



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODOLOGI PENELITIAN

3.1. Objek Penelitian

Objek penelitian yang akan digunakan dalam proses penelitian ini adalah staf *student service* dan mahasiswa Universitas Multimedia Nusantara. Objek tersebut dipilih sebagai variabel utama dalam mengukur tingkat keberhasilan *response* yang diberikan oleh program *chatbot* dan menilai kemiripan *chatbot* yang bertindak layaknya *Student Service Staff* dalam melayani pertanyaan mahasiswa.

3.1.1. Preparation

Environment disiapkan terlebih dahulu sebelum dilakukan proses *development* dengan menggunakan *command-line interface* dari Node.js sebagai *development server*. Ionic dan Cordova di *install* melalui *node.js command prompt* seperti pada gambar 3.1 dengan *syntax* “*npm install -g cordova ionic*” –*g* digunakan untuk *global installation*. Ionic *application* di *generate* menggunakan *cli* dengan *syntax* “*ionic start ChatbotModule blank*” yang akan membuat *directory* dengan nama *ChatbotModule* yang berisi *package standard ionic* yang menggunakan *blank template*. Kemudian dibuatlah beberapa *pages* dengan *syntax* “*ionic g page nama_halaman*”, pembuatan *pages* dilakukan untuk pembuatan

front-end application yang akan diakses sebelum pengguna mengakses *chatbot module*.

3.1.2. Chatbot Module

Implementasi modul *chatbot* dilakukan dengan memanfaatkan API dari Dialogflow. Dialogflow *web interface* disambungkan dengan *container* yang telah dibuat pada *front-end application* dengan *client token* yang ada pada *menu setting* pada *dashboard agent* Dialogflow, kemudian *intent* dibuat agar dapat menerima *input message* dari pengguna sesuai dengan klasifikasi konteks pada *keyword* yang berasal dari kalimat pesan yang diberikan oleh pengguna. *Agent* pada Dialogflow menggunakan algoritma *Rule-based keyword matching* untuk mendapatkan *keyword* dari konteks pembicaraan yang dilakukan oleh *knowledge base* yang dimiliki oleh tiap *agent* yang dibuat pada Dialogflow. *Knowledge base* dapat dikembangkan dengan menggunakan *training* pada Dialogflow yang telah disediakan dengan cara memberikan keterangan (anotasi) pada kalimat dengan suatu entitas yang dapat dibuat sendiri ataupun menggunakan *prebuilt entities* dari Dialogflow. Setiap *intent* yang telah dibuat dilakukan aktivasi *webhook* agar dapat memberikan rangkaian jawaban yang acak sesuai dengan *intent* yang telah teridentifikasi dan diakses oleh *front-end application* untuk diberikan *response* balasan dari *chatbot*. *Service component* pada Dialogflow akan menerima *input message*

dari pengguna sebagai *payload*, dan setelah menerima *response* yang sesuai dari *intent* pada Dialogflow maka *response* akan di *return* kepada *front-end application* yang telah disambungkan.

3.2. Tahap Pengembangan Sistem

Tahap pengembangan sistem, pada tahap ini pengembangan program *Chatbot* dapat dilakukan melalui beberapa metode. Metode-metode tersebut perlu dianalisa dari sisi kelebihan dan sisi kekurangannya, kemudian perlu dipilih metode apa yang memiliki tingkat kecocokan tertinggi dan memenuhi kebutuhan pengembangan program *Chatbot*. Dapat dilihat pada tabel 3.1 perbandingan dari beberapa metode yang dapat digunakan dalam penelitian ini:

Tabel 3.1. Perbandingan Metode Pengembangan

No.	Nama Metode	Kelebihan (+)	Kekurangan (-)
1.	Waterfall Model	Proses pengembangan dilakukan secara berurut dan terorganisir, sehingga dapat menghasilkan kualitas sebuah sistem yang baik karena tiap tahapan perlu terpenuhi sebelum dapat melanjutkan kepada tahap selanjutnya.	Proses memerlukan waktu pengembangan yang panjang, dan pengembangan tidak dapat dilakukan secara berulang sebelum produk selesai dihasilkan.
2.	Prototyping Model	Proses pengembangan mengurangi ambiguitas antara <i>customer</i> dengan <i>developer</i> , dan proses pengembangan mengurangi resiko	Proses pengembangan tidak fleksibel jika terjadi banyak perubahan, karena jangka waktu yang diharapkan sangat singkat maka algoritma yang dikembangkan

No.	Nama Metode	Kelebihan (+)	Kekurangan (-)
		yang dapat terjadi pada tingkat perangkat lunak.	biasanya tidak kompleks dan hanya bertujuan untuk menyelesaikan fungsi yang ada.
3.	Rapid Application Development Model	Proses pengembangan tergolong cepat, sehingga dapat menghemat waktu dalam keseluruhan fase pada proyek, metode ini dapat dikembangkan secara berulang karena menggunakan teknik <i>iterative</i> . metode ini juga dapat mengurangi kebutuhan yang berkaitan dengan biaya proyek.	Proses pengembangan yang tergolong cepat memerlukan proses Analisa yang cepat, maka dari itu jika Analisa yang dilakukan tidak akurat maka dapat membahayakan pengembangan proses kedepannya.
4.	Agile Model	Proses pengembangan ini dapat melakukan testing setiap saat dibutuhkan, cocok digunakan jika tidak memiliki estimasi waktu yang dibutuhkan, proses pengembangan dapat beradaptasi dengan perubahan yang ada dengan mudah.	Proses ini tidak begitu mementingkan dokumentasi, memerlukan konsistensi hasil akhir dari <i>customer</i> karena perubahan yang terjadi dapat memengaruhi hasil akhir yang membuat ketidaksesuaian pada harapan <i>customer</i> .

Sumber: (Rastogi, 2015)

Berdasarkan perbandingan metode-metode pada tabel 3.1, maka metode pengembangan modul *Chatbot* yang paling cocok untuk digunakan adalah metode *Agile*. Metode *Agile* merupakan salah satu

dari beberapa metode yang digunakan dalam pengembangan software. *Agile method* adalah jenis pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dan pengembang terhadap perubahan dalam bentuk apapun. (Hyde, 2010). *Agile Development Method* memiliki prinsip dimana komunikasi antara anggota tim, antara orang-orang teknis, antara developer dan managernya tergolong sangat penting. Ciri lain dari *Agile Method* adalah klien menjadi bagian dari tim pembangun *software*. Ciri-ciri ini didukung oleh 12 prinsip yang ditetapkan oleh *Agile Alliance*. Menurut *Agile Alliance*, 12 prinsip ini perlu dilakukan jika ingin berhasil dalam penerapan *Agile Software Development*:

1. Kepuasan klien adalah prioritas utama dengan menghasilkan produk lebih awal dan terus menerus.
2. Menerima perubahan kebutuhan, sekalipun diakhir pengembangan.
3. Penyerahan hasil/software dalam hitungan waktu beberapa minggu sampai beberapa bulan.
4. Pihak bisnis dan pengembang harus bekerja sama setiap hari selama pengembangan berjalan.
5. Membangun proyek dilingkungan orang-orang yang bermotivasi tinggi yang bekerja dalam lingkungan yang mendukung dan yang dipercaya untuk dapat menyelesaikan proyek.
6. Komunikasi dengan berhadapan langsung adalah komunikasi yang efektif dan efisien.

7. Software yang berfungsi adalah ukuran utama dari kemajuan proyek.
8. Dukungan yang stabil dari sponsor, pembangun, dan pengguna diperlukan untuk menjaga perkembangan yang berkesinambungan.
9. Perhatian kepada kehebatan teknis dan desain yang bagus meningkatkan sifat *agile*.
10. Kesederhanaan penting.
11. Arsitektur, kebutuhan dan desain yang bagus muncuk dari tim yang mengatur dirinya sendiri.
12. Secara periodik tim evaluasi diri dan mencari cara untuk lebih efektif dan segera melakukannya. (12 Principles Behind the Agile Manifesto)

Teknik dari *Agile Development Methodology* yang digunakan ialah *Scrum*.

Teknik *Scrum* merupakan salah satu framework pengembangan perangkat lunak yang termasuk dalam *Agile development*. Dalam prosesnya, pengembangan dibagi ke dalam beberapa iterasi kecil yang disebut dengan *sprint*. Teknik *Scrum* merupakan metode yang menerapkan konsep gabungan antara pendekatan *iterative* dan *incremental*. Suatu *sprint* merupakan iterasi dan semakin bertambahnya *sprint* maka fitur yang diimplementasikan semakin banyak (*incremental*). (Schwaber, 1997)

Fase-fase pada *Scrum Development Framework* antara lain:

1. *Initiate*

Fase ini merupakan tahap awal dari proses pengembangan dengan *Scrum*. Fase ini meliputi pembentukan tim, pembuatan *project vision*, dan penentuan *product backlog* yang dimulai dari penentuan modul-modul apa yang perlu dibuat untuk aplikasi “UMN *Bulletin*” dan diurutkan berdasarkan prioritas modul tersebut. Untuk mendukung pembuatan *project vision* pada *project* pembuatan “UMN *Bulletin*” maka akan dilakukan pengumpulan data dan analisa kebutuhan dari pihak *Student Service* dalam melayani kebutuhan informasi mahasiswa Universitas Multimedia Nusantara.

2. *Plan and Estimate*

Pada fase ini dilakukan perencanaan untuk memulai pelaksanaan *sprint*, meliputi penulisan *user story*, penjabaran *task* pada tiap *user story*, melakukan estimasi nilai terhadap setiap *user story* dan *task*, serta menentukan *sprint backlog*.

3. *Implement*

Fase ini merupakan tahap untuk mengeksekusi setiap *task* yang telah didefinisikan serta melakukan aktivitas-aktivitas untuk membentuk produk. Pada tahap ini juga terdapat *daily standup meeting*, dimana tim melakukan evaluasi apa yang telah dilakukan oleh anggota tim selama jangka waktu tertentu. Pada saat tersebut juga

dapat diceritakan masalah apa saja yang ditemui selama melakukan implementasi.

4. *Review and Retrospect*

Pada tahap ini dilakukan *review* terhadap hasil pekerjaan tim (*deliverable product*) selama satu sprint. Penilaian tidak hanya dilakukan pada hasilnya, tetapi juga pada proses bekerja tim yang dinilai sehingga dapat menentukan apa saja yang perlu ditingkatkan untuk proses pengembangan pada *sprint* berikutnya. *Sprint* dianggap berhasil apabila pada *sprint* tersebut dihasilkan (*deliverable product*) yang sesuai dengan *acceptance criteria* yang telah ditentukan dan disetujui oleh *product owner*.

5. *Release*

Pada tahap ini produk yang telah memenuhi seluruh *acceptance criteria* dikirimkan kepada klien. Selain itu juga dilakukan dokumentasi mengenai pelajaran apa saja yang didapat selama melakukan pengembangan. Dan pada tahap ini dapat dilakukan riset untuk mengecek keandalan produk yang telah dibuat dengan dilakukannya *acceptance test*.

Acceptance test dapat dilakukan dengan melakukan pengumpulan data. Metode yang digunakan untuk mengumpulkan data dalam

penelitian ini adalah penyebaran *User Acceptance Test* untuk mengumpulkan data kuantitatif dari responden. *User Acceptance Test* merupakan sebuah teknik pengumpulan data penerimaan pengguna akan objek yang diuji dengan cara memberi serangkaian pertanyaan tertulis dan responden akan menjawab pertanyaan tersebut sesuai dengan tingkat jawaban. Target dari responden kuesioner dalam penelitian ini adalah *Student Service Staff* dan mahasiswa Universitas Multimedia Nusantara yang akan diambil menggunakan teknik *proportional stratified random sampling* bagi mahasiswa. Hasil dari *user acceptance test* yang telah dijawab oleh responden akan dikumpulkan dan digunakan kembali untuk pengembangan aplikasi.

