



### **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

#### 2.1 Program Dual Degree Informatika UMN

Program Dual Degree Informatika UMN mempersiapkan mahasiswa-mahasiswa dengan kemampuan dibidang teknologi dan informasi yang dapat diterapkan pada berbagai macam bidang industri. UMN bekerja sama dengan Swinburne University of Technology yang berada di Melbourne, Australia untuk program ini (Hernawa, 2019a). Mahasiswa akan belajar di UMN selama 5 semester pertama dan 3 semester berikutnya dilanjutkan di Swinburne University. Mahasiswa dari lulusan ini akan mendapatkan dua gelar yaitu S.Kom. dari UMN dan Bachelor of IT (BIT) dari Swinburne University. Dengan adanya dua gelar itu maka mahasiswa lulusan *dual degree* ini mampu bekerja di Indonesia maupun di luar negeri atau global (UMN, 2019).

Program Dual Degree Informatika UMN sudah dibuka sejak tahun 2017. Sebelum dipilihnya seorang koordinator, maka koordinasi program ini pada awalnya masih digabung dengan Program Studi Informatika UMN (Hansun, 2019). Program *dual degree* ini memiliki kelas khusus yang menggunakan Bahasa Inggris dan kurikulum yang berbeda dengan program *regular* Informatika UMN. Kriteria kelulusan mahasiswa program *dual degree* dilihat dari jumlah SKS yang telah diambil selama kuliah di UMN dan di Swinburne University. Mahasiswa *dual degree* tidak akan mengambil mata kuliah magang di UMN dan mereka akan melaksanakan skripsi di Swinburne University of Technology (Hansun, 2019).

## 2.2 Voice Chatbot

*Voice recognition chatbot* atau singkatnya disebut *voice chatbot* adalah perkembangan dari *chatbot* berbasis tulisan, yang mana *chatbot* ini menggunakan masukan dan keluaran berupa suara. *Chatbot* merupakan program atau aplikasi yang dapat melakukan percakapan dengan manusia dalam bahasa natural dengan kecerdasan buatan (Mauldin, 1994). Program ini dapat diakses baik dengan menggunakan teks atau dengan menggunakan suara. Namun awal perkembangan *chatbot* ini lebih dominan dengan basis teks atau tulisan.

Seiring dengan kemajuan teknologi dan untuk meningkatkan kenyamanan dalam penggunaan aplikasi, maka tidak hanya melalui teks tetapi *chatbot* juga dapat menggunakan pengenalan suara (Preez dkk, 2009). Dengan pengenalan suara manusia tidak perlu mengetik lagi, tetapi proses interaksi atau komunikasi dengan *chatbot* dapat dilakukan seperti berbicara dengan manusia biasanya. Proses pengenalan suara atau *voice recognition* membutuhkan dua bagian proses yaitu menangkap atau menerima dan menganalisis suara (Kacur dkk, 2007). Setelah itu, dari suara bisa didapatkan bentuk teksnya yang kemudian akan diproses seperti *chatbot* berbasis teks untuk memperoleh informasi. Informasi yang didapat akan diproses dan dikeluarkan kembali menjadi suara.

*Voice chatbot* dan *chatbot* berbasis teks memiliki kesamaan di mana keduanya memerlukan *natural language processing*. Masukan yang diberikan merupakan bahasa yang digunakan manusia sehari-hari dan tentu tidak akan dimengerti mesin. Oleh karena itu *chatbot* harus dapat memproses bahasa tersebut sehingga dapat mengerti dan memberikan keluaran hasil yang dapat dimengerti manusia juga.

### 2.3 Wit.ai

Wit.ai merupakan sebuah platform *chatbot* yang dikembangkan oleh perusahaan besar ternama yaitu Facebook. Wit.ai berfungsi sebagai penyedia layanan untuk *natural language processing* maupun *natural language understanding*. Selain Wit.ai, penyedia layanan untuk NLP juga terdapat API.AI (sekarang bernama Dialogflow milik Google), IBM Watson, dan Amazon Lex. Di antara semuanya, Wit.ai menyediakan layanan gratis tanpa batasan dan juga memiliki cakupan bahasa yang luas beserta *machine learning* (Canonico dan Russis, 2018).

Wit.ai bekerja dengan mengekstrak *intent* atau maksud dari kalimat yang diberikan. Model pemberian label *intent* tersebut dilatih oleh user dengan memberikan contoh kalimat dan *intent* yang benar. Selain *intent*, sebuah kalimat juga dapat diekstrak *entities* atau entitas yang penting dalam kalimat tersebut. *Entities* merupakan kata-kata dalam kalimat yang dapat diperoleh informasinya. Dengan adanya *intent* dan *entities* yang diperoleh dari suatu kalimat maka maksud yang ditangkap oleh *chatbot* dapat lebih baik dan fleksibel serta tanggapan yang diberikan dapat lebih tepat (WIT.AI, 2019).

Wit.ai sebagai sebuah platform *chatbot* digunakan untuk mengerti maksud dari pertanyaan-pertanyaan yang akan diajukan user. Semakin banyak contoh pertanyaan yang diberikan maka akan semakin mengerti *chatbot* akan maksud dari suatu pertanyaan (WIT.AI, 2019). Namun untuk tanggapan dari maksud suatu pertanyaan tidak disediakan dalam platform ini maka jawaban-jawaban seputar Program Dual Degree Informatika UMN dari *intent* atau maksud pertanyaan akan disimpan pada platform berbeda.

## 2.4 Algoritma Jaro-Winkler

Algoritma Jaro-Winkler atau Jaro-Winkler Distance adalah algoritma Jaro dengan modifikasi yang dilakukan Winkler. Algoritma Jaro Distance umum digunakan untuk mencari jarak perbedaan antar dua buah *string*. Algoritma ini berdasarkan *matching* / pencocokan dan *transposition* / transposisi (Coster dkk, 2011).

*Matching* ( $m$ ) adalah menghitung skor berdasarkan kemunculan karakter pada kedua *string* (Coster dkk, 2011). Setiap karakter dari *string* A ( $s$ ) akan dicari karakter yang sama dari *string* B ( $t$ ) dengan jarak pada karakter B yang belum ditemukan. Jaraknya adalah setengah jumlah *string* terpanjang dikurang satu. Transposisi ( $t_r$ ) adalah menghitung banyaknya karakter *string* A yang tidak sejajar dengan karakter *string* B (Coster dkk, 2011). Persamaan untuk Jaro Distance ( $d_j$ ) adalah (Dreßler dan Ngomo, 2016).

$$d_j = \begin{cases} 0 & \text{jika } m = 0 \\ \frac{1}{3} \left( \frac{m}{|s|} + \frac{m}{|t|} + \frac{m-t_r}{m} \right) & \text{jika } m \neq 0 \end{cases} \dots(2.1)$$

Winkler memodifikasi algoritma Jaro berdasarkan perbandingan kata depan setiap *string*. Dengan adanya modifikasi ini maka akan ada penambahan skor pada saat membandingkan. Ukuran dari kata depan ( $l$ ) dapat menjadi sebuah variabel. Namun ukuran kata depan yang biasa digunakan adalah empat (Coster dkk, 2011). Modifikasi dilakukan karena Winkler mengamati bahwa umumnya kesalahan terjadi pada tengah dan akhir kata. Penambahan skor yang dilakukan juga berdasarkan suatu nilai batasan  $b_r$  yang umumnya 0.7 (Dreßler dan Ngomo, 2016). Persamaan untuk Jaro Winkler ( $d_w$ ) adalah

$$d_w = \begin{cases} d_j & \text{jika } d_j < b_t \\ d_j + (l p(1 - d_j)) & \text{jika } d_j \geq b_t \end{cases} \quad \dots(2.2)$$

Dalam penelitian ini, algoritma Jaro-Winkler akan digunakan untuk menghitung jarak perbedaan antara nama yang dimasukkan pengguna dengan nama yang terdapat pada daftar nama. Tujuannya adalah untuk mencari nilai terbesar yang dianggap sebagai nama sesungguhnya yang dimaksud pengguna. Terdapat berbagai macam nama yang umum digunakan baik yang di Indonesia atau yang di luar Indonesia (Ernawati, t.th.; Kantrowitz dan Ross, 1991a; Kantrowitz dan Ross, 1991b; PanduanIbu, t.th.).

## 2.5 End User Computing Satisfaction (EUCS)

*End user computing satisfaction* merupakan evaluasi secara keseluruhan atas sistem informasi berdasarkan pengalaman pengguna akhir dalam menggunakan sistem (Chin dan Matthew, 2000). Sejumlah studi telah dilakukan untuk menilai keseluruhan evaluasi yang mana dimaksud adalah berupa kepuasan dari pengguna akhir dan juga faktor-faktor penentunya. Model evaluasi ini dikembangkan oleh Doll & Torkzadeh (1988) dengan faktor atau dimensi kepuasan seperti *content*, *accuracy*, *format*, *ease of use*, dan *timeliness*.

Penerapan EUCS ini dilakukan dengan pertanyaan-pertanyaan yang mewakili setiap dimensi kepuasan yang ada. Contoh pertanyaan untuk dimensi *content* seperti “apakah situs ini menyediakan informasi yang sesuai kebutuhan anda?” (Helm dkk, 2005). Contoh pertanyaan untuk dimensi *accuracy* seperti “apakah anda puas dengan keakuratan situs ini?” (Helm dkk, 2005). Contoh pertanyaan untuk dimensi *format* seperti “Apakah situs ini menyediakan informasi yang jelas?” (Helm dkk, 2005). Contoh pertanyaan untuk dimensi *ease of use* seperti “apakah situs ini mudah

digunakan?” (Helm dkk, 2005). Contoh pertanyaan untuk dimensi *timeliness* seperti “apakah anda mendapatkan informasi yang anda butuhkan cepat?” (Helm dkk, 2005).

Dengan melihat kepuasan dari pengguna maka dapat ditarik kesimpulan mengenai apakah aplikasi yang dibuat berguna dan diterima oleh umum. Berdasarkan dimensi kepuasan yang ada maka akan ada pertanyaan-pertanyaan yang mengenai masing-masing dimensi. Jawaban dari pertanyaan akan dievaluasi lebih lanjut dengan skala Likert. Skala Likert digunakan untuk mengukur tingkat sikap secara ilmiah. Skala ini membobotkan pilihan sikap yang ada ke dalam nilai (Joshi dkk, 2015).

