

Penerapan Pohon Keputusan pada Pemilihan Bahasa Pemrograman untuk Pemula

Kevin Basuki / 13516071

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

13516071@stei.itb.ac.id

Abstract—Seseorang yang baru memulai belajar pemrograman dapat mengalami kesulitan yang diakibatkan karena pendekatan paradigma ataupun pemilihan bahasa pemrograman yang tidak tepat untuknya. Memilih bahasa pemrograman untuk seorang pemula dapat memerhatikan berbagai pertimbangan yang bersesuaian dengan kondisi orang tersebut. Pertimbangan tersebut dapat berupa usia, motivasi, tingkat kesulitan, dan bidang yang ingin didalami. Pertimbangan – pertimbangan yang ada dapat dimodelkan dengan Pohon Keputusan sehingga solusi dari persoalan dapat diraih yaitu bahasa pemrograman yang sekiranya tepat dipelajari olehnya. Pengembangan pemodelan sangat dimungkinkan apabila terdapat kondisi – kondisi lain yang berpengaruh terhadap solusi ataupun solusi – solusi lain berupa bahasa pemrograman yang sekiranya tepat untuk kondisi tertentu.

Keywords—bahasa pemrograman, pemula, pertimbangan, pohon keputusan.

I. PENDAHULUAN

Zaman sekarang ini, teknologi berkembang sangat pesat terutama di bidang teknologi informasi. Muncul berbagai software yang sangat membantu di berbagai bidang, halaman web yang berisi informasi – informasi yang berguna dan menarik, bahkan koran, taksi, ktp, kartu tol dan contoh – contoh lainnya sudah menggunakan teknologi agar waktu yang diperlukan menjadi lebih ringkas dan tenaga yang dikeluarkan menjadi lebih sedikit. Dikarenakan perkembangan yang sangat pesat tersebut, para pelajar baik di perguruan tinggi maupun tingkat pendidikan yang lebih rendah disarankan untuk belajar pemrograman walaupun mungkin bidangnya tidak fokus mendalami dunia informatika. Contohnya di Institut Teknologi Bandung ini, hampir semua mahasiswa tahun pertama dengan jurusan dan fakultas yang berbeda – beda diwajibkan mengambil mata kuliah Pengantar Teknologi Informasi yang sebagian materinya belajar tentang pemrograman dengan bahasa pemrograman yang berbeda – beda tergantung kebutuhan jurusan masing – masing. Di beberapa sekolah menengah juga sudah memberi murid – muridnya sedikit pengantar pemrograman dengan bahasa pemrograman yang lebih mudah dimengerti untuk anak – anak.

Memulai untuk belajar hal yang baru adalah tindakan yang menantang, menarik untuk dilakukan, tetapi terkadang juga sulit untuk dilakukan jika memang orang yang belajar tidak menyukai hal baru tersebut atau bisa jadi pendekatan

pembelajarannya tidak tepat. Dalam hal belajar pemrograman, pelajar terkadang sulit memahami paradigma yang ada dalam bahasa tersebut ataupun sintaks bahasa yang digunakan, hal ini bisa disebabkan karena salah memilih paradigma dan bahasa pemrograman untuk dipelajari pertama kali.

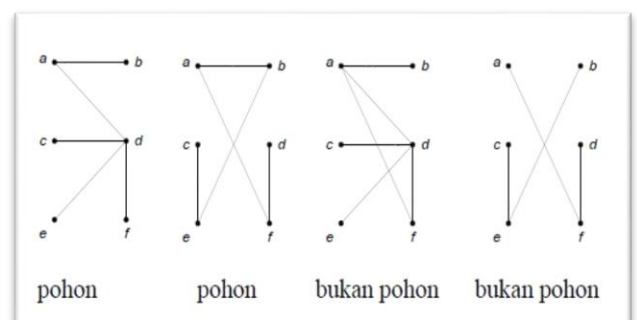
Tentu setiap orang yang ingin memulai belajar pemrograman mempunyai kondisi yang berbeda satu sama lain: ada yang baru berumur 13 tahun, ada yang ingin belajar karena tertarik dengan game, ada yang tertarik dengan web, ada yang tertarik dengan startup, ada yang tertarik dengan prospek kerja dan gaji yang ditawarkan, ada juga yang hanya iseng ingin belajar pemrograman. Semua kondisi tersebut tidak tepat didekati dengan pendekatan paradigma dan bahasa pemrograman yang sama, akan lebih tepat jika mereka memulai pemrograman dengan bahasa pemrograman yang berbeda. Untuk memilih bahasa pemrograman yang sekiranya tepat untuk dipelajari pertama kali, dapat digunakan Pohon Keputusan sebagai salah satu metode untuk memilih.

II. DASAR TEORI

2.1 Pohon

A. Definisi Pohon

Pohon adalah graf tak-berarah terhubung yang tidak mengandung sirkuit.^[1] Penekanannya, ada 2 ciri khas dari pohon yaitu terhubung dan tidak mengandung sirkuit.



Gambar 1. Graf 1 dan Graf 2 adalah pohon, Graf 3 dan Graf 4 bukan pohon

(Sumber:

<https://imeldaflorensia91.wordpress.com/2013/05/08/pohon-tree-matematika-diskrit/>

Diakses 3 Desember 2017 pukul 17.00 WIB)

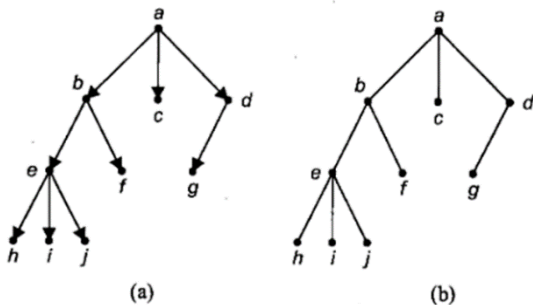
Pada Gambar 1, Graf 3 bukanlah pohon karena mengandung sirkuit dan Graf 4 bukan pohon karena tidak terhubung.

Sifat – sifat pohon dinyatakan dengan sebuah teorema:^[1] Misal $G = (V, E)$ adalah graf tak berarah sederhana dan jumlah simpulnya adalah n , maka semua pernyataan di bawah ini ekuivalen:

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 (jembatan adalah sisi yang bila dihapus menyebabkan sisi terpecah menjadi dua komponen).

B. Pohon Berakar^[1]

Pohon berakar (*rooted tree*) adalah pohon yang sebuah simpulnya diperlakukan sebagai akar dan sisi – sisinya diberi arah menjauh dari akar. Akar mempunyai derajat masuk nol dan simpul – simpul lainnya berderajat masuk satu. Simpul yang mempunyai derajat keluar nol disebut daun atau simpul terminal. Simpul yang mempunyai derajat keluar tidak nol disebut simpul dalam atau simpul cabang. Setiap simpul di pohon dapat dicapai dari akar dengan sebuah lintasan tunggal dan unik.



(a) Pohon berakar, (b) sebagai konvensi, arah panah pada sisi dapat diabaikan

Gambar 2. Pohon berakar dengan simpul a adalah akarnya

(Sumber: <http://poetra70.blogspot.co.id/2015/09/pohon-matematika-diskrit.html>)

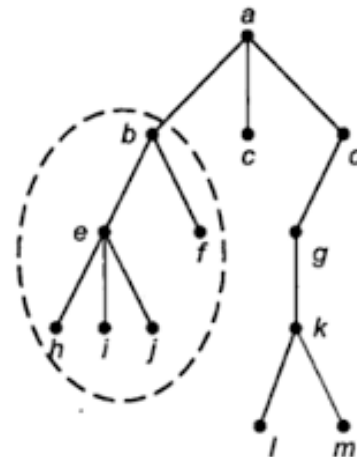
Diakses 3 Desember 2017 pukul 17.30 WIB)

Sebagai konvensi, arah sisi di dalam pohon tidak perlu digambar. Karena setiap simpul di pohon harus dicapai dari akar, maka lintasan di dalam pohon berakar selalu dari bagian “atas” ke “bawah”. Sembarang pohon tak – berakar dapat diubah menjadi pohon berakar dengan memilih sebuah simpul sebagai akar dari pohon tersebut. Pemilihan simpul yang berbeda sebagai akar menghasilkan pohon berakar yang berbeda pula.

C. Terminologi pada Pohon Berakar

Di bawah ini adalah beberapa terminologi yang penting pada pohon berakar:^[1]

1. Anak (*child*) dan Orang tua (*parent*)
Simpul y dikatakan anak simpul x jika ada sisi dari simpul x ke y . Dalam hal tersebut, x disebut orang tua dari y . Pada Gambar 2, b, c, dan d adalah anak – anak simpul a, dan a adalah orang tua dari b, c, dan d.
2. Lintasan (*path*)
Dari Gambar 2, lintasan dari a ke j adalah a, b, e, j. Panjang lintasan adalah jumlah sisi yang dilalui dalam suatu lintasan. Panjang lintasan dari a ke j adalah 3.
3. Keturunan (*descendant*) dan Leluhur (*ancestor*)
Jika terdapat lintasan dari simpul x ke simpul y di dalam pohon, maka x adalah leluhur dari simpul y , dan y adalah keturunan simpul x . Pada Gambar 2, b adalah leluhur simpul h, dan dengan demikian h adalah keturunan b.
4. Saudara kandung (*sibling*)
Simpul yang mempunyai orang tua yang sama adalah saudara kandung satu sama lain. Pada Gambar 2, f adalah saudara kandung e, tetapi g bukan saudara kandung e karena orang tua g dan e berbeda.
5. Upapohon (*subtree*)
Misalkan x adalah simpul di dalam pohon T , yang dimaksud dengan upapohon dengan x sebagai akarnya ialah upagraf $T' = (V', E')$ sedemikian sehingga V' mengandung x dan semua keturunannya dan E' mengandung sisi – sisi dalam semua lintasan yang berasal dari x .



Gambar 3. Upapohon

(Sumber:

<http://poetra70.blogspot.co.id/2015/09/pohon-matematika-diskrit.html>)

Diakses 3 Desember 2017 pukul 17.30 WIB)

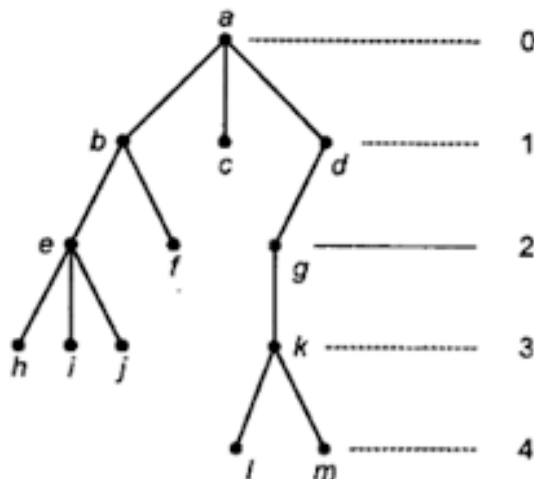
Pada Gambar 3, bagian yang dilingkari adalah upapohon dengan b sebagai akarnya.

6. Derajat (*degree*)
Derajat sebuah simpul pada pohon berakar adalah

jumlah upapohon (atau jumlah anak) pada simpul tersebut. Pada Gambar 2, derajat a adalah 3, derajat b adalah 2, derajat d adalah 1, dan derajat c adalah 0.

Derajat maksimum dari semua simpul adalah derajat pohon itu sendiri. Pohon pada Gambar 2 berderajat 3 karena derajat tertinggi dari semua simpulnya adalah 3.

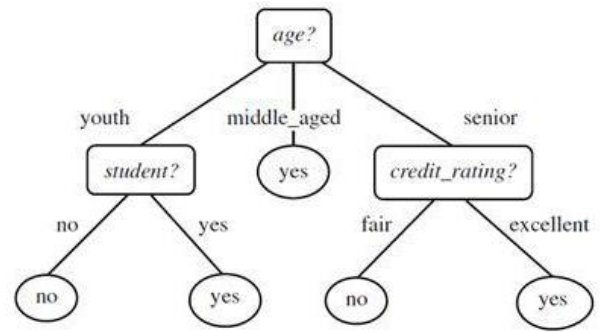
7. Daun (*leaf*)
Daun adalah simpul yang berderajat nol atau simpul yang tidak mempunyai anak. Pada Gambar 2, simpul h, i, j, f, c, dan g adalah daun.
8. Simpul dalam (*internal nodes*)
Simpul dalam adalah simpul yang mempunyai anak. Pada Gambar 2, simpul b, d, dan e adalah simpul dalam.
9. Aras (*level*) atau Tingkat
Akar mempunyai aras = 0, sedangkan aras simpul lainnya = 1 + panjang lintasan dari akar ke simpul tersebut. Beberapa buku memulai nomor aras dari 0, sedangkan buku lainnya dari 1.



Gambar 4. Aras tiap simpul suatu pohon
(Sumber: <http://poetra70.blogspot.co.id/2015/09/pohon-matematika-diskrit.html>
Diakses 3 Desember 2017 pukul 17.30 WIB)

10. Tinggi (*height*) atau Kedalaman (*depth*)
Aras maksimum dari suatu pohon disebut tinggi atau kedalaman pohon tersebut. Dengan kata lain, tinggi pohon adalah panjang maksimum lintasan dari akar ke daun. Pohon pada Gambar 4 mempunyai tinggi atau kedalaman 4.

- D. Pohon Keputusan^[1]
Pohon keputusan digunakan untuk memodelkan persoalan yang terdiri dari serangkaian keputusan yang mengarah ke solusi. Tiap simpul dalam menyatakan keputusan sedangkan daun menyatakan solusi.



Gambar 5. Contoh pohon keputusan untuk menunjukkan apakah seorang pembeli akan membeli sebuah komputer

(Sumber: <https://wargabasdat2009.wordpress.com/2009/06/10/dat-a-mining-firdi/>
Diakses 3 Desember 2017 pukul 20.25 WIB)

2.2 Bahasa Pemrograman

Ada banyak sekali bahasa pemrograman, mulai dari bahasa tingkat rendah, tingkat menengah, maupun bahasa tingkat tinggi. Bahasa tingkat rendah lebih dekat ke bahasa mesin sedangkan bahasa tingkat tinggi lebih dekat ke bahasa manusia sehari – hari. Pada makalah ini, hanya akan dibahas beberapa bahasa tingkat tinggi yang populer saja, karena tidak mungkin membahas semua bahasa pemrograman walaupun hanya secara umum dan bahasa yang dibahas di makalah ini hanya akan dipertimbangkan oleh para pemula untuk memulai belajar pemrograman.

- A. Python^{[2][3]}
 - Bahasa tingkat tinggi yang dikenal banyak orang sebagai bahasa pemrograman yang paling mudah untuk dipelajari dan paling tepat untuk pemula.
 - Meskipun mudah dipelajari, aplikasinya juga terdapat di banyak bidang seperti kecerdasan buatan, komputasi ilmiah, dan web programming.
 - Digunakan untuk membuat aplikasi terkenal seperti Youtube, Instagram, Spotify.
 - Digunakan oleh beberapa organisasi besar seperti Google, NASA, CIA, dan Disney.



Gambar 6. Logo Python
(Sumber: <https://www.python.org/community/logos/>
Diakses 3 Desember 2017 pukul 21.00 WIB)

- B. Java^{[2][3]}
 - Bahasa tingkat tinggi modern yang didesain oleh Sun Microsystems sekitar tahun 1990 dan sekarang

dimiliki oleh Oracle.

- Agak susah untuk dipelajari karena paradigma berbasis objek.
- Salah satu bahasa yang memberikan peluang kerja yang luas dan gaji yang tinggi untuk para programmer yang menguasai bahasa tersebut.
- Bahasa yang portabel, dinamis, dan populer di semua platform, sistem operasi, dan perangkat karena portabel.
- Slogan: *Write once, Run Anywhere.*
- Lebih dari 3 juta perangkat dapat menjalankan Java.
- Digunakan untuk membuat sebagian besar aplikasi Android, Gmail, Minecraft, antivirus, aplikasi web, dan aplikasi – aplikasi perusahaan seperti bank dan perusahaan lainnya.



Gambar 7. Logo Java

(Sumber:

<https://java-runtime-environment.en.softonic.com/>
Diakses 3 Desember 2017 pukul 21.17 WIB)

C. C^[2]

- Salah satu bahasa yang paling tua dan paling banyak digunakan di dunia.
- Agak susah untuk dipelajari karena salah satu bahasa yang tua dan terkesan “manual”.
- Bahasa yang populer digunakan untuk sistem dan perangkat keras.

D. C++^{[2][3]}

- Versi kompleks dari bahasa C dengan banyak fitur tambahan.
- Lebih susah untuk dipelajari karena banyaknya fitur dan ditambahkan paradigma berbasis objek.
- Digunakan untuk berbagai tujuan. Banyak digunakan untuk membuat *video game* dan berbagai jenis aplikasi.
- Sangat disarankan memiliki mentor yang dapat mendampingi jika ingin belajar bahasa ini.

E. JavaScript^{[2][3]}

- Meskipun namanya mirip dengan bahasa Java, bahasa ini memiliki fungsi yang berbeda.
- Bahasa yang populer digunakan untuk membuat sebagian besar *front-end* halaman web dan aplikasi karena membuat halaman web dan aplikasi lebih interaktif.
- Sangat disarankan untuk dipelajari jika ingin

menjadi pengembang *front-end* web (Ditambah HTML dan CSS).

F. C#^[2]

- Mirip dengan bahasa Java dalam sintaks dan beberapa fitur.
- Populer digunakan untuk membuat halaman web oleh perusahaan dan aplikasi Windows menggunakan kerangka .NET.
- Dapat dipelajari jika ingin mendalami *back-end*.

III. PENERAPAN POHON KEPUTUSAN PADA PEMILIHAN BAHASA PEMROGRAMAN

Dalam memilih bahasa pemrograman untuk dipelajari oleh pemula, perlu diperhatikan beberapa pertimbangan agar tidak terlalu mudah atau terlalu sulit atau tidak sesuai dengan tujuan seseorang belajar pemrograman. Beberapa pertimbangan yang harus diperhatikan untuk memilih bahasa pemrograman:

1. Usia

Usia memegang peran yang cukup penting untuk memilih bahasa pemrograman. Jika yang ingin belajar adalah anak – anak, jangan memilih bahasa – bahasa yang sukar untuk dipelajari. Anak – anak dapat memulai dengan belajar Python (mudah dipelajari) ataupun didahului dengan Scratch (tidak dibahas).

2. Alasan / Motivasi belajar pemrograman

Alasan seseorang belajar pemrograman sangat mempengaruhi bahasa yang akan digunakan. Seseorang yang belajar untuk kesenangan, belajar untuk menjadi Sarjana Teknik Informatika, dan belajar untuk mengembangkan bidang lain yang disukainya ataupun mencari pekerjaan akan membutuhkan bahasa pemrograman yang berbeda agar tepat guna dan tepat sasaran sesuai dengan alasan seseorang belajar pemrograman.

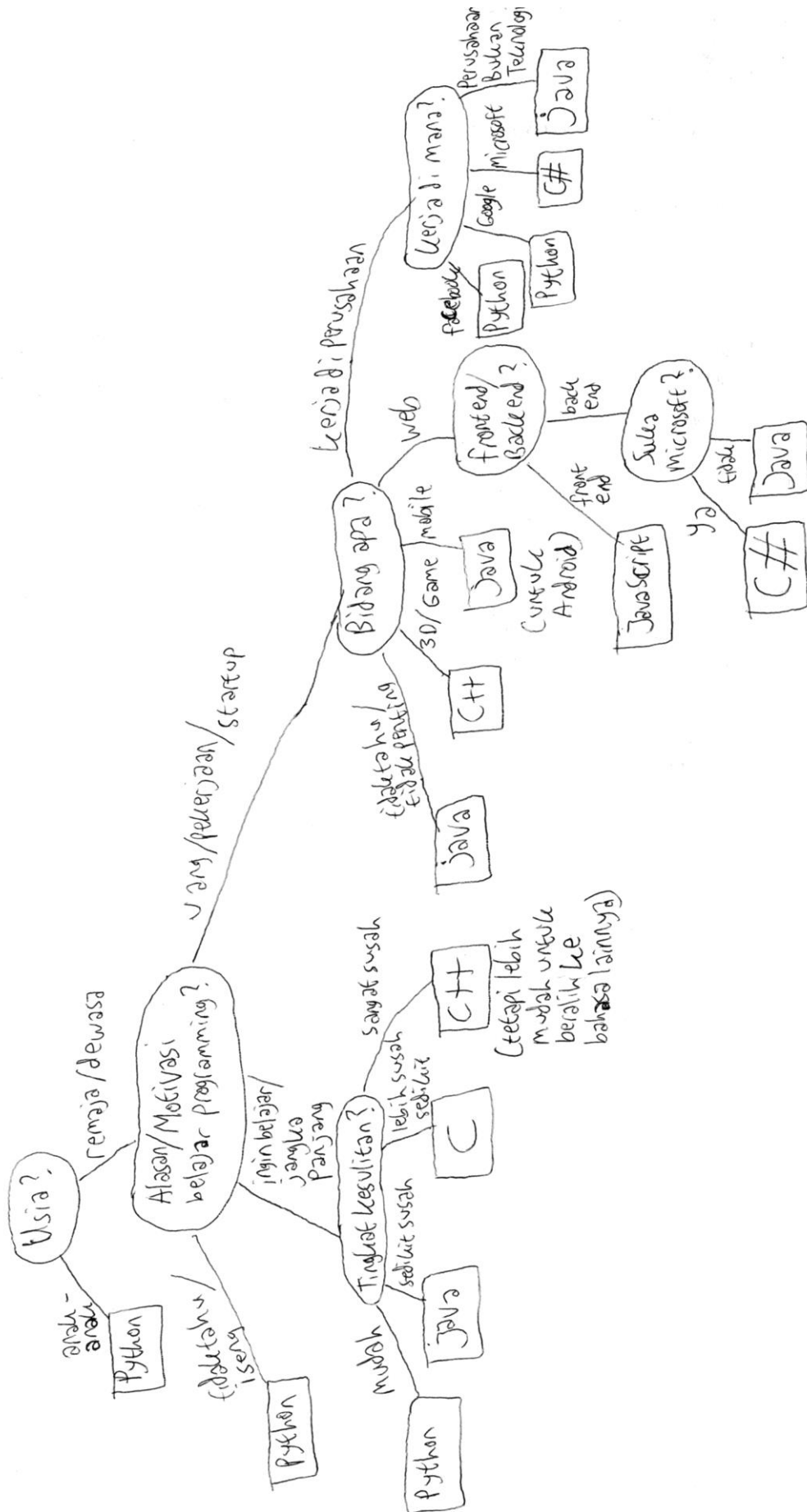
3. Tingkat kesulitan yang dikehendaki

Bahasa pemrograman yang berbeda memiliki tingkat kesulitan yang berbeda pula karena paradigma yang berbeda, tujuan penggunaan yang berbeda, maupun jumlah fitur yang berbeda. Seseorang hendaknya memilih tingkat kesulitan bahasa yang ingin dipelajarinya agar tidak terlalu mudah sehingga menyepelkan ataupun terlalu sulit sehingga menjadi beban.

4. Bidang yang ingin didalami

Setiap bahasa mempunyai tujuan penggunaan masing – masing, atau dapat dikatakan lebih kuat dalam bidangnya masing – masing. Bahasa yang tepat digunakan untuk membuat *video game* mungkin tidak cocok untuk membuat halaman web. Seseorang hendaknya menjadikan bidang yang ingin didalaminya sebagai salah satu pertimbangan agar bahasa yang dipilih tepat untuk bidang tersebut.

Berikut adalah contoh pohon keputusan yang dapat dibentuk berdasarkan pertimbangan – pertimbangan di atas:



Gambar 8. Contoh Pohon Keputusan untuk memilih bahasa pemrograman

IV. KESIMPULAN

Pohon Keputusan mempunyai banyak kegunaan, salah satunya dapat diaplikasikan dalam pemilihan bahasa pemrograman yang tepat untuk pemula. Pemodelan dalam bentuk Pohon Keputusan ini diharapkan dapat membantu menyelesaikan persoalan dalam memilih bahasa pemrograman. Pemodelan ini masih dapat dikembangkan dengan berbagai pertimbangan dan kondisi yang dihadapi seseorang sehingga solusi yang ditawarkan menjadi lebih sah. Selain itu, terdapat banyak sekali bahasa pemrograman yang tidak dibahas di makalah ini, tentunya dapat menjadi alternatif solusi jika pemodelan Pohon Keputusan ini ingin dikembangkan.

REFERENSI

- [1] M. Rinaldi, "Matematika Diskrit", 3rd ed. Bandung: Informatika Bandung, 2010, ch. 9.
- [2] <http://carlcheo.com/startcoding>. Diakses 2 Desember 2017 pukul 22.00 WIB.
- [3] <https://play.google.com/store/apps/details?id=com.sololearn>. Diakses 2 Desember 2017 pukul 23.15 WIB.

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, 3 Desember 2017



Kevin Basuki / 13516071