



## Implementasi Sistem Tanya Jawab Berbasis Skenario untuk Mendukung Proses Akademik dengan IBM Watson Assistant

Bryan Wijaya<sup>#1</sup>, Hapnes Toba<sup>#2</sup>

<sup>#</sup>Fakultas Teknologi Informasi, Universitas Kristen Maranatha  
Jl. Suria Sumantri No. 65 Bandung 40164, Jawa Barat, Indonesia

<sup>1</sup>bryan.wijaya@outlook.co.id

<sup>2</sup>hapnestoba@it.maranatha.edu

**Abstrak**— Dalam makalah ini disampaikan sebuah hasil penelitian dengan memanfaatkan teknologi dari IBM, yaitu *Watson Assistant*. *Watson Assistant* digunakan untuk membuat *chatbot* terkait proses akademik. Analisis dan pengumpulan data dilakukan dengan berbasiskan skenario. Data-data tersebut dibuat ke dalam sebuah *graph search*. *Watson Assistant* akan menentukan *node* dengan nilai kepercayaan tertinggi untuk diberikan sebagai jawaban. Skenario percakapan yang ditanamkan dalam *chatbot* ini telah diimplementasikan ke dalam bentuk laman *web*, Facebook Messenger, dan Slack untuk membantu interaksi antara pihak fakultas dengan mahasiswa. *Chatbot* berperan pula sebagai sistem pendamping forum tanya jawab di dalam *course learning system (CLS)* untuk pertanyaan-pertanyaan rutin. Berdasarkan hasil uji coba, *chatbot* berbasis skenario telah dapat menjawab kebutuhan dasar mahasiswa untuk bertanya seputar hal akademis, sebagaimana tercantum dalam buku panduan, khususnya untuk proses perwalian dan deskripsi mata kuliah.

**Kata kunci**— *belief engine, chatbot, google translate, sistem tanya jawab, watson assistant*

**Abstract**— *Watson Assistant* is a technology offered by IBM which could be implemented as *chatbot* to answer questions from users. *Watson Assistant* requires data in its learning. The data processing is based on scenarios. The data is stored into nodes in the form of *assertion graph*. In every node the value of confidence will be calculated, therefore *Watson Assistant* will determine the highest value of confidence to be given as an answer. Our research has successfully implemented the integration of *Watson Assistant* in the faculty website, Facebook Messenger, and Slack to support students' activities, especially in academic processes. The *chatbot* has also shown its ability as a supporting tool for the regular forum question answering activities in *course learning system (CLS)*.

**Keywords**— *belief engine, chatbot, google translate, question answering system, watson assistant*

### I. PENDAHULUAN

Proses akademik terkadang menjadi sebuah hal yang cukup kompleks bagi beberapa mahasiswa/i, contohnya pada saat perwalian. Seringkali pada saat mahasiswa/i tersebut membutuhkan arahan dari dosen terdapat beberapa kendala. Misalnya, ketika mahasiswa/i ingin memperoleh panduan mata kuliah yang harus diambil dalam perencanaan studi dan dosen sedang tidak berada di tempat, maka informasi tidak dapat diperoleh secara cepat dan menghambat proses selanjutnya.

Di sisi lain, berbagai perkembangan ilmu dan teknologi menghadirkan berbagai alternatif baru yang memungkinkan kemudahan dalam berinteraksi [1]. Salah satunya adalah alternatif untuk mengotomatiskan sebuah proses percakapan sehingga dapat mempercepat kinerja [2]. Terkait dengan hal itulah maka dalam penelitian ini diangkat sebuah rumusan masalah, yaitu: bagaimana menghasilkan sebuah sistem yang dapat menjawab pertanyaan-pertanyaan mendasar dari mahasiswa/i untuk mendukung proses akademik secara cepat dan tepat.

Tujuan utama dari penelitian ini adalah menghasilkan sistem yang memungkinkan interaksi secara otomatis antara mahasiswa/i dengan pihak fakultas dalam menemukan informasi terkait proses akademik. Otomatisasi dilakukan melalui percakapan berbasis skenario dengan dukungan teknologi *Watson Assistant* dari IBM.

### II. KAJIAN LITERATUR

#### A. IBM Watson & IBM Watson Services

IBM Watson adalah sistem komputer yang menerapkan pemrosesan bahasa alami dan konten yang bagus dalam kompetisi sehingga cukup cepat untuk bersaing dan bahkan dapat menang melawan manusia dalam kontes Jeopardy [1]. IBM telah menyediakan sebuah layanan bagi pengguna sehingga dimungkinkan untuk membuat sistem interaksi

percakapan secara otomatis dengan menggunakan IBM Watson *Conversation Service*.

Dengan layanan ini perangkat lunak dapat diarahkan untuk mengerti masukan dalam bahasa alami dan berkomunikasi dengan pengguna seperti halnya percakapan antar manusia. *Conversation Service* sangat bergantung pada proses pembelajaran mesin untuk merespons pengguna dalam mensimulasikan percakapan antar manusia. Melalui layanan ini beberapa perusahaan besar telah memanfaatkan IBM Watson dalam proses interaksi dengan para pelanggannya, seperti misalnya: Coca-Cola, Thomson Reuters, dan Honda [2].

### B. Watson Assistant

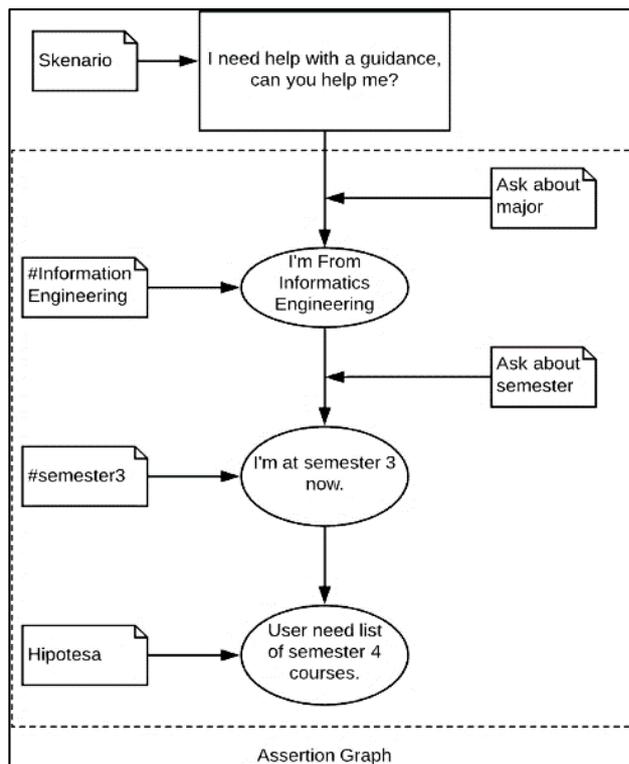
*Watson Assistant* adalah perangkat yang ditawarkan oleh IBM Watson untuk membuat antarmuka pengguna berupa percakapan ke dalam aplikasi ataupun media lainnya. *Watson Assistant* menggunakan *belief computation* untuk menentukan keputusan dalam sebuah *assertion graph*. *Watson Assistant* dapat mencari beberapa jalur sekaligus dengan melihat probabilitas dari setiap jalur dalam *assertion graph* [3], [4], [5]. Bentuk visualisasi *assertion graph* tergambar seperti pada Gambar 1.

Visualisasi *assertion graph* pada Gambar 1 menjelaskan mengenai skenario perwalian yang diajukan oleh seorang mahasiswa. Pada skenario awal dicontohkan seorang mahasiswa membutuhkan bantuan dalam perwalian dan menanyakan kepada sistem “*I need help with a guidance, can you help me?*”. Sistem akan memberikan respons dengan menanyakan kepada pengguna berada di program studi apa, untuk kemudian diarahkan pada program studi yang sesuai.

Setelah itu, sistem akan menanyakan kepada pengguna ada di semester berapakah dia sekarang, untuk menentukan semester. Hal ini dilakukan untuk mengarahkan hipotesis yang pengguna butuhkan. Dalam kasus ini pengguna membutuhkan daftar mata kuliah semester 4 (semester selanjutnya pada saat perwalian) sebagai domain skenario pengetahuan.

Terkait dengan pengembangan skenario dalam *assertion graph*, terdapat beberapa sub-komponen untuk mendukung pembentukan hipotesis dalam sistem, yaitu:

- *Node*: sebuah simpul yang menjadi penghubung / koneksi dalam rangkaian dialog.
- *Intent*: dalam *Watson Assistant* digunakan untuk menentukan niat atau tujuan yang ingin dicapai.
- *Entity*: dalam *Watson Assistant* digunakan untuk menentukan hipotesis jawaban dan dapat pula menyimpan pengetahuan yang dibutuhkan oleh *chatbot* sebelum menjawab pertanyaan.

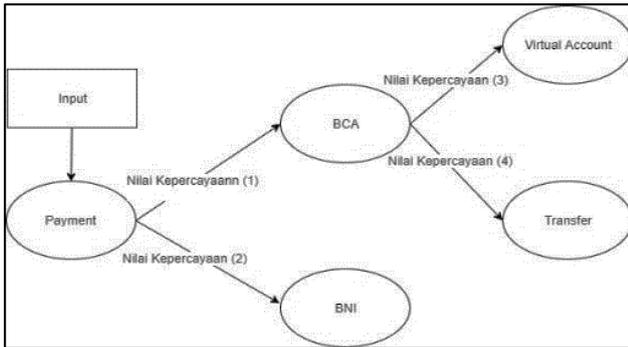


Gambar. 1 Contoh *assertion graph* pada *watson assistant*

### C. Belief Engine

Salah satu pendekatan untuk menentukan keputusan dari beberapa alternatif hipotesis dalam *assertion graph* adalah *probabilistic inference* melalui sebuah *graphical model* [3]. *Watson Assistant* merujuk ke komponen ini sebagai *belief engine*. Meskipun tujuan utama dari mesin ini adalah untuk menyimpulkan kepercayaan terhadap hipotesis, tetapi juga memiliki tujuan sekunder yaitu untuk menyimpulkan kepercayaan pada simpul yang tidak diketahui yang juga bukan hipotesis. *Belief engine* perlu memberikan nilai kepercayaan pada setiap simpul, bukan hanya hipotesis. Setiap *node* mewakili pernyataan, jadi bisa salah satu dari dua pernyataan: benar atau salah.

Sebuah *graphical model* dengan *node* sejumlah  $k$ , dapat memberikan kombinasi  $2^k$  kemungkinan pernyataan. Inferensi probabilistik pada graf akan digunakan untuk menentukan besar nilai kemungkinan dari masing-masing pernyataan tersebut. *Belief engine* menggunakan nilai kemungkinan ini untuk menghitung probabilitas marjinal pada setiap *node* yang berpotensi benar. Probabilitas marjinal inilah yang kemudian dianggap sebagai nilai kepercayaan.



Gambar. 2 Contoh Belief Engine dalam Pembentukan Hipotesis Skenario

Sebagai contoh, visualisasi *belief engine* pada Gambar 2 menjelaskan mengenai pembayaran yang memiliki 2 metode pembayaran.

Jika pengguna memasukkan jawaban terhadap pertanyaan pembayaran, maka *belief engine* tersebut akan menghitung nilai kepercayaan terhadap setiap *node* yang ada. *Node* dengan nilai kepercayaan tertinggi akan menjadi jawaban selanjutnya dari sistem. Jika terdapat lebih dari sebuah *parent node* maka perhitungan nilai kepercayaan dapat menggunakan kombinasi dari *noisy-OR* dengan menggunakan formula 1 [3]. Dalam formula 1 dapat dilihat bahwa nilai probabilistik sebuah *node* dihitung dengan mengambil nilai “bersih” setelah probabilistik sebuah *node* dikurangi dengan hasil perkalian seluruh *node* yang terkait sebelumnya (*chain rule*).

$$(1 - \prod_{r \in R} (1 - r)) = \bigoplus_{r \in R} r \quad \dots (1)$$

Dengan:

- $r$  : himpunan *node* yang terhubung
- $\prod_{r \in R} (1 - r)$  : hasil agregasi (perkalian) antara himpunan *node* yang terhubung
- $\bigoplus r$  : nilai probabilitas marjinal

Probabilitas dari masing-masing *node* dapat dihitung menggunakan formula 2. Sebagai contoh pada kasus yang terdapat pada Gambar 2, *parent node* hanya terdapat 1. Maka nilai  $r$  adalah  $(1 - (1 - 1))$ ,  $r = 1$ . Jika nilai kepercayaan (1) adalah 0.8 dan nilai kepercayaan (2) adalah 0.3, maka nilai probabilitas untuk menjawab BCA adalah  $0.8 * 1 = 0.8$  dan BNI adalah  $0.3 * 1 = 0.3$ . Untuk menghitung probabilitas pada *node Virtual Account* dan *Transfer*, nilai  $r$  haruslah dihitung berdasarkan aturan *Bayesian Chain Rule* [3]. Dengan demikian, nilai  $r$  untuk *Virtual Account* adalah  $(1 - ((1 - 0.8) * (1 - 0.7))) = 0.94$ .

$$P(x|R_x, Q_x) = \left[ \bigoplus_{r \in R_x} (s_r p_r) \right] \left[ 1 - \bigoplus_{q \in Q_x} (s_q p_q) \right] \quad \dots (2)$$

Dengan:

$P(x|R_x, Q_x)$  : *Bayesian chain rule* untuk *node x* dengan diketahui *node R* dan *Q* yang terhubung dengan  $x$

$\bigoplus (s_x, p_x)$  : nilai probabilitas marjinal seperti pada formula 1

#### D. Respons Watson Assistant

Respons yang diberikan oleh *Watson Assistant* disampaikan dalam bentuk JSON. Terdapat beberapa atribut penting dalam struktur JSON tersebut. Pertama adalah “*output*” memiliki empat sub-atribut lainnya, yaitu: “*generic*”, “*debug*”, “*intents*”, dan “*entities*” (lihat Gambar 3). Sebelum dapat membuka sub-elemen diperlukan akses terhadap “*output*” terlebih dahulu.

Di dalam atribut “*generic*” terdapat dua atribut terkait yaitu “*response\_type*” dan “*text*”. “*Response\_type*” digunakan untuk mengetahui bentuk respons apa yang diberikan. Atribut “*text*” digunakan untuk mengetahui isi dari responsnya (dapat dilihat dalam Gambar 4), yang akan diakses untuk mendapatkan jawaban dari *Watson Assistant*.

Ketiga atribut lain “*debug*”, “*intents*”, dan “*entities*” digunakan untuk pengembangan dalam mencari dan mengurangi “*bug*”. “*Debug*” dapat digunakan untuk melihat *node* apa yang dikunjungi. Atribut “*intents*” dapat digunakan untuk melihat perhitungan nilai kepercayaan terhadap *intents* dan *intent* apa yang dipilih oleh *Watson Assistant*. Atribut “*entities*” digunakan untuk melihat *entity* apa yang masuk atau yang terbaca oleh *Watson Assistant*.

```

1  {
2  |   "output": {
3  |     |   "generic": [ -
8  |     |     |
9  |     |     |   "debug": { -
18 |     |     |   },
19 |     |     |   "intents": [ -
60 |     |     |   ],
61 |     |     |   "entities": []
62 |     |   }
63 |   }

```

Gambar. 3 Respons dari *Watson Assistant* melalui *Application Programming Interface (API)*

#### E. Korelasi Watson Assistant dengan Sistem Pakar

Menurut definisinya, sebuah sistem pakar dapat dinyatakan sebagai sebuah sistem yang mampu memberikan solusi untuk sebuah permasalahan tertentu ataupun untuk memberikan saran yang disusun berdasarkan pengetahuan para ahli dalam bidangnya [6]. Pada dasarnya *Watson Assistant* sudah menyediakan layanan tersebut melalui mekanisme tanya jawab sehingga dapat memberikan arahan untuk kebutuhan pengguna berdasarkan data yang sudah dipersiapkan, dan dengan demikian dapat dianalogikan sebagai sebuah sistem pakar.

Secara umum sebuah sistem pakar tersusun atas komponen: antarmuka sistem untuk berinteraksi, mesin inferensi, dan sumber pengetahuan. Di dalam *Watson Assistant*, terdapat antarmuka sistem yang dapat digunakan oleh pengguna, serta untuk terhubung dengan dengan sistem-sistem pihak ketiga. Adapun mesin inferensi yang digunakan dalam *Watson Assistant* diwujudkan melalui *belief engine*, dengan sumber pengetahuan yang dibentuk melalui struktur *assertion graph*.

F. Chatbot

Teknologi *chatbot* sangat berkembang pesat dalam beberapa tahun terakhir ini. Misalnya dalam sistem Alexa dapat mematikan lampu rumah hanya dengan cara kita memerintahnya melalui suara. Dalam kasus *chatbot*, dengan diluncurkannya *bot* pada *platform* seperti pada Slack ataupun Facebook Messenger dapat membuat pertumbuhan pembuatan *bot* menjadi lebih cepat [7]. Dengan perkembangan pesat perangkat teknologi informasi, *chatbot* dapat diarahkan untuk memberikan layanan asistensi kepada pengguna. Misalnya *chatbot* dapat digunakan untuk memberikan layanan *customer service* 24 jam, yang biasa dilakukan oleh manusia dalam waktu terbatas.

*Chatbot* dapat pula menjadi pendukung layanan *Frequently Ask Questions* (FAQ) untuk mendapatkan kandidat-kandidat jawaban terbaik [5]. Dalam penelitian

yang dibahas dalam makalah ini, *chatbot* digunakan sebagai penyedia informasi dan melengkapi buku panduan akademik.

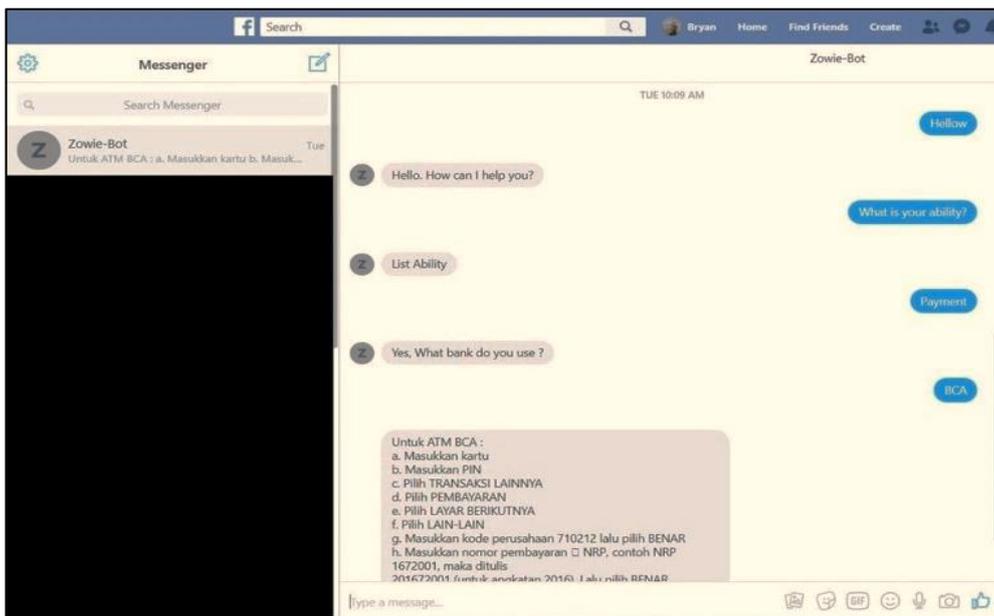
Dengan penetrasi sosial media dan konektifitas dari internet disertai dengan kemajuan dalam pemrosesan bahasa alami serta kecerdasan buatan, *chatbot* diharapkan dapat mendominasi pasar. Seorang *developer* perlu untuk memahami apa yang *chatbot* dapat tawarkan dan kategori apa yang masuk ke dalam *chatbot* tersebut. Pengetahuan tersebut dapat membantu dalam memilih algoritma atau *platform* dan alat yang tepat dalam membuat *chatbot* [7].

```

1  {
2    "output": {
3      "generic": [
4        {
5          "response_type": "text",
6          "text": "Hello. How can I help you?"
7        }
8      ],
9      "debug": { --
10     },
11     "intents": [ --
12     ],
13     "entities": []
14   }
15 }

```

Gambar. 4 Respons dari Watson berisi konten jawaban



Gambar. 5 Integrasi Chatbot Facebook Messenger dan Watson

Gambar 5 menunjukkan sebuah contoh percakapan melalui *chatbot* untuk menjelaskan proses pembayaran melalui ATM. Dapat dilihat pula dalam Gambar 5 tersebut bahwa *chatbot* tersebut - yang sudah diimplementasikan menggunakan

*Watson Assistant* - dapat terintegrasi dengan Facebook Messenger. Sebagai sapaan awal dalam percakapan tersebut, seorang pengguna (dengan warna biru) memberikan salam

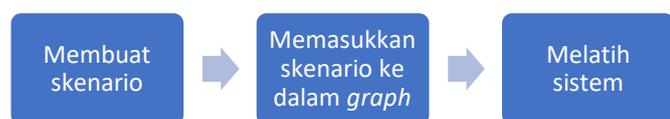
“Hellow”. *Chatbot* langsung memberi tanggapan dengan memberikan salam “Hello. How can I help you?”.

III. ANALISIS & RANCANGAN SISTEM TANYA JAWAB

A. Metodologi

Secara garis besar metode kerja yang akan dilakukan dalam penelitian ini tergambarkan dalam bentuk skema seperti pada Gambar 6.

1. Dalam tahapan menganalisis skenario terdapat beberapa bagian, yaitu:
  - Memahami kebutuhan yang diperlukan. Dalam konteks ini, peneliti menentukan kebutuhan mahasiswa/i, misalnya mengenai proses akademik yang akan diimplementasikan ke dalam sistem.
  - Mengumpulkan informasi yang relevan dan merancang skenario terkait kebutuhan tersebut.
2. Dalam tahapan memasukkan skenario ke dalam *graph* terdapat beberapa bagian, yaitu :
  - Setelah skenario dianggap telah mewakili kebutuhan, maka skenario tersebut dimasukkan ke dalam *graph* pada *engine* yang terdapat dalam IBM Watson.
  - Menjadikan skenario-skenario tersebut menjadi *node-node* pada *graph* tersebut.
3. Dalam tahapan melatih sistem terdapat beberapa bagian, yaitu:
  - Melatih dengan memberikan pertanyaan-pertanyaan yang sudah ada maupun yang belum ada dalam skenario agar melihat hasil yang diberikan sesuai atau tidak.
  - Jika sudah dirasa cukup, *chatbot* kemudian di-*deploy* ke dalam laman *web*, sehingga kemudian dapat diuji coba oleh pengguna.
4. Jika dirasa masih terdapat kekurangan maka rangkaian proses ini dapat dilakukan kembali dari awal.



Gambar. 6 Metode kerja pengembangan sistem tanya jawab

B. Perancangan Skenario Sistem Tanya Jawab

Adapun perancangan skenario uji coba dalam penelitian ini disampaikan pada Tabel 1. Level 1 digunakan oleh skenario untuk pertanyaan-pertanyaan yang dianggap paling umum, disebut pula sebagai domain deskriptif.

Level 2 dan 3 digunakan oleh skenario yang bersifat prosedural. Penting untuk diketahui pula bahwa *level* di dalam *Watson Assistant* tidak dibatasi dan dapat dikembangkan sesuai kebutuhan skenario.

TABEL I  
CONTOH SKENARIO SISTEM

No	Level 1	Level 2	Level 3
1	Welcome	Thanks	-
2	Joke	#General_About_You	#General_About_You
3	Security	-	-
4	Guidance	Informatics Engineering	Semester3
5	Payment	BCA	Virtual Account
		BNI	Transfer
6	Closing	-	-
7	Ability	-	-
8	Anything Else	-	-

C. Perancangan Pemanfaatan Sistem

Perancangan pemanfaatan sistem tergambar dengan *use case diagram* seperti pada Gambar 7. Di dalam sistem terdapat setidaknya 2 aktor, yaitu pengguna (mahasiswa dan dosen), serta administrator. Pengguna inilah yang dapat melakukan komunikasi dengan menggunakan skenario pada *bot*.

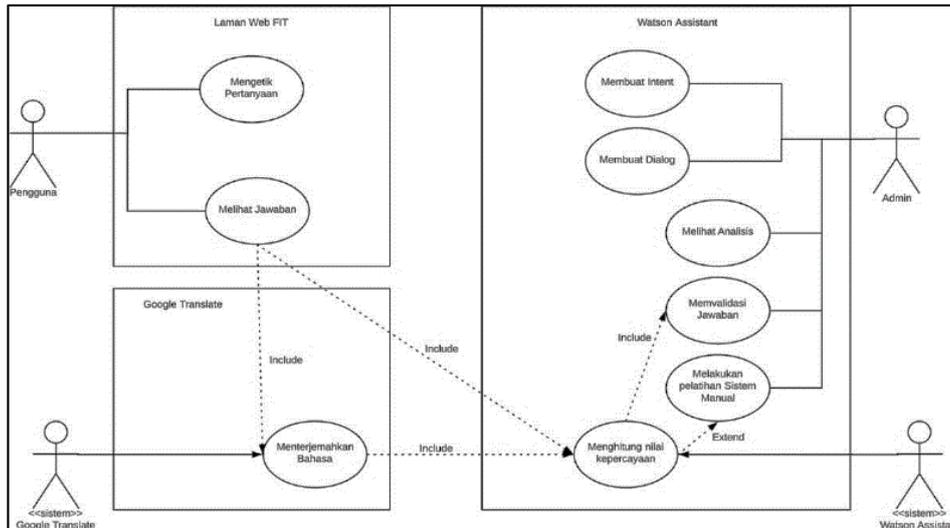
D. Sequence Diagram dalam Pengetikan Pertanyaan dan Melihat Jawaban

Perancangan *sequence diagram* dari *use case* dapat dilihat pada Gambar 8. Seorang admin / pengguna mengajukan pertanyaan dalam bahasa Inggris ataupun dalam bahasa Indonesia (penerjemahan dilakukan di dalam sistem) melalui tempat percakapan yang telah tersedia. Data pernyataan tersebut akan diteruskan kepada mesin *Watson Assistant* dan ditelusuri di dalam *graph* menggunakan *belief engine* yang sudah dijelaskan pada bagian sebelumnya. Setelah mendapat jawaban, jawaban terbaik menurut sistem akan dikirimkan kepada pengguna. Komunikasi akan terus berlangsung hingga admin / pengguna selesai mengajukan pertanyaan.

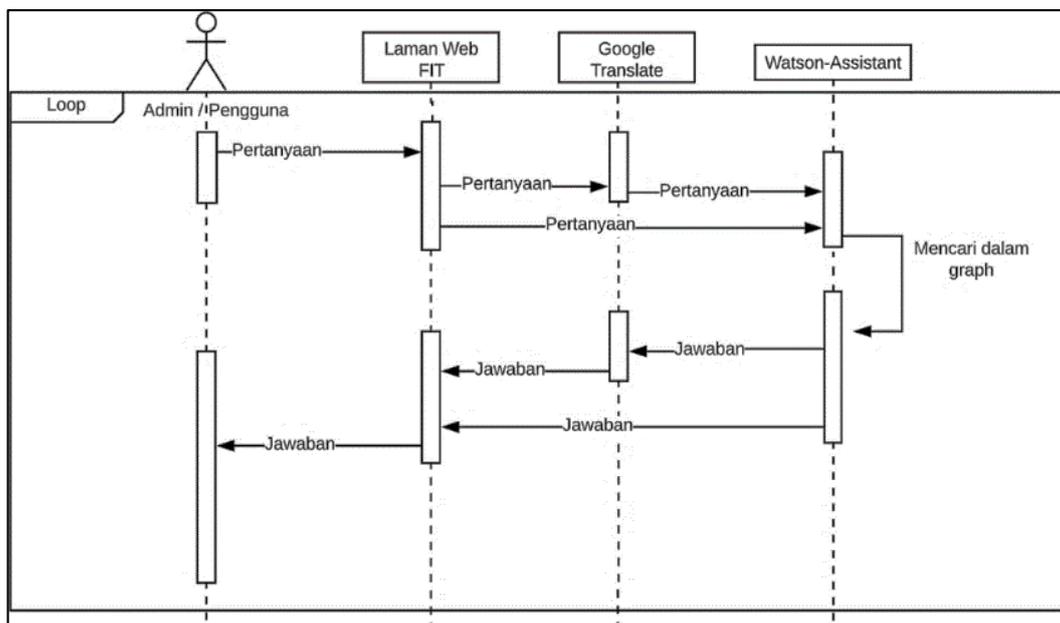
Dalam *use case* pada Gambar 7, terdapat *case* untuk menerjemahkan bahasa yang dilakukan oleh sistem *Google Translate*, sebagai salah satu sistem penerjemah bahasa dengan performa baik saat ini [8]-[9]. Terdapat pula *case* untuk menghitung nilai kepercayaan yang dilakukan oleh *Watson Assistant*. *Watson Assistant* akan menerima “message” atau dalam prakteknya adalah pertanyaan dari pengguna.

Mesin pencari dalam *Watson Assistant (belief engine)* akan mencari apakah ada *intents* atau *entities* yang dituju. Jika ada, maka akan mencari dialog yang sesuai dengan *intents* atau *entities* tersebut dan kemudian memberikan jawaban kembali. Jika tidak ada kandidat jawaban terbaik, maka akan mengembalikan jawaban *default* sesuai jalur atau skenario yang telah dimasukkan sebelumnya.

Ekosistem IBM Watson menyediakan layanan untuk penerjemahan bahasa menggunakan *Language Translator* namun pada saat pembuatan riset ini masih belum menyediakan penerjemahan langsung ke dalam Bahasa Indonesia. Oleh sebab itu, mesin penerjemah yang akan digunakan adalah *Google Translate*. *Google Translate* dapat digunakan melalui *Google Cloud Platform*. Terdapat pula alternatif lain, yaitu dengan mengunduh *library* berkode sumber terbuka, seperti misalnya pustaka “*PHP Google Translate Free*” [10].



Gambar. 7 Perancangan Use Case Integrasi Aplikasi Web dan Chatbot Watson Assistant



Gambar. 8 Sequence Diagram Pemanfaatan Sistem Secara Keseluruhan

#### IV. IMPLEMENTASI

##### A. Garis Besar Pemanfaatan Teknologi

Secara garis besar unsur-unsur teknologi yang digunakan dalam implementasi *chatbot* ini tergambar dalam bentuk diagram seperti pada Gambar 9. *Knowledge* adalah pengetahuan mengenai data yang diperlukan untuk menjawab pertanyaan yang ada. *Backend* adalah lokasi penyimpanan data yang tidak dapat diakses langsung oleh pengguna.

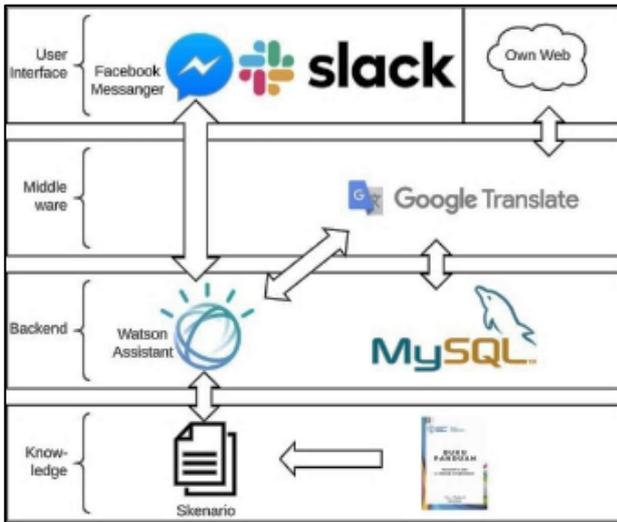
*Middleware* adalah sebuah perangkat lunak yang terdapat pada aplikasi yang memiliki tujuan tertentu dan bekerja secara tersembunyi. *User interface* adalah perangkat lunak yang dapat diakses langsung oleh pengguna.

##### B. Skenario

Uji coba skenario dalam penelitian ini dilakukan dengan memetakan buku panduan akademik mahasiswa ke dalam sebuah rangkaian skenario. Rangkaian skenario dibutuhkan untuk menentukan alur percakapan di dalam *Watson Assistant*.

Skenario dibuat untuk pengguna *chatbot* agar mendapatkan jawaban yang mereka butuhkan. Sebagai contoh pada kasus seperti Gambar 10, pengguna ingin mengetahui penjelasan mengenai mata kuliah Pemograman Game 2D.

Kasus seperti dalam Gambar 10 adalah salah satu kasus yang mudah untuk dideteksi oleh *chatbot* karena langsung menuju inti pertanyaan. Namun pada kenyataan sehari-hari percakapan tidak berjalan seperti pada kasus tersebut. Oleh karena itu, diperlukan *engine Watson Assistant* untuk memudahkan dalam melakukan pencarian dalam struktur data *graph* yang sudah dijelaskan pada bagian Kajian Literatur. Sebagai contoh percakapan yang tidak langsung menuju pada inti pertanyaan terlihat pada Gambar 11.



Gambar. 9 Pemanfaatan teknologi dalam sistem yang dikembangkan



Gambar. 10 Contoh skenario dengan penjelasan mata kuliah dengan hasil penerjemahan *Google Translate*

Skenario dibuat untuk pengguna *chatbot* agar mendapatkan jawaban yang mereka butuhkan. Sebagai contoh pada kasus seperti Gambar 10, pengguna ingin mengetahui penjelasan mengenai mata kuliah Pemograman Game 2D.

Kasus seperti dalam Gambar 10 adalah salah satu kasus yang mudah untuk dideteksi oleh *chatbot* karena langsung menuju inti pertanyaan. Namun pada kenyataan sehari-hari percakapan tidak berjalan seperti pada kasus tersebut. Oleh karena itu, diperlukan *engine Watson Assistant* untuk memudahkan dalam melakukan pencarian dalam struktur data *graph* yang sudah dijelaskan pada bagian Kajian Literatur. Sebagai contoh percakapan yang tidak langsung menuju pada inti pertanyaan terlihat pada Gambar 11.



Gambar. 11 Contoh Skenario yang memerlukan perhitungan *Belief Engine*

### C. Pengembangan Aplikasi Chatbot

IBM Watson tidak dapat menyimpan *Knowledge* dalam Bahasa Indonesia, oleh karena itu pada laman *web* diimplementasikan “*Google Translate*” sebagai penterjemah dari Bahasa Indonesia ke dalam bahasa Inggris dan sebaliknya.

*Deployment chatbot* dapat dilakukan ke dalam beberapa *platform* yang sudah bekerja sama dengan IBM. Beberapa diantaranya adalah Facebook Messenger, Slack, dan juga dapat diimplementasikan ke dalam laman *web* yang dibuat sendiri dengan memanfaatkan *API*.

## V. PENGUJIAN

Pengujian sistem tanya jawab dilakukan dalam beberapa skenario, yang mencakup pada: pengujian domain deskriptif dan prosedural, pengujian terhadap skenario negatif (penyangkalan), penelusuran *node*, pengujian terhadap hasil penerjemahan, pengujian hasil pembelajaran dalam *belief engine*, dan pengujian *assertion graph*.

Pengujian dilakukan dengan cara memasukkan percakapan menggunakan aplikasi “*Postman*” untuk dapat melihat respons dalam bentuk JSON (*JavaScript Object Notation*) [11]. Bentuk respons terlihat seperti pada Gambar 3 dan 4. Pengujian dilakukan untuk menguji dialog

dengan domain yang bersifat deskriptif dan prosedural. Pengujian juga dilakukan guna untuk mendapatkan kandidat jawaban berkualitas yang diharapkan. Pengujian berikutnya dilakukan terhadap dialog dengan *intents* dan *entities* yang sudah dipersiapkan sebelumnya, dengan harapan bahwa *intent* dan *entity* yang ditelusuri oleh *Watson Assistant* sesuai dengan skenario yang ditanamkan.

#### A. Domain Deskriptif

Pengujian domain deskriptif ditujukan untuk melihat berapakah nilai kepercayaan yang dihasilkan oleh *Watson-assistant* pada tiap-tiap *node* yang diuji [12]. Berikut ini adalah skenario percakapan dalam uji coba mulai dari tahap penyapaan sampai, memberikan informasi tentang deskripsi mata kuliah, dan penutup percakapan.

1. Skenario : Menyapa dan mengaktifkan *bot*.  
=> Masukan : *hai*  
=> *Intent* yang diharapkan : *#General\_greetings*  
=> *Intent* yang sesungguhnya : *#General\_greetings*  
=> Nilai kepercayaan : 0.315961
2. Skenario : Mengetahui kemampuan *bot*.  
=> Masukan : *What is your ability?*  
=> *Intent* yang diharapkan : *#Ability*  
=> *Intent* yang sesungguhnya : *#Ability*  
=> Nilai kepercayaan : 1
3. Skenario : Mengetahui deskripsi perkuliahan.  
=> Masukan : *Can you explain me about basic programming?*  
=> *Intent* yang diharapkan : *#Basic\_programming*  
=> *Intent* yang sesungguhnya : *#Basic\_programming*  
=> Nilai kepercayaan : 0.938578
4. Skenario : Mengetahui deskripsi perkuliahan.  
=> Masukan : *What do you know about 2d game prog?*  
=> *Intent* yang diharapkan : *#2d\_game\_programming*  
=> *Intent* yang sesungguhnya : *#2d\_game\_programming*  
=> Nilai kepercayaan : 0.8885
5. Skenario : Mengetahui informasi umum fakultas  
=> Masukan : *Can you tell me the history of our faculty?*  
=> *Intent* yang diharapkan : *#History\_faculty*  
=> *Intent* yang sesungguhnya : *#History\_faculty*  
=> Nilai kepercayaan : 1
6. Skenario : Mengetahui informasi umum fakultas  
=> Masukan : *Can you tell me the mission of our faculty?*  
=> *Intent* yang diharapkan : *#Mission\_faculty*  
=> *Intent* yang sesungguhnya : *#Mission\_faculty*  
=> Nilai kepercayaan : 0.901823
7. Skenario : Mengetahui informasi umum fakultas  
=> Masukan : *What is the vision of our faculty?*  
=> *Intent* yang diharapkan : *#Vision\_faculty*  
=> *Intent* yang sesungguhnya : *#Vision\_faculty*  
=> Nilai kepercayaan : 0.945253
8. Skenario : Mengetahui dosen dan struktur organisasi  
=> Masukan : *Can you tell me the complete organization structure in our faculty?*  
=> *Intent* yang diharapkan : *#Organization\_structure*

- => *Intent* yang sesungguhnya : *#Organization\_structure*  
=> Nilai kepercayaan : 0.923019
9. Skenario : Mengetahui dosen dan struktur organisasi  
=> Masukan : *Can you show me the leaders in our faculty?*  
=> *Intent* yang diharapkan : *#Structural\_lecturer*  
=> *Intent* yang sesungguhnya : *#Structural\_lecturer*  
=> Nilai kepercayaan : 0.965362
  10. Skenario : Penutupan  
=> Masukan : *Thank you*  
=> *Intent* yang diharapkan : *#General\_positive\_feedback*  
=> *Intent* yang sesungguhnya : *#General\_positive\_feedback*  
=> Nilai kepercayaan : 1

#### B. Domain Prosedural

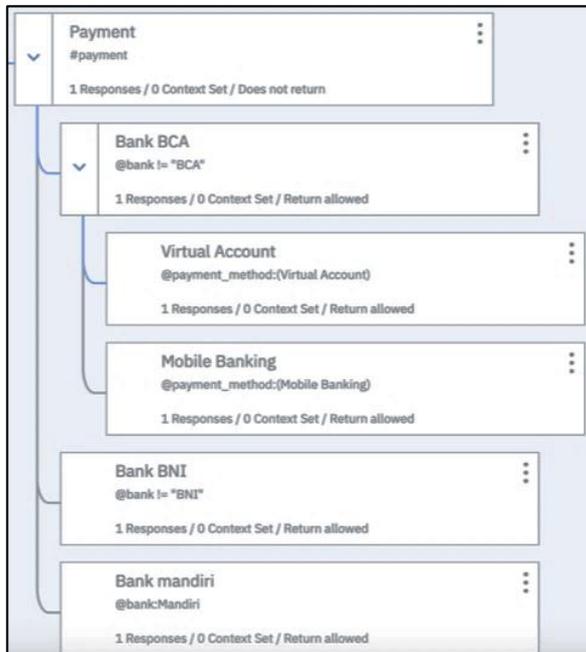
Pengujian domain prosedural ditujukan untuk melihat kesinambungan antara setiap *node* yang diuji [12]. Selain itu, ditujukan juga untuk melihat apakah pengaruh dari *node parent* terhadap skenario untuk tingkatan di bawahnya. Berikut ini adalah sebuah contoh skenario tanya-jawab persiapan mahasiswa untuk proses perwalian.

1. Skenario : Mengetahui cara pembayaran  
=> *Node Parent* : -  
=> Masukan : *Can you help me with payment?*  
=> *Intent* yang diharapkan : *#Payment*  
=> *Intent* yang sesungguhnya : *#Payment*  
=> Nilai kepercayaan : 1
2. Skenario : Mengetahui cara pembayaran  
=> *Node Parent* : *Payment*  
=> Masukan : *BCA*  
=> *Intent* yang diharapkan : *@bca*  
=> *Intent* yang sesungguhnya : *@bca*  
=> Nilai kepercayaan : 0.859009
3. Skenario : Mengetahui cara pembayaran => *Node Parent* : *BCA*  
=> Masukan : *Virtual Account*  
=> *Intent* yang diharapkan : *@virtual\_account*  
=> *Intent* yang sesungguhnya : *@virtual\_account*  
=> Nilai kepercayaan : 0.9580
4. Skenario : Mengetahui cara pembayaran  
=> *Node Parent* : *Payment* => Masukan : *BNI*  
=> *Intent* yang diharapkan : *@bni*  
=> *Intent* yang sesungguhnya : *@bni*  
=> Nilai kepercayaan : 0.87538
5. Skenario : Mengetahui cara pembayaran  
=> *Node Parent* : *Payment*  
=> Masukan : *Maybank*  
=> *Intent* yang diharapkan : *#anything\_else*  
=> *Intent* yang sesungguhnya : -  
=> Nilai kepercayaan : Semua intents rendah.
6. Skenario : Mengetahui perwalian / mata kuliah semester depan.  
=> *Node Parent* : -  
=> Masukan : *Can you help me with guidance?*  
=> *Intent* yang diharapkan : *#Guidance*

- => *Intent* yang sesungguhnya : #Guidance
- => Nilai kepercayaan : 0.7575317
- 7. Skenario : Mengetahui perwalian / mata kuliah per semester.
- => *Node Parent* : Guidance
- => Masukan : *Informatics Engineering*
- => *Intent* yang diharapkan : #InformationEngineering
- => *Intent* yang sesungguhnya : #InformationEngineering
- => Nilai kepercayaan : 0.98143
- 8. Skenario : Mengetahui perwalian / mata kuliah per semester.
- => *Node Parent* : Information Engineering
- => Masukan : semester 3
- => *Intent* yang diharapkan : #Semester3
- => *Intent* yang sesungguhnya : #Semester3
- => Nilai kepercayaan : 0.93664

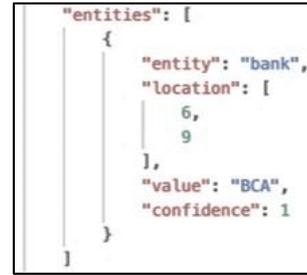
C. Skenario Penyangkalan

Untuk melakukan pengujian ini, dirancanglah dialog khusus skenario pembayaran pada Gambar 12. Pengujian dilakukan dengan tujuan apakah *Watson Assistant* mampu menentukan sebuah nilai sebagai *node* negatif.



Gambar. 12 Contoh skenario penyangkalan

Contoh pengujian dilakukan dengan memasukkan jawaban “*I use BCA*”, pada saat *chatbot* bertanya bank apa yang digunakan. Pada Gambar 13 terlihat bahwa *entity* yang dituju adalah “*bank*” dengan *value* “*BCA*”. Sedangkan dalam Gambar 12, terlihat bahwa kondisi yang dibuat pada *node* anak pertama dari *node payment* adalah jika “*@bank != “BCA”*”, yang bila diterjemahkan ke dalam bahasa alamiah adalah jika *entity* dari “*bank*” bukan “*BCA*”.

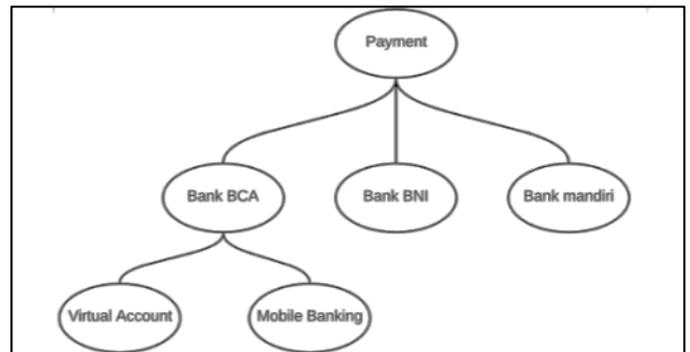


Gambar. 13 Respons terhadap penyangkalan

Adapun masukan dari pengguna menunjukkan bahwa *entity* dari “*bank*” sekarang adalah “*BCA*”. Maka dari itu, kondisi pada *node* pertama menjadi tidak terpenuhi. Jika dimasukkan nilai selain daripada “*BCA*” maka *node* yang akan dikunjungi adalah *node* pertama yaitu *node “Bank BCA”*.

D. Penelusuran Node

Pengujian ini dilakukan untuk memperlihatkan metode temu balik pada *Watson Assistant* adalah dengan menggunakan metode BFS (*Breadth-First Search*). Pengujian ini dilakukan dengan melakukan pengamatan terhadap jawaban dari *Watson Assistant* melalui penelusuran *node* [13]. Dengan menggunakan skenario percakapan seperti pada Gambar 12, dapat direpresentasikan ke dalam bentuk *graph* seperti pada Gambar 14.



Gambar. 14 Pembentukan node untuk skenario pada gambar 12

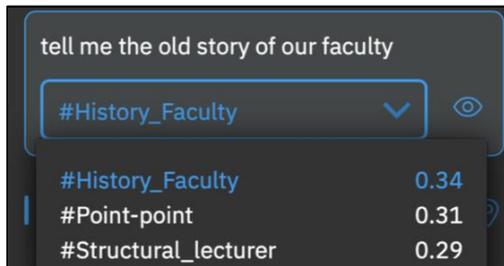
Metode BFS melakukan pencarian dengan cara *traversing node* dari *root node* jika pada Gambar 14 adalah *payment* lalu akan mengunjungi *node “tetangga”* dari *payment*. *Node “tetangga”* itu berarti *node* yang berada sejajar atau 1 level dengan *node* tersebut pada kasus ini *payment* tidak memiliki *node “tetangga”* maka dari itu metode BFS akan mencari ke *child node* dari *payment*. *Node payment* memiliki 3 *child node* yaitu Bank BCA, Bank BNI dan Bank mandiri. Metode BFS akan mengunjungi *node* dari *node* awal atau paling kiri hingga *node* akhir atau paling kanan. Hal ini terbukti dalam penggambaran hasil jawaban pada Gambar 13, bahwa jika pengguna memasukkan *value* *BCA* maka *Watson Assistant* akan masuk ke dalam *node “tetangganya”* yaitu Bank BNI.

E. Hasil Penerjemahan Google

Pengujian ini difokuskan untuk mengetahui apakah terjemahan yang dilakukan melalui pemanfaatan pustaka “PHP GoogleTranslate” sudah sesuai dengan yang diterjemahkan oleh Google Translate. Selain itu, pengujian ini juga dilakukan untuk melihat apakah *Watson Assistant* mampu memberikan jawaban yang sesuai jika pertanyaan yang dimasukkan pengguna dalam Bahasa Indonesia diterjemahkan terlebih dahulu ke dalam Bahasa Inggris oleh *library* tersebut. Pengujian ini akan memperlihatkan kalimat-kalimat dalam Bahasa Indonesia serta jawaban yang diberikan dari masing-masing mesin penerjemah. Berikut ini adalah beberapa contoh hasil penerjemahan untuk pertanyaan-pertanyaan uji coba:

1. Saya ingin mengetahui pelajaran dasar pemrograman.  
=> *I want to know the basic lessons of programming.*
2. Bisa tolong jelaskan mengenai sejarah Fakultas Teknologi Informasi?  
=> *Could you please explain about the history of the Faculty of Information Technology?*
3. Saya sekarang berada di semester 3.  
=> *I am now in semester 3.*
4. Apakah anda mempunyai kontak dosen?  
=> *Do you have lecturer contacts?*
5. Tolong bantu saya dalam perwalian.  
=> *Please help me in guardianship.*
6. Semester 2.  
=> *2<sup>nd</sup> semester.*

Berdasarkan pengujian penerjemahan yang dilakukan pada nomor 3 dan 6, terdapat perbedaan hasil. Dapat dilihat bahwa jika pengguna hanya memasukkan kata “2 semester” maka yang hasil terjemahannya jadi memiliki tambahan “*nd*” dibelakang angkanya. Kekurangan ini bukan hanya disebabkan oleh mesin penerjemah itu saja, melainkan hal ini bisa terjadi karena penggunaan *entity*. Suatu kata / kalimat akan dianggap sebagai *entity* jika kata tersebut memiliki ciri yang sesuai dengan *entity* yang dituju.



Gambar. 15 Masukan awal dari pengguna

F. Hasil Perhitungan dalam Belief Network

Pembelajaran *Watson Assistant* dapat dilakukan dengan cara merubah *intent* atau memastikan kembali *intent* apa yang seharusnya dituju oleh *Watson Assistant*. Sebagai contoh dalam pengujian ini dibuatlah masukan awal dari pengguna pertama kali yang tidak terdapat secara spesifik dalam *intent* seperti dalam Gambar 15.

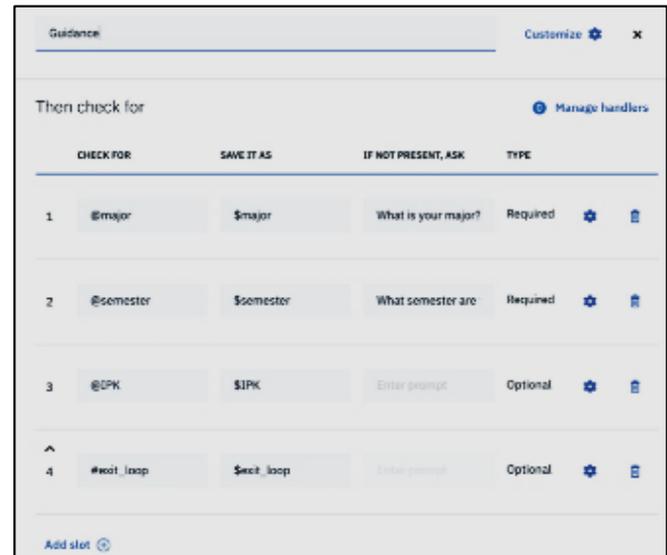


Gambar. 16 Masukan selanjutnya dari pengguna

Walaupun sudah sesuai dengan *intent* awal yang ingin dituju akan tetapi nilai kepercayaan dari *Watson-assistant* sangatlah kecil (dibawah 0.5). Maka dari itu, pembelajaran dapat dilakukan dengan memilih *intent* yang ‘dituju’. Setelah *#History\_Faculty* dipilih kembali maka jika pengguna memasukkan kembali pertanyaan yang sama hasilnya akan seperti pada Gambar 16, dengan nilai keyakinan yang meningkat drastis (dari 0.34 menjadi 1.00).

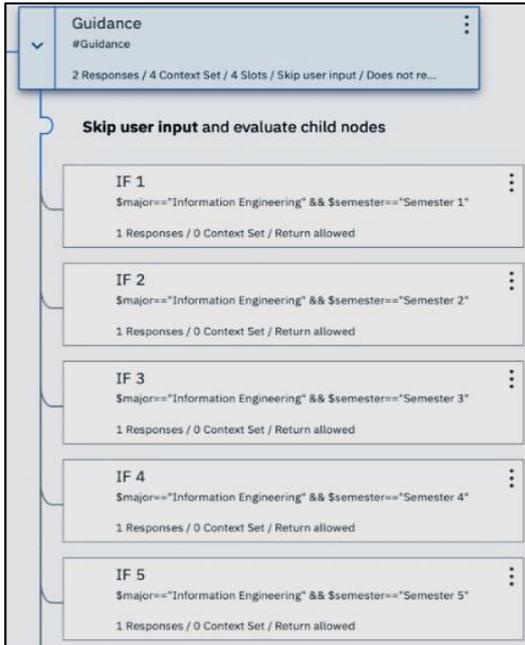
G. Pengujian Assertion Graph

Dalam pengujian ini akan menguji apakah *assertion graph* sudah berjalan pada skenario yang dibuat. Pengujian ini dilakukan dengan melihat apakah terdapat pengaruh dari pernyataan sebelumnya terhadap jawaban yang akan diberikan.



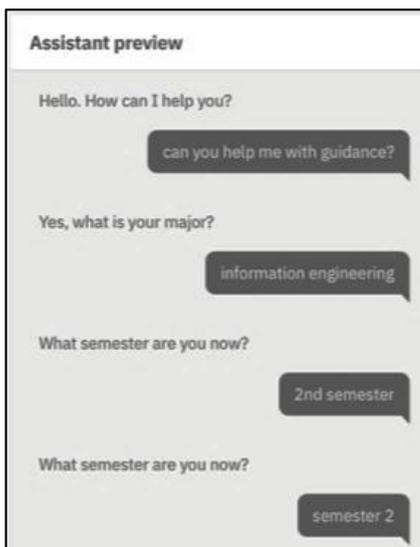
Gambar. 17 Pengujian Assertion Graph dengan Entities dan Slots

Dengan memanfaatkan *entities* jawaban yang diberikan oleh pengguna dapat disimpan. Sehingga jawaban tersebut nantinya dapat diperhitungkan sesuai dengan yang dibutuhkan. Dalam penyusunan dialog, pemanfaatan *entitites* dapat digunakan di dalam *slots*. Setiap *slot* akan menampung sebuah nilai dari *entity* tersebut dalam variabel yang dilambangkan dengan “\$”. Pengujian ini dilakukan terhadap susunan dialog seperti pada Gambar 17.



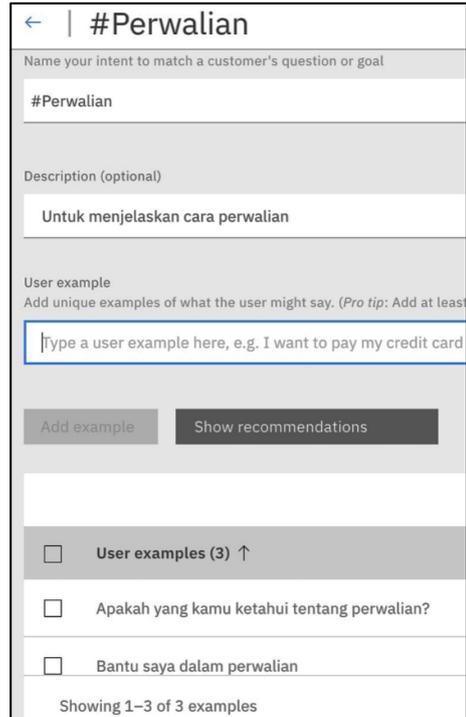
Gambar. 18 Jawaban yang diharapkan dalam Assertion Graph

Jawaban yang diharapkan sudah tersimpan dalam dialog seperti pada Gambar 18. Jawaban ini berdasarkan pengecekan terhadap variabel penampung yang diambil dari slots. Setiap nilai yang diharapkan akan dicocokkan apakah sudah sesuai atau belum sesuai dengan *entities* yang sudah dibuat.



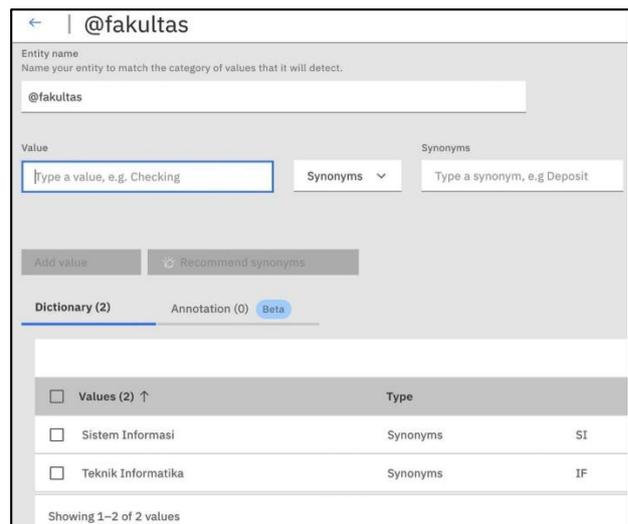
Gambar. 19 Contoh percakapan dalam pengujian Assertion Graph

Dialog dengan *slots* seolah akan memaksa pengguna untuk memberikan jawaban yang diminta [14]. Jika pengguna tidak memberikan jawaban yang sesuai maka *Watson Assistant* akan memberikan pertanyaan yang sama terus menerus. Kondisi ini dapat memperlihatkan terjadinya *assertion* atau pengaruh dari jawaban sebelumnya.



Gambar. 20 Contoh Intent dalam Bahasa Indonesia

Setiap jawaban yang diberikan juga akan memiliki hasil yang berbeda. Jawaban yang diberikan berdasarkan *graph* yang sudah disusun untuk memenuhi setiap kondisi yang ada. Gambar 19 adalah contoh dari pengujian yang dilakukan, dengan jawaban yang sesuai *graph* untuk menampilkan jadwal bagi mahasiswa semester kedua.

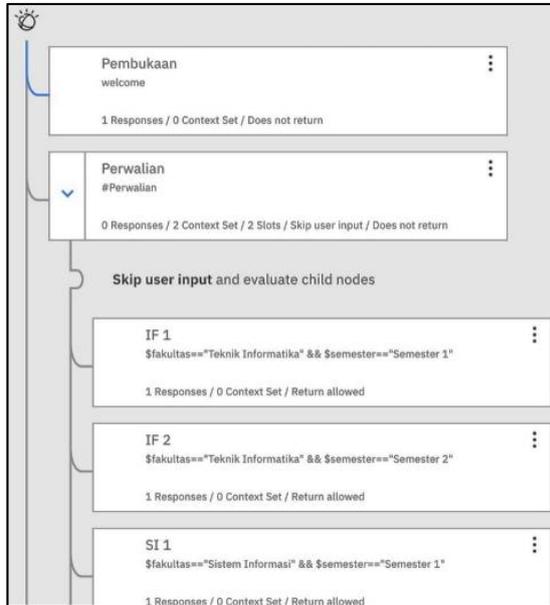


Gambar. 21 Contoh Entity dalam Bahasa Indonesia

#### H. Pengujian Bahasa Indonesia

*Watson Assistant* menyediakan modul khusus untuk sekitar 15 bahasa [1], diantaranya adalah Bahasa Inggris, Bahasa Jepang, dan Bahasa Arab. Namun, Bahasa Indonesia belum didukung secara khusus di dalam *Watson Assistant*. Oleh karena itu, perlu dilakukan pengujian untuk

melihat apakah *Watson Assistant* dapat menangani proses untuk bahasa yang belum didukung dalam modul khusus [15]. Diujicobakan sebuah *intent* dengan nama *#Perwalian* sebagai skenario percakapan dalam Bahasa Indonesia seperti pada Gambar 20.



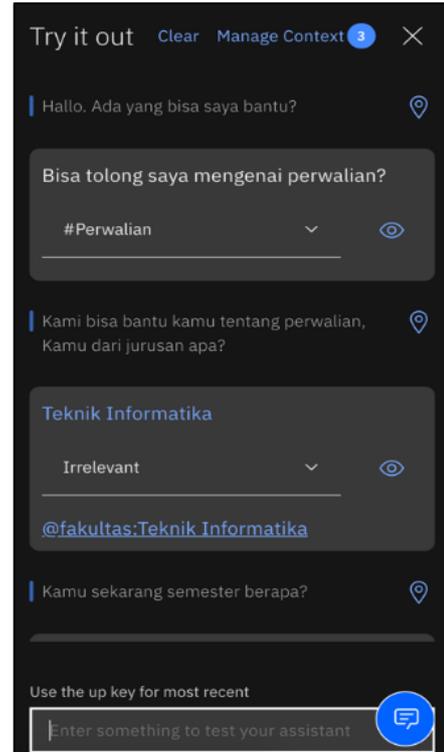
Gambar. 22 Contoh dialog dalam Bahasa Indonesia

Di samping itu, terdapat pula *entity* *@fakultas* dengan konten dalam Bahasa Indonesia seperti pada Gambar 21. Berikutnya *intent* dan *entity* dalam Gambar 20 dan 21 digunakan pada dialog seperti pada Gambar 22, dengan pengaturan *Watson Assistant* yang sudah dipersiapkan semua dalam Bahasa Indonesia. Pengujian ini dilakukan dalam skenario “*Try it out*”.

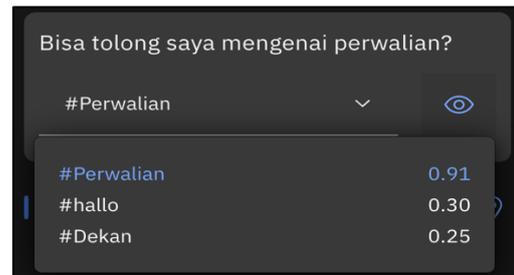
Terlihat pada Gambar 23 hasil percakapan yang dilakukan dengan memberikan pertanyaan: “Bisa tolong saya mengenai perwalian?”, *Watson Assistant* menjawab sesuai dengan dialog yang sudah dipersiapkan yaitu dengan masuk ke dalam *intent* *#Perwalian*.

Gambar 24 menunjukkan nilai kepercayaan yang dihasilkan oleh *Watson Assistant*, *#Perwalian* mendapatkan nilai 0.91. Nilai tersebut termasuk tinggi dikarenakan pertanyaan yang diberikan tidak ada pada contoh pertanyaan dalam *intent* *#Perwalian*. Namun dapat dilihat pula dalam Gambar 20 bahwa setiap contoh pertanyaan dalam *intent* tersebut mengandung kata *perwalian*, sehingga hasil perhitungan probabilitas dalam *node* tersebut tetap mendapatkan nilai yang tinggi.

Melalui pengujian pada Gambar 20-24 diperlihatkan bahwa meskipun bahasa Indonesia tidak didukung secara penuh dalam *Watson Assistant*, tetap dapat disusun skenario tanya jawab dalam bahasa Indonesia. Syarat utama yang diperlukan adalah dibentuknya skenario dan alur percakapan sesuai dengan domain deskriptif dan prosedural yang diinginkan.



Gambar. 23 Contoh percakapan dalam Bahasa Indonesia



Gambar. 24 Nilai kepercayaan pada *#Perwalian*.

## VI. KESIMPULAN

Berdasarkan pada hasil pengujian dalam pengembangan *chatbot* yang dilakukan menggunakan *Watson Assistant*, dapat ditarik kesimpulan bahwa *Watson Assistant* dapat menyediakan kandidat-kandidat jawaban mengacu pada skenario yang telah dirancang sebelumnya. *Watson Assistant* terbukti mampu menjawab pertanyaan mahasiswa/i untuk beberapa topik umum yang ada di dalam buku panduan akademik. Layanan *chatbot* dengan menggunakan *Watson Assistant* dapat disediakan ke dalam beberapa *platfom* seperti Facebook Messenger, Slack ataupun pada aplikasi *web* mandiri. Selain itu, di dalam *Watson Assistant* juga dimungkinkan implementasi skenario dalam berbagai bahasa alami, meskipun tidak disediakan modul khusus untuk bahasa tersebut. Kondisi utama yang perlu diperhatikan untuk implementasi adalah terbentuknya skenario serta alur percakapan melalui domain deskriptif dan prosedural sesuai kebutuhan di lapangan.

Dalam kasus pembuatan skenario untuk membantu interaksi mahasiswa, didapati bahwa skenario yang dihasilkan sebagian besar merupakan skenario dengan domain deskriptif karena ditujukan untuk membuat penjelasan mata kuliah. *Intent* yang dibuat menjadi sangat banyak untuk dapat menangani setiap kasus (mata kuliah dalam hal ini). Di sisi lain, *Watson Assistant* akan sangat baik digunakan untuk menangani kasus dengan domain prosedural dikarenakan tingkat kesulitan untuk penanganannya relatif lebih rendah pada domain ini.

#### REFERENSI

- [1] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A.A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, N. Schlaefer, and C. Welty, "Building Watson: An overview of the DeepQA project," *AI magazine* vol. 31, no. 3: 59-79, 2010.
- [2] A. Azraq, H. Aziz, N. Nappe, C. R. Bravo and L. Sri, *Building Cognitive Applications with IBM Watson Services Vol. 2 Conversation*, IBM Redbooks, 2017.
- [3] A. Lally, S. Bagchi, M.A. Barborak, D.W. Buchanan, J. Chu-Carroll, D.A. Ferrucci, M.R. Glass, A. Kalyanpur, E.T. Mueller, J.W. Murdock, and S. Patwardhan, "WatsonPaths: scenario-based question answering and inference over unstructured information," *AI Magazine*, vol. 38 no. 2, pp. 59-76, 2017.
- [4] M. Biswas, *IBM Watson Chatbots: Beginning AI Bot Frameworks*. Berkeley: Apress, 2018.
- [5] J. da Silva Oliveira, D.B. Espindola, R. Barwaldt, L.M. Ribeiro, and M. Pias, "IBM Watson Application as FAQ Assistant about Moodle", in *Proc. IEEE Frontiers in Education Conference (FIE)*, 2019, pp. 1-8.
- [6] P. J. Lucas and L. C. v. d. Gaag, *Principles of Expert Systems*, Amsterdam: Addison-Wesley, 1991.
- [7] K. Nimavat and T. Champaneria, "Chatbots: An overview. Types, Architecture, Tools, and Future Possibilities," *International Journal for Scientific Research & Development (IJSRD)*, vol. 5, no. 7, pp. 1019-1024, 2017.
- [8] J.L. Jackson, A. Kuriyama, A. Anton, A. Choi, J-P. Fournier, A-K Geier, F. Jacquerioz, D. Kogan, C. Scholcoff, and R. Sun. "The Accuracy of Google Translate for Abstracting Data From Non-English-Language Trials for Systematic Reviews," *Annals of Internal Medicine*, vol. 171, no. 9, pp. 677-679, 2019.
- [9] E.T. Murtisari, R. Widiningrum, J. Branata, and R.D. Susanto. "Google Translate in Language Learning: Indonesian EFL Students' Attitudes," *Journal of Asia TEFL*, vol. 16, no. 3, pp. 978-986, 2019.
- [10] A. Barrio. (2017) Github PHP Google Translate Free. [Online]. Available: <https://github.com/statickidz/php-google-translate-free>
- [11] T. Marris, *JSON at work: practical data integration for the web*, Sebastopol: O'Reilly Media, Inc., 2017.
- [12] Paladines, José, and Jaime Ramírez, "An Intelligent Tutoring System for Procedural Training with Natural Language Interaction", in *Proc. International Conference on Computer Supported Education (CSEDU)*, vol. 2, 2019, pp. 307-314.
- [13] L. Szeremeta, and D. Tomaszuk, Document-oriented RDF graph store, *Studia Informatica*, vol. 38, no. 2, pp. 31-43, 2017.
- [14] J. Collinaszy, M. Bundzel, and I. Zolotova, Implementation of intelligent software using IBM Watson and Bluemix, *Acta Electrotechnica et Informatica*, vol. 17, no. 1, pp. 58-63, 2017.
- [15] K. Ralston, C. Yuhao, H. Isah, and F. Zulkernine, A Voice Interactive Multilingual Student Support System using IBM Watson, in *Proc. IEEE International Conference on Machine Learning And Applications (ICMLA)*, 2019, pp. 1924-1929.