
PENGGUNAAN DATA ODONTOGRAM DALAM BENTUK FORMAT JSON DARI MATRIKS KONDISI GIGI DAN PERFORMANYA PADA APLIKASI SISTEM IDENTIFIKASI BERBASIS *WEB*

Fredy Rendra Taursia Wisnu*, Sri Kusumadewi**, Intan Ruspita***

* Program Studi Teknik Informatika Program Magister, FTI Universitas Islam Indonesia

** Fakultas Teknologi Industri, Universitas Islam Indonesia

*** Fakultas Kedokteran Gigi, Universitas Gadjah Mada

email: ndra16@gmail.com

KATA KUNCI

antemortem, aplikasi,
JSON, odontogram,
postmortem

ABSTRAK

Pendahuluan: Data odontogram sangat dibutuhkan dalam proses identifikasi dan merupakan salah satu data primer selain sidik jari dan DNA. Bentuk tipe data pada basis data sangat berpengaruh pada besar kecilnya ukuran ruang penyimpanan dan *resource* komputer yang dibutuhkan. Penelitian ini bertujuan untuk menguji penggunaan data odontogram berbentuk matriks kondisi gigi dalam format *Javascript Object Notation* (JSON) dan performanya dalam aplikasi identifikasi berbasis web. **Metode:** Penelitian ini menguji kinerja penggunaan data odontogram yang berbentuk matriks kondisi gigi dalam format JSON; yaitu dalam proses penyimpanan dan proses perbandingan saat melakukan identifikasi menggunakan aplikasi identifikasi berbasis *web* dan basis data *PostgreSQL* dengan melihat besar ruang penyimpanan terpakai dan lama waktu *query* yang dibutuhkan. **Hasil:** Hasil dari penelitian ini menunjukkan rata-rata ukuran ruang penyimpanan yang dibutuhkan untuk menyimpan satu data matriks kondisi gigi odontogram *antemortem* dalam bentuk format JSON pada basis data sebesar 2,2 *kiloBytes (kB)* dan membutuhkan waktu *query* antara 0,3 – 07 *milliseconds (ms)* untuk menyimpan data ke dalam basis data *PostgreSQL* dan waktu *query* yang dibutuhkan untuk satu kali perbandingan data odontogram *postmortem* dan satu data odontogram *antemortem* rata-rata selama 0,4ms, sedangkan waktu yang diperlukan untuk melakukan *rendering* data JSON hasil pencarian dan pencocokan odontogram ke dalam antarmuka aplikasi berbasis *web* membutuhkan waktu 54,1 *ms*. Hasil uji *usability* dari aplikasi identifikasi berbasis *web* dengan menggunakan data odontogram berbentuk format JSON menunjukkan tingkat ketergunaan yang tinggi. **Simpulan:** Penggunaan data odontogram berbentuk matriks kondisi gigi dalam format JSON pada aplikasi identifikasi berbasis web memiliki performa yang baik dengan kebutuhan ruang penyimpanan yang kecil dan waktu yang singkat pada proses *query*.

KEYWORDS

antemortem,
application, JSON,
odontogram,
postmortem

ABSTRACT

Introduction: The data in the odontogram is essential in the identification process and served as one of the primary data besides fingerprints and DNA. The choice of data type in the database affect the required size of the storage space and computer resources. This study aims to examine the use of odontogram data in the form of a dental condition matrix in the Javascript Object Notation (JSON) format and its performance in a web-based identification application. **Methods:** In this study, we tested the performance of odontogram data in the form of dental condition matrix with the Javascript Object Notation (JSON) format in the process of storing and comparison during the identification process using a web-based identification application and PostgreSQL database by looking at the amount of storage space used and the required time for the query process. **Findings:** The results of this study are the average storage space needed to store one row of antemortem dental condition data matrix of the odontogram in the JSON format on the database is 2.2 kiloBytes (kB) and requires between 0.3 – 0.7 milliseconds (ms) query time to store the data into the PostgreSQL database and the average query time needed to compare one postmortem and one antemortem odontogram data is 0.4 ms, while the time needed to render JSON data for search results and odontogram matching into the web-based application interface takes 54.1 ms. Usability test results from a web-based identification application by using odontogram data in JSON format shows a high level of usability. **Conclusion:** The use of odontogram data in the form of dental condition matrix in JSON format in a web-based identification application has good performance with small storage space required and short query processing time.

PENDAHULUAN

Data odontogram atau data rekam kondisi gigi merupakan salah satu dari tiga data primer yang dibutuhkan dalam proses identifikasi untuk menentukan identitas dari mayat yang tidak dikenali baik karena tindak kriminal maupun disebabkan oleh bencana

alam maupun kecelakaan.¹ Penggunaan data odontogram sebagai salah satu data primer disamping data sidik jari dan DNA didasarkan pada tingginya tingkat keunikan data antara dua individu yang berbeda. Berdasarkan hasil penelitian, kemungkinan kejadian dua data odontogram yang sama

antara dua individu di dunia adalah satu berbanding dua milyar atau bisa dibilang kemungkinannya sangat kecil.² Dalam beberapa kejadian bencana dan musibah kecelakaan, banyak kasus identitas dapat ditemukan berdasarkan kecocokan data odontogram.³

Identifikasi yang dilakukan berdasarkan data odontogram saat ini masih terkendala dengan belum tersedianya data odontogram *antemortem* yang memadai sehingga sangat menyulitkan proses identifikasi. Proses identifikasi berdasarkan data rekam gigi saat ini dilakukan dengan mengumpulkan data *antemortem* dan membandingkan dengan data *postmortem* yang terdapat pada jenazah yang belum teridentifikasi. Namun, hingga saat ini di Indonesia belum memiliki data odontogram yang tersentralisasi bahkan berdasarkan penelitian dari Dewanto pada 2016 hanya sedikit dokter gigi yang menuliskan odontogram sesuai dengan aturan yang benar.^{4,5}

Tingginya tingkat keunikan yang ada pada data odontogram juga menyebabkan besarnya data yang harus dibandingkan saat proses identifikasi. Untuk menentukan dua buah data odontogram merupakan milik individu yang sama atau berbeda, data yang dibandingkan merupakan kombinasi antara 52 gigi (32 gigi dewasa dan 20 gigi sulung/susu), 5 permukaan (mesial, distal, occlusal, vestibular dan lingual) pada masing-masing gigi dan kemungkinan kondisi yang menyertai dimana dalam buku

panduan rekam medik kedokteran gigi yang dikeluarkan oleh Kementerian Kesehatan RI pada tahun 2014 menyebutkan ada 34 kondisi, dimana setiap permukaan gigi dapat mengalami satu atau lebih kondisi.⁶ Dengan besarnya data yang harus diperbandingkan ini, akan sangat memakan waktu dan juga membutuhkan sumber daya komputer yang besar jika jumlah data seiring waktu juga akan semakin besar. Besarnya data dan kemungkinan kombinasi antara gigi, permukaan dan kondisi yang menyertai inilah yang menjadi dasar peneliti untuk mencoba penggunaan format *Javascript Object Notation* (JSON), dimana penggunaan format JSON ini bertujuan untuk memperkecil jumlah kolom yang akan digunakan pada basisdata agar tabel basisdata lebih ringkas dan efisien, selain itu format JSON merupakan format yang ringan dan mudah dibaca oleh berbagai bahasa pemrograman dibanding format penulisan odontogram manual yang sulit diterjemahkan oleh mesin dan bahasa pemrograman.⁷⁻⁹

METODE

Penelitian dilakukan pada proses konversi data odontogram kedalam bentuk matriks kondisi gigi dan menyimpannya kedalam basisdata dengan tipe data berbentuk format *Javascript Object Notation* (JSON). Tujuan dilakukan konversi kedalam bentuk matriks kondisi gigi dengan format JSON adalah untuk mempermudah dalam proses perbandingan antara data *antemortem* dan

postmortem, karena JSON merupakan sebuah format yang sangat ringan digunakan dalam proses pertukaran data dan sangat mudah dibaca baik oleh mesin maupun oleh manusia, selain itu juga sudah banyak bahasa pemrograman yang bisa memproses format JSON saat ini. Penyimpanan kedalam bentuk format JSON juga bertujuan memperkecil ukuran tabel basis data yang disimpan dengan mengurangi jumlah kolom yang harus dibuat menjadi cukup hanya satu kolom berisi data JSON sehingga tabel basis data yang harus dibuat menjadi lebih ringkas dan efisien.⁷⁻⁹

Sistem basis data yang digunakan pada penelitian ini adalah *PostgreSQL* yang berjalan di *server Linux* dengan sistem operasi yang berjalan adalah *Ubuntu Server*. Aplikasi identifikasi berbasis *web* yang digunakan adalah aplikasi yang sedang peneliti kembangkan dengan menggunakan *framework Ruby on rails* dimana *framework* ini mengadopsi metode *model, View, Controller (MVC)*.

Tahapan yang dilakukan dalam menguji penggunaan data odontogram berbentuk matriks kondisi gigi dalam format JSON pada aplikasi identifikasi berbasis *web* adalah sebagai berikut :

1. *Input* data odontogram dalam bentuk matriks kondisi gigi yang kemudian diubah kedalam bentuk format JSON saat penyimpanan kedalam basis data. Data yang digunakan dalam penelitian ini, merupakan data yang diambil dari data odontogram pasien

yang datang ke Rumah Sakit Gigi dan Mulut (RSGM) Baiturrahmah Padang yang berjumlah 54 odontogram yang diambil dari data rekam medis 33 pasien.

2. Format JSON yang digunakan dalam penelitian ini adalah sebagai berikut :

```
gigi x {
  "m_x" => "k1, k2,...k-n",
  "o_x" => "k1, k2,...k-n",
  "d_x" => "k1, k2,...k-n",
  "v_x" => "k1, k2,...k-n",
  "l_x" => "k1, k2,...k-n"
}
```

Keterangan :

x = notasi gigi (FDI)
 m,o,d,v,l = permukaan gigi
 k = kondisi gigi

Sebagai contoh, untuk menuliskan kondisi **gigi 11: MODVL rot cw-nvt-rct** dalam bentuk format JSON adalah sebagai berikut :

```
gigi 11 {
  "m_11" => "rot cw,nvt,rct",
  "o_11" => "rot cw,nvt,rct",
  "d_11" => "rot cw,nvt,rct",
  "v_11" => "rot cw,nvt,rct",
  "l_11" => "rot cw,nvt,rct"
}
```

Bentuk format JSON ini saat dilakukan penerjemahan ke dalam bentuk matriks kondisi gigi pada aplikasi akan tampak seperti pada gambar 1 berikut :

	M	O	D	V	L
11	rot cw,nvt,rct				

Gambar 1. Matrik kondisi gigi pada aplikasi identifikasi berbasis *web*

Terlihat pada gambar 1, bahwa informasi data odontogram yang dikonversi dari model manual menjadi model matrik, menjadi lebih mudah dibaca dan dipahami dengan

menampilkan informasi kondisi gigi pada masing – masing kolom permukaan gigi.

- Setelah semua kondisi gigi dari semua odontogram selesai dimasukkan dan dirubah kedalam bentuk format JSON saat disimpan ke dalam basis data, berikutnya dengan menggunakan *script query* yang dijalankan pada tools pgAdmin 4 untuk menghitung besar ruang penyimpanan yang digunakan dalam basis data *PostgreSQL*.¹⁰ Guna melihat lama waktu *query* saat menyimpan data odontogram dalam bentuk format JSON ini, peneliti akan membaca *logfile* yang tersimpan dari aplikasi identifikasi berbasis *web*.^{11,12}
- Tahapan berikutnya setelah data odontogram *antemortem* tersimpan dalam basisdata, adalah membuat simulasi proses identifikasi guna melihat performa dari penggunaan data dengan format JSON dalam aplikasi identifikasi berbasis *web* yang dikembangkan oleh peneliti. Dalam simulasi ini, peneliti membuat sebuah data matrik kondisi gigi dimana dalam setiap permukaan gigi dewasa akan diisi dengan 3 kondisi berbeda. Sehingga dari data ini, sistem akan melakukan pencocokan terhadap total 480

kondisi dari 160 permukaan gigi dewasa.

- Setelah data indentifikasi atau data *postmortem* selesai dimasukkan ke dalam sistem, kemudian dilakukan proses identifikasi pada aplikasi berbasis *web*, sampai sistem memberikan hasil identifikasi.
- Proses identifikasi dilakukan menggunakan aplikasi berbasis *web* dengan cara menghitung tingkat kemiripan antara data *antemortem* yang terdapat dalam basisdata dengan data simulasi yang diberikan pada langkah 4 di atas. Dalam proses ini, aplikasi akan melakukan pencarian berdasarkan jenis kelamin dan umur terlebih dahulu, hasil pencarian akan dikalkulasi kemiripannya satu persatu. Performa dari penggunaan data odontogram dalam bentuk format JSON dilihat berdasarkan lama waktu *query* yang dibutuhkan oleh sistem. Lama waktu *query* ini akan dilihat berdasarkan catatan yang terdapat pada *logfile* setelah proses identifikasi selesai.

HASIL

Script query SQL yang dijalankan pada toolsPgAdmin 4 untuk melihat seberapa besar ukuran ruang penyimpanan yang digunakan oleh sistem berbasis *web* yang sedang dikembangkan oleh peneliti untuk menyimpan data odontogram dalam bentuk

format JSON menunjukkan hasil seperti pada gambar 2 dibawah ini.

	what text	bytes/ct bigint	bytes_pretty text	bytes_per_row bigint
1	core_relation_size	65536	64 kB	1213
2	visibility_map	8192	8192 bytes	151
3	free_space_map	24576	24 kB	455
4	table_size_incl_toast	106496	104 kB	1972
5	indexes_size	16384	16 kB	303
6	total_size_incl_toast_and_indexes	122880	120 kB	2275
7	live_rows_in_text_representation	44236	43 kB	819
8	-----	[null]	[null]	[null]
9	row_count	54	[null]	[null]
10	live_tuples	54	[null]	[null]
11	dead_tuples	21	[null]	[null]

Gambar 2. Hasil *query SQL* dengan menggunakan tools pgAdmin 4

Hasil *query SQL* diatas menunjukkan bahwa dari 54 baris data yang telah tersimpan dalam tabel basisdata odontogram besar ruang penyimpanan untuk menyimpan data relasional adalah sebesar 65536 *bytes* atau 64 *kilobytes (kB)*, jika dihitung rata-rata kebutuhan ruang penyimpanan untuk satu baris data odontogram adalah 1213 *bytes* atau 1,18 *kB*. Sedangkan total ruang penyimpanan yang dibutuhkan setelah ditambahkan *TOAST* dan *indexes* adalah 122880 *bytes* atau 120 *kB*. Dari hasil ini dapat dihitung bahwa rata-rata kebutuhan ruang penyimpanan yang dibutuhkan oleh satu baris data odontogram dalam bentuk JSON adalah 2,2 *kB*.

Waktu yang dibutuhkan untuk menyimpan data odontogram satu gigi dalam bentuk format JSON kedalam basis data *PostgreSQL* dapat dilihat dalam *query log* (diambil dari *logfile* yang dibuat oleh aplikasi saat proses dijalankan) berikut :

```

1. ...
2. ...[1m[36mOdontogram Update (0.7ms)[0m
   ...[1m[33mUPDATE "odontograms" SET "a11" =
   $1, "updated_at" = $2 WHERE "odontograms".
   "id" = $3_[0m [{"a11", "{\"m_11\": \"rot c
   w,nvt,rct\", \"o_11\": \"rot cw,nvt,rct\", \"
   d_11\": \"rot cw,nvt,rct\", \"v_11\": \"rot c
   w,nvt,rct\", \"l_11\": \"rot cw,nvt,rct\"}"]
   , ["updated_at", "2019-09-
   11 06:01:39.359428"], [{"id", "fd298009-
   8791-455b-8efc-05546fc85aad"}]]
3. ...

```

Gambar 3. *Query log* saat proses penyimpanan data JSON kondisi gigi 11

```

1. ...
2. ...[1m[36mOdontogram Load (1.1ms)[0m [1m[3
   4mSELECT "odontograms".* FROM "odontogram
   s" WHERE "odontograms"."no_rm" = $1 ORDER
   BY "odontograms"."recording_date" DESC LIM
   IT $2_[0m [{"no_rm", "1111"}, [{"LIMIT", 1
   }
3. ...
   b, app/controllers/patients_controller.rb
   :14
4. ...
   Rendering patients/show.html.erb within
   layouts/application
5. ...
   Rendered patients/show.html.erb within l
   ayouts/application (5.4ms)
6. ...

```

Gambar 4. *Query log* saat proses *view* data odontogram menjadi matrik kondisi gigi

Seperti tampak pada *querylog* pada gambar 3, waktu yang dibutuhkan untuk menjalankan *queryUPDATE* untuk *field* "a11" dengan data "{ \"m_11\": \"rot cw,nvt,rct\", \"o_11\": \"rot cw,nvt,rct\", \"d_11\": \"rot cw,nvt,rct\", \"v_11\": \"rot cw,nvt,rct\", \"l_11\": \"rot cw,nvt,rct\"}" adalah selama 0,7 *milliseconds (ms)*, Waktu yang dibutuhkan untuk melakukan proses memanggil dan menampilkan data odontogram dalam bentuk matrik kondisi gigi ke dalam halaman detil pasien pada antarmuka pengguna, waktu yang dibutuhkan adalah 1,1 *ms* untuk membaca dan 5,4 *ms* untuk melakukan *rendering* sehingga bisa ditampilkan dalam bentuk matrik kondisi gigi sesuai notasi gigi dan permukaannya seperti tampak pada *query log* gambar 4. Hasil ini menunjukkan bahwa proses *query INSERT / UPDATE* yang dilakukan untuk menyimpan data

odontogram dalam format JSON serta proses pembacaan dan *rendering* untuk ditampilkan pada antar muka pengguna dalam bentuk matrik kondisi gigi terbilang cepat karena waktu yang dibutuhkan untuk menjalankan proses *query* yang hanya dalam hitungan *milliseconds*.

Pada proses pencarian dan pencocokan antara data *antemortem* yang tersimpan dalam basis data dengan data identifikasi atau data *postmortem* yang dimasukkan melalui menu pada aplikasi berbasis *web*, menunjukkan bahwa total waktu yang dibutuhkan oleh sistem untuk membandingkan satu data *postmortem* dengan 37 data *antemortem* yang telah berhasil ditemukan dengan melakukan *filter* pada jenis kelamin dan usia yang sesuai adalah 16,1 *ms* sehingga rata-rata waktu yang dibutuhkan untuk membandingkan setiap data *antemortem* dengan data *postmortem* yang di *input* hanya 0,4 *ms*. Hasil detail waktu yang dibutuhkan untuk masing-masing proses *query* untuk 37 data *antemortem* terdapat pada tabel 1.

Tabel 1. Catatan waktu lama *query* pencocokan antara data identifikasi yang memiliki *uuid* 31ae620b-0b31-45f3-9392-a25f9833baaddengan data *antemortem* dari basisdata

<i>Odontograms uuid</i>	<i>Time (ms)</i>
77f35356-5f6c-42ff-b7b7-db660bcfad5e	0,6
4b26f1bb-bb30-4340-a151-3e59be9839be	0,6
9a37f037-7182-43d4-ba88-b1ec8c030cff	0,9
595f3a10-ca2f-45bf-8ac1-edbaf924ba8f	0,6
720ee07f-f03c-46a0-8c58-3f089548f03f	1,4
90ce9e81-51a3-4080-8eb9-efbba42278a9	0,5
e123e831-6b60-4daf-90f5-8d82ea247642	0,6
5251e2be-1002-47fa-b060-fcc684fa6525	0,3
fbff890-3962-4acb-8fbd-2fa1cc93c349	0,3
0192323b-6aa5-495a-a55b-a8c7f93689ce	0,3
334a6e14-520e-41cc-af5e-237c47face3b	0,4

f0821028-a52a-44a8-9d97-bf0f015426fd	0,4
1ec04f6d-3b34-4128-9eff-cc0c49f9cbda	0,3
3478ee71-ace7-432c-be85-c417445a86f6	0,3
e9634e88-552c-4e4e-91b2-fb969fa0f2ae	0,3
1a6a1243-5b14-4b31-b2a7-1975705fc2f3	0,7
a6928830-1578-4f0b-97d0-ab98231ff2f2	0,3
58fc270e-c78e-493f-bd16-31772ce817e0	0,3
e81594b3-376b-4177-8375-236cb8af437a	0,7
50500542-ab45-4821-9ab4-4328dc702fd4	0,3
7158bff1-e9b8-4dd0-ade7-b4ee1b6dc674	0,3
7a6373ea-d1d2-431f-a8a7-1c2033cf9bc6	0,7
064c81e3-d8d1-4897-8297-40055d39ed02	0,3
cef00d9c-962f-41d8-bca9-8ac3e1247c17	0,3
cd62555b-82d8-43c5-9033-ec83f176c4e8	0,6
95435868-67d6-49a8-80c2-e878e132458a	0,3
4a1c9f55-f7dd-44e4-9142-113e0d2764bd	0,3
eb4e8f17-a0a4-449f-b297-4d2769e79431	0,3
f27a6768-4f97-44d9-a0e2-b2c82d1a3b38	0,3
dd65fe8b-196e-4d57-b759-9bd47a1217b3	0,3
dabd1858-2e27-4c65-b86c-fec240fc224f	0,3
3ccb32cb-0fad-4dc8-8b20-a6275929b374	0,3
e0bcd5e2-0622-4cca-9965-b63279104b47	0,3
8f963d9d-c066-495d-a033-91274ce7339d	0,4
bcf025df-7a31-41bc-b943-879526700590	0,4
8b298ece-bf9c-4f96-a13d-f330b3736c24	0,3
fd298009-8791-455b-8efc-05546fc85aad	0,3
TOTAL	16,1
Rata - rata	0,4

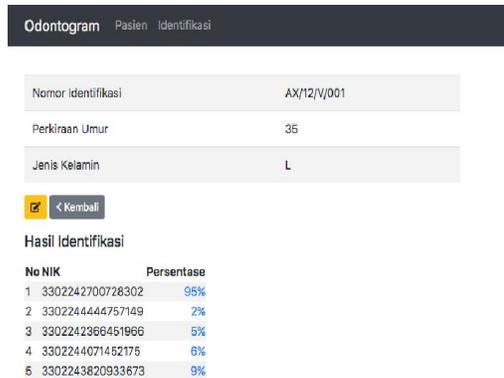
Setelah proses pencocokan ini, sistem melakukan *rendering* data dan akan menampilkan hasil pencarian berupa daftar odontogram *antemortem* disertai persentase kemiripan dengan data *postmortem*. Waktu yang dibutuhkan untuk proses *rendering* ini adalah 54,7 *ms* seperti dapat dilihat pada *query log* gambar 5 dan hasil *rendering* data pencarian pada gambar 6.

```

1. ...
2. [1m[36mOdontogram Load (0.3ms)[0m
   [1m[34mSELECT "odontograms".* FROM "
   odontograms" WHERE "odontograms"."id"
   = $1 LIMIT $2[0m [{"id", "fd298009-
   8791-455b-8efc-
   05546fc85aad"}, ["LIMIT", 1]]
3.   â†³ app/views/identifications/_resu
   lt.html.erb:15
4.   Rendered identifications/_result.ht
   ml.erb (54.7ms)
5. ...

```

Gambar 5. *Query log* saat proses *rendering* data hasil identifikasi



Gambar 6. Antarmuka hasil pencarian pada aplikasi identifikasi berbasis web

Waktu *query* yang dibutuhkan untuk memproses komparasi setiap permukaan gigi dari data identifikasi, sebagai contoh, kami mengambil *logfile* proses identifikasi dengan `uuid31ae620b-0b31-45f3-9392-`

`a25f9833baad` dan data odontogram *antemortem* dengan `uuidfd298009-8791-455b-8efc-05546fc85aadd` waktu yang diperlukan selama *7,6 ms*, *Query log* proses komparasi dapat dilihat pada Gambar 7 berikut ini.

```

1. ...
2. [1m[36mIdentification Load (0.8ms)[0m [1m[34mSELECT "identifications". * FROM "identifications" WHERE "identifications"."id" = $1 LIMIT $2[0m [{"id", "31ae620b-0b31-45f3-9392-a25f9833baad"}], [{"LIMIT", 1}]
3. â†³ app/controllers/identifications_controller.rb:74
4. [1m[36mOdontogram Load (0.5ms)[0m [1m[34mSELECT "odontograms".* FROM "odontograms" WHERE "odontograms"."id" = $1 LIMIT $2[0m [{"id", "fd298009-8791-455b-8efc-05546fc85aad"}], [{"LIMIT", 1}]
5. â†³ app/controllers/identifications_controller.rb:65
6. Rendering identifications/compariso n.html.erb within layouts/application
7. Rendered identifications/compariso n.html.erb within layouts/application (7.6ms) |
8. ...
    
```

Gambar 7. *Query log* dari proses komparasi antara data *antemortem* dan *postmortem*

Hasil komparasi antara data odontogram *antemortem* dan *postmortem* akan ditampilkan dalam antarmuka pengguna dengan pemberian perbedaan warna, dimana untuk gigi dengan kondisi sama akan berwarna hijau muda, sedangkan untuk gigi dengan kondisi yang berbeda akan berwarna merah muda seperti tampak pada gambar 8.

Ante Mortem					Post Mortem				
M	O	D	V	L	M	O	D	V	L
11	-	-	-	-	11	-	-	-	-
12	-	-	-	-	12	-	-	-	-
13	-	-	-	-	13	-	-	-	-
14	-	-	-	-	14	car,caf	car,caf	-	-
15	car,caf	car,caf	-	-	15	car,caf	-	-	-
16	car,caf	-	-	-	16	car,caf	-	-	-
17	-	-	-	-	17	-	-	-	-
18	une	une	une	une	18	une	une	une	une
21	-	-	-	-	21	-	-	-	-
22	ret,esp	ret,esp	ret,esp	ret,esp	22	-	-	-	-
23	-	-	-	-	23	-	-	-	-
24	car	car	-	-	24	car	car	-	car
25	-	-	-	-	25	-	-	-	-
26	mls	mls	mls	mls	26	mls	mls	mls	mls

Gambar 8. Antarmuka hasil komparasi data *antemortem* dan *postmortem*

PEMBAHASAN

Penggunaan format JSON untuk menyimpan data matrik kondisi gigi pada aplikasi identifikasi berbasis web terbukti memiliki kinerja yang baik. Hasil dari pengujian yang telah dilakukan dalam penelitian menunjukkan bahwa kapasitas kebutuhan ruang penyimpanan untuk menyimpan satu baris data odontogram dalam basis data *postgreSQL* hanya memerlukan ruang sebesar *2,2 kB*. Dari besar data ini dapat kita hitung dalam sebuah media penyimpanan yang berkapasitas *1 TeraByte (TB)* diperkirakan akan mampu menyimpan data JSON odontogram lebih dari *465 juta* baris data. Kinerja bagus juga ditunjukkan oleh waktu yang dibutuhkan untuk menjalankan *query* dalam setiap proses yang berjalan pada

aplikasi identifikasi berbasis *web* yang dikembangkan oleh peneliti dan digunakan dalam penelitian ini. Kinerja waktu *query* yang diperlukan untuk semua proses berjalan membutuhkan waktu tidak lebih dari satu detik atau dibawah 1000 *ms*. Dengan demikian, proses identifikasi dengan cara membandingkan data odontogram *postmortem* dari identifikasi dengan data *antemortem* yang telah tersimpan dalam basisdata relatif lebih cepat jika dibandingkan dengan cara manual yang selama ini dilakukan. Meskipun dalam proses identifikasi data gigi bukan merupakan satu - satunya data penentu, karena harus ditunjang dengan data primer dan data sekunder lainnya, namun dari proses pencocokan menggunakan aplikasi berbasis *web* dengan data berbentuk format JSON ini, ahli forensik bisa mendapatkan rekomendasi dan masukan tentang identitas yang dicari dengan lebih cepat.

Selanjutnya, untuk menguji ketergunaan (*usability*) dari aplikasi identifikasi berbasis *web* menggunakan data odontogram dalam format JSON ini, peneliti menggunakan metode pengujian *USE Questionnaire* yang memiliki empat faktor penilaian yaitu *usefulness*, *ease of use*, *ease of learning* dan *satisfaction*.¹³ Uji ini dilakukan dengan memberikan pertanyaan dalam bentuk kuesioner terkait penggunaan aplikasi yang telah dicoba oleh 11 responden dokter gigi yang menjadi pengguna. Kuesioner yang diberikan menggunakan jawaban dalam skala *Likert* dimana masing-masing pertanyaan terdapat empat pilihan jawaban yaitu Sangat

tidak setuju (STS), tidak setuju (TS), setuju (S) dan sangat setuju (SS).¹⁴ Uji ketergunaan (*usability*) ini dihitung berdasarkan persentase jumlah jawaban setuju (S) dan sangat setuju (SS). Hasil persentase jawaban setuju dan sangat setuju dari uji ketergunaan ini dapat dilihat pada tabel 2.

Tabel 2. Hasil uji ketergunaan (*usability*)

Faktor pengujian	Persentase Jawaban
<i>Usefulness</i>	100%
<i>Ease of Use</i>	97,7%
<i>Ease of Learning</i>	100%
<i>Satisfaction</i>	100%

Menurut Tabel 2, hasil uji ketergunaan menunjukkan bahwa semua faktor pengujian memiliki persentase jawaban setuju dan sangat setuju yang tinggi. Hasil ini membuktikan bahwa aplikasi identifikasi berbasis *web* dengan penggunaan data odontogram dalam bentuk format JSON memiliki tingkat ketergunaan (*usability*) yang baik menurut pengguna, baik dari segi kegunaan (*usefulness*), kemudahan penggunaan (*ease of use*), kemudahan dipelajari (*ease of learning*) maupun kepuasan pengguna terhadap aplikasi (*satisfaction*).

SIMPULAN

Penggunaan data matrik kondisi gigi dengan format JSON menunjukkan hasil uji performa yang bagus dari segi besar kebutuhan ruang penyimpanan maupun catatan waktu *query* yang diperlukan untuk menyelesaikan proses penyimpanan, proses pencarian, pencocokan dan perbandingan. Hal ini menunjukkan bentuk matrik kondisi

gigi dalam format JSON dapat dipertimbangkan untuk digunakan dalam penyimpanan pada database dari aplikasi identifikasi berbasis *web*. Penggunaan bentuk format JSON ini juga akan memungkinkan pertukaran data yang lebih mudah mengingat format JSON merupakan format ringan yang mudah dibaca baik sesama aplikasi berbasis *web*, maupun aplikasi berbasis sistem lainnya seperti *desktop* atau *mobile* dengan penggunaan *web service* atau *Application Programming Interface (API)*.

DAFTAR PUSTAKA

1. Interpol. INTERPOL Disaster victim identification guide [Internet]. 2014. Available from: <https://www.interpol.int/INTERPOL-expertise/Forensics/DVI>
2. Murniwati M. Peran Rekam Medik Gigi Sebagai Sarana Identifikasi. *Maj Kedokt Andalas* [Internet]. 2012 Aug 30 [cited 2018 Aug 24];36(2):163. Available from: <http://jurnalangka.fk.unand.ac.id/index.php/article/view/124>
3. Prawestiningtyas E, Algozi AM. Identifikasi Forensik Berdasarkan Pemeriksaan Primer dan Sekunder Sebagai Penentu Identitas Korban pada Dua Kasus Bencana Massal. *J Kedokt Brawijaya* [Internet]. 2009 Mar 19 [cited 2018 Aug 24];25(2):88–94. Available from: <http://jkb.ub.ac.id/index.php/jkb/article/view/175>
4. Dewanto I. Gambaran Rekam Medik Gigi sebagai Posisi Sentral bagi Dokter Gigi di Yogyakarta. *Mutiara Med J Kedokt dan Kesehat*. 2016 Jun 2;7(2):83–7.
5. Trisnowahyuni, Rahim AH, Doloksaribu EI. Rekam Medis Odontogram Sebagai Alat Identifikasi Dan Kepentingan Pembuktian Di Pengadilan. *SOEPRA J Huk Kesehat*. 2017;3(1):117–31.
6. Kementerian Kesehatan RI. Panduan Rekam Medik Kedokteran Gigi [Internet]. 2014 [cited 2018 Aug 2]. Available from: <http://pdgi.or.id/wp-content/uploads/2015/03/Buku-Rekam-Medik-KG-20141.pdf>
7. Bourhis P, Reutter JL, Suárez F, Vrgoč D. JSON: Data model, Query languages and Schema specification. In: *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. 2017. p. 123–35.
8. Rosid MA. Implementasi JSON untuk Minimasi Penggunaan Jumlah Kolom Suatu Tabel Pada Database PostgreSQL. *JOINCS (Journal Informatics, Network, Comput Sci* [Internet]. 2017 Jul 21 [cited 2019 Aug 15];1(1):33. Available from: <http://ojs.umsida.ac.id/index.php/JOINCS/article/view/802>
9. Yusof MK, Man M. Efficiency of JSON approach for data extraction and query retrieval. *Indones J Electr Eng Comput Sci*. 2016;4(1):203–14.
10. Brandstetter E. performance - Measure the size of a PostgreSQL table row - Database Administrators Stack Exchange [Internet]. 2012 [cited 2019 Sep 1]. Available from: <https://dba.stackexchange.com/questions/23879/measure-the-size-of-a-postgresql-table-row/23933#23933>
11. Hasani Z, Jakimovski B, Kon-popovska M, Velinov G. Real Time Analytic of SQL Queries Based on Log Analytic. In: *ICT Innovations 2015 Web Proceedings* [Internet]. 2015 [cited 2019 Aug 11]. p. 78–87. Available from: <https://proceedings.ictinnovations.org/attachment/paper/377/real-time-analytic-of-sql-queries-based-on-log-analytic.pdf>
12. Vernekar SS, Buchade A. MapReduce based log file analysis for system threats and problem identification. In: *Proceedings of the 2013 3rd IEEE International Advance Computing Conference, IACC 2013*. 2013. p. 831–5.
13. Lund AM. Measuring Usability with the USE Questionnaire [Internet]. 2001 [cited 2019 Sep 5]. Available from: https://www.researchgate.net/publication/230786746_Measuring_Usability_with_the_USE_Questionnaire
14. Asnawi N. Pengukuran Usability Aplikasi Google Classroom Sebagai E-learning Menggunakan USE Questionnaire (Studi Kasus: Prodi Sistem Informasi UNIPMA). *Res Comput Inf Syst Technol Manag*. 2018;1(1):17.