

OPTIMASI SISTEM *INSPECTION REPORT* DENGAN METODE *SUBQUERY* DI *DEPARTMENT QUALITY ASSURANCE* PT. JVC ELECTRONICS INDONESIA

Nurul Ayuningtyas Noviana
Yudiana

ABSTRAK

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

1. 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*).

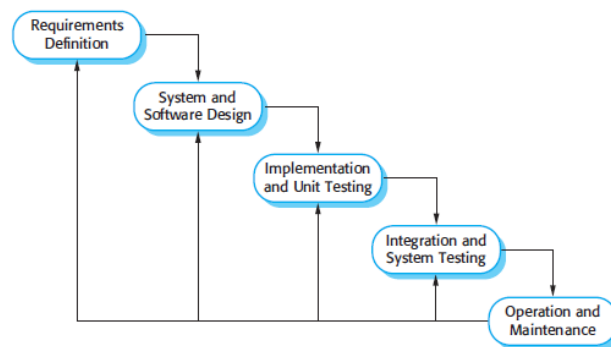
Permasalahan teletak pada kinerja sistem yang berbasis *web* di QA yang lambat. Hal ini terjadi karena pada saat itu terdapat transaksi tambahan pada *website*, yaitu proses pengambilan data bulan sebelumnya oleh *admin*. Sistem *inspection report* di PT. JVC Electronics Indonesia khususnya di *Department Quality Assurance* sudah menggunakan sistem yang terkomputerisasi. Namun, perubahan sistem ini sudah pasti tidak memiliki arti jika bekerja secara lambat. Berdasarkan hal tersebut maka perlu dilakukan *reverse engineering* untuk dapat menemukan letak permasalahan yang selanjutnya dilakukan optimasi terhadap sistem yang sudah ada.

2. METODE 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. 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.

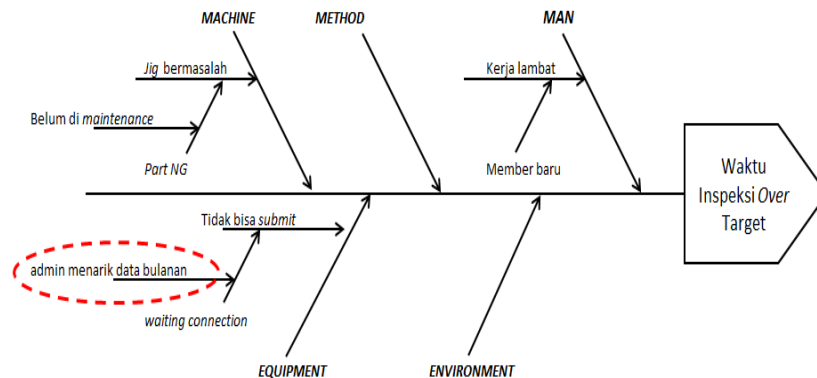
Dalam penelitian ini model SDLC yang akan digunakan yaitu *waterfall model* dengan gambar sebagai berikut :



Gambar 1. *Waterfall Model* (Sommerville,2011)

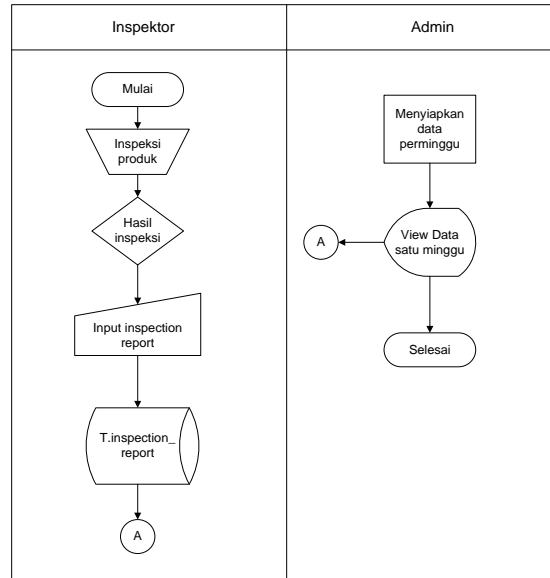
3. ANALISIS DAN PEMBAHASAN

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



Gambar 2. Diagram Tulang Ikan

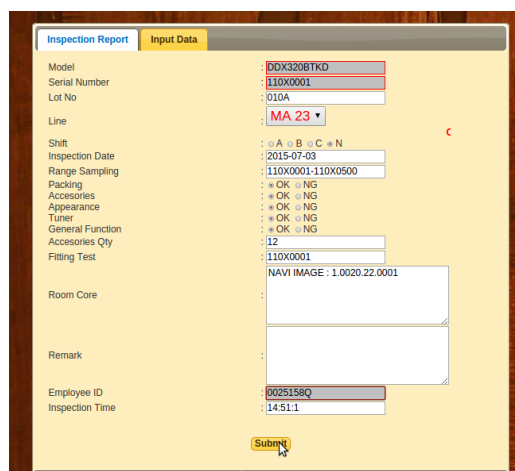
Selain itu juga alur proses pada *Departement QA* memiliki alur proses sebagai berikut :



Gambar 3. Flowchart sistem yang berjalan di QA

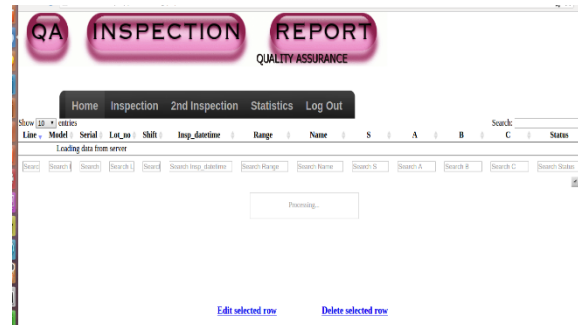
Menu search pada *website* yang diakses oleh departemen QA akan bekerja dengan menarik data bulanan memerlukan waktu lebih dari 3 menit. Dalam satu kali transaksi maka sistem akan mengambil 18 buah tabel yang berisi data bulanan.

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*.



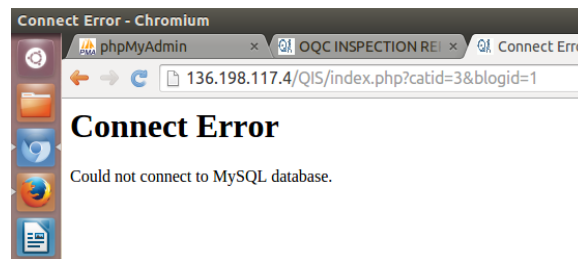
Gambar 4. *Form inspection report* tidak bisa submit

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



Gambar 5. Data tidak muncul pada *monitoring.php*

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*

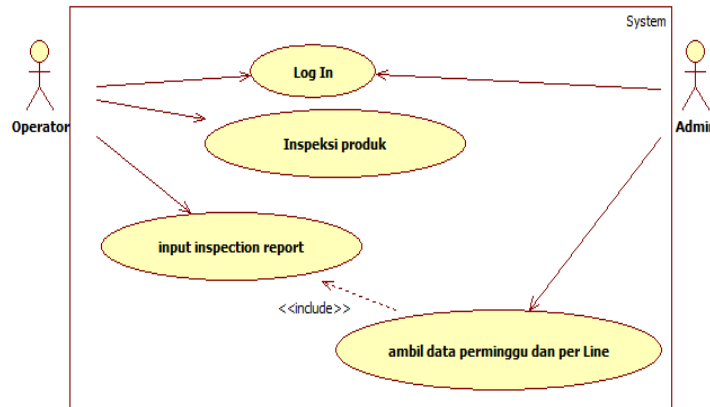
Setelah mengetahui mengenai permasalahan yang ada selanjutnya dilakukan analisa semantik terhadap perintah *query* yang digunakan.

```

SELECT SQL_CALC_FOUND_ROWS t1.Line, t1.Model, t1.Lot_no, 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.Lot_no, A.Inspection_dateime,
           IF(RIGHT(A.Range_sampling,4)='0000',10000,
              RIGHT(A.Range_sampling,4))-MID(A.Range_sampling,5,4)
           )+1 AS Merchandise
    FROM inspection_report AS A
    GROUP BY A.Line, TRIM(A.Model), A.Lot_no, 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.Lot_no = t2.lotno
WHERE MONTH(t1.Inspection_dateime) = '$Month'
AND YEAR(t1.Inspection_dateime) = '$Year'
GROUP BY t1.Model,t1.Line,t1.Lot_no
    
```

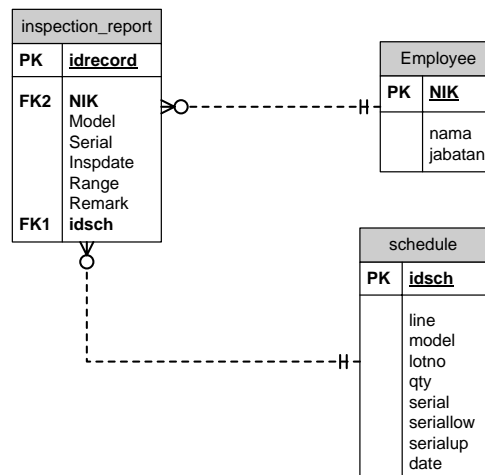
Gambar 7. *Query MySQL* yang digunakan oleh *admin* untuk menampilkan data bulanan.

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 redundansi 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.

```

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 '". $tq11. "%' AND '". $tq12. "%'
        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

D	= T x M x S
Mt	= WM x D
Y	= WY x D

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 :

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

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

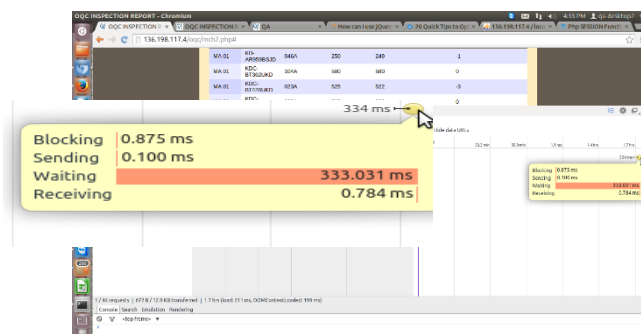
$$Y = 249 \times 3.600 \text{ record / hari}$$

$$Y = 896.400 \text{ record / year}$$

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*.

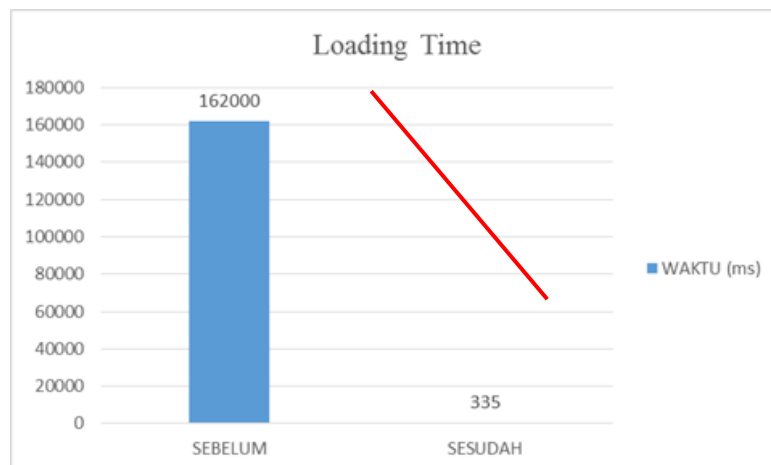
Hasil dari perubahan tersebut dapat terlihat adanya perbaikan dalam kecepatan



Gambar 12. *Loading Time* setelah dilakukan optimasi

4. KESIMPULAN

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



- b. 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.

5. DAFTAR PUSTAKA

- G.K, Gupta. 2011, Sistem Manajemen *Database*. CV. Andi Offset, Yogyakarta.
- Jogiyanto, H.M. 2010, Analisis dan Desain Sistem Informasi, Edisi III. CV. Andi Offset, Yogyakarta.
- Kadir, Abdul. 2014, Pengenalan Sistem Informasi. CV. Andi Offset, Yogyakarta.
- M.Z., T. Yuri, Nurcahyo, Rahmat. 2013, TQM Manajemen Kualitas Total dalam Perspektif Teknik Industri. Index Permata Puri Media, Jakarta.
- Sommerville, Ian, 2011, Software Engineering 9th, Pearson Education, Massachusetts
- Sugiyono. 2010, Metode Penelitian Kuantitatif Kualitatif Dan R&D. Alfabeta, Bandung.
- Sutabri, Tata. 2012, Konsep Dasar Informasi. CV. Andi Offset, Yogyakarta.
- Sutanta, Edhi. 2011, Basis Data Dalam Tinjauan Konseptual. CV. Andi Offset, Yogyakarta.
- Rouf, Abdul. Pengujian Perangkat Lunak dengan Menggunakan Metode *White Box* Dan *Black Box* diakses tanggal 16 Juni 2015 pukul 13.50 WIB
- Wahyu Ariani, Dorothea. 2003, Manajemen Kualitas Pendekatan Sisi Kualitatif. Ghalia Indonesia, Jakarta.
- Wahyu W, Tri. Pengoptimasian Pencarian Data Dengan Algoritma Subset Query. ISSN:1978-9491 | Vol 2, No.1, Th.2008 diakses tanggal 28 April 2015 pukul 15.10 WIB
- <http://www.softwaretestinghelp.com/penetration-testing-guide/> diakses 16 Juni 2015 14.10 wib
- <http://www.softwaretestinghelp.com/web-application-testing/> diakses 16 Juni 2015 14.10 wib
- <http://www.softwaretestinghelp.com/web-testing-example-test-cases/> diakses 16 Juni 2015 14.10 wib
- <http://www.softwaretestinghelp.com/what-is-client-server-and-web-based-testing-and-how-to-test-these-applications/> diakses 16 Juni 2015 14.10 wib
- <http://id.wikipedia.org/wiki/MySQL> diakses 16 Juni 2015 14.10 wib