Implementasi Pohon Keputusan Dengan Natural Languange Understanding Dalam *Chatbot*

Muhammad Akram Al Bari 13519142
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13519142@std.stei.itb.ac.id

Abstrak*— Dalam beberapa dekade terakhir, teknologi telah mengalami kemajuan yang pesat. Akibatnya, lahirlah disipin-disiplin ilmu baru yang kemudian juga melahirkan banyak penemuan-penemuan baru. Salah satunya adalah Artificial Intelligence (AI) atau kecerdasan buatan. Dengan berkembangnya AI, mesin komputer menjadi semakin 'cerdas', yang ditunjukkan dengan mulai mampunya komputer untuk melakukan prediksi-prediksi, memberikan pilihan-pilihan ketika dihadapkan pada kasus-kassu tertentu. Salah satu implementasi AI yang cukup sering kita temui dalam keseharian adalah Chatbot. Chatbot merupakan kecerdasan buatan yang mampu mensimulasikan percakapan dengan pengguna dalam Natural Language. Chatbot banyak dimanfaatkan dalam dunia industri dewasa ini untuk membantu memberikan pengarahan kepada pengguna dalam menjalankan aplikasi, website, dan lain sebagainya. Dalam pengimplementasian chatbot, biasanya dibuat pohon keputusan yang nantinya akan menjadi 'otak' dari chatbot yang digunakan.

Keywords*—Chatbot, Artificial Intelligence, Pohon Keputusan, *Natural Language

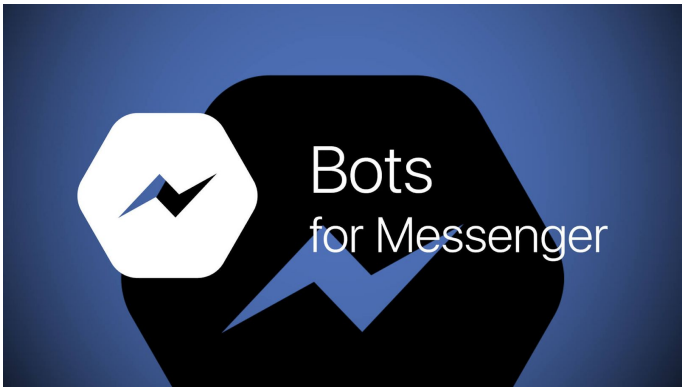
I. PENDAHULUAN

Saat ini, penggunaan teknologi komputer telah ada di berbagai lini kehidupan. Di industri, kantor-kantor, instansi pendidikan, bahkan di tiap-tiap rumah, semuanya memiliki komputer. Salah satu pemanfaatan komputer yang saat ini sering digunakan adalah dengan hadirnya *Artificial Intelligence (AI)* atau biasa disebut juga kecerdasan buatan.

AI merupakan hasil dari majunya teknologi kita saat ini. Secara sederhana, dengan adanya AI, komputer menjadi mampu untuk melakukan keputusan-keputusan atau memberikan informasi yang berbeda-beda sesuai dengan apa yang diberikan oleh pengguna. Dalam implementasinya, untuk membuat AI diperlukan data-data yang sangat banyak, *Big Data*, yang nantinya dari data-data ini lah *kecerdasan* AI dapat dibuat. Hadirnya AI juga membuat banyak implementasi-implementasi teknologi yang memanfaatkan AI, salah satunya adalah Chatbot.

Chatbot merupakan suatu aplikasi yang mampu mensimulasikan percakapan dengan pengguna dalam *Natural Languange* atau bahasa sehari-hari. Kemampuan Chatbot untuk bisa melakukan percakapan ini tidak lepas dari hadirnya AI. Dalam kehidupan sehari-hari, pemanfaatan chatbot sangat luas, mulai dari dunia hiburan, sampai dengan dunia industri.

Pemanfaatan chatbot misalnya dapat kita lihat pada *online messaging apps* seperti Facebook Messenger yang bisa kita unduh pada ponsel Android masing-masing.

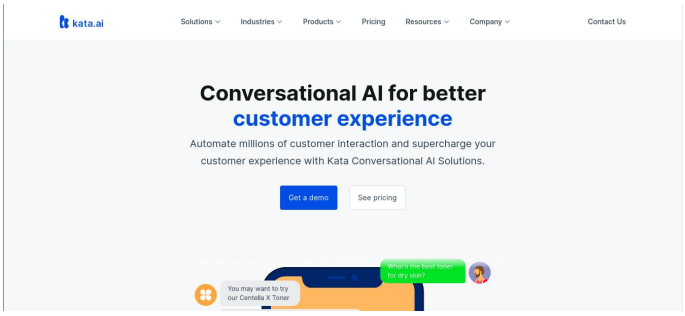


Gambar 1. chatbot aplikasi Facebook Messenger
Sumber:
https://marketingland.com/facebook-messenger-adds-option-chat-bots-avoid-chatting-208255

Dalam hal ini, pengguna Messenger mampu berkomunikasi dengan chatbot yang nantinya chatbot ini akan memberikan layanan jawaban, apabila pengguna memberikan *query* pertanyaan yang dikenali oleh chatbot.

Selain itu, chatbot pun saat ini telah menjadi suatu komoditi yang diperjualbelikan di dunia industri. Kita bisa mengambil contoh dari Perusahaan Kata.ai. Kata.ai adalah sebuah start-up yang bergerak di bidang pengadaan layanan chatbot, yang nantinya dapat langsung digunakan oleh customer. Produk chatbot dari Kata.ai ini banyak dimanfaatkan dalam dunia industri. Hal ini bisa menjadi parameter bahwa chatbot telah menjadi suatu alat yang sangat bermanfaat dalam menjalankan bisnis pada dewasa ini.

Dengan adanya chatbot, suatu perusahaan dapat memberikan layanan kepada customernya tanpa harus mempekerjakan manusia untuk memberikan tanggapan kepada pengguna. Bahkan saat ini, chatbot juga telah banyak dikembangkan untuk menjadi alat bantu belajar di dunia pendidikan.

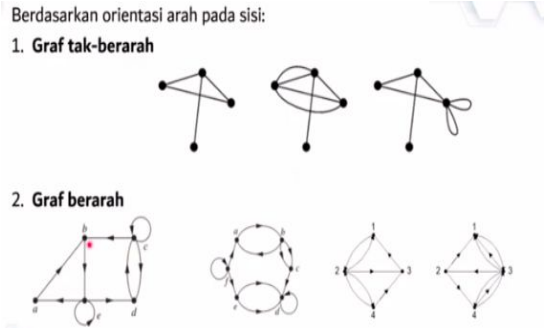


Gambar 2. Produk chatbot dari Kata.ai

II. LANDASAN TEORI

A. Graf

Graf adalah himpunan simpul-simpul (*vertices*) dan juga himpunan sisi (*edges*) yang menghubungkan sepasang simpul. Graf memiliki beberapa jenis yang tiap-tiap jenisnya dibedakan berdasarkan kriteria tertentu. Berdasarkan orientasi arah pada sisi graf, graf dibedakan menjadi dua jenis, yakni Graf Tak-Berarah dan Graf Berarah.

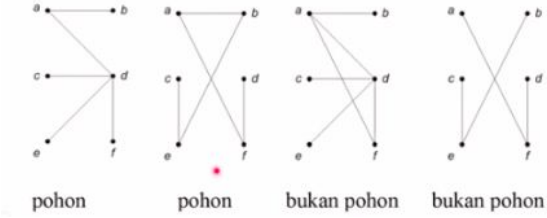


Gambar 3. Jenis-jenis graf
Sumber: http://informatika.stei.itb.ac.id

Kita juga mengenal beberapa terminologi khusus yang berkaitan dengan graf. Seperti Lintasan, Sirkuit, dan Keterhubungan. Dalam sebuah graf G, barisan berselang-seling simpul-simpul dan sisi-sisi disebut sebagai Lintasan. Sirkuit, ialah Lintasan yang berawal dan berakhir pada simpul yang sama. Sedangkan Keterhubungan, suatu graf disebut terhubung jika untuk setiap pasang simpul v_i dan v_j dalam himpunan V, terdapat lintasan dari v_i ke v_j.

B. Pohon

Secara definisi, pohon adalah graf tak-berarah terhubung yang tidak mengandung sirkuit



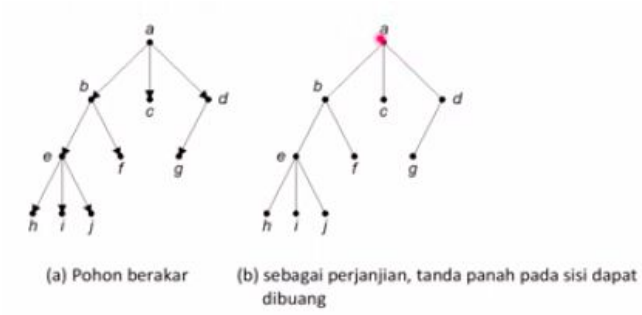
Gambar 4. Contoh Pohon
Sumber: http://informatika.stei.itb.ac.id

Misalkan G = (V,E) adalah graf tak-berarah yang memiliki n buah simpul. Maka G adalah pohon jika memenuhi sifat-sifat berikut:

1. G adalah pohon.
2. Setiap pasang simpul di dalam G terhubung dengan lintasan tunggal.
3. G terhubung dan memiliki m = n - 1 buah sisi.
4. G tidak mengandung sirkuit dan memiliki m = n -1 buah sisi.
5. G tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya satu sirkuit.
6. G terhubung dan semua sisinya adalah jembatan

C. Pohon Berakar (*rooted tree*)

Pohon berakar adalah pohon yang satu buah simpulnya diperlakukan sebagai akar dan sisi-sisinya diberi arah sehingga menjadi graf terarah.



Gambar 5. Pohon Berakar
Sumber: http://informatika.stei.itb.ac.id

dalam pohon berakar, kita mengenal terminologi-terminologi berikut:

1. Anak (*child*) dan Orangtua (*parent*)
pada gambar di atas, b,c, dan d adalah anak dari simpul a, a adalah orangtua dari simpul b,c, dan d
2. Lintasan (path)
Lintasan dari a ke j adalah a,b,e,j. Panjang lintasan dari a ke j adalah 3.
3. Saudara kandung (*sibling*)
f adalah saudara kandung e, tetapi g bukan saudara kandung e, karena orangtua mereka berbeda.
4. Upapohon (*subtree*)
Pada dasarnya, pohon adalah struktur rekursif yang didefinisikan dengan dirinya sendiri. Suatu pohon adalah kumpulan dari pohon-pohon lainnya sehingga kita bisa memiliki upapohon.
5. Derajat (*degree*)
Derajat sebuah simpul adalah jumlah upapohon (atau jumlah anak) pada simpul tersebut. Derajat a adalah 3, derajat b adalah 2, derajat d adalah satu, dan derajat c. Derajat maksimum dari semua simpul yang ada pada suatu pohon, adalah derajat dari pohon itu sendiri.
6. Daun (*leaf*)
Simpul yang berderajat nol (atau tidak mempunyai anak) disebut sebagai daun. simpul h,i,j,f,c, dan g adalah daun.
7. Simpul Dalam (*internal nodes*)
Simpul yang mempunyai anak disebut simpul dalam.

Simpul b, d, e, g, dan k adalah simpul dalam.

8. Aras (*level*) atau tingkat

Aras adalah penomoran berdasarkan letak suatu simpul terhadap simpul utama (simpul paling atas) dalam suatu pohon. Penomoran untuk simpul utama dimulai dari 0, dan bertambah 1 seiring dengan perpindahan ke anak-anak yang ada pada pohon.

9. Tinggi (*height*) atau Kedalaman (*depth*)

Aras maksimum dari suatu pohon disebut tinggi atau kedalaman pohon tersebut.

D. Pohon Terurut

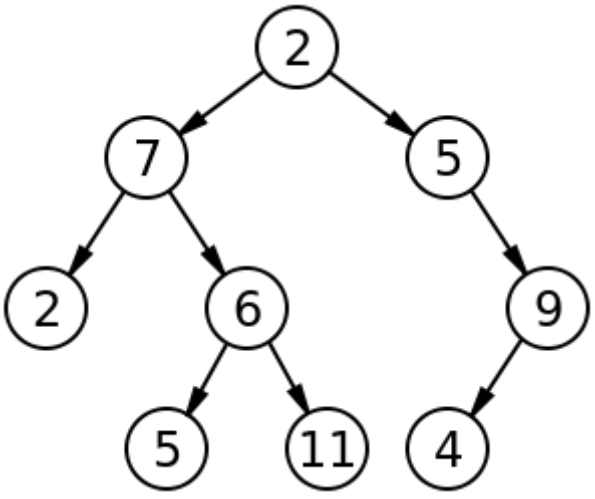
Pohon berakar yang urutan anak-anaknya diperhatikan disebut sebagai pohon terurut (*ordered tree*).

E. Pohon N-Ary

Pohon berakar yang setiap simpul cabangnya mempunyai paling banyak *n* buah anak disebut pohon *n-ary*. Pohon *n-ary* dikatakan teratur atau penuh jika setiap simpul cabangnya mempunyai tepat *n* anak.

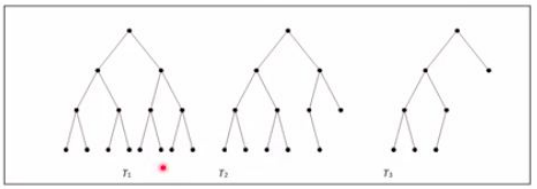
F. Pohon Biner

Pohon biner merupakan pohon *n-ary* dengan $n = 2$. Pohon biner dapat dikatakan sebagai pohon yang paling penting karena banyak pengaplikasiannya, terutama dalam ranah informatika. Setiap simpul di dalam pohon biner mempunyai paling banyak dua buah anak. Anak pada pohon biner dibedakan antara anak kiri (*left chlid*) dan anak kanan (*right child*). Karena ada perbedaan urutan anak, pohon biner merupakan pohon terurut.



Gambar 6. Pohon Biner
Sumber: http://informatika.stei.itb.ac.id

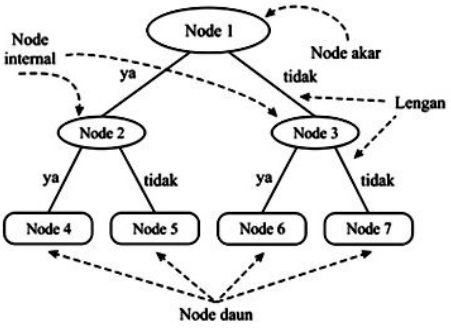
Dalam terminologi pohon biner, kita juga mengenal istilah Pohon Biner Seimbang. Pohon Biner Seimbang adalah pohon yang tinggi upapohon kiri dan tinggi upapohon kanannya seimbang, yaitu berbeda maksimal satu.



Gambar 7. Pohon biner seimbang dan tidak seimbang
Sumber: http://informatika.stei.itb.ac.id

G. Pohon Keputusan

Pohon Keputusan adalah pohon yang digunakan sebagai prosedur penalaran untuk mendapatkan kesimpulan atau informasi dari suatu *query* yang dimasukkna. Pohon yang dibentuk tidak selalu berupa pohon biner. Jika dalam data set menggunakan dua macam nilai kategorikal maka bentuk pohon keputusan yang didapatkan adalah pohon biner. Jika data set menggunakan lebih dari dua macam nilai, maka bentuk pohon yang didapatkan bukanlah pohon biner.



Gambar 8. Contoh Pohon Keputusan
Sumber: http://informatika.stei.itb.ac.id

Simpul pada pohon keputusan merepresentasikan *conditional* yang akan dievaluasi, dan sisi merepresentasikan hasil evaluasi. Hasil akhir dari pohon keputusan akan berada pada simpul daun, yang artinya tidak ada lagi *conditional*. Pohon keputusan berguna untuk memecah suatu kasus yang membutuhkan pengambilan keputusan yang kompleks agar menjadi lebih sederhana dengan melakukan eliminasi terhadap kemungkinan yang tidak diperlukan, sehingga pengambilan keputusan akan menjadi lebih jelas dan dapat dikaitkan secara langsung dengan solusi permasalahan.

Pada gambar di atas, pembacaan pohon keputusan dimulai dari atas ke bawah. Simpul yang berada pada posisi paling atas adalah simpul akar. Simpul yang ada di bawahnya dinamakan simpul keputusan. Cabang-cabang yang mengarah ke arah kanan dan kiri dari cabang keputusan adalah alternatif-alternatif keputusan yang dapat dipilih.

H. Natural Language Understanding

Natural Language Understanding (NLU) adalah sub topic dari Natural Language Processing. NLU melibatkan proses memecah persoalan dalam bahasa

manusia, bahsa sehari-hari, menjadi bahasa yang bisa dipahami oleh mesin. NLU menggunakan aturan gramatikal dan syntax dasar untuk memahami konteks atau maksud dari bahasa manusia, natural language. Tujuan dari NLU adalah untuk memahami bahasa tertulis atau lisan yang digunakan oleh manusia pada umumnya.

NLU digunakan dalam NLP seperti proses rekognisi bahasa, dan proses analisis.

III. IMPLEMENTASI POHON KEPUTUSAN

A. Pohon Keputusan Chatbot

Pohon keputusan adalah salah satu komponen yang sangat penting dalam mengimplementasikan chatbot. Pohon keputusan akan mendefinisikan bagaimana chatbot menangani kasus-kasus yang nantinya akan diberikan oleh pengguna. Apabila kita tidak meluangkan cukup waktu dan usahan untuk menyusun algoritma pohon keputusan chatbot yang baik dan mangkus, maka chatbot yang dihasilkan bisa jadi akan bekerja kurang baik

Sebagai contoh, chatbot yang didesain dengan pohon keputusan yang kurang baik, akan memiliki kemungkinan besar untuk gagal menangani *query* yang diajukan pengguna dengan benar. Hal ini tentunya akan mempengaruhi *feedback* dari pengguna. Secara ringkas, poin-poin yang perlu diperhatikan apabila pohon keputusan tidak diimplementasikan dengan baik dan benar adalah:

1. Chatbot akan gagal memberikan informasi yang sesuai dengan *query*
2. Chatbot memberikan tanggapan, namun informasi yang diberikan chatbot tidak sesuai dengan *query* yang dimasukkan.
3. Apabila percabangan pada pohon keputusan terlalu pendek, maka chatbot hanya akan mampu mengenali pola-pola yang terbatas pula.
4. Namun bila terlampau panjang, maka chatbot bisa jadi malah akan memberikan respons yang terlalu bertele-tele dan tidak sesuai dengan apa yang dibutuhkan oleh pengguna. Chatbot dengan pohon keputusan yang baik, akan mampu membawa pengguna kepada *experience* yang baik pula dan mampu menghasilkan *output* sesuai dengan yang diharapkan.

B. Desain Chatbot dengan NLU

Berdasarkan referensi bacaan yang digunakan oleh penulis, alur chatbot dapat dibagi menjadi dua. Yang pertama adalah *floating flow*. Chatbot dengan *floating flow* biasanya tidak memiliki diagram pohon keputusan yang kompleks. Secara struktur, biasanya chatbot ini terbagi menjadi:

1. *Greetings*
2. Daftar topik/kasus yang akan ditangani
3. *Fallbacks*

Chatbot jenis ini memungkinkan pengguna untuk mendapatkan objektif sederhana secara langsung.

Gambar 9. Representasi Chatbot
Sumber: https://chatbotslife.com/chatbot

Tipe yang lain dari chatbot adalah tipe dengan diagram yang miirip seperti pohon yang disebut sebagai *layerd flow*. Tipe ini biasanya didesain untuk chatbot yang dibuat agar mampu menganani cakupan topik dan konten yang luas.

Kemudian, kita akan bersinggungan dengan topik bahasan terkait penggunaan *keyword* pada desain chatbot atau memilih untuk menggunakan *Natural Language Understanding*. Apabila dianalisis lebih lanjut, sebetulnya chatbot berbasis *keyword* memang jauh lebih mudah untuk diimplementasikan. Namun implementasi menggunakan cara ini, akan menmui suatu persoalan yang cukup sulit untuk dipecahkan. Yakni, chatbot yang diimplementasikan berbasis *keyword* akan menemui masalah overlapping. Kita tidak dapat mengimplementasikan suatu kata yang sama untuk dua atau lebih kasus yang berbeda. Sehingga, pengimplementasian dengan NLU menjadi pilihan yang lebih baik karena NLU memberikan fleksibilitas dalam melakukan mapping untuk state-state chatbot yang berbeda.

Setelah kita melakukan pemilihan state mana saja yang memerlukan NLU, maka kita perlu menyiapkan data untuk chatbot kita. Data ini merupakan koleksi daru kalimat yang mungkin dimasukkan oleh pengguna.

Data yang telah dikumpulkan, perlu kita klasifikasikan kedalam tiap-tiap state respons. Dengan kumpulan kalimat ini, kita bisa mendefinisikan komponen yang diperlukan untuk membuat NLU.

Kita perlu mengidentifikasikan maksud dari pengguna yang bisa ditangani oleh chatbot. Secara garis besar, terdapat dua jenis maksud yang biasa kita temui: yakni *action* dan *redo*. Langkah selanjutnya adlah menentukan entitas mana yang sesuai dengan maksud yang disampaikan oleh pengguna. Kita dapat menentukan object yang dimaksud dalam query yang dimasukan pengguna.

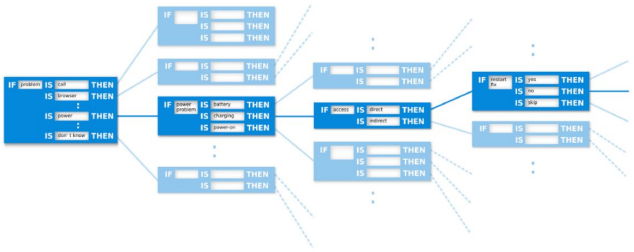
Langkah terakhir, adalah memastikan bahwa setiap state dalam chatbot memiliki map NLnya masing-masing. Setelah semua set-up tersebut berhasil dilakukan, maka NL siap digunakan dalam penyusunan pohon keputusan untuk chatbot. Untuk lebih membuat NL semakin mutakhir, maka akan semakin baik bila kita memiliki banyak input data yang dapat diberikan ke program supaya NL bisa terus terupdate dan dapat menyesuaikan diri secara lebih baik lagi terhadap variasi query dari pengguna.

C. Memadukan NLU dan Pohon Keputusan

Setelah kita selesai mempersiapkan segala sesuatu yang berkaitan dengan NLU, sekarang saatnya kita untuk menggabungkan NLU ini dengan pohon keputusan.

Pada dasarnya, yang akan kita lakukan adalah tidak jauh berbeda dengan ketika kita membuat suatu algoritma *if-else*,

yakni pohon keputusan kita akan melakukan pengecekan terhadap kondisi-kondisi tertentu. Hanya saja, saat ini kondisi yang akan dicek bukanlah berbasis keyword, melainkan suatu state tertentu yang telah kita definisikan saat membangun NLU.



Gambar 10. Gambaran Kompleksitas Pohon
Sumber: https://chatbotslife.com/chatbot

Gambar di atas adalah contoh representasi pohon keputusan apabila kita menggunakan keyword. Bisa diperhatikan, bahwa akan banyak sekali kasus yang ditangani dan pastinya, persoalan berupa satu keyword hanya bisa menangani satu kasus masih terjadi jika kita menggunakan keyword.

Mengimplementasikan NLU merupakan alternatif yang lebih baik karena sekarang, kondisonal yang perlu kita cek tidak lagi berbasis keyword, melainkan sesuai dengan assigment NLU yang telah kita rancang.

Hal ini akan membuat chatbot yang dibuat menjadi lebih diverse dan bisa dimanfaatkan dengan lebih baik untuk menangani beragam kasus.

Meskipun begitu, pada akhirnya pemilihan penggunaan pohon keputusan ini memiliki konstrain yang tidak bisa dihindari. Bahwa untuk menangani setiap kasus, maka kompleksitas dari pohon keputusan ini akan terus meningkat seiring dengan banyaknya kasus yang ingin diimplementasikan untuk bisa ditangani.

Jika ditinjau dari sisi kompleksitasnya, maka banyaknya opsi pohon yang akan ditambahkan (dengan mengasumsikan bahwa pohon seimbang) akan berkembang mengikuti rumusan sebagai berikut

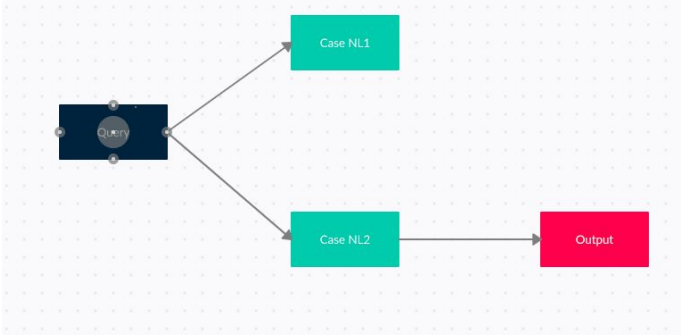
$$\text{Kompleksitas: } a^n$$

Keterangan:

a : banyaknya opsi tiap pohon

n: kasus yang ingin ditangan

Secara sederhana, hasil akhir dari pohon keputusan yang telah diintegrasikan dengan NLU adalah seperti gambar di bawah ini



Gambar 11. Pohon Keputusan Akhir

Perbedaan yang jelas teramati pada pohon ini adalah, apabila pada pengimplementasian sebelumnya test casenya adalah keyword, pada pohon ini test casenya berupa hasil olaha dari NL yang telah dibuat sebelumnya. Keuntungan dengan metode seperti ini adalah chatbot yang dihasilkan mampu menangani lebih banyak kasus.

Meskipun begitu, pengimplementasian dengan menggunakan NL ini, bila ditinjau dari sisi persiapannya, jauh lebih sulit dibandingkan dengan pengimplementasian langsung menggunakan keyword. Karena resource untuk *set-up* nya jauh lebih besar dikarenakan kita memerlukan sekumpulan data terlebih dahulu untuk bisa menyiapkan NL yang kemudian bisa digunakan dalam pohon keputusan.

IV. KESIMPULAN

Chatbot yang baik sangat bergantung dari desain pohon keputusan pada chatbot tersebut. Apabila pohon keputusan pada chatbot kurang baik, maka chatbot pun akan memberikan hasil yang tidak maksimal.

Pada bab sebelumnya, penulis telah memaparkan secara sederhana, bagaimana proses implementaswi dari pohon keputusan dalam mendesain suatu chatbot.

Secara fundamental, pengimplementasian pohon keputusan untuk chatbot adalah seperti yang sudah dijelaskan pada bab selanjutnya. Namun, contoh-contoh yang diberikan oleh penulis memang masih contoh yang sederhana, sehingga masih kurang representatif terhadap pohon keputusan yang digunakan pada chatbot di dunia industri.

Dalam merancang pohon keputusan untuk chatbot, perlu diperhatikan bahwa pohon keputusan jangan sampai terlalu pendek, karena akan memberikan batasan terhadap kasus yang bisa ditangani. Namun, juga jangan terlalu panjang karena prosesnya nanti akan menjadi terlalu bertele-tele

Pada akhirnya, pohon keputusan dalam mengimplementasikan chatbot merupakan alat bantu yang memudahkan kita untuk melakukan mapping terhadap kondisi-kondisi yang ingin ditangani, disesuaikan dengan query atau masukan dari pengguna.

Chatbot dengan pohon keputusan dapat didesain berdasarkan NL sebagai alternatif dari desain yang berdasarkan pada keyword. Membuat design NL ini dapat dimulai dengan membuat outline atau gambarn kasar dari alur kerja chatbot yang ingin dirancang. Perancangan outline ini juga memanfaatkan pohon keputusan dalam hal pemodelannya. Setelah desain dari NL telah berhasil dibentuk, maka kita juga masih perlu untuk membuat mapping NL tersebut kepada aksi yang sebenarnya ingin dieksekusi. Keuntungan dari pembuatan berbasis NL ini adalah chatbot menjadi lebih *maintanable* karena ia dapat terus menyesuaikan diri dengan mendapatkan input data yang lebih banyak lagi.

V. UCAPAN TERIMA KASIH

Puji syukur kehadirat Tuhan Yang Maha Esa, karena rahmatNya, penulis dapat menyelesaikan karya tulis ilmiah ini. Penulis juga berterima kasih kepada seluruh pihak yang baik

secara langsung, maupun tidak langsung telah memberikan dorongan motivasi dan semangat, sehingga membantu penulis dalam menyelesaikan karya tulis ini. Penulis juga mengucapkan terima kasih yang sebesar-besarnya kepada Bu Harlili selaku dosen pengampu Mata Kuliah Matematika Diskrit di Kelas 02. Atas bimbingan beliaulah akhirnya penulis dapat memilki bekal untuk menuliskan karya tulis ini.

REFERENSI

[1] Lamber, Martin. Chatbot Decision Trees. Diakses pada Desember 10, 2020, dari https://chatbotslife.com/chatbot-decision-trees-a42ed8b8cf32

[2] Lobo, Joe. What is a decision tree?. Diakses pada Desember 10, 2020, dari https://chatbotslife.com/chatbot-decision-trees-a42ed8b8cf32

[3] Rembulan. How I Design NL for Chatbots With Decision Tree Model of Flow. Diakses pada Desember 10, 2020, dari https://medium.com/swlh/how-i-design-nl-for-chatbots-with-decision-tree-model-of-flow-83b0566edca5

[4] Khanna, Anirudh., Pandey, Bishwajeet., Vashista, Kusagra., Kalia, Kartik., Pradeepkumar, Bhale., Das Terath. A Study of Today's A.I. through Chatbots and Rediscovery of Machine Intelligence. Diakses pada Desember 10, 2020, dari https://www.researchgate.net/publication/287110028_A_Study_of_Today's_AI_through_Chatbots_and_Rediscovery_of_Machine_Intelligence

[5] Munir, R. Pohon. Diakses pada Desember 10, 2020, dari http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/matdis20-21.html

[6] Expert System Team. Chatbot: What is a Chatbot? Why are Chatbots Important?. Diakses pada Desember 10, 2020, dari https://www.expert.ai/blog/chatbot/

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11 Desember 2020

Muhammad Akram Al Bari 13519142