

Penerapan Pohon pada *Game* Berbasis *Artificial Intelligence*

Rifki Afina Putri (13511066)¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

13511066@std.stei.itb.ac.id

Abstrak — Pohon merupakan terapan khusus dari teori graf. Pohon merupakan suatu struktur data yang sangat penting. Banyak sekali penerapan pohon yang bermanfaat dalam pembuatan software misalnya saja dalam pembuatan *game*. Penerapan pohon yang sering digunakan adalah pohon keputusan dan pohon pencarian biner. Penerapan pohon ini sering digunakan pada *game* berbasis *Artificial Intelligence* seperti Akinator (*game* yang menjamur sejak tahun 2008). Akinator sendiri sebenarnya merupakan suatu pengembangan dari permainan '20 Questions' yang berkembang di Amerika pada abad ke-19. Karena itu, Akinator dan 20 Questions memiliki kemiripan dalam struktur permainan.

Kata Kunci — akinator, *artificial intelligence*, *game*, pohon keputusan, pohon pencarian biner.

I. PENDAHULUAN

Pada abad ke-19, permainan 20 Questions berkembang di Amerika. Permainan ini mencapai puncak popularitasnya pada sekitar akhir tahun 1940 ketika permainan ini diadaptasi menjadi sebuah program kuis mingguan dalam sebuah radio. Cara bermain 20 Questions sebenarnya sangat sederhana. Permainan ini dimainkan oleh dua orang. Satu orang harus memilih suatu subjek atau karakter. Pilihan karakter disini memiliki banyak sekali pilihan kemungkinan, orang yang terkenal, orang yang dikenalnya, atau bahkan dirinya sendiri. Kemudian satu orang lainnya harus menebak siapa orang yang dipikirkan oleh lawannya. Orang yang menebak dapat bertanya sebanyak 20 pertanyaan agar bisa menebak karakter yang dipikirkan lawannya dengan tepat. Untuk kasus yang sederhana, jawaban dari pertanyaan-pertanyaan tersebut berupa 'yes' atau 'no'. Untuk kasus yang lebih kompleks, jawaban juga bisa dikembangkan menjadi 'maybe' atau 'don't know'. Strategi dalam memilih pertanyaan yang akan ditanyakan pun sangat berpengaruh pada permainan ini.



Gambar 1.1 Logo Permainan 20 Questions

(Sumber Gambar:

http://en.wikipedia.org/wiki/Twenty_Questions)

Permainan ini menarik karena penebak harus memikirkan 20 pertanyaan yang efektif agar bisa menebak karakter yang dipikirkan lawannya secara tepat. Karena itu, pilihan pertanyaan merupakan suatu yang penting dalam permainan ini. Pertanyaan pertama biasanya merupakan pertanyaan yang umum, kemudian pertanyaan berikutnya merupakan pertanyaan yang dipilih berdasarkan jawaban 'yes' atau 'no' yang mengarah kepada kemungkinan pilihan karakter.

Permainan 20 Questions ini kemudian dikembangkan menjadi sebuah aplikasi bernama 'Akinator the Genie'. Permainan ini dapat dimainkan pada iPhone, iPad, tablet pc, dan mobile phone. Selain itu, Akinator juga dapat dimainkan pada web. Akinator sering kali disebut sebagai 'Web Genius' karena kemampuannya untuk menebak secara akurat. Sehingga tidak salah jika *game* ini dikategorikan sebagai *game* yang berbasis *Artificial Intelligence* karena memiliki tingkat kecerdasan yang hampir menyetarai manusia. Ibaratnya, Akinator ini dapat 'membaca' pikiran manusia.



Gambar 1.2 Screenshoot dari Game Akinator

(Sumber Gambar: <http://www.androidup.com/akinator-the-genie-il-genio-che-indovina-i-nostri-pensieri/>)

Cara bermain *game* Akinator ini memiliki banyak kemiripan dengan permainan *20 Questions*. Akinator dapat menebak siapa karakter yang sedang dipikirkan oleh pemain berdasarkan beberapa pertanyaan. Dalam permainan Akinator ini, karakter yang dipikirkan tidak terbatas. Karakter dapat berupa karakter sesungguhnya atau karakter fiktif, bahkan Akinator dapat menebak dengan tepat meskipun karakter yang kita pikirkan adalah diri kita sendiri.

Permainan dimulai dengan menanyakan 20 pertanyaan dengan pilihan jawaban *yes*, *no*, *don't know*, *probably*, dan *probably not*. Apabila jawaban telah ditemukan sebelum 20 pertanyaan ditanyakan, program akan secara otomatis 'menebak' jawaban dan menanyakan apakah tebakan tersebut benar. Apabila tebakan benar, maka permainan berakhir. Namun apabila tebakan salah, pemain dapat melanjutkan permainan atau menghentikan permainan sesuai dengan keinginannya. Jika kesalahan tebakan terjadi sebanyak tiga kali berturut-turut maka program akan meminta pemain untuk memasukkan nama karakter yang dipikirkannya untuk ditambahkan pada database.

II. TEORI DASAR

Berikut ini adalah beberapa teori dasar yang memiliki kaitan erat dengan permainan ini.

2.1 Pohon

Pohon adalah graf tak-berarah terhubung yang tidak mengandung sirkuit. Karena sebuah pohon tidak boleh mengandung sirkuit, pohon tidak boleh mengandung sisi ganda atau loop. Oleh karena itu, setiap pohon merupakan sebuah graf sederhana. (Rosen, 2007 : 683)

Salah satu jenis pohon adalah pohon biner. Pohon biner adalah pohon yang mempunyai paling banyak dua cabang. Pohon biner ini merupakan struktur yang penting dalam ilmu komputer karena terapan dari pohon biner sangat banyak dan aplikatif. Beberapa terapan pohon biner adalah pohon ekspresi, pohon keputusan, kode prefix,

kode Huffman, pohon pencarian biner, dan masih banyak lagi. Pada makalah ini, penulis akan membahas lebih rinci mengenai pohon keputusan dan pohon pencarian biner. Kedua terapan pohon ini merupakan dasar dari permainan ini untuk bekerja.

2.2 Pohon Keputusan

Pohon keputusan digunakan untuk memodelkan persoalan yang terdiri dari serangkaian keputusan yang mengarah kepada solusi. Tiap simpul dalam menyatakan keputusan, dan daun menyatakan solusi. (Rosen, 2007 : 698)

Pohon keputusan biasanya digunakan agar proses pengambilan keputusan yang kompleks dapat dibuat menjadi lebih sederhana. Pohon keputusan berguna untuk mengeksplorasi data, dan juga menemukan hubungan tersembunyi antara sejumlah variabel input dengan variabel target. Namun hasil dari kualitas keputusan tentu saja bergantung pada desain pohon keputusan tersebut.

Penggunaan metode pohon keputusan ini memiliki beberapa kelebihan dan kekurangan. Kelebihannya, daerah pengambilan keputusan yang sebelumnya kompleks dan global dapat diubah menjadi lebih sederhana dan spesifik, tidak memerlukan eliminasi perhitungan-perhitungan karena sampel diuji berdasarkan kriteria atau kelas tertentu, serta fleksibel untuk memilih fitur dari *internal node* yang berbeda.

Sedangkan kekurangan dari metode pohon keputusan ialah terjadinya *overlap* terutama ketika kelas-kelas dan kriteria yang digunakan berjumlah sangat banyak. Jumlah yang banyak ini juga menambah waktu pengambilan keputusan dan jumlah memori yang diperlukan. Selain itu, pengakumulasian jumlah error pada setiap tingkat bernilai besar, dan sulitnya untuk mendesain pohon keputusan yang optimal.

2.3 Pohon Pencarian Biner

Pohon Pencarian Biner (*Binary Search Tree*) adalah pohon biner yang simpulnya memiliki satu anak kiri dan satu anak kanan, tidak ada simpul yang memiliki lebih dari satu anak kiri atau kanan, dan setiap simpul memiliki kunci (*key*). (Rosen, 2007 : 696)

Ketentuan pengaturan kunci adalah semua simpul di upapohon kiri mempunyai kunci yang lebih kecil daripada kunci pada *parent*-nya, dan semua simpul di upapohon kanan mempunyai kunci yang lebih besar daripada kunci pada *parent*-nya. Struktur pohon dimaksudkan untuk memberikan akses yang cepat terhadap data di simpul. Pencarian selalu dimulai dari simpul akar dan nilai yang dicari akan dibandingkan dengan kunci. Pencarian akan berlanjut ke upapohon kiri atau kanan bergantung pada nilainya, apakah lebih besar atau lebih kecil dari nilai pada *parent*-nya.

Pohon pencarian biner ini sangat efektif karena merupakan gabungan antara algoritma pengurutan dan pencarian sehingga pencarian objek dapat dilakukan secara lebih efisien. Pohon pencarian biner ini merupakan

struktur data dasar yang biasanya digunakan untuk membentuk suatu struktur data yang lebih abstrak.

2.4 Artificial Intelligence

Artificial Intelligence (kecedasan buatan) merupakan cabang dari ilmu komputer yang mempelajari bagaimana sebuah komputer atau robot dapat berperilaku seperti manusia. Menurut H. A. Simon (1987), *Artificial Intelligence* adalah kawasan penelitian, aplikasi, dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas.

Ada banyak bahasa pemrograman yang digunakan untuk mengembangkan *Artificial Intelligence*, diantaranya IPL (bahasa pertama yang digunakan), LISP, Prolog, STRIPS, juga bahasa standar yang sering digunakan seperti bahasa C.

Beberapa aplikasi dari *Artificial Intelligence* diantaranya sistem pakar (*Expert System*), pengolahan bahasa alami (*Natural Language Processing*), pengenalan ucapan (*Speech Recognition*), robotika dan sistem sensor, *computer vision*, serta *game playing*. Pada mulanya, *Artificial Intelligence* dipusatkan pada pengembangan *game playing*.

Permainan berbasis *Artificial Intelligence* lebih mengacu kepada teknik yang digunakan pada komputer dan game agar dapat menghasilkan sebuah ilusi kecerdasan pada perilaku dari *non-player character*. *Game* berbasis *Artificial Intelligence* biasanya juga memiliki *gameplay* yang bagus.

Pencarian dan penelusuran merupakan salah satu teknik *Artificial Intelligence* yang paling penting. Karena itu, pemanfaatan pohon sangat berguna dalam pembuatan *game* yang berbasis *Artificial Intelligence*.

Ada beberapa beberapa cara pencarian yang dapat digunakan untuk mencari kemungkinan penyelesaian, yaitu Depth-First Search, Breadth-First Search, Hill-Climbing Search, Least-Cost Search, dan Best-Fist Search. Evaluasi dari hasil pencarian tersebut tentu akan sangat kompleks. Karena itu, teknik pencarian yang tepat tentu akan membuat program yang dibuat akan menjadi semakin efektif.

III. PENERAPAN POHON PADA AKINATOR

3.1 Cara Kerja Program

Seperti yang telah disebutkan sebelumnya, pada dasarnya permainan *20 Questions* dan Akinator memiliki cara kerja yang serupa. Oleh karena itu, pada bagian ini penulis akan lebih membahas mengenai Akinator.

Akinator memiliki database utama yang sangat besar berupa database karakter, pertanyaan, dan statistik. Permainan sangat ini mengandalkan teknik pencarian dan penelusuran database berdasarkan masukan jawaban dari pemain.

Pada awal permainan, program akan mulai bertanya pertanyaan dasar dan umum seperti '*Is your character a*

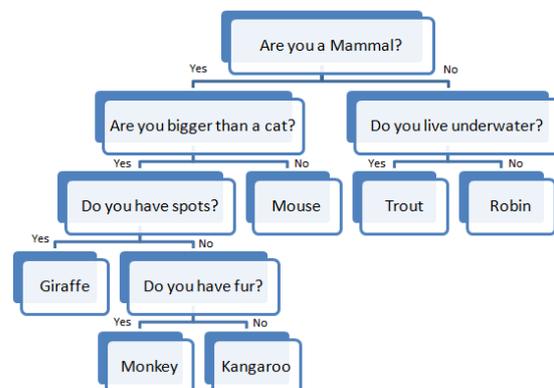
man?' atau '*Is your character real?*'. Pertanyaan tersebut akan dilanjutkan dengan pertanyaan-pertanyaan yang akan mengarah kepada kemungkinan karakter yang akan ditebak.

Misalnya saja, karakter yang kita pikirkan adalah seorang wanita. Maka kita menjawab '*no*' pada pertanyaan '*Is your character a man?*' Kemudian program akan merekam bahwa karakter adalah seorang wanita, lalu pertanyaan yang akan ditanyakan berikutnya adalah pertanyaan yang dikhususkan untuk karakter wanita.

Pada setiap pengulangan, program menanyakan pertanyaan yang akan mengeliminasi setengah dari kemungkinan pilihan jawaban. Secara matematis, jika jumlah kemungkinan jawaban adalah N, maka kita akan mendapatkan jawaban setelah $\log_2(N)$ pertanyaan. Maka 20 pertanyaan dapat memungkinkan program untuk membedakan 2^{20} atau 1.048.576 subjek. Dengan demikian, strategi paling efektif agar subjek berhasil ditebak secara akurat adalah dengan mengajukan pertanyaan yang dapat membagi sisa kemungkinan menjadi setengahnya pada setiap pertanyaan. Proses pencarian ini menggunakan prinsip dari pohon pencarian biner. Parameter untuk dapat menghasilkan pencarian yang efektif adalah seberapa cepat metode *search* tersebut menemukan penyelesaian yang baik.

Pohon keputusan juga merupakan struktur yang sangat bermanfaat untuk program ini. Pohon keputusan digunakan untuk menentukan pertanyaan terbaik pada setiap cabang untuk membedakan koleksi database yang direpresentasikan sebagai anak kanan dan kirinya. Desain pohon keputusan ini tentunya memiliki pengaruh yang besar terhadap keakuratan jawaban. Karena itu, dengan memahami cara membuat pohon keputusan yang baik, maka kita dapat membuat program seperti Akinator yang dapat menebak jawaban secara akurat.

Berikut ini adalah contoh dari penggunaan pohon keputusan, namun dalam bentuk yang masih sangat sederhana.



Gambar 3.1 Contoh Penggunaan Pohon Keputusan Sederhana

Dari contoh di atas, pertanyaan awal tentunya merupakan pertanyaan yang paling umum, kemudian pertanyaan akan semakin spesifik kepada pilihan jawaban

yang memungkinkan. Pada contoh ini, kemungkinan pilihan jawaban hanya dua, *yes* atau *no*. Pohon keputusan sebenarnya yang ada pada struktur program Akinator, tentu akan jauh lebih kompleks daripada contoh di atas, karena terdapat lebih banyak kemungkinan jawaban yaitu *yes*, *no*, *don't know*, *probably*, dan *probably not*. Pohon yang digunakan pun bisa jadi bukan merupakan pohon biner, tetapi pengembangannya. Mengingat begitu banyak dan kompleksnya struktur program dari Akinator ini.

Agar dapat memberikan jawaban yang akurat pada setiap permainan, maka kita juga harus memikirkan berapa peluang suatu objek akan dipilih ketika mendesain pohon. Karena beberapa objek mungkin akan lebih sering dipilih daripada objek yang lain.

Pada permainan ini, pemain juga dapat menambahkan database karakter. Namun, pemain hanya dapat menambahkan karakter yang terkenal. Di versi web, setiap kali permainan berakhir dan program menebak karakter, pemain dapat menambahkan karakter baru. Jika tebakan benar, pemain dapat memperbaiki nama karakter tersebut jika dirasa terdapat kesalahan nama pada database. Pemain hanya tinggal memilih opsi '*Suggest a New Name*', kemudian pemain akan diminta untuk mengisi nama yang baru dan deskripsi mengenai karakter tersebut, sehingga tidak akan bertukar dengan karakter lain yang mungkin memiliki nama yang sama. Nama yang baru akan diimplementasikan setelah dilakukan validasi oleh moderator. Jika program tidak menebak dengan benar, pemain juga dapat memasukkan karakter yang seharusnya, dengan cara yang sama.

Apabila program menemukan suatu kemiripan antara karakter yang kita tambahkan dengan karakter yang sudah ada dalam database, maka program akan memberikan list karakter. Jika karakter yang kita tambahkan berada pada list tersebut, maka kita hanya tinggal memilih karakter tersebut, dan proses penambahan karakter tidak akan dilakukan. Jika karakter tidak terdapat pada list, maka pemain diharuskan menginput kembali nama dan deskripsi karakter untuk ditambahkan.

Selain menambah karakter, pemain pun dapat menambahkan pertanyaan pada akhir permainan dengan memilih opsi '*Add a Question*'. Pemain akan diminta untuk memasukan pertanyaan dan kata kunci untuk pertanyaan tersebut. Jika program menemukan kemiripan dengan pertanyaan yang telah ada, maka program akan menampilkan list pertanyaan. Apabila pertanyaan yang ditambahkan telah ada pada list, maka kita hanya tinggal memilih pertanyaan tersebut dan pertanyaan baru tidak akan ditambahkan. Jika pertanyaan tidak ada pada list, maka pemain akan diminta untuk menambahkan lagi pertanyaan dan info untuk pertanyaan tersebut. Kemudian pertanyaan tersebut akan ditambahkan pada database setelah dilakukan verifikasi oleh moderator.

Adanya fitur penambahan karakter dan pertanyaan ini tentunya akan menambah kaya koleksi database yang dimiliki oleh Akinator. Semakin banyak permainan ini dimainkan, akan semakin meningkat pula kemampuan

Akinator untuk menebak, karena permainan ini menambah 'pengetahuan' dari pemain berdasarkan jawaban-jawaban yang direkam dalam sebuah statistik. Database permainan akan selalu diperbarui setiap kali permainan ini dimainkan. Maka *game* ini akan terus berkembang jika terus menerus dimainkan.

Jawaban dari pemain bukan berupa boolean karena kemungkinan jawaban tidak hanya *yes* dan *no*, tetapi juga terdapat kemungkinan jawaban *don't know*, *probably*, dan *probably not*. Oleh karena itu, jawaban dari pemain direkam dalam sebuah integer dengan interval dari 0 sampai 1. Misalnya saja *yes* bernilai 1, *probably* bernilai 0.75, *don't know* bernilai 0.5, *probably not* bernilai 0.25, dan *no* bernilai 0. Logika ini dikenal juga sebagai *fuzzy logic* (logika *fuzzy*).

Ada kemungkinan pohon keputusan yang ada pada program dibuat ketika program dimainkan. Sehingga struktur pohon merupakan struktur pohon yang dinamis, karena Akinator tidak selalu menanyakan pertanyaan yang sama pada setiap permainan. Terdapat suatu elemen *random* yang 'memilih' pertanyaan. Beberapa pertanyaan pun tidak harus selalu diaplikasikan secara tetap pada suatu karakter. Banyak juga orang yang berspekulasi bahwa algoritma yang digunakan untuk membangun pohon keputusan pada permainan ini adalah algoritma C4.5.

3.2 Simulasi Permainan

Simulasi ini merupakan simulasi permainan Akinator versi web (en.akinator.com).

Sebelum memulai permainan, pemain harus mendaftarkan diri terlebih dahulu dengan mengisi nama, umur, dan jenis kelamin. Sebenarnya, pengisian ini hanya untuk mencatat statistik jumlah pemain dan tidak memiliki pengaruh yang begitu besar kepada permainan. Kemudian pertanyaan akan muncul satu persatu dengan lima pilihan jawaban yaitu *yes*, *no*, *don't know*, *probably*, dan *probably not*. Pertanyaan dimulai dari pertanyaan yang paling umum sampai pertanyaan khusus yang semakin mengarah kepada pilihan karakter seperti yang telah dijelaskan pada bagian sebelumnya.

Pada simulasi kali ini, Akinator dapat menebak karakter setelah 25 pertanyaan. Ternyata karakter yang ditebak oleh Akinator sesuai dengan karakter yang dipikirkan oleh pemain. Permainan pun berakhir. Selanjutnya, terdapat beberapa opsi, yaitu kembali bermain, melihat *game report*, menambah pertanyaan, dan membetulkan nama karakter. Karena penambahan karakter dan pertanyaan telah dijelaskan pada bagian sebelumnya, maka pada bagian ini penulis akan memaparkan mengenai *game report*.

Pada opsi *game report*, terdapat daftar pertanyaan yang muncul pada permainan beserta jawaban yang dijawab oleh pemain. Kemudian, terdapat juga perkiraan jawaban yang seharusnya atau yang diharapkan oleh program. Statistik pertanyaan dan jawaban disajikan dalam bentuk tabel di bawah ini.

Character to find: Katy Perry

Question	Answer given	Answer expected
Is your character a girl?	Yes	Yes
Is your character from your family?	No	No
Is your character a star?	Yes	Yes
Does your character live in America?	Yes	Yes
Is your character a singer?	Yes	Yes
Is your character famous thanks to Disney Channel?	No	No
Is your character blond?	No	No
Has your character sung a hit single in the 90s?	No	No
Is your character dark-skinned?	No	No
Is your character an actor?	No	No
Does your character sleep in gas station?	Don't know	No
Does your character ever ask you to call her maybe?	No	No
Is your character Asian?	No	No
Does your character represent a country?	No	No
Does your character play in a rock band?	No	No
Is your character a teenager?	No	No
Is your character still alive?	Yes	Yes
Is your character a priest?	No	No
Did your character play in a Polish blues band?	No	No
Does your character wear a space suit?	No	No
Is your character black-haired?	Yes	Yes
Does your character play several music instruments?	Yes	NONE
Has your character ever dated a celebrity?	Yes	Yes
Has your character been pope?	Don't know	No
Is your character white?	Yes	Yes

Dari tabel di atas, kita dapat membayangkan 'jalur' yang harus ditempuh di dalam pohon untuk bisa sampai kepada daun dengan kata kunci 'Katy Perry'. Akar dari pohon tersebut adalah pertanyaan 'Is your character a girl?' yang merupakan pertanyaan dasar. Karena jawaban dari pemain adalah 'yes', maka kita hanya perlu mencari database karakter dengan jenis kelamin perempuan. Dengan ini, kita telah mengeliminasi banyaknya kemungkinan jawaban. Begitu seterusnya, hingga sampai kepada pertanyaan yang terakhir.

Ada banyak hal menarik yang bisa kita temukan dalam permainan ini. Misalnya, tidak hanya tokoh-tokoh yang terkenal saja, kita juga bisa memikirkan karakter apapun,

tokoh kartun, anime, orang yang kita kenal seperti keluarga kita, atau bahkan diri kita sendiri. Selain itu, meskipun terdapat beberapa jawaban dari pemain yang salah atau tidak sesuai dengan jawaban yang diharapkan, Akinator tetap dapat menebak karakter dengan tepat. Karena itu, algoritma yang digunakan pada Akinator merupakan algoritma dengan tingkat kerumitan yang tinggi namun cukup efisien.

IV. KESIMPULAN

Dari berbagai pemaparan pada bagian sebelumnya, didapat kesimpulan sebagai berikut:

- *Artificial Intelligence* merupakan perkembangan ilmu komputer yang membuat suatu program dapat bekerja dengan tingkat kecerdasan yang hampir menyerupai manusia. *Game* yang berbasis *Artificial Intelligence*, menghasilkan suatu ilusi kecerdasan pada *non-player character*. Salah satu contoh dari game berbasis *Artificial Intelligence* adalah 'Akinator the Genie', dengan tokoh jin Akinator sebagai *non-player character* yang memiliki 'kecerdasan'.
- Pohon merupakan struktur yang memiliki pengaruh paling besar pada cara kerja permainan ini. Terutama pohon keputusan dan pohon pencarian biner. Namun, karena struktur dan database yang dimiliki Akinator sangat besar dan kompleks, kedua aplikasi pohon tersebut hanya merupakan ide dasar untuk membuat *game* ini. Algoritma yang sesungguhnya dari *game* ini merupakan pengembangan yang lebih lanjut dari struktur pohon tersebut, namun masih memiliki prinsip dan ide dasar yang sama.
- Secara umum, Akinator dapat menebak karakter yang dipikirkan oleh pemain dengan menggunakan metode pencarian yang dapat mengeliminasi kemungkinan-kemungkinan pilihan karakter berdasarkan jawaban dari beberapa pertanyaan yang diajukan hingga akhirnya menyisakan satu solusi.

REFERENSI

- [1] George Luger & William Stubblefield. 2004. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving, 5th ed.* The Benjamin/Cummings Publishing Company.
- [2] Munir, Rinaldi. 2008. *Diktat Kuliah IF2091 – Struktur Diskrit.* Bandung: Program Studi Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung.
- [3] Rosen, Kenneth H. 2007. *Discrete Mathematics and It's Applications, Sixth Edition.* New York: McGraw-Hill.
- [4] Syamsuddin, Aries. 2012. *Algoritma Decision Tree C4.5.* Dari <http://blogs.itb.ac.id/aiceware/2012/09/23/algoritma-decision-tree-c4-5/>, 15 Desember 2012.
- [5] _____. *Everything You've Always Wanted to Know About Akinator.* Dari <http://en.akinator.com/content/6/everything-you-ve-always-wanted-to-know-about-akinator>, 15 Desember 2012.
- [6] _____. *How Do 20 Questions AI Algorithms Work?* Dari <http://stackoverflow.com/questions/887533/how-do-20-questions-ai-algorithms-work>, 15 Desember 2012.
- [7] _____. *How Does the Web Service Behind the Akinator Work?* Dari <http://www.quora.com/How-does-the-web-service-behind-the-Akinator-work>, 16 Desember 2012.

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, 16 Desember 2012

A handwritten signature in black ink, appearing to read 'Rifki Afina Putri', with a stylized flourish at the end.

Rifki Afina Putri (13511066)