



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 II

LANDASAN TEORI

Dalam melakukan perancangan dan pembangunan *chatbot* menggunakan algoritma Bigram, ada beberapa telaah literatur yang berhubungan, yaitu *chatbot*, *text pre-processing*, algoritma Bigram, *Technology Acceptance Model (TAM)*, skala Likert, dan algoritma Jaro-Winkler *Distance*.

2.1 Chatbot

Chatbot merupakan sebuah program komputer yang dapat menjalankan *intelligent conversation* dengan pengguna melalui media suara ataupun teks, sering kali dilakukan dengan percakapan yang singkat (Tillotson, 2012). *Chatbot* bekerja dengan mengartikan pesan yang diberikan oleh pengguna, kemudian memproses maksud dari pesan tersebut, selanjutnya menentukan dan mengeksekusi apa yang perlu *chatbot* lakukan berdasarkan perintah dari pengguna, dan terakhir menyampaikan hasil dari eksekusi program kepada pengguna (Shawar, 2007).

Penelitian ini menggunakan platform LINE untuk menjalankan *chatbot*, dengan cara membuat *Official Account (OA)* dari LINE lalu menghubungkannya dengan program *chatbot* yang dirancang dan dibangun.

2.2 Text Pre-Processing

Text pre-processing merupakan proses untuk mempersiapkan data mentah sebelum dilakukan proses lain (Mujlahwati, 2016). Berikut ini merupakan tahap-tahap dalam *text pre-processing*.

2.2.1 Tokenisasi

Tokenisasi merupakan proses pemisahan setiap kata (*term*) yang terkandung dalam sebuah dokumen (Pradnyana, 2013). Pada tahap ini juga dilakukan *case folding* yaitu proses penyamaan bentuk (*case*) suatu kata, biasanya menjadi huruf kecil. Setelah itu, dilakukan proses penghilangan tanda baca atau simbol-simbol yang tidak diperlukan.

Tabel 2.1 Contoh Tokenisasi

Input	Hasil Tokenisasi
SELAMAT SIANG	array("selamat","siang")
Bagaimana jika KTM hilang?	array("bagaimana","jika","ktm","hilang")
Apa bedanya KHS dan KRS?	array("apa","bedanya","khs","dan","krs")
Apakah ada IPK minimal, untuk pindah prodi?	array("apakah","ada","ipk","minimal","untuk","pindah","prodi")
Terima Kasih	array("terima","kasih")

2.2.2 Stopwords removal

Stopwords removal merupakan proses penghilangan kata-kata tidak penting (*stopwords*) dari suatu dokumen. Proses penghilangan dilakukan dengan pencocokan kata-kata yang ada dalam dokumen terhadap daftar kata-kata tidak penting (*stoplist*) (Pradnyana, 2013). Stopwords yang digunakan diambil dari Epadi (2018). Beberapa contoh kata yang terdapat dalam stopwords, yaitu 'dapat', 'yang', 'dan', 'agak', 'hanya', 'karena', 'oleh', 'untuk', 'dari', 'sudah'.

Tabel 2.2 Contoh Stopwords Removal

Input	Hasil Stopwords Removal
array("selamat","siang")	array("selamat","siang")
array("bagaimana","jika","ktm","hilang")	array("ktm","hilang")
array("apa","bedanya","khs","dan","krs")	array("bedanya","khs","krs")

Tabel 2.2 Contoh Stopwords Removal (Lanjutan)

Input	Hasil Stopwords Removal
array("apakah","ada","ipk","minimal","untuk","pindah","prodi")	array("ipk","minimal","pindah","prodi")
array("terima","kasih")	array("terima","kasih")

2.2.3 Stemming

Stemming merupakan suatu proses yang terdapat dalam sistem *Information Retrieval* (IR) yang mentransformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata akarnya (*root word*) dengan menggunakan aturan-aturan tertentu (Pramudita, 2014). Algoritma *stemming* untuk beberapa bahasa telah dikembangkan, seperti algoritma Porter untuk teks berbahasa Inggris, algoritma Porter untuk teks berbahasa Indonesia, algoritma Nazief & Andriani untuk teks berbahasa Indonesia (Andriani dkk, 2007).

Tahap-tahap dari algoritma Nazief & Andriani adalah sebagai berikut.

1. Cari kata yang akan diistem dalam kamus kata dasar. Jika ditemukan maka diasumsikan kata adalah *root word*. Maka algoritma berhenti.
2. *Inflection Suffixes* ("-lah", "-kah", "-ku", "-mu", atau "-nya") dibuang. Jika berupa *particles* ("-lah", "-kah", "-tah" atau "-pun") maka langkah ini diulangi lagi untuk menghapus *Possesive Pronouns* ("-ku", "-mu", atau "-nya"), jika ada.
3. Hapus *Derivation Suffixes* ("-i", "-an" atau "-kan"). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a
 - a. Jika "-an" telah dihapus dan huruf terakhir dari kata tersebut adalah "-", maka "-k" juga ikut dihapus. Jika kata tersebut ditemukan dalam

- kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
- b. Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan, lanjut ke langkah 4.
4. Hapus *Derivation Prefix*. Jika pada langkah 3 ada sufiks yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b.
 - a. Periksa tabel kombinasi awalan-akhiran yang tidak diijinkan. Jika ditemukan maka algoritma berhenti, jika tidak pergi ke langkah 4b.
 - b. For $i = 1$ to 3, tentukan tipe awalan kemudian hapus awalan. Jika root word belum juga ditemukan lakukan langkah 5, jika sudah maka algoritma berhenti. Catatan: jika awalan kedua sama dengan awalan pertama algoritma berhenti.
 5. Melakukan recoding, yakni penyusunan kembali kata-kata yang mengalami proses *stemming* berlebih.
 6. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai *root word*. Proses selesai.

Tabel 2.3 Contoh Stemming

Input	Hasil Stemming
array(“selamat”, “siang”)	array(“selamat”, “siang”)
array(“ktm”, “hilang”)	array(“ktm”, “hilang”)
array(“bedanya”, “khs”, “krs”)	array(“beda”, “khs”, “krs”)
array(“ipk”, “minimal”, “pindah”, “prodi”)	array(“ipk”, “minimal”, “pindah”, “prodi”)
array(“terima”, “kasih”)	array(“terima”, “kasih”)

2.3 Algoritma Bigram

N-gram adalah potongan n karakter dalam suatu string tertentu atau potongan kata dalam suatu kalimat tertentu (Permadi, 2008). Teknik n-gram didasarkan pada pemisahan teks menjadi string dengan panjang n mulai dari posisi tertentu dalam suatu teks. Posisi n-gram berikutnya dihitung dari posisi yang sebenarnya bergeser sesuai dengan *offset* yang diberikan. Nilai *offset* bergantung pada pembagian yang digunakan dalam n-gram. N-gram dapat berupa unigram (n=1), bigram (n=2), trigram (n=3), dan seterusnya (Lisangan, 2013).

Di dalam bigram terdapat *sentence similarity measurement* atau kesamaan semantik sebagai pengukur tingkat kemiripan suatu teks. Kesamaan semantik memberikan skor untuk hubungan semantik antara dua kalimat atau *string* (Setiaji & Wibowo, 2016). Jika ada dua kalimat atau lebih maka dapat dihitung tingkat kesamaannya. Semakin tinggi skor kesamaannya, maka semakin tinggi pula tingkat kesamaan kalimat tersebut. Skor untuk kesamaan semantik ini bernilai dari 0 sampai 1. Rumus untuk menghitung kesamaan semantik ditunjukkan seperti pada Persamaan 2.1 berikut.

$$\text{Similarity} = \frac{\text{count}(S_1 \cap S_2) + \text{count}(S_2 \cap S_1)}{\text{count}(S_1) + \text{count}(S_2)} \quad \dots(2.1)$$

Metode untuk menghitung kesamaan semantik antara dua kalimat dapat ditulis sebagai berikut (Setiaji & Wibowo, 2016).

- a. Setiap kalimat dibagi menjadi deretan token
- b. Kalimat yang telah dibagi dapat menerapkan bigram set
- c. Setelah dibagi-bagi dan menerapkan bigram set, maka akan dihitung dan dimasukkan ke dalam Persamaan 2.1.

Contoh untuk metode tersebut yaitu sebagai berikut.

$s_1 = \text{mahasiswa}$

$s_2 = \text{masihsiswaa}$

s_1 dan s_2 dapat dibagi menjadi bigram set sebagai berikut.

$s_1 = \{\text{"ma"}, \text{"ah"}, \text{"ha"}, \text{"as"}, \text{"si"}, \text{"is"}, \text{"sw"}, \text{"wa"}\} = 8$

$s_2 = \{\text{"ma"}, \text{"as"}, \text{"si"}, \text{"ih"}, \text{"hs"}, \text{"si"}, \text{"is"}, \text{"sw"}, \text{"wa"}, \text{"aa"}\} = 10$

$s_1 \in s_2 = \{\text{"ma"}, \text{"as"}, \text{"si"}, \text{"is"}, \text{"sw"}, \text{"wa"}\} = 6$

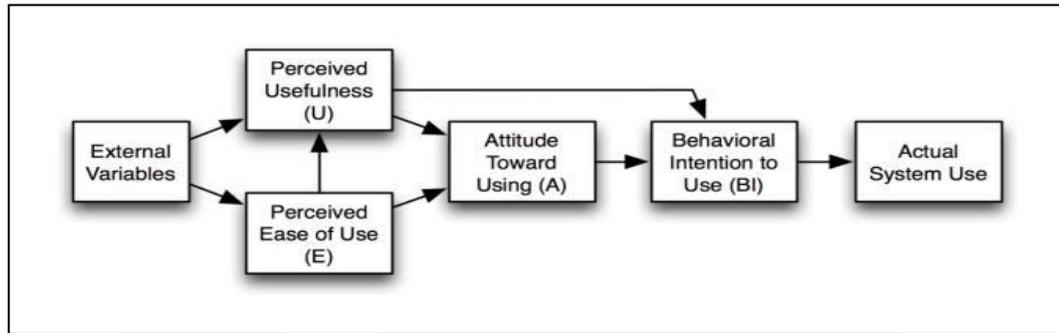
$s_2 \in s_1 = \{\text{"ma"}, \text{"as"}, \text{"si"}, \text{"is"}, \text{"sw"}, \text{"wa"}\} = 6$

Untuk mendapatkan skor kesamaan semantik, maka hasil tersebut dimasukkan ke dalam Persamaan 2.1 dan hasilnya sebagai berikut.

$$\text{Similarity} = \frac{6 + 6}{8 + 10} = \frac{12}{18} = 0.6666$$

2.4 Technology Acceptance Model (TAM)

Salah satu metode yang dapat digunakan untuk mengukur penerimaan dan penggunaan teknologi adalah metode *Technology Acceptance Model* (TAM). TAM merupakan suatu model hasil pengembangan dari *Theory of Reasoned Action* (TRA) oleh Davis (1986) yang mempelajari perilaku penerimaan seorang individu dalam menerima suatu sistem informasi. Menurut Davis (1986), tujuan utama TAM adalah untuk memberikan dasar untuk penelusuran pengaruh faktor eksternal terhadap kepercayaan, sikap, dan tujuan pengguna. TAM menganggap bahwa 2 keyakinan individual, yaitu persepsi manfaat (*perceived usefulness*, disingkat PU) dan persepsi kemudahan penggunaan (*perceived ease of use*, disingkat PEOU), adalah pengaruh utama untuk perilaku penerimaan komputer. Secara skematik teori TAM tergambar dalam Gambar 2.1.



Gambar 2.1 Technology Acceptance Model (Davis, dkk. 1989)

Persepsi kegunaan (*perceived usefulness*) menggambarkan tingkat kepercayaan seseorang bahwa penggunaan sistem akan meningkatkan kinerjanya. Orang menggunakan teknologi informasi karena mempunyai keyakinan bahwa prestasi dan kinerja akan meningkat. Konsep ini menggambarkan ukuran di mana penggunaan suatu teknologi dipercaya akan mendatangkan manfaat bagi pemakainya (Rahayu, 2017).

Kemudahan penggunaan yang dirasakan (*perceived ease of use*) menggambarkan tingkat kepercayaan seseorang bahwa penggunaan sistem informasi merupakan hal yang mudah dan tidak memerlukan usaha keras dari pemakainya. Kemudahan ini akan mengurangi tenaga, pikiran dan waktu yang digunakan untuk mempelajari dan menggunakan sistem informasi. Orang yang bekerja dengan sistem informasi, bekerja lebih mudah dibandingkan orang yang bekerja secara manual tanpa sistem informasi (Rahayu, 2017).

Sikap terhadap penggunaan (*attitude towards use*) merupakan kecenderungan tanggapan awal atas kondisi yang menyenangkan maupun tidak menyenangkan pada suatu objek tertentu. Secara teoritis, sikap merupakan refleksi perasaan seseorang atas objek dalam kondisi baik atau buruk, menguntungkan atau merugikan. Sikap muncul karena seseorang memiliki nilai yang ditentukan oleh kepercayaan atas objek tersebut. Pada kondisi lain, perilaku tertentu juga dapat

mempengaruhi kepercayaan baru seseorang sehingga membawa perubahan pada sikap (Rahayu, 2017).

Perilaku (*behavior*) dilakukan karena individu mempunyai niat atau keinginan untuk melakukan atau niat berperilaku akan menentukan perilakunya. Niat berperilaku (*behavioral intention*) adalah suatu keinginan seseorang untuk melakukan suatu perilaku tertentu atau kecenderungan seseorang untuk tetap menggunakan teknologi tertentu. Tingkat penggunaan teknologi seseorang dapat diprediksi dari sikap perhatiannya terhadap teknologi tersebut, misal motivasinya untuk tetap menggunakan maupun memotivasi penggunaan lain atau menambah perangkat pendukung (Rahayu, 2017).

Perilaku adalah penggunaan sesungguhnya (*actual usage*) teknologi itu sendiri atau kondisi nyata penggunaan sistem informasi. Perilaku atau penggunaan sesungguhnya sulit diobservasi dan diukur melalui daftar pertanyaan. Hasil penelitian TAM, menunjukkan bahwa penggunaan sistem informasi dapat diprediksi dengan baik dengan menggunakan variabel niat berperilaku (*behavioral intention*) (Venkatesh, 2000).

2.4.1 Contoh Pertanyaan

Pada penelitian sebelumnya, Rahayu (2017) menggunakan TAM untuk menganalisis tingkat penerimaan terhadap e-learning pada Universitas Atma Jaya Yogyakarta. Selain itu, Ratri (2016) juga menggunakan TAM untuk menganalisis penggunaan e-learning oleh guru SMK Negeri 2 Yogyakarta.

Gambar 2.2 menunjukkan contoh pertanyaan untuk variabel *perceived ease of use* yang bertujuan untuk mengetahui apakah sistem mudah digunakan.

1. Perceived Ease of Use of E-Learning System
Berikut ini sejumlah pernyataan tentang persepsi terhadap kemudahan pengguna sistem informasi akademik (Situs Kuliah) UAJY.

No	Pernyataan	Alternatif Jawaban				
		SS	S	N	TS	STS
1	Menurut saya penggunaan situs kuliah mudah untuk dipelajari					
2	Menurut saya mendapatkan informasi yang saya butuhkan dari situs kuliah merupakan hal yang mudah					
3	Menurut saya situs kuliah menggunakan bahasa yang mudah dimengerti dan dipahami					
4	Menu – menu pada situs kuliah tersusun dengan baik sehingga fitur – fitur yang tersedia dapat mudah digunakan					
5	Menurut saya situs kuliah sangat fleksibel untuk berinteraksi					
6	Menurut saya langkah-langkah dalam menggunakan situs kuliah mudah diingat.					

Gambar 2.2 Contoh Pertanyaan Perceived Ease of Use (Rahayu, 2017)

Gambar 2.3 menunjukkan contoh pertanyaan untuk variabel *perceived usefulness* yang bertujuan untuk mengetahui apakah sistem bermanfaat untuk pengguna.

2. Perceived Usefulness of E-Learning System
Berikut ini sejumlah pernyataan tentang persepsi terhadap kebermanfaatan penggunaan sistem informasi akademik (Situs Kuliah) UAJY.

No	Pernyataan	Alternatif Jawaban				
		SS	S	N	TS	STS
1	Menggunakan situs kuliah membantu saya menjadi lebih cepat dalam menemukan materi perkuliahan					
2	Menggunakan situs kuliah membantu saya untuk meningkatkan kinerja belajar saya					
3	Menggunakan situs kuliah meningkatkan produktivitas saya dalam perkuliahan					
4	Menggunakan situs kuliah meningkatkan keefektifan belajar saya					
5	Menggunakan situs kuliah membuat saya lebih mudah untuk mengerjakan tugas-tugas perkuliahan					
6	Secara keseluruhan penggunaan situs kuliah sangat bermanfaat bagi saya					

Gambar 2.3 Contoh Pertanyaan Perceived Usefulness (Rahayu, 2017)

Gambar 2.4 menunjukkan contoh pertanyaan untuk variabel *attitude toward use* yang bertujuan untuk mengetahui apakah sistem menyenangkan bagi pengguna.

3. Attitude Toward Using of E-Learning System
Berikut ini sejumlah pernyataan tentang sikap terhadap penggunaan sistem informasi akademik (Situs Kuliah) UAJY.

No	Pernyataan	Alternatif Jawaban				
		SS	S	N	TS	STS
1	Saya merasa senang menggunakan situs kuliah					
2	Saya merasa nyaman dan menikmati saat menggunakan situs kuliah					
3	Saya tidak suka menggunakan situs kuliah					
4	Saya merasa bosan menggunakan situs kuliah					

Gambar 2.4 Contoh Pertanyaan Attitude Toward Use (Rahayu, 2017)

Gambar 2.5 menunjukkan contoh pertanyaan untuk variabel *behavioral intention* yang bertujuan untuk mengetahui apakah pengguna tetap berniat menggunakan suatu sistem.

4. Behavioral Intention of E-Learning System
Berikut ini sejumlah pernyataan tentang minat berperilaku terhadap penggunaan sistem informasi akademik (Situs Kuliah) UAJY.

No	Pernyataan	Alternatif Jawaban				
		SS	S	N	TS	STS
1	Saya ingin menggunakan situs kuliah saat perkuliahan maupun dalam mengerjakan tugas.					
2	Saya ingin selalu menggunakan situs kuliah kapanpun					
3	Saya berniat ingin terus menggunakan situs kuliah untuk mendukung proses belajar saya					
4	Saya berharap untuk terus menggunakan situs kuliah dalam perkuliahan saya					
5	Saya ingin memotivasi teman saya untuk menggunakan situs kuliah					

Gambar 2.5 Contoh Pertanyaan Behavioral Intention (Rahayu, 2017)

Gambar 2.6 menunjukkan contoh pertanyaan untuk variabel *actual usage* yang bertujuan untuk mengetahui apakah pengguna merasa puas menggunakan suatu sistem.

7. Penggunaan Sesungguhnya (*Actual Technology Use*)

Berikut ini adalah sejumlah pernyataan tentang penggunaan *e-learning* PINTER yang sesungguhnya oleh guru.

No	Pernyataan	Alternatif Jawaban			
		SS	S	TS	STS
26	Saya mengakses <i>e-learning</i> PINTER selama hari aktif kegiatan pembelajaran maupun hari libur.				
27	Setiap pembelajaran di kelas, saya menyempatkan untuk menggunakan <i>e-learning</i> PINTER.				
28	Saya mengakses <i>e-learning</i> PINTER hampir setiap hari.				
29	Saya mengakses <i>e-learning</i> PINTER minimal rata-rata 10 menit setiap kali mengunjungnya.				
30	Secara keseluruhan saya puas dengan kinerja <i>e-learning</i> PINTER.				
31	Saya menyampaikan kepuasan terhadap kinerja <i>e-learning</i> PINTER kepada sesama rekan guru.				

Gambar 2.6 Contoh Pertanyaan Actual Usage (Ratri, 2016)

2.5 Skala Likert

Penelitian ini menggunakan metode penelitian kuantitatif dengan skala Likert. Menurut Sugiyono (2014), skala likert yaitu skala yang digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial.

Biasanya skala Likert menggunakan lima skala, namun pada penelitian ini hanya menggunakan empat skala dengan interval penilaian sebagai berikut.

1. Sangat Tidak Setuju = 0% – 24,99%
2. Tidak Setuju = 25% – 49,99%
3. Setuju = 50% – 74,99%
4. Sangat Setuju = 75% – 100%

Menurut Hadi (1991), modifikasi terhadap skala Likert dimaksudkan untuk menghilangkan kelemahan yang terkandung oleh skala lima tingkat, dengan alasan sebagai berikut.

1. Kategori *undeciden* itu mempunyai arti ganda, bisa diartikan belum dapat memutuskan atau memberi jawaban, bisa juga diartikan netral, setuju tidak, tidak setuju pun tidak, atau bahkan ragu-ragu. Kategori jawaban ganda arti atau *multi interpretable* ini tentu saja tidak diharapkan dalam suatu instrumen
2. Tersedianya jawaban yang ditengah itu menimbulkan jawaban ke tengah atau *central tendency effect*, terutama bagi mereka yang ragu-ragu atas arah kecenderungan pendapat responden, ke arah setuju atau ke arah tidak setuju. Jika disediakan kategori jawaban itu akan menghilangkan banyak data penelitian sehingga mengurangi banyaknya informasi yang dapat dijangkau para responden.

2.6 Algoritma Jaro-Winkler Distance

Jaro-Winkler *Distance* adalah algoritma untuk menghitung nilai jarak kedekatan antara dua teks. Menurut Winkler (1999), Jaro-Winkler *Distance* mempunyai 3 komponen dasar: (1) menghitung panjang string atau kata, (2) mencari nomor huruf pada kedua kata, (3) mencari transposisi. Semakin tinggi nilai Jaro-Winkler *Distance* antara dua teks berarti semakin ada kemiripan.

Pada algoritma Jaro-Winkler, jarak antara dua kata s_1 dan s_2 dihitung menggunakan rumus pada Persamaan 2.2.

$$Jaro(s_1, s_2) = \frac{1}{3} \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m - t}{m} \right) \dots(2.2)$$

Di mana s_1 dan s_2 adalah dua teks atau kata yang dihitung jarak kedekatannya, m adalah jumlah karakter yang cocok antara s_1 dengan s_2 , t adalah jumlah transposisi. Jumlah karakter pencocokan yang urutannya berbeda dibagi dua

menjadi nilai transposisi. Nilai Jaro yang tinggi menunjukkan kemiripan yang lebih besar antara kedua kata. Nilai Jaro 0 menunjukkan bahwa kata tidak sama dan nilai 1 menunjukkan bahwa keduanya sama (Agarwal, 2013).

Jaro-Winkler *Distance* menggunakan *prefix scale* (p) yang menentukan seberapa besar pengaruh *prefix length* (L) terhadap kesamaan dari *string* yang dibandingkan, dan *prefix length* (L) yang menyatakan panjang awalan yaitu panjang karakter yang sama dari *string* yang dibandingkan sampai ditemukannya ketidaksamaan. Bila *string* s_1 dan s_2 yang diperbandingkan, maka Jaro-Winkler *distance*-nya (dw) seperti pada Persamaan 2.3.

$$dw(s_1, s_2) = Jaro(s_1, s_2) + (L * p * (1 - Jaro(s_1, s_2))) \quad \dots(2.3)$$

Di mana Jaro merupakan Jaro *Distance* untuk *string* s_1 dan s_2 , t merupakan panjang prefiks umum di awal *string* yang nilai maksimalnya adalah 4 karakter, sedangkan p merupakan konstanta *scaling factor* dengan nilai standar menurut Winkler adalah 0,1. (Pandya dan Virparia, 2011).

