

**INTEGRASI CHATBOT BERBASIS AIML PADA WEBSITE E-COMMERCE
SEBAGAI VIRTUAL ASSISTANT DALAM Pencarian dan Pemesanan Produk
(STUDI KASUS TOKO BUKU ONLINE EDU4INDO.COM)**

Egga Bahartyan, Nurdin Bahtiar, dan Indra Waspada

Ilmu Komputer / Informatika FSM Universitas Diponegoro
eightghax@gmail.com, nurdinbahtiar@gmail.com, indrawaspada@gmail.com

Abstrak

Perkembangan teknologi chatbot semakin canggih hingga saat ini. Mulai dari *chatbot* yang bernama ALICE, hingga aplikasi Siri milik perusahaan Apple sebagai *virtual assistant* yang berjalan pada iPhone masih dikembangkan agar semakin cerdas dan canggih untuk mempermudah pekerjaan manusia. *Chatbot* dapat diintegrasikan ke berbagai layanan, aplikasi atau sumber data lainnya. Salah satunya adalah mengintegrasikan *chatbot* dengan situs web e-commerce untuk menggantikan tugas manusia sebagai *customer service*. Salah satu caranya adalah memanfaatkan layanan AIML *interpreter* milik Pandorabots. AIML berfungsi sebagai “otak” pada *chatbot*, berisi basis pengetahuan yang dipersiapkan untuk menghasilkan jawaban. Pengembangan aplikasi *chatbot* dilakukan menggunakan metodologi Ripple yang merupakan kombinasi metodologi *berorientasi objek* dengan fase-fase klasik dari proses pengembangan perangkat lunak. Aplikasi yang dihasilkan diberi nama EIVA, *chatbot* yang berperan sebagai *customer service* pada toko buku online Edu4indo.com. Interaksi pelanggan dengan chatbot yang menggunakan bahasa alami diharapkan dapat menjadikan *chatbot* lebih memahami apa yang dimaksudkan pelanggan bila dibandingkan dengan penggunaan alat bantu pencarian tradisional dengan banyak pilihan atau bahkan *textbox* tunggal untuk memasukkan kata kuncinya.

Kata kunci : chatbot, bahasa alami, e-commerce, AIML, Pandorabots

1. PENDAHULUAN

Edu4indo adalah usaha jasa penyedia buku dan sarana pendidikan yang dikelola oleh praktisi pendidikan dan penyedia sarana pendidikan berpengalaman. Edu4indo memanfaatkan teknologi informasi dalam pemasaran produknya melalui sistem layanan elektronik. Salah satu pemanfaatan teknologi informasi pada Edu4indo yaitu situs web e-commerce dapat diakses melalui Edu4indo.com. Pelanggan dapat dengan mudah melakukan pencarian dan pemesanan produk milik Edu4indo melalui situs web e-commerce tersebut. Demi meningkatkan pelayanan kepada pelanggan, Edu4indo memiliki layanan *customer service* berupa media chatting dimana pelanggan dapat langsung bertanya menggunakan Yahoo! Messenger seputar Edu4indo dan produknya atau bahkan untuk melakukan pemesanan.

Namun, dari penjelasan di atas ternyata memiliki kendala. Pelanggan yang tidak sedang menggunakan atau belum memiliki akun Yahoo! Messenger tidak dapat menggunakan layanan *customer service*. Disamping itu, Edu4indo tidak

menyediakan layanan *customer service* selama 24 jam per hari, yaitu hanya dapat digunakan pada saat jam kerja. Dari permasalahan tersebut, dapat disimpulkan bahwa penggunaan layanan *customer service* menggunakan Yahoo! Messenger dirasa masih kurang cukup untuk memenuhi kualitas pelayanan bagi pelanggan Edu4indo, sehingga dibutuhkan suatu solusi untuk mengatasi permasalahan tersebut. Dalam hal ini, solusi yang dimaksud adalah bagaimana Edu4indo dapat memberikan pelayanan lebih kepada pelanggannya melalui customer service tanpa melalui Yahoo! Messenger dan dapat berkerja selama 24 jam per hari.

Dalam perkembangan teknologi saat ini terdapat banyak penelitian yang memanfaatkan *chatbot* sebagai *virtual assistant* atau *shop assistant* [5] [10]. *Chatbot* itu sendiri merupakan sebuah program komputer yang mampu berinteraksi dengan penggunanya dengan menggunakan bahasa alami [14]. Saat ini mulai banyak *chatbot* berbasis AIML (Artificial Intelligence Markup Language) yang digunakan sebagai *virtual assistant* pada beberapa aplikasi

virtual assistant seperti *Voice Actions*, *Skyvi*, *Iris*, dan *Call Mom* (Pandorabots.com). AIML sering digunakan sebagai basis pengetahuannya (*knowledge base*) [17].

Dari permasalahan yang telah dipaparkan sebelumnya, yaitu mengenai peningkatan layanan Edu4indo terhadap pelanggannya dan penjelasan tentang pemanfaatan *chatbot*, maka dilakukan penelitian mengenai pemanfaatan *chatbot* sebagai *virtual assistant* untuk pelanggan Edu4indo. Untuk selanjutnya *chatbot* ini akan disebut dengan *Edu4indo Interactive Virtual Assistant* disingkat dengan EIVA..

2. DASAR TEORI

Metodologi Pengembangan Perangkat Lunak *Ripple*

Ripple merupakan metodologi berorientasi objek yang diperkenalkan oleh Mike O'Docherty pada bukunya yang berjudul *Object-Oriented Analysis and Design Understanding System Development with UML 2.0*. *Ripple* terlihat seperti versi berorientasi objek dari semua fase klasik pada model pengembangan perangkat lunak. Karena object oriented sangat terjangkau, pengembang dapat terlibat dalam semua fase, pelanggan dapat terlibat dalam tahap awal, yang membantu pengembang untuk melakukan pekerjaan mereka, dan manajer tidak disingkirkan dari dunia pengembang, sehingga komunikasi semakin terjalin. Tabel 1 berikut menjelaskan gambaran singkat dari metodologi *Ripple* mengenai artefak apa saja yang dihasilkan pada setiap fasenya. Fase-fase pada metodologi *Ripple* meliputi *Genesis*, *Requirements*, *Analysis*, *Class Specification*, *Implementation*, *Testing*, *Deployment*, dan *Maintenance*.

Pengertian *Chatbot*

Chatbot adalah karakter bahasa alami yang berkomunikasi dengan penggunanya, atau orang-orang yang sedang chatting di messenger, web instan, email, *usenet*, forum *web*, atau bahkan melalui komunikasi suara seperti telepon. *Chatbot* juga kadang-kadang disebut chat robot, *bot*, *chatterbot*, *botchatting*, *chatterbox*, *V-Host*, *V-People*, *agent*, dan manusia *virtual*. Sebuah *chatbot* dapat dikaitkan dengan sebuah *avatar*, animasi yang juga dapat mencakup sintesis

pidato sehingga *chatbot* mungkin muncul lebih hidup melalui animasi *virtual reality* dan suara. Sebuah *chatbot* juga dapat memiliki teknologi pengenalan suara, sehingga *bot* hanya melalui teks. *Chatbot* memiliki botmaster, yaitu orang di belakang layar yang bertanggung jawab untuk menciptakan kepribadian bot dan melepaskannya ke tempat umum melalui internet. Membuat *bot* lebih seperti menciptakan karakter untuk sebuah novel atau skenario daripada seperti menulis sebuah program komputer [24].

Situs Pandorabots.com

Pandorabots.com adalah situs *web* bebas berbasis *open-source* yang dapat digunakan untuk mengembangkan dan mempublikasikan *chatbot* di web. "Pandorabots adalah komunitas terbesar *chatbot* di internet". Sejak Februari 2012 layanan komunitas gratis Pandorabots adalah rumah bagi lebih dari 166.000 *botmaster* dan 201.000 *chatbots* dalam berbagai bahasa. Lebih dari 1,4 miliar interaksi percakapan terjadi antara klien dan Pandorabots, dan jumlah ini meningkat cepat (dapat dilihat di dekat bagian atas tengah halaman *web* Pandorabots). Pandorabots digunakan di seluruh *web* – silahkan masukkan kata kunci "talk?botid=" (dan termasuk tanda kutip) ke dalam kotak pencarian google untuk perkiraan jumlah terbaru.

Pandorabots sedang dalam pembangunan konstan dan memiliki komunitas yang aktif yang sangat besar. Pandorabots mematuhi standar terbuka untuk penerbitan *chatbot*. Untuk Pandorabots sejauh mungkin sendiri telah dibangun dari perangkat lunak open source. Pandorabots menggunakan AIML sebagai bahasa *markup* konten pengetahuan [25]. Untuk menggunakan layanan Pandorabots, hal pertama yang perlu dilakukan adalah mengunjungi situs Pandorabots pada <http://pandorabots.com> dan melakukan pendaftaran untuk sebuah akun. Setelah pendaftaran selesai dilakukan, kemudian Pandorabots akan memberikan halaman kontrol termasuk halaman *training* yang berfungsi untuk mengelola basis pengetahuan pada *bot* [24].

Setelah dilakukan *training* pada *bot* sehingga *bot* tersebut memiliki pengetahuan, maka untuk memanfaatkan *bot* tersebut agar dapat digunakan pada aplikasi pada pihak ketiga yaitu dengan

cara menggunakan Pandorabots API yang diakses pada <http://www.pandorabots.com/pandora/talk-xml> dengan memberikan tiga parameter melalui method POST atau GET, yaitu:

- 1) *botid*, merupakan identitas bot yang didapat setelah berhasil melakukan registrasi
- 2) *input*, yaitu teks yang diberikan kepada *bot*
- 3) *custid*, digunakan untuk merekam percakapan yang berbentuk teks unik sebagai identitas pengguna bot agar percakapan dapat tetap pada konteksnya [24]

AIML dan Unit-unit AIML

AIML (*Artificial Intelligence Markup Language*) adalah bahasa pelengkap XML yang memungkinkan user untuk meng-*customize* suatu *chatbot* (memasukan *knowledge* ke dalam *chatbot*) atau bahkan merancang sendiri suatu *chatbot* dalam waktu singkat dengan menggunakan teknologi *free software* A.L.I.C.E.. AIML dikembangkan oleh komunitas Alicebot dan telah menjadi dasar dari Alicebot pertama, yakni A.L.I.C.E. Berikut adalah unit-unit terpenting dari AIML

- 1) `<aiml>`
Merupakan tag yang menandai awal dan akhir suatu dokumen AIML
- 2) `<category>`
Merupakan tag yang menandai "unit pengetahuan"
- 3) `<pattern>`
Digunakan untuk menyimpan pola sederhana yang sekiranya cocok dengan apa yang diketikkan oleh seorang pengguna pada Alicebot. Dengan kata lain, *pattern* merupakan pertanyaan hasil masukan pengguna. *Pattern* dalam bahasa AIML sangatlah sederhana dan hanya terdiri dari abjad, angka, karakter spasi, serta simbol wildcard `'_'` dan `'*'`
- 4) `<template>`
Berisi respon atau jawaban Alicebot terhadap masukan pengguna. Dalam bentuk paling sederhana, *template* hanya terdiri dari teks datar. Namun umumnya, *template* dapat ditransformasikan ke dalam program komputer yang dapat menyimpan data, membuka program lain, dan melakukan beberapa tugas lainnya.

Selain yang disebutkan di atas, terdapat 20 atau lebih *tag* tambahan yang umumnya ditemukan dalam file AIML. Aplikasi A.L.I.C.E. memiliki *knowledge* yang mencakup sekitar 41.000 kategori.

Terdapat 2 (dua) tipe konteks opsional dalam AIML, yakni "that" dan "topic". Tag `<that>` dalam AIML terletak didalam kategori dan bertugas untuk menyimpan jawaban terakhir *chatbot*. Hal ini penting apabila *chatbot* mengajukan pertanyaan pada *user*. Sedangkan *tag* `<topic>` terletak diluar kategori dan bertugas untuk menyatukan kelompok kategori. Sebagai catatan, apabila respon terakhir *chatbot* adalah berupa paragraf (lebih dari satu kalimat), nilai "that" akan di-set menjadi kalimat terakhir dalam paragraf tersebut. Umumnya aplikasi `<that>` digunakan dalam *Yes/No Question*.

Prestashop

Prestashop adalah suatu *platform* perbelanjaan toko *online* yang gratis dan *open-source*. Prestashop sudah digunakan lebih dari 125.000 toko *online* yang aktif pada lebih dari 180 negara di seluruh dunia. Software Prestashop ringan, cepat, dan sangat fleksibel (www.prestashop.com/en/faq). CMS ini cukup populer di dunia dan banyak yang menggunakannya. Untuk mengetahui bagaimana struktur basis data Prestashop yang nantinya akan diintegrasikan dengan EIVA, dapat diunduh melalui alamat <http://doc.prestashop.com/download/attachments/9404440/pdw-1.5-20130605.pdf>.

3. ANALISIS DAN PERANCANGAN

Kebutuhan

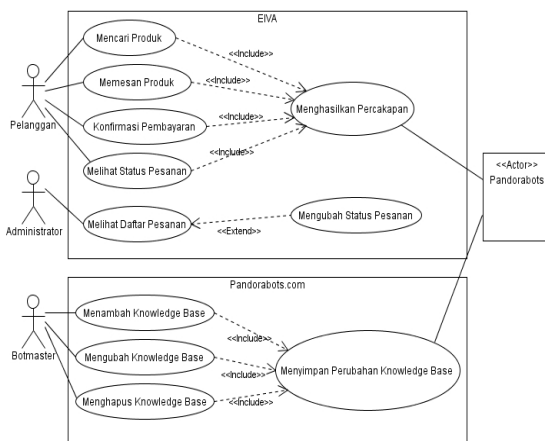
Edu4indo *Interactive Virtual Assistant* atau disingkat dengan EIVA, merupakan sebuah sistem berbasis *chatbot* yang berfungsi layaknya *customer service* dimana penggunaanya berinteraksi melalui *chatting*. Pengguna yang dimaksud disini adalah siapapun yang menggunakan atau berinteraksi dengan EIVA. Selanjutnya disebut dengan *customer*. *Customer* meminta dan EIVA melayani layaknya *customer service*. *Customer* dapat mencari produk dengan kriteria yang diinginkannya melalui bertanya. Selain itu, *customer* dapat memesan produk yang diinginkannya termasuk pada saat proses

checkout. Selain itu, EIVA dapat melayani konfirmasi pembayaran pada *customer* yang sudah melakukan proses pembayaran dan mampu memberikan informasi status pemesana *customer*. Semua kegiatan tersebut dilakukan oleh *customer* dan EIVA melalui percakapan dengan bahasa alami (melakukan percakapan menggunakan bahasa Indonesia).

Semua pesanan *customer* disimpan sehingga pihak perusahaan dapat mengetahui detail *customer* berupa produk apa saja yang sudah dipesan beserta alamat dan detail pembayaran yang mendukung dalam proses transaksi. Semua daftar pesanan tersebut dikelola oleh pegawai perusahaan dapat disebut juga dengan administrator. Administrator bertugas melakukan validasi pembayaran jika terdapat *customer* yang sudah melakukan proses pembayaran. Setelah melakukan proses pengecekan, administrator dapat mengubah status pesannya, misalnya dari “Menunggu Validasi Administrator” ke “Lunas, Sedang Dikirim” atau status pesanan lainnya.

Diagram Use case

Use case diagram disusun berdasarkan daftar actor dan daftar use case dan disusun berdasarkan hubungan keduanya, use case diagram sistem dapat dilihat pada Gambar 1.



Gambar 1. Use Case Diagram Sistem

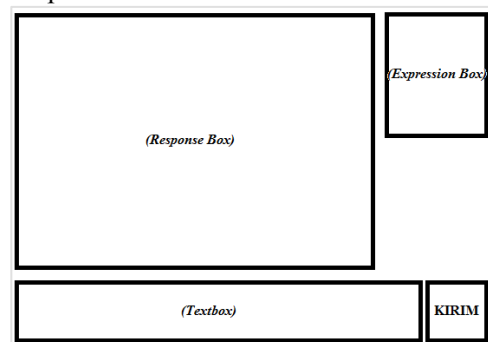
Perancangan Antarmuka

Sistem EIVA terdiri dari dua tampilan utama, yaitu EIVA untuk customer/pelanggan, dan untuk administrator. Aplikasi EIVA untuk customer merupakan aplikasi yang digunakan oleh customer dalam pencarian dan pemesanan

produk. Oleh karena itu, kontrol utama yang ada dalam tampilannya terdiri dari:

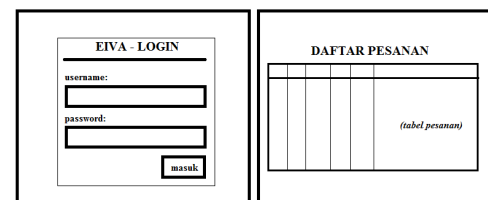
- 1) *Textbox*, sebagai tempat untuk mengetikkan pertanyaan dari pelanggan
- 2) *Response Box*, digunakan untuk menghasilkan respon dari pertanyaan yang dikirim oleh pelanggan
- 3) Kotak Ekspresi, menghasilkan berbagai ekspresi yang berbeda sesuai dengan respon yang diberikan agar lebih interaktif

Untuk lebih jelasnya, rancangan antarmuka EIVA untuk customer dapat dilihat pada gambar 2. Hanya terdapat satu bagian tampilan, tetapi pada bagian response box dapat berubah-ubah.



Gambar 2. Rancangan Antarmuka EIVA untuk Customer

Selanjutnya akan dijelaskan mengenai tampilan untuk administrator. Pada tugas akhir ini, sesuai judulnya mengenai integrasi situs web e-commerce dengan virtual assistant, maka tampilan administrator hanya dibuat dalam bentuk sederhana. Sehingga, komponen yang ditampilkan dalam antarmuka EIVA untuk administrator hanya berupa form untuk login dan tabel berisi daftar pesanan, dapat dilihat pada gambar 3 berikut.

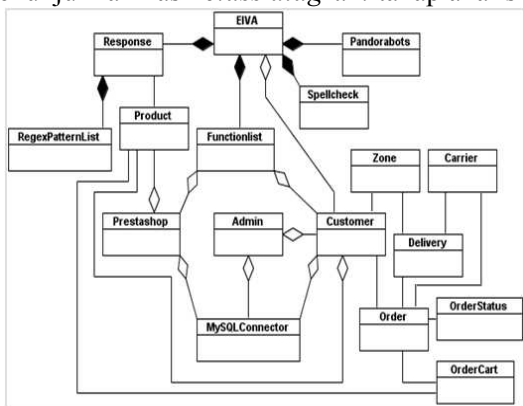


Gambar 3. Rancangan Antarmuka EIVA untuk Administrator

Class Diagram Tahap Analisis

EIVA merupakan kelas utama yang menggunakan fungsi dari kelas lain. EIVA

menggunakan kelas *Spellcheck* untuk membetulkan kesalahan pengetikan yang dilakukan pengguna (pelanggan). Kemudian menggunakan kelas *Pandorabots* untuk mengambil teks jawaban polos yang terkadang berisi kata yang diberi tanda khusus. Kata tersebut merupakan nama fungsi untuk dipanggil dari kelas *FunctionList* berbagai fungsi yang juga terdapat fungsi yang menggunakan kelas lain, yaitu kelas *Customer* untuk mengelola data pelanggan, dan kelas *Prestashop* untuk menghubungkan dengan data produk pada basis data CMS Prestashop yang masih menggunakan kelas *MySQLConnector*. Kelas *Customer* memiliki atribut yang berkaitan, seperti produk, alamat pengiriman, daftar pesanan, keranjang belanja (cart) dan status pesanan. Kelas-kelas tersebut diberi nama *Product*, *Order*, *Zone*, *Delivery*, *Carrier*, *OrderStatus*, dan *OrderCart* dan saling berasosiasi. Hasil akhir proses dikemas pada kelas *Response* beserta masukan jawaban yang dibutuhkan berupa *regular expression* yang polanya sudah dipersiapkan pada kelas *RegexPatternList*. Gambar 4 berikut menunjukkan hasil *class diagram* tahap analisis.

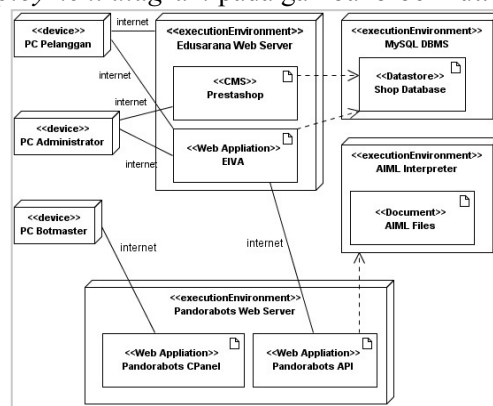


Gambar 4. *Class Diagram* Tahap Analisis

Arsitektur Sistem

Arsitektur sistem yang digunakan EIVA menggunakan konsep *three-tier*. *Customer*, *administrator* dan *botmaster* yang menggunakan *browser* untuk membuka halaman *web* berada pada *client tier*. *Customer* dan *administrator* terhubung dengan EIVA melalui internet. *Customer* dapat juga mencari dan memesan produk melalui halaman web Prestashop. Sedangkan *botmaster* mengelola AIML menggunakan *Pandorabots control panel*.

Aplikasi EIVA berjalan di dalam *web server*, begitu juga dengan CMS Prestashop. Komponen pemroses tersebut berada pada *middle tier* dan mengolah data yang terletak pada *data tier*. Perangkat pada *data tier* meliputi DBMS untuk menyimpan data produk dan pesanan. AIML *interpreter* menghasilkan respon berdasarkan *file* AIML pada server Pandorabots. Rancangan arsitektur sistem disajikan dalam bentuk *deployment diagram* pada gambar 5 berikut.



Gambar 5. *Deployment Diagram* Sistem EIVA

Teknologi yang Digunakan

EIVA merupakan sebuah sistem berbasis *web* sehingga teknologi yang digunakan merupakan komponen yang mendukung dalam pengembangan aplikasi *web*. Pada sisi *client* digunakan HTML dan *javascript* sebagai *client-side scripting* dan semakin dipermudah dalam proses pengembangannya dengan setelah menggunakan jQuery. Melalui *browser* yang digunakan *customer*, pengiriman data (yang berupa pertanyaan) dilakukan melalui AJAX sehingga halaman tidak perlu *reload* dalam proses percakapan antara *customer* dengan EIVA. Kemudian pertanyaan tersebut diproses pada sisi *server* menggunakan bahasa pemrograman yang mendukung *server-side scripting*, yaitu PHP. Bahasa pemrograman PHP memiliki fungsi yang dapat menghubungkan dan mengolah data menggunakan MySQL. PHP juga dapat digunakan untuk mengambil atau memanfaatkan *web service*, sehingga pertanyaan yang diberikan customer dapat diolah menggunakan PHP yang memanfaatkan Pandorabots API.

4. IMPLEMENTASI DAN PENGUJIAN

Penulisan Knowledge Base EIVA

Knowledge Base digunakan agar EIVA dapat menjawab pertanyaan *customer* menggunakan bahasa alami. EIVA menggunakan AIML sebagai *knowledge base*. EIVA hanya memiliki pattern sebagai pertanyaan *customer* yang terbatas, sehingga tidak semua pertanyaan yang diberikan *customer* dapat dimengerti. Jika EIVA menemui kondisi dimana pertanyaan yang diberikan *customer* tidak ada atau tidak dikenal oleh EIVA, maka secara *default* EIVA akan menjawab “Maaf, saya tidak mengerti maksud Anda”. *Syntax* AIML yang berbentuk XML dikonversi menjadi tabel agar lebih mudah dibaca. Berikut merupakan daftar pertanyaan yang dikenal dan respon yang diberikan EIVA. Pada kode 1 berikut merupakan salah satu contoh *category* pada AIML yang digunakan EIVA.

```
<category>
<pattern>cari * tentang *</pattern>
<template>
<think>
<set name="lastcommand"> cari_kat_key</set>
<set name="kategori">
<star index="1" /></set>
<set name="keyword">
<star index="2" /></set>
</think>
<random>
<li>OK, </li>
<li>Baik.. </li>
<li>Yup, </li>
</random>
Berikut hasil pencarian <star index="1" />
tentang
<star index="2" />
[[search_products_by_category_and_keyword#<st
ar
index="1"/>|<star index="2" />]]
</template>
</category>
```

Kode 1. Contoh *Category* pada AIML yang Digunakan EIVA

Diantara *tag pattern* terdapat kalimat “cari * tentang *” yang berarti, EIVA dapat menerima pola dengan kata pertama “cari”, kemudian diikuti kata atau kalimat apa saja sebelum kata “tentang” yang diikuti kata atau kalimat apa saja.

Misalnya “cari buku tentang binatang malam”. Respon yang dihasilkan EIVA berada di dalam *tag* template selain pada *tag think*. *Tag think* mengeksekusi *syntax* yang ada di dalamnya, tetapi tidak dicetak. *Tag set* menyimpan kata atau kalimat yang dimasukkan ke dalam variabel dengan atribut *name*, sehingga variabel *lastcommand* berisi *cari_kat_key*, variabel kategori berisi kata pada pengganti tanda bintang (*) yang menggunakan *tag star*. Sehingga, sesuai contoh sebelumnya, variabel kategori berisi “buku” dan variabel *keyword* berisi “binatang malam”. *Tag random* menghasilkan respon acak pada *tag li*. Setelah itu diikuti kalimat pengantar dan selanjutnya terdapat tanda “[[“ diikuti nama fungsi yang sudah dipersiapkan. Diantara tanda “[[” dan”]]” terdapat tanda “#” yang dibuat untuk membatasi antara fungsi dengan parameter. Sedangkan tanda “|” digunakan untuk membatasi parameter yang diberikan. Sehingga dalam kasus ini, Pandorabots akan memberikan jawaban seperti “Baik.. Berikut hasil pencarian buku tentang binatang malam [[search_products_by_category_and_keyword#buku|binatang malam]]” yang kemudian akan dilakukan *parsing* pada kelas EIVA berdasarkan hasil respon tersebut.

Implementasi Antarmuka

Gambar 6 berikut merupakan halaman awal EIVA. Saat pengguna (*customer*) membuka halaman EIVA, maka akan ditampilkan halaman dengan satu *textbox* untuk memasukkan kalimat pertanyaan, satu kotak di atasnya untuk menampilkan hasil jawaban, dan satu gambar wanita dengan berbagai ekspresi yang dapat berubah-ubah sebagai bentuk ekspresi.

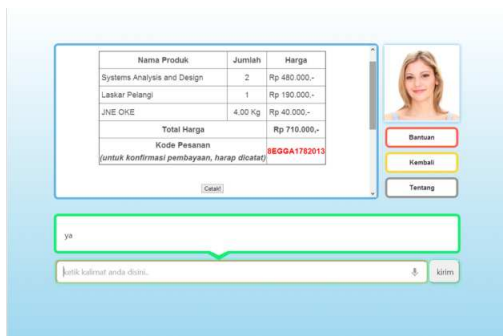


Gambar 6. Tampilan Awal EIVA

Gambar 7 menunjukkan bagaimana interaksi EIVA dalam melayani pelanggan untuk pencarian produk. Sedangkan tampilan EIVA saat melayani pemesanan dapat dilihat pada gambar 8.

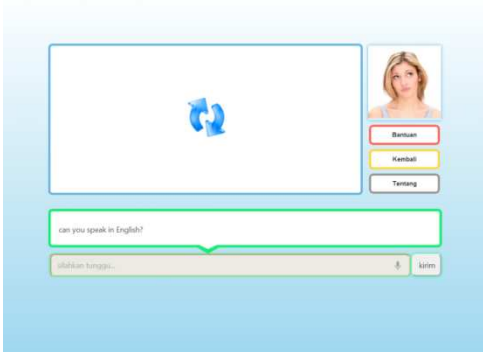


Gambar 7. Tampilan EIVA Saat Melayani Pencarian

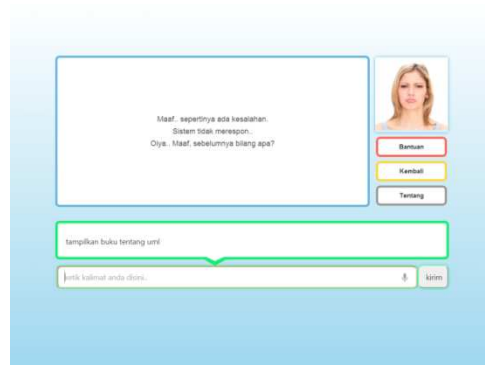


Gambar 8. Tampilan EIVA Saat Melayani Pemesanan

EIVA dapat memiliki berbagai ekspresi. Gambar 9 dan gambar 10 menunjukkan bagaimana EIVA memberikan respon saat berada pada kondisi-kondisi tertentu. Respon yang dihasilkan tidak hanya merubah kalimat, tetapi juga dapat menampilkan berbagai macam ekspresi wajah.



Gambar 9. Ekspresi EIVA Saat Proses



Gambar 10. Ekspresi EIVA Saat Terjadi Gangguan

Rencana Pengujian

Terdapat 6 *use case* yang digunakan sebagai acuan pada tahap pengujian ini, yaitu Mencari Produk, Memesan Produk, Konfirmasi Pembayaran, Melihat Status Pesanan, Melihat Daftar Pesanan, dan Mengubah Status Pesanan. Rencana pengujian EIVA ditunjukkan pada tabel 1 berikut.

Tabel 1. Rencana Pengujian

No	Use Case	Pengujian	Jenis Pengujian	Identifikasi Pengujian
1.	Mencari Produk	Mencari produk yang ada pada basis data	blackbox	U-1-01
		Mencari produk yang tidak ada pada basis data	blackbox	U-1-02
2.	Memesan Produk	Memesan produk yang ada pada basis data	blackbox	U-2-01
		Memesan produk yang tidak ada pada basis data	blackbox	U-2-02
3.	Konfirmasi Pembayaran	Menggunakan kode pesanan yang terdaftar	blackbox	U-3-01
		Menggunakan kode pesanan yang tidak terdaftar	blackbox	U-3-02
4.	Melihat Status Pesanan	Menggunakan kode pesanan yang terdaftar	blackbox	U-4-01
		Menggunakan kode pesanan yang tidak terdaftar	blackbox	U-4-02
5.	Melihat Daftar Pesanan	Login menggunakan username dan password yang benar	blackbox	U-5-01
		Login menggunakan username dan password yang salah	blackbox	U-5-02
6.	Mengubah Status Pesanan	Memasukkan angka pada kolom "Terbayar" (masukkan valid angka)	blackbox	U-6-01
		Memasukkan huruf pada kolom "Terbayar" (masukkan valid angka)	blackbox	U-6-02

Pelaksanaan Pengujian

Pelaksanaan pengujian dilakukan dengan cara mengimplementasikan rencana pengujian yang telah disusun. Tabel 2 hingga tabel 7 berikut menunjukkan hasil pengujian yang telah dilakukan.

Tabel 2. Hasil Pengujian Pencarian Produk

Identifikasi	U-1-01	U-1-02
Deskripsi	Mencari produk yang ada pada basis data	Mencari produk yang tidak ada pada basis data
Prosedur Pengujian	Memasukkan pertanyaan untuk mencari produk yang ada pada basis data	Memasukkan pertanyaan untuk mencari produk yang tidak ada pada basis data
Masukan	Pertanyaan untuk mencari produk yang ada pada basis data, seperti "tampilkan buku tentang uml"	Pertanyaan untuk mencari produk yang tidak ada pada basis data "tampilkan buku tentang zzzzzzzz"
Keluaran yang Diharapkan	EIVA menampilkan produk yang dicari	Terdapat pemberitahuan bahwa produk yang dicari sedang kosong
Hasil yang Didapat	Produk yang dicari ditampilkan	EIVA menunjukkan bahwa produk yang dicari tidak ada / kosong
Kesimpulan	Diterima	Diterima

Tabel 3. Hasil Pengujian Pemesanan Produk

Identifikasi	U-2-01	U-2-02
Deskripsi	Memesan produk yang ada pada basis data	Memesan produk yang tidak ada pada basis data
Prosedur Pengujian	Memberikan kode produk yang ada pada basis data	Memberikan kode produk yang tidak ada pada basis data
Masukan	Kode produk yang ada pada basis data	Kode produk yang tidak ada pada basis data
Keluaran yang Diharapkan	Menuju ke proses pemesanan selanjutnya	Memberikan notifikasi bahwa kode produk salah
Hasil yang Didapat	Menanyakan berapa jumlah produk yang ingin dipesan	Memberikan informasi bahwa kode produk yang diberikan salah dan menawarkan untuk kembali ke hasil pencarian sebelumnya
Kesimpulan	Diterima	Diterima

Tabel 4. Hasil Pengujian Konfirmasi Pembayaran

Identifikasi	U-3-01	U-3-02
Deskripsi	Menggunakan kode pesanan yang terdaftar	Menggunakan kode pesanan yang tidak terdaftar
Prosedur Pengujian	Memberikan kode pesanan yang ada pada basis data	Memberikan kode pesanan yang tidak ada pada basis data
Masukan	Kode produk yang ada pada basis data	Kode produk yang tidak ada pada basis data
Keluaran yang Diharapkan	Notifikasi bahwa konfirmasi pembayaran sudah disimpan	Notifikasi bahwa kode pesanan tidak terdaftar
Hasil yang Didapat	Notifikasi bahwa konfirmasi pembayaran sudah disimpan	Notifikasi bahwa kode pesanan tidak terdaftar
Kesimpulan	Diterima	Diterima

Tabel 5. Hasil Pengujian Melihat Status Pembayaran

Identifikasi	U-4-01	U-4-02
Deskripsi	Menggunakan kode pesanan yang terdaftar	Menggunakan kode pesanan yang tidak terdaftar
Prosedur Pengujian	Memasukkan kode pesanan yang sudah terdaftar	Memasukkan kode pesanan yang tidak terdaftar
Masukan	Kode pesanan yang sudah terdaftar	Kode pesanan yang tidak terdaftar
Keluaran yang Diharapkan	Informasi status pesanan berdasarkan kode pesanan	Notifikasi bahwa kode pesanan yang dimasukkan salah
Hasil yang Didapat	Informasi status pesanan yang sesuai dengan kode pesanan	Notifikasi bahwa kode pesanan yang dimasukkan salah
Kesimpulan	Diterima	Diterima

Tabel 6. Hasil Pengujian Melihat Pesanan

Identifikasi	U-5-01	U-5-02
Deskripsi	Login menggunakan username dan password yang benar	Login menggunakan username dan password yang salah
Prosedur Pengujian	Memasukkan username dan password yang benar	Memasukkan username dan password yang salah
Masukan	Username dan password yang benar	Username dan password yang salah
Keluaran yang Diharapkan	Dialihkan ke halaman yang berisi daftar pesanan	Diberi notifikasi bahwa username dan password salah
Hasil yang Didapat	Dialihkan ke halaman yang berisi daftar pesanan	Diberi notifikasi bahwa username dan password salah
Kesimpulan	Diterima	Diterima

Tabel 7. Mengubah Status Pesanan

Identifikasi	U-6-01	U-6-02
Deskripsi	Memasukkan angka pada kolom "Terbayar" (masukkan <i>valid</i> angka)	Memasukkan huruf pada kolom "Terbayar" (masukkan <i>valid</i> angka)
Prosedur Pengujian	Pada kolom "Terbayar" dari daftar pesanan terdapat <i>textbox</i> , kemudian memasukkan angka sebagai nominal angka dari pembayaran <i>customer</i>	Pada kolom "Terbayar" dari daftar pesanan terdapat <i>textbox</i> , kemudian memasukkan huruf acak
Masukan	Angka sembarang	Huruf sembarang
Keluaran yang Diharapkan	Data diperbarui, jumlah kekurangan pembayaran sesuai	Terdapat notifikasi yang menyatakan bahwa data yang dimasukkan salah
Hasil yang Didapat	Data diperbarui, jumlah kekurangan pembayaran sesuai	Terdapat notifikasi yang menyatakan bahwa data yang dimasukkan salah
Kesimpulan	Diterima	Diterima

Evaluasi Pengujian

Berdasarkan hasil pengujian pada tabel 4.8 hingga 4.13 dapat dilihat bahwa pengujian yang telah dilakukan dapat diterima. Hasil yang didapat dari pelaksanaan pengujian menunjukkan kesesuaian dengan keluaran yang diharapkan. Adanya kesesuaian tersebut merupakan suatu bukti bahwa sistem berjalan sesuai dengan definisi kebutuhan. Sehingga, dalam hal ini dapat dinyatakan bahwa EIVA telah memenuhi syarat dan mampu untuk:

- 1) Mencari Produk
- 2) Memesan Produk
- 3) Konfirmasi Pembayaran
- 4) Melihat Status Pesanan
- 5) Melihat Daftar Pesanan
- 6) Mengubah Status Pesanan

5. KESIMPULAN DAN SARAN

Kesimpulan

Kesimpulan yang dapat diambil dalam pengerjaan tugas akhir ini adalah dihasilkannya sistem bernama Edu4indo *Interactive Virtual Assistant* (EIVA). Sistem ini dapat digunakan sebagai alternatif lain dalam pencarian dan pemesanan produk pada situs *web e-commerce* yang dalam kasus ini milik Edu4indo.com. EIVA berinteraksi dengan *customer* menggunakan bahasa Indonesia kemudian ditambah ekspresi yang bervariasi

menjadikannya semakin interaktif sehingga dapat berfungsi layaknya *customer service*.

5.1. Saran

EIVA menggunakan AIML sebagai basis pengetahuannya. Semakin besar basis pengetahuannya, semakin banyak pertanyaan yang dikenalnya, maka akan semakin cerdas. Sehingga diharapkan *botmaster* dapat memperbanyak basis pengetahuan EIVA. Selain itu, diharapkan EIVA dapat menjadi sebuah modul Prestashop yang *portable, customizable* dan fleksibel, sehingga dapat digunakan di berbagai toko *online*.

6. DAFTAR PUSTAKA

- [1] Allan, Dennis. 2009. "System Analysis and Design with UML". Hoboken: Wiley.
- [2] Aloysius, SW. 2011. "Website Super Canggih dengan Plugin jQuery Terbaik". Jakarta Selatan: Mediakita.
- [3] Ariesto, HS. 2001. "Analisis dan Desain Berorientasi Objek". Yogyakarta: J&J Learning.
- [4] Arulkumaran, Kumaraswamipillai. 2007. "Java/J2EE Job Interview Companion".: Lulu.com.
- [5] Bayan, AS. 2007. "Chatbots: Are They Really Useful?": LDV-Forum.
- [6] Bernd, Bruegge and Allen H, Dutoit, Eds. 2004. "Object Oriented Software Engineering Using UML, Pattern, and Java 2nd Edition". Munich: Pearson Education.
- [7] Booch, G, et all. 2005. "The Unified Modelling Language User Guide 2nd Edition".: Pearson Education.
- [8] C, Simon. 2012. "Advanced Web Technologies".: University of Westminster.
- [9] Charles, MK. 2005. "The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference".: No Starch Press.
- [10] Eduardo, M. 2007. "Intelligent web navigation using virtual assistant". Granada: University of Granada.
- [11] Frans. 2002. "Membuat Web Dinamis dan Interaktif dengan CGI". Yogyakarta: Penerbit Andi.

- [12] Ian, S. 2000. "Software Engineering 6th Edition". Jakarta: Erlangga.
- [13] Joyce, C. 2002. "Natural Language Assistant: A Dialog System for Online Product Recommendation".: AI Magazine.
- [14] Macrae, Callum. 2013. "Learning From JQuery". Sebastopol: O'Reilly.
- [15] Martin, F. 2003. "UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition". Boston: Addison Wesley.
- [16] Militiadis, DL and Naeve, Ambjorn , Eds. 2006. "Intelligent Learning Infrastructure for Knowledge Intensive Organizations: A Semantic Web Perspective".: Idea Group Inc.
- [17] O'Docherty, Mike. 2005. "Object-Oriented Analysis and Design : Understanding System Development with UML 2.0". Chichester. England: John Wiley & Sons Inc.
- [18] Oracle, Corp. (2012) MySQL. [Online]. <http://dev.mysql.com>
- [19] Peranginangi, Kasiman. 2006. "Aplikasi Web dengan PHP dan MySQL". Yogyakarta: Penerbit Andi.
- [20] Rajib, M. 2009. "Fundamentals of Software Engineering 3rd Edition". New Delhi: PHI Learning Private Limited.
- [21] Richardson, L and S, Ruby , Eds. 2007. "RESTful Web Services". Sebastopol: O'Reilly.
- [22] S. Wallace, Richard. (2010, Dec.) Alicebot. [Online]. <http://alicebot.blogspot.com/2010/12/integrating-Pandorabots-with-third.html>
- [23] S. Wallace, Richard. 2005. "Be Your Own Botmaster".: ALICE A.I. Foundation.
- [24] S. Wallace, Richard. (2010) Pandorabots. [Online]. <http://www.pandorabots.com/botmaster/en/>
- [25] S. Wallace, Richard. 2009. "Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer". The Anatomy of A.L.I.C.E. ed. Epstein, Robert , Roberts, Gary , and Beber, Grace , Eds. San Diego. United States of America: Springer.
- [26] Samisa, A. 2008. "Restful PHP Web Services". Mumbai: PACKT Publishing.
- [27] Schmuller, Joseph. 2004. "Sams Teach Yourself UML in 24 Hours". United States of America: Sams Publishing.
- [28] Scott, W. Ambler. 2005. "The Elements of UML 2.0 Style". Cambridge: Cambridge University Press.
- [29] Sylvia, Q , "A Chatbot-based Interactive Question Answering System," in Proceedings of The 11th Workshop on The Semantics and Pragmatics of Dialogue. Trento, 2007, vol. XI.