat dilihat pada Subbab A, *paraphrase of answers* yang dapat dilihat pada Subbab B, dan *learning* yang dapat dilihat pada Subbab C.

A. Rancangan Antarmuka Summary of Questions

Pada Subbab ini dijelaskan rancangan antarmuka untuk menampilkan hasil ringkasan berdasarkan satu sesi pengguna dan setiap satu bulan yang telah dilakukan oleh modul CLEVEREE. Rancangan antarmuka untuk tampilan halaman ringkasan berdasarkan satu sesi pengguna ditunjukkan pada Gambar 3.14 dan halaman ringkasan berdasarkan setiap satu bulan ditunjukkan pada Gambar 3.16.

JACOB Admin Register name

List Answer Samples Entites Logs **Summaries** Paraphrase Learning

Last Month Summaries Show History

- Summary 1
- Summary 2
- Summary 3

Search

No	User Session	Date	Summary	Intent	Entites Count	Action
1	Kevin	1 March 2019	Summary1	Intent1	1	<input type="checkbox"/>
2	Toni	1 March 2019	Summary2	Intent2	0	<input type="checkbox"/>

Previous 1 Next

Gambar 3.14 Rancangan antarmuka *summaries per user session*

Berdasarkan Gambar 3.14, rancangan antarmuka terdiri dari dua bagian, yaitu *card* yang berisi ringkasan pertanyaan untuk satu bulan terakhir dan tabel yang berisi ringkasan pertanyaan berdasarkan satu sesi pengguna. Untuk melihat keseluruhan ringkasan untuk bulan sebelumnya dapat dilihat dengan menekan “Show History” pada bagian atas kanan *card* dan menampilkan tampilan seperti pada Gambar 3.16. Selain itu, pada tabel ringkasan terdapat kolom *action* yang berisi *button* untuk menampilkan modal yang berisi detail dari satu baris pada tabel yang ditunjukkan pada Gambar 3.15.

Modal Title (summary)

Text

Intent

Entities 1

Entities 2

Validate Delete Cancel

Gambar 3.15 Rancangan antarmuka modal *summary*

Dalam Gambar 3.15 terdapat beberapa *text field* yang menunjukkan detail informasi dari kalimat pertanyaan pada fitur ringkasan. Informasi yang ditampilkan berupa kalimat pertanyaan, nilai *intent* dari pertanyaan, nilai *entity key*, dan *entity value* dari pertanyaan. Jika terdapat lebih dari satu *entity* akan muncul *text field* baru pada bagian *entitites*. Admin dapat melakukan validasi kalimat pertanyaan dengan menekan *button* “Validate”, menghapus kalimat pertanyaan dengan menekan *button* “Delete”, dan menutup modal dengan menekan *button* “Cancel”. Jika divalidasi oleh admin, data ringkasan tersebut berpindah ke *tab samples*.

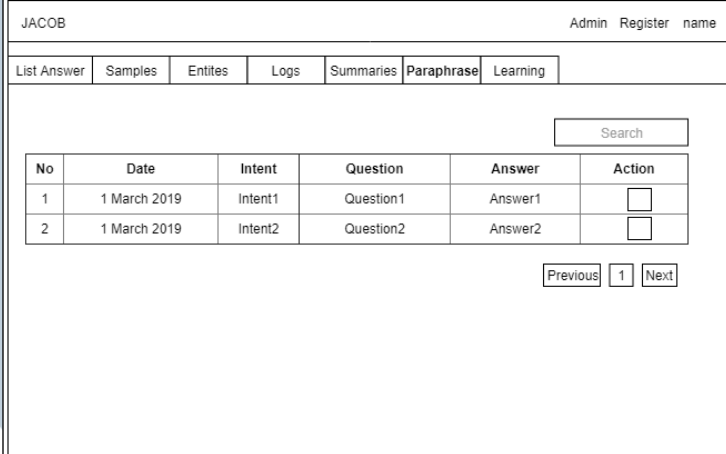
No	Month	User Session	Summary
1	February 2019	Kevin	<ul style="list-style-type: none"> • Summary1 • Summary2 • Summary3 • Summary4 • Summary5
2	February 2019	Toni	<ul style="list-style-type: none"> • Summary1 • Summary2

Gambar 3.16 Rancangan antarmuka *monthly summaries*

Gambar 3.16 merupakan gambar rancangan antarmuka untuk halaman yang menampilkan riwayat *monthly summaries*. Pada tabel *monthly summaries* terdapat kolom bulan dari pertanyaan ditanyakan, nama pengguna, dan kolom ringkasan yang menyatakan ringkasan pertanyaan yang dipilih pada bulan tersebut. Untuk kembali ke halaman *summaries*, dapat dilakukan dengan menekan *button* “Back” pada bagian kiri atas tabel.

B. Rancangan Antarmuka Paraphrase of Answers

Pada Subbab ini dijelaskan rancangan antarmuka untuk menampilkan hasil parafrasa jawaban yang dilakukan oleh modul CLEVEREE. Rancangan antarmuka untuk tampilan halaman parafrasa jawaban ditunjukkan pada Gambar 3.17.



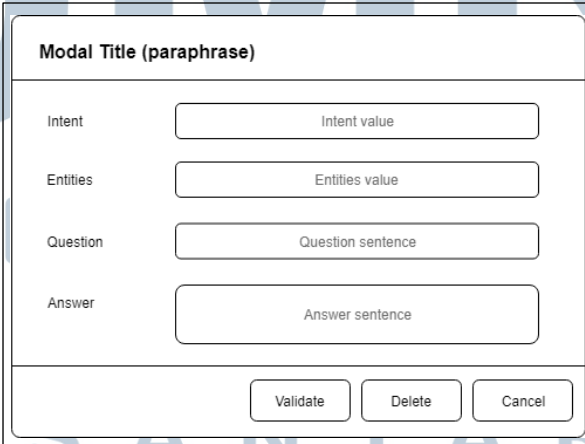
The screenshot shows a web interface for 'Paraphrase'. At the top, there is a header with 'JACOB' on the left and 'Admin Register name' on the right. Below the header is a navigation menu with tabs: 'List Answer', 'Samples', 'Entites', 'Logs', 'Summaries', 'Paraphrase', and 'Learning'. The 'Paraphrase' tab is active. Below the menu is a search bar with the text 'Search'. The main content area contains a table with the following data:

No	Date	Intent	Question	Answer	Action
1	1 March 2019	Intent1	Question1	Answer1	<input type="checkbox"/>
2	1 March 2019	Intent2	Question2	Answer2	<input type="checkbox"/>

Below the table are navigation buttons: 'Previous', '1', and 'Next'.

Gambar 3.17 Rancangan antarmuka *paraphrase of answers*

Pada tabel Gambar 3.17 diperlihatkan tanggal parafrasa jawaban dihasilkan, pertanyaan, nilai *intent* dari pertanyaannya, dan hasil parafrasa pada kolom *answer*. Selain itu, terdapat kolom *action* yang berisi *button* untuk memunculkan modal dan untuk rancangan antarmuda modal ini dapat dilihat pada Gambar 3.18.



The screenshot shows a modal window titled 'Modal Title (paraphrase)'. It contains four input fields with labels and placeholder text:

- Intent: Intent value
- Entities: Entities value
- Question: Question sentence
- Answer: Answer sentence

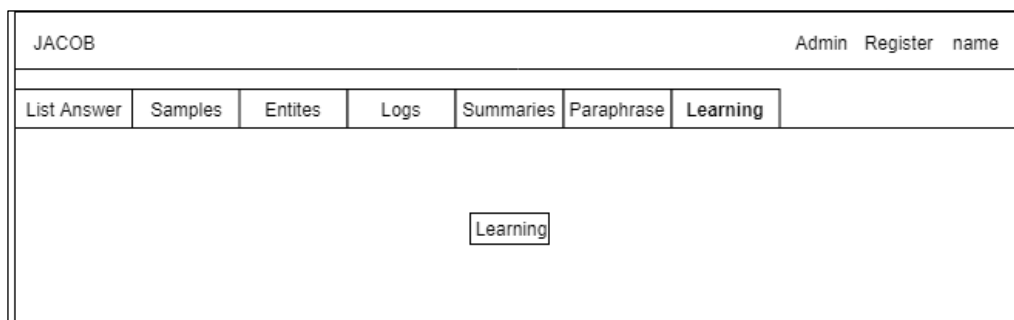
At the bottom of the modal are three buttons: 'Validate', 'Delete', and 'Cancel'.

Gambar 3.18 Rancangan antarmuka modal *paraphrase of answers*

Dalam Gambar 3.18 terdapat beberapa *text field* yang menunjukkan detail informasi dari kalimat parafrasa jawaban. Informasi yang ditampilkan berupa kalimat pertanyaan, nilai *intent* dari pertanyaan, nilai *entity* dari pertanyaan, dan hasil parafrasa jawaban. Jika terdapat lebih dari satu *entity* akan muncul *text field* baru pada bagian *entities*. Admin dapat melakukan validasi hasil parafrasa dengan menekan *button* “Validate”, menghapus dengan menekan *button* “Delete”, dan menutup modal dengan menekan *button* “Cancel”. Sebelum dilakukan validasi, hasil parafrasa jawaban dapat diganti oleh admin jika terdapat kesalahan *grammar* atau informasi tidak lengkap. Jika divalidasi oleh admin, data parafrasa jawaban berpindah ke *tab list answer*.

C. Rancangan Antarmuka Learning

Di dalam Subbab ini dirancang tampilan antarmuka untuk halaman *learning*. Pada rancangan tampilan ini hanya terdapat satu tombol yang dapat digunakan untuk melakukan pelatihan yang ditunjukkan pada Gambar 3.19.

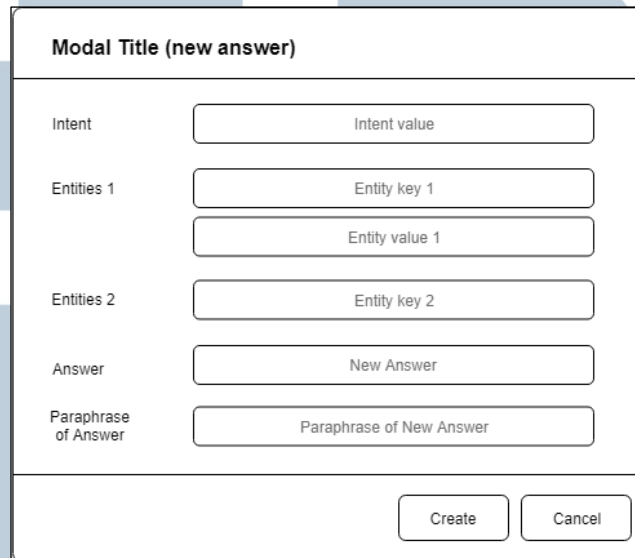


Gambar 3.19 Rancangan antarmuka *learning*

D. Rancangan Antarmuka Pembaharuan Modal Add Answer

Pada Subbab ini dirancang perubahan antarmuka modal *add new answer* pada *tab* List Answer seperti pada Gambar 3.20. Ditambahkan *field* untuk menampilkan hasil dari parafrasa kalimat ketika admin ingin menambahkan

jawaban baru. Perubahan ini dilakukan agar hasil kalimat baru yang dibuat oleh admin dapat langsung ditampilkan dan ditambahkan kalimat parafrasanya ke dalam *database* ketika *button* “Create” ditekan.



The image shows a modal window titled "Modal Title (new answer)". It contains several input fields:

- Intent: Intent value
- Entities 1: Entity key 1, Entity value 1
- Entities 2: Entity key 2
- Answer: New Answer
- Paraphrase of Answer: Paraphrase of New Answer

At the bottom right, there are two buttons: "Create" and "Cancel".

Gambar 3.20 Rancangan antarmuka modal *create new answer*

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA