

Naskah Publikasi

PROYEK TUGAS AKHIR

**IMPLEMENTASI CHATTER BOT SEBAGAI MEDIA INFORMASI
SEKOLAH DENGAN PENDEKATAN NATURAL LANGUAGE
PROCESSING BERBASIS WEB**

Program Studi Teknik Informatika
Fakultas Teknologi Informasi dan Elektro

Disusun oleh:

FAUZI ZAKI MA'RUF

5140411119

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN ELEKTRO
UNIVERSITAS TEKNOLOGI YOGYAKARTA
2019**

Naskah Publikasi

**IMPLEMENTASI CHATTER BOT SEBAGAI MEDIA INFORMASI
SEKOLAH DENGAN PENDEKATAN NATURAL LANGUAGE
PROCESSING BERBASIS WEB**

Disusun oleh:
FAUZI ZAKI MA'RUF
5140411119

Telah disetujui oleh pembimbing

Pembimbing

Rianto, S.Kom., M.Eng.

Tanggal :.....

Implementasi *Chatter Bot* Sebagai Media Informasi Sekolah dengan Pendekatan *Natural Language Processing* Berbasis Web

Fauzi Zaki Ma'ruf

*Program Studi Teknik Informatika, Fakultas Teknologi Informasi dan Elektro
Universitas Teknologi Yogyakarta
Jl. Ringroad Utara Jombor Sleman Yogyakarta
E-mail : fauzizaki15@gmail.com*

ABSTRAK

Pertumbuhan madrasah yang ada di Indonesia dianggap belum menunjukkan kemajuan dalam ranah pemanfaatan teknologi. Penulis berasumsi bahwa di ranah madrasah belum memiliki kesan pemanfaatan teknologi yang ada pada lingkungan sekolah, baik dari sisi pegawai, maupun siswa. Padahal jika penulis lihat, bahwasanya ada beberapa masalah yang dapat diambil pada lingkungan madrasah. Masalah tersebut salah satunya yaitu penanganan pusat informasi. Disamping itu, meningkatnya pengguna gadget untuk mengakses informasi yang dibutuhkan juga mengalami peningkatan yang cukup signifikan. Pada masalah tersebut, penulis berusaha untuk mencari solusi yang tepat untuk bisa menangani masalah tanya jawab yang dilakukan oleh penanya dan penyedia informasi secara langsung dan bersamaan dengan bantuan teknologi yang dapat di implementasikan pada lingkungan sekolah. Teknologi yang paling tepat diimplementasikan kedalam masalah tanya jawab tersebut yaitu implementasi teknologi dari Chatter Bot atau Question Answering. Chatter Bot merupakan program komputer yang dapat berinteraksi dengan pengguna (manusia) menggunakan bahasa natural. Dalam tahapan belajar komputer untuk dapat memahami kalimat dari pengguna, tentu komputer memiliki teknik belajar yang berbeda-beda untuk dapat memahami kalimat yang diajukan oleh pengguna dan tergantung dari tingkat kompleksitas pemrograman yang penulis bangun. Supaya komputer dapat memahami kalimat dari pengguna, ada beberapa teknik yang dapat dipakai dari pendekatan Natural Language Processing (NLP). Teknik dasar yang digunakan untuk memahamkan komputer yaitu sebagian besar menggunakan implementasi dari tokenize, stopword, stemming, correction dan analyzing.

Kata kunci : *Chatter Bot, Natural Language Processing, Informasi Sekolah, Question Answering*

1. PENDAHULUAN

1.1. Latar Belakang

Pertumbuhan madrasah yang ada di Indonesia dianggap belum menunjukkan kemajuan dalam ranah pemanfaatan teknologi, sehingga penulis berasumsi bahwa di ranah madrasah belum memiliki kesan pemanfaatan teknologi yang ada pada lingkungan sekolah, baik dari sisi pegawai, siswa maupun orangtua. Padahal jika penulis lihat, bahwasanya ada beberapa masalah yang dapat diambil pada lingkungan madrasah. Masalah tersebut salah satunya yaitu penanganan pusat informasi. Disamping itu, meningkatnya pengguna gadget untuk mengakses informasi yang dibutuhkan juga mengalami peningkatan yang cukup signifikan. Dengan adanya peningkatan jumlah pengguna tersebut, tentunya perlu perhatian khusus upaya untuk memfasilitasi pengguna pegawai, siswa maupun orangtua dalam mencari informasi yang berkaitan dengan sekolah. Jika penulis lihat bahwa ada beberapa kendala yang terjadi pada lingkungan sekolah. Dimulai dari sisi pelayanan dari pegawai Tata Usaha (TU), yaitu tidak dapat menangani banyak permintaan pertanyaan dari pegawai, siswa maupun orangtua secara langsung dan

bersamaan. Teknologi yang paling tepat diimplementasikan kedalam masalah tanya jawab tersebut, yaitu implementasi teknologi dari Chatter Bot. Chatter Bot adalah sebuah program yang dapat melakukan obrolan dalam bahasa alami mengenai sebuah topik yang ada dalam model pengetahuan Chatter Bot tersebut. Artinya, Chatter Bot harus bisa mengenali dan merespon kata-kata yang diminta oleh pengguna secara otomatis.

2. LANDASAN TEORI

2.1. Informasi

Informasi adalah data yang diolah menjadi suatu bentuk yang berguna bagi penerimanya dan memiliki nilai bagi pengambilan keputusan saat ini atau di masa mendatang.[2]

2.2. Chatter Bot

Chatter Bot adalah program komputer yang dirancang untuk dapat melakukan interaksi dengan manusia melalui pesan teks, maupun suara. Chatter Bot biasanya juga dibekali dengan kecerdasan buatan dan pemrosesan bahasa alami yang membuatnya menjadi program komputer yang cerdas dan dapat

menjawab pertanyaan yang diberikan oleh manusia.[7]

2.3. Pedoman Umum Ejaan Bahasa Indonesia (PUEBI)

PUEBI adalah pedoman umum tentang bagaimana cara memahami aturan huruf, kata dan tanda baca yang bisa di pakai, guna memperoleh aturan bahasa Indonesia yang baik dan benar.[6]

2.4. *Natural Language Processing* (NLP)

Natural Language Processing (NLP) adalah bertujuan untuk mengembangkan program komputer yang dapat memproses bahasa alami yang dapat dipahami oleh manusia. Tujuannya yaitu untuk memungkinkan komputer dapat berkomunikasi dengan manusia dengan cara yang sama seperti manusia berkomunikasi dengan manusia lain.[9]

2.5. *Flowchart*

Flowchart adalah suatu diagram yang menggambarkan susunan logika suatu program.[1]

2.6. Basis Data

Basis data adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi.[3]

2.7. Diagram Relasi Entitas (ERD)

Diagram Relasi Entitas (ERD) merupakan kumpulan dari objek-objek dengan sifat (property) yang sama dan diidentifikasi oleh enterprise yang mempunyai eksistensi yang bersifat independent.[4]

2.8. Diagram Alir Data (DAD)

Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengatur dari masukan (input) dan keluaran (output).[10]

3. METODOLOGI PENELITIAN

3.1. Metode Penelitian

Objek dari penelitian ini adalah memanfaatkan data pegawai, data siswa, data pembayaran, data mata pelajaran dan data kelas yang dapat diolah menjadi sebuah informasi. Tentunya sistem sudah bisa membedakan antara pengguna pegawai, siswa maupun orangtua untuk membatasi informasi yang dapat tersampaikan kepada masing-masing pengguna

melalui hak akses yang telah dibuat. Data tersebut akan digunakan sebagai pengelolaan data (*resource*).

3.2. Metode Pengumpulan Data

Pengumpulan data adalah suatu metode yang digunakan untuk mendapatkan suatu informasi yang harus dikerjakan pada saat pembuatan sistem. Untuk mempermudah penelitian yang sedang dilakukan, peneliti menggunakan beberapa metode pengumpulan data yang penulis pakai, diantaranya yaitu:

a. Wawancara

Wawancara sangat berguna bagi penulis untuk mendapatkan informasi dari narasumber secara langsung. Jenis wawancara yang penulis pakai yaitu dengan wawancara terstruktur, yang mengedepankan pertanyaan yang spesifik. Penulis juga menggaris bawahi topik tentang informasi dari hasil melakukan wawancara dengan salah satu kepala bidang akademik madrasah dan pegawai yang mengampu disana, yaitu bapak Fajar Rahmadi.

b. Studi Literatur

Studi literatur digunakan oleh penulis untuk melakukan kajian teori melalui buku-buku dan media web yang berkaitan dengan media informasi yang akan dikembangkan.

c. Menganalisis Data

Langkah selanjutnya adalah menganalisis data-data yang sudah terkumpul, kemudian mengkaitkan antara data-data yang sudah terkumpul dari proses pengumpulan data melalui wawancara, buku-buku dan jurnal untuk memperoleh hasil yang lebih akurat.

3.3. Metode Pengembangan Aplikasi

Dalam melakukan penelitian, penulis melakukan beberapa langkah pengembangan aplikasi. Tahapan tersebut dilakukan agar dapat memperoleh hasil yang diharapkan. Adapun tahapan-tahapan sebagai berikut:

a. Analisis

Analisis yang dilakukan pada tahap ini merupakan analisis terhadap data yang sudah diperoleh dari tahap pengumpulan data sebelumnya.

b. Desain dan Perancangan

Pada tahap ini, penulis melakukan perancangan sistem yang dilakukan dengan dua tahap, yaitu:

1) Pemodelan Sistem

Pemodelan sistem dilakukan agar penulis dapat menjelaskan sistem yang akan dibangun. Pemodelan dilakukan dengan Diagram Alir Data (DAD). Dimulai dengan diagram konteks kemudian diagram alir data level n.

2) Perancangan Antar Muka

Perancangan antar muka dilakukan penulis agar aplikasi yang telah dibuat memiliki tampilan yang baik bagi pengguna. Dalam tahap ini, penulis merancang antar muka dengan memberikan

rancangan fitur dan menu yang akan ada pada aplikasi.

3) Pengkodean (Coding)

Tahap ini dilakukan penulis untuk mengkoneksikan antara antar muka aplikasi dengan basis data pada aplikasi menggunakan Bahasa pemrograman Node JS sebagai pemrograman di sisi server dan MYSQL sebagai basis datanya.

4) Implementasi

Pada tahap implementasi dilakukan dengan mencoba aplikasi yang telah dibuat, mencari sebuah kesalahan kode program yang ada, memastikan agar aplikasi yang akan digunakan benar-benar bersih dari bug program.

3.4. Perangkat Pendukung

Spesifikasi minimum untuk menjalankan sistem ini adalah sebagai berikut:

- Perangkat Keras yang Digunakan:
 - Intel(R) Core(TM) i3-6100U CPU @ 2.30GHz (4 CPUs).
 - RAM 4 GB.
 - Hardisk 500 GB.
- Perangkat Lunak yang Digunakan:
 - Sistem Operasi Windows 10.
 - XAMPP 7.0.15.
 - Node JS 10.1.0.

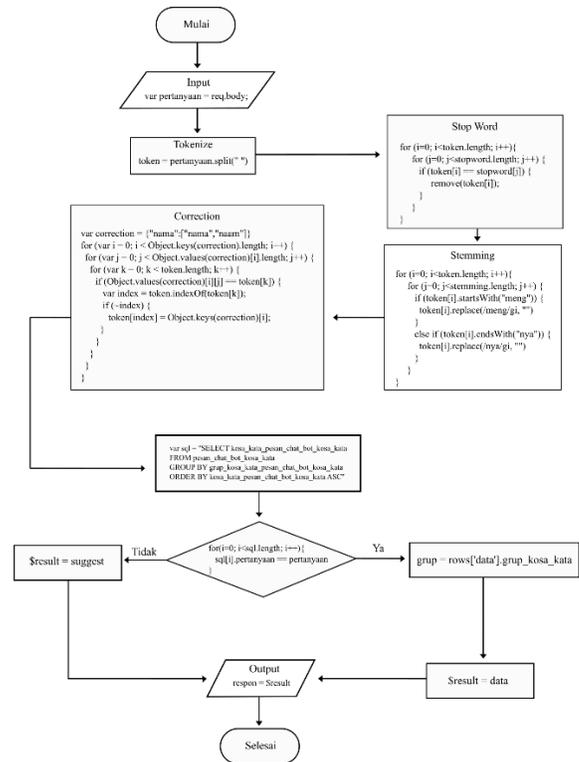
4. HASIL DAN PEMBAHASAN

4.1. Perancangan Sistem Aplikasi Chatter Bot

Pada bagian ini akan dibahas perancangan sistem dari aplikasi *Chatter Bot*. Sistem ini dirancang melalui beberapa tahap disesuaikan dengan kebutuhan sistem yang ada, yaitu: perancangan aplikasi, perancangan proses, perancangan basis data, dan perancangan antarmuka sistem.

a. Perancangan Arsitektur Model Pemrograman

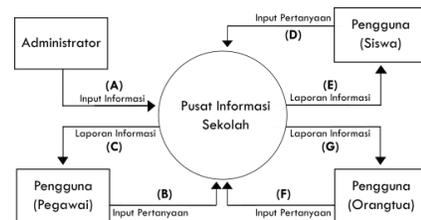
Arsitektur model pemrograman yang dipakai oleh penulis salah satunya menggunakan kosa kata yang memiliki masing-masing grup kosa kata berupa pegawai, siswa, kelas, mata pelajaran dan pembayaran, Untuk contoh kasis arsitektur pemrograman bisa dilihat seperti pada Gambar 4. 1.



Gambar 4. 1 Flowchart Model Pemrograman

b. Diagram Konteks

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh input ke dalam sistem atau output dari sistem yang memberi gambaran tentang keseluruhan sistem, yaitu:

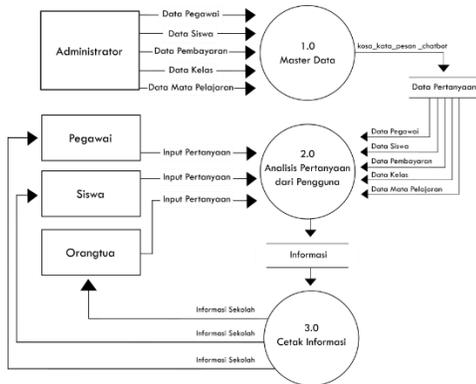


Gambar 4. 2 Diagram Konteks Pusat Informasi Sekolah

c. Perancangan Diagram Alir Data (DAD)

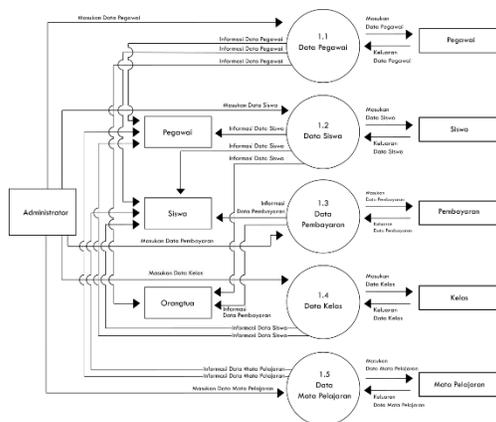
Diagram alir data merupakan suatu model logika yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan dan data apa yang di hasilkan dari proses tersebut. Adapun gambar DAD menurut per levelnya, yaitu:

1. DAD Level 0



Gambar 4. 3 DAD Level 0

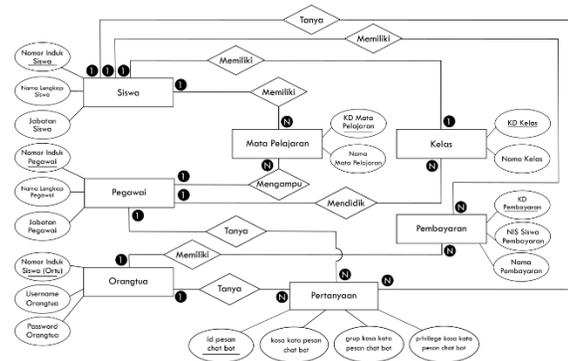
2. DAD Level 1 Proses 1



Gambar 4. 4 DAD Level 1 Proses 1

d. Diagram Relasi Antar Entitas (ERD)

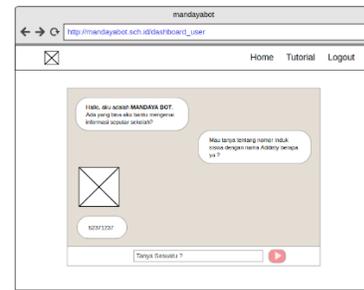
ERD (*Entity Relationship Diagram*) adalah suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. Adapun tampilan ERD seperti pada Gambar 4.5, yaitu:



Gambar 4. 5 Diagram Relasi Antar Entitas

4.2. Perancangan Antar Muka Aplikasi

Antarmuka (Interface) merupakan mekanisme komunikasi antara pengguna (user) dengan sistem. Adapun tampilan antar muka seperti terlihat pada Gambar 4. 6, yaitu:



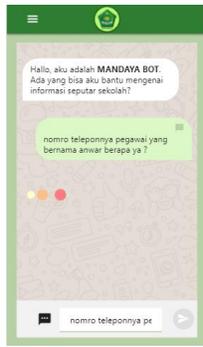
Gambar 4. 6 Interface Halaman Chatting

4.3. Perancangan Aplikasi Chatter Bot

Implementasi pada *Natural Language Processing* pada *Chatter Bot* yang penulis bangun memiliki beberapa teknik. Teknik pemrosesan bahasa yang dipakai oleh penulis sudah dijelaskan dasar teori pada landasan teori tentang *Natural Language Processing*. Ada 2 proses yang harus dipenuhi supaya aplikasi dapat menjawab pertanyaan dari pengguna, yaitu:

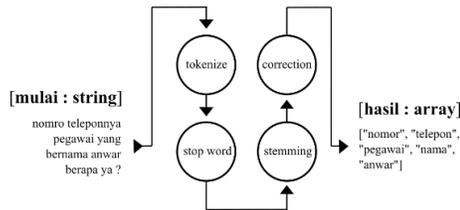
a. Pemrosesan Kalimat

- 1) Pengguna memasukan pertanyaan berupa "nomor teleponnya pegawai yang bernama anwar berapa ya?", seperti yang terlihat pada gambar berikut:



Gambar 4. 7 Memulai Pertanyaan ke Sistem

- 2) Kalimat tersebut akan masuk kedalam beberapa proses seperti *tokenize*, *stop word*, *stemming*, *correction* dan *stemming*, seperti pada Gambar 4. 8:



Gambar 4. 8 Proses Pemecahan array Pertanyaan Pengguna

Pada proses *tokenize*, yaitu memungkinkan perubahan kalimat string “nomro teleponnya pegawai yang bernama anwar berapa ya ?” menjadi array [“nomro”, “teleponnya”, “pegawai”, “yang”, “bernama”, “anwar”, “berapa”, “ya”, “?”]. Kemudian masuk ke proses *stop word*, yaitu proses penghapusan kata yang tidak digunakan pada aplikasi *Chatter Bot* tentang informasi sekolah. Dalam kasus ini, kata yang terhapus yaitu: [“yang”, “berapa”, “ya”, “?”]. Kemudian yaitu masuk pada proses *stemming*, yaitu mendeteksi kata yang berimbuhan prefix, suffiks dan konfiks akan diubah menjadi kalimat dasar. Kemudian yaitu masuk kedalam proses *correction*, yaitu memungkinkan kata dari pertanyaan pengguna yang memiliki potensi salah ketik akan diperbaiki oleh sistem.

- 3) Setelah melewati setidaknya empat proses tersebut, kemudian dihasilkan kata yang telah tersaring, yaitu: [“nomor”, “telepon”, “pegawai”, “nama”, “anwar”].
- 4) Kemudian langkah terakhir *analyzing*. Proses *analyzing* memungkinkan kata didalam array dari pertanyaan pengguna yang sudah melewati proses-proses sebelumnya akan dicocokkan dengan seluruh pertanyaan yang ada di basis

data. Pada kasus ini, pertanyaan yang tersedia pada basis data hanya dicontohkan 5 pertanyaan saja. Kemudian array tersebut di *looping* dengan seluruh pertanyaan yang ada di basis data, sehingga akan menampilkan proses sebagai berikut:

```

    nomor {
      "0. nomor == nomor telepon pegawai",
      "1. nomor == nomor telepon siswa",
      "2. null == daftar pengampu mata pelajaran kelas",
      "3. null == nama lengkap siswa",
      "4. null == detail pembayaran siswa",
    }
    telepon {
      "0. telepon == nomor telepon pegawai",
      "1. telepon == nomor telepon siswa",
      "2. null == daftar pengampu mata pelajaran kelas",
      "3. null == nama lengkap siswa",
      "4. null == detail pembayaran siswa",
    }
    pegawai {
      "0. pegawai == nomor telepon pegawai",
      "1. null == nomor telepon siswa",
      "2. null == daftar pengampu mata pelajaran kelas",
      "3. null == nama lengkap siswa",
      "4. null == detail pembayaran siswa",
    }
    nama {
      "0. null == nomor telepon pegawai",
      "1. null == nomor telepon siswa",
      "2. null == daftar pengampu mata pelajaran kelas",
      "3. nama == nama lengkap siswa",
      "4. null == detail pembayaran siswa",
    }
    anwar {
      "0. null == nomor telepon pegawai",
      "1. null == nomor telepon siswa",
      "2. null == daftar pengampu mata pelajaran kelas",
      "3. null == nama lengkap siswa",
      "4. null == detail pembayaran siswa",
    }
  
```

Gambar 4. 9 Proses Pencocokan Pertanyaan Pengguna ke-1

Pada Gambar 4. 9 menjelaskan tentang proses *looping* array kalimat pertanyaan dari pengguna dengan *looping* semua pertanyaan yang ada pada basis data. Pada gambar tersebut dijelaskan bahwa ada keterangan null dan lain sebagainya. Null berarti bernilai “0” dan selain null akan bernilai “1”. Hasil tersebut dikumpulkan menjadi satu, hingga mendapati hasil seperti pada Gambar 4. 10:

```

    [
      1,1,0,0,0,1,1,0,0,0,1,0,
      0,0,0,0,0,0,1,0,0,0,0,0
    ]
  
```

Gambar 4. 10 Proses Pencocokan Pertanyaan Pengguna ke-2

tahap selanjutnya yaitu proses memecah sekumpulan array yang berjumlah 25 seperti pada Gambar 4.10 dengan total array yang terdapat pada pertanyaan pengguna. Pada kasus ini array pertanyaan dari pengguna yang tersaring dari proses-proses sebelumnya yaitu berjumlah 5, yaitu: [“nomor”, “telepon”, “pegawai”, “nama”, “anwar”], sehingga akan menampilkan output sebagai berikut:

```
[
  nomor -> [1,1,0,0,0],
  telepon -> [1,1,0,0,0],
  pegawai -> [1,0,0,0,0],
  nama -> [0,0,0,1,0],
  anwar -> [0,0,0,0,0]
]
```

Gambar 4. 11 Pemotongan Array Sesuai dengan Hasil Saring

Kemudian setelah array tersebut dipecah menjadi 5 bagian, sistem akan mengelompokannya menjadi array lagi, seperti pada gambar 4.12 dan 4.13. Langkah ini digunakan untuk mencari total persamaan pertanyaan kata dari pengguna dengan daftar seluruh pertanyaan yang sudah disiapkan pada basis data.

```
[[1, 1, 0, 0, 0], [1, 1, 0, 0, 0], [1, 0, 0, 0, 0], [0, 0, 0, 1, 0], [0, 0, 0, 0, 0]]
```

Gambar 4. 12 Penjumlahan Array dari Pertanyaan Pengguna ke-1

```
[
  1,1,1,0,0,1,1,0,0,0,0,0,
  0,0,0,0,0,0,1,0,0,0,0,0
]
```

Gambar 4. 13 Penjumlahan Array dari Pertanyaan Pengguna ke-2

Pada langkah selanjutnya yaitu mengelompokan array, seperti pada Gambar 4.13, selanjutnya array tersebut akan dibagi menjadi 5 bagian. Terlihat seperti pada Gambar 4.14, untuk dijumlahkan total array pertanyaan dari pengguna yang paling cocok dengan seluruh daftar pertanyaan yang ada di basis data. Pencocokan tersebut di dapatkan dari hasil jumlah dari masing-masing array yang telah dikelompokan menjadi 5 tersebut, terlihat pada Gambar 4.15 bahwa total kesamaan berhasil dijumlahkan dan dilanjutkan dengan membuat variabel untuk menampung array untuk data pertanyaan yang ada di basis data dengan menggunakan *looping* dan menampung bersamaan dengan jumlah kecocokan dari pertanyaan pengguna, terlihat seperti pada Gambar 4.16.

```
[
  [1,1,1,0,0],
  [1,1,0,0,0],
  [0,0,0,0,0],
  [0,0,0,1,0],
  [0,0,0,0,0]
]
```

Gambar 4. 14 Pra Penghitungan Array Kecocokan

```
[3,2,0,1,0]
```

Gambar 4. 15 Pasca Penghitungan Array Kecocokan

```
{
  "id":0,"total_match":3,"kalimat":"nomor telepon pegawai","split_total_kalimat":3},
  {"id":1,"total_match":2,"kalimat":"nomor telepon siswa","split_total_kalimat":3},
  {"id":2,"total_match":0,"kalimat":"daftar pengampu mata pelajaran kelas","split_total_kalimat":5},
  {"id":3,"total_match":1,"kalimat":"nama lengkap siswa","split_total_kalimat":3},
  {"id":4,"total_match":0,"kalimat":"detail pembayaran siswa","split_total_kalimat":3}
}
```

Gambar 4. 16 Tampilan Data dengan Total Kecocokan

Proses selanjutnya yaitu menjumlahkan semua data di dalam array pada Gambar 4.15 dan mencari pertanyaan pengguna yang memiliki kecocokan tertinggi. Setelah data array tertinggi telah didapatkan, seperti pada gambar berikut:

```
[3]
```

Gambar 4. 17 Proses Pengambilan Data total_match Tertinggi ke-1

Kemudian dicari kalimat yang ada di basis data, yang memiliki hasil penjumlahan total_match seperti pada Gambar 4.17. Pada kasus ini kebetulan hanya memiliki 1 kalimat pertanyaan yang ada di basis data yang tersaring seperti pada Gambar 4.18.

```
{
  "id":0,"total_match":3,"kalimat":"nomor telepon pegawai","split_total_kalimat":3}
}
```

Gambar 4. 18 Proses Pengambilan Data total_match Tertinggi ke-2

Setelah data yang dikumpulkan dari key “total_match” yang terkumpul, selanjutnya yaitu mencari data pada key split_total_kalimat yang terendah dari array pada Gambar 4.19, seperti pada gambar berikut:

```
[3]
```

Gambar 4. 19 Kumpulan split_total_kalimat

Pada kasus ini kebetulan hanya memiliki 1 kalimat pertanyaan yang ada di basis data yang memiliki split_total_kalimat dengan jumlah seperti pada Gambar 4.19, dan hasil terendah akan di proses hingga menghasilkan data seperti pada Gambar 4.20.

```
[3]
```

Gambar 4. 20 Hasil split_total_kalimat Terendah

Setelah split_total_kalimat terendah berhasil ditemukan, langkah selanjutnya yaitu melihat isi

hasil saringan pertanyaan dengan `split_total_kalimat` terendah. Adapun hasil kalimat dengan `split` terendah yang berhasil di simpulkan:

```
{ "id":0,"total_match":3,"kalimat":"nomor telepon pegawai","split_total_kalimat":3}
```

Gambar 4. 21 Hasil Kesimpulan dari Pertanyaan 1

```
["nomor", "telepon", "pegawai"]
```

Gambar 4. 22 Hasil Kesimpulan dari Pertanyaan 2

b. Pemrosesan Nama (Objek yang dicari)

Setelah proses mencari pertanyaan yang diajukan dari pengguna ditemukan pada basis data, langkah selanjutnya yaitu proses mencari nama tujuan yang dimaksud oleh pengguna. Kembali lagi pada pertanyaan pengguna yang telah disimpulkan menjadi array oleh sistem yaitu: ["nomor","telepon","pegawai","nama","anwar"]. Pada array pertanyaan tersebut *dilooping* dengan seluruh nama yang ada di basis data. Setelah nama *terlooping*, kemudian kalimat yang sudah dibentuk array tersebut diolah dan sistem kebetulan mendeteksi ada nama pegawai yang bernama "anwar". Adapun fungsi untuk menghapus seluruh array sebelum nama, sebagai berikut:

```
a). awal : ["nomor", "telepon", "pegawai", "nomor", "anwar"]
b). terdeteksi : ["nomor", "telepon", "pegawai", "nomor", "anwar"]
                |-----|
                |terhapus|
c). hasil : ["anwar"]
```

Gambar 4. 23 Output Pada Array Setelah Terdeteksi Nama Pegawai

Selanjutnya yaitu membuat perulangan yang bersarang antara data nama siswa yang ada pada basis data dengan data yang ada pada Gambar 4.23 untuk dicarikan kondisi nama yang paling tepat dengan cara pemotongan string dengan fungsi `pop` yang hasilnya akan disimpan pada variabel "arr". Terlihat pada Gambar 4.23 bagian c, pada kasus pertanyaan ini menjelaskan bahwa tidak ditemukannya array kata lagi setelah nama di deteksi. Untuk mempermudah penjelasan mengenai pencarian nama, maka penulis telah menyediakan tampilan visual seperti pada Gambar 4.24 untuk menjelaskan cara mencari nama beserta nama panjangnya.

Contoh Output (hanya contoh pada Gambar 4.24 yaitu proses untuk mencari nama panjang atau bukan merupakan kasus pertanyaan yang penulis angkat pada contoh pertanyaan pengguna sebelumnya):

```
[0] ["anwar", "berapa", "?", █████ ]
[1] ["anwar", "berapa", █████ ]
[2] ["anwar", █████ ]
```

Gambar 4. 24 Output Proses Cari Nama 1

Pada Gambar 4.24 menjelaskan bahwa, proses tersebut berfungsi sebagai proses seleksi untuk mencari nama yang di maksud beserta dengan nama panjangnya. Proses tersebut dilakukan dengan cara menghapus array satu per satu dengan menggunakan fungsi dari Javascript berupa `array pop()`. Untuk proses pencocokan nama, lebih detailnya bisa dilihat pada Gambar 4.25.

```
0 : anwar berapa ? = Achmad Ramzi Adriansyah
0 : anwar berapa ? = Addisty Ayu Rinjani
0 : anwar berapa ? = Anwar Hermanto
0 : anwar berapa = Achmad Ramzi Adriansyah
0 : anwar berapa = Addisty Ayu Rinjani
0 : anwar berapa = Anwar Hermanto
0 : anwar = Achmad Ramzi Adriansyah
0 : anwar = Addisty Ayu Rinjani
1 : anwar = Anwar Hermanto
```

Gambar 4. 25 Output Proses Cari Nama 2

```
["anwar"]
```

Gambar 4. 26 Output Akhir Cari Nama

Pada Gambar 4.25 menjelaskan bahwa array yang lolos pada proses saring akan melalui proses lagi untuk mencari nama yang dimaksud. Pada kasus ini, angka 1 berarti nama yang dicari pengguna telah ditemukan. Seperti pada Gambar 4.26 menjelaskan bahwa nama telah ditemukan. Setelah kalimat pertanyaan dari pengguna berhasil tersimpulkan dan nama yang dimaksud berhasil ditemukan pada basis data, maka langkah selanjutnya yaitu menangkap grup kalimat pertanyaan yang ada pada basis data sesuai dengan kalimat pertanyaan dari pengguna yang berhasil di identifikasi. Grup kalimat pertanyaan berfungsi untuk mengarahkan kalimat yang tersimpulkan kepada kolom tabel pada basis data yang memiliki data yang sedang dicari oleh pengguna. Untuk contoh pengambilan grup pertanyaan pada basis data sebagai berikut:

328	alamat rumah pegawai	alamat_pegawai
329	nomor telepon pegawai	no_handphone_pegawai
330	nomor induk pegawai	nip_pegawai

↓

nip_pegawai	nama_pegawai	no_handphone_pegawai
19581112679031002	Anwar Hermanto	08121668218

Gambar 4. 27 Pengambilan Keputusan Jawaban

Seperti yang terlihat pada Gambar 4.27 bahwa data yang dicari berhasil di index pada basis data. Data tersebut melampirkan data nomor induk pegawai (NIP), nama pegawai dan nomor *handphone* pegawai. Yang menjadi perhatian pada data yang

terlampir tersebut yaitu data nomor induk pegawai. Nomor induk pegawai digunakan untuk mencantumkan foto pegawai yang di tuju (dalam kasus ini yaitu data atas nama pegawai “Anwar”). Foto pegawai tersimpan dengan identitas nomor induk pegawai. Selanjutnya aplikasi hanya akan memanggil ekstensi foto tersebut sesuai dengan nomor induk pegawai yang dimaksud (“Anwar”). Kemudian selanjutnya Chatter Bot akan memberikan balasan atau jawaban dengan mengirimkan 2 parameter data yang terdiri foto dan informasi terkait, seperti pada gambar berikut:



Gambar 4. 28 Hasil Jawaban Chatter Bot

5. PENUTUP

5.1. Kesimpulan

Dari penelitian dan tulisan yang telah penulis uraikan, maka dapat ditarik kesimpulan sebagai berikut:

- Aplikasi *Chatter Bot* yang dibangun menggunakan pendekatan Natural Language Processing berhasil dibuat.
- Pemrosesan *stemming* dapat membedakan kata yang memiliki gabungan atau tidak. Contoh yaitu kata: “mata pelajaran”. Sehingga aplikasi *Chatter Bot* tidak akan melakukan stem pada kata pelajaran dengan konfiks pel- dan -an, dikarenakan gabungan kata tersebut sudah masuk dalam satu kesatuan kata “mata pelajaran”.
- Adanya kelemahan pada algoritma *analyzing*. Semakin banyak pertanyaan yang disiapkan pada basis data, maka semakin baik *Chatter Bot* dalam menganalisis jawaban yang tepat.
- Aplikasi *Chatter Bot* mampu mengoreksi setengah kata yang typo, dimulai dari akhir dari total kata yang dimasukkan oleh pengguna.

5.2. Saran

Penulis menyadari masih ada kekurangan yang ada pada aplikasi *Chatter Bot*. Masih banyak hal yang dapat dilakukan untuk mengembangkan aplikasi ini agar menjadi lebih baik lagi, antara lain:

- Menambahkan algoritma *classification* untuk mempercepat aplikasi dalam menentukan jawaban yang tepat.
- Meningkatkan algoritma lemmatization pada aplikasi Chatter Bot untuk mendeteksi dan melakukan aksi ketika adanya bentuk lain dari kata di dalam kalimat pertanyaan.
- Meningkatkan algoritma untuk *correction*, sehingga dapat mendeteksi seluruh jenis salah ketik kata yang diinputkan melalui pengguna.

DAFTAR PUSTAKA

- Elsa, H.A.S.N., (2014), Teori Flow Chart, (<http://hasneteamblogspot.blogspot.com/2014/10/teori-flowchart-flowchart-sop-pajak-1.html>), akses 15 Agustus 2018.
- Hartono, B., (2013), Sistem Informasi Manajemen Berbasis Komputer, Jakarta: Jakarta: Rineka Cipta.
- Hindrianto, H., (2012), Pengertian Database, (<https://www.termasmedia.com/lainnya/software/69-pengertian-database.html>), akses 14 Juli 2018.
- Indrajani, (2018), Database Design All in One : Theory, Practice and Case Study, Jakarta: Elex Media Komputindo.
- Maskur, (2016), Perancangan CHATBOT Pusat Informasi Mahasiswa Menggunakan AIML Sebagai Virtual Assistant Berbasis Web, Kinetik, Vol 1(3), 123-128.
- Mustakim, Harimansyah, G., Qodratillah, M.T., Ruskhan, A.G., Sriyanto, Sasangka, S.S.T.W., Zahra, S., Sitanggang, S.R.H., Amalia, D., Solihah, A., Darnis, A.D., (2016), Pedoman Umum Ejaan Bahasa Indonesia (Edisi Keempat), Jakarta: Badan Pengembangan dan Pembinaan Bahasa.
- Nila, E.S.C.P. dan Afrianto, I., (2015), Rancang Bangun Aplikasi chatbot Informasi Objek Wisata Kota Bandung dengan Pendekatan Natural Language Processing, Jurnal Ilmiah Komputer dan Informatika (KOMPUTA), Vol 4(1), 2089-9033.
- Pernanda, A.Y., (2017), Stopword Removal Bahasa Indonesia dengan Python Sastrawi, (<https://devtrik.com/python/stopword-removal-bahasa-indonesia-python-sastrawi/>), akses 20 Desember 2018.
- Sibarani, E.M., (2013), A Study of Parsing Process on Natural Language Processing in Bahasa Indonesia, (<https://www.researchgate.net/publication/262215218>), akses 9 Desember 2018.
- Sukamto, R. A. dan Shalahuddin, M., (2013), Modul Pembelajaran Rekayasa Perangkat Lunak, Bandung: Modula.

