

Rancang Bangun Aplikasi Chatbot Sebagai Media Pencarian Informasi Anime Menggunakan Regular Expression Pattern Matching

David Domarco¹, Ni Made Satvika Iswari²

Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia
david.domarco@student.umn.ac.id¹, satvika@umn.ac.id²

Diterima 12 April 2017

Disetujui 16 Mei 2017

Abstract— *Technology development has affected many areas of life, especially the entertainment field. One of the fastest growing entertainment industry is anime. Anime has evolved as a trend and a hobby, especially for the population in the regions of Asia. The number of anime fans grow every year and trying to dig up as much information about their favorite anime. Therefore, a chatbot application was developed in this study as anime information retrieval media using regular expression pattern matching method. This application is intended to facilitate the anime fans in searching for information about the anime they like. By using this application, user can gain a convenience and interactive anime data retrieval that can't be found when searching for information via search engines. Chatbot application has successfully met the standards of information retrieval engine with a very good results, the value of 72% precision and 100% recall showing the harmonic mean of 83.7%. As the application of hedonic, chatbot already influencing Behavioral Intention to Use by 83% and Immersion by 82%.*

Index Terms—*anime, chatbot, information retrieval, Natural Language Processing (NLP), Regular Expression Pattern Matching*

I. PENDAHULUAN

Memasuki awal abad ke-20, seniman grafis Jepang mulai merasakan pengaruh dari perkembangan teknologi Barat, yaitu film digital. Perubahan tersebut merupakan pemicu lahirnya sebuah seni modern Jepang, yang dikenal dunia dengan nama *anime* atau film animasi Jepang. *Anime* sendiri merupakan film animasi dengan teknik penggambaran yang melibatkan emosi dari setiap karakter dengan alur yang kompleks. Berdasarkan rekap data distribusi *anime* yang dikeluarkan oleh AJA[1], distribusi *anime* di internet berkembang secara pesat setiap tahunnya. Hal ini dipicu oleh bertambah besarnya jumlah penggemar *anime* yang mengekspresikan ketertarikan mereka dengan mencari segala informasi yang berhubungan dengan kegemaran mereka.

Berdasarkan survey yang dilakukan Shawar, Atwell, dan Roberts[2] *chatbot* telah sukses dikembangkan dan diterapkan pada bidang edukasi, pencarian informasi, bisnis, dan *e-commerce*. Banyak *chatbot anime* yang telah beredar di internet baik *chatbot* yang dibuat oleh profesional maupun kaum awam. Namun, belum ada aplikasi *chatbot anime* yang dibangun untuk dapat menangani permintaan pengguna dalam mencari informasi spesifik dari seluruh film *anime* yang ada. Pada penelitian ini dibangun sebuah aplikasi *chatbot* yang dikhususkan sebagai sebuah media pencarian informasi seputar film *anime* yang diberi nama Reikobot. Reikobot dibangun sebagai sebuah fasilitas untuk memudahkan penggemar *anime* dalam berinteraksi dan mencari informasi seputar film *anime* selayaknya bercakap dengan manusia secara natural.

Saat ini, sudah banyak dikembangkan aplikasi *chatter-boot*, atau dikenal sebagai *chatbot* untuk berbagai tujuan tertentu, atau hanya untuk media hiburan[3]. Aplikasi *chatbot* akan mencocokkan kalimat input dari pengguna dengan *pattern* yang ada pada *knowledge* yang sudah disimpan sebelumnya. *Knowledge* didapatkan dari berbagai sumber dan menjadi bahan dari pola *chat* antara pengguna dengan *chatbot*[4]. Dalam penelitian ini dapat dilihat seberapa besar pengaruh *chatbot* dalam memberikan informasi yang relevan bagi pengguna serta rasa nyaman selama dalam menggunakan aplikasi.

II. PENELITIAN TERKAIT

Sebuah *chatbot* pencarian informasi yang telah berhasil dibangun adalah FAQchat[2], yang digunakan sebagai sebuah aplikasi *question answering* seputar Frequently Ask Question (FAQ) di sebuah *School of Computing* (SoC), University of Leeds. FAQchat adalah sebuah *chatbot* pencarian informasi yang dikembangkan dengan menggunakan AIML (*Artificial Intelligence Markup Language*) *pattern matching* yang mengambil informasi dari *data base* FAQ (*Frequently Ask Question*). FAQchat

memperoleh informasi menggunakan kata yang signifikan sebagai kata kunci, kemudian mencoba mencari *pattern* terpanjang yang sesuai tanpa menggunakan *linguistic tools* atau menganalisa makna dari *query* pengguna. FAQchat tidak membutuhkan modul pengetahuan *linguistic* dan menerapkan prinsip *language independent*. Proses kerja FAQchat adalah sebagai berikut [2]:

1. Semua pertanyaan dan jawaban diekstrak dari *database* setelah melakukan *process filtering* untuk menghilangkan *tag* yang tidak dibutuhkan.
2. *Database* FAQ tersusun atas pertanyaan dan jawaban. Dengan menggunakan pola tersebut, sebuah daftar *link* dibangun, yang berisi *link* dari FAQ ke halaman web yang berisi jawaban.
3. Disusunlah sebuah kamus yang berisikan semua kata dalam pertanyaan dengan frekuensi kejadian. Kemudian, kata pertama atau kedua yang paling signifikan diambil dari setiap pertanyaan yang telah diajukan.
4. AIML *patern matching rules*, atau dikenal sebagai “kategori” dibuat. Terdapat dua tipe *pattern matching* pada FAQchat, yaitu seluruh pertanyaan, atau hanya kata pertama atau kedua yang paling signifikan yang mengalami kecocokan dengan data pertanyaan pada *database* FAQ. Terdapat dua kemungkinan respon yang diberikan FAQchat, yaitu sebuah jawaban jika terdapat satu pertanyaan yang cocok atau beberapa jawaban jika kata pertama atau kedua yang paling signifikan ditemukan dalam beberapa pertanyaan.

III. KOMPONEN CHATBOT

A. Part of Speech Tagging

Part of Speech Tagging (POS Tagging) merupakan sebuah proses yang bekerja untuk mengidentifikasi tata bahasa pada suatu kalimat [5]. POS Tagging digunakan untuk mengelompokkan kata-kata yang terdapat dalam kalimat natural guna memahami kategori kata di dalam suatu kalimat. Sebagai contoh berdasarkan tata bahasa natural, kalimat “*What is rezero anime?*” tersusun atas empat kata yaitu “*what*”, “*is*”, “*rezero*” dan “*anime*”. Pada proses *tagging* ketiga kata tersebut akan diidentifikasi menjadi “*what*” sebagai WH Word, “*is*” sebagai *verb*, dan “*rezero anime*” sebagai *noun*. WH Word merupakan kata pertanyaan untuk menanyakan pertanyaan, seperti *what*, *where*, *when*, *who*, *whom*, *why*, dan *how*. Disebut sebagai WH Word karena dalam Bahasa Inggris, kebanyakan kata tersebut dimulai dengan huruf *wh*. *Noun* merupakan objek dari suatu perbincangan di dalam kalimat sehingga berdasarkan pengelompokkan di atas, sistem dapat mengenali bahwa topik yang terdapat di dalam bahasa natural tersebut adalah “*rezero anime*”. Topik yang didapat dapat digunakan sebagai *keywords* dalam pencarian judul film *anime*.

Aplikasi *chatbot* pencarian informasi *anime* (Reikobot) menggunakan Brill Tagger, yaitu sebuah

algoritma POS *Tagging* yang menggunakan metode induksi berbasis transformasi untuk memahami tata bahasa natural[6]. Teknik pemberian *tag* ini menggunakan seperangkat aturan yang telah ditetapkan sebelumnya dalam bentuk sebuah kumpulan *vocabulary*. Jika kata diketahui, maka kata akan diberikan *tag* sesuai dengan sebuah kamus *vocabulary* yang telah disiapkan. Jika kata tersebut tidak diketahui, maka secara naif kata tersebut akan diberikan *tag noun* sedangkan jika kata tersebut diketahui, maka kata tersebut akan diberikan *tag* sesuai *record* pada kamus *vocabulary*. Kemudian, kata tersebut akan memasuki seperangkat *if rules* yang secara bertahap akan mengubah *tag* awal menjadi *tag* lain yang lebih akurat bila kata tersebut memenuhi kondisi yang ditentukan pada suatu *rules*.

Gambar 1 menampilkan contoh *pseudocode* dari penerapan metode Brill *tagger*.

```
FOR each word as word
  IF the word IS SET IN dictionary THEN
    tag word WITH dictionary tag
  ELSE
    tag word AS noun
  END IF.

  IF word tag AS verb AND the word before IS 'the' THEN
    change word tag TO noun
  END IF.

  IF the word before IS 'would'
    change word tag TO verb
  END IF.

  ..... // apply any if rules

  IF the word END WITH 'ly'
    change word tag TO adverb
  END IF.

  // and so on
END FOR.
```

Gambar 1. Pseudocode Brill Tagger

B. Regular Expression Pattern Matching

Menurut Navarro dan Raffinot [7] *regular expression* memberikan solusi yang sangat kuat dalam mengekspresikan sederet pencarian *pattern*. *Regular Expression Pattern Matching* menggunakan sekumpulan *regular expression* yang disusun menjadi sebuah *pattern*. Bila kalimat masukan cocok dengan salah satu *pattern* yang ada, maka sistem akan melakukan proses sesuai perintah yang ditentukan pada *pattern* tersebut. Gambar 2 menampilkan contoh *pattern* menggunakan *regular expression* dalam pencarian informasi *anime*.

```
$patterns = array(
'desc' => "/((description)|(summary)|(info)|(story)|(synop))/", // desc
'type' => "/( ) -((type)|(series)|(movie)|(ova)|(music))/", // type
'eps' => "/((episode)|(chapter)|(long)|(day))/", // episode
'stat' => "/((status))/", // status
'date' => "/((date)|(month))/", // aired
'season' => "/((season)|(start)|(aired)|(end)|(period)|(year))/", // season
'studio' => "/((studio)|(make)|(produce))/", // studio
'source' => "/(source)|(adapt)/", // source
'genre' => "/((genre))/", // genre
'rank' => "/((rank))/", // ranked
'pop' => "/((popular))/", // popularity
'url' => "/((link).*(anime)|())|((url).*(anime)|())/", // link ... anim
//
);
```

Gambar 2. Regular Expression Pattern

C. Precision, Recall, dan Harmonic Mean

Menurut Manning dkk. [5] dalam bukunya yang berjudul “*Introduction to Information Retrieval*” unsur utama yang diukur dalam sistem pencarian informasi adalah *precision* dan *recall*. *Precision* merupakan seberapa banyak jumlah dokumen relevan yang didapatkan dari seluruh dokumen yang berhasil diambil. *Recall* mendefinisikan seberapa banyak sistem mengembalikan dokumen yang relevan dari seluruh dokumen relevan yang tersedia.

Tabel 1 Tabel Kontingensi

	<i>Relevant</i>	<i>Non Relevant</i>
<i>Retrieved</i>	<i>true positive (tp)</i>	<i>false positive (fp)</i>
<i>Non Retrieved</i>	<i>false negative (fn)</i>	<i>true negative (tn)</i>

Tabel 1 menunjukkan beberapa kondisi yang perlu diperhitungkan dalam *information retrieval*. Kondisi *true positive (tp)* merupakan jumlah dokumen relevan yang diambil. Kondisi *false positive (fp)* merupakan jumlah dokumen tidak relevan yang diambil. Kondisi *false negative (fn)* merupakan jumlah dokumen relevan yang tidak ditemukan. Sementara kondisi *true negative (tn)* merupakan jumlah dokumen tidak relevan yang tidak ditemukan.

Berdasarkan Tabel Kontingensi (Tabel 1) gagasan *precision* dan *recall* dapat ditafsirkan menjadi sebuah persamaan berikut [5]:

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} = \left(\frac{tp}{tp + fp} \right) \times 100\% \quad (1)$$

$$\text{Recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})} = \left(\frac{tp}{tp + fn} \right) \times 100\% \quad (2)$$

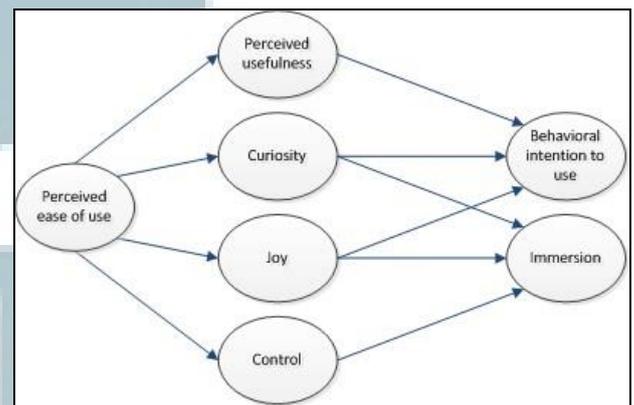
Oleh karena *precision* dan *recall* merupakan unsur yang saling berlawanan (*trade-off*), dalam mengevaluasi toleransi *trade-off* tersebut digunakanlah penghitungan F-Measure yang merupakan *weighted harmonic mean* dari *precision* dan *recall*. Nilai minimum *harmonic mean* yang harus dicapai adalah sebesar 70% dikarenakan pada titik tersebut nilai dari *precision* dan *recall* mencapai keseimbangan *trade-off* [11].

$$\text{F_Measure} = \frac{\#(2PR)}{P+R} \quad (3)$$

IV. HEDONIC MOTIVATION SYSTEM ADOPTION MODEL (HMSAM)

HMSAM [8] adalah sebuah model untuk mengukur sebuah sistem yang mengadaptasi motivasi hedonis. Ada lima faktor yang menjadi fokus pengukuran pada HMSAM, yaitu *perceived usefulness*, *perceived ease of use*, *curiosity*, *control*, dan *joy* yang akan mempengaruhi *behavioral intentional to use* dan *immersion* suatu aplikasi. Aspek-aspek tersebut adalah sebagai berikut:

- *Perceived usefulness*, mengukur peningkatan kinerja ketika menggunakan suatu sistem.
- *Perceived ease of use*, mengukur kemudahan penggunaan suatu sistem.
- *Curiosity*, sejauh mana suatu sistem dapat meningkatkan rasa ingin tahu dalam aspek kognitif.
- *Control*, persepsi pengguna bahwa dirinya yang diajak berinteraksi oleh sistem.
- *Joy*, aspek kesenangan yang didapatkan dari interaksi dengan sistem.
- *Behavioral intentional to use*, keinginan pengguna untuk menggunakan aplikasi.
- *Focussed Immersion*, total keterlibatan yang dilakukan antara pengguna dengan aplikasi yang mengukur seberapa dalam pengguna terfokus dalam menggunakan sistem.



Gambar 3. Overview of HMSAM [3]

V. PERANCANGAN SISTEM

Penelitian ini membuat sebuah rancang bangun sistem chatbot sebagai media pencarian informasi *anime* yang ditulis dalam bahasa PHP. Sistem chatbot ini diberi nama Reikobot.

A. Peancangan Sistem

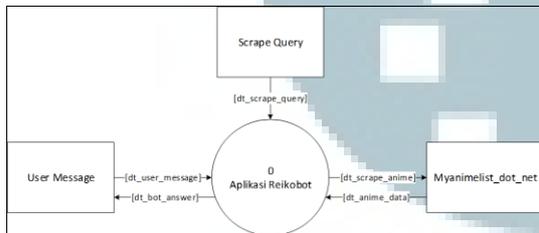
Desain rancangan sistem Reikobot adalah sebagai berikut.

1. Kalimat natural atau pertanyaan yang dimasukkan pengguna di pecah menjadi sekumpulan *array* kata, kemudian dilakukan transformasi (*normalisasi*, *stop word* dan *strip punctuation*)

terhadap setiap kata sehingga didapatkan kata dalam bentuk kata dasar. Sesudah itu, dibentuklah kalimat baru (kalimat sederhana) dengan unsur kata yang telah disederhanakan.

2. Dilakukan teknik *tagging* pada kalimat sederhana sehingga setiap kata mempunyai sebuah *tag grammar*. Lalu, dilakukan penyeleksian berdasarkan *tag* agar tercipta sebuah kalimat baru (kalimat *tag*) yang didalamnya hanya berisi kata dengan *tag* yang bermakna.
3. Sistem memiliki beberapa topik yang dikelompokkan ke dalam *pattern* tertentu yaitu *pattern* untuk pertanyaan terbuka dan *pattern* untuk pertanyaan khusus atau spesifik. Kalimat *tag* akan dicocokkan dengan *pattern* sehingga sistem dapat mengkategorikan jenis pertanyaan yang terkandung di dalam kalimat *tag*.
4. Jenis pertanyaan yang telah dikategorikan akan diproses sesuai perintah yang diprogram di dalam sistem dan akan mengembalikan keluaran berupa jawaban yang memenuhi kriteria. Jika kriteria tidak ditemukan, maka sistem akan memberikan jawaban seperti "Sorry dear, Nee-san can't find the data you are looking for".

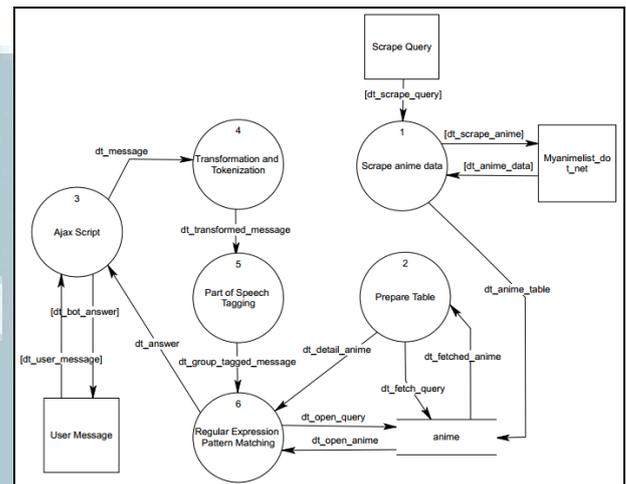
B. Data Flow Diagram (DFD)



Gambar 4. Diagram Konteks Aplikasi Reikobot

Gambar 4 menunjukkan Diagram Konteks dari Aplikasi Reikobot yang dibangun. Sementara Gambar 5 menampilkan alur data tingkat satu dari sistem *chatbot* pencarian informasi *anime*. Terdapat lima buah proses dan dua buah entitas di dalam diagram alur data. Alur data *dt_scrape_query* dan *dt_anime_data* berisi *query* dengan hasil berupa seluruh data dan informasi mengenai film *anime* yang berasal dari entitas *Myanimelist_dot_net* dan didapatkan melalui proses *scrape anime data*. Data dan informasi *anime* yang berhasil diambil disimpan ke dalam tabel *anime*. Data *dt_message* dari entitas user diterima oleh proses *ajax script*, kemudian message tersebut diolah menjadi data *dt_ajax_post* yang akan melewati tiga buah proses *natural language processing*, yaitu *transformation and tokenization*, *part of speech tagging*, dan *regular expression pattern matching* yang masing-masing menghasilkan data berupa *dt_transformed_message*, *dt_group_tagged_message*, dan *dt_json_encode*.

Proses *regular expression pattern matching* membutuhkan data *dt_detail_anime* dan *dt_open_anime* yang akan diolah menjadi sebuah keluaran dalam bentuk JSON (JavaScript Object Notation). Data *dt_detail_anime* merupakan data yang berisi informasi *anime* terkait pertanyaan tertutup sedangkan *dt_open_anime* merupakan data hasil *query* yang berisi informasi *anime* terkait pertanyaan terbuka. Setelah melewati ke tiga buah proses tersebut, *dt_json_encode* yang dikirimkan kembali ke proses *ajax script* akan diolah menjadi sebuah jawaban bernama *dt_answer*. Lalu, data *dt_answer* diterima oleh entitas *user*.



Gambar 5. DFD Level 1

VI. IMPLEMENTASI



Gambar 6. Hasil Implementasi Reikobot

Setelah berhasil diimplementasikan aplikasi *chatbot* pencarian informasi *anime* (Reikobot) memiliki tampilan seperti pada Gambar 6. Tampilan antar muka Reikobot memiliki dua buah elemen utama yaitu *chat box* yang terletak di sisi kiri dan *instruction box* yang terletak di sisi kanan. *Chat box* adalah tempat pengguna berinteraksi dengan aplikasi Reikobot. *Instruction box* berisi *hint* yang membantu pengguna untuk berinteraksi dengan aplikasi.

Reikobot. Selain kedua elemen utama tersebut terdapat elemen *online survey* yang terletak di pojok kanan bawah dari tampilan.

VII. EVALUASI SISTEM

Setelah aplikasi *chatbot* pencarian informasi *anime* (Reikobot) telah selesai diimplementasikan tahap selanjutnya adalah pengujian terhadap sistem. Terdapat dua buah pengujian yang dilakukan yaitu evaluasi *chatbot* sebagai mesin pencarian informasi dan evaluasi *chatbot* sebagai sebuah aplikasi hedonis.

A. Evaluasi Chatbot sebagai Mesin Pencarian Informasi

Berdasarkan *rule of thumb* dari pencarian informasi, jumlah minimum pencarian data dalam suatu tes adalah 50 buah [5]. Relevansi dari dokumen ditentukan oleh satu orang *judge* sebagai titik pengukuran dengan mengikuti standar pengukuran *unranked retrieval sets*. Pengukuran menggunakan nilai biner, yaitu nilai "0" untuk menandakan dokumen yang tidak relevan dan nilai "1" untuk menandakan dokumen yang relevan. Berikut merupakan penghitungan *Recall* dan *Precision* sesuai hasil tes relevansi yang dinotasikan pada Tabel 1.

Tabel 2. Notasi Hasil Tes Relevansi

	<i>Relevant</i>	<i>Non Relevant</i>
<i>Retrieved</i>	36	14
<i>Non Retrieved</i>	0	0

Berdasarkan Tabel 2, dilakukan penghitungan *precision*, *recall*, dan *F-Measure*. Dari hasil penghitungan didapatkan *precision* sebesar 72%, *recall* sebesar 100%. Nilai *harmonic mean* yang tercapai adalah sebesar 83.7%.

B. Evaluasi Chatbot dengan Metode HMSAM

Chatbot merupakan aplikasi *entertainment* atau hiburan yang mengarah pada aktivitas dengan unsur intrinsik di dalamnya sehingga *chatbot* dikategorikan sebagai sebuah sistem hedonis. Menurut Lowry dkk. [5] teknik evaluasi yang digunakan untuk menguji sistem hedonis adalah dengan menggunakan aspek-aspek dalam *Hedonic Motivation System Adoption Model* yang disingkat sebagai HMSAM.

Menurut Gay and Diehl [9] jumlah minimum *sample* yang diperlukan dalam melakukan suatu penelitian adalah sebanyak 30 buah. Melalui kuisoner *online* yang disebar di dapatkan 33 orang responden. Pertanyaan yang diajukan kepada 33 responden ini adalah lima buah pertanyaan mengenai *perceived ease of use*, *perceived usefulness*, *curiosity*, *control*, dan *joy* yang akan mempengaruhi *behavioral intention to use* dan *immersion* suatu aplikasi hedonis. Respon yang diberikan oleh ke tiga puluh tiga responden dapat dilihat pada Tabel 3.

Tabel 3 Respon Kuisoner HMSAM

#	<i>Strongly Disagree</i>	<i>Dis-agree</i>	<i>Nei-ther</i>	<i>Agree</i>	<i>Strongly Angree</i>	<i>Aspek HMSAM</i>
1	0	1	1	19	12	<i>Perceived Ease of Use</i>
2	0	1	1	22	9	<i>Perceived Usefulness</i>
3	1	1	7	12	12	<i>Curiosity</i>
4	1	1	5	8	18	<i>Control</i>
5	0	2	7	15	9	<i>Joy</i>

Berdasarkan Tabel 3, dilakukan penghitungan Likert scale [10] yang mewakili aspek-aspek HMSAM. Hasil dari penghitungan disajikan pada Tabel 4.

Aspek *Behavioral Intention to Use* dan *Immersion* dihitung menggunakan rata-rata aritmatika dan didapatkan *Behavioral Intention to Use* sebesar 83% serta *Immersion* sebesar 82%.

Tabel 4 Hasil Penghitungan Likert Scale HMSAM

Aspek HMSAM	Penghitungan Likert
<i>Perceived Ease of Use</i>	0.854545 (<i>Strongly Agree</i>)
<i>Perceived Usefulness</i>	0.836364 (<i>Strongly Agree</i>)
<i>Curiosity</i>	0.8 (<i>Strongly Agree</i>)
<i>Joy</i>	0.848485 (<i>Strongly Agree</i>)
<i>Control</i>	0.787879 (<i>Agree</i>)
<i>Behavioral Intention to Use</i>	0.834848 (<i>Strongly Agree</i>)
<i>Immersion</i>	0.81812 (<i>Strongly Agree</i>)

VIII. SIMPULAN

Aplikasi *chatbot* sebagai media interaktif dalam mendapatkan informasi seputar *anime* berbasis teks menggunakan *regular expression pattern matching* telah berhasil dirancang dan dibangun dengan nama Reikobot. Aplikasi Reikobot menghasilkan nilai *precision* sebesar 72% dan *recall* sebesar 100% yang menghasilkan nilai *harmonic mean* sebesar 83,7%. Adapun nilai *harmonic mean* tersebut bermakna bahwa aplikasi dapat menyajikan informasi dengan *precision* dan *recall* yang harmonis atau seimbang dengan bobot yang tidak terlalu jauh berbeda.

Berdasarkan penghitungan Likert, aplikasi Reikobot menghasilkan tingkat *Behavioral intention to use* (BIU) sebesar 83% yang berarti pengguna sangat setuju bahwa aplikasi meningkatkan minat pengguna dalam pencarian informasi *anime* dan menghasilkan tingkat *Immersion* sebesar 82% yang berarti pengguna sangat terfokus ketika menggunakan aplikasi.

DAFTAR PUSTAKA

- [1] The Association of Japanese Animation. March 2015. *Anime Industry 2014 Summary*. The Association of Japanese Animation, Database Working Group, Akihabara.
- [2] Shawar, B. A., Atwell, E; dan Roberts, A. 2005. FAQchat as in Information Retrieval system. Web: eprints.whiterose.ac.uk/4663, diakses pada 4 April 2016.
- [3] Augello, A., Pilato, G., Machi, A., dan Gaglio, S. 2012. *An Approach to Enhance Chatbot Semantic Power and Maintainability: Experiences Within The FRASI Project*.

- Proc. Of 2012 IEEE Sixth International Conference on Semantic Computing.
- [4] Setiaji, B. dan Wibowo, F.W. 2016. *Chatbot Using A Knowledge in Database*. Proc. Of 2016 7th International Conference on Intelligent System, Modelling and Simulation.
- [5] Manning, C. D., Raghavan, P., dan Schütze, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press.
- [6] Brill, Eric. 1995. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging. *Computational Linguistics*, December 1995.
- [7] Navarro, G., dan Raffinot M. 2002. *Flexible Pattern Matching in String: Practical On-Line Search Algorithms for Texts and Biological Sequence*. Cambridge University Pers.
- [8] Lowry, P. B. dkk. 2013. *Taking 'fun and games' seriously: Proposing the hedonic-motivation system adoption model (HMSAM)*. *Journal of the Association for Information Systems (JAIS)*, vol. 14(11), 617–671.
- [9] Gay L.R. dan Diehl P.L. 1992. *Research Method for Business and Management*.
- [10] Likert, Rensis. 1932. *A Technique for the Measurement of Attitudes*. *Archives of Psychology* 140.
- [11] van Rijsbergen, C. J. 1979. *Information Retrieval*. London: Butterworths.

