

# PERANCANGAN APLIKASI MONITORING AUTHORITATIVE DOMAIN NAME SYSTEM SERVER MENGGUNAKAN FRAMEWORK CODEIGNITER

Kartika Eka Yanindra<sup>\*)</sup>, Aghus Sofwan, and Maman Somantri

Departemen Teknik Elektro, Universitas Diponegoro  
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

<sup>\*)E-mail: eka.yanindra@gmail.com</sup>

## Abstrak

Teknologi jaringan mengalami perkembangan yang cepat. Perkembangan teknologi membuat banyak alat yang mempermudah kerja manusia. Salah satu alat yang digunakan disebut dengan DNS server yang berguna untuk menerjemahkan domain website menjadi alamat IP. DNS yang banyak digunakan pada jaringan adalah DNS recursor dan DNS authoritative. Penggunaan DNS server dalam sebuah jaringan membutuhkan juga aplikasi untuk memonitor kerja dari DNS server. Monitoring DNS server yang saat ini digunakan adalah menampilkan grafik kinerja dari DNS server. Salah satu grafik yang ditampilkan pada aplikasi monitoring adalah grafik CPU load dari server DNS, sehingga dapat diketahui kinerja dari server. Fokus dari penelitian ini adalah perancangan aplikasi monitoring authoritative DNS menggunakan framework CodeIgniter, database menggunakan MariaDB, dan webserver menggunakan Apache. Penggunaan aplikasi monitoring authoritative DNS diharapkan dapat memonitor kerja dari DNS server dan dapat menambahkan konfigurasi untuk kebutuhan pada jaringan yang diimplementasikan.

*Kata Kunci : DNS Server, CodeIgniter, MariaDB, Apache Webserver*

## Abstract

Network technology is experiencing rapid development. The development of technology makes a lot of tools that simplify human work. One of the tools is called DNS server which is useful to translate website domain into IP address. DNS that is widely used on the networks are DNS recursor and DNS authoritative. The use of DNS server in a network requires an application to monitor the work of the DNS server. Monitoring DNS servers currently in use are displaying performance graphs from DNS servers. One of the graphs shown in the monitoring application is the CPU load graph of the DNS server, so it can be known the performance of the server. The focus of this research is the design of the authoritative DNS monitoring application using the CodeIgniter framework, database using MariaDB, and the webserver using Apache. The use of DNS authoritative monitoring applications is expected to monitor the work of the DNS server and can add configuration to the implemented network.

*Keywords : DNS Server, CodeIgniter, MariaDB, Apache Webserver*

## 1. Pendahuluan

Teknologi jaringan mengalami perkembangan yang cepat, dengan banyaknya *tools* yang mempermudah kinerja manusia. Menyuguhkan kemudahan dalam mengakses perangkat ataupun program secara cepat.

Salah satu *tools* yang mempermudah manusia adalah penggunaan DNS *server* pada sebuah jaringan. DNS *server* berguna untuk menerjemahkan domain yang diketikkan pengguna pada browser kedalam alamat IP dari website yang akan dituju. Penggunaan DNS *server* dalam sebuah jaringan mempermudah akses *website* oleh

pengguna dikarenakan pengguna tidak perlu mengingat alamat IP dari *website*, melainkan hanya mengingat kata kunci yang berkaitan dengan laman web tersebut. Penggunaan DNS *server* juga berguna untuk menjaga keamanan jaringan seperti yang disebutkan dalam jurnal penelitian sebelumnya yang menjelaskan tentang penggunaan *cache* DNS dalam mengantisipasi serangan yang disebabkan oleh pihak luar pada jaringan yang digunakan [1].

Penggunaan DNS *server* dalam jaringan memerlukan sebuah *server* tambahan yang berfungsi untuk menyimpan aplikasi dari DNS *server* tersebut. Ketika sebuah jaringan menggunakan DNS *server*, maka diperlukan juga alat untuk memonitor kerja dari DNS *server*. Metode

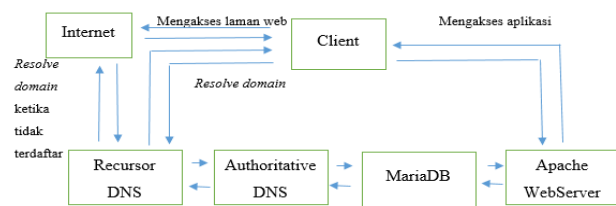
monitoring DNS server yang sudah ada adalah metode aktif dan pasif, seperti yang dijelaskan pada penelitian “*Optimal positioning of Active and Passive Monitoring Device*” [2]. Penelitian lain yang menjelaskan tentang metode monitoring pasif dilakukan oleh Samuel Marchal, Jerome Francois, Cynthia Wagner, Radu State, Alexandre Dulaunoy, Thomas Egnel, dan Olivier Festor yang menjelaskan tentang metode pasif pada jaringan berskala besar [3]. Seperti halnya berbagai macam tipe monitoring sebuah DNS dan perangkat jaringan lain, software yang digunakan untuk memonitor kerja dari sebuah DNS atau perangkat jaringan menggunakan berbagai macam protokol seperti SNMP (*Simple Networking Monitoring Protocol*) yang digunakan pada penelitian oleh Ahmed Kijazi dan Kisangiri Michael [4]. Pada penelitian ini, aplikasi monitoring DNS menggunakan *plugin RRDtools* untuk mengumpulkan informasi yang dibutuhkan untuk memonitor kerja dari DNS server.

Berdasarkan penelitian terdahulu tersebut, maka muncul ide untuk membuat sebuah aplikasi monitoring DNS yang juga berfungsi sebagai aplikasi untuk mengontrol fungsi dari sebuah DNS oleh administrator jaringan. Aplikasi yang dibuat dalam penelitian ini menggunakan software DNS server dari powerDNS dan basis data yang digunakan adalah MariaDB yang merupakan cabang pengembangan dari MySQL. *Webserver* yang digunakan pada penelitian ini menggunakan Apache *webserver*. Perancangan aplikasi monitoring authoritative DNS pada penelitian ini menggunakan *framework CodeIgniter* sehingga aplikasi hasil penelitian ini dapat dibuka baik melalui komputer ataupun *smartphone*. Digunakannya beberapa aplikasi tambahan pada DNS server dapat berakibat dalam penurunan kerja dari DNS server tersebut, sehingga dilakukanlah pengujian pengiriman *query* seperti yang dilakukan pada penelitian terdahulu dengan judul “*DNS Performance and the Effectiveness of Caching*” untuk melihat performa dari sebuah sistem DNS [5].

## 2. Metode

### 2.1. Deskripsi Sistem

Dalam penelitian ini, penulis membuat sebuah sistem DNS dan sistem monitoring authoritative DNS berbasis *web*. Konsep yang dibahas adalah bagaimana *interface web* yang ada pada server dapat diakses melalui komputer ataupun *smartphone* sehingga dapat mengatur zona yang ada pada authoritative server tanpa harus berhubungan secara langsung dengan database yang digunakan yang pada hal ini menggunakan MariaDB. Sistem ini menggunakan 1 buah server yang berisi DNS authoritative, DNS recursor, basis data, dan web server.

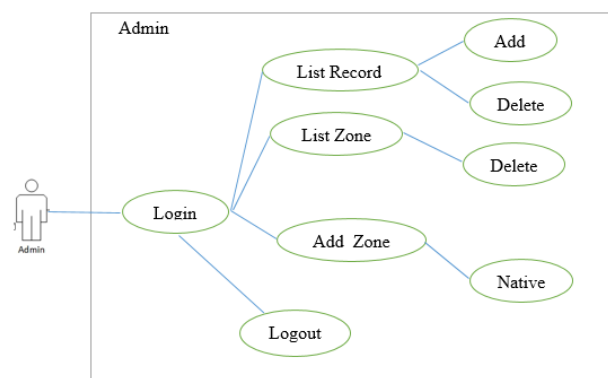


Gambar 1. Desain Sistem

Gambar 1 adalah desain sistem Aplikasi Monitoring Authoritative Domain Name System Server menggunakan Framework CodeIgniter menggunakan 1 buah *recursor DNS*, 1 buah *Authoritative DNS*, MariaDB, dan Apache *Webserver* yang terdapat dalam 1 buah server. Server tersebut terhubung ke dalam jaringan yang dapat diakses oleh pengguna. Cara kerja dari sistem ini adalah ketika pengguna akan mengakses internet, maka komputer pengguna akan menghubungi *recursor DNS* untuk meminta alamat ip dari *website* yang akan dibuka, kemudian *recursor DNS* akan memeriksa basis data dari *authoritative server*. Jika *web* yang akan diakses terdapat dalam *record* maka *authoritative DNS* akan menyediakan alamat *IP* yang dibutuhkan oleh pengguna. Sedangkan jika alamat yang diminta tidak tersedia maka *recursor DNS* akan meminta ke *Internet* untuk alamat yang diminta oleh pengguna. Untuk kerja dari aplikasi monitoring, pengguna akan mengakses *web* menggunakan *browser* dan akan mengakses *webserver* yang ada di server. *Web* monitoring yang diakses oleh pengguna terhubung dengan basis data MariaDB dan basis data tersebut terhubung dengan *Authoritative DNS* menggunakan *MySQL backend*.

## 2.2. Unified Model Language (UML)

### 2.2.1. Diagram Use Case



Gambar 2 Diagram Use Case

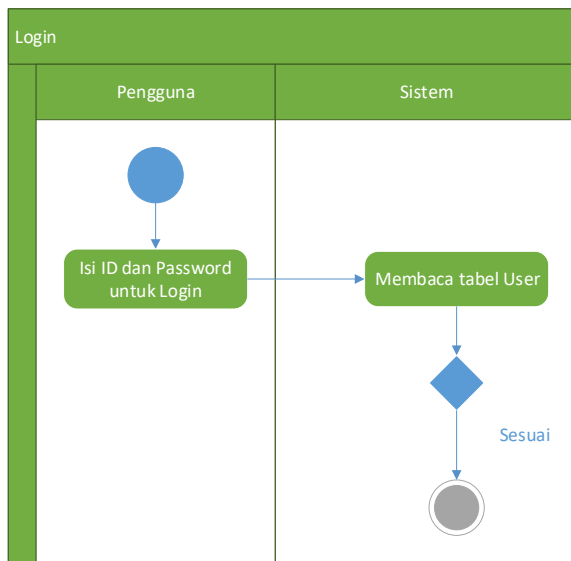
Pada Gambar 2 terlihat bahwa pengguna memiliki fitur *Add* dan *Delete* pada *list record*, menampilkan *list zone* dan menghapus *zone* yang terdaftar, dapat menambahkan *zone* dengan tipe *native* dan *logout*.

Fungsi-fungsi yang dapat digunakan oleh pengguna yaitu:

1. *Add Zone*, berfungsi untuk menambahkan zona baru yang merupakan awal dari sebuah *Authority records* dan merupakan *root domain* untuk *sub level domain* lainnya.
2. *List Zone*, berfungsi untuk menampilkan zona yang telah didaftarkan pada fitur *Add Zone*.
3. *List Record*, berfungsi untuk menampilkan *record* yang telah didaftarkan pada basis data.
4. *Add* dan *Delete* berfungsi untuk menambahkan atau menghapus *sub-level domain* dari sebuah zona yang telah dipilih dari *List Record*.
5. *Logout*, berfungsi untuk menutup halaman akses pengguna.

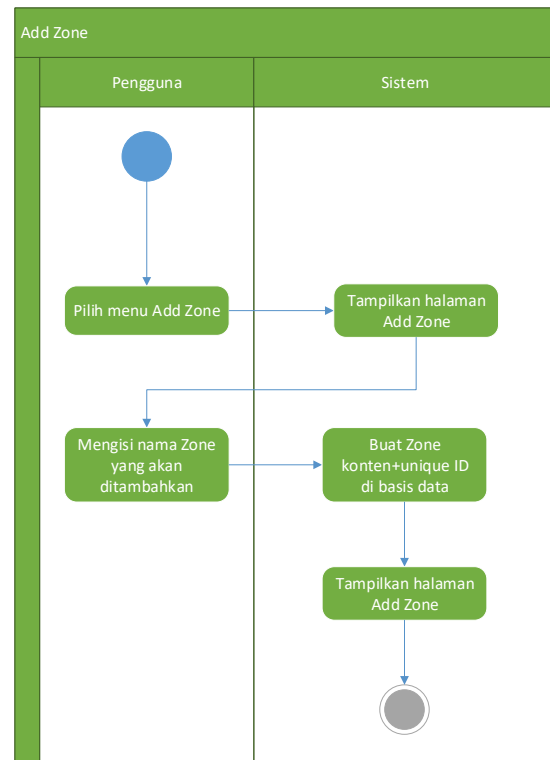
### 2.2.2. Diagram Aktifitas

Diagram aktivitas menggambarkan logika *procedural*, proses bisnis dan aliran kerja dalam sistem yang sedang dirancang. Diagram aktivitas dapat digunakan untuk menggambarkan proses parallel yang mungkin terjadi pada beberapa eksekusi. Diagram aktivitas dapat dibagi menjadi beberapa *swimlane object* untuk menggambarkan objek mana yang bertanggung jawab terhadap aktifitas tertentu. Gambar 3 merupakan diagram aktifitas untuk proses *login*.



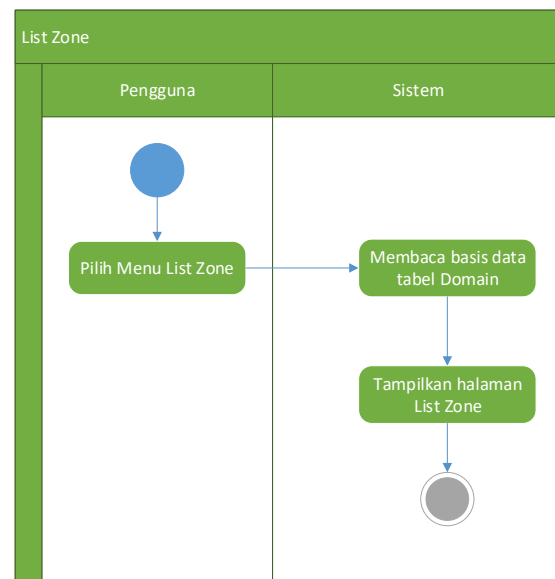
Gambar 3. Diagram aktifitas login

Aktivitas kedua adalah menu *Add Zone*, yang ketika dipilih pengguna diarahkan ke halaman *Add Zone* untuk dapat menambahkan zona baru. Aktivitas ini ditunjukkan pada Gambar 4.



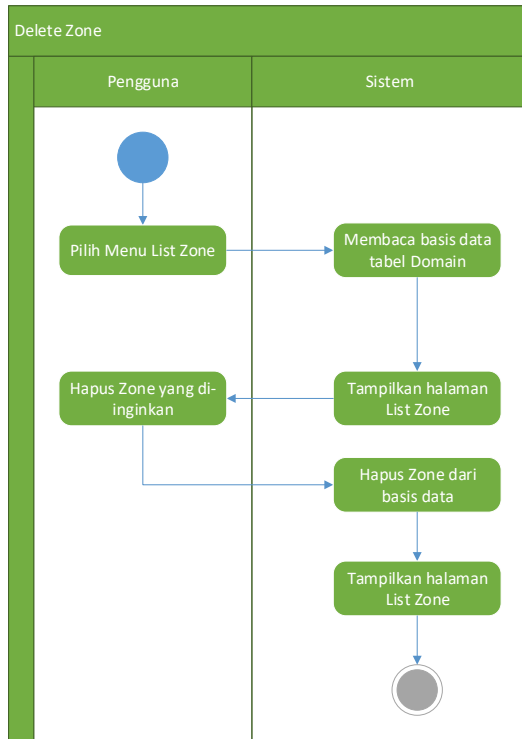
Gambar 4. Diagram aktifitas Add Zone

Aktivitas ketiga adalah menu *List Zone* yang menampilkan zona yang telah terdaftar dalam basis data. Gambar 5 menunjukkan diagram aktivitas dari menu *List Zone*.



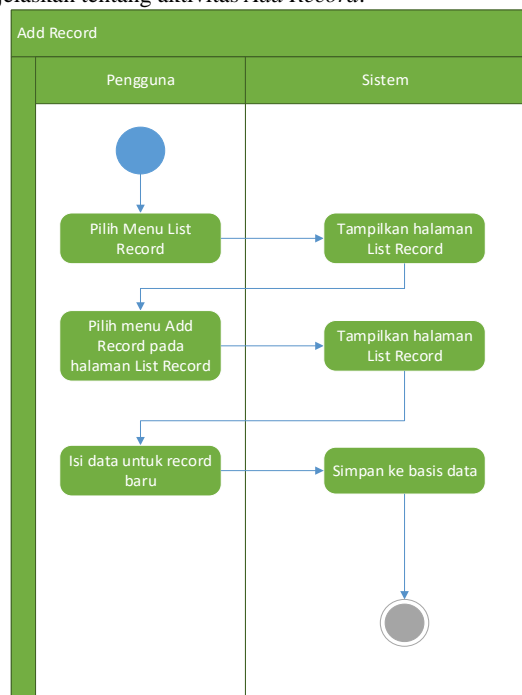
Gambar 5. Diagram aktifitas List Zone

Aktivitas ke empat adalah menu *Delete Zone*, yang ketika dipilih akan menghapus zona yang telah terdaftar. Gambar 6 menjelaskan tentang aktivitas *Delete Zone*.



Gambar 6. Diagram aktivitas Delete Zone

Aktivitas ke lima adalah menu *Add Record* yang akan menambahkan *record* baru kedalam basis data. Gambar 7 menjelaskan tentang aktivitas *Add Record*.



Gambar 7. Diagram aktivitas Add Record

Aktivitas terakhir adalah aktivitas *Delete Record* untuk menghapus *record* yang ada pada *List Record*. Aktivitas ini akan dijelaskan pada Gambar 8.

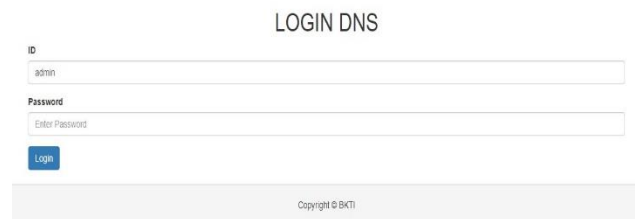


Gambar 8. Diagram aktivitas Delete Record

### 3. Hasil dan Analisa

#### 3.1. Implementasi Aplikasi

Pada saat pengguna membuka aplikasi pertama menggunakan *web browser* maka tampilan pertama yang terbuka adalah tampilan *login*. Berikut adalah tampilan *login* dari aplikasi monitoring *authoritative domain name system server*.



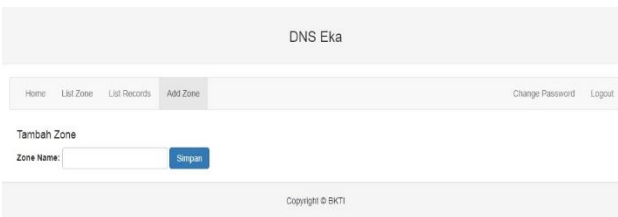
Gambar 9. Tampilan login aplikasi

Setelah pengguna memasukkan *ID* dan *Password* dan berhasil *login* maka yang akan ditampilkan oleh sistem adalah tampilan utama dari aplikasi monitoring *authoritative domain name system server*. Gambar 10 adalah tampilan utama dari aplikasi.



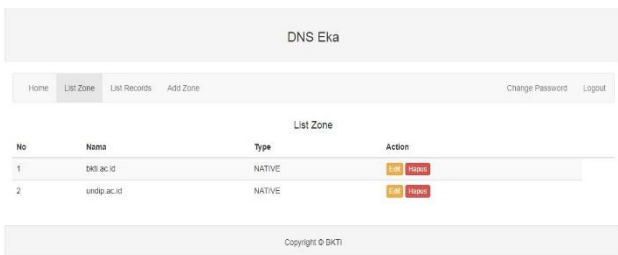
Gambar 10. Tampilan utama aplikasi

Gambar 10 merupakan tampilan utama dari aplikasi. Terdapat 6 buah menu yang terdapat pada baris navigasi, *Home*, *List Zone*, *List Record*, *Add Zone*, *Change password*, *Logout*. Gambar 11 merupakan tampilan aplikasi ketika menampilkan menu *Add Zone*.



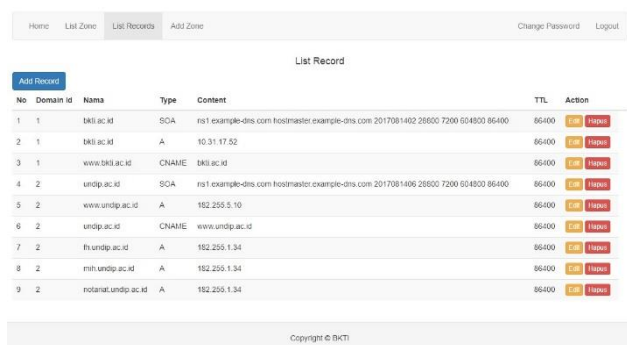
Gambar 11. Tampilan menu Add Zone

Gambar 12 merupakan tampilan aplikasi pada menu *List Zone* yang terdapat pada baris navigasi.



Gambar 12. Tampilan List Zone

Gambar 13 merupakan tampilan aplikasi pada menu *List Record* yang terdapat pada baris navigasi. Pada tampilan *List Record* terdapat menu lain yaitu menu *Add Record* yang digambarkan pada Gambar 14.



Gambar 13. Tampilan menu List Record pada aplikasi



Gambar 14. Tampilan menu Add Record pada aplikasi

### 3.3. Pengujian Sistem

#### 3.3.1. Pengujian Blackbox

Pada pengujian *blackbox* terdapat beberapa metode yang dapat digunakan, salah satunya pengujian alfa. Pengujian alfa bertujuan untuk identifikasi dan menghilangkan masalah sebelum akhirnya sampai ke pengguna. Tabel 1 menunjukkan hasil dari pengujian yang dilakukan.

Tabel 1. Pengujian Blackbox aplikasi

Pengujian	Hasil
Login	Berhasil
Menambah zona baru	Berhasil
Menghapus zona terdaftar	Berhasil
Menambah record baru	Berhasil
Menghapus record terpilih	Berhasil

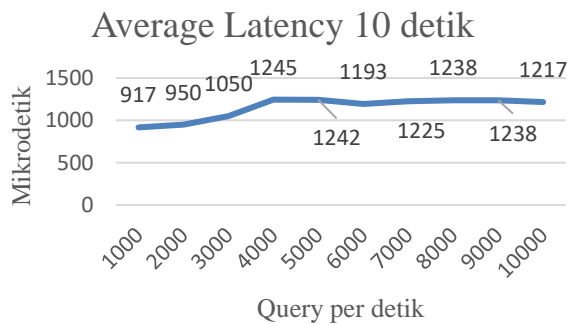
#### 3.3.2. Pengujian Pengiriman Query

Pengujian pengiriman *query* bertujuan untuk melihat performa sistem pada kondisi beban tertentu dan selama periode waktu yang telah ditentukan. Pengujian ini menggunakan aplikasi *dnsperf* yang di install pada komputer lain. Pengujian menggunakan aplikasi *dnsperf* bertujuan untuk mengetahui *average latency* dari sistem DNS. Pengujian yang dilakukan menggunakan *dnsperf* terbagi menjadi 3 bagian 10,100,dan 1000 detik lama pengujian.

Tabel 2. Pengujian pengiriman query selama 10 detik

Query	Average Latency ( $\mu$ S)
1000	917
2000	950
3000	1050
4000	1245
5000	1242
6000	1193
7000	1225
8000	1238
9000	1238
10000	1217

Dengan menggunakan tabel 2 didapatkan grafik hubungan antara *query* dengan *average latency* yang dijelaskan pada Gambar 15.

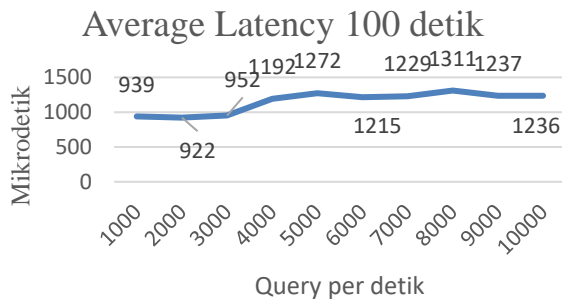


Gambar 15 Average Latency dalam 10 detik

Tabel 3. Pengujian pengiriman query selama 100 detik

Query	Average Latency ( $\mu$ S)
1000	939
2000	922
3000	952
4000	1192
5000	1272
6000	1215
7000	1229
8000	1311
9000	1237
10000	1236

Dengan menggunakan Tabel 3 dapat digambarkan grafik hubungan antara pengiriman *query* dengan *average latency* pada kurun waktu 100 detik yang dijelaskan pada Gambar 16.

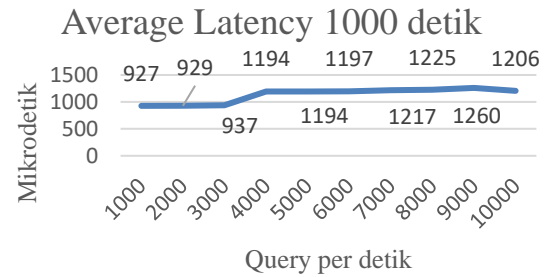


Gambar 16 Average Latency dalam 100 detik

Tabel 4. Pengujian pengiriman query selama 1000 detik

Query	Average Latency ( $\mu$ S)
1000	927
2000	929
3000	937
4000	1194
5000	1194
6000	1197
7000	1217
8000	1225
9000	1260
10000	1206

Dengan menggunakan hasil pengujian pada Tabel 4 bisa digambarkan pada Gambar 17 hubungan antara *query* dan *average latency* selama 1000 detik pengujian.



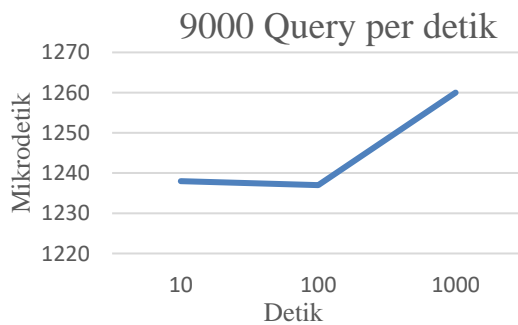
Gambar 17 Average Latency selama 1000 detik

Dari beberapa pengujian mulai 10,100, dan 1000 detik bisa diambil salah satu data patokan untuk melihat hasil pengiriman query yang berjumlah sama namun dalam kurun waktu yang berbeda. Tabel 5 menjelaskan *average latency* 9000 *query* per detik selama 10, 100, dan 1000 detik.

Tabel 5 Average Latency 9000 query per detik

Kurun Waktu	Average Latency ( $\mu$ S)
10	1238
100	1237
1000	126

Dengan menggunakan data pada Tabel 5 dihasilkan grafik hubungan antara *average latency* dan waktu yang digambarkan pada Gambar 18.



Gambar 18. Average Latency 9000 query per detik

#### **4. Kesimpulan**

Kesimpulan yang didapat dari hasil penelitian adalah aplikasi monitoring *authoritative* DNS tidak hanya dapat menambah zona, namun juga dapat menghapus zona yang telah terdaftar. Hasil uji menggunakan *dnstperf* menunjukkan bahwa sistem dapat menjawab seluruh *query* yang dikirimkan pada saat pengujian. Pengujian pengiriman *query* dari beberapa waktu yang berbeda menunjukkan bahwa sistem mengalami average latency yang meningkat lebih dari 200 mikrodetik saat sistem diberi 4000 *query* per detik. Saran untuk aplikasi monitoring ini adalah masih perlu dikembangkan aplikasi dengan menggunakan *API* dari *powerDNS*. Melakukan penelitian lebih lanjut dengan mengaktifkan replikasi pada sistem DNS dan menambahkan kemampuan *master/slave* pada aplikasi monitoring.

#### **Referensi**

- [1] B. Manning, "Intermediate DNS Caching as an Attack Vector," *Cisco*, vol. 12, no. 2, pp. 2-6, 2009.
- [2] C. Claude, F. Eric and R. Herve, "Optimal Positioning of Active and Passive Monitoring Devices," *CoNEXT*, 2005.
- [3] S. Marchal, J. Francois, C. Wagner, R. State, A. Dulaunoy, T. Engel and O. Festor, "A Large Scale Passive DNS Security Monitoring Framework," *IEEE*, p. 1, 2012.
- [4] A. Kijazi and K. Michael, "A Step on Developing Network Monitoring Tools," *ISSN*, vol. 5, no. 8, 2014.
- [5] E. S. H. B. a. R. M. Jaeyeon Jung, "DNS Performance and the Effectiveness of Caching," *IEEE*, vol. 10, no. 5, pp. 589-603, 2002.
- [6] Incognito Software, Understanding DNS (the Domain Name System), Incognito Software Inc., January 2007.
- [7] DNS made easy, DNS for newbies, expert & everyone in between, DNS made easy, 2002.
- [8] F. Razzoli, Mastering MariaDB, Birmingham: Packt publishing, 2014.
- [9] A. G. Matthew Helmke, The Official Ubuntu Book 7th Edition, Prentice hall, 2012.
- [10] I. D. Id, Framework CodeIgniter Sebuah Panduan dan Best Practice, Pekanbaru, 2011.
- [11] Nominum.inc, DNS performance tools manual, redwood city, 2007.
- [12] J. Rama, Sistem Informasi Akuntansi, Jakarta: Salemba, 2008.
- [13] J. W. a. L. Bentley, System Analysis & Design Methods, New York: McGraw-Hill, 2007.
- [14] S. Mneimmeh, Computer Network UDP and TCP, New York: Hunter College, 2008.
- [15] J. Spurlock, Bottstrap: Responsive Web Development, O'Reilly, 2013.