

ANALISIS TERHADAP SKYLINE *QUERY* DAN TOP-K *QUERY* PADA CONTEXT PREFERENCE AWARE SERVICE

Ecko Fernando Manalu

Program Studi Teknik Informatika
Institut Teknologi Bandung
Jl. Ganesha 10 Bandung
e-mail: ecko_mnl@yahoo.com

ABSTRAK

CONTEXT PREFERENCE AWARE (CPA) Service adalah suatu jenis sistem informasi geografis yang memberikan suatu layanan informasi tepat guna dan relevan kepada pengguna karena tanggap (*aware*) terhadap konteks dan preferensi pengguna. CPA menggunakan arsitektur basisdata yang khusus, serta strategi algoritma khusus dalam hal memfiltrasi hasil querynya agar dapat memberi hasil yang baik kepada pengguna. Dua strategi algoritma tersebut yang berbasiskan pada Greedy Algorithm adalah skyline dan top-k. Keduanya menjadi strategi yang sangat andal dalam CPA. Dalam tulisan ini akan diungkapkan secara lebih dekat mengenai apa yang menjadikan kedua strategi algoritma ini menjadi sangat baik.

Kata kunci: Context Preference Aware, Sistem Informasi Geografis, top-k, skyline, greedy.

1. PENDAHULUAN

Context Preference Aware Service (CPA) dalam bahasa Indonesia diartikan sebagai sebuah layanan yang merespon (*aware*) atau tanggap akan konteks dan preferensi pengguna layanan tersebut. CPA merupakan suatu layanan yang merupakan pengembangan atau lebih tepatnya jawaban akan kekurangan yang dimiliki oleh layanan berbasis lokasi (*Location Based Service / LBS*). Jika LBS hanya akan memberikan layanan informasi seputar lokasi dimana pengguna berada, maka CPA lebih jauh lagi melakukan penyaringan terhadap informasi apa saja yang relevan kepada pengguna dengan memperhitungkan lokasi, preferensi, dan konteks pengguna saat ini. Keduanya merupakan jenis layanan yang berbasis atas Sistem Informasi Geografis sebagai intinya. Dalam artian bahwa layanan ini sangat terkait dengan data spasial (geografis) pengguna, terutamanya adalah lokasi lintang dan bujur (koordinat bumi), yang dapat ditentukan melalui suatu alat yang disebut GPS (*Global Positioning System*) atau dengan metode

penghitungan Base Transmission Station (BTS) yang menandakan bahwa layanan ini dapat diakses dengan menggunakan suatu perangkat genggam (*mobile device*)

Layanan CPA lebih jelasnya dapat dilihat pada contoh kasus berikut. Seorang wisatawan yang belum begitu mengenal daerah Bandung ingin agar dia dapat melakukan kegiatan wisatanya dengan aman dan lancar, tanpa ada rasa takut akan kehilangan arah, atau tersesat, atau kurang puas dengan wisatanya karena tempat yang dia kunjungi kurang relevan akibat kekurangan informasi. Untuk itu dia melakukan penjelajahan di Internet untuk melihat lokasi-lokasi menarik di Bandung, lengkap dengan harga yang harus dibayar untuk bisa ke lokasi tersebut, bagaimana cara untuk bisa ke lokasi tersebut. Namun hal tersebut tidak mudah dilakukan, dan membutuhkan waktu yang lama, karena dia harus melakukan lagi penghitungan akan biaya, waktu, jarak yang ditempuh, menyimpan navigasi untuk mencapai lokasi yang dia inginkan.

Kesulitan yang dialami tersebut sebenarnya dapat diatasi dengan menggunakan CPA, karena dengan CPA, wisatawan tidak lagi perlu untuk melakukan perencanaan yang rumit seperti di atas. Karena dengan hanya menggunakan suatu perangkat genggam yang memadai, informasi penting dan relevan berdasarkan lokasi dimana wisatawan berada, dan berdasarkan konteks dan preferensi pengguna (seperti lokasi, umur, ketinggian tempat, hidangan kesukaan, lokasi yang disukai, dan preferensi lainnya) akan dapat diperoleh kapan saja dimana saja, dengan asumsi bahwa semua lokasi pada daerah tersebut telah terdapat di dalam basisdata sistem CPA.

2. ARSITEKTUR CPA

Di atas telah disinggung dengan apa yang disebut dengan basisdata CPA. CPA memang menggunakan basisdata untuk menyimpan semua data yang diperlukan untuk dapat memberi informasi yang tepat dan relevan kepada pengguna, layaknya sistem informasi lainnya. Secara garis besar basis data dibagi atas kelompok berikut :

1. Basisdata yang menyimpan konteks dan preferensi pengguna.
2. Basisdata yang menyimpan lokasi-lokasi dan informasi detilnya.
3. Basisdata yang menyimpan informasi akan lingkungan sekitar pengguna saat ini.

Untuk contoh kasus di atas, pada suatu saat wisatawan ingin mencari lokasi kuliner, dimana lokasi itu tidak hanya dekat, tapi juga yang sesuai dengan dietnya, sesuai dengan menu kesukaannya, sesuai dengan budget yang dia sanggupi, kondisi lingkungannya nyaman, jalur ke tempat tersebut tidak sedang macet. Dapat dibayangkan akan betapa sulitnya dia melakukan hal tersebut apabila hanya memanfaatkan jasa layanan peta, atau situs pariwisata bandung, dimana hal tersebut juga akan memboroskan waktu sehingga mengganggu wisatanya. Dengan penggunaan CPA, maka dia cukup menyampaikan 'query' sehubungan dengan lokasi kuliner kepada sistem. Maka dengan memanfaatkan konteks dan preferensi pengguna yang sebelumnya telah dientri oleh pengguna (umur, diet, budget), sistem CPA akan melakukan pencarian informasi dengan mencocokkan konteks dan preferensi pengguna tersebut dengan lokasi-lokasi yang sesuai yang terdapat pada basisdata lokasinya, dan selanjutnya akan mencocokkan juga dengan kondisi lingkungan (jalanan tidak macet, kondisi lingkungan sejuk dan nyaman) sekitar dari lokasi yang telah didapatkan.

2. DETIL SISTEM CPA

Dari pemaparan di atas, secara garis besar dapat dilihat bahwa yang menjadi kunci dalam suksesnya penyampaian informasi yang relevan kepada pengguna adalah proses penyimpanan di basisdata, serta bagaimana melakukan pemrosesan *query* dari pengguna. Pada tulisan ini akan dibahas mengenai aspek bagaimana pemrosesan *query* pengguna dilakukan. Dalam tulisan ini akan diperkenalkan mengenai dua strategi *query* yang juga menjadi strategi algoritma untuk penyelesaian masalah penyampaian informasi yang relevan.

2.1 Query

Dalam hal transaksi basis data, dikatakan juga bahwa *query* adalah proses dimana dilakukan permintaan akan suatu informasi spesifik dari basisdata. Dalam bidang basisdata dikenal suatu istilah yang disebut dengan SQL (Structured *Query* Language) yang menjadi standar dalam pengaksesan basisdata.

Pada CPA, *query* yang dilakukan adalah yang bersifat multi-objektif. Misalnya bagaimana cara mencari daftar

tempat yang paling dekat, paling murah, paling nyaman, paling bagus pelayanannya. Hal tersebut dilakukan dengan memperkenalkan teknik *query* multi-objektif skyline dan top-k.

2.2 Skyline Query

Skyline *Query* adalah suatu strategi untuk memfilter hasil *query* dengan hanya mengambil hasil *query* yang memiliki nilai tertinggi dari setiap kriteria. Misalnya pada suatu pemrosesan awal *query* didapati beberapa tempat yang dekat dengan lokasi pengguna saat ini, maka dari beberapa tempat ini hanya akan diambil tempat yang memiliki nilai tertinggi, misalnya yang paling murah, yang paling dekat, paling nyaman, paling bagus pelayanannya. Tidak menutup kemungkinan bahwa satu tempat bisa saja menjadi yang paling murah sekaligus paling dekat.

Contoh:

Table 1 Contoh Lokasi

No	Nama Tempat	Jarak	Harga Rata-rata	Rating Pelayanan
1.	Dago Pakar	100m	50.000 IDR	4 dari 5
2.	Sapu Lidi	200m	40.000 IDR	3 dari 5
3.	Bancakan	400m	60.000 IDR	2 dari 5
4.	Kafemali	300m	20.000 IDR	2 dari 5
5.	Kantin Borju	5m	40.000 IDR	2 dari 5

Dari ke-5 tempat di atas, jika berdasarkan kriteria yang diminta maka yang akan ditampilkan sebagai hasil adalah

1. Dago Pakar (Rating Pelayanan Tertinggi)
2. Kafemali (Harga Termurah)
3. Kantin Borju (Jarak Terdekat)

Sementara itu 2 hasil *query* lainnya akan dibuang atau bisa saja dijadikan kandidat sekunder.

2.3 Top-K Query

TOP-K *Query* adalah suatu strategi untuk menjawab multi objektif *query* yang perilakunya dalam memroses hasil *query* berbeda dengan Skyline. Jika Skyline mengambil hasil *query* dengan memilah berdasarkan nilai tertinggi dari masing-masing kriteria yang dimiliki oleh hasil *query* (hasil tempat) maka top-k melakukan fungsi agregasi, bukan melakukan pendekatan satu-persatu seperti yang terdapat pada skyline.

Untuk tabel di atas, maka top-k *query* akan melakukan filtrasi penyaringan hasil *query* dengan cara melakukan

fungsi agregasi seperti jumlah terkecil dari masing-masing kriteria, atau rata-rata dari setiap nilai kriteria. Misalnya jika dianggap 3 jumlah nilai masing-masing kriteria terkecil (Jarak + Harga + Rating) sebagai fungsi penyaring, maka secara kasar akan didapati hasil serta urutan sebagai berikut :

1. Kafemali. Total Skor (300 + 20.000 + 2) = 20.302
2. Kantin Borju (5 + 40.000 + 2) = 40.007
3. Sapu Lidi (200+ 40.000 + 2) = 40.202

Dua hasil *query* lainnya tidak akan disertakan dalam hasil karena yang dibutuhkan adalah top-3 dari hasil *query* dengan fungsi agregasi adalah penjumlahan minimum dari semua nilai kolom (kriteria).

3. APLIKASI QUERY PADA CPA

Aplikasi dari kedua strategi pemrosesan *query* ini pada CPA adalah dengan melakukan gabungan dari TOP-K dan Skyline *query*. Proses penggabungan terjadi dengan cara memilah kriteria mana saja yang dapat digabung (agregasi) serta kriteria mana saja yang dapat dianggap saling lepas (di proses satu per-satu). Untuk kriteria yang dapat diagregasi maka akan dilakukan top-k *query* terhadap hasil *query* tersebut, dengan memperhitungkan kriteria tertentu saja, selanjutnya setelah dilakukan top-k *query*, maka dilakukan skyline *query* terhadap hasil *query* dengan memperhitungkan kriteria yang dapat dianggap saling lepas.

Pada contoh tabel di atas, rating pelayanan dan harga bisa dianggap sebagai suatu nilai yang bisa diagregasi dengan melakukan fungsi rata-rata, tentunya dengan terlebih dahulu menyamakan bobot dari masing-masing nilai. Pembobotan nilai dilakukan dengan terlebih dahulu mentransformasi tabel di atas menjadi urutan paling murah dan paling bagus layanannya, tidak seperti pada contoh di atas, menjadi sebagai berikut:

Table 2 Tabel Transformasi

No	Nama Tempat	Jarak	Harga Rata-rata	Rating Pelayanan
1.	Dago Pakar	100m	3	1
2.	Sapu Lidi	200m	2	2
3.	Bancakan	400m	4	2
4.	Kafemali	300m	1	3
5.	Kantin Borju	5m	2	3

Untuk top-3 *query* berdasarkan harga dan pelayanan, dengan fungsi rata-rata terkecil sebagai penyaring, maka didapati hasil:

1. Dago Pakar (Rata-rata = 2)
2. Sapu Lidi (Rata-rata =2)
3. Kafemali (Rata-rata=2)

Selanjutnya, hasil tersebut akan digabungkan dengan proses *query* skyline, dengan kriteria jarak sebagai acuan. Maka tanpa perlu mentransformasi lagi tabel, dapat dilihat bahwa urutan hasil yang akan keluar dari *query* adalah :

1. Dago Pakar (Jarak 100)

Kedua hasil lainnya dapat dijadikan kandidat.

4. ANALISIS

Setelah memaparkan dengan contoh yang sedikit lebih detil mengenai bagaimana kedua strategi *query* di atas dapat digunakan untuk menghasilkan informasi yang lebih tepat kepada pengguna, maka analisis yang diperoleh mengenai kedua strategi di atas adalah, bahwa kedua strategi diatas tergolong solusi yang sangat baik, karena proses penyaringan dilakukan berkali-kali terhadap hasil *query*, sehingga sangat memungkinkan munculnya jawaban yang sesuai dan relevan terhadap pengguna. Namun yang menjadi persoalan dalam penggunaan kedua strategi ini adalah bagaimana menentukan fungsi agregasi yang tepat, dan bagaimana menentukan kolom atau kriteria mana yang dapat diagregasikan, dalam hal penggunaan TOP-K, serta bagaimana menentukan kolom mana yang nilai tertinggi masing-masing kriterianya dapat dianggap bisa dikalkulasikan secara satu-per-satu. Kekurangan lainnya dari strategi *query* ini adalah proses pemangkasan hasilnya, tidak tertutup kemungkinan bahwa bisa saja hasil yang dipangkas adalah hasil yang juga diinginkan oleh pengguna, sesuai dengan nature dari manusia yang berpikir secara naluri, terkadang tidak begitu mengikuti aturan (preferensi) yang telah ditetapkan sendiri.

5. KESIMPULAN DAN SARAN

5.1 Saran

Setelah diakui bahwa kedua strategi *query* di atas adalah sangat baik, meski memiliki beberapa kesulitan atau kekurangan maka saran yang dapat diajukan lewat tulisan ini untuk pengembangannya adalah :

1. Agar hasil proses pemangkasan hasil tidak dilakukan secara kaku, memandang bisa saja hasil yang dipangkas adalah hasil yang diinginkan pengguna. Oleh karena itu disarankan agar diberi juga fasilitas untuk menampilkan beberapa urutan teratas hasil yang dipangkas.
2. Agar ditambahkan proses learning pada CPA ini, dimana setiap kali user memilih suatu hasil, agar

diberikan bobot kepada hasil tersebut, dan dapat digunakan untuk proses *query* berikutnya, agar proses pemberian urutan hasil berdasarkan preferensi semakin relevan dan tepat.

5.1 Kesimpulan

Dengan pemrosesan hasil *query* yang berlapis dengan menggunakan filtrasi bertingkat dengan menggunakan top-k dan skyline *query* sebagai strategi algoritma penyelesaian masalah penyampaian informasi yang relevan, layanan CPA ini menjadi suatu layanan yang sangat mangkus dan mutakhir untuk saat ini, meski belum ada penerapannya di Indonesia, oleh karena keterbatasan pengembang, dan juga teknologi.

REFERENSI

- [1] Rinaldi Munir, "Strategi Algoritmik", ITB, 2006
- [2] Mokbel, Mohamed F. , Justin J. Levandoski. (2009). "Toward Context and Preference Aware Location based." 8th International ACM Workshop on Data Engineering for Wireless and Mobile Access.
- [3] Schmidt, A., Beigle, M., Gellersen, H.W (1998) "*There is more to context than location. In Proc. Of the Intl Workshop on Interactive Applications of Mobile Computing (IMC 98)*
- [4] Ak Dey, "*Understanding and using context. Personal and Ubiquitous Computing 5: 20-24.*" (2001)
- [5] Kolari, Juha., Timo Laakko, Tapio Hiltunen, Veikko Ikonen, Minna Kulju, Raisa Suihkonen, Santtu Toivonen & Tytti Virtanen, "*Context Aware Services for Mobile User*", (2004)
- [6] Watson, Thomas J., Norman Cohen, James Black, Paul Castro, Maria Ebling, Barry Leiba, Archan Misra, Wolfgang Segmuller. "*Building Context Aware Applications with Context Weaver*". (IBM Research Report). (2004)