

Aplikasi Pohon Prefix pada Pencarian Kontak di Database

Willy / 13512070

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

13512070@std.stei.itb.ac.id

Abstract—Pencarian kontak pada sebuah database merupakan hal yang paling sering dihadapi di setiap aplikasi. Pencarian data di database dengan untaian kata sering memerlukan waktu yang cukup lama. Hal ini dikarenakan banyak data yang dapat dimungkinkan dari untaian kata / info yang diberikan.

I. PENDAHULUAN

Pencarian kontak merupakan salah satu masalah yang sering dihadapi dalam membuat suatu program. Sering kali kita menggunakan mesin pencarian kontak di keseharian kita. Salah satu contohnya, kita mencari kontak seseorang saat kita ingin menelepon orang tersebut dengan telepon genggam kita. Pertama kali yang dilakukan adalah mengetik beberapa bagian dari nama atau kriteria kontak tersebut di mesin pencari. Bila kriteria yang kita berikan dapat memuat di sebagian orang, maka hasil pencariannya akan memunculkan sebagian orang yang memenuhi kriteria tersebut. Pencarian tersebut dapat dilakukan dengan menggunakan pohon prefix.

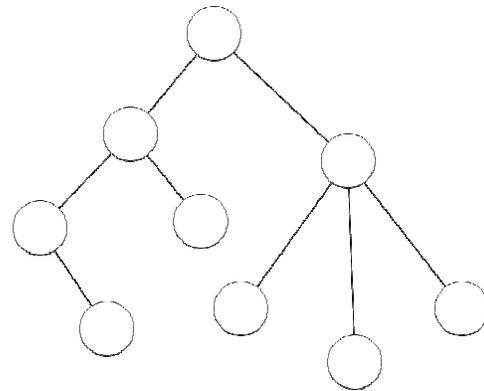
Pencarian yang kita lakukan berdasarkan keyword yang dimasukkan dan kemudian kita mencarinya dalam database. Di dalam database tersebut memuat nama orang dan kriteria orang tersebut.

Pembagian kategori tersebut juga ada prioritas misalnya kategori negara dan kota, lebih diutamakan negaranya daripada kota.

Pembahasan mengenai topik ini menggunakan struktur data pohon dikarenakan pohon dapat memiliki cabang yang tak terhingga terus menerus yang bila setiap simpul atau sisi menandakan suatu huruf/kata. Maka sebuah jalan dari akar menuju suatu simpul data membentuk kata.

II. DASAR TEORI

Suatu graf terhubung yang tidak mempunyai sirkuit disebut dengan pohon. Struktur data pohon telah sering digunakan untuk menyelesaikan beberapa problema di berbagai bidang. Aplikasi pohon pertama kali digunakan pada tahun 1857, seorang matematikawan bernama Arthur Cayley yang mengaplikasikannya untuk menghitung beberapa jenis-jenis struktur kimia.



Contoh pohon

Pohon adalah struktur data yang paling fundamental di bidang komputer. Sering kali informasi disimpan ke dalam struktur data yang mirip dengan pohon. Simulasi rekursif suatu program juga dapat dimodelkan dalam pohon. Pohon biasanya digambarkan dari tingkat atas hingga bawah, dimana tingkat atas disebut dengan akar, dan setiap sisi mengubungkan dengan pohon lain.

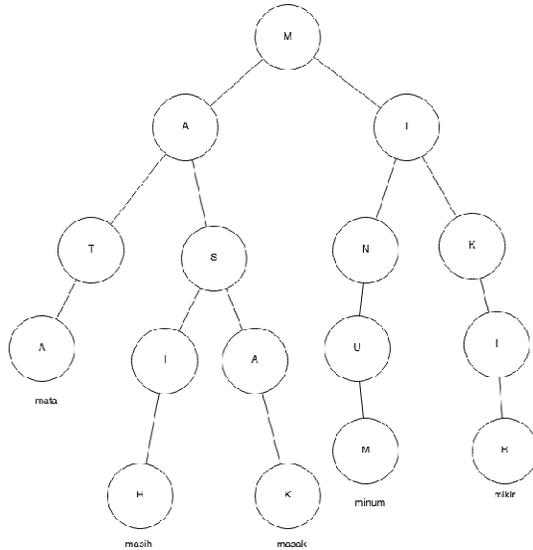
Sifat - sifat dari pohon adalah :

- Segala sesuatu yang terhubung dengan pohon adalah pohon.
- Ada satu jalan yang unik dari setiap simpul ke simpul lainnya.
- Menambahkan sebuah sisi antara 2 simpul di pohon yang tidak berhubungan membentuk suatu siklus
- Membuang sebuah sisi dari pohon tersebut membentuk graf yang tidak terhubung.
- Bila suatu pohon terdapat setidaknya 2 simpul maka setidaknya pohon tersebut terdapat 2 daun
- Jumlah simpul di dalam suatu pohon pasti lebih dari jumlah sisi

Akar dari sebuah pohon adalah sebuah simpul yang sudah ditetapkan sebagai pusatnya dan setiap sisi diarahkan untuk menjauh dari pusatnya. Bila v adalah sebuah simpul di sebuah graf selain dari akar, maka v disebut dengan *parent* dari u dimana ada sisi yang menghubungkan antara u dan v . Bila u adalah *parent* dari v , v disebut anak dari u . Simpul yang memiliki *parent* yang sama disebut dengan *siblings*. Sebuah simpul disebut dengan daun bila tidak mempunyai anak. *Ancesor* dari sebuah simpul selain dari akar adalah

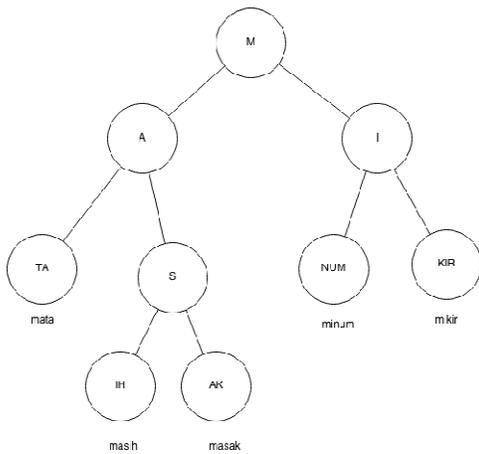
simpul dari jalan antara akar dan simpul tersebut. *Descendant* dari sebuah simpul v adalah simpul-simpul yang mempunyai v sebagai *ancestornya*.

Pohon prefix adalah struktur data yang biasanya menyimpan 2 pasang set yang biasanya labelnya adalah kata. Semua *descendant* dari sebuah simpul mempunyai prefix dari kata yang sama. Akar dari sebuah pohon prefix adalah karakter kosong. Pada kasus ini disebut dengan prefix karena tiap karakter dari yang paling depan dimasukkan ke dalam pohon secara langsung sebagai simpul.



Contoh upapohon prefix

Namun penyimpanan seperti ini dapat membuang memori dikarenakan banyak simpul yang hanya menyimpan sebuah karakter. Pohon Patricia adalah penggabungan simpul - simpul yang hanya terdapat 1 anak.

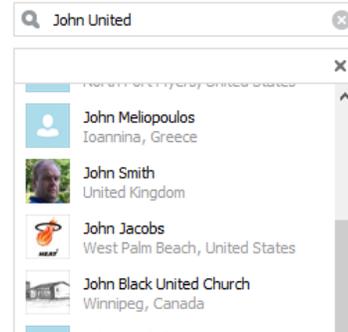


Contoh upapohon Patricia

III. LATAR BELAKANG MASALAH

Pada zaman sekarang, terlalu banyak data yang perlu disimpan dalam suatu jaringan/program yang berbasis sosial. Seperti contohnya mesin pencarian kontak di telepon genggam berdasarkan nama belakang atau nama depan. Pencarian nama ini memerlukan waktu yang lama bila struktur data tidak memadai kepada programmer untuk memberikan algoritma untuk mencari data tersebut.

Salah satu contoh program berbasis sosial yang membutuhkan mesin pencarian yang cepat adalah Skype. Saat kita ingin mencari kontak pada databasenya, dia akan mencari kontak tersebut pada database yang menyimpan semua kontak.



Contoh pencarian "John United" pada program Skype

Pada contoh tersebut, pencariannya dapat berupa beberapa maksud. Salah satunya, dapat dicari kontak yang mempunyai nama John pada nama depan, tengah ataupun belakang dan berwarga negara yang mempunyai kata United ataupun mencari kontak yang mempunyai kata "John" dan "United" dan berbagai bentuk maksud. Pada program skype, query yang dimasukkan digunakan untuk mencari username, nama depan, nama belakang, kota, dan negara. Sehingga bila seorang user tidak mengetahui username dari kontak yang ingin dia cari, dia dapat mencarinya berdasarkan informasi yang dia ketahui mengenai dia. Semakin banyak informasi yang dia berikan di mesin pencari, kemungkinan untuk menemukan orang tersebut semakin besar, sebab semakin sedikit informasi yang diberi, semakin banyak orang yang memenuhi kriteria yang diberikan.

IV. DASAR DARI MESIN PENCARIAN

Dalam melakukan pencarian kontak, biasanya terdapat tabel yang memuat data-data disebut dengan database. Database biasanya hanya menyimpan data-data yang diperlukan oleh program atau jaringannya. Pencarian biasanya dilakukan terhadap beberapa kategori dari data tersebut saja. Hal ini disebabkan karena ada data privasi yang tidak boleh diketahui oleh orang lain kecuali user dan pembuat program sendiri. Salah satu contohnya adalah password, sebuah password tidak pernah dipublikasikan atau bisa dicari oleh user lain karena privasi antara user dan pembuat program sendiri.

No	Nama	Kriteria A	Kriteria B	...
1	John	Apel	Indonesia	...
2	Martin	Mangga	Malaysia	...
3	Sam	Jeruk	Singapore	...
...

Tabel Database

Untuk pemisalan, database pada tabel di atas terdapat n user yang berarti banyaknya user dan C buah kriteria. Kita juga misalkan kita diberikan k kata, sebanyak n buah yang berbeda-beda. Dimana tujuan pemberian kata adalah mencari kontak yang cukup sama dengan kriteria pada database tersebut.

Misalkan :

Pencarian : Jo Ap

Hasil Pencarian : John, Indonesia

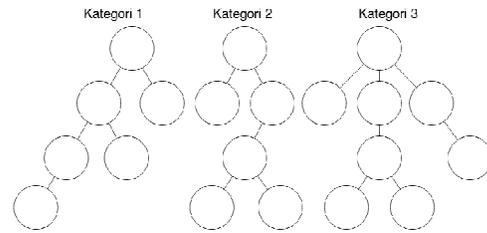
Berdasarkan contoh, hasil pencarian hanya diberikan beberapa kriteria yang boleh diberi tahu oleh publik. Meski kata "Ap" yang memuat Apel pada data John, tidak diberikan secara publik karena privasi tersendiri dari program/user.

V. MESIN PENCARIAN TANPA POHON PREFIX

Karena urutan dari kata yang diberikan tidak sesuai dengan tabel dan banyak kata bukanlah berarti banyak kriteria yang harus dipenuhi seperti contohnya 2 kata seperti "John Smith" memuat satu kriteria saja. Karena itu dimungkinkan n buah kata memenuhi 1 sampai n buah kriteria sebuah user. Sebuah kriteria perlu dicocokkan dengan sebuah kata, diperlukan pencocokan dengan tiap hurufnya. Waktu yang diperlukan untuk mencocokkan setiap kata terhadap tabel adalah :

VI. MESIN PENCARIAN DENGAN POHON PREFIX

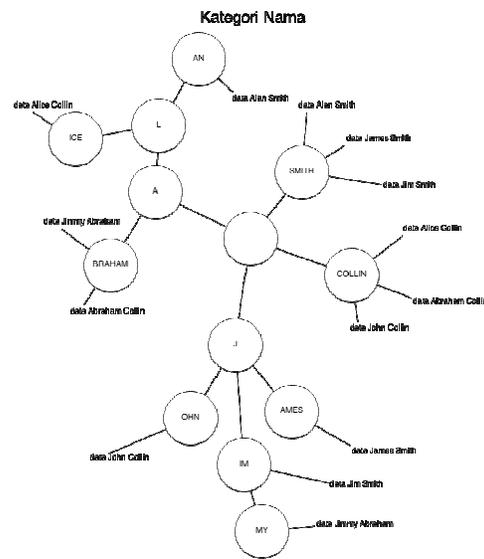
Yang dimaksud dengan pencarian dengan pohon prefix adalah setiap kata sudah dimasukkan ke dalam kategori yang bersangkutan. Karena itu terdapat C buah pohon yang menyimpan kata-kata bersangkutan. Kemudian setiap daun dari pohon tersebut menunjuk ke tabel yang bersangkutan. Misalkan, terdapat data : John, Amerika, Washington DC. Maka "John" dimasukkan ke dalam tree yang berkategori nama. "Amerika" ke kategori negara. "Washington DC" ke kategori kota. Bila terdapat dari 2 kata yang mempunyai kategori yang sama, maka simpul tersebut menunjuk ke 2 data.



Contoh database dengan 3 kategori

Setiap daun dari pohon kategori tersebut menunjuk ke data di tabel tersebut.

Bila setiap simpul digunakan untuk menyimpan satu huruf saja, terlalu memakan ruang memori, karena itu kadang pohon prefix dapat diubah ke pohon patricia (yang sering disebut dengan pohon prefix juga karena konsepnya tidak jauh beda). Setiap simpul yang mempunyai 1 anak saja dapat digabung.



Contoh pohon prefix

Algoritma untuk pencarian data

Saat ingin mencari apakah sebuah kata terdapat pada kategori tersebut, hanya diperlukan mencari dari akar pohon kategori tersebut. Huruf demi huruf dicari menulsi dari akar, bila tidak ada huruf/kata yang terdapat pada anak dari simpul yang sedang kita telusuri, maka kata tersebut tidak terdapat pada kategori tersebut. Bila kata yang dicari sudah ditelusuri selesai, berarti hasil dari pencarian adalah penelusuran dari simpul tersebut hingga ke daun.

Hasil dari pencarian adalah data yang ditemukan oleh setiap kata pada semua kriteria yaitu minimum banyak kata.

Bila terdapat lebih dari 1 kata yang dicari, maka setiap kata di tes cari di setiap kategori. Semua hasil yang ditemukan lebih dari n kali, dimana n adalah banyaknya kata yang ingin dicari, disebut sebagai salah satu dari solusi dari pencarian tersebut.

Pada saat menambah suatu kata ke dalam kategori juga hanya diperlukan penelusuran sebuah pohon kategori sebanyak panjang katanya saja.

Contohnya adalah apabila user memasukkan "J" ke dalam mesin pencari. Hasil yang didapatkan dari menelusuri pohon kategori nama adalah data John Collin, James Smith, Jim Smith, Jimmy Abraham. Hal ini disebabkan dari akar pohon, kemudian menuju simpul yang kuncinya bernilai J. Karena query dari user telah dicari selesai, maka dari simpul tersebut, pencarian dilakukan ke semua arah anak. Contoh lainnya adalah apabila user memasukkan "J C" ke dalam mesin pencari. Hasil yang didapatkan dari pencarian adalah data John Collin saja. Hal ini didapatkan karena saat menelusuri kata "J", jalan yang ditempuhnya adalah menuju simpul J, kemudian menelusuri semua *descendent* dari J, karena kata "J" sudah ditelusuri sampai ujung. Hasil dari menggunakan kata J saja didapatkan Jim Smith, John Collin, Jimmy Abraham, James Smith. Kemudian melakukan pencarian dengan kata "C" pada kategori nama, maka dimulai dari akar pohon tersebut. Pada pencarian "C" dari akar langsung menuju simpul "COLLIN" karena hanya simpul tersebut yang memiliki prefix yang sama. Maka didapatkan hasil data Alice Collin, Abraham Collin, John Collin dari kata "C". Bila dihitung frekuensi ketemu data tersebut dapat dipresentasikan dalam tabel.

Data	Frekuensi
Alice Collin	1
Abraham Collin	1
John Collin	2
Jim Smith	1
Jimmy Abraham	1
James Smith	1

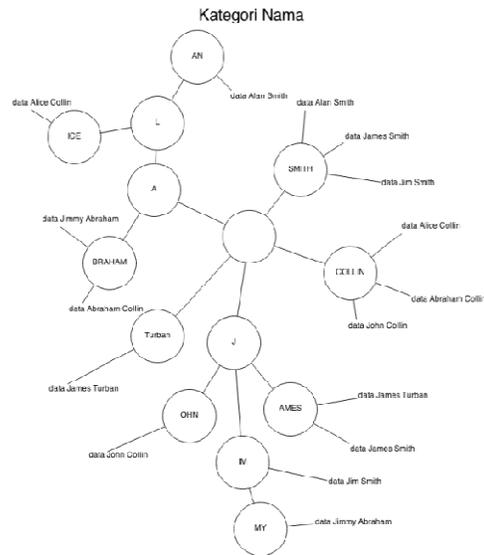
Maka hasil dari pencarian dari "J C" adalah data John Collin, karena hanya data John Collin yang ditemukan pada setiap pencarian kata. Data dapat disebut solusi dari pencarian saat jumlah frekuensi lebih besar sama dengan jumlah kata. Hal ini disebabkan, setiap solusi data tersebut harus sudah ditemukan minimum oleh semua kata dalam setiap kategori.

Waktu yang diperlukan untuk mencari adalah

Bila waktu ini dibandingkan dengan pencarian tanpa prefix tree sangatlah jauh, hal ini disebabkan kita tidak mengetes permutasi setiap kategori dengan setiap kata.

Algoritma untuk penambahan data

Ditelusuri dari akar suatu pohon kategori yang ingin ditambah, bila sudah ada simpul yang huruf/katanya sesuai dengan data yang ingin dimasukkan maka ditelusuri menuju huruf/kata berikutnya. Bila belum ada huruf/kata di simpul, maka dibentuk simpul pada pohon tersebut.



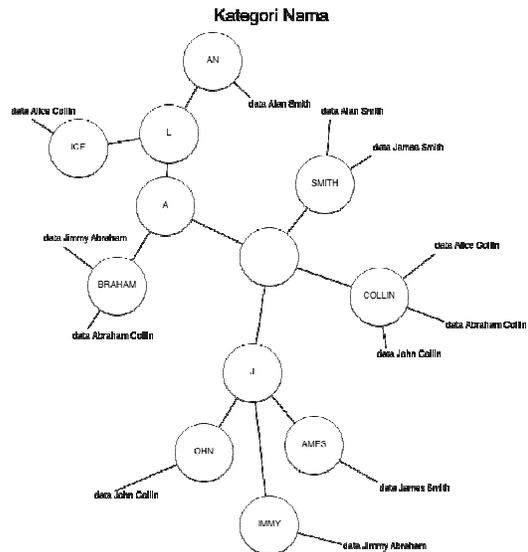
Penambahan data James Turban pada pohon kategori Nama

Pada saat menghapus suatu data dari database, maka pohon tiap kategori juga perlu ada yang dihapus.

Algoritma untuk pengurangan data

Ditelusuri setiap pohon, sesuai dengan data yang ingin dihapus dari kategori tersebut. Setiap simpul yang hanya berhubungan dengan kata tersebut dihapus juga, kecuali simpul yang berhubungan dengan kata lain.

Misalkan dikurangkan sebuah data Jim Smith pada prefix tersebut.



Pengurangan data Jim Smith pada pohon kategori Nama

V. CONCLUSION

Karena itu bila pencarian data pada sebuah database

menggunakan pohon prefix lebih menghemat waktu daripada mencari data pada tabel secara satu per satu .

VIII. ACKNOWLEDGMENT

Puji syukur kepada Tuhan Yang Maha Esa karena penulis dapat menyelesaikan tugas makalah "Matematika Diskrit" yang diberikan pada akhir semester 3. Penulis mengucapkan terima kasih kepada dosen mata kuliah Matematika Diskrit yaitu Ibu Harlili dan Pak Rinaldi Munir.

REFERENCES

- [1] Kenneth H. Rosen, Discrete Mathematics and Its Application, Mc Graw Hill, 1999.
- [2] Halim, Steven, Competitive Programming 3, Steven & Felix Halim, 2013.
- [3] Eric Lehman, F Tom Leighton, Albert R Meyer, Mathematics for Computer Science, MIT Handbook, 2010.
- [4] <http://en.wikipedia.org/wiki/Trie>
diakses pada : 14 Desember 2013 pafa pukul 12:00

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, 15 November 2013



Willy 13512070