PERANCANGAN DAN IMPLEMENTASI SISTEM MONITORING JARINGAN MENGGUNAKAN ICINGA DAN POWERDNS DI PT. PACIFIC TELEMATIKA INDONESIA

Selamet Puji Ramadhan

Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
E-mail : selamet.ramadhan@gmail.com

ABSTRAK

PT. Pacific Telematika Indonesia merupakan perusahaan yang bergerak di bidang Teknologi Informasi, salah satu layanan yang diberikan adalah *Internet Service Provider* (ISP) selain layanan tersebut PT. Pacific Telematika Indonesia memiliki layanan *colocation* yaitu menempatkan komputer server pelanggan di rak PT. Pacific Telematika Indonesia yang digunakan oleh, baik itu instansi pemerintahan, swasta atau perorangan. Dalam menjaga kondisi jaringan ditemukan beberapa masalah, yaitu sulitnya memantau kondisi jaringan terhadap perubahan-perubahan yang terjadi seperti perubahan status jaringan dari hidup ke mati dan sebaliknya. Kondisi tersebut tidak terawasi sepenuhnya dikarenakan keterbatasan Technical Network & System yang harus mengawasi dengan cara mengecek kondisi server satu per satu. Selain itu, penanggulangan terhadap server jika mengalami gangguan, masih dilakukan secara manual.

Berdasarkan hal tersebut dibutuhkan sebuah sistem monitoring jaringan yang dapat digunakan untuk mengatasi masalah tersebut. Dengan tujuan untuk memudahkan Technical Network & System dalam memonitoring jaringan dan dapat menanggulangi permasalahan terhadap server jika mengalami gangguan, dengan cara pengalihan jalur DNS terhadap server alternatif yang disebut *DNS failover*. Sistem yang diimplementasikan menggunakan *Application Programming Interface* (API) sebagai jembatan untuk komunikasi antara Sistem Monitoring jaringan dengan Sistem Domain Name System.

Berdasarkan hasil pengujian yang telah dilakukan dapat disimpulkan bahwa Sistem Monitoring Jaringan yang diimplementasikan membuat penyampaian informasi status perubahan pada objek yang di monitoring menjadi lebih cepat diketahui oleh Technical Network & System, sehingga penanganan masalah pada jaringan menjadi lebih cepat, dan dapat meminimalkan waktu *downtime*. Serta *DNS Failover* yang diimplementasikan dapat menanggulangi permasalahan yang terjadi. jika terjadi masalah pada server.

Kata kunci : Sistem Monitoring Jaringan, *DNS Failover*, API.

1. PENDAHULUAN

Pada bagian ini akan di paparkan mengenai latar belakang masalah, maksud dan tujuan.

1.1 Latar Belakang

PT. Pacific Telematika Indonesia merupakan perusahaan yang bergerak di bidang Teknologi Informasi, salah satu layanan yang diberikan adalah *Internet Service Provider* (ISP) selain layanan tersebut PT. Pacific Telematika Indonesia memiliki layanan *colocation* yaitu menempatkan komputer server pelanggan di rak PT. Pacific Telematika Indonesia yang digunakan oleh lebih dari 30 pengguna, baik itu instansi pemerintahan, swasta atau perorangan. Dalam memberikan pelayanan terhadap pelanggan, perusahaan terus berupaya meningkatkan jaringan dan sistem yang ada, hal tersebut untuk memberikan layanan terbaik kepada pelanggan. Dalam menjaga kondisi jaringan ditemukan beberapa masalah, yaitu sulitnya memantau kondisi jaringan terhadap perubahan-perubahan yang terjadi seperti perubahan status jaringan dari hidup ke mati dan sebaliknya, kondisi yang mengakibatkan terjadinya jaringan mati antara lain kerusakan yang terjadi pada *software* maupun *hardware* contohnya pada *software* mengalami *bug* yang tidak diketahui dari mana penyebabnya, dan pada *hardware* misalnya *ethernet* mati dan sering terjadinya tegangan listrik yang tidak stabil, seperti terjadi pada tanggal 28 November 2016 pukul 20:00 WIB. tentunya kondisi tersebut berpengaruh terhadap pelayanan *service* yang diberikan terhadap pelanggan. kondisi tersebut tidak terawasi sepenuhnya dikarenakan keterbatasan Technical Network & System yang harus mengawasi dengan cara mengecek kondisi server satu per satu. hal ini mengakibatkan penanganan dari suatu kinerja server tidak optimal. Salah satu sistem yang dapat digunakan agar terjaganya kondisi jaringan adalah menggunakan Sistem Monitoring Jaringan.

Berdasarkan wawancara yang telah dilakukan terhadap Technical Network & System di PT. Pacific Telematika Indonesia, maka dibutuhkan

sebuah sistem yang dapat memantau kondisi jaringan di satu *zone* server, karena selama ini untuk mengetahui kondisi jaringan hanya dengan melakukan pemeriksaan terhadap *software* maupun *hardware* yang ditinjau secara manual. Di samping itu perlu adanya mekanisme penanggulangan terhadap server jika mengalami gangguan, dengan pengalihan jalur DNS terhadap server alternatif yang disebut *DNS failover*. Karena ketika terjadi server mati diperlukan konfigurasi pengalihan terhadap server yang mengalami gangguan untuk dialihkan terhadap server alternatif namun konfigurasi tersebut masih dilakukan secara manual. Hal ini dirasa kurang efektif karena tidak menentunya penanganan terhadap pengalihan server mengingat perubahan DNS memerlukan waktu yang dikenal sebagai Time To Live (TTL), yang mengakibatkan *downtime* dalam pemulihan server tergantung dengan waktu yang ditetapkan. TTL tersebut memiliki rentang waktu yang variatif, contoh acuan waktu untuk *refresh* DNS memerlukan waktu 86400 detik atau 24 jam [1].

Salah satu *tools* Sistem Monitoring Jaringan yaitu ICINGA, merupakan sistem monitoring jaringan bersifat open *source* sehingga dapat di konfigurasi sesuai dengan kebutuhan, kegunaannya antara lain untuk memantau kondisi jaringan di lingkungan yang kompleks di beberapa lokasi, ICINGA juga menghasilkan data kinerja dari hasil monitoring untuk pelaporan. ICINGA mampu memonitor segala sesuatu yang terhubung dengan jaringan yang akan di tangani, selain itu dapat mengirimkan notifikasi jika terdapat perubahan status dan ICINGA memungkinkan *multithreaded*. *DNS failover* adalah suatu mekanisme perubahan DNS dari suatu server ke server lainnya, salah satu *tools* DNS yaitu PowerDNS, PowerDNS adalah nameserver autoritatif advanced dengan performansi tinggi. dapat membaca informasi dari MySQL, Oracle, PostgreSQL dan database lainnya. PowerDNS di gunakan untuk *record* DNS untuk menunjang mekanisme *DNS Failover*.

Berdasarkan hasil pengujian yang telah dilakukan dapat disimpulkan bahwa Sistem Monitoring Jaringan yang diimplementasikan dapat digunakan oleh pihak perusahaan untuk memonitoring server dan mempermudah Technical Network & System, Serta DNS Failover yang diimplementasikan dapat menanggulangi permasalahan yang terjadi. jika terjadi masalah pada server.

1.2 Maksud dan Tujuan

Maka maksud dari penelitian ini adalah merancang dan mengimplementasi sebuah sistem monitoring jaringan menggunakan ICINGA sebagai server Sistem Monitoring Jaringan dan PowerDNS sebagai DNS Server.

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah untuk mempermudah Technical Network & System dalam memonitor kondisi jaringan yang nantinya dapat menanggulangi permasalahan yang akan terjadi.

1.3 Manajemen Jaringan

Manajemen jaringan (Network Management) terdiri dari serangkaian fungsi untuk mengontrol, rencana, mengalokasikan, menyebarkan, mengkoordinasikan dan memonitor sumber daya jaringan. Manajemen jaringan digunakan untuk menjadi sebuah pertimbangan di banyak arsitektur jaringan. Sebagai contoh, sebagian besar jaringan arsitektur dan desain dikembangkan tanpa berpikir tentang pengguna yang berbahaya, yang umumnya benar sampai beberapa tahun yang lalu. Mempertimbangkan perubahan yang baru-baru ini telah dibuat dalam keamanan SNMP. Hari ini, dan di masa depan, Jaringan adalah sumber daya yang integritas harus diukur dan diverifikasi. Arsitektur manajemen jaringan, seperti dengan arsitektur komponen lain, dimulai dengan persyaratan dan analisis aliran [5]. Daerah yang harus ditangani selama proses analisis meliputi:

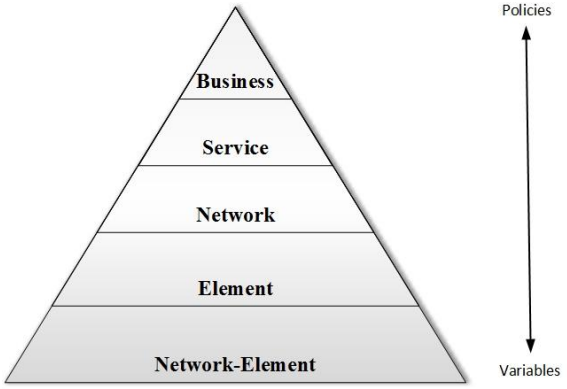
a. Memilih protokol manajemen jaringan.
b. Pelaksana manajemen aset tingkat tinggi sebagai bagian dari jaringan manajemen arsitektur.
c. Konfigurasi ulang jaringan sering untuk memenuhi persyaratan berubah.
d. Pemantauan seluruh sistem dari satu lokasi atau perangkat.
e. Pengujian penyedia layanan sesuai dengan perjanjian tingkat layanan *Service Level Agreement* (SLA) dan kebijakan.
f. Pemantauan kinerja untuk menghindari masalah.
g. *Out-of-band* akses persyaratan.

Berawal mendefinisikan dan mencirikan manajemen untuk arsitektur jaringan, dan bagaimana rencana untuk pemantauan, konfigurasi ulang, dan merencanakan pemecahan masalah jaringan. kemudian memeriksa persyaratan instrumentasi dan protokol manajemen jaringan. Hal ini akan menyebabkan pertimbangan untuk mengembangkan arsitektur pengelolaan jaringan [5]. Manajemen jaringan dapat dilihat sebagai suatu struktur yang terdiri dari beberapa lapisan:

a. Manajemen Bisnis: pengelolaan aspek bisnis jaringan misalnya, pengelolaan anggaran sumber-sumber, perencanaan, dan kesepakatan.
b. Layanan Manajemen: manajemen pengiriman layanan kepada pengguna misalnya, penyedia layanan ini akan mencakup manajemen *bandwidth* akses, Penyimpanan data dan pengiriman aplikasi.
c. Manajemen jaringan: manajemen perangkat jaringan semua di seluruh jaringan.

d. Elemen manajemen: pengelolaan koleksi serupa perangkat jaringan misalnya, akses *router* atau sistem manajemen pelanggan.
e. Manajemen jaringan-elemen: manajemen perangkat jaringan individu misalnya, satu *router*, *switch* atau *hub*.

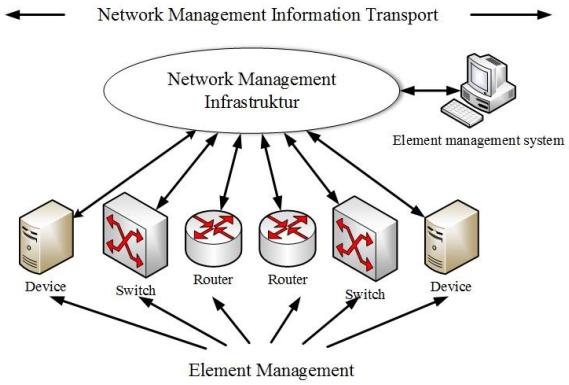
Pada Gambar 1, struktur ini adalah pendekatan top-*down*, dengan komponen yang paling abstrak (manajemen bisnis) di bagian atas dari *hirarki*, dan *specified* kebanyakan, komponen nyata (jaringan-elemen manajemen) di bagian bawah dari *hirarki*. Ini ditunjukkan dalam Gambar 1. Sejalan dengan itu, karena komponen menjadi lebih abstrak, bahwa cara mereka diterapkan dan diukur (informasi unsur) berubah. Dengan demikian, di bagian bawah dari *hirarki* ini (jaringan-elemen, elemen, Jaringan), manajemen diaplikasikan dengan variabel dan parameter, sementara di bagian atas *hirarki* ini (layanan, bisnis), manajemen yang diterapkan secara lebih abstrak, menggunakan kebijakan.



Gambar 1 Network Management Hierarchy [5]

Umum untuk semua komponen arsitektur, dan kami akan mencari bahwa kebijakan dapat digunakan untuk setiap komponen. manajemen jaringan dapat dibagi menjadi dua fungsi dasar: pengangkutan manajemen informasi di seluruh sistem, dan manajemen elemen informasi NM Gambar 2. Fungsi-fungsi ini, seperti yang ditunjukkan pada Gambar 2, terdiri dari berbagai tugas monitoring, konfigurasi ulang, pemecahan masalah, dan perencanaan yang dilakukan oleh pengguna, administrator, dan personil jaringan. Salah satu tantangan pertama dalam mengembangkan arsitektur manajemen jaringan adalah untuk mendefinisikan apa yang manajemen jaringan benar-benar berarti bagi organisasi yang akan melakukan tugas dan menerima jasa-yaitu akhir, pengguna, atau pelanggan, dari sistem [5]. Ada empat kategori tugas pengelolaan jaringan, sesuai dengan empat tugas yang disebutkan:

1. Monitoring untuk aksi pemberitahuan notifikasi.
2. Pemantauan untuk analisis *trend* dan perencanaan.
3. Konfigurasi ulang parameter jaringan.
4. Mengatasi masalah jaringan.



Gambar 2 Mengelola Elemen dan Data Manajemen Transportasi Data [5]

1.4 Sistem Monitoring Jaringan

Sistem monitoring jaringan adalah salah satu sub sistem dari manajemen jaringan yang berfungsi menggambarkan kondisi suatu sistem, yang terus menerus memonitor kondisi jaringan dengan rentang waktu pengecekan tertentu, baik itu memonitor kondisi *hardware* maupun *software*, misalnya memeriksa ketersediaan perangkat yang dimonitor apakah status hidup atau mati, sehingga jika terjadi gangguan dapat diketahui secepatnya dengan melakukan notifikasi kepada seorang Technical Network & System.

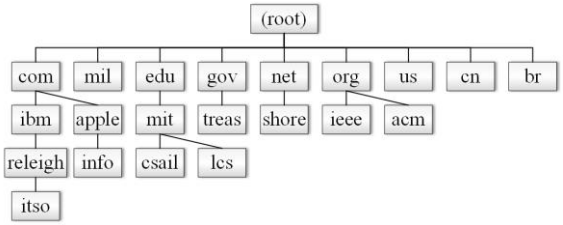
Ada dua pendekatan utama yang umumnya digunakan dalam memantau kinerja layanan jaringan dari dukungan QoS (Quality of Service) untuk menentukan apakah SLA (Service Level Agreement) telah dipenuhi:

a. Pemantauan jaringan pasif: dengan pemantauan jaringan pasif, perangkat jaringan merekam statistik pada jaringan *traffic*, yang dapat memberikan indikasi status pada elemen jaringan tertentu. *Polling* berkala biasanya digunakan untuk mengumpulkan data-data ini untuk pelaporan dan analisis. Ini adalah sebagian kecil yang tampak pada masing-masing perangkat dalam isolasi; dengan melihat beberapa elemen jaringan, pandangan agregat status layanan jaringan dapat disimpulkan. Pemantauan jaringan pasif tidak memerlukan tambahan apapun yang akan digunakan untuk tujuan pengukuran *traffic*.
b. monitoring jaringan yang aktif: Tidak seperti pemantauan pasif, pengawasan aktif melibatkan pengiriman tambahan *traffic* ke jaringan. *stream* tes sintetis yang terdiri dari "probe" paket dikirim di seluruh jaringan semata-mata untuk tujuan mengetahui karakteristik kinerja jaringan; analisis aliran diterima digunakan untuk karakterisasi ini. pengawasan aktif memberikan ukuran sebagian kecil SLA (Service Level Agreement) jaringan bahwa laporan kinerja yang diukur di sejumlah elemen jaringan sebagai suatu sistem.

1.5 Domain Name System

Domain Name System (DNS) adalah layanan pencarian *direktori* yang menyediakan pemetaan antara nama host di Internet dan alamat numerik. DNS penting untuk fungsi internet. Hal ini didefinisikan dalam RFC 1034 dan RFC 1035. Empat elemen DNS terdiri dari [3]:

a. Domain *name space*: DNS menggunakan ruang nama pohon-terstruktur untuk mengidentifikasi sumber daya di Internet.
b. DNS *database*: Secara konseptual, setiap *node* dan daun dalam nama struktur ruang nama pohon set informasi (misalnya, alamat IP, jenis sumber daya) yang terkandung dalam catatan *resource record* (RR). Koleksi semua RRS diatur ke dalam *database* terdistribusi.
c. *Name servers*: Ini adalah program server yang menyimpan informasi tentang bagian dari struktur pohon nama domain dan RRS terkait.
d. *Resolvers*: Ini adalah program yang mengekstrak informasi dari server nama dalam menanggapi permintaan *client*. Permintaan *client* khas adalah untuk alamat IP yang sesuai dengan nama domain tertentu.



Gambar 3 Bagian Internet Domain Tree [3]

1.6 JSON (JavaScript Object Nation)

JSON (JavaScript Object Nation) adalah format pertukaran data ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari bahasa pemograman JavaScript, Standar ECMA-262 Edisi ke-3-Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Phyton dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data [6].

1.7 API (Application Programming Interface)

Application programming interface (API) adalah sekumpulan perintah, fungsi, komponen, dan protokol yang disediakan oleh sistem operasi ataupun bahasa pemograman tertentu yang dapat digunakan oleh programmer saat membangun perangkat lunak

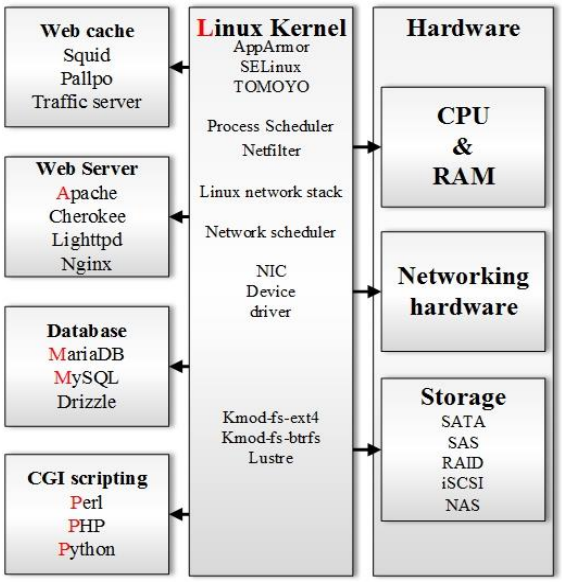
Dalam suatu pemograman dibutuhkan setidaknya ribuan system call per detik oleh karena itu banyak programmer yang menggunakan API. Di dalam API terdapat fungsi-fungsi/perintah untuk menggantikan bahasa yang digunakan dalam system calls dengan bahasa yang lebih mudah dimengerti oleh programmer. Fungsi yang dibuat dengan menggunakan API tersebut kemudian akan memanggil system call sesuai dengan sistem operasinya. Tidak tertutup kemungkinan nama dari system call sama dengan nama API.

1.8 Centos

Distribusi Linux CentOS (singkat dari Community ENTerprise Operating System) adalah linux kelas enterprise tersedia secara bebas, sistem operasi open source yang berasal dari kode sumber dari Red Hat Enterprise Linux (RHEL) dan dikembangkan dan dikelola oleh proyek CentOS. CentOS Proyek adalah lebih dari sekedar sebuah distribusi Linux. Anggota proyek bekerja untuk memberikan dukungan, materi pelatihan, dan di masa depan bahkan sertifikasi. Meskipun pada intinya proyek ini adalah distribusi CentOS, Proyek CentOS meliputi seluruh ekosistem perangkat lunak, pengembang dan proyek-proyek [7].

1.9 LAMP

LAMP (Linux, Apache, MariaDB, PHP) merupakan sebuah paket perangkat lunak bebas, bebas dalam arti menggunakan, mempelajari, mengubah, menyalin atau menjual sebuah perangkat lunak, pengguna tidak perlu meminta izin dari siapapun.Gambar Arsitektur LAMP dapat di lihat pada Gambar 4.



Gambar 4 Arsitektur LAMP

1.10 ICINGA

ICINGA merupakan salah satu tools sistem monitoring jaringan bersifat open source sehingga dapat di konfigurasi sesuai dengan kebutuhan, kegunaannya antara lain untuk memantau kondisi jaringan di lingkungan yang kompleks di beberapa lokasi, dapat memberi notifikasi pengguna jika sumber daya jaringan yang di monitor mati, juga

menghasilkan data kinerja untuk pelaporan. dengan tingkat fleksibel yang tinggi ICINGA mampu memonitor segala sesuatu yang terhubung dengan jaringan yang akan di tangani [8]. Gambar Arsitektur ICINGA dapat di lihat pada Gambar 5.

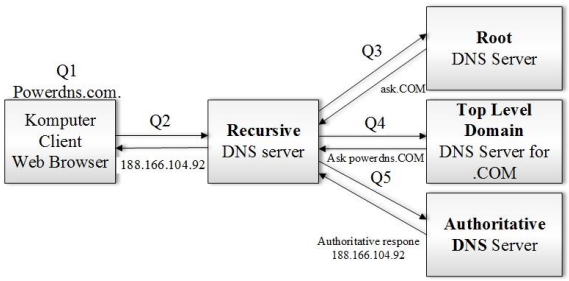


Gambar 5 Arsitektur ICINGA

1.11 PoworDNS

PowerDNS Recursor adalah *high-end*, dengan kinerja tinggi menyelesaikan nama server yang kekuatan resolusi DNS setidaknya seratus juta pelanggan. Memanfaatkan beberapa prosesor dan mendukung kemampuan skripting yang sama kuat dari *Authoritative* Server, *Recursor* memberikan performa terbaik sementara tetap mempertahankan fleksibilitas modern dalam penyebaran DNS bagi yang memerlukan [9].

PowerDNS yang digunakan berfungsi sebagai DNS server dan memanfaatkan API untuk media kontrol terhadap DNS, selain itu PowerDNS memiliki beberapa keunggulan seperti dukungan terhadap standar yang relavan, dengan langkah-langkah anti-spoofing, konfigurasi ulang tanpa *downtime*, tentunya pemanfaatan PowerDNS untuk memungkinkan *Dynamic Update* DNS yang bertujuan untuk DNS Failover.



Gambar 6 Mekanisme DNS

Dari Gambar 6 dapat dijelaskan mekanisme DNS dari permintaan *client* hingga *client* mendapat tanggapan dari DNS *Authoritative*. Langkah penjelasannya adalah sebagai berikut:

a. Client membuka Web Browser dan ingin mengakses domain powerdns.com (Q1). Sistem Operasi dan Web Browser menentukan apakah mengetahui domain powerdns.com. jika tidak mengetahui, Sistem Operasi meminta DNS Server untuk menyelesaikannya. Sistem Operasi melakukan permintaan terhadap DNS Server *Recursive*.
b. DNS Server *Recursive* apakah mengenali domain powerdns.com jika tidak DNS Server Recursive melakukan permintaan terhadap DNS Server *Root* (Q3).
c. DNS Server *Root* memberikan tanggapan tidak mengetahui domain powerdns.com tetapi DNS Server *Root* mengetahui di mana dapat menemukan DNS Server COM.
d. DNS Server *Recursive* menyimpan semua informasi dari DNS Server *Root*, menempatkan dalam *cache* dan kemudian melakukan permintaan terhadap DNS Server COM (Q4).
e. DNS Server COM memberikan tanggapan tidak mengetahui domain powerdns.com tetapi DNS Server COM mengetahui di mana dapat menemukan domain powerdns.com.
f. DNS Server *Recursive* menyimpan semua informasi dari DNS Server COM menempatkannya dalam *cache* dan kemudian melakukan permintaan terhadap DNS Server *Authoritative* (Q5).
g. DNS Server *Authoritative* memberikan tanggapan mengetahui domain powerdns.com dan memberikan informasi IP Address yang dapat di akses oleh Web Browser
h. DNS Server *Recursive* menyimpan semua informasi dari DNS Server *Authoritative* menempatkannya dalam *cache* dan kemudian memberikan jawaban ke Sistem Operasi.
i. Kemudian Sistem Operasi memberikan terhadap Web Browser.
j. Web browser kemudian membuat koneksi terhadap IP *address* dan meminta halaman web untuk domain powerdns.com

1.12 Kali Linux

Kali Linux adalah sebuah distro Linux berbasis Debian yang ditujukan untuk pengujian penetrasi dan audit keamanan yang canggih. Kali berisi beberapa ratus tools yang mengarah kepada berbagai tugas keamanan informasi, seperti pengujian penetrasi, riset keamanan, komputer forensik dan Reverse Engineering. Kali Linux dikembangkan, didanai dan dikelola oleh keamanan ofensif, perusahaan pelatihan keamanan informasi terkemuka [10].

2 ISI PENELITIAN

Analisis sistem pada tahap ini adalah untuk mengidentifikasi permasalahan yang terjadi pada ruang lingkup penelitian meliputi hardware dan software. Analisis ini diperlukan sebagai dasar untuk tahapan perancangan sistem dan untuk menentukan kebutuhan dari sistem yang akan diterapkan.

2.1 Analisis Masalah

Permasalahan dari penelitian ini adalah bagaimana mempermudah Technical Network & System dalam mengelola jaringan yang ditangani kemudian dapat menanggulangi masalah yang terjadi.

2.2 Analisis Sistem Berjalan dan Permasalahan

Analisis Sistem Berjalan dan Permasalahan pada tahap ini adalah memberikan gambaran meliputi sistem yang sedang berjalan, dapat dilihat pada Tabel 2.1.1.

Tabel 2.1.1 Identifikasi Permasalahan

Objek Analisa	Sistem Berjalan	Permasalahan
Pemeriksaan koneksi	Dilakukan secara manual, dengan melakukan pengujian terhadap *host* ataupun *service* satu per satu	Memerlukan banyak waktu untuk menguji ketersediaan *host* maupun *service* menyulitkan dalam penanggulangan masalah
Pencatatan kondisi *host* dan *service*	Belum ada	Tidak tersedianya dokumentasi keadaan jaringan
Sistem Notifikasi	Belum ada	Tidak ada dokumentasi pelaporan keadaan jaringan
Pemantauan kondisi perangkat pada jaringan	Melakukan *remote* melalui *port* SSH terhadap perangkat satu per satu	Sulit dilakukan pada banyak perangkat dalam waktu rutin
Sistem penanggulangan	Dilakukan secara manual	Sulit dilakukan karena memerlukan waktu dalam pengalihan *server*

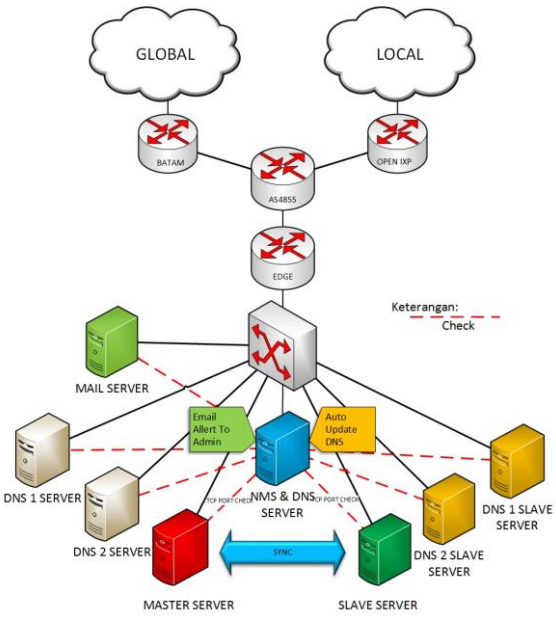
Setelah melakukan identifikasi permasalahan maka selanjutnya mengajukan sistem yang ditawarkan untuk mengatasi permasalahan-permasalahan tersebut, dapat dilihat pada tabel Tabel 2.1.2

Tabel 2.1.2 Sistem Yang Ditawarkan

Objek Analisa	Sistem Berjalan	Sistem Ditawarkan
Pemeriksaan koneksi	Dilakukan secara manual, dengan melakukan pengujian terhadap *host* ataupun *service* satu per satu dengan	Pemeriksaan seluruh *host* dan *service* secara otomatis
Pencatatan kondisi *host* dan *service*	Belum ada	Pencatatan otomatis terhadap kondisi *host* dan *service*
Sistem Notifikasi	Belum ada	Notifikasi berbasis email memberikan notifikasi terkait dengan kondisi yang ada kepada Technical Network & System
Pemantauan kondisi perangkat pada jaringan	Melakukan pemantauan terhadap perangkat satu per satu	Pemantauan terdistribusi keseluruh perangkat secara otomatis
Sistem penanggulangan	Dilakukan secara manual	Penanggulangan secara otomatis dengan metode *DNS Failover*

2.3 Analisis Arsitektur Sistem

Analisis Arsitektur Sistem berguna untuk menggambarkan bagaimana sistem monitoring akan diterapkan dan menggambarkan interaksi antar komponen arsitektur di dalamnya, untuk lebih jelasnya dapat di lihat pada gambar Gambar 7.



Gambar 7 Arsitektur Sistem Monitoring Jaringan

2.4 Analisis Kebutuhan Hardware dan Software

Tahap ini menjelaskan kebutuhan hardware dan software serta parameter yang dibutuhkan terkait untuk mendukung pada tahapan implementasi.

1. Hardware

   Hardware/Perangkat keras yang dimaksud dalam kebutuhan pada penelitian ini adalah seperangkat komputer yang di install Sistem Operasi yang akan di gunakan untuk melakukan implementasi. Perangkat keras yang dibutuhkan dapat di lihat pada Tabel 2.1.3.

Tabel 2.1.3 Kebutuhan Minimum Perangkat Keras

Perangkat keras	Spesifikasi
Processor	Intel Pentium Dual-Core 800MHz
Ram	3 GB
Harddisk	40 GB
Monitor	CRT/LCD minimum resolusi 1024x768
Mouse+Keyboard	-

2.5 Perancangan Komunikasi Antar Sistem

Pada tahap ini perancangan Komunikasi Antar sistem menjelaskan cara kerja dalam komunikasi antara Sistem Monitoring Jaringan dengan Domain Name System. Perancangan Komunikasi Antar Sistem dapat dilihat pada Gambar 8.

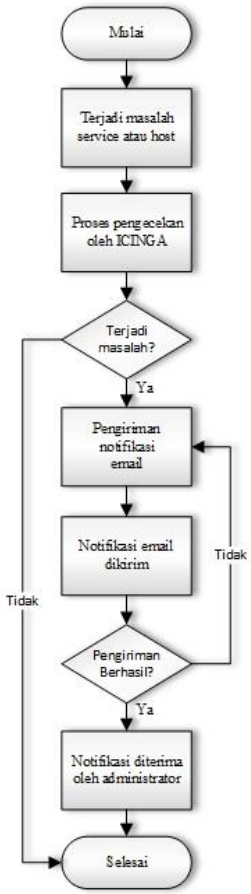


Gambar 8 Komunikasi Anatar Sistem

Pada Gambar 8 menjelaskan komunikasi yang dilakukan oleh Sistem Monitoring Jaringan dengan Domain Name System, dilakukan dengan memanfaatkan API sebagai media berkomunikasi untuk perubahan DNS.

2.6 Perancangan Sistem Notifikasi

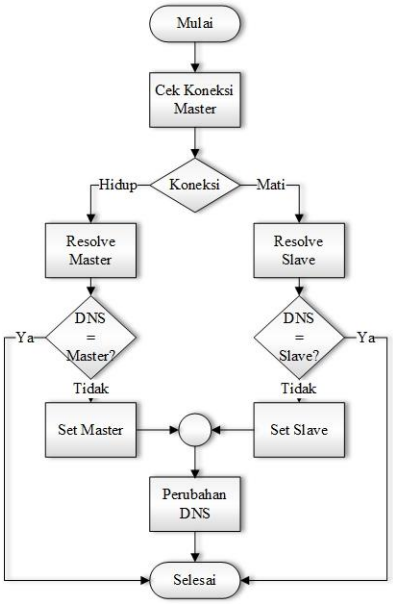
Pada tahap ini perancangan Sistem Notifikasi menjelaskan cara kerja dalam notifikasi ketika terjadi perubahan status. Perancangan Sistem Notifikasi dapat dilihat pada Gambar 9.



Gambar 9 Diagram Alir Notifikasi Email

2.7 Perancangan Metode DNS Failover

Pada tahap ini perancangan DNS Failover menjelaskan cara kerja dalam menanggulangi server yang mengalami gangguan. Perancangan DNS Failover dapat dilihat pada Gambar 10.



Gambar 10 Diagram Alir DNS Failover

2.8 Lingkungan Implementasi

Lingkungan implementasi merupakan kebutuhan lingkungan meliputi *hardware* maupun *software* yang direkomendasikan untuk menjalankan atau mengakses sistem.

1. Lingkungan Hardware
   Berikut ini adalah spesifikasi *hardware* yang digunakan dalam menjalankan sistem, perangkat keras yang dibutuhkan dapat di lihat pada Tabel 2.1.4.

Tabel 2.1.4 Implementasi Hardware

No.	Perangkat keras	Spesifikasi
1	Processor	i5 3.30 Ghz
2	RAM	4GB
3	Harddisk	250 GB
4	Monitor	CRT/LCD minimum resolusi 1024x768
5	Mouse+Keyboard	-

2. Lingkungan Software
   spesifikasi software yang digunakan dalam menjalankan sistem, perangkat lunak yang dibutuhkan.

2.9 Pengujian Sistem

Pengujian sistem adalah tahapan untuk menguji sistem yang telah diimplementasikan, dengan tujuan untuk menemukan kekurangan maupun kelebihan yang ada pada sistem tersebut. Pengujian yang dilakukan mencakup parameter-parameter dan komponen-komponen yang terkait dengan Sistem Monitoring Jaringan dan DNS Failover.

2.10 Rencana Pengujian

Rencana pengujian merupakan cara yang akan dilakukan untuk memastikan bahwa sistem yang diimplementasikan telah memenuhi persyaratan yang telah dirancang sebelumnya. Pada tahapan pengujian dilakukan pengujian untuk fungsional yang ada apakah telah sesuai atau masih ada kekurangan.

2.11 Skenario Pengujian

Skenario pengujian menjelaskan urutan dari pengujian yang akan dilakukan pada perangkat lunak yang telah diimplementasikan. Skenario pengujian yang akan dilakukan fokus pada permasalahan yang terjadi. Maka akan dibuat dua skenario pengujian yang akan dilakukan yaitu pengujian dengan manual dan pengujian dengan *tools*. Skenario-skenario pengujian tersebut diantaranya:

1. Skenario manual, dengan mematikan *service http* server master yang berkaitan dengan *triger* DNS Failover, dilakukan secara manual terhadap *service* tersebut.
2. Skenario *tools*, dengan menggunakan *tools* yang dilakukan dengan mengirimkan beban *trafic* terhadap server master. Detail dari skenario pengujian *trafic* dapat dihihat pada. Adapun besaran paket adalah *KiloByte*.

Tabel 2.1.5 Pengujian Trafic

No	Percobaan	Jumlah paket	Ukuran Paket (KB)	Port
1	Percobaan ke-1	10	50	80
2		10	60	80
3		10	70	80
4		10	80	80
5		10	90	80
6		10	100	80
7		10	120	80
8	Percobaan ke-2	100	50	80
9		100	60	80
10		100	70	80
11		100	80	80
12		100	90	80
13		100	100	80
14		100	120	80
15	Percobaan ke-3	1000	50	80
16		1000	60	80
17		1000	70	80
18		1000	80	80
19		1000	90	80
20		1000	100	80
21		1000	120	80
22	Percobaan ke-4	10000	50	80
23		10000	60	80
24		10000	70	80
25		10000	80	80
26		10000	90	80
27		10000	100	80
28		10000	120	80
29	Percobaan ke-5	100000	50	80
30		100000	60	80
31		100000	70	80
32		100000	80	80
33		100000	90	80
34		100000	100	80
35		100000	120	80
36	Percobaan ke-6	1000000	50	80
37		1000000	60	80
38		1000000	70	80
39		1000000	80	80
40		1000000	90	80
41		1000000	100	80
42		1000000	120	80
43	Percobaan ke-7	1000000	50	80
44		1000000	60	80
45		1000000	70	80
46		1000000	80	80
47		1000000	90	80
48		1000000	100	80
49		1000000	120	80

2.12 Hasil Pengujian

Berikut ini adalah hasil pengujian yang telah diskenariokan pada bahasan sebelumnya.

1. Pengujian pertama akan dilakukan berupa simulasi ketika *service* dimatikan secara manual yang nantinya sistem monitoring akan mengecek ketersediaan *service http* yang berjalan terhadap *service http* server master yang dimatikan.
2. Pengujian kedua akan dilakukan berupa simulasi ketika *service* diberi beban *trafic* tinggi dengan *tools* yang nantinya sistem monitoring akan mengecek ketersediaan *service* yang berjalan terhadap service server dibebani beban *trafic* tinggi. Pengujian tersebut merupakan jenis serangan terhadap komputer atau server yang dikenal *DDoS* (*Distributed Denial of Service attacks*) dengan menghabiskan *resource bandwith* yang dimiliki server master. Dengan menggunakan *tools goldeneye* sebagai *tools DDoS* dan Kali Linux sebagai Sistem Operasi.

2.13 Evaluasi Pengujian

Berdasarkan hasil pengujian yang telah dilakukan, baik itu pengujian yang dilakukan manual maupun pengujian dengan *tools* dapat ditarik kesimpulan bahwa:

1. Fungsional yang ada pada sistem monitoring jaringan berjalan sesuai dengan perancangan yang telah dibuat sebelumnya, dalam penanganan kesalahan yang ada pada sistem.
2. Kemampuan sistem dalam menanggulangi permasalahan dengan DNS Failover dapat dilihat bahwa sistem dapat menanggulangi dengan cepat sesuai dengan waktu yang ditetapkan.

3 PENUTUP

3.1 Kesimpulan

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan pada sistem Sistem Monitoring Jaringan menggunakan ICINGA dan DNS Failover menggunakan PowerDNS dapat ditarik beberapa kesimpulan sebagai berikut:

1. Sistem Monitoring Jaringan yang diimplementasikan dapat digunakan oleh pihak perusahaan untuk memonitoring server.
2. Sistem Monitoring Jaringan yang diimplementasikan dapat mempermudah Technical Network & System dalam memonitor kondisi jaringan serta membantu menanggulangi permasalahan yang akan terjadi

3.2 Saran

Adapun saran untuk pengembangan Sistem Monitoring Jaringan adalah sebagai berikut:

1. Pengembangan Sistem Monitoring Jaringan lebih lanjut agar bisa digunakan untuk semua zone server.
2. Hasil log dari ICINGA dapat dijadikan informasi nilai tambah untuk pengembangan dalam network management

DAFTAR PUSTAKA

[1] P. Beertema, “Common DNS Data File Configuration Errors,” Oktober 1993. [Online]. Available: https://tools.ietf.org/html/rfc1537.

[2] Sugiyono, “Perspektif Metode Penelitian Kualitatif dan Kuantitatif,” dalam Metode Penelitian Kuantitatif Kualitatif dan R&D, Bandung, Alfabeta, 2014, p. 4.

[3] W. Stallings, Data and Computer Communications (Tenth Edition), Tenth edition penyunt., New Jersey: Pearson Prentice Hall, 2014, p. 917.

[4] W. Stallings, Data and Computer Communications (Eighth Edition), Pearson Prentice Hall, 2006.

[5] I. B. J. E. A. F. C. F. H.-G. H. J. D. M. Sebastian Abeck, Network Management, Elsevier, 2009.

[6] JSON, “Introducing JSON,” [Online]. Available: http://www.json.org/json-id.html.

[7] T. V. R. A. Peter Membrey, The Definitive Guide to CentOS, Apress, 2009.

[8] ICINGA, “About Icinga 2 : Icinga web,” [Online]. Available: http://docs.icinga.org/icinga2/latest/doc/module/icinga2/toc#!/icinga2/latest/doc/module/icinga2/chapter/about-icinga2#what-is-icinga2. [Diakses 5 Oktober 2016].

[9] PowerDNS, “PowerDNS Recursor,” [Online]. Available: https://www.powerdns.com/recursor.html.

[10] Kali Linux, “Introduction Kali Linux,” [Online]. Available: http://docs.kali.org/introduction/what-is-kali-linux.