

Analisis Dan Perancang Optimasi Sistem *Inspection Report* Dengan Metode *Subquery* Di *Department Quality Assurance* Pt. Jvc Electronics Indonesia

Apit Priatna¹, Arif Maulana Yusuf^{2*}, Nurul Ayuningtyas Noviana³

Program Studi Sistem Informasi^{1,3}, Komputerisasi Akuntansi^{2*}, STMIK ROSMA

Jln. Kertabumi No. 62, Karawang, INDONESIA

¹apit.priatna@dosen.rosma.ac.id, ²arif@rosma.ac.id, ³nurul@mhs.rosma.ac.id

INTISARI

Seiring dengan banyaknya wacana untuk perbaikan sistem yang berjalan yaitu dengan merubah sistem yang tadinya manual menjadi otomatis atau terkomputerisasi, ternyata dibutuhkan kinerja sistem yang mendukung aktifitas user secara cepat. PT. JVC Electronics Indonesia merupakan perusahaan asing yang bergerak dibidang elektronik, yang tentu saja selalu melakukan perbaikan pada sistem kerjanya. Objek penelitian yang penulis ambil yaitu di departemen QA (Quality Assurance). Tujuan dari penelitian ini yaitu untuk mengoptimalkan kinerja sistem yang sedang berjalan. Selama pelaksanaan penelitian, penulis menemukan adanya permasalahan pada kinerja sistem yang berbasis web di QA. Pada sistem yang berjalan sekarang ditemukan adanya masalah yang dihadapi user karena kinerja website yang lambat. Hal ini terjadi karena pada saat itu terdapat transaksi tambahan pada website, yaitu proses pengambilan data bulan sebelumnya oleh admin.

Kata kunci— Sistem informasi, inspection report, quality assurance, optimasi.

ABSTRACT

Along with many discourse to improve the running system by changing the system from manual to automated or computerized, evidently it needs performance of system which support user activity rapidly. PT. JVC Electronics Indonesia is one of foreign company in electronics field, and of course always do improvement in their work system. The research object that the author take is in QA (Quality Assurance) Department. The purpose of this research is to optimize the work system of running system. During doing the research, the authors found a problem on system performance of web-based in QA. On current system found a problem which faced by user due to the slow performance of website. It caused by at the same time there are the additional transactions on the website, that is data collection process of previous month which conduct by admin.

Kata kunci— Information System, Inspection report, Quality Assurance, Optimizing.

I. PENDAHULUAN

PT. JVC Electronics Indonesia merupakan perusahaan asing yang bergerak dibidang elektronik, yang tentu saja selalu melakukan perbaikan pada sistem kerjanya. Objek penelitian dilakukan di departemen QA (Quality Assurance).

Permasalahan terletak pada kinerja sistem yang berbasis web di QA yang lambat. Hal ini terjadi karena pada saat itu terdapat transaksi tambahan pada website, yaitu proses pengambilan data bulan sebelumnya oleh admin. Sistem *inspection report* di PT. JVC Electronics Indonesia khususnya di *Department Quality Assurance* sudah menggunakan sistem yang terkomputerisasi. Namun, perubahan sistem ini sudah pasti tidak memiliki arti jika bekerja secara lambat[1][2]. Berdasarkan hal tersebut maka perlu

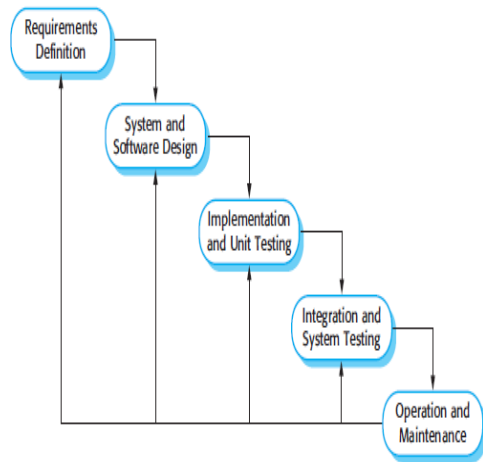
dilakukan *reverse engineering* untuk dapat menemukan letak permasalahan yang selanjutnya dilakukan optimasi terhadap sistem yang sudah ada[3][4][5].

II. METODOLOGI PENELITIAN

Metode analisis data yang digunakan dalam penelitian di PT. JVC Electronics Indonesia yaitu menggunakan SDLC (Systems Development Life Cycle) atau Siklus Hidup Pengembangan Sistem. SDLC adalah tahapan-tahapan suatu pekerjaan yang dilakukan oleh seorang analisis sistem dan programmer dalam membangun sebuah sistem informasi supaya berjalan dengan disiplin dan sistematis[6][7]. Dan pada penulisan penelitian ini penulis memilih *Waterfall* Model untuk metode SDLC dengan tahapan proses *reverse engineering* hal ini karena sistem yang sudah ada akan

dioptimasi. Langkah yang digunakan meliputi tahapan, seperti analisis kebutuhan, desain, dan coding[7][6].

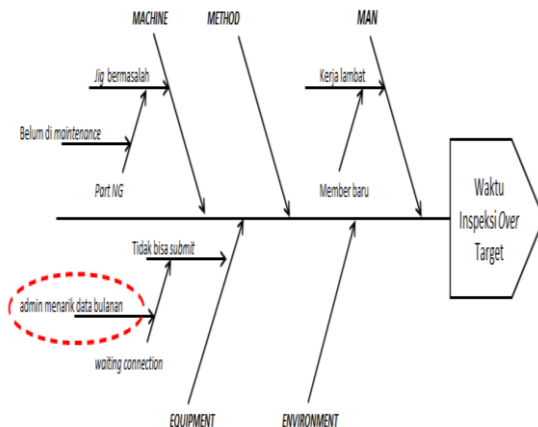
Dalam penelitian ini model SDLC yang akan digunakan yaitu *waterfall model* dengan gambar sebagai berikut :



Gambar 1. Waterfall Model
Sumber : Sommerville [8]

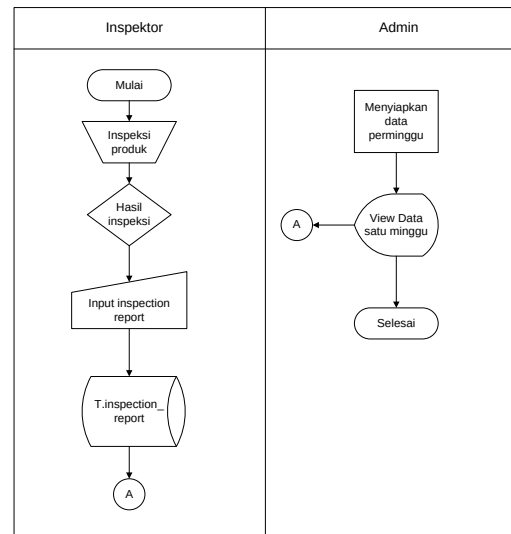
III. HASIL DAN PEMBAHASAN

Untuk menampilkan detail permasalahan yang memiliki hubungan antara kinerja dan aplikasi web ini, maka dibuat dalam bentuk diagram tulang ikan (ishikawa diagram)[9].



Gambar 2. Diagram Tulang Ikan

Selain itu juga alur proses pada Departement QA memiliki alur proses sebagai berikut.



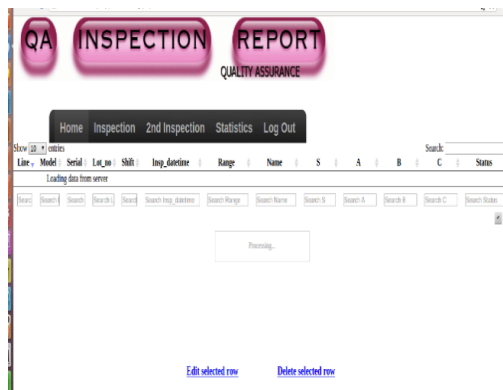
Gambar 3. Flowchart sistem yang berjalan di QA

Menu search pada *website* yang diakses oleh departemen QA akan bekerja dengan menarik data bulanan memerlukan waktu lebih dari 3 menit. Dalam satu kali transaksi maka sistem akan mengambil 18 buah tabel yang berisi data bulanan.

Saat proses penarikan data maupun setelah data muncul di halaman *admin*, ternyata permasalahan tidak cukup sampai disini. Dari sisi *client* pun terjadi masalah yaitu *form inspection report* di halaman *inspection_report.php* tidak bisa di *submit*, selain itu halaman lain yang digunakan user pun menjadi *hang up*.

Gambar 4. Form inspection report tidak bisa submit

Selain halaman *inspection_report.php*, halaman untuk melihat data hasil pengecekan yaitu *monitoring.php* pun tidak bisa memunculkan data.



Gambar 5. monitoring

Selain halaman yang digunakan inspektor dan *admin*, aplikasi lain yang masih terdapat dalam *server* yang sama dan menggunakan *MySQL* memiliki permasalahan yang sama.



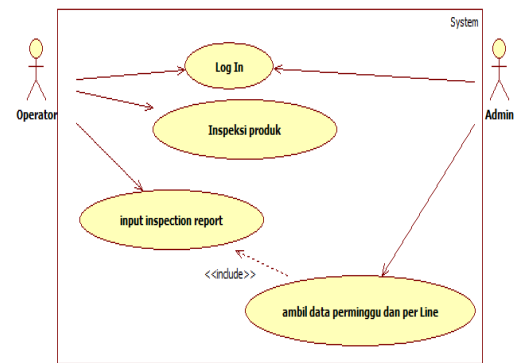
Gambar 6. Pesan error di aplikasi QIS

Setelah mengetahui mengenai permasalahan yang ada selanjutnya dilakukan analisa semantik terhadap perintah *query* yang digunakan[10].

```
SELECT SQL_CALC_FOUND_ROWS t1.Line, t1.Model, t1.Lotno, t2.qty AS Lot_Quantity,
SUM(t1.Merchandise) AS Merchandise,
SUM(t1.Merchandise)*t2.qty AS Balance
FROM
(
SELECT A.Line, TRIM(A.Model) AS Model, A.Lotno, A.Inspdatetime,
IF(RIGHT(A.Range_sampling,4)='0000',10000,
RIGHT(A.Range_sampling,4))-MID(A.Range_sampling,5,4)
)AS Merchandise
FROM inspection_report AS A
GROUP BY A.Line, TRIM(A.Model), A.Lotno, A.Range_sampling
) t1
INNER JOIN
(
SELECT B.Line, B.model, B.lotno, qty, B.date
FROM schedule AS B
GROUP BY B.Line, B.model, B.lotno
) t2
ON t1.Line LIKE CONCAT('%',t2.Line,'%')
AND CONCAT('%',t2.model,'%') LIKE CONCAT('%',t1.Model,'%')
AND t1.Lotno = t2.lotno
WHERE MONTH(t1.Inspdatetime) = '$Month.'
AND YEAR(t1.Inspdatetime) = '$Year.'
GROUP BY t1.Model, t1.Line, t1.Lotno
```

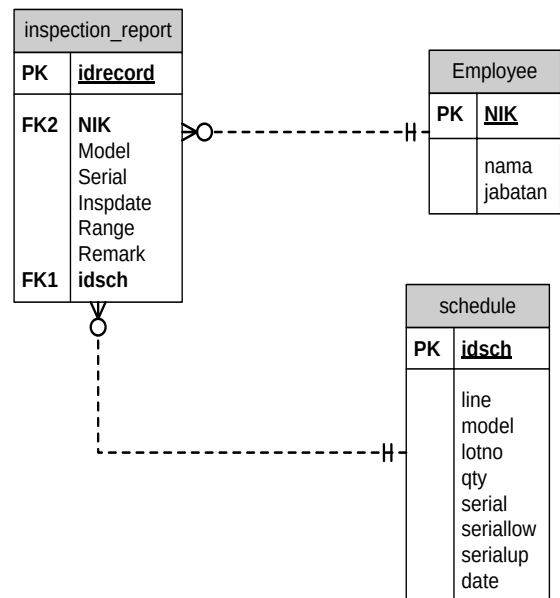
Gambar 7. Query MySQL yang digunakan oleh admin

Setelah dilakukannya perubahan pada prosedur yang berjalan, maka pada *Use Case Diagram* pun mengalami perubahan yang sebelumnya admin melakukan penarikan data per bulan menjadi per minggu. Tampilan *Use Case* pada penggunaan aplikasi sistem usulan di QA yaitu tampak seperti gambar di bawah ini :



Gambar 8. Use Case Diagram Aplikasi Web QA

Di bawah ini merupakan tampilan dari *ERD* setiap tabel yang digunakan dalam *database* sistem usulan. Munculnya relasi pada bab ini, karena ingin mencegah terjadinya redudansi data pada *field line* dan *lotno*.



Gambar 9. Normalisasi tabel di database QA

Selain merubah relasi setiap tabel di *database*, penulis juga harus merubah *query* yang dikirim ke *database*. *Query* yang sebelumnya dikondisikan untuk menarik data bulanan, pada sistem usulan ini dirubah ke

mingguan dan penarikan dilakukan per *Line Production*. Di bawah ini merupakan *query* MySQL yang digunakan oleh *admin* untuk menampilkan data[10].

```
SELECT SQL_CALC_FOUND_ROWS t1.line, t1.Model, t1.lotno,
t1.qty AS Lot_Quantity,
SUM(t1.Merchandise) AS Merchandise,
SUM(t1.Merchandise)-t1.qty AS Balance
FROM
(
SELECT B.line,A.Model, B.lotno,A.Inspdate,B.qty
, IF(RIGHT(A.Range,4)='0000',10000,
RIGHT(A.Range,4))-MID(A.Range,5,4)+1 AS Merchandise
FROM inspection_report AS A
LEFT JOIN schedule AS B
ON A.idsch = B.idsch
WHERE A.Inspdate BETWEEN '%$tgl1.%' AND '%$tgl2.%'
AND B.line LIKE '%$line.%'
GROUP BY B.line, A.Model, B.lotno,A.Range
) t1
GROUP BY t1.Model,t1.line,t1.lotno
```

Gambar 10. Perubahan *query* yang digunakan oleh *admin* untuk menampilkan data

Beban *database* dalam menampung jumlah *record* yang masuk dapat dihitung dengan rumus berikut[10] :

$$\begin{aligned} D &= T \times M \times S \\ Mt &= WM \times D \\ Y &= WY \times D \end{aligned}$$

Keterangan :

D : Jumlah record / hari
 Mt : Jumlah record / bulan
 Y : Jumlah record / tahun
 T : Target inspeksi / orang
 M : Jumlah Orang
 S : Jumlah Shift
 WM : Jumlah hari kerja sebulan
 WY : Jumlah hari kerja setahun

Contoh kasus :

Target inspeksi setiap orang yaitu 50 produk, setara dengan 50 *record* yang masuk ke *database*. Jumlah orang yang melakukan inspeksi 36 orang. Sementara jumlah *shift* di QA yaitu 2 *shift*. Jumlah hari kerja rata-rata dalam sebulan yaitu 21 hari kerja, dan di tahun 2015 ini jumlah hari kerja yaitu 249 hari kerja. Maka jumlah *record* yang masuk ke *database* adalah sebagai berikut :

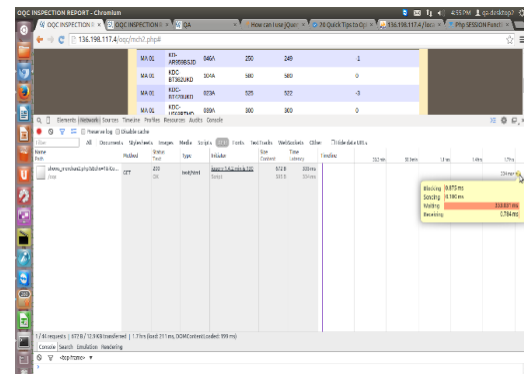
$$\begin{aligned} D &= T \times M \times S \\ D &= 50 \times 36 \times 2 \\ D &= 3.600 \text{ record / hari} \end{aligned}$$

$$\begin{aligned} Mt &= WM \times D \\ Mt &= 21 \times 3.600 \text{ record / hari} \\ Mt &= 75.600 \text{ record / bulan} \\ Y &= MY \times D \\ Y &= 249 \times 3.600 \text{ record / hari} \\ Y &= 896.400 \text{ record / year} \end{aligned}$$

Jadi dari hasil perhitungan di atas jumlah *record* yang harus ditarik dalam satu bulan oleh *admin* sebanyak 75.600 *record*.

Berdasarkan hal tersebut maka perlu dilakukan perubahan dalam perintah *query*-nya yang sebelumnya dikondisikan untuk menarik data bulanan, pada sistem usulan ini dirubah ke mingguan dan penarikan dilakukan per *Line Production*.

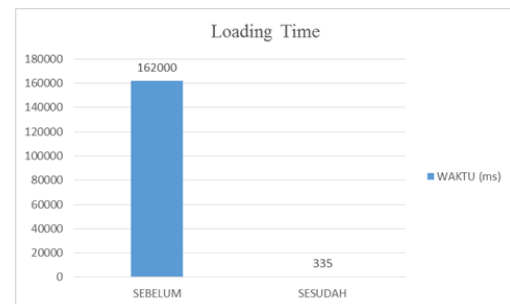
Hasil dari perubahan tersebut dapat terlihat adanya perbaikan dalam kecepatan.



Gambar 11. Loading Time setelah dilakukan optimasi

KESIMPULAN

1. Perbaikan dari sistem lama memiliki hasil yang cukup signifikan pada kinerja sistem. Berikut tabel perbandingan loading time dari kedua sistem.



Gambar 12. Grafik Loading Time

2. Dengan diimplementasikannya sistem baru, ada penyusutan pada waktu *loading time* sebesar 161.665 ms. Hal ini menunjukkan bahwa sistem baru telah mempercepat kinerja *admin* pada saat menarik data untuk laporan bulanan, sementara pengguna sistem lain tidak ada yang terganggu pada saat *admin* melakukan proses tersebut.

REFERENSI

- [1] A. Maulana and Y. K. Suhada, "Perancangan Aplikasi Repository Intranet Pada Laboratorium Komputer Kampus A STMIK Rosma," *Infoman's*, 2018.
- [2] R. D. Beger *et al.*, "Towards quality assurance and quality control in untargeted metabolomics studies," *Metabolomics*, 2019.
- [3] C. Raibulet, F. Arcelli Fontana, and M. Zanoni, "Model-driven reverse engineering approaches: A systematic literature review," *IEEE Access*, 2017.
- [4] H. Ghafoor, "Reverse engineering in orthodontics," *Turkish J. Orthod.*, 2018.
- [5] S. W. Asher, S. Jan, G. Tsaramirsis, F. Q. Khan, A. Khalil, and M. Obaidullah, "Reverse engineering of mobile banking applications," *Comput. Syst. Sci. Eng.*, 2021.
- [6] A. M. Yusuf, H. Hasmizal, and N. Dini, "Sistem Informasi Perhitungan Penyusutan Aktiva Tetap Menggunakan Metode Garis Lurus Berbasis Vb.Net Pada CV Ginanjar Sejahtera Mandiri Karawang," *J. Interkom J. Publ. Ilm. Bid. Teknol. Inf. dan Komun.*, 2021.
- [7] M. V. L. P. Krishna, G. S. Reddy, M. V. Madhav, J. P. Kumar, and K. Amarendra, "Component based software development (SDLC)," *Turkish J. Physiother. Rehabil.*, 2021.
- [8] A. . Rosa and M. Shalahuddin, "Model Sdlc," *Informatika Bandung*, 2018.
- [9] Y. Hisprastin and I. Musfiroh, "Ishikawa Diagram dan," *Artik. Mini Rev.*, 2021.
- [10] G. L. Ginting, P. N. Dian, and Pristiwanto., "Perancangan Aplikasi Pendeteksi Kesalahan Perintah SQL Query Menggunakan Algoritma Knuth Morris Pratt," *J. Ris. Komput. ISSN 2407-389X*, 2018.