

Penerapan Pohon Keputusan dalam Pemilihan Bahasa Pemrograman yang Tepat untuk Suatu Proyek

Aloysius Gilang Pramudya - 13520147¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

¹author@itb.ac.id

Abstrak—Dalam sebuah proyek pembangunan software memilih bahasa pemrograman yang paling cocok adalah suatu keharusan. Jika pemrogram gagal memilih bahasa pemrograman yang tepat maka akan berpengaruh buruk terhadap keberjalanan dan perkembangan proyek. Memilih bahasa pemrograman bisa menjadi tugas yang menantang dan memakan waktu karena banyaknya pilihan bahasa pemrograman yang tersedia. Pemrogram harus mempertimbangkan beberapa faktor saat memilih bahasa pemrograman. Paper ini membahas bagaimana penerapan pohon keputusan dapat membantu pemrogram dalam menentukan bahasa pemrograman yang tepat untuk suatu proyek.

Kata Kunci— Bahasa pemrograman, pohon keputusan, proyek, software.

I. PENDAHULUAN

Membangun proyek *software* merupakan proses yang sangat rumit. Setiap *software* mempunyai siklus hidupnya masing-masing. Semua proyek *software* akan melalui tahapan yang sama yaitu : konsepsi, pengumpulan persyaratan/eksplorasi/*modelling*, desain, *coding* dan *debugging*, *testing*, *release*, pemeliharaan, dan *retirement*^[1]. Konsepsi dan pengumpulan persyaratan merupakan tahap awal yang sangat penting dalam pembangunan *software*. Tahapan ini mencakup daftar spesifikasi program dan fitur apa saja yang harus ada ketika program dijalankan. Pemrogram harus bisa menjawab pertanyaan yang spesifik seperti : apa masalah yang bisa diselesaikan dengan fitur ini? bagaimana cara menggunakannya untuk pengguna? apa saja jenis data masukan dan luaran? apakah perlu untuk mengintegrasikannya dengan program lain? bagaimana sistem keamanan dan privasinya?. Pada tahap ini juga pemrogram harus memilih bahasa pemrograman untuk proyek tersebut. Jika pemrogram ingin membangun *software* yang bagus, pemrogram harus mempunyai pemahaman yang baik terhadap persyaratan program untuk bisa memilih bahasa pemrograman yang paling tepat.

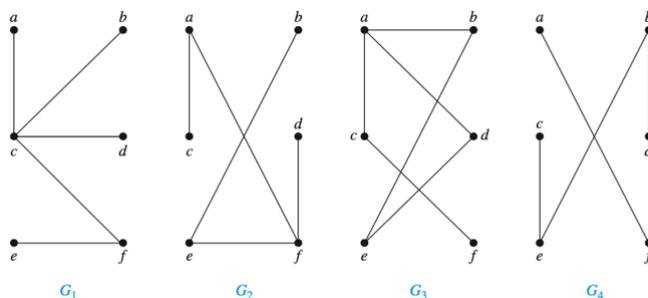
Memilih bahasa pemrograman dapat menjadi tantangan juga hambatan bagi pemrogram. Pemrogram dapat terjebak dalam *analysis paralysis* karena banyaknya pilihan bahasa yang tersedia sehingga memakan banyak waktu dan energi. Salah satu cara untuk membantu pemrogram adalah dengan menggunakan pohon keputusan. Dengan pohon keputusan, pemrogram dapat menentukan bahasa pemrograman berdasarkan persyaratan dengan cepat. Paper ini membahas penerapan pohon keputusan

dalam memilih bahasa pemrograman untuk suatu proyek.

II. DASAR TEORI

A. Pohon^[2]

Pohon adalah graf tak-berarah terhubung yang tidak mengandung sirkuit. Karena pohon tidak mempunyai sirkuit maka pohon tidak mempunyai sisi ganda atau *loop*.



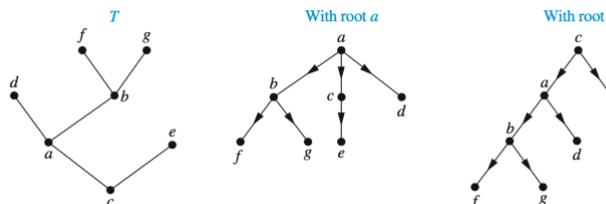
Gambar 1. Graf G1 dan G2 adalah pohon, Graf G3 dan G4 bukan pohon.

Sumber : Discrete mathematics and its applications, 8th ed, by Kenneth H. Rosen

Graf tak-berarah adalah pohon jika dan hanya jika ada lintasan unik diantara pasangan simpul-simpul. Beberapa graf yang tidak mengandung sirkuit yang tidak terhubung disebut hutan (*forest*). Setiap komponen terhubung dari hutan merupakan pohon.

B. Pohon berakar^[2]

Pohon berakar adalah pohon yang salah satu simpulnya ditetapkan sebagai akar dan semua sisinya diarahkan menjauh dari akarnya.

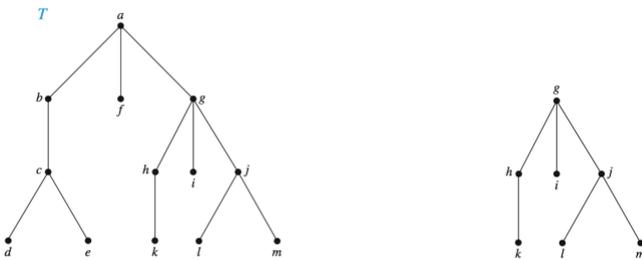


Gambar 2. Pohon dan pohon berakar dengan akar yang berbeda

Sumber : Discrete mathematics and its applications, 8th ed, by Kenneth H. Rosen

Pohon berakar dapat dibuat secara rekursif. Kita bisa mengubah pohon tak-berakar menjadi pohon berakar dengan memilih salah satu simpulnya menjadi akar. Pohon berakar biasa digambarkan dengan akarnya berada di posisi paling atas.

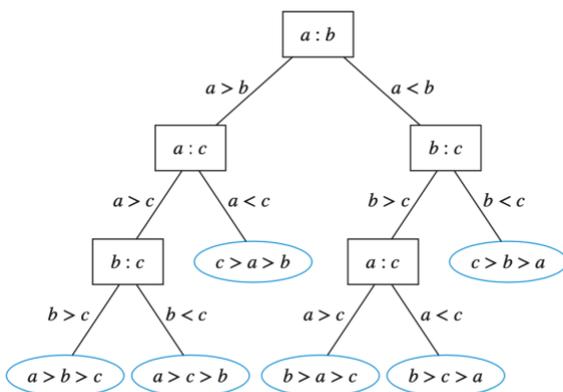
Terminologi untuk pohon berasal dari penamaan botani. Misalkan T adalah pohon berakar. Jika v adalah sebuah simpul selain simpul akar, Orang tua (*parent*) dari v adalah simpul unik u sedemikian sehingga ada sisi dari u ke v . Jika u adalah orang tua dari v , v adalah anak (*child*) dari u . Simpul dengan orang tua yang sama disebut saudara kandung (*sibling*). Leluhur (*ancestor*) dari suatu simpul selain akar adalah simpul-simpul dari akar menuju simpul tersebut, termasuk akar dan tidak termasuk simpul itu sendiri. Keturunan (*descendants*) dari simpul v adalah simpul-simpul yang memiliki leluhur v . Simpul pada pohon disebut daun (*leaf*) jika simpul tersebut tidak memiliki anak. Akar adalah simpul dalam kecuali jika akar merupakan simpul satu-satunya pada pohon. Jika a merupakan simpul pada pohon, upapohon (*subtree*) dengan a sebagai akar adalah upagraf (*subgraph*) dari pohon yang terdiri dari a , keturunannya, dan semua sisi yang bersisian dengan keturunan tersebut.



Gambar 3. Pohon T dan upapohon berakar di g
 Sumber : Discrete mathematics and its applications, 8th ed, by Kenneth H. Rosen

C. Pohon Keputusan^[2]

Pohon keputusan (*decision tree*) adalah pohon berakar dengan setiap simpul dalamnya berhubungan dengan suatu keputusan, dan upapohon dari setiap simpul menggambarkan setiap hasil dari keputusan tersebut.

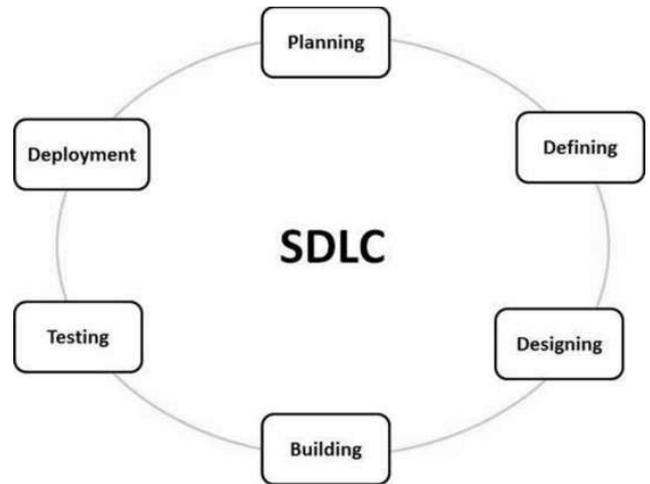


Gambar 4. Pohon keputusan untuk mengurutkan 3 elemen berbeda

Kompleksitas dari algoritma pengurutan pada pohon keputusan diukur dari berapa banyak perbandingan yang digunakan. Kebanyakan perbandingan menggunakan lintasan terpanjang pada pohon keputusan. Karena ketinggian dari suatu pohon biner dengan $n!$ daun minimal $\lceil \log_2 n! \rceil$, maka banyaknya perbandingan minimal $\lceil \log_2 n! \rceil$.

D. Software Development Life Cycle^[3]

Software Development Life Cycle (SDLC) adalah suatu proses untuk mendesain, membangun, mengetes *software*.



Gambar 5. Tahapan dalam SDLC
 Sumber : https://www.tutorialspoint.com/sdlc/sdlc_overview.htm

Tahap pertama yaitu perencanaan (*planning*) merupakan tahap paling fundamental di SDLC. Tim akan menerima masukan dari *customer*, masukan ini kemudian digunakan untuk merencanakan pendekatan proyek dan melakukan uji kelayakan produk di bidang ekonomi, operasional, dan teknis.

Tahap kedua dari SDLC adalah pengumpulan persyaratan (*defining requirement*). Tahap ini membahas dokumen yang berisikan semua persyaratan produk yang akan dirancang dan dikembangkan selama siklus hidup proyek.

Tahap ketiga dari SDLC adalah mendesain (*designing*). Tim akan mendesain semua arsitektur produk, aliran data, dan penggunaan alat dari pihak ketiga dengan jelas dan detail.

Tahap selanjutnya adalah membangun produk (*building*). Pada tahap inilah pemrogram akan membuat produk berdasarkan pedoman yang sudah dibuat seperti bahasa pemrograman, *compiler*, *debugger* yang digunakan.

Tahap kelima setelah program dibuat adalah *testing*. Semua cacat produk akan dilaporkan, diperbaiki, dan diuji ulang sampai produk mencapai standar kualitas yang ditetapkan.

Tahap terakhir dari SDLC adalah *deployment and maintenance*. Setelah produk telah lulus uji, maka produk akan diluncurkan di pasar yang sesuai. Pemrogram juga harus melakukan pemeliharaan secara berkala setelah produk diluncurkan.

E. Bahasa Pemrograman

Bahasa pemrograman merupakan suatu bahasa yang mengekspresikan suatu set instruksi untuk dipahami komputer. Instruksi tersebut kemudian dapat dieksekusi secara langsung ketika sudah dalam bentuk bahasa mesin. Secara umum bahasa pemrograman diklasifikasikan menjadi dua tipe yaitu *Low level language* dan *High level language*^[4]. Makalah ini hanya akan membahas pemilihan bahasa pemrograman bertipe *High level language*.

Ada banyak sekali bahasa pemrograman yang tersedia sampai saat ini. Setiap bahasa pemrograman mempunyai karakteristik masing-masing. Berikut adalah beberapa contoh bahasa pemrograman yang paling populer.

1. Python^{[5][6]}

Python adalah bahasa pemrograman berorientasi objek yang diluncurkan pada tahun 1992. Python terkenal luas sebagai bahasa pemrograman terbaik untuk pemula. Python banyak digunakan dibidang sains, teknik, dan akademik, seperti *artificial intelligence* dan *machine learning*. Kita juga bisa membangun website menggunakan web framework python yang terkenal yaitu Django.

Python juga digunakan membuat aplikasi terkenal seperti Youtube, Instagram, dan Spotify.



Gambar 6. Logo Python

Sumber :

https://commons.wikimedia.org/wiki/File:Python_logo_and_wordmark.svg

2. Java^{[5][6]}

Java adalah bahasa pemrograman berorientasi objek serbaguna yang sangat populer untuk semua perangkat dan OS karena portabilitasnya. Sesuai dengan slogannya "write once, work everywhere". Mirip dengan python, java juga menjadi bahasa pemrograman yang paling dicari dan digaji tinggi.

Java juga digunakan untuk membuat aplikasi dan game terkenal seperti Gmail, Minecraft, aplikasi-aplikasi android, dan software bisnis lainnya.



Gambar 7. Logo Java

Sumber : [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))

3. C^{[5][6]}

Bahasa C adalah salah satu bahasa paling tua yang masih banyak digunakan sampai sekarang. Bahasa C cukup sulit untuk dipelajari oleh pemula. Bahasa C pertama kali diluncurkan pada tahun 1972. bahasa C banyak berpengaruh dan menjadi inspirasi bahasa pemrograman lain seperti C, C#, Java, Python dan banyak lagi. Bahasa ini banyak digunakan untuk membangun sistem operasi seperti Windows dan Linux. Kita juga bisa

membangun gim, grafis dan aplikasi-aplikasi yang menggunakan banyak perhitungan.



Gambar 8. Logo C++

Sumber :

https://commons.wikimedia.org/wiki/File:C_Programming_Language.svg

4. C++^{[5][6]}

Bahasa C++ adalah versi yang lebih kompleks dari C dengan fitur yang lebih banyak. Bahasa ini banyak digunakan untuk membuat gim, sistem operasi, *browser*, dan berbagai aplikasi industri. Bahasa ini terkenal sulit untuk dipelajari bagi pemula.



Gambar 9. Logo C++

Sumber : https://commons.wikimedia.org/wiki/File:ISO_C%2B%2B_Logo.svg

5. JavaScript^{[5][6]}

Terlepas dari namanya, JavaScript tidak berhubungan dengan Java. JavaScript adalah bahasa berorientasi objek yang diluncurkan pada tahun 1995. JavaScript juga termasuk bahasa pemrograman yang mudah untuk dipelajari JavaScript digunakan untuk menulis kode yang berjalan pada web. JavaScript biasanya digunakan bersama dengan HTML dan CSS untuk membuat aplikasi web yang interaktif dan responsif. Bahasa ini menjadi bahasa yang wajib dikuasai bagi *front-end developer*.



Gambar 10. Logo JS

Sumber :

https://commons.wikimedia.org/wiki/File:Unofficial_JavaScript_logo_2.svg

6. PHP^[6]

PHP adalah bahasa pemrograman berorientasi objek yang gratis dan bersifat *open source*. PHP juga sering digunakan bersama dengan HTML dan CSS dalam membangun web. Bedanya dengan JavaScript adalah PHP merupakan *server-side scripting language* dibanding *client-side*.



Gambar 11. Logo PHP

Sumber : <https://commons.wikimedia.org/wiki/File:PHP-logo.svg>

7. SQL^[6]

Structured Query Language (SQL) adalah bahasa yang didesain spesifik untuk mengelola data yang disimpan dalam *database*. Tidak seperti kebanyakan bahasa pemrograman lain, bahasa SQL bukanlah bahasa yang serbaguna dan penggunaannya lebih sempit.



Gambar 12. Logo SQL

Sumber :

https://commons.wikimedia.org/wiki/File:Sql_data_base_with_logo.png

III. PEMILIHAN BAHASA PEMROGRAMAN DENGAN POHON KEPUTUSAN

A. Faktor yang Perlu Diperhatikan

Dalam memilih bahasa pemrograman yang tepat, pemrogram perlu memiliki pemahaman yang baik tentang persyaratan yang diberikan oleh *client*. Beberapa faktor yang perlu diperhatikan pemrogram saat memilih bahasa pemrograman adalah :

1. Apa bentuk/tipe aplikasi yang akan dibuat ?

Pemrogram perlu memiliki tujuan yang jelas terlebih dahulu sebelum memilih bahasa pemrograman. Pemilihan bahasa pemrograman sangat berpengaruh kepada tipe aplikasi yang akan dibuat. Aplikasi dapat berbentuk aplikasi web, aplikasi *mobile*, sistem operasi, embedded firmware, *enterprise*, dan sebagainya.

2. Apa platform yang akan menjalankan aplikasi tersebut ?

Faktor lain yang perlu diperhatikan adalah jenis *platform* yang akan menjalankan program. Pemilihan bahasa pemrograman yang tepat untuk *platform* tertentu akan berpengaruh kepada efisiensi program.

Misalkan jika kita menulis program dengan bahasa C untuk platform Windows dan Linux, maka kita memerlukan *platform compiler* dan dua *executables* yang berbeda. Sedangkan jika kita menulis program dengan bahasa Java kita bisa menjalankan program tersebut dimanapun selama terinstal Java Virtual Machine pada *platform* tersebut.

Hal yang sama dengan aplikasi *website* jika kita menggunakan HTML5 dan CSS3. Tampilan dan perilaku *website* yang dibuat mungkin akan berbeda pada *browser* yang berbeda.

3. Seberapa besar skala dan performa aplikasi tersebut?

Performa dari aplikasi yang dibuat sangat bergantung pada bahasa pemrograman yang dipakai. pemrogram perlu memperkirakan berapa banyak orang yang akan memakai aplikasi tersebut? apakah aplikasi ini sudah bisa menampungnya? apakah bertambahnya user akan memperlambat performa aplikasi?. Beberapa stack dengan skalabilitas yang baik adalah Ruby on Rails, .NET, MEAN (MongoDB, Express, Angular, Node), LAMP (Linux, Apache,

MySQL, PHP/Perl/Python).

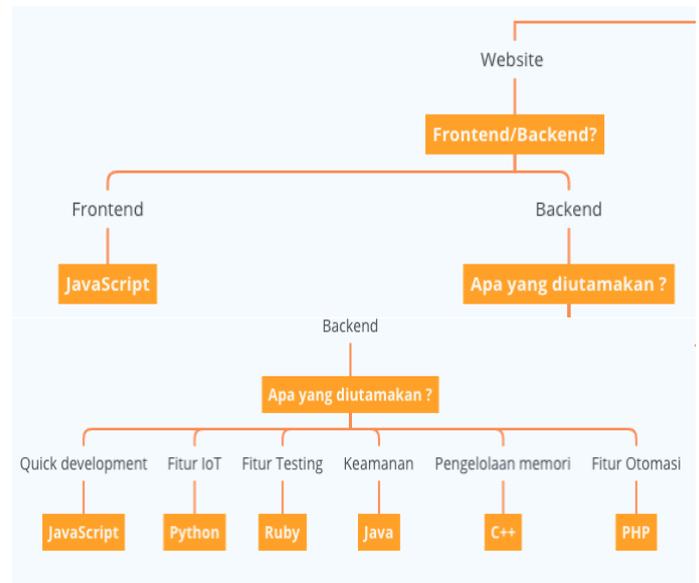
B. Pohon keputusan

Perbandingan pertama dan yang terpenting yaitu menentukan tipe aplikasi yang akan dibuat.



Gambar 13. Perbandingan tipe aplikasi

Setelah menentukan tipe aplikasi yang akan dibuat, pemrogram melanjutkan perbandingan selanjutnya sesuai tipe aplikasi. Berikut ini adalah upapohon keputusan untuk aplikasi website. Penilaian diambil berdasarkan perbandingan bahasa pemrograman pada website <https://blog.back4app.com/best-backend-language> ^[7].



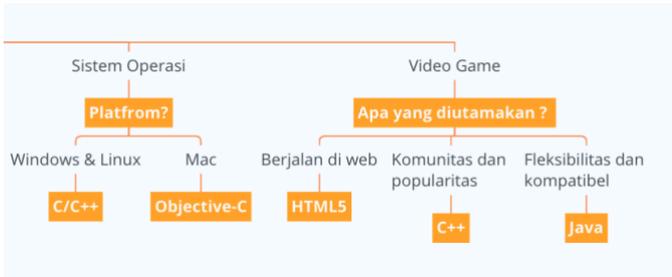
Gambar 13. Upapohon keputusan untuk aplikasi website

Berikut ini adalah upapohon keputusan untuk aplikasi mobile



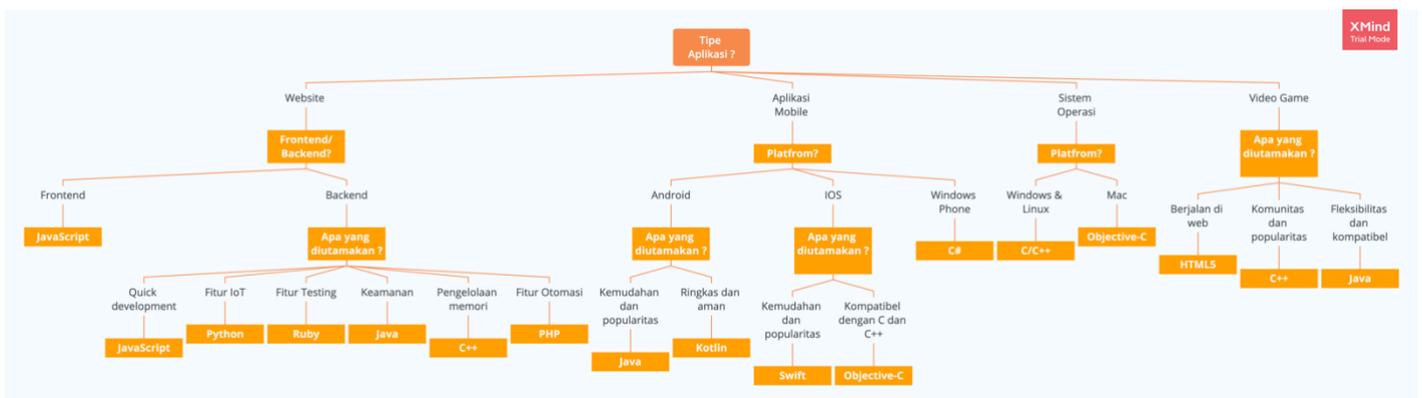
Gambar 14. Upapohon keputusan untuk aplikasi mobile

Berikut ini adalah upapohon keputusan untuk aplikasi sistem operasi dan video game.



Gambar 15. Upapohon keputusan untuk sistem operasi dan video game

Maka keseluruhan pohon keputusan bisa digambarkan sebagai berikut.



Gambar 15. Keseluruhan pohon keputusan pemilihan bahasa pemrograman

C. Contoh Kasus

Berikut ini adalah beberapa contoh penerapan pohon keputusan dalam pemilihan bahasa pemrograman.

1. Kasus 1

Fikri adalah seorang mahasiswa Informatika ITB. Saat ini dia berserta kelompoknya ditugaskan untuk membuat website. Waktu yang diberikan untuk menyelesaikan website tersebut hanya 5 hari. Fikri mendapat tugas untuk menangani bagian *back-end* website.

Fikri menggunakan pohon keputusan yang sudah dibuat untuk membantunya memilih bahasa pemrograman yang tepat. Pada level pertama Fikri harus menentukan tipe aplikasi yang dia buat. Selanjutnya Fikri harus memilih antara *front-end* atau *back-end*. Pada level terakhir, karena Fikri memerlukan bahasa pemrograman dengan pengembangan yang cepat, maka Fikri akan memilih bahasa JavaScript. Dengan menggunakan bahasa yang sama pada bagian *front-end* dan *back-end*, pembuatan website pun akan lebih cepat.

2. Kasus 2

Yoseph adalah seorang mahasiswa Informatika ITB. Saat ini dia berserta kelompoknya ditugaskan untuk membuat aplikasi *mobile* android. Karena Yoseph belum pernah membuat aplikasi *mobile* sebelumnya, dia lebih mempertimbangkan bahasa pemrograman yang populer digunakan untuk membuat aplikasi *mobile* dan memiliki komunitas yang besar.

Yoseph menggunakan pohon keputusan yang sudah dibuat untuk membantunya memilih bahasa pemrograman yang tepat. Pada level pertama Yoseph harus menentukan tipe aplikasi yang dia buat yaitu aplikasi *mobile*. Selanjutnya Yoseph harus memilih pada platform apa aplikasi tersebut akan berjalan. Pada level terakhir, karena Yoseph lebih memilih bahasa pemrograman yang populer dan memiliki komunitas yang besar, maka Yoseph akan memilih bahasa Java. Popularitas dan komunitas yang besar akan membantu Yoseph ketika ia menghadapi kesulitan dalam membuat aplikasi tersebut.

3. Kasus 3

Ranjabi adalah seorang mahasiswa Informatika ITB. Saat ini dia berserta kelompoknya ditugaskan untuk membuat website. Ranjabi berada di kelompok yang sama dengan Fikri pada kasus yang pertama. Dia ingin menambahkan *game* kecil pada website yang dibuat.

Ranjabi menggunakan pohon keputusan yang sudah dibuat untuk membantunya memilih bahasa pemrograman yang tepat. Pada level pertama Ranjabi harus menentukan tipe aplikasi yang dia buat yaitu *game*. Pada level terakhir, karena Ranjabi memerlukan bahasa pemrograman untuk membuat *game* yang bisa berjalan pada website, maka Ranjabi akan memilih bahasa HTML5.

IV. SIMPULAN

Pemilihan bahasa pemrograman untuk suatu proyek bisa menjadi tugas yang menantang dan memakan waktu bagi pemrogram. Salah satu alat yang bisa membantu pemrogram adalah dengan menggunakan pohon keputusan. Melalui penerapan pohon keputusan, pemrogram diharapkan untuk mampu memilih bahasa pemrograman dengan tepat dan cepat sehingga tidak memakan waktu dan menghambat proses-proses selanjutnya.

Ada banyak sekali bahasa pemrograman yang tersedia sampai saat ini yang tidak dibahas pada makalah ini. Model pohon keputusan yang telah dibuat bisa dikembangkan lebih jauh dengan bahasa dan faktor pemilihan yang lebih beragam.

V. UCAPAN TERIMAKASIH

Penulis mengucapkan syukur kepada Tuhan Yang Maha Esa atas segala berkat dan rahmat-Nya sehingga penulis dapat menyelesaikan makalah ini. Selain itu penulis juga mengucapkan terimakasih kepada :

1. Dr. Ir. Rinaldi Munir, MT., Dr. Nur Ulfa Maulidevi, S.T, M.Sc., Dra. Harlili M.Sc., selaku dosen pengajar mata kuliah Matematika Diskrit.
2. Keluarga dan teman-teman penulis yang selalu mendukung penulis.

REFERENSI

- [1] F. Dooley, Software development, design and coding: With patterns, debugging, unit testing, and refactoring. New York: APRESS, 2018,pp. 25-27.
- [2] K. H. Rosen, Discrete mathematics and its application, 8th ed. London: McGraw-Hill, 1995.
- [3] M. Bennatan, Software project management: A practitioner's approach, 2nd ed. London, England: McGraw-Hill Publishing, 1994.
- [4] M. Gabbrielli and S. Martini, Programming languages: Principles and paradigms, 2010th ed. Guildford, England: Springer, 2010..
- [5] C. Cheo, "Which programming language should I learn first? [infographic]," Carlcheo.com. [Online]. Available: <http://carlcheo.com/startcoding>. [Diakses: 10-Des-2021].
- [6] D. Hemmendinger, "computer programming language," Encyclopedia Britannica. 29-Jan-2021..
- [7] "The best ten backend languages," Back4app.com, 27-Feb-2021. [Online]. Available: <https://blog.back4app.com/best-backend-language/>. [Diakses: 11-Des-2021].

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.

Kuningan, 11 Desember 2020



Aloysius Gilang Pramudya