

Rancang Bangun *Chatbot Helpdesk* untuk Sistem Informasi Terpadu Universitas Sam Ratulangi

Ruspandi R. Benedictus, Hans Wowor, Alwin Sambul

Teknik Informatika Universitas Sam Ratulangi Manado, Indonesia.
ruspandireynaldi@gmail.com, hanswowor@unsrat.ac.id, asambul@gmail.com

Abstrak – Universitas Sam Ratulangi sudah menggunakan teknologi informasi dalam bentuk aplikasi untuk mempermudah kegiatan civitas akademika. Aplikasi-aplikasi tersebut tergabung dalam Sistem Informasi Terpadu.

Layanan *helpdesk* merupakan hal yang pertama dicari oleh *user* ketika mengalami masalah atau memerlukan informasi dan bantuan atas suatu produk atau jasa. Akan tetapi, Universitas Sam Ratulangi belum mempunyai layanan *helpdesk* untuk memberikan pelayanan kepada *user*. Karena itu dibuatlah sebuah aplikasi *chatbot helpdesk* yang bisa menjawab pertanyaan-pertanyaan seputar penggunaan aplikasi-aplikasi dalam Sistem Informasi Terpadu.

Chatbot ini menggunakan teknologi *web*, algoritma *bigram* sebagai algoritma untuk pencocokan pola kalimat dan metode *forward chaining* sebagai metode penalaran ketika pertanyaan yang diberikan memerlukan kondisi-kondisi tertentu agar bisa dijawab. Metodologi pengembangan perangkat lunak yang digunakan adalah metode *Prototyping* dan pengujian aplikasi menggunakan metode *Black Box*.

Penelitian ini menghasilkan aplikasi *Chatbot Helpdesk* yang bisa membantu *user* dengan menjawab pertanyaan seputar penggunaan aplikasi-aplikasi dalam Sistem Informasi Terpadu Universitas Sam Ratulangi.

Kata kunci : *Helpdesk, Chatbot, Bigram, Forward Chaining, Inference Engine*

I. PENDAHULUAN

Universitas Sam Ratulangi (Unsrat) merupakan universitas yang sudah memanfaatkan teknologi informasi dalam beberapa proses bisnisnya. Semua teknologi tersebut tergabung dalam Sistem Informasi Terpadu (SIT) Unsrat. Dalam SIT Unsrat, tergabung sejumlah aplikasi, salah satu contohnya adalah Portal Akademik.

Aplikasi yang beragam serta banyaknya fungsionalitas yang ada, menyebabkan *user* memerlukan waktu untuk beradaptasi agar bisa menggunakan aplikasi-aplikasi tersebut dengan benar, apalagi bagi mahasiswa baru. Untuk Portal Akademik, walaupun sudah ada panduan tertulis yang bisa diunduh, itu tidak menjamin bahwa *user* dapat langsung bisa mengerti penjelasannya. Ketika ingin mendapatkan informasi mengenai aplikasi-aplikasi itu, *user* harus pergi ke lokasi tertentu dan hanya bisa pada jam kantor saja. Kurangnya informasi tentang penggunaan SIT menyebabkan *user*, terutama mahasiswa bertanya kepada mahasiswa lain sehingga informasi yang didapatkan bisa saja tidak akurat.

Dengan demikian, perlu dibangun sebuah aplikasi *chatbot helpdesk* yang bisa menjawab / memberikan informasi seputar penggunaan aplikasi-aplikasi tersebut kepada *user*. Dari hal tersebut maka penulis mengangkat judul “Rancang Bangun *Chatbot Helpdesk* untuk Sistem Informasi Terpadu Universitas Sam Ratulangi”.

II. LANDASAN TEORI

Chatbot yang dibuat merupakan aplikasi *helpdesk* bertipe *Question Answering*, sehingga bantuan yang ditawarkan adalah menjawab pertanyaan *user*. *Chatbot* ini menggunakan algoritma *bigram* sebagai algoritma untuk *pattern matching*, dan *forward chaining inference engine* sebagai metode penalaran jika untuk menjawab pertanyaan *user* memerlukan informasi tambahan.

Chatbot yang dibuat merupakan *chatbot* dengan pengetahuan yang bersifat *close domain*. *Chatbot* dibuat dengan menggunakan teknologi *web*. Untuk *client side* menggunakan HTML, CSS, dan jQuery sebagai *library* Javascript. Sedangkan untuk *server side* menggunakan bahasa pemrograman PHP dengan *framework* CodeIgniter. Proses pengembangan *software* menggunakan metode *Prototyping* dan pengujian menggunakan metode *Black Box*.

A. Helpdesk

Helpdesk merupakan struktur atau program yang menangani segala bentuk keluhan dari berbagai pihak dengan menyediakan layanan untuk memberi informasi dan solusi kepada *user*[1]. Salah satu kebutuhan *user* terhadap *helpdesk* adalah meminta informasi[2], sehingga *helpdesk* harus mempunyai pengetahuan terhadap suatu produk atau layanan yang diberikan.

Di Unsrat sendiri, terutama di Unit Pelaksana Teknis Teknologi Informasi dan Komunikasi (UPT-TIK), tidak ada *helpdesk* fisik yang mengarahkan *user* apabila mereka ingin mendapatkan bantuan teknis ataupun menanyakan informasi. Oleh karena itu, dibuat sistem *helpdesk* yang bisa mengotomatisasi jawaban atas pertanyaan-pertanyaan *user* dan bisa diakses kapan saja dan dimana saja.

B. Chatbot

Chatbot merupakan program komputer yang berinteraksi dengan *user* menggunakan bahasa natural. Teknologi *chatbot* pertama dimulai pada tahun 1960an. Tujuan pembuatan *chatbot* ini adalah pengujian apakah *chatbot* dapat menipu *user* agar mereka mengira sedang berkomunikasi dengan manusia[3]. Pengujian ini dikenal dengan nama “*Turing Test*”.

Chatbot adalah sebuah program yang dapat melakukan obrolan dalam bahasa alami mengenai sebuah topik yang ada dalam model pengetahuan *chatbot* tersebut. Artinya, *chatbot* harus bisa mengenali dan merespon kata-kata

yang diberikan *user*. Kemampuan *chatbot* dalam mengenali dan memberikan respons ditentukan oleh ruang lingkup dari pengetahuan *chatbot*. Terdapat 2 macam ruang lingkup dari *chatbot*, yaitu :

1. *Open Domain*. *User* dapat membawa arah percakapan kemana saja. Jumlah topik yang dibahas tidak terbatas. Dibutuhkan pengetahuan tentang dunia yang luas dalam basis pengetahuan agar bisa mengenali dan memberikan respon terhadap masukan *user*.
2. *Close Domain*. *Chatbot* dirancang untuk memenuhi tujuan yang spesifik. Basis pengetahuan *chatbot* hanya mencakup topik seputar tujuan *chatbot*[4]. Dalam penelitian ini, dibuat *chatbot* bertipe *close domain* sehingga hanya bisa mengenali dan merespon masukan seputar penggunaan SIT Unrat.

C. Pattern Matching

Pattern matching dalam ilmu komputer adalah kegiatan pemeriksaan serangkaian *string* atau karakter yang ada untuk menemukan pola yang konstituen antar *string* [5]. Salah satu algoritma dalam *pattern matching* adalah *bigram*. Algoritma ini membagi *string* menjadi *bigram*. *Bigram* adalah sepasang unit tertulis yang berurutan, berupa huruf, suku kata atau kata[6].

Contoh *bigram* adalah :

String : "saya"

Bigram: {"sa", "ay", "ya"}

Kemudian, dengan menggunakan rumus *sentence similarity measurement* :

$$similarity = \frac{count(S_1 \cap S_2) + count(S_2 \cap S_1)}{count(S_1) + count(S_2)} \quad (1)$$

S1 adalah *string* pertama dan s2 adalah *string* kedua. $Count(s1 \cap s2)$ adalah jumlah irisan antar *bigram* s1 dan s2. $Count(s1)$ adalah jumlah *bigram* s1 dan $count(s2)$ adalah jumlah *bigram* s2.

Algoritma ini digunakan pada tugas akhir ini untuk mencari pertanyaan dalam *database* yang paling mirip dengan pertanyaan yang dimasukkan oleh *user*. Jika ada, *chatbot* akan menampilkan pasangan jawaban dari pertanyaan yang ditemukan tersebut.

D. Inference Engine

Inference engine merupakan *software* yang menghasilkan fakta baru atau asosiasi dari informasi-informasi yang ada. *Inference engine* merupakan *finite state machine* yang terdiri dari 3 *state*, yaitu mencocokkan *rule*, memilih *rule*, dan mengeksekusi *rule*[7]. Pada tahap pertama, *chatbot* akan memilih semua *rule* yang terpenuhi berdasarkan data yang dimasukkan oleh *user*. *Rule-rule* tersebut masuk dalam kelompok *conflict set*. Pada tahap kedua, *chatbot* akan menerapkan sejumlah aturan penyelesaian konflik sehingga didapatkan *rule* yang akan dieksekusi. Pada tahap ketiga, *chatbot* mengeksekusi *rule* yang terpilih pada tahap kedua.

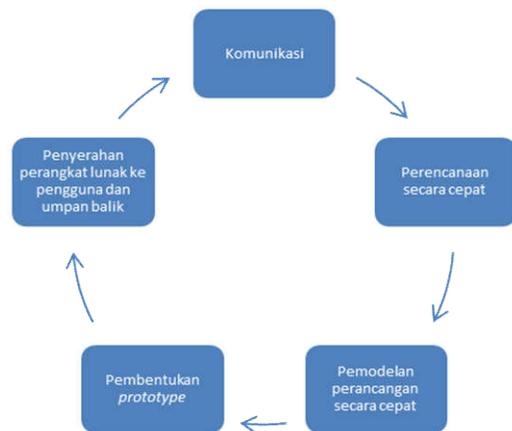
E. Forward Chaining

Metode *forward chaining* merupakan sebuah penalaran yang akan mengevaluasi fakta terlebih dahulu

dan kemudian menghasilkan kesimpulan berdasarkan fakta-fakta yang ada[8]. Pada tiap *rule* yang ada di *database*, akan diuji kondisi benar atau salah dan pada akhirnya akan menghasilkan kesimpulan berdasarkan *rule-rule* yang sudah dimasukkan dalam *database*. Metode penalaran *forward chaining* digunakan dalam tugas akhir ini untuk menentukan *output* dari pertanyaan yang memiliki beberapa kondisi yang dituangkan dalam *rule-rule*. Setelah bertanya dan terdapat *rule* untuk pertanyaan itu, *user* harus menjawab pertanyaan tambahan sesuai *rule* yang ada. Lalu *chatbot* akan menggunakan *inference engine forward chaining* untuk menentukan *output* setelah terlebih dahulu memproses masukan serta *input* pertanyaan tambahan.

F. Prototyping

Prototyping merupakan sebuah paradigma pengembangan perangkat lunak yang menawarkan pendekatan yang paling baik dan membantu pengembang dan *stakeholder* untuk memahami lebih baik apa yang akan dikembangkan saat spesifikasi kebutuhan belum jelas[9]. Spesifikasi kebutuhan yang belum jelas biasanya terjadi ketika *stakeholder* maupun pengembang masih belum bisa mengidentifikasi secara detail spesifikasi yang rinci untuk fungsi-fungsi dan fitur-fitur, algoritma yang akan digunakan, maupun komponen-komponen lainnya seperti desain tampilan dan spesifikasi sistem.



Gambar 1. Metode Prototyping

G. Black Box

Pengujian *black box* merupakan pengujian dimana kasus uji didesain berdasarkan spesifikasi dan berfokus pada *output* yang dihasilkan sebagai respon dari *input* yang dipilih dan kondisi-kondisi eksekusi[10]. Kelebihan dari pengujian *black box* adalah :

1. Penguji tidak memerlukan pengetahuan akan bahasa pemrograman yang digunakan.
2. Membantu menemukan ambiguitas dan ketidakkonsistenan yang terjadi antara penguji dan pengembang.

III. METODOLOGI PENELITIAN

A. Identifikasi Masalah

Identifikasi masalah dari hasil observasi di Unsrat adalah sebagai berikut:

1. Sebagian mahasiswa yang bingung saat pertama

kali menggunakan Portal Akademik Unsrat.

2. Kesulitan untuk mencari informasi mengenai penggunaan SIT Unsrat karena harus pergi ke lokasi tertentu dan pada jam kantor saja.

B. Pengumpulan Data

Metode pengumpulan data yang digunakan peneliti pada penelitian ini terdiri dari :

1. Studi Literatur
Menggunakan panduan yang ada di Buku Panduan Portal Akademik Unsrat.
2. Observasi
Observasi dilakukan dengan melihat keadaan lapangan di Unsrat, khususnya UPT-TIK dan survei di Forum Diskusi Portal Akademik Unsrat.
3. Kuisisioner
Kuisisioner disebarakan kepada 100 mahasiswa Unsrat.
4. Wawancara
Dilakukan wawancara dengan Kepala UPT-TIK Unsrat.

C. Metode Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak yang digunakan adalah metode *Prototyping* yang terdiri dari 5 tahap, yaitu Tahap Komunikasi, Tahap Perencanaan secara cepat, Tahap Pemodelan Perancangan secara cepat, Tahap Pembentukan *Prototype*, Tahap Penyerahan Sistem.

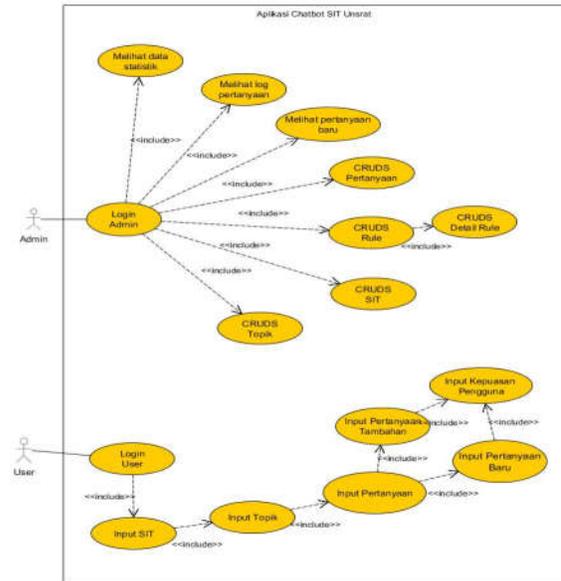
1. Tahap Komunikasi
Pada tahap ini dikumpulkan data melalui kuesioner tentang fitur apa saja pada Portal Akademik Unsrat yang masih sulit untuk digunakan. Pada kuesioner juga terdapat contoh pertanyaan-pertanyaan seputar Portal Akademik. Data juga diambil dari Buku Panduan Portal Akademik Unsrat sebagai panduan dasar untuk pengoperasian.
2. Tahap Perencanaan secara cepat
Pada tahap ini dilakukan perancangan cepat desain *interface* aplikasi *web helpdesk* dan alur kerja program.
3. Tahap Pemodelan Perancangan Secara Cepat
Pada tahap ini dilakukan pemodelan perangkat lunak berdasarkan hasil desain dari tahap dua. Pemodelan tersebut mencakup *Use Case Diagram*, *Flowchart* dan *Entity Relationship Diagram*.
4. Tahap Pembentukan Prototype
Pada tahap ini hasil pemodelan dituangkan dalam *database* dan kode sumber aplikasi.
5. Tahap Penyerahan Sistem/Perangkat lunak Prototype yang berhasil dibangun akan diserahkan kepada *user*. Setelah pengujian selesai, *user* akan memberikan umpan balik mengenai kekurangan aplikasi. Perbaikan *chatbot* dilakukan berdasarkan hasil umpan

balik.

D. Perancangan Perangkat Lunak

Perancangan yang dibuat mencakup *Use Case Diagram*, *Entity Relationship Diagram*, *flowchart* algoritma *bigram*, dan *flowchart* alur dialog *chatbot*.

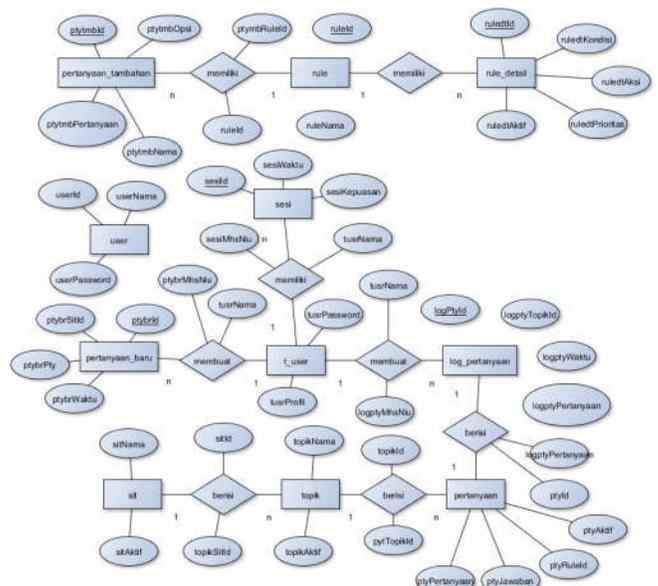
Use Case Diagram



Gambar 2. Use Case Diagram

Dari Gambar 2 bisa dilihat bahwa administrator bisa melakukan manajemen terhadap pertanyaan, *rule*, daftar SIT, topik serta melihat statistik untuk mempermudah penambahan data pada *database chatbot*. *User* hanya bisa menginputkan pertanyaan dan diakhir sesi ada permintaan penilaian kualitas *chatbot*.

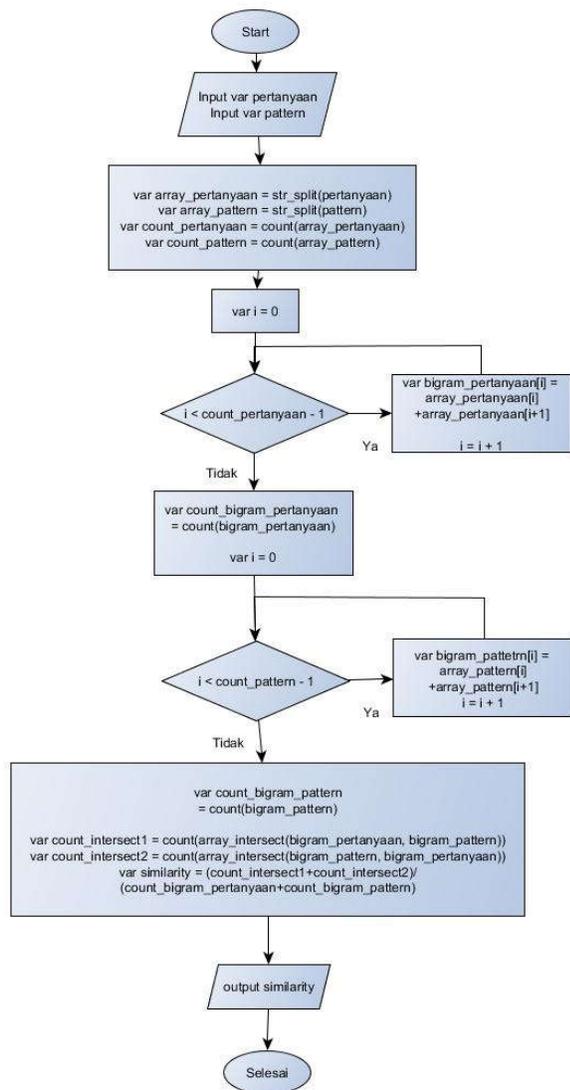
Entity Relationship Diagram



Gambar 3. Entity Relationship Diagram

Pada Gambar 3 dapat dilihat rancangan struktur *database chatbot* yang akan diimplementasikan.

Flowchart Algoritma Bigram



Gambar 4. *Flowchart* algoritma *bigram*

Gambar 4 merupakan *flowchart* algoritma *bigram* yang dibuat berdasarkan persamaan (1).

Langkah awal pada algoritma *bigram* adalah *input* dua variabel *string* yang akan dibandingkan, yaitu pertanyaan dan *pattern*. Variabel pertanyaan merupakan *string* pertanyaan yang diinputkan oleh *user*, sedangkan variabel *pattern* merupakan pertanyaan yang ada di *database*. Kedua *string* tersebut dipecah menggunakan fungsi *str_split* sehingga menghasilkan *array* yang beranggotakan huruf-huruf dari masing-masing *string*.

Sehingga hasilnya adalah dua buah *array* yang berisikan pecahan huruf dari masing-masing *string*, yaitu *array_pertanyaan* dan variabel *array_pattern*. Jumlah anggota masing-masing *array* juga dihitung dan dimasukkan dalam variabel *count_pertanyaan* dan *count_pattern*.

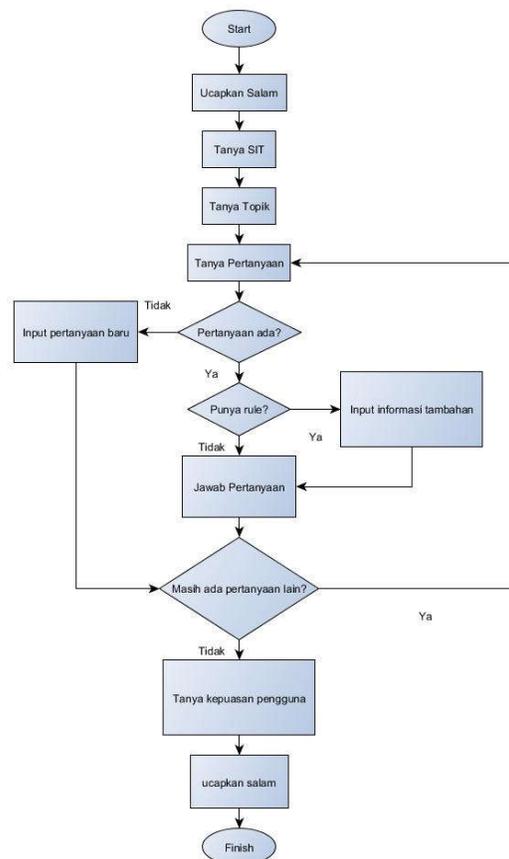
Langkah berikutnya adalah melakukan perulangan sebanyak jumlah anggota *array-1* untuk masing-masing

array. Akan dibuat dua buah *array* baru, yaitu *bigram_pertanyaan* dan *bigram_pattern* yang berisi *bigram* dari *array_pertanyaan* dan *array_pattern*. Jumlah anggota *bigram_pertanyaan* dan *bigram_pattern* juga akan dihitung dan dimasukkan dalam variabel *count_bigram_pertanyaan* dan *count_bigram_pattern*.

Langkah selanjutnya adalah menghitung jumlah irisan antara dua *bigram_pertanyaan* dan *bigram_pattern* yang disimpan dalam variabel *count_intersect1* dan *count_intersect2*. Lalu nilai kemiripan dihitung berdasarkan jumlah *count_intersect1* dan *count_intersect2* dibagi dengan jumlah *count_bigram_pertanyaan* dan *count_bigram_pattern*. Keluaran dari proses ini adalah nilai kemiripan tersebut.

Alur Dialog Chatbot

Chatbot yang dibuat memiliki alur dialog sebagai berikut:



Gambar 5. Alur dialog *chatbot*

IV. HASIL DAN PEMBAHASAN

A. Implementasi Database

Pada tahap ini peneliti membangun database berdasarkan rancangan Entity Relationship Diagram yang telah dibuat.

Tabel 1. Tabel log pertanyaan

log pertanyaan			
Nama Kolom	Tipe Data	Panjang	PK
logptyId	int	11	Ya
logptyTopikId	int	11	

logptyPertanyaan	text		
logptyMhsNiu	varchar	20	
logptyWaktu	datetime		

Tabel 2. Tabel pertanyaan

pertanyaan			
Nama Kolom	Tipe Data	Panjang	PK
ptyId	int	11	Ya
ptyTopikId	int	11	
ptyPertanyaan	text		
ptyJawaban	text		
ptyRuleId	int	11	
ptyAktif	tinyint	1	

Tabel 3. Tabel pertanyaan baru

pertanyaan baru			
Nama Kolom	Tipe Data	Panjang	PK
ptybrId	int	11	Ya
ptybrSitId	int	11	
ptybrTopikId	int	11	
ptybrPty	text		
ptybrMhsNiu	varchar	20	
ptybrWaktu	datetime		

Tabel 4. Tabel pertanyaan tambahan

pertanyaan tambahan			
Nama Kolom	Tipe Data	Panjang	PK
ptytmbId	int	11	Ya
ptytmbRuleId	int	11	
ptytmbNama	varchar	25	
ptytmbPertanyaan	varchar	255	
ptytmbOps	int	255	

Tabel 5. Tabel rule

rule			
Nama Kolom	Tipe Data	Panjang	PK
ruleId	int	11	Ya
ruleNama	varchar	50	

Tabel 6. Tabel rule detail

rule_detail			
Nama Kolom	Tipe Data	Panjang	PK
ruledtId	int	11	Ya
ruledtRuleId	int	11	
ruledtKondisi	varchar	255	
ruledtAksi	text	255	
ruledtPrioritas	tinyint	2	
ruledtAktif	tinyint	1	

Tabel 7. Tabel sesi

sesi			
Nama Kolom	Tipe Data	Panjang	PK
sesiId	varchar	32	Ya
sesiMhsNiu	varchar	20	
sesiWaktu	datetime		
sesiKepuasan	tinyint	1	

Tabel 8. Tabel SIT

Sit			
Nama Kolom	Tipe Data	Panjang	PK
sitId	int	11	Ya
sitNama	varchar	50	
sitAktif	tinyint	1	

Tabel 9. Tabel Topik

Topic			
Nama Kolom	Tipe Data	Panjang	PK
topikId	int	11	Ya
topikSitId	int	11	
topikNama	varchar	50	
topikAktif	tinyint	1	

Tabel 10. Tabel user

user			
Nama Kolom	Tipe Data	Panjang	PK
usrId	int	11	Ya
usrNama	varchar	20	
usrPassword	varchar	255	

Tabel sit memuat daftar aplikasi yang bisa ditanyakan kepada *chatbot*. Tabel topik memuat daftar topik yang tersedia pada masing-masing aplikasi pada tabel sit. Sedangkan tabel pertanyaan memuat daftar pertanyaan yang tersedia pada masing-masing topik. Pada tabel sit, topik dan pertanyaan memiliki kolom untuk menentukan pertanyaan tersebut aktif atau tidak. Sehingga administrator nantinya bisa mengaktifkan atau menonaktifkan suatu SIT, topik, atau pertanyaan.

Pada tabel pertanyaan terdapat jawaban dari masing-masing pertanyaan dan juga *rule*, jika pertanyaan tersebut mempunyai *rule*. Suatu pertanyaan memiliki *rule* apabila pertanyaan tersebut memerlukan informasi tambahan untuk dijawab. Daftar *rule* yang tersedia dapat dilihat pada tabel *rule*. Setiap *rule* memiliki pertanyaan tambahan yang ada pada tabel pertanyaan_tambahan dan harus dijawab oleh *user* dan jawaban tersebut akan diproses berdasarkan kondisi yang ada pada tabel *rule_detail*. Tabel *rule_detail* berisi daftar kondisi jawaban *user* atas pertanyaan tambahan yang mungkin dan juga aksi (jawaban) yang sesuai berdasarkan kondisi yang ada.

Tabel *user* memuat data administrator. Tabel sesi memuat data sesi *login* beserta waktu *login* dan juga

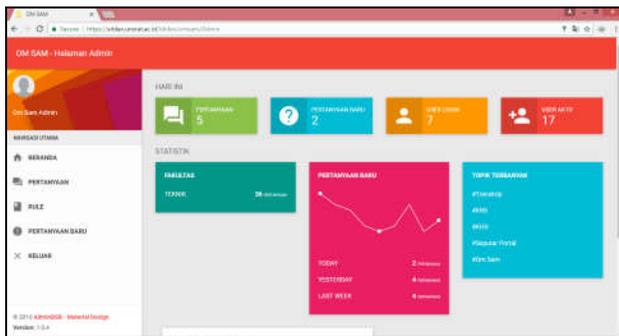
kepuasan dari tiap sesi yang nantinya akan digunakan untuk menilai kualitas *chatbot*. Tabel *t_user* memuat data *user* yang bisa login yang diambil dari *database* *akademika_portal* Unsrat, sehingga setiap *user* dapat login menggunakan username dan password Portal Akademik mereka.

Tabel *log_pertanyaan* memuat data semua pertanyaan yang pernah diajukan oleh *user*. Tabel *log* ini juga berisi topik pertanyaan. Sehingga dari data pada tabel ini dapat terlihat pertanyaan serta topik mana yang paling banyak dan paling sedikit ditanyakan.

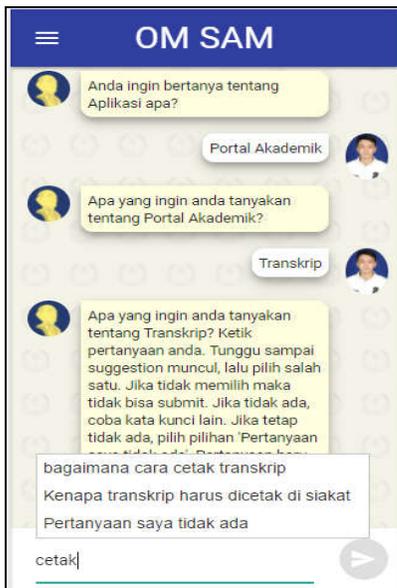
Tabel *pertanyaan_baru* merupakan tabel untuk menampung pertanyaan *user* yang belum ada di tabel pertanyaan. Nantinya pertanyaan yang ada pada *pertanyaan_tambahan* akan ditambahkan oleh administrator ke tabel pertanyaan.

B. Implementasi Sistem

Pada tahap ini, rancangan *flowchart* dan alur dialog *chatbot* dituangkan dalam kode sumber dan dibuat antarmuka, baik untuk administrator maupun untuk *user*.



Gambar 6. Antarmuka administrator



Gambar 7. Antarmuka user

C. Pengujian Sistem

Pada tahap ini digunakan metode pengujian *Black Box*. Disiapkan 16 kasus uji, dan setelah dilakukan pengujian, aplikasi berhasil lolos 100% dari 16 kasus uji tersebut.

Tabel 11. Hasil pengujian *Black Box*

No.	Komponen Pengujian	Masukan	Status
1	Login	User memasukkan NIP/NIM	Diterima
2	Operasi CRUD untuk pertanyaan	Admin menambah, mengubah dan menghapus pertanyaan	Diterima
3	Operasi CRUD untuk rule	Administrator menambah, mengubah dan menghapus rule	Diterima
4	Operasi CRUD untuk SIT	Administrator menambah, mengubah dan menghapus SIT	Diterima
5	Operasi CRUD untuk Topik	Administrator menambah, mengubah dan menghapus topik	Diterima
6	Operasi CRUD untuk user	Administrator menambah, mengubah dan menghapus user	Diterima
7	Log Pertanyaan	User menanyakan pertanyaan	Diterima
8	Pertanyaan tidak terjawab	User menanyakan pertanyaan yang tidak bisa dijawab oleh chatbot	Diterima
9	Statistik penggunaan	User menanyakan pertanyaan baik yang bisa dijawab maupun tidak	Diterima
10	Kebenaran jawaban	User menanyakan pertanyaan	
10.1		Siapa nama kamu?	Diterima
10.2		Siapakah nama anda???	Diterima
10.3		bagaimana cara menghapus mata kuliah di krs	Diterima
10.4		bagaimana cara menambah mata kuliah di krs	Diterima
10.5		bagaimana cara print khs	Diterima

V. PENUTUP

A. Kesimpulan

Berdasarkan analisis, perancangan, implementasi dan pengujian yang telah dilakukan terhadap aplikasi *chatbot*, dapat diambil kesimpulan sebagai berikut :

1. Aplikasi *Chatbot* sebagai *helpdesk* telah berhasil dibangun dengan menggunakan algoritma *Bigram* dan *Forward Chaining*.
2. Aplikasi *Chatbot* dapat menjawab pertanyaan sesuai dengan yang diharapkan (pertanyaan yang sudah ada dalam *database*), sehingga dapat membantu *user* untuk menjawab pertanyaan seputar Portal Akademik.

B. Saran

1. Agar pihak universitas dapat dengan cepat dan tepat tanggap menambah pengetahuan *chatbot* apabila ada pertanyaan yang tidak bisa terjawab, karena keberhasilan *chatbot* sangat bergantung pada hal itu, sehingga perlahan-lahan pengetahuannya akan lebih baik.
2. Untuk penelitian selanjutnya agar menambahkan *dialog manager* sehingga lebih mudah mengatur jalannya percakapan serta respon-respon standar yang dihasilkan *chatbot*.
3. Menambahkan basis pengetahuan untuk SIT dan topik lainnya.
4. Melakukan pengujian di Unsrat.

DAFTAR PUSTAKA

- [1] Manopo, R. I. 2016. "Perancangan Aplikasi Help Desk di UPT-TIK Unsrat", Tekno. vol. 8. no.1, 2016.
- [2] Gibson, Darril. 2015. *Effective Help Desk Specialist Skills*. USA: Pearson Education, Inc.
- [3] Shawar, A. & Atwell, B. 2007. "Chatbot: Can They Serve as Natural Language Interfaces to QA Corpus?", LDV Forum 2007 – Band 22(1) 31-50.
- [4] Britz, Denny. 2016. "Deep Learning for Chatbots, Part 1 – Introduction" [Online]. Available: <http://www.wildml.com/2016/04/deep-learning-for-chatbots-part-1-introduction>. Diakses tanggal 12 Juni 2017.
- [5] Dewi, A. & Setiaji, B. 2014. "Pemanfaatan Sentence-Similarity Measurement untuk Proses Pencarian Pola pada Chatbot Berbasis Pattern-Matching," Seminar Nasional Teknologi Informasi dan Multimedia 2014, ISSN : 2302-3805, 8 Februari 2014.
- [6] Oxford Living Dictionaries [Online]. Available: <https://en.oxforddictionaries.com/definition/bigram>. Diakses tanggal 12 Juni 2017
- [7] Singh, Swapna & Karwayun, Ragini. 2010. "A Comparative Study of Inference Engines" Seventh International Conference on Information Technology 2010.
- [8] Baiti, Zifora., Nugroho, Fresy. 2013. "Aplikasi Chatbot IM3 untuk Informasi Jurusan Teknik Informatika Berbasis Sistem Pakar menggunakan Forward Chaining", MATICS vol.5 no.3, September 2013.
- [9] Pressman, Roger. 2012. *Rekayasa Perangkat Lunak 7th ed.* Yogyakarta: ANDI.
- [10] Nidhra, Srinivas dan Dondeti, Jagruthi. 2012, *Black Box And White Box Testing Techniques –A Literature*

Review, International Journal of Embedded Systems and Applications (IJESA) Vol.2, No.2, 8-9.

- [11] Wickramanayake, Kamal. "PhpExpertSystem" [Online]. Available: <http://www.deadschool.com/phpexpertsystem>. Diakses tanggal 12 Juni 2017.

SEKILAS TENTANG PENULIS

Saya bernama Ruspandi Reynaldi Benedictus. Lahir pada tanggal 03 Oktober 1994 di Manado.

Saya mulai menempuh pendidikan di SD Frater Don Bosco Manado (1999-2006). Kemudian melanjutkan ke SMP Frater Don Bosco Manado (2006-2009). Setelah itu saya menempuh pendidikan di SMA Frater Don Bosco Manado (2009-2012). Setelah lulus, di tahun 2012 saya melanjutkan pendidikan di Universitas Sam Ratulangi Manado, mengambil Program Studi S-1 Teknik Informatika di Jurusan Elektro Fakultas Teknik.

