

**OPTIMALISASI KINERJA SERVER MENGGUNAKAN MANAJEMEN DNS
OPTIMIZING SERVER PERFORMANCE USING DNS MANAGEMENT
(STUDI KASUS : IST AKPRIND YOGYAKARTA)**

¹Taufik Hidayat, ²Catur Iswahyudi, ³Suraya

^{1,2,3} Teknik Informatika, Institut Sains & Teknologi AKPRIND, Yogyakarta
32hidayattaufik@gmail.com, catur@akprind.ac.id, Suraya@akprind.ac.id

ABSTRACT

The development of the world of information technology, especially digital on era this very fast that encourages an individual or group to apply it in everyday life. The advantages of information technology today is providing access speed, delivery of information fast, efficient in terms of power and cost and easy use. The Internet is one of the media delivery of information that is very attached to the life of the community. The need for internet in a large agency such as companies, campuses, and other kinds of services to support services and information needs.

This study discusses the optimization of server performance using DNS management, DNS management is done to provide efficiency to the server. The design is to create server virtualization where virtualization is used as a server to improve service performance. Management is done is to create a master-slave DNS which will be between two servers will be a backup when one of the server experience constraints and the konfigurasi of DNS using BIND. Testing is done by using two connections ie public connection and also local connection. The results showed that by doing DNS management and also on the server virtualization can improve service and save physical space.

Keywords : DNS, Virtualization, Master-Slave, Server, Bind.

INTISARI

Perkembangan dunia teknologi informasi terutama di era digital ini yang sangat cepat sehingga mendorong seorang individu maupun kelompok untuk menerapkannya dalam kehidupan sehari-hari. Kelebihan dari teknologi informasi sekarang ini adalah menyediakan kecepatan akses, penyampaian informasi yang cepat, efisien dalam hal waktu tenaga dan biaya serta penggunaannya yang mudah. Internet merupakan salah satu media penyampaian informasi yang sudah sangat melekat pada kehidupan masyarakat. Kebutuhan akan internet pada suatu instansi besar seperti perusahaan, kampus, dan macam-macam lainnya untuk menunjang pelayanan dan kebutuhan informasi.

Penelitian ini membahas tentang optimalisasi kinerja server menggunakan manajemen DNS (Domain Name System), manajemen DNS ini dilakukan untuk memberikan efisiensi terhadap server tersebut. Rancangan yang dilakukan adalah membuat virtualisasi server pada virtualisasi tersebut digunakan sebagai server untuk meningkatkan kinerja layanan. Manajemen yang dilakukan adalah membuat DNS *master-slave* yang nantinya antara 2 server akan menjadi backup apabila salah satu server mengalami kendala, konfigurasi yang digunakan adalah menggunakan BIND. Pengujian dilakukan dengan menggunakan dua koneksi yaitu koneksi *public* dan juga koneksi *local*. Hasil penelitian menunjukkan bahwa dengan melakukan manajemen DNS dan juga pada server dilakukan virtualisasi dapat meningkatkan layanan dan menghemat ruang fisik.

Kata kunci : DNS, Virtualisasi, Master-Slave, Server, Bind.

PENDAHULUAN

Perkembangan dunia teknologi informasi terutama di era digital ini yang sangat cepat sehingga mendorong seorang individu maupun kelompok untuk menerapkannya dalam kehidupan sehari-hari. Kelebihan dari teknologi informasi sekarang ini adalah menyediakan kecepatan akses, penyampaian informasi yang cepat, efisien dalam hal waktu tenaga dan biaya serta penggunaannya yang mudah. Internet merupakan salah satu media penyampaian informasi yang sudah sangat melekat pada kehidupan masyarakat. Kebutuhan akan internet

pada suatu instansi besar seperti perusahaan, kampus, dan macam-macam lainnya untuk menunjang pelayanan dan kebutuhan informasi.

IST AKPRIND Yogyakarta merupakan kampus yang memiliki konfigurasi atau ketersediaan layanan internet yang cukup kompleks dan sudah sesuai kebutuhan. Kapasitas server baik itu *server website*, *database*, dan macam-macam, yang dimiliki oleh pihak IST AKPRIND cukup banyak untuk menunjang kegiatan, dan juga dengan *bandwidth* yang lumayan besar tentu diharapkan akan mempermudah dalam pengaksesan internet ataupun *website-website* yang digunakan dalam menunjang kegiatan. Akan tetapi semua itu berbanding terbalik dengan kenyataannya, sebagai contoh ketika mahasiswa sedang melakukan KRS Online dan mahasiswa mengakses *website* KRS Online aksesnya sangat rendah atau *throughput*-nya kecil penyebab itu dikarenakan jalur masuk untuk mengakses *website* terlalu kecil akibatnya ketika *user* yang sangat banyak mengakses secara bersamaan akan terjadi antrian untuk masuk ke dalam *website*.

Berdasarkan latar belakang tersebut dikarenakan yang bermasalah adalah lambatnya akses ke *serverwebsite*, maka perlu mengoptimalkan kinerja server untuk meningkatkan kualitas layanan dan mengurangi beban kinerjanya yaitu dengan mengoptimalkan pada DNS-nya menggunakan manajemen DNS. Manajemen DNS yang dilakukan nantinya dengan melakukan *forward* DNS yaitu akan terdapat 2 *network server* yang dikonfigurasi atau ditempatkan NS1 di ISP dan NS2 berada pada IST AKPRIND. Diharapkan dalam melakukan optimalisasi server ini dapat meningkatkan kualitas dalam penyediaan informasi bagi mahasiswa dan pelayanan akademik kampus.

Karena banyaknya topic pembahasan yang dapat dibahas dalam penelitian ini maka penulis membatasi penelitian ini dengan

1. Hanya melakukan konfigurasi pada server *local* atau NS2 yang berada pada IST AKPRIND
2. Tidak melakukan sinkronasi dengan web server.
3. Tidak melakukan pengujian keamanan.

TINJAUAN PUSTAKA

(Juliharta, Antar, & Aryani, 2016) melakukan penelitian dengan membuat manajemen domain name server menggunakan standar *national institute of standard and technology (NIST) 800-81r1*. Melakukan perbandingan antara *DNS server* standar dan *DNS* menggunakan NIST standar, menggunakan 3 alat yaitu *nessus*, *openVas*, dan *nmap*. Untuk DNS standar NIST ada enam tahapan yang dilakukan yaitu manajemen versi, manajemen *spoofing*, manajemen zona transfer, manajemen *dynamic update*, manajemen *query*, dan manajemen *access list*. Hasil yang diperoleh telah dibuktikan bahwa DNS dengan standar NIST memiliki performa yang lebih baik. Kekurangan atau hal yang belum dilakukan dalam penelitian ini adalah dalam pengujiaanya hanya melakukan scanning tidak melakukan uji coba serangan secara langsung atau *real time*.

(Kurniawan, Sumaryono, & Suwastono, 2010) melakukan penelitian Pengelolaan dinamik *domain name system* berbasis DJBDNS, Dinamik DNS digunakan untuk menerjemahkan nama *host* menjadi alamat IP dengan data alamat IP yang sering berubah-ubah yang tidak dapat dilayani oleh DNS biasa. Sistem pengelolaan dinamik DNS berbasis *djbdns* dibuat dengan menggunakan bahasa pemrograman PHP dan basisdata *MySQL* sebagai penyimpanan data. Pengelolaan dinamik DNS berbasis *djbdns* sebagai *content* *DNS server* dapat berfungsi sebagai dinamik *content* *DNS server* dengan data RR yang berubah-ubah dalam interval waktu tertentu dengan bantuan VegaDNS. Untuk mengatasi *DNS Propagation* disarankan pengguna layanan dinamik DNS ini menggunakan *proxy* *DNS server* yang juga disediakan dalam layanan ini. Kekurangan atau hal yang belum dilakukan dalam penelitian ini adalah tidak melakukan perbandingan antara BIND dan DJBDNS sebagai pengelola DNS yang dikatakan lebih baik dibandingkan dengan BIND, tidak melakukan pengujian keamanan terhadap DJBDNS.

(Hidayat & Rahmatulloh, 2015) melakukan penelitian dengan membuat Optimasi server SIMAK menggunakan *memcache* dan *mirrorserver* untuk meningkatkan kecepatan akses layanan akademik Universitas Siliwangi. Peningkatan jumlah mahasiswa berimbas pada banyaknya *user system* yang mengakses sistem akademik dan menurunnya kecepatan *response* sistem dalam menangani *request* kondisi ini dapat ditangani dengan optimasi sistem dengan *Memcached* dan *mirror server* *Memcached* adalah suatu script tambahan yang

diletakkan dalam suatu Server Mirror sebagai jembatan (bridge system) antara server utama dengan web service. Sehingga saat penggunaan, setiap kali komunikasi data terjadi, request dari pengguna akan diseleksi apabila permintaan tersebut tersedia di Cached Memory pada Mirror Server maka langsung ditampilkan ke antarmuka pengguna tanpa melalui eksekusi ke database sehingga beban sistem utama dan database berkurang. Hasil pengujian Memcached dan Mirror Server menggunakan Jmeter dengan skenario standar adalah terjadi peningkatan throughput sekitar 490 transaksi per menit lebih besar dibandingkan sebelum implementasi yaitu 171 transaksi per menit kekurangan atau yang belum dilakukan dalam penelitian ini adalah, dalam pengujiannya hanya menggunakan jmeter saja tidak menggunakan beberapa software pengujian lainnya, belum melakukan pengujian keamanan atau *security* dalam penerapan optimalisasinya.

Dari beberapa tinjauan pustaka di atas, dapat disimpulkan kinerja dari pada sebuah server sangat penting guna memberikan *performance* yang lebih baik bagi para penggunanya. Dalam penelitian ini untuk meningkatkan kinerja server yaitu dengan melakukan *forward* DNS yaitu DNS server akan berada di ISP dengan akses bandwidth yang tinggi dan server fisik berada pada IST AKPRIND. Diharapkan dalam penelitian ini dapat bermanfaat untuk kebutuhan akademik di lingkungan IST AKPRIND Yogyakarta. Penelitian ini penting untuk dilakukan adalah untuk membantu IST AKPRIND.

LANDASAN TEORI

Server merupakan hati dari jaringan. Server biasanya merupakan komputer berkecepatan tinggi dengan kapasitas memori (RAM) dan simpanan yang besar, dan dihubungkan dengan kartu jaringan yang cepat (*fast network interface*). Untuk mendukung kinerja yang tinggi, pada server dimungkinkan untuk dipasang beberapa prosesor secara simetrik. Demikian juga peralatan lain seperti *main board*, sumber daya dan RAM dipilih kualitas yang tinggi, sehingga server mampu beroperasi terus-menerus tanpa berhenti untuk melayani permintaan. Sistem operasi jaringan juga dipilih untuk bekerja pada komputer tersebut, bersama perangkat lunak aplikasi dan *file* data yang diperlukan. Server terdiri dari bermacam-macam server seperti DNS Server, Web Server, Mail server dan macam-macam lainnya. (Wagito, 2007)

Virtualisasi adalah sebuah teknik yang saat ini banyak diterapkan untuk memenuhi kebutuhan TI yang semakin tinggi namun diikuti dengan tuntutan untuk mengefisienkan biaya yang digunakan semaksimal mungkin. Virtualisasi adalah teknologi yang telah diterapkan secara luas saat ini dengan dampak peningkatan operasional dan finansial yang positif. Virtualisasi merupakan konsep bahwa akses ke sebuah *hardware* seperti server diatur sehingga beberapa *operating system* dapat berbeaigi sebuah hardware. Tujuan dari virtualisasi adalah kinerja tingkat tinggi, ketersediaan, keandalan, ketangkasan, atau untuk membuat dasar keamanan dan manajemen terpadu. Beberapa perangkat yang dapat divirtualisasi diantaranya server, desktop, storage, application, dan network. (Limantara, 2014)

DNS (*Domain name system*) merupakan sebuah fasilitas dalam suatu sistem operasi untuk memetakan alamat/domain ke alamat IP. Ketika sedang membuka *website*, yang diketikkan adalah merupakan alamat dari *website* tersebut yang sebenarnya adalah merupakan alamat IP. Sebagai contoh yaitu ketika sedang diketik www.google.com maka yang sebenarnya dituju adalah alamat IP dari google, yaitu 173.194.38.146. IP ini dapat diketahui apabila menjalankan ping dari CMD dari komputer dan mengetikkan ping google.com, maka ketika komputer telah terkoneksi dengan internet akan terlihat IP dari google. (Komputer, 2013)

BIND adalah perangkat lunak *open source* yang memungkinkan untuk mempublikasikan informasi sistem nama domain (DNS) di internet, dan untuk menyelesaikan *query* DNS untuk pengguna. Nama BIND adalah singkatan dari "*Berkeley Internet Name Domain*", perangkat lunak ini berasal pada awal tahun 1980an di *University of California* di *Berkeley*. BIND menerapkan protokol DNS yaitu menentukan proses bahwa satu komputer dapat menemukan komputer lain berdasarkan namanya. (Consurtium, 2017)

PowerDNS adalah *name server* berwibawa tingkat tinggi yang canggih, yang kompatibel dengan sejumlah *backends*. PowerDNS dapat menggunakan file konfigurasi BIND, membaca informasi dari MariaDB, MySQL, Oracle, PostgreSQL dan banyak database lainnya. *Backends* dapat dengan mudah ditulis dalam bahasa apapun. (Hartmann, 2014)

Cache DNS Server Jenis server ini dikenal sebagai resolver karena menangani *query* rekursif dan umumnya dapat menangani pekerjaan kasar yang melacak data DNS dari server

lain. Ketika *cacheserver DNS* melacak jawaban dari *query client*, kemudian *server* mengembalikan jawaban ke *client*, dan juga menyimpannya di *canche* untuk jangka waktu yang diizinkan oleh nilai *TTL*. *Canche* kemudian dapat digunakan sebagai sumber permintaan berikutnya untuk mempercepat waktu *round-trip total*. *Canche Server DNS* adalah pilhan yang baik untuk banyak situasi jika tidak ingin bergantung pada *DNS ISP* atau *server DNS* lain yang tersedia untuk umum. (Ellingwood, 2014)

DNS Forwarding hampir sama dengan *canche DNS server* jika dilihat dari sudut pandang *client*, namun mekanisme dan beban kerja sangat berbeda. *Server DNS forwarding* menawarkan keuntungan yang sama untuk menjaga *canche* agar memperbaiki waktu resolusi *DNS* bagi *client*. Akan tetapi tidak ada *query* rekursif, sebaliknya yaitu dengan meneruskan semua permintaan ke *server* penyelesaian di luar dan kemudian menyimpan hasilnya untuk *query* berikutnya. Memungkinkan *server* penerus merespon dari temboloknya, meski tidak mengharuskannya melakukan semua *query* rekursif. Hal ini memungkinkan *server* hanya membuat permintaan tunggal (permintaan *client* yang diteruskan). (Ellingwood, 2014)

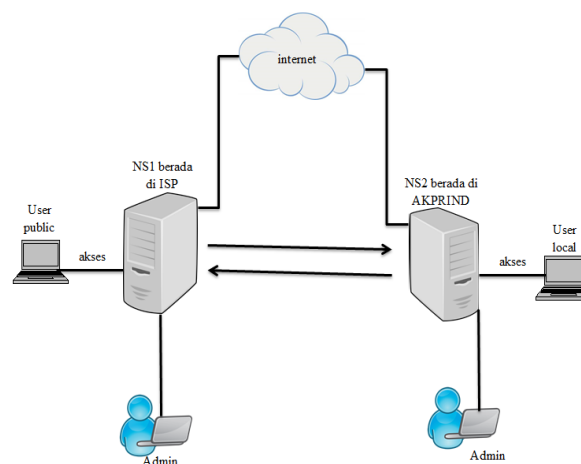
Name server DNS authoritative bertanggung jawab untuk memberikan jawaban atas *name server DNS* rekursif dengan pemetaan IP dari situs web yang dimaksud. Respons *name server authoritative* terhadap *name server* rekursif berisi informasi penting untuk setiap domain, seperti alamat IP yang sesuai dari data *DNS* lain yang diperlukan. (Frost, 2014)

Linux adalah sebuah program *open source* yang gratis dibawah lisensi GNU sistem operasi 32-64 bit, yang merupakan turunan dari UNIX dan dapat dijalankan pada berbagai macam *platform* perangkat keras mulai dari Intel (x86), hingga prosesor RISC. Selain karena gratis, dengan lisensi GNU (*Gnu Not Unix*) pengguna dapat memperoleh program lengkap dengan kode dan sumbernya (*source code*). Pengguna diberikan hak untuk meng-copy sebanyak yang diinginkan, atau bahkan mengubah kode sumbernya. (Sofyan, 2000)

ISP berasal dari kata internet yang berarti hubungan komputer dengan berbagai tipe yang membentuk sistem jaringan yang mencakup seluruh dunia, *service* berarti layanan dan *provider* yang berarti penyedia layanan atau jasa sehingga pengertian ISP adalah penyedia layanan internet. Penyedia jasa internet yakni suatu lembaga atau pengusaha yang menghubungkan komputer pengguna dengan komputer internet. ISP adalah perusahaan yang bergerak dalam jasa pelayanan internet, perusahaan ini menginvestasikan dananya untuk membangun infrastruktur jaringan internet. Jika sebuah instansi ingin terhubung dengan jaringan internet terlebih dahulu harus menghubungkan komputer kesebuah ISP tertentu dengan mematuhi syarat-syarat yang diberikan oleh ISP tersebut mulai dari besarnya biaya yang dibebankan, kecepatan transfer data, dan juga batas waktu untuk mengakses internet. ISP di Indonesia tergabung dalam asosiasi penyelenggara jasa internet Indonesia (APJII). (Santoso, 2012)

PEMBAHASAN

Rancangan Arsitektur Sistem

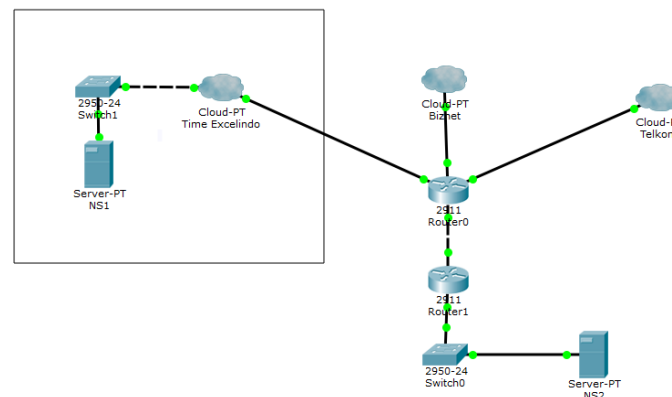


Gambar 1 Rancangan Arsitektur Sistem

Pada gambar 1 dapat dijelaskan bahwa pada rancangan arsitektur ini akan terdapat 2 buah server, 2 buah server tersebut yaitu NS1 (*name server 1*) berada di ISP dan NS2 (*name server 2*) berada pada IST AKPRIND. Apabila seorang user akan mengakses akprind.ac.id dengan menggunakan jaringan *public* atau jaringan internet maka akan mengakses pada NS1, tetapi jika user tersebut mengakses dengan koneksi jaringan *local* yang berada pada IST AKPRIND maka akan mengakses pada NS2, selain itu NS2 berfungsi sebagai backup apabila server NS1 mengalami masalah maka akan di backup oleh NS2.

Topologi konfigurasi server

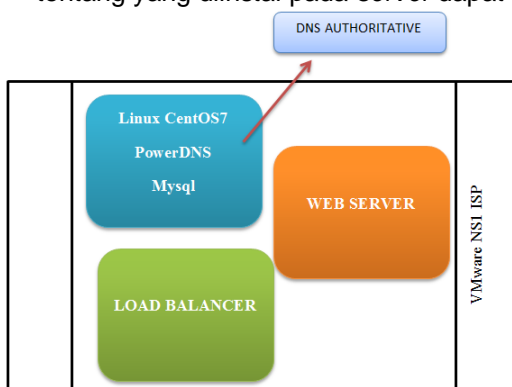
Topologi jaringan yang digunakan agar server dapat terhubung dengan internet secara *public* ataupun secara *local* adalah dengan menggunakan topologi star. Topologi ini sangat fleksibel untuk dilakukan pengembangan, server NS1 terhubung atau berada pada jaringan ISP (*Internet Service Provider*) dan untuk yang NS2 berada pada jaringan atau bertempat di IST AKPRIND. Gambar topologi jaringan dapat dilihat pada gambar 2



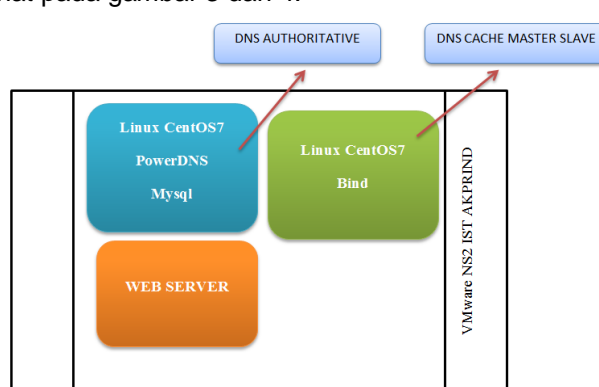
Gambar 2 Topologi konfigurasi server

Rancangan instalasi

Rancangan instalasi *server* ini adalah membahas apa saja yang diinstal pada *server* yang dikonfigurasi. Instalasi *software* adalah melakukan pemasangan VMware atau melakukan virtualisasi. Dalam mesin *virtual* tersebut akan dilakukan pemasangan sistem operasi baru, pada sistem operasi tersebut dilakukan konfigurasi DNS. Rancangan atau gambar sederhana tentang yang diinstal pada *server* dapat dilihat pada gambar 3 dan 4.



Gambar 3 Instalasi Software NS 1

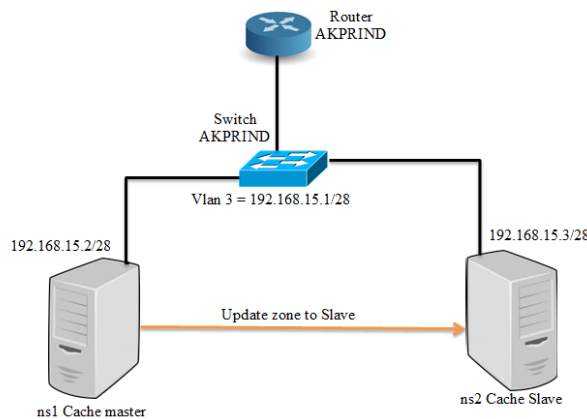


Gambar 4 Instalasi software NS2

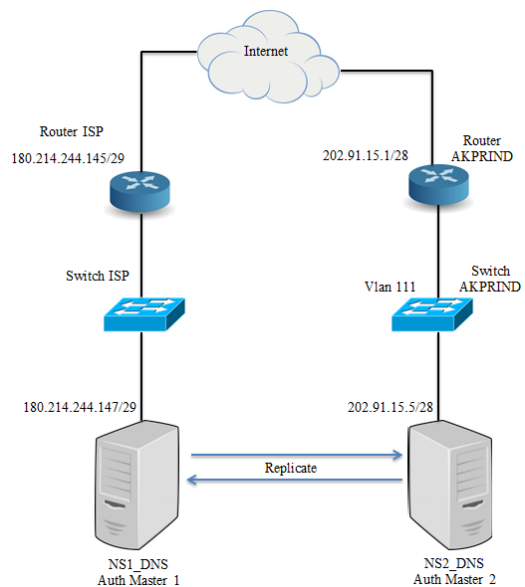
Instalasi DNS

Pada konfigurasi atau instalasi DNS ini akan dikonfigurasi menjadi 2 jenis yaitu DNS *resolver (cache)*, dan juga DNS *authoritative (authority)*. DNS *resolver (cache)* ini akan memiliki fungsi yaitu mempercepat akses ke *website local* IST AKPRIND, bahwa DNS cache ini hanya

digunakan untuk jaringan *local* IST AKPRIND saja. Sedangkan untuk konfigurasi DNS *authoritative* dimana NS1 dan NS2 akan selalu tersinkronasi, perubahan yang dilakukan pada salah satu *server* maka akan secara otomatis dikirim pada *server* yang lainnya. Semisal melakukan perubahan pada NS2 maka akan dikirim secara otomatis pada NS1, begitu pula dengan sebaliknya. Aplikasi yang digunakan sebagai konfigurasi adalah BIND dan juga PowerDNS, rancangan konfigurasinya BIND akan digunakan sebagai DNS *resolver* (*cache*) dan untuk konfigurasinya berupa *Master-Slave* sedangkan untuk PowerDNS digunakan sebagai *authoritative* dan untuk konfigurasinya berupa *Master-Master*. instalasi dns dapat dilihat pada gambar 5 dan 6



Gambar 5 DNS Cache



Gambar 6 DNS Authoritative

Konfigurasi DNS

Konfigurasi DNS *Cache master*

```
options {
    listen-on port 53 { 127.0.0.1; 192.168.15.2; };
    listen-on-v6 port 53 { ::1; };
    directory "/var/named";
    dump-file "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_mem_stats.txt";
    allow-query { localhost; 192.168.15.0/28; };
    allow-transfer { 192.168.15.3; };
};

zone "akprind.ac.id" IN {
    type master;
    file "/var/named/forward.akp";
    allow-update { none; };
};

zone "15.168.192. in-addr.arpa" IN {
    type master;
    file "/var/named/reverse.akp";
    allow-update { none; };
};
```

Gambar 7 Konfigurasi named.conf

```

GNU nano 2.3.1      File: /var/named/forward.akp
$TTL 86400
@           IN      SOA    master.akprind.ac.id. root.akprind.ac.id. (
                                2011071001      ;Serial
                                3600             ;Refresh
                                1800             ;Retry
                                60400            ;Expire
                                86400            ;Minimum TTL
)
@           IN      NS     master.akprind.ac.id.
@           IN      NS     slave.akprind.ac.id.
@           IN      A      192.168.15.2
@           IN      A      192.168.15.3
master      IN      A      192.168.15.2
slave      IN      A      192.168.15.3

```

Gambar 8 Konfigurasi forward.akp

```

GNU nano 2.3.1      File: /var/named/reverse.akp
$TTL 86400
@           IN      SOA    master.akprind.ac.id. root.akprind.ac.id. (
                                2011071001      ;Serial
                                3600             ;Refresh
                                1800             ;Retry
                                60400            ;Expire
                                86400            ;Minimum TTL
)
@           IN      NS     master.akprind.ac.id.
@           IN      NS     slave.akprind.ac.id.
@           IN      PTR    akprind.ac.id.
@           IN      A      192.168.15.2
@           IN      A      192.168.15.3
master      IN      A      192.168.15.2
slave      IN      A      192.168.15.3
2           IN      PTR    master.akprind.ac.id.
3           IN      PTR    slave.akprind.ac.id.

```

Gambar 9 Konfigurasi file reverse

Konfigurasi DNS Slave

```

options {
    listen-on port 53 { 127.0.0.1; 192.168.15.3; };
    listen-on-v6 port 53 { ::1; };
    directory "/var/named";
    dump-file "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_mem_stats.txt";
    allow-query { localhost; 192.168.15.0/28; };
    allow-transfer { 192.168.15.3; };
}

zone "akprind.ac.id" IN {
    type slave;
    file "/var/named/slaves/forward.slaveakp";
    masters { 192.168.15.2; };
};
zone "15.168.192.in-addr.arpa" IN {
    type slave;
    file "/var/named/slaves/reverse.slaveakp";
    masters { 192.168.15.2; };
};

```

Gambar 10 Konfiigurasi Named.conf Pada Slave

Pada DNS Slave ini karena tidak perlu melakukan konfigurasi pada file *forward* dan juga *reverse* apabila konfigurasi dari kedua server sudah benar maka akan secara otomatis akan tersinkronasi, untuk mengetahuinya dapat dilihat pada gambar 11.

```
[root@slave ~]# ls /var/named/slaves/
forward.slaveakp reverse.slaveakp
[root@slave ~]#
```

Gambar 11 Cek Konfigurasi Forward dan Reverse Pada Slave

Pengujian

Pengujian ini dilakukan dengan menggunakan akses internet, pengujian yang dilakukan adalah pengujian untuk mengetahui apakah dns sudah dikenali secara public dan pengujian dengan menggunakan jaringan *local* dan jaringan *public* pengujian yang pertama adalah menggunakan <https://dnschecker.org/> guna mengetahui DNS dikenali secara *public* dan juga dikenali di mana saja, pengujian dapat dilihat pada gambar 11 sampai dengan gambar 13.

DNS Propagation Check		
Canoga Park, CA, United States (Sprint)	180.214.244.148	✓
Montreal, Canada (iWeb Technologies)	180.214.244.148	✓
Broomfield CO, United States (Verizon)	180.214.244.148	✓
Mountain View CA, United States (Google)	180.214.244.148	✓
Holtsville NY, United States (OpenDNS)	180.214.244.148	✓
Holtsville NY, United States (OpenDNS)	180.214.244.148	✓
Yekaterinburg, Russian Federation (SkyDNS)	180.214.244.148	✓
Johannesburg, South Africa (HETZNER)	180.214.244.148	✓
Zwolle, Netherlands (Ziggo)	180.214.244.148	✓
Fort-de-france, France (Orange)	180.214.244.148	✓

Gambar 11 Akprind.ac.id

DNS Propagation Check		
Canoga Park, CA, United States (Sprint)	180.214.244.147	✓
Montreal, Canada (iWeb Technologies)	180.214.244.147	✓
Broomfield CO, United States (Verizon)	180.214.244.147	✓
Mountain View CA, United States (Google)	180.214.244.147	✓
Holtsville NY, United States (OpenDNS)	180.214.244.147	✓
Holtsville NY, United States (OpenDNS)	180.214.244.147	✓
Yekaterinburg, Russian Federation (SkyDNS)	180.214.244.147	✓
Johannesburg, South Africa (HETZNER)	180.214.244.147	✓
Zwolle, Netherlands (Ziggo)	180.214.244.147	✓
Fort-de-france, France (Orange)	180.214.244.147	✓
Palma, Spain (Ono)	180.214.244.147	✓
Horn, Switzerland (Swisscom)	180.214.244.147	✓
Melbourne, Australia (Pacific Internet)	180.214.244.147	✓
Gloucester, United Kingdom (Fasthosts Internet)	180.214.244.147	✓
Midtjylland, Denmark (NM NET APS)	180.214.244.147	✓

Gambar 12 NS1.Akprind.ac.id

DNS Propagation Check		
Canoga Park, CA, United States (Sprint)	202.91.15.5	✓
Montreal, Canada (iWeb Technologies)	202.91.15.5	✓
Broomfield CO, United States (Verizon)	202.91.15.5	✓
Mountain View CA, United States (Google)	202.91.15.5	✓
Holtsville NY, United States (OpenDNS)	202.91.15.5	✓
Holtsville NY, United States (OpenDNS)	202.91.15.5	✓
Yekaterinburg, Russian Federation (SkyDNS)	202.91.15.5	✓
Johannesburg, South Africa (HETZNER)	202.91.15.5	✓
Zwolle, Netherlands (Ziggo)	202.91.15.5	✓

Gambar 13 NS2.Akprind.ac.id

Pengujian selanjutnya adalah melakukan pengujian untuk mengetahui kecepatan dari DNS pengujian ini dilakukan dengan menggunakan DNS ultratools ketika diakses oleh user hasil dari pengujian dapat dilihat pada tabel 1

Tabel1 Pengujian dengan ultratools

Nama Pengujian	A	TXT	SOA	SRV	SPF	CNAME	MX	AAAA
Pengujian dengan jaringan <i>public</i>	298	292	289	297	278	295	276	282
	284	280	270	271	281	281	278	288
	285	282	286	281	286	269	282	281
	277	276	267	285	275	282	276	275
	282	289	273	268	280	286	277	278
Pengujian dengan jaringan Local	274	279	286	274	283	286	283	283
	282	280	284	281	270	287	282	286
	285	283	268	270	272	277	293	286
	281	281	275	275	277	277	291	276
	277	272	276	288	275	287	291	281
Rata-Rata	282,5	281,4	277,4	279	277,7	282,7	282,9	281,6

Pada tabel 1 dapat dijelaskan bahwa hasil yang didapatkan atau waktu rata-rata ketika DNS diakses, waktu yang di butuhkan tidak lebih dari 300 ms (mili second) atau sekitar 0,3 detik ketika diakses dan memberikan jawaban kepada *user*. Pengujian berikutnya adalah pengujian dengan menggunakan jaringan *local* dan jaringan *public* apabila terkoneksi dengan jaringan local kemudian mengakses akprind.ac.id maka akan mengakses pada server *local* dengan IP 202.91.15.13, dan apabila terkoneksi dengan jaringan *public* maka akan mengakses pada server public yaitu dengan IP 180.214.244.148 gambar pengujian dapat dilihat pada gambar 14 dan 15

```

C:\WINDOWS\system32\cmd.exe
C:\Users\hidayat_ratnasari>nslookup akprind.ac.id
Server: Unknown
Address: 192.168.15.2

Non-authoritative answer:
Name:   akprind.ac.id
Address: 202.91.15.13

C:\Users\hidayat_ratnasari>ping akprind.ac.id

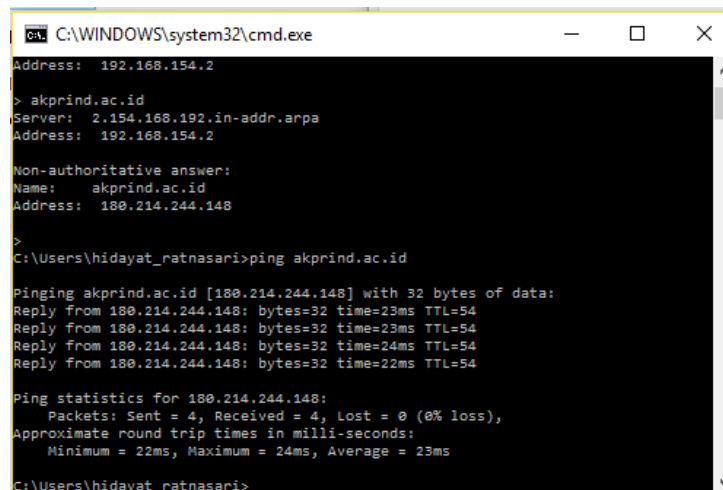
Pinging akprind.ac.id [202.91.15.13] with 32 bytes of data:
Reply from 202.91.15.13: bytes=32 time=38ms TTL=62
Reply from 202.91.15.13: bytes=32 time=49ms TTL=62
Reply from 202.91.15.13: bytes=32 time=43ms TTL=62
Reply from 202.91.15.13: bytes=32 time=62ms TTL=62

Ping statistics for 202.91.15.13:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 38ms, Maximum = 62ms, Average = 48ms

C:\Users\hidayat_ratnasari>

```

Gambar 14 pengujian dengan menggunakan jaringan local IST AKPRIND



```

C:\WINDOWS\system32\cmd.exe
Address: 192.168.154.2
> akprind.ac.id
Server: 2.154.168.192.in-addr.arpa
Address: 192.168.154.2

Non-authoritative answer:
Name: akprind.ac.id
Address: 180.214.244.148
>
C:\Users\hidayat_ratnasari>ping akprind.ac.id

Pinging akprind.ac.id [180.214.244.148] with 32 bytes of data:
Reply from 180.214.244.148: bytes=32 time=23ms TTL=54
Reply from 180.214.244.148: bytes=32 time=23ms TTL=54
Reply from 180.214.244.148: bytes=32 time=24ms TTL=54
Reply from 180.214.244.148: bytes=32 time=22ms TTL=54

Ping statistics for 180.214.244.148:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 22ms, Maximum = 24ms, Average = 23ms
C:\Users\hidayat_ratnasari>

```

Gambar 15 pengujian dengan menggunakan jaringan public

Kesimpulan

Berdasarkan dari hasil perancangan dan ujicoba dalam melakukan optimalisasi server menggunakan DNS manajemen dapat diambil kesimpulan bahwa:

1. Dalam melakukan peningkatan server tidak harus hanya dengan meningkatkan bandwidth ataupun kapasitas internet melainkan dalam pengelolaan servernya, hal ini dapat dilihat dari konfigurasi servernya.
2. Infrastruktur suatu server tidak harus membutuhkan server fisik yang banyak dapat hanya menggunakan beberapa server fisik saja. Hal ini dapat dilihat instalasi server yang menggunakan teknologi virtualisasi yang dapat mengurangi biaya dan efisiensi tempat.
3. Melakukan manajemen DNS dapat memberikan peningkatan performance dan mengurangi beban kinerja dari server hal ini dapat dilihat pada saat user *public* mengakses akprind.ac.id maka akan mengakses pada server NS1 yang berada pada ISP dan pada saat user *local* akan mengakses akprind.ac.id, maka akan mengakses NS2 yang berada pada jaringan IST AKPRIND. Tentunya user yang berada pada jaringan *local* saat mengakses akan menjadi lebih cepat karena dalam satu lingkup jaringan.
4. Kecepatan dari DNS ketika diakses oleh *user* cukup baik hal ini dapat dilihat pada pengujian DNS speed menggunakan ultratool, rata-rata kecepatan ketika diakses kemudian memberikan jawaban kepada user tidak lebih dari 300ms (mili second) atau sekitar 0,3 detik waktu yang diperlukan.
5. Melakukan peningkatan pada kinerja server sangatlah penting terutama dalam *backup* suatu server. Hal ini dapat dilihat dengan melakukan DNS master-slave guna untuk melakukan backup apabila salah satu server mengalami masalah dapat di tangani oleh server lainnya.

Saran

Dalam melakukan penelitian ini yaitu optimalisasi kinerja server dengan menggunakan manajemen DNS yang berstudi kasus di IST AKPRIND, masih banyak kekurangan dalam memberikan laporan. Berikut saran yang diberikan.

1. Penelitian selanjutnya dapat melakukan pengujian keamanan guna mengetahui tingkat dari kewanitaan suatu server atau manajemen DNS-nya.
2. Penelitian selanjutnya dapat melakukan sinkronasi antara DNS dengan *web server* atau *database server*.
3. Penelitian selanjutnya adalah melakukan konfigurasi pada server yang berada pada ISP guna mengetahui bagaimana konfigurasi.
4. Penelitian selanjutnya melakukan pengujian *traffic* terhadap server pada saat digunakan

DAFTAR PUSTAKA

- Consurtium, I. S. (2017, Juli 31). *BIND Versatile, Classic, Complete Name Server Software*. Retrieved Oktober 31, 2017, from ISC (Internet Systems Consurtium): <https://www.isc.org/downloads/bind/>
- Ellingwood, J. (2014, Juli 1). *How to Configure Bind as a Caching or Forwarding DNS Server on Ubuntu 14.04*. Retrieved Oktober 31, 2017, from Digitalocean: <https://www.digitalocean.com/community/tutorials/how-to-configure-bind-as-a-caching-or-forwarding-dns-server-on-ubuntu-14-04>
- Frost, C. (2014, July 16). *difference authoritative recursive dns nameservers*. Retrieved November 14, 2017, from Umbrella Cisco: <https://umbrella.cisco.com/blog/2014/07/16/difference-authoritative-recursive-dns-nameservers/>
- Hartmann, W. (2014, Desember 2). *Cara Menginstal dan Mengkonfigurasi PowerDNS dengan MariaDB di Ubuntu 14.04*. Retrieved Oktober 31, 2017, from DigitalOcean: <https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-powerdns-with-a-mariadb-backend-on-ubuntu-14-04>
- Hidayat, E. W., & Rahmatulloh, A. (2015, April). Optimasi Server SIMAK Menggunakan Memcached dan Mirror Server Untuk Meningkatkan Kecepatan Akses Layanan Akademik Universitas Siliwangi. *Jurnal Ilmu Komputer dan Sains Terapan*, 5, 69-78.
- Juliharta, I. G., Antar, I. G., & Aryani, M. H. (2016, Februari). Manajemen Domain Name Server Menggunakan Standar National Institute of Standard and Technology (NIST) 800-81r1. *Seminar Nasional Teknologi Informasi dan Multimedia 2016*.
- Komputer, W. (2013). *Konsep dan Implementasi Jaringan Menggunakan Windows Server 2012*. Yogyakarta: Andi.
- Kurniawan, A., Sumaryono, S., & Suwastono, A. (2010, September). Pengelolaan Dinamik Domain Name System Berbasis DJBDNS. *Jurnal Penelitian Teknik Elektro*, 3.
- Limantara, N. (2014, Oktober 11). *Konsep Dasar Virtualisasi*. Retrieved November 13, 2017, from BINUS University: <https://sis.binus.ac.id/2014/10/11/konsep-dasar-virtualisasi/>
- Wagito. (2007). *Jaringan Komputer Teori dan Implementasi Berbasis Linux*. Yogyakarta: Gava Media.