

Aplikasi Pohon Dalam Pengelolaan *Database*

Andrey Simaputera (13512058)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
andrey.simaputera@yahoo.com

Abstract — Semua orang di dunia ini pasti memiliki banyak hal yang harus dikelola. Hal-hal tersebut termasuk mengelola jadwal, mengelola data pengeluaran, mengelola prioritas, mengelola biodata, dan lain-lain. Di dalam makalah ini akan dibahas bagaimana cara pengelolaan berbagai macam data yang ada di dunia ini dalam struktur pohon.

Index Terms — Data, Database, Pengelolaan, Struktur Pohon

I. PENDAHULUAN

Dewasa ini, pengelolaan *data* dalam hidup manusia sangatlah esensial dikarenakan oleh beberapa hal, di antaranya yaitu untuk memudahkan pencarian sebuah *data*, mengatur skala prioritas dalam hidup kita, memantau pengeluaran kita berdasarkan waktu yang ditentukan (misal per minggu atau per bulan), dan juga mengetahui sebuah *requirement* sebelum kita melakukan sebuah aksi.

Data yang sangat banyak tersebar di mana-mana di dunia ini, di perkantoran, di rumah sakit, di bank, dan bahkan di *personal computer* yang kita miliki. Banyak dari *data* tersebut yang dibiarkan begitu saja dan tidak dikelola sehingga terkadang kita mengalami kesulitan untuk mencarinya dan mengaksesnya dalam waktu yang cepat saat kita terburu-buru. Oleh karena alasan itu, manusia menciptakan sistem-sistem yang memudahkan kita untuk mengelola *database* dan untuk mencari sebuah *datum* yang berada di sekumpulan *data* yang kita miliki. Dengan *data* yang dikelola dengan rapi, niscaya pengaksesan dan pencarian *data* akan dilakukan dalam waktu yang relatif lebih cepat dan singkat.

Data yang tersebar di masyarakat bisa juga berbentuk dalam sebuah rangkaian aktivitas atau rutinitas yang seseorang miliki di dalam kehidupannya sehari-hari, mulai dari jadwal sehari-hari ataupun juga berbentuk sebagai penentuan skala prioritas untuk jadwal dari kegiatan-kegiatan yang dimiliki oleh masing-masing orang. Struktur pohon dapat digunakan untuk mengaplikasikan hal ini dan lebih lanjut akan dibahas kemudian di dalam makalah ini.

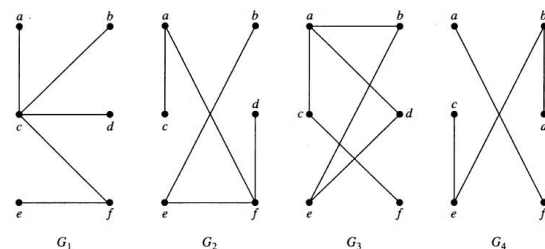
Di dalam bank atau rumah sakit, misalnya,

pengelolaan *database* adalah sebuah hal yang sangat penting dan esensial karena *data* dari seorang pasien atau nasabah bank mungkin akan dicari dan diperlukan sewaktu-waktu. Jika *data* tersebut tidak dikelola dengan baik, maka waktu yang dibutuhkan untuk mencari *data* tersebut akan cenderung lama dan merugikan nasabah atau pasien tersebut. Oleh karena itu sebuah sistem yang sangkil dan mangkus sangat diperlukan dalam pengelolaan *database*, apalagi jika bank atau rumah sakit yang bersangkutan adalah sebuah bank atau rumah sakit yang berskala besar atau internasional.

Data yang dikelola dapat juga berupa aksi-aksi yang saling berhubungan satu sama lain. Contoh aksi yang dimaksud misalnya, sebelum seseorang tidur, dia harus sudah menggosok giginya, mencuci kakinya, mencuci mukanya, dan juga berdoa. Lalu misalnya lagi sebelum seseorang diperbolehkan mengikuti ujian dia harus sudah membayar penuh uang sekolah dalam semester yang bersangkutan, mengenakan pakaian yang sopan selama ujian berlangsung, dan membawa kartu tanda peserta ujian.

II. TEORI POHON

Pohon adalah sebuah graf tak berarah terhubung yang tidak memiliki sirkuit. Karena pohon tidak memiliki sirkuit, pohon tidak mungkin bisa memiliki *loop* yang banyak dan lebih lanjut bisa dikatakan bahwa semua pohon adalah sebuah graf sederhana.

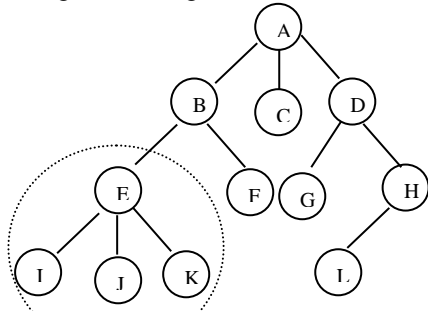


Gambar 1. Sekumpulan Graf

G1 dan G2 pada gambar adalah pohon
G3 bukan pohon karena mengandung sirkuit di dalamnya
G4 bukan pohon karena tidak terhubung

Sebenarnya pohon bisa dikelompokkan ke dalam berbagai macam jenis, tetapi dalam makalah ini hanya akan dibahas beberapa saja, yaitu:

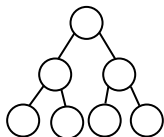
1. Pohon berakar (*rooted tree*): pohon berakar adalah sebuah pohon yang salah satu simpulnya dijadikan sebagai akar (*root*) dan sisanya diletakkan di bawah akar dan dihubungkan dengan sebuah garis.



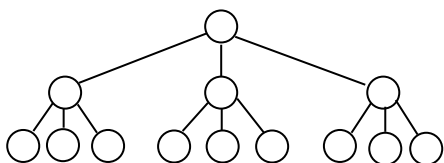
Gambar 2. Sebuah Pohon Berakar T

Pada Gambar Pohon T di atas ada beberapa hal yang dapat disimpulkan, yaitu:

- A adalah akar dari pohon T
 - E adalah *parent* dari J, K, dan L
 - G adalah *sibling* dari H
 - Gambar yang dilingkari adalah upapohon dari pohon T
 - Kedalaman/*Level* dari A yaitu 0, E yaitu 2
 - Derajat/*Degree* dari A adalah 3, B adalah 2, C adalah 0
 - Lintasan (*path*) dari A ke L adalah A,D,H,L dengan panjang 3
 - Simpul dalam adalah simpul yang memiliki anak, yaitu B, D, E, H
 - Daun adalah simpul berderajat 0, yaitu C, I, J, K, dan L
 - dan seterusnya
2. Pohon *n-ary*: pohon *n-ary* adalah pohon berakar yang setiap cabangnya paling banyak memiliki *n* buah anak. Pohon *n-ary* yang semua simpul dalam dan akarnya tepat memiliki *n* anak disebut juga pohon *n-ary* teratur atau pohon *n-ary* penuh. Contoh dari pohon *n-ary* penuh adalah sebagai berikut:



Gambar 3. Pohon 2-ary/binary penuh

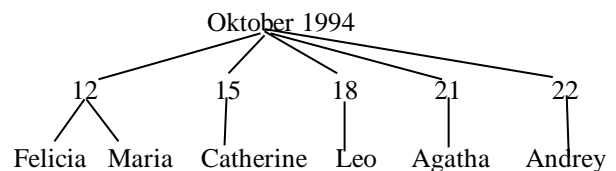


Gambar 4. Pohon 3-ary/tertiary penuh

III. APLIKASI STRUKTUR POHON DALAM PENGELOLAAN DATABASE

A. Data Sederhana

Untuk mengelola *data* yang sederhana, tidak dibutuhkan cara yang terlalu rumit, seperti namanya, sederhana. Pengelolaan *data* daftar kelahiran misalnya, tidaklah begitu sulit. Daftar kelahiran bayi yang ada di rumah sakit dapat dikelola menggunakan struktur pohon. *Data* dari seluruh bayi yang pernah lahir di rumah sakit XYZ, contohnya, dikelola menjadi sekumpulan pohon, atau bisa juga disebut hutan (*forest*). Pohon-pohon ini dipisahkan berdasarkan tahun kelahiran, jika ingin lebih spesifik lagi berdasarkan bulan pada tahun tertentu. Anak dari simpul bulan sebanyak *n* buah dengan *n* adalah tanggal-tanggal di mana ada bayi yang lahir. Lebih lanjut anak dari tanggal tersebut ada sebanyak *m* buah dengan *m* adalah nama-nama bayi yang lahir di tanggal yang bersangkutan.



Gambar 5. Contoh pohon tanggal kelahiran

Selain pohon tanggal kelahiran, pengelolaan *data* yang sederhana bisa juga diaplikasikan untuk mengelola *data* nasabah pada sebuah bank. Pengategorian nasabah pada bank mungkin diatur berdasarkan status nasabah ataupun bisa juga berdasarkan tanggal registrasi, tergantung kebijakan dari bank masing-masing. Bayangkan jika data nasabah tidak dikelola dengan baik dan semestinya. Kemungkinan akan terjadi hal-hal yang tidak diinginkan. Contohnya jika ada nasabah yang kehilangan uang tabungannya, masalah ini tidak akan bisa diproses dengan cepat dan tepat jika data nasabah tidak dikelola dengan baik. Jika ada pengelolaan data yang tepat dan benar maka masalah pasti akan cepat terselesaikan.

Pengelolaan *data* sederhana bisa juga diaplikasikan di dalam komputer Anda masing-masing dalam hal mengelola *file* yang Anda miliki. Kita bisa mengategorikan semua *file* yang ada di komputer kita sesuai dengan kategori *file* tersebut masing-masing. Kategori paling umum dimisalkan sebagai akar dan semakin ke bawah (semakin *level* pohon naik/ maka pengategorian akan semakin khusus). Di komputer Anda, pemisalan seperti ini adalah sebagai berikut. *Folder* yang ada di komputer anda digambarkan sebagai simpul dalam (termasuk akar) dan *file* atau *data* yang kita cari digambarkan sebagai daun atau akhir dari pencarian kita. Contoh dari sebuah pengategorian misalnya pengategorian lagu atau pengategorian film.

Pada pengategorian lagu, *folder* lagu dikategorikan sebagai akar, lalu anak dari *folder* lagu adalah *folder genre* lagu. Anak dari *folder genre* adalah penyanyi, anak dari penyanyi adalah album, dan anak dari album adalah daftar lagu yang kita bisa telusuri dan cari lagu mana yang akan kita pilih untuk kita putar. Setiap *folder* yang ada pada pohon pengategorian ini diibaratkan sebagai sebuah simpul

Tidak berbeda jauh halnya dengan kasus pengategorian *film/movie* yang juga ada di komputer anda, *folder film* bisa dikategorikan dengan setidaknya dua buah cara, yaitu dengan cara dikategorikan berdasarkan *genre* film tersebut, *action*, *drama*, *comedy*, *fantasy*, *animation*, ataupun kita bisa juga mengkategorikan *film-film* berdasarkan tahun rilis dari *film* tersebut. Selain kedua cara tersebut, pengguna bisa saja mengategorikan *film* yang dia miliki dengan kategori yang ingin dia buat sendiri, misal: berdasarkan sutradara *film*, atau aktor dan aktris yang berperan di dalam film tersebut.

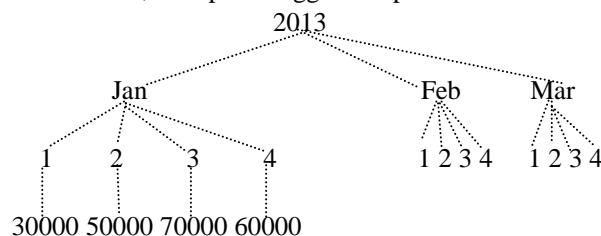


Gambar 6. Pengaksesan lagu dari Taylor Swift - *Love Story*

Gambar 6 di atas menunjukkan contoh pengategorian lagu dan cara pengaksesan sebuah lagu bergenre *country* yang dinyanyikan oleh Taylor Swift yang berjudul *Love Story* di dalam folder *Music* dan di dalam album *Fearless*. Pengelolaan data dengan cara pengategorian seperti ini memberikan beberapa keuntungan bagi para pengguna komputer. Keuntungan yang terutama dan yang pasti yaitu kesanggupan waktu yang diperlukan untuk mencari *file* yang kita maksud. Yang kedua adalah *file* Anda akan terlihat lebih teratur dan tertata dengan rapi sehingga lebih bagus untuk dilihat. Keuntungan-keuntungan inilah yang ditawarkan oleh pengelolaan *data* menggunakan struktur pohon. Tanpa kita sadari, aplikasi struktur pohon telah digunakan dan diterapkan oleh para pembuat sistem operasi di dalam sistem penyimpanan dan pengorganisasian *data* yang berada di komputer Anda, apa pun sistem operasi yang Anda jalankan di komputer anda baik *Windows*, *Linux*, *Macintosh*, dan lain-lain. Penerapan dan pengaplikasian

struktur pohon di dalam sistem operasi tersebut tidaklah sama persis, tetapi memiliki dasar yang sama.

Lebih lanjut, pengelolaan *data* sederhana juga dapat diaplikasikan ke berbagai macam hal dan bidang di hidup ini. Selain untuk pengelolaan *file* yang ada di komputer Anda, pengelolaan data nasabah bank dan pasien rumah sakit, *data* yang dikelola dalam bentuk pohon dapat juga berupa daftar pengeluaran dalam durasi tertentu, baik per minggu atau per bulan.



Gambar 7. Contoh pohon pengeluaran per minggu yang ada di bulan tertentu (hanya bulan Januari yang diisi karena keterbatasan tempat)

Data pengeluaran tersebut dapat dibuat secara terpisah, yakni dengan membuatnya sebagai subpohon dari sebuah pohon yang lebih besar nantinya. Subpohon yang dimaksud digambarkan dengan bulan dan pohon yang lebih besar yang dimaksud adalah pohon dengan tahun sebagai akar pohon. Pengelolaan *data* pengeluaran ini dimaksudkan agar orang yang membuatnya dapat mengontrol pengeluaran yang dia lakukan sehingga tidak kehabisan uang sebelum dapat gaji, misalnya.

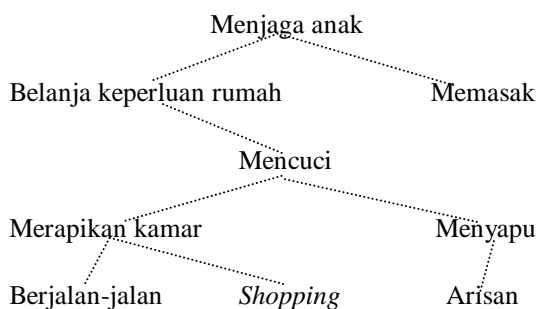
B. Pohon Prioritas

Pengelolaan *data* menggunakan struktur pohon juga dapat mencakup *data* yang memiliki skala prioritas tertentu (misalnya daftar aktivitas) yang dimiliki oleh setiap orang. Setiap orang pasti memiliki kegiatan yang berbeda dan sudah pasti memiliki prioritas masing-masing yang juga berbeda yang menyebabkan pohon prioritasnya akan berbeda apalagi ditambah faktor umur dan juga pekerjaan. Seorang siswa kelas 1 SD sudah tentu beda prioritasnya dengan seorang siswa SMA. Prioritas seorang anak kelas 1 SD biasanya adalah bermain dan siswa SMA biasanya adalah belajar karena harus bersiap-siap untuk memasuki perguruan tinggi yang diinginkan. Jika siswa SMA memprioritaskan bermain maka dia bisa kehilangan kesempatan untuk memasuki perguruan tinggi yang dia impikan. Selain itu, prioritas seorang ibu rumah tangga dan seorang *businesswoman* sudah tentulah berbeda. Seorang *businesswoman* mungkin akan lebih mementingkan sebuah rapat yang penting bersama direktur dibandingkan waktu makan malam bersama keluarga besar karena menurutnya makan malam bisa dilakukan kapan-kapan namun peluang bisnis tidak akan datang dua kali. Sebaliknya, seorang ibu rumah tangga mungkin

akan lebih mementingkan kepentingan di rumahnya dibandingkan berjalan-jalan dengan teman-temannya.

Sebuah struktur pohon prioritas, seperti namanya, disusun berdasarkan prioritas yang ditentukan oleh seseorang. Setiap aktivitas diberi atau dibubuhkan sebuah "level of priority" seperti layaknya level/depth dari sebuah pohon. Semakin dalam atau semakin tinggi level sebuah pohon prioritas maka semakin rendah tingkat kepentingannya dibandingkan simpul yang berada di level yang lebih kecil. Sebaliknya, semakin dangkal atau semakin rendah level sebuah pohon prioritas maka semakin tinggi tingkat kepentingan yang dimiliki dan aktivitas tersebut yang harus dikerjakan terlebih dahulu dibandingkan aktivitas-aktivitas yang memiliki tingkat prioritas yang lebih rendah.

Hal yang penting dan poin yang harus diperhatikan dari struktur pohon prioritas ini bukanlah hubungan antar simpul seperti anak (*children*), saudara (*siblings*), ataupun orangtua (*parents*) dari sebuah simpul melainkan tingkat (*level*)/kedalaman (*depth*) dari simpul yang bersangkutan. Maksudnya hubungan di sini yaitu tidak penting kegiatan apa anak dari kegiatan apa atau kegiatan apa adalah orang tua dari kegiatan apa. Satu-satunya kegunaan dari hubungan anak atau orangtua yaitu untuk membedakan skala prioritas/level antar simpul atau dalam hal ini yaitu kegiatan. Dan kegunaan dari hubungan saudara (*sibling*) yaitu untuk menentukan kegiatan-kegiatan yang mana saja yang memiliki tingkat prioritas yang sama yang artinya aktivitas mana saja yang dikerjakan duluan bukanlah masalah.



Gambar 8. Contoh pohon prioritas untuk ibu rumah tangga pada umumnya

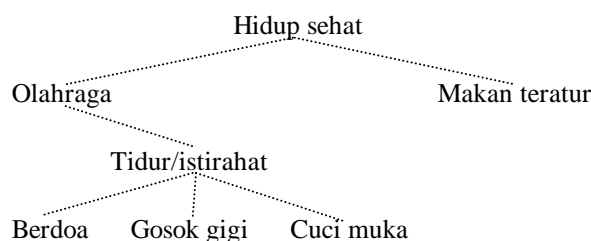
Kita bisa lihat bahwa gambar 8 di atas menunjukkan sebuah pohon prioritas untuk seorang ibu rumah tangga pada umumnya. Bisa dilihat di gambar di atas bahwa kegiatan memasak dan mencuci tidaklah terhubung. Tidak ada garis yang menghubungkan antara kegiatan memasak dan kegiatan mencuci sebab jika ada garis yang menghubungkan kedua kegiatan tersebut maka struktur yang ada di atas tidak akan menjadi sebuah struktur pohon melainkan hanya menjadi sebuah struktur graf karena akan terjadi sebuah sirkuit jika ada garis yang menghubungkan kegiatan memasak dan mencuci. Kegunaan struktur pohon di atas yaitu untuk menentukan

kegiatan mana saja yang harus dilakukan pertama kalinya dengan kegiatan yang ada di simpul akar (*level 0*) sebagai kegiatan yang harus diutamakan untuk dilakukan terlebih dulu.

Contoh kasusnya adalah sebagai berikut. Seorang ibu rumah tangga sudah tentu akan menjaga anaknya terlebih dulu sebelum melakukan hal-hal lainnya. Setelah dia mengurus anaknya, maka simpul "Menjaga anak" akan dihapus dari pohon. Setelah itu dia punya pilihan untuk belanja keperluan rumah atau memasak. Mungkin kegiatannya memang bergantung dari situasi yang terjadi, tapi di sini diasumsikan kedua kegiatan tersebut memiliki tingkat prioritas yang sama. Misalnya ibu rumah tangga tersebut memilih untuk memasak terlebih dahulu, maka setelah dia selesai memasak, simpul memasak akan dihapus dari pohon dan kegiatan yang selanjutnya dilakukan sudah pasti belanja keperluan rumah karena sekarang kegiatan itulah yang memiliki level yang paling rendah, dan begitu seterusnya sampai semua kegiatan di pohon prioritas telah dijalani. Perlu diingat bahwa daftar kegiatan di pohon prioritas ini tidaklah *default* yang berarti setiap orang dapat dengan sesuka hati menyesuaikan pohon prioritasnya dengan kegiatan yang mereka jalani masing-masing.

C. Pohon Syarat

Sebuah struktur pohon syarat, seperti namanya, adalah sebuah struktur pohon yang simpul-simpulnya memuat data berupa daftar kegiatan yang saling berkaitan jika memiliki hubungan anak-orangtua (*children-parents*). Pada dasarnya struktur pohon syarat ini mirip dengan struktur pohon prioritas. Hanya saja, bedanya, struktur pohon syarat ini tidak terlalu mementingkan level dari sebuah simpul. Yang lebih penting dan lebih diperhatikan pada struktur ini adalah hubungan anak-orangtua (*children-parents*). Inti dari struktur ini yaitu kegiatan yang ditetapkan sebagai simpul anak harus terlebih dulu dilakukan sebelum kita melakukan sebuah simpul yang ditetapkan sebagai simpul orangtua dari simpul anak tersebut. Agar lebih bisa dimengerti, berikut adalah contoh pohon syarat:



Gambar 9. Contoh pohon syarat

Dapat kita lihat gambar 9 di atas menunjukkan sebuah pohon syarat untuk kita supaya dapat hidup sehat. Untuk bisa hidup dengan sehat, kita harus makan teratur dan

juga berolahraga, Sebelum berolahraga, kita harus sudah tidur/istirahat terlebih dahulu supaya kita memiliki tenaga yang cukup untuk berolahraga karena olahraga tanpa istirahat yang cukup akan berdampak buruk bagi kesehatan. Sebelum tidur/istirahat, hal-hal yang biasa kita lakukan yaitu berdoa, menggosok gigi, dan mencuci muka. Hal unik dari pohon syarat yaitu jika ada salah satu saja kegiatan yang merupakan *siblings* yang belum dilakukan, maka kita tidak bisa melakukan kegiatan yang berada di simpul orangtua dari simpul yang bersangkutan. Misalnya jika kita hanya berolahraga yang cukup saja tanpa makan teratur dan malah makan-makanan yang buruk bagi kesehatan, *junk food* misalnya, maka kita pasti tidak akan bisa hidup sehat. Malah kita akan mudah terserang penyakit. Dan jika kita sudah berdoa dan cuci muka namun belum menggosok gigi, sebaiknya kita tidak tidur/beristirahat dulu.

Struktur pohon syarat ini memang mirip dengan struktur pohon prioritas. Perbedaan yang jelas selain keterkaitan kegiatan yaitu cara memandang pohon. Pada pohon prioritas, kita memandang pohon dari akar (kegiatan yang paling penting) ke daun (kegiatan yang tidak begitu penting untuk dilakukan). Sedangkan pada pohon syarat, kita memandang sebuah pohon dari daun (syarat awal) menuju akar (tujuan yang hendak dicapai/kegiatan yang ingin kita lakukan).

Pohon syarat ini sebenarnya sudah cukup banyak diterapkan saat ini penggunaannya di berbagai bidang di dunia ini, baik pada bidang teknologi, permainan, pabrik, dan lain-lain. Misalnya saja pada permainan *online* yang memperbolehkan setiap karakternya memiliki *job* (pekerjaan) tertentu di *game* tersebut. Struktur pohon syarat ini biasanya digunakan dalam pengaplikasian *skill tree* (pohon kemampuan) dari karakter dengan *job* (pekerjaan) yang bersangkutan di dalam *game*. Misalnya saja seorang pemain *game online* yang memiliki *job* sebagai *mage* (penyihir) perlu mempelajari *skill fire*, *blizzard*, dan *thunder* di pohon kemampuannya sebelum dia bisa mempelajari *skill flare* (*skill magic* lanjutan yang bisa memberikan *damage* yang jauh lebih besar dan efektif dari *skill* dasar yang didapatkan secara *default* di awal permainan). Dengan memantau terus dan mengetahui syarat-syarat *skill tree* selama permainan berlangsung, diharapkan seorang pemain dapat mendapatkan dan memilih *skill* yang sangat bagus secara sangkil karena biasanya di permainan *online*, poin untuk mempelajari *skill* pada *skill tree* terbatas dan tidak semua *skill* dapat dipelajari oleh pemain. Oleh karena itu dibutuhkan strategi yang matang dan juga pengamatan yang baik terhadap pohon syarat (dalam hal ini *skill tree*) tersebut.

IV. KESIMPULAN

Pengelolaan *data* dengan struktur pohon sangatlah bermanfaat jika kita menggunakan struktur yang tepat.

Pemilihan struktur pohon disesuaikan dengan tipe *data* yang ingin dikelola oleh seorang pengguna. Misalnya jika seorang pengguna ingin mengelola *data* yang ada di dalam komputernya masing-masing atau jika seseorang ingin mengurus *database* yang ada di rumah sakit, maka struktur pohon yang digunakan sebaiknya adalah struktur pohon sederhana karena si pengguna tidak memerlukan banyak peraturan untuk mengelola *datanya*, hanya perlu mengelompokkan saja. Lalu selanjutnya jika seseorang sedang dalam masa-masa yang sibuk dan memiliki jadwal yang padat dan kemudian bingung aktivitas mana yang harus dilakukan duluan, maka orang itu sebaiknya membuat sebuah pohon prioritas agar waktu padatnya dapat terkontrol dan semua aktivitasnya dapat dia lakukan tanpa ada satu kegiatan pun yang terlewatkan.

V. UCAPAN TERIMA KASIH

Rasa terima kasih penulis sampaikan kepada Tuhan Yang Maha Esa karena hanya dengan berkat dan kemauan-Nyalah penulis dapat menyelesaikan makalah yang berjudul “Aplikasi Pohon Dalam Pengelolaan *Database*” ini dengan tepat waktu. Rasa terima kasih juga penulis sampaikan kepada Dr. Ir. Rinaldi Munir, MT dan Harlili S., M.Sc. atas bimbingan dan juga pengajarannya selama kuliah Matematika Diskrit berlangsung sehingga penulis mendapatkan ilmu yang sangat berguna dalam menyelesaikan makalah ini dan juga keperluan kuliah selanjutnya. Selain itu penulis juga tidak lupa menyampaikan terima kasih kepada orang tua penulis dan juga teman-teman atas segala masukan, bantuan, semangat dan juga doa mereka.

DAFTAR PUSTAKA

- [1] Rosen, Kenneth. H., “*Discrete Mathematics and Its Applications 6th edition*”, McGraw-Hill International Edition, New York:2007
- [2] Munir, Rinaldi, “*Matematika Diskrit*”, Informatika, Bandung: 2010

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 2013



Andrey Simaputera 13512058