

Penerapan Graf dan Pohon pada Klasifikasi Aplikasi di Play Store

Amal Qurany 13514078
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
quranyamal@students.itb.ac.id

Abstract—Google Play Store merupakan pusat distribusi aplikasi untuk perangkat berbasis Android. Berdasarkan statistik yang dirilis oleh appfigures.com, total aplikasi yang terdapat di Google Play Store pada tahun 2014 lalu adalah sebanyak 1,43 juta aplikasi. Di dalam makalah ini akan dibahas terapan dari teori pohon atau *tree* dan teori graf untuk membantu menyelesaikan persoalan pengelompokan aplikasi pada Google Play Store sehingga pengguna android dapat melakukan pencarian aplikasi dengan mudah serta memperoleh hasil pencarian yang relevan.

Keywords—Pohon, Graf, Play Store, Android, Google

I. PENDAHULUAN

Android merupakan Sistem Operasi *mobile* yang tidak asing lagi di kalangan masyarakat saat ini. Android dikembangkan oleh raksasa perusahaan IT, Google, dengan mengadopsi kernel Linux. Sejak android ditemukan oleh Andy Rubin pada tahun 2003, sistem operasi ini berkembang sangat pesat. Berdasarkan statistik yang dirilis oleh idc.com, pada kuartal ke-2 tahun 2015 Android digunakan oleh 82,8% dari total seluruh smartphone yang ada di dunia ^[1].

Peningkatan pengguna Android ini tentu tidak terlepas dari faktor keunggulan yang dimilikinya. Salah satu daya Tarik yang diberikan sistem operasi milik Google ini dibanding pesaingnya adalah keterbukaan kode sumber atau yang biasa dikenal dengan istilah *Open Source*.

Dengan sistem open source, para pengembang atau *developer* bahkan pengguna biasa sekalipun dapat dengan mudah mempelajari dan membuat aplikasi berbasis Android. Oleh karena itu jumlah aplikasi yang dibuat pun meningkat pesat.

Google menyediakan kakas khusus tempat mengunduh segala aplikasi berbasis Android bernama Google Play Store. Karena banyaknya aplikasi yang diunggah oleh para pengembang ke Play Store ini, timbul berbagai persoalan diantaranya; Bagaimana caranya agar pengguna Android dapat menemukan aplikasi yang diinginkan diantara lebih dari satu juta aplikasi yang ada di Play Store? Bagaimana jika pengguna tersebut tidak tahu nama

dari aplikasi yang diinginkannya tetapi dia ingin mencari aplikasi dengan fungsi tertentu? Di sisi pengembang, bagaimana agar pengembang bisa mengetahui aplikasi apa saja yang telah dibuat oleh pengembang X, seeta berbagai persoalan lainnya.

Salah satu teknik yang digunakan oleh Google untuk menyelesaikan persoalan tersebut adalah dengan menerapkan teori pohon dan graf.

II. LANDASAN TEORI

II. 1 Graf

Teori Graf merupakan salah satu pokok bahasan dari ilmu matematika yang membahas tentang objek-objek diskrit dan hubungan antara objek tersebut. Pembahasan tentang graf telah dimulai sejak lama dan telah diterapkan di banyak bidang hingga saat ini.

Graf G didefinisikan sebagai pasangan himpunan (V, E) , yang dalam hal ini:

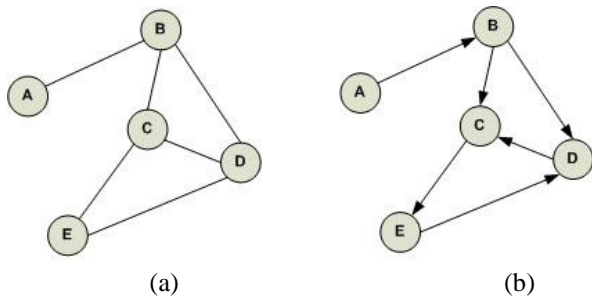
- V = himpunan tidak-kosong dari simpul-simpul (*vertices* atau *node*) = $\{v_1, v_2, v_3, \dots, v_n\}$
- E = himpunan sisi (*edge* atau *arcs*) yang menghubungkan antara dua simpul = $\{e_1, e_2, e_3, \dots, e_n\}$.

Dari definisi diatas, diketahui bahwa sebuah graf dapat tidak memiliki sisi, akan tetapi harus memiliki minimal satu simpul. Graf yang tidak memiliki sisi dan hanya memiliki satu simpul ini disebut **graf trivial**.

Sisi pada graph dapat memiliki orientasi atau arah. Berdasarkan ada atau tidaknya arah pada sisi, graf dibagi menjadi dua seperti pada gambar 1, yaitu:

1. **Graf tak berarah**
2. **Graf berarah**

Agar lebih jelas, perhatikan gambar di bawah ini:



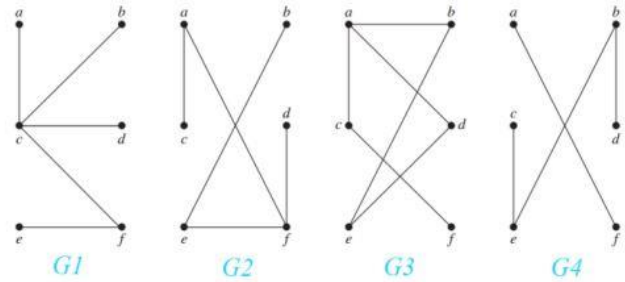
Gambar 1. a) Graf tak berarah b) Graf berarah
Sumber: www.codediesel.com/

Dalam teori graf terdapat beberapa terminologi dasar sebagai berikut:

- Bertetangga** atau *adjacent*
Dua buah simpul dikatakan bertetangga jika kedua simpul tersebut dihubungkan secara langsung oleh sebuah sisi.
- Bersisian** atau *incident*
Untuk sembarang sisi $e = (v_j, v_k)$, sisi e dikatakan bersisian dengan simpul v_j dan v_k .
- Simpul terpencil**
Sebuah simpul dikatakan terpencil jika simpul tersebut tidak memiliki tetangga
- Graf Kosong**
Graf kosong atau disebut juga *Null Graph* atau *Empty Graph* merupakan graph yang tidak memiliki sisi (himpunan sisinya adalah himpunan kosong).
- Derajat**
Derajat sebuah simpul pada suatu graf merupakan banyak sisi yang bersisian dengan simpul tersebut.
- Lintasan**
Lintasan pada graf adalah kumpulan simpul dan sisi yang berselang seling sehingga membentuk jalur dari simpul ke simpul
- Sirkuit**
Sirkuit adalah lintasan yang dimulai dari suatu simpul dan berakhir di simpul yang sama.
- Terhubung**
Sebuah graf dikatakan graf terhubung jika setiap kombinasi dua simpul dalam himpunan simpul memiliki lintasan sehingga setiap simpul dapat dicapai dari simpul yang lainnya.
- Upagraf** atau *subgraph*
Upagraf merupakan himpunan simpul dan sisi yang merupakan himpunan bagian dari sebuah graf. Himpunan simpul pada upagraf tidak boleh kosong sedangkan himpunan sisi boleh kosong, sisi pada upagraf harus merupakan penghubung antara dua simpul pada himpunan simpul upagraf.

II. 2 Pohon

Pohon merupakan suatu graf yang terhubung yang tidak memiliki arah dan tidak memiliki sirkuit. Berikut adalah contoh pohon dan bukan pohon:



Gambar 2. G1 dan G2 merupakan pohon sedangkan G3 dan G4 bukan pohon karena pada G3 terdapat sirkuit sedangkan graf G4 bukanlah graf terhubung^[2]

Misalkan terdapat suatu graf G dengan simpul (V) dan sisi (E) dimana $G = (V, E)$ dalam graf tak berarah sederhana dengan jumlah simpul n . Maka sebuah pernyataan dibawah ini adalah ekuivalen:

- G adalah pohon
- Setiap pasang simpul di dalam G terhubung dalam satu lintasan tunggal
- Jumlah sisi pada G adalah $e = n - 1$ sisi
- G tidak mengandung sirkuit
- Penambahan satu sisi pada graf hanya akan membuat satu sirkuit saja
- Pada G terhubung semua sisinya adalah jembatan

Berikut adalah istilah-istilah yang sering digunakan dalam teorema pohon atau *tree*.

- Akar** merupakan simpul teratas dari sebuah pohon (memiliki aras 0).
- Daun** merupakan simpul yang tidak memiliki percabangan kebawah (anak).
- Simpul Dalam** adalah semua simpul pada pohon yang bukan merupakan akar dan bukan merupakan daun.
- Cabang** adalah jalur yang menghubungkan akar dengan daun
- Simpul orang tua** atau *parent*. Simpul v_1 dikatakan parent dari v_2 jika v_1 terhubung secara langsung dengan v_2 dan v_1 memiliki aras lebih besar satu daripada v_2
- Simpul anak** atau *child*. Simpul v_1 dikatakan anak dari simpul v_2 jika simpul v_2 merupakan parent dari simpul v_1
- Simpul bersaudara**. Dua atau lebih simpul dikatakan bersaudara apabila simpul-simpul tersebut memiliki parent yang sama.
- Leluhur** atau *Ancestor* dari suatu simpul A merupakan setiap simpul yang merupakan jalur dari simpul A menuju akar (termasuk akar tetapi tidak termasuk simpul A sendiri).
- Upapohon** atau *Subtree* adalah pohon yang

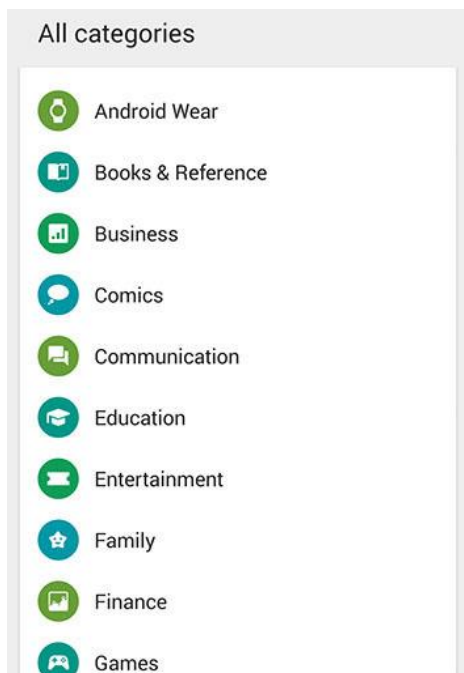
akarnya merupakan salah satu cabang dari suatu pohon.

10. **Derajat** atau *Degree* adalah jumlah upapohon atau jumlah anak dari suatu simpul
11. **Aras** atau *Level* adalah jumlah sisi yang menghubungkannya dengan akar
12. **Tinggi/Kedalaman** atau *Depth* adalah aras maksimum dari suatu pohon

III. PENERAPAN GRAF DAN POHON PADA GOGLE PLAY STORE

1. Penerapan Pohon

Gambar berikut ini merupakan tampilan kategori yang ada di Gogle Play Store.

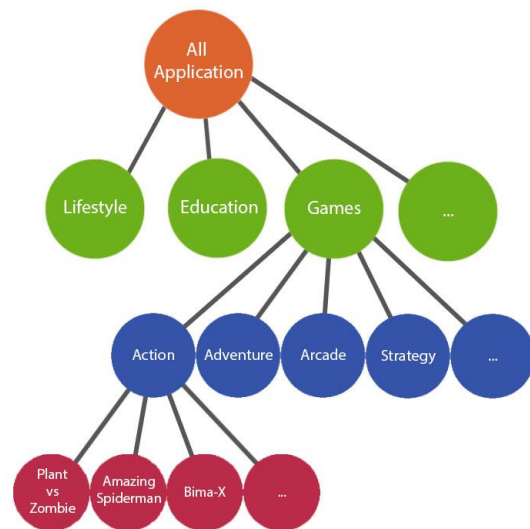


Gambar 4. Kategori aplikasi di Play Store

Dari gambar diatas dapat kita ketahui bahwa Google mengelompokkan setiap aplikasi yang diunggah ke Play Store berdasarkan kategori aplikasi. Hal ini dapat mempermudah user untuk melakukan pencarian aplikasi yang memiliki fungsi tertentu sehingga dapat dijelajahi berdasarkan kategori.

Saat kita memilih salah satu kategori pada gambar diatas maka akan muncul tampilan baru yang menampilkan bagian-bagian dari kategori yang dipilih. Bagian-bagian tersebut merupakan sub kategori dari kategori yang dipilih sebelumnya. Sebagai contoh jika kita memilih kategori *games* maka akan muncul sub kategori berupa *action*, *adventure*, *arcade*, *board*, *card*, *casino*, dll. Lebih lanjut lagi jika kita memilih salah satu sub kategori maka akan ditampilkan aplikasi-aplikasi yang merupakan bagian dari sub kategori tersebut..

Pengkategorian yang dilakukan Google pada Play Store kurang lebih ditunjukkan oleh gambar berikut.



Gambar 5. Pohon Klasifikasi Aplikasi di Play Store

Dari gambar diatas kita dapat mengetahui bahwa Google mengkasifikasikan aplikasi menggunakan struktur pohon. Sesuai teori yang dijelaskan pada Bab II, lingkaran yang berwarna oranye pada gambar diatas merupakan akar atau *root* serta merupakan leluhur atau *ancestor* dari seluruh simpul dan daun di bawahnya. Lingkaran berwarna merah merupakan daun yang merepresntasikan aplikasi-aplikasi pada Play Store. Sedangkan lingkaran yang berwarna hijau dan yang berwarna biru adalah simpul dalam. Diagram diatas merupakan bentuk sederhana dari klasifikasi atau pengelompokan yang dilakukan oleh Google pada aplikasi-aplikasi di Play Store. Sisitim yang terjadi sebenarnya tentu lebih rumit daripada penjelasan diatas.

2. Penerapan Graf

Pada sub bab sebelumnya telah dijelaskan penerapan pohon pada klasifikasi aplikasi di Play Store. Pada bagian teori juga telah dijelaskan bahwa pohon adalah graf terhubung yang tidak memiliki arah maupun sirkuit. Klasifikasi dari aplikas-aplikasi menggunakan teorema pohon sebenarnya merupakan salah satu pemanfaatan graf karena pohon juga merupakan graf. Namun, di sub bab ini akan dijelaskan penerapan graf pada Play Store dalam menghubungkan aplikasi-apliaksi yang ada di dalamnya.

Terdapat beberapa hubungan antar aplikasi yang terdapat pada Play Store seperti:

- a. Kesamaan kategori atau fungsi
- b. Kesamaan pengembang atau *developer*

Beberapa Aplikasi dari developer yang sama mungkin memiliki kategori yang berbeda. Untuk itu dibutuhkan sebuah cara agar pengguna Android atau pengembang aplikasi dapat melihat daftar aplikasi yang telah dibuat oleh sebuah *developer X* misalnya.

Saat kita membuka halaman dari suatu aplikasi di Play Store, pada bagian bawah terdapat pilihan "*more from developer*" atau jika menggunakan bahasa Indonesia

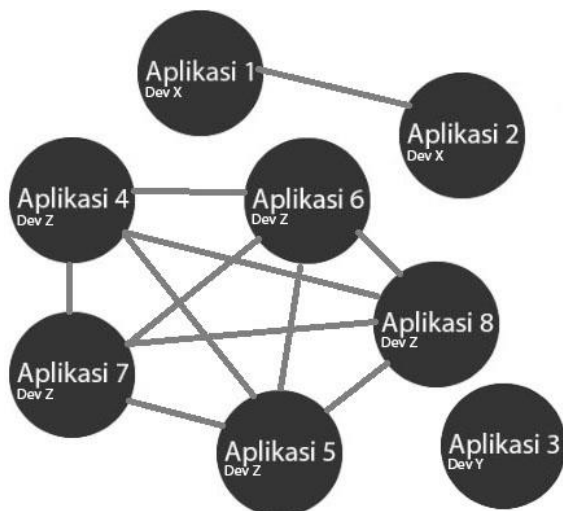
“lainnya dari pengembang”. Jika kita membuka pilihan tersebut, akan ditampilkan daftar aplikasi dari pengembang yang sama dengan pengembang aplikasi yang kita lihat sebelumnya. Hubungan antar aplikasi tersebut dapat dijelaskan menggunakan teorema graf.

Representasi graf dalam pengelompokan aplikasi berdasarkan pengembangnya adalah sebagai berikut. Simpul menunjukkan suatu aplikasi, sedangkan sisi yang menghubungkan antar dua simpul menunjukkan kedua aplikasi tersebut dibuat atau dikembangkan oleh pengembang yang sama. Sebagai contoh, terdapat beberapa aplikasi dan pengembang seperti pada table berikut.

DAFTAR APLIKASI ANDROID DAN PENGEMBANGNYA	
Nama Aplikasi	Pengembang
Aplikasi 1	Dev X
Aplikasi 2	Dev X
Aplikasi 3	Dev Y
Aplikasi 4	Dev Z
Aplikasi 5	Dev Z
Aplikasi 6	Dev Z
Aplikasi 7	Dev Z
Aplikasi 8	Dev Z

Tabel 1. Contoh daftar aplikasi Android dan pengembangnya

Tafsiran graf dari tabel diatas adalah sebagai berikut



Gambar 6. Contoh kasus graf pada Play Store

Gambar diatas memperlihatkan graf dari aplikasi-aplikasi beserta hubungan kesamaan pengembang. Jika kita amati, setiap pengembang pada graf seluruh aplikasi akan memiliki upagraf atau *subgraph* berupa graf lengkap. Graf lengkap adalah graf yang setiap simpulnya terhubung ke seluruh simpul lain pada graf itu. Hal ini

memungkinkan halaman sebuah aplikasi dapat menampilkan seluruh aplikasi dengan pengembang yang sama.

Pada contoh kasus diatas, simpul aplikasi 1 memiliki derajat satu (hanya memiliki satu sisi yang bersisian). Artinya simpul tersebut mempunyai satu aplikasi yang *developer* nya sama dengan *developer* aplikasi tersebut yaitu Dev X, aplikasi yang dimaksud adalah aplikasi 2. Simpul aplikasi 3 memiliki derajat nol. Artinya aplikasi 3 merupakan karya satu satunya dari Dev Y yang diupload ke Play Store. Jadi, pada halaman aplikasi 3 tidak terdapat pilihan “*more from developer*”. Selanjutnya simpul aplikasi 4 memiliki derajat 4. Oleh karena itu pada halaman aplikasi 4 akan ditampilkan empat aplikasi dengan *developer* yang sama yaitu aplikasi 5, aplikasi 6, aplikasi 7, dan aplikasi 8. Begitu juga pada halaman dari keempat aplikasi tersebut, akan ditampilkan 4 aplikasi lainnya dari Dev Z.

IV. KESIMPULAN

Dari pembahasan diatas kita dapat mengetahui bahwa Google menggunakan pohon atau *tree* dalam mengklasifikasikan aplikasi-aplikasi yang ada di Play Store. Dengan adanya klasifikasi semacam ini, pengguna Android menjadi lebih mudah dalam menemukan aplikasi yang ingin dicari. Pada tampilan menu sebuah aplikasi di Play Store terdapat pilihan “*more from developer*”. Pilihan ini akan menampilkan aplikasi-aplikasi yang dibuat oleh pengembang atau *developer* yang sama dengan pengembang aplikasi sebelumnya. Fungsi “*more from developer*” ini memanfaatkan teori graf dalam menghubungkan aplikasi. Tanpa penerapan teori pohon dan graf pada Play Store, pencarian sebuah aplikasi diantara jutaan aplikasi yang ada di Play Store akan susah, tidak efisien, serta menghabiskan banyak waktu.

UCAPAN TERIMAKASIH

Puji syukur penulis ucapkan kepada Allah S.W.T atas segala nikmat yang telah diberikan baik berupa nikmat iman, kesehatan maupun kekuatan dalam menyusun makalah ini. Penulis juga mengucapkan terimakasih kepada kedua orang tua yang berada di kampung halaman yang telah mendidik dan membesarkan penulis dengan penuh kasih sayang. Terimakasih kepada kedua saudara penulis yang selalu memberikan motivasi untuk meraih impian sehingga penulis dapat melanjutkan pendidikan di kampus ini. Selanjutnya penulis ingin menyampaikan terimakasih kepada dosen Matematika Diskrit, Bapak Dr.Ir. Rinaldi Munir, MT. dan Ibu Dra. Harlili, M.Sc., yang telah mencurahkan banyak ilmu kepada kami “warga labtek V”. Semoga ilmu yang beliau berikan dapat kami pergunakan dengan semestinya. Semoga Allah membalasi kebaikan orang-orang yang

disebut diatas dengan kebaikan yang berlipat ganda, amin.

REFERENCES

- [1] Munir, Rinaldi. "Struktur Diskrit". Program Studi Teknik Informatika, 2008.
- [2] <http://blog.appfigures.com/app-stores-growth-accelerates-in-2014/>
Diakses tanggal 6 Desember 2015
- [3] <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>
Diakses tanggal 7 Desember 2015
- [4] Rahmatullah, Bintang, "Pembuktian Teorema Goodstein Aplikasi Penggunaan Graf Pada Sistem Website Video Streaming Youtube". Program Studi Teknik Informatika 2011.
- [5] Warnita, Tifani. "Representasi Model Pohon Berakar pada Permainan Hydra dalam". Program Studi Teknik Informatika, 2013.

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, 10 Desember 2015

Amal Qurany
13514078