Quad Tree dan Contoh-Contoh Penerapannya

Muhammad Reza Mandala Putra - 13509003
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung 40132, Indonesia
13509003@std.stei.itb.ac.id

***Sari*—Pohon merupakan salah satu terapan graf yang konsepnya banyak dimanfaatkan dalam memecahkan berbagai permasalahan. Dalam kehidpan sehari-hari orang telah lama menggunakan pohon untuk menggambarkan hirarki. Beberapa contohnya adalah pohon silsilah keluarga, struktur organisasi, dan organisasi pertandingan. Pohon juga ada banyak jenisnya, seperti pohon biner (pohon yang setiap cabangnya memiliki maksimum 2 simpul / *node*), pohon *m-ary* (memiliki jumlah simpul maksimum sebanyak *m* untuk setiap cabangnya), pohon keputusan, dan lain sebagainya. Salah satu contoh dari pohon *m-ary* adalah *Quad Tree*. Pada makalah ini akan dibahas mengenai *Quad Tree* dan contoh-contoh penerapannya.**

***Kata Kunci*—Graf, Pohon, *Quad Tree,* Rekursif.**

I. PENDAHULUAN

Pohon merupakan salah satu terapan dari teori graf. Konsep pohon banyak digunakan untuk memecahkan berbagai permasalahan. Konsep pohon sering diterapkan dalam bidang komputasi. Misalnya adalah pohon diterapkan sebagai representasi dari *file system*. Contoh berikutnya, konsep pohon digunakan dalam pembuatan beberapa algoritma, seperti algoritma BFS, DFS, *Branch and Bound*, dan lain sebagainya. Selain itu konsep pohon juga diterapkan dalam pembuatan aplikasi permainan, baik aplikasi permainan dua dimensi maupun tiga dimensi.

Salah satu contoh pohon adalah *Quad Tree*. *Quad Tree* adalah pohon yang memiliki 4 simpul. Simpul tersebut dapat berupa simpul anak, daun, atau kombinasi dari keduanya. Sebelum membahas lebih jauh mengenai pohon *Quad Tree*, alangkah baiknya apabila kita mengenal dulu teori mengenai pembentukannya, yaitu teori graf dan pohon.

II. GRAF

A. Pengertian Graf

Graf adalah suatu struktur diskrit yang terdiri dari *vertex* (simpul) dan sisi, dimana sisi menghubungkan *vertex-vertex* yang ada. Secara matematis, graf didefinisikan sebagai berikut:

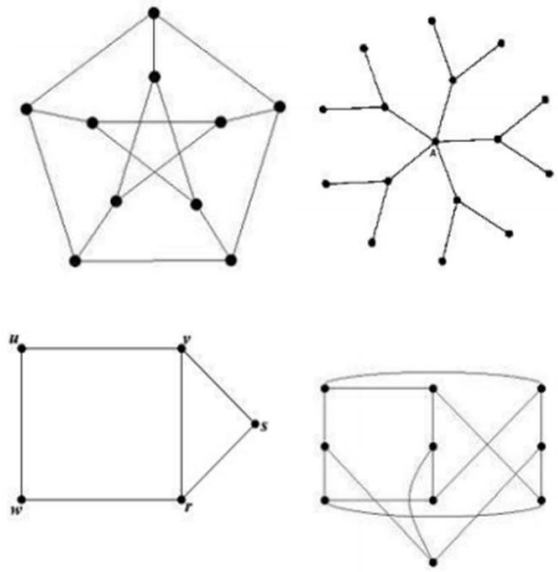
Graf G didefinisikan sebagai pasangan himpunan (V,E) yang dalam hal ini:

V = himpunan tidak-kosong dari simpul-simpul (vertices atau node) = {v1,v2,...,vn} dan

E = himpunan sisi (edges atau arcs) yang menghubungkan sepasang simpul = {e1,e2,...,en}

Atau dapat ditulis singkat notasi G = (V,E), dengan V tidak boleh kosong, sedangkan E boleh kosong.

Jadi, sebuah graf dimungkinkan tidak mempunyai sisi satu buah pun, tetapi simpulnya harus ada, minimal satu. Secara geometri, graf digambarkan sebagai sekumpulan noktah (simpul) di dalam bidang dwimatra yang dihubungkan dengan sekumpulan garis (sisi).



Gambar 1. Contoh-contoh graf

B. Terminologi Dasar

Beberapa terminologi dasar yang sering dipakai

1. Bertetangga
   Dua buah simpul pada graf tak berarah G dikatakan bertetangga bila keduanya terhubung langsung dengan sebuah sisi.
2. Bersisian
   Untuk sembarang sisi e = (u,v), sisi e dikatakan bersisian dengan simpul u dan simpul v.
3. Simpul terpencil
   Simpul terpencil ialah simpul yang tidak mempunyai sisi yang bersisian dengannya. Atau, dapat juga simpul terpencil adalah simpul yang tidak satupun bertetangga dengan simpul-simpul lainnya.

4. Graf kosong

   Graf kosong adalah graf yang himpunan sisinya merupakan himpunan kosong.
5. Derajat

   Derajat suatu simpul pada graf tak berarah adalah jumlah sisi yang bersisian dengan simpul tersebut.
6. Siklus atau sirkuit

   Siklus atau sirkuit adalah lintasan yang berawal dan berakhir pada simpul yang sama.
7. Terhubung

   Graf tak berarah G disebut graf terhubung jika untuk setiap pasang simpul u dan v di dalam himpunan V terdapat lintasan dari u ke v.
8. Upagraf

   Misalkan G = (V,E) adalah sebuah graf. G1 = (V1,E1) adalah upagraf dari G jika V1 merupakan bagian dari V dan E1 merupakan bagian dari E.
9. Graf berbobot

   Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga (bobot).

C. Jenis-Jenis Graf

Graf dapat dikelompokkan menjadi beberapa kategori bergantung pada sudut pandang pengelompokannya. Pengelompokan graf dapat dipandang berdasarkan ada tidaknya sisi ganda atau sisi kalang, berdasarkan jumlah simpul, atau berdasarkan orientasi arah pada sisi.

Berdasarkan orientasi arah pada sisi, maka secara umum graf dibedakan atas 2 jenis:

1. Graf tak berarah

   Graf tak berarah adalah graf yang sisinya tidak mempunyai orientasi arah. Pada graf tak berarah, urutan pasangan simpul yang dihubungkan oleh sisi tidak diperhatikan. Jadi $(v_j,v_k) = (v_k,v_j)$ adalah sisi yang sama.
2. Graf berarah

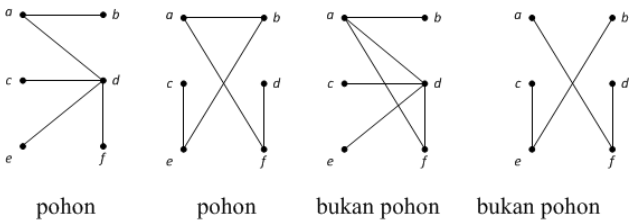
   Graf berarah adalah graf yang setiap sisinya diberikan orientasi arah. Jadi $(v_j,v_k) = (v_k,v_j)$ adalah sisi yang berbeda.

III. POHON

A. Pengertian Pohon

Pohon adalah suatu graf tak berarah terhubung yang tidak mengandung sirkuit. Karena definisi pohon tersebut diacu dari teori graf, maka sebuah pohon dapat mempunyai sebuah simpul tanpa sebuah sisipun. Dengan kata lain, jika G = (V,E) merupakan sebuah pohon, maka V tidak boleh berupa himpunan kosong, tapi E boleh merupakan himpunan kosong.

Berdasarkan definisi tersebut, ada dua sifat penting pada pohon, yaitu terhubung dan tidak mengandung sirkuit. Yang dimaksud dengan terhubung adalah pada setiap pasang simpul pada pohon terdapat lintasan yang menghubungkannya. Sedangkan yang dimaksud tidak mengandung sirkuit berarti tidak terdapat lebih dari satu lintasan yang menghubungkan setiap pasang simpul pada pohon. Gambar berikut adalah gambar mengenai perbedaan pohon dan bukan pohon.



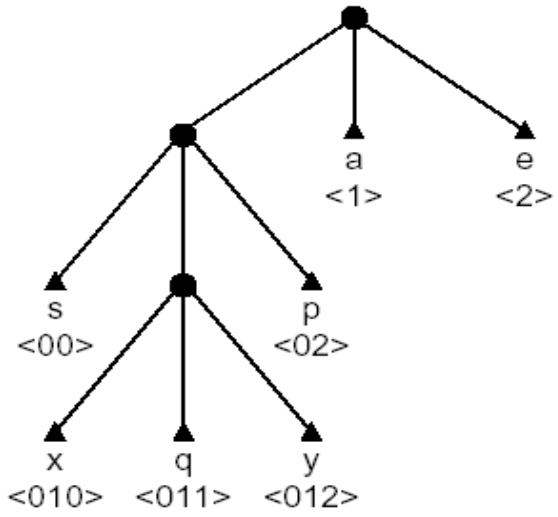
Gambar 2. Perbedaan pohon dan bukan pohon

B. Jenis-Jenis Pohon

Sama seperti graf, pohon juga mempunyai beberapa jenis. Jenis-jenis pohon antara lain pohon berakar, pohon keputusan, pohon biner (*binary tree*) dan pohon *m-ary*. Di antara jenis-jenis pohon yang ada, yang akan dibahas adalah pohon berakar dan pohon *m-ary*. Pohon berakar adalah pohon yang mana sebuah simpulnya diperlakukan sebagai akar dan sisi-sisinya diberi arah menjauh dari akar. Ada beberapa terminologi yang perlu diketahui pada pohon berakar, yaitu terminologi anak dan orangtua.

Suatu simpul *a* dikatakan sebagai anak dari simpul *b* jika ada sisi yang berarah dari simpul *b* ke simpul *a*. Sedangkan pohon *m-ary* adalah pohon berakar yang setiap simpul cabangnya mempunyai paling banyak *m* buah anak.

Salah satu contoh penggunaan pohon *m-ary* adalah pohon *ternary Huffman*. Pohon ini digunakan sebagai struktur data untuk melakukan kompresi dan dekompresi data pada sebuah *file*. Berikut adalah gambar dari pohon *ternary Huffman* tersebut.

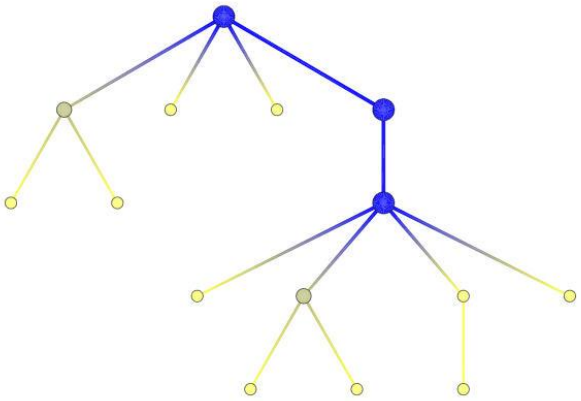


Gambar 3. Pohon ternary Huffman

IV. *QUAD TREE*

Salah satu contoh dari pohon *m-ary* adalah *Quad Tree*. *Quad Tree* adalah pohon *m-ary* yang setiap simpulnya tepat memiliki 4 cabang anak. Cabang-cabang pada *Quad*

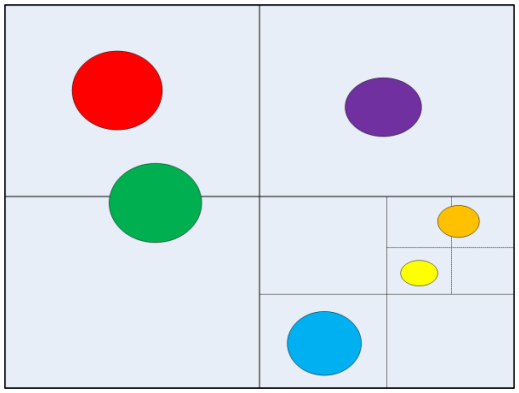
Tree dapat berupa simpul yang memiliki cabang lagi sebanyak 4 cabang. Cabang-cabang *Quad Tree* juga dapat langsung berupa daun atau kombinasi antara cabang dan daun. Daun-daun pada *Quad Tree* biasanya berisi informasi mengenai sesuatu. Penggambaran pohonnya adalah sebagai berikut.



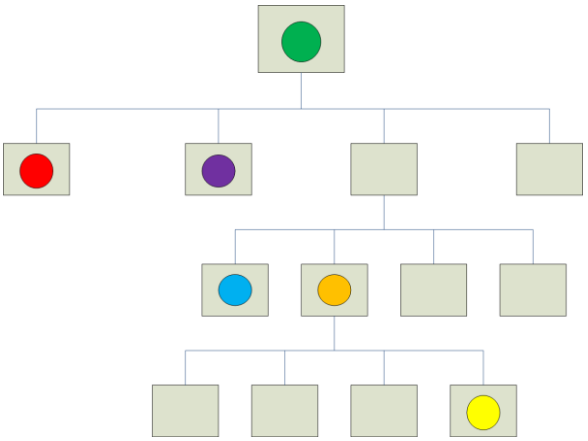
Gambar 4. Penggambaran Quad Tree

V. CONTOH PENGGUNAAN *QUAD TREE*

Quad Tree banyak dijumpai contoh-contoh penerapannya. *Quad Tree* paling sering diterapkan untuk pembuatan aplikasi permainan. Salah satu contoh penerapan *Quad Tree* pada aplikasi permainan adalah tentang pengecekan *collision* (benturan / tabrakan) dua objek yang berbeda pada arena permainan dua dimensi secara efisien (untuk pengecekan *collision* pada arena permainan tiga dimensi digunakan *Oct-Tree*, pohon dengan jumlah maksimum 8 cabang pada setiap simpulnya). Cara melakukan pengecekannya adalah dengan membagi arena permainan menjadi empat bagian yang berukuran sama. Apabila dua objek atau lebih berada pada satu wilayah yang sama, bagi lagi wilayah tersebut menjadi empat bagian yang berukuran sama. Pembagian wilayah terus dilakukan hingga masing-masing tidak ada objek yang berada pada wilayah yang sama. Pembagian wilayah ini dilakukan secara rekursif. Berikut ini adalah contoh pembagiannya.

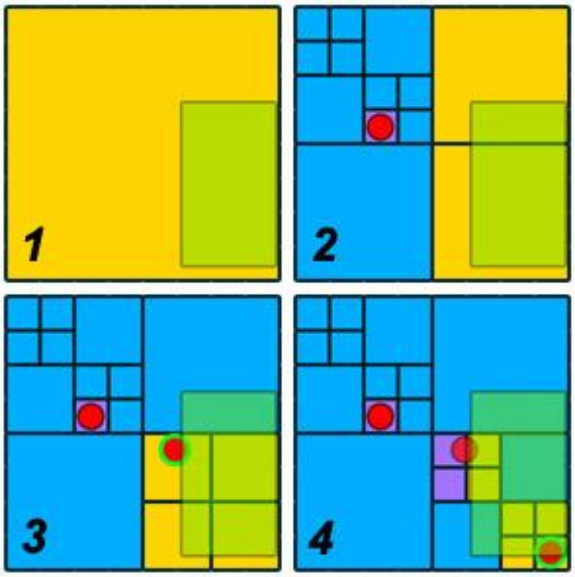


Gambar 5. Pembagian wilayah objek-objek pada arena permainan



Gambar 6. Pemetaan wilayah objek-objek ke dalam Quad Tree

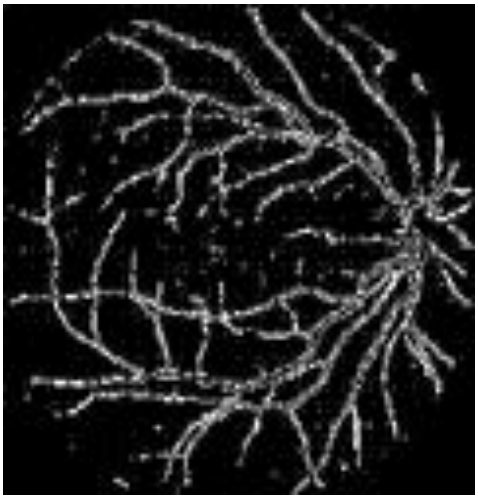
Setelah dilakukan pemetaan objek-objek ke dalam *Quad Tree*, selanjutnya akan dilakukan pendeteksian terhadap objek-objek yang mengalami *collision.* Berikut ini adalah ilustrasi dari pendeteksian objek-objek yang mengalami *collision*.



Gambar 7. Pendeteksian objek-objek yang bertabrakan dengan kotak berwarna hijau

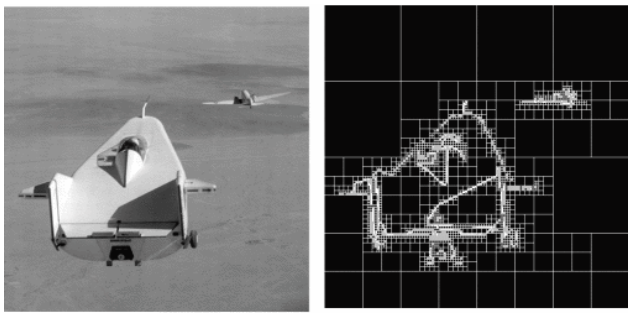
Selain pada aplikasi permainan, *Quad Tree* juga diterapkan dalam dunia medis. Salah satu contoh penerapannya adalah pendeteksian pembuluh darah pada retina. Cara pengambaran *Quad Tree*-nya adalah dengan mengambil foto dari retina yang akan diperiksa. Kemudian foto retina tersebut dilakukan dekomposisi dengan menggunakan *Quad Tree* sehingga didapatkan gambar pembuluh darah yang akan digunakan untuk analisis selanjutnya. Foto hasil dekomposisi tersebut berisi informasi mengenai berbagai jenis blok dan intensitas dari *pixel* dalam blok.

Gambar 8. Foto retina untuk pendeteksian pembuluh darah



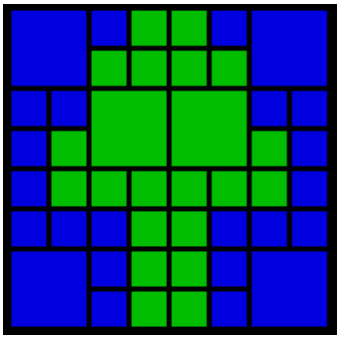
Gambar 9. Foto retina setelah dilakukan dekomposisi

Pohon *Quad Tree* juga diterapkan pada bidang grafis. Salah satu contohnya adalah pada pengolahan citra (*image processing*). Pada pengolahan citra, Sebuah gambar didefinisikan sebagai *array* dua dimensi yang mana *array* tersebut berisi informasi-informasi mengenai pewarnaan gambar. Setiap warna memiliki nomor yang berbeda-beda. Ukuran gambar dinyatakan dalam satuan *pixel* dan setiap *pixel*-nya mewakili satu warna. Berikut adalah contoh penggunan *Quad Tree* dalam *image processing*.

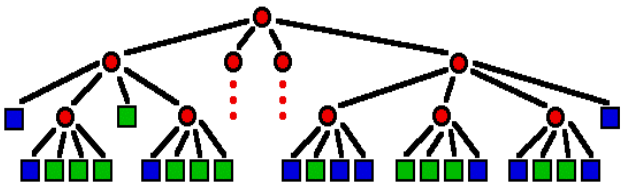


Gambar 10. Gambar dan hasil dekomposisnya dengan menggunakan Quad Tree

Untuk melakukan penyimpanan informasi mengenai pewarnaan gambar secara cepat dan hemat *memory* diperlukan suatu struktur data dan algoritma yang tepat. Salah satu struktur data yang digunakan adalah struktur data *Quad Tree* dan algoritman yang digunakan adalah algoritma *Divide and Conquer*. Cara penerapan algoritma dan struktur data tersebut adalah dengan cara membagi gambar menjadi empat bagian yang berukuran sama. Kemudian dilakukan pengecekan terhadap warna-warna pada suatu bagian. Apabila didapat warna-warna yang berbeda pada suatu bagian, bagi lagi bagian tersebut menjadi empat bagian berukuran sama. Pembagian tersebut dilakukan secara rekursif hingga tidak ada lagi beberapa warna yang berbeda pada suatu bagian. Berikut ini adalah contoh hasil pembagian dan representasi warna pada *Quad Tree.*



Gambar 11. Pembagian wilayah warna pada suatu gambar



Gambar 12. Penyimpanan informasi warna suatu gambar dalam struktur data Quad Tree

VII. KESIMPULAN

Dari pembahasan mengenai *Quad Tree* di atas dapat ditarik kesimpulan mengenai beberapa hal sebagai berikut:

1. Salah satu contoh dari pohon *m-ary* adalah Quad Tree, dengan $m = 4$.
2. *Quad Tree* dapat diterapkan di berbagai bidang, seperti pada pembuatan aplikasi permainan, pada dunia medis, bidang grafis, dan lain sebagainya.
3. *Quad Tree* pada umumnya menggunakan algoritma *Divide and Conquer* secara rekursif untuk melakukan *generate* pada simpul dan daun-daunnya, yaitu dengan cara membagi simpul menjadi 4 cabang.
4. Daun-daun pada Quad Tree berisi informasi mengenai suatu objek yang disimpan.

REFERENSI

[1] Munir, Rinaldi. *Matematika Diskrit Revisi Keempat.* Bandung : INFORMATIKA. 2010. halaman 353 – 490

[1] http://acm.tju.edu.cn/toj/showp1497.html
tanggal akses 11 Desember 2011

[2] http://gravite.labri.fr/?Want_to_work_with_us_%3F:Hiring_puzzles:Tidy_Tree_Layouts
tanggal akses 12 Desember 2011

[3] http://www.cs.ubc.ca/~pcarbo/cs251/welcome.html
tanggal akses 11 Desember 2011

[4] http://www.codeproject.com/KB/recipes/QuadTree.aspx
tanggal akses 12 Desember 2011

[5] http://www.kyleschouviller.com/wsuxna/quadtree-source-included/
tanggal akses 11 Desember 2011

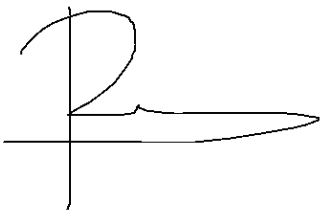
[6] http://www.mathworks.com/help/toolbox/images/ref/qtdecomp.html
tanggal akses 11 Desember 2011

[7] http://www.springerlink.com/content/9388076688260565/
tanggal akses 11 Desember 2011

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, 12 Desember 2011



Muhammad Reza Mandala Putra - 13509003