

OPTIMASI QUERY MENGGUNAKAN ALGORITMA INGRES PADA PORTAL LUMBUNG DATA PENDIDIKAN JAWA TENGAH

Galih Setyo Wibowo¹, Ajib Susanto², Ibnu Utomo W.M.³

^{1, 2, 3} Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro

Jl. Nakula 1 No. 5-11 Semarang

Email: galih@dinustek.com¹, ajib.susanto@dsn.dinus.ac.id², ibnu.utomo.wm@dsn.dinus.ac.id³

Abstrak

Portal web Lumbung Data Pendidikan Jawa Tengah memiliki jumlah data record yang besar yaitu 46.442.378 record, dan terdapat tabel sebanyak 31, untuk mendapatkan informasi yang diinginkan diperlukan relasi tabel yang banyak sehingga diperlukan waktu lebih lama dalam memproses *query*, dikarenakan tidak menggunakan metode khusus dalam melakukan *query* atau hanya menggunakan teknik *query* konvensional hasilnya waktu yang dibutuhkan untuk memproses *query* menjadi relatif lambat. Tujuan penelitian ini untuk mengoptimalkan *query* menggunakan algoritma INGRES yang nantinya akan diuji dan dibandingkan hasilnya dengan *query* konvensional untuk mengetahui sejauh mana pemrosesan *query* dapat dipercepat. Hasil eksperimen yang dilakukan menunjukkan bahwa algoritma INGRES dapat mempercepat proses eksekusi akan tetapi masih memiliki kekurangan, pada pemrosesan awal penggunaan algoritma INGRES justru mengakibatkan penurunan kecepatan hingga 122%, akan tetapi pada proses berikutnya dapat menaikkan kecepatan hingga 60%. Hal ini dikarenakan pada pemrosesan *query* dengan menerapkan algoritma INGRES diperlukan pemecahan *query* dan pengekseskuan secara berurutan yang hasilnya ditampung ke dalam *temporary* tabel terlebih dahulu, hingga didapatkan *query* akhir yang paling optimal.

Kata kunci : Optimasi Query, INGRES, Optimasi Query Basis Data Tersentralisasi

1. PENDAHULUAN

1.1. Latar Belakang

Data merupakan catatan fakta-fakta dalam kehidupan sehari-hari yang dapat menggambarkan suatu peristiwa tertentu yang sebenarnya (Lubis, 2014), data dapat berupa angka atau *quantitative data* yang menghasilkan informasi numerik dan juga dapat berupa kata atau *qualitative data* yang menghasilkan informasi deskriptif, kumpulan data-data dari berbagai peristiwa dalam kehidupan sehari-hari apabila dicari hubungannya antara satu dengan yang lain dapat menghasilkan suatu informasi penting yang dapat digunakan manusia ketika menjalani kehidupannya sehari-hari untuk mengambil keputusan dalam berbagai persoalannya.

Seiring dengan perkembangan zaman dimana manusia hidup dalam era teknologi saat ini banyak hal dapat dimudahkan dengan adanya teknologi komputerisasi, pengolahan data dapat dilakukan dengan lebih mudah dan cepat untuk menghasilkan informasi yang bermanfaat terutama dalam suatu kegiatan-kegiatan besar yang dilakukan kurun waktu yang lama dan berulang-ulang yang tentunya memiliki data yang besar pula untuk diolah. Dalam penyajian informasi saat ini aplikasi berbasis *web* yang semakin canggih dan didukung dengan basis data yang baik mampu menjadikan dunia ini tidak terbatas ruang maupun waktu (Askar, Susanto, 2018).

Di dalam sistem komputerisasi tempat dimana data dikumpulkan dan dikelola dikenal dengan basis data / *database* (Connolly, Begg, 2005), yaitu tempat berkumpulnya data yang saling berhubungan dalam suatu wadah yang dikelola secara terstruktur bertujuan agar dapat mempermudah dan mempercepat untuk pemanggilan kembali data-data yang diinginkan sehingga dapat dimanfaatkan dengan segera (Lubis, 2014), selanjutnya untuk mengelola basis data tersebut diperlukan suatu sistem basis data yaitu suatu sistem penyusunan dan pengelolaan *record-record* data dengan menggunakan komputer, dengan tujuan untuk menyimpan atau merekam serta memelihara data secara lengkap, sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk kepentingan proses pengambilan keputusan (Lubis, 2014), hal ini memiliki peran yang sangat penting untuk menunjang keberlangsungan suatu kegiatan, terutama kegiatan-kegiatan yang menggunakan data yang besar untuk dikelola.

Indonesia saat ini memiliki suatu sistem pengelolaan data pendidikan yang dinamakan Data Pokok Pendidikan (DAPODIK), DAPODIK merupakan sistem pengelolaan data pokok pendidikan yang merupakan data-data mikro peserta didik, satuan pendidikan, dan tenaga pendidik secara *online* dan *real time* dengan jangkauan data berskala nasional (Dapodik, 2014). Sistem ini mengelola data-data pendidikan di seluruh Indonesia yang didapatkan dari sekolah-sekolah di daerah, setiap sekolah menyetorkan data mereka secara langsung dengan sistem tersebut ke pusat data yang dibagi menjadi dua bagian yaitu Pendidikan Dasar (Dikdas), dan Pendidikan menengah (Dikmen) yang selanjutnya diolah oleh Pusat Data dan Statistik Pendidikan (PDSP) di Kemdikbud. Salah satu tujuan dari dapodik adalah untuk dapat memberikan informasi kepada pelaku dan penyelenggara pendidikan terutama kepada pengambil kebijakan mengenai realita kondisi pendidikan di Indonesia untuk dapat merencanakan penyelenggaraan pendidikan dengan lebih baik dan melakukan penghematan biaya pendidikan yang kurang tepat sasaran sehingga penghematan ini dapat ditujukan untuk pembiayaan program-program pendidikan lain yang lebih sesuai dengan kebutuhan.

Pada awalnya setelah dilakukan pengumpulan data oleh sekolah-sekolah di daerah ke pusat, pemanfaatan dapodik masih kurang efektif di daerah karena untuk mendapatkan data-data tersebut pengelola pendidikan di daerah harus datang langsung ke Kemdikbud di Jakarta, oleh karena itu dibangunlah sistem *backbone* Dapodik, sistem ini berfungsi untuk mendistribusikan data-data Dapodik yang sudah dikumpulkan di pusat agar pemanfaatannya di daerah menjadi lebih mudah. Jawa Tengah merupakan salah satu provinsi yang mulai mengawali pembuatan *backbone* ini, dan untuk pengelolaannya Dinas Pendidikan Provinsi Jawa Tengah mengembangkan portal *web* Dapodik Jawa Tengah yang diberi nama Lumbung Data Pendidikan Jawa Tengah, portal *web* tersebut dapat diakses dengan alamat <http://dapodik.pdkjateng.go.id> (Dapodik Jawa Tengah, 2018).

Portal *web* ini mengelola data yang sangat besar hanya untuk lingkup provinsi Jawa Tengah saja, selain jumlah *data record* yang besar yaitu 46.442.378 *record*, basis data portal lumbung Data Pendidikan Jawa Tengah juga memuat tabel sebanyak 31 yang mana untuk mendapatkan informasi yang diinginkan banyak diperlukan relasi tabel. Dalam pengembangannya terdapat permasalahan berkenaan dengan waktu yang diperlukan dalam memproses *query*, dikarenakan tidak menggunakan metode khusus dalam melakukan *query* atau dengan kata lain hanya menggunakan teknik *query* konvensional hasilnya waktu yang dibutuhkan untuk memproses *query* menjadi relatif lambat sehingga proses penyajian data kepada pengguna menjadi kurang efisien dalam hal waktu. Untuk memecahkan permasalahan tersebut peneliti melakukan studi pustaka dengan mencari penelitian-penelitian sebelumnya yang berkaitan dengan optimasi *query* (Silberschatz, Korth, and Sudarshan, 2011) dan mempelajari metode-metode *query* yang digunakan untuk dapat diterapkan pada kasus Lumbung Data Pendidikan Jawa Tengah. Optimasi *query* menjadi salah satu cara yang dapat digunakan untuk melakukan sebuah proses pengolahan data dalam mencari jalur tercepat pada *database management system (DBMS)* (Dewi, Rezeki, 2016).

Penelitian mengenai optimasi *query* sebelumnya telah dilakukan oleh Wisnu Dwi Jatmiko, Satrio Agung Wicaksono, dan Aditya Rachmadi dengan penelitiannya yang berjudul “Optimasi Query Pada Sistem Basis Data Terdistribusi Menggunakan Metode Dynamic Approach” (Jatmiko, Wicaksono, Rachmadi, 2014). Pada penelitian tersebut dilakukan optimasi pada 3 *query* yang berbeda yang menghasilkan peningkatan kecepatan sebesar 145.87% untuk *query* 1 dengan data yang diproses sejumlah 1.393.735 data kemudian peningkatan kecepatan sebesar 264.81% untuk *query* 2 dengan jumlah data yang diproses sejumlah 47.144 data, dan *query* 3 dengan peningkatan kecepatan sebesar 2015.96% dengan total data yang diproses sebanyak 1.393.763 data.

Penelitian selanjutnya mengenai optimasi *query* dilakukan oleh Mohamad Syauqi Reza, Yusi Tyroni M., dan Satrio Agung W. dalam penelitiannya yang berjudul “Optimasi Query Pada Sistem Basis Data Terdistribusi Dengan Metode Semijoin Based Approach (SBA)” (Reza, Yusi, Satrio, 2013) pada penelitian tersebut dilakukan optimasi pada 3 *query* dengan peningkatan kecepatan sebesar 1259% untuk *query* 1 dan peningkatan kecepatan sebesar 816% untuk *query* 2 dengan total data yang dieksekusi sebesar 1.393.735 data, dan peningkatan kecepatan sebesar 648% untuk *query* 3 dengan total data yang dieksekusi sebesar 3.393 data.

Penelitian mengenai optimasi *query* juga dilakukan oleh Manahan Siallagan, Mira Kania Sabariah, dan Malanita Sontya dengan penelitiannya yang berjudul “Optimasi Query Database Menggunakan Algoritma Genetik” (Siallagan, Sabariah, Sontya, 2008), Pada penelitian ini tidak

ditunjukkan peningkatan kecepatan dibandingkan dengan *query* konvensional, akan tetapi membandingkan dua metode *crossover* dalam algoritma genetik yaitu *CHUNK Crossover* dan *M2S Crossover* untuk mengetahui metode mana yang lebih baik, hasil dari penelitian ini didapatkan bahwa metode terbaik untuk optimasi *query database* yang dilakukan pada percobaan tersebut adalah metode *M2S Crossover*.

Membandingkan hasil dari ketiga penelitian sebelumnya, pada penelitian kali ini peneliti melakukan optimasi *query* dengan menerapkan algoritma *INGRES* (Jatmiko, Wicaksono, Rachmadi, 2014) dikarenakan dilihat dari hasilnya pada penelitian ini ditemukan peningkatan kecepatan yang paling signifikan dibandingkan dengan penelitian yang lain, selain itu pada penelitian tersebut menggunakan database dengan jumlah *data record* yang besar sesuai dengan objek penelitian kali ini pada database Lumbung Data Pendidikan Jawa Tengah. Perbedaan penelitian kali ini dan sebelumnya adalah pada penelitian ini menggunakan objek database Lumbung Data Pendidikan Jawa Tengah yang mana hanya menggunakan sistem basis data tersentralisasi berbeda dengan penelitian sebelumnya yang menggunakan sistem database terdistribusi. Data dari portal Pendidikan Jawa Tengah yang digunakan untuk penelitian ini adalah data pada semester ganjil tahun ajaran 2016/2017 sesuai dengan aslinya.

1.2. Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah bagaimana melakukan optimasi *query* pada database Lumbung Data Pendidikan Jawa Tengah dengan menggunakan algoritma *INGRES* untuk mengoptimalkan portal *web backbone* Dapodik Jawa Tengah ?

2. METODOLOGI

2.1 Objek Penelitian

Objek penelitian yang digunakan adalah *database* Portal Lumbung Data Pendidikan Jawa Tengah dengan data pada semester ganjil tahun ajaran 2016/2017

Tabel 1. Tabel Data Penelitian

No	Nama Tabel	Jml Atribut	Jml Data	
1	mst_wilayah	13	97.410	46.442.378
2	Sekolah	42	87.347	
3	ptk.vld_master	4	399.174	
4	Ptk	56	682.706	
5	ptk_terdaftar	26	2.791.346	
6	jenis_ptk	16	28	
7	peserta_didik	59	10.281.252	
8	pd.vld_master	16	5.293.404	
9	sarana	14	2.906.377	
10	sarana_longitudinal	8	15.601.940	
11	jenis_sarana	9	6.167	
12	prasarana	40	595.711	
13	prasarana_longitudinal	54	3.422.899	
14	jenis_prasarana	6	54	
15	rombongan_belajar	17	1.796.934	
16	agama	6	8	
17	tingkat_pendidikan	8	22	
18	prestasi	13	145.783	
19	tingkat_prestasi	6	7	

20	jenis_prestasi	6		4
21	beasiswa_peserta_didik	11		363.765
22	jenis_beasiswa	9		7
23	sumber_dana	6		27
24	jenjang_pendidikan	8		4
25	bidang_studi	15		1.784
26	rwy_pend_formal	18		1.914.638
27	gelar_akademik	8		285
28	penghargaan	12		53.274
29	tingkat_penghargaan	6		7
30	jenis_penghargaan	6		2
31	status_kepegawaian	6		12

2.2 Teknik Analisis Data

Tahapan analisa yang dilakukan pada penelitian ini antara lain:

1. Memperoleh data berupa tabel-tabel dalam objek penelitian dan *query* yang akan dioptimasi.
2. *Query* konvensional akan diuji dan dicatat hasilnya.
3. *Query* konvensional dioptimasi menggunakan algoritma *INGRES*.
4. Setelah *query* dioptimasi kemudian diuji dan dicatat hasilnya dan selanjutnya dibandingkan kecepatan pemrosesannya dengan *query* konvensional.

2.3 Algoritma INGRES

INGRES menggunakan metode optimasi *query* dinamis yang secara rekursif melakukan penguraian *query* untuk mendapatkan *query* yang lebih kecil dan kemudian dieksekusi secara berurutan (Özsu, Valduriez, 2011). Mula-mula *query* didekomposisi menjadi urutan *query* yang memiliki relasi unik, kemudian setiap monorelasi *query* yang dihasilkan diproses dengan pemilihan metode akses yang terbaik yaitu dengan *index*. Algoritma *INGRES* akan mengeksekusi operasi *unary* yaitu operasi *query* yang hanya melibatkan satu relasi atau satu tabel saja dan kemudian memperkecil ukurannya untuk dieksekusi secara berurutan dengan operasi *binary* yaitu operasi *query* yang melibatkan lebih dari satu relasi atau lebih dari satu tabel, misalnya terdapat satu *query* q yang dapat diuraikan menjadi dua sub *query* q_1 dan q_2 , maka q_1 akan dieksekusi terlebih dahulu dan hasilnya digunakan untuk q_2 . Contoh lain bila diberikan *query* q dengan n -relasi, maka *query* q akan diuraikan menjadi n sub*query*, $q_1 \rightarrow q_2 \rightarrow q_3 \rightarrow \dots \rightarrow q_n$, dan kemudian setiap monorelasi akan dieksekusi sampai tidak ada lagi monorelasi dan menyisakan multirelasi akhir.

Contoh *query* yang menampilkan nama-nama pendidik dan tenaga kependidikan beserta nip yang mempunyai beasiswa prestasi:

Tabel 2. Ujicoba *Query*

Obyek <i>Query</i>	<i>Query</i>
Q :	<pre>SELECT ptk.nama, ptk.nip FROM ptk, beasiswa_ptk, jenis_beasiswa WHERE ptk.nip = beasiswa_ptk.nip AND beasiswa_ptk.jenis_beasiswa_id= jenis_beasiswa.jenis_beasiswa_id AND jenis_beasiswa.nama = "Prestasi"</pre>

Setelah dilakukan proses pemecahan dan eksekusi pada monorelasi *query* maka didapatkan *query* sebagai berikut:

Tabel 3. Hasil Monorelasi *Query*

Obyek <i>Query</i>	<i>Query</i>
Q1 :	SELECT jenis_basiswa.jenis_basiswa_id INTO #jb_siswa FROM jenis_basiswa WHERE jenis_basiswa.nama = "Prestasi"
Q' :	SELECT ptk.nama, ptk.nip FROM ptk, beasiswa_ptk, #jb_siswa WHERE ptk.nip = beasiswa_ptk.nip AND beasiswa_ptk.jenis_basiswa_id = #jb_siswa.jenis_basiswa_id

Selanjutnya hasil dari proses pemecahan yaitu Q' dapat menghasilkan *query-ruery* berikutnya sebagai berikut:

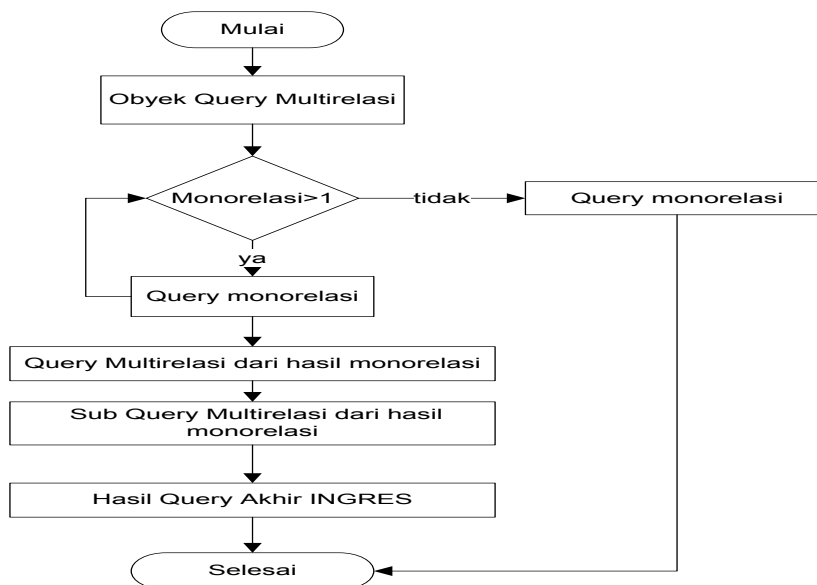
Tabel 4. Hasil Pemecahan *Query Q'*

Obyek <i>Query</i>	<i>Query</i>
Q2 :	SELECT beasiswa_ptk.nip INTO #b_siswa FROM beasiswa_ptk, #jb_siswa WHERE beasiswa_ptk.jenis_basiswa_id = #jb_siswa.jenis_basiswa_id
Q3 :	SELECT ptk.nama, ptk.nip FROM ptk, #b_siswa WHERE ptk.nip = #b_siswa.nip

Query Q3 merupakan *query* multirelasi yang sudah tidak dapat dilakukan pemecahan lagi, sehingga *query* Q3 merupakan *query* akhir yang paling optimal.

3. HASIL DAN PEMBAHASAN

3.1. Alur Diagram Flowchart

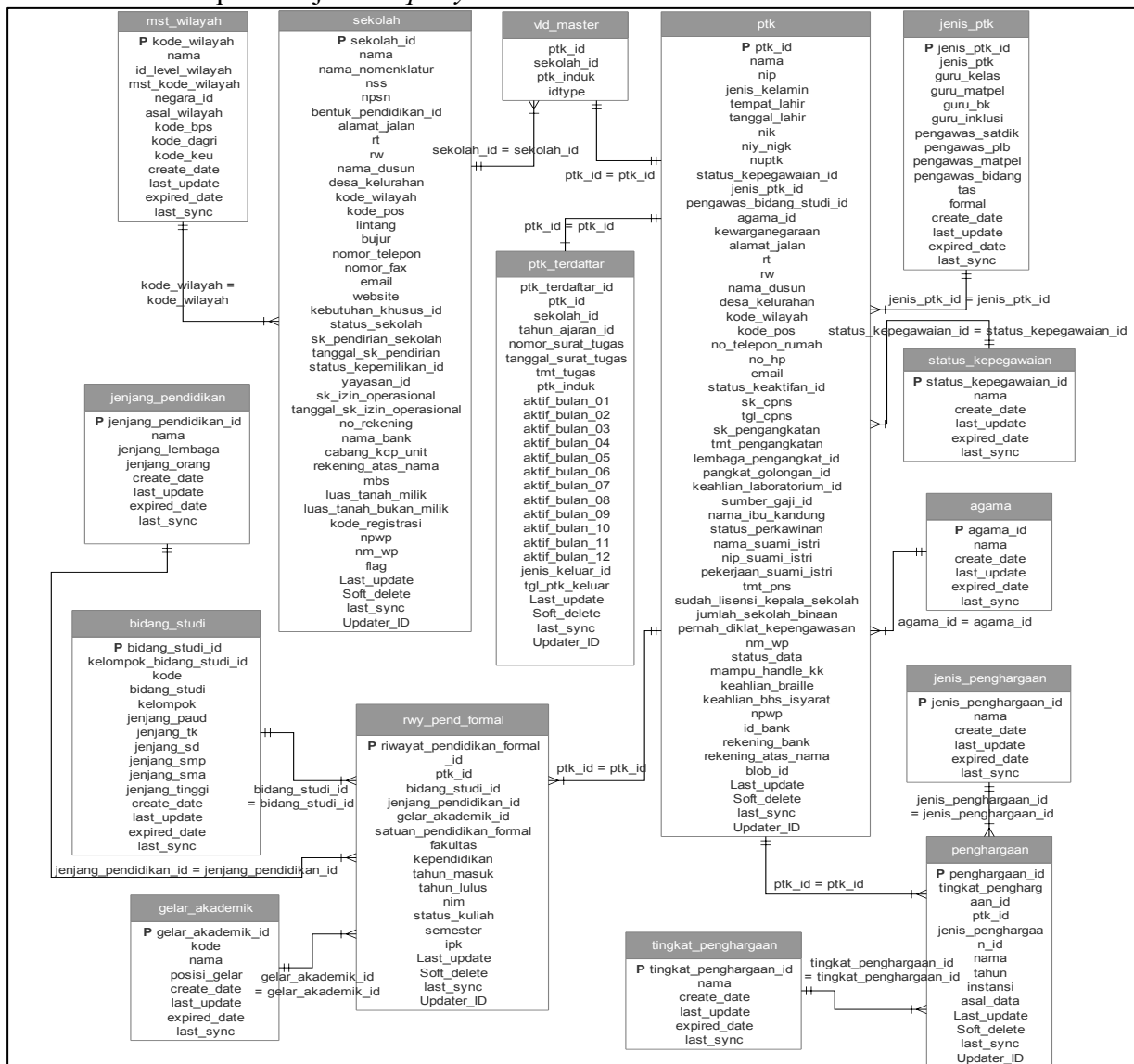


Gambar 1. Flowchart *Query* Algoritma INGRES

Obyek *query* multirelasi dicek untuk mengetahui ada monorelasi atau tidak, jika tidak ada maka *query* langsung proses, apabila terdapat monorelasi akan dipisah dan diproses terlebih dahulu sampai tidak ada lagi *query* monorelasi. Selanjutnya hasil dari proses *query* monorelasi diproses seperti obyek *query* awal, kemudian diproses kembali menjadi sub-sub *query* secara berurutan sampai menghasilkan *query* akhir yang optimal.

3.2. Pengujian Query

Pengaksesan database dan pengujian *query* menggunakan *tools Navicat 10.1.6*, berikut ini relasi tabel untuk proses uji coba *query* :



Gambar 2. Relasi Tabel

Uji *query* yaitu *QueryQ6* berisi perintah SQL untuk menampilkan data pendidik dan tenaga kependidikan (ptk) di Jawa Tengah yang memiliki jenjang pendidikan S1 dengan IPK di atas 3 dan mempunyai penghargaan tingkat nasional, adapun atribut yang ditampilkan oleh *queryQ6* berupa: nama kota/kabupaten, nama kecamatan, nama sekolah, nama ptk, nip, jenis kelamin, tempat lahir, tanggal lahir, nik, nuptk, status kepegawaian, jenis ptk, agama, jenjang pendidikan, bidang studi, gelar akademik, tahun masuk pendidikan, tahun lulus pendidikan, ipk, penghargaan, tahun_penghargaan, jenis penghargaan, tingkat penghargaan, dan instansi pemberi penghargaan.

QueryQ6 mempunyai 15 tabel yang saling berelasi, diantaranya adalah: tabel ptk, tabel ptk_terdaftar, tabel ptk.vld_master, tabel sekolah, tabel mst_wilayah, tabel status kepegawaian, tabel jenis_ptk, tabel agama, tabel rwy_pend_formal, tabel jenjang_pendidikan, tabel bidang_studi, tabel gelar_akademik, tabel penghargaan, tabel jenis_penghargaan, dan tabel tingkat_penghargaan. Berikut merupakan proses pada *queryQ6*:

```

SELECT kotakab.nama AS kotakab, kec.nama AS kecamatan, skh.nama AS nama_sekolah,
ptk.nama AS nama_ptk, ptk.nip, ptk.jenis_kelamin, ptk.tempat_lahir, ptk.tanggal_lahir,
ptk.nik, ptk.nuptk, skep.nama AS status_kepegawaian, jptk.jenis_ptk,
agm.nama AS agama, jp.nama AS jenjang_pendidikan, bs.bidang_studi, ga.nama AS
gelar_akademik, rpend.tahun_masuk, rpend.tahun_lulus, rpend.ipk,
penghargaan.nama AS penghargaan, penghargaan.tahun, jpeng.nama AS
jenis_penghargaan, tpeng.nama AS tingkat_penghargaan, penghargaan.instansi
FROM ptk
JOIN ptk.vld_master AS vld ON vld.ptk_id = ptk.ptk_id
JOIN sekolah AS skh ON skh.sekolah_id = vld.sekolah_id
JOIN ref.mst_wilayah AS kec ON kec.kode_wilayah = skh.kode_wilayah
JOIN ref.mst_wilayah AS kotakab ON kotakab.kode_wilayah =
LEFT(skh.kode_wilayah,4)+'00'
JOIN ref.status_kepegawaian AS skep ON skep.status_kepegawaian_id =
ptk.status_kepegawaian_id
JOIN ref.jenis_ptk AS jptk ON jptk.jenis_ptk_id = ptk.jenis_ptk_id
JOIN ref.agama AS agm ON agm.agama_id = ptk.agama_id
JOIN rwy_pend_formal AS rpend ON rpend.ptk_id = ptk.ptk_id
JOIN ref.jenjang_pendidikan AS jp ON jp.jenjang_pendidikan_id =
rpend.jenjang_pendidikan_id
JOIN ref.bidang_studi AS bs ON bs.bidang_studi_id = rpend.bidang_studi_id
JOIN ref.gelar_akademik AS ga ON ga.gelar_akademik_id = rpend.gelar_akademik_id
JOIN penghargaan ON penghargaan.ptk_id = ptk.ptk_id
JOIN ref.jenis_penghargaan AS jpeng ON jpeng.jenis_penghargaan_id =
penghargaan.jenis_penghargaan_id
JOIN ref.tingkat_penghargaan AS tpeng ON tpeng.tingkat_penghargaan_id =
penghargaan.tingkat_penghargaan_id
WHERE ptk.Soft_delete = 0
AND vld.idtype > 7 AND vld.idtype < 20 AND rpend.Soft_delete = 0
AND rpend.ipk > 3 AND jp.nama = 'S1' AND penghargaan.Soft_delete = 0
AND tpeng.nama = 'Nasional' AND skh.Soft_delete = 0
ORDER BY kotakab.kode_wilayah, kec.kode_wilayah, skh.npsn, ptk.nip

```

Gambar 3. Query pada Obyek QueryQ6

Semua monorelasi query dalam query Q6 dipisah dan diproses terlebih dahulu, hasil dari pemisahan monorelasi query dari query Q6 adalah sebagai berikut:

Query Q61:

```

SELECT ptk_id, nama, nip, jenis_kelamin, tempat_lahir, tanggal_lahir,
nik, nuptk, status_kepegawaian_id, jenis_ptk_id, agama_id INTO #temp_ptk
FROM ptk WHERE Soft_delete = 0

```

Query Q62:

```

SELECT ptk_id, sekolah_id INTO #temp_vld_master
FROM ptk.vld_master WHERE idtype > 7 AND idtype < 20

```

Query Q63:

```

SELECT ptk_id, tahun_masuk, tahun_lulus, ipk, jenjang_pendidikan_id,
bidang_studi_id, gelar_akademik_id INTO #temp_rwy_pend_formal
FROM rwy_pend_formal WHERE Soft_delete = 0 AND ipk > 3

```

Query Q64:

```

SELECT jenjang_pendidikan_id, nama INTO #temp_jenjang_pendidikan
FROM ref.jenjang_pendidikan WHERE nama = 'S1'

```

Query Q65:

```

SELECT ptk_id, tingkat_penghargaan_id, jenis_penghargaan_id, nama, tahun,
instansi INTO #temp_penghargaan
FROM penghargaan WHERE Soft_delete = 0

```

Query Q66:

```

SELECT tingkat_penghargaan_id, nama INTO #temp_tingkat_penghargaan
FROM ref.tingkat_penghargaan WHERE nama = 'Nasional'

```

Query Q67:

```

SELECT sekolah_id, nama, kode_wilayah, npsn INTO #temp_sekolah
FROM sekolah WHERE Soft_delete = 0

```

Gambar 4. Pemisahan Monorelasi Query pada Obyek QueryQ6

Hasil dari setiap pemrosesan *query* monorelasi dikembalikan ke *query* awal, hasilnya sebagai berikut:

Query Q6':

```

SELECT kotakab.nama AS kotakab, kec.nama AS kecamatan, skh.nama AS
nama_sekolah, ptk.nama AS nama_ptk, ptk.nip, ptk.jenis_kelamin,
ptk.tempat_lahir, ptk.tanggal_lahir, ptk.nik, ptk.nuptk, skeep.nama AS
status_kepegawaian, jptk.jenis_ptk,
    agm.nama AS agama, jp.nama AS jenjang_pendidikan, bs.bidang_studi,
ga.nama AS gelar_akademik, rpend.tahun_masuk, rpend.tahun_lulus, rpend.ipk,
    penghargaan.nama AS penghargaan, penghargaan.tahun, jpeng.nama AS
jenis_penghargaan, tpeng.nama AS tingkat_penghargaan, penghargaan.instansi
FROM #temp_ptk AS ptk
JOIN #temp_vld_master AS vld ON vld.ptk_id = ptk.ptk_id
JOIN #temp_sekolah AS skh ON skh.sekolah_id = vld.sekolah_id
JOIN ref.mst_wilayah AS kec ON kec.kode_wilayah = skh.kode_wilayah
JOIN ref.mst_wilayah AS kotakab ON kotakab.kode_wilayah =
LEFT(skh.kode_wilayah,4)+'00'
JOIN ref.status_kepegawaian AS skeep ON skeep.status_kepegawaian_id =
ptk.status_kepegawaian_id
JOIN ref.jenis_ptk AS jptk ON jptk.jenis_ptk_id = ptk.jenis_ptk_id
JOIN ref.agama AS agm ON agm.agama_id = ptk.agama_id
JOIN #temp_rwy_pend_formal AS rpend ON rpend.ptk_id = ptk.ptk_id
JOIN #temp_jenjang_pendidikan AS jp ON jp.jenjang_pendidikan_id =
rpend.jenjang_pendidikan_id
JOIN ref.bidang_studi AS bs ON bs.bidang_studi_id = rpend.bidang_studi_id
JOIN ref.gelar_akademik AS ga ON ga.gelar_akademik_id =
rpend.gelar_akademik_id
JOIN #temp_penghargaan ON penghargaan.ptk_id = ptk.ptk_id
JOIN ref.jenis_penghargaan AS jpeng ON jpeng.jenis_penghargaan_id =
penghargaan.jenis_penghargaan_id
JOIN #temp_tingkat_penghargaan AS tpeng ON tpeng.tingkat_penghargaan_id =
penghargaan.tingkat_penghargaan_id
ORDER BY kotakab.kode_wilayah, kec.kode_wilayah, skh.npsn, ptk.nip

```

Gambar 5. *Query* pada Obyek *QueryQ6'*

Setelah tidak ada lagi monorelasi *query*, selanjutnya *queryQ6'* diuraikan lagi menjadi urutan sub-sub *query* dan kemudian dieksekusi secara berurutan.

Query Q68:

```

SELECT pg.ptk_id, tp.nama AS tingkat_penghargaan,
pg.jenis_penghargaan_id, pg.penghargaan, pg.tahun, pg.instansi INTO
#temp_penghargaan1
FROM #temp_penghargaan AS pg, #temp_tingkat_penghargaan AS tp
WHERE tp.tingkat_penghargaan_id = pg.tingkat_penghargaan_id

```

Query Q69:

```

SELECT pg.ptk_id, pg.tingkat_penghargaan, jp.nama AS jenis_penghargaan,
pg.penghargaan, pg.tahun, pg.instansi INTO #temp_penghargaan2
FROM #temp_penghargaan1 AS pg, ref.jenis_penghargaan AS jp
WHERE jp.jenis_penghargaan_id = pg.jenis_penghargaan_id

```

Query Q610:

```

SELECT rp.ptk_id, rp.tahun_masuk, rp.tahun_lulus, rp.ipk, jp.nama as
jenjang_pendidikan, rp.bidang_studi_id, rp.gelar_akademik_id INTO
#temp_rwy_pend_formal1
FROM #temp_rwy_pend_formal AS rp, #temp_jenjang_pendidikan AS jp
WHERE jp.jenjang_pendidikan_id = rp.jenjang_pendidikan_id

```

Query Q611:

```

SELECT rp.ptk_id, rp.tahun_masuk, rp.tahun_lulus, rp.ipk,
rp.jenjang_pendidikan, bs.bidang_studi, rp.gelar_akademik_id INTO
#temp_rwy_pend_formal2
FROM #temp_rwy_pend_formal1 AS rp, ref.bidang_studi AS bs
WHERE bs.bidang_studi_id = rp.bidang_studi_id

```


Query Q612:

```
SELECT rp.ptk_id, rp.tahun_masuk, rp.tahun_lulus, rp.ipk,
rp.jenjang_pendidikan, rp.bidang_studi, ga.nama AS gelar_akademik INTO
#temp_rwy_pend_formal3
FROM #temp_rwy_pend_formal2 AS rp, ref.gelar_akademik AS ga
WHERE ga.gelar_akademik_id = rp.gelar_akademik_id
```

Query Q613:

```
SELECT #temp_penghargaan2.*, tahun_masuk, tahun_lulus, ipk,
jenjang_pendidikan, bidang_studi, gelar_akademik INTO #pendidikan_penghargaan
FROM #temp_penghargaan2, #temp_rwy_pend_formal3
WHERE #temp_penghargaan2.ptk_id = #temp_rwy_pend_formal3.ptk_id
```

Query Q614:

```
SELECT ptk.*, tahun_masuk, tahun_lulus, ipk, jenjang_pendidikan,
bidang_studi, gelar_akademik, tingkat_penghargaan, jenis_penghargaan,
penghargaan, tahun, instansi
INTO #temp_ptk_pp
FROM #temp_ptk AS ptk, #pendidikan_penghargaan AS pp
WHERE ptk.ptk_id = pp.ptk_id
```

Query Q615:

```
SELECT ptk.*, vld.sekolah_id
INTO #temp_ptk1
FROM #temp_ptk_pp AS ptk, #temp_vld_master AS vld
WHERE ptk.ptk_id = vld.ptk_id
```

Query Q616:

```
SELECT ptk.*, skh.nama AS nama_sekolah, skh.npsn, skh.kode_wilayah
INTO #temp_ptk2
FROM #temp_ptk1 AS ptk, #temp_sekolah AS skh
WHERE skh.sekolah_id = ptk.sekolah_id
```

Query Q617:

```
SELECT ptk.*, kotakab.nama AS kotakab, kec.nama AS kecamatan,
kotakab.kode_wilayah AS kode_wilayah1, kec.kode_wilayah AS kode_wilayah2
INTO #temp_ptk3
FROM #temp_ptk2 AS ptk, ref.mst_wilayah AS kotakab, ref.mst_wilayah AS
kec
WHERE LEFT(ptk.kode_wilayah,4)+'00' = kotakab.kode_wilayah
AND ptk.kode_wilayah = kec.kode_wilayah
```

Query Q618:

```
SELECT ptk.*, sk.nama AS status_kepegawaian
INTO #temp_ptk4
FROM #temp_ptk3 AS ptk, ref.status_kepegawaian AS sk
WHERE sk.status_kepegawaian_id = ptk.status_kepegawaian_id
```

Query Q619:

```
SELECT ptk.*, jp.jenis_ptk
INTO #temp_ptk5
FROM #temp_ptk4 AS ptk, ref.jenis_ptk AS jp
WHERE jp.jenis_ptk_id = ptk.jenis_ptk_id
```

Query Q620 (hasil akhir):

```
SELECT kotakab, kecamatan, nama_sekolah, nama_ptk, nip, jenis_kelamin,
tempat_lahir, tanggal_lahir, nik, nuptk, status_kepegawaian, jenis_ptk,
agama.nama AS agama, jenjang_pendidikan, bidang_studi, gelar_akademik,
tahun_masuk, tahun_lulus, ipk,
penghargaan, tahun, jenis_penghargaan, tingkat_penghargaan, instansi
FROM #temp_ptk5, ref.agama
WHERE agama.agama_id = #temp_ptk5.agama_id
ORDER BY kode_wilayah1, kode_wilayah2, npsn, nip
```

Gambar 6. Sub Query pada Obyek QueryQ6'

3.3. Hasil Uji Coba

Hasil dari penelitian ini dibagi menjadi 2 bagian, yaitu uji coba eksekusi pada *query* konvensional dan hasil uji coba eksekusi pada *query* yang sudah dioptimasi menggunakan algoritma *INGRES*.

Tabel 5. Hasil uji coba eksekusi pada *query* konvensional

Objek <i>Query</i>	Percobaan										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
Q1	0.529	0.511	0.505	0.520	0.502	0.511	0.512	0.523	0.502	0.669	0.5365
Q2	6.238	3.489	2.681	2.833	2.669	2.699	2.513	2.639	2.687	2.621	3.1069
Q3	12.169	1.068	1.061	1.391	1.238	1.068	1.063	1.091	1.454	1.205	2.2808
Q4	14.058	0.752	0.69	0.687	0.672	0.679	0.652	0.683	0.636	0.698	2.0207
Q5	21.677	4.957	3.638	2.9	3.403	6.091	2.936	4.167	3.718	3.854	5.7341
Q6	11.758	1.183	1.438	1.236	1.712	1.429	1.787	1.343	1.226	1.226	2.4338
Rata-rata:											2.685467

Tabel 6. Hasil uji coba eksekusi pada *query* yang sudah dioptimasi menggunakan *algoritma INGRES*.

Objek <i>Query</i>	Percobaan										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
Q1	9.886	0.172	0.185	0.186	0.214	0.249	0.172	0.169	0.193	0.171	1.1597
Q2	40.818	1.259	1.301	1.307	1.28	1.314	1.324	1.315	1.307	1.31	5.2535
Q3	19.478	0.626	0.6	0.599	0.583	0.594	1.078	0.578	0.605	0.595	2.5336
Q4	6.031	0.26	0.27	0.269	0.261	0.276	0.326	0.247	0.255	0.245	0.844
Q5	52.593	3.537	4.42	2.828	3.438	2.333	2.866	2.387	2.282	3.985	8.0669
Q6	10.614	0.58	0.645	0.923	0.863	0.81	0.712	0.623	1.331	0.903	1.8004
Rata-rata:											3.27635

3.4. Pembahasan

Berdasarkan hasil pengujian di atas dapat dilihat bahwa hanya dua *query* dari *query* yang sudah dioptimasi yang dapat menghasilkan waktu eksekusi yang lebih cepat dibandingkan dengan hasil dari eksekusi *query* konvensional.

Tabel 7. Perbandingan hasil kecepatan eksekusi *query*.

Q	Konvensional	Optimized	Kesimpulan:	
1	0.5365	1.1597	216%	Lebih Lambat
2	3.1069	5.2535	169%	Lebih Lambat
3	2.2808	2.5336	111%	Lebih Lambat
4	2.0207	0.844	42%	Lebih Cepat
5	5.7341	8.0669	141%	Lebih Lambat

Namun apabila diteliti hasil dari 10 kali percobaan tersebut ternyata dapat dilihat bahwa eksekusi *query* pada kali pertama memerlukan waktu yang lebih banyak dibandingkan proses eksekusi setelahnya, dan hasil eksekusi pertama pada *query* yang sudah dioptimasi ternyata memakan waktu jauh lebih banyak dibandingkan hasil eksekusi pertama pada *query* konvensional, akan tetapi proses eksekusi berikutnya *query* yang sudah dioptimasi memakan waktu yang lebih singkat daripada *query*

konvensional. berikut merupakan hasil rata-rata kecepatan akses apabila percobaan pertama tidak dimasukkan:

Tabel 8. Hasil uji coba eksekusi pada *query* konvensional tanpa memasukkan percobaan pertama.

Objek										Rata-rata
<i>Query</i>	2	3	4	5	6	7	8	9	10	
Q1	0.511	0.505	0.520	0.502	0.511	0.512	0.523	0.502	0.669	0.5365
Q2	3.489	2.681	2.833	2.669	2.699	2.513	2.639	2.687	2.621	2.759
Q3	1.068	1.061	1.391	1.238	1.068	1.063	1.091	1.454	1.205	1.182111
Q4	0.752	0.69	0.687	0.672	0.679	0.652	0.683	0.636	0.698	0.683222
Q5	4.957	3.638	2.9	3.403	6.091	2.936	4.167	3.718	3.854	3.962667
Q6	1.183	1.438	1.236	1.712	1.429	1.787	1.343	1.226	1.226	1.397778
Rata-rata:										1.753546

Tabel 9. Hasil uji coba eksekusi pada *query* yang sudah dioptimasi tanpa memasukkan percobaan pertama.

Objek										Rata-rata
<i>Query</i>	2	3	4	5	6	7	8	9	10	
Q1	0.172	0.185	0.186	0.214	0.249	0.172	0.169	0.193	0.171	0.190111
Q2	1.259	1.301	1.307	1.28	1.314	1.324	1.315	1.307	1.31	1.301889
Q3	0.626	0.6	0.599	0.583	0.594	1.078	0.578	0.605	0.595	0.650889
Q4	0.26	0.27	0.269	0.261	0.276	0.326	0.247	0.255	0.245	0.267667
Q5	3.537	4.42	2.828	3.438	2.333	2.866	2.387	2.282	3.985	3.119556
Q6	0.58	0.645	0.923	0.863	0.81	0.712	0.623	1.331	0.903	0.821111
Rata-rata:										1.058537

Tabel 10. Perbandingan hasil kecepatan eksekusi *query* tanpa percobaan pertama.

Q	Konvensional	Optimized	Kesimpulan:	
1	0.5365	0.190111111	35%	Lebih Cepat
2	2.759	1.301888889	47%	Lebih Cepat
3	1.182111111	0.650888889	55%	Lebih Cepat
4	0.683222222	0.267666667	39%	Lebih Cepat
5	3.962666667	3.119555556	79%	Lebih Cepat
6	1.397777778	0.821111111	59%	Lebih Cepat

4. SIMPULAN

Berdasarkan percobaan yang dilakukan dengan mengoptimasi *query* menggunakan algoritma *INGRES* pada basis data portal Lumbung Data Pendidikan Jawa Tengah hasilnya pada pemrosesan *query* pertama kali yang menerapkan algoritma *INGRES* justru lebih memakan banyak waktu dibandingkan dengan metode konvensional sehingga mengakibatkan penurunan kecepatan hingga 122% dengan rata-rata hasil dari algoritma *INGRES* selama 3.276350 sec, dan rata-rata hasil dari *query* konvensional selama 2.685467 sec. Peningkatan kecepatan pemrosesan *query* dengan algoritma *INGRES* terjadi pada percobaan ke 2 dan seterusnya dengan rata-rata hasil dari algoritma *INGRES* selama 1.058537 sec, dan rata-rata hasil dari *query* konvensional selama 1.753546 sec. Dengan kata lain hasilnya 60% lebih cepat dibandingkan dengan *query* konvensional.

DAFTAR PUSTAKA

- A. Silberschatz, H. F. Korth, and S. Sudarshan. 2011. *Database System Concepts - 6th. ed.*, vol. 4.
- DAPODIK. 2014. *Tentang Program Data Pokok Pendidikan (DAPODIK)*, [Online]. Available: <http://dapodik.org/tentang-dapodik/>. [Accessed: 24-Maret-2018].
- DAPODIK Jawa Tengah. 2018. *Lambung Data Pendidikan Provinsi Jawa Tengah*, [Online]. Available: <http://dapodik.pdkjateng.go.id/tentang> . [Accessed: 24-Maret-2018].
- Melany Mustika Dewi , Novandi Rezeki. 2017. *Analisis Perbandingan Optimasi Query Nasted Join dan Hash Join pada MySQL Server*, CSRID Journal, Vol. 9 No.1, Medan.
- M. K. Adyanata Lubis, S.Kom. 2014. *Basis Data Dasar*, Yogyakarta: Deepublish.
- M. S. Manahan Siallagan, Mira Kania Sabariah. 2018. *Optimasi Query Database Menggunakan Algoritma Genetik. ISSN 1907-5022 OPTIMASI*, Vol. 2008. Yogyakarta: SNATI.
- M. T. Özsu and P. Valduriez. 2011. *Principles of Distributed Database Systems*, vol. 12, New York: Springer.
- S. A. W. Mohamad Syauqi Reza, Yusi Tyroni M. 2013. *Optimasi Query Pada Sistem Basis Data Terdistribusi Dengan Metode Semijoin Based Approach (SBA)*. *Repos. J. Mhs. PTIIK UB*, Vol. 1.
- S. S. Wisnu Dwi Jatmiko, Satrio Agung Wicaksono, S.Kom, M.Kom, Aditya Rachmadi. 2014. *Optimasi Query Pada Sistem Basis Data Terdistribusi Menggunakan Metode Dynamic Approach*. *Repos. J. Mhs. PTIIK UB*, vol. 3, pp. 1–9.
- Syah Askar Akbar, Ajib Susanto. 2012. *Teknik Proteksi Terhadap Serangan Sql Injection Menggunakan Konsep Amnesia Pada Aplikasi Web*. *Journal of Intellegent Systems and Business Intellegence*, Vol. 1, No. 2, Semarang : COWISBI.
- T. Connolly and C. Begg. 2005. *Database Systems: A Practical Approach to Design, Implementation, and Management*. pp. 1–1425.