

Rancangan Desain Implementasi Pengujian Konektivitas Antara NMS Nagios Dan ITSM iTop Dengan Aspek Otomatisasi

1st Muhammad Rafi
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
rafichan@student.telkomuniversity.ac.id

2nd Adityas Widjarto
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
adtwjrt@telkomuniversity.ac.id

3rd Muhammad Fathinuddin
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
muhammadfathinuddin@telkomuniversity.ac.id

Abstrak—Penelitian ini berfokus pada implementasi dan profiling perangkat lunak open-source untuk meningkatkan efisiensi dalam pengelolaan layanan TI dan infrastruktur jaringan dengan aspek otomasi. Dengan menggunakan metode Network Development Life Cycle (NDLC), penelitian ini melibatkan tahap analisis, desain, dan simulasi terhadap tiga pengujian utama, yaitu Nagios sebagai platform Network Monitoring System (NMS), iTop sebagai platform IT Service Management (ITSM), dan pengujian otomasi antara kedua platform tersebut. Hasil pengujian menunjukkan keberhasilan lima pengujian pada Nagios yang berkaitan dengan proses monitoring jaringan serta dua pengujian pada iTop yang terkait dengan proses pembuatan tiket. Namun, pada pengujian otomasi, proses pembuatan tiket otomatis melalui script createTicket-nagios.php tidak berhasil. Analisis menunjukkan bahwa event handler yang diinisiasi oleh perintah create-iTop-ticket terbaca, namun script createTicket-nagios.php tidak mampu mengonversi notifikasi dari Nagios menjadi tiket pada iTop, menunjukkan adanya masalah dalam komunikasi antara script tersebut dan iTop. Berdasarkan temuan ini, penelitian memberikan rekomendasi untuk menguji konektivitas antara script Nagios dengan API iTop. Solusi yang diajukan melibatkan perancangan script baru yang mendasari script createTicket-nagios.php, serta penyusunan flowchart, Data Flow Diagram (DFD), dan Class Diagram untuk memastikan komunikasi yang efektif dan otomatisasi yang berhasil antara kedua platform tersebut.

Kata kunci— Otomatisasi, otomasi, NMS, ITSM, monitoring, ticketing, NDLC

I. PENDAHULUAN

Perkembangan teknologi informasi (TI) yang pesat telah membawa implikasi signifikan terhadap kompleksitas dan dinamika infrastruktur teknologi informasi di berbagai organisasi. Untuk menghadapi tantangan ini, proses otomasi menjadi kunci utama dalam meningkatkan efisiensi dan responsivitas operasional [1]. Khususnya, manajemen layanan teknologi informasi dan pemantauan infrastruktur menjadi fokus kritis dalam mengelola pertumbuhan yang cepat dari perangkat keras dan aplikasi.

Manajemen layanan teknologi informasi atau biasa dikenal dengan istilah *Information Technology Service Management* (ITSM) merupakan metode/praktik manajemen sistem teknologi informasi yang berupaya menyelaraskan TI dengan kebutuhan bisnis untuk mengelola penyediaan

layanan TI yang efisien [2]. Dalam hal ini, salah satu langkah strategis yang tepat untuk diambil dalam memenuhi kebutuhan terhadap proses pengelolaan infrastruktur jaringan yang efisien adalah menggunakan aplikasi ITSM berbasis open-source [3]. *Software ITSM open-source* adalah solusi perangkat lunak yang dirancang untuk mengelola layanan TI secara efisien, dengan akses terbuka terhadap kode sumbernya. *Software ITSM open-source* memungkinkan organisasi untuk menyesuaikan dan memodifikasi sistem sesuai dengan kebutuhan spesifik mereka tanpa biaya lisensi yang tinggi. *Software* ini juga menawarkan berbagai fitur yang sama seperti ITSM berbayar seperti manajemen insiden, manajemen perubahan, dan pengelolaan aset, serta integrasi dengan alat lain untuk mendukung operasi TI yang lebih baik. Keuntungan utama dari penggunaan aplikasi open-source yaitu tidak perlu mengeluarkan biaya dalam penggunaannya, fleksibilitas serta komunitas pengguna yang aktif yang dapat memberikan dukungan dan pengembangan berkelanjutan [4]. Ini memungkinkan organisasi untuk mengadopsi dan mengadaptasi solusi ITSM yang sesuai dengan anggaran dan kebutuhan mereka secara lebih efektif.

Di sisi lain dalam konteks pemantauan infrastruktur jaringan, terdapat beberapa *aplikasi Network Monitoring System* (NMS) yang ada di berbagai perusahaan. Profilisasi dengan NMS memungkinkan organisasi untuk melakukan pemantauan proaktif terhadap kinerja, ketersediaan, dan keamanan sistem jaringan. Dengan notifikasi *real-time* dan kemampuan analisis, NMS menjadi salah satu elemen penting dalam strategi manajemen jaringan TI sebuah perusahaan.

Integrasi antara sistem ITSM dengan sistem NMS menitikberatkan pada aspek otomatisasi. Dengan mengintegrasikan manajemen layanan TI yang terpadu bersama pemantauan infrastruktur yang efektif, diharapkan dapat terbentuk ekosistem yang responsif terhadap perubahan, meminimalkan *downtime*, serta meningkatkan produktivitas organisasi.

Oleh karena itu, penelitian ini bertujuan untuk memberikan wawasan yang mendalam mengenai bagaimana implementasi dan *profiling* perangkat lunak open-source dapat meningkatkan otomatisasi dan efisiensi dalam mengelola layanan TI dan infrastruktur jaringan.

Penelitian ini menggunakan metode *Network Development Life Cycle* (NDLC), yang merupakan pendekatan terstruktur untuk perancangan, implementasi, dan pemeliharaan solusi jaringan. NDLC mencakup beberapa tahap penting, mulai dari analisis kebutuhan, perancangan arsitektur jaringan, hingga implementasi dan evaluasi. Metode ini memberikan panduan sistematis dalam mengidentifikasi dan mengatasi masalah potensial, memastikan bahwa sistem dapat beroperasi secara efisien dengan baik. Dengan mengikuti siklus hidup desain jaringan ini, penelitian ini bertujuan untuk mengoptimalkan proses otomatisasi dan memastikan bahwa koneksi antara NMS dan ITSM dapat memenuhi standar operasional dan teknis yang diharapkan.

II. KAJIAN TEORI

A. Otomatisasi

Otomatisasi didefinisikan sebagai "penciptaan dan penerapan teknologi untuk memantau dan mengendalikan produksi dan pengiriman produk dan layanan". Definisi ini menyoroti cakupan otomatisasi yang luas, yang tidak hanya mencakup manufaktur tetapi juga sektor-sektor seperti transportasi, utilitas, dan teknologi [5].

Otomatisasi adalah cara pelaksanaan prosedur dan tata kerja secara otomatis, dengan pemanfaatan yang menyeluruh dan seefisien mungkin atau mesin, sehingga bahan dan sumber yang ada dapat dimanfaatkan. Tujuan otomatisasi, yakni pemanfaatan yang seefisien mungkin atas, uang, tenaga kerja, material, waktu dan mesin atau alat-alat kerja, Pemeliharaan kecepatan dan ketepatan prosedur kerja, dan meningkatkan produksi [6].

B. Network Monitoring System

Network Monitoring System (NMS) merupakan sistem yang sangat penting untuk operasi jaringan apapun karena memiliki fungsi untuk mengamati atau memonitor sistem jaringan komputer yang sedang berjalan dan memungkinkan deteksi dini kesalahan jaringan. *Monitoring* mencakup pemantauan performa dan ketersediaan serta semua operasi jaringan, yang semuanya sangat penting dalam operasi jaringan. *Notification Alert* adalah salah satu fitur penting yang disediakan oleh NMS yang memungkinkan untuk mengirimkan notifikasi kepada *Administrator* jika sistem mendeteksi masalah pada *host* atau *service* [7].

Pengiriman notifikasi dalam sistem *monitoring* jaringan biasanya ditentukan oleh nilai ambang batas tertentu yang dapat diatur untuk setiap objek yang dimonitor sesuai kebutuhan. Media notifikasi yang digunakan juga dapat disesuaikan dalam sistem *monitoring* jaringan, seperti melalui *email* atau SMS [8]. Beberapa alasan mengapa *monitoring* diperlukan untuk sebuah jaringan adalah:

- Untuk memantau kinerja jaringan secara keseluruhan.
- Untuk mengetahui status layanan (*up/down*) dari *host* yang dimonitor secara *real-time* dengan sistem notifikasi peringatan.
- Sebagai media pencatatan *log* terhadap layanan, sehingga semua aktivitas dari *host* yang dimonitor dapat direkam.

C. Nagios

Nagios merupakan sistem *monitoring* berbasis *open-source* yang berfungsi sebagai penjaga kesehatan sistem jaringan dengan melakukan pemantauan secara terus-menerus terhadap *host*, layanan, atau sumber daya dengan kemampuannya untuk mendeteksi dan memberikan notifikasi *real-time* terkait masalah potensial. Nagios secara rutin memeriksa apakah mesin beroperasi dengan baik dan memverifikasi berbagai layanan di dalamnya untuk memastikan semuanya berjalan dengan lancar [9].

Sistem *monitoring* di Nagios dibedakan menjadi dua kategori, yaitu *hosts* dan *service*. *Hosts* mewakili sebuah perangkat fisik atau virtual di dalam jaringan seperti *server*, *router*, dan lain-lain. *Services* adalah sebuah fungsi partikular, sebagai contoh, sebuah proses *Secure Shell* (SSH) pada *server* bisa didefinisikan sebagai *service* yang akan dimonitor. Setiap *service* pasti berhubungan dengan sebuah *host* yang menjalankannya. Berikut adalah fitur-fitur yang tersedia di Nagios:

- Pemantauan layanan jaringan (seperti SMTP, POP3, HTTP, PING, dan lainnya)
- Pemantauan sumber daya *host* (seperti beban prosesor, penggunaan disk, dan lainnya)
- Pemeriksaan layanan yang dapat dilakukan secara paralel
- Antarmuka *web* opsional untuk melihat status jaringan, urutan masalah, notifikasi, *log file*, dan lainnya
- Kemampuan untuk mendefinisikan kejadian yang ditangani selama layanan atau *host* beroperasi, memudahkan pemecahan masalah
- Perputaran *log file* secara otomatis
- Mengirim notifikasi ketika ada masalah terhadap layanan atau *host* dengan berbagai metode seperti *email*, *pager*, SMS, WhatsApp, Telegram, atau metode yang didefinisikan pengguna

D. Information Technology Service Management

Information Technology Service Management (ITSM) merupakan seperangkat layanan yang terstruktur dan terdefinisi dengan baik yang berfokus pada manajemen infrastruktur TI, komponen serta aplikasi bisnis dan proses terkait [10]. ITSM mengacu pada jumlah proses dan praktik yang dibutuhkan untuk mengelola dan mendukung layanan teknologi informasi. ITSM bertujuan untuk mendukung vendor secara independen dan sepenuhnya spektrum layanan TI dari jaringan ke aplikasi hingga layanan bisnis lengkap.

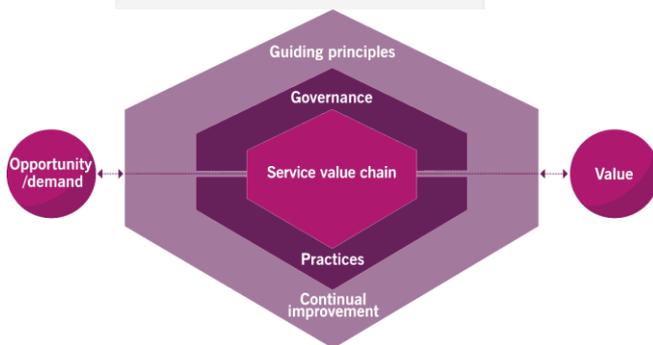
E. Information Technology Infrastructure Library

Information Technology Infrastructure Library (ITIL) merupakan praktik untuk mengelola layanan TI dan meningkatkan dukungan TI serta tingkat layanan. Tujuan utama ITIL adalah memastikan bahwa layanan TI selaras dengan tujuan bisnis, bahkan saat tujuan bisnis berubah. ITIL memberikan deskripsi detail tentang praktik layanan TI dengan daftar cek, tugas, serta prosedur yang menyeluruh yang dapat disesuaikan dengan segala jenis organisasi TI [11].

ITIL v4 terdiri dari lima bagian praktik manajemen yang mendukung struktur inti pada *Service Value System*. Kelima bagian tersebut adalah sebagai berikut [12]:

- 1) *Guiding Principles*
Guiding Principles berisi prinsip-prinsip yang memberikan panduan bagi organisasi dalam membuat keputusan dan menentukan arah dalam konteks manajemen layanan.
- 2) *Governance*
Governance menyediakan kerangka kerja untuk tata kelola dan kontrol dalam organisasi, memastikan bahwa semua aktivitas selaras dengan tujuan dan kebijakan organisasi.
- 3) *Service Value Chain*
Service Value Chain (SVC) merupakan bagian inti dari SVS, yang menggambarkan serangkaian aktivitas yang harus dilalui untuk menyediakan layanan dan menciptakan nilai. Ini mencakup enam aktivitas kunci: *Engage, Plan, Design & Transition, Obtain/Build, Deliver & Support, dan Improve.*
- 4) *Practices*
Praktik dalam ITIL v4 mencakup panduan dan aktivitas yang terkait dengan manajemen layanan. Ada 34 praktik yang dibagi menjadi tiga kategori: *General Management Practices, Service Management Practices, dan Technical Management Practices.*
- 5) *Continual Improvement Model*
Continual Improvement Model merupakan model yang digunakan untuk melakukan perbaikan berkelanjutan pada semua aspek organisasi, mulai dari proses hingga layanan dan teknologi.

Kelima bagian dari ITIL v4 ini bekerja secara sinergis untuk mendukung SVS dengan memastikan bahwa semua aktivitas dan keputusan dalam manajemen layanan TI selaras dengan tujuan bisnis. Bersama-sama, komponen-komponen ini memastikan bahwa organisasi dapat menciptakan dan memelihara nilai layanan TI yang efektif dan relevan dalam konteks yang dinamis. Struktur SVS dapat dilihat pada Gambar dibawah ini.



GAMBAR 1
(LIMA BAGIAN PENDUKUNG PADA STRUKTUR VALUE SYSTEM)

F. *Service Desk*

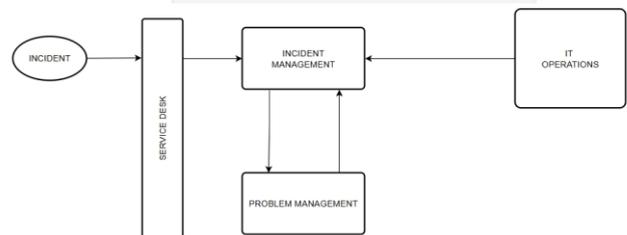
Dalam ITIL, *Service Desk* adalah fungsi kunci dalam dalam tahap *Practices* bagian *Service Management Practices* yang dirancang untuk memenuhi kebutuhan komunikasi antara pengguna dan dukungan TI. *Service Desk* juga berfungsi sebagai titik pusat dalam integrasi berbagai proses dukungan ITIL, seperti *Incident Management, Problem Management, dan Change Management.* Tujuan utama

Service Desk adalah untuk menerima dan menangani insiden serta permintaan layanan, lalu meneruskan insiden dan permintaan tersebut ke proses TI terkait. Oleh karena itu, penting bagi *Service Desk* untuk terintegrasi dengan baik dengan proses ITIL lainnya.

Keuntungan dari penerapan *Service Desk* dapat dirasakan oleh seluruh elemen yang mendukungnya. Beberapa manfaat implementasi *Service Desk* diuraikan sebagai berikut [13]:

1. Akses layanan yang lebih baik melalui satu titik kontak bagi pengguna, serta peningkatan kualitas layanan yang berujung pada kepuasan pengguna yang lebih tinggi.
2. Kemampuan yang lebih baik untuk berkomunikasi dan berbagi informasi dengan pengguna.
3. Peningkatan kapasitas untuk memberikan layanan secara proaktif kepada pengguna.
4. Pengendalian aktivitas pengguna yang lebih baik dan penggunaan sumber daya TI yang lebih efisien.
5. Penyelesaian masalah yang lebih cepat.
6. Peningkatan produktivitas bagi pengguna dan departemen TI.
7. Kualitas informasi yang dikumpulkan yang mendukung proses pengambilan keputusan.

Posisi *Service Desk* dalam kerangka ITIL, bersama dengan fungsi dan proses lainnya, ditunjukkan pada gambar dibawah ini.



GAMBAR 2
(POSISI *SERVICE DESK* DIANTARA PROSES DAN FUNGSI LAIN PADA ITIL)

Gambar 2 menggambarkan bahwa sebuah insiden harus melewati *Service Desk* sebagai titik kontak utama dengan pengguna sebelum diteruskan ke proses dan fungsi lain dalam *framework* ITIL. Setelah mengkategorikan sebuah insiden, selanjutnya *IT Operations* yang mengoperasikan insiden yang diterima berdasarkan praktik *Incident Management* ataupun *Problem Management.*

G. iTop

iTop adalah aplikasi ITSM berbasis *open-source* yang menggabungkan beberapa proses operasional TI perusahaan. Aplikasi iTop memiliki banyak fitur, termasuk manajemen kejadian dan permintaan pengguna, transparansi SLA, manajemen masalah, dan *chat messenger.* Aplikasi iTop membantu bisnis mendapatkan laporan tentang layanan pelanggan yang baik. [14].

iTop menyediakan berbagai fitur untuk mengelola berbagai aspek infrastruktur TI, yaitu:

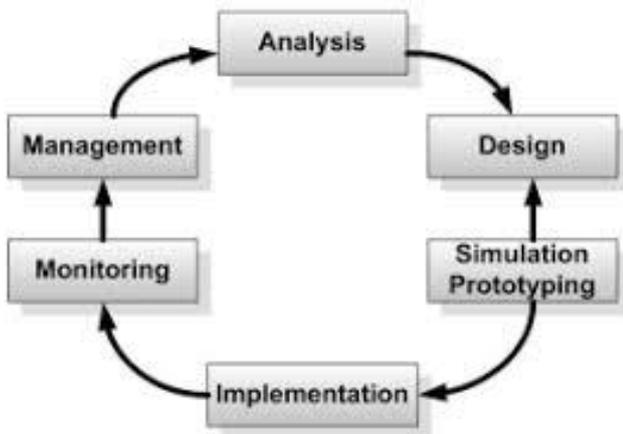
1. *Incident Management;* Mengelola dan menyelesaikan insiden atau masalah yang terjadi dalam lingkungan TI.

2. *User Request*; Menangani permintaan layanan dari pengguna atau pelanggan.
3. *Problem Management*; Identifikasi, analisis, dan pemecahan masalah yang berdampak pada layanan TI.
4. *Change Management*; Mengelola perubahan dalam lingkungan TI dengan cara yang terstruktur dan terkoordinasi.
5. *Configuration Management*; Mengelola dan memelihara informasi tentang konfigurasi dan komponen sistem TI.
6. *Asset Management*; Melacak dan mengelola aset TI, termasuk perangkat keras dan perangkat lunak.

iTop juga menyediakan antarmuka berbasis *web* yang memungkinkan pengguna untuk mengakses dan mengelola data serta fitur dari berbagai perangkat. Sistem ini sering digunakan untuk meningkatkan efisiensi operasional TI, memperbaiki pelayanan pelanggan, dan memastikan kepatuhan terhadap standar dan kebijakan TI.

H. Network Development Life Cycle

Network Development Life Cycle (NDLC) adalah pendekatan sistematis untuk merancang, mengembangkan, dan mengelola jaringan komputer. Proses ini sering digunakan untuk memastikan bahwa jaringan dibangun secara efektif dan efisien. Metode tersebut terdiri dari *analysis, design, simulation prototype, implementation, dan monitoring*. Gambar dibawah ini menjelaskan tahapan dari metode NDLC [15].



GAMBAR 3
(TAHAPAN NDLC)

1) Tahap *Analysis*

Tahap analisis adalah langkah awal dalam proses yang mencakup pemahaman kebutuhan, permasalahan yang ada, keinginan pengguna, serta topologi jaringan yang sudah ada. Pada tahap ini, data yang dibutuhkan untuk merumuskan masalah dan solusi dikumpulkan. Proses ini melibatkan identifikasi sistem yang sedang berjalan dan menganalisis jenis pengembangan sistem yang akan diterapkan untuk mengatasi kendala yang ada.

2) Tahap *Design*

Pada tahap ini, berdasarkan data yang telah diperoleh pada tahap analisis, selanjutnya membuat desain topologi jaringan yang akan dibangun, termasuk desain akses data dan lainnya.

3) Tahap *Simulation Prototype*

Tahap *Simulation Prototype* melibatkan pengembangan jaringan dalam bentuk simulasi. Tujuan dari tahap ini adalah untuk mengevaluasi kinerja jaringan yang akan dibangun dan menyediakan materi untuk presentasi serta berbagi informasi dengan tim pengembangan sistem jaringan.

4) Tahap *Implementation*

Tahap ini mungkin memakan waktu lebih lama. Selama proses implementasi, diterapkan semua rencana dan desain yang telah dibuat sebelumnya. Pada tahap ini, akan terlihat bagaimana pengembangan yang dilakukan mempengaruhi sistem yang ada.

5) Tahap *Monitoring*

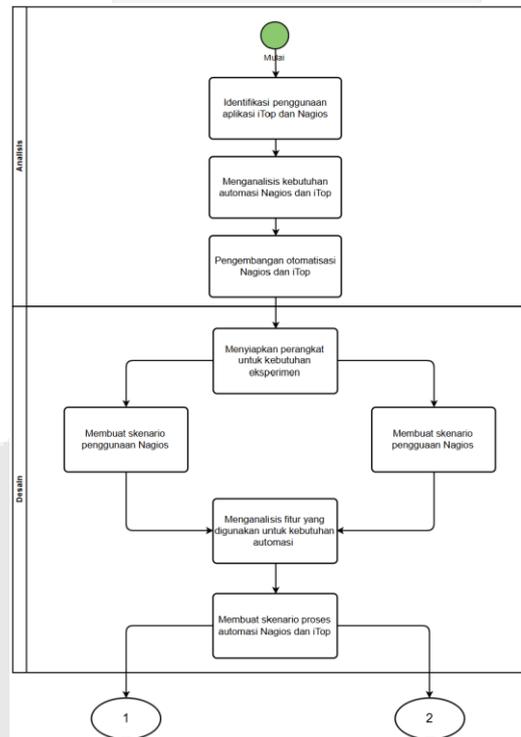
Setelah tahap implementasi, tahap *monitoring* menjadi sangat penting untuk memastikan bahwa jaringan dan komunikasi berfungsi sesuai dengan keinginan dan tujuan yang ditetapkan pada tahap analisis awal.

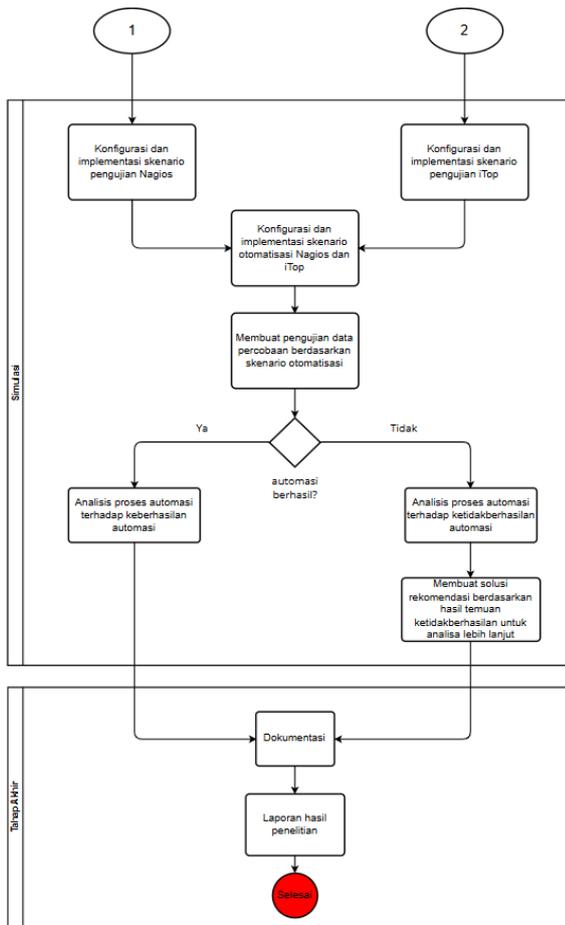
6) Tahap *Management*

Pada tahap ini, perhatian khusus diberikan pada masalah kebijakan, yang mencakup aktivitas, pemeliharaan, dan pengelolaan sistem. Kebijakan perlu dirumuskan untuk mengatur dan memastikan bahwa sistem yang telah dibangun dan berfungsi dengan baik dapat terus beroperasi dalam jangka panjang.

III. METODE

Dalam penelitian ini, berikut merupakan sistematika penyelesaian masalah yang dilakukan:





GAMBAR 4 (SISTEMATIKA PENYELESAIAN MASALAH)

A. Analisis

Pokok fokus penelitian ini adalah untuk memahami kebutuhan dari penggunaan antara NMS, dalam hal ini Nagios dan ITSM yaitu iTop. Tinjauan literatur pada penelitian sebelumnya tentang Nagios, ITSM, dan integrasi memberikan wawasan dan evaluasi pada penelitian ini untuk tujuan dalam pengembangan potensi otomatisasi diantara kedua sistem untuk operasional dan respons yang lebih baik terhadap masalah teknologi informasi. Secara keseluruhan, tujuan penelitian ini adalah untuk membuat pendekatan metode yang efektif dalam hal otomasi Nagios dengan ITSM iTop. Diharapkan otomatisasi ini dapat meningkatkan kemampuan manajemen sistem teknologi informasi, analisis, dan pemantauan sistem jaringan yang terpusat pada satu platform.

B. Desain

Sebelum menguji proses otomasi Nagios dengan iTop, tahap desain penelitian ini menekankan pada perencanaan otomatisasi sistem berdasarkan kebutuhan khusus dengan membuat skenario penggunaan terhadap kedua aplikasi. Skenario ini dirancang untuk dapat menunjukkan alur dan cara kerja suatu fitur pada masing-masing aplikasi untuk menentukan skenario utama pada penelitian ini, yaitu proses otomasi. Tahap desain juga memberikan fondasi untuk implementasi otomasi Nagios dengan iTop, memastikan solusi skenario yang sesuai dengan kebutuhan otomatisasi.

C. Simulasi

Tahap ini dilakukan penerapan skenario Nagios dan iTop dengan melakukan konfigurasi pada kedua aplikasi untuk menerapkan fitur yang akan dijadikan landasan eksperimen otomatisasi, yaitu fitur *monitoring* pada Nagios dan fitur *ticketing* pada iTop. Setelah konfigurasi kedua aplikasi, dilakukan penerapan fitur kedua aplikasi untuk implementasi skenario otomatisasi. Langkah terakhir dilakukan pengujian dari implementasi konfigurasi yang telah dilakukan untuk melihat hasil percobaan untuk dianalisa pada tahap berikutnya.

Pada tahap ini juga dilakukan pemeriksaan terhadap data hasil percobaan skenario otomatisasi. Terdapat pengkondisian yang dimana hasil yang sudah diuji dapat menemui keberhasilan atau tidak. Ini dilakukan sehingga penelitian ini dapat menganalisa (*profiling*) dengan tujuan untuk mengidentifikasi bagian-bagian yang dapat mempengaruhi proses otomasi.

D. Tahap Akhir

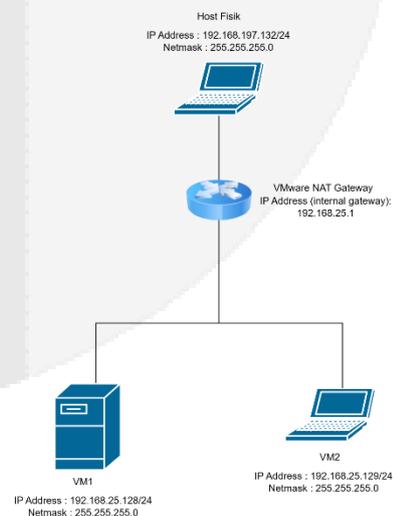
Pada tahap ini, data pengujian dan data analisis yang dilakukan di tahap sebelumnya akan dibuat laporan yang disusun menjadi laporan penelitian. Setelah laporan dibuat, selanjutnya ditentukan apakah otomatisasi Nagios dengan iTop merupakan pengembangan yang dapat diimplementasikan memberikan atau tidak.

IV. HASIL DAN PEMBAHASAN

Bagian ini menjelaskan proses persiapan perangkat serta rancangan skenario terhadap sistem Nagios dan iTop serta skenario otomatisasi hingga hasil yang diperoleh dari penerapannya.

A. Diagram Konektivitas

Diagram Konektivitas disusun sebagai panduan untuk memastikan interaksi antar komponen berjalan dengan baik selama pengujian eksperimen. Diagram ini membantu menetapkan hubungan antar elemen yang akan digunakan dalam proses eksperimen. Gambar IV.1 merupakan diagram konektivitas yang mencerminkan hubungan di antara komponen-komponen sistem.



GAMBAR 5 (TOPOLOGI JARINGAN EKSPERIMEN)

Untuk penjelasannya terdapat pada tabel dibawah ini.

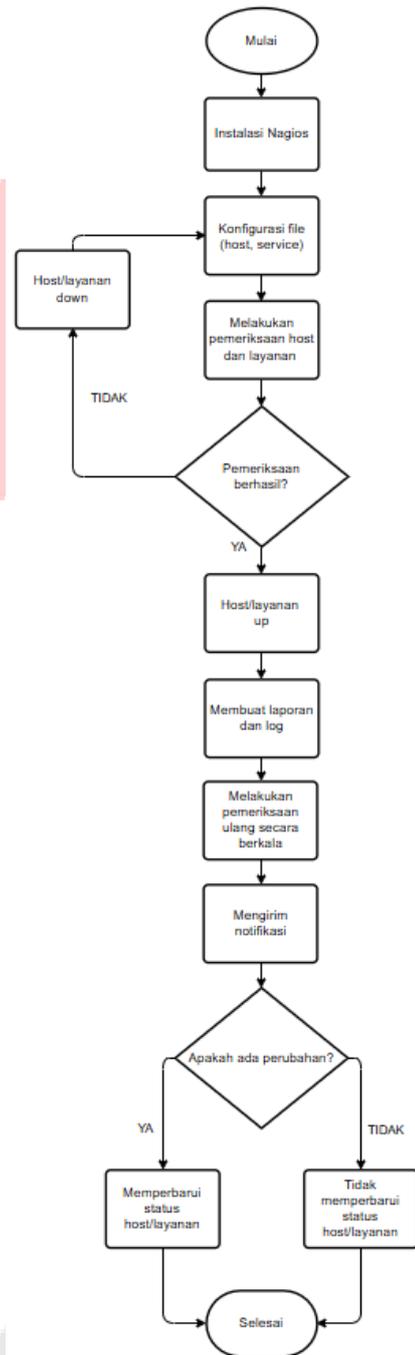
TABEL 1
(IP ADDRESS)

No	Hostname	IP Address	Gateway
1	Host fisik (Menghubungkan ke jaringan eksternal dan menyediakan koneksi NAT untuk VM).	192.168.197.132 /24	192.168.197.1
2	VMware NAT Gateway (Mengelola translasi alamat antara jaringan eksternal dan internal).	192.168.25.1/24	
3	VM 1 (Menjalankan Nagios dan iTop, dengan memantau VM2).	192.168.25.128 /24	192.168.25.1
4	VM 2 (Host yang dimonitor)	192.168.25.129 /24	192.168.25.1

B. Skenario Percobaan

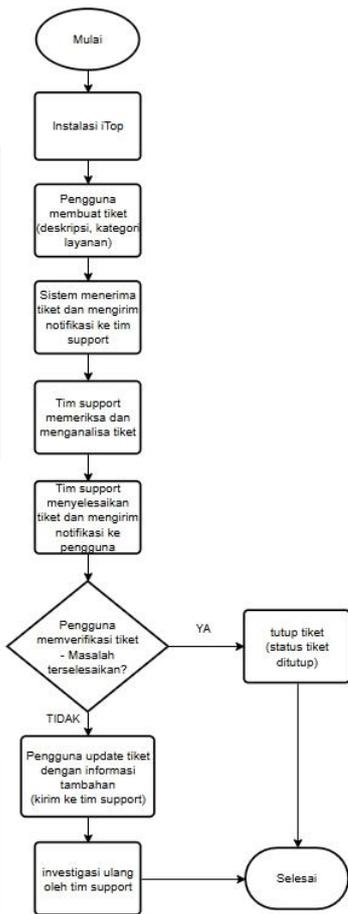
Skenario percobaan adalah serangkaian langkah terstruktur yang digunakan untuk menguji sistem, aplikasi, atau proses guna memastikan kinerja, keandalan, dan fungsionalitasnya sesuai dengan persyaratan yang ditetapkan. Skenario percobaan ini dirancang dalam bentuk *flowchart*, berikut daftar skenario yang dibuat:

1. Skenario Eksperimen Nagios: *Monitoring* Kondisi Sistem Host



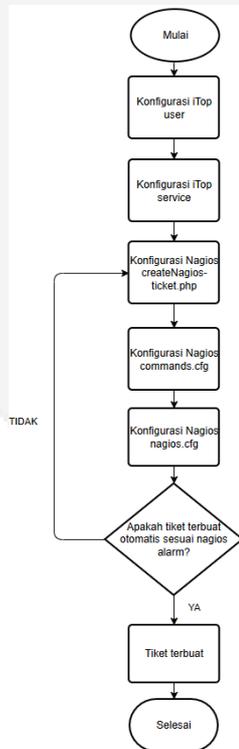
GAMBAR 6
(FLOWCHART NAGIOS)

2. Skenario Eksperimen iTop: Alur Pembuatan Tiket



GAMBAR 7 (FLOWCHART ITOP)

3. Skenario Otomatisasi



GAMBAR 8 (FLOWCHART OTOMATISASI)

C. Implementasi

Berikut tahapan implementasi yang dilakukan berdasarkan pada skenario percobaan sebelumnya:

1. Konfigurasi Nagios

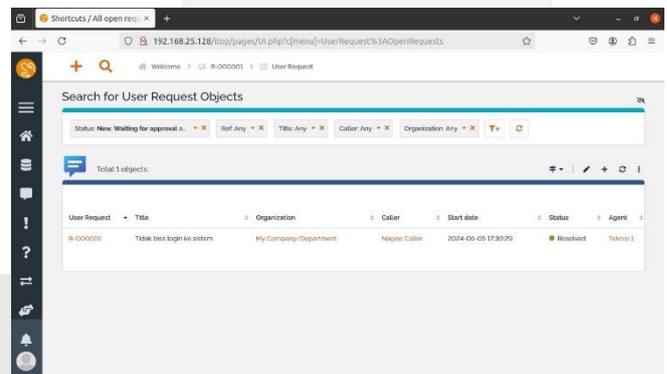
Konfigurasi Nagios dilakukan untuk menguji skenario terhadap proses *monitoring* terhadap *host* dengan menginstalasi Nagios serta mengkonfigurasi *host*, pendefinisian layanan, pemantauan status, dan deteksi masalah. Hal ini dilakukan agar dapat jadi acuan untuk mengkonversi notifikasi menjadi tiket pada iTop untuk proses otomatisasi. Berikut tampilan daftar layanan yang dimonitor oleh Nagios:



GAMBAR 9 (LAYANAN YANG DIMONITOR PADA HOST)

2. Konfigurasi iTop

Konfigurasi iTop dilakukan untuk menguji skenario terhadap sistem pembuatan tiket yang terdapat pada iTop dengan menginstalasi iTop serta mengidentifikasi mekanisme sistem *ticketing* mulai dari dibuatnya tiket sampai tiket tersebut diselesaikan sehingga dapat mengoptimalkan proses konversi notifikasi dari NMS ke tiket yang terstruktur. Berikut tampilan hasil pembuatan tiket yang sudah terselesaikan:



GAMBAR 10 (TIKET YANG TERBUAT)

3. Konfigurasi Otomatisasi

Setelah mengimplementasikan fungsi *monitoring* pada Nagios dan fungsi *ticketing* pada iTop, maka langkah terakhir adalah memadukan kedua skenario aplikasi tersebut untuk membuat skenario otomatisasi dalam penelitian ini. Implementasi ini mencakup verifikasi konfigurasi awal pada kedua sistem yaitu Nagios dan iTop, lalu mengkonfigurasi berbagai *script* dan *file* konfigurasi yang digunakan untuk kebutuhan proses otomatisasi, serta pemantauan kondisi awal dan akhir dari proses yang diotomatisasi. Tahap pertama

yaitu mengkonfigurasi *script* createTicket-nagios.php yang disediakan oleh iTop. Konfigurasi yang dilakukan yaitu menyesuaikan alamat *server* iTop yang dituju serta penyesuaian tiket yang akan di otomasi sesuai kebutuhan eksperimen seperti pengkategorian layanan, kategori tiket, dan kategori lainnya. Berikut tampilan terhadap *script* createTicket-nagios.php yang sudah dikonfigurasi sesuai kebutuhan.

```

GNU nano 4.8 createTicket-nagios.php
// Adjust these settings to fit your iTop installation
// iTop Server location
define('ITOP_SERVER_URL', 'http://192.168.25.128/webservices/itop.wsdl.php');
// iTop credentials for connecting to iTop
define('ITOP_USERNAME', 'admin');
define('ITOP_PASSWORD', 'admin');
// Defaults settings for creating the incident ticket
define('DEFAULT_IMPACT', 2); // 1 = department, 2 = service, 3 = person
define('DEFAULT_URGENCY', 2); // 1 = high, 2 = medium, 3 = low
define('DEFAULT_CALLER_NAME', 'Caller Nagios');
define('DEFAULT_ORGANIZATION_NAME', 'My Company/Department');
define('DEFAULT_SERVICE_NAME', 'Server');
define('DEFAULT_SERVICE_SUBCATEGORY', 'Ubuntu');
define('DEFAULT_WORKGROUP_NAME', 'Department Customer');
// End of instance-specific parameters

define('DEBUG', false); // Set to true to see the actual SOAP envelope output

# Parameters: from Nagios when executing automatic actions on alarms
# =====
# $Service=$argv[2]; # $HOSTNAMES
# $ServiceState=$argv[3]; # $SERVICESTATES
# $ServiceType=$argv[4]; # $SERVICESTATYPES
# $ServiceMessage=$argv[5]; # $LONGSERVICEOUTPUTS

// Main program
require_once(dirname(realpath(__FILE__)).'/ItopsoaTypes.class.inc.php');

echo "===== create_ticket.php called =====\n";

```

GAMBAR 9
(KONFIGURASI createTicket-nagios.php)

Selanjutnya mendefinisikan perintah baru pada *commands.cfg*. *commands.cfg* dalam Nagios adalah *file* konfigurasi yang digunakan untuk mendefinisikan perintah-perintah kustom yang dapat dijalankan oleh Nagios. Pendefinisian perintah baru untuk proses otomasi yaitu *create-iTop-ticket* yang didalam perintah tersebut akan memanggil *script* PHP (*createTicket-nagios.php*) yang akan dieksekusi saat perintah dipicu. Berikut tampilan terhadap pendefinisian *create-iTop-ticket* yang sudah ditambahkan.

```

GNU nano 4.8 objects/commands.cfg
# SAMPLE PERFORMANCE DATA COMMANDS
#
# These are sample performance data commands that can be used to send performance
# data output to two text files (one for hosts, another for services). If you
# place on a single printing performance data out to a file, consider using the
# host_perfdata_file and service_perfdata_file options in the main config file.
# =====
#
# 'process-host-perfdata' command definition
define command{
  command_name process-host-perfdata
  command_line /usr/bin/printf "%b" "$LASTHOSTCHECKS;$HOSTNAMES;$HOSTSTATES;$HOSTSTATETYPES;$HOSTSTATETYPES;$HOSTEXECUTED"
}

# 'process-service-perfdata' command definition
define command{
  command_name process-service-perfdata
  command_line /usr/bin/printf "%b" "$LASTSERVICECHECKS;$HOSTNAMES;$SERVICEDESCS;$SERVICESTATES;$SERVICESTATETYPES;$SERVICEEXECUTED"
}

# Create incident tickets in iTop command definition
define command{
  command_name create-iTop-ticket
  command_line /usr/bin/php /etc/nagios4/createTicket-nagios.php "$HOSTNAMES" "$SERVICEDESCS" "$SERVICESTATES" "$SERVICESTATETYPES"
}

```

GAMBAR 10
(KONFIGURASI Commands.cfg)

Langkah terakhir yaitu mengkonfigurasi *file* nagios.cfg. *File* nagios.cfg adalah *file* konfigurasi utama untuk Nagios. Proses terakhir pada skenario ini yaitu mengaktifkan fungsi *event handlers* di *nagios.cfg* yang untuk menghubungkan perintah *create-iTop-ticket* yang telah dibuat sebelumnya dengan peristiwa (*event*) di Nagios untuk menentukan kapan dan bagaimana perintah tersebut akan dijalankan untuk membuat tiket otomatis di iTop. Berikut tampilan hasil konfigurasi *nagios.cfg*:

```

GNU nano 4.8 nagios.cfg
# passive checks are logged
log_external_commands=1

# PASSIVE CHECKS LOGGING OPTION
# If you don't want Nagios to log passive host and service checks, set
# this value to 0. If passive checks should be logged, set
# this value to 1.
log_passive_checks=1

# GLOBAL HOST AND SERVICE EVENT HANDLERS
# These options allow you to specify a host and service event handler
# command that is to be run for every host or service state change.
# The global event handler is executed immediately prior to the event
# handler that you have optionally specified in each host or
# service definition. The command argument is the short name of a
# command definition that you define in your host configuration file.
# Read the HTML docs for more information.
global_host_event_handler=create-iTop-ticket
global_service_event_handler=create-iTop-ticket

# SERVICE INTER-CHECK DELAY METHOD
# This is the method that Nagios should use when initially

```

GAMBAR 11
(KONFIGURASI Nagios.cfg)

D. Hasil Pengujian Nagios

Berikut ini adalah tabel yang menggambarkan langkah-langkah pengujian dan kriteria keberhasilan dari proses *monitoring* pada Nagios.

TABEL 2
(HASIL PENGUJIAN NAGIOS)

No	Pengujian	Kriteria keberhasilan	Hasil pengujian	Status
1	Melakukan pemeriksaan <i>host</i>	Status <i>host</i> dapat ditampilkan dengan status <i>UP</i>	<i>Host</i> menunjukkan status <i>UP</i>	Berhasil
2	Melakukan pemeriksaan layanan yang dimonitor	Status layanan dapat dipantau dan ditampilkan	Status layanan terlihat	Berhasil
3	Mengirimkan notifikasi terkait kondisi sistem/ <i>service</i> pada <i>host</i> yang terdaftar	Status <i>host</i> dapat diperbarui sesuai kondisi sistem/ <i>service</i> yang dipantau	Status <i>host/service</i> diperbarui	Berhasil
4	Menunggu interval waktu sebelum pemeriksaan ulang	Nagios menunggu sesuai interval yang ditentukan	Interval waktu sesuai	Berhasil
5	Membuat laporan dan <i>log</i>	Laporan dan <i>log</i> dibuat sesuai dengan hasil pemeriksaan	Laporan dan <i>log</i> ada dan terlampir pada <i>nagios.log</i>	Berhasil

Dalam pengujian yang disajikan pada tabel diatas dapat dilihat bahwa masing-masing pengujian berhasil sesuai dengan kriteria keberhasilan yang ditetapkan dalam kebutuhan *monitoring*. Dengan ini, sistem Nagios dalam

memonitor *host* untuk memantau kondisi layanan sistem maupun jaringan melalui notifikasi pada halaman Nagios berfungsi dengan baik sehingga dapat menjadi acuan dalam menentukan alur otomasi dengan aplikasi iTop.

E. Hasil Pengujian iTop

Berikut ini adalah tabel yang menggambarkan langkah-langkah pengujian dan kriteria keberhasilan dari sistem *ticketing* pada iTop.

TABEL 3
(HASIL PENGUJIAN ITOP)

No	Pengujian	Kriteria keberhasilan	Hasil pengujian	Status
1	Tiket dapat dibuat ke berbagai bagian <i>management</i>	Tiket yang dikirim dapat muncul di <i>Problem Management</i> dan <i>Incident Management</i> .	Tiket dapat dibuat dan dikirim ke bagian <i>Problem Management</i> maupun <i>Incident Management</i> .	Berhasil
2	Tim support menerima tiket	Tim support dapat melihat tiket pada <i>dashboard Problem Management</i> maupun <i>Incident Management</i> .	Tiket terlihat	Berhasil

Dalam pengujian ini, masing-masing daftar pengujian pada tabel 3 membuahkan hasil yang sesuai dengan kriteria keberhasilan yang ditetapkan dalam kebutuhan sistem *ticketing*. Tiket berhasil dibuat dan dapat dikirim ke platform *Problem Management* maupun *Incident Management* serta informasi waktu tiket dibuat yang ditampilkan secara *realtime* kepada tim *support*. Dengan ini, sistem iTop dalam menangani proses *ticketing* berfungsi dengan baik sehingga dapat menjadi acuan dalam mengklasifikasikan notifikasi Nagios terhadap *host* dengan status *WARNING* maupun *CRITICAL*.

F. Hasil Pengujian Otomatisasi

Berikut ini adalah tabel yang menggambarkan langkah-langkah pengujian dan kriteria keberhasilan dari proses otomasi antar kedua aplikasi

TABEL 4
(HASIL PENGUJIAN OTOMATISASI)

No	Pengujian	Kriteria keberhasilan	Hasil pengujian	Status
1	Inisialisasi konfigurasi iTop dan Nagios	Nagios dan iTop berhasil	Kedua sistem berjalan	Berhasil

		dikonfigurasi tanpa <i>error</i>		
2	Nagios mendeteksi <i>host down</i>	<i>Host</i> atau layanan yang bermasalah terdeteksi oleh Nagios	Layanan yang dimonitor terdeteksi	Berhasil
3	Nagios mengirim notifikasi ke iTop melalui <i>script</i>	Notifikasi dikirim dan diterima oleh iTop dalam bentuk tiket.	Notifikasi tidak diterima/tiket tidak terbuat otomatis di iTop	Tidak berhasil

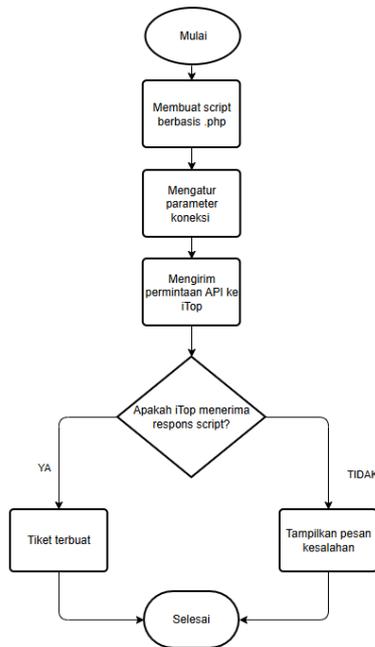
Berdasarkan hasil pengujian yang telah dilakukan terhadap proses otomasi antara Nagios dan iTop pada tabel 4, pengujian inisialisasi iTop dan Nagios berhasil tanpa kesalahan, menunjukkan bahwa kedua sistem dapat berjalan dan beroperasi sesuai dengan konfigurasi awal sehingga siap untuk proses integrasi dan otomasi lebih lanjut. Selain itu, pengujian menunjukkan bahwa Nagios berhasil mendeteksi ketika *host* mengalami gangguan (*down*), yang menunjukkan bahwa sistem *monitoring* Nagios berfungsi dengan baik dalam mendeteksi masalah pada *host* yang dipantau. Namun, pengujian untuk mengirim notifikasi dari Nagios ke iTop melalui *script* tidak berhasil.

G. Rekomendasi

Berdasarkan hasil temuan diatas, diusulkan strategi solusi dengan membuat rancangan *script* khusus untuk aplikasi Nagios untuk menguji konektivitas dengan layanan iTop berdasarkan struktur kode yang terdapat pada *createTicket-nagios.php*. Rancangan *Script* ini dibuat agar dapat diimplementasikan untuk dieksekusi secara manual melalui *terminal* Linux untuk memastikan bahwa API iTop dapat diakses oleh Nagios dan menguji apakah tiket dapat dibuat. Langkah-langkah dalam rancangan ini meliputi:

1. Flowchart

Flowchart untuk menyediakan gambaran visual tentang langkah-langkah yang terlibat dalam cara kerja *script* dan bagaimana *script* berinteraksi dengan iTop. *Flowchart* ditampilkan pada gambar berikut:

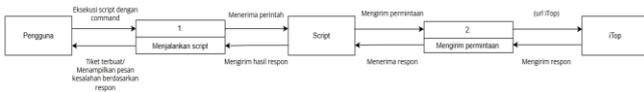


GAMBAR 12

(FLOWCHART UJI KONEKTIVITAS NAGIOS DENGAN ITOP)

2. Data Flow Diagram

Data Flow Diagram untuk menggambarkan bagaimana data mengalir melalui sistem *script* Nagios dan bagaimana *input* dan *output* diproses oleh *script*. *Data Flow Diagram* ditampilkan pada gambar berikut:

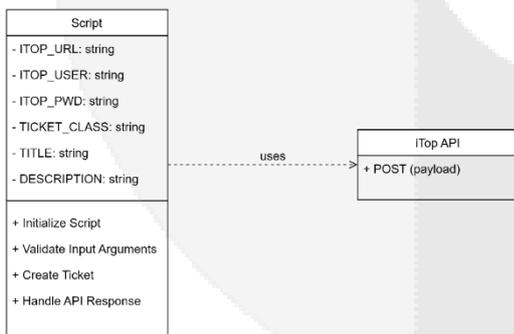


GAMBAR 13

(DFD UJI KONEKTIVITAS NAGIOS DENGAN ITOP)

3. Class Diagram

Class Diagram untuk menyediakan model struktural dari *script* yang mencakup kelas, atribut, dan metode yang terlibat dalam pembuatan *script*. *Class Diagram* ditunjukkan pada gambar berikut:



GAMBAR 14

(CLASS DIAGRAM SCRIPT UJI KONEKTIVITAS)

V. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan dalam “Implementasi dan Profiling Software Opensource iTop

Dengan Software Opensource Nagios Berdasarkan Aspek Otomatisasi”, didapat kesimpulan bahwa:

1. Hasil pengujian terhadap Nagios dalam proses *monitoring* yang terdiri dari lima pengujian, yaitu pemeriksaan *host*, pemeriksaan layanan yang dimonitor, pengiriman notifikasi terkait kondisi sistem/*service* pada *host* yang terdaftar, pemeriksaan berulang dengan waktu interval yang sudah ditetapkan, serta pembuatan laporan dan *log* mendapatkan status berhasil. Nagios dapat mendeteksi potensi masalah pada *host* dan mengirimkan notifikasi sesuai dengan temuan yang sudah diklasifikasikan dalam konfigurasi Nagios. Alur kerja pada Nagios melibatkan beberapa langkah kunci; konfigurasi dengan menambahkan *host*, mendefinisikan layanan yang ingin dimonitor, pemantauan berkelanjutan terhadap status *host* dan layanan, deteksi otomatis ketika ada masalah atau potensi masalah. Proses pengiriman notifikasi pada Nagios dalam hal *monitoring* menjadi acuan dalam proses otomatis mengkonversi notifikasi yang diinisiasi Nagios ke iTop dalam bentuk tiket.
2. Hasil pengujian terhadap iTop dalam proses pembuatan tiket yang terdiri dari dua pengujian, yaitu tiket dapat dibuat ke pendekatan *Problem Management* maupun *Incident Management* serta tim *support* dapat menerima tiket mendapatkan status berhasil. Dalam identifikasi alur kerja *ticketing* pada iTop, ditemukan bahwa sistem ini memiliki mekanisme yang terstruktur untuk membuat tiket dalam *Incident Management* dan *Problem Management*. Berdasarkan klasifikasi yang diberikan, tiket dibuat lalu dikategorikan sesuai kebutuhan. Tiket kemudian ditugaskan kepada tim *support* untuk investigasi dan penyelesaian. Proses ini mencakup pembaruan status tiket, pencatatan tindakan yang diambil, dan pemberitahuan kepada pengguna terkait perkembangan penyelesaian masalah. Hal ini memastikan bahwa setiap laporan pada tiket diatasi dengan metode yang sistematis dan efisien. Proses pembuatan tiket digunakan sebagai konversi notifikasi Nagios yang berisi ketidaknormalan atau temuan masalah pada *host* untuk dikategorikan sesuai klasifikasi Nagios ke pendekatan *Problem Management* maupun *Incident Management*.
3. Hasil pengujian terhadap proses otomatisasi yang terdiri dari tiga pengujian, yaitu inisialisasi konfigurasi iTop dan Nagios serta Nagios dapat mendeteksi *host* yang *down* mendapatkan status berhasil. Meskipun hasil pengujian menunjukkan keberhasilan dalam inisialisasi kedua sistem dan deteksi masalah oleh Nagios, proses pengiriman notifikasi dari Nagios ke iTop melalui *script* tidak berhasil. *Script* `createTicket_nagios.php` telah dikonfigurasi dan disesuaikan dengan benar, pendefinisian `create-iTop-ticket` pada `commands.cfg` sudah sesuai, dan `event handler` pada `nagios.cfg` telah diaktifkan. Meskipun `event handler create-iTop-ticket` telah terbaca oleh Nagios, iTop tetap tidak membuat tiket otomatis berdasarkan notifikasi Nagios, dan tidak ada *log* yang menunjukkan keberhasilan proses otomatisasi ini. Hal ini menunjukkan perlunya investigasi lebih lanjut untuk memastikan *script* dan konfigurasi dijalankan dengan benar serta memastikan tidak ada masalah komunikasi atau izin yang menghalangi eksekusi *script* tersebut. Untuk itu dibuat solusi rekomendasi untuk menguji konektivitas antara *script* Nagios dengan API iTop

dengan membuat rancangan *script* berpedoman terhadap struktur kode yang ada pada *script* createTicket-nagios.php. Rancangan yang dibuat berupa alur kerja pada *flowchart*, proses aliran data dan komunikasi pada DFD, dan model data *script* pada Class Diagram.

REFERENSI

- [1] A. T. Atmaja, D. Santoso, and P. Ninghardjanti, "Penerapan Sistem Otomatisasi Administrasi Untuk Meningkatkan Efektivitas dan Efisiensi Kerja di Bidang Pendapatan Dinas Perdagangan Kota Surakarta," *Jurnal Informasi dan Komunikasi*, vol. 2, Feb. 2018.
- [2] J. Iden and T. R. Eikebrokk, "Implementing IT Service Management: A systematic literature review," *Int J Inf Manage*, vol. 33, no. 3, pp. 512–523, Jun. 2013, doi: 10.1016/j.ijinfomgt.2013.01.004.
- [3] Z. Apriesta, "Pemeringkatan Open Source Software ITSM Bagi Small-Medium Enterprises (SMEs) Menggunakan Metode Fuzzy AHP Berdasarkan Kriteria ITIL V3 Pada Proses Service Operation," Universitas Islam Negeri Syarif Hidayatullah, Jakarta, 2018.
- [4] Salamun and Sukri, "Analisa Pemanfaatan dan Peran Software Open-source bagi Mahasiswa Universitas Abdurrab," *Jurnal Buana Informatika*, vol. 12, pp. 49–57, Apr. 2021.
- [5] ISA, "What is Automation?"
- [6] Sedarmayanti, *Dasar-dasar Pengetahuan tentang Manajemen Perkantoran: Suatu Pengantar*. Bandung: Mandar Maju, 2009.
- [7] D. O. Saputra and H. Saptono, "Implementasi Network Monitoring System Terintegrasi Dengan Ticketing System Menggunakan Nagios dan osTicket," *Jurnal Informatika Terpadu*, vol. 5, Mar. 2019.
- [8] W. C. Prasetia, "Implementasi Sistem Monitoring Jaringan Menggunakan Nagios Dengan SMS Alert Menggunakan Ozeki NG SMS Gateway," Universitas Telkom, Bandung, 2012.
- [9] W. Kocjan, "Introducing Nagios," in *Learning Nagios 4*, March 2014., Birmingham: Packt Publishing Ltd., 2014, p. 8.
- [10] A. J. Keel, M. A. Orr, R. R. Hernandez, E. A. Patrocinio, and J. Bouchard, "From a technology-oriented to a service-oriented approach to IT management &," 2007.
- [11] Budiyo, E. Nugroho, and W. W. Winarno, "Implementasi ITIL V3 Framework pada Perancangan Aplikasi Service Desk Management Berorientasi User," *JNETI*, vol. 1, Aug. 2012.
- [12] AXELOS, "The ITIL service value system," in *ITIL Foundation*, ITIL 4 Edition., Norwich: The Stationery Office, 2019.
- [13] CA Technologies, *CA Service Desk Manager*. New York, 2007.
- [14] A. Rahmawati and A. F. Wijaya, "Analisis Risiko Teknologi Informasi Menggunakan ISO 31000 Pada Aplikasi iTop," *Jurnal SITECH*, vol. 2, Jun. 2019.
- [15] S. Kosasi, "Penerapan Network Development Life Cycle Untuk Pengembangan Teknologi Thin Client Pada Pendidikan KSM Pontianak," *JIKE*, vol. 1, pp. 125–141, May 2011.