

Kajian Optimasi Query Dengan Pendekatan Heuristik

Julia Kurniasih¹, Henderi²

^{1,2}Magister Teknik Informatika, Universitas AMIKOM Yogyakarta

e-mail: ¹asihseketi@gmail.com, ²henderi@mail.ugm.ac.id

Intisari

Semakin kompleks proses bisnis sebuah sistem, maka semakin besar pula data yang dimiliki. Proses transaksi data juga akan semakin intensif, sehingga jika tidak di optimasi dapat berdampak pada kinerja sistemnya. Proses optimasi query akan memberikan beberapa rancangan query yang akan di uji untuk mengidentifikasi dan memilih query yang paling optimal. Salah satu pendekatan optimasi yang dapat digunakan untuk melakukan optimasi query adalah pendekatan heuristic. Pendekatan heuristic akan mengubah pohon-query (query tree) dengan menggunakan seperangkat aturan (rule) yang bertujuan untuk meningkatkan kinerja eksekusi query. Query direpresentasikan ke dalam bentuk struktur data query tree dengan menggunakan algoritma optimasi query heuristic. Dengan pendekatan heuristic pada proses optimasi query akan menghasilkan jumlah tuple yang lebih sedikit dikarenakan tuple-tuple sudah dipilih sesuai dengan kondisi melalui proses transformasi aljabar relasional. Dengan jumlah tuple yang lebih sedikit maka waktu eksekusi query untuk operasi join tabel dapat menjadi lebih cepat..

Kata kunci—optimasi, query, heuristic

Abstract

The more complex the system data, the greater the data owned. The process of data transactions will also be more intensive, if not able to optimize it can affect to the system performance. Query optimization generates some selected query algorithms to select the most optimal query. One of the optimization approach that can be used to perform query optimization is the heuristic approach. A heuristic approach optimization using a set of rules that aims to improve the quality of the execution query. The query is represented in the query tree data structure using the heuristic query optimization algorithm. The heuristic approach to the query optimization process produces a smaller number of tuples because the tuples are selected according to conditions through the relational algebra transformation process. With fewer tuples the query execution time for join table operations can become faster.

Keywords—optimization, query, heuristic

PENDAHULUAN

Hampir semua sistem membutuhkan media penyimpanan data. Pada umumnya semakin kompleks proses bisnis sebuah sistem, maka semakin besar pula data yang dimiliki. Proses transaksi data juga akan semakin intensif, sehingga jika tidak di optimasi dapat berdampak pada kinerja sistemnya. Database merupakan salah satu bentuk penyimpanan data yang digunakan oleh sebagian besar sistem yang ada pada saat ini. Dibandingkan dengan penggunaan file sebagai bentuk penyimpanan data untuk aplikasi tunggal, database lebih efisien digunakan sebagai tempat penyimpanan data karena memiliki kemampuan keterkaitan antara data dan lebih mudah untuk dikembangkan. Pada sistem penyimpanan data yang besar, merupakan hal yang penting untuk mengoptimasi kerja query agar sistem dapat dimanfaatkan secara maksimal dan meminimalkan waktu proses [1].

Optimasi query adalah proses menentukan strategi eksekusi yang optimal untuk pemrosesan query [2]. Optimasi dari suatu database dapat dilakukan pada bagian 1) struktur database, yaitu dengan mengatur struktur tabel agar tidak terjadi redundansi yang dapat menghambat kinerja dari sistem dan 2) query yang di eksekusi untuk melakukan pengambilan data dari tabel database. Teknik optimasi query memberikan beberapa rancangan query yang diuji untuk mengidentifikasi dan memilih query yang paling optimal, yaitu yang meminimalkan penggunaan sumber daya data tanpa merubah output [3]. Optimasi query mencakup beberapa proses seperti transformasi query ke dalam bentuk logika yang sama, memilih jalan akses yang optimal dan mengoptimalkan penyimpanan data. Secara umum optimasi query dapat dilakukan dengan pendekatan heuristic dan pendekatan berbasis biaya. Dibandingkan optimasi dengan pendekatan berbasis biaya, pendekatan heuristic dinyatakan lebih murah dikarenakan pendekatan heuristic meminimalkan jumlah akses dengan mengurangi jumlah tuple dan kolom yang dicari [4]. Optimasi dengan menggunakan pendekatan heuristik mengubah pohon-query (query tree) dengan menggunakan seperangkat aturan yang umumnya dapat meningkatkan kinerja eksekusi query.

Beberapa penelitian terkait :

1. Experiment of Query Optimization Techniques To Get The Efficient Query [5]. Penelitian ini ditulis oleh Hakim dkk. Penelitian ini melakukan perbandingan antara query modifikasi hasil optimasi dengan query non optimasi. Hasilnya menunjukkan query optimasi lebih efisien dibandingkan dengan query non optimasi dilihat dari waktu prosesnya.
2. Perbandingan Optimasi Query Database Menggunakan Algoritma Genetic dan Metode Heuristic Untuk Meningkatkan Kecepatan Pengaksesan Query [6]. Penelitian ini ditulis oleh Nuraisa. Pada penelitian ini digunakan dua metode untuk mengoptimasi query yaitu metode heuristic dan algoritma genetic yang diuji dan dibandingkan. Hasil eksperimen menunjukkan bahwa metode heuristic memiliki performa yang baik dalam meningkatkan kecepatan akses query berdasarkan waktu akses yang dihasilkan yaitu 0.05315 second dan 0.05548 second untuk metode genetic.
3. Query Optimization Techniques-Tips for Writing Efficient and Faster SQL Queries [7]. Penelitian ini ditulis oleh Habimana. Pada penelitian ini dilakukan pemilihan tips dan trik optimasi query dalam mendapatkan kinerja yang lebih baik.

METODE PENELITIAN

Pada penelitian ini dilakukan kajian terhadap pendekatan heuristic yang digunakan untuk optimasi query. Algoritma heuristic yang digunakan terlihat pada Gambar 1.

Sebelum melakukan optimasi, query direpresentasikan ke dalam bentuk struktur data yaitu dapat menggunakan query tree (struktur data yang sesuai dengan ekspresi aljabar relasional) dan query graph (struktur data yang sesuai dengan ekspresi kalkulus relasional). Query diterjemahkan ke bentuk yang setara dengan penjabaran ekspresi aljabar relasional yang dinyatakan dalam bentuk query tree. Untuk merubah sebuah query ke bentuk query tree, sebuah query diuraikan menjadi

query block. Query block adalah unit dasar yang dapat dioptimasi dan diterjemahkan ke dalam operator aljabar. Query block berisi ekspresi SELECT-FROM-WHERE tunggal, serta klausa GROUP BY dan HAVING jika merupakan bagian dari block.

<p>Input: Relational Algebra Query Output: Best Evaluation Plan Algorithm: Step 1: Design the initial canonical tree of the query Step 2: Move the SELECT operation down the tree query Step 3: Apply more restrictive SELECT operation first. Step 4: Replace CARTESIAN PRODUCT and SELECT with JOIN operation Step 5: Move PROJECT operations down the query tree</p>
--

Gambar 1. Algoritma Optimasi Query Heuristic

Penelitian ini menggunakan representasi query tree untuk menggambarkan proses optimasi heuristic-nya.

Kebutuhan untuk simulasi :

1. Tabel-tabel

- Tabel_Penerbangan (pno : string, pstno : string, asal : string, tujuan : string, jarak : integer, berangkat : time, tiba : time, harga : currency, pstno : string)
- Tabel_Pesawat (pstno : string, pstnama : string, jaraktempuh : integer)
- Tabel_Pilot (pilotid : string, pstno : string, pilot_nama : string, gaji : currency)

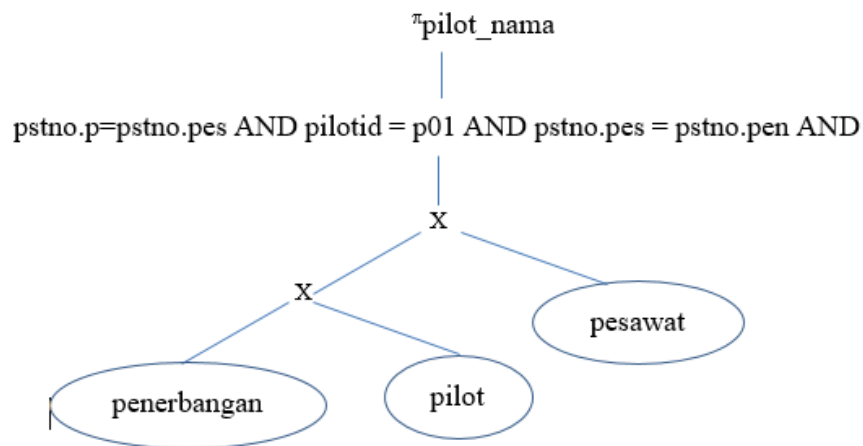
2. Query yang dioptimasi

SELECT pilot_nama FROM pilot p, pesawat pes, penerbangan pen WHERE pstno.p=pstno.pes AND pilotid = p01 AND pstno.pes = pstno.pen AND asal = "semarang";

HASIL DAN PEMBAHASAN

Dengan menggunakan tabel-tabel dan query untuk simulasi, proses optimasi query heuristic terlihat pada urutan langkah-langkah berikut :

1. Inisial query tree



Gambar 2. Inisial Query Tree

Menjalankan tree ini secara langsung akan membentuk sebuah file yang besar yang berisi CARTESIAN PRODUCT dari keseluruhan file-file Pilot, Penerbangan dan Pesawat. Namun query ini hanya memerlukan satu record dari relasi Pilot untuk id = p01 dan hanya record Penerbangan untuk yang berasal dari kota semarang.

2. Menambahkan operasi SELECT ke dalam query tree

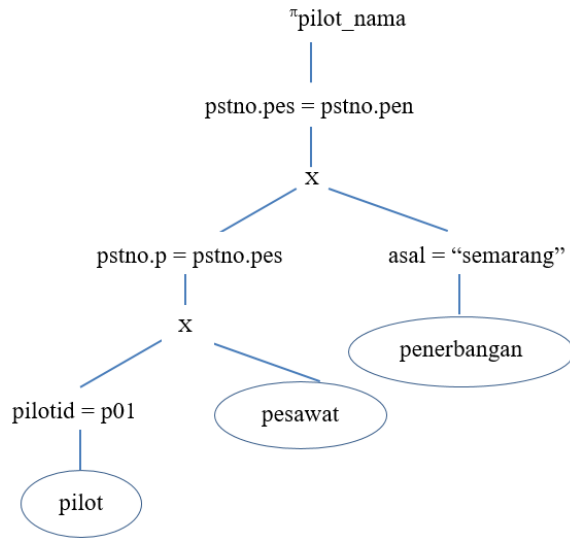
Gambar 3 menunjukkan perbaikan query tree untuk mengurangi banyaknya tuple yang terlihat pada CARTESIAN PRODUCT.

3. Membatasi penggunaan operasi SELECT terlebih dahulu

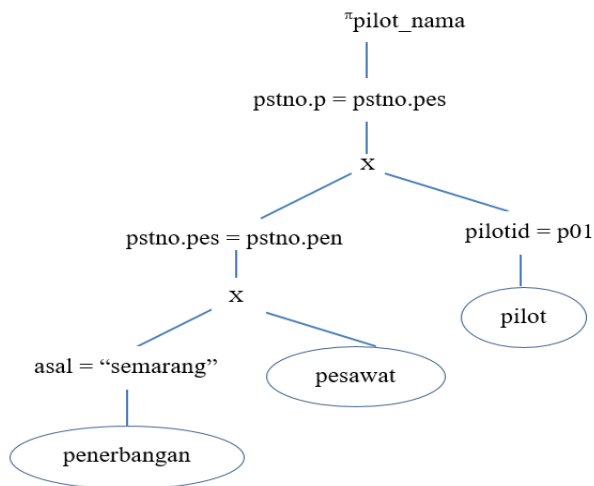
Perbaikan selanjutnya dicapai dengan menukar posisi dari relasi Pilot, Pesawat, Penerbangan di dalam tree. Prosesnya terlihat pada Gambar 4.

4. Mengganti CARTESIAN PRODUCT dan SELECT dengan operasi-operasi JOIN

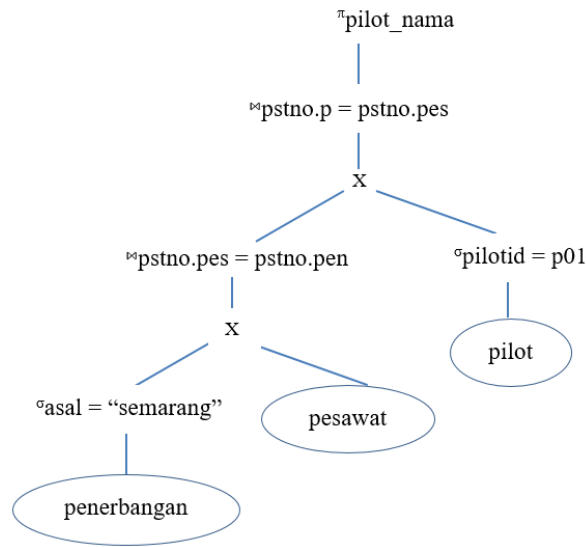
Selanjutnya query tree diperbaiki dengan cara mengembalikan beberapa operasi CARTESIAN PRODUCT yang diikuti dengan sebuah kondisi JOIN. Prosesnya terlihat pada Gambar 5.



Gambar 3. Proses Penambahan Operasi SELECT



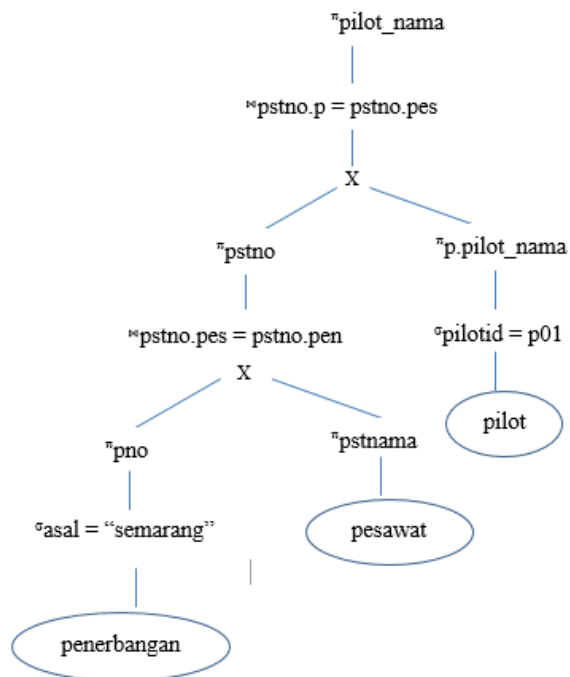
Gambar 4. Proses Pembatasan Penggunaan Operasi SELECT



Gambar 5. Proses Pergantian CARTESIAN PRODUCT dan SELECT Dengan Operasi JOIN

5. Memindahkan operasi PROJECT ke bawah Tree

Perbaikan selanjutnya adalah menyimpan atribut-atribut yang diperlukan oleh operasi berikutnya, hal ini dapat mengurangi atribut/kolom dari relasi-relasinya.



Gambar 6. Proses Memindahkan Operasi PROJECT Ke Bawah Tree

KESIMPULAN

Optimasi query perlu dilakukan untuk meningkatkan kinerja eksekusi query. Pendekatan heuristic yang digunakan untuk optimasi query akan menghasilkan jumlah tuple yang lebih sedikit dikarenakan tuple-tuple sudah dipilih sesuai dengan kondisi melalui proses transformasi aljabar relasionalnya. Dengan jumlah tuple yang lebih sedikit maka waktu eksekusi query untuk operasi join tabel dapat menjadi lebih cepat.

DAFTAR PUSTAKA

- [1]Thangam, A. Regita and Peter, S. John; 2016, En Extensive Survey on Various Query Optimization Techniques, IJCSMC Vol 5 Issue 8.
- [2]Elmasri, R. and Navathe, S. B.; 2011, Fundamentals of Database Systems, Sixth Edition, Addison-Wesley, United States of America.
- [3]Muhammad, L. J., Zakariyau, Y. B. and Ali, A. G.; 2015, Efficient Multi-Query Optimization Algorithm, International Journal of Database Theory and Application Vol. 8 No. 6.
- [4]Hatmode, V. and Rangdale, S.; 2014, Heuristic based Query Optimization, International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 6, June 2014.
- [5]Nuraisha, S., 2016, Perbandingan Optimasi Query Database Menggunakan Algoritma Genetic dan Metode Heuristic Untuk Meningkatkan Kecepatan Pengaksesan Query, Skripsi, Teknik Informatika FIK UDINUS.

- [6]Habimana, J., 2015, Query Optimization Techniques-Tips for Writing Efficient and Faster SQL Queries, International Journal of Scientific & Technology Research Vol. 4, Issue 10.
- [7]Hakim, L., Nugroho, S., and Waluyo, S. H.; 2017, Experiment of Query Optimization Techniques To Get The Efficient Query, International Journal of Computer Applications Vol 170 – No. 10.
- [8]Vincent, M., Kaushik, A., and Amandeep, S. B.; 2014, Issues in Query Processing and Optimization International Journal of Modern Trends in Engineering and Research (IJMTER) Vol. 1, Issue 5.

Kajian Optimasi Query Dengan Pendekatan Heuristik

(Julia Kurniasih, Henderi)