

APLIKASI CLOUD SQL BERBASIS WEB

Rizky Sulistyo¹, Aan Erlansari², Funny Farady Coastera³

^{1,2,3}Program Studi Teknik Infomatika, Fakultas Teknik, Universitas Bengkulu.
Jl. WR. Supratman Kandang Limun Bengkulu 38371A INDONESIA
(telp: 0736-341022; fax: 0736-341022)

¹rizkylab@gmail.com,
²ffaradyc@unib.ac.id,
³aan_erlanshari@unib.ac.id

Abstrak : Aplikasi cloud *SQL* berguna untuk memudahkan administrator database untuk manajemen database secara terpusat. Pada umumnya, aplikasi manajemen database harus diinstall pada masing – masing server database, sehingga perlu aplikasi cloud *SQL* agar dapat terpusat. Aplikasi ini menggunakan arsitektur *software as a service* dengan tujuan membagi resource database dalam bentuk *json API*. Aplikasi ini dibangun dengan bahasa pemrograman node.js. Sedangkan metode pengembangan sistem yang digunakan adalah *Unified Modeling Language* (UML) sebagai perancangan sistem. Implementasi *cloud* pada aplikasi ini dengan menggunakan model *private cloud*. Pada penelitian ini diujicobakan untuk mengintegrasikan 4 database server yang dapat diolah pada aplikasi *cloud*. Hasil akhir dari penelitian ini adalah sebuah aplikasi cloud yang dapat manajemen database MySQL dan PostgreSQL, dan fungsi cloud sebagai *cloud software as a service* adalah dapat berbagi *resource* data pada database yang diolah dalam bentuk *json API*.

Kata Kunci: aplikasi *cloud*, *private cloud*, *json API*, *cloud software as a service*, Node.js, MySQL, PostgreSQL.

Abstract: Cloud *SQL* application is beneficial to ease database administrator to manage databases centralized. Generally, database management application ought to be install to each database server, such that it is required a cloud *SQL* application to be centralized. This application utilize software as a service architecture in order to deploy database resource in a form *Json API*. This application is built by using Node.js programming language and system development method is *Unified Modelling Language* (UML) as the system design. Cloud implementation on this application is the cloud private model. On this study there are 4 tested server database, so that it will be managed at cloud application. The end

result of this research is a cloud application, capable of administering MySQL and PostgreSQL database, where the cloud is software as a service is employed to share data resource within an organized *JSON-API* database.

Keywords: *cloud application*, *private cloud*, *json API*, *cloud software as a service*, Node.js, MySQL, PostgreSQL

I. PENDAHULUAN

A. Latar Belakang

Berkembangnya teknologi informasi yang sangat pesat, khususnya *cloud computing* sangatlah mempermudah pengguna dalam melakukan pekerjaannya sehari-hari. Hal ini

memacu para pengguna *cloud* untuk saling berlomba dalam menggunakan dengan teknologi yang baik dan sesuai dengan kebutuhan. *Cloud computing* memiliki beberapa keunggulan dibandingkan dengan sistem konvensional. Perhatian utama mengenai *cloud computing* ini adalah masalah kehandalan. Apakah sistem ini mampu memberikan kebutuhan komputasi sesuai yang dibutuhkan dan memberikan stabilitas. Membagi beberapa proses secara bersama-sama, mengirimkan dan menerima file secara bersama-sama adalah sebagian resiko dari pada server *cloud computing*.

Dalam beberapa tahun terakhir, *cloud computing* (komputasi awan) menjadi tren teknologi baru atas efisiensi penggunaan teknologi informasi dengan memanfaatkan perkembangan Internet. *Cloud computing* pada dasarnya adalah Internet-based service untuk berbagi sumber daya komputasi yang terkonfigurasi, baik dari sisi jaringan, server, aplikasi, penyimpanan dan layanan. *Cloud SQL* menawarkan kolaborasi arsitektur *Software as a Service* (SaaS), serta kemampuan akses kapan pun dan di mana pun.

Dengan adanya *cloud computing* maka dapat dibangun suatu arsitektur layanan pengolahan database dengan menggunakan *Cloud SQL*. Aplikasi ini nantinya menerapkan *cloud computing*, yaitu berupa layanan SaaS dalam bentuk *Database As a Service* (DBaaS) dan sebagai media pengolahan database. Di dalam arsitektur layanannya, *Cloud SQL* akan menyediakan infrastruktur dan pengolahan database secara terpusat di *cloud server*. Pengguna tidak perlu tahu bagaimana dan darimana sumber daya komputasinya dipenuhi oleh *cloud provider*. Selain itu untuk manajemen database, setiap database server tidak perlu direpotkan dengan

penyediaan aplikasi sendiri yang harus terinstall di masing – masing *server*. Administrator server cukup menggunakan layanan pada *Cloud SQL* yang nantinya bisa mengolah database. Harapan dengan adanya *Cloud SQL* adalah agar terciptanya keseragaman dan kemudahan dalam hal pengolahan database sehingga akses menjadi lebih cepat, mudah, terpusat, dapat dilakukan kapan saja dan di mana saja. Berdasarkan latar belakang yang telah dijelaskan, maka penelitian tugas akhir ini diberi judul yaitu Aplikasi *Cloud SQL* Berbasis Web.

B. Rumusan Masalah

Berdasarkan latar belakang di atas adapun yang menjadi rumusan masalah adalah “Bagaimana Membangun Aplikasi *Cloud SQL* Berbasis Web?”.

C. Batasan Masalah

Batasan masalah dalam penelitian ini sebagai berikut:

1. Aplikasi dibuat pada perangkat yang menggunakan sistem operasi berbasis Linux.
2. Aplikasi yang dibangun untuk manajemen database berupa editor yang dapat mengeksekusi perintah - perintah (*query*) pada SQL.
3. Database yang di manage pada pada aplikasi *cloud* hanya MySQL dan PostgreSQL.
4. Jenis *Cloud* dibangun dengan menggunakan *private cloud*
5. *Cloud* berbasis *software as a service*.

D. Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah :

1. Merancang dan membangun aplikasi *cloud SQL* berbasis web.
2. Mengimplementasikan layanan *software as a service* dalam berbagi resource database yang nantinya bisa digunakan oleh siapapun jika diperlukan.

E. Manfaat Penelitian

Adapun manfaat yang ingin dicapai dari penelitian ini adalah :

1. Dapat memberikan kemudahan dalam manajemen database MySQL dan PostgreSQL secara terpusat.

II. TINJAUAN PUSTAKA

A. Cloud Computing

Cloud computing adalah sebuah arsitektur teknologi informasi yang dimana sumber daya komputasi tersedia sebagai layanan yang dapat diakses melalui internet (Sasono: 2011).

Cloud pada *Cloud Computing* merupakan penyedia atau hal-hal yang berkaitan dari tenaga komputasi hingga infrastruktur komputasi, aplikasi-aplikasi, proses bisnis hingga kolaborasi yang muncul sebagai layanan yang dapat diakses pada saat dibutuhkan kapanpun dan dimanapun. Teknologi komputasi komputer dengan memanfaatkan internet sebagai terminal utamanya guna mengelola piranti lunak hingga infrastruktur sebagai suatu bentuk layanan. *Cloud computing* pada dasarnya adalah menggunakan *internet-based service* untuk mendukung proses bisnis.

B. Model Layanan Cloud Computing

Secara umum, layanan pada *Cloud Computing* dibagi menjadi tiga model layanan, yaitu aplikasi/perangkat lunak, *platform*, dan infrastruktur (*Software as a Service*, *Platform as a Service*, dan *Infrastructure as a Service*).

1. *Software as a Service* (SaaS)

Model layanan *Software as a Service* (SaaS) adalah model layanan yang paling banyak dikembangkan untuk sistem *cloud computing*. Model layanan ini berupa aplikasi atau *software* berbasis web, yang diberikan kepada berbagai pengguna oleh *vendor* atau pemilik sistem tersebut. Pengguna pun tidak perlu memiliki aplikasi tersebut untuk menggunakannya, melainkan pengguna memerlukan koneksi internet untuk dapat mengakses ataupun menggunakannya. Contoh perusahaan yang mengembangkan layanan ini adalah Google dengan contoh aplikasinya yaitu Google Docs, yang berfungsi sebagai *word processor*, *spreadsheet*, *presentation creator*.

2. Platform as a Service (PaaS)

Pada lapis *platform*, kita dapat mendeploy aplikasi yang kita buat di *Cloud*. Menyediakan *platform* (bahasa pemrograman, Tools dsb) guna pengembangan aplikasi berbasis konsumen pada infrastruktur *Cloud*.

Pengguna dapat membuat aplikasi, dan aplikasi tersebut pengguna host di *Cloud provider*. Atau pengguna tidak perlu menginstall perangkat lunak untuk membuat aplikasi, namun dapat mendesain aplikasi, membangun, mendeploy, dan meng-host aplikasi di internet. PaaS juga dikenal dengan *Cloudware*. Artinya model layanan *platform as a service* merupakan layanan yang menyediakan *platform* seperti bahasa pemrograman, *tools*, ataupun sistem operasi yang nantinya mampu mengembangkan aplikasi berbasis konsumen. Keuntungan dari PaaS adalah bahwa perusahaan yang menggunakan layanan ini tidak perlu khawatir untuk instalasi, perawatan, dan keamanan pada server karena *provider* PaaS akan

menangani itu semua. Contoh implementasi dari PaaS adalah Microsoft Windows Azure dan Google App Engine.

3. *Infrastruktur as a Service* (IaaS)

Pada lapis infrastruktur ini, menyediakan sumber daya pemroses, *storage* atau penyimpanan, kapasitas jaringan, dan sumber daya komputasi lainnya. Dimana konsumen dapat mengembangkan dan menjalankan aplikasi khusus seakan-akan mempunyai perangkat keras dan segala isinya pada *remote server*, termasuk perangkat lunak di dalamnya. Maka dari itu, sinonim dari layer ini adalah *Everything as a Service*.

C. Jenis *Cloud Computing*

1. Public Cloud

Model pengembangan ini adalah model pengembangan yang paling populer. Model ini diimplementasikan kepada berbagai server yang berjalan pada berbagai data center agar dapat diakses di belahan bumi manapun. Model pengembangan ini telah dilakukan oleh salah satunya adalah Google.

2. Private Cloud

Model pengembangan ini biasanya dilakukan oleh suatu perusahaan atau organisasi dengan memegang atribut dari cloud computing yaitu, elastisitas, Self Service, Pay-by-Use, dan programmability. Layanan ini berada pada jaringan LAN (bisa berupa suatu perusahaan, instansi, sekolah yang berada pada area yang sama), dan biasanya untuk *private cloud* bersifat terbatas, atau artinya hanya orang tertentu yang bisa mengaksesnya.

3. Hybrid Cloud

Model ini merupakan kombinasi atau gabungan dari model public dan private yang

masih memiliki entitas unik namun terikat bersama-sama oleh standar teknologi atau kepemilikan data dan portabilitas aplikasi.

4. Community Cloud

Model ini diimplementasikan ketika beberapa perusahaan atau organisasi memiliki kesamaan konteks data.

D. Node.js

Node.js adalah sebuah *platform* yang dirancang untuk *webserver*. Aplikasi ini ditulis dalam bahasa javascript dan berbasis pada *event* (*event driven*). Tidak seperti kebanyakan bahasa javascript yang dijalankan di *client*, Node.js dieksekusi sebagai aplikasi *server*. Node.js memiliki efisiensi *memory* yang lebih baik apabila bekerja dalam keadaan muatan yang banyak. Pengguna tidak perlu khawatir dengan terjadinya *deadlock* karena tidak ada *lock* dalam Node.js. Aplikasi ini terdiri dari V8.

Javascript *engine* buatan google dan beberapa modul bawaan yang terintegrasi (Nodejs.org). Di dalam Node.js terdapat operasi-operasi yang sifatnya *synchronous* dan *asynchronous*, tetapi keunggulan Node.js ini terdapat pada *asynchronousnya* atau bisa juga disebut dengan *non-blocking I/O*. Node.js menggunakan pendekatan *event-driven* berbasis *infinite event loop* dalam satu *thread* untuk mengurangi jumlah *memory*. Manuel Kiessling (2012) berpendapat bahwa node.js merupakan model yang efisien untuk membangun sebuah aplikasi yang harus bersinggungan dengan *concurrency* (dua hal yang bekerja secara bersamaan).

Semenjak kemunculannya, Node.js telah menyita perhatian dari kalangan industri. Banyak pelaku industri yang menggunakan *node* untuk membuat layanan jaringan yang cepat dan

scalable. Salah satu alasannya adalah karena node.js menggunakan javascript. Javascript adalah salah satu bahasa pemrograman yang paling banyak digunakan. Kebanyakan programmer sudah terbiasa menggunakan javascript pada *browser* (Teixeira, P. 2013, p4).

E. Teknologi *Rest*

Ada beberapa pendekatan teknologi web services yang dapat digunakan, yaitu metode SOAP dan REST. Dalam penelitian ini akan digunakan pendekatan REST. REST merupakan singkatan dari REpresentative State Transfer. Pertama kali dikemukakan dalam disertasi seorang program doktor bernama Roy Thomas Fielding pada tahun 2000 (Rozali, 2011). REST adalah sebuah metode dalam menyampaikan resource melalui media web. Sedangkan resource sendiri didefinisikan sebagai segala sesuatu yang dapat disimpan didalam sebuah komputer dan ditampilkan sebagai urutan bit, misalnya sebuah dokumen, tabel dalam sistem basis data, atau hasil dari sebuah perhitungan (Sandoval, 2009).

REST (*Representational State Transfer*) adalah model arsitektur yang pada dasarnya memanfaatkan teknologi dan protokol yang sudah ada seperti HTTP (*Hypertext Transfer Protocol*) dan XML (Riyadi, 2013).

F. Basis Data

Basis Data merupakan file yang berisikan tabel-tabel yang saling berinteraksi sehingga dapat diproses dan digunakan dengan cepat dan mudah. Tabel merupakan kumpulan data yang tersusun menurut aturan tertentu dan berhubungan dengan topik tertentu. Tabel diorganisasikan dalam dua bagian, bagian menurun atau kolom disebut dengan *field* dan bagian mendatar atau baris disebut dengan *record*. Berkas basis data

dibuktikan dengan adanya media penyimpanan fisik dengan adanya media penyimpanan fisik berupa *magnetic-tipe*, piringan (disket, CD).

Adapun Basis Data yang digunakan yaitu MySQL dan PostgreSQL, yang merupakan salah satu *Basis Data Management System* (DBMS) dari sekian banyak DBMS seperti Oracle, MS SQL, Vertica, dan lainnya. MySQL berfungsi untuk mengolah Basis Data menggunakan bahasa SQL.

MySQL merupakan Basis Data yang paling digemari kalangan programmer web, dengan alasan bahwa program ini merupakan Basis Data yang sangat kuat dan cukup stabil untuk digunakan sebagai media penyimpanan data. Sebagai sebuah Basis Data server yang mampu untuk manajemen Basis Data dengan baik, mysql terhitung merupakan Basis Data yang paling digemari dan paling banyak digunakan dibanding Basis Data lainnya. Selain mysql masih terdapat beberapa jenis Basis Data server yang juga memiliki kemampuan yang juga tidak bisa dianggap enteng, Basis Data itu adalah PostgreSQL. (Nugroho, 2005).

PostgreSQL, atau yang sering disingkat menjadi pgsq dan pg* untuk yang berhubungan dengannya, merupakan sistem basis data *open source* yang sangat handal dan mampu menangani objek data yang sangat besar. Banyak yang mempercayai PostgreSQL saat ini menjadi sistem database yang paling baik dan efektif, tentu saja tidak terlepas dari harga yang gratis dan mampu bekerja sama baiknya di hampir seluruh sistem operasi saat ini, Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), dan Windows.

III. METODE PENELITIAN

A. Jenis Penelitian

Penelitian terapan ini bertujuan untuk merancang dan membangun aplikasi *cloud SQL* berbasis web yang nantinya dapat mengintegrasikan server database berbasis MySQL dan PostgreSQL serta manajemen database secara terpusat.

B. Jenis dan Sumber Data

Dalam mengumpulkan data, teknik pengumpulan data yang digunakan dalam penelitian ini yaitu:

Perangkat server yang digunakan pada sistem *Cloud SQL* beserta database server yang berjumlah tiga perangkat server.

Studi pustaka dilakukan dengan mengumpulkan data dan informasi yang dijadikan sebagai acuan perancangan aplikasi *Cloud SQL*. Data dan informasi tersebut diantaranya: tutorial pemrograman jaringan menggunakan bahasa pemrograman Nodejs, ResfulAPI, web service, desain diagram UML, instalasi database server, dan sumber-sumber tertulis baik tercetak maupun elektronik yang berhubungan dengan penelitian ini.

C. Metode Pengembangan Sistem

Sistem yang dikembangkan dalam penelitian ini menggunakan model pengembangan sistem *sekuensial*, tahap-tahap yang akan dilakukan adalah :

1. Pada tahap pertama adalah penentuan basis awal dari sebuah penelitian yaitu latar belakang penelitian.
2. Tahap kedua yaitu menentukan tujuan, manfaat dan ruang lingkup penelitian. Target pencapaian dalam tahap ini adalah diketahuinya tujuan dan manfaat dari aplikasi *cloud SQL*. Sedangkan batasan masalah digunakan untuk membatasi pembahasan dan

ruang lingkup penelitian agar tidak terlalu luas.

3. Pada tahap ketiga dilakukan pengumpulan data-data yang diperlukan dalam penelitian, meliputi jenis data masukan dan keluaran, spesifikasi fungsi, kemampuan dan fasilitas dari aplikasi yang akan dibangun serta data-data deskripsi maupun abstrak hasil penelitian. Pengumpulan data tersebut akan diperoleh dari proses dokumentasi dan studi pustaka.
4. Tahap keempat adalah proses analisis kebutuhan. Analisis kebutuhan yang ditentukan oleh peneliti terdiri dari analisis kebutuhan sistem, analisis kebutuhan proses, analisis kebutuhan input, analisis kebutuhan output, dan analisis kebutuhan perangkat lunak dan perangkat keras.
5. Tahap kelima yaitu perancangan *Unified Modelling Language* (UML). Pada tahapan ini akan diketahui semua entitas luar, input, dan output yang terlibat dalam sistem serta diagram *use-case*, diagram kelas, diagram aktifitas, dan diagram *sequence* yang digunakan dalam analisis sistem.
6. Tahap ketujuh yaitu perancangan *flowchart* dan antarmuka (*human interface*). Tahap perancangan *flowchart* akan digunakan untuk menggambarkan sistem baru yang akan dikembangkan secara logis tanpa mempertimbangkan terlebih dahulu lingkungan sistem. Tahap perancangan antarmuka akan dibuat dalam beberapa rancangan tata letak system sesuai dengan analisis kebutuhan dari sistem.
7. Tahapan kedelapan yaitu implementasi sistem secara keseluruhan. Implementasi berdasarkan hasil dari tahapan keempat hingga ketujuh, yaitu dari tahapan analisis

- kebutuhan, UML, *flowchart* dan antarmuka. Pada tahap ini juga dilakukan pengkodean menggunakan node.js dan *restful* pada sistem.
8. Tahapan kesembilan yaitu tahapan pengujian hasil implementasi sistem secara umum. Pengujian meliputi pengujian input, proses dan output system apakah telah sesuai dengan perancangan sistem sebelumnya. Pada tahapan ini juga dilakukan pengujian pada proses integrasi database server ke *cloud SQL*. Tahapan ini menguji implementasi proses manajemen database, menjalankan dan mengeksekusi proses *query* pada sistem, serta menguji implementasi web service pada *cloud SQL*. Proses pengujian ini dilakukan menggunakan metode *black-box testing* dan *white-box testing*.
 9. Tahapan kesebelas adalah operasi dan pemeliharaan sistem. Tahapan ini dilakukan setelah proses pengujian sistem telah berhasil dilakukan.
 10. Tahapan terakhir adalah penarikan kesimpulan dan saran. Tahap tersebut dilakukan untuk mengetahui hasil yang telah diperoleh selama melakukan penelitian.

D. Metode Pengujian

Metode pengujian sistem yang digunakan dalam penelitian ini adalah *black-box testing* dan *white-box testing*.

Hal yang akan diuji pada pengujian *black-box* antara lain:

1. Pengujian fungsional sistem dari aplikasi *cloud SQL*.
 2. Pengujian proses integrasi 4 database server pada aplikasi *cloud SQL*.
 3. Pengujian untuk mengamati proses *query* pada setiap database baik MySQL dan PostgreSQL.
 4. Pengujian pada fitur exporting file json dari hasil *query* kedalam bentuk .csv atau excel.
 5. Pengujian untuk mengamati jumlah penggunaan *running time* pada setiap *query* yang di jalankan.
 6. Pengujian untuk mengamati implemetasi dari web service.
- Sedangkan hal yang akan diuji pada *white-box testing* antara lain:
1. Pengujian *method* dan *attribute* pada aplikasi *cloud SQL*.
 2. Pengujian antarmuka aplikasi *cloud SQL*.

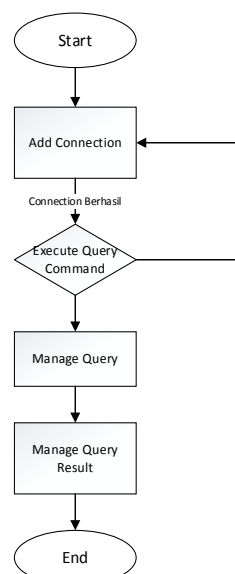
IV. ANALISIS DAN PERANCANGAN SISTEM

A. Analisis Fungsional

Analisis fungsional merupakan analisis yang dilakukan untuk menganalisis fungsi apa yang dapat dilakukan oleh sistem nantinya. Fungsi pada aplikasi *cloud SQL* ini adalah untuk memudahkan administrator database untuk manage database MySQL dan PostgreSQL secara terpusat dan dapat melakukan *sharing* data kepada siapapun yang membutuhkan data dalam cloud tersebut.

B. Analisis Alur Kerja Sistem

Adapun diagram alir kerja sistem dapat dilihat pada Gambar 4.1.

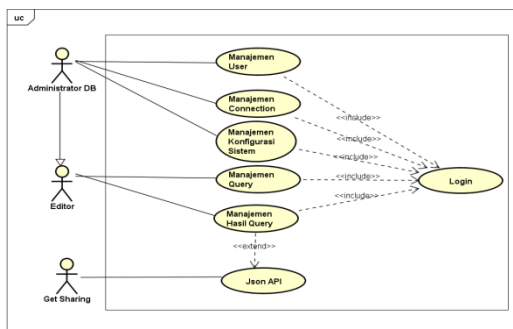


Gambar 4.1 Alur Kerja Sistem

C. Perancangan Unified Modelling Language (UML)

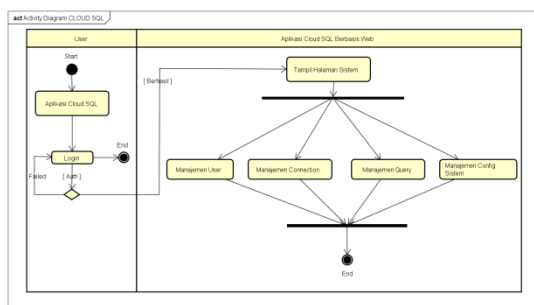
1. Use Case Diagram

Use case diagram terdiri dari sebuah aktor dan interaksi yang dilakukannya. Aktor tersebut dapat berupa manusia, perangkat keras, sistem lain, ataupun yang berinteraksi dengan sistem. Aktor dalam hal ini bertindak sebagai pengguna yang melakukan interaksi terhadap sistem. Interaksi yang dilakukan diantaranya melakukan login, manajemen *user*, manajemen *connection*, manajemen *query*, dan manajemen konfigurasi sistem. Adapun *Use case diagram* dari aplikasi *cloud SQL* dapat dilihat pada Gambar 4.2 :



Gambar 4.2 Use Case Diagram

2. Activity Diagram Administrator

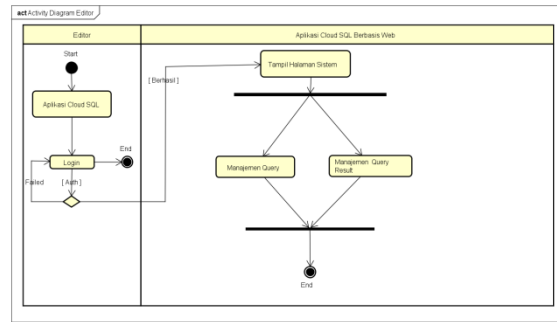


Gambar 4.3 Activity Diagram Administrator

Pada *activity diagram* aplikasi *cloud SQL* berbasis web penggambaran diatas, diagram menunjukkan bagaimana aktifitas

yang terjadi ketika Administrator menjalankan aplikasi tersebut.

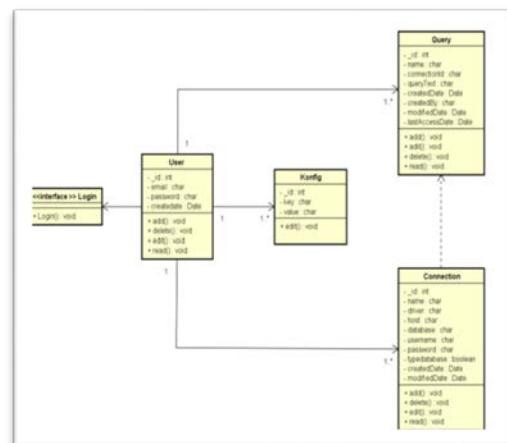
3. Activity Diagram Editor



Gambar 4.4 Activity Diagram Editor

Diagram menunjukkan bagaimana aktifitas yang terjadi ketika *editor* menjalankan aplikasi tersebut.

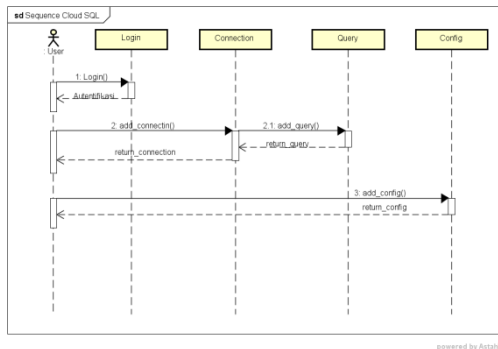
4. Class Diagram



Gambar 4.5 Class Diagram

Class Diagram menggambarkan keadaan (*attribute/property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (*metode/fungsi*). *Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Pada sistem ini disetiap menu yang ditampilkan memiliki operasi dan atribut-aribut yang mengikuti.

5. Sequence Diagram



Gambar 4.6 Sequence Diagram

Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh objek-objek yang melakukan suatu tugas atau aksi tertentu. Pada sistem ini akan ditentukan langkah-langkah dari pengerjaan sistem sesuai dengan model diagram *Use Case*

V. IMPLEMENTASI

A. Langkah kerja sistem rekomendasi

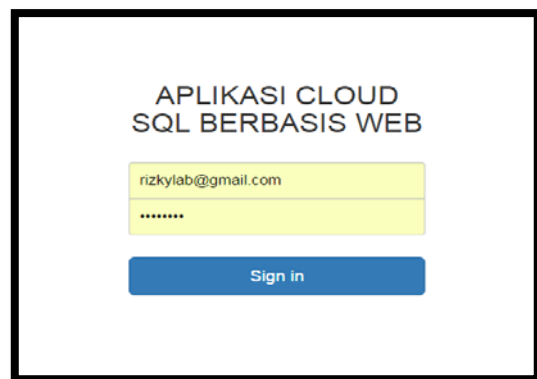
1. Pada saat aplikasi dijalankan proses pertama yang akan dilakukan adalah menambah connection database yang ingin di manage oleh administrator database.
2. Setelah connection dibuat, maka administrator database bisa melakukan manage query.
3. Administrator database bisa melakukan management query dengan melakukan perintah-perintah SQL sesuai kebutuhan. Jika connection berhasil dilakukan, maka mengeksekusi perintah-perintah SQL bisa berhasil dilakukan. Namun, jika koneksi gagal, maka admin harus memperbaiki connection terlebih dahulu.
4. Setelah manage query, admin bisa menyimpan hasil query untuk dimasukkan ke halaman manage query result. Setiap hasil query yang telah dijalankan dapat

disimpan dan data tersebut dapat dibagikan oleh siapapun dalam bentuk API.

B. Implementasi Antarmuka

1. Tampilan halaman *sign in*

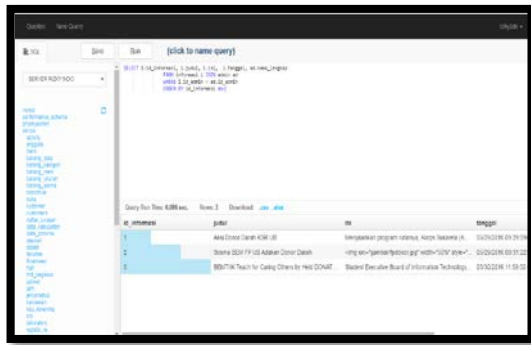
Halaman login merupakan halaman pertama yang akan keluar saat sistem dijalankan. Pada halaman *login* ini terdapat judul dari aplikasi yang telah dibuat ini yaitu “Aplikasi *Cloud SQL* Berbasis Web”. Halaman login ini juga memiliki dua buah *field* yaitu email dan password yang digunakan sebagai autentifikasi ketika user ingin masuk kedalam sistem seperti pada Gambar 5.1



Gambar 5.1 Halaman *Sign in*

2. Tampilan hasil manajemen *query*

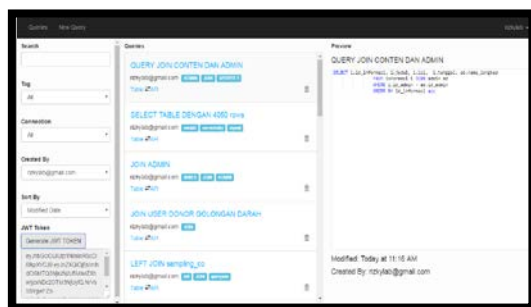
Aplikasi *cloud* ini mengambil studi kasus dengan menggunakan 3 database server, dimana terdapat tiga MySQL server dan tiga database PostgreSQL server. Dengan demikian, didalam aplikasi ini dicantumkan sebuah halaman yang di dalamnya menambahkan mengenai informasi-informasi yang berhubungan dengan *connection* setiap database server yang di tambahkan seperti pada Gambar 5.2



Gambar 5.2 Halaman Hasil Manajemen Query

3. Halaman Query Result

Halaman ini adalah halaman data *query* yang telah di simpan. Setiap *id query* yang disimpan, maka ada fasilitas *preview* perintah apa pada *query* tersebut, serta ada waktu dan pembuat. Pada setiap *query* yang disimpan terdapat pula tombol ke hasil tabel dan tombol untuk membuat *json api*. Pada *api* inilah data dapat di sahning kepada siapapun yang membutuhkan. Namun, proses membagi data tidak bisa langsung diberikan, melainkan untuk mendapat data tersebut diharuskan menggunakan token sebagai validasi agar data itu dapat di peroleh oleh pengguna yang membutuhkan data tersebut seperti pada Gambar 5.3.



Gambar 5.3 Halaman Query Result

4. Halaman JSON API

Pada setiap *query* yang disimpan terdapat pula tombol ke hasil tabel dan tombol untuk membuat *json api*. Pada *api* inilah data dapat di

sahring kepada siapapun yang membutuhkan. Namun, proses membagi data tidak bisa langsung diberikan, melainkan untuk mendapat data tersebut diharuskan menggunakan token sebagai validasi agar data itu dapat di peroleh oleh pengguna yang membutuhkan data tersebut seperti Gambar 5.4



Gambar 5.4 Halaman Json API

VI. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan analisis perancangan sistem, implementasi, dan pengujian sistem, maka dapat disimpulkan hal berikut :

- 1) Aplikasi cloud SQL berbasis web dibangun dengan menggunakan layanan *software as a service* yang dapat memanajemen database MySQL dan PostgreSQL serta dapat berbagi *resource* database ke siapapun yang membutuhkan.

B. Saran

Berdasarkan analisis perancangan sistem, implementasi, dan pengujian sistem, maka untuk pengembangan penelitian selanjutnya penulis menyarankan hal berikut:

- 1) Aplikasi ini masih menerapkan jenis *private cloud*, agar hendaknya di gembangkan kedepan untuk menjadi

public cloud dengan akses level user yang lebih baik.

- 2) Aplikasi ini dapat terus dikembangkan lebih lanjut kedepannya diharapkan untuk dapat mengembangkan setiap *generate* token yang dapat digunakan agar dapat di simpan dalam database.

REFERENSI

- [1] Kadhira, Abdul. 2008. *Belajar Database Menggunakan SQL Server*. Yogyakarta : Penerbit Andi.
- [2] Nugroho, Andi. 2005. *Rasional Rose untuk Pemodelan Berorientasi Objek*, Bandung: Penerbit Informatika.
- [3] Pedro Teixeira, 2013. *Professional Node.js*. Indianapolis, United States of America: John Wiley & Sons, Inc.
- [4] Roger S. Pressman, 2002. *Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku Satu)*, ANDI Yogyakarta.
- [5] Riyadi, 2013, *Rancang Bangun REST Web Service untuk Perbandingan Harga Pengiriman dengan Metode Web Scrapping*, Skripsi, Teknik Informatika AMIKOM, Yogyakarta.
- [6] Sandoval, Jose, 2009, *RESTful Java Web services, Master core REST concepts and create RESTful web services in Java*, Packt Publishing Ltd, Birmingham, UK.
- [7] Sasono, Norman. 2011. *Memahami Cloud Computing (Bagian 2)*. (Online). bagian-2/ semua-halaman, diakses 14 Juli 2016).
- [8] Sugeng, Winarno. 2005. *Jaringan Komputer dengan TCP/IP*, Informatika, Bandung.