

Pencarian dan Pengolahan File Menggunakan Metode Pohon Biner pada Program

Geraldi Anggapardana (13511097)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
¹13511097@std.stei.itb.ac.id

ABSTRAK

Makalah ini menjelaskan tentang cara mencari sebuah kata atau lebih pada program dengan menggunakan metode pohon biner. Metode ini efektif karena tidak akan “menyulitkan” program dalam mencari suatu kata yang dimaksud. Dengan melihat dari tiap char apakah sama dengan akar. Bila lebih kecil, maka akan ke kiri, kalau lebih besar ke kanan. Jika sama, dilihat char kedua, dan seterusnya. Jika pada pencarian sudah mencapai daun dan belum ditemukan suatu data yang dituju, maka dapat disebutkan bahwa data yang dicari tidak ada. Penambahan dan penghapusan jumlah buku juga dapat dilakukan dengan menggunakan pohon biner tersebut.

Kata kunci—pohon biner, sorting, pembagian abjad pada pohon

I. PENDAHULUAN

Bumi ini memiliki berbagai macam makhluk hidup. Ada tiga jenis makhluk hidup utama, yaitu manusia, hewan, dan tumbuhan. Pada, tumbuhan banyak kita jumpai bermacam-macam ragam dan jenis, salah satunya adalah pohon. Tidak mungkin orang tidak mengetahui jenis makhluk hidup yang satu ini. Pohon adalah tumbuhan yang memiliki tiga unsur atau bagian utama yang ada padanya, yaitu akar, batang, dan daun. Setiap bagian memiliki fungsi yang berbeda-beda. Akar adalah bagian terpenting suatu pohon karena akar berfungsi untuk menyerap air dan makanan lainnya. Dindingnya dapat menyerap air dan akar berfungsi sebagai penyangga pohon agar tetap tegak. Batang tumbuh memanjang dan memiliki lapisan yang kuat dan keras, sedangkan daun adalah bagian terujung dari suatu pohon dan berfungsi menangkap cahaya matahari agar tetap hijau.

Bagian atau struktur inilah yang digunakan ilmu matematika sebagai pola implementasi graf. Graf disini memiliki tiga unsur, akar, batang, dan daun. Pohon ini akan dibahas lebih lanjut pada bab berikutnya.

Pada teknologi sekarang yang sudah maju sangat pesat, banyak sekali teknologi yang baru atau teknologi inovasi. Hampir semua pekerjaan, benda, atau apapun dalam kehidupan kita. Salah satunya adalah program untuk mencari sebuah data, yang dapat berbentuk kata/string atau angka / integer.

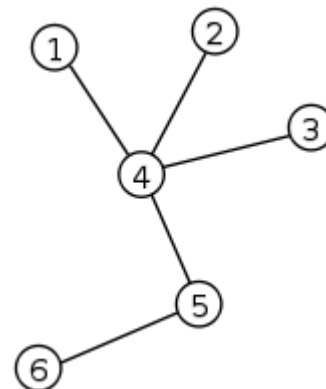
Pada makalah ini, akan dibahas mengenai pohon biner yang diimplementasikan pada sebuah kasus pencarian file yang ada pada sebuah suatu tempat. Data-data file yang tersedia akan tersimpan pada sebuah pohon biner terurut dan juga kasus kosong pada pohon tersebut tersimpan jika file yang dicari tidak ada. Proses ini bertujuan agar program tidak terlalu sulit mencari suatu data dan tidak perlu memakan waktu bila mencari data dalam skala besar. Dengan cara ini, pencarian bisa dilakukan dengan cepat, hanya dengan melihat urutan karakter abjad. Dengan dilakukan eksperimen nanti, kita bisa melihat tingkat keefektifan dari penggunaan pohon biner ini dalam mencari suatu data.

II. DASAR TEORI

II.a POHON

II.a.1 Definisi Pohon

Pohon adalah graf khusus. Graf khusus disini berarti graf tak-berarah terhubung yang tidak mengandung sirkuit. Contoh dari pohon adalah sebagai berikut :



Gambar 2.1

Pohon juga bersifat unik untuk setiap lintasan pasang simpul yang ada. Dari contoh gambar 2.1, dari 1 ke 4 hanya satu lintasan, dari 2 ke 4 hanya satu lintasan, dan seterusnya. Berikut ini adalah sifat-sifat dari pohon[1] yang dimisalkan oleh $G = (V, E)$ adalah:

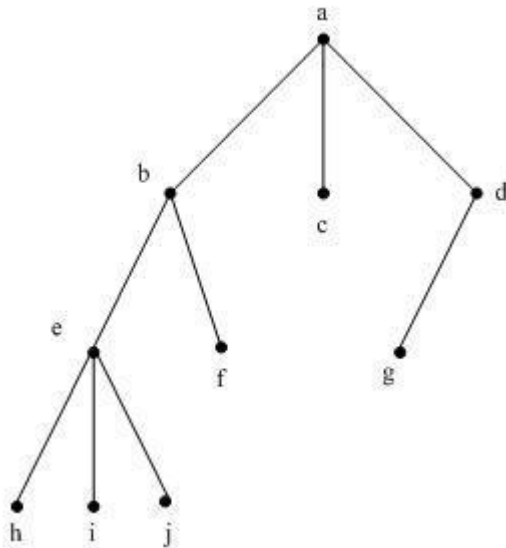
- G adalah pohon
- setiap pasang simpul di dalam G terhubung dengan lintasan tunggal.
- G terhubung dan memiliki $m = n - 1$ buah sisi

- d. G tidak mengandung sirkuit dan memiliki $m = n - 1$ buah sisi
- e. G tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya satu sirkuit.
- f. G terhubung dan semua sisinya adalah jembatan.

Jembatan adalah sisi yang bila dihapus menyebabkan graf terpecah menjadi dua komponen.

II.a.2 Pohon Berakar

Pohon yang sebuah simpulnya diperlakukan sebagai akar dan sisi-sisinya diberi arah menjauh dari akar dinamakan pohon berakar. Bisa dilihat pada gambar berikut :



Gambar 2.2

Keterangan lebih lanjut akan dibahas pada sub-bab selanjutnya.

II.a.3 Terminologi pada Pohon Berakar

Secara keseluruhan, pohon berakar memiliki lintasan yang selalu bergerak dari “atas” ke “bawah”. Dengan meninjau gambar 2.2 (selanjutnya akan menggunakan contoh gambar 2.2) simpul a akan bergerak ke b, c, d kemudian dari b bergerak ke e, f dan seterusnya. Berikut adalah terminology yang digunakan pada pohon berakar :

a. anak dan orangtua

Pada gambar, simpul a dikatakan orang tua dari b,c,d dan b,c,d dikatakan anak dari simpul a jika ada sisi dari simpul a ke b,c,d dan a terletak di atas b,c,d. e dan f adalah anak-anak dari b, dan seterusnya.

b. lintasan

lintasan dari simpul v_1 ke simpul v_k adalah runtunan simpul v_1, v_2, \dots, v_k sedemikian sehingga v_i adalah orangtua dari v_{i+1} untuk $1 \leq i < k$. contoh, lintasan dari a ke l adalah a,d,g,k,l. panjang lintasan adalah jumlah sisi yang dilalui dalam suatu lintasan, $k - 1$.

c. keturunan

pada gambar, b adalah leluhur h dan h adalah

keturunan a.

d. saudara kandung

Simpul dikatakan saudara kandung bilamana memiliki orang tua yang sama. e adalah saudara kandung f, tapi bukan saudara kandung g.

e. upapohon

jika x adalah simpul, maka akar tiap-tiap upapohon x disebut anak , dan x adalah orangtua setiap akar upapohon.

f. derajat

Jumlah upapohon pada simpul tersebut. Misal, a berderajat 3, c berderajat 0, b berderajat 2. Derajat maksimum semua simpul merupakan derajat pohon tersebut.

g. daun

Simpul yang berderajat 0, disebut daun.

h. simpul dalam

Simpul yang mempunyai anak adalah simpul dalam. Contoh, simpul b,e,d.

i. aras / tingkat

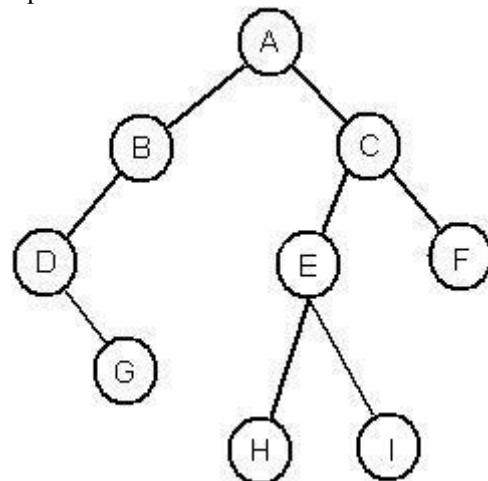
Aras adalah panjang lintasan dari akar ke suatu simpul + 1. Akar memiliki aras 0.

j. tinggi / kedalaman

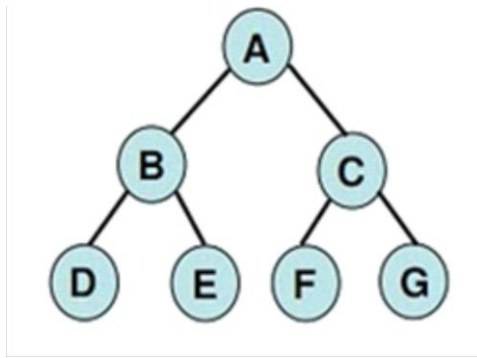
Aras maksimum suatu pohon.

II.a.4 Pohon Biner

Pohon biner adalah pohon yang setiap simpul cabangnya mempunyai paling banyak dua buah anak. Dapat disebut anak kiri dan anak kanan. Pohon yang akarnya adalah anak kiri disebut **upapohon kiri**, sedangkan yang kanan disebut **upapohon kanan**. Berikut adalah contoh gambar pohon biner :



Gambar2.3



Gambar2.4

Gambar 2.4 adalah contoh pohon biner lengkap, yaitu dimana setiap akar memiliki dua anak yaitu upapohon kiri dan upapohon kanan dan derajat daun yang sejajar. Sedangkan gambar 2.3 bukan pohon biner lengkap, karena ada yang hanya memiliki satu buah anak dan derajat daun berbeda-beda.

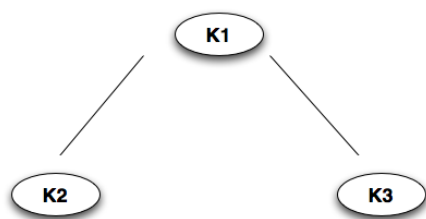
Pohon biner memiliki banyak jenis dan kegunaannya, ada pohon ekspresi, pohon keputusan, kode awalan, kode Huffman, traversal pohon biner, dan pohon pencarian atau biasa disebut *binary search tree* (BST). BST inilah yang akan digunakan pada percobaan kali ini.

II.a.5 Pohon Pencarian Biner (*binary search tree*)

Pohon pencarian biner mungkin adalah pohon biner yang paling penting. BST ini dapat melakukan operasi pencarian, penyisipan, dan penghapusan elemen. Simpul pada BST dapat berupa kunci pada data *record*, atau data itu sendiri. Kunci bersifat unik, karena tidak ada dua atau lebih simpul yang memiliki nilai yang sama.

Ketentuan pengaturan kunci adalah sebagai berikut. Jika R adalah akar, dan semua kunci yang tersimpan pada setiap simpul bersifat unik, maka :

- a. Semua simpul pada upapohon kiri memunyai kunci lebih kecil dari kunci(R)
- b. semua simpul pada upapohon kanan memunyai kunci lebih besar dari kunci (R).



Gambar 2.5

Keterangan :

- $K2 < K1$
- $K3 > K1$

III. PENEMPATAN DATA KEPADA POHON PENCARIAN BINER

Banyak problem yang dapat diselesaikan dengan metode ini. Percobaan kali ini akan mensimulasikan

pendataan buku yang ada di suatu perpustakaan atau taman bacaan. Metode pohon pencarian biner ini dapat menghitung urutan nilai integer maupun urutan nilai *char*. Pada percobaan ini kedua nilai tersebut dapat dipakai, namun dalam kondisi berbeda.

Dimisalkan dalam sebuah taman bacaan, terdapat banyak buku pelajaran dan buku hiburan lainnya seperti komik, majalah, dll. Buku-buku yang ada di taman bacaan ini sudah terdata dan terolah dalam suatu program dimana baik admin maupun pengunjung yang ingin mencari buku tertentu dapat dimudahkan dalam mencari buku yang diinginkan.

Pada suatu tipe data tertentu yang memuat data-data buku, seperti judul buku, nomor buku, banyak buku, status buku (dipinjam/tidak) atau yang lainnya. Pohon pencarian ini cukup menggunakan nomor atau judul buku untuk dapat menyimpan semua tipe data dalam pohon biner. Berikut terdapat beberapa nomor dan judul buku pada taman bacaan berikut:

no.	judul buku
1	bahasa indonesia
2	english
3	matematika
4	biologi
5	kimia
6	fisika
7	planet di sekitar kita
8	pemrograman 1
9	times

Tabel 1

Pada tabel diatas, terdapat dua nilai pada suatu tipe data, yaitu nomor buku dan judul buku. Pada kenyataannya pasti ada ratusan atau ribuan buku dalam suatu taman bacaan, namun tabel ini hanya contoh dari implementasi program yang menggunakan acuan pohon biner.

Pada penempatannya, nilai tengah dari suatu data yang ada itu yang akan dijadikan nilai akar pertama pada pohon. Misal A adalah data nilai terendah, B adalah data nilai terbesar, dan C adalah nilai tengahnya, maka

$$C = (A + B) / 2$$

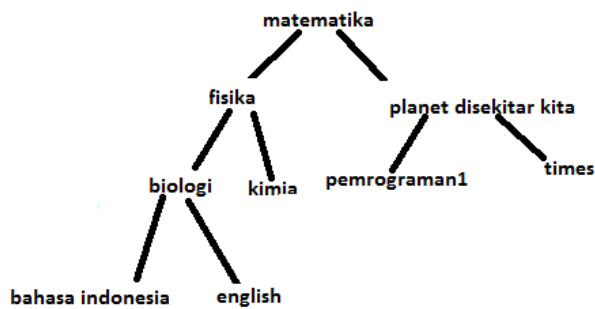
Berlaku untuk nilai integer maupun nilai karakter. Rumus diatas juga ditetapkan dibulatkan ke nilai atas yang paling dekat yang ada, jadi jika ada data 4,8,9, sesuai rumus yang ada diatas :

$$C = (4+9) / 2 = 6,5 \rightarrow 8$$

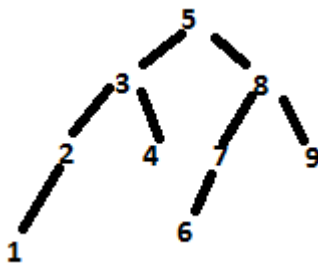
Jadi nilai akarnya adalah 8 dan 4 adalah upapohon kiri, sedangkan upapohon kanan adalah 9.

Tabel 1 tersebut akan direpresentasikan kedalam pohon biner berdasarkan nilai *char* dan nilai *integernya*.

Kedua pohon biner tersebut adalah sebagai berikut :



Gambar 3.1



Gambar 3.2

Pada gambar 3.1 adalah pohon yang berisi data *string* sebagai komponen pencarian, sedangkan gambar 3.2 menggunakan *integer* sebagai komponen pencarian. Pembuatan pohon dibuat dua cara dengan dua nilai berbeda karena pencarian bisa digunakan dengan menulis nomor buku atau judul buku sesuai kebutuhan pembaca atau penjaga taman bacaan tersebut.

Pada gambar 3.1, pohon dibuat sedemikian rupa dan sesuai dengan rumus diatas karena :

- Fisika < matematika
- Biologi < fisika
- Kimia > fisika
- Bahasa inggris < bahasa Indonesia < biologi
- Planet.....> matematika
- Pemrograman< planet
- Pemrograman>matematika
- Times > planet > matematika

Nilai yang tertera diatas mengacu pada urutan dari karakter pertama yang dicek terlebih dahulu. Jika karakter pertama sama, maka diatur dari karakter kedua, dan seterusnya. Pada gambar 3.2 sudah terlihat dari nilai *integer* itu sendiri dengan melihat tabel 1, nilai 5 dijadikan sebagai acuan nilai tengah dan dijadikan sebagai akar utama pada pohon pencarian biner.

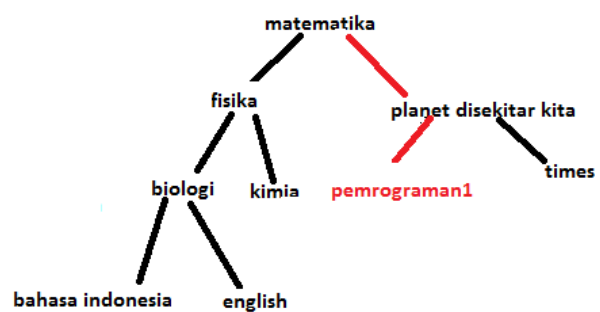
IV. HASIL DAN ANALISIS PENCARIAN DAN PENGOLAHAN DATA PADA POHON BINER

IV.a Proses, Hasil, dan Analisis Pencarian Data

Pada proses percobaan kali ini, percobaan akan mengacu pada gambar 3.1 yaitu dengan menggunakan pohon biner dengan data *string* sebagai acuan dalam percobaan kali ini karena pada dasarnya percobaan ini akan menghasilkan hasil dan cara yang sama.

Seperti gambar yang terlihat pada gambar 3.1, proses pencarian akan dimulai dari data paling atas, yaitu dari matematika. Kemudian akan diperiksa apakah sama atau tidak. Jika tidak maka diperiksa lagi apakah karakternya lebih besar atau lebih kecil dari karakter pada matematika tersebut. Jika lebih kecil, akan turun ke kiri, dan jika lebih besar, akan turun kekanan. Proses bersifat rekursif sampai data ditemukan atau data sudah mencapai daun.

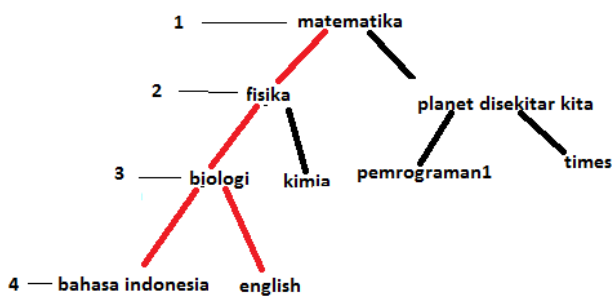
Dilakukan percobaan mencari buku pemrograman1. Percobaan mengacu pada gambar 3.1. Pertama, proses bermula dari buku matematika. Dibandingkan dengan kata yang dicari. Karakter 'P' lebih besar dari karakter 'M', sehingga lintasannya akan bergerak ke upapohon kanan. Pada upapohon kanan, buku "Planet di Sekitar Kita" menjadi kuncinya. Kemudian dibandingkan lagi dengan kata yang dicari. Karena karakter pertama sama, maka diuji karakter kedua. Karakter 'E' pada pemrograman1 lebih kecil pada 'L' pada planet, sehingga pencarian bergerak lagi ke upapohon kiri. Upapohon kiri selanjutnya menjadi kunci, dan ternyata ketika dibandingkan hasilnya sama dengan kata yang dicari. maka buku yang dicari ada dan akan muncul data-data yang ada pada tipe data yang berisi judul buku tersebut.



Gambar 4.1

Gambar menunjukkan cara kerja proses pencarian dilakukan.

Jika dilihat pada gambar 3.1, pencarian terpanjang yang bisa dilakukan oleh pohon pencarian biner tersebut adalah sebanyak cabang atau lintasan terbanyak yang dimiliki oleh pohon pencarian tersebut ditambah data yang ada pada akar utama itu sendiri, yaitu sebanyak 4 kali pencarian.



Gambar 4.2

Sedangkan, jika proses pencarian tidak melakukan pohon dan mencarinya satu per satu seperti pada tabel 1 bab III, pencarian maksimum adalah 9 kali, jika buku yang dicari saat itu adalah Times.

Jadi pada percobaan pencarian ini bisa dibilang jauh lebih cepat dengan pohon biner karena proses pencarian tidak harus dengan menyeluruh, tapi hanya sebagian saja data bisa ditemukan dengan tepat dan cepat.

IV.b Hasil dan Analisis Pengolahan Data

IV.b.1 Penambahan Data

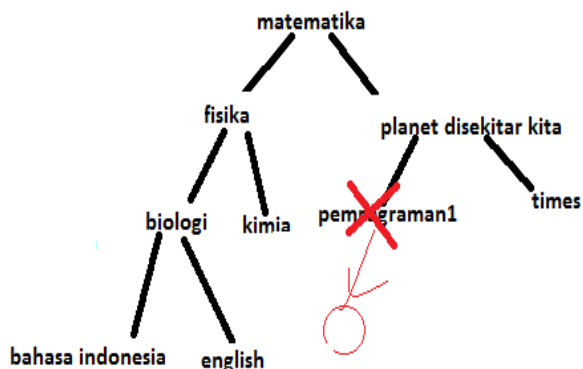
Dilakukan analisa dengan cara penambahan data pada pohon pencarian di gambar 3.1. Diberikan tambahan buku yang ingin dimasukkan ke dalam daftar buku yang kemudian akan dimasukkan ke dalam pohon.

no	judul buku
10	pemrograman2
11	sistem digital
12	geografi

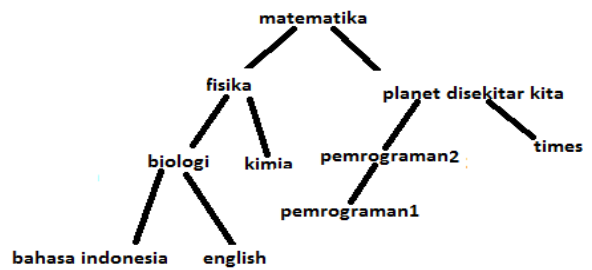
Tabel 2

Sesuai dengan metode dan cara yang sudah disebutkan pada bab II dan bab III, maka penambahan buku pada pohon pencarian memiliki hasil sebagai berikut :

Penambahan buku ke 10

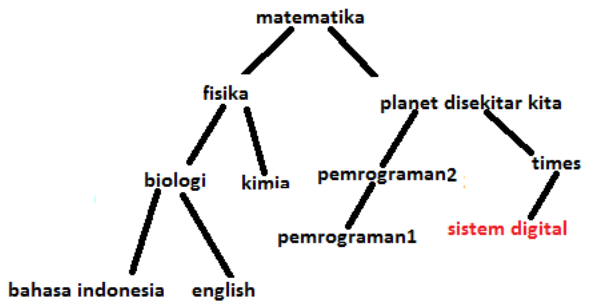


Gambar 4.3



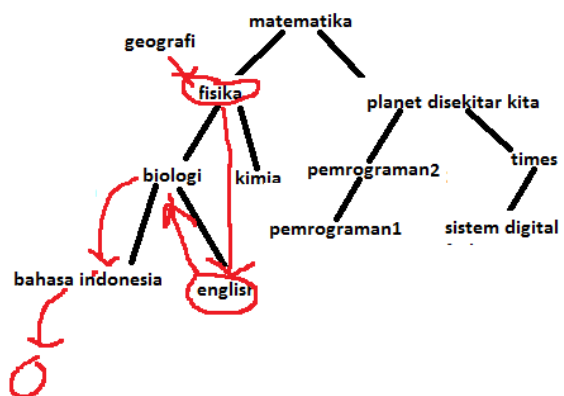
Gambar 4.4

Penambahan buku ke 11

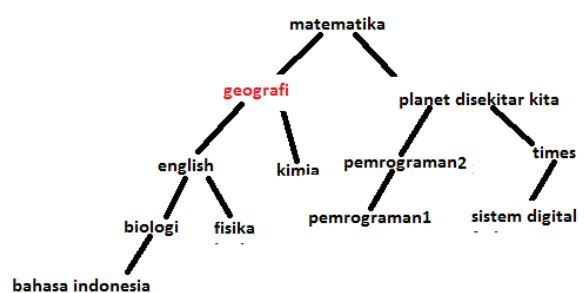


Gambar 4.5

Penambahan buku ke 12



Gambar 4.6



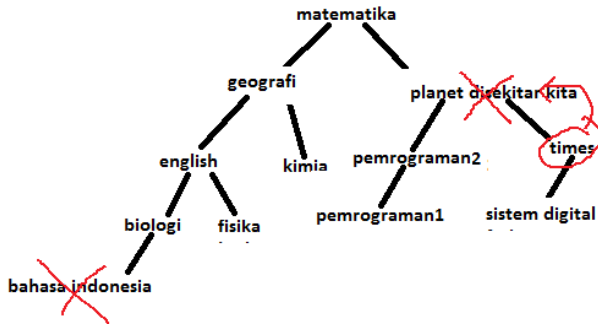
Gambar 4.7

Inti dari tiga percobaan yang ada di atas adalah dengan melakukan pencarian terlebih dahulu, apakah data penambahan lebih kecil atau lebih besar dari data baru. Bila data baru lebih kecil dari data lama dan data lama tersebut daun, maka otomatis data baru akan menjadi upapohon kiri dari key daun tersebut. Bila data baru lebih besar, maka semua key dan keturunannya akan dirombak kembali menjadi sebuah pohon dengan urutan yang baru

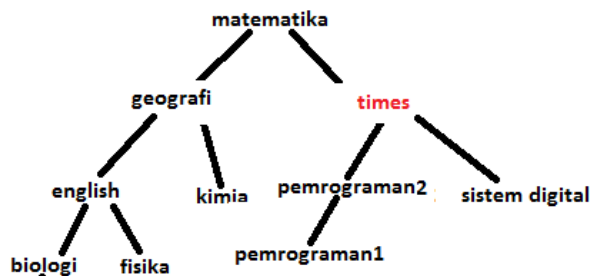
sesuai dengan materi pada bab II dan bab III, dengan data baru menjadi key menggantikan posisi key yang sebelumnya.

IV.b.2 Penghapusan Data

Dilakukan analisa penghapusan elemen data dari sebuah pohon pencarian. Dengan memakai gambar 4.7 sebagai hasil akhir, akan dihapus beberapa buku, yaitu bahasa Indonesia dan Planet di Sekitar Kita. Berdasarkan cara-cara yang sudah disepakati dalam bab II dan bab III, maka proses penghapusan dan pembuatan pohon baru ialah sebagai berikut :



Gambar 4.8



Gambar 4.9

Proses penghapusan ini dilakukan dengan melakukan pencarian kata yang dimaksud terlebih dahulu. Jika kata sama dengan kata yang dimaksud, maka data tersebut akan dihapus dan akan digantikan dengan upapohon kiri atau kanan, sesuai dengan metode penghitungan pada bab III. Jadi semua keturunan dari kata yang dihapus dibandingkan lagi dan dicari titik tengahnya. Jika yang dihapus adalah daun, tidak perlu mengubah struktur pohon.

Pada kedua hasil dan cara pengolahan data, baik penambahan atau penghapusan data membutuhkan proses pencarian terlebih dahulu. Dengan metode pohon pencarian, untuk mencari suatu data menjadi lebih cepat sehingga pada penambahan dan penghapusan elemen pun dapat dilakukan dengan lebih cepat dan tidak memakan waktu lama.

V. KESIMPULAN

Dengan memanfaatkan pohon pencarian biner, proses pencarian suatu data akan dapat dilakukan jauh lebih cepat daripada dengan mencari tanpa menggunakan pohon karena dengan dibuatnya dalam pohon biner

pencarian tidak perlu dilakukan secara keseluruhan. Begitu juga dengan penambahan dan penghapusan data. Dengan proses pencarian ini tentu ada sedikit kendala di penambahan karena harus tetap dilakukan proses pencarian dan 'rekonstruksi' ulang pohon, namun pada proses penghapusan cara ini tetap jauh lebih efektif. Secara keseluruhan metode pohon ini dapat bekerja lebih efektif dan lebih hemat waktu jika data yang tersimpan dalam jumlah yang sangat besar. Dengan demikian bagi seseorang yang sedang mencari atau mengolah sebuah data dalam sebuah file yang banyak terdapat data bisa mencari dan mengolahnya dengan cepat dan tidak memakan waktu.

DAFTAR PUSTAKA

- [1] Munir, Rinaldi. *Matematika Diskrit rev. 5*. Informatika Bandung, 2012
- [2] <http://www.cs.sysu.edu.cn/~zxc/course/discrete/tree02.pdf> , 12 Desember 2012, pk.11.30 WIB
- [3] <http://thesonofdevil.wordpress.com/2009/12/07/pohon-keputusan/> , 14 Desember pk 15.00
- [4] <http://rangga07.wordpress.com/2009/01/02/pembacaan-pohon-tree-dengan-metode-preorder/> , 14 Desember pk 15.35
- [5] <http://udinensdin.wordpress.com/2010/11/> , 14 Desember , pk 15.37 WIB
- [6] <http://www.people.westminstercollege.edu/faculty/ggagne/fall2012/306/handouts/balanced-trees/index.html> , 14 Desember pk 15.52 WIB
- [7] Praworohartono, Slamet dan Sri Hidayati. *Sains Biologi 2*. Bumi Aksara, 2007.

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, 14 Desember 2012

Geraldi Anggapardana / 13511097