

## **CHATBOT PADA SMART COOPERATIVE ORIENTED PROBLEM MENGUNAKAN NATURAL LANGUAGE PROCESSING DAN NAIVE BAYES CLASSIFIER**

### ***CHATBOT ON SMART COOPERATIVE ORIENTED PROBLEMS USING NATURAL LANGUAGE PROCESSING AND NAIVE BAYES CLASSIFIER***

**Yogi Yunefri<sup>1</sup>, Yogi Ersan Fadrial<sup>2</sup>, Sutejo<sup>3</sup>**

<sup>123</sup>Universitas lancing Kuning  
[yogiyunefri@unilak.ac.id](mailto:yogiyunefri@unilak.ac.id),

#### **ABSTRACT**

*The development of science and technology requires universities as formal educational institutions, to be able to produce qualified and competent graduates. Learning in higher education should be more innovative and creative in producing graduates and responsive to the needs of the workforce. The obstacle experienced by lecturers in teaching Data Structure courses is that there is no learning model that brings students closer to abstract theories that are difficult for students to understand. To overcome these problems, it is necessary to have a learning model that can improve the quality of learning that students have by conducting learning chatbots. The method used in this study is the Naive Bayes Classifier, from the software that was built it can help the lecturers in the data structure course in the student tutoring process. The Naive Bayes Classifier method can be implemented to build a chatbot software for student tutoring.*

**Keywords:** *Naive Bayes Classifier, Chatbot, Learning Model.*

#### **ABSTRAK**

Perkembangan ilmu pengetahuan dan teknologi menuntut perguruan tinggi sebagai lembaga pendidikan yang formal, untuk dapat menghasilkan lulusan yang bermutu dan kompeten. Pembelajaran pada perguruan tinggi seharusnya lebih inovatif dan kreatif dalam menghasilkan lulusan serta responsif akan kebutuhan tenaga kerja. Kendala yang dialami dosen dalam mengajar matakuliah Struktur Data adalah belum adanya model pembelajaran yang mendekatkan mahasiswa dengan teori abstrak yang sulit untuk dipahami mahasiswa, untuk mengatasi permasalahan tersebut maka perlu adanya model pembelajaran yang dapat meningkatkan kualitas belajar yang dimiliki mahasiswa dengan cara melakukan chatbot belajar. Metode yang digunakan dalam penelitian ini adalah Naive Bayes Classifier, dari Perangkat lunak yang dibangun dapat membantu pihak dosen pada mata kuliah struktur data dalam proses bimbingan belajar mahasiswa. Metode Naive Bayes Classifier dapat diimplementasikan untuk membangun perangkat lunak chatbot bimbingan belajar mahasiswa.

**Kata Kunci :** Naive Bayes Classifier, Chatbot, Model Pembelajaran.

#### **PENDAHULUAN**

Akibat covid- 19, pembelajaran di tingkatan Sekolah Dasar, Sekolah Menengah Pertama, Sekolah Menengah Atas/ Kejuruan, dan Sekolah Tinggi/ Universitas, Semua jenjang pendidikan yang terletak dibawah Kementerian Pendidikan seluruhnya mendapatkan dampak *negative*, aplikasi web yang semakin pesat sejak munculnya teknologi internet sangat membantu dalam kemudahan serta kecepatan

pengiriman penyampaian dan penerimaan informasi, *E- Learning* adalah merupakan jasa pelayanan pendidikan berbasis *website* yang diharapkan bisa sediakan fasilitas belajar yang bisa diakses tanpa dibatasi jarak serta waktu, chatbot telah banyak digunakan dalam kehidupan sehari misalnya *polling*, *game*, dan lain lain. Aplikasi berbasis *Web Smart Cooperative Oriented Problem* atau SCOP membuat chatbot yang bisa

menjawab pertanyaan secara otomatis pada mata kuliah struktur data di fakultas ilmu komputer, universitas lancing kuning, *Natural Language Processing* dan *Naive Bayes Classifier* pada *chatbot* untuk proses bimbingan belajar sehingga mahasiswa di bantu dalam memberikan jawaban berkaitan dengan pertanyaan seputar mata kuliah yang sedang di pelajari.

Natural Language Processing (NLP) adalah salah satu bidang ilmu komputer yang merupakan cabang dari kecerdasan buatan, dan bahasa (linguistik) yang berkaitan dengan interaksi antara komputer dan bahasa alami manusia, seperti bahasa Indonesia atau bahasa Inggris. Tujuan utama dari studi NLP adalah membuat mesin yang mampu mengerti dan memahami makna bahasa manusia lalu memberikan respon yang sesuai. (Alamanda, dkk 2016)

*text preprocessing* untuk data yang akan di gunakan dalam proses analisa sentimen, dimana data yang kita proses akan kita ambil informasi yang terkandung didalamnya dalam hal sentimen penulisnya yaitu negaitf atau positif. (Aris Tri Jaka H 2015)

Text Preprocessing merupakan proses persiapan data teks dokumen atau dataset. Proses ini berfungsi untuk mengubah data teks yang tidak terstruktur menjadi data yang terstruktur, terdiri dari *Case folding*, *Tokenizing*, *Normalization*, *Stopwords Removal*, *Stemming*.

Algoritma Enhanced Confix Stripping Stemmer merupakan perbaikan dari algoritma sebelumnya yaitu, algoritma Confix Stripping Stemmer. (Andita dkk, 2010)

Metode Term Frequency-Inverse Document Frequency (TF-IDF) adalah cara pemberian bobot hubungan suatu kata (term) terhadap dokumen. TF-IDF ini adalah sebuah ukuran statistik yang digunakan untuk mengevaluasi seberapa

penting sebuah kata di dalam sebuah dokumen atau dalam sekelompok kata, TF-IDF digunakan rumus untuk menghitung bobot (W) masing masing dokumen terhadap kata kunci dengan rumus yaitu:

$$W_{dt} = tf_{dt} * Id_{ft}$$

Dimana:

$W_{dt}$  = bobot dokumen ke-d terhadap kata ke-t  $tf_{dt}$  = banyaknya kata yang dicari pada sebuah dokumen

$Id_{ft}$  = Inversed Document Frequency ( $\log(N/df)$ )

N = total dokumen

Df = banyak dokumen yang mengandung kata yang dicari.

Term frequency (TF) adalah metode yang paling sederhana dalam pembobotan kata yang bergantung pada jumlah kemunculan term dalam dokumen dengan menghitung skor antara term dan dokumen berdasarkan bobot (weight) term di dalam dokumen.

$$TF(d, t) = f(d, t)$$

Keterangan :

$F(d,t)$  : frekuensi kemunculan term t pada dokumen d.

Inverse Document Frequency (IDF) adalah metode yang memperhatikan kemunculan kata di dalam dokumen dengan cara menghitung bobot kemunculan kata pada kumpulan dokumen.

$$ID(t) = \log \left( \frac{N}{df(t)} \right)$$

Keterangan:

$N_m$  : jumlah seluruh dokumen.

$df(t)$  : jumlah dokumen yang mengandung term t.

Persamaan adalah menggabungkan konsep perhitungan TF-IDF dapat dijabarkan sebagai berikut:

$$\begin{aligned} W(t, d) &= TF(d, t) \times IDF \\ &= TF(d, t) \times \log \left( \frac{N}{df(t)} \right) \end{aligned}$$

Keterangan :

$W(d,t)$  : bobot term t pada dokumen d.

TF(d,t) : total kemunculan term t pada dokumen d

Nd : total seluruh dokumen.

df(t) : total dokumen yang memiliki term t.

*Naive Bayes Classifier* juga menunjukkan akurasi dan kecepatan yang tinggi ketika diterapkan ke dalam *database* yang besar dan mengasumsikan bahwa efek dari nilai atribut pada kelas tertentu tidak bergantung pada nilai atribut lain. (Handkk, 2012)

Maximum likelihood bertujuan untuk menghitung probabilitas kemunculan setiap kata pada dokumen.

$$P(C) = \frac{Nc}{N}$$

Keterangan:

P(C) : probabilitas kemunculan kata pada dokumen.

Nc : jumlah dokumen dalam kelas c.

N : jumlah dokumen yang dilatih.

conditional probabilities sebagai frekuensi relatif dari term dalam dokumen milik kelas c menggunakan laplace smoothing, yaitu penambahan angka satu (1) agar tidak ada nilai nol (0) untuk kombinasi kelas term yang terjadi dalam data pelatihan dengan menggunakan persamaan.

$$P(tk|c) = \frac{Wct+1}{(\sum_{W' \in V} Wct') + B'}$$

Keterangan:

P(tk|c) : conditional probabilities term yang memiliki kategori.

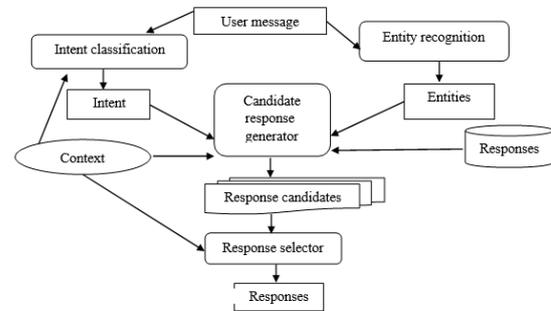
Wct : nilai pembobotan tfidf atau W dari kata t di kelas c.

$\sum_{W' \in V} Wct'$  : jumlah total W dari seluruh kata (termasuk frequency) yang berada di kelas c.

B' : jumlah W kata unik (tidak dikalikan dengan TF) di semua kelas.

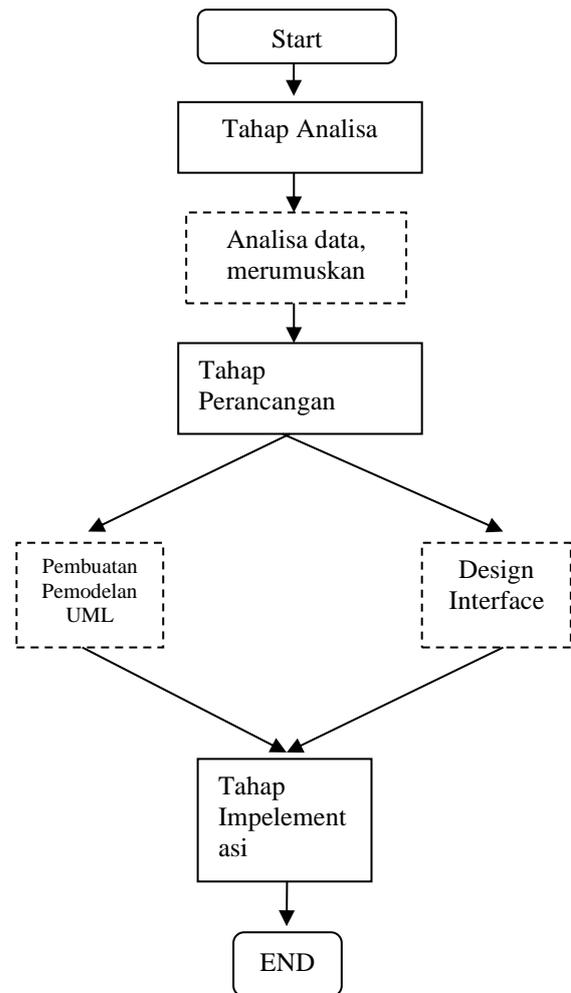
*Chatbot* adalah program komputer yang memproses bahasa alami dari pengguna dan menghasilkan respon kemudian dikirim kembali ke pengguna. (Khan & Das, 2018). *Chatbot* bekerja

pada pengenalan pola dan sebuah kumpulan algoritma Secara struktural sebuah chatbot mempunyai tiga komponen fungsional utama, *Intent* (Maksud), *Entities* (Konteks), *Responses* (Tanggapan).



Gambar 1. Cara Kerja Chatbot

## METODE



Gambar 2. Tahapan Penelitian

### 1. Tahap Analisa

Pada tahap ini yang dilakukan adalah analisa. Yaitu Membahas data-data yang diperoleh dari hasil pengolahan data sehingga lebih memperjelas pengolahan data. Pada tahapan ini bagian yang akan dianalisa yaitu sistem yang sedang berjalan dan sistem yang diusulkan.

### 2. Tahap Perancangan

Pada tahap ini yang dilakukan adalah :

#### a. Pemodelan sistem menggunakan pendekatan UML (*Unified Modelling Language*).

Pemodelan sistem baru dilakukan dengan perancangan sistem logis, dimana perancangan ini terdiri dari 5 diagram, yaitu *use case diagram*, *activity diagram*, *sequence diagram*, *collaboration diagram* dan *class diagram*.

#### b. Desain Interface

Setelah melakukan pemodelan sistem menggunakan pendekatan UML maka dibuatlah perancangan antar muka (*interface*) untuk sistem usulan.

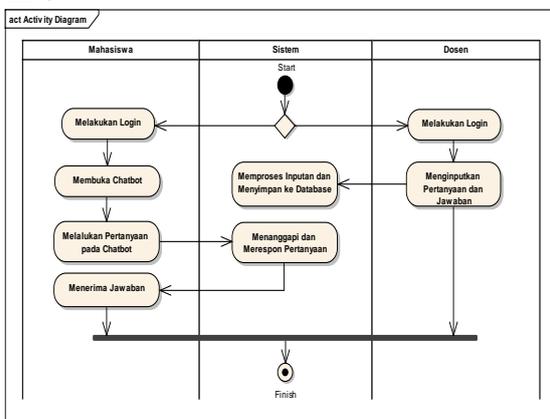
### 3. Tahap Implementasi

Pada tahap ini yang dilakukan adalah melakukan proses pengkodean dalam pengembangan sistem.

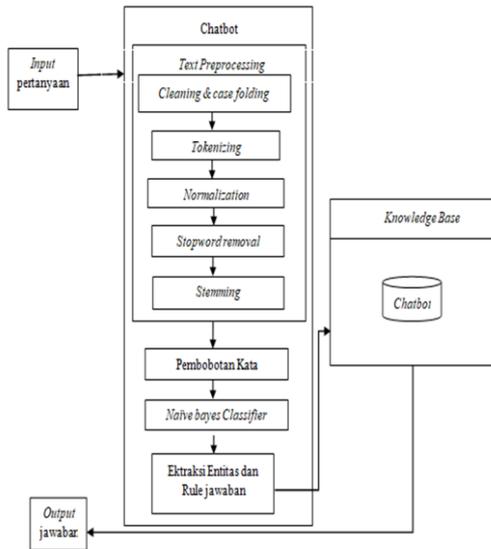
Tahapan analisa program *chatbot* dilakukan dengan menganalisa cara kerja *chatbot* dalam memberikan jawaban. Pengguna dapat berinteraksi dengan *chatbot* dengan cara memasukan pertanyaan berupa kalimat. Sebelum kategori pertanyaan diketahui, maka akan dilakukan tahapan *text preprocessing* terlebih dahulu agar pertanyaan tersebut nanti dapat diolah oleh *chatbot*. Setelah melakukan *text preprocessing*, langkah selanjutnya adalah melakukan pembobotan kata dengan menggunakan TF IDF. Kemudian data tersebut diolah dan dicari kategori pertanyaan dengan menggunakan *Naïve Bayes Classifier*. Kategori pertanyaan berguna untuk mengetahui kategori pertanyaan (maksud). Setelah kategori pertanyaan didapat, sistem memeriksa kata kunci berdasarkan informasi apa yang ingin dicari dari pertanyaan tersebut. Setelah itu mencocokkan kategori pertanyaan dengan kata kunci. Jika kategori dan kata kunci ditemukan maka akan didapat aturan (*rule*) jawaban, maka proses selanjutnya adalah mencari jawaban dari pertanyaan tersebut berdasarkan aturan (*rule*) *query database* yang telah ditentukan di dalam *database*. Setelah jawaban berdasarkan aturan *query database* ditemukan, maka *chatbot* akan menampilkan respon atau jawaban dalam layar sebagai *output*. Alur analisa program *chatbot*.

## HASIL DAN PEMBAHASAN

Proses-proses *activity diagram* dibawah ini :



Gambar 3. Activity Diagram



Gambar 4. Tahapan Kerja Chatbot

Analisa pengenalan pertanyaan berfungsi untuk melakukan analisa masukan pertanyaan yang diberikan oleh pengguna berupa kalimat berdasarkan kategori pertanyaan yang ada di chatbot. Terdapat dua metode yang digunakan untuk tahapan ini yaitu *text preprocessing* untuk mengolah teks sebelum diklasifikasi dan *Naive Bayes Classifier* digunakan untuk tahapan klasifikasi kategori pertanyaan. Sebelum melakukan tahapan klasifikasi, kata (*term*) yang telah dilakukan *text preprocessing* akan melalui proses pembobotan kata.

Tabel 1. Contoh Data Perhitungan Manual

| Kategori Pertanyaan       | Data Pertanyaan                 |
|---------------------------|---------------------------------|
| Materi Struktur Data      | Apa itu struktur data?          |
| Materi Bahasa Pemrograman | Apa itu bahasa pemrograman C++? |
| Materi Tipe Data          | Pengertian tipe data?           |

### 1. Text Preprocessing

*Text preprocessing* adalah langkah penting sebelum melakukan proses klasifikasi kategori pertanyaan. Tahapan ini bertujuan untuk membersihkan unsur-unsur yang tidak dibutuhkan dalam kalimat pertanyaan agar lebih mudah diolah oleh chatbot.

#### a. Cleaning and Case Folding

Langkah awal tahapan *text preprocessing* dalam chatbot adalah melakukan *cleaning* yang bertujuan untuk menghilangkan karakter atau simbol (@#\$%^&\*()\_+";{}<>.,?!~/[]). Kemudian dilanjutkan tahapan *case folding* yang bertujuan untuk mengubah semua huruf besar menjadi huruf kecil agar tidak ada perbedaan ukuran huruf. Hasil *cleaning and case folding* untuk contoh data dapat dilihat pada tabel berikut.

Tabel 2. Data Setelah Cleaning and Case Folding

| Kategori Pertanyaan       | Data Pertanyaan              |
|---------------------------|------------------------------|
| Materi Struktur Data      | apa itu struktur data        |
| Materi Bahasa Pemrograman | apa itu bahasa pemrograman c |
| Materi Tipe Data          | pengertian tipe data         |

#### b. Tokenizing

Tahapan *tokenizing* bertujuan untuk memecah kalimat menjadi sebuah kata atau token dengan cara memisahkan kata tersebut dan menentukan struktur sintaksis dari setiap kata. Hasil *tokenizing* untuk contoh data dapat dilihat pada table berikut.

Tabel 3. Data Setelah Tokenizing

| P1       | P 2         | P 3        |
|----------|-------------|------------|
| apa      | apa         | pengertian |
| itu      | itu         | tipe       |
| struktur | bahasa      | data       |
| data     | pemrograman | c          |

#### c. Normalization

Tahapan *normalization* bertujuan untuk menormalkan kata menjadi ejaan yang tepat. *Normalization* pada penelitian tugas akhir ini menggunakan tabel normalisasi untuk mengubah kata tidak baku menjadi baku dan *laveinsthein distance* untuk *spelling correction* (perbaiki ejaan) dengan mengambil jarak kesalahan 1. Pembuatan kamus kata baku dan tidak

baku serta kamus untuk *spelling correction* pada penelitian ini dibuat secara manual berdasarkan kebutuhan penelitian. Hasil *normalization* untuk contoh data dapat dilihat pada tabel berikut.

**Tabel 4. Data Setelah Normalization**

| P1       | P 2         | P 3        |
|----------|-------------|------------|
| apa      | apa         | pengertian |
| itu      | itu         | tipe       |
| struktur | bahasa      | data       |
| data     | pemrograman |            |
| c        |             |            |

**d. Stopword Removal**

Tahapan *stopword removal* bertujuan menghilangkan kata-kata yang tidak penting atau yang tidak diperlukan guna mengurangi dimensi ruang ketika diproses. Kata-kata tersebut adalah kata preposisi, kata tanya dan kata-kata lain yang tidak berhubungan dengan kata kunci yang penting untuk pemrosesan. Pembuatan kamus *stopword* ini menggunakan kamus *stopword* yang diambil dari <https://github.com/nolimitid/nolimit-kamus> dan disesuaikan dengan kebutuhan penelitian. Hasil *stopword removal* untuk contoh data dapat dilihat pada tabel berikut.

**Tabel 5. Data Setelah Stopword Removal**

| P1       | P 2         | P 3        |
|----------|-------------|------------|
| apa      | apa         | pengertian |
| itu      | itu         | tipe       |
| struktur | bahasa      | data       |
| data     | Pemrograman |            |
| c        |             |            |

**c. Stemming**

Tahapan *stemming* bertujuan untuk mengubah kata yang memiliki imbuhan menjadi kata dasar. Pada penelitian tugas akhir ini algoritma *stemming* yang digunakan adalah algoritma *Enhanced Confix Stripping* (ECS) dan pembuatan kamus kata dasar ini menggunakan kamus kata dasar yang diambil dari [https://github.com/nolimitid/nolimit-](https://github.com/nolimitid/nolimit-kamus)

[kamus](#) dan disesuaikan dengan kebutuhan penelitian. Proses *stemming* diawali dengan melakukan pengecekan kata ke dalam kamus kata dasar yang berfungsi untuk mengetahui apakah kata yang dimasukkan sudah berbentuk kata dasar atau belum. Jika kata yang dicari belum berbentuk kata dasar, maka proses akan menghapus imbuhan berdasarkan aturan algoritma *stemming* ECS dan kemudian mengubah kata tersebut menjadi kata dasar. Namun jika kata yang dicari sudah berbentuk kata dasar, maka proses akan mengembalikan kata awal tersebut. Hasil *stemming* untuk contoh data dapat dilihat pada tabel berikut.

**Tabel 6. Data Setelah Stemming**

| P1       | P 2         | P 3  |
|----------|-------------|------|
| struktur | bahasa      | tipe |
| data     | pemrograman | data |
| c        |             |      |

Data hasil setelah *stemming* adalah data hasil *text preprocessing*. Data ini yang akan digunakan untuk tahapan selanjutnya yaitu pembobotan kata.

**2. Pembobotan Kata**

Pembobotan kata bertujuan untuk mengubah kata menjadi sebuah vektor, di mana setiap kata dihitung sebagai satu fitur. Pada penelitian tugas akhir ini menggunakan pembobotan TF-IDF. Pembobotan ini merupakan penggabungan antara TF (*term frequency*) untuk menghitung bobot kemunculan kata dalam dokumen dan IDF untuk menghitung bobot kemunculan kata pada kumpulan dokumen. Tabel perhitungan manual untuk pembobotan kata setiap pertanyaan dapat dilihat pada tabel berikut.

**Tabel 7. Data Hasil Perhitungan Manual Pembobotan Kata Setiap Pertanyaan**

| term        | tf     |        |        | d<br>f | D df | IDF (log D df) | W       |         |         |
|-------------|--------|--------|--------|--------|------|----------------|---------|---------|---------|
|             | P<br>1 | P<br>2 | P<br>3 |        |      |                | P1      | P2      | P3      |
| bahasa      | 0      | 1      | 0      | 1      | 3    | 0,477121255    | 0       | 0,47712 | 0       |
| c           | 0      | 1      | 0      | 1      | 3    | 0,477121255    | 0       | 0,47712 | 0       |
| data        | 1      | 0      | 1      | 2      | 1,5  | 0,176091259    | 0,17609 | 0       | 0,17609 |
| pemrograman | 0      | 1      | 0      | 1      | 3    | 0,477121255    | 0       | 0,47712 | 0       |
| struktur    | 1      | 0      | 0      | 1      | 3    | 0,477121255    | 0,47712 | 0       | 0       |
| tipe        | 0      | 0      | 1      | 1      | 3    | 0,477121255    | 0       | 0       | 0,47712 |
| Jumlah      |        |        |        |        |      | 2,561697533    | 0,65321 | 1,43136 | 0,65321 |

Berikut ini penjelasan perhitungan manual TF untuk kata struktur menggunakan Persamaan 2.1.

$$TF(P1, \text{struktur}) = 1, TF(P2, \text{struktur}) = 0, TF(P3, \text{struktur}) = 0$$

Berikut ini penjelasan perhitungan manual IDF untuk kata ahli menggunakan Persamaan 2.2.

$$IDF(\text{struktur}) = \log(3) = 0,477121255$$

Berikut ini penjelasan perhitungan manual bobot (w) untuk kata ahli menggunakan Persamaan 2.3 :

$$W(P1) = TF(P1, \text{struktur}) \times IDF \Rightarrow 1 \times 0,477121255 = 0,477121255$$

$$W(P2) = TF(P2, \text{struktur}) \times IDF \Rightarrow 0 \times 0,477121255 = 0$$

$$W(P3) = TF(P3, \text{struktur}) \times IDF \Rightarrow 0 \times 0,477121255 = 0$$

Berikut ini penjelasan perhitungan manual jumlah Bobot (W) kata untuk setiap kategori :

$$\text{Jumlah } W(P1) = 0,65321$$

$$\text{Jumlah } W(P2) = 1,43136$$

$$\text{Jumlah } W(P3) = 0,65321$$

$$\text{Jumlah IDF} = 2,561697533$$

### 3. Naïve Bayes Classifier

Tahapan *naïve bayes classifier* dilakukan setelah text preprocessing, bertujuan untuk menentukan kategori pertanyaan. Tahapan ini memiliki dua bagian, yaitu training dan testing.

#### a. Training

Training dilakukan untuk mengetahui model klasifikasi yang akan menjadi acuan untuk testing. Langkah pertama dalam tahapan training adalah menghitung nilai prior probabilities P(c) dengan menggunakan maximum

likelihood dan langkah selanjutnya adalah menghitung nilai conditional probabilities dengan menggunakan laplace smoothing.

i. Menghitung Nilai prior probabilities P(c) dengan menggunakan Maximum Likelihood

Tahapan ini menghitung prior probabilities P(c) atau probabilitas kemunculan kata pada dokumen dari setiap kategori menggunakan Persamaan 2.4.

$$P(\text{Materi Struktur Data}) = 1/6 = 0,167$$

$$P(\text{Materi Bahasa Pemrograman}) = 1/6 = 0,167$$

$$P(\text{Materi Tipe Data}) = 1/6 = 0,167$$

ii. Menghitung conditional probabilities dengan Laplace Smoothing (LS) Tahapan Laplace Smoothing bertujuan untuk menghilangkan nilai 0 (nol) pada tahapan akhir proses training. Hasil yang didapat akan menjadi model untuk melakukan klasifikasi pada tahapan testing. Perhitungan manual untuk tahap training dapat dilihat pada tabel berikut.

**Tabel 8. Perhitungan Manual untuk Tahap Training**

| LS          |                      |                           |                  |
|-------------|----------------------|---------------------------|------------------|
| Term        | Materi Struktur Data | Materi Bahasa Pemrograman | Materi Tipe Data |
| bahasa      | 0,311050694          | 0,500868845               | 0,311050694      |
| c           | 0,311050694          | 0,500868845               | 0,311050694      |
| data        | 0,622101387          | 0,250434423               | 0,622101387      |
| pemrograman | 0,311050694          | 0,500868845               | 0,311050694      |
| struktur    | 0,622101387          | 0,250434423               | 0,311050694      |
| tipe        | 0,311050694          | 0,250434423               | 0,622101387      |

Berikut ini penjelasan perhitungan manual *conditional probabilities* kata ahli dengan menggunakan *laplace*

smoothing untuk setiap kategori menggunakan persamaan dibawah

$$P(\text{struktur} | \text{Materi Struktur Data}) = \frac{0,65321+2,5616976}{0+1}$$

$$P(\text{struktur} | \text{Materi Bahasa Pemrograman}) = \frac{1,43136+2,5616976}{0+1}$$

$$P(\text{struktur} | \text{Materi Tipe Data}) = \frac{0,65321+2,5616976}{0+1}$$

**b. Testing**

Pada tahapan testing, data yang sudah dilatih (training) akan diuji untuk klasifikasi. Contoh data testing untuk perhitungan manual dapat dilihat pada tabel berikut.

**Tabel 9. Data Testing**

| Jenis Pertanyaan | Isi Pertanyaan                          |
|------------------|---|
| ?                | Sebutkan pengertian dari struktur data? |

Hasil *Text Preprocessing* untuk data testing dapat dilihat pada tabel berikut

**Tabel 10. Hasil Text Processing Data Testing**

| Testing    |
|------------|
| pengertian |
| struktur   |
| data       |

Setelah hasil data *text preprocessing* didapat, selanjutnya melakukan langkah *matching* dan menghitung probabilitas.

**i. Matching**

Tahapan *matching* untuk mencari kata (*term*) yang terdapat kesamaan antara data *training* dan *testing*. Hasil *matching* dapat dilihat pada tabel berikut.

**Tabel 11. Hasil Matching Data Training dan Data Testing**

| Testing  |
|----------|
| struktur |
| data     |

**ii. Menghitung Probabilitas**

Tahapan ini untuk menentukan kategori pertanyaan data *testing*.

Perhitungan manual probabilitas *testing* dapat dilihat pada tabel berikut.

**Tabel 12. Data Perhitungan Manual Probabilitas Testing**

| Term     | TF Testing | LS                   |                           |                  |
|----------|------------|----------------------|---------------------------|------------------|
|          |            | Materi Struktur Data | Materi Bahasa Pemrograman | Materi Tipe Data |
| data     | 1          | 0,622101387          | 0,250434423               | 0,622101387      |
| struktur | 1          | 0,622101387          | 0,250434423               | 0,311050694      |

Setelah melakukan perhitungan probabilitas *testing* berdasarkan nilai *laplace smoothing* (LS), selanjutnya memangkatkan nilai LS dengan nilai TF. Perhitungan manual pemangkatan nilai LS dan TF dapat dilihat pada tabel berikut.

**Tabel 13. Data Perhitungan Manual Probabilitas Testing**

| Term                        | TF Testing | LS                   |                           |                  |
|-----------------------------|------------|----------------------|---------------------------|------------------|
|                             |            | Materi Struktur Data | Materi Bahasa Pemrograman | Materi Tipe Data |
| data                        | 1          | 0,622101387          | 0,250434423               | 0,622101387      |
| struktur                    | 1          | 0,622101387          | 0,250434423               | 0,311050694      |
| Hasil perkalian             |            | 0,387010136          | 0,0627174                 | 0,193505068      |
| Perkalian Prior Probability |            | 0,064630693          | 0,010473806               | 0,032315346      |
| Nilai Maksimal              |            | 0,064630693          |                           |                  |

Berikut ini penjelasan perhitungan manual probabilitas kata “bahas” dalam data *testing* :

- a. Memangkatkan nilai *laplace smoothing* dari setiap kategori dengan nilai TF, di tabel kata bahas memiliki nilai TF = 1.  
 $P(\text{struktur} | \text{Materi Struktur Data})$   
 $0,622101387^1 = 0,622101387$   
 $P(\text{struktur} | \text{Materi Bahasa Pemrograman})$   
 $0,250434423^1 = 0,250434423$   
 $P(\text{struktur} | \text{Materi Tipe Data})$   
 $0,311050694^1 = 0,311050694$
- b. Mengalikan nilai *prior probabilities* dengan total perkalian pangkat TF dengan nilai *laplace smoothing* untuk mendapatkan nilai probabilitas setiap kelas.

Probabilitas setiap kelas terhadap kelas *testing*:

$$P(\text{Materi Struktur Data}|\text{Testing}) = 0,167 * 0,387010136 = 0,064630693$$

$$P(\text{Materi Bahasa Pemrograman}|\text{Testing}) = 0,167 * 0,0627174 = 0,010473806$$

$$P(\text{Materi Tipe Data}|\text{Testing}) = 0,167 * 0,193505068 = 0,032315346$$

Berdasarkan hasil perhitungan manual probabilitas, diketahui bahwa probabilitas kelas yang paling tinggi dengan menggunakan persamaan 2.6 terdapat pada kategori deskripsi mata kuliah, sehingga pertanyaan tersebut termasuk ke dalam kategori materi struktur data (hasil uji benar).

### Rancangan Antar Muka

Rancangan antar muka ini akan menggambarkan bagaimana tampilan yang ada didalam sistem. Letak navigasi menu yang akan ditampilkan.

#### a. Halaman Depan

Antar muka Main Menu merupakan antar muka yang akan tampil disetiap halaman ketika membuka sistem *e-Smart Cooperative Oriented Problem*.



Gambar 5. Halaman Depan

#### b. Halaman Login

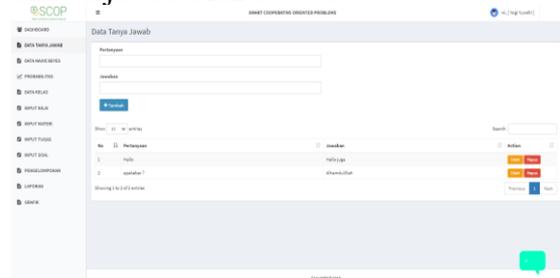
Antar muka *Login* adalah halaman yang pertama kali muncul saat *website* dibuka yang digunakan untuk verifikasi *user* dengan memasukkan *username* dan *password* yang benar.



Gambar 6. Halaman Login

#### c. Antar Muka Menu Utama Dosen

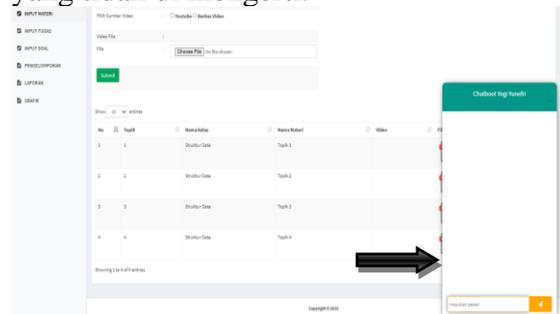
Antar muka menu utama dosen merupakan tampilan yang muncul setelah dosen melakukan login dimana dosen dapat mengolah data pertanyaan dari mahasiswa dan jawaban yang belum bisa terjawab oleh *chatbot*.



Gambar 7. Halaman Utama Dosen

#### d. Halaman chatbot

Antar muka menu *chatbot* merupakan tampilan yang muncul setelah mahasiswa mengklik menu *chat*, sehingga mahasiswa dapat menggunakannya sebagai alat bantu tanya apa bila ada materi perkuliahan yang tidak di mengerti.



Gambar 8. Halaman chatbot

## SIMPULAN

### Kesimpulan

Berdasarkan pembahasan, maka dapat ditarik kesimpulan yaitu melalui

perancangan sistem chatbot ini, pihak-pihak terkait dapat merealisasikan kedalam sistem Chatbot, sehingga mahasiswa dapat melakukan perkuliahan dengan fitur yang baru, Masalah tentang cara menjawab pertanyaan dari mahasiswa yang masih konvensional, memerlukan waktu yang cukup lama untuk mengetahui jawaban dari dosen, Harapannya masalah-masalah tersebut dapat terselesaikan serta proses perkuliahan dapat dilaksanakan secara efektif, efisien, cepat dan transparan.

#### DAFTAR PUSTAKA

- Alamanda, R., Suhery, C., Brianorman, Y., & Komputer, J. S. (2016). Jurnal Coding , Sistem Komputer Untan APLIKASI PENDETEKSI PLAGIAT TERHADAP KARYA TULIS BERBASIS WEB MENGGUNAKAN NATURAL LANGUAGE PROCESSING DAN Jurnal Coding , Sistem Komputer Untan ISSN: 2338-493x. *Jurnal Coding, Sistem Komputer Untan*, 04(1).
- Amrizal, V. (2018). Penerapan Metode Term Frequency Inverse Document Frequency (Tf-Idf) Dan Cosine Similarity Pada Sistem Temu Kembali Informasi Untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus: Hadits Shahih Bukhari-Muslim). *Jurnal Teknik Informatika*, 11(2), 149–164. <https://doi.org/10.15408/jti.v11i2.8623>
- Andita Dwiyoga Tahitoe, D. P. (2010). Implementasi Modifikasi Enhanced Confix Stripping Stemmer Untuk Bahasa Indonesia Dengan Metode Corpus Based Stemming. *Jurnal Ilmiah*, 1–15.
- Feng, X., Li, S., Yuan, C., Zeng, P., & Sun, Y. (2018). Prediction of Slope Stability using Naive Bayes Classifier. *KSCE Journal of Civil Engineering*, 22(3), 941–950. <https://doi.org/10.1007/s12205-018-1337-3>
- H, A. T. J. (2015). Preprocessing Text untuk Meminimalisir Kata yang Tidak Berarti dalam Proses Text Mining. *Informatika UPGRIS*, 1, 1–9.
- Hengki, M., & Wahyudi, M. (2020). Klasifikasi Algoritma Naïve Bayes dan SVM Berbasis PSO Dalam Memprediksi Spam Email Pada Hotline-Sapto. *Paradigma - Jurnal Komputer Dan Informatika*, 22(1), 61–67. <https://doi.org/10.31294/p.v22i1.7842>
- Kang, H., Yoo, S. J., & Han, D. (2012). Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews. *Expert Systems with Applications*, 39(5), 6000–6010. <https://doi.org/10.1016/j.eswa.2011.11.107>
- Nwankwo, W. (2018). Interactive Advising with Bots: Improving Academic Excellence in Educational Establishments. *American Journal of Operations Management and Information Systems*, 3(1), 6. <https://doi.org/10.11648/j.ajomis.20180301.12>
- Rusmarasy, B., Priyambadha, B., & Pradana, F. (2019). Pengembangan Chat Bot pada CoMa untuk Memberikan Motivasi Kepada Pengguna Menggunakan AIML. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(5), 4484–4490.
- Smith, D. (2006). *Informa . on Retrieval*.
- Utari, M. I., & Medyawati, H. (2020). Classification of News Types By Implementing Enhanced Confix Stripping Stemmer. *International Journal of Engineering*

*Technologies and Management  
Research*, 6(5), 135–141.  
[https://doi.org/10.29121/ijetmr.v6.i  
5.2019.380](https://doi.org/10.29121/ijetmr.v6.i5.2019.380)