

PENCARIAN BARANG DALAM BASIS DATA ONLINE SHOP MENGGUNAKAN ALGORITMA BFS, DFS, DAN KMP

Zulva Fachrina - 13513010
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
zulva.fachrina@students.itb.ac.id

Abstrak—Online Shop merupakan salah satu bisnis yang cukup berkembang pesat akhir-akhir ini. Semakin besar suatu Online Shop, maka semakin besar data yang harus disimpan sehingga harus dibangun sistem basis data yang besar untuk menangani seluruh data tersebut. Salah satu cara untuk mendapatkan informasi mengenai barang tertentu pada basis data Online Shop adalah dengan memanfaatkan algoritma pencarian Breadth First Search atau Depth First Search yang dikombinasikan dengan algoritma String Matching Knutt-Morris Pratt. Fitur pencarian ini sering digunakan pemilik Online Shop untuk menarik minat pelanggan.

Kata Kunci—BFS, Basis data, DFS, KMP, Online Shop

I. PENDAHULUAN

Perkembangan teknologi dan internet yang semakin pesat membuat segala kegiatan dapat dilakukan dengan lebih mudah dan praktis, tanpa memakan cukup banyak waktu dan biaya. Salah satu dampak dapat dilihat dari pertumbuhan pesat pasar e-commerce. Kemudahan berbelanja dengan hanya menggunakan internet dan kartu kredit/debit tanpa harus pergi ke toko yang bersangkutan membuat online shopping cukup banyak digemari.

Di Indonesia sendiri, berdasarkan data dari Menkominfo, tercatat bahwa pada tahun 2013 nilai transaksi e-commerce mencapai angka 300 triliun, atau dengan kata lain 8% dari 30% pengguna internet di Indonesia pernah berbelanja secara online. Angka ini terus bertambah dengan semakin berkembangnya smartphone. Perkembangan ini sejalan dengan perkembangan perusahaan-perusahaan e-commerce di Indonesia. Sebut saja beberapa perusahaan besar yang sudah tidak asing lagi, seperti Olx.com, Lazada, Zalora, Tokopedia, dan BukaLapak. Tidak salah banyak individu maupun kelompok yang mulai tertarik untuk berkecimpung dalam bisnis jenis ini.

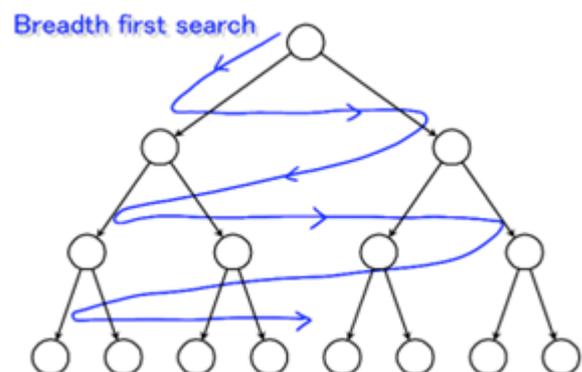
Salah satu fasilitas yang disediakan oleh berbagai online shop seperti yang disebutkan di atas adalah kemudahan dalam mencari barang yang diinginkan. Pelanggan cukup memasukkan query pencarian barang yang ingin dicari pada kolom yang telah disediakan, kemudian akan ditampilkan berbagai hasil pencarian. Dalam melakukan pencarian, pelanggan bisa memilih

kategori apa yang ingin dicari untuk membatasi pencarian, atau jika tidak program akan mencari barang tersebut dari berbagai kategori. Setiap barang pasti masuk dalam salah satu dari banyak kategori dan dalam setiap pencarian program akan menelusuri kategori-kategori tersebut.

Pencarian yang paling umum dilakukan untuk kasus tersebut adalah pencarian menggunakan query yang memanfaatkan operator relasional yang disediakan database. Namun dalam makalah ini, dijabarkan alternatif lain berupa Algoritma Pencarian BFS dan DFS, serta algoritma String Matching KMP yang dapat diterapkan pada pencarian barang-barang pada online shop. Setiap kategori dan sub kategori dapat diperlakukan sebagai simpul, dimana setiap simpul pasti akan berujung pada sebuah daun yang mewakilkan barang-barang yang tersedia. Pencarian dapat dilakukan dengan menerapkan algoritma BFS dan DFS dalam menelusuri setiap simpul sambil melakukan string matching untuk mendapatkan hasil yang sesuai

II. TEORI DASAR

A. Breadth First Search (BFS)



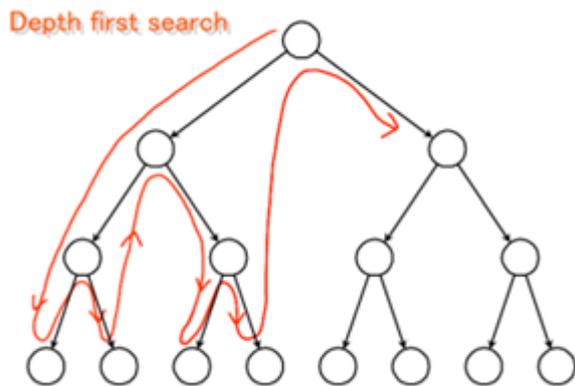
Breadth-first search (BFS) adalah algoritma pencarian/penelusuran traversal melebar pada sebuah graf atau pohon ruang status. *BFS* mengawali penelusuran dengan cara mengunjungi suatu simpul pada graf (atau pohon) status. Kemudian, *BFS* mengunjungi semua

simpul yang bertetangga dengan simpul tersebut secara *preorder*. Selanjutnya, simpul yang belum dikunjungi dan bertetangga dengan simpul-simpul yang tadi dikunjungi, demikian seterusnya. Jika graf berbentuk pohon berakar, maka semua simpul pada aras d dikunjungi lebih dulu dari simpul-simpul pada aras $d+1$. Atau dalam kata lain, BFS akan melakukan pencarian secara melebar terlebih dahulu (*breadth-first*).

Dalam penggunaannya, algoritma ini memerlukan sebuah antrian q untuk menyimpan simpul-simpul yang telah dikunjungi. Antrian ini diperlukan sebagai acuan dalam memilih simpul-simpul yang akan dikunjungi berikutnya, yaitu simpul-simpul yang bertetangga dengan simpul yang sudah ada di antrian. Tiap simpul yang telah dikunjungi masuk ke dalam antrian hanya satu kali. Algoritma ini juga membutuhkan tabel boolean untuk menyimpan simpul yang telah dikunjungi sehingga tidak ada simpul yang dikunjungi lebih dari satu kali. Urutan algoritma dari Breadth Search Firsh adalah sebagai berikut:

- Masukkan simpul *root* ke dalam antrian
- Periksa antrian terdepan apakah memiliki anak simpul
- Jika ya, masukan semua anak simpul ke dalam antrian
- Hapus antrian terdepan
- Jika antrian kosong berhenti, tapi jika tidak kembali ke langkah dua

B. Depth Search First (DFS)



Depth-first search (DFS) adalah algoritma pencarian/penelusuran traversal mendalam pada sebuah graf atau pohon ruang status. *DFS* mengawali penelusuran dengan cara mengunjungi suatu simpul pada graf (atau pohon) status. Kemudian, *DFS* mengunjungi simpul tetangga dari simpul tersebut yang dipilih secara *preorder*, terus menerus ke simpul tetangga dari simpul tetangga yang dikunjungi tidak memiliki tetangga lagi. Kemudian, *DFS* akan melakukan *backtrack*-ing ke simpul sebelumnya. Jika simpul ini masih memiliki tetangga yang belum dikunjungi, *DFS* akan kembali melakukan pencarian rekursif pada tetangga tersebut sampai menemukan simpul yang tidak memiliki tetangga lagi. Jika simpul ini tidak memiliki anak yang belum dikunjungi, *DFS* akan melanjutkan *backtrack*-nya ke

simpul sebelumnya lagi secara terus menerus dan rekursif sampai ditemukan sebuah simpul yang masih memiliki tetangga, dan kemudian melakukan pencarian rekursif pada simpul tetangga tersebut. Jika graf berbentuk pohon berakar, maka semua simpul pada aras d dikunjungi lebih dulu dari simpul-simpul pada aras $d+1$. Atau dalam kata lain, DFS akan melakukan pencarian secara mendalam terlebih dahulu (*depth-first*). Algoritma ini dapat diimplementasikan menggunakan *stack* atau secara rekursif. Urutan Algoritm DFS dengan menggunakan stack:

- Masukkan simpul *root* ke dalam tumpukan dengan push
- Ambil dan simpan isi elemen (berupa simpul pohon) dari tumpukan teratas
- Hapus isi stack teratas dengan prosedur pop
- Periksa apakah simpul pohon yang disimpan tadi memiliki anak simpul
- Jika ya, push semua anak simpul yang dibangkitkan ke dalam stack
- Jika tumpukan kosong berhenti, tapi jika tidak kembali ke langkah dua

C. Knuth-Morris-Pratt

Algoritma Knuth-Morris-Pratt adalah salah satu algoritma pencarian string, dikembangkan secara terpisah oleh Donald E. Knuth pada tahun 1967 dan James H. Morris bersama Vaughan R. Pratt pada tahun 1966, namun keduanya mempublikasikannya secara bersamaan pada tahun 1977.

Algoritma KMP melakukan proses awal terhadap pattern dengan menghitung fungsi pinggiran. Dengan adanya fungsi pinggiran ini, maka dapat dicegah pergeseran yang tidak berguna, seperti yang terjadi pada algoritma brute force. Fungsi pinggiran hanya bergantung pada karakter yang ada di dalam pattern, tidak bergantung kepada karakter di dalam teks.

I	N	F	O		I	N	F	O	R	M	A	S	I
F	O	R	M	A	S	I							
	F	O	R	M	A	S	I						
		F	O	R	M	A	S	I					
					F	O	R	M	A	S	I		
						F	O	R	M	A	S	I	
							F	O	R	M	A	S	I

Penjelasannya adalah sebagai berikut, mula-mula pattern dan teks sejajar pada posisi paling kiri, dan dibandingkan karakter pertamanya. Langkah 1 dan 2 sama dengan algoritma *brute force*, tetapi perhatikan pada langkah 3, ketidakcocokan terjadi saat membandingkan karakter ke-3 dari pattern "R" dan karakter ke-5 dari teks (spasi). Metode KMP akan memeriksa apakah pada teks

yang dilewati pada langkah ini (F-O-spasi) terdapat karakter awal pattern 'F', dan ternyata hanya cocok di karakter pertama yang sudah dibandingkan pada langkah ini.

Karena itu, tidak perlu menggeser pattern satu per satu, pattern dapat langsung digeser sejauh 3 karakter pada langkah selanjutnya. Demikian seterusnya hingga semuanya cocok.

Kompleksitas algoritma KMP meliputi kompleksitas perhitungan fungsi pinggiran yaitu: $O(m)$ dan kompleksitas pencarian *string* yaitu $O(n)$. Sehingga keseluruhan kompleksitas waktu algoritma KMP adalah $O(m+n)$.

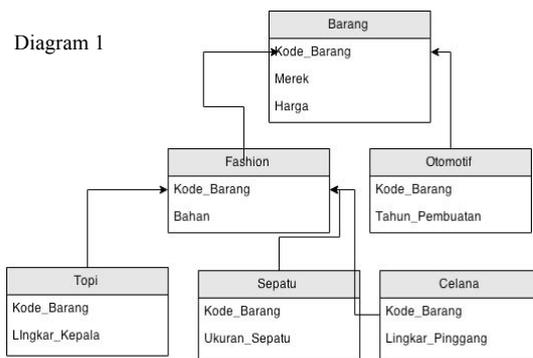
III. STRUKTUR PENYIMPANAN DAFTAR BARANG ONLINE SHOP

Pada setiap online shop pastilah terdapat daftar barang yang akan diperjual-belikan. Daftar barang-barang tersebut disimpan dalam sebuah relasi yang terhubung dengan relasi-relasi lainnya dalam sebuah basis data yang mengatur sistem secara keseluruhan.

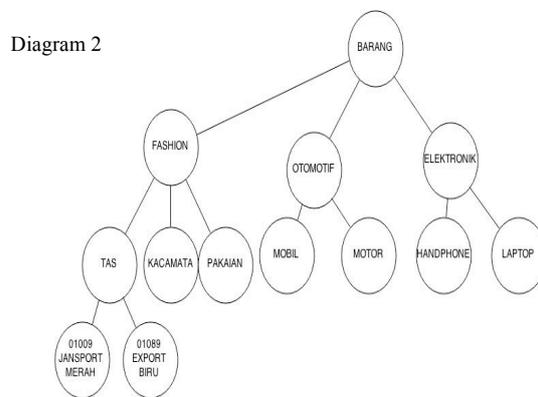
Misalkan relasi yang menyimpan daftar seluruh barang disebut sebagai relasi "Barang". Relasi "Barang" akan memiliki beberapa atribut seperti kode barang, merek barang, jenis barang, dan harga. Untuk setiap jenis barang terdapat atribut tambahan yang berbeda-beda, misalkan barang dengan jenis *pakaian* memiliki ukuran, barang dengan jenis *kendaraan* memiliki atribut tahun pembuatan, barang dengan jenis *Handphone* memiliki atribut seri. Jika seluruh barang dari setiap jenis disimpan dalam satu relasi "Barang", maka isi dari relasi akan tidak terhingga banyaknya dan akan banyak terjadi redundansi pada atribut jenis. Selain itu, seluruh atribut tambahan seperti ukuran, tahun pembuatan, dan seri akan dimasukkan dalam relasi "Barang" dan akan mengakibatkan banyak tuple yang diset dengan Null pada bagian di mana barang tersebut tidak memiliki atribut yang dimasukkan.

Untuk mengakali hal tersebut, maka pada basisdata, atribut jenis dipecah lagi menjadi relasi-relasi baru. Hal ini mengakibatkan akan terbentuk banyak relasi seperti "Fashion", "Elektronik", "Komputer", dan "Handphone". Masing-masing relasi akan memiliki atribut kode barang yang me-reference kode barang pada relasi "Barang". Jika pada relasi "Fashion" kembali terdapat kategori seperti pakaian, tas, sepatu, maka setiap kategori tersebut akan kembali dipecah menjadi relasi baru, dan begitu seterusnya. Masing-masing relasi akan memiliki key yang me-reference primary key pada relasi yang lebih besar.

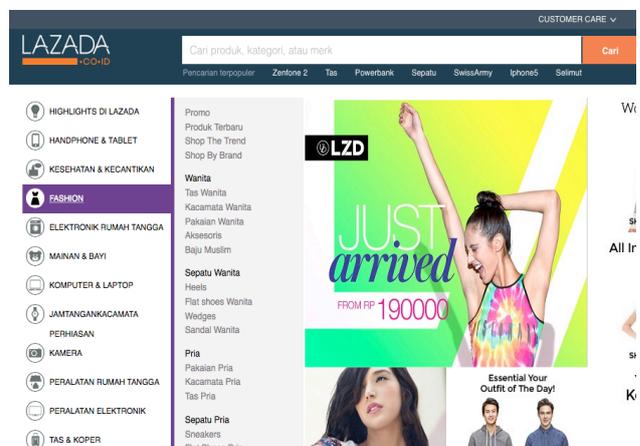
Dengan cara tersebut kita telah memiliki banyak relasi yang mewakili setiap kategori atau jenis dari barang. Setiap relasi terhubung satu sama lain melalui atribut kode_barang dan dapat digambarkan seperti gambar di bawah ini



Relasi tersebut dapat pula digambarkan dengan graph pohon. Jika untuk setiap jenis barang dibuat simpulnya masing-masing, kemudian setiap kategori memiliki beberapa kategori lain yang dilambangkan dengan subsimpul, hingga pada akhirnya berujung pada daun yang merupakan daftar barang yang diperjual-belikan.



Penggambaran struktur barang dan kategorinya menggunakan graph pohon seperti di atas lebih gampang untuk dimengerti karena hampir mirip dengan skema yang digunakan dalam website-online shop. Dalam hampir setiap website online shop, ditampilkan sejumlah kategori barang yang diperjual-belikan, kemudian user dapat memilih kategori atau jenis barang apa yang ingin dicari. Kategori-kategori tersebut merupakan bagian dari kategori yang lebih besar, sehingga jatuhnya seperti penyimpanan file pada direktori folder, dengan nama barang sebagai file di dalam folder.



Gambar 1: Tampilan Halaman Depan Website Online Shop
Sumber: www.Lazada.co.id

IV. PENCARIAN BARANG DENGAN MENGGUNAKAN ALGORITMA BFS, DFS, DAN KMP

Sebelum berbelanja, tentunya kita harus tahu terlebih dahulu benda apa yang ingin kita beli. Begitupula saat berbelanja melalui online shop. Jika kita sudah tahu barang spesifik yang ingin kita cari dan jenisnya, kita bisa langsung memasukkan query dalam kolom pencarian. Sedangkan jika kita ingin melihat terlebih dahulu berbagai macam barang yang disediakan sebelum membeli, kita bisa memilih salah satu kategori yang disediakan, yang kemudian akan menampilkan seluruh daftar barang yang berada dalam kategori tersebut.

Pencarian barang baik dengan memasukkan query ataupun dengan memilih kategori dapat menggunakan query pada SQL. Pencarian dengan menggunakan query pada database menggunakan prinsip select, join, dan berbagai operator relasional yang memanfaatkan keterhubungan antar relasi.

Dalam mencari barang yang diinginkan, perlu ditentukan query yang pas sehingga pencarian dapat ditemukan. Query biasanya menggunakan berbagai operator yang disediakan, dan berkaitan dengan menghubungkan satu relasi dengan relasi lain untuk mendapatkan informasi yang diinginkan.

Misalkan seorang pelanggan memasukkan query “Nike” dalam kolom pencarian, maka berdasarkan Diagram 1, query yang dimasukkan adalah mencari semua barang dalam relasi barang dimana Merek = “Nike”. Dalam SQL, query dapat ditulis:

```
SELECT *
FROM Barang
WHERE Merek = “Nike”
```

Query tersebut akan menampilkan hasil seluruh barang yang memiliki merek Nike. Namun, informasi yang didapatkan hanya informasi yang terkandung dalam relasi barang, yaitu kode_barang, merek, dan harga. Padahal, kita memerlukan seluruh informasi yang terkait dengan sepatu dengan Merek Nike, misalnya ukuran dari sepatu tersebut. Agar kita bisa mendapatkan seluruh informasi tersebut, kita harus mengganti query menjadi:

```
SELECT *
FROM Barang NATURAL JOIN Sepatu
WHERE Merek = “Nike”
```

Query ini kemudian menjadi masalah ketika Merek Nike tidak hanya merupakan merek dari sepatu, namun juga dapat merupakan merek dari baju ataupun tas, dan kita tidak tahu jenis barang apa yang diinginkan oleh pelanggan. Karena itu akan terdapat banyak relasi yang terlibat dalam query pencarian

```
SELECT *
```

```
FROM Barang NATURAL JOIN Sepatu
NATURAL JOIN Tas NATURAL JOIN Baju
NATURAL JOIN ...
WHERE Merek = “Nike”
```

Salah satu alternatif pencarian query yang dapat digunakan dalam mencari barang tertentu pada database Online Shop selain menggunakan SQL adalah dengan mengkombinasikan algoritma BFS atau DFS dengan String Matching. Cara ini terbilang lebih memakan waktu dan kompleks, namun lebih mudah untuk dipahami dan diimplementasikan. Pencarian dengan algoritma BFS dan DFS memanfaatkan struktur penyimpanan yang seperti pohon untuk menelusuri seluruh simpul sampai menemukan hasil yang diinginkan. Pada makalah ini hanya akan dibahas pencarian dimana pelanggan memasukkan query di dalam kolom pencarian, yaitu pelanggan sudah tahu barang yang ingin dibeli. Pencarian juga akan dilakukan dengan asumsi setiap barang pasti berada dalam satu kategori dan tidak ada barang yang berada pada dua kategori yang berlainan.

A. Pencarian Menggunakan BFS

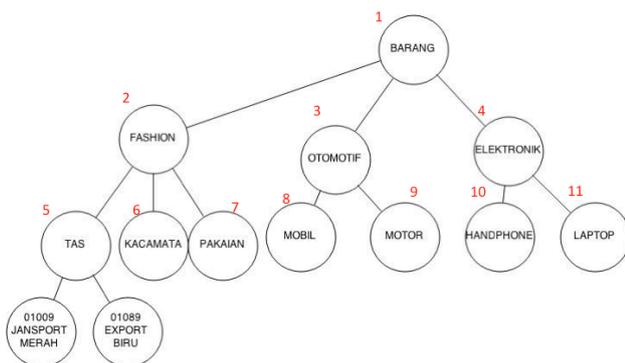
Penelusuran simpul-simpul kategori dimulai dari simpul paling atas atau akar. Sebelumnya terlebih dahulu dibuat sebuah queue kosong yang akan menampung setiap simpul yang dikunjungi. Karena diasumsikan setiap barang tidak mungkin masuk dalam dua kategori berbeda, maka tidak diperlukan penandaan apakah suatu simpul sudah dikunjungi atau belum.

Pertama-tama, program akan menelusuri semua kategori yang terdapat pada akar. Setiap kategori akan dimasukkan ke dalam Queue. Program kemudian akan mengambil simpul pertama yang berada di dalam Queue sesuai dengan prinsip Queue yaitu First In First Out (FIFO). Simpul yang diambil tersebut akan kembali ditelusuri setiap simpul-simpul anaknya. Setiap simpul anak yang didapat dari simpul tersebut kembali dimasukkan ke dalam Queue. Begitu seterusnya sampai didapat simpul daun yang di dalamnya mengandung query yang ingin dicari. Pencocokan query dapat dilakukan dengan menggunakan string matching. Ketika sejumlah simpul cocok, maka program akan menampilkan seluruh kemungkinan tersebut.

Pencarian tersebut dapat digambarkan sebagai berikut. Jika seorang pelanggan memasukkan query pencarian “Jansport”, maka berdasarkan Diagram 2, yang dilakukan program adalah sebagai berikut:

- Seluruh simpul anak Barang, yaitu Fashion, Otomotif, dan Elektronik ke dalam Queue. Isi Queue: Fashion, Otomotif, Elektronik.
- Fashion dikeluarkan dari Queue, kemudian seluruh anak simpul dari Fashion dimasukkan ke dalam Queue, Isi Queue: Otomotif, Elektronik, Tas, Kacamata, dan Pakaian.
- Otomotif dikeluarkan dari Queue, kemudian seluruh anak simpul dari otomotif dimasukkan ke dalam Queue. Isi Queue: Elektronik, Tas, Kacamata, Pakaian, Motor, Mobil.

- Elektronik dikeluarkan dari Queue, kemudian seluruh anak simpul dari elektronik dimasukkan ke dalam Queue. Isi Queue: Tas, Kacamata, Pakaian, Motor, Mobil, Laptop, Handphone
- Tas dikeluarkan dari Queue. Di dalam simpul Tas terdapat banyak simpul daun, dan beberapa di antaranya merupakan simpul yang mengandung Jansport seperti pada diagram. Sambil menelusuri setiap simpul dan mencocokkan string, program juga menuliskan hasil pencarian ke layar. Dengan begitu, semua barang yang memenuhi query pencarian sudah ditemukan



B. Pencarian Menggunakan DFS

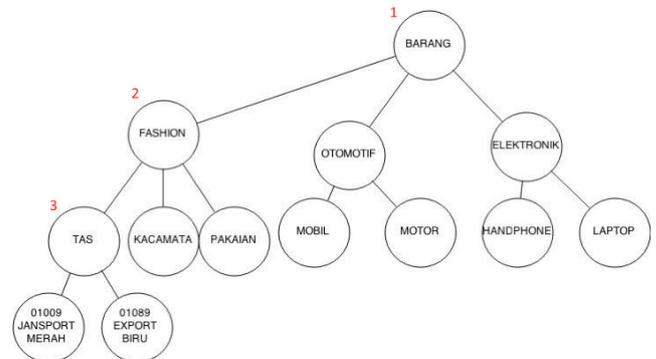
Pencarian menggunakan DFS dapat menggunakan stack ataupun secara rekursif. Pada makalah ini akan dijabarkan pemanfaatan pencarian menggunakan algoritma DFS.

Pertama-tama dibuat sebuah stack dan dimasukkan kosong dan masukkan akar ke dalam stack. Ambil simpul kategori yang berada pada posisi paling atas dan bangkitkan semua simpul anaknya. Simpul anak tersebut kemudian kembali dimasukkan ke dalam stack. Ambil lagi simpul teratas pada stack dan ulangi langkah sebelumnya sampai seluruh stack kosong atau pencarian sudah ditemukan.

Pencarian tersebut untuk lebih mudahnya digambarkan sebagai berikut. Jika seorang pelanggan memasukkan query “Jansport” pada pencarian, maka berdasarkan diagram 2, yang dilakukan proram adalah sebagai berikut:

- Seluruh simpul anak akar yaitu Elektronik, Otomotif, dan Fashion dimasukkan ke dalam stack. Isi stack: Elektronik, Otomotif, Fashion.
- Fashion yang berada pada tumpukan teratas stack di-pop, kemudian semua anak simpul Fashion dibangkitkan dan dimasukkan ke dalam stack. Isi stack: Elektronik, Otomotif, Pakaian, Kacamata, Tas.
- Tas yang berada pada tumpukan teratas stack di-pop, kemudian dibangkitkan seluruh simpul anaknya. Simpul-simpul anak tersebut kembali dimasukkan ke dalam stack. Ketika simpul anak teratas di-pop dari stack, simpul tersebut tidak

memiliki anak lagi karena merupakan simpul daun. Untuk setiap simpul daun akan dilakukan pencocokan string apakah sesuai dengan query yang diminta. Jika sesuai, maka hasil akan ditampilkan di layar. Proseuder ini terus dilakukan untuk setiap simpul daun yang berada dalam stack hingga seluruh simpul habis.



C. String Matching Menggunakan Algoritma KMP

Pada setiap simpul yang dikunjungi, algoritma string matching diterapkan untuk mencari simpul yang memiliki kecocokan dengan query yang diberikan. Algoritma string matching yang dipilih untuk digunakan pada makalah ini adalah algoritma Knutt-Morris Pratt karena algoritma ini memiliki kompleksitas yang cukup optimal, terutama jika dibandingkan dengan algoritma Brute Force.

Langkah pertama yang harus dilakukan adalah membentuk fungsi pinggiran dari query yang didapatkan. Tabel Fungsi Pinggiran KMP berukuran 1 x m dengan m merupakan panjang query dan masing-masing kolom m mewakili satu karakter pada pattern. Masing-masing kolom k diisi dengan ukuran paling besar prefix dari pattern[1..k] yang juga merupakan suffix dari pattern[1..k]

Setelah fungsi pinggiran terbentuk, maka pencocokan string yang terdapat pada simpul dengan string pada query dapat dilakukan. Pencocokan string dimulai dari paling kiri, kemudian satu per satu dilakukan pengecekan apakah karakter pada text simpul sama dengan pada query. Pada kasus ini terdapat 3 kemungkinan:

- Jika karakter sama, maka teruskan pengecekan tanpa menggeser query
- Jika karakter tidak sama dan karakter tersebut merupakan karakter pertama pada query, maka geser query satu karakter ke kanan
- Jika karakter tidak sama dan karakter tersebut bukan merupakan karakter pertama pada query, maka geser query sejauh indeks karakter sebelum terjadi kesalahan dikurangi isi tabel fungsi pinggiran pada indeks tersebut

Langkah tersebut diulangi sampai string ditemukan atau query tidak lagi dapat digeser atau dengan kata lain string tidak ditemukan.

Mari kembali ambil contoh ketika seorang pelanggan mencari query “Jansport” dan salah satu simpul daun memiliki merek “Jas Jansport”. Pencarian menggunakan KMP dapat diilustrasikan sebagai berikut:

Fungsi pinggiran JANSPO

1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0

J A S J A N S P O R T
 J A N S P O R T
 J A N S P O R T
 J A N S P O R T
 J A N S P O R T

Pada baris ketiga dapat dilihat bahwa query tidak hanya digeser sejauh satu karakter, melainkan 2 karakter sekaligus. Pergeseran ini didapat melalui fungsi pinggiran. Indeks karakter terakhir sebelum terjadi ketidakcocokan adalah 2, dan nilai dari table dengan indeks 2 adalah 0, maka query akan digeser sejauh $2 - 0 = 2$ karakter.

V. KESIMPULAN

Dalam pencarian query pada suatu basis data, selain menggunakan operator relasional yang disediakan SQL, dapat juga menggunakan perpaduan antara algoritma Pencarian Breadth First Search atau Depth First Search dengan algoritma String Matching Knutt-Morris Pratt. Salah satu contohnya adalah pencarian query yang dimasukkan pelanggan dalam sebuah website Online Shop. Struktur penyimpanan dapat dibuat menyerupai struktur pohon pada graph, sehingga setiap kategori memiliki sub kategori yang diwakilkan dengan simpul. Algoritma BFS atau DFS digunakan untuk menelusuri setiap simpul yang mewakili sebuah kategori, sedangkan algoritma string matching KMP akan digunakan untuk mencocokkan data yang tersimpan pada simpul daun dengan query pencarian pelanggan.

VI. UCAPAN TERIMA KASIH

Penulis ingin mengucapkan terima kasih kepada Allah SWT, karena atas rahmat dan karunia-Nya lah makalah ini dapat selesai pada waktunya. Penulis juga ingin mengucapkan terima kasih kepada kedua orang tua yang tidak pernah letih mendukung dan mendoakan anaknya, serta Bapak Dr. Ir. Rinaldi Munir dan Ibu Dr. Nur Ulfa Maulidevi, S.T., M.Sc selaku dosen mata kuliah Strategi Algoritma. Tidak lupa penulis juga ingin mengucapkan terima kasih kepada pihak-pihak yang secara langsung

maupun tidak telah membantu dalam merampungkan makalah ini.

REFERENCES

- [1] Munir, Rinaldi. 2008. *Diktat Kuliah IF2091 Matematika Diskrit*. Bandung: Penerbit Informatika.
- [2] “Data Statistik Mengenai Pertumbuhan Pangsa E-Commerce di Indonesia Saat Ini” <http://startubisnis.com/data-statistik-mengenai-pertumbuhan-pangsa-pasar-e-commerce-di-indonesia-saat-ini/> (Diakses Tanggal 3 Mei 2015 pukul 20.00)
- [3] www.lazada.co.id (Diakses Tanggal 4 Mei 2015 pukul 15.00)

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, 4 Mei 2015



Zulva Fachrina/13513010