

Implementasi JSON untuk Minimasi Penggunaan Jumlah Kolom Suatu Tabel Pada Database PostgreSQL

¹Mochamad Alfian Rosid

Universitas Muhammadiyah Sidoarjo

e-mail: 1alfanrosid@umsida.ac.id

Abstrak

Pada perkembangannya database saat ini telah menjadi suatu hal yang sangat penting, setiap akan merencanakan sesuatu, mempertimbangkan hal apapun kita akan memerlukan dan menggunakan data yang ada didalam suatu database, PostgreSQL merupakan salah satu Object Relational Database Management System(DBMS) open source dengan banyak fitur yang tidak kalah canggihnya dengan database komersial. Pada penggunaannya terkadang dijumpai suatu kasus yang memerlukan kolom yang dinamis sehingga harus membuat kolom baru untuk memenuhi kebutuhan tersebut, akan tetapi bila tiap baris suatu tabel memerlukan kolom yang berbeda-beda dan berpotensi membutuhkan kolom yang lain lagi maka hal ini akan merepotkan dan terjadi pemborosan penggunaan kolom.

Pada penelitian memanfaatkan teknologi JavaScript Object Notation (JSON) dalam hal ini fungsi json_encode dengan bahasa pemrograman PHP untuk menyederhanakan penggunaan kolom sehingga ketika ada kebutuhan penambahan kolom tidak perlu lagi membuat kolom baru di tabel tersebut, Dari hasil penelitian menunjukkan bahwa JSON dapat menyederhanakan tabel suatu basis data, tabel yang sebelumnya membutuhkan banyak kolom menjadi satu kolom yang berisi string JSON.

Kata kunci: Database, PostgreSQL, ORDBMS, JSON, PHP

Abstract

On the development of the database currently has become a very important thing, each will plan something, consider any thing we will need and use existing data in a database, The development of the database currently has become a very important thing, when we want to plan something and consider everything, we will need and use existing data in a database, PostgreSQL is one Object Relational Database Management Systems (DBMS) open source with a lot of features that are not less sophisticated to commercial databases. On daily use, it sometimes occurs a case that requires a dynamic column, therefore we should make the new columns to meet those needs. However, if each row of the table column requires different column and potentially need another ones, then it is going to be useless.

On the research of utilizing the technology of JavaScript Object Notation (JSON), json_encode PHP programming language can simplify the use of the column so that a new column is no longer needed when we want to create a new column on the table, we only need a new column that stores an array that has been converted into a JSON string forms the result of json_encode that represents the column name and the content of the columns of a table.

Keywords: Database, PostgreSQL, ORDBMS, JSON, PHP

1. Pendahuluan

Database dewasa ini menjadi salah satu kebutuhan yang sangat penting dimiliki oleh sebuah perusahaan, database dapat diibaratkan sebagai lemari arsip yang menyimpan data-data perusahaan tersebut [2], semakin besar perusahaan otomatis akan

membutuhkan lemari arsip yang besar pula, untuk menggantikan peran lemari arsip tersebut, muncul berbagai macam software database salah satunya PostgreSQL.

Sebagai ORDBMS (Object Relational Database Management System) yang ada saat ini, PostgreSQL memiliki berbagai macam kemampuan yang dimiliki oleh database komersial umum lainnya, seperti dukungan akan perintah-perintah SQL, dimana dengan menggunakan perintah-perintah SQL memungkinkan database administrator lebih mudah berinteraksi dengan database PostgreSQL, baik dalam manipulasi data seperti : insert, update, ataupun delete[7]. Selain bersifat *open source*, keunggulan database PostgreSQL ini dapat mendukung sebagai media penyimpanan pada banyak bahasa pemrograman yang ada, baik itu pemrograman berbasis desktop, seperti : Java, Gambas, dll ataupun pemrograman berbasis web, seperti : bebePython, PHP, Java Server Pages, Perl, dll.

Pada penggunaannya seperti membuat tabel terkadang dijumpai suatu kasus yang memerlukan kolom yang dinamis sehingga harus membuat kolom baru untuk memenuhi kebutuhan tersebut, akan tetapi bila tiap baris suatu tabel memerlukan kolom yang berbeda-beda dan berpotensi membutuhkan kolom yang lain lagi maka hal ini akan merepotkan dan terjadi pemborosan penggunaan kolom. Oleh karena itu dibutuhkan sebuah solusi agar ketika membuat tabel yang memiliki kolom dinamis artinya ada baris yang membutuhkan kolom baru ada yang tidak membutuhkan kolom yang baru tersebut, administrator database tidak perlu lagi berkali-kali mengupdate tabel.

Berdasarkan permasalahan tersebut, pada penelitian ini mencoba menerapkan JSON dengan memanfaatkan fungsi *json_encode* dengan bahasa pemrograman PHP untuk meminimalisasi jumlah kolom pada beberapa kasus tertentu. Dengan memanfaatkan fungsi *json_encode* ini, administrator database cukup menyediakan sebuah kolom yang mana kolom tersebut berisikan nilai hasil enkripsi *json_encode*, yaitu sebuah array bisa 1 dimensi 2 dimensi atau seterusnya yang diubah menjadi string JSON. String JSON tersebut mewakili nama kolom dan isi dari kolom tersebut. Array ini harus dirubah menjadi bentuk JSON karena kalau tidak, maka array tersebut otomatis tidak bisa dimanfaatkan dan hanya terisi kata 'array' saja.

JSON [1][5] adalah obyek yang pada awalnya dirancang dan dikembangkan dengan format pertukaran data yang ringan, berbasis teks, bahasa-independen pertukaran data berasal dari objek literal ECMAScript (JavaScript) pemrograman bahasa standar. JSON objek yang dianalisis sebagai array string, dengan parsing yang lebih tinggi, efisiensi dan format yang lebih mudah daripada format transportasi seperti XML. JSON terbuat dari dua struktur data: kumpulan pasangan nama-nilai dan daftar ordered nilai-nilai. Memiliki format data yang dapat dipertukarkan dengan struktur *built-in* data bahasa pemrograman mengurangi kompleksitas dan waktu pemrosesan. Selain itu [3], dasar dari XML juga hadir dalam JSON, jadi tidak ada yang secara signifikan dihilangkan [6].

Penelitian sebelumnya membandingkan XML dengan JSON dan hasilnya JSON memiliki tingkat efisiensi parsing yang lebih tinggi dan lebih mudah dalam penerapannya dibandingkan dengan XML, serta diusulkan bagaimana menerjemahkan dokumen XML ke format JSON dan mengusulkan enkripsi format JSON sebagai alternatif enkripsi XML [1].

2. Metode Penelitian

2.1 PostgreSQL

PostgreSQL (dibaca Post-Gres-Q-L) atau postgres adalah *Object Relational Database Management System* yang bersifat *open source*. PostgreSQL menekankan ekstensibilitas, kreativitas, serta kompatibilitas. Bersaing dengan vendor database relasional yang besar seperti Oracle, MySQL, SQL Server, dan lain-lain. Hal ini digunakan oleh berbagai sektor termasuk lembaga pemerintah dan publik dan sektor swasta. PostgreSQL adalah cross-platform DBMS, dan berjalan pada sistem operasi

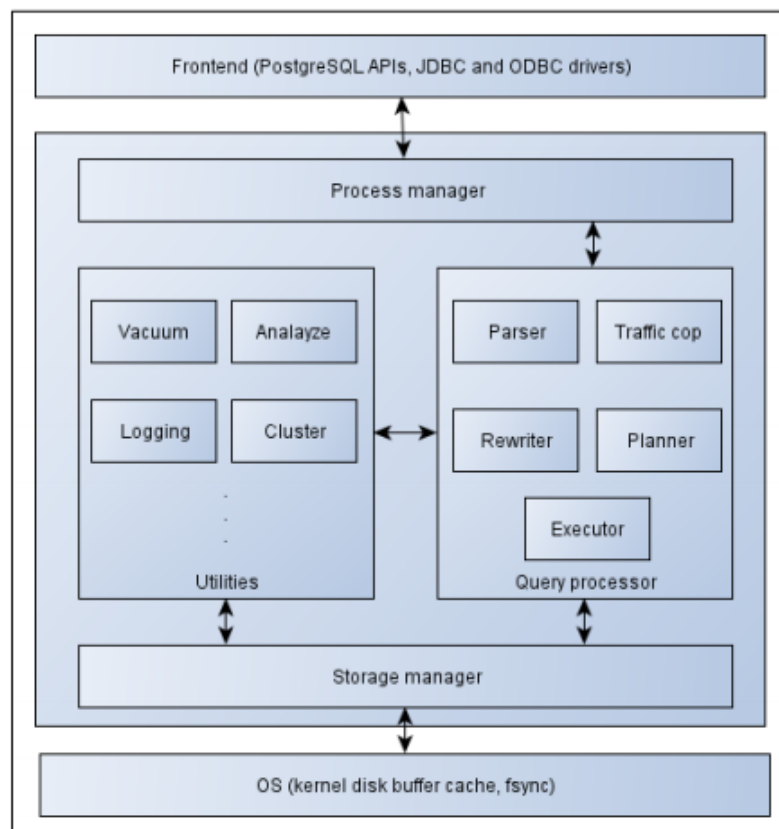
yang paling modern termasuk Windows, MAC, dan Linux rasa. Ini sesuai dengan SQL, dan ACID compliant [4].

2.2 Arsitektur PostgreSQL

Secara ringkas arsitektur PostgreSQL dapat dijelaskan konsep kemampuan PostgreSQL dan interaksi dengan klien, dan sistem operasi. PostgreSQL server dapat dibagi secara garis besar menjadi empat subsistem, sebagai berikut [4]:

- *Process manager*: mengelola proses manajemen koneksi klien seperti Forking dan proses mengakhiri.
- *Query Processor*: ketika klien mengirimkan *query* ke PostgreSQL, *query* diurai oleh parser, dan kemudian menentukan subsistem *traffic cop* jenis permintaan. *Query* utilitas dilewatkan ke subsistem utilitas. Pilih, insert, Update, dan delete *query* ditulis ulang oleh *rewriter* yang akhirnya, *query* dijalankan dan Hasilnya dikembalikan ke klien.
- *Utilites*: Subsistem utilitas menyediakan sarana untuk mempertahankan database mengklaim penyimpanan, memperbarui statistik ,mengekspor dan mengimpor data dengan format dan penguncian.
- *Storage manager*: *Storage manager* menangani cache memori, disk buffer, dan alokasi penyimpanan.

Gambar 1 adalah gambar diagram yang menggambarkan arsitektur konseptual PostgreSQL:

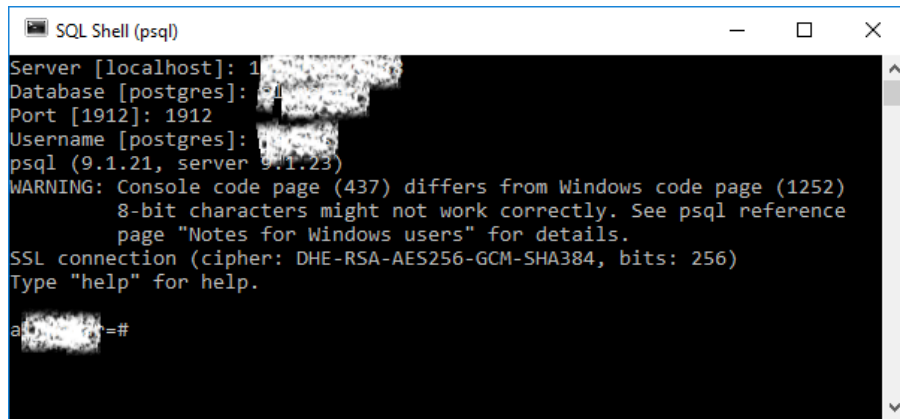


Gambar 1. Arsitektur Konseptual PostgreSQL

2.3 Pembuatan objek di PostgreSQL

Pada dasarnya untuk membuat objek-objek yang diperlukan seperti database, tabel, view, user, dll di database PostgreSQL dapat menggunakan dua cara yakni mode

teks, dan graphical user interface (GUI). Untuk mode teks, perintah-perintah SQL ditulis di psql client yang ditunjukkan gambar 2, contoh pembuatan tabel di psql ditunjukkan oleh gambar 3.

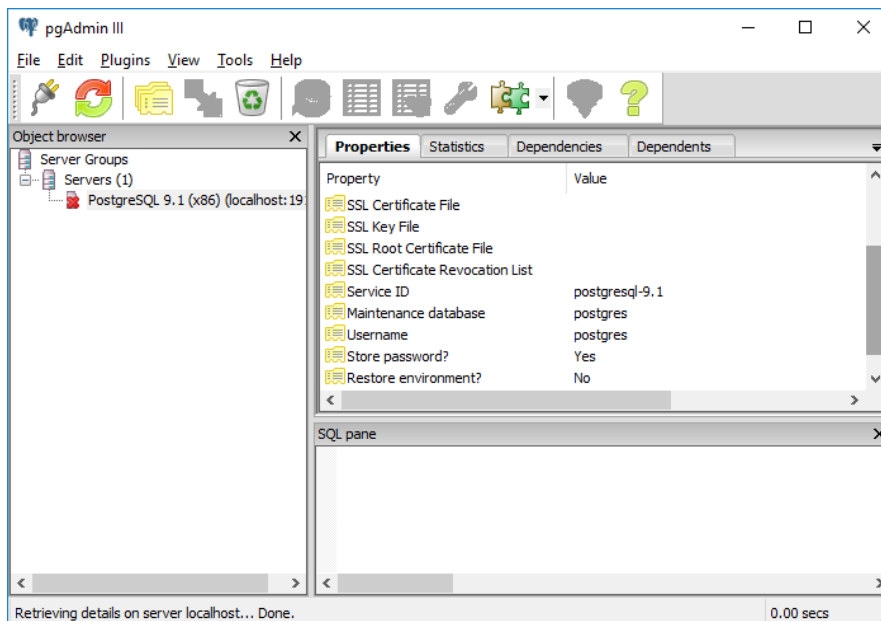


Gambar 2. Tampilan PSQL untuk manajemen database PostgreSQL

```
CREATE TABLE new_table_name (
    table_column_title TYPE_OF_DATA column_constraints,
    next_column_title TYPE_OF_DATA column_constraints,
    table_constraint
    table_constraint
) INHERITS existing_table_to_inherit_from;
```

Gambar 3. Contoh pembuatan tabel di psql

sedangkan secara GUI, pengguna dapat menggunakan pgAdmin III seperti yang ditunjukkan oleh gambar 4. Dengan pgAdmin III pembuatan objek-objek yang diperlukan cukup klik fitur-fitur yang disediakan sehingga tidak perlu menggunakan syntax-syntax SQL.

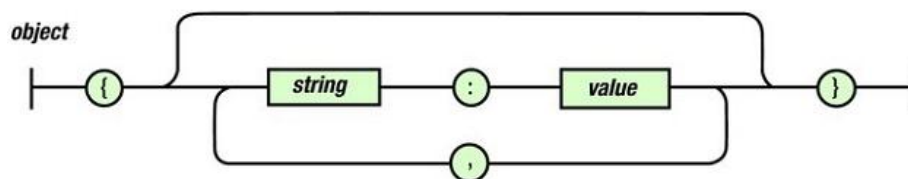


Gambar 4. Tampilan pgAdmin III

2.4 JSON

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON adalah format teks yang tidak bergantung pada bahasa pemrograman apapun sehingga dapat digunakan di bahasa pemrograman C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON sangat ideal sebagai bahasa pertukaran data [5].

JSON memiliki dua buah struktur yang diwakili dalam bentuk objek dan array. Crockford menguraikan dua penggambaran struktur JSON melalui serangkaian sintaks diagram. Diagram ini menyerupai rel kereta api dari tampilan yang merangkum dan dengan demikian juga dirujuk sebagai diagram kereta api. Gambar 5 menggambarkan representasi tata bahasa untuk koleksi pasangan nilai string [8].



Gambar 5. Diagram sintaks koleksi pasangan nilai string

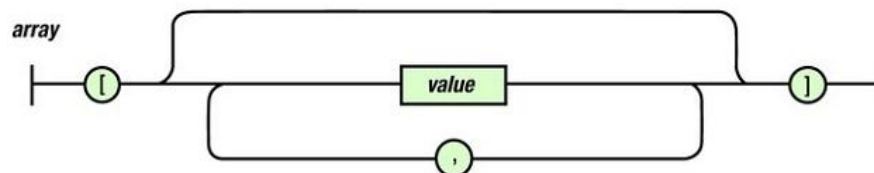
(Sumber: Smith Ben. Beginning JSON, The Expert's Voice in Web Development. New York: Apress. 2015. Page:56)

Seperti yang ditunjukkan oleh gambar 5 objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap string diikuti dengan : (titik dua) dan setiap pasangan string/nilai dipisahkan oleh , (koma). Listing 1 berikut adalah contoh dari *object*.

Listing 1. Contoh *object* dari JASON

```
//Empty Collection Set
{};
//Single string/value pair
{"abc":"123"};
//Multiple string/value pairs
{"captainsLog":"starDate 9522.6","message":"I've never trusted Klingons, and I never will."};
```

Pada gambar 6 menggambarkan representasi tata bahasa kumpulan nilai-nilai yang terurutkan atau array. Di sini kita bisa menyaksikan bahwa daftar urutan dimulai dengan menggunakan bracket terbuka ([]) dan berakhir dengan menggunakan bracket dekat (]).



Gambar 6. Diagram sintaks pengurutan array

(Sumber: Smith Ben. Beginning JSON, The Expert's Voice in Web Development. New York: Apress. 2015. Page:57)

Listing 2 berikut ini adalah contoh yang merepresentasikan array berdasarkan tata bahasa JSON:

Listing 2. Contoh array berdasarkan tata bahasa JSON

```
//Empty Ordered List
[];
//Ordered List of multiple values
["abc"];
//Ordered List of multiple values
["0",1,2,3,4,100];
```

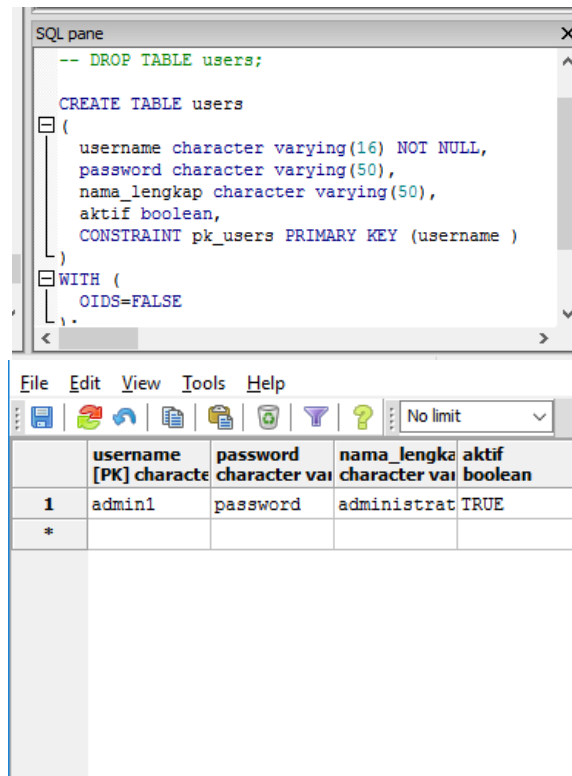
Format JSON sebenarnya kalau diperhatikan sama dengan fungsi sebuah tabel yang berisi kolom dan baris atau bisa juga hanya kolom dan baris saja, untuk lebih jelas listing 3 berikut adalah contoh dari JASON:

Listing 3. Contoh string JSON

```
{ "mahasiswa": [ { "nama": "budi", "alamat": "sidokare" }, { "nama": "toni", "alamat": "bungurasih" }, { "nama": "selamat", "alamat": "wage" } ] }
```

pada contoh yang ditunjukkan listing 3, memuat data mahasiswa yang berisi nama dan alamat sejumlah tiga baris data. Dari contoh tersebut terlihat bahwa JSON merepresentasikan sebuah objek tabel dengan kolom nama dan alamat.

Pada penelitian ini, penulis mencoba menerapkan JSON untuk meminimasi jumlah kolom pada tabel, sebagai contoh tabel user dimana terdapat hak akses untuk masuk ke beberapa aplikasi dan menu didalamnya. Pada percobaan ini penulis menggunakan database PostgreSQL dan bahasa pemrograman PHP. Desain awal dari tabel ini memiliki kolom username, password, nama_lengkap, aktif seperti yang diperlihatkan oleh gambar 7.



Gambar 7. Struktur tabel users

struktur tabel yang ditunjukkan gambar 7 tersebut merupakan struktur tabel yang biasa digunakan untuk manajemen user suatu aplikasi. Hal ini tidak akan menjadi masalah apabila tabel ini digunakan untuk masuk ke satu aplikasi yang tidak membutuhkan batasan pengaksesan menu-menu tiap user didalamnya. Namun apabila aplikasi yang manajemen usernya menggunakan tabel tersebut membutuhkan rule untuk tiap user dalam hal pengaksesan menu-menu yang ada di dalam aplikasi tersebut sebagai contoh aplikasi keuangan yang didalamnya terdapat menu pengajuan, pencairan dan pelaporan dengan rule yang disajikan tabel 1. Maka kita harus merubah struktur tabel yang ada dengan menambahkan kolom pengajuan, pencairan dan pelaporan, kolom-kolom ini nantinya bisa berisi angka 1 jika diberi akses angka 0 jika tidak diberi akses.

Apabila suatu saat aplikasi keuangan ini dikembangkan dan terdapat penambahan menu maka apabila kita memakai metode menambahkan kolom hak akses tambahan, hal ini jelas sangat tidak efisien dan merepotkan administrator database nantinya. Disamping itu juga akan terjadi pemborosan penggunaan resource database karena tiap user mempunyai hak akses yang berbeda.

Tabel 1. Rule hak akses user

username	pengajuan	pencairan	Pelaporan
admin	√	√	√
alfan	√	√	
rosid	√		√
yazid			√

Dari tabel 1 terlihat bahwa masing-masing user mempunyai hak akses yang berbeda-beda. Kolom pengajuan, pencairan dan pelaporan digunakan untuk menentukan hak akses dari tiap user. Maka dari itu penulis mencoba merubah metode penambahan kolom hak akses ini dengan format JSON.

Listing 4. Sintax PHP merubah ke format JSON

```
<?php
$a = array('pengajuan','pencairan','pelaporan');
$akses=array("KEUANGAN"=>$a);
echo json_encode($akses);
?>
```

Listing 4 merupakan sintax PHP yang merubah array ke dalam format JSON dengan fungsi **json_encode**, seperti yang sudah dijelaskan diawal bahwa array tidak dapat disimpan di database sehingga harus diubah formatnya terlebih dahulu. Listing 4 menghasilkan format JSON pada listing 5 sebagai berikut:

Listing 5. Format JSON hasil listing 4

```
{"KEUANGAN":["pengajuan","pencairan","pelaporan"]}
```

dengan format yang ditunjukkan listing 5 maka sebuah user mempunyai hak akses ke aplikasi keuangan dan dapat mengakses menu pengajuan, pencairan dan pelaporan, apabila hak akses hanya pengajuan saja maka sintax untuk member aksesnya menjadi seperti listing 6.

Listing 6. Sintax PHP merubah ke format JSON 2

```
<?php
$a = array('pengajuan');
$akses=array("KEUANGAN"=>$a);
echo json_encode($akses);
?>
```

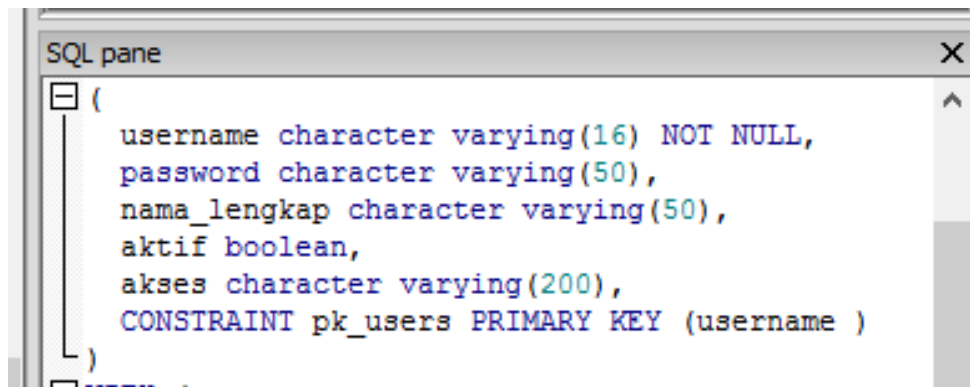
Listing 6 akan menghasilkan format JSON {"KEUANGAN":["pengajuan"]}. Format ini dapat disimpan di satu kolom pada tabel user seperti yang ditunjukkan oleh tabel 2.

Tabel 2. Rule hak akses user format JSON

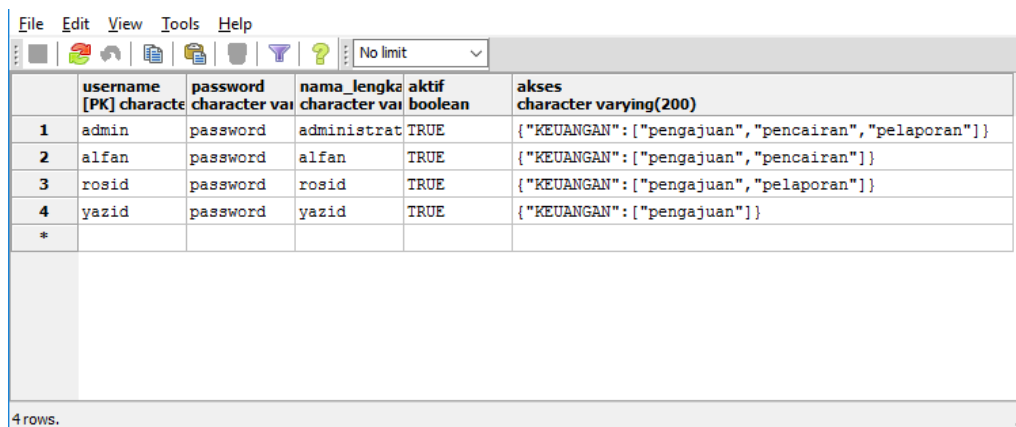
username	Akses
admin	{"KEUANGAN":["pengajuan","pencairan","pelaporan"]}
alfan	{"KEUANGAN":["pengajuan","pencairan"]}
rosid	{"KEUANGAN":["pengajuan","pelaporan"]}
yazid	{"KEUANGAN":["pelaporan"]}

3. Hasil Penelitian dan Pembahasan

Dari percobaan yang dilakukan didapatkan pengurangan penggunaan kolom di tabel user, yang awalnya membutuhkan tiga buah kolom menjadi hanya satu kolom saja, pada tabel 2 posisi kolom pengajuan, pencairan dan pelaporan digantikan dengan kolom akses yang menyimpan string JSON. Struktur tabel user setelah menerapkan JSON menjadi seperti gambar 8.



Gambar 8. Struktur tabel sesudah menerapkan JSON



	username [PK] character varying(16)	password character varying(50)	nama_lengkap character varying(50)	aktif boolean	akses character varying(200)
1	admin	password	administrat	TRUE	{"KEUANGAN":["pengajuan","pencairan","pelaporan"]}
2	alfan	password	alfan	TRUE	{"KEUANGAN":["pengajuan","pencairan"]}
3	rosid	password	rosid	TRUE	{"KEUANGAN":["pengajuan","pelaporan"]}
4	yazid	password	yazid	TRUE	{"KEUANGAN":["pengajuan"]}
*					

Gambar 9. Contoh isi tabel user sesudah menerapkan JSON

Dari sini untuk mengembalikan string JSON ini ke bentuk semula dapat menggunakan fungsi `json_decode` pada bahasa pemrograman PHP. Hasil dari `decode` ini selanjutnya dapat diolah untuk mengatur hak akses user sesuai data rule yang tersimpan di kolom akses tabel user. Contoh sintaks `decode` JSON dapat dilihat di listing 7.

Listing 7. Contoh sintaks `decode` pada bahasa pemrograman PHP.

```
$login=pg_fetch_array(pg_query("SELECT * FROM user WHERE
username='$username' AND password='$pass'"));
$access = json_decode($login['akses'],true);
    if(is_array($access)){
        if(array_key_exists('KEUANGAN',$access)){
            $akses = $access;
        }
    }
    $sacc = $akses['KEUANGAN'];
    echo"<ul>";
        if(in_array('pengajuan',$sacc)){?>
            <li><a href="pengajuan">Pengajuan</a></li>
        <?php }
        if(in_array('pencairan',$sacc)){?>
            <li><a href="pencairan">Pencarian</a></li>
        <?php }
        if(in_array('pelaporan',$sacc)){?>
            <li><a href="pelaporan">Pelaporan</a></li>
        <?php }
    echo"</ul>";
```

perintah `$access = json_decode($login['akses'],true);` berfungsi untuk merubah JSON ke array, `if(array_key_exists('KEUANGAN',$access))` akan memeriksa apakah user memiliki hak akses untuk masuk ke aplikasi KEUANGAN, sedangkan `if(in_array('pengajuan',$sacc))` memeriksa apakah user mempunyai akses menu pengajuan. Dengan menerapkan metode ini, maka ketika terjadi penambahan hak akses maka cukup menambahkan nilai di array-nya saja, sebagai contoh ada penambahan menu disposisi dari contoh gambar 9 maka tinggal menambahkan ke JSON menjadi

Listing 8. Format JSON setelah ditambah menu disposisi

```
{"KEUANGAN":["pengajuan","pencairan","pelaporan","disposisi"]}
```

Pada Listing 8 adalah pemberian hak akses user untuk mengakses aplikasi keuangan menu pengajuan, pencairan, pelaporan dan disposisi. Format JSON juga dapat digunakan untuk memberi akses user ke aplikasi lain misalnya aplikasi JURNAL maka bentuk format JSON-nya seperti listing 9 berikut:

Listing 9. Format JSON setelah ada penambahan hak akses aplikasi jurnal.

```
{"KEUANGAN":["pengajuan","pencairan","pelaporan","disposisi"],"JURNAL":[]}
```

4. Kesimpulan

JSON pada dasarnya adalah sebuah format pertukaran data yang ringan dan jauh lebih cepat dibandingkan XML, dari hasil penelitian, JSON ternyata juga dapat digunakan untuk menyelesaikan kasus yang memerlukan tabel dengan kolom dinamis atau tabel tersebut berpotensi bertambah sesuai kebutuhan aplikasi yang menggunakan tabel tersebut. JSON dapat mereduksi dengan hanya menggunakan sebuah kolom yang berisi string format JSON. Untuk merubah array menjadi format JSON menggunakan fungsi `json_encode`, sedangkan untuk mengembalikan ke bentuk semula dapat menggunakan fungsi `json_decode`.

Referensi

- [1] E-Aziz A.Abd, Kannan A. JSON Encryption. In Proceedings of 2014 International Conference on Computer Communication and Informatics (ICCCI -2014), Jan. 03 – 05, 2014, Coimbatore, INDIA, 2014
- [2] Fathansyah, Basis Data, Informatika, Bandung, 2007.
- [3] G. Wang. Improving Data Transmission in Web Applications via the Translation between XML and JSON. In Proceedings of 2011 Third International Conference on Communications and Mobile Computing (CMC), pages 182 – 185, 18-20 April, 2011
- [4] J. Salahadin, V. Achim, V. Andrey. Learning PostgreSQL, Create, Develop, and manage relational databases in real-world applications using PostgreSQL. Birmingham-Mumbai:PACKT Publishing. 2015.
- [5] Jackson W, JSON Quick Syntax Reference, The Expert's Voice in Web Development. Lompoc, California, USA:Apress. 2016.
- [6] L. Griffin, B. Butler, E. de Leastar, B. Jennings, and D. Botvich. On the Performance of Access Control Policy Evaluation. In Proceedings of 2012 IEEE International Symposium on Policies for Distributed Systems and Networks (POLICY), pages 25 – 32, 16-18 July, 2012.
- [7] Munawaroh Siti. 2005. Mengeksplorasi Database PostgreSQL dengan PgAdmin III. Jurnal Teknologi Informasi DINAMIK.
- [8] Smith Ben. Beginning JSON, The Expert's Voice in Web Development. New York: Apress. 2015