

Performance Tuning Oracle 11g Database Melalui Inisial Paramater, Structure Database dan SQL Tuning (Studi Pada ERP SISFORBUN Dana Pensiun Perkebunan (DAPENBUN))

Oracle 11g Database Performance Tuning Through Initial Parameters, Database Structure, and SQL Tuning (Study on ERP SISFORBUN Dana Pensiun Perkebunan (DAPENBUN))

Samidi¹, Hariyanto²

^{1,2} Program Studi Magister Ilmu Komputer, Fakultas Teknologi Informasi, Universitas Budi Luhur, Jakarta Selatan

E-mail: ¹samidi@budiluhur.ac.id , ²2211601097@student.budiluhur.ac.id

Abstrak

Dana Pensiun Perkebunan (DAPENBUN) sebagai pengelola manfaat pensiun bagi karyawan PTPN seluruh Indonesia beserta lembaga afiliasi dengan jumlah peserta per 31 Desember 2021 sebanyak 284.934 orang. Aplikasi SISFORBUN ini digunakan untuk memproses manfaat pensiun bagi seluruh peserta. Aplikasi berbasis web ini menggunakan database Oracle 11g dan sudah digunakan sejak tahun 2013, dengan seiring berjalanya waktu perkembangan data semakin banyak dengan jumlah record terbesar dalam satu table sebesar 26.696.667 record data, sehingga performa proses data semakin menurun. Atas dasar permasalahan tersebut perlu dilakukan penelitian untuk meningkat performa dari Aplikasi. Penelitian ini menggunakan metode SQL Tuning, Structuring Object dan Initial Parameter Database. Setelah dilakukan pengujian dengan melakukan proses pembayaran Manfaat Pensiun pada laporan manajemen nomor 18 (LM18), penulis dapat menyimpulkan bahwa setelah dilakukan optimasi, waktu yang diperlukan untuk proses pembayaran Manfaat Pensiun (LM18) menjadi lebih cepat dibandingkan sebelum dilakukan optimasi.

Kata kunci: Database, Inisial Parameter, Struktur Database, Oracle 11g, SQL Tuning, Dana Pensiun

Abstract

Dana Pensiun Perkebunan (DAPENBUN) is the management of pension benefits for PTPN employees throughout Indonesia and affiliated institutions with 284,934 participants as of December 31, 2021. This SISFORBUN application is used to process pension benefits for all participants. This web-based application uses the Oracle 11g database and has been in use since 2013, with the development of more and more data with the largest number of records in one table of 26,696,667 data records, data processing performance is decreasing. On the basis of these problems, research needs to be done to improve the performance of the application. This research uses SQL Tuning, Structuring Objects, and Initial Parameter Database methods. After testing by processing the payment of Pension Benefits (LM18 report), the author can conclude that after optimization, the time required to process the payment of Pension Benefits (LM18 report) is faster than before optimization.

Keywords: Database, Initial Parameter, Structure Database, Oracle 11g, SQL Tuning, Pension Fund.

1. PENDAHULUAN

Keberadaan sistem informasi di suatu perusahaan atau institusi sangatlah penting, sistem informasi membawa berbagai keuntungan, misalnya dalam hal kecepatan dan ketepatan informasi, percepatan pengambilan keputusan, pengurangan birokrasi, optimalisasi sumber daya dan operasional, penghematan biaya. Untuk mengembangkan Relational Database Management System (RDMS) yang optimal, perlu dilakukan analisis dan perancangan database yang optimal untuk meningkatkan efisiensi dan efektifitas kinerja database perusahaan. RDBMS adalah sistem manajemen basis data relasional yang menyimpan data dalam bentuk tabel[1]. Dan untuk mengambil informasi yang dibutuhkan dari data yang disimpan menggunakan SQL (Structure Query Language). SQL adalah syntax database yang komprehensif, yang berisikan statement untuk manipulasi data (Query, Update)[2], [3]

Dana Pensiun Perkebunan (DAPENBUN) sebagai pengelola manfaat pensiun bagi karyawan PTPN seluruh Indonesia beserta lembaga afiliasi dengan jumlah peserta per 31 Desember 2021 sebanyak 284.934 orang[4]. Sejak tahun 2013, DAPENBUN menggunakan teknologi informasi dalam melakukan pengelolaan pembayaran manfaat pensiun yang dinamakan Sistem Informasi Dana Pensiun Perkebunan (SISFORBUN). SISFORBUN adalah core system/Sistem Utama yang berbasis Web Based yang dibangun menggunakan bahasa pemrograman Java dengan Database Oracle 11g. SISFORBUN terdapat beberapa modul diantaranya modul Keuangan, Human Resources, Investasi dan modul Kepesertaan yang memiliki sub modul Peserta Aktif, Peserta Pasif, Mutasi Peserta, Pensiun Gugur, Pensiun Tunda, Mutasi Pemberi Kerja, Mutasi Unit kerja dan beberapa modul dan sub modul lainnya.

Dengan seiring berjalannya waktu dan penambahan jumlah transaksi yang semakin meningkat mengakibatkan terjadinya penurunan kinerja dari sebuah aplikasi hal ini ditandai dengan pengambilan data dan proses transaksi menjadi lambat dan beberapa kali ditemukan Service Oracle down, sehingga perlu dilakukan optimasi pada Database.

Salah satu cara yang dilakukan untuk optimasi sebuah database adalah Performance Tuning. Performance Tuning adalah kegiatan menganalisis dan meningkatkan kinerja sistem dalam DBMS[5][6]. Proses performance tuning ini meliputi perubahan struktur data, parameter system database, konfigurasi Sistem Operasi atau termasuk perangkat keras.

Pada penelitian sebelumnya Virmansyah et al[7] dengan menggunakan Database PostgreSQL, melakukan optimasi Database dengan metode Indexing dan partisi tabel dan mendapatkan hasil bahwa penggunaan Primary Key, Foreign Key, Index dan Partition table dianggap lebih baik respon timenya. Selanjutnya Tavares et al[8] melakukan pengujian dengan melakukan restrukturisasi kueri agar didapatkan optimalisasi kueri dengan basis data menggunakan MySQL, serta S. Indah Nurhafida melakukan penulisan terkait dengan Optimasi Query Database Web Scraping Pada Jurnal SINTA yang menggunakan database MySQL menghasilkan dengan menggunakan metode query index, load data yang diperlukan lebih singkat dibandingkan tanpa menggunakan metode index[9].

Dari beberapa penulisan yang ada, pengujian untuk meningkatkan performace aplikasi metode yang banyak dilakukan adalah dengan melakukan pengindeksan pada tabel. Pengindeksan menjadi teknik kunci untuk meningkatkan kinerja dalam database[10], namun untuk mendukung kueri yang efisien, mekanisme pengindeksan yang tepat harus ada[11].

Perbedaan tulisan ini dengan yang sebelumnya adalah pada penulisan ini menggunakan Database Oracle. Berdasarkan tulisan dari Wood et al[12] telah mengidentifikasi solusi potensial untuk performance diantaranya System Issue (Initial Parameter), Design Issue (Phsysical Layout) dan SQL Issues (Indexing), oleh karena itu penulis akan melakukan pengujian pontensial solusi yang sudah teridentifikasi tersebut sehingga diharapkan setelah penulisan ini, mendapatkan cara yang terbaik untuk meningkatkan performance dari aplikasi.

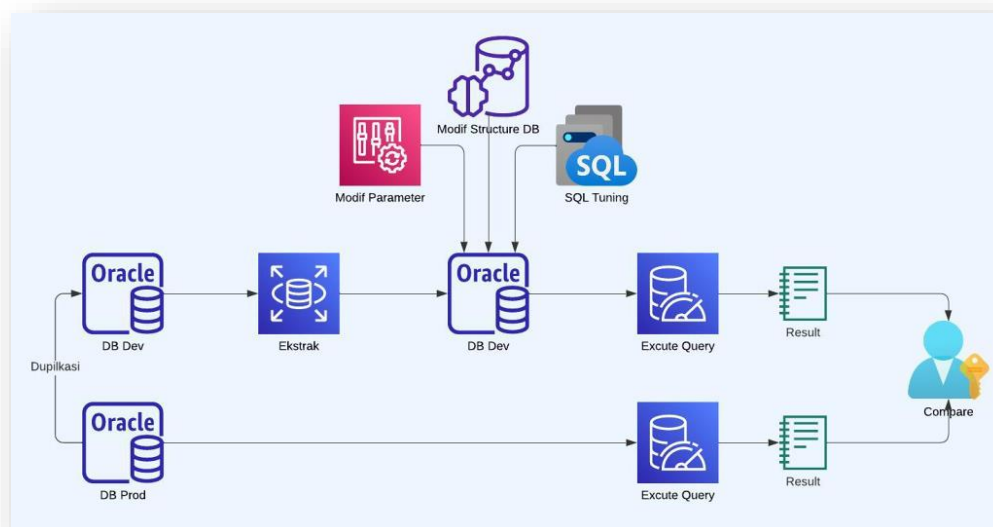
2. METODE PENELITIAN

Penelitian ini menggunakan metode eksperimental dengan memilih modul Pengajuan Pembayaran Manfaat Pensiun sebagai objectnya. Modul ini akan memproses berapa jumlah peserta pensiun dan nominal manfaat pensiun yang akan diterima oleh peserta Pensiun Normal, Pensiun Dipercepat, Pensiun Dipercepat (Ditunda), Pensiun Cacat, Pensiun Janda, Pensiun Anak dan Pensiun Tunda serta berapa jumlah peserta dan nominal Pengembalian Iuran, Pengembalian Sisa, Pensiun Sekaligus, Penerimaan Kembali dan Pensiun Gugur. Data yang dihasilkan akan menjadi sumber utama bagi Direksi untuk persetujuan pengeluaran dana untuk melakukan pembayaran Manfaat pensiun ke Peserta.

Pengujian dilakukan pada database Sisforbun versi development dengan menggunakan beberapa table dan jumlah record terbesar berisikan 26.696.667 record data. Dalam Pengujian ini penulis melakukan perubahan pada konfigurasi Parameter Database, melakukan modifikasi struktur Database dan melakukan beberapa perintah SQL pada table yang akan dilakukan uji coba serta mengukur waktu respon untuk menampilkan data dari perintah SQL. Untuk pengujian atau pengukuran waktu respon dilakukan dalam dua kondisi yang berbeda, yaitu menggunakan Database server production dan Database server development yang telah dilakukan Performance Tuning serta diasumsikan dalam kondisi tidak ada user lain yang mengakses ke server production.

Pada gambar 1, ditampilkan langkah-langkah yang dilakukan dalam penelitian ini adalah:

1. Melakukan duplikasi Database Production Menjadi Database Development.
2. Mengekstrak database development.
3. Melakukan modifikasi beberapa parameter di database development.
4. Melakukan perubahan struktur database development.
5. Melakukan SQL Tuning di database development
6. Menjalankan perintah SQL di database production dan database development dan catat waktu respon.
7. Membandingkan hasil test di database production dan database development.



Gambar 1: Tahapan Penelitian

Dalam melakukan penelitian dan pengujian. Perangkat yang digunakan oleh penulis berupa server HCI (Hyper Converged Infrastructure)-Development, dengan spesifikasi sistem operasi Windows Server 2008 R2 64-bit, Database Oracle 11g, Processor 16 Core, Ram 64 GB, tools pl-sql developer.

3. HASIL DAN PEMBAHASAN

Dalam rangka mendukung kegiatan operasional Dana Pensiun Perkebunan menggunakan ERP Sisforbun yang terdiri dari modul Peserta Aktif, Peserta Pasif, Pasar Modal dan Pasar Uang, Investasi Langsung, Human Resources, Keuangan dan masih banyak modul lainnya.

Pada penelitian kali ini adalah fokus kepada Database Sisforbun khususnya tabel yang sering dilakukan proses query, yaitu tabel mutasi golongan, table transaksi_mp, table mutasi golongan, table attestatie de vita dan tabel tarnsaksi iuran.

Tahapan yang akan dilakukan pengujian dalam penelitian ini adalah melakukan intial parameter database, Struktur Database dan SQL Tuning.

1. Initial Parameter Database

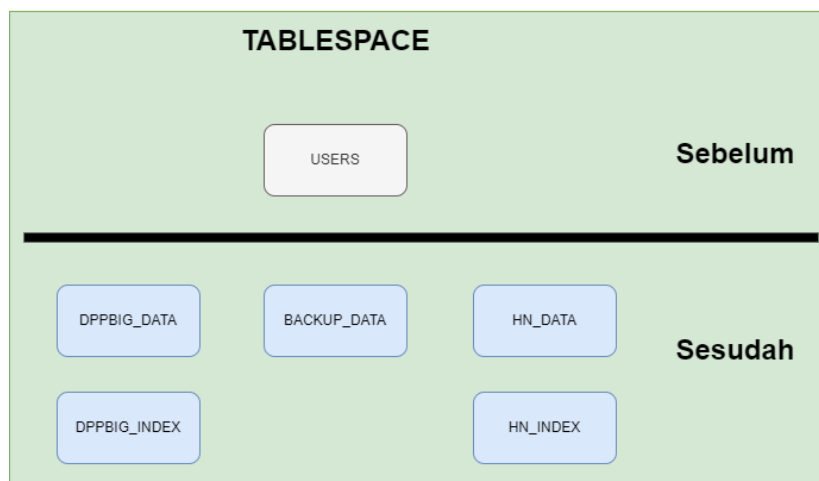
Dalam arsitektur Database Oracle bahwa optimasi memory merupakan kunci dari performance [13], sehingga hal pertama yang dilakukan untuk optimalisasi terhadap proses Database adalah alokasi memory dari proses yang akan dijalankan.

SEBELUM	SESUDAH	KETERANGAN
db_writer_processes = 2	db_writer_processes ==16	Penambahan background proses penulisan data reord(redolog Transaction) ke dalam tablespace
sga_max_size = 17179869184	sga_max_size = 21474836480	Penambahan akses data ke dalam memory
shared_pool_size=0	shared_pool_size= 10737418240	Ada beberapa proses library cache yang mereduca memory yang besar yang selama ini diterapkan dengan konsep AMM (automatic Memory Management)
Java_pool_size=0	Java_pool_size= 2147483648	Ada beberapa proses library cache yang mereduca memory yang besar yang selama ini diterapkan dengan konsep AMM (automatic Memory Management)

Gambar 2. Perbandingan Initial Parameter

2. Structuring Database

Melakukan reorganisasi tempat penyimpanan table dan index, awalnya hanya menggunakan 1 tablespace USERS menjadi beberapa tablespace sesuai dengan transaksi yang ada. Pemindahan object (table dan index) yang dilakukan dengan menambahkan tablespace dan konfigurasi logik[14]. Hal ini untuk mempermudah administrasi dan pengelolaan pertumbuhan database[15]



Gambar 3. Perbandingan Struktur Tablespace

	TABLESPACE_NAME	FILE_NAME		FREE_MB	ALLOCATED_MB	CAPACITY_MB
1	SYSTEM	D:\ORADATA\DPPPROD2\SYSTEM01.DBF	***	24447	10000	32767
2	SYSAUX	D:\ORADATA\DPPPROD2\SYSAUX01.DBF	***	31789	2000	32767
3	HN_INDEX	D:\ORADATA\DPPPROD2\HN_INDEX02.DBF	***	25723	10240	32767
4	BACKUP_DATA	D:\ORADATA\DPPPROD2\BACKUP_DATA01.DBF	***	32759	200	32767
5	DPPBIG_DATA	D:\ORADATA\DPPPROD2\DPPBIG_DATA01.DBF	***	20734	20480	32767
6	USERS	D:\ORADATA\DPPPROD2\USERS01.DBF	***	32598	177	32767
7	USERS	D:\ORADATA\DPPPROD2\USERS02.DBF	***	32582	200	32767
8	DPPBIG_INDEX	D:\ORADATA\DPPPROD2\DPPBIG_INDEX01.DBF	***	12686	20480	32767
9	UNDOTBS1	D:\ORADATA\DPPPROD2\UNDOTBS01.DBF	***	32669	4533	32767
10	HN_INDEX	D:\ORADATA\DPPPROD2\HN_INDEX01.DBF	***	1243	10240	10240
11	HN_DATA	D:\ORADATA\DPPPROD2\HN_DATA01.DBF	***	634	10240	10240
12	DPPBIG_INDEX	D:\ORADATA\DPPPROD2\DPPBIG_INDEX02.DBF	***	2518	5120	5120
13	HN_DATA	D:\ORADATA\DPPPROD2\HN_DATA02.DBF	***	4430	5120	5120

Gambar 4 List Tablespace di Database Oracle

Keterangan Tablespace

- DPPBIG_DATA, digunakan untuk object table besar
- DPPBIG_INDEX, digunakan untuk index besar
- BACKUP_DATA, digunakan untuk backup tabel
- HN_DATA, digunakan untuk object table menengah
- HN_INDEX, digunakan untuk index menengah
- USERS, digunakan untuk develop table

Untuk selanjutnya jika ada penambahan table maupun index, maka harus menyertakan nama tablespace yang akan ditempatkan. Apakah masuk kedalam golongan object menengah atau besar, contoh:

```
Create table golongan_tua_tbl (id number,Name varchar2(50))
tablespace HN_DATA storage (initial 128k);
```

3. SQL Tuning

Sebelum dilakukan performance tuning, dilakukan pemeriksaan, nama table yang menggunakan disk dalam proses penampilan dengan syntax seperti dibawah, sehingga diketahui query mana yang sering di eksekusi serta penggunaan disk yang besar,

```
select a.USERNAME,
       DISK_READS,
       EXECUTIONS,
       round(DISK_READS / decode(EXECUTIONS, 0, 1, EXECUTIONS)),
       SQL_TEXT
from   dba_users a, sys.v_$session, sys.v_$sqlarea
where  PARSING_USER_ID=USER_ID
and    ADDRESS=SQL_ADDRESS(+)
and    DISK_READS > 10000
       order by DISK_READS desc, EXECUTIONS desc;
```

Gambar 5 Query List Table yang sering digunakan

Hasil yang di dapat:

	USERNAME	DISK_READS	EXECUTIONS	ROUND(DISK_READS/DECODE(EXECUT	SQL_TEXT
61	DPPUSER	470440	13826921	0	SELECT SUM(NVL(IURAN_PESERTA,0)) + SUM(NVL(PI.KOREKSI_PESERTA,0
33	DPPUSER	873639	3452883	0	SELECT SUM(NVL(MANFAAT_PENSIUN,0)) + SUM(NVL(TUNJANGAN_PEF
164	DPPUSER	195436	234872	1	SELECT 1 FROM transaksi_mp_tbl WHERE company_id=:1 AND no_peser
87	DPPUSER	385347	179892	2	INSERT INTO penerimaan_manfaat_pensiun_tbl(company_id,no_peserta,r
178	DPPUSER	151671	179892	1	SELECT 1 FROM penerimaan_manfaat_pensiun_tbl WHERE company_id :
478	DPPUSER	22937	78922	0	SELECT access_granted from role_object_operation_tbl WHERE role_id in
181	SYSMAN	137796	26367	5	DECLARE job BINARY_INTEGER := :job; next_date DATE := :mydate; brok
383	SYSMAN	38283	24375	2	CALL MGMT_ADMIN_DATA.EVALUATE_MGMT_METRICS(target_guid,me
1	DPPUSER	221630792	11340	19544	Select 1 from mutasi_golongan_tbl where company_id=:1 and no_pesert
354	DPPUSER	43417	10716	4	SELECT SOURCE FROM SYS.EXU8SPS WHERE OBJ# = :1 ORDER

Gambar 6 Hasil dari Query

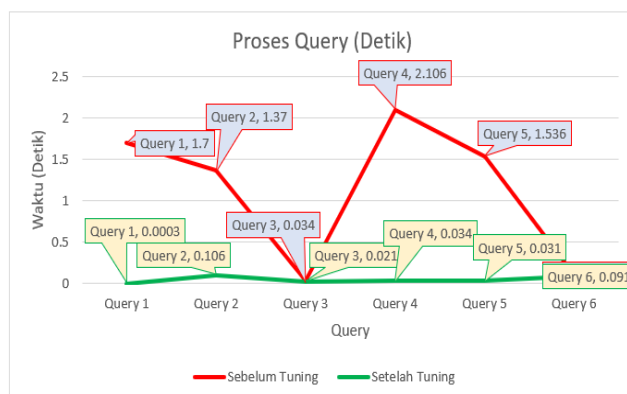
Selanjutnya dilakukan tuning dengan solusi penambahan Index, optimalisasi query mempertimbangkan desain fisik yang sesuai dengan Database dalam memori (menentukan kumpulan indeks yang optimal) [16]:

Tabel 1 – Query SQL Sebelum dan Sesudah optimalisasi

SEBELUM	SESUDAH
Query 1	
Select 1 from mutasi_golongan_tbl where company_id = :1 and no_peserta = :2 and kd_pemberi_kerja = :3 and periode_kepesertaan = :4 and state != 'Dibatalkan'	create index MUTASI_GOLONGAN_IX2 on MUTASI_GOLONGAN_TBL (company_id, No_Peserta, Kd_Pemberi_Kerja,periode_kepesertaan) tablespace DPPBIG_INDEX storage (initial 128K maxextents unlimited);
Query 2	
select count(*) total from (select no_peserta from transaksi_mp_tbl tmp where sekaligus = 'N' and company_id = :1 and periode_kepesertaan = :2 and tmp.kd_pemberi_kerja = :3 and tmp.golongan_pensiun = :4 and tmp.no_pensiun not like '04%' and tmp.state = 'Disetujui' union select no_peserta from data_rekon_pensiun_tbl drp, periode_kepesertaan_tbl pk where drp.company_id = pk.company_id and drp.berlaku_mulai_pensiun <= pk.berlaku_mulai and drp.company_id = :5 and drp.kd_pemberi_kerja = :6 and sekaligus = 'N' and pk.periode_kepesertaan = :7 and drp.kel_golongan = :8 and drp.kd_jenis_pensiun != '04'	create index TRANSAKSI_MP_ix2 on TRANSAKSI_MP_TBL (sekaligus, COMPANY_ID, PERIODE_KEPESERTAAN, kd_pemberi_kerja,golongan_pensiun,state) tablespace DPPBIG_INDEX pctfree 10 initrans 20 maxtrans 255 storage (initial 128K next 512K minextents 1 maxextents unlimited);

and drp.accounting_year = :9 and drp.semester = :10)	
Query 3	
SELECT 1 FROM mutasi_golongan_tbl WHERE company_id=:1 AND no_peserta=:2 AND state NOT IN ('Disetujui', 'Dibatalkan')	Melakukan perbaikan pada state not in karena tidak menggunakan index melainkan menggunakan IN (state yang ada di tables tersebut)
Query 4	
select * from attestatie_de_vita_tbl adv where adv.no_peserta = :nopes and adv.accounting_year = :tahunskd	create index ATTESTATIE_DE_VITA_IX2 on ATTESTATIE_DE_VITA_TBL (no_peserta, accounting_year) tablespace HN_INDEX initrans 20 storage (initial 64K minextents 1 maxextents unlimited);
Query 5	
select SUM(ti.phdp) phdp FROM golongan_tbl g, kelompok_golongan_tbl kg, pemberi_kerja_tbl pk, transaksi_iuran_tbl ti WHERE ti.company_id = g.company_id and ti.kd_golongan = g.kd_golongan and g.company_id = kg.company_id and g.kd_kel_golongan = kg.kd_kel_golongan and ti.company_id = pk.company_id and ti.kd_pemberi_kerja = pk.kd_pemberi_kerja and ti.company_id = 'DPP' and ti.periode_kepesertaan = :1 and pk.kd_pemberi_kerja = :2 and kg.group_kelompok_golongan = :3 and ti.state = 'Disetujui'	create index TRANSAKSI_IURAN_IX4 on TRANSAKSI_IURAN_TBL (COMPANY_ID, kd_golongan,kd_pemberi_kerja, periode_kepesertaan,STATE) tablespace hn_index pctfree 10 initrans 20 maxtrans 255 storage (initial 128K next 512K minextents 1 maxextents unlimited);
Query 6	
Select no_referensi from mutasi_golongan_tbl where company_id = 'DPP' and state not in ('Disetujui', 'Dibatalkan') and kd_pemberi_kerja = '12'	create index MUTASI_GOLONGAN_IX3 on MUTASI_GOLONGAN_TBL (company_id, kd_pemberi_kerja, state) tablespace HN_INDEX initrans 20 storage (initial 64K minextents 1 maxextents unlimited);

Query	Jumlah Record	Waktu Proses (Detik)	
		Sebelum Tuning	Setelah Tuning
Query 1	855,783	1.7	0.0003
Query 2	24,959,380	1.37	0.106
Query 3	855,783	0.034	0.021
Query 4	1,828,488	2.106	0.034
Query 5	12,342,772	1.536	0.031
Query 6	855,783	0.111	0.091



Gambar 7 Tabel dan Grafik Perbandingan Waktu Proses Query Sebelum dan Sesudah Tuning

Dari hasil table dan grafik perbandingan pada gambar 2, didapat performance yang sangat signifikan peningkatannya dengan mengimplementasikan index di beberapa table yang sering di lakukan eksekusi.

4. KESIMPULAN DAN SARAN

Hasil pengujian menunjukkan perbandingan waktu yang dibutuhkan Database untuk menjalankan perintah query sebelum dan sesudah dilakukan tuning Database. Query berjalan lebih cepat setelah dilakukan tuning Database. Hal ini mendukung penelitian sebelumnya yang menyatakan bahwa ada beberapa cara yang dapat dilakukan untuk meningkatkan performance Database.

Untuk meningkatkan performance secara keseluruhan, perlu dilakukan tuning Database diseluruh modul aplikasi yang ada di Dalam Sisforbun. Penelitian yang telah dilakukan pada modul Pengajuan Pembayaran Manfaat Pensiun dapat dijadikan contoh untuk diterapkan pada modul lainnya yang menghasilkan data record dalam jumlah cukup besar diantaranya modul Pasar Modal dan Pasar Uang, Investasi Langsung, Human Resource serta modul Keuangan.

Dan mengingat lokasi kantor Dana Pensiun Perkebunan yang berada di beberapa kota di Indonesia serta peningkatan jumlah data yang cukup besar, untuk penelitian berikutnya kami menyarankan untuk dilakukan kajian penerapan Clustering Database.

DAFTAR PUSTAKA

- [1] A. Bachhav, V. Kharat, and M. Shelar, "Query Optimization for Databases in Cloud Environment: A Survey," *International Journal of Database Theory and Application*, vol. 10, no. 6, pp. 1–12, Jun. 2017, doi: 10.14257/ijdta.2017.10.6.01.
- [2] R. Elmasri and S. B. Navathe, "Database Systems SEVENTH EDITION," 2016.
- [3] A. Abbas and K. Ahmad, "Query Performance in Database Operations."
- [4] www.dapenbun.co.id, "AR 2021," pp. 66–74, 2021, Accessed: Nov. 03, 2022. [Online]. Available: <https://dapenbun.co.id/laporan-tahunan>
- [5] F. Almeida, P. Silva, and F. Araújo, "Performance analysis and optimization techniques for Oracle Relational Databases," *Cybernetics and Information Technologies*, vol. 19, no. 2, pp. 117–132, Jun. 2019, doi: 10.2478/cait-2019-0019.
- [6] D. Colley and C. Stanier, "Identifying New Directions in Database Performance Tuning," in *Procedia Computer Science*, 2017, vol. 121, pp. 260–265. doi: 10.1016/j.procs.2017.11.036.
- [7] Y. Virmansyah, R. Yulyanto Suladi, A. Bambang Lesmana, F. Teknologi Informasi, and U. Budi Luhur, "Optimasi Database dengan Metode Index dan Partisi Tabel Database Postgresql pada Aplikasi E-Commerce. Studi pada Aplikasi Tokopintar."

- [8] O. M. I. Tavares, S. M. Rangkoly, S. B. D. Bawan, K. B. Ahmad, E. Utami, and M. S. Mustafa, "ANALYSIS OF QUERY RESTRUCTURIZATION MODELS IN OPTIMIZING DATA EXECUTION RESPONSE TIME ON MYSQL PHP 7.2.27 RELATIONAL DATABASES," *Jurnal Komputer dan Informatika*, vol. 9, no. 1, pp. 1–10, Feb. 2021, doi: 10.35508/jicon.v9i1.3598.
- [9] S. Indah Nurhafida *et al.*, "Optimasi Query Database Web Scraping Pada Jurnal SINTA," 2022. [Online]. Available: <http://jurnal.mdp.ac.id>
- [10] J. M. Medina, C. D. Barranco, and O. Pons, "Evaluation of Indexing Strategies for Possibilistic Queries Based on Indexing Techniques Available in Traditional RDBMS," *International Journal of Intelligent Systems*, vol. 31, no. 12, pp. 1135–1165, Dec. 2016, doi: 10.1002/int.21820.
- [11] G. Guzun and G. Canahuate, "Hybrid query optimization for hard-to-compress bit-vectors," *VLDB Journal*, vol. 25, no. 3, pp. 339–354, Jun. 2016, doi: 10.1007/s00778-015-0419-9.
- [12] G. Wood, U. Shaft, and J. Beresniewicz, "Performance Fundamentals for Oracle Database 10g and 11g," 2008.
- [13] Q. Li, H. Xu, and G. Yan, "Research on performance optimization and implementation of oracle database," in *3rd International Symposium on Intelligent Information Technology Application, IITA 2009*, 2009, vol. 3, pp. 520–523. doi: 10.1109/IITA.2009.7.
- [14] S. C. Yao, "Optimization design scheme of oracle database," in *Applied Mechanics and Materials*, 2014, vol. 608–609, pp. 589–592. doi: 10.4028/www.scientific.net/AMM.608-609.589.
- [15] L. Yingcheng and L. Ling, "RESEARCH ON SPATIAL DATABASE DESIGN AND TUNING BASED ON ORACLE AND ARCSDE."
- [16] P. L. Bajaj, "A Survey on Query Performance Optimization by Index Recommendation," 2015.