

**PERBANDINGAN METODE
IDENTIFIKASI *HOSTNAME* PADA PROTOKOL HTTPS
UNTUK PENERAPAN *TRANSPARENT WEB PROXY*
DI POLITEKNIK NEGERI BANDUNG**

*Hostname Identification on HTTPS Protocol Method Comparison for
Transparent Web Proxy Implementation in Politeknik Negeri Bandung*

TUGAS AKHIR

Laporan ini disusun untuk memenuhi salah satu syarat menyelesaikan
pendidikan Diploma Empat Program Studi Teknik Informatika
di Jurusan Teknik Komputer dan Informatika

Oleh:

MUHAMMAD SAIFUL ISLAM

NIM: 141524020



**POLITEKNIK NEGERI BANDUNG
2018**

**PERBANDINGAN METODE
IDENTIFIKASI *HOSTNAME* PADA PROTOKOL HTTPS
UNTUK PENERAPAN *TRANSPARENT WEB PROXY*
DI POLITEKNIK NEGERI BANDUNG**

Oleh:

MUHAMMAD SAIFUL ISLAM

NIM: 141524020

Menyetujui,

Bandung, 9 Agustus 2018

Pembimbing I,

Yudi Widhiyasana, S.Si., M.T.

NIP 197407182001121002

Pembimbing II,

Setiadi Rachmat, B.Eng., M.Eng.

NIP 196904041998031001

Ketua Jurusan Teknik Komputer dan Informatika,

Drs. Eddy Bambang Soewono, M.Kom.

NIP 196101141992021001



**PERBANDINGAN METODE
IDENTIFIKASI *HOSTNAME* PADA PROTOKOL HTTPS
UNTUK PENERAPAN *TRANSPARENT WEB PROXY*
DI POLITEKNIK NEGERI BANDUNG**

Oleh:

MUHAMMAD SAIFUL ISLAM

NIM: 141524020

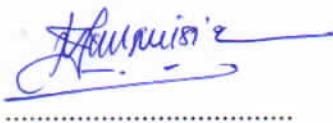
Tugas akhir ini telah disidangkan pada tanggal 25 Juli 2018
sesuai dengan ketentuan.

Tim Penguji:

Ketua : Joe Lian Min, B.Eng., M.Eng.
NIP 196610181995121001



Anggota : Transmissia Semiawan, BSCS., M.IT., Ph.D.
NIP 196111091993032001



PERNYATAAN PENULIS

Dengan ini menyatakan bahwa laporan tugas akhir dengan judul Perbandingan Metode Identifikasi *Hostname* pada Protokol HTTPS untuk Penerapan *Transparent Web Proxy* di Politeknik Negeri Bandung adalah karya ilmiah yang bebas dari unsur tindakan plagiarisme, dan sesuai dengan ketentuan tata tulis yang berlaku.

Apabila di kemudian hari ditemukan adanya unsur plagiarisme, maka hasil penilaian dari tugas akhir ini dicabut dan bersedia menerima sanksi sesuai dengan ketentuan yang berlaku.

Demikian pernyataan ini dibuat dengan sesungguhnya dalam keadaan sadar sepenuhnya.

Bandung, // Agustus 2018



Muhammad Saiful Islam

NIM 141524020

Feedback Studio - Google Chrome

Secure | https://ev.turnitin.com/app/carta/en_us/?u=1063722671&lang=en_us&s=1&o=989123904

turnitin Perbandingan Metode Identifikasi Hostname pada Protokol HTTPS. < 2 of 9 > ?

Match Overview

5%

Rank	Source	Percentage
1	www.rhymesoft.in Internet Source	<1%
2	id.123dok.com Internet Source	<1%
3	sistemaoperacionalser... Internet Source	<1%
4	digilib.its.ac.id Internet Source	<1%
5	repository.usu.ac.id Internet Source	<1%
6	xss.cx Internet Source	<1%
7	slideplayer.info Internet Source	<1%
8	text-id.123dok.com Internet Source	<1%

PERBANDINGAN METODE
DENTIFIKASI *HOSTNAME* PADA PROTOKOL HTTPS
UNTUK PENERAPAN *TRANSPARENT WEB PROXY*
DI POLITEKNIK NEGERI BANDUNG

Oleh:
MUHAMMAD SAIFUL ISLAM
NIM: 141524020

Menyetujui,

Page: 2 of 128 Word Count: 28274 Text-only Report | High Resolution On



Nama : Muhammad Saiful Islam

NIM : 141524020

Tempat, Tanggal Lahir : Probolinggo, 24 Agustus 1996

SD Lulus Tahun : 2008 dari SDN Sukarasa 5, Kota Bandung

SLTP Lulus Tahun : 2011 dari SMPN 2, Kota Bandung

SLTA Lulus Tahun : 2014 dari SMAN 2, Kota Bandung

Prestasi yang pernah dicapai:

- *Best Position Paper in Healthcare Accessibility* dan *Runner Up, Committee Pitch Competition*, S. Rajaratnam Endowment – Youth Model ASEAN Conference 2017
- Finalis Mahasiswa Berprestasi Nasional 2017
- Mahasiswa Berprestasi Utama Politeknik Negeri Bandung 2017
- Penghargaan dari Pembantu Direktur Bidang Kemahasiswaan Polban sebagai inisiator kegiatan “*Friday Open Mic*,” kegiatan untuk meningkatkan minat dan kemampuan *public speaking* mahasiswa
- Penghargaan dari Pembantu Direktur Bidang Kemahasiswaan Polban atas pengabdian sebagai pengelola infrastruktur IT di JTK Polban
- Juara 1 Agricode IPB Programming Competition 2015
- Mahasiswa Berprestasi Akademik 2014/2015, 2015/2016, 2016/2017
- Tim *competitive programming* terbaik di tingkat diploma pada perlombaan ACM-ICPC Multi-Provincial di Indonesia sejak tahun 2015 s.d. 2018 dengan nama tim “SunsetInCiwaruga”, “Rumah Besar Cipali”, dan “Kembali untuk Syafira”

ABSTRAK

Aplikasi masa kini mulai banyak membutuhkan akses langsung ke internet. Penggunaan aplikasi tersebut tidak dapat digunakan di Politeknik Negeri Bandung karena penerapan *explicit web proxy* untuk melakukan autentikasi dan otorisasi pengguna serta pencatatan akses web. Dengan *transparent web proxy*, masalah tersebut dapat diselesaikan dan sesuai dengan rekomendasi dari para praktisi. Namun, peningkatan penggunaan protokol HTTPS di seluruh dunia menyebabkan identifikasi *hostname* pada *transparent web proxy* lebih sulit untuk dilakukan.

Ada empat metode untuk mengidentifikasi *hostname* pada *transparent web proxy*, yaitu metode *reverse lookup* berdasarkan entri DNS, *reverse lookup* berdasarkan rekaman *query* pengguna, metode *server name indication* (SNI), dan metode berdasarkan atribut *common name* (CN) pada sertifikat TLS. Hasil identifikasi dari keempat metode tersebut dibandingkan terhadap hasil identifikasi dari *explicit web proxy*. Hasilnya, metode SNI menghasilkan nilai F_1 score sebesar 100% jika klien mendukung penggunaan SNI. Alternatif dari metode ini adalah metode *reverse lookup* berdasarkan rekaman *query* pengguna dengan nilai F_1 score sebesar 75,82%.

Arsitektur jaringan di Politeknik Negeri Bandung perlu dimodifikasi untuk menerapkan *transparent web proxy* dengan metode SNI. Tugas akhir ini merumuskan model autentikasi pengguna, topologi, dan konfigurasi jaringan yang baru. Model tersebut diimplementasi dalam skala lab dan digunakan untuk menguji tiga belas aplikasi yang digunakan dalam pembelajaran di Politeknik Negeri Bandung. Hasilnya, seluruh aplikasi yang diujikan dapat diidentifikasi dengan baik, dan model yang dihasilkan dapat diterapkan di Politeknik Negeri Bandung.

Kata kunci: arsitektur jaringan komputer, HTTPS, identifikasi layanan, *web proxy*.

ABSTRACT

Politeknik Negeri Bandung uses explicit web proxy to authenticate, authorize, and log web access. That enforcement became a problem because many applications recently require direct connection to the internet which is not possible in a network with an explicit web proxy. Practitioners made a recommendation to use transparent web proxy instead of explicit web proxy, but increasing usage on HTTPS protocol worldwide made it difficult to identify hostname being accessed.

Four methods to identify hostname in transparent web proxy: reverse lookup based on users' query, reverse lookup based on DNS entries, server name indication (SNI), and lookup on common name attribute on TLS certificates are compared with identification results from explicit web proxy being used in Politeknik Negeri Bandung. The result shows that SNI method offers 100% of F₁ score if the clients support SNI. The alternative of SNI method is reverse lookup based on users' query with F₁ score of 75,82%.

Network architecture in Politeknik Negeri Bandung then needs to be modified in order to enforce transparent web proxy with SNI hostname identification method. New authentication flow, network topology and configuration is modeled in this final project. The model is implemented in lab scale and used to test thirteen applications used in academic classes. It shows that all hostnames accessed by those applications can be correctly identified and the model can be implemented in Politeknik Negeri Bandung.

Keywords: computer network architecture, HTTPS, service identification, web proxy.

KATA PENGANTAR

Penulis memuji dan bersyukur kepada Allah *subhanahu wa ta'ala* atas rahmat, nikmat, dan karunia-Nya sehingga penulis dapat menyelesaikan laporan tugas akhir dengan judul Perbandingan Metode Identifikasi *Hostname* pada Protokol HTTPS untuk Penerapan *Transparent Web Proxy* di Politeknik Negeri Bandung ini. Selawat beserta salam semoga senantiasa tercurah kepada Nabi Muhammad *shallallahu 'alaihi wa sallam* beserta keluarganya, sahabatnya, dan umatnya hingga akhir zaman.

Laporan ini merupakan uraian tertulis mengenai pekerjaan tugas akhir yang penulis lakukan sebagai bagian dari pendidikan pada semester kedelapan tahun akademik 2017/2018.

Pada kesempatan ini penulis berterima kasih kepada:

1. kedua orang tua, bapak Nur'ain Syaiful Toharoh, A.Md. dan ibu Erna Wahyu Mustika, S.Pd. serta adik penulis, Muhammad Faris Al Hafidh yang telah senantiasa memberikan dukungan, doa, dan restu;
2. bapak Yudi Widhiyasana, S.Si., M.T. dan bapak Setiadi Rachmat, B.Eng., M.Eng. sebagai pembimbing atas bimbingan dan motivasi dalam penelitian dan penulisan laporan tugas akhir ini;
3. bapak Joe Lian Min, B.Eng., M.Eng. dan ibu Transmissia Semiawan, BSCS., M.IT., Ph.D. sebagai penguji yang juga turut memberikan masukan dalam tugas akhir ini;
4. ibu Ani Rahmani, S.Si., M.T. dan ibu Rahil Jumiyani, S.ST., M.Sc. atas koreksinya dalam hal tata tulis laporan serta abstrak dan judul bahasa Inggris dari tugas akhir ini;
5. bapak Drs. Mulyadi Yuswandono, Dipl.Ing., M.T. sebagai Pembantu Direktur IV Bidang Perencanaan dan Pengembangan bersama bapak Billy Muhammad Iqbal, A.Md. dan bapak Megi Donni Daradjat, S.T., M.Kom. dari Sub Bagian Pengembangan Sistem Informasi (PSI) atas dukungan dan bantuan teknis yang diberikan pada tugas akhir ini;

6. bapak Urip Teguh Setijohatmo, BSCS., M.Kom. sebagai wali kelas D-IV Teknik Informatika 2014 dan bersama bapak Suprihanto, BSEE., M.Sc. sebagai koordinator mata kuliah tugas akhir;
7. ibu Santi Sundari, S.Si., M.T. sebagai ketua program studi D-IV Teknik Informatika dan bapak Drs. Eddy Bambang Soewono, M.Kom. sebagai ketua Jurusan Teknik Komputer dan Informatika;
8. rekan-rekan *network and system administrator* JTK Polban: bapak Ghifari Munawar, S.Kom., M.T., Sukma Setyaji, Kiki Pratiwi, Ilham Gibran Achmad Mudzakir, Ali Piqri Sopandi, dan Refdinal Tubagus atas diskusi dan bantuan teknis dalam pengerjaan tugas akhir ini;
9. Fadhlwan Ridhwanallah, Muhammad Imam Fauzan Putra Perdana Nasution, Sukma Setyaji, dan Novia Sukmasari Putri, empat orang rekan penulis yang memberikan semangat, dukungan, dan motivasi sepanjang pengerjaan tugas akhir ini;
10. rekan-rekan D-IV Teknik Informatika 2014 atas kebersamaannya dalam semester terakhir tahun akademik 2017/2018 ini, terutama di mata kuliah tugas akhir;
11. Via Vallen dan Nella Kharisma atas karya-karyanya yang telah membantu menghadirkan lingkungan pengerjaan tugas akhir yang kondusif; serta
12. rekan-rekan penulis lainnya yang tak dapat dituliskan satu persatu atas seluruh bantuan, diskusi, dan motivasi yang diberikan selama pengerjaan tugas akhir ini.

Penulis berharap laporan ini bermanfaat bagi kemajuan bersama, khususnya bagi Politeknik Negeri Bandung. Penulis juga berharap seluruh pengalaman yang penulis dapatkan pada pelaksanaan tugas akhir ini menjadi jalan penulis untuk mengamalkannya dalam kebaikan, serta menjadi amal ibadah bagi pihak-pihak yang sudah mendukungnya.

Penulis memohon maaf atas segala kekurangan dalam laporan ini.

Bandung, Agustus 2018

Muhammad Saiful Islam

DAFTAR ISI

ABSTRAK	i
ABSTRACT	ii
KATA PENGANTAR	iii
DAFTAR ISI.....	v
DAFTAR LAMPIRAN	viii
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	xi
DAFTAR RUMUS.....	xii
BAB I PENDAHULUAN	1
I.1 Latar Belakang	1
I.2 Rumusan Masalah	4
I.3 <i>Research Question</i> dan Hipotesis.....	4
I.4 Tujuan	5
I.5 Luaran	5
I.6 Manfaat	6
I.7 Ruang Lingkup.....	6
I.8 Sistematika Penulisan.....	7
BAB II TINJAUAN PUSTAKA.....	9
II.1 Dasar Teori.....	9
II.1.1 Model Referensi OSI dan TCP/IP	9
II.1.2 <i>Hypertext Transfer Protocol (HTTP)</i>	11
II.1.3 <i>HTTP over TLS (HTTPS)</i>	12
II.1.4 <i>Transport Layer Security (TLS)</i>	13
II.1.5 <i>Server Name Indication (SNI)</i>	15
II.1.6 <i>Domain Name System (DNS)</i>	16
II.1.7 <i>Web Proxy</i>	18
II.1.8 <i>Authentication, Authorization, and Accounting (AAA)</i>	22
II.1.9 <i>Confusion Matrix</i>	25

II.1.10	Koefisien Korelasi Pearson.....	26
II.2	Karya Ilmiah Sejenis Sebelumnya	27
	BAB III METODOLOGI PENELITIAN.....	33
III.1	Jenis Penelitian.....	33
III.2	Subjek Penelitian.....	33
III.3	Objek Penelitian	34
III.4	Variabel Penelitian	34
III.5	Data Penelitian	36
III.6	Tahapan Penelitian	37
III.6.1	Studi Pustaka	37
III.6.2	Analisis <i>Problem Domain</i>	39
III.6.3	Pengumpulan Data Log <i>Web Proxy</i>	40
III.6.4	Pengumpulan Aplikasi Pengguna	41
III.6.5	Pembuatan Aplikasi Eksperimen.....	42
III.6.6	Penyiapan Data	43
III.6.7	Eksperimen	46
III.6.8	Analisis Hasil Eksperimen.....	49
III.6.9	Perancangan Jaringan dengan <i>Transparent Web Proxy</i>	49
III.6.10	Pembuatan Implementasi Skala Lab.....	50
III.6.11	Percobaan Aplikasi Pengguna dan Evaluasi Hasil	53
	BAB IV PENENTUAN METODE IDENTIFIKASI <i>HOSTNAME</i>.....	54
IV.1	Analisis <i>Problem Domain</i>	54
IV.1.1	Permasalahan Identifikasi <i>Hostname</i> Akses HTTPS.....	54
IV.1.2	Analisis Kebutuhan Aplikasi Eksperimen	60
IV.2	Pembuatan Aplikasi Eksperimen	62
IV.3	Penyiapan Data.....	63
IV.4	Eksperimen.....	64
IV.5	Analisis Hasil Eksperimen	64
IV.5.1	Pengaruh dari Dukungan SNI pada Klien	64
IV.5.2	Pengaruh dari Jumlah Alamat IP Hasil <i>Lookup</i>	67
IV.5.3	Pengaruh dari Jumlah <i>Mutual Hostname</i>	72

IV.5.4	Pengaruh dari Jumlah Alamat IP Pengakses <i>Hostname</i>	75
IV.5.5	Pengaruh dari Jumlah Kunjungan per <i>Hostname</i>	77
IV.5.6	Analisis Keseluruhan	79
BAB V PENERAPAN <i>TRANSPARENT WEB PROXY</i>		83
V.1	Analisis <i>Problem Domain</i>	83
V.1.1	Analisis Permasalahan Saat Ini.....	83
V.1.2	Analisis Topologi Jaringan Kampus.....	85
V.1.3	Analisis Pengelolaan Akses Web ke Internet	88
V.1.4	Evaluasi Hasil Analisis	91
V.2	Perancangan Jaringan dengan <i>Transparent Web Proxy</i>	93
V.2.1	Alur Autentikasi Hingga Akses Web ke Internet	93
V.2.2	Topologi Jaringan yang Dibutuhkan	95
V.2.3	Konfigurasi yang Dibutuhkan.....	97
V.3	Pembuatan Implementasi Skala Lab	98
V.3.1	Implementasi di Lingkungan GNS3	99
V.3.2	Implementasi di Lingkungan <i>Deployment</i>	101
V.4	Percobaan Aplikasi Pengguna.....	103
V.5	Evaluasi Hasil Percobaan.....	103
BAB VI KESIMPULAN DAN SARAN		104
VI.1	Kesimpulan	104
VI.2	Saran.....	106
DAFTAR PUSTAKA		108

DAFTAR LAMPIRAN

- Lampiran 1. Laporan Analisis Kebutuhan Layanan IT JTK Polban
- Lampiran 2. Standar Pengelolaan Jaringan SI Politeknik Negeri Bandung
- Lampiran 3. Dokumen Formal Perizinan Permintaan Data dan Fasilitas Penelitian
- Lampiran 4. *Source Code* Program Eksperimen
- Lampiran 5. Implementasi Konfigurasi Skala Lab pada Lingkungan GNS3
- Lampiran 6. Implementasi Konfigurasi Skala Lab pada Lingkungan *Deployment*

DAFTAR GAMBAR

Gambar II.1.	Model referensi OSI.....	10
Gambar II.2.	Model referensi TCP/IP	10
Gambar II.3.	Pertukaran pesan <i>request</i> dan <i>response</i> pada protokol HTTP	12
Gambar II.4.	Posisi protokol TLS dalam protokol HTTPS pada <i>layer</i> TCP/IP	13
Gambar II.5.	Urutan pertukaran pesan pada <i>handshake</i> TLS.....	14
Gambar II.6.	SNI pada <i>handshake</i> TLS.....	15
Gambar II.7.	<i>Field SNI</i> dilihat menggunakan Wireshark.....	16
Gambar II.8.	Keluaran <i>console</i> pada <i>lookup www.google.com</i>	17
Gambar II.9.	Gambaran umum sebuah <i>proxy</i>	18
Gambar II.10.	Penerapan <i>transparent web proxy</i> di ISP.....	21
Gambar II.11.	Interaksi komponen AAA ketika pengguna mengakses jaringan	23
Gambar II.12.	Alur komunikasi umum di RADIUS.....	25
Gambar II.13.	<i>Confusion matrix</i>	26
Gambar II.14.	Struktur data pada metode kedua	28
Gambar III.1.	Hubungan antar variabel penelitian	36
Gambar III.2.	Tahapan penelitian	38
Gambar III.3.	Contoh log dari <i>explicit web proxy</i>	41
Gambar III.4.	Alur penyiapan data hingga eksperimen	43
Gambar IV.1.	Visualisasi <i>reverse lookup</i> berdasarkan rekaman <i>query</i>	56
Gambar IV.2.	<i>Sequence diagram</i> eksperimen pada aplikasi utama	62
Gambar IV.3.	Bot Telegram untuk notifikasi eksperimen	64
Gambar IV.4.	Jumlah alamat IP hasil <i>lookup</i> dan jumlah <i>hostname</i> -nya	68
Gambar IV.5.	Nilai <i>F₁ score</i> berdasarkan jumlah alamat IP hasil <i>lookup</i>	69
Gambar IV.6.	Distribusi <i>hostname</i> berdasarkan jumlah <i>mutual hostname</i>	73
Gambar IV.7.	Distribusi <i>hostname</i> dengan <i>outliers</i> yang dibuang	73
Gambar IV.8.	Nilai <i>F₁ score</i> berdasarkan jumlah <i>mutual hostname</i>	74
Gambar IV.9.	Nilai <i>F₁ score</i> berdasarkan jumlah alamat IP pengakses	76
Gambar IV.10.	Nilai <i>F₁ score</i> berdasarkan jumlah kunjungan per <i>hostname</i>	78
Gambar V.1.	Topologi <i>existing</i> akses web intranet kampus.....	87

Gambar V.2.	Contoh <i>report</i> yang dikeluarkan LightSquid	91
Gambar V.3.	Log pada EWP ₁ yang berhenti dianalisis	93
Gambar V.4.	Alur autentikasi klien hingga akses web ke internet	94
Gambar V.5.	Topologi usulan akses web intranet kampus.....	96
Gambar V.6.	Implementasi di lingkungan GNS3	100
Gambar V.7.	Perangkat implementasi skala lab yang dipasang di JTK	102

DAFTAR TABEL

Tabel II.1.	Jenis entri DNS	17
Tabel II.2.	Perbandingan <i>explicit</i> dengan <i>transparent web proxy</i>	21
Tabel III.1.	Subjek studi pustaka	37
Tabel III.2.	Data yang akan dikumpulkan	39
Tabel III.3.	Daftar aplikasi pengguna yang membutuhkan akses internet	42
Tabel III.4.	Konfigurasi eksperimen.....	48
Tabel III.5.	<i>Template</i> hasil eksperimen	51
Tabel IV.1.	Nilai F_1 score berdasarkan dukungan SNI pada klien.....	65
Tabel IV.2.	Perbandingan atribut CN dengan dan tanpa dukungan SNI	66
Tabel IV.3.	Beberapa <i>hostname</i> dengan alamat IP lebih dari delapan buah.....	69
Tabel IV.4.	Perincian nilai F_1 score ketika jumlah alamat IP hasil <i>lookup</i> = 6.70	
Tabel IV.5.	Korelasi jumlah alamat IP hasil <i>lookup</i> dengan nilai F_1 score.....	71
Tabel IV.6.	Korelasi jumlah <i>mutual hostname</i> dengan nilai F_1 score	75
Tabel IV.7.	Korelasi jumlah alamat IP pengakses dengan nilai F_1 score	77
Tabel IV.8.	Beberapa <i>hostname</i> populer pada log.....	78
Tabel IV.9.	Nilai F_1 score minimal, rata-rata, dan maksimal berdasarkan jumlah kunjungan	79
Tabel IV.10.	Korelasi jumlah kunjungan per <i>hostname</i> dengan nilai F_1 score ..	79
Tabel IV.11.	Pengaruh masing-masing variabel bebas terhadap nilai F_1 score ..	82
Tabel V.1.	Sebagian akses pada <i>explicit web proxy</i> tanpa autentikasi	85
Tabel V.2.	Segmen intranet kampus beserta <i>web proxy</i> -nya.....	86
Tabel V.3.	<i>Web proxy</i> kampus	86
Tabel V.4.	Hasil percobaan pada implementasi skala lab	103

DAFTAR RUMUS

Rumus II.1. <i>Recall</i>	25
Rumus II.2. <i>Precision</i>	26
Rumus II.3. <i>F₁ score</i>	26
Rumus II.4. Koefisien penentuan	27
Rumus II.5. Koefisien korelasi	27

BAB I

PENDAHULUAN

Bab ini menjelaskan latar belakang, rumusan masalah yang dihadapi, *research question* dan hipotesis, tujuan, luaran, manfaat, batasan ruang lingkup, serta sistematika penulisan laporan tugas akhir ini.

I.1 Latar Belakang

Kebutuhan akses internet semakin penting dalam kehidupan sehari-hari, termasuk dalam proses belajar mengajar (PBM) (Bulman dan Fairlie, 2016). Di Jurusan Teknik Komputer dan Informatika Politeknik Negeri Bandung, laporan analisis kebutuhan layanan IT pada awal tahun 2018 (terlampir di Lampiran 1) menunjukkan bahwa 56,25% s.d. 57,5% mata kuliah akan terganggu target capaiannya jika kebutuhan tersebut tidak dipenuhi dengan baik.

Untuk memenuhi kebutuhan tersebut, Politeknik Negeri Bandung berlangganan akses internet kepada *internet service provider* (ISP). Akses internet ini kemudian dikelola oleh Sub Bagian Pengembangan Sistem Informasi (PSI) di bawah Pembantu Direktur Bidang Perencanaan dan Pengembangan. Selain untuk akses pengguna dari dalam kampus, internet juga digunakan untuk mengakses sistem informasi yang ada di dalam kampus.

Untuk mengakses internet dari dalam kampus, perangkat pengguna bergabung di dalam intranet yang menggunakan alamat Internet Protocol (IP) privat dan tidak terkoneksi langsung ke internet. PSI kemudian menempatkan *web proxy* sebagai penghubung kedua jaringan ini sehingga pengguna dapat mengakses layanan yang ada di internet.

Intranet tersebut hanya dapat diakses oleh warga kampus yang sudah terdaftar di PSI. Autentikasi dan otorisasi akses ini dilakukan menggunakan *web proxy*. Dengan *web proxy* pula, PSI mengidentifikasi dan mencatat akses web yang dilakukan sebagai bahan analisis untuk pengembangan layanan selanjutnya. Salah satu hal

yang penting untuk diidentifikasi adalah *hostname* yang dituju oleh aplikasi (misalnya: www.polban.ac.id).

Aplikasi yang digunakan pengguna perlu dikonfigurasi. Konfigurasi dilakukan agar aplikasi mengakses web melalui *web proxy* tersebut. Kredensial pengguna (berupa pasangan *username* dan *password*) juga perlu diatur ketika melakukan konfigurasi. Dengan demikian, aplikasi tidak akan mencoba terhubung ke server yang dituju di internet secara langsung, melainkan hanya akan terhubung ke *web proxy* untuk kemudian meminta disambungkan ke internet. Perilaku tersebut menyebabkan jenis *web proxy* seperti ini disebut sebagai *explicit web proxy* (Rabinovich dan Spatscheck, 2001).

Perilaku tersebut menyebabkan aplikasi yang akan mengakses internet harus mendukung penggunaan *explicit web proxy* (atau disebut *proxy-aware*). Pada *explicit web proxy* yang membutuhkan autentikasi pengguna seperti di Politeknik Negeri Bandung, selain harus *proxy-aware*, aplikasi yang digunakan juga harus mampu mengirim kredensial pengguna ke *explicit web proxy*.

Namun, Wilson (2017) dan Yeh (2017) mengungkapkan bahwa tidak semua aplikasi saat ini *proxy-aware*. Aplikasi selain *web browser* yang meminta akses langsung ke internet semakin banyak, termasuk aplikasi yang digunakan di Politeknik Negeri Bandung (lihat Lampiran 1 sebagai gambaran).

Selain itu, perangkat pribadi pun mulai banyak ditemui penggunaannya secara kasat mata di area kampus. Pemakaian perangkat pribadi mengharuskan pengguna mengonfigurasi ulang semua aplikasi yang digunakannya setiap berpindah jaringan (dari luar ke dalam kampus maupun sebaliknya).

Transparent web proxy, jenis *web proxy* yang bekerja secara transparan di dalam jaringan (Rabinovich dan Spatscheck, 2001) kemudian diharapkan menjadi solusi yang dapat mempermudah akses pengguna di dalam kampus. Jaringan yang menggunakan *web proxy* jenis ini dikonfigurasi untuk memastikan bahwa akses dari aplikasi akan melalui *web proxy* secara otomatis. Dengan demikian, aplikasi

yang digunakan pengguna tidak perlu *proxy-aware*. Pengguna juga tidak perlu melakukan konfigurasi pada perangkatnya setiap kali berpindah jaringan.

Meski mempermudah pengguna, penerapan *transparent web proxy* tidak langsung begitu saja menjadi solusi. Seiring meningkatnya penggunaan web di seluruh dunia, protokol HTTPS – protokol HTTP yang dienkripsi menggunakan protokol TLS (Rescorla, 2000) – dibuat dan terus meningkat penggunaannya secara signifikan pada tahun-tahun terakhir ini hingga mayoritas akses web kini merupakan akses dengan protokol HTTPS (Felt *et al.*, 2017). Karena pesan HTTP berada dalam keadaan terenkripsi, *transparent web proxy* yang berada di antara pengguna dengan server tujuan tidak dapat mengetahui layanan apa yang sedang diakses. Pada *explicit web proxy* hal ini tidak menjadi masalah karena aplikasi pengguna memberitahukan tujuan aksesnya kepada *web proxy*.

Studi yang dilakukan lima tahun terakhir kemudian mengungkapkan metode-metode berikut untuk mengidentifikasi *hostname* dari akses yang terenkripsi:

- melakukan *reverse lookup* terhadap alamat IP berdasarkan entri pada *domain name system* (DNS) (Bermudez *et al.*, 2012);
- melakukan *reverse lookup* terhadap alamat IP berdasarkan rekaman *query* dari pengguna pada server DNS milik organisasi (Bermudez *et al.*, 2012; Foremski, Callegari dan Pagano, 2014);
- menggunakan *server name indication* (SNI), ekstensi dari protokol TLS, mensyaratkan penggunaan aplikasi yang mendukung SNI (Rao, 2013; Shbair, 2017); dan
- menggunakan atribut *common name* (CN) pada sertifikat yang diberikan server pada protokol TLS (Rao, 2013; Shbair, 2017).

Metode-metode tersebut memiliki cara kerja yang jauh berbeda dan memiliki kompleksitas teknis implementasi yang cukup tinggi sehingga tidak *feasible* jika harus diterapkan semuanya pada jaringan untuk dibandingkan.

Selain itu, kedua jenis *web proxy* memiliki cara integrasi yang berbeda di dalam jaringan, sebagaimana diungkapkan oleh Agarwal dan Leonetti (2013), Li dan Clark (2015), McAfee (2014) pada *white paper*-nya, Rabinovich dan Spatscheck

(2001), dan oleh Yeh (2017). Dengan demikian, model jaringan yang sekarang diimplementasi di Politeknik Negeri Bandung juga perlu dianalisis, dievaluasi, dan dimodifikasi dalam rangka persiapan jika *transparent web proxy* diterapkan.

I.2 Rumusan Masalah

PSI menerapkan *explicit web proxy* di Politeknik Negeri Bandung untuk melakukan autentikasi dan otorisasi pengguna serta mencatat akses web. Pada *explicit web proxy* tersebut, *hostname* pada akses dengan protokol HTTPS sudah dapat diidentifikasi. Namun, penerapan *explicit web proxy* tersebut menjadi masalah karena aplikasi masa kini lebih banyak meminta akses langsung ke internet.

Masalah tersebut juga diungkapkan oleh Wilson (2017) dan Yeh (2017). Wilson dan Yeh mengungkapkan bahwa *explicit web proxy* sebaiknya tidak lagi digunakan. *Transparent web proxy* kemudian diharapkan menjadi solusi.

Namun, *web proxy* yang beroperasi secara *transparent* tidak dapat melakukan identifikasi *hostname* pada akses dengan protokol HTTPS seperti yang dilakukan jika *web proxy* beroperasi secara *explicit*. Metode identifikasi akses yang berkembang pada studi-studi terkini kemudian perlu dibandingkan untuk mengetahui mana yang dapat mengidentifikasi *hostname* dengan hasil serupa yang didapatkan dari *explicit web proxy* di lingkungan Politeknik Negeri Bandung.

I.3 *Research Question* dan Hipotesis

Dari masalah tersebut, timbul pertanyaan sebagai berikut:

- RQ₁. Metode apa yang tepat untuk mengidentifikasi *hostname* pada akses melalui protokol HTTPS dengan hasil yang sama dengan yang diperoleh *explicit web proxy* untuk pemakaian di Politeknik Negeri Bandung?

Hipotesis: metode yang tepat untuk mengidentifikasi *hostname* pada akses melalui protokol HTTPS dengan hasil yang sama dengan yang diperoleh *explicit web proxy* adalah metode SNI pada *handshake TLS*.

- RQ₂. Bagaimana arsitektur jaringan yang dibutuhkan untuk menerapkan *transparent web proxy* dengan metode identifikasi *hostname* jawaban RQ₁ di Politeknik Negeri Bandung?

I.4 Tujuan

Tugas akhir ini bertujuan untuk memudahkan pengguna untuk menggunakan aplikasi yang dibutuhkan di dalam kampus, namun mempertahankan kebutuhan PSI untuk melakukan autentikasi, otorisasi, serta pencatatan akses.

Untuk membantu mencapai tujuan tersebut, maka selain melakukan penelitian untuk menemukan metode yang dapat mengidentifikasi *hostname* dari akses dengan protokol HTTPS pada *transparent web proxy* sesuai kebutuhan PSI, arsitektur jaringan di Politeknik Negeri Bandung juga dievaluasi untuk penerapan *transparent web proxy* dengan metode terpilih.

Selain itu, hasil penelitian ini ditujukan untuk melihat kemaungkinan implementasi metode identifikasi terpilih dengan perubahan yang minimal terhadap arsitektur yang saat ini ada di Politeknik Negeri Bandung.

I.5 Luaran

Luaran tugas akhir ini adalah:

1. metode terpilih yang dapat mengidentifikasi *hostname* dari akses dengan protokol HTTPS pada *transparent web proxy* sesuai kebutuhan PSI;
2. model jaringan dan konfigurasi yang dibutuhkan untuk penerapan *transparent web proxy* dengan metode terpilih di Politeknik Negeri Bandung; dan
3. implementasi model jaringan dan konfigurasi dalam skala lab untuk pemakaian di Politeknik Negeri Bandung.

Implementasi *transparent web proxy* dengan metode identifikasi terpilih juga berkontribusi terhadap studi Agarwal dan Leonetti (2013). Studi Agarwal dan Leonetti menggunakan Squid, *web proxy* yang digunakan di Politeknik Negeri Bandung, namun belum dapat mengidentifikasi akses dengan protokol HTTPS.

I.6 Manfaat

Identifikasi *hostname* pada akses web merupakan *requirement* dari berbagai metode yang mengidentifikasi akses web pada *service level* (Bermudez *et al.*, 2012; Foremski, Callegari dan Pagano, 2014; Yeh, 2017). Identifikasi pada *service level* merupakan identifikasi yang menggabungkan *hostname-hostname* yang ada menjadi satu label layanan yang mudah dimengerti oleh pengelola. Label layanan tersebut misalnya akses Facebook, Twitter, *e-mail*, dan *video streaming*.

Dengan demikian, selain membantu mewujudkan tujuan yang telah dinyatakan pada subbab I.4, metode yang terpilih pada tugas akhir ini merupakan fondasi awal untuk identifikasi akses web di Politeknik Negeri Bandung pada *service level*. Tugas akhir ini juga bermanfaat untuk mengevaluasi dan menerapkan pengelolaan jaringan yang lebih baik di Politeknik Negeri Bandung.

Tugas akhir ini dilakukan berdasarkan kondisi di Politeknik Negeri Bandung. Namun, luaran dari tugas akhir ini juga relevan bagi organisasi lain yang memiliki kebutuhan pengelolaan akses web yang serupa.

I.7 Ruang Lingkup

Pekerjaan tugas akhir ini dibatasi sebagai berikut:

- dalam menentukan metode identifikasi *hostname*, data yang menjadi *ground truth* adalah log yang dihasilkan dari *explicit web proxy* yang beroperasi di Politeknik Negeri Bandung saat ini. Pemilihan ini dibangun dengan asumsi bahwa log yang saat ini dihasilkan sudah sesuai dengan kebutuhan PSI;
- akses dengan protokol HTTPS yang diidentifikasi hanyalah yang melalui *port* standar (443) dengan asumsi bahwa akses yang menuju *port* 443 pasti merupakan akses dengan protokol HTTPS (atau TLS pada umumnya);
- aplikasi-aplikasi di luar *web browser* yang dicoba pada implementasi skala lab diambil dari penggunaan mahasiswa di Jurusan Teknik Komputer dan Informatika, karena karakteristik penggunaannya mencerminkan *superset* dari penggunaan rata-rata di Politeknik Negeri Bandung berdasarkan keterangan PSI. Daftar aplikasi yang dicobakan dituliskan pada subbab III.6.4;

- autentikasi pada implementasi skala lab menggunakan protokol *remote authentication dial in user service* (RADIUS). Hal ini dilakukan karena *transparent web proxy* tidak dapat menangani autentikasi pengguna secara mandiri (McAfee, 2014). Protokol RADIUS dipilih karena merupakan protokol *de facto* untuk autentikasi yang didukung oleh banyak vendor perangkat jaringan (Feng, 2009);
- pemilihan *backend* dari protokol RADIUS sebagai sumber informasi autentikasi (seperti *user directory*) dan komunikasinya berada di luar lingkup tugas akhir ini, karena tugas akhir ini fokus pada permasalahan identifikasi dan pencatatan akses saja;
- korelasi antara variabel bebas dengan variabel terikat dianalisis secara bivariat;
- eksperimen implementasi *transparent web proxy* yang sedang dilakukan oleh PSI di jaringan nirkabel gedung Direktorat, Pendopo Tonny Soewandito, dan gedung pascasarjana tidak dijadikan bahan pertimbangan ketika melakukan analisis *problem domain* karena merupakan eksplorasi dan percobaan PSI;
- penelitian ini tidak berupaya menggabungkan metode-metode yang ada menjadi metode baru;
- metrik *quality of service*, performa *web proxy* terkait dengan *concurrent user* atau *concurrent connection* berada di luar lingkup tugas akhir ini.

I.8 Sistematika Penulisan

Laporan tugas akhir ini disusun dengan sistematika berikut:

BAB I PENDAHULUAN, berisi latar belakang, rumusan masalah yang dihadapi, *research question* dan hipotesis, tujuan, luaran, manfaat, batasan ruang lingkup, serta sistematika penulisan laporan tugas akhir ini;

BAB II TINJAUAN PUSTAKA, berisi dasar-dasar teori yang digunakan pada tugas akhir ini serta kajian terhadap karya ilmiah sejenis;

- BAB III METODOLOGI PENELITIAN, berisi metodologi penelitian tugas akhir secara rinci, meliputi jenis, subjek, objek, variabel, data, dan tahapan penelitian;
- BAB IV PENENTUAN METODE IDENTIFIKASI *HOSTNAME*, berisi analisis *problem domain* terkait metode identifikasi *hostname*, pembuatan aplikasi eksperimen, penyiapana data, dan hasil eksperimen beserta analisisnya. Bab ini menjawab RQ₁;
- BAB V PENERAPAN *TRANSPARENT WEB PROXY*, berisi analisis *problem domain* terkait kondisi di Politeknik Negeri Bandung, perancangan jaringan untuk implementasi *transparent web proxy*, pembuatan implementasi skala lab, percobaan aplikasi pengguna dan hasilnya. Bab ini menjawab RQ₂ dan menguatkan jawaban RQ₁; serta
- BAB VI KESIMPULAN DAN SARAN, berisi kesimpulan dari penelitian yang dilakukan dan saran untuk pengembangan selanjutnya.

BAB II

TINJAUAN PUSTAKA

Bab ini menjelaskan dasar teori yang digunakan pada tugas akhir ini serta kajian terhadap karya ilmiah sejenis.

II.1 Dasar Teori

Subbab ini menjelaskan dasar-dasar teori yang digunakan pada tugas akhir ini.

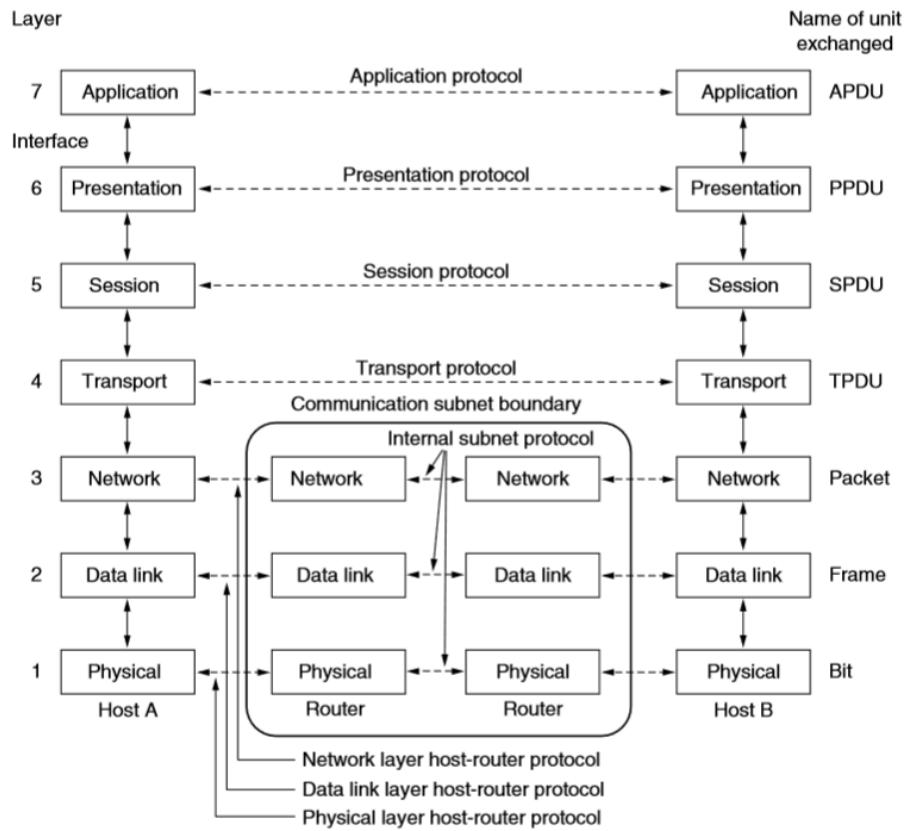
II.1.1 Model Referensi OSI dan TCP/IP

International Standards Organization (ISO) memodelkan standarisasi protokol-protokol yang digunakan dalam komunikasi data (Tanenbaum dan Wetherall, 2011). Model ini disebut model referensi *open system interconnection* (OSI) karena menggambarkan hubungan antar sistem yang terbuka untuk berkomunikasi dengan sistem lainnya.

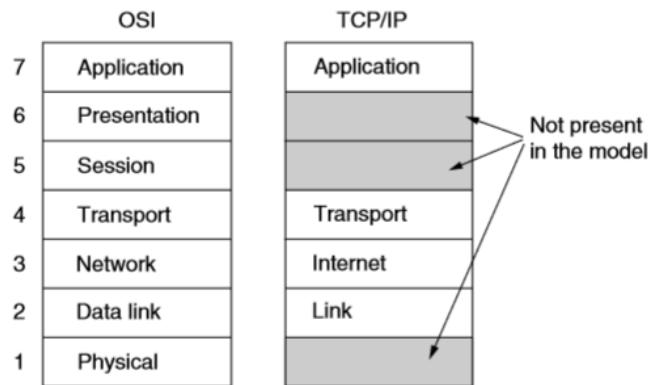
Model referensi OSI menggambarkan fungsi komunikasi yang dilakukan dalam tujuh *layer*, mulai dari *physical layer* hingga *application layer*. Model ini dibangun sebelum protokol-protokol yang melayaninya dibuat. Dengan demikian, model ini netral dan tidak bias terhadap teknologi tertentu. Tanenbaum dan Wetherall (2011) menggambarkan urutan tujuh *layer* tersebut pada Gambar II.1.

Model referensi TCP/IP dibangun berdasarkan arsitektur ARPANET, jaringan riset Departemen Pertahanan Amerika Serikat yang kemudian berkembang menjadi internet (Tanenbaum dan Wetherall, 2011). Model referensi TCP/IP dibangun sesudah protokol-protokolnya dibuat. Dengan demikian, model referensi ini hanya cocok digunakan untuk menggambarkan implementasi jaringan yang menggunakan *protocol stack* TCP/IP.

Berbeda dengan model referensi OSI, model TCP/IP memiliki empat *layer*, yaitu *link layer*, *internet layer*, *transport layer*, dan *application layer*. Tanenbaum dan Wetherall (2011) menggambarkan urutan empat *layer* tersebut beserta perbandingannya dengan model referensi OSI pada Gambar II.2.



Gambar II.1. Model referensi OSI



Gambar II.2. Model referensi TCP/IP

Komunikasi antar perangkat jaringan (termasuk klien) yang terhubung secara langsung merupakan lingkup *link layer*. Komunikasi antar perangkat jaringan dilakukan menggunakan alamat MAC, bukan alamat IP. Perangkat-perangkat yang terhubung langsung pada *link layer* kemudian disebut saling berbagi *bandwidth domain* dan *broadcast domain* (Oppenheimer, 2011). Kedua *domain* ini, menurut Oppenheimer, harus dijaga agar cakupannya sekecil mungkin; dengan kata lain,

jumlah perangkat yang berada pada *bandwidth domain* dan *broadcast domain* yang sama harus diminimalkan.

Komunikasi antar perangkat jaringan (termasuk klien) yang tidak terhubung secara langsung merupakan lingkup dari *internet layer*. Komunikasi antar perangkat jaringan pada *layer* ini menggunakan alamat IP. Pada *layer* inilah terdapat bahasan mengenai *routing* yang dapat menjelaskan permasalahan yang ada di Politeknik Negeri Bandung saat ini.

Tanenbaum dan Wetherall menyatakan ruang lingkup *transport layer* dengan hal terkait *connection-oriented* dan *connectionless*. Pada *protocol stack* TCP/IP, protokol *connection-oriented* adalah protokol TCP, sementara protokol *connectionless* adalah protokol UDP. Protokol HTTPS yang menjadi kajian merupakan protokol yang beroperasi dengan protokol TCP pada *transport layer*.

Protokol yang spesifik mengenai aplikasi yang berhadapan dengan pengguna berada dalam ruang lingkup *application layer*. Protokol HTTP, HTTPS, dan TLS merupakan protokol yang berada pada *layer* ini. Meski demikian, protokol TLS akan lebih mudah dipahami jika ditempatkan pada *presentation layer* di model referensi OSI, karena protokol TLS melakukan enkripsi terhadap seluruh protokol yang ada di atasnya.

Kedua model referensi ini digunakan pada subbab V.1.1 untuk menganalisis permasalahan identifikasi akses dengan protokol HTTPS serta permasalahan yang diangkat pada latar belakang tugas akhir ini.

II.1.2 Hypertext Transfer Protocol (HTTP)

HTTP adalah protokol *application layer* untuk sistem informasi yang terdistribusi dan kolaboratif (Fielding *et al.*, 1999). HTTP adalah protokol yang menjadi fondasi web di internet.

Protokol HTTP berbasis *request-response*. Klien mengirim *request* ke server berupa *request method*, *uniform resource identifier* (URI), dan versi protokol yang digunakan (hingga kini protokol HTTP sudah berkembang hingga versi HTTP/2), diikuti *header-header* HTTP, dan dilanjutkan dengan *body request* jika ada. Server

kemudian akan merespons dengan *status code*, diikuti *header-header* HTTP, dan dilanjutkan dengan *body response*. Visualisasi dari alur *request-response* ini ditunjukkan pada Gambar II.3.



Gambar II.3. Pertukaran pesan *request* dan *response* pada protokol HTTP

Protokol ini bersifat *plaintext*, sehingga pihak-pihak yang berada di antara klien dan server (semua perangkat jaringan yang dilalui, misalnya) dapat membaca semua pertukaran pesan. Dengan demikian, *transparent web proxy* tidak memiliki masalah untuk mengidentifikasi akses web dengan protokol HTTP.

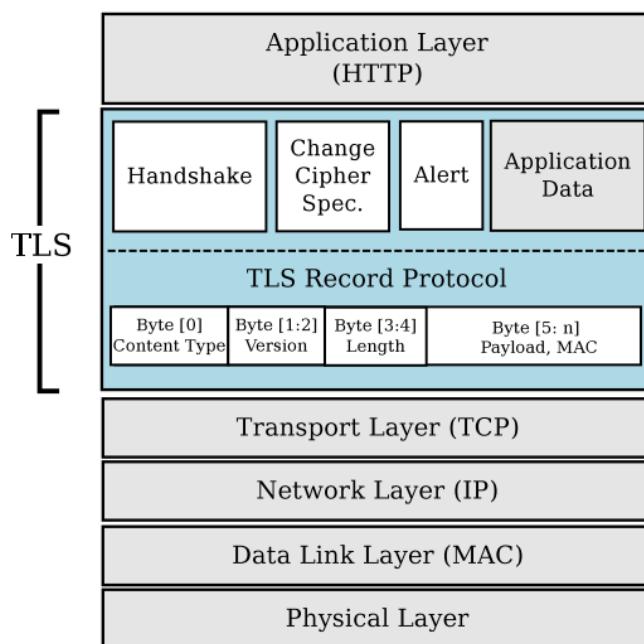
Namun, karena sifat *plaintext* ini, pihak-pihak di antara klien dan server dapat melakukan modifikasi terhadap pesan yang dipertukarkan. Hal ini berbahaya jika pesan yang dipertukarkan merupakan pesan yang sensitif, misalnya pesan-pesan pada layanan perbankan. Oleh karena itulah, protokol HTTPS dikembangkan.

II.1.3 HTTP over TLS (HTTPS)

Netscape Communications mulai mengembangkan protokol HTTPS pada tahun 1994 (Walls, 2006). Protokol HTTPS digunakan untuk mentransmisikan pesan HTTP yang dienkripsi dengan protokol *transport layer security* (TLS) dan diformalkan pada tahun 2000 (Rescorla, 2000).

Pesan HTTPS dienkripsi dengan protokol TLS ketika meninggalkan klien. Pesan HTTPS kemudian didekripsi ketika tiba di server. Pesan balasan juga dienkripsi ketika meninggalkan server dan didekripsi ketika tiba di klien. Dengan mekanisme ini, jalur yang dilalui di antara klien dan server tidak dapat melihat isi transmisi dan memodifikasinya, termasuk *header* dan *payload* HTTP yang terkandung di dalamnya.

Posisi protokol TLS dalam protokol HTTPS pada *layer TCP/IP* ditunjukkan oleh Shbair (2017) dengan Gambar II.4.



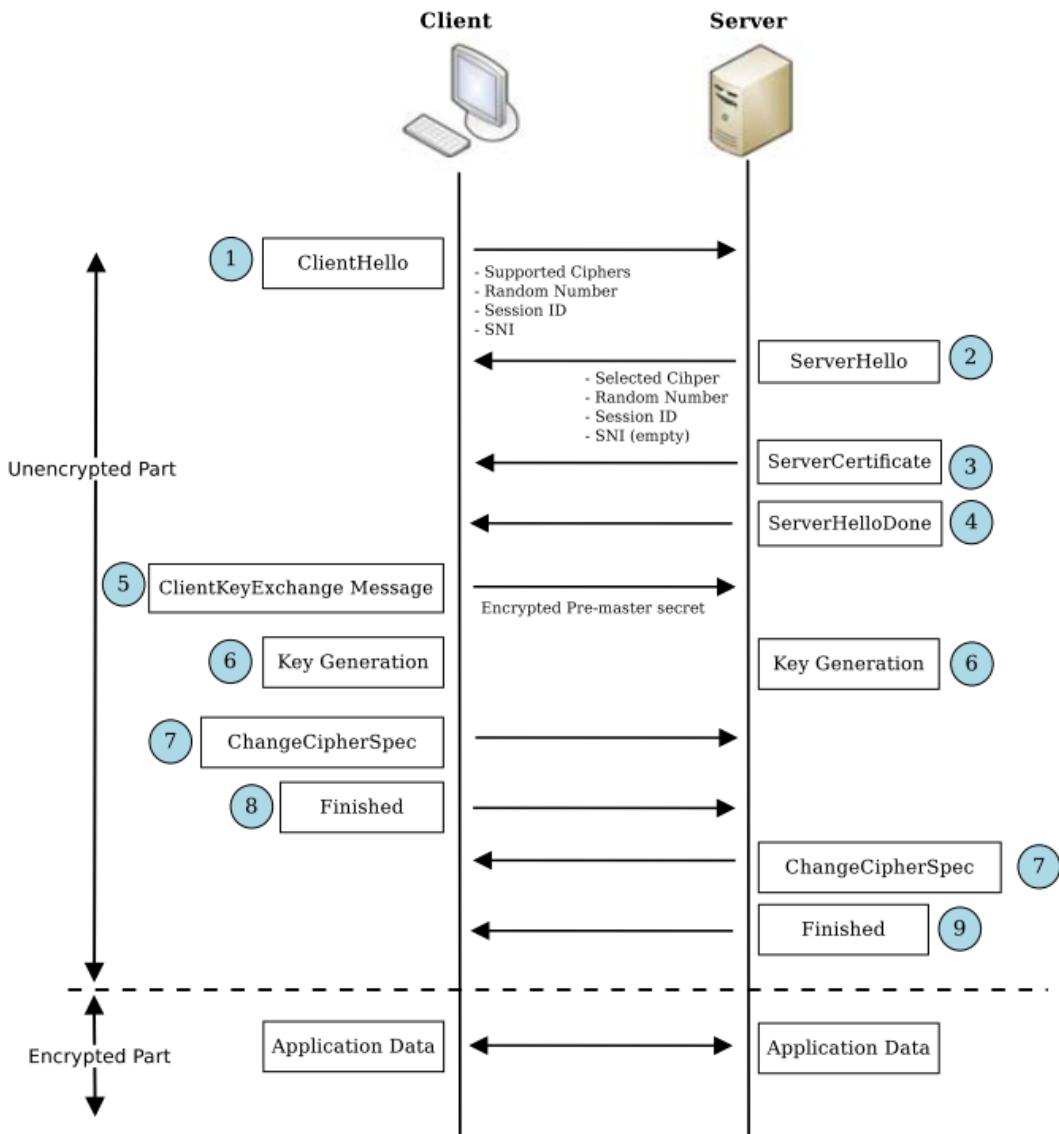
Gambar II.4. Posisi protokol TLS dalam protokol HTTPS pada *layer TCP/IP*

II.1.4 *Transport Layer Security (TLS)*

Tujuan utama dari protokol TLS adalah untuk memberikan saluran yang aman antara dua pihak autentik untuk saling berkomunikasi (Shbair, 2017). Pihak ketiga tidak dapat melihat konten komunikasi sama sekali. Klien tidak perlu dikonfigurasi untuk dapat menggunakan protokol ini.

Ada sembilan pertukaran pesan yang dilakukan ketika *handshake* sebagaimana digambarkan oleh Shbair (2017) pada Gambar II.5. Sembilan pertukaran pesan ini tidak terenkripsi, kecuali pesan Pre-master Secret yang dikirimkan oleh klien pada

pesan kelima. Sesudah sembilan pertukaran pesan ini, data dari *layer* aplikasi di atasnya (lihat Gambar II.4) dikirimkan dalam bentuk terenkripsi.



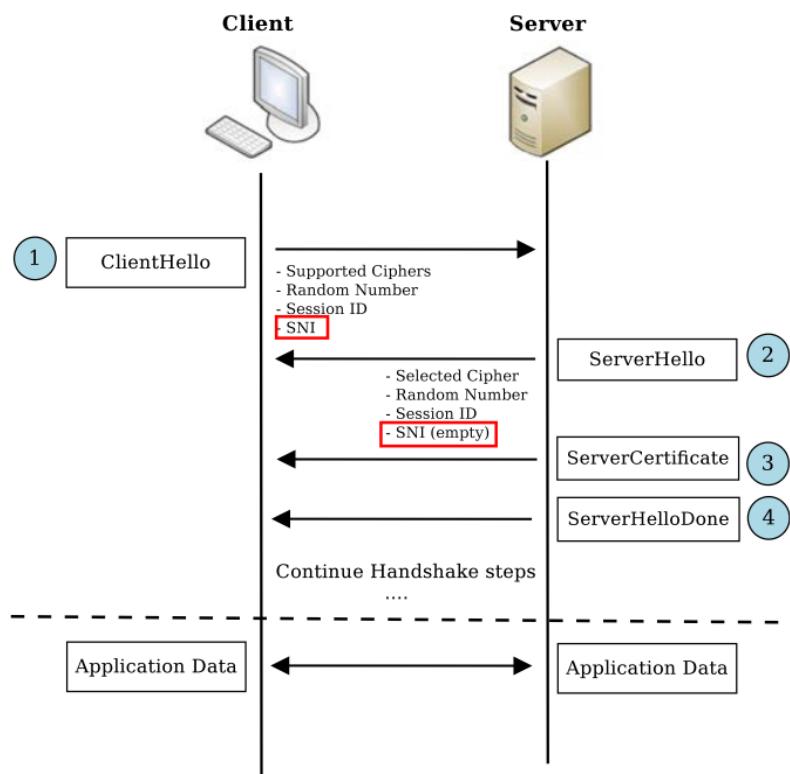
Gambar II.5. Urutan pertukaran pesan pada *handshake* TLS

Terdapat beberapa atribut informasi pada sertifikat TLS. Salah satunya adalah atribut *common name* (CN) yang mengandung informasi *hostname* utama apa yang diwakili oleh sertifikat tersebut. Atribut ini penting karena digunakan pada salah satu metode identifikasi *hostname* yang dijelaskan pada subbab II.2.

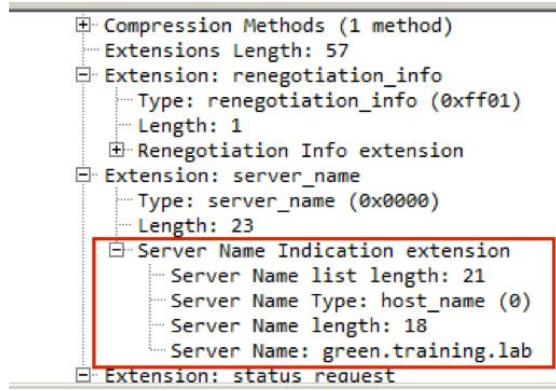
II.1.5 Server Name Indication (SNI)

Pada tahun 2003, diperkenalkan *extension-extension* dari protokol TLS yang digunakan untuk menambah fungsionalitas protokol TLS ketika melakukan *handshake* (Blake-Wilson *et al.*, 2003). Salah satu *extension* yang diperkenalkan adalah SNI. Dengan SNI, klien yang menggunakan protokol TLS dapat mengirimkan *hostname* dari server yang dituju ketika melakukan *handshake*. Dengan informasi tambahan ini, server yang melakukan *virtual hosting* (satu alamat IP digunakan untuk melakukan *hosting* banyak layanan) dapat menghadirkan informasi yang tepat sejak *handshake* dilakukan.

Informasi SNI dibungkus dalam suatu struktur data yang diletakkan pada akhir bagian dari pesan pertama yang dipertukarkan ketika *handshake*, yaitu pesan `ClientHello`. Ilustrasi dari informasi SNI ini ditunjukkan pada Gambar II.6 oleh Shbair (2017), sementara Campos (2015) menghadirkan Gambar II.7 untuk menunjukkan bagaimana informasi SNI dilihat menggunakan *software Wireshark*.



Gambar II.6. SNI pada *handshake* TLS



Gambar II.7. *Field SNI dilihat menggunakan Wireshark*

II.1.6 Domain Name System (DNS)

Tanenbaum dan Wetherall (2011) mendefinisikan *domain name system* (DNS) sebagai sistem pemetaan dari *hostname* menjadi alamat IP dan sebaliknya yang diimplementasi secara terdistribusi. DNS dibangun karena komunikasi antar komputer sebenarnya membutuhkan alamat IP, sementara pengguna lebih mudah mengingat *hostname* seperti `www.polban.ac.id` daripada alamat IP seperti `103.209.131.20`.

Sebuah entri pada basis data DNS memiliki lima buah informasi, yaitu (Tanenbaum dan Wetherall, 2011):

- *hostname* yang mewakili entri ini;
- *time to live*, merupakan waktu dalam detik yang menunjukkan berapa lama entri ini boleh di-*cache* oleh klien DNS;
- *class*, selalu bernilai IN untuk penggunaan di internet;
- jenis entri, bernilai salah satu dari sepuluh jenis entri DNS (diuraikan pada Tabel II.1);
- *value*, merupakan nilai entri tersebut sesuai dengan *type*-nya.

Untuk nama domain dan jenis entri DNS yang sama, bisa terdapat beberapa entri. Sebagai contoh, *lookup* terhadap `www.google.com` dapat menghasilkan enam buah alamat IP dengan keluaran *console* yang ditunjukkan pada Gambar II.8.

Tabel II.1. Jenis entri DNS

Type	Meaning	Value
SOA	Start of authority	Parameters for this zone
A	IPv4 address of a host	32-Bit integer
AAAA	IPv6 address of a host	128-Bit integer
MX	Mail exchange	Priority, domain willing to accept email
NS	Name server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
SPF	Sender policy framework	Text encoding of mail sending policy
SRV	Service	Host that provides it
TXT	Text	Descriptive ASCII text

```
$ dig www.google.com

; <>> DiG 9.11.3-1ubuntu1.1-Ubuntu <>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30025
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;www.google.com.           IN      A

;; ANSWER SECTION:
www.google.com.      192      IN      A      74.125.24.104
www.google.com.      192      IN      A      74.125.24.147
www.google.com.      192      IN      A      74.125.24.103
www.google.com.      192      IN      A      74.125.24.105
www.google.com.      192      IN      A      74.125.24.99
www.google.com.      192      IN      A      74.125.24.106

;; Query time: 2 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Mon Mar 19 06:58:49 UTC 2018
;; MSG SIZE  rcvd: 139
```

Gambar II.8. Keluaran *console* pada *lookup* www.google.com

Pada penelitian ini, metode *reverse lookup* terhadap alamat IP berdasarkan entri pada DNS yang diungkapkan oleh Bermudez *et al.* (2012) menggunakan entri berjenis PTR yang menyimpan *hostname* dari alamat IP tertentu. Entri jenis ini bergantung pada apakah pemilik alamat IP menyimpan *hostname*-nya pada DNS atau tidak.

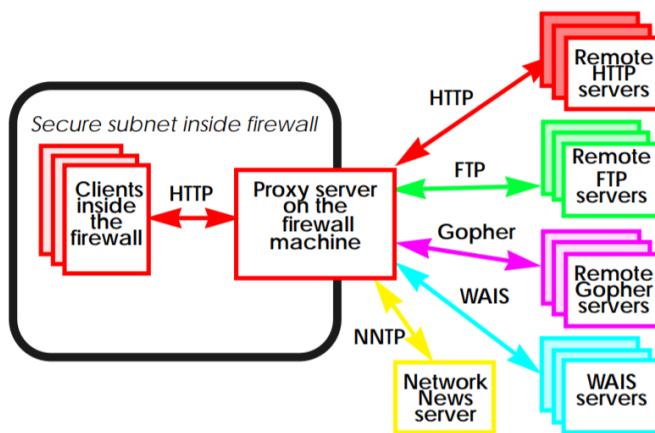
Untuk melakukan *reverse lookup*, alamat IP ditulis dari oktet terakhir hingga oktet pertama, kemudian ditambah .in-addr.arpa. Sebagai contoh, untuk mencari *hostname* dari alamat IP 222.124.203.161, dilakukan *query* terhadap DNS dengan

domain 161.203.124.222.in-addr.arpa dengan jenis PTR. Jika entri ini ditemukan di DNS, maka hasilnya akan berupa *hostname*, misalnya *polban.ac.id*.

Untuk meningkatkan performa DNS, maka terdapat informasi *time to live* pada entri basis data DNS. Dengan informasi ini, klien dapat melakukan *caching* terhadap hasil dari *query* DNS yang dilakukannya selama jumlah detik yang tertera pada *time to live*, sehingga untuk akses berikutnya ke *hostname* yang sama, klien tidak perlu lagi melakukan *query* selama belum melampaui waktu *time to live* (Tanenbaum dan Wetherall, 2011).

II.1.7 Web Proxy

Server *proxy* adalah aplikasi yang menyediakan akses di antara dua jaringan; umumnya kedua jaringan tersebut adalah intranet dengan internet (Villarroel-Acosta *et al.*, 2017). *Web proxy* adalah aplikasi *proxy* yang melayani protokol umum di web, yaitu HTTP dan HTTPS. Selain *web proxy*, ada juga *proxy* untuk aplikasi-aplikasi lain seperti FTP (Luotonen dan Altis, 1994). Gambar II.9 adalah gambaran umum sebuah *proxy* sebagaimana digambarkan oleh Luotonen dan Altis (1994).



Gambar II.9. Gambaran umum sebuah *proxy*

Web proxy bertindak sebagai *middleware* antara klien dengan server yang dituju oleh klien. *Web proxy* menerima permintaan dari klien, kemudian membuat permintaan yang sama ke server yang dituju. Setelah menerima respons dari server, *web proxy* meneruskan respons tersebut ke klien.

Pada dasarnya, penggunaan *web proxy* memiliki beberapa *use case* sebagai berikut:

1. membagi koneksi Internet ke banyak pengguna (Yeh, 2017). Pada awal 1990-an, internet baru dapat diakses melalui koneksi *dial-up*. Agar koneksi dapat digunakan oleh banyak pengguna, *web proxy* disambungkan dengan internet, kemudian diakses oleh pengguna lain di jaringan intranet yang tidak tersambung ke internet. Namun, sejak penggunaan koneksi *broadband* meluas dan *network address translation* (NAT) diformalkan, *use case* ini sudah tidak relevan lagi;
2. meningkatkan performa dan menghemat *bandwidth* (Yeh, 2017). Pada akhir tahun 1990-an dan awal tahun 2000-an, penggunaan internet mengalami peningkatan eksponensial, sedangkan *resource bandwidth* yang dimiliki tidak dapat ditingkatkan secepat permintaannya. *Web proxy* kemudian melakukan *caching* terhadap akses *resource* yang umumnya dilayani dengan protokol HTTP, sehingga dapat meningkatkan performa dan menghemat *bandwidth*;
3. melakukan autentikasi, otorisasi, dan pencatatan akses (Tanenbaum dan Wetherall, 2011). *Explicit web proxy* (dijelaskan lebih lanjut pada subbab II.1.7.1) dilengkapi fitur autentikasi dan otorisasi. Dengan autentikasi, pengguna diidentifikasi secara individual, kemudian hak akses atas berbagai *resource* dapat diizinkan, ditolak, dan/atau dibatasi (misalnya dengan *bandwidth throttling*). Penerapan *use case* ini lebih efektif jika *explicit web proxy* dijadikan satu-satunya jalan akses pengguna ke internet.

Uraian *use case* dari *web proxy* tersebut selaras dengan alasan mengapa Politeknik Negeri Bandung menggunakan *explicit web proxy* setidaknya sejak tahun 2003 berdasarkan keterangan PSI. Penggunaan *explicit web proxy* tersebut pernah optimal dan tepat guna pada masanya.

Meskipun *web proxy* bukan merupakan teknologi baru, namun kajian tugas akhir ini tetap diperlukan, karena sebagaimana ditunjukkan oleh Wilson (2017) dan Yeh (2017) terjadi perkembangan teknologi web yang semakin dinamis (terutama hadirnya protokol HTTPS yang semakin meluas serta aplikasi-aplikasi masa kini yang meminta akses langsung ke internet) menyebabkan praktik lama seperti penggunaan *explicit web proxy* perlu ditinjau ulang.

II.1.7.1 Web Proxy Deployment

Rabinovich dan Spatscheck (2001) mengungkapkan dua cara *deployment web proxy* dalam suatu jaringan, yaitu secara *non-transparent* dan secara *transparent*. *Web proxy* yang di-deploy secara *non-transparent* umum disebut sebagai *explicit web proxy*.

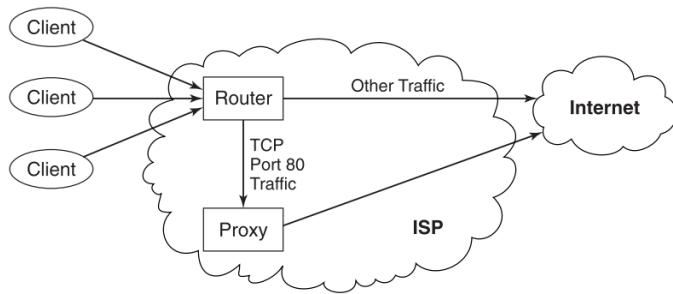
Pada *non-transparent web proxy deployment*, klien mengetahui adanya *web proxy* yang perlu dituju agar dapat melakukan akses web ke internet. Dengan demikian, klien akan mengirim semua *request*-nya ke *web proxy*, bukan ke server yang diindikasikan pada URI.

Hal yang dilakukan oleh klien jika klien tidak mengetahui keberadaan *web proxy* ketika akan melakukan akses web adalah:

1. klien akan melakukan *lookup* terhadap *hostname* dari server yang dituju ke suatu server DNS, kemudian server DNS akan mengembalikan alamat IP yang dapat dituju oleh klien untuk menyampaikan *request*-nya; kemudian
2. berbekal alamat IP yang sudah diperoleh, klien kemudian membuat koneksi langsung ke server tujuan, kemudian mengirimkan *request* berupa *path* dari *resource* yang ingin diakses. Server tujuan kemudian mengembalikan *resource* yang dimaksud.

Dengan keberadaan *explicit web proxy*, klien akan langsung membuat koneksi ke *web proxy*, kemudian mengirimkan seluruh URI dari *resource* yang ingin diakses (termasuk *hostname* dari server yang dituju). *Lookup* terhadap *hostname* kemudian akan dilakukan oleh *web proxy*.

Cara lainnya untuk melakukan *deployment web proxy* adalah dengan melakukannya secara *transparent*. Pada cara ini, klien tidak mengetahui bahwa ada *web proxy* yang akan dituju. Dengan demikian, cara klien mengakses *resource* tidak ada bedanya dengan ketika tidak ada *web proxy* sama sekali. Gambar II.10 menunjukkan penerapan *transparent web proxy* di ISP sebagaimana digambarkan oleh Rabinovich dan Spatscheck (2001).



Gambar II.10. Penerapan *transparent web proxy* di ISP

II.1.7.2 Perbandingan *Explicit* dengan *Transparent Web Proxy*

Tabel II.2 menunjukkan perbandingan antara *explicit* dengan *transparent web proxy* sebagaimana dijelaskan oleh Agarwal dan Leonetti (2013), Li dan Clark (2015), McAfee (2014) pada *white paper*-nya, Rabinovich dan Spatscheck (2001), dan oleh Yeh (2017).

Tabel II.2. Perbandingan *explicit* dengan *transparent web proxy*

Aspek	<i>Explicit Web Proxy</i>	<i>Transparent Web Proxy</i>
Routing akses ke web proxy	Pengaturan <i>web proxy</i> perlu dilakukan di setiap klien.	Tidak perlu ada pengaturan <i>web proxy</i> yang dilakukan di klien.
Pengecualian akses ke web proxy	Jika suatu <i>resource</i> perlu diakses tanpa melalui <i>web proxy</i> , maka hal tersebut dapat diatur di sisi klien maupun di sisi jaringan.	Pengaturan pengecualian harus dilakukan di sisi jaringan, karena tidak ada konfigurasi di sisi klien.
Autentikasi	<i>Web proxy</i> dapat melakukan autentikasi jika diakses dengan aplikasi yang <i>proxy-aware</i> .	Autentikasi harus dilakukan di luar <i>web proxy</i> .
Sesi autentikasi	Tidak ada sesi yang perlu dikelola, sepanjang setiap koneksi terautentikasi.	Ada sesi yang perlu dikelola oleh server autentikasi. Sesi ini dapat bersifat <i>time-based</i> ataupun <i>cookie-based</i> .
Akses web dengan HTTPS	<i>Hostname</i> tujuan dapat diketahui oleh <i>web proxy</i> berdasarkan <i>request</i> dari aplikasi pengguna.	Hanya alamat IP tujuan yang dapat dilihat oleh <i>web proxy</i> , kecuali klien mengirim <i>request</i> dengan tambahan SNI.
DNS	<i>Lookup</i> alamat IP hanya perlu dilakukan oleh <i>web proxy</i> .	<i>Lookup</i> alamat IP dilakukan oleh dua pihak, yaitu klien dan <i>web proxy</i> .
Jenis aplikasi	Aplikasi yang digunakan oleh pengguna harus <i>proxy-aware</i> .	Aplikasi yang digunakan oleh pengguna tidak harus <i>proxy-aware</i> .

II.1.8 Authentication, Authorization, and Accounting (AAA)

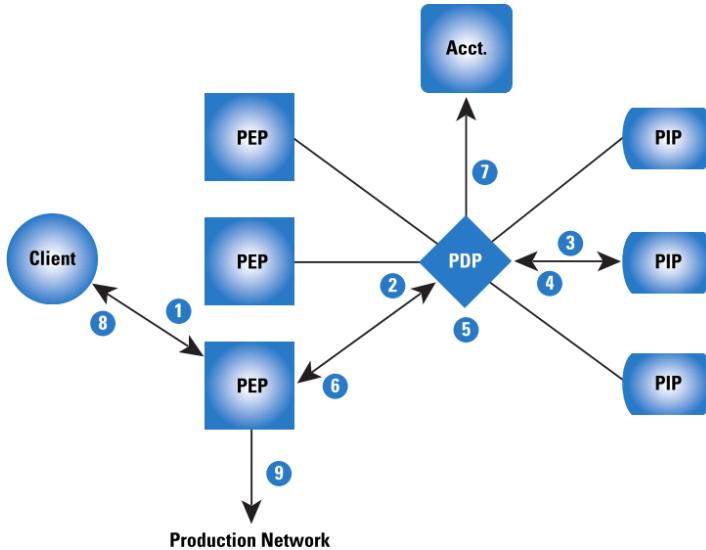
Masing-masing dari AAA merupakan unsur-unsur dasar dalam keamanan jaringan (Convery, 2007a). Pertanyaan, “siapa Anda?” ditanyakan pada proses autentikasi. Pertanyaan, “apa yang boleh Anda lakukan?” ditanyakan pada proses otorisasi. Terakhir, pertanyaan, “apa yang Anda lakukan?” ditanyakan pada proses akuntansi.

Pada jaringan komputer tanpa AAA, suatu jaringan harus dikonfigurasi secara statis untuk mengendalikan akses pengguna. Alamat IP harus diatur secara statis dan perangkat pengguna tidak boleh berpindah lokasi. Dengan AAA, pengguna dapat mengakses layanan dari berbagai tempat, sehingga jaringan harus dikonfigurasi secara dinamis.

Ada lima komponen AAA sebagaimana diungkapkan oleh Convery (2007a), yaitu:

- klien, yaitu perangkat yang akan mengakses jaringan;
- *policy enforcement point* (PEP/authenticator), yaitu perangkat yang melayani permintaan akses dari klien. Perangkat ini bisa berupa banyak hal, di antaranya *manageable switch*, *wireless access point*, *inline security gateway*, atau bahkan *web proxy*. PEP bertanggung jawab meng-*enforce* izin akses untuk klien;
- *policy information point* (PIP), yaitu *repository* informasi yang membantu dalam pengambilan keputusan apakah akses diberikan atau tidak. PIP bisa berupa *database* dari ID perangkat yang diizinkan, atau *database* pengguna. Komunikasi dengan PIP dapat dilakukan dengan protokol tertentu, misalnya *lightweight directory access protocol* (LDAP);
- *policy decision point* (PDP/server AAA), yaitu pengambil keputusan AAA. PDP menerima permintaan akses dari PEP. PDP kemudian melakukan *query* ke satu atau lebih PIP untuk mendapatkan informasi. Sesudah informasi yang cukup diperoleh, PDP memutuskan apakah akses ke jaringan diberikan atau tidak, serta bagaimana akses tersebut di-*enforce*. Keputusan ini dikirimkan ke PEP untuk kemudian dilakukan *enforcement*;
- *accounting and reporting system*, yang dengannya AAA dapat menginformasikan siapa yang berada di jaringan, dari mana akses masuknya, serta akses apa saja yang diizinkan untuk pengguna tersebut.

Komponen-komponen tersebut berinteraksi ketika pengguna akan mengakses jaringan dengan *flow* yang digambarkan Convery (2007a) pada Gambar II.11.



Gambar II.11. Interaksi komponen AAA ketika pengguna mengakses jaringan

Kelima komponen tersebut tidak harus menjadi lima jenis perangkat yang terpisah. Komponen-komponen tersebut dapat dikombinasikan dalam satu perangkat. Jika kita melihat *explicit web proxy* dengan autentikasi yang diterapkan di Politeknik Negeri Bandung saat ini, *explicit web proxy* tersebut mengambil peran sebagai PEP, PDP, dan *accounting and reporting system* secara bersamaan.

II.1.8.1 Autentikasi Klien dengan PEP

Autentikasi klien dengan PDP dapat berjalan pada *layer* kedua atau *layer* yang lebih tinggi pada model OSI. Ada banyak protokol yang dapat digunakan untuk autentikasi klien dengan PDP, di antaranya PPP, PPPoE, IEEE 802.1X, IPSec, SSL VPN, dan HTTP *captive portal* (Hole, Dyrnes dan Thorsheim, 2005; Convery, 2007b). Protokol-protokol tersebut sangat tergantung pada dukungan perangkat yang menjadi PEP.

II.1.8.2 Komunikasi PEP dengan PDP

Sesudah klien berkomunikasi dengan PEP, PEP perlu berkomunikasi dengan PDP sebagai pengambil keputusan. Ada tiga protokol utama yang digunakan pada tahap ini, yaitu TACACS+, RADIUS, dan Diameter (Convery, 2007b).

TACACS+ dikembangkan oleh Cisco sebagai protokol *proprietary*. Meski dikembangkan oleh Cisco, TACACS+ didukung pula oleh perangkat-perangkat Juniper. Sifatnya yang *proprietary* menyebabkan tidak banyak vendor perangkat jaringan yang mendukung protokol ini.

Protokol kedua adalah *remote authentication dial in user service* (RADIUS). Protokol ini didukung sangat luas oleh berbagai vendor perangkat jaringan. Karena dukungan yang luas ini, RADIUS menjadi protokol utama untuk komunikasi PEP dengan PDP, dan bahkan secara *de facto* menjadi standar (Feng, 2009).

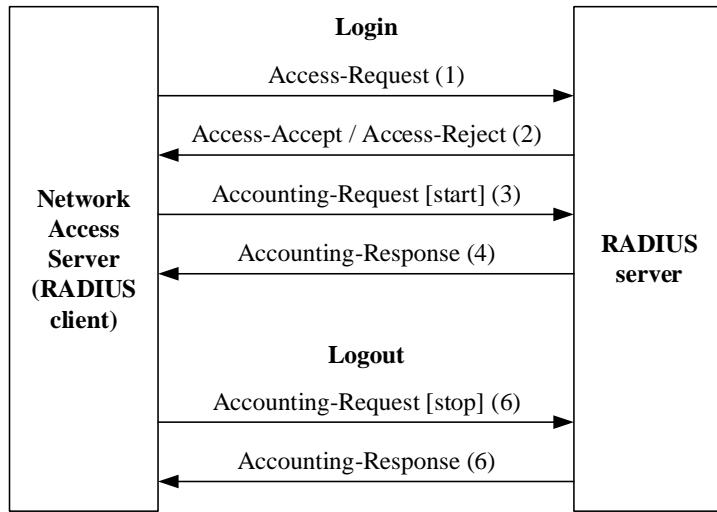
Protokol ketiga adalah Diameter yang merupakan pengembangan dari RADIUS. Meski merupakan pengembangan yang lebih baru, pasar korporat enggan mengadopsi Diameter, sehingga tidak banyak perangkat jaringan yang mendukung protokol ini.

Demi dukungan perangkat yang luas, maka pada tugas akhir ini RADIUS dipilih sebagai protokol yang digunakan untuk komunikasi PEP dengan PDP.

II.1.8.3 *Remote Authentication Dial In User Service (RADIUS)*

Protokol ini berbasis klien-server (Convery, 2007b). Pada protokol ini, PEP mengirimkan kredensial pengguna dan informasi koneksi lainnya ke PDP. PDP melakukan autentikasi dan otorisasi terhadap permintaan PEP, kemudian mengirimkan respons kepada PEP. PEP kemudian mengirimkan pesan akuntansi kepada PDP. Alur komunikasi ketika seorang pengguna mengakses jaringan hingga keluar dari jaringan umumnya terjadi sebagaimana yang digambarkan oleh Feng (2009) pada Gambar II.12.

Studi Agarwal dan Leonetti (2013) yang akan dibahas pada subbab II.2 membahas mengenai metode autentikasi pada *transparent web proxy*. Pada studi Agarwal dan Leonetti, *web proxy* yang digunakan melakukan *lookup* terhadap alamat IP yang melakukan akses web. *Lookup* dilakukan untuk mengetahui identitas pengguna. Identitas ini tersimpan dalam sebuah *database* terpusat. Namun, mekanisme autentikasi pada studi Agarwal dan Leonetti tidak menggunakan protokol yang standar.



Gambar II.12. Alur komunikasi umum di RADIUS

Jika menggunakan protokol RADIUS, alamat IP yang dimaksud akan dikirimkan oleh PEP dalam pesan akuntansi (pada Gambar II.12 ditunjukkan pada pesan ke-3). Pada standar RADIUS Accounting, PEP dapat mengirimkan alamat IP yang benar-benar digunakan oleh klien pada atribut bernama `Framed-IP-Address`, sedangkan identitas pengguna dikirimkan pada atribut bernama `User-Name` (Rigney, 2000). Berbekal dua atribut yang disimpan pada *accounting and reporting system* yang tersedia di jaringan, *web proxy* dapat melakukan *lookup* untuk menentukan identitas pengguna.

II.1.9 Confusion Matrix

Confusion matrix adalah matriks dua dimensi yang merepresentasikan performa klasifikasi dari sebuah pengklasifikasi berdasarkan suatu data uji (Ting, 2017). Umumnya, *confusion matrix* menggunakan dua kelas, yaitu *positive class* dan *negative class*. Empat elemen pada matriks yang terbentuk dari dua kelas itu kemudian dinyatakan sebagai *true positive*, *false positive*, *true negative*, dan *false negative*. Hal ini digambarkan pada Gambar II.13 (Ting, 2017).

Ukuran performa klasifikasi kemudian dinyatakan dengan tiga nilai berikut:

$$Recall = \frac{TP}{TP + FN} \quad (\text{II.1})$$

		Assigned class	
		Positive	Negative
Actual class	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Gambar II.13. *Confusion matrix*

$$Precision = \frac{TP}{TP + FP} \quad (\text{II.2})$$

$$F_1 score = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} \quad (\text{II.3})$$

Ketiga nilai akan berada di rentang $0 \leq recall, precision, F_1 score \leq 1$.

Dalam kasus khusus ketika TP bernilai nol, maka:

- $recall, precision$, dan $F_1 score$ akan bernilai 1 jika kedua nilai dari FP dan FN bernilai nol; dan
- $recall, precision$, dan $F_1 score$ akan bernilai 0 jika salah satu atau kedua nilai dari FP dan FN bernilai lebih dari nol.

Pada penelitian ini, *confusion matrix* digunakan untuk mengukur akurasi dari metode identifikasi *hostname* yang dibandingkan. Akurasi berupa nilai $F_1 score$ yang diperoleh dari nilai *recall* dan *precision*.

II.1.10 Koefisien Korelasi Pearson

Suatu variabel bebas X dapat berkorelasi dengan suatu variabel terikat Y (Suprapto, 2000). Korelasi kedua variabel ini bisa bernilai positif atau negatif. Korelasinya dikatakan positif apabila kenaikan/penurunan X umumnya diikuti oleh kenaikan/penurunan Y , dan dikatakan negatif apabila terjadi kebalikannya.

Kuat dan tidaknya hubungan antara X dan Y apabila dapat dinyatakan dengan fungsi linear (paling tidak mendekati), diukur dengan suatu nilai yang disebut koefisien korelasi (r). Nilai r berada pada rentang $-1 \leq r \leq 1$. Jika nilai r mendekati 1, maka hubungannya semakin kuat dan positif; jika nilai r mendekati -1, maka

hubungannya semakin kuat dan negatif; jika nilai r mendekati 0, maka hubungannya lemah sekali.

Kontribusi dari X terhadap naik-turunnya Y kemudian dihitung melalui suatu koefisien penentuan (KP) dengan rumus berikut:

$$KP = r^2 \quad (\text{II.4})$$

Sedangkan nilai r dapat dihitung sebagai berikut:

$$r = \frac{n \sum X_i Y_i - \sum X_i \sum Y_i}{\sqrt{n \sum X_i^2 - (\sum X_i)^2} \sqrt{n \sum Y_i^2 - (\sum Y_i)^2}} \quad (\text{II.5})$$

Kedua rumus tersebut disebut sebagai koefisien korelasi Pearson. Pada penelitian ini, koefisien korelasi Pearson digunakan untuk menentukan besaran pengaruh variabel bebas terhadap variabel terikat yang ada.

II.2 Karya Ilmiah Sejenis Sebelumnya

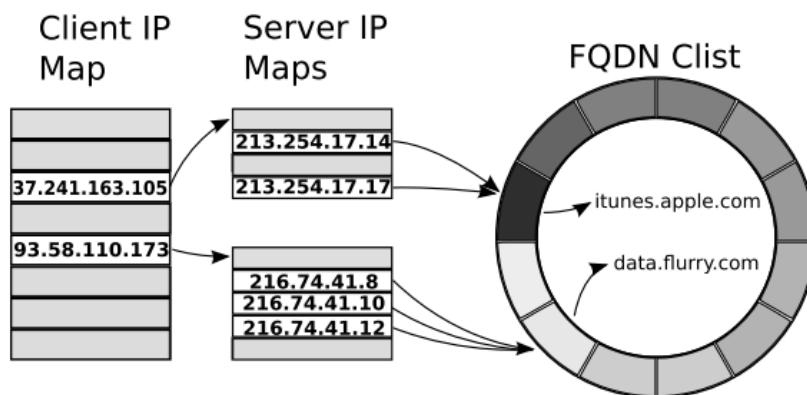
Hal-hal yang dijelaskan pada subbab II.1 merupakan teori-teori fundamental yang terkait dengan akses web. Untuk mengetahui bagaimana cara untuk mengidentifikasi *hostname* pada akses web dengan protokol HTTPS, perlu ditinjau studi-studi terkini.

Bermudez *et al.* (2012) melakukan studi untuk menangkap berbagai fenomena yang ada di dunia internet saat ini. Untuk melakukan studi tersebut, Bermudez *et al.* melakukan *traffic capturing* dari ISP besar di Eropa dan Amerika Utara. Namun, peningkatan penggunaan protokol HTTPS juga menyulitkan Bermudez *et al.* untuk mengetahui secara pasti akses yang terekam, sehingga Bermudez *et al.* menggunakan DNS untuk mengetahui akses yang dilakukan.

Bermudez *et al.* kemudian mengungkapkan metode pertama, yaitu melakukan *reverse lookup* terhadap alamat IP berdasarkan entri pada DNS. *Reverse lookup* dilakukan dengan mencari *hostname* pada entri berjenis PTR dari alamat IP yang ada di DNS. Mekanisme *reverse lookup* ini dijelaskan pada subbab II.1.6. Dari *hostname* yang sudah diperoleh, akses yang terekam kemudian dapat diidentifikasi.

Bermudez *et al.* kemudian mengusulkan metode kedua yang dinamai DN-Hunter. Metode ini tidak melakukan *reverse lookup* pada entri berjenis PTR, melainkan merekam seluruh *query* DNS yang dilakukan oleh pengguna terhadap server DNS di dalam jaringan.

Lebih detail, metode ini merekam alamat IP pengguna beserta *hostname* yang dicari oleh pengguna. Ketika DNS mengembalikan alamat IP dari *hostname* yang dicari, alamat-alamat IP ini juga disimpan. Penyimpanan ini memiliki struktur data yang digambarkan oleh Bermudez *et al.* (2012) pada Gambar II.14. Pada struktur data yang digambarkan tersebut, *transparent web proxy* nantinya bisa menemukan bahwa jika alamat IP 93.58.110.173 melakukan akses ke alamat IP 216.74.41.8, maka *hostname* dari akses tersebut adalah `data.flurry.com`.



Gambar II.14. Struktur data pada metode kedua

Foremski, Callegari dan Pagano (2014) pada studi terpisah kemudian juga mempublikasikan sistem identifikasi akses mereka yang bernama DNS-Class. Meski dilakukan dalam studi terpisah, Foremski, Callegari dan Pagano kemudian mengakui studi Bermudez *et al.* sebagai studi yang menghasilkan produk yang mirip dengan DNS-Class. Perbedaannya, Bermudez *et al.* mengembangkan DN-Hunter untuk keperluan analisis rekaman data ISP mereka tanpa sengaja; Foremski, Callegari dan Pagano dari awal memang bertujuan membangun sistem identifikasi menggunakan DNS.

Rao (2013) pada tesisnya melakukan studi untuk mengetahui bagaimana perilaku aplikasi yang mengakses internet pada perangkat *mobile*. Untuk hal tersebut, Rao

perlu merekam akses yang dilakukan dari perangkat *mobile* menggunakan protokol HTTPS. Rao kemudian menggunakan metode ketiga dan keempat yang akan dikaji pada tugas akhir ini, yaitu menggunakan atribut CN pada sertifikat TLS serta menggunakan SNI.

SNI digunakan oleh Rao sebagai informasi utama untuk menentukan *hostname* dari akses dengan protokol HTTPS. Namun, Rao tidak berharap banyak pada metode ini karena pada saat studi Rao dilakukan penggunaan SNI masih sangat terbatas. Rao kemudian menggunakan atribut CN pada sertifikat yang diberikan server pada protokol TLS. Atribut CN mengandung informasi *hostname* utama apa yang diwakili oleh sertifikat tersebut.

Selain menggunakan SNI dan atribut CN pada sertifikat TLS, Rao juga menggunakan metode DNS yang mirip dengan metode yang dipublikasikan Bermudez *et al.* (2012). Perbedaannya, ketika Bermudez *et al.* menggunakan SNI sebagai metode utama, Rao hanya menggunakan metode DNS ketika SNI dan atribut CN tidak berhasil digunakan untuk mengidentifikasi akses dengan protokol HTTPS tersebut.

Shbair (2017) kemudian pada tesisnya melakukan studi untuk mengidentifikasi akses dengan protokol HTTPS. Metode-metode yang ada dikumpulkan oleh Shbair dan dikelompokkan sebagai berikut:

1. *website fingerprinting*. Metode ini mampu mengidentifikasi secara spesifik halaman mana yang dikunjungi dari suatu situs tertentu. Namun, metode ini hanya dapat mengidentifikasi *website* yang spesifik (misalnya, dapat mengetahui halaman spesifik apa yang dibuka dari suatu *website*, namun tidak dapat mengetahui sama sekali kegiatan yang dilakukan di *website* lain yang belum dilakukan *fingerprinting*);
2. *machine learning*. Studi-studi yang dikelompokkan pada metode ini menggunakan Support Vector Machine, Markov Chain, *state machine*, dan statistik untuk mengidentifikasi layanan email, layanan *video streaming*, *keyword* pencarian yang dimasukkan di mesin pencari, dan penggunaan Google Maps. Perilaku metode yang dikelompokkan pada *machine learning* ini juga

- sama dengan *website fingerprinting*, yaitu hanya dapat mengidentifikasi *website* yang sudah di-*training* terlebih dahulu;
3. metode berbasis DNS dan alamat IP. Shbair mengungkapkan metode *reverse lookup* terhadap entri DNS pada bagian ini. Shbair kemudian mengungkapkan bahwa *reverse lookup* tidak efektif karena fenomena *virtual hosting*. Sayangnya, Shbair tidak mengulas metode yang dilakukan Bermudez *et al.* (2012) dan Foremski, Callegari dan Pagano (2014) yang memberikan alternatif baru untuk identifikasi *hostname* menggunakan DNS;
 4. berdasarkan sertifikat yang diberikan server ketika melakukan *handshake* pada protokol TLS (digambarkan oleh Shbair pada Gambar II.5). Shbair menyatakan metode ini tidak efektif karena banyak penyedia layanan menggunakan satu sertifikat untuk banyak layanan, terlepas dari atribut manapun yang akan digunakan. Sayangnya, Shbair tidak merujuk pada studi manapun, terutama studi Rao (2013) yang menggunakan sertifikat TLS untuk melakukan identifikasi;
 5. Server Name Indication (SNI). Metode ini mulai berkembang sejak tahun 2015 untuk mengidentifikasi *hostname* tujuan berdasarkan sebuah *field* yang dipertukarkan ketika melakukan *handshake* pada protokol TLS (lihat Gambar II.5). Shbair mengungkapkan bahwa metode ini praktis digunakan, banyak diterapkan pada solusi *firewall* yang canggih, dan belakangan dimanfaatkan untuk melakukan *pre-processing* sebelum metode seperti *machine learning* diterapkan lebih jauh.

Dari pengelompokan di atas, Shbair langsung fokus pada metode SNI. Sejauh yang diketahui hingga saat ini, belum ada studi yang secara langsung membandingkan keempat metode tersebut dalam satu waktu. Tugas akhir ini diharapkan dapat memberikan pembahasan mengenai keempat metode tersebut dan membandingkannya dalam konteks penggunaan di Politeknik Negeri Bandung.

Sebelumnya dijelaskan bahwa SNI merupakan *extension* dari protokol TLS (lihat subbab II.1.5). Dengan demikian, secara prinsip terbuka kemungkinan bahwa ada klien dan/atau server yang tidak mendukung penggunaan SNI. Kekhawatiran ini kemudian dijawab oleh studi Nygren (2017), peneliti Akamai yang menyatakan

bahwa tingkat penggunaan SNI pada akses web menggunakan protokol HTTPS telah mencapai 99,4% pada bulan Maret 2017. Peningkatan protokol HTTPS juga didorong oleh *requirement* protokol HTTP/2 yang mensyaratkan klien yang akan menggunakan protokol tersebut harus menggunakan HTTPS ketika menginisiasi koneksi dengan server.

Meski demikian, studi Nygren tidak menjelaskan apakah penggunaan SNI dengan tingkat 99,4% itu hanya mencakup *web browser* saja atau juga berlaku pada seluruh jenis aplikasi yang ada. Karena di Politeknik Negeri Bandung tidak hanya menggunakan *web browser*, maka percobaan tetap perlu dilakukan untuk mengetahui dukungan SNI pada aplikasi-aplikasi lain yang digunakan itu.

Di luar studi mengenai metode identifikasi *hostname*, studi Agarwal dan Leonetti (2013) mengenai penerapan autentikasi pada *transparent web proxy* juga ditelaah. Studi Agarwal dan Leonetti ini relevan karena seperti ditunjukkan pada Tabel II.2, *transparent web proxy* tidak dapat berlaku sebagai *authenticator* (Rabinovich dan Spatscheck, 2001).

Agarwal dan Leonetti merancang mekanisme autentikasi pengguna pada *transparent web proxy* memanfaatkan skrip dalam bahasa Perl. Mekanisme yang dibangun Agarwal dan Leonetti mencatat semua pengguna yang sudah dilakukan autentikasi berdasarkan alamat IP mereka dalam suatu *database* terpusat. *Web proxy* kemudian akan melakukan *lookup* terhadap alamat IP yang melakukan akses web ke *database* terpusat ini, sehingga akses web yang dilakukan juga dapat dilakukan autentikasi oleh *web proxy*. Agarwal dan Leonetti bahkan melakukan implementasi mekanisme ini menggunakan Squid, *web proxy* yang juga digunakan di Politeknik Negeri Bandung.

Meski demikian, model yang dihasilkan Agarwal dan Leonetti belum dapat melakukan identifikasi akses web melalui protokol HTTPS. Selain itu, model Agarwal dan Leonetti juga tidak menggunakan protokol autentikasi standar yang diungkapkan oleh Convery (2007a). Dengan demikian, hasil dari tugas akhir ini berkontribusi pada studi Agarwal dan Leonetti.

Slameta (2013) meninjau infrastruktur jaringan komputer Politeknik Negeri Bandung. Menurut Slameta, topologi model jaringan *existing* saat ini (dibahas lebih lanjut pada subbab V.1.2) perlu diganti menjadi model jaringan VLAN. Model jaringan ini memberikan unjuk kerja yang lebih baik. Rekomendasi Slameta ini akan menjadi dasar bagi penerapan segmentasi jaringan di model yang diusulkan pada tugas akhir ini.

BAB III

METODOLOGI PENELITIAN

Bab ini menjelaskan metodologi penelitian tugas akhir secara rinci, meliputi jenis, subjek, objek, variabel, data, dan tahapan penelitian.

III.1 Jenis Penelitian

Penelitian ini dilakukan dengan pendekatan kuantitatif dengan teknik penelitian eksperimental. Eksperimen dilakukan untuk membandingkan akurasi dari metode-metode identifikasi *hostname* berikut, sehingga dapat disimpulkan mana metode yang dapat menghasilkan identifikasi yang serupa dengan hasil identifikasi *explicit web proxy* yang digunakan di Politeknik Negeri Bandung:

- melakukan *reverse lookup* terhadap alamat IP berdasarkan entri pada DNS (Bermudez *et al.*, 2012);
- melakukan *reverse lookup* terhadap alamat IP berdasarkan rekaman *query* dari pengguna pada server DNS milik organisasi (Bermudez *et al.*, 2012; Foremski, Callegari dan Pagano, 2014);
- menggunakan SNI, ekstensi dari protokol TLS, mensyaratkan penggunaan aplikasi yang mendukung SNI (Rao, 2013; Shbair, 2017); dan
- menggunakan atribut CN pada sertifikat yang diberikan server pada protokol TLS (Rao, 2013; Shbair, 2017).

Untuk mengetahui hal tersebut, disimulasikan pengguna yang melakukan akses ke internet. Hal yang disimulasikan adalah aplikasi pengguna dan *transparent web proxy* yang bekerja dengan metode-metode identifikasi *hostname* yang dibandingkan. Sedangkan pengguna-pengguna disimulasikan berdasarkan log *explicit web proxy* yang dijadikan data penelitian (dijelaskan pada subbab III.5).

III.2 Subjek Penelitian

Subjek penelitian ini adalah hasil identifikasi dari metode-metode identifikasi *hostname* yang dibandingkan. Hasil identifikasi kemudian dikuantifikasi dengan menggunakan nilai *true positive*, *false positive*, dan *false negative* yang kemudian

dihitung menjadi nilai F_1 score. Nilai F_1 score dikelompokkan berdasarkan *hostname* masing-masing yang kemudian dianalisis hubungannya dengan variabel-variabel bebas penelitian (dijelaskan pada subbab III.4).

III.3 Objek Penelitian

Objek penelitian ini adalah *hostname-hostname* yang terdapat pada log yang dijadikan data penelitian (dijelaskan pada subbab III.5). *Hostname-hostname* tersebut dikelompokkan berdasarkan variabel bebas penelitian (dijelaskan pada subbab III.4).

III.4 Variabel Penelitian

Variabel-variabel yang terlibat dalam penelitian ini adalah:

1. variabel bebas (*independent variables*). Terdapat lima variabel bebas pada penelitian ini. Variabel pertama akan dimanipulasi, sedangkan keempat variabel lainnya merupakan faktor-faktor yang digunakan untuk mengevaluasi mana metode yang paling unggul di berbagai kondisi. Kelima variabel tersebut adalah:
 - a. apakah aplikasi pengguna mendukung SNI atau tidak.

Karena salah satu metode yang dibandingkan adalah metode SNI, maka disimulasikan dua jenis aplikasi, yaitu aplikasi pengguna yang mendukung SNI atau tidak. Aplikasi pengguna yang mendukung SNI ditandai dengan akses yang mengandung informasi SNI ketika melakukan *handshake* protokol TCP;

- b. jumlah alamat IP yang diberikan dari hasil *lookup hostname* pada DNS.
Sebagaimana yang dijelaskan pada subbab II.1.6, *lookup* terhadap sebuah *hostname* bisa mengembalikan lebih dari satu alamat IP. Menurut Callahan, Allman dan Rabinovich (2013), *hostname* dengan lebih dari satu alamat IP yang terdaftar pada DNS bertujuan untuk memberikan alternatif bagi klien jika gagal mengakses salah satu alamat IP yang ada. Selain itu, dengan lebih dari satu alamat IP, server DNS akan mengacak urutan dari alamat-alamat IP, sehingga menimbulkan perilaku *load balancing*.

Variabel ini merepresentasikan skala penyedia layanan yang diakses; penyedia layanan berskala besar seperti Google membutuhkan *high availability* yang dapat diperoleh dari *load balancing*, sehingga meningkatkan kemungkinan pengembalian alamat IP lebih dari satu. Pengelompokan *hostname* pada variabel ini dilakukan berdasarkan hasil *lookup* pada DNS;

- c. jumlah *mutual hostname* (*hostname* lain yang berbagi alamat IP).

Lookup terhadap dua atau lebih *hostname* yang berbeda dapat mengembalikan alamat IP yang sama. Sebagai contoh, alamat IP 23.220.203.17 dan 23.220.203.18 sama-sama dikembalikan dari hasil *lookup* terhadap *hostname* fcdn-sphotos-f-a.akamaihd.net, fcdn-sphotos-e-a.akamaihd.net, dan fcdn-sphotos-d-a.akamaihd.net. Dengan demikian, masing-masing *hostname* tersebut memiliki dua *hostname* lain yang berbagi alamat IP;

Variabel ini merepresentasikan fenomena *virtual hosting* (satu alamat IP digunakan untuk melakukan *hosting* banyak layanan) yang dapat mempengaruhi akurasi identifikasi *hostname* tersebut. Pengelompokan *hostname* pada variabel ini dilakukan berdasarkan hasil *lookup* pada DNS;

- d. jumlah alamat IP pengakses *hostname*.

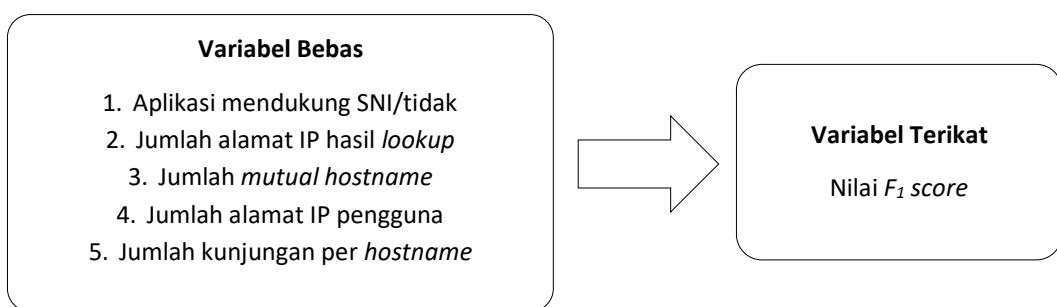
Variabel ini merepresentasikan popularitas dari *hostname* tersebut. Semakin banyak jumlah alamat IP pengguna yang mengakses *hostname* tersebut, maka *hostname* tersebut semakin populer. *Hostname* yang populer penting untuk bisa diidentifikasi secara akurat oleh metode identifikasi yang sedang dibandingkan, karena kegagalan identifikasi *hostname* yang populer dapat memberikan informasi yang salah kepada PSI. Pengelompokan *hostname* pada variabel ini dilakukan berdasarkan log pada data penelitian;

- e. jumlah kunjungan per *hostname* pada log.

Variabel ini merepresentasikan frekuensi akses suatu *hostname* oleh pengguna di lingkungan Politeknik Negeri Bandung. Sama halnya dengan variabel jumlah alamat IP pengakses *hostname*, suatu *hostname* yang sering

- diakses merupakan *hostname* yang penting untuk bisa diidentifikasi dengan tepat oleh metode yang dipilih;
2. variabel terikat (*dependent variable*), yaitu nilai F_1 score yang diperoleh dari nilai *true positive*, *false positive*, dan *false negative*. Nilai ini merepresentasikan akurasi identifikasi *hostname* dari metode yang sedang dibandingkan.

Variabel-variabel tersebut didapatkan untuk setiap *hostname* yang terdapat pada data penelitian. Hubungan variabel-variabel tersebut digambarkan pada Gambar III.1.



Gambar III.1. Hubungan antar variabel penelitian

III.5 Data Penelitian

Data utama pada penelitian ini adalah log dari *explicit web proxy* PSI yang melayani jaringan nirkabel. Log memiliki 20.985.562 baris yang dimulai pada 6 Juli 2014 dan berakhir pada 19 Januari 2018. Log yang diberikan merupakan log berformat standar Squid, yaitu *software web proxy* yang digunakan di Politeknik Negeri Bandung. Log ini dijadikan sebagai *ground truth* untuk menentukan akurasi identifikasi *hostname* metode-metode yang dibandingkan.

Log tersebut mengandung data yang sensitif, yaitu data identitas berupa *username* dari pengguna yang mengakses. Untuk menjaga kerahasiaan dan privasi, analisis tidak dilakukan untuk mengidentifikasi perilaku pengguna yang terekam pada log. Laporan tugas akhir ini juga tidak mengungkap hal tersebut karena tidak terkait dengan kajian yang dilakukan.

Data lainnya yang digunakan pada penelitian ini adalah nama-nama aplikasi yang digunakan sebagai perangkat pembelajaran di Jurusan Teknik Komputer dan

Informatika. Aplikasi digunakan untuk memastikan bahwa metode terpilih benar-benar dapat mengidentifikasi seluruh aplikasi yang dijadikan sampel dari penggunaan di Politeknik Negeri Bandung. Terdapat tiga belas aplikasi yang digunakan.

Penjelasan lebih lengkap mengenai data penelitian kemudian dijelaskan lebih lanjut pada subbab III.6.3 dan subbab III.6.4.

III.6 Tahapan Penelitian

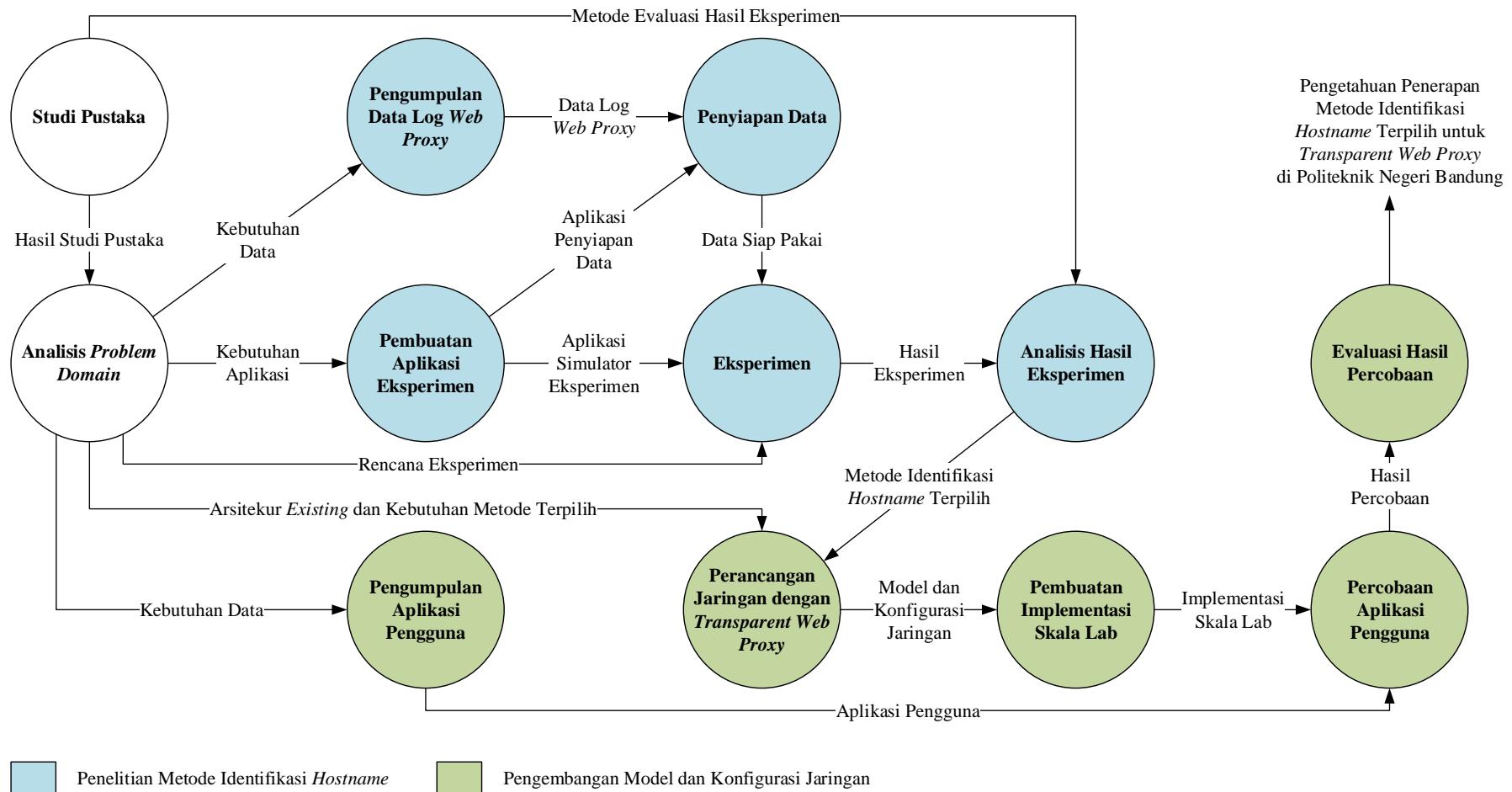
Penelitian ini dilakukan dalam dua belas tahapan yang secara umum digambarkan pada Gambar III.2. Penelitian ini juga akan diikuti dengan pengembangan model dan konfigurasi jaringan. Subbab ini kemudian menjelaskan lebih rinci mengenai dua belas tahapan tersebut.

III.6.1 Studi Pustaka

Studi pustaka bertujuan mencari dan mempelajari sumber-sumber yang relevan dan terpercaya (buku teks, *paper* di jurnal dan prosiding, dan dokumen standar teknis seperti dokumen RFC yang berlaku di industri) terkait permasalahan yang dihadapi sebagai landasan teori dan bahan kajian dari penelitian ini. Hal-hal yang distudi diuraikan pada Tabel III.1.

Tabel III.1. Subjek studi pustaka

Subjek Studi	Keperluan Studi
Protokol HTTPS dan cara-cara identifikasinya	Untuk mengetahui metode identifikasi akses web dengan protokol HTTPS yang terpilih beserta cara kerjanya sebagai bahan kajian
AAA jaringan komputer	Untuk memilih protokol AAA yang akan digunakan pada perancangan jaringan baru dengan <i>transparent web proxy</i>
Penggunaan <i>web proxy</i> untuk mengelola akses web ke internet	Mempelajari jenis-jenis <i>deployment web proxy</i> serta hal-hal yang perlu diperhatikan dari <i>web proxy</i>
Rekomendasi arsitektur jaringan Politeknik Negeri Bandung	Arsitektur jaringan yang menjadi asumsi untuk membangun model usulan
<i>Confusion matrix</i>	Sebagai alat bantu untuk melakukan analisis hasil eksperimen
Koefisien korelasi Pearson	Sebagai alat bantu untuk melakukan analisis korelasi antara variabel bebas dengan variabel terikat



Gambar III.2. Tahapan penelitian

Pengetahuan yang diperoleh dari tahap studi pustaka ini akan digunakan untuk menganalisis *problem domain* yang ada pada tugas akhir ini.

III.6.2 Analisis *Problem Domain*

Pada tahap ini dilakukan analisis dan pembahasan permasalahan penelitian yang meliputi:

1. analisis permasalahan identifikasi akses HTTPS serta metode identifikasi *hostname* pada akses HTTPS yang diungkapkan pada subbab II.2;
2. analisis kebutuhan aplikasi yang digunakan untuk eksperimen;
3. analisis permasalahan pada latar belakang tugas akhir (terkait penggunaan *explicit web proxy* di Politeknik Negeri Bandung);
4. analisis topologi jaringan Politeknik Negeri Bandung; serta
5. analisis pengelolaan akses ke internet yang dilakukan oleh PSI.

Selain berdasarkan studi pustaka, data-data tambahan juga dikumpulkan untuk melakukan analisis terkait kejadian di Politeknik Negeri Bandung. Data-data tersebut ditunjukkan pada Tabel III.2.

Tabel III.2. Data yang akan dikumpulkan

Data	Sumber dan Cara Perolehan
Topologi <i>existing</i> jaringan Politeknik Negeri Bandung yang berkaitan dengan akses web ke internet	PSI, dengan wawancara serta melakukan <i>tracing</i> terhadap konfigurasi perangkat jaringan terkait (utamanya <i>router</i> terluar dan <i>firewall</i> pada <i>gateway</i> internal)
Konfigurasi <i>web proxy</i> (baik <i>explicit</i> maupun <i>transparent</i>) <i>existing</i> di Politeknik Negeri Bandung	PSI, dengan melakukan <i>tracing</i> terhadap konfigurasi dari <i>web proxy</i> yang sedang berjalan
Kebutuhan log yang dihasilkan dari <i>web proxy</i> sebagai bahan analisis PSI	PSI, dengan mengambil arsip log dari <i>web proxy</i> yang sedang berjalan, serta wawancara dengan staf PSI
Kebutuhan pengelolaan akses web di Politeknik Negeri Bandung	PSI, melalui wawancara serta dokumen standar pengelolaan yang berlaku

Analisis-analisis tersebut pada laporan tugas akhir ini disebar ke dalam dua bab, yaitu bab IV dan bab V, sesuai dengan relevansinya pada bab tersebut.

III.6.3 Pengumpulan Data Log Web Proxy

Data yang akan digunakan sebagai bahan eksperimen adalah log dari *explicit web proxy* PSI yang melayani jaringan nirkabel. Log memiliki 20.985.562 baris yang dimulai pada 6 Juli 2014 dan berakhir pada 19 Januari 2018. Log yang diberikan merupakan log berformat standar Squid, yaitu *software web proxy* yang digunakan di Politeknik Negeri Bandung.

Data yang terdapat pada log terdiri dari:

- waktu *request*;
- berapa lama permintaan menunggu hingga mendapatkan respons dari server tujuan;
- alamat IP pengguna;
- indikator apakah akses diotorisasi atau tidak;
- indikator apakah *response* diambil dari *cache* milik *web proxy* atau tidak;
- protokol akses (HTTP atau bukan);
- kode respons HTTP, jika diakses dengan protokol HTTP;
- ukuran respons;
- kode *method* HTTP, jika diakses dengan protokol HTTP;
- URI dari *resource* yang diakses (jika diakses dengan protokol HTTP) atau *hostname* dan *port* yang dituju (jika diakses dengan protokol HTTPS);
- identitas berupa *username* dari pengguna yang mengakses;
- alamat server yang dituju oleh *web proxy*; dan
- jenis *resource* yang diakses, jika diakses dengan protokol HTTPS.

Contoh log yang diperoleh ditunjukkan pada Gambar III.3.

Log tersebut mengandung data yang sensitif, yaitu data identitas berupa *username* dari pengguna yang mengakses. Alamat IP pengguna yang terekam juga digunakan sebagai variabel dan dibutuhkan pada metode *reverse lookup* berdasarkan rekaman *query*. Untuk kerahasiaan data, analisis pada tugas akhir ini tidak menyentuh aspek personal tersebut. Alamat IP pengguna juga tidak ditampilkan pada laporan tugas akhir ini.

```
1404595783.383 30432 [alamat IP] TCP_MISS/200 839 CONNECT twitter.com:443
[username] DIRECT/199.59.149.198 -
1404595783.830 30974 [alamat IP] TCP_MISS/200 6489 CONNECT twitter.com:443
[username] DIRECT/199.59.148.10 -
1404595785.175 30686 [alamat IP] TCP_MISS/200 2721 CONNECT
syndication.twitter.com:443 [username] DIRECT/199.59.149.201 -
1404595796.382 116566 [alamat IP] TCP_MISS/200 4711 CONNECT
www.facebook.com:443 [username] DIRECT/173.252.73.52 -
1404595799.644 70919 [alamat IP] TCP_MISS/200 9095 CONNECT fcdn-sphotos-g-
a.akamaihd.net:443 [username] DIRECT/165.254.42.10 -
1404595808.538 833647 [alamat IP] TCP_MISS/200 1364949 CONNECT fcdn-profile-
a.akamaihd.net:443 [username] DIRECT/125.160.16.113 -
1404595809.542 80810 [alamat IP] TCP_MISS/200 39770 CONNECT fcdn-sphotos-f-
a.akamaihd.net:443 [username] DIRECT/125.160.16.96 -
1404595814.633 74997 [alamat IP] TCP_MISS/200 16417 CONNECT fcdn-sphotos-h-
a.akamaihd.net:443 [username] DIRECT/165.254.42.8 -
1404595817.166 117577 [alamat IP] TCP_MISS/200 10679 CONNECT fcdn-sphotos-c-
a.akamaihd.net:443 [username] DIRECT/96.6.122.80 -
1404595817.374 201 [alamat IP] TCP_MISS/200 29649 GET
http://img.berniaga.co.id/images/55/5588965364.jpg [username]
DIRECT/202.158.20.54 image/jpeg
```

Gambar III.3. Contoh log dari *explicit web proxy*

Hostname pada log tersebut kemudian dijadikan sebagai *ground truth* untuk menentukan akurasi identifikasi *hostname* metode-metode yang dibandingkan.

III.6.4 Pengumpulan Aplikasi Pengguna

Data lainnya yang digunakan pada penelitian ini adalah nama-nama aplikasi yang digunakan sebagai perangkat pembelajaran di Jurusan Teknik Komputer dan Informatika. Aplikasi digunakan untuk memastikan bahwa metode terpilih benar-benar dapat mengidentifikasi seluruh aplikasi yang dijadikan sampel dari penggunaan di Politeknik Negeri Bandung.

Data ini dikumpulkan dengan cara menghubungi perwakilan kelas-kelas yang ada di Jurusan Teknik Komputer dan Informatika. Seluruh perwakilan kelas dihubungi, kecuali kelas yang melaksanakan tugas akhir. Perwakilan kelas kemudian diminta menyebutkan aplikasi-aplikasi yang digunakan selama satu tahun akademik terakhir, yaitu tahun akademik 2017/2018, baik semester ganjil maupun semester genap. Aplikasi yang dikumpulkan adalah aplikasi yang membutuhkan akses ke internet dan bukan buatan mahasiswa itu sendiri.

Dari para perwakilan kelas, diperoleh lima puluh aplikasi yang digunakan sepanjang tahun akademik 2017/2018, dengan tiga belas aplikasi di antaranya membutuhkan akses web ke internet. Aplikasi tersebut diuraikan pada Tabel III.3.

Tabel III.3. Daftar aplikasi pengguna yang membutuhkan akses internet

Nama Aplikasi	Mata Kuliah
Android Studio	Pemrograman Perangkat Bergerak (2/D3-TI) dan Perancangan Antarmuka (2/D4-TI)
Docker	Pemrograman Perangkat Lunak Berorientasi Obyek (3/D4-TI)
Dropbox	Digunakan untuk <i>support</i> perkuliahan
Git	Proyek 1 (1/D4-TI), Pemrograman Perangkat Bergerak (2/D3-TI), Proyek Perangkat Lunak 3 (2/D3-TI), dan Proyek Perangkat Lunak 4 (2/D3-TI)
GitHub Desktop	Proyek 1 (1/D4-TI), Pemrograman Perangkat Bergerak (2/D3-TI), Proyek Perangkat Lunak 3 (2/D3-TI), dan Proyek Perangkat Lunak 4 (2/D3-TI)
GitKraken	Proyek 1 (1/D4-TI), Pemrograman Perangkat Bergerak (2/D3-TI), Proyek Perangkat Lunak 3 (2/D3-TI), dan Proyek Perangkat Lunak 4 (2/D3-TI)
Google Chrome	<i>Web browser</i> , digunakan untuk <i>support</i> perkuliahan
LINE	Digunakan untuk <i>support</i> perkuliahan
Mozilla Firefox	<i>Web browser</i> , digunakan untuk <i>support</i> perkuliahan
NetBeans	Pemrograman Berorientasi Obyek (2/D3-TI) dan Proyek Perangkat Lunak 3 (2/D3-TI)
NPM Package Manager	Pengembangan Web (3/D4-TI)
PyCharm	Proyek Perangkat Lunak 4 (2/D3)
Visual Studio	Pemrograman Perangkat Lunak Berorientasi Obyek (3/D4-TI)

III.6.5 Pembuatan Aplikasi Eksperimen

Karena eksperimen ini menyimulasikan pengguna yang melakukan akses internet dan diproses oleh *transparent web proxy*, maka perlu dibuat aplikasi untuk hal tersebut. Aplikasi eksperimen utama akan dibuat berdasarkan analisis *problem domain* terkait keempat metode identifikasi *hostname* yang ada (dijelaskan pada subbab IV.1.1).

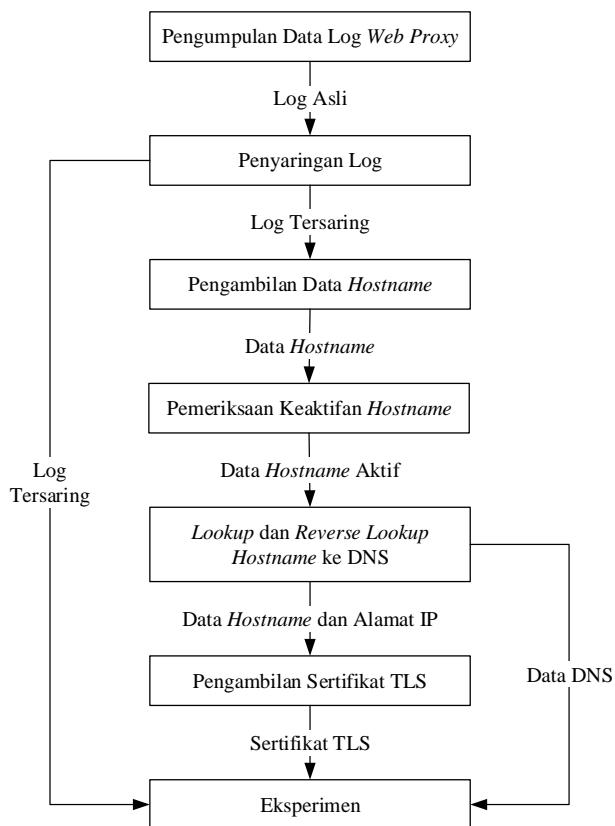
Selain aplikasi untuk eksperimen utama, aplikasi-aplikasi yang lebih kecil juga dibuat untuk masing-masing tahap penyiapan data (dijelaskan pada subbab III.6.6) dan untuk memproses delapan buah log yang dikeluarkan dari aplikasi eksperimen (dijelaskan pada akhir subbab III.6.7.3).

Aplikasi-aplikasi ini dikembangkan sesuai kebutuhan berdasarkan proses yang dijelaskan pada masing-masing subbab. Aplikasi-aplikasi ini juga dikembangkan dengan prinsip pengembangan perangkat lunak di lingkungan UNIX, “*do one thing and do it well*,” yang oleh Raymond (2003) diformalkan sebagai *rule of modularity*.

Seluruh aplikasi pada eksperimen ini dibuat menggunakan bahasa pemrograman Python 3.5.2 dengan implementasi CPython.

III.6.6 Penyiapan Data

Sebelum data utama dapat digunakan pada eksperimen, data utama perlu disiapkan terlebih dahulu. Penyiapan data dilakukan untuk memastikan bahwa data bersih dari *noise* dan siap pakai. Dalam penyiapan data juga dilakukan pengambilan data tambahan berdasarkan data utama yang diperoleh pada subbab III.6.3. Penyiapan data dilakukan dalam lima tahap yang digambarkan pada Gambar III.4 dan dijelaskan pada subbab ini.



Gambar III.4. Alur penyiapan data hingga eksperimen

Untuk memastikan kontrol dan *reproducibility* penelitian ini, maka data yang berasal dari luar (data DNS dan data sertifikat TLS) diambil terlebih dahulu semuanya dan disimpan. Tahap-tahap eksperimen selanjutnya yang membutuhkan data DNS dan data sertifikat TLS kemudian mengacu pada data yang sudah disimpan ini. Hal ini dilakukan dengan menyimulasikan server DNS dan server

TLS dari masing-masing *hostname*. *Transparent web proxy* yang disimulasikan kemudian menggunakan simulator server DNS dan server TLS ini untuk melakukan eksperimen.

Tiga dari lima tahap yang ada, yaitu tahap pemeriksaan keaktifan *hostname*, tahap *lookup* dan *reverse lookup hostname* ke DNS, serta tahap pengambilan sertifikat TLS membutuhkan akses ke internet. Untuk mempercepat proses dan menghindari risiko terganggunya akses internet di Politeknik Negeri Bandung, maka kedua tahap ini dilakukan prosesnya menggunakan *virtual private server* dari layanan Microsoft Azure. Spesifikasi *virtual private server* yang digunakan adalah:

- *instance type* Azure B2S,
- *region* Southeast Asia (berlokasi secara fisik di Singapura),
- CPU Intel® Haswell 2,4 GHz E5-2673 v3 dengan 2 *virtual core*,
- RAM 4 GB,
- *local SSD storage* 8 GB, dan
- sistem operasi Ubuntu 18.04 LTS.

Lokasi dari *virtual private server* di Singapura menyebabkan hasil *lookup* dapat berbeda jika dilakukan dari lokasi lain (misalnya di Bandung). Namun, perbedaan tersebut tidak signifikan mempengaruhi hasil eksperimen pada tugas akhir ini.

III.6.6.1 Penyaringan Log

Log yang diberikan dari PSI merekam seluruh akses, baik menggunakan protokol HTTP maupun HTTPS. Log juga mencatat akses yang tidak menuju *port standar* (80 untuk HTTP dan 443 untuk HTTPS). Untuk menyesuaikan dengan ruang lingkup tugas akhir, log disaring sehingga hanya entri log yang memenuhi kriteria berikut yang digunakan, yaitu:

- akses dengan protokol HTTPS, ditandai dengan *method code* CONNECT);
- akses ke *port standar* HTTPS (443);
- koneksi yang tercatat di log berhasil dilakukan, ditandai dengan kode koneksi 200 (koneksi yang tidak berhasil dilakukan dan tercatat dianggap sebagai *noise* karena tidak dapat disimulasikan pada eksperimen);

- *hostname* yang tercatat bukan merupakan alamat IP (alamat IP yang tercatat berarti bahwa koneksi dari klien memang mengarah langsung ke alamat IP tanpa menyebutkan *hostname*).

Suatu *string* dinyatakan sebagai alamat IP yang valid bila hanya terdiri dari empat buah bilangan bulat dalam rentang 0–255 yang dipisahkan dengan titik.

III.6.6.2 Pengambilan Data *Hostname*

Log yang sudah disaring pada subbab III.6.6.1 kemudian diambil *hostname*-nya saja. Dua atau lebih entri dengan *hostname* yang sama hanya dicatat *hostname*-nya satu kali. Tahap ini menghasilkan daftar *hostname* unik yang ada pada log yang sudah disaring.

III.6.6.3 Pemeriksaan Keaktifan *Hostname*

Daftar *hostname* unik yang dihasilkan pada subbab III.6.6.2 kemudian dicoba diakses satu per satu. Tahap ini perlu dilakukan karena log yang didapatkan pada penelitian ini terakhir kali mencatat pada tanggal 19 Januari 2018, dan ada kemungkinan terdapat entri log yang *hostname*-nya tidak lagi dapat diakses.

Akses dilakukan terhadap *port* 443 dari *hostname* tersebut. Tahap ini hanya memastikan bahwa server dengan *hostname* tersebut masih menerima akses dari pengguna luar.

Tahap ini menambahkan keterangan pada masing-masing *hostname* apakah *hostname* tersebut masih aktif atau tidak.

III.6.6.4 *Lookup* dan *Reverse Lookup* *Hostname* ke DNS

Untuk melakukan simulasi server DNS pada eksperimen, dibutuhkan data hasil *lookup* dan *reverse lookup* dari data seluruh *hostname* aktif yang diperoleh pada subbab III.6.6.3.

Data *lookup* diperoleh dengan melakukan *lookup* untuk setiap *hostname* ke DNS, kemudian menyimpan *hostname* tersebut beserta seluruh alamat IP-nya ke dalam *file*. Sementara data *reverse lookup* diperoleh dengan melakukan *reverse lookup* terhadap setiap alamat IP hasil *lookup* kemudian menyimpannya ke dalam *file*.

Kedua data ini menjadi bagian dari masukan untuk eksperimen.

III.6.6.5 Pengambilan Sertifikat TLS

Untuk melakukan simulasi server TLS pada eksperimen, dibutuhkan data sertifikat TLS dari semua *hostname* dan alamat IP yang mungkin dikunjungi oleh pengguna (yang disimulasikan). Data *hostname* dan alamat IP yang mungkin dikunjungi diperoleh dari data *lookup* yang diperoleh pada subbab III.6.4.

Karena pada eksperimen akan disimulasikan dua jenis aplikasi, yaitu aplikasi yang mendukung SNI dan tidak, maka sertifikat TLS akan diambil dari alamat-alamat IP tersebut dua kali. Pengambilan pertama akan menggunakan SNI, sehingga *hostname* juga turut dikirimkan ke server sebenarnya. Pada alamat IP dengan beberapa *hostname* yang terasosiasi, sertifikat TLS akan dicoba diambil untuk masing-masing *hostname* tersebut. Pengambilan kedua tidak akan menggunakan SNI, sehingga *hostname* tidak turut dikirimkan ke server sebenarnya.

Seluruh sertifikat TLS yang sudah diperoleh kemudian disimpan ke dalam *file* untuk digunakan sebagai bagian dari masukan untuk eksperimen.

III.6.7 Eksperimen

Tahap penyiapan data yang dibahas pada subbab III.6.6 menghasilkan tiga buah data yang digunakan pada eksperimen ini, yaitu log yang sudah tersaring untuk disimulasikan aksesnya, data DNS untuk menyimulasikan server DNS, serta data sertifikat TLS untuk menyimulasikan server-server tujuan.

Eksperimen kemudian dilakukan dengan menyimulasikan *transparent web proxy* yang memproses akses yang dilakukan oleh pengguna. Akses yang dilakukan oleh pengguna disimulasikan dari log yang sudah tersaring. Pada simulasi ini, *transparent web proxy* akan melakukan identifikasi *hostname* dengan keempat metode yang dibandingkan.

Untuk memastikan kontrol dan *reproducibility* penelitian ini, data DNS dan data sertifikat TLS digunakan untuk menyimulasikan server DNS dan server dari masing-masing *hostname* yang akan dihubungi oleh pengguna dan oleh *transparent web proxy*.

III.6.7.1 Tujuan Eksperimen

Eksperimen ini dilakukan untuk mengetahui metode apa yang dapat menghasilkan identifikasi yang serupa dengan hasil identifikasi *explicit web proxy* yang digunakan di Politeknik Negeri Bandung.

III.6.7.2 Alat, Bahan, Perlengkapan, dan Unit Eksperimen

Alat, bahan, perlengkapan, dan unit eksperimen pada penelitian ini adalah sebagai berikut:

1. Alat eksperimen

Alat yang digunakan pada eksperimen ini adalah sebuah *virtual private server* dengan spesifikasi yang sama seperti yang digunakan pada tahap penyiapan data (lihat subbab III.6.6).

2. Bahan eksperimen

Bahan eksperimen ini adalah log yang sudah tersaring untuk disimulasikan aksesnya, data DNS untuk menyimulasikan server DNS, serta data sertifikat TLS untuk menyimulasikan server-server tujuan.

3. Perlengkapan eksperimen

Perlengkapan pada eksperimen ini adalah klien dan server SSH yang digunakan untuk mengakses *virtual private server*. Selain itu, program DB Browser for SQLite juga digunakan untuk melakukan *query* terhadap hasil eksperimen dalam rangka analisis hasil, dan Microsoft Excel digunakan untuk melakukan analisis lebih lanjut (termasuk analisis korelasi variabel bebas dengan variabel terikat) serta pembuatan grafik.

4. Unit eksperimen

Unit eksperimen ini adalah aplikasi simulator *transparent web proxy* dengan empat metode identifikasi *hostname* yang telah dibuat pada subbab III.6.5.

III.6.7.3 Metode Eksperimen

Untuk memenuhi tujuan eksperimen pada subbab III.6.7.1, eksperimen perlu dilakukan sebanyak delapan kali dengan konfigurasi yang ditunjukkan pada Tabel III.4. Delapan kali eksperimen tersebut menggunakan data log yang sama.

Tabel III.4. Konfigurasi eksperimen

Nomor Eksperimen	Metode Identifikasi <i>Hostname</i>	Dukungan SNI pada Klien (variabel bebas)
1	<i>Reverse lookup</i> ke DNS	Aktif
2		Tidak aktif
3	<i>Reverse lookup</i> dari rekaman <i>query</i>	Aktif
4		Tidak aktif
5	SNI pada <i>handshake TLS</i>	Aktif
6		Tidak aktif
7	Atribut CN pada sertifikat TLS	Aktif
8		Tidak aktif

Eksperimen dilakukan dengan menyimulasikan klien-klien yang melakukan akses ke internet berdasarkan log yang menjadi masukan eksperimen dengan alur sebagai berikut untuk setiap entri log:

1. klien menggunakan alamat IP yang tercatat pada entri log, kemudian melakukan akses ke internet berdasarkan *hostname* yang tercatat pada entri log.
 - a. Untuk bisa melakukan akses internet, klien melakukan *lookup* ke server DNS (yang disimulasikan) untuk mendapatkan alamat IP dari *hostname* yang dituju. Alamat IP ini kemudian disimpan di klien selama *time to live* yang dikembalikan dari server DNS (Tanenbaum dan Wetherall, 2011);
 - b. akses disimulasikan dengan sebuah pesan yang berisi alamat IP klien dan alamat IP tujuan;
 - c. jika konfigurasi eksperimen menunjukkan dukungan SNI pada klien dalam keadaan aktif, maka pada pesan yang dikirimkan di poin (b) di atas ditambahkan atribut SNI berupa *hostname* yang didapatkan dari entri log;
2. pesan dari klien yang merepresentasikan akses diterima oleh *transparent web proxy* yang kemudian melakukan identifikasi sesuai dengan metode yang sedang aktif pada konfigurasi eksperimen;
3. *transparent web proxy* menuliskan entri log baru berisi *hostname* yang teridentifikasi berdasarkan metode yang sedang dicoba.

Hasil dari delapan kali eksperimen akan menghasilkan delapan buah log baru yang berisi alamat IP klien, *hostname* yang sebenarnya (berdasarkan *ground truth*), serta *hostname* yang dideteksi berdasarkan metode pada konfigurasi eksperimen

tersebut. Delapan log ini kemudian diproses untuk menghasilkan data yang diilustrasikan pada Tabel III.5.

Pada implementasinya, data pada Tabel III.5 tersebut akan berwujud sebuah dokumen berformat SQLite yang dapat di-*query*. *Query* terhadap dokumen SQLite ini akan dilakukan dalam menganalisis hasil eksperimen.

III.6.8 Analisis Hasil Eksperimen

Hasil eksperimen yang ditunjukkan pada Tabel III.5 akan dianalisis untuk mencari tahu mana metode yang unggul yang dapat mengidentifikasi *hostname* pada protokol HTTPS untuk penggunaan di Politeknik Negeri Bandung. Analisis dilakukan dengan melihat nilai F_1 *score* dari masing-masing metode pada berbagai kondisi variabel bebas.

Selain mencari tahu mana metode yang unggul, analisis juga dilakukan untuk melihat mana variabel bebas yang berpengaruh terhadap nilai F_1 *score*. Analisis dilakukan menggunakan koefisien korelasi dan koefisien pengaruh menggunakan rumus II.4 dan II.5.

Pada akhir analisis, hipotesis yang dinyatakan untuk RQ₁ (lihat subbab I.3) dinyatakan apakah diterima atau ditolak. Hipotesis dinyatakan diterima apabila metode SNI pada *handshake* TLS unggul dalam berbagai kondisi variabel bebas berdasarkan nilai F_1 *score*-nya.

Metode apa pun yang unggul pada eksperimen ini kemudian akan dijadikan sebagai bahan perancangan jaringan dengan *transparent web proxy* yang dilakukan pada subbab III.6.9.

III.6.9 Perancangan Jaringan dengan *Transparent Web Proxy*

Pada tahap ini, arsitektur jaringan yang sudah dianalisis pada subbab III.6.2 dimodifikasi untuk penerapan metode identifikasi *hostname* yang sudah dipilih berdasarkan analisis hasil eksperimen pada subbab III.6.8. Modifikasi akan dibuat sekecil mungkin, sehingga meminimalkan *cost* yang perlu dikeluarkan oleh PSI untuk melakukan implementasi jika model jaringan dan konfigurasi yang dibutuhkan akan diwujudkan di Politeknik Negeri Bandung.

Lebih rinci, hal-hal yang akan dirancang adalah:

- topologi dari pengguna, *web proxy*, PEP, PDP (berupa server RADIUS), dan komponen lain yang dibutuhkan (jika ada) serta hubungannya ke internet;
- alur autentikasi pengguna sejak perangkat bergabung di jaringan hingga dapat mengakses web; serta
- konfigurasi dari *web proxy*, PEP, PDP, komponen lain yang dibutuhkan (jika ada) dan perangkat jaringan yang menghubungkan seluruh komponen tersebut dengan pengguna.

Hasil dari tahap ini adalah model jaringan beserta konfigurasi yang dibutuhkan jika PSI ingin menerapkan *transparent web proxy* dengan metode identifikasi *hostname* yang diusulkan pada tugas akhir ini. Implementasi dalam skala lab kemudian akan dilakukan untuk menguji model dan konfigurasi jaringan yang sudah dihasilkan. Hasil dari tahap ini juga menjawab RQ₂.

III.6.10 Pembuatan Implementasi Skala Lab

Metode identifikasi *hostname* yang sudah diperoleh pada subbab III.6.8 kemudian diimplementasikan dalam skala lab yang mencerminkan model dan konfigurasi jaringan yang dirancang pada subbab III.6.10.

Implementasi dilakukan dalam dua tahap. Tahap pertama, setiap komponen diimplementasi masing-masing dalam sebuah *virtual machine* menggunakan sistem operasi ber-*kernel* Linux. Masing-masing *virtual machine* kemudian bergabung dalam suatu topologi jaringan yang disimulasikan menggunakan *software GNS3*. Tahap ini menghasilkan konfigurasi yang ideal yang siap diimplementasikan.

Pada tahap kedua, konfigurasi yang sudah dihasilkan dari tahap pertama diimplementasi pada *hardware* berikut:

- Raspberry Pi 3 Model B sebagai *web proxy*, PDP/server RADIUS, dan *accounting and reporting system* dengan spesifikasi:
 - CPU *quad-core* 1,2 GHz Broadcom BCM2837 64-bit berarsitektur ARM,
 - RAM sebesar 1 GB,
 - satu buah *port* 100-Base Ethernet, dan

Tabel III.5. *Template* hasil eksperimen

<i>Hostname</i>	Variabel Bebas					Metode	Variabel Terikat					
	Jumlah Alamat IP Hasil <i>Lookup</i> DNS	Jumlah <i>Mutual</i> <i>Hostname</i>	Jumlah Alamat IP Pengguna	Jumlah Kunjungan per <i>Hostname</i>	Dukungan SNI pada Klien		TP⁽¹⁾	FP⁽²⁾	FN⁽³⁾	Recall	Precision	F₁ score
<i>H₁</i>	Aktif	<i>Reverse lookup ke DNS</i>
						Tidak aktif	
						Aktif	<i>Reverse lookup dari rekaman query</i>
						Tidak aktif	
						Aktif	<i>SNI pada handshake TLS</i>
						Tidak aktif	
						Aktif	<i>Atribut CN pada sertifikat TLS</i>
						Tidak aktif	
<i>H₂</i>
...
<i>H_N</i>

Keterangan: (1) TP: *true positive*; (2) FP: *false positive*; (3) FN: *false negative*.

Nilai *recall* dihitung dengan rumus II.1, *precision* dengan rumus II.2, dan *F₁ score* dengan rumus II.3.

- bersistem operasi Raspbian Stretch (berbasis Debian 9), dipilih karena merupakan sistem operasi ber-*kernel* Linux;
- MikroTik hEX (RB750Gr3) sebagai *router* dan PEP dengan spesifikasi:
 - lima buah *port* Gigabit Ethernet,
 - CPU *dual-core* 880 MHz,
 - RAM sebesar 256 MB, dan
 - bersistem operasi RouterOS dengan lisensi level 4, memiliki fitur *captive portal* (bernama *hotspot* dalam terminologi RouterOS).

Kedua perangkat dipilih karena dimensinya yang ringkas sehingga memungkinkan percobaan dilakukan secara *mobile* (tidak terikat ruangan di Jurusan Teknik Komputer dan Informatika).

Sebagaimana yang dijelaskan di subbab II.1.8.1, pertimbangan pemilihan protokol autentikasi klien dengan PEP mengikuti ketersediaan teknologi yang ada pada perangkat PEP. Pada implementasi skala lab ini, protokol yang digunakan untuk melakukan autentikasi klien dengan PEP adalah HTTP dengan mekanisme *captive portal* dari RouterOS. Protokol ini digunakan karena sudah tersedia sebagai bawaan perangkat MikroTik hEX (RB750Gr3) yang digunakan, sehingga memudahkan implementasi.

Sementara PDP (server RADIUS), *accounting and reporting system* serta *web proxy* ditempatkan pada *node* yang sama; komunikasi *web proxy* dengan *accounting and reporting system* dilakukan melalui aplikasi kecil yang dibangun pada tugas akhir ini. Aplikasi kecil ini akan memanggil program `radwho` untuk melihat sesi yang sedang tercatat aktif oleh server RADIUS. Aplikasi kecil dibangun dengan *shell script* menggunakan dialek Bash dan dibangun sesuai kebutuhan.

Pada kedua tahap tersebut, Squid digunakan sebagai *web proxy*. Squid dipilih karena Squid merupakan *web proxy* yang digunakan di Politeknik Negeri Bandung. Dengan demikian, ketika akan diusulkan untuk diimplementasikan (sesudah tugas akhir ini selesai), PSI tidak perlu mempelajari *web proxy* baru. Versi Squid yang digunakan adalah Squid versi 4.1, karena fitur SNI diimplementasikan pada Squid

sejak versi 3.5 dan terdapat beberapa *bug fix* terkait identifikasi menggunakan SNI dari versi 3.5 ke versi 4.1.

Untuk implementasi server RADIUS, dipilih FreeRADIUS. FreeRADIUS memiliki dokumentasi yang cukup baik, komunitas yang sudah cukup aktif, serta *resource* tutorial yang tersedia di internet. Dengan demikian, implementasi pada tugas akhir ini menjadi lebih mudah.

III.6.11 Percobaan Aplikasi Pengguna dan Evaluasi Hasil

Pada tahap ini aplikasi pengguna yang sudah dikumpulkan pada subbab III.6.4 dicoba menggunakan jaringan implementasi skala lab. Percobaan ini dilakukan untuk:

1. mengetahui apakah aplikasi mendapatkan akses ke internet, serta
2. apakah metode identifikasi *hostname* yang dihasilkan dari subbab III.6.8 dapat mengidentifikasi aplikasi-aplikasi yang digunakan secara langsung.

Kegiatan ini dilakukan dengan menghubungkan komputer ke jaringan implementasi skala lab, kemudian mencoba fitur-fitur dari aplikasi yang membutuhkan akses ke internet. Selama percobaan fitur, log dari *transparent web proxy* akan diamati.

Hasil dari percobaan ini diharapkan dapat memberikan kesimpulan yang bulat apakah metode identifikasi yang terpilih dapat memenuhi kebutuhan atau tidak.

BAB IV

PENENTUAN METODE IDENTIFIKASI *HOSTNAME*

Bab ini menjelaskan analisis *problem domain* terkait metode identifikasi *hostname*, pembuatan aplikasi eksperimen, penyiapan data, dan hasil eksperimen beserta analisisnya. Bab ini menjawab RQ₁.

IV.1 Analisis *Problem Domain*

Subbab ini menjelaskan dua hal yang dianalisis untuk menentukan metode identifikasi *hostname*, yaitu permasalahan identifikasi *hostname* pada akses HTTPS, metode-metode identifikasi *hostname* pada akses HTTPS, serta analisis kebutuhan aplikasi eksperimen. Analisis *problem domain* lainnya terkait penerapan metode identifikasi yang terpilih serta analisis mengenai kondisi di Politeknik Negeri Bandung akan dijelaskan pada Bab V.

IV.1.1 Permasalahan Identifikasi *Hostname* Akses HTTPS

Identifikasi akses HTTPS merupakan permasalahan yang berkembang sejak penggunaan protokol TLS dan kedulian masyarakat akan keamanan transportasi data meningkat. Peningkatan yang sangat terasa dan dibicarakan dalam media diskusi informal (blog, forum, media sosial) adalah peningkatan penggunaan protokol TLS pada protokol HTTPS sebagaimana dipublikasikan secara akademik oleh Felt *et al.* (2017). Hal ini menjadi permasalahan bagi pengelola jaringan karena protokol-protokol *application layer* perlu dimonitor jika ingin mendapatkan *insight* mengenai pemakaian jaringan dalam suatu institusi.

Untuk dapat memahami permasalahan identifikasi akses HTTPS ini, diperlukan model referensi OSI, meskipun implementasi jaringan menggunakan *protocol stack* TCP/IP. Pada model referensi OSI (lihat Gambar II.1), terdapat *presentation layer* yang dapat melakukan enkripsi dan dekripsi protokol yang ada pada *application layer* (Tanenbaum dan Wetherall, 2011). Pada *layer* inilah protokol TLS ditempatkan.

Dengan demikian, untuk melakukan identifikasi *hostname* pada akses HTTPS, diperlukan data yang bersumber dari *presentation layer* ke bawah. Alternatif lain adalah menggunakan data lain selain komunikasi akses HTTPS yang dilakukan; misalnya dengan menggunakan DNS.

Pada subbab ini, akan dilakukan analisis lebih lanjut terhadap empat metode identifikasi *hostname* yang diungkapkan oleh Bermudez *et al.* (2012), Foremski, Callegari dan Pagano (2014), Rao (2013), dan Shbair (2017).

IV.1.1.1 Metode *Reverse Lookup* Berdasarkan Entri DNS

Metode *reverse lookup* berdasarkan entri DNS merupakan metode yang paling sederhana, karena untuk mengetahui *hostname* dari alamat IP, *transparent web proxy* cukup melakukan *lookup* terhadap entri DNS dengan jenis PTR.

Barr (1996) pada RFC 1912 menyatakan bahwa setiap alamat IP harus memiliki sebuah entri DNS berjenis PTR sehingga dapat dilakukan *reverse lookup*. *Hostname* yang dinyatakan pada entri jenis PTR ini juga harus mencatat alamat IP yang sama sebagai sebuah entri DNS berjenis A. Dengan demikian, secara teori, metode ini dapat diandalkan untuk melakukan identifikasi *hostname* dari alamat IP yang ada, karena terjadi simetri fungsi $\text{reverse_lookup}(\text{IP}) = \text{hostname}$ dan $\text{lookup}(\text{hostname}) = \text{IP}$.

Namun, spesifikasi DNS juga mengizinkan entri jenis A lebih dari satu, sehingga satu *hostname* bisa mengembalikan lebih dari satu alamat IP ketika di-*lookup*. Perilaku seperti ini biasanya dilakukan oleh penyedia layanan dengan skala besar yang ingin menghadirkan *load balancing* langsung sejak *query* DNS dilakukan oleh klien. Dengan perilaku ini, ada kekhawatiran bahwa simetri fungsi yang sebelumnya diungkapkan tidak terjadi, sehingga melanggar RFC 1912. Selain itu, penggunaan *virtual hosting* juga akan membatalkan simetri fungsi, karena akan terdapat H_x , sembarang *hostname*, dengan $\text{lookup}(H_x) = \text{IP}$ namun $\text{reverse_lookup}(\text{IP}) \neq H_x$.

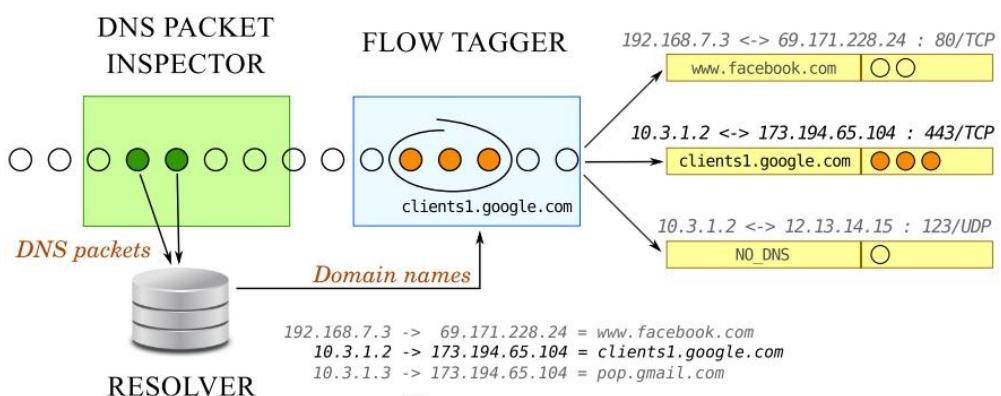
Kekhawatiran ini merupakan salah satu alasan jumlah alamat IP yang dikembalikan dari suatu *lookup hostname* serta jumlah *mutual hostname* menjadi variabel bebas

yang akan dilihat pengaruhnya terhadap nilai *F₁ score* dari hasil identifikasi metode pada penelitian ini.

IV.1.1.2 Metode *Reverse Lookup* Berdasarkan Rekaman *Query*

Metode *reverse lookup* berdasarkan rekaman *query* yang diusulkan oleh Bermudez *et al.* (2012) dan Foremski, Callegari dan Pagano (2014) lebih menjanjikan daripada metode *reverse lookup* berdasarkan entri DNS. Pada metode ini, server DNS yang ada di dalam institusi merekam seluruh *query* yang dilakukan oleh pengguna. Rekaman ini kemudian akan menjadi sumber *reverse lookup* terhadap *hostname* dari alamat IP yang diakses oleh pengguna. Metode ini lebih menjanjikan karena entri DNS berjenis PTR sama sekali tidak digunakan pada metode ini. Dengan demikian, kekhawatiran bahwa RFC 1912 tidak diikuti oleh banyak penyedia layanan tidak berlaku pada metode ini.

Metode ini berangkat dari kondisi normal bahwa pengalaman antar komputer untuk komunikasi di internet yang menggunakan TCP/IP menggunakan alamat IP, bukan menggunakan *hostname*. Jika tujuan komunikasi hanya dinyatakan dalam bentuk *hostname*, maka pengguna perlu melakukan *lookup* ke DNS terlebih dahulu sebelum dapat berkomunikasi. *Lookup* yang dilakukan pengguna inilah yang menjadi sumber data yang digunakan ketika *reverse lookup* pada metode ini. Visualisasinya digambarkan oleh Foremski, Callegari dan Pagano (2014) pada Gambar IV.1.



Gambar IV.1. Visualisasi *reverse lookup* berdasarkan rekaman *query*

Jika metode ini ingin diimplementasikan, maka pengelola jaringan perlu memastikan bahwa semua pengguna di dalam jaringannya melakukan *query* ke satu server DNS yang sama yang dikelola oleh pengelola jaringan. Dengan demikian, pengelola IT dapat merekam seluruh *query* yang dilakukan oleh pengguna di dalam jaringannya.

Alternatif lain dari menyediakan satu server DNS di dalam jaringan adalah dengan melakukan *sniffing* terhadap seluruh *traffic* DNS yang terkirim di dalam jaringan. Dengan teknik *sniffing*, pengelola jaringan tidak perlu menyiapkan server DNS di dalam jaringan, jika belum ada.

Rekaman *query* kemudian perlu diproses dengan program untuk menjadi struktur data seperti yang digambarkan Bermudez *et al.* (2012) pada Gambar II.14. Struktur data yang cocok untuk mengimplementasi visualisasi tersebut adalah struktur data *hash table*. Dengan *hash table*, pencarian terhadap alamat IP pengguna dan alamat IP yang sedang diakses (lihat Gambar II.14) dapat dilakukan dengan kompleksitas algoritma yang kecil, sehingga tidak meningkatkan *latency* secara signifikan dari akses internet yang dilakukan pengguna.

Hal yang perlu diwaspadai dan bisa berpengaruh terhadap akurasi metode ini adalah kombinasi praktik *virtual hosting* dan adanya atribut *time to live* pada entri DNS. Praktik *virtual hosting* menyebabkan satu alamat IP bisa terasosiasi dengan lebih dari satu *hostname*, sedangkan atribut *time to live* menyebabkan klien dapat melakukan *caching* terhadap jawaban dari DNS.

Skenario yang dapat menjelaskan mengapa *virtual hosting* dan *time to live* dapat mempengaruhi akurasi metode ini adalah jika seorang pengguna mengakses situs S_1 . Pada kali pertama pengguna mengakses S_1 , klien akan melakukan *query* DNS untuk mengetahui alamat IP dari S_1 . Pada DNS, S_1 tercatat dapat diakses melalui alamat IP A. Klien kemudian membuka koneksi alamat IP A. Struktur data kemudian akan mencatat bahwa $hostname(klien, A) = S_1$; bahwa klien pernah melakukan *lookup* untuk S_1 dan alamat IP-nya adalah A.

Namun, karena pengguna belum terautentikasi pada situs S_1 , maka situs S_1 mengarahkan pengguna ke situs untuk melakukan autentikasi dan otorisasi pada S_2 (kasus nyata dari S_1 dan S_2 ini misalnya `mail.google.com` dan `accounts.google.com`). Pada kali pertama pengguna mengakses S_2 , klien akan melakukan *query DNS* untuk mengetahui alamat IP dari S_2 . Pada DNS, S_2 tercatat dapat diakses melalui alamat IP A pula. Klien kemudian membuka koneksi baru ke alamat IP A. Struktur data kemudian akan mencatat bahwa *hostname*(klien, A) = S_2 akibat *query DNS* yang dilakukan terhadap S_2 .

Sesudah pengguna terautentikasi pada S_2 , S_2 mengarahkan pengguna kembali ke S_1 . Jika klien sudah pernah mengakses S_1 dalam waktu yang tidak terlalu lama (masih di bawah *time to live* entri DNS S_1), maka metode ini akan mengidentifikasi akses pengguna tersebut sebagai akses ke S_2 akibat *hostname*(klien, A) masih bernilai S_2 .

Fenomena ini juga merupakan alasan lain mengapa jumlah alamat IP hasil *lookup* terhadap *hostname* dan jumlah *mutual hostname* dijadikan variabel bebas pada penelitian ini. Dua variabel tersebut merupakan dugaan variabel bebas yang akan memengaruhi nilai *F₁ score* metode ini, dan akan dibuktikan sesudah analisis hasil eksperimen dilakukan.

IV.1.1.3 Metode Server Name Indication (SNI)

Metode SNI ini diungkapkan oleh Rao (2013) dan Shbair (2017). Metode ini terlihat sederhana, yakni cukup dengan membaca atribut SNI yang merupakan *extension* dari protokol TLS untuk menentukan *hostname* yang sedang diakses.

Ketidaksederhanaan metode ini terletak pada ketergantungannya pada atribut SNI yang sifatnya opsional: klien yang menggunakan protokol TLS tidak wajib mengirimkan atribut SNI ketika melakukan *handshake* TLS. Jika klien tidak mendukung penggunaan SNI, maka metode ini akan gagal total mengidentifikasi *hostname* yang dituju oleh pengguna.

Berdasarkan hal tersebut, maka dugaan terhadap akurasi hasil identifikasi metode ini akan bersifat biner: 100% jika klien mendukung SNI, dan 0% jika klien tidak

mendukung SNI. Studi Nygren (2017) menyatakan bahwa tingkat penggunaan SNI pada akses web menggunakan protokol HTTPS sebenarnya telah mencapai 99,4% pada bulan Maret 2017. Namun, Nygren melakukan studi ini dalam perannya sebagai peneliti Akamai, salah satu penyedia *content distribution network* (CDN) besar. Mengingat umumnya CDN digunakan untuk membantu penyedia layanan menghadirkan konten menggunakan *web browser*, serta ketiadaan informasi pada studi Nygren aplikasi apa saja yang mengakses data Akamai untuk menentukan tingkat penggunaan SNI, maka perlu ditelusuri apakah aplikasi-aplikasi yang digunakan di Politeknik Negeri Bandung memang sudah mendukung penggunaan SNI jika ingin diterapkan.

IV.1.1.4 Metode Atribut *Common Name* (CN) pada Sertifikat TLS

Rao (2013) dan Shbair (2017) kemudian mengungkapkan metode terakhir, yaitu menggunakan atribut CN pada sertifikat yang diberikan ketika *handshake* TLS. Atribut CN merupakan bagian dari identitas utama pada sebuah sertifikat TLS yang menyatakan *hostname* utama yang dilayani oleh sertifikat TLS ini.

Persoalan penggunaan atribut CN ini adalah spesifikasi bahwa atribut CN boleh bernilai *wildcard* (misalnya: *.polban.ac.id). Atribut CN dengan *wildcard* menunjukkan bahwa sertifikat itu memang digunakan untuk melayani seluruh *hostname* yang berada satu tingkat di bawah polban.ac.id pada hierarki DNS. Atribut CN *wildcard* ini memiliki implikasi sebagai berikut:

1. ketika digunakan untuk mencatat akses, maka harus dipilih sebuah *hostname* untuk dicatat. Oleh karena itu, *wildcard* perlu dibuang sebelum *hostname* dicatat;
2. ketika digunakan untuk melakukan filter pada *access control list* (ACL, yaitu konfigurasi berupa sekumpulan *rules* yang ada pada *firewall* atau *web proxy*), maka *wildcard* tidak perlu dibuang. Pengujian apakah baris ACL akan *match* atau tidak cukup dengan memeriksa apakah *hostname* yang sedang diperiksa *match* dengan *wildcard* atau tidak. Sebagai contoh, www.polban.ac.id tentu *match* dengan *.polban.ac.id, namun tidak dengan www.jtk.polban.ac.id (karena atribut CN *wildcard* hanya berlaku satu tingkat), atau bahkan situs yang benar-benar berbeda seperti www.itb.ac.id.

Karena tugas akhir ini fokus pada pencatatan akses sebagaimana kebutuhan PSI yang menjadi latar belakang pada subbab I.1, maka *wildcard* yang akan dibuang sebelum *hostname* dicatat. Pembuangan *wildcard* berpotensi mengurangi akurasi metode ini, karena hasil identifikasi tidak akan sama dengan *ground truth* yang ada. Hal ini akan dibuktikan pada analisis hasil eksperimen.

IV.1.2 Analisis Kebutuhan Aplikasi Eksperimen

Keempat metode di atas akan dieksperimenkan dalam simulasi akses dari pengguna dalam jaringan yang memiliki *transparent web proxy*. Untuk setiap akses dari pengguna, *transparent web proxy* akan mencoba melakukan identifikasi *hostname* untuk keperluan pencatatan log, sehingga keluaran dari setiap simulasi yang dilakukan adalah sebuah baris log layaknya yang dikeluarkan oleh *web proxy*.

Aplikasi perlu memiliki fungsi dari peran-peran yang akan disimulasikan berikut:

- Experimenter, berfungsi untuk:
 - membaca log dari *explicit web proxy* yang akan disimulasikan,
 - memberikan masukan kepada Client yang akan menyimulasikan sebuah *request* akses ke internet untuk setiap baris yang dibaca oleh Experimenter,
 - menerima hasil identifikasi yang dilakukan oleh ProxyServer, serta
 - menulis log hasil simulasi berisi *ground truth* dan hasil identifikasinya;
- Client, berfungsi untuk menyimulasikan kegiatan yang dilakukan oleh klien, yaitu:
 - melakukan *lookup DNS* jika klien belum memiliki informasi alamat IP dari suatu *hostname*.

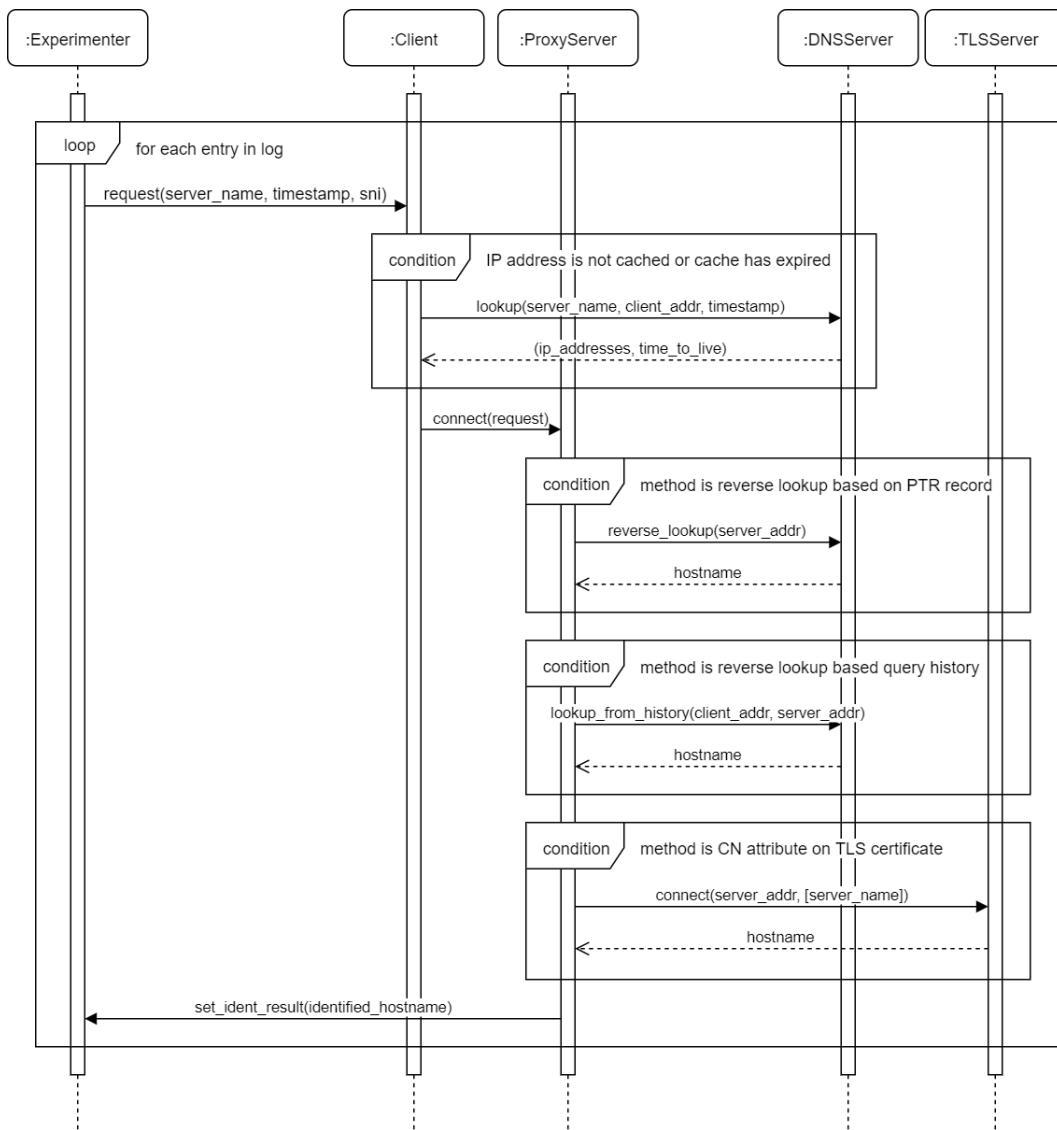
Lookup DNS tidak akan dilakukan apabila klien sudah pernah melakukan *lookup* terhadap *hostname* yang sama, dan data tersebut belum kadaluarsa (ditandai dengan *time to live* yang dikirimkan oleh server DNS);

- membuat sebuah *request* yang akan diterima oleh ProxyServer.
Request yang dibuat oleh Client berisi dua data yang pasti akan diterima oleh *transparent web proxy* sesungguhnya, yaitu alamat IP klien dan alamat IP dari server yang dituju. Jika dukungan SNI pada klien sedang disimulasikan, maka *request* ini akan berisi satu data tambahan, yaitu atribut SNI yang berisi *hostname* yang dituju oleh klien;

- ProxyServer, berfungsi untuk:
 - menerima *request* dari Client,
 - melakukan identifikasi berdasarkan metode identifikasi *hostname* yang sedang disimulasikan, serta
 - menyampaikan hasil identifikasi *hostname* ke Experimenter yang akan menuliskan log hasil simulas;
- DNSServer, berfungsi untuk menyimulasikan sebuah server DNS yang:
 - menerima *query* dari Client untuk melakukan *lookup* terhadap *hostname* yang akan dituju oleh Client,
 - menerima *query* dari ProxyServer untuk melakukan *reverse lookup* terhadap alamat IP berdasarkan entri DNS, serta
 - menerima *query* dari ProxyServer untuk melakukan *reverse lookup* berdasarkan rekaman *query* yang dilakukan Client sebelumnya. Untuk menjalankan fungsi ini, DNSServer perlu mencatat semua *query* yang dilakukan oleh Client dalam suatu struktur data yang sudah dianalisis pada subbab IV.1.1.2;
- TLSServer, berfungsi untuk menyimulasikan server tujuan Client yang mengembalikan sertifikat TLS berdasarkan *request* yang diterima dari ProxyServer. Sertifikat TLS yang dikirimkan kepada ProxyServer bisa berbeda, tergantung apakah *request* dilakukan menggunakan atribut SNI atau tidak.

Kelima peran tersebut beserta alur dalam sebuah simulas digambarkan menggunakan *sequence diagram* pada Gambar IV.2.

DNSServer dan TLSServer perlu disimulasikan untuk memastikan bahwa eksperimen dengan data yang sama akan menghasilkan data yang sama pula. Untuk kebutuhan simulas, maka DNSServer dan TLSServer perlu mendapatkan masukan berupa data DNS dan data sertifikat TLS dari seluruh *hostname* dan alamat IP yang mungkin dikunjungi oleh Client berdasarkan log yang menjadi data masukan. Hal ini yang menyebabkan pada tahap penyiapan data, data DNS dan data sertifikat TLS akan diambil sebagai bagian dari tahap penyiapan data (dengan metode yang dijelaskan pada subbab III.6.6.4 dan III.6.6.5).



Gambar IV.2. *Sequence diagram eksperimen pada aplikasi utama*

Pada implementasinya, masing-masing peran tersebut direpresentasikan sebagai sebuah *class*. Garis tegas antar *instance class* pada Gambar IV.2 kemudian menyatakan *method-method* yang perlu ada di masing-masing *class*, sementara garis putus-putus menyatakan nilai kembalian dari masing-masing *method* yang mendahuluinya.

IV.2 Pembuatan Aplikasi Eksperimen

Seluruh aplikasi yang digunakan pada tugas akhir ini diimplementasi menggunakan bahasa pemrograman Python 3.5.2 dengan implementasi CPython, kecuali proses

mendapatkan *hostname* unik yang dilakukan menggunakan GNU Awk 4.1.3. Pengembangan dilakukan dengan perangkat komputasi berikut:

- CPU Intel® Core™ i7-8550U @ 1,80 GHz;
- RAM 8 GB;
- sistem operasi Ubuntu 18.04 LTS yang berjalan di atas Windows Subsystem for Linux (WSL) pada Windows 10 Home.

Sistem operasi Ubuntu 18.04 LTS di atas WSL dipilih untuk menyamakan lingkungan *development* dengan lingkungan *production* (tempat eksperimen berjalan, di *virtual private server* yang dijelaskan pada subbab III.6.7.2).

Program-program kecil yang mengakses internet untuk memeriksa keaktifan *hostname*, mengambil data DNS, dan mengambil data sertifikat TLS diimplementasi dengan memanfaatkan fitur *multithreading* untuk mempercepat proses pengambilan data. *Multithreading* dilakukan hingga maksimal 128 *threads*. Jumlah maksimal *threads* yang aktif dalam satu waktu dijaga menggunakan sinkronisasi *semaphore*.

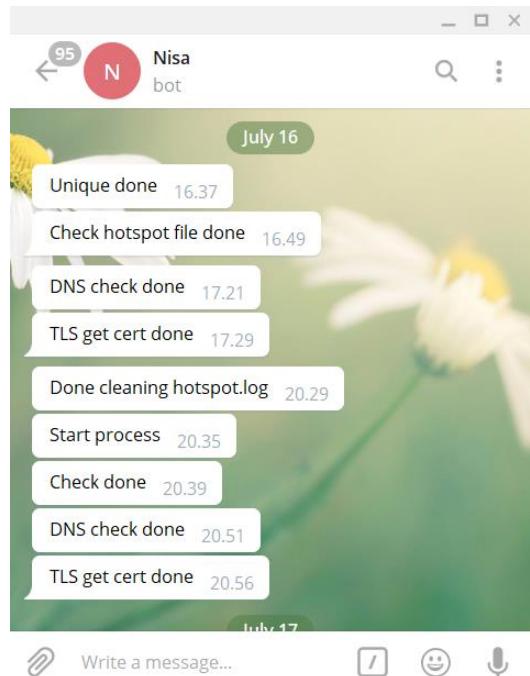
Selain program-program yang bekerja, sebuah bot Telegram juga disiapkan untuk memberikan notifikasi bahwa proses yang dikerjakan di *virtual private server* sudah selesai. Dengan adanya bot, koneksi ke *virtual private server* bisa dilakukan seperlunya saja – tidak perlu menunggu proses selesai menggunakan klien SSH. Bot ini ditunjukkan pada Gambar IV.3.

Program-program yang dibuat pada tahap ini disertakan pada Lampiran 4.

IV.3 Penyiapan Data

Data log dari *explicit web proxy* yang disediakan PSI disiapkan menggunakan program-program yang sudah dibuat pada subbab IV.2 mengikuti metode yang sudah dinyatakan pada subbab III.6.6. Data disiapkan di *virtual private server*.

Tahap ini menghasilkan empat buah *file*, yaitu data sertifikat TLS, data hasil *lookup* dan *reverse lookup* dari entri DNS, serta log yang sudah disaring. Keempat *file* tersebut digunakan pada tahap eksperimen.



Gambar IV.3. Bot Telegram untuk notifikasi eksperimen

IV.4 Eksperimen

Eksperimen dilakukan di *virtual private server* menggunakan aplikasi yang sudah dibuat pada subbab IV.2 dengan masukan berupa empat buah *file* yang dihasilkan dari tahap penyiapan data. Eksperimen menghasilkan delapan buah log simulasi *transparent web proxy* menggunakan metode-metode yang dibandingkan. Delapan buah log tersebut kemudian diproses dan menghasilkan sebuah *file* berformat SQLite yang siap di-*query* untuk memenuhi kebutuhan analisis hasil eksperimen.

IV.5 Analisis Hasil Eksperimen

Hasil eksperimen dianalisis dengan memperhatikan pengaruh dari masing-masing variabel bebas terhadap nilai *F₁ score* hasil identifikasi. Variabel bebas yang dimanipulasi – yaitu dukungan SNI pada klien – dianalisis terlebih dahulu sebelum variabel bebas yang tidak dimanipulasi.

IV.5.1 Pengaruh dari Dukungan SNI pada Klien

Hasil identifikasi dari masing-masing metode jika diperhatikan terhadap dukungan SNI pada klien ditunjukkan pada Tabel IV.1.

Tabel IV.1. Nilai F_1 score berdasarkan dukungan SNI pada klien

Dukungan SNI pada Klien	Metode Identifikasi	F_1 Score
Aktif	<i>Reverse lookup</i> dari entri DNS	1,53%
	<i>Reverse lookup</i> dari rekaman <i>query</i>	75,82%
	SNI pada <i>handshake TLS</i>	100%
	Atribut CN pada sertifikat TLS	18,72%
Tidak Aktif	<i>Reverse lookup</i> dari entri DNS	1,53%
	<i>Reverse lookup</i> dari rekaman <i>query</i>	75,82%
	SNI pada <i>handshake TLS</i>	0%
	Atribut CN pada sertifikat TLS	13,60%

Pada tabel tersebut, metode yang bergantung pada DNS (*reverse lookup* berdasarkan entri DNS serta *reverse lookup* dari rekaman *query*) memiliki nilai F_1 score yang sama, baik ketika SNI didukung oleh klien maupun tidak. Hal ini menunjukkan bahwa nilai F_1 score pada kedua metode tersebut tidak terpengaruh oleh dukungan SNI pada klien.

Nilai F_1 score pada metode *reverse lookup* berdasarkan entri DNS sangat rendah, yaitu 1,53%. Dengan nilai F_1 score serendah ini, metode ini tidak dapat digunakan untuk melakukan identifikasi *hostname* pada penggunaan di Politeknik Negeri Bandung.

Adapun nilai F_1 score pada metode *reverse lookup* dari rekaman *query* pengguna cukup tinggi, yaitu 75,82%. Dengan nilai F_1 score tersebut, metode ini cukup menjanjikan untuk digunakan. Namun, masih terdapat metode lain yang perlu dianalisis untuk dapat menjawab RQ₁.

Di sisi lain, metode atribut CN pada sertifikat TLS terpengaruh oleh dukungan SNI pada klien. Terjadi sedikit perbedaan pada nilai F_1 score antara ketika klien mendukung SNI (nilainya lebih tinggi) dibandingkan dengan ketika klien tidak mendukung SNI.

Perbedaan tersebut terjadi akibat penggunaan *virtual hosting*. Pada penerapan *virtual hosting*, sebuah alamat IP dapat memberikan lebih dari satu layanan dengan *hostname* yang berbeda. Masing-masing *hostname* tersebut kemudian dapat menggunakan sertifikat TLS yang berbeda. Agar server memahami sertifikat TLS

mana yang perlu diberikan kepada klien ketika *handshake*, server membutuhkan informasi SNI dari klien.

Hal tersebut dapat dikonfirmasi jika log yang dihasilkan dari simulator *transparent web proxy* dibandingkan, dengan sebagian hasilnya ditunjukkan pada Tabel IV.2. Server-server yang tertulis pada Tabel IV.2 tersebut memberikan sertifikat TLS yang berbeda ketika diakses menggunakan klien yang mendukung SNI dibandingkan dengan klien yang tidak mendukung SNI.

Tabel IV.2. Perbandingan atribut CN dengan dan tanpa dukungan SNI

Ground Truth	Dengan Dukungan SNI	Tanpa Dukungan SNI
accounts.google.co.id	google.co.id	google.com
apis.google.com	apis.google.com	google.com
c1.staticflickr.com	yimg.com	yahoo.com
c2.staticflickr.com	yimg.com	yahoo.com
choices.truste.com	truste.com	13.35.8.17
clients1.google.co.id	google.co.id	google.com
d5nxst8fruw4z.cloudfront.net	cloudfront.net	13.35.8.71
geo.query.yahoo.com	sp.analytics.yahoo.com	yahoo.com
global.adserver.yahoo.com	ads.yahoo.com	yahoo.com
go.padsdel.com	go.padsdel.com	188.42.162.135
mygpuid.com	mygpuid.com	188.72.202.234
s1.yimg.com	yimg.com	yahoo.com
s2.yimg.com	yimg.com	yahoo.com
s3.yimg.com	yimg.com	yahoo.com
socialprofiles.zenfs.com	www1.zenfs.com	yahoo.com
sp.yimg.com	yimg.com	yahoo.com
translate.google.co.id	google.co.id	google.com
w.soundcloud.com	soundcloud.com	13.35.8.34
www.google.co.id	google.co.id	google.com
www.googleadservices.com	www.googleadservices.com	g.doubleclick.net
www.google-analytics.com	google-analytics.com	google.com
wwwpromoter.com	sni61479.cloudflaressl.com	104.24.96.231

Dalam beberapa kasus, misalnya akses ke `apis.google.com`, `go.padsdel.com`, `mygpuid.com`, dan `www.googleadservices.com`, hasil identifikasi ketika klien mendukung SNI sama persis dengan *ground truth*. Ini menunjukkan bahwa server memberikan sertifikat TLS dengan *common name* yang bernilai sama persis dengan yang diminta oleh klien.

Di banyak kasus lainnya, hasil identifikasi ketika klien mendukung SNI memang tidak sama dengan *ground truth*. Namun, sebagian besar hasil identifikasi ketika klien mendukung SNI merupakan *hostname* dengan level yang lebih umum dari

ground truth. Dengan demikian, sebenarnya hasil identifikasi masih cukup membantu PSI menentukan ke mana (secara umum) akses dilakukan. Contohnya www.google-analytics.com yang diidentifikasi sebagai google-analytics.com; s1.yimg.com, s2.yimg.com, s3.yimg.com, dan sp.yimg.com yang diidentifikasi sebagai yimg.com; atau accounts.google.co.id, clients1.google.co.id, translate.google.co.id, dan www.google.co.id yang diidentifikasi sebagai google.co.id.

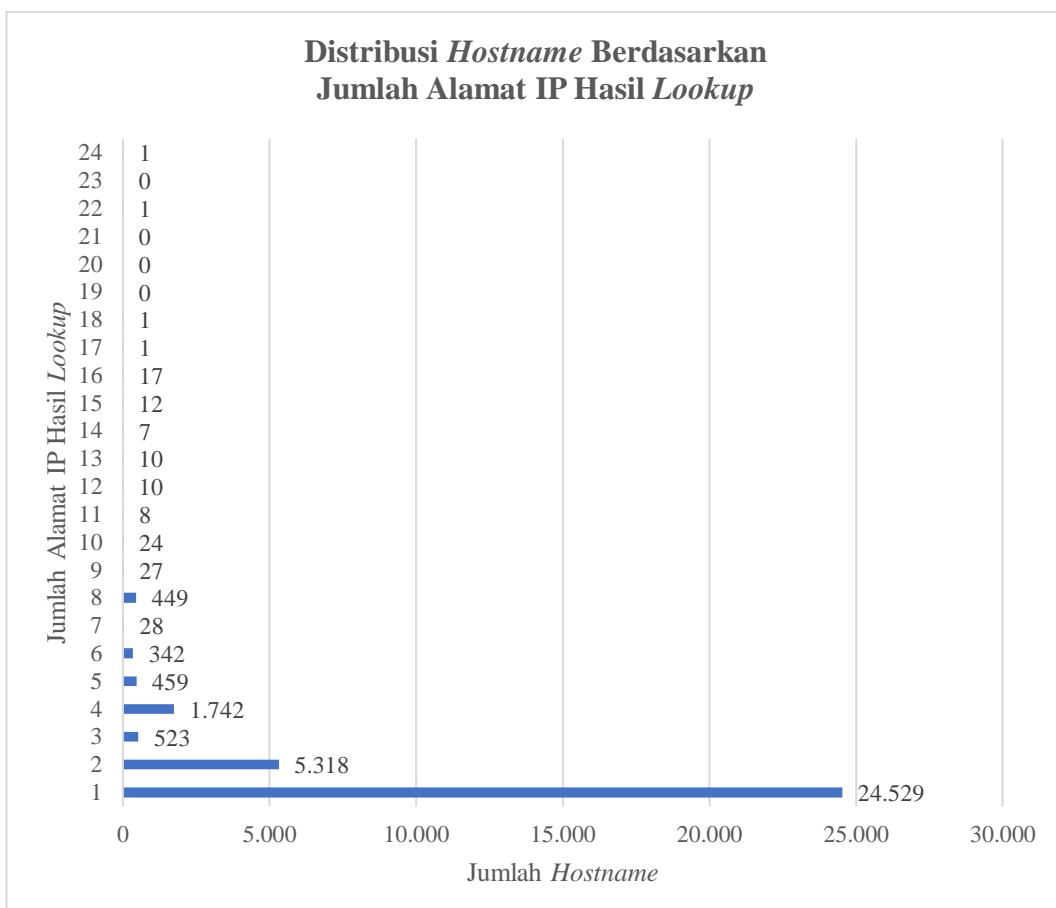
Namun, nilai F_1 score dari metode atribut CN pada sertifikat TLS juga cukup kecil. Nilai F_1 score tertinggi diperoleh ketika klien mendukung SNI, yaitu sebesar 18,72%. Sedangkan ketika klien tidak mendukung SNI, F_1 score dari metode ini hanya bernilai 13,60%. Sama halnya pada metode *reverse lookup* berdasarkan entri DNS, nilai ini menunjukkan bahwa metode atribut CN pada sertifikat TLS tidak dapat digunakan untuk melakukan identifikasi *hostname* pada penggunaan di Politeknik Negeri Bandung.

Adapun metode identifikasi berdasarkan SNI pada *handshake* TLS hanya terpengaruh oleh variabel ini. Hal ini ditunjukkan dengan nilai F_1 score yang bersifat biner: F_1 score bernilai sempurna (100%) ketika klien mendukung SNI, namun bernilai 0% ketika klien tidak mendukung SNI. Dengan demikian, metode ini bisa menjadi solusi hanya jika aplikasi yang digunakan di Politeknik Negeri Bandung mendukung SNI.

Karena metode SNI pada *handshake* TLS hanya terpengaruh pada variabel dukungan SNI pada klien ini, maka metode ini tidak akan dilihat pada analisis variabel-variabel bebas lainnya.

IV.5.2 Pengaruh dari Jumlah Alamat IP Hasil *Lookup*

Data dari entri DNS menunjukkan bahwa minimal satu *hostname* mengembalikan satu buah alamat IP, sementara satu *hostname* maksimal mengembalikan 24 buah alamat IP. Distribusi tersebut digambarkan pada Gambar IV.4.



Gambar IV.4. Jumlah alamat IP hasil *lookup* dan jumlah *hostname*-nya

Pada gambar tersebut, dapat dilihat bahwa mayoritas *hostname* (berjumlah 24.529) mengembalikan satu alamat IP saja. Sementara itu, sisanya kurang lebih maksimal mengembalikan hingga delapan alamat IP.

Pada analisis ini ditemukan hal yang menarik, yaitu *hostname counter.yadro.ru* yang mengembalikan 22 alamat IP sesudah di-*lookup* ternyata merupakan *hostname* yang menyajikan *adware* di komputer pengguna. Log *explicit web proxy* menyatakan terdapat 34 pengguna di Politeknik Negeri Bandung yang pernah mengakses *hostname* ini, dengan jumlah akses sebanyak 1.376 kali.

Sementara itu, beberapa *hostname* yang mengembalikan alamat IP lebih dari delapan buah ditunjukkan pada Tabel IV.3. Nama-nama pada *hostname* tersebut merupakan nama-nama penyedia layanan skala besar (misalnya *akamaihd.net* merupakan *hostname* milik Akamai, penyedia layanan CDN). Dengan demikian,

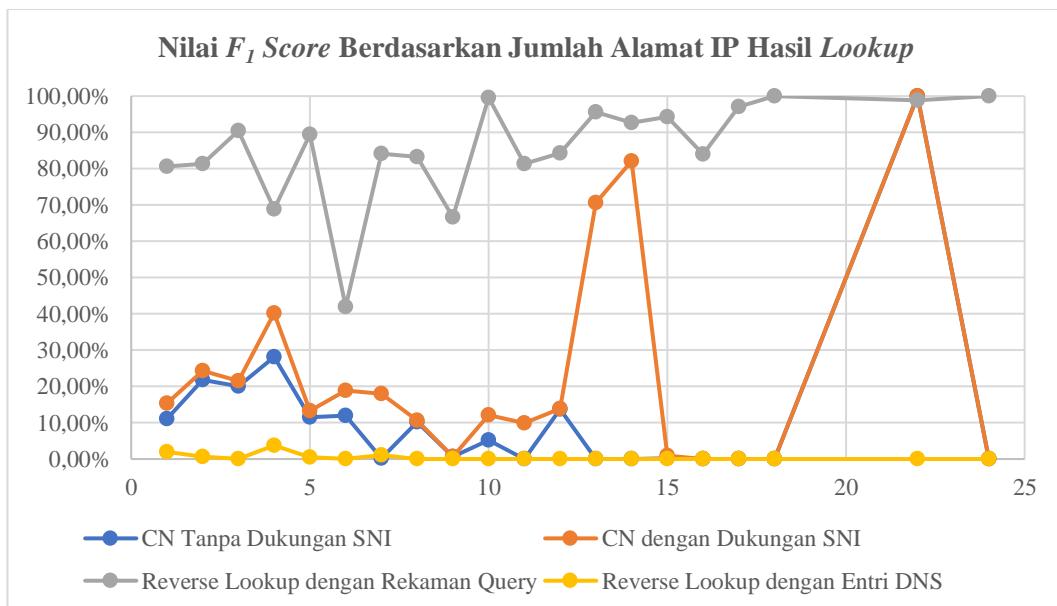
analisis bahwa skala layanan dapat direpresentasikan dengan jumlah alamat IP sebagaimana tertulis pada subbab IV.1.1.1 terbukti.

Tabel IV.3. Beberapa *hostname* dengan alamat IP lebih dari delapan buah

Hostname	Jumlah Alamat IP
android.clients.google.com	16
www.youtube.com	16
booking3.airasia.com	15
registry.npmjs.org	12
www.lazada.co.id	11
*.hotmail.com	9
*.akamaihd.net	9

Jika jumlah alamat IP dari suatu *hostname* ini dikaitkan dengan nilai F_1 score dari metode identifikasinya, hasilnya dapat dilihat pada Gambar IV.5.

Gambar tersebut menunjukkan bahwa pada jumlah alamat IP hasil *lookup* di atas dua belas, terjadi fluktuasi yang cukup signifikan dari nilai F_1 score metode yang ada. Hal ini terjadi karena jumlah *hostname* yang memiliki jumlah alamat IP hasil *lookup* yang tinggi cukup sedikit, sehingga nilai F_1 score lebih terpengaruh oleh variabel lain. Dengan demikian, data dengan jumlah alamat IP di atas delapan tidak dianalisis.



Gambar IV.5. Nilai F_1 score berdasarkan jumlah alamat IP hasil *lookup*

Gambar tersebut secara umum menunjukkan urutan F_1 score yang serupa dengan hasil analisis sebelumnya pada subbab IV.5.1. Urutan tersebut menunjukkan metode *reverse lookup* dengan rekaman *query* menempati urutan pertama. Urutan tersebut diikuti oleh metode atribut CN dengan kondisi SNI didukung oleh klien. Kemudian, metode atribut CN dengan kondisi SNI tidak didukung oleh klien mengikuti. Terakhir, metode *reverse lookup* berdasarkan entri DNS menempati urutan terendah.

Metode *reverse lookup* berdasarkan entri DNS menunjukkan hasil yang konsisten: maksimal memiliki nilai akurasi 2,70%, mayoritas bernilai di bawah 0,6%, dan gagal sama sekali mengidentifikasi *hostname* ketika jumlah alamat IP berada di atas tujuh. Hasil ini menguatkan dugaan pada analisis *problem domain* yang menyatakan bahwa semakin banyak jumlah alamat IP yang dikembalikan dari hasil *lookup* suatu *hostname* akan mengakibatkan simetri fungsi *reverse_lookup(IP)* dengan *lookup(hostname)* semakin berpeluang tidak terjadi.

Terdapat hal yang menarik perhatian pada data dengan jumlah alamat IP hasil *lookup* = 6. Data tersebut menunjukkan penurunan nilai F_1 score yang cukup signifikan pada metode *reverse lookup* dengan rekaman *query*. Untuk mengetahui mengapa hal tersebut terjadi, dilakukan analisis lanjutan. Analisis lanjutan mengungkapkan data yang ditunjukkan pada Tabel IV.4, yaitu jumlah *mutual hostname* dari *hostname* dengan jumlah alamat IP hasil *lookup* = 6.

Tabel IV.4. Perincian nilai F_1 score ketika jumlah alamat IP hasil *lookup* = 6

Jumlah alamat IP hasil <i>lookup</i>	Jumlah <i>mutual hostname</i>	Nilai F_1 score
6	0	98,14%
6	1	66,84%
6	2	93,19%
6	3	70,45%
6	4	84,44%
6	7	98,00%
6	8	76,11%
6	29	91,31%
6	189	28,13%

Pada tabel tersebut terlihat bahwa nilai F_1 score dari metode *reverse lookup* dengan rekaman *query* turun drastis ketika jumlah *mutual hostname*-nya 189. *Hostname*

dengan jumlah *mutual hostname* tersebut adalah *hostname* dari kelompok Google, ditandai dengan *hostname* *.google.com, *.youtube.com, *.ampproject.net, dan *.googleapis.com.

Meski turun drastis, namun sebagaimana ditunjukkan pada subbab IV.5.3, bukan berarti jumlah *mutual hostname* berpengaruh terhadap nilai F_1 score. Hal ini ditunjukkan dengan nilai koefisien pengaruh yang rendah (4,45% pada metode ini).

Adanya fluktuasi dari nilai F_1 score dari metode-metode yang ada menyebabkan perlu dilakukan analisis korelasi secara statistik. Dari rumus II.4 dan II.5, nilai koefisien korelasi beserta koefisien penentuan diketahui. Kedua nilai tersebut diuraikan pada Tabel IV.5.

Tabel IV.5. Korelasi jumlah alamat IP hasil *lookup* dengan nilai F_1 score

Metode	Koefisien korelasi	Koefisien penentuan
Reverse <i>lookup</i> dengan rekaman <i>query</i>	0,5504	30,29%
Reverse <i>lookup</i> dengan entri DNS	-0,502	25,20%
Atribut CN ketika SNI didukung klien	0,1275	1,62%
Atribut CN ketika SNI tidak didukung klien	0,1136	1,29%

Tabel tersebut menunjukkan bahwa jumlah alamat IP hasil *lookup* hanya menyumbang cukup besar terhadap nilai F_1 score pada metode yang terkait dengan DNS. Kontribusi dari variabel ini pada kedua metode tersebut cukup besar, yaitu 25,20% untuk metode *reverse lookup* dengan entri DNS dan 30,29% untuk metode *reverse lookup* dengan rekaman *query*.

Namun, korelasi kedua variabel tersebut pada masing-masing metode memiliki perbedaan yang cukup drastis. Pada metode *reverse lookup* dengan rekaman *query*, korelasi yang terjadi adalah korelasi positif dengan nilai 0,5504. Hal ini menunjukkan bahwa semakin banyak jumlah alamat IP hasil *lookup*, metode tersebut memiliki kecenderungan untuk menghasilkan nilai F_1 score yang lebih tinggi.

Kejadian tersebut dapat dipahami jika kita memperhatikan bahwa *virtual hosting* menyebabkan satu alamat IP dapat melayani lebih dari satu *hostname*. Metode *reverse lookup* dengan rekaman *query* dapat mengeluarkan hasil identifikasi yang

salah jika pengguna mengakses lebih dari satu *hostname* dengan alamat IP yang sama. Dengan jumlah alamat IP hasil *lookup* yang banyak, kemungkinan pengguna mengakses lebih dari satu *hostname* dengan alamat IP yang sama menjadi lebih kecil.

Sebaliknya, pada metode *reverse lookup* dengan entri DNS terjadi korelasi negatif dengan nilai -0,502. Hal ini menunjukkan bahwa semakin banyak jumlah alamat IP hasil *lookup*, metode tersebut memiliki kecenderungan untuk menghasilkan nilai F_1 *score* yang lebih rendah.

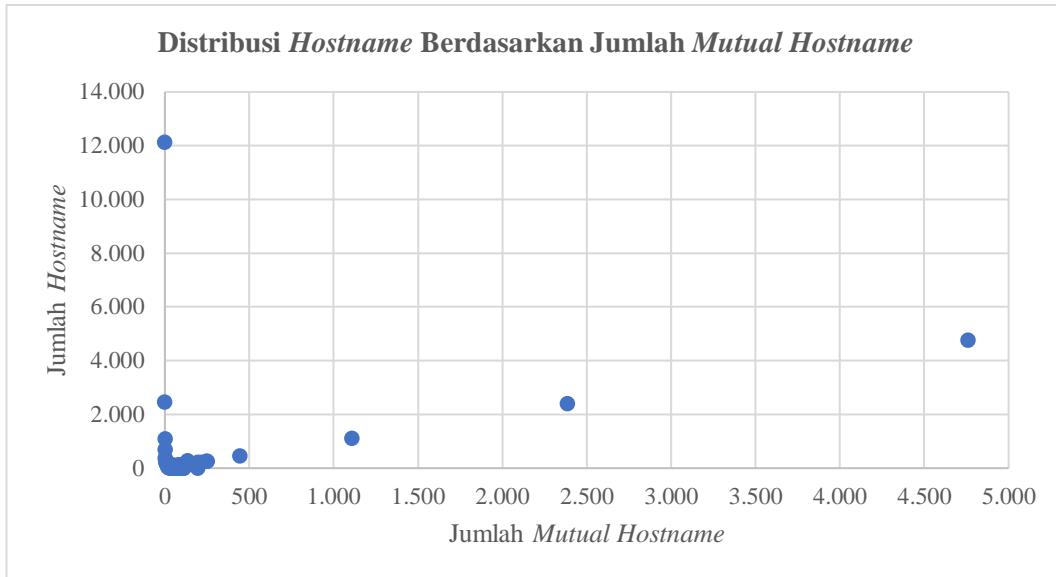
Kejadian tersebut juga dapat dimengerti dengan pemahaman tentang *virtual hosting*. Satu alamat IP dapat melayani lebih dari satu *hostname*, sedangkan hanya satu entri PTR yang dapat dilekatkan pada satu alamat IP. Simetri fungsi $\text{reverse_lookup}(\text{IP}) = \text{hostname}$ dan $\text{lookup}(\text{hostname}) = \text{IP}$ kemudian tidak dapat terjadi, sehingga menurunkan nilai F_1 *score* dari metode *reverse lookup* berdasarkan entri DNS ini.

Sementara itu, kontribusi jumlah alamat IP hasil *lookup* terhadap nilai F_1 *score* pada metode yang tidak terkait dengan DNS sangat kecil (maksimal hanya 1,62%). Dengan demikian dapat disimpulkan bahwa pada kedua metode identifikasi lainnya, variabel ini tidak berpengaruh terhadap nilai F_1 *score*.

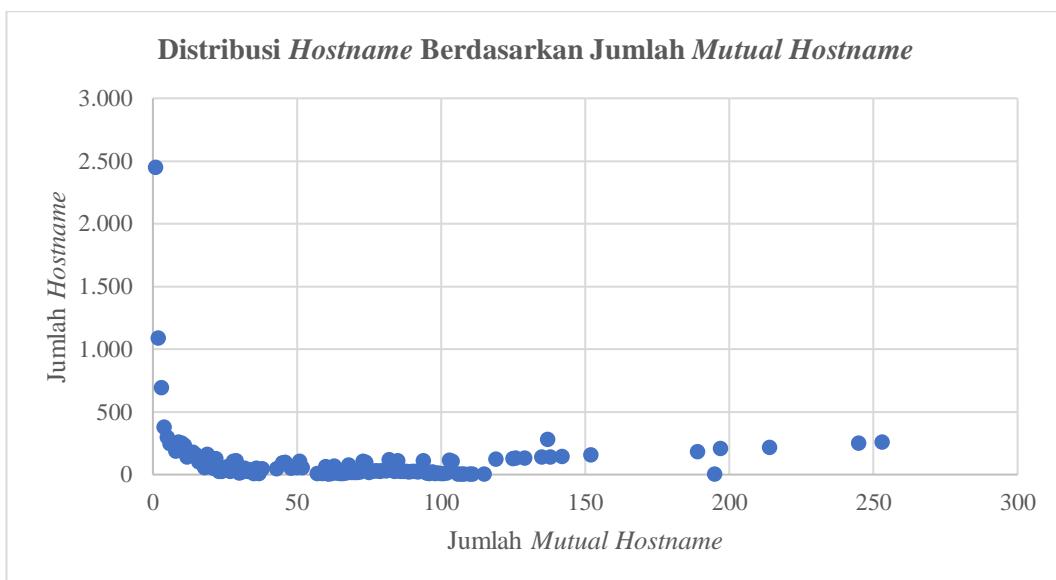
IV.5.3 Pengaruh dari Jumlah *Mutual Hostname*

Dari data DNS, diperoleh bahwa terdapat 12.123 *hostname* yang tidak memiliki *mutual hostname*, dan maksimal terdapat 4.762 *hostname* yang memiliki 4.761 *mutual hostname*. Distribusi ini jika digambar menggunakan *scatter plot* secara keseluruhan dapat dilihat pada Gambar IV.6.

Gambar tersebut menunjukkan dua titik ekstrem, yaitu *hostname* yang tidak memiliki *mutual hostname* dan *hostname* yang memiliki 4.761 *mutual hostname*. Jika nilai-nilai *outliers* (yang berjauhan dengan nilai-nilai lainnya) dibuang dan digambar ulang pada *scatter plot*, hasilnya ditunjukkan pada Gambar IV.7.



Gambar IV.6. Distribusi *hostname* berdasarkan jumlah *mutual hostname*



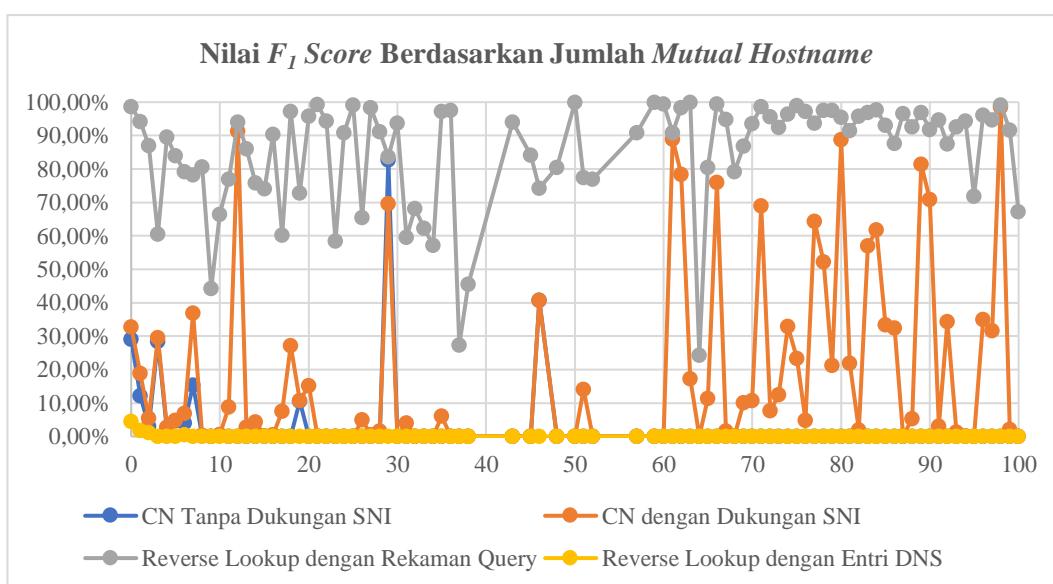
Gambar IV.7. Distribusi *hostname* dengan *outliers* yang dibuang

12.123 *hostname* yang tidak memiliki *mutual hostname* mewakili 36,17% dari keseluruhan *hostname* yang ada pada data penelitian. Sisanya, 63,83% *hostname* memiliki minimal satu *mutual hostname*. Hal ini menunjukkan bahwa penggunaan *virtual hosting* memiliki porsi yang cukup besar dari semesta *hostname* yang terdapat pada data penelitian. Salah satu kelompok *hostname* yang berbagi alamat IP adalah kelompok *hostname* dari Google (*.blogspot.com,

`*.googleusercontent.com`, `*.ampproject.org`, dan `*.gghpht.com` sebanyak 1.178 *hostname*).

Dari distribusi tersebut, digambarkan nilai F_1 score dari masing-masing metode berdasarkan jumlah *mutual hostname* pada Gambar IV.8. Gambar tersebut hanya menunjukkan data dengan jumlah *mutual hostname* kurang dari 100, dengan total 23.664 *hostname* atau 70,61% *hostname* yang ada.

Sama halnya dengan Gambar IV.5, Gambar IV.8 juga secara umum menunjukkan urutan F_1 score yang serupa dengan hasil analisis sebelumnya pada subbab IV.5.1. Urutan tersebut menunjukkan metode *reverse lookup* dengan rekaman *query* menempati urutan pertama. Urutan tersebut diikuti oleh metode atribut CN dengan kondisi SNI didukung oleh klien. Kemudian, metode atribut CN dengan kondisi SNI tidak didukung oleh klien mengikuti. Terakhir, metode *reverse lookup* berdasarkan entri DNS menempati urutan terendah.



Gambar IV.8. Nilai F_1 score berdasarkan jumlah *mutual hostname*

Metode *reverse lookup* dengan entri DNS menunjukkan hasil yang konsisten dengan pengaruh dari jumlah alamat IP hasil *lookup* sebelumnya: akurasinya sangat rendah dan sama sekali gagal mengidentifikasi *hostname* yang berbagi dengan lebih dari tiga *hostname* lainnya. Sekali lagi, hal ini menguatkan analisis bahwa metode

ini akan sangat terdampak terhadap penggunaan *virtual hosting* yang masih dicerminkan dari jumlah *hostname* yang saling berbagi alamat IP ini.

Hal yang sama juga ditunjukkan metode atribut CN dari sertifikat TLS tanpa dukungan SNI: metode tersebut hanya dapat mengidentifikasi dengan maksimal ketika *hostname* tersebut memiliki kurang dari sepuluh *mutual hostname*. Di atas itu, metode ini hanya dapat sesekali mengidentifikasi *hostname*. Pada analisis ini ditunjukkan bahwa metode ini juga termasuk yang terdampak penggunaan *virtual hosting* oleh para penyedia layanan.

Sisanya, metode atribut CN dari sertifikat TLS dengan dukungan SNI serta metode *reverse lookup* berdasarkan rekaman *query* menunjukkan hasil yang fluktuatif. Secara umum, metode *reverse lookup* berdasarkan rekaman *query* tetap unggul. Sangat kecil kejadian metode atribut CN menghasilkan akurasi yang lebih tinggi.

Fluktuasi nilai F_1 score dari kedua metode tersebut menyebabkan perlu dilakukan analisis korelasi secara statistik. Dari rumus II.4 dan II.5, nilai koefisien korelasi beserta koefisien penentuan diketahui. Kedua nilai tersebut diuraikan pada Tabel IV.6.

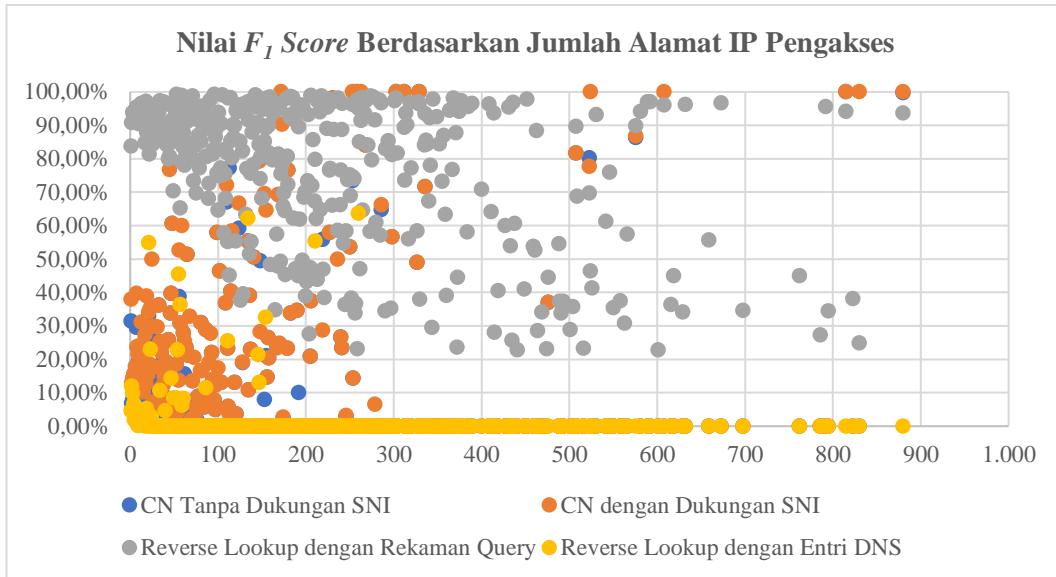
Tabel IV.6. Korelasi jumlah *mutual hostname* dengan nilai F_1 score

Metode	Koefisien korelasi	Koefisien penentuan
<i>Reverse lookup</i> dengan rekaman <i>query</i>	-0,2108	4,45%
<i>Reverse lookup</i> dengan entri DNS	-0,0426	0,18%
Atribut CN ketika SNI didukung klien	-0,0687	0,47%
Atribut CN ketika SNI tidak didukung klien	-0,0522	0,27%

Dari tabel tersebut dapat dilihat bahwa variabel jumlah *mutual hostname* ini maksimal hanya berkontribusi 4,45% terhadap nilai F_1 score yang ada. Dengan demikian, variabel ini bisa disimpulkan tidak berkorelasi dengan nilai F_1 score.

IV.5.4 Pengaruh dari Jumlah Alamat IP Pengakses *Hostname*

Analisis selanjutnya akan diambil dari Gambar IV.9 yang merupakan *scatter plot* nilai F_1 score berdasarkan jumlah alamat IP pengakses.



Gambar IV.9. Nilai F_1 score berdasarkan jumlah alamat IP pengakses

Secara visual, gambar tersebut menunjukkan bahwa semakin sedikit jumlah alamat IP pengguna (artinya, semakin tidak populer suatu *hostname*), maka kemungkinan akurasi dari metode *reverse lookup* berdasarkan rekaman *query* untuk meningkat semakin tinggi. Hal ini dapat diamati dengan visualisasi bahwa semakin populer *hostname* tersebut, meski kecenderungan metode ini tetap memberikan akurasi yang tinggi, namun kemunculan akurasi yang semakin rendah pun meningkat. Hal ini menunjukkan titik lemah dari metode ini.

Hal yang berbeda ditunjukkan oleh metode atribut CN pada sertifikat TLS dengan dukungan SNI pada klien. Pada metode ini, meski akurasinya tetap jauh lebih rendah secara keseluruhan dibandingkan metode *reverse lookup* berdasarkan rekaman *query*, namun ternyata memiliki kecenderungan untuk meningkat akurasinya seiring dengan semakin populernya *hostname* tersebut. Meski demikian, kecenderungan ini hanya terlihat secara visual; analisis dengan metode statistik kemudian menunjukkan bahwa kecenderungan ini tidak signifikan.

Kedua metode lainnya cenderung stabil – tidak menunjukkan peluang bahwa akurasinya akan meningkat ketika jumlah alamat IP pengguna di atas 300.

Analisis ini perlu menjadi pertimbangan untuk memilih salah satu dari keempat metode ini, mengingat jumlah *hostname* yang menikmati popularitas (setidaknya diakses lebih dari 500 orang) hanya 1% dari seluruh *hostname* yang ada.

Analisis korelasi secara statistik dilakukan untuk membantu memahami bagaimana korelasi jumlah alamat IP pengakses terhadap nilai *F₁ score* dari keseluruhan metode yang ada. Korelasi tersebut diuraikan pada Tabel IV.7.

Tabel IV.7. Korelasi jumlah alamat IP pengakses dengan nilai *F₁ score*

Metode	Koefisien korelasi	Koefisien penentuan
Reverse <i>lookup</i> dengan rekaman <i>query</i>	-0,4243	18,01%
Reverse <i>lookup</i> dengan entri DNS	-0,1505	2,27%
Atribut CN ketika SNI didukung klien	-0,0367	0,13%
Atribut CN ketika SNI tidak didukung klien	-0,0415	0,17%

Tabel tersebut menunjukkan bahwa variabel jumlah alamat IP pengakses ini hanya memiliki korelasi yang cukup besar dengan nilai *F₁ score* pada metode *reverse lookup* dengan rekaman *query*. Korelasi yang terjadi merupakan korelasi negatif dengan nilai -0,4243, sehingga berkontribusi terhadap nilai *F₁ score* sebesar 18,01%. Korelasi ini menyatakan bahwa semakin tinggi jumlah alamat IP pengakses, ada kecenderungan bahwa nilai *F₁ score* pada metode tersebut menurun.

Selain pada metode tersebut, dengan nilai koefisien penentuan maksimal 2,27%, dapat disimpulkan bahwa variabel ini tidak berkorelasi terhadap nilai *F₁ score* pada metode lainnya. Dengan demikian, kecenderungan nilai *F₁ score* yang terlihat pada metode atribut CN pada sertifikat TLS dengan dukungan SNI pada klien merupakan kecenderungan yang hanya terlihat secara visual dan bukan disebabkan oleh variabel ini.

IV.5.5 Pengaruh dari Jumlah Kunjungan per *Hostname*

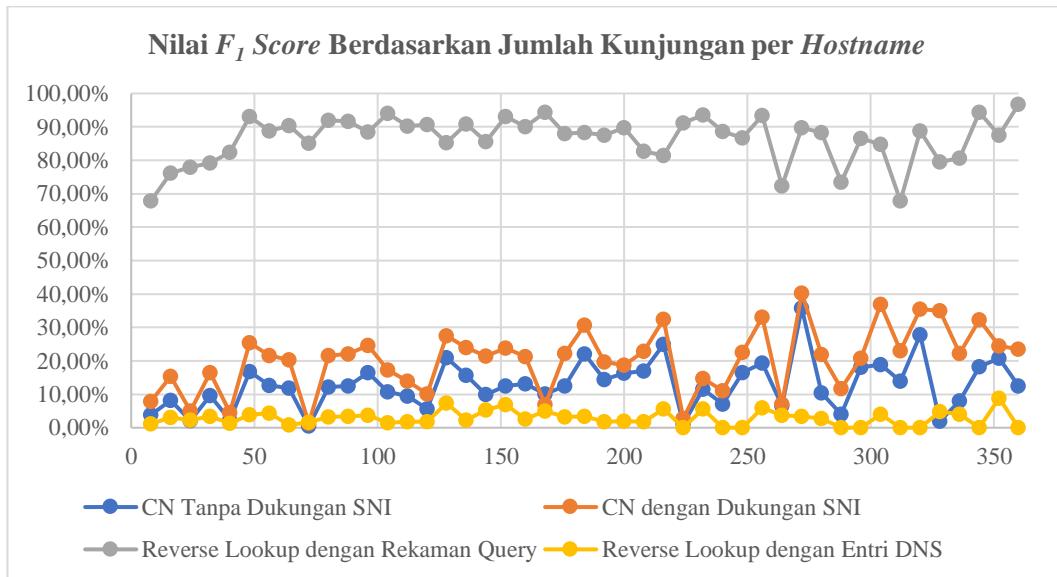
Log data penelitian menunjukkan bahwa paling sedikit suatu *hostname* diakses delapan kali selama empat tahun. Jumlahnya cukup signifikan, yaitu 14.231 *hostname* (42,47%). Sementara, 90% dari *hostname* yang ada (30.216 *hostname*) hanya diakses maksimal 360 kali selama empat tahun.

10% dari *hostname* sisanya diakses cukup sering – ribuan kali – hingga maksimal diakses sebanyak 1.134.512 kali. *Hostname* yang paling banyak diakses tersebut adalah www.youtube.com. Selain itu, beberapa *hostname* populer lainnya ditunjukkan pada Tabel IV.8.

Tabel IV.8. Beberapa *hostname* populer pada log

<i>Hostname</i>	Jumlah Kunjungan
www.youtube.com	1.134.512
r3---sn-2uuxa3vhug5onpu-cuie.googlevideo.com	1.009.312
r2---sn-2uuxa3vhug5onpu-cuie.googlevideo.com	946.480
r1---sn-2uuxa3vhug5onpu-cuie.googlevideo.com	933.960
fbcdn-profile-a.akamaihd.net	759.144
www.facebook.com	707.000
v10.vortex-win.data.microsoft.com	667.544
play.google.com	636.560
www.google.com	614.912

Karena rentang jumlah kunjungan yang besar, maka diambil 90% *hostname* yang diakses maksimal sebanyak 360 kali selama empat tahun untuk divisualisasikan pada Gambar IV.10.



Gambar IV.10. Nilai F_1 score berdasarkan jumlah kunjungan per *hostname*

Sama halnya dengan Gambar IV.5 dan Gambar IV.8, Gambar IV.10 juga secara umum menunjukkan urutan F_1 score yang serupa dengan hasil analisis sebelumnya pada subbab IV.5.1. Urutan tersebut menunjukkan metode *reverse lookup* dengan

rekaman *query* menempati urutan pertama. Urutan tersebut diikuti oleh metode atribut CN dengan kondisi SNI didukung oleh klien. Kemudian, metode atribut CN dengan kondisi SNI tidak didukung oleh klien mengikuti. Terakhir, metode *reverse lookup* berdasarkan entri DNS menempati urutan terendah.

Gambar tersebut hanya dapat menunjukkan bagaimana nilai F_1 score dari keempat metode pada 90% *hostname* yang ada. Untuk melihat gambaran secara keseluruhan, nilai F_1 score minimal, rata-rata, dan nilai maksimal dihitung dan ditunjukkan pada Tabel IV.9. Pada tabel tersebut ditunjukkan bahwa urutan yang disimpulkan dari Gambar IV.10 berlaku secara umum.

Tabel IV.9. Nilai F_1 score minimal, rata-rata, dan maksimal berdasarkan jumlah kunjungan

Metode	Nilai Minimal	Rata-rata	Nilai Maksimal
<i>Reverse lookup</i> dengan rekaman <i>query</i>	15,81%	84,80%	100,00%
<i>Reverse lookup</i> dengan entri DNS	0,00%	2,47%	100,00%
Atribut CN ketika SNI didukung klien	0,00%	15,50%	100,00%
Atribut CN ketika SNI tidak didukung klien	0,00%	11,55%	100,00%

Analisis korelasi dilakukan untuk membantu memahami bagaimana korelasi jumlah kunjungan per *hostname* terhadap nilai F_1 score dari keseluruhan metode yang ada. Korelasi tersebut diuraikan pada Tabel IV.10.

Tabel IV.10. Korelasi jumlah kunjungan per *hostname* dengan nilai F_1 score

Metode	Koefisien korelasi	Koefisien penentuan
<i>Reverse lookup</i> dengan rekaman <i>query</i>	-0,1493	2,23%
<i>Reverse lookup</i> dengan entri DNS	-0,0429	0,18%
Atribut CN ketika SNI didukung klien	-0,0256	0,07%
Atribut CN ketika SNI tidak didukung klien	-0,0247	0,06%

Dari tabel tersebut dapat dilihat bahwa variabel jumlah kunjungan per *hostname* ini maksimal hanya berkontribusi 2,23% terhadap nilai F_1 score yang ada. Dengan demikian, dapat disimpulkan bahwa variabel ini tidak berkorelasi terhadap nilai F_1 score dari seluruh metode yang dieksperimenkan.

IV.5.6 Analisis Keseluruhan

Analisis pada subbab IV.5.1 menunjukkan bahwa metode SNI pada *handshake TLS* merupakan metode terbaik jika aplikasi yang digunakan di Politeknik Negeri

Bandung mendukung penggunaan SNI dengan nilai F_1 score sebesar 100%. Percobaan perlu dilakukan untuk melihat apakah penggunaan SNI sudah didukung oleh aplikasi yang ada atau belum. Hasil percobaan ini ditunjukkan pada subbab V.4. Berdasarkan analisis pada subbab IV.5.1 ini, maka analisis pada subbab IV.5.2, IV.5.3, IV.5.4, dan IV.5.5 tidak mempertimbangkan metode SNI pada *handshake TLS*.

Analisis pada subbab IV.5.2 menunjukkan bahwa dua metode yang terkait dengan DNS cukup terpengaruh oleh jumlah alamat IP hasil *lookup* yang merepresentasikan skala layanan. Kontribusi variabel ini terhadap nilai F_1 score juga cukup besar, yaitu 25,20% untuk metode *reverse lookup* dengan entri DNS dan 30,29% untuk metode *reverse lookup* dengan rekaman *query*.

Analisis pada subbab IV.5.3 menunjukkan bahwa terdapat fluktuasi nilai F_1 score dari metode-metode yang ada. Fluktuasi tersebut kemudian dijawab oleh analisis korelasi yang menunjukkan bahwa variabel jumlah *mutual hostname* tidak berpengaruh terhadap nilai F_1 score.

Analisis pada subbab IV.5.4 kemudian menunjukkan bahwa secara visual, metode *reverse lookup* berdasarkan rekaman *query* memiliki kecenderungan untuk berperforma baik ketika *hostname* tersebut tidak populer (diakses oleh sedikit alamat IP) dengan penurunan performa seiring semakin populernya suatu *hostname*. Interpretasi visual tersebut kemudian dikonfirmasi oleh analisis korelasi yang menunjukkan bahwa pada metode tersebut, variabel jumlah alamat IP pengakses memang berpengaruh terhadap nilai F_1 score, meskipun hanya berkontribusi sebesar 18,01%.

Terakhir, analisis pada subbab IV.5.5 menunjukkan bahwa jumlah kunjungan per *hostname* tidak berpengaruh terhadap nilai F_1 score.

Analisis korelasi yang dilakukan terhadap empat variabel numerik serta variabel nominal pertama (dukungan SNI pada klien) kemudian diringkas pada Tabel IV.11 yang menunjukkan variabel bebas mana saja yang terbukti berpengaruh terhadap variabel terikat pada penelitian ini. Analisis tersebut menunjukkan bahwa variabel-

variabel bebas yang dipilih pada penelitian ini belum menggambarkan seluruh variabel bebas yang mempengaruhi nilai F_1 score dari suatu metode identifikasi *hostname*.

Meski terdapat variabel bebas yang tidak berkorelasi dengan variabel terikat, penelitian ini menggunakan variabel-variabel bebas tersebut untuk mengevaluasi metode identifikasi *hostname* dalam rangka menjawab RQ₁. Metode-metode tersebut dibandingkan sebagai alternatif dari metode SNI pada *handshake* TLS yang diunggulkan pada analisis di subbab IV.5.1.

Dari keempat analisis tersebut, alternatif yang paling kuat jika metode SNI tidak dapat diimplementasi adalah metode *reverse lookup* berdasarkan rekaman *query*. Akurasi dari metode ini secara umum selalu jauh lebih tinggi dari metode-metode lainnya. Namun, opsi lainnya adalah dengan menggabungkan keempat metode tersebut, jika metode SNI tidak dapat diimplementasi sama sekali, sebagaimana yang dilakukan oleh Rao (2013).

Dari analisis tersebut, RQ₁ dapat dijawab: metode yang tepat untuk mengidentifikasi *hostname* pada akses melalui protokol HTTPS dengan hasil yang sama dengan yang diperoleh *explicit web proxy* adalah metode SNI pada *handshake* TLS.

Tabel IV.11. Pengaruh masing-masing variabel bebas terhadap nilai F_1 score

Variabel bebas	Koefisien korelasi								Koefisien penentuan		Mempengaruhi?		
	<i>Reverse lookup</i> dengan entri DNS				<i>Reverse lookup</i> dengan rekaman <i>query</i>				Attribut CN ketika SNI tidak didukung klien		<i>Reverse lookup</i> dengan rekaman <i>query</i>		
Dukungan SNI pada klien											×	×	√
Jumlah alamat IP hasil <i>lookup hostname</i>	0,5504	-0,5021	0,1275	0,1137	30,30%	25,21%	1,63%	1,29%	√	√	×	×	×
Jumlah <i>mutual hostname</i>	-0,2108	-0,0427	-0,0687	-0,0523	4,45%	0,18%	0,47%	0,27%	×	×	×	×	×
Jumlah alamat IP pengakses	-0,4244	-0,1505	-0,0367	-0,0415	18,01%	2,27%	0,13%	0,17%	√	×	×	×	×
Jumlah kunjungan per <i>hostname</i>	-0,1494	-0,0429	-0,0256	-0,0247	2,23%	0,18%	0,07%	0,06%	×	×	×	×	×
<i>Reverse lookup</i> dengan rekaman <i>query</i>													

BAB V

PENERAPAN *TRANSPARENT WEB PROXY*

Bab ini menjelaskan analisis *problem domain* terkait kondisi di Politeknik Negeri Bandung, perancangan jaringan untuk implementasi *transparent web proxy*, pembuatan implementasi skala lab, percobaan aplikasi pengguna dan hasilnya. Bab ini menjawab RQ₂ serta menguatkan jawaban RQ₁.

V.1 Analisis *Problem Domain*

Subbab ini menjelaskan tiga hal yang dianalisis selain penentuan metode identifikasi *hostname* yang sudah dijelaskan pada Bab IV. Tiga hal tersebut adalah analisis permasalahan saat ini, analisis topologi jaringan Politeknik Negeri Bandung, serta analisis bagaimana akses web ke internet dikelola oleh PSI. Hasil analisis ketiga hal tersebut kemudian dievaluasi.

V.1.1 Analisis Permasalahan Saat Ini

Politeknik Negeri Bandung melalui PSI mewajibkan pengguna jaringan intranet untuk menggunakan *explicit web proxy*. Dengan *explicit web proxy*, PSI melakukan autentikasi dan otorisasi pengguna serta pencatatan akses web yang dilakukan. Untuk jangka waktu yang lama (setidaknya sejak tahun 2003 menurut PSI), penerapan *explicit web proxy* ini tidak bermasalah.

Masalah baru muncul akhir-akhir ini ketika mulai terdapat aplikasi yang tidak *proxy-aware* dan meminta akses langsung ke internet, sebagaimana diungkapkan oleh Wilson (2017) dan Yeh (2017). Dalam artikelnya, Wilson bahkan meminta para pengelola jaringan untuk beranjak dari penggunaan *explicit web proxy*.

Dalam menganalisis permasalahan ini, *explicit web proxy* tidak dapat dilihat secara individu. Analisis harus melihat secara utuh arsitektur jaringan yang ada, karena *explicit web proxy* hanyalah salah satu komponen yang menyusun fungsionalitas yang ada pada satu jaringan. Permasalahan yang baru muncul akhir-akhir ini kemudian dapat dijelaskan dengan model referensi TCP/IP (lihat Gambar II.2).

Pada kondisi saat ini di mana *explicit web proxy* diterapkan, kontrol jaringan terhadap akses pengguna hanya diterapkan pada *application layer*. Pengguna (melalui aplikasi yang digunakannya) bertanggung jawab untuk memastikan bahwa aksesnya ke internet akan melalui *explicit web proxy*. Dengan kata lain, pengguna bertanggung jawab untuk memastikan bahwa pada *internet layer*, aksesnya diterima oleh *explicit web proxy*. Hal ini dilakukan oleh pengguna dengan mengkonfigurasi aplikasi yang digunakannya supaya mengakses *explicit web proxy* pada alamat IP dan *port* tertentu.

Permasalahan timbul ketika aplikasi yang digunakan tidak *proxy-aware*: aplikasi tidak memiliki kemampuan untuk memastikan bahwa pada *internet layer*, aksesnya akan tiba di *explicit web proxy*. Sementara jaringan tidak bertanggung jawab untuk memastikan bahwa akses yang dilakukan oleh klien akan sampai pada *explicit web proxy*.

Dengan penerapan *transparent web proxy*, jaringan akan mengambil alih tanggung jawab tersebut dari klien. Jaringan secara aktif akan mengarahkan seluruh akses yang dilakukan oleh pengguna kepada *transparent web proxy* yang ada di dalam jaringan. Dengan kata lain, pada penerapan *transparent web proxy*, komponen-komponen jaringan bertanggung jawab hingga *internet layer*.

Pada penerapan *explicit web proxy* di Politeknik Negeri Bandung, selain aplikasi pengguna harus bertanggung jawab pada *internet layer*, aplikasi pengguna juga harus bertanggung jawab melakukan autentikasi dan otorisasi pengguna dengan berkomunikasi dengan *explicit web proxy*. Tidak semua aplikasi yang digunakan di Politeknik Negeri Bandung melakukan hal ini. Log dari *explicit web proxy* yang disediakan oleh PSI menunjukkan ada banyak akses yang berhasil tiba di *explicit web proxy*, namun tidak ada pesan autentikasi yang dikirimkan dari aplikasi yang digunakan pengguna; sebagian dari akses tersebut ditunjukkan pada Tabel V.1.

Dengan data tersebut, peninjauan kembali penggunaan *explicit web proxy* di Politeknik Negeri Bandung menjadi hal yang penting.

Tabel V.1. Sebagian akses pada *explicit web proxy* tanpa autentikasi

Hostname	Jumlah Akses
ksn-url-geo.kaspersky-labs.com	53.429
d.dropbox.com	46.796
android.clients.google.com	44.546
api.onedrive.com	29.097
ksn-file-geo.kaspersky-labs.com	26.726
graph.instagram.com	20.156
www.googleapis.com	19.128
api.parse.com	18.452
clients4.google.com	18.301
client.wns.windows.com	14.659
clientservices.googleapis.com	14.547
tools.google.com	13.726
policy.ccs.mcafee.com	13.269
watson.telemetry.microsoft.com	6.959
armmf.adobe.com	5.213
client-office365-tas.msedge.net	3.984

V.1.2 Analisis Topologi Jaringan Kampus

Untuk memenuhi kebutuhan akademik, Politeknik Negeri Bandung berlangganan koneksi internet ke dua ISP. Kedua ISP menyediakan *redundant link* untuk koneksi internet. Manajemen *redundant link* ini dilakukan oleh pihak ketiga, sehingga PSI tidak melakukan kontrol terhadap mekanisme *redundant link* yang dilakukan. Karena alasan tersebut pula, pada tugas akhir ini mekanisme koneksi ke ISP dianggap sebagai *blackbox*.

Sebagai bagian dari langganan ini, Politeknik Negeri Bandung mendapatkan alokasi IP publik dengan blok 103.209.131.0/24 dan dipublikasikan dengan nomor AS133355. Blok ini tentu tidak cukup jika dialokasikan ke seluruh perangkat yang ada di dalam kampus. Oleh karena itu, di dalam kampus dibuat intranet menggunakan IP privat dengan segmentasi yang diuraikan pada Tabel V.2.

Intranet tidak memiliki *default route* ke internet, kecuali jaringan WiFi gedung Direktorat, Pendopo Tonny Soewandito, dan gedung Pascasarjana yang memiliki *default route* melalui *transparent web proxy*. Agar dapat mengakses web, PSI kemudian menyediakan *web proxy* dengan satu alamat IP di masing-masing segment jaringan.

Tabel V.2. Segmen intranet kampus beserta *web proxy*-nya

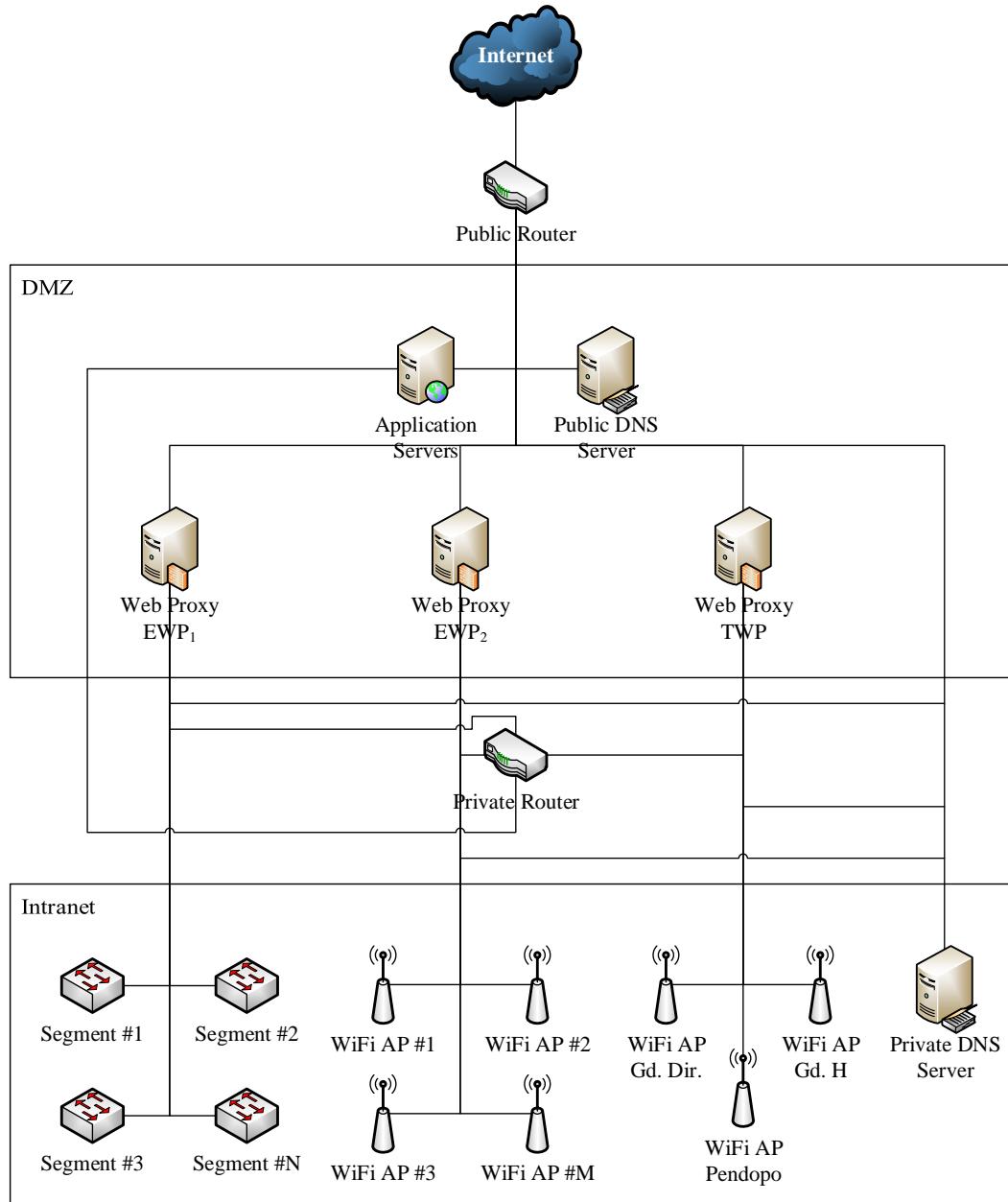
Segmen Jaringan	Peruntukan	Alamat Gateway	Web Proxy
192.168.29.0/24	J. T. Energi	192.168.29.9	EWP ₁ , 192.168.29.1
192.168.32.0/24	J. T. Mesin	192.168.32.3	EWP ₁ , 192.168.32.1
192.168.34.0/24	Prodi. MTRI	192.168.34.3	EWP ₁ , 192.168.34.1
192.168.36.0/24	Gedung P2T	192.168.36.3	EWP ₁ , 192.168.36.1
192.168.41.0/24	UPT Studio Gambar	192.168.41.3	EWP ₁ , 192.168.41.1
192.168.42.0/24	UPT Bahasa	192.168.44.3	EWP ₂ , 192.168.42.1
192.168.44.0/24	Pusat/Direktorat	192.168.44.3	EWP ₂ , 192.168.44.1
192.168.72.0/24	J. T. Komputer	192.168.72.3	EWP ₁ , 192.168.72.1
192.168.73.0/24	J. Administrasi Niaga	192.168.73.3	EWP ₁ , 192.168.73.4
192.168.74.0/24	J. Akuntansi	192.168.74.3	EWP ₁ , 192.168.74.1
192.168.75.0/24	J. T. Refrigerasi	192.168.75.3	EWP ₁ , 192.168.75.1
192.168.76.0/24	J. T. Sipil	192.168.76.7	EWP ₁ , 192.168.76.1
192.168.77.0/24	J. T. Kimia	192.168.77.3	EWP ₁ , 192.168.77.1
192.168.78.0/24	J. T. Elektro	192.168.78.3	EWP ₁ , 192.168.78.1
192.168.92.0/24	UPT Komputer	192.168.92.3	EWP ₁ , 192.168.92.1
10.50.0.0/16	Jaringan Hotspot	10.50.10.3	EWP ₁ , 10.50.20.4 EWP ₂ , 10.50.10.1 TWP, 10.50.20.1
10.69.0.0/16	UPT Komputer	192.168.36.3	EWP ₁ , 192.168.36.1 (via <i>web proxy</i> internal UPTK)

Ada tiga *web proxy* yang dimiliki oleh PSI dengan alamat IP publik dan keperluan yang diuraikan pada Tabel V.3. Ketiga *web proxy* tersebut kemudian dikodekan dengan awalan EWP untuk *explicit web proxy* dan TWP untuk *transparent web proxy*. Alokasi ketiga *web proxy* tersebut juga ditunjukkan pada Tabel V.2. Kode tersebut akan dirujuk pada penjelasan selanjutnya di laporan tugas akhir ini. Namun sebagaimana dijelaskan pada ruang lingkup tugas akhir (lihat subbab I.7), TWP tidak akan diperhatikan karena merupakan eksplorasi dan percobaan PSI.

Tabel V.3. *Web proxy* kampus

Kode	Alamat IP Publik	Area Layanan	Software Web Proxy
EWP ₁	103.209.131.4	Jaringan kabel	Squid 2.6.STABLE22
EWP ₂	103.209.131.3	Jaringan UPT Bahasa, gedung Direktorat, dan jaringan WiFi	Squid 2.6.STABLE22
TWP	103.209.131.9	Jaringan WiFi gedung Direktorat, Pendopo Tonny Soewandito, dan gedung Pascasarjana	Squid 3.1.23

Secara *logic*, topologi tersebut digambarkan pada Gambar V.1.



Gambar V.1. Topologi *existing* akses web intranet kampus

Gambar tersebut menunjukkan hubungan antar komponen dalam perspektif *internet layer* pada model TCP/IP. Garis antara dua komponen menunjukkan bahwa kedua komponen tersebut berada pada satu segmen IP yang sama.

Gambar tersebut menunjukkan bahwa untuk setiap perangkat yang bergabung ke jaringan, dalam kondisi standar akan terhubung langsung dengan tiga komponen

lain dalam satu segmen IP yang sama, yaitu Web Proxy, Private Router, dan Private DNS Server. Namun, meski tersegmentasi berdasarkan *internet layer*, pada *link layer* seluruh perangkat di dalam intranet berada pada *broadcast domain* dan *bandwidth domain* yang sama.

Pada gambar tersebut juga ditunjukkan bahwa Private Router terhubung dengan Application Servers. Hal ini merupakan implementasi dari kebijakan PSI bahwa komputer klien sebaiknya mengakses Application Servers tidak melalui *web proxy*. Pada penerapannya, ketika melakukan konfigurasi *web proxy* di perangkat pengguna, sebuah *exception* terhadap *hostname *.polban.ac.id* perlu dipasang.

Pada intranet, alamat IP untuk perangkat pengguna diatur secara manual oleh unit/jurusank masing-masing. Pengecualian terdapat pada jaringan WiFi, di mana komputer pengguna diberikan alamat IP oleh jaringan melalui protokol *dynamic host configuration protocol* (DHCP).

V.1.3 Analisis Pengelolaan Akses Web ke Internet

Kebutuhan pengelolaan akses web ke internet di Politeknik Negeri Bandung saat ini adalah melakukan autentikasi pengguna yang melakukan akses serta mencatat seluruh akses web ke internet.

Untuk mengetahui kebutuhan pengelolaan lebih dalam, konfigurasi *web proxy* dianalisis. *File* konfigurasi *web proxy* EWP₁ dan EWP₂ diperoleh dari PSI. Sebelum dianalisis, hal-hal berikut dilakukan untuk membersihkan *file* konfigurasi *web proxy* tersebut:

- semua baris konfigurasi yang diawali tanda # (menunjukkan komentar) dihapus,
- semua deklarasi *access control list* (ACL) yang tidak dirujuk pada *rule* mana pun dihapus,
- semua *consecutive newline* dibersihkan, untuk memudahkan pembacaan.

Hasil pembersihan *file* konfigurasi *web proxy* kemudian menunjukkan bahwa konfigurasi pada *web proxy* EWP₁ dan EWP₂ secara prinsip merupakan konfigurasi yang sama.

V.1.3.1 Autentikasi Pengguna

Autentikasi pengguna dilakukan dengan sumber data berformat `htpasswd`. Autentikasi dilakukan ketika aplikasi yang digunakan mengirimkan *request* tanpa kredensial pengguna. *Request* yang tidak mengirimkan kredensial pengguna akan ditolak.

Namun, pada konfigurasi yang sama, terdapat perlakuan bahwa akses ke beberapa situs tertentu tidak memerlukan autentikasi. PSI menyatakan bahwa perlakuan tersebut merupakan pelanggaran *requirement* karena akses ke situs-situs ini dilakukan oleh aplikasi yang tidak mendukung autentikasi pada *web proxy*, antara lain:

- *update* sistem operasi Windows dan OSX,
- akses ke layanan vendor uji kompetensi GMetrix dan Certiport,
- *update* perangkat DataStore 3PAR Blade System,
- fitur-fitur Windows dan Office yang membutuhkan akses web ke internet.

Konfigurasi ini juga melanggar Standar Pengelolaan Jaringan SI bagian 3.2 (lihat Lampiran 2) yang menyatakan bahwa, “untuk semua pengguna internet di lingkungan Politeknik Negeri Bandung diberlakukan autentikasi ketika akan membuka halaman web internet, agar bisa terkontrol penggunaannya.”

V.1.3.2 Identifikasi Akses dengan Protokol HTTPS

Terdapat deklarasi *rule* yang menyatakan bahwa ketika klien meminta *web proxy* membuat akses web dengan protokol HTTPS, server tujuan harus dinyatakan dengan *hostname*, bukan hanya dengan alamat IP yang dituju. Namun, tidak ada *requirement* bahwa URI lengkap dari *resource* yang diakses perlu dicatat. PSI mengonfirmasi kesimpulan tersebut dengan menyatakan bahwa pada akses dengan protokol HTTPS, yang perlu dicatat cukup *hostname*-nya saja.

V.1.3.3 Pencatatan Akses Web

Akses web ke internet yang dilakukan pengguna kemudian dilakukan pencatatan. Pencatatan dilakukan untuk setiap permintaan HTTP atau HTTPS dengan format standar Squid. Hal-hal yang dicatat adalah:

- waktu *request*;

- berapa lama permintaan menunggu hingga mendapatkan respons dari server tujuan;
- alamat IP pengguna;
- indikator apakah akses diotorisasi atau tidak;
- indikator apakah *response* diambil dari *cache* milik *web proxy* atau tidak;
- protokol akses (HTTP atau bukan);
- kode respons HTTP, jika diakses dengan protokol HTTP;
- ukuran respons;
- kode *method* HTTP, jika diakses dengan protokol HTTP;
- alamat lengkap *resource* yang diakses (jika diakses dengan protokol HTTP) atau *hostname* dan *port* yang dituju (jika diakses dengan protokol HTTPS);
- identitas berupa *username* dari pengguna yang mengakses;
- alamat server yang dituju oleh *web proxy*; dan
- jenis *resource* yang diakses, jika diakses dengan protokol HTTPS.

Log yang dihasilkan dari *web proxy* kemudian dianalisis oleh PSI dengan bantuan program LightSquid. Program LightSquid menerima log dengan format standar Squid. Dengan LightSquid, PSI melihat:

- statistik penggunaan per hari pada bulan tertentu, terdiri dari jumlah pengguna, jumlah data yang ditransfer, rata-rata transfer per pengguna, serta *hit rate* dari *cache* di *web proxy*;
- informasi penggunaan dalam satu hari, terdiri dari daftar pengguna, jumlah permintaan akses web, jumlah data yang ditransfer, dan persentase akses pengguna tersebut dibandingkan dengan pengguna lainnya pada hari tersebut;
- situs yang diakses oleh pengguna tertentu pada tanggal tertentu, dikelompokkan berdasarkan *hostname* dan *port* tujuan dengan informasi jumlah permintaan, jumlah data yang ditransfer, beserta persentase akses situs tersebut dibandingkan dengan situs lainnya oleh pengguna pada tanggal tersebut;
- daftar *resource* berukuran besar yang diakses pada tanggal tertentu, ditampilkan dengan alamat lengkap (jika menggunakan protokol HTTP) atau dengan *hostname* dan *port* (jika menggunakan protokol HTTPS), disusun secara kronologis dan dilengkapi dengan identitas pengguna yang mengakses;

- statistik penggunaan per kelompok pengguna (berdasarkan unit dan jurusan) dalam tanggal, bulan, atau tahun tertentu.

Contoh tampilan laporan penggunaan yang dihasilkan oleh LightSquid ditunjukkan pada Gambar V.2.

The screenshot shows the LightSquid user access report interface. At the top left is a "Calendar" with months from 2013 to 2018 and days from 01 to 12. At the top right is the title "Squid user access report" and the date "Date: 23 May 2018 (update :: 23:07 :: 23 May 2018)". Below the calendar are two tables:

Date	Group	Users	Oversize	Bytes	Average	Hit %
27 May 2018	grp	40	22	9.6 G	246.0 M	0.14%
26 May 2018	grp	103	23	37.8 G	375.8 M	0.13%
25 May 2018	grp	702	380	193.0 G	281.5 M	0.20%
24 May 2018	grp	814	405	163.9 G	206.1 M	0.13%
23 May 2018	grp	951	522	279.3 G	300.7 M	0.10%
22 May 2018	grp	863	456	197.9 G	234.8 M	0.23%
21 May 2018	grp	828	424	183.7 G	227.2 M	0.17%
20 May 2018	grp	90	45	27.4 G	312.0 M	0.08%
19 May 2018	grp	101	62	42.4 G	430.3 M	0.05%
18 May 2018	grp	653	355	180.5 G	283.0 M	0.11%
17 May 2018	grp	794	431	166.0 G	214.0 M	0.13%
16 May 2018	grp	756	418	183.0 G	247.8 M	0.17%
15 May 2018	grp	819	455	249.3 G	311.7 M	0.11%
14 May 2018	grp	830	454	247.1 G	304.8 M	0.12%
13 May 2018	grp	113	51	47.0 G	425.9 M	0.09%
12 May 2018	grp	134	70	73.9 G	365.0 M	0.24%
11 May 2018	grp	544	309	213.3 G	401.5 M	0.10%
10 May 2018	grp	124	60	93.6 G	772.6 M	0.08%
09 May 2018	grp	832	464	290.1 G	357.1 M	0.12%
08 May 2018	grp	870	488	302.5 G	356.0 M	0.13%
07 May 2018	grp	901	491	298.6 G	339.4 M	0.10%
06 May 2018	grp	89	37	57.7 G	663.5 M	0.09%
05 May 2018	grp	119	28	93.0 G	800.4 M	0.07%
04 May 2018	grp	648	353	240.5 G	380.1 M	0.12%
03 May 2018	grp	728	400	238.2 G	335.0 M	0.08%
02 May 2018	grp	744	403	192.8 G	265.4 M	0.20%
01 May 2018	grp	106	60	38.8 G	375.1 M	0.12%

Total/Average: 529 286 4340.7 G 370.8 M 0.13%

#	Time	User	Real Name	Connect	Bytes	%	Group
1	00:00:00	[REDACTED]	[REDACTED]	2 398	13.5 G	4.8%	
2	00:00:00	[REDACTED]	[REDACTED]	4 894	12.5 G	4.4%	
3	00:00:00	[REDACTED]	[REDACTED]	5 466	12.1 G	4.3%	
4	00:00:00	[REDACTED]	[REDACTED]	4 730	8.7 G	3.1%	
5	00:00:00	[REDACTED]	[REDACTED]	6 326	7.0 G	2.4%	
6	00:00:00	[REDACTED]	[REDACTED]	4 339	5.3 G	1.8%	
7	00:00:00	[REDACTED]	[REDACTED]	15 519	5.0 G	1.7%	
8	00:00:00	[REDACTED]	[REDACTED]	2 422	4.8 G	1.7%	
9	00:00:00	[REDACTED]	[REDACTED]	1 109	4.8 G	1.7%	
10	00:00:00	[REDACTED]	[REDACTED]	14 932	4.6 G	1.6%	
11	00:00:00	[REDACTED]	[REDACTED]	17 015	4.2 G	1.5%	
12	00:00:00	[REDACTED]	[REDACTED]	1 594	4.2 G	1.5%	
13	00:00:00	[REDACTED]	[REDACTED]	2 049	4.0 G	1.4%	
14	00:00:00	[REDACTED]	[REDACTED]	3 575	3.9 G	1.3%	
15	00:00:00	[REDACTED]	[REDACTED]	4 432	3.8 G	1.3%	
16	00:00:00	[REDACTED]	[REDACTED]	2 862	3.6 G	1.2%	
17	00:00:00	[REDACTED]	[REDACTED]	16 027	3.2 G	1.1%	
18	00:00:00	[REDACTED]	[REDACTED]	3 641	3.1 G	1.1%	
19	00:00:00	[REDACTED]	[REDACTED]	4 386	2.7 G	0.9%	
20	00:00:00	[REDACTED]	[REDACTED]	169 026	2.6 G	0.9%	
21	00:00:00	[REDACTED]	[REDACTED]	1 996	2.5 G	0.9%	
22	00:00:00	[REDACTED]	[REDACTED]	4 426	2.4 G	0.8%	
23	00:00:00	[REDACTED]	[REDACTED]	1 178	2.4 G	0.8%	
24	00:00:00	[REDACTED]	[REDACTED]	3 959	2.4 G	0.8%	
25	00:00:00	[REDACTED]	[REDACTED]	2 835	2.3 G	0.8%	
26	00:00:00	[REDACTED]	[REDACTED]	5 866	2.3 G	0.8%	

Gambar V.2. Contoh *report* yang dikeluarkan LightSquid

V.1.4 Evaluasi Hasil Analisis

Dari analisis pada subbab V.1.1, V.1.2, dan V.1.3 tersebut, terdapat hal-hal yang perlu dievaluasi sebagai berikut:

- distribusi beban dari EWP₁ dan EWP₂ tidak berdasar.

Jumlah unit/jurusan yang dilayani oleh kedua *web proxy* tersebut tidak seimbang, sedangkan PSI tidak mengungkapkan mengapa unit/jurusan tertentu dilayani oleh *web proxy* tertentu;

- PSI menggunakan Squid dengan versi lawas dan berbahaya.

EWP₁ dan EWP₂ menggunakan Squid versi 2.6 yang dirilis tahun 2006, sementara TWP menggunakan Squid versi 3.1 yang dirilis tahun 2010. Kedua versi tersebut sudah dinyatakan berbahaya berdasarkan dokumen *vulnerability*

CVE-2014-7141, CVE-2014-7142, CVE-2014-6270, CVE-2014-3609, CVE-2014-0128, dan CVE-2009-0801 dan harus segera dihentikan penggunaannya. Selain itu, hal tersebut menunjukkan bahwa pemeliharaan PSI terhadap komponen jaringan yang digunakan sangat minim;

- seluruh perangkat jaringan intranet (termasuk perangkat pengguna) berada pada *bandwidth domain* dan *broadcast domain* yang sama.

Oppenheimer (2011) menyatakan kondisi tersebut tidak baik. Selain itu, PSI sendiri juga mengungkapkan sering terjadi masalah akibat *bandwidth domain* dan *broadcast domain* yang sama. Masalah tersebut adalah *broadcast storm* yang melumpuhkan seluruh jaringan kampus serta adanya pengguna yang salah melakukan konfigurasi alamat IP sehingga mengakibatkan konflik dengan pengguna di segmen jaringan lain.

Slameta (2013) yang juga melakukan analisis topologi jaringan Politeknik Negeri Bandung mengungkapkan bahwa VLAN perlu digunakan untuk melakukan pemisahan *link layer* antar segmen yang sudah terpisah pada *internet layer*. Slameta bahkan sudah membuktikan bahwa dengan penggunaan VLAN, performa jaringan bahkan meningkat secara signifikan;

- manajemen kredensial pengguna dilakukan dengan cara yang rawan kesalahan. Subbab V.1.3.1 menunjukkan bahwa kredensial pengguna disimpan dengan *file* berformat `htpasswd`. Dengan format *file* tersebut, kredensial pengguna yang sudah tidak berhak mengakses jaringan (mahasiswa yang sudah lulus maupun *drop out*, atau staf yang sudah tidak bekerja lagi) pada kenyataannya masih dapat mengakses jaringan. Pada log yang dianalisis, hal tersebut benar-benar terjadi;

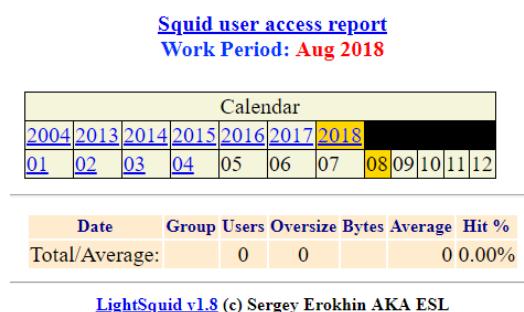
- tidak semua akses web dilakukan autentikasi.

Subbab V.1.3.1 juga menunjukkan bahwa terdapat beberapa layanan yang dapat diakses tanpa autentikasi. Konfigurasi tersebut tidak karena menyebabkan Standar Pengelolaan Jaringan SI yang ditetapkan oleh Politeknik Negeri Bandung tidak dipenuhi;

- pada EWP₁, log tidak dianalisis sejak bulan Mei 2018.

Gambar V.3 menunjukkan bahwa pada EWP₁, log yang dihasilkan dari *web proxy* tidak dianalisis oleh LightSquid. Hal tersebut menunjukkan bahwa

pencatatan akses yang dilakukan oleh PSI tidak konsisten dan tidak ditindaklanjuti, disimpulkan dari dibiarkannya kondisi ini hingga laporan tugas akhir ini diterbitkan.



Gambar V.3. Log pada EWP₁ yang berhenti dianalisis

Hal lain yang perlu dicermati adalah bahwa pengguna saat ini hanya mendapatkan akses DNS melalui Private DNS Server yang tidak terhubung ke internet. Private DNS Server ini juga hanya dapat melakukan *resolve* terhadap *hostname* *.polban.ac.id. Pada penerapan *transparent web proxy*, pengguna perlu mendapatkan akses DNS ke internet, sehingga pada jaringan yang baru, server DNS yang ada di jaringan juga perlu melakukan *resolve* terhadap *hostname* yang valid di internet. Server DNS ini juga perlu bisa diakses oleh pengguna.

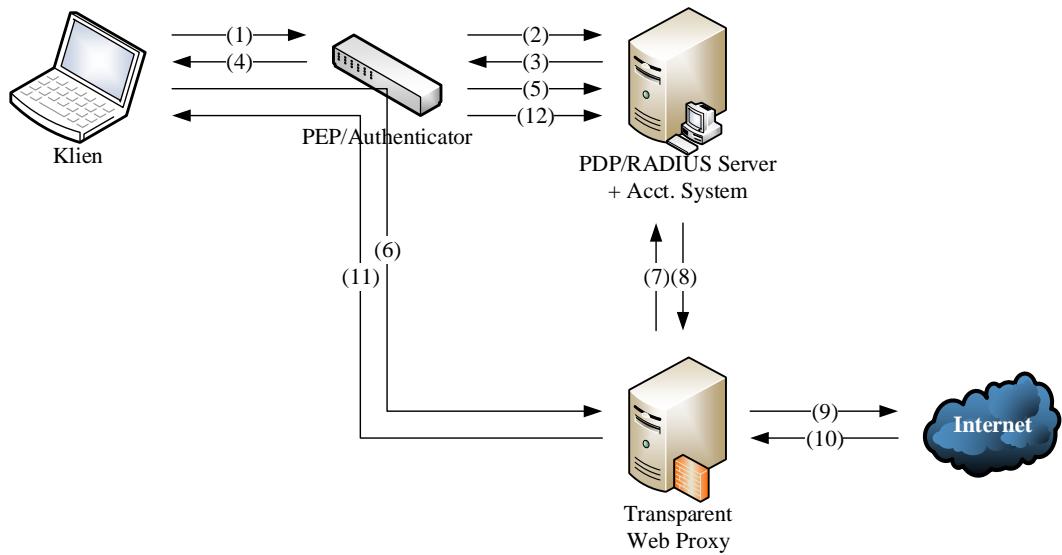
V.2 Perancangan Jaringan dengan *Transparent Web Proxy*

Ada tiga hal yang perlu dirancang pada jaringan yang akan menggunakan *transparent web proxy*: bagaimana autentikasi pengguna dilakukan hingga pengguna mendapatkan akses ke internet, bagaimana topologi jaringan yang dibutuhkan, serta bagaimana konfigurasi yang dibutuhkan.

Uraian pada subbab ini menjadi jawaban untuk RQ₂.

V.2.1 Alur Autentikasi Hingga Akses Web ke Internet

Berdasarkan teori yang dibahas oleh Convery (2007b) serta kebutuhan *transparent web proxy* untuk mengetahui siapa pengguna yang sedang mengakses jaringan, alur autentikasi hingga klien mendapatkan akses web ke internet dirancang dan digambarkan pada Gambar V.4. Alur autentikasi ini menggunakan protokol RADIUS berdasarkan teori pada subbab II.1.8.2.



Gambar V.4. Alur autentikasi klien hingga akses web ke internet

Alur tersebut adalah sebagai berikut:

1. klien terhubung ke PEP dan dimintai kredensial oleh PEP. Klien mengirimkan kredensial pengguna ke PEP;
2. PEP mengirimkan kredensial tersebut ke PDP/server RADIUS. Server RADIUS kemudian menggunakan PIP untuk memperoleh informasi sebagai dasar keputusan;
3. PDP mengirimkan keputusan apakah akses diberikan atau ditolak;
4. PEP mengirimkan hasil keputusan PDP kepada klien.

Mekanisme komunikasi antara klien dengan PEP menggunakan protokol IEEE 802.1X jika perangkat PEP merupakan *wireless access point* yang sudah mendukung protokol tersebut. Jika tidak, maka protokol HTTP dengan mekanisme *captive portal* diterapkan.

Jika akses diotorisasi, maka langkah ke-5 hingga langkah ke-8 dijalankan:

5. PEP mengirimkan pesan akuntansi (berupa pesan Accounting-Request dengan jenis Start) kepada PDP sebagai informasi bahwa sesi akses sudah diberikan kepada klien. PEP harus mengirimkan minimal dua informasi penting bagi *web proxy*, yaitu **Framed-IP-Address** yang berisi alamat IP klien dan **User-Name** yang berisi identitas pengguna;

6. klien melakukan akses web ke internet, dan ditangkap oleh *transparent web proxy*. *Transparent web proxy* melakukan identifikasi akses (dengan SNI jika menggunakan protokol HTTPS);
7. *transparent web proxy* melakukan *lookup* terhadap alamat IP klien untuk memperoleh identitas pengguna. *Lookup* dilakukan terhadap *accounting and reporting system*;
8. *Accounting and reporting system* mengirimkan hasil *lookup* terhadap alamat IP klien.

Jika alamat IP klien berhasil di-*lookup* dan akses tersebut diizinkan dari sudut pandang *web proxy*, maka langkah ke-9 hingga langkah ke-11 dijalankan:

9. *transparent web proxy* meneruskan permintaan klien tersebut ke server tujuan;
10. server tujuan mengirimkan respons terhadap permintaan *web proxy*;
11. *transparent web proxy* meneruskan respons ke klien.

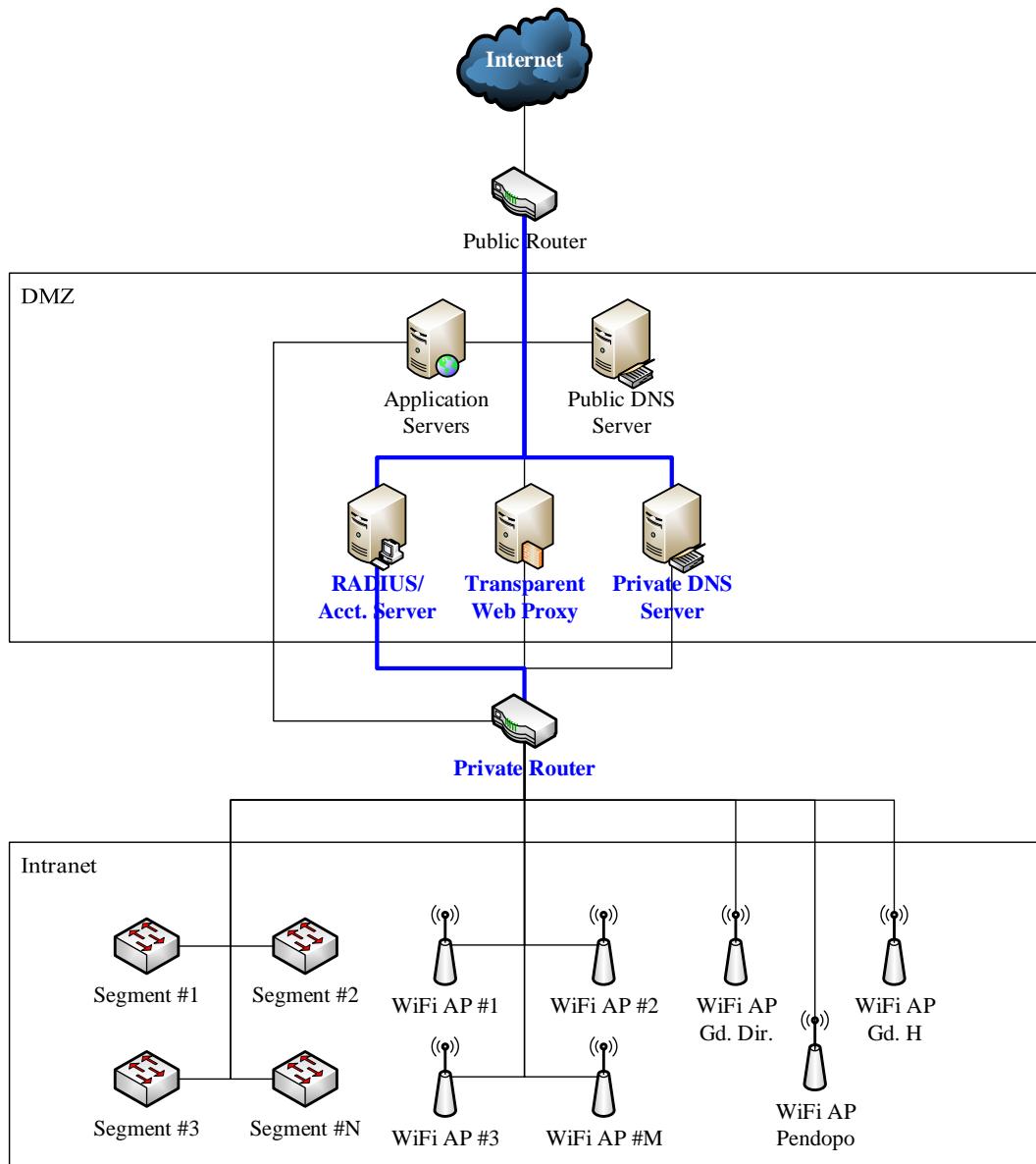
Jika klien keluar dari jaringan, maka langkah ke-12 dijalankan:

12. PEP mengirimkan pesan akuntansi (berupa pesan Accounting-Request dengan jenis Stop) kepada PDP sebagai informasi bahwa sesi akses klien sudah selesai.

V.2.2 Topologi Jaringan yang Dibutuhkan

Untuk mendukung alur autentikasi tersebut serta memenuhi kebutuhan yang ada untuk penerapan *transparent web proxy*, topologi seperti pada Gambar V.5 dibutuhkan. Meski terlihat hanya mengubah *explicit web proxy* menjadi *transparent web proxy*, namun topologi jaringan perlu diubah. Perubahan perlu dilakukan *web proxy* hanyalah salah satu komponen dari jaringan yang bekerja sama dalam sistem besar. Selain itu, alur autentikasi yang dirumuskan pada subbab V.2.1 melibatkan komponen lain di luar *transparent web proxy*.

Pada gambar tersebut, yang dicetak tebal berwarna biru merupakan perubahan pada topologi jaringan yang baru bila dibandingkan dengan topologi *existing*.



Gambar V.5. Topologi usulan akses web intranet kampus

Perubahan kedua adalah pemindahan server DNS yang semula berada di intranet ke jaringan DMZ. Hal ini karena server DNS yang mulanya hanya melakukan *resolve* terhadap *.polban.ac.id saja, kini harus melakukan *resolve* terhadap *hostname* apa pun yang valid di internet. Dengan dipindahkannya server DNS ini ke jaringan DMZ, server DNS mendapatkan akses langsung ke internet dan dapat melakukan semua fungsi administratif DNS dengan lancar. Pengguna juga kemudian dapat melakukan *resolve* terhadap *hostname* apa pun yang valid di internet. Peningkatan jumlah *lookup* DNS hingga dua kali lipat akibat dua pihak (klien dan *web proxy*) akan melakukan *lookup* terhadap *hostname* yang sama (lihat

Tabel II.2) juga mengakibatkan penting bagi *transparent web proxy* dan pengguna untuk menggunakan server DNS yang sama.

Perubahan ketiga dan keempat tentu saja adalah mengubah semua *explicit web proxy* menjadi *transparent web proxy* serta menambah server RADIUS sebagai PDP dan *accounting and reporting system*.

Pada alur autentikasi menggunakan RADIUS, *web proxy* akan mengetahui identitas pengguna melalui alamat IP yang di-intercept oleh *transparent web proxy*. Dengan demikian, penting bagi jaringan tersebut untuk memastikan bahwa satu alamat IP tersebut terasosiasi dengan satu identitas pengguna saja. Kondisi tersebut bisa tidak terjadi jika di bawah *web proxy* terdapat klien yang melakukan Network Address Translation (NAT) bagi banyak klien di bawahnya. Oleh karena itu, secara kebijakan, perlu dipastikan bahwa di intranet tidak ada penggunaan NAT sama sekali.

V.2.3 Konfigurasi yang Dibutuhkan

Dari topologi yang sudah diuraikan tersebut, konfigurasi yang dibutuhkan adalah sebagai berikut:

- pada *router*:
 - semua akses web ke internet melalui *port 80* (untuk HTTP) dan *443* (untuk HTTPS) harus diarahkan ke *transparent web proxy* (sehingga menjadi *default route*);
- pada *transparent web proxy*:
 - menerima semua akses web yang diarahkan dari *router* agar diproses oleh *web proxy* (misalnya pada *kernel Linux* dengan fitur NAT pada *iptables*),
 - melakukan identifikasi akses web dengan protokol HTTPS menggunakan SNI,
 - melakukan *resolve hostname* server tujuan ke server DNS yang sama dengan yang digunakan oleh klien dan yang melakukan *caching* terhadap permintaan DNS klien,
 - melakukan *resolve* terhadap alamat IP klien ke *accounting and reporting system*,

- melakukan pencatatan akses sedemikian rupa sehingga menghasilkan setidaknya lima jenis laporan penggunaan, yaitu:
 - statistik penggunaan per hari pada bulan tertentu,
 - informasi penggunaan dalam satu hari,
 - situs yang diakses oleh pengguna tertentu pada tanggal tertentu,
 - daftar *resource* berukuran besar yang diakses pada tanggal tertentu, dan
 - statistik penggunaan per kelompok pengguna (berdasarkan unit dan jurusan) dalam tanggal, bulan, atau tahun tertentu;
- pada server DNS (yang dipindahkan dari intranet ke jaringan DMZ):
 - melakukan *resolve* terhadap semua *hostname* yang valid di internet,
 - melakukan *caching* terhadap permintaan DNS klien beserta responsnya;
- pada PEP/*authenticator*:
 - melakukan komunikasi dengan PDP menggunakan protokol RADIUS,
 - mengirimkan pesan akuntansi ke PDP terhadap setiap sesi yang dibuat dengan klien, dengan minimal dua informasi penting bagi *web proxy*: *Framed-IP-Address* yang berisi alamat IP klien dan *User-Name* yang berisi identitas pengguna;
- pada PDP/server RADIUS:
 - melakukan komunikasi dengan PIP untuk mengambil keputusan,
 - meneruskan pesan akuntansi ke *accounting and reporting system* sebagai bahan *lookup* bagi *web proxy*;
- pada *accounting and reporting system* (digambarkan satu *node* dengan PDP pada gambar):
 - menyimpan setiap sesi yang dikirim PEP melalui PDP untuk menjadi bahan *lookup* bagi *web proxy*,
 - menyediakan data sesi bagi *web proxy*.

V.3 Pembuatan Implementasi Skala Lab

Untuk menjawab pertanyaan yang diungkapkan pada subbab IV.5 mengenai apakah aplikasi-aplikasi yang digunakan di Politeknik Negeri Bandung sudah mendukung SNI, maka jaringan yang sudah dimodelkan pada subbab V.2 diimplementasi dalam skala lab.

Pada skala lab, hanya akan dimodelkan segmen jaringan Jurusan Teknik Komputer dan Informatika (192.168.72.0/24) sebagai jaringan pengguna, kemudian jaringan DMZ dengan dua blok, yaitu blok 192.168.99.0/24 dari sisi intranet dan blok 172.16.16.0/24 dari sisi internet (mewakili blok alamat IP publik 103.209.131.0/24 milik Politeknik Negeri Bandung), serta akses ke internet. Private Router dan Public Router masing-masing tetap diimplementasi di lingkungan GNS3, tetapi pada lingkungan *deployment* hanya akan diimplementasi dalam satu buah *router* dengan konfigurasi sedemikian rupa sehingga mewakili dua fungsi *router* pada model tersebut. Application Servers juga tidak diimplementasi pada skala lab ini.

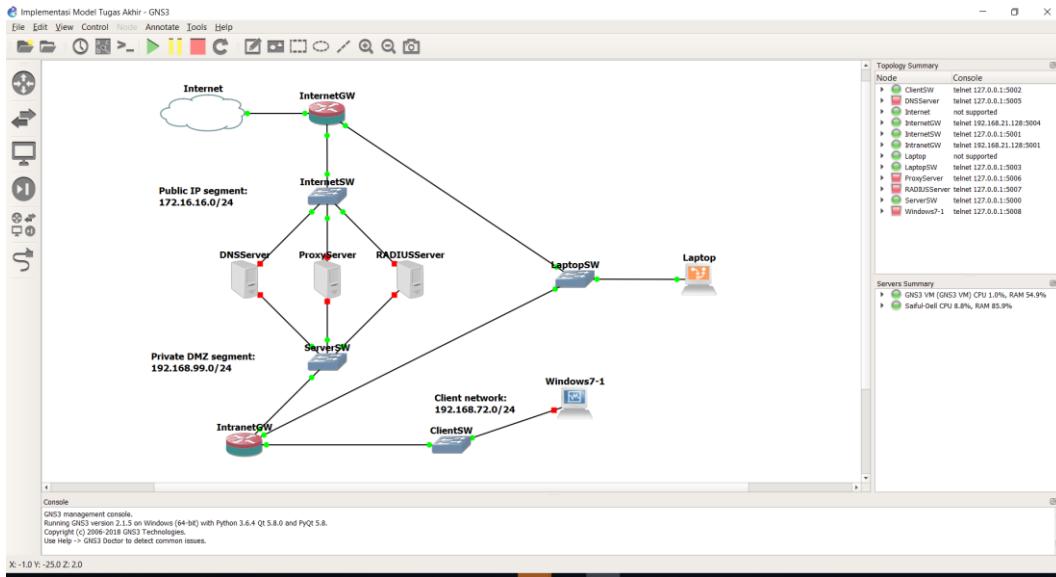
V.3.1 Implementasi di Lingkungan GNS3

Sebelum implementasi di lingkungan *deployment*, dilakukan terlebih dahulu implementasi di lingkungan simulasi menggunakan GNS3. GNS3 adalah *software* yang dapat memodelkan suatu topologi jaringan dan interaksi antar komponen yang diimplementasi berupa *virtual machine*.

Topologi yang sudah dirumuskan pada subbab V.2.2 disimulasikan di GNS3 (lihat Gambar V.6). Hanya satu dari segmen intranet kampus (lihat Tabel V.2) yang disimulasikan, yaitu segmen 192.168.72.0/24. Satu segmen ini cukup mewakili segmen-segmen lainnya karena memiliki karakteristik yang sama.

Gambar V.6 menunjukkan simulasi segmen jaringan lainnya, yaitu segmen IP publik yang memiliki alamat 172.16.16.0/24 (merepresentasikan segmen IP 103.209.131.0/24 di Politeknik Negeri Bandung) serta segmen intranet 192.168.99.0/24. Kedua segmen ini tersambung pada jaringan DMZ tempat server-server aplikasi milik Politeknik Negeri Bandung ditempatkan. Pada jaringan DMZ kemudian diletakkan server DNS, server RADIUS, dan server *transparent web proxy* mengikuti topologi yang dirumuskan pada subbab V.2.2.

Pada gambar tersebut, *node* Laptop merupakan laptop fisik yang digunakan untuk melakukan administrasi GNS3. Pada laptop fisik, sebuah *interface loopback* digunakan untuk tersambung ke kedua *router* melalui *data link layer*.



Gambar V.6. Implementasi di lingkungan GNS3

Pada segmen jaringan intranet pengguna yang disimulasikan di GNS3 (segmen **192.168.72.0/24**), dipasang sebuah *virtual machine* bersistem operasi Windows 7 yang bertindak sebagai pengguna. Pada *virtual machine* ini beberapa *web browser* (Internet Explorer, Mozilla Firefox, dan Google Chrome) digunakan untuk melakukan autentikasi pengguna ke jaringan (lihat subbab V.2.1) hingga beraktivitas dengan melakukan *browsing*.

Sesuai dengan uraian pada subbab III.6.10, *router* yang ada di lingkungan GNS3 menggunakan *virtual machine* bersistem operasi MikroTik RouterOS. Seluruh server yang ada menggunakan sistem operasi Debian 9 untuk menyerupai Raspbian yang akan digunakan pada Raspberry Pi 3 Model B.

Squid digunakan sebagai *software* untuk implementasi *transparent web proxy* sesuai subbab III.6.10. Identifikasi dengan metode SNI telah tersedia pada Squid 4.1. Namun, pada *binary* Squid yang disediakan dari *repository* Debian modul identifikasi dengan metode SNI tersebut tidak dikompilasi karena konflik lisensi antara Debian dengan OpenSSL, *library* yang dibutuhkan untuk melakukan fungsi terkait protokol TLS.

Untuk menangani hal tersebut, maka Squid perlu di-*build* ulang dari *source*-nya dengan menambah tiga buah *flag*, yaitu `--enable-ssl`, `--enable-ssl-crtd`, dan `--with-openssl`. Di lingkungan GNS3, *build* ulang ini memakan waktu 30 menit.

Pada Private Router yang menggunakan MikroTik RouterOS, akses web dengan protokol HTTPS diarahkan ke *transparent web proxy* dengan fitur *mangle*. Fitur NAT pada MikroTik RouterOS tidak digunakan untuk mengarahkan akses web karena Squid membutuhkan akses ke tabel NAT dari sistem operasi tempatnya beroperasi (Debian 9) untuk menentukan alamat IP klien. Fitur NAT baru digunakan di server *transparent web proxy* untuk menangkap akses di *port* 80 dan 443. Masing-masing kemudian diarahkan ke *port* 3129 dan 3130 yang melayani *interception* HTTP dan HTTPS.

Pada implementasi di GNS3 ini, Application Servers dan Public DNS Server tidak diimplementasi karena eksperimen tidak membutuhkan akses ke aplikasi yang berada di jaringan DMZ (tidak ada aplikasi Politeknik Negeri Bandung yang ditempatkan di jaringan skala lab). Kedua hal ini juga tidak esensial untuk dicoba pada jaringan skala lab karena tidak berkaitan langsung dengan akses pengguna ke internet yang menjadi kajian tugas akhir ini.

Seluruh konfigurasi yang dihasilkan pada implementasi di lingkungan GNS3 ini dapat dilihat pada Lampiran 5.

V.3.2 Implementasi di Lingkungan *Deployment*

Karena implementasi dilakukan dalam skala lab, maka dua *hardware* yang disebutkan pada subbab III.6.10 nyatanya memegang lebih dari satu peran yang disebutkan dalam model. Implementasi lebih dari satu peran ini dilakukan dengan konfigurasi yang hati-hati, sehingga tetap memenuhi model usulan.

MikroTik hEX (RB750Gr3) difungsikan sebagai Private Router, Public Router, dan PEP/*authenticator*. Protokol autentikasi yang digunakan adalah HTTP dengan mekanisme *captive portal* (bernama *hotspot* dalam terminologi RouterOS). Sementara itu, Raspberry Pi 3 Model B difungsikan sebagai *transparent web proxy*, PDP/server RADIUS, *accounting and reporting system*, serta Private DNS Server.

Sama halnya dengan implementasi di lingkungan GNS3, *binary* Squid 4.1 yang disediakan dari *repository* Raspbian tidak menyertakan modul identifikasi menggunakan metode SNI. *Build* ulang dengan cara yang serupa dengan cara pada implementasi di GNS3 dilakukan dan memakan waktu 110 menit.

Karena memodelkan pengelolaan akses web di Politeknik Negeri Bandung, dibutuhkan akses langsung ke internet tanpa melalui *web proxy* yang dikelola PSI. Atas dukungan yang diberikan Pembantu Direktur IV Bidang Perencanaan dan Pengembangan serta PSI (dokumen perizinan terlampir pada Lampiran 3), diberikan jalur internet tanpa melalui *web proxy* yang disambungkan melalui *distribution switch* yang berada di Jurusan Teknik Komputer dan Informatika. Alamat IP yang digunakan untuk implementasi tugas akhir ini adalah **103.209.131.60**.

Tampilan dari kedua perangkat ini ditunjukkan pada Gambar V.7.



Gambar V.7. Perangkat implementasi skala lab yang dipasang di JTK

Konfigurasi pada Raspberry Pi 3 Model B mengikuti konfigurasi di lingkungan GNS3, sedangkan konfigurasi pada Mikrotik hEX disertakan pada Lampiran 6.

V.4 Percobaan Aplikasi Pengguna

Aplikasi yang sudah dikumpulkan pada subbab III.6.4 kemudian dicoba menggunakan implementasi skala lab ini. Percobaan dilakukan dengan menggunakan perangkat komputer yang juga digunakan untuk melakukan *development* pada tugas akhir ini (disebutkan pada subbab IV.2). Hasil dari percobaan ini dijelaskan pada subbab V.5.

V.5 Evaluasi Hasil Percobaan

Hasil percobaan aplikasi pada implementasi skala lab ditunjukkan pada Tabel V.4.

Tabel V.4. Hasil percobaan pada implementasi skala lab

Nama Aplikasi	Hasil Percobaan
Android Studio	Teridentifikasi
Docker	Teridentifikasi
Dropbox	Teridentifikasi
Git	Teridentifikasi
GitHub Desktop	Teridentifikasi
GitKraken	Teridentifikasi
Google Chrome	Teridentifikasi
LINE	Teridentifikasi
Mozilla Firefox	Teridentifikasi
NetBeans	Teridentifikasi
NPM Package Manager	Teridentifikasi
PyCharm	Teridentifikasi
Visual Studio	Teridentifikasi

Pada percobaan tersebut, tiga belas aplikasi yang dikumpulkan pada subbab IV.2 berhasil mendapat akses ke internet serta dapat diidentifikasi seluruhnya pada *transparent web proxy*. Aplikasi-aplikasi tersebut juga dapat bekerja dengan lancar tanpa konfigurasi *explicit web proxy* apa pun.

Dengan demikian, hasil percobaan ini menguatkan jawaban terhadap RQ₁, bahwa metode SNI yang sudah terpilih dapat mengidentifikasi *hostname* dari akses dengan protokol HTTPS dengan hasil yang sama dengan yang didapatkan oleh *explicit web proxy* untuk pemakaian di Politeknik Negeri Bandung. Pemakaian ini diwakili oleh ketiga belas aplikasi yang dicobakan pada implementasi skala lab tersebut.

BAB VI

KESIMPULAN DAN SARAN

Bab ini mengungkapkan kesimpulan dari penelitian yang dilakukan dan saran untuk pengembangan selanjutnya.

VI.1 Kesimpulan

Tugas akhir ini berangkat dari latar belakang utama bahwa penerapan *explicit web proxy* di Politeknik Negeri Bandung mulai mengalami kendala akibat munculnya aplikasi-aplikasi yang membutuhkan akses langsung ke jaringan internet. Solusi yang diharapkan, yaitu menggunakan *transparent web proxy*, tidak bisa langsung diimplementasi, karena terdapat akses dengan protokol HTTPS yang membutuhkan metode khusus untuk mengidentifikasinya.

Studi-studi terkini kemudian mengungkapkan empat buah metode identifikasi *hostname* pada akses dengan protokol HTTPS, yaitu dengan *reverse lookup* terhadap entri DNS, *reverse lookup* terhadap rekaman *query* DNS pengguna, menggunakan SNI, serta menggunakan atribut CN dari sertifikat TLS yang diberikan ketika *handshake* TLS. Masing-masing studi mengungkapkan metode-metode tersebut, tetapi perlu dilakukan perbandingan terhadap keempat metode ini karena belum ada studi yang membandingkan keempat metode tersebut. Perbandingan juga perlu dilakukan karena situasi mengenai SNI dan adopsi protokol HTTPS yang sudah meningkat. Fenomena *virtual hosting* juga menjadi perhatian dalam tugas akhir ini.

Tugas akhir ini kemudian melakukan eksperimen untuk menemukan metode mana yang paling cocok diterapkan di Politeknik Negeri Bandung. Penelitian dilakukan secara kuantitatif dengan teknik eksperimental dengan melakukan simulasi akses internet yang dilakukan oleh pengguna terhadap *transparent web proxy* yang mencoba melakukan identifikasi *hostname* dengan keempat metode tersebut. Ada lima variabel bebas yang diamati, yaitu dukungan SNI pada klien, jumlah alamat IP hasil *lookup* terhadap suatu *hostname*, jumlah *mutual hostname*, jumlah alamat

IP pengakses, serta jumlah kunjungan per *hostname*. Variabel terikat yang diamati adalah nilai *F₁ score* yang mewakili akurasi dari keempat metode tersebut untuk masing-masing *hostname*. Data yang digunakan sebagai *ground truth* untuk simulasi akses pengguna adalah log dari *explicit web proxy* yang diberikan oleh PSI.

Dari eksperimen tersebut, metode SNI ditemukan memberikan akurasi 100% terhadap *ground truth* ketika klien mendukung penggunaannya. Sementara, terdapat metode *reverse lookup* berdasarkan rekaman *query* pengguna yang memberikan hasil cukup baik, yaitu sebesar 75,82%. Metode melihat atribut CN pada sertifikat TLS memberikan hasil yang rendah sebesar 18,72% ketika klien mendukung SNI dan 13,60% ketika klien tidak mendukung SNI. Namun, metode *reverse lookup* dengan entri DNS memberikan hasil paling rendah, yaitu sebesar 1,53%. Hal ini menunjukkan bahwa kedua metode tersebut tidak layak sama sekali dijadikan pilihan.

Analisis korelasi yang dilakukan pada tugas akhir ini juga menemukan bahwa jumlah *mutual hostname* serta jumlah kunjungan per *hostname* sebenarnya tidak berkorelasi dengan nilai *F₁ score* dari masing-masing metode.

Kondisi *existing* di Politeknik Negeri Bandung terkait topologi jaringan serta pengelolaan akses web kemudian dianalisis untuk melihat apakah sudah memadai untuk menerapkan *transparent web proxy* dengan metode identifikasi SNI. Hasil analisis menunjukkan bahwa kondisi *existing* saat ini belum memadai. Bahkan, ditemukan hal-hal yang tidak sesuai dengan teori dasar serta melanggar standar pengelolaan yang ditetapkan Politeknik Negeri Bandung. Evaluasi tersebut dapat dibaca pada subbab V.1.4.

Perancangan jaringan pun dilakukan dengan modifikasi sekecil mungkin terhadap topologi jaringan yang ada. Hasil dari perancangan jaringan ini adalah sebuah model jaringan yang terdiri dari alur autentikasi, topologi, dan konfigurasi yang dibutuhkan untuk mengimplementasi *transparent web proxy* dengan metode identifikasi SNI.

Metode SNI tentu menjadi harapan, namun terdapat keraguan apakah aplikasi-aplikasi yang digunakan di Politeknik Negeri Bandung sudah mendukung SNI dan bisa diidentifikasi *hostname*-nya. Untuk menjawab hal tersebut, maka model jaringan yang sudah dihasilkan diimplementasi dalam skala lab. Untuk mewakili aplikasi yang digunakan di Politeknik Negeri Bandung, digunakanlah tiga belas aplikasi yang digunakan mahasiswa Jurusan Teknik Komputer dan Informatika.

Hasilnya, seluruh aplikasi yang digunakan mendukung SNI dan seluruhnya dapat diidentifikasi pada *transparent web proxy*. Dengan demikian, maka metode SNI dibuktikan dapat digunakan untuk mengidentifikasi akses internet dari aplikasi yang digunakan di Politeknik Negeri Bandung. Tugas akhir ini juga menghasilkan model jaringan yang siap diimplementasi oleh Politeknik Negeri Bandung jika dibutuhkan.

VI.2 Saran

Hasil analisis pada subbab IV.5.6 menunjukkan bahwa dua hal berikut dapat dilakukan untuk mengembangkan tugas akhir ini, yaitu:

- mencari variabel lain yang mempengaruhi nilai *F₁ score* dari metode identifikasi *hostname* dengan protokol HTTPS; serta
- menganalisis korelasi antara variabel bebas dengan variabel terikat pada penelitian ini secara multivariat.

Studi lain yang dapat dilakukan di atas pekerjaan tugas akhir ini adalah:

- mengkaji bagaimana mengidentifikasi akses dengan protokol HTTPS di luar *port* standar 443 serta bagaimana mengidentifikasi akses dengan protokol selain HTTPS di *port* standar 443;
- meneliti apakah memungkinkan dilakukan penggabungan metode yang ada (misalnya, metode SNI pada *handshake TLS* yang dikombinasikan dengan metode *reverse lookup* berdasarkan rekaman *query* pengguna) serta bagaimana implementasinya dalam jaringan;
- meneliti dampak secara performa akses web (terutama *latency*) ketika *transparent web proxy* perlu memeriksa semua permintaan akses web dengan berkonsultasi ke *accounting and reporting system*;

- meneliti bagaimana performa *web proxy* terkait dengan *concurrent user* atau *concurrent connection* ketika menggunakan *transparent web proxy* dibandingkan dengan menggunakan *explicit web proxy*;
- mengembangkan mekanisme yang memungkinkan PDP/server RADIUS dengan *web proxy* berjalan pada *node* yang terpisah (misalnya, dengan pertukaran informasi menggunakan *in-memory database* seperti Redis);
- mengkaji *backend* yang digunakan oleh server RADIUS dari segi fungsional dan performa – misalnya, integrasi terhadap data warga kampus (mahasiswa, dosen, dan tenaga kependidikan), atau mengkaji bagaimana komunikasi yang efisien antara server RADIUS dengan *backend* yang digunakan sebagai sumber informasi sehingga meminimalkan waktu pengguna untuk melakukan autentikasi ke jaringan intranet;
- mengembangkan implementasi teknis untuk menerapkan metode-metode identifikasi *hostname* selain SNI pada *handshake TLS* pada *web proxy* Squid; serta
- mengkaji pemantauan akses internet secara *real-time* untuk memberikan edukasi bagi pengguna jaringan dalam menggunakan *resource* jaringan yang terbatas, sehingga tidak perlu dilakukan secara manual dengan mengamati keluaran dari *log processor* seperti LightSquid yang rentan luput dikerjakan.

DAFTAR PUSTAKA

- Agarwal, T. dan Leonetti, M. (2013) “Design and Implementation of an IP based authentication mechanism for Open Source Proxy Servers in Interception Mode,” *Advanced Computing: An International Journal (ACIJ)*, 4(1), hal. 23–33. doi: 10.5121/acij.2013.4103.
- Barr, D. (1996) *Common DNS Operational and Configuration Errors*. Tersedia pada: <https://tools.ietf.org/html/rfc1912>.
- Bermudez, I. N. *et al.* (2012) “DNS to the Rescue: Discerning Content and Services in a Tangled Web,” *Proceedings of the 2012 ACM SIGCOMM Internet Measurement Conference (IMC '12)*, hal. 413–426. doi: 10.1145/2398776.2398819.
- Blake-Wilson, S. *et al.* (2003) *Transport Layer Security (TLS) Extensions*. Tersedia pada: <https://tools.ietf.org/html/rfc3546>.
- Bulman, G. dan Fairlie, R. W. (2016) “Technology and Education: Computers, Software, and the Internet,” *Handbook of the Economics of Education*, 5(9432), hal. 239–280. doi: 10.1016/B978-0-444-63459-7.00005-1.
- Callahan, T., Allman, M. dan Rabinovich, M. (2013) “On modern DNS behavior and properties,” *ACM SIGCOMM Computer Communication Review*, 43(3), hal. 7–15. doi: 10.1145/2500098.2500100.
- Campos, J. (2015) *Citrix NetScaler: How to Apply Multiple Certificates to One Virtual Server*. Tersedia pada: <http://bit.ly/2IlQyl4> (Diakses: 1 Maret 2018).
- Convery, S. (2007a) “Network Authentication, Authorization, and Accounting, Part One: Concepts, Elements, and Approaches,” *The Internet Protocol Journal*, 10(1), hal. 2–11.
- Convery, S. (2007b) “Network Authentication, Authorization, and Accounting, Part Two: Protocols, Applications, and the Future of AAA,” *The Internet Protocol*

Journal, 10(2), hal. 2–15.

Felt, A. P. *et al.* (2017) “Measuring HTTPS Adoption on the Web,” in *USENIX Security*.

Feng, J. (2009) “Analysis, implementation and extensions of RADIUS protocol,” in *2009 International Conference on Networking and Digital Society*, hal. 154–157. doi: 10.1109/ICNDS.2009.44.

Fielding, R. T. *et al.* (1999) *Hypertext Transfer Protocol – HTTP/1.1*. Tersedia pada: <https://tools.ietf.org/html/rfc2616> (Diakses: 5 Maret 2018).

Foremski, P., Callegari, C. dan Pagano, M. (2014) “DNS-Class: immediate classification of IP flows using DNS,” *International Journal of Network Management*, 24, hal. 272–288. doi: 10.1002/nem.1864.

Hole, K. J., Dyrnes, E. dan Thorsheim, P. (2005) “Securing Wi-Fi networks,” *Computer*, 38(7), hal. 28–34. doi: 10.1109/MC.2005.241.

Li, Q. dan Clark, G. (2015) *Security Intelligence: A Practitioner’s Guide to Solving Enterprise Security Challenges*. Indianapolis: John Wiley & Sons, Inc.

Luotonen, A. dan Altis, K. (1994) “World Wide Web proxies,” *Computer Networks and ISDN Systems*, 27(2), hal. 147–154.

McAfee (2014) *Direct or Transparent Proxy?* Tersedia pada: <http://bit.ly/2JegIr5>.

Nygren, E. (2017) *Reaching toward universal TLS SNI, The Akamai Blog*. Tersedia pada: <http://bit.ly/2Gsk8Jk> (Diakses: 20 Maret 2018).

Oppenheimer, P. (2011) *Top-Down Network Design*. Third Edit. Indianapolis: Cisco Press.

Rabinovich, M. dan Spatscheck, O. (2001) *Web Caching and Replication*. Addison-Wesley Professional.

Rao, A. (2013) *Improving Transparency and End-User Control in Mobile*

Networks. Université Nice Sophia Antipolis.

Raymond, E. S. (2003) *The Art of Unix Programming*. Addison-Wesley Professional.

Rescorla, E. (2000) *HTTP Over TLS*. Tersedia pada: <https://tools.ietf.org/html/rfc2818> (Diakses: 5 Maret 2018).

Rigney, C. (2000) *RADIUS Accounting*. Tersedia pada: <https://tools.ietf.org/html/rfc2866>.

Shbair, W. M. (2017) *Service-Level Monitoring of HTTPS Traffic*. Université de Lorraine.

Slameta (2013) “Pengembangan Infrastruktur Jaringan Komputer (Studi Kasus: Politeknik Negeri Bandung),” *TEDC*, 6(1).

Suprapto, J. (2000) *Statistik: Teori & Aplikasi, Jilid 1*. 6 ed. Jakarta: Erlangga.

Tanenbaum, A. S. dan Wetherall, D. J. (2011) *Computer Networks*. 5 ed. Boston: Prentice Hall.

Ting, K. M. (2017) “Confusion Matrix,” in Sammut, C. dan Webb, G. I. (ed.) *Encyclopedia of Machine Learning and Data Mining*. 2nd ed. Boston: Springer, hal. 260.

Villarroel-Acosta, A. A. et al. (2017) “Method of Auto-configuration for Corporate Proxies,” *Ingeniería Solidaria*, 13(21), hal. 9–18.

Walls, C. (2006) *Embedded Software: The Works*. Elsevier.

Wilson, A. (2017) *The future is not to proxy, but what about security?*, *IT News Africa*. Tersedia pada: <http://bit.ly/2IVZnXx> (Diakses: 11 Mei 2018).

Yeh, J. (2017) *Key factors in building a Secure Web Gateway*. The University of Waikato.

LAMPIRAN 1

LAPORAN ANALISIS KEBUTUHAN LAYANAN IT

JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA



**LAPORAN ANALISIS KEBUTUHAN LAYANAN IT
DI JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA
TAHUN 2018**

JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA
POLITEKNIK NEGERI BANDUNG
2018

LEMBAR PENGESAHAN

Dokumen ini dibuat untuk memberikan informasi mengenai kebutuhan layanan IT di Jurusan Teknik Komputer dan Informatika (JTK) untuk sudut pandang pengelola IT di Politeknik Negeri Bandung.

Dibuat di Bandung Barat, 8 Januari 2018

Kepala Laboratorium Jaringan & Server,



Ghifari Munawar, S.Kom., M.T.

NIP. 19860412 201404 1 001

Disahkan di Bandung Barat, 8 Januari 2018

a.n. Ketua Jurusan Teknik Komputer dan Informatika,

Sekretaris I,



Ani Rahmani, S.Si., M.T.

NIP. 19681014 199303 2 002

DAFTAR ISI

Lembar Pengesahan	i
Daftar Isi.....	ii
Bab I Latar Belakang	1
1.1. Kebutuhan Layanan IT di JTK.....	1
1.2. Permasalahan yang Dihadapi	6
1.3. Infrastruktur Jaringan JTK (<i>Existing</i>).....	10
Bab II Analisis Kebutuhan Layanan	12
2.1. Kebutuhan <i>Incoming Network Access</i> ke JTK	12
2.2. Kebutuhan <i>Outgoing Network Access</i> dari JTK.....	13
2.3. Kebutuhan Layanan Pendukung di Luar JTK	14
2.4. Kebutuhan Layanan Pendukung di Dalam JTK	15
Bab III Kesimpulan	16

BAB I

LATAR BELAKANG

1.1. Kebutuhan Layanan IT di JTK

Kebutuhan akan layanan IT **yang optimal** pada Jurusan Teknik Komputer dan Informatika (JTK) merupakan salah satu hal yang mutlak dipenuhi. Sebagai salah satu jurusan dengan capaian lulusan yang berkecimpung dalam dunia teknologi informasi, layanan IT merupakan salah satu prasyarat yang mutlak dipenuhi dalam tercapainya proses pembelajaran yang baik. JTK memiliki dua program studi, yakni D-III Teknik Informatika dan D-IV Teknik Informatika dengan jumlah mahasiswa keseluruhan mencapai ±300 mahasiswa ditambah 30 dosen dan 4 staf administrasi.

Salah satu layanan IT yang besar perannya adalah **akses internet**. Di JTK, akses internet tidak hanya digunakan dalam akses informasi saja seperti *browsing*, *download artikel*, *download software*, kirim *e-mail*, *upload tugas*, dll., namun juga digunakan **saat pembelajaran berlangsung** seperti mengadakan kuis *online*, pengembangan produk (*coding/development*), publikasi produk (*deployment*), manajemen *source code*, *version control* (GitLab, GitHub, atau SVN), pameran/demo produk, dll.

Besarnya peran akses internet ini ditunjukkan dengan perincian mata kuliah yang terpengaruh dalam PBM setiap semesternya jika terjadi kendala terhadap akses layanan internet di JTK **berdasarkan kurikulum tahun 2016** sebagai berikut:

- **D-III Teknik Informatika**

No.	Kode MK	Nama MK	SKS	Semester
1	16TKO1054	Dasar-dasar Pemrograman	4	1
2	16TKO1083	Proyek Perangkat Lunak 1	3	1
3	16TKO2044	Struktur Data & Algoritma	4	2
4	16TKO2073	Proyek Perangkat Lunak 2	3	2
5	16TKO3014	Basis Data	4	3
6	16TKO3024	Pengantar Rekayasa Perangkat Lunak	4	3
7	16TKO3043	Pemrograman Berorientasi Object	3	3
8	16TKO3073	Proyek Perangkat Lunak 3	3	3
9	16TKO4014	Analisa dan Peranc. Sistem Informasi	4	4
10	16TKO4024	Pengembangan Perangkat Lunak I	4	4
11	16TKO4043	Pemrog. Perangkat Bergerak (Mobile)	3	4
12	16TKO4063	Proyek Perangkat Lunak 4	3	4
13	16TKO5024	Pengembangan Perangkat Lunak II	4	5

14	16TKO5043	Komputer Grafik	3	5
15	16TKO5073	Proyek Perangkat Lunak 5	3	5
16	16TKO6022	Pengolahan Citra Digital	2	6
17	16TKO6043	Manajemen Proyek	3	6
18	16TKO6054	Tugas Akhir	4	6
TOTAL SKS			61	

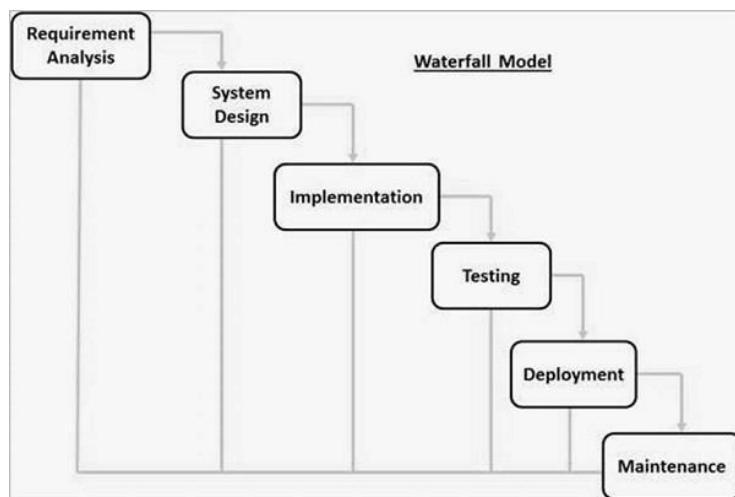
Terdapat 18 mata kuliah dari total 32 **mata kuliah program studi** di program studi D3 yang membutuhkan akses internet yang intens dalam mendukung PBM, sehingga ada ±56,25% dari jumlah mata kuliah yang secara tidak langsung akan berdampak pada target capaian PBM jika tidak difasilitasi dengan baik. Terlebih lagi mata kuliah tersebut adalah mata kuliah inti yang memuat kompetensi utama dari lulusan program studi D3.

- **D-IV Teknik Informatika**

No.	Kode MK	Nama MK	SKS	Semester
1	16TIN1034	Dasar-dasar Pemrograman	4	1
2	16TIN1064	Konsep Teknologi Informasi dan Komunikasi	4	1
3	16TIN2043	Proyek Perangkat Lunak 1	3	2
4	16TIN2054	Teknik Pemrograman	4	2
5	16TIN2074	Struktur Data dan Algoritma	4	2
6	16TIN3053	Sistem Basis Data	3	3
7	16TIN3063	Prinsip Bahasa Pemrograman	3	3
8	16TIN3073	Komunikasi Data dan Jaringan	3	3
9	16TIN4014	Analisis dan Perancangan Perangkat Lunak 1	4	4
10	16TIN4063	Proyek Perangkat Lunak 2	3	4
11	16TIN4043	Perancangan Antarmuka	3	4
12	16TIN5024	Sistem Informasi	4	5
13	16TIN5054	Pengembangan Web	4	5
14	16TIN5063	Manajemen Proyek Rekayasa Perangkat Lunak	3	5
15	16TIN6013	Analisis dan Perancangan Perangkat Lunak 2	3	6
16	16TIN6023	Sistem Terdistribusi	3	6
17	16TIN6043	Pengolahan Citra Digital	3	6
18	16TIN6053	Pengujian Perangkat Lunak	3	6
19	16TIN6064	Pemrograman Perangkat Lunak Berorientasi Objek	4	6
20	16TIN6073	Proyek Perangkat Lunak 3	3	6
21	16TIN7034	Proyek Perangkat Lunak 4	2	7
22	16TIN8042	Kualitas Perangkat Lunak	2	8
23	16TIN8054	Proyek Perangkat Lunak 5 (Tugas Akhir)	4	8
TOTAL SKS			76	

Terdapat 23 mata kuliah dari total 40 **mata kuliah program studi** di program studi D4 yang membutuhkan akses internet yang intens dalam mendukung PBM, sehingga ada ±57,50% dari jumlah mata kuliah yang secara tidak langsung akan berdampak pada target capaian PBM jika tidak difasilitasi dengan baik. Terlebih lagi mata kuliah tersebut adalah mata kuliah inti yang memuat kompetensi utama dari lulusan program studi D4.

Salah satu mata kuliah yang paling intens dalam hal ini adalah mata kuliah **Proyek Perangkat Lunak** (total 5 mata kuliah pada D3, dan 4 mata kuliah pada D4). Mata kuliah proyek perangkat lunak merupakan suatu mata kuliah yang mengimplementasikan *lifecycle* (daur hidup) dari pengembangan perangkat lunak, mulai dari tahap analisis, tahap desain, tahap implementasi/*coding*, tahap pengujian, hingga tahap publikasi (*deployment*).



Gambar 1. Daur hidup perangkat lunak

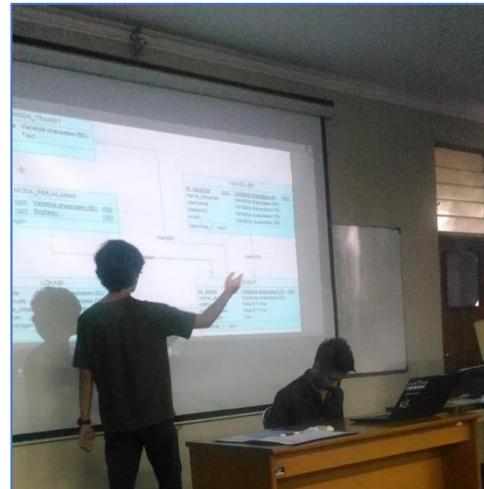
Mata kuliah ini menjadi mata kuliah unggulan di JTK, di mana mahasiswa dituntut untuk melakukan setiap tahapan pengembangan perangkat lunak dari tahap analisis hingga tahap publikasi (*deployment*). Setiap tahapannya membutuhkan akses internet yang intens, seperti akses ke *project management online*, *source code management*, *version control*, API, dll. Suasana perkuliahan mata kuliah ini ditunjukkan pada Gambar 2.



- a. Mahasiswa melakukan PBM secara berkelompok dan didampingi oleh dosen mata kuliah proyek.



- b. Mahasiswa mempresentasikan hasil produk berupa web *online*



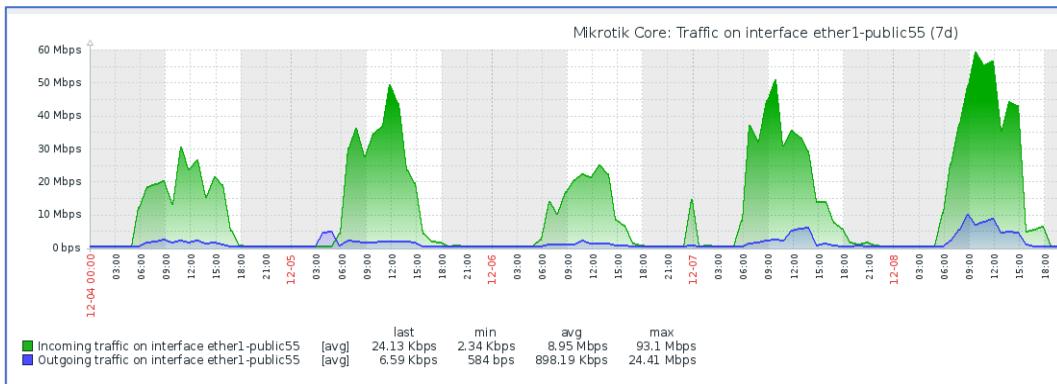
- c. Mahasiswa mempresentasikan proses analisis produk

Gambar 2. Suasana PBM mata kuliah Proyek Perangkat Lunak

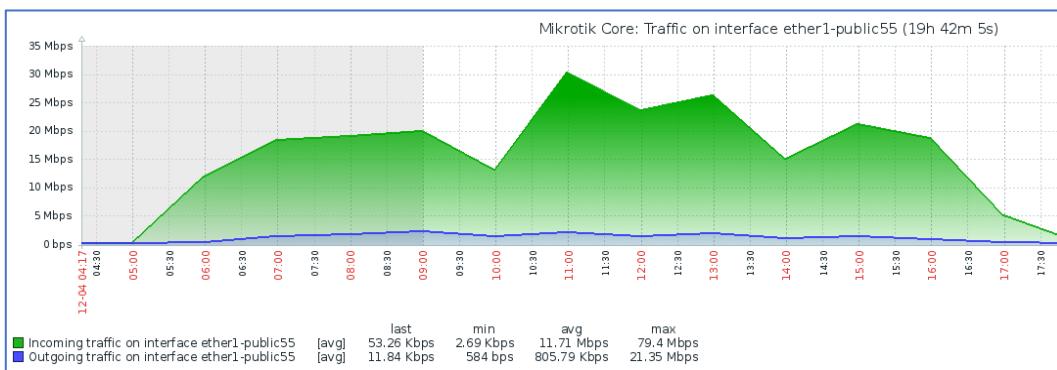
Selain mata kuliah proyek yang sepenuhnya menggunakan akses internet dalam PBM, di samping itu beberapa mata kuliah lain pun cukup intens menggunakan akses internet saat jam perkuliahan berlangsung. Hal ini dapat dilihat pada grafik statistik penggunaan internet di JTK saat jam perkuliahan. Gambar 3 menunjukkan salah satu contoh gambaran penggunaan akses internet di minggu perkuliahan mulai pukul 06.00 s.d. 18.00 WIB pada tanggal 4 s.d. 9 Desember 2017.

Statistik tersebut merupakan lalu lintas data yang tercatat pada *router* di jaringan JTK. Berdasarkan gambar tersebut dapat dilihat bahwa di jam perkuliahan antara pukul 07.00 s.d. 16.00 WIB adalah jam di mana akses internet digunakan secara intens. Hal ini menjadi salah satu gambaran bahwa kebutuhan akan akses layanan internet dalam mendukung PBM di JTK sangat tinggi.

Gambar 4 menunjukkan statistik apabila dirinci dalam satu hari (sebagai contoh tanggal 4 Desember 2017).



Gambar 3. Statistik *traffic* internet dalam satu pekan perkuliahan



Gambar 4. Statistik *traffic* internet dalam satu hari perkuliahan

Selain untuk memenuhi kebutuhan akses secara langsung oleh peserta PBM, akses internet juga diperlukan untuk **operasional perangkat server dan jaringan**, seperti pengiriman *email* dari Mail Transfer Agent (MTA), akses *controller* perangkat jaringan, dll.

Akses internet tidak hanya digunakan oleh para *stakeholders* untuk berkegiatan di dalam JTK, namun juga **memfasilitasi berbagai kebutuhan dari luar JTK**. Di antaranya akses ke layanan-layanan internal seperti portal *e-learning*, server penelitian, maupun server pembelajaran beberapa mata kuliah seperti Sistem Terdistribusi atau Komunikasi Data dan Jaringan yang membutuhkan akses SSH. Akses ke layanan-layanan tersebut akan dikelola melalui sebuah server VPN di dalam JTK yang menggunakan IP publik JTK yang sedang dalam proses pengembangan.

Tidak hanya akses internet yang memenuhi kebutuhan, **penggunaan e-mail institusi** (dengan domain JTK bagi dosen dan domain Polban bagi mahasiswa) juga

merupakan kebutuhan penting dalam kehidupan akademik. Bagi para dosen yang memiliki tugas meneliti sebagai salah satu pemenuhan Tridarma Perguruan Tinggi, **availability** dari layanan *e-mail* sangat diandalkan sebagai media komunikasi dengan sesama peneliti dan dicantumkan dalam publikasi ilmiah. *E-mail* juga digunakan oleh dosen untuk menerima pengumpulan tugas serta berkomunikasi dengan mahasiswa.

Tak lupa, sebagai jurusan dengan lima laboratorium fisik dengan unit komputer yang disediakan untuk pelaksanaan PBM, JTK memerlukan **lisensi** sistem operasi serta aplikasi-aplikasi terkait yang menunjang pelaksanaan PBM. Dalam hal ini, penggunaan lisensi yang resmi sesuai *resource* yang dimiliki Polban sangat ditekankan karena menyangkut aspek legalitas.

1.2. Permasalahan yang Dihadapi

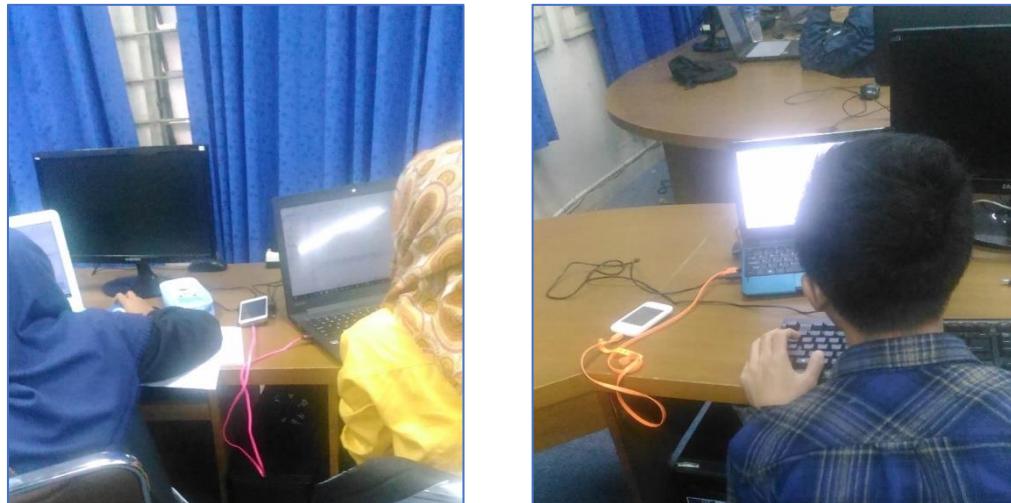
Dari uraian di atas dapat dilihat bahwa kebutuhan akan layanan akses internet yang optimal menjadi kebutuhan utama bagi JTK untuk mendukung PBM. Hal ini perlu diwujudkan demi tercapainya tujuan pembelajaran yang sesuai dengan profil lulusan dari masing-masing program studi D3 dan D4 Teknik Informatika JTK.

Sayangnya, kebutuhan-kebutuhan tersebut hingga saat ini belum difasilitasi dengan baik, dengan timbulnya permasalahan-permasalahan berikut:

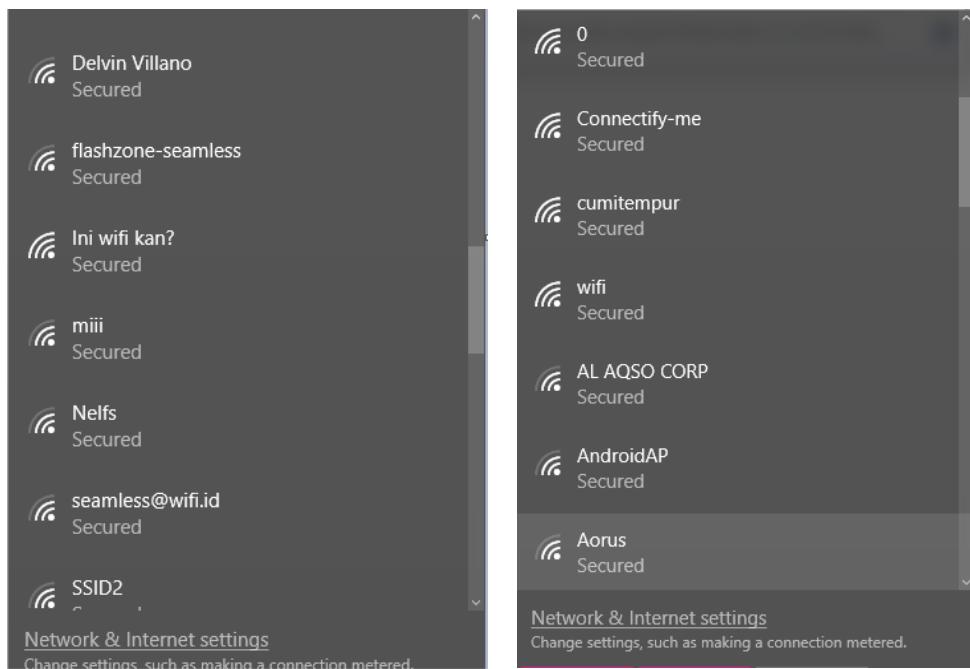
- **Pembatasan akses internet menggunakan proxy** mengakibatkan kurang optimalnya kegiatan PBM pada beberapa mata kuliah, terutama yang terkait dengan pengembangan produk, *version control*, manajemen *source code*, publikasi produk, serta pameran/demo produk. Kebutuhan akses internet seperti mengakses API (Application Programming Interface), sinkronisasi *source code* (seperti Git, SVN, dll.), akses *software development kit* (SDK) untuk pemrograman *mobile* seperti Android/Java, akses *cloud database/web* akan **menjadi terhambat** akibat tidak kompatibelnya kebutuhan-kebutuhan tersebut dengan *proxy*.

Sementara ini, para *stakeholder* (mahasiswa, dan dosen) dibebankan / dipaksa untuk memiliki akses jaringan sendiri, sehingga akses internet yang dibatasi proxy harus dikoneksikan melalui jalur lain (di luar jaringan JTK seperti *hotspot* melalui *handphone* atau modem WiFi pribadi – lihat Gambar

5 dan Gambar 6). Hal ini menjadi tidak optimal apabila tidak ada solusi ke depannya, di mana pembelajaran menjadi terhambat dan menambah beban bagi para *stakeholder* (mahasiswa dan dosen).



Gambar 5. Penggunaan akses internet pribadi pada PBM Proyek Perangkat Lunak



Gambar 6. WiFi pribadi pada PBM Proyek Perangkat Lunak

- **Tidak stabilnya proxy yang disediakan Polban untuk JTK** mengakibatkan layanan internet tidak dapat diakses ketika jam kerja dan padat penggunaan. Sistem *monitoring* dari dalam JTK menunjukkan **dalam tiga bulan terakhir, terjadi 89 kejadian proxy Polban yang tidak dapat diakses** (diilustrasikan pada Gambar 7). Selain tidak dapat diakses, **terjadi**

pula degradasi kualitas layanan (berupa *reply with zero bytes*) dengan intensitas serupa dalam jangka waktu yang sama. Hal ini sangat disayangkan, mengingat koneksi Polban ke internet tidak bermasalah, tetapi karena *proxy* yang tidak stabil mengakibatkan kegiatan *stakeholders* terganggu.

Problem	Duration
Proxy Polban 1 is unavailable by ICMP	1d 5h 29m
Proxy Polban 1 is unavailable by ICMP	1d 13h 33m
Proxy Polban 1 is unavailable by ICMP	7m
Proxy Polban 1 is unavailable by ICMP	1m
Proxy Polban 1 is unavailable by ICMP	12m
Proxy Polban 1 is unavailable by ICMP	1m
Proxy Polban 1 is unavailable by ICMP	8m
Proxy Polban 1 is unavailable by ICMP	9m

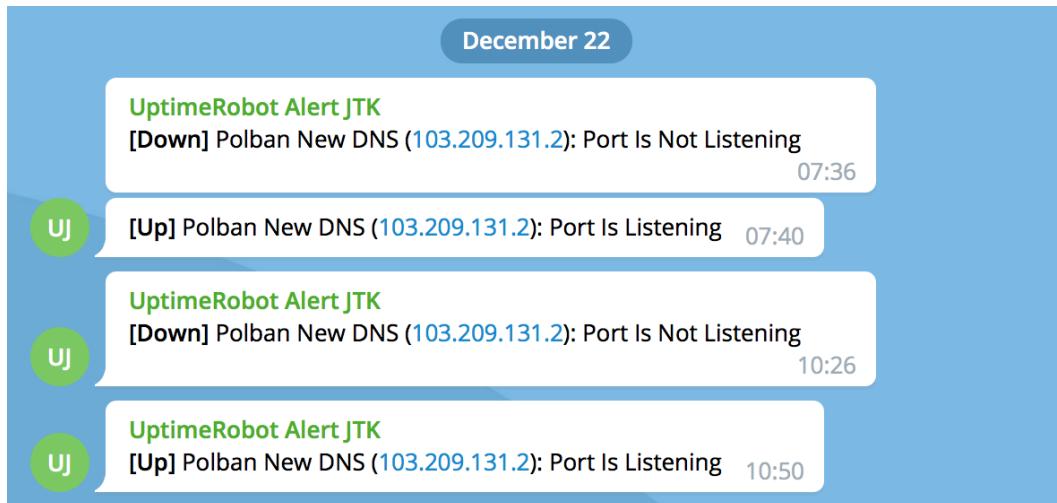
Gambar 7. *Proxy* Polban tidak dapat diakses berdasarkan sistem *monitoring* JTK

Dari tidak stabilnya *proxy* ini, selain mengakibatkan penggunaan akses pribadi, juga diketahui beberapa pengguna memilih menggunakan akses VPN yang banyak beredar di internet. Penggunaan akses VPN ini **membuat koneksi tidak lagi dapat dikendalikan** menggunakan *proxy* sebagaimana yang diharapkan Polban.

- Layanan *e-mail* tidak *reliable* akibat **tidak stabilnya DNS Polban** yang dikelola secara *in-house*. Penyediaan layanan *e-mail* yang menggunakan layanan Google dengan *reliability* yang tinggi sayangnya tidak diimbangi dengan layanan DNS yang dengan tingkat *reliability* serupa. Peran DNS sangat vital, dalam hal ini sebagai *resolver* untuk mengarahkan *e-mail* ke layanan Google.

Pengelolaan DNS secara *in-house* yang saat ini dilakukan – alih-alih menggunakan layanan yang sudah teruji – membutuhkan *effort* besar untuk menghadirkan tingkat *reliability* yang sebanding. *Monitoring* yang dilakukan JTK dari luar menunjukkan tidak stabilnya layanan DNS (lihat

Gambar 8), dan sering menjadi penyebab terganggunya pelaksanaan PBM; salah satunya gagalnya pelaksanaan ujian *online* pada PBM tertentu yang membutuhkan akses *e-mail*.



Gambar 8. *Reporting* layanan DNS Polban yang *unavailable*

- Pemberian lisensi yang **membutuhkan waktu yang tidak sebentar** dari Polban menyebabkan kebutuhan *maintenance* laboratorium JTK menjadi terhambat. *By nature*, laboratorium JTK yang digunakan oleh publik (termasuk ruang *e-library* yang disediakan oleh Polban di JTK) membuat adanya keperluan untuk melakukan *refresh* secara berkala. Selain itu, akses lisensi berupa *volume license* (yang merupakan lisensi yang tepat untuk penggunaan laboratorium) dari Polban juga **terbatas** sehingga penyediaan layanan ke *stakeholders* terhambat.

Permasalahan yang terjadi pada layanan IT di JTK secara tidak langsung dapat mempengaruhi target capaian PBM. Banyak target pembelajaran yang sudah direncanakan, beberapa terhambat karena terkendala layanan IT belum menyediakan performa layanan yang diharapkan.

Dengan kata lain, dapat dikatakan bahwa kesuksesan target PBM-nya tergantung dari siapnya layanan IT. Terhambatnya akses internet, tidak stabilnya *proxy* Polban dan layanan DNS Polban sepatutnya dapat diminimalkan. Penyelesaian masalah-masalah tersebut merupakan kebutuhan yang tidak dapat dihindari bagi optimalisasi pembelajaran di JTK.

1.3. Infrastruktur Jaringan JTK (*Existing*)

Secara *logic*, Polban mengalamatkan jaringan JTK pada *subnet* 192.168.72.0/24. Karena kebutuhan JTK yang besar dan kompleks dan tidak mampu ditampung jika harus ditempatkan pada *subnet* tersebut, maka JTK membuat jaringan *logic* sendiri dengan menempatkan *router* pada alamat 192.168.72.5. *Router* yang dipasang ini kemudian menerapkan Network Address Translation (NAT), sehingga dari sisi Polban semua *traffic* dari JTK akan keluar dari alamat tersebut.

Secara umum, jaringan JTK kemudian dibagi-bagi ke dalam jaringan berikut:

- Jaringan server (untuk operasional dan eksperimen, masing-masing dalam jaringan terpisah, berupa kumpulan server fisik maupun virtual serta *NAS/network attached storage*)
- Jaringan manajemen perangkat jaringan (UniFi)
- Jaringan multimedia (untuk *streaming* kegiatan jurusan dari ruang serbaguna ke Internet)
- *Customer access*
 - Jaringan untuk dosen
 - Jaringan untuk tenaga kepegawaian
 - Jaringan untuk mahasiswa
 - Jaringan untuk ruangan *e-library*
 - Jaringan untuk setiap laboratorium

Selain itu, untuk kebutuhan JTK yang perlu berhadapan langsung dengan Internet, JTK diberikan IP publik oleh Polban dengan alamat 103.209.131.55. IP publik ini di-*assign* di *router* yang sama dengan *router* yang beralamat di 192.168.72.5 di atas.

IP publik JTK ini tidak dapat digunakan untuk keperluan akses melalui *port* 80 (HTTP) dan 443 (HTTPS). Untuk keperluan tersebut, Polban menyediakan server *proxy* di alamat 192.168.72.1 dengan satu pasang *username-password* yang dikhususkan untuk JTK.

Untuk kegiatan sehari-hari warga JTK, dipasang server *proxy* internal JTK. *Proxy server* ini akan meneruskan semua *traffic*-nya ke server *proxy* Polban di

192.168.72.1 menggunakan pasangan *username-password* yang diberikan Polban untuk JTK.

Selain sebagai penerus ke server *proxy* Polban, server *proxy* internal JTK juga melakukan *caching* terhadap *repository* berbasis Ubuntu/Debian secara *on-demand*. Dengan demikian, proses *software update* semua mesin di JTK yang berbasis Ubuntu/Debian dapat dilakukan tanpa memakan *bandwidth* yang besar.

Dalam menyediakan akses ke warga JTK, JTK mengelola *access point* sendiri dengan SSID berikut di gedung JTK:

- AP Dosen JTK Lt. 1
- AP Dosen JTK Lt. 2
- AP Mahasiswa JTK Lt. 1
- AP Mahasiswa JTK Lt. 2

Access point yang digunakan adalah *access point* Ubiquiti UniFi AP AC PRO, dengan konfigurasi bahwa setiap SSID akan *broadcast* di frekuensi 2,4 GHz dan 5 GHz.

Untuk kebutuhan manajemen semua perangkat JTK dari luar jaringan JTK, digunakan *port forwarding*. Untuk setiap mesin yang butuh diakses dari luar, akan dibuka satu *port* di IP publik JTK yang akan diteruskan ke mesin terkait. Sebagai contoh, untuk mengakses layanan SSH (*port* 22) dari server Moodle JTK (dengan alamat internal 10.10.100.15), pengguna layanan dapat mengakses *port* 2215 dari IP publik JTK.

BAB II

ANALISIS KEBUTUHAN LAYANAN

2.1. Kebutuhan *Incoming Network Access* ke JTK

Kebutuhan *incoming network access* ke JTK melalui IP publik JTK secara umum adalah *port forwarding* dengan aplikasi-aplikasi berikut:

- **Web access** berupa *traffic* ke *port* 80 dan 443, digunakan untuk mengakses sistem-sistem di lingkungan JTK (seperti *website* JTK, *website* Moodle JTK, *website* himpunan, portal *knowledge base* JTK, sistem monitoring JTK, dan sistem lainnya). Karena peranannya vital (*website* JTK terutama berfungsi sebagai sumber informasi bagi calon pendaftar PMDK/SMB), maka *availability web access* ke JTK harus sama dengan *availability web access* ke situs Polban lainnya, terutama *website* PMDK dan SMB;
- **VPN access**, berupa *traffic* ke *port* dari VPN server JTK yang akan ditentukan selanjutnya (menunggu pengembangan RADIUS *authentication* di dalam JTK selesai), digunakan untuk memberikan akses bagi dosen dan pengelola jaringan. *Use case* dari VPN access ini adalah:
 - Dosen peneliti dan tim untuk mengakses server penelitian di JTK;
 - Pengelola jaringan untuk mengakses perangkat JTK dari luar;
 - Dosen pengajar dan mahasiswa untuk mengakses server pembelajaran di luar aplikasi berbasis *web access* (misalnya SSH untuk mata kuliah Sistem Operasi, Komunikasi Data dan Jaringan, maupun Sistem Terdistribusi);
- **SSH access**, berupa *traffic* ke *port-port* yang jumlahnya dinamis (sesuai dengan kebutuhan di semester berjalan), digunakan untuk memenuhi kebutuhan dari *use case* VPN access di atas selama VPN server JTK belum siap;
- Aplikasi lainnya yang digunakan secara **insidental**, misalnya *iperf* yang digunakan untuk menguji kapasitas *bandwidth* JTK dari luar. Untuk keperluan ini dibutuhkan *port forwarding* ke *port* yang dinamis dalam waktu cepat dan singkat.

2.2. Kebutuhan *Outgoing Network Access* dari JTK

Kebutuhan *outgoing network access* dari dalam JTK dapat dibagi ke dalam **tiga kategori besar** yang membutuhkan dua alamat IP publik terpisah, yaitu:

- Kebutuhan untuk me-*reply incoming network access* yang diuraikan pada subbab 2.1 di atas. Untuk keseragaman, seluruh *reply* dari *incoming network access* ke JTK harus dibalas melalui IP publik dan jalur yang sama (dalam hal ini, 103.209.131.55).
- Kebutuhan *traffic* ke luar yang merepresentasikan **sistem di dalam JTK** secara *official* (bukan pemakaian secara langsung oleh *customer*), yaitu:
 - *Outgoing mail* dari Mail Transfer Agent (MTA) di *server* JTK. *Outgoing mail* ini merupakan *e-mail* dari sistem, seperti *forget password* dari aplikasi operasional di JTK, *e-mail* yang dikirimkan dari sistem informasi yang dibuat mahasiswa di mata kuliah, atau *e-mail* dari *network monitoring system* internal JTK. Sesuai protokol baku, *outgoing mail* melewati *port* 25;
 - Sinkronisasi UniFi *controller* di JTK dengan UniFi Cloud melalui *port* 443 (HTTPS dan WebSocket, serta tidak bisa melalui *proxy*) untuk mengelola perangkat jaringan yang *managed* di JTK;
 - Pemanggilan API-API eksternal seperti Google API dan Telegram API untuk sistem informasi yang dibuat oleh mahasiswa, terutama di mata kuliah proyek. *Port* yang dituju adalah *port* 80 dan *port* 443 dan tidak bisa melalui *proxy*;
 - Kebutuhan aplikasi lain di masa depan sesuai perkembangan.

Seluruh kebutuhan di atas harus keluar melalui IP publik JTK di 103.209.131.55, terutama *outgoing mail* karena hanya IP publik JTK tersebut yang diotorisasi di DNS JTK untuk mengirimkan *e-mail*. Selain itu, seluruh kebutuhan di atas hanya keluar dari jaringan *server*, bukan jaringan *customer*.

- Kebutuhan lainnya yang tidak terkait dengan *incoming network access* dan **langsung digunakan oleh customer.**

Kebutuhan *customer* ini **harus melalui IP publik lain** (boleh dibagi dengan jurusan lain) karena risiko aktivitasnya yang dapat mengganggu reputasi IP publik JTK untuk keperluan yang bersifat *official* seperti *outgoing mail*.

Persyaratan bahwa *traffic* harus melalui IP publik lain sebenarnya sudah dipenuhi dengan penggunaan *proxy* Polban, tetapi ada beberapa aplikasi yang **tidak dapat berjalan melalui proxy** dan dibutuhkan, yaitu:

- *Dependency management* dari Android Studio dan Visual Studio, dibutuhkan untuk proses *building* semua aplikasi berbasis Android dan .NET;
- Pemanggilan API dari sistem informasi yang dibangun mahasiswa di mata kuliah yang sifatnya *development* (belum di-host di server JTK/masih di komputer lab atau *laptop* mahasiswa);
- Sinkronisasi *source code* dengan layanan *repository online* seperti GitLab dan GitHub menggunakan protokol HTTPS.

Selain itu, seperti diuraikan pada Bab I, akses ke *proxy* Polban saat ini juga tidak stabil, sehingga layanan internet bisa tidak dapat berfungsi, tetapi layanan-layanan lain di luar *port* 80 dan 443 dapat berfungsi.

Tiga kebutuhan di atas yang membutuhkan dua IP publik dapat dilakukan dengan pengaturan di sisi *router* menggunakan *multi gateway* dan *traffic marking*.

2.3. Kebutuhan Layanan Pendukung di Luar JTK

Kebutuhan layanan pendukung di luar JTK adalah:

- ***E-mail* yang reliable.** Layanan *e-mail* dibutuhkan sebagai sarana korespondensi akademik antar dosen (baik di dalam Polban maupun dengan luar Polban) serta sarana komunikasi antara mahasiswa dengan dosen. Layanan ini membutuhkan *reliability* yang tinggi, salah satunya diukur dengan *availability* layanan yang tinggi.
- **Layanan DNS yang reliable.** Layanan DNS dibutuhkan sebagai *name resolver* dari berbagai layanan di internet ke layanan yang dimiliki Polban. Layanan DNS harus memiliki tingkat *reliability* yang tinggi, sesuai dengan

layanan dengan *reliability* paling tinggi yang dimiliki Polban (dalam hal ini *e-mail* yang disediakan Google). Tidak tersedianya layanan DNS menyebabkan layanan lain terganggu aksesnya meski layanan lain tersebut tidak bermasalah sama sekali.

Selain itu, layanan DNS yang *reliable* dibutuhkan untuk memberikan opsi bagi JTK agar dapat memiliki layanan yang di-*host* di luar lingkungan JTK (seperti *cloud service* atau layanan monitor).

2.4. Kebutuhan Layanan Pendukung di Dalam JTK

Kebutuhan layanan pendukung di dalam JTK adalah *software* berikut yang digunakan di laboratorium:

- Windows 10 Pro
- Microsoft Office 2016
- Microsoft Project 2016
- Microsoft Visio 2016

Lisensi untuk keempat jenis *software* tersebut haruslah lisensi yang diperuntukkan untuk laboratorium, bukan diikat ke perorangan (seperti Office 365 atau DreamSpark individu). Dengan demikian, penggunaan lisensi yang bersifat *volume license* menjadi hal mutlak untuk dipenuhi.

BAB III

KESIMPULAN

Pada laporan ini telah dipaparkan kebutuhan terhadap akses layanan internet di JTK. Penyusunan laporan ini dilakukan sebagai kajian analisis yang memuat permasalahan yang dihadapi selama ini oleh para *stakeholder* (mahasiswa, dan dosen) dalam mengakses layanan internet di JTK untuk mendukung PBM. Selain itu telah disampaikan pula secara teknis akan kebutuhan layanan internet dan infrastruktur yang dimiliki JTK saat ini. Kendala yang dihadapi dalam mengakses layanan internet di JTK secara tidak langsung akan berdampak pada capaian target dari PBM-nya. Sebagai salah satu jurusan dengan capaian lulusan yang berkecimpung dalam dunia teknologi informasi, akses internet merupakan salah satu prasyarat yang mutlak dipenuhi demi tercapainya proses pembelajaran yang baik di JTK. Harapan kami ke depan adalah kebutuhan terhadap akses layanan internet dapat dioptimalkan demi terwujudnya capaian pembelajaran yang baik di JTK.

Demikian laporan yang dapat kami susun, semoga dapat menjadi pertimbangan Bapak/Ibu dalam merencanakan kebijakan yang strategis bagi kemajuan JTK pada khususnya dan kemajuan institusi Polban pada umumnya sehingga visi JTK sebagai **“jurusan unggulan dan terdepan di bidang pengkajian, penerapan dan pengembangan teknologi informasi, yang diakui baik di tingkat nasional maupun internasional”** dapat diwujudkan.

LAMPIRAN 2
STANDAR PENGELOLAAN JARINGAN SI
POLITEKNIK NEGERI BANDUNG

 POLBAN	POLITEKNIK NEGERI BANDUNG	Kode/No : PL1.R1/STD/Pend/Polban/BAAK-17
	Standar Pengelolaan Jaringan SI	Tanggal: 30 Agustus 2016
		Revisi : 0
		Halaman: 1 dari 12

STANDAR PENGELOLAAN JARINGAN SI
POLITEKNIK NEGERI BANDUNG



	Jabatan	Nama	Tanda Tangan
Dikaji ulang oleh:	Pembantu Direktur Bidang Akademik		
Dikendalikan oleh:	Satuan Penjaminan Mutu		
Disetujui oleh:	Direktur		



1. Definisi Istilah

Jaringan Lokal (*Local Area Network*, LAN) adalah sekelompok komputer dengan perangkat pendukungnya yang terhubung dan dapat berkomunikasi dalam area kerja tertentu.

Jaringan Jarak Jauh (*Wide Area Network*, WAN) adalah dua atau lebih LAN yang terhubung dan dapat berkomunikasi.

Jaringan Komputer Lokal Berbasis Internet (*Intranet*) adalah suatu jaringan komputer yang menggunakan fasilitas LAN dan atau WAN untuk keperluan internal.

Jaringan Komputer Global (*Internet*) adalah kumpulan jaringan komputer yang saling terhubung dan menganut konsep terbuka, sehingga informasi yang ada didalamnya dapat diakses secara luas.

Penyedia Layanan Internet (*Internet Service Provider*, ISP) adalah suatu kegiatan usaha yang menyediakan layanan akses ke jaringan internet.

Kartu Antarmuka Jaringan (*Network Interface Card*) adalah perangkat keras pada komputer yang digunakan sebagai *interface* dari komputer ke jaringan komputer serta mengatur pengiriman dan penerimaan data dari dan ke dalam jaringan.

Perangkat Lunak Jaringan adalah sarana untuk dapat berhubungan dengan komputer lain melalui jaringan, sehingga pertukaran data dapat terjadi dengan mudah.

Penyimpan Data (*Disk Storage, Storage Devices*) adalah perangkat keras yang digunakan sebagai sarana menyimpan data dalam bentuk elektronik.

Sistem Informasi (*Electronic Office, e-Office*) adalah aplikasi perkantoran yang mengganti proses administrasi berbasis manual ke proses berbasis elektronis dengan memanfaatkan fasilitas LAN.

Kode Akses (*Password*) adalah kombinasi huruf, angka dan karakter khusus sebagai pengenal dan pengaman dalam mengakses sistem komputer.

Identitas Pengguna (*Account*) adalah data pengguna yang perlu



dicatat untuk mendapatkan alokasi ruang dalam mengoperasikan Sistem Informasi dengan memasukkan kode akses.

Sistem Pengamanan (*Security System*) adalah sistem yang dibangun untuk mencegah pengaksesan secara tidak sah dan perusakan, serta menjamin kerahasiaan data.

Penampil Informasi/Penjelajah (*Browser*) adalah perangkat lunak untuk menjelajah data dan informasi yang terdapat pada jaringan komputer baik melalui *intranet* maupun *Internet*.

Pengaman Sistem Jaringan Komputer (*Firewall*) adalah perangkat lunak dan/atau perangkat keras untuk **menjamin pengguna yang memiliki otorisasi dalam mengakses jaringan**.

Domain Name System (DNS) adalah *distribute database system* yang digunakan untuk pencarian nama komputer (name resolution) di jaringan yang menggunakan TCP/IP (Transmission Control Protocol/Internet Protocol). DNS biasa digunakan pada aplikasi yang terhubung ke Internet seperti web browser atau e-mail, dimana DNS membantu memetakan host name sebuah komputer ke IP address.

Proxy server (peladen **proxy**) adalah sebuah komputer **server** atau program komputer yang dapat bertindak sebagai komputer lainnya untuk melakukan request terhadap content dari Internet atau intranet.

Manajemen jaringan merupakan kemampuan untuk **mengontrol dan memonitor sebuah jaringan** komputer dari sebuah lokasi.

Wireless adalah jika dari arti katanya dapat diartikan “tanpa kabel”, yaitu melakukan suatu hubungan telekomunikasi menggunakan gelombang elektromagnetik sebagai pengganti media kabel.

Virtual Local Area Network (VLAN) adalah metode untuk menciptakan jaringan-jaringan yang secara logika tersusun sendiri-sendiri. **VLAN** sendiri berada dalam jaringan Local Area Network (LAN), sehingga dalam jaringan (LAN) bisa terdapat satu atau lebih **VLAN**.

Router adalah sebuah alat yang mengirimkan paket data melalui sebuah jaringan atau Internet menuju tujuannya, melalui sebuah proses yang dikenal sebagai routing.



POLITEKNIK NEGERI BANDUNG

Standar Pengelolaan Jaringan SI

Kode/No :

PL1.R1/STD/Pend/Polban/BAAK-17

Tanggal: 30 Agustus 2016

Revisi : 0

Halaman: 4 dari 12

2. Rationale	<p>Standar Pengelolaan Jaringan SI dimaksudkan agar Tim Pengelola Jaringan dan semua pihak yang terlibat dalam pengelolaan jaringan SI di Politeknik Negeri Bandung memiliki petunjuk tentang tata cara yang jelas, baku, dan terukur serta mengetahui bagaimana dan kapan harus dilakukan, di mana, dan oleh siapa dilakukan. Pada dasarnya Standar ini merupakan pedoman yang berisi prosedur-prosedur operasional standar yang digunakan untuk memastikan bahwa semua keputusan dan tindakan, serta penggunaan fasilitas-fasilitas proses yang dilakukan oleh Tim Pengelola Jaringan dalam memberikan layanan pengelolaan jaringan SI berjalan secara efisien dan efektif, konsisten, standar dan sistematis.</p> <p>Standar Pengelolaan Jaringan SI bertujuan untuk menciptakan tata kelola yang baik dan terstandardisasi sehingga semua kegiatan layanan akan dapat dilakukan secara konsisten oleh siapapun yang sedang bertugas melakukan layanan.</p> <p>Layanan-layanan yang berbelit, tumpang tindih dan tidak jelas prosedur operasinya akan semakin terminimalisir.</p> <p>Selanjutnya dengan adanya Standar Pengelolaan Jaringan SI ini setiap stakeholder baik pengguna jaringan maupun petugas yang melakukan pengelolaan dan memberi layanan akan dapat meningkatkan layanan yang semakin hari semakin baik dan semakin cepat karena terjadinya proses pembelajaran yang secara terus menerus. Dengan demikin dapat diharapkan melalui Standar Pengelolaan Jaringan SI ini akan dapat meningkatkan efisiensi dan efektifitas kerja pengelolaan jaringan di Politeknik Negeri Bandung.</p>
3. Pernyataan Isi Standar	<p>1. Sistem Jaringan Komputer (LAN/WAN)</p> <p>Dalam pengelolaan sistem jaringan, khususnya intranet Politeknik Negeri Bandung harus didukung oleh beberapa faktor, diantaranya:</p> <p>1.1. Perangkat Keras (Hardware)</p> <p>Untuk membangun sistem jaringan komputer diperlukan perangkat pendukung berupa <i>hardware</i>, diantaranya adalah:</p> <p>a. Media transmisi (<i>Wireline dan Wireless</i>) yang dapat mengkomunikasikan data (kabel <i>UTP</i>, serat optik, access point,</p>



- dan lain-lain);
- b. Konektor kabel transmisi ke peralatan (*Ethernet card, switch, router*, dan lain-lain);
 - c. *Network interface card (NIC)*;
 - d. Perangkat lunak jaringan (driver dari *NIC*);
 - e. Server Penyimpanan data (*storage device*).

1.2. Standardisasi Pengkabelan

Jaringan yang dibuat dan dikembangkan akan dapat terakses dengan baik apabila didukung oleh pengkabelan yang baik dan benar. Untuk itu dalam kerangka sistem jaringan di lingkungan Politeknik Negeri Bandung ditetapkan standar kabel yang digunakan dalam pemanfaatan sistem jaringan. Untuk Network Cabling diantaranya yaitu kable jenis UTP *Verified Category 5E* dan *Category 6*, dengan konektor RJ 45, *Fiber Optic* untuk koneksi antar gedung serta wireless untuk koneksi ke pengguna laptop, notebook atau *gadget* lainnya.

1.3. Standardisasi Alamat (*Internet Protocol/IP Address*)

Alamat IP ditulis berdasarkan standar yang dikeluarkan oleh InterNIC yaitu suatu organisasi yang bertanggung jawab dalam administrasi pengalaman IP Internet sedangkan untuk alamat lokal ditentukan berdasarkan otoritas penomoran internet yaitu Internet Assigned Numbers Authority (IANA).

1.4. Standardisasi Perangkat Lunak (*Software*)

- Antivirus

Antivirus merupakan program yang berguna untuk menjaga, mendekripsi dan menghapus virus dari sistem komputer.

Dengan demikian perlu digunakan standardisasi penggunaan *antivirus* untuk mencegah jaringan dari serangan virus yang dapat menyebabkan gangguan dalam menggunakan jaringan.

- Sistem operasi

Sistem operasi merupakan kumpulan program yang bertanggung jawab mengelola perangkat keras dan menyediakan berbagai fasilitas operasi dasar, misalnya penyimpanan file,



akses ke jaringan, eksekusi program dan pemanfaatan memori. Sistem operasi berbasis windows yang banyak digunakan pada di lingkungan Politeknik Negeri Bandung yaitu Windows XP Profesional, Windows 7, Windows 8, Windows 10 Profesional. Sedangkan sistem operasi yang digunakan untuk server adalah Linux Fedora, Linux CentOS, Linux Debian, Linux Ubuntu. dan distro Linux lainnya.

1.5. Standardisasi Koneksi Jaringan

Standardisasi koneksi jaringan yang digunakan di lingkungan Politeknik Negeri Bandung, yaitu koneksi jaringan antar gedung menggunakan *fiber optic*, dan antar komputer dalam satu gedung menerapkan desain jaringan topologi star dengan teknologi kabel atau tanpa kabel (wireless).

1.6. Standardisasi Pengelolaan Jaringan

Standardisasi pengelolaan jaringan untuk koneksi jaringan di pusat dan backbone jaringan antar gedung yang bertanggungjawab adalah pengelola pusat, sedangkan untuk pengelolaan di unit lain yang bertanggung jawab adalah jurusan atau unit masing-masing, kecuali ada surat perintah dari pejabat yang berwenang yaitu Pembantu Direktur Bidang Perencanaan dan Pengembangan dan Ka BAPSI, untuk pengelolaan jaringan di unit atau jurusan tertentu.

2. Autentifikasi User Internet

Untuk semua pengguna Internet di lingkungan Politeknik Negeri Bandung diberlakukan autentifikasi ketika akan membuka halaman web internet, agar bisa terkontrol penggunaanya. Setiap pengguna baik mahasiswa, staff, dan dosen wajib mempunyai Kode Akses (*password*) untuk akses Internet.

3. Pemeliharaan dan Perawatan Jaringan

Setelah seluruh sistem jaringan yang dibangun dan dikembangkan selesai, maka sebagai tahap selanjutnya diperlukan pemeliharaan sebagai upaya jangka panjang guna mempermudah dan memperlancar akses dalam pemakaian fasilitas jaringan. Dalam hal ini peran Unit



Pengelola Sistem Informasi Politeknik Negeri Bandung untuk menyiapkan tenaga teknisi yang handal dan mampu memahami bidangnya secara profesional. Standardisasi pemeliharaan jaringan adalah *updating service* serta *updating* dan *upgrade* sistem operasi. Dan yang paling penting dalam pemeliharaan untuk komputer client yaitu: peremajaan (*Updating*) *antivirus*, *Windows* atau *microsoft office*, pengecekan pada *hardware* apakah masih layak pakai atau terdapat *hardware* yang sudah lemah, serta harus ganti *hardware* atau di tingkatkan (*upgrade*), atau penggantian jaringan bila terjadi kegagalan koneksi. Pengecekan pada sistem jaringan seperti *IP address*, *dns*, *domain* atau *workgroup*, *subnet mask*, dan *gateway* apakah sudah terkonfigurasi dengan benar atau tidak. Standardisasi pemeliharaan yang dilakukan setelah sistem informasi atau program yang telah selesai dibuat adalah pembuatan dokumentasi sistem, petunjuk operasional sistem, mengadakan pelatihan untuk pengguna sistem, memperbaiki sistem apabila terdapat kesalahan (*bug*) pada sistem tersebut, serta menambahkan fasilitas pada sistem agar sistem tetap yang terkini (*up-to-date*).

4. Manajemen Jaringan

The International Organization for Standardization (ISO) mendefinisikan sebuah model konseptual untuk menjelaskan fungsi manajemen jaringan.

a. Manajemen Kesalahan (FaultManagement),

menyediakan fasilitas yang memungkinkan administrator jaringan untuk mengetahui kesalahan (fault) pada perangkat yang dikelola, jaringan, dan operasi jaringan, agar dapat segera menentukan apa penyebabnya dan dapat segera mengambil tindakan (perbaikan). Untuk itu, manajemen kesalahan memiliki mekanisme untuk :

- Melaporkan terjadinya kesalahan
- Mencatat laporan kesalahan (logging)
- Melakukan diagnosis



	<ul style="list-style-type: none">• Mengoreksi kesalahan (dimungkinkan secara otomatis) <p>b. Manajemen Konfigurasi (Configuration Management), memonitor informasi konfigurasi jaringan sehingga dampak dari perangkat keras atau pun lunak tertentu dapat dikelola dengan baik. Hal tersebut dapat dilakukan dengan kemampuan untuk inisialisasi, konfigurasi ulang, pengoperasian, dan mematikan perangkat yang dikelola. Dalam pengelolaan biasanya diperlukan sebuah konfigurasi jaringan untuk mengatasi sebuah masalah, contohnya untuk mengatasi Broadcast maka cara yang dilakukan adalah contohnya dengan menerapkan konsep VLAN atau juga dengan menerapkan metode routing antar unit atau jurusan menggunakan <i>Router</i>.</p> <p>c. Pelaporan (Accounting), mengukur utilisasi jaringan dari pengguna atau grup tertentu untuk:</p> <ul style="list-style-type: none">• Menghasilkan informasi tagihan (billing)• Mengatur pengguna atau grup• Membantu dalam menjaga performa jaringan pada level tertentu yang dapat diterima <p>d. Manajemen Performa (Performance Management), mengukur berbagai aspek dari performa jaringan termasuk pengumpulan dan analisis dari data statistik sistem sehingga dapat dikelola dan dipertahankan pada level tertentu yang dapat diterima. Untuk itu, manajemen performa memiliki kemampuan untuk:</p> <ul style="list-style-type: none">• Memperoleh utilisasi dan tingkat kesalahan dari perangkat jaringan• Mempertahankan performa pada level tertentu dengan memastikan prangkat memiliki kapasitas yang mencukupi <p>d. Manajemen Keamanan (Security Management), mengatur akses ke sumber daya jaringan sehingga informasi tidak dapat diperoleh tanpa izin. Hal tersebut dilakukan dengan cara:</p>
--	--



	<ul style="list-style-type: none">• Membatasi akses ke sumber daya jaringan• Memberi pemberitahuan akan adanya usaha pelanggaran dan pelanggaran keamanan
4. Strategi	<p>A. Program Peningkatan Kemampuan di sisi Pengelola</p> <p>Program peningkatan kemampuan Sumber Daya Manusia (SDM) mempunyai dua kebutuhan dasar yang menjadi patokan dalam aktivitas inventarisasi dan kebutuhan, yaitu:</p> <ol style="list-style-type: none">1. Kebutuhan untuk memperkenalkan program kerja yang didukung teknologi elektronis dengan sendirinya membutuhkan penguasaan keahlian baru (<i>instructional needs</i>).2. Kebutuhan untuk dapat mencapai/memenuhi standar sertifikasi keahlian direalisasikan melalui pelatihan SDM dibidang keahlian baru (<i>need assessment</i>). <p>Tujuan utama peningkatan kemampuan SDM adalah untuk dapat memenuhi kebutuhan peningkatan kemampuan tersebut diatas. Secara spesifik perlu digambarkan struktur pencapaiannya, yang pada prinsipnya menjelaskan bagaimana tujuan global direncanakan akan dicapai. Struktur umum pencapaian tujuan utama, digambarkan secara berjenjang dalam urutan beberapa pencapaian tujuan antara, sehingga keberhasilan pencapaian tujuan utama dengan mudah dapat dievaluasi. Adapun urutan pencapaian tujuan tersebut dimulai dari tujuan utama sampai dengan tujuan elementer adalah sebagai berikut :</p> <p>a. Tujuan Utama</p> <p>Merupakan tujuan akhir program dari peningkatan kemampuan SDM. Keberhasilan pemanfaatan Sistem Informasi dalam rangka otomasi prosedur kerja harian instansi pemerintah untuk meningkatkan efektifitas dan efisiensi kerja. Sebagai contoh: metode manual pengolahan data digantikan dengan metode elektronis secara lintas instansi; pengarsipan manual digantikan dengan pengarsipan elektronis lintas instansi.</p>



b. Tujuan Program

Merupakan tujuan antara turunan level pertama dari tujuan utama yang spesifik diterapkan dalam masing-masing unit program. Tujuan utama memiliki beberapa tujuan program. Sebagai contoh untuk dapat merealisasikan tujuan utama pada butir a, harus disiapkan SDM yang kompeten untuk mengaplikasikan program *networking* (LAN/WAN) sehingga komunikasi elektronis internal dan lintas instansi dapat direalisasikan untuk dapat mengoperasikan penyiapan dan pengolahan data elektronis harus disiapkan personal yang mampu mengoperasikan program *database* instansi pemerintah tersebut.

c. Tujuan Kursus/Unit Peningkatan Kemampuan

Merupakan tujuan antara turunan level kedua dari tujuan utama. Setiap tujuan program memiliki beberapa tujuan kursus. Sebagai contoh untuk menjadi kompeten didalam mengoperasikan jaringan (LAN/WAN), SDM harus mampu mengimplementasikan pengetahuan dasar protocol komunikasi dasar; juga menguasai pengoperasian *hardware/Perangkat lunak (Software)* komunikasi yang terkait dengan peralatan komunikasi seperti router, switch, hub dan lain sebagainya melalui unit pelatihan pengoperasian peralatan komunikasi.

d. Tujuan penguasaan Kemampuan Operasional Elementer (*Enabling Objectives*)

Merupakan tujuan antara turunan level ketiga dari tujuan utama. Setiap tujuan kursus/unit peningkatan kemampuan didalamnya terkandung beberapa tujuan, berupa kemampuan penguasaan operasional elementer. Sebagai contoh kursus/unit peningkatan kemampuan protokol komunikasi dasar mempunyai beberapa *enabling objectives*, misalnya kemampuan mendesain dan mempersiapkan cetak biru pengembangan LAN; kemampuan mendesain dan mempersiapkan cetak biru WAN. Contoh lain *enabling objectives* ditingkat paling elementer untuk program



	<p>aplikasi MS-Word, antara lain kemampuan memformat dokumen, kemampuan menggabungkan beberapa file menjadi satu file dan lain sebagainya.</p> <p>B. Program Peningkatan kemampuan disisi Pengguna</p> <p>Program peningkatan kemampuan disisi pengguna sangat diperlukan untuk membantu dalam pengelolaan jaringan di Politeknik Negeri Bandung, dalam hal ini pengguna diharapkan minimal bisa mengetahui tentang informasi yang dibutuhkan agar komunikasi dan informasi yang dibutuhkan bisa terpenuhi.</p> <p>1. Pelatihan Netiquette</p> <p>Pelatihan ini diperlukan agar pengguna Jaringan SI bisa mengetahui aturan-aturan yang di terapkan di Politeknik Negeri Bandung dalam pengelolaannya,</p> <p>2. Pelatihan Instalasi Antivirus</p> <p>Pelatihan ini diperlukan agar pengguna bisa mengimplementasikan bagaimana cara-cara instalasi Antivirus di setiap komputer masing-masing individu pengguna Jaringan SI di Politeknik Negeri Bandung.</p>
5. Indikator	<ol style="list-style-type: none">1. Terciptanya Layanan Internet dan Intranet yang handal, yang sesuai dengan kebutuhan akses Sistem Informasi di semua unit kerja.2. Pemakaian komputer yang stabil dan bebas virus di semua unit kerja.3. <i>Update</i> dan <i>upgrade</i> terhadap teknologi yang berkembang dan mengikuti perubahannya.
6. Subyek atau Pihak yang bertanggung-jawab untuk mencapai/memenuhi isi standar	<ol style="list-style-type: none">1. Pembantu Direktur Bidang Perencanaan dan Pengembangan2. Ka. BAPSI
7. Referensi	<ol style="list-style-type: none">1. Undang-Undang Republik Indonesia Nomor 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik.2. Peraturan Menteri Pendidikan dan Pendidikan Tinggi Republik Indonesia tahun 2006 tentang Statuta Politeknik Negeri Bandung.



**POLITEKNIK NEGERI
BANDUNG**

Standar Pengelolaan Jaringan SI

Kode/No :	PL1.R1/STD/Pend/Polban/BAAK-17
Tanggal:	30 Agustus 2016
Revisi :	0
Halaman:	12 dari 12

	<p>3. Peraturan Komisi Informasi Pusat Republik Indonesia tentang Standar Layanan Informasi Publik No.1 Tahun 2010.</p> <p>4. Masterplan IT Politeknik Negeri Bandung tahun 2016.</p>
8. Verifikasi	<p>Standar ini telah dikaji ulang oleh Pembantu Direktur Bidang Perencanaan dan Pengembangan telah diperiksa oleh Pejabat Satuan Penjaminan Mutu dan disetujui oleh Kepala Bagian Administrasi Perencanaan dan Sistem Informasi. Standar ini telah diketahui dan disimpan di Satuan Penjaminan Mutu untuk pengendalian melalui proses Audit Mutu, sedangkan pengendalian lapangan dilakukan oleh Gugus Penjamin Mutu unit kerja.</p>

LAMPIRAN 3
DOKUMEN FORMAL PERIZINAN PERMINTAAN DATA
DAN FASILITAS PENELITIAN



KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI POLITEKNIK NEGERI BANDUNG

Jln. Gegerkalong Hilir, Ds. Ciwaruga, Bandung 40012, Kotak Pos 1234, Telepon (022) 2013789, Fax. (022) 2013889

Homepage : www.polban.ac.id Email : polban@polban.ac.id

Nomor : 1330/PL1.KO/KN/2018
Perihal : Permohonan Survey Data

17 JANUARI 2018

Yth. Pembantu Direktur IV
Bidang Perencanaan
Politeknik Negeri Bandung

Sehubungan dengan mahasiswa Jurusan Teknik Komputer dan Informatika yang sedang mengerjakan mata kuliah **Tugas Akhir**, pada pelaksanaannya membutuhkan data / informasi tentang **Topologi Jaringan, Traffic Flow, Konfigurasi dan Log Proxy, Routing, Firewall, dan Quality Of Service (Qos Terkait Klasifikasi dan Prioritas Traffic Serta Alokasi Bandwidth)**.

Untuk itu kami mengajukan permohonan kepada Bapak, kiranya dapat menerima mahasiswa melakukan survey dan pengumpulan data/ informasi dimaksud di atas.

Mahasiswa yang akan melakukan survey dan pengumpulan data/ informasi, sebagai berikut :

Nama : Muhammad Saiful Islam
NIM : 141524020
Kelas : 4A/D-IV
Durasi : 18 Januari 2018 s.d 26 Januari 2018

Atas perhatian dan dikabulkan Bapak, diucapkan terima kasih.

Jurusan Teknik Komputer dan Informatika

Ketua

EDDY B. SORWONO, Drs., M.Kom.
NIP.196101141992021001

TEMBUSAN :
Pertinggal.

Bandung, 5 April 2018

Hal: Permohonan Resource IP Publik untuk Tugas Akhir

Yth. Drs. Mulyadi Yuswandono, Dipl.Ing., M.T.
Pembantu Direktur IV Bidang Perencanaan dan Pengembangan
di Politeknik Negeri Bandung

*Bismillahirrahmanirrahim,
Assalamu'alaikum warahmatullahi wabarakatuh.*

Dengan hormat,

Terima kasih atas dukungan dari Bapak untuk tugas akhir saya dengan topik **Model Pengelolaan Akses Web dengan Transparent Web Proxy di Politeknik Negeri Bandung** yang diawali dengan surat dari Ketua Jurusan saya bernomor 1330/PL1.KO/KN/2018. *Alhamdulillah* tugas akhir ini sudah diseminarkan dua kali di Jurusan Teknik Komputer dan Informatika.

Selanjutnya, menyambung pertemuan dengan Bapak kemarin, 4 April 2018, melalui surat ini saya memohon bantuan kepada Bapak untuk dapat menyediakan *resource* terkait eksperimen tugas akhir saya.

Resource tersebut berupa **satu buah IP publik** yang terpisah dari IP publik Jurusan Teknik Komputer dan Informatika dari blok 103.209.131.0/24 milik Politeknik Negeri Bandung. IP publik ini juga **perlu mendapat akses langsung ke Internet tanpa pembatasan firewall**.

IP publik ini akan digunakan sebagai media eksperimen usulan model *web proxy* Politeknik Negeri Bandung dan **akan selesai digunakan di akhir bulan Agustus 2018** sesuai batas akhir penyerahan laporan untuk yudisium I tahun ini.

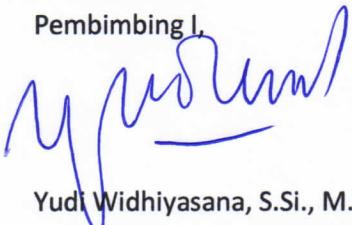
Besar harapan saya Bapak dapat membantu saya dalam penyelesaian tugas akhir ini.

Atas perhatian Bapak, saya mengucapkan terima kasih.

*Wabillahi taufiq wal hidayah,
Wassalamu'alaikum warahmatullahi wabarakatuh.*

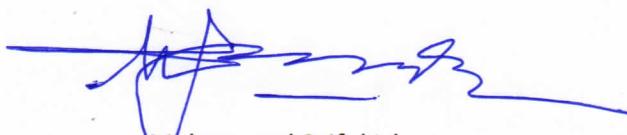
Mengetahui,

Pembimbing I,



Yudi Widhiyasana, S.Si., M.T.
NIP 197407182001121002

Hormat saya,



Muhammad Saiful Islam
NIM 141524020
Program Studi D4-Teknik Informatika

LAMPIRAN 4
SOURCE CODE PROGRAM EKSPERIMEN

Program 4-1. Penyaring log

```
#!/usr/bin/python3 -u

import ipaddress
import sys

for line in sys.stdin:
    line = line.strip()
    split = line.split()

    # Buang entri log koneksi HTTPS
    if split[5] != 'CONNECT':
        continue

    # Buang entri log koneksi di port 443
    if split[6][-3:] != '443':
        continue

    # Buang entri log koneksi yang berhasil (kode 200)
    if split[3][-3:] != '200':
        continue

    # Buang entri log dengan hostname yang merupakan alamat IP
    #
    # Blok except ValueError tidak akan dimasuki jika Python berhasil
    # melakukan parsing hostname menjadi struktur data alamat IP
    try:
        ipaddress.ip_address(split[6][:4])
        continue
    except ValueError:
        print(line)
```

Program 4-2. Pengambil data *hostname*

```
#!/usr/bin/awk -Mf

{
    sub(/:443$/,"", $7);
    domain[$7] = 1;
}

END {
    for (d in domain) {
        print d;
    }
}
```

Program 4-3. Pemeriksa keaktifan *hostname*

```
#!/usr/bin/python3 -u

from threading import Thread, Semaphore
import ssl
import sys
import requests
```

```

import socket

sph = Semaphore(128)

available = {}

def connect(hostname, sni):
    try:
        context = ssl.create_default_context()
        sock = socket.socket(socket.AF_INET)

        if sni:
            conn = context.wrap_socket(sock, server_hostname=hostname)
        else:
            context.check_hostname = False
            conn = context.wrap_socket(sock)

        conn.settimeout(10)
        conn.connect((hostname, 443))
        conn.shutdown(socket.SHUT_RDWR)
        conn.close()

        return True
    except:
        return False

class CheckerThread(Thread):
    def __init__(self, domain):
        super(CheckerThread, self).__init__()
        self.domain = domain

    def run(self):
        res = connect(self.domain, False)
        if res:
            print(self.domain, 'ok')
        else:
            res = connect(self.domain, True)
            if res:
                print(self.domain, 'ok')
            else:
                print(self.domain, 'fail')

        sph.release()

    def main():
        for line in sys.stdin:
            domain = line.strip()

            sph.acquire()
            CheckerThread(domain).start()

main()

```

Program 4-4. Penyaring log dari *hostname* yang tidak aktif

```

#!/usr/bin/python3 -u

import sys
import os

```

```

status = {}

def check_file(file):
    if not os.path.isfile(file):
        print("File path {} does not exist".format(file), file=sys.stderr)
        sys.exit(1)

def load_status_file(file):
    print('Loading status file from {} ...'.format(file), file=sys.stderr)
    check_file(file)

    with open(file) as fp:
        for i, line in enumerate(fp):
            line = line.strip().split()
            status[line[0]] = (line[1] == 'ok')

def main():
    status_file = sys.argv[1]
    load_status_file(status_file)

    print('Filtering inactive hostnames from stdin ...', file=sys.stderr)
    for line in sys.stdin:
        line = line.strip()
        split = line.split()

        if status.get(split[6][:4], False):
            print(line)

main()

```

Program 4-5. *Lookup* dan *reverse lookup* hostname ke DNS

```

#!/usr/bin/python3 -u

from threading import Thread, Semaphore
import ipaddress
import sys
import socket

sph = Semaphore(128)

domains = {}
ips = {}

def get_addresses(hostname):
    try:
        data = socket.gethostbyname_ex(hostname)
        return [(ip, 60 * 5) for ip in data[2]] # TTL: 5 minutes
    except socket.error:
        return []

def get_hostname(ip):
    try:
        data = socket.gethostbyaddr(ip)
        return (data[0], 60 * 5) # TTL: 5 minutes
    except socket.error:
        return (None, None)

class CheckerThread(Thread):
    def __init__(self, domain):

```

```

super(CheckerThread, self).__init__()
self.domain = domain

def run(self):
    try:
        ipaddress.ip_address(self.domain)

        # self.domain is a valid IP address
        ips[self.domain] = get_hostname(self.domain)

    except ValueError:
        # self.domain is not a valid IP address
        domains[self.domain] = get_addresses(self.domain)

        for ip in domains[self.domain]:
            ips[ip[0]] = get_hostname(ip[0])

    sph.release()

def main():
    exp_name = sys.argv[1]

    threads = []

    for domain in sys.stdin:
        domain = domain.strip()

        sph.acquire()
        t = CheckerThread(domain)
        threads.append(t)

        t.start()

    for t in threads:
        t.join()

    with open('{}.hosts'.format(exp_name), 'w') as f:
        for d in domains:
            for i in domains[d]:
                print(d, i[0], i[1], file=f)

    with open('{}.ips'.format(exp_name), 'w') as f:
        for i in ips:
            print(i, ips[i][0], ips[i][1], file=f)

main()

```

Program 4-6. Pengambil sertifikat TLS

```

#!/usr/bin/python3 -u

from threading import Thread, Semaphore
import ssl
import sys
import socket
import pprint
import json

sph = Semaphore(128)

```

```

results = {}

def common_name(cert):
    for x in cert['subject']:
        if x[0][0] == 'commonName':
            return x[0][1]

    return None

def subject_alt_name(cert):
    return [x[1] for x in cert['subjectAltName'] if x[0] == 'DNS']

def connect(ip, domain):
    context = ssl.create_default_context()
    result = {
        'cert': None,
        'binary_cert': None,
        'result': {}
    }

    try:
        sock = socket.socket(socket.AF_INET)
        if domain is None:
            context.check_hostname = False
            conn = context.wrap_socket(sock)
        else:
            conn = context.wrap_socket(sock, server_hostname=domain)

        conn.settimeout(10)
        conn.connect((ip, 443))
        cert = conn.getpeercert()
        result['binary_cert'] = conn.getpeer cert(True)
        conn.shutdown(socket.SHUT_RDWR)
        conn.close()

        result['result']['error'] = None
        result['result']['common_name'] = common_name(cert)
        result['result']['subject_alt_name'] = subject_alt_name(cert)
    except ssl.SSLError as ex:
        result['result']['error'] = ex.reason
        result['binary_cert'] = None
    except ssl.CertificateError as ex:
        result['result']['error'] = 'ssl.CertificateError'
        result['binary_cert'] = None
    except socket.timeout:
        result['result']['error'] = 'socket.timeout'
        result['binary_cert'] = None
    except OSError:
        result['result']['error'] = 'OSError'
        result['binary_cert'] = None

    return result

class GetterThread(Thread):
    def __init__(self, domain, ip):
        super(GetterThread, self).__init__()
        self.domain = domain
        self.ip = ip

    def run(self):
        result = {}

        sni_res = connect(self.ip, self.domain)

```

```

non_sni_res = connect(self.ip, None)

result['sni'] = sni_res['result']
result['non_sni'] = non_sni_res['result']
result['same_cert'] = (sni_res['binary_cert'] ==
non_sni_res['binary_cert'])

if self.ip not in results:
    results[self.ip] = {}

results[self.ip][self.domain] = result

sph.release()

def main():
    threads = []

    for line in sys.stdin:
        line = line.strip().split()
        domain = line[0]
        ip = line[1]

        sph.acquire()
        t = GetterThread(domain, ip)
        threads.append(t)

        t.start()

    for t in threads:
        t.join()

    with open('certificates.tls', 'w') as fp:
        json.dump(results, fp)

main()

```

Program 4-7. Eksperimen utama

```

#!/usr/bin/python3 -u

import sys
import os
import ipaddress
import time
import json

def check_file(file):
    if not os.path.isfile(file):
        print("File path {} does not exist".format(file), file=sys.stderr)
        sys.exit(1)

class DNSError(Exception):
    pass

class TLSError(Exception):
    pass

class DNSServer():
    def __init__(self):
        self.domains = {}

```

```

        self.ips = {}
        self.reverse_log = {}

    def populate_from_file(self, exp_file):
        for file in ['{}.hosts'.format(exp_file), '{}.ips'.format(exp_file)]:
            check_file(file)

            with open('{}.hosts'.format(exp_file)) as fp:
                for i, line in enumerate(fp):
                    line = line.strip().split()

                    domain = line[0]
                    ip = line[1]
                    ttl = int(line[2])

                    if domain not in self.domains:
                        self.domains[domain] = []

                    self.domains[domain].append((ip, ttl))

            with open('{}.ips'.format(exp_file)) as fp:
                for i, line in enumerate(fp):
                    line = line.strip().split()

                    ip = line[0]
                    domain = line[1] if line[1] != 'None' else None
                    ttl = int(line[2]) if line[2] != 'None' else None

                    self.ips[ip] = (domain, ttl)

    def lookup(self, domain, client_addr, timestamp):
        ips = self.domains.get(domain, [])

        for ip in ips:
            self.reverse_log[(client_addr, ip[0])] = domain

        return [(ip[0], timestamp + ip[1]) for ip in ips]

    def reverse_lookup(self, ip):
        return self.ips.get(ip, (None, None))

    def get_domain_from_history(self, client_addr, ip):
        return self.reverse_log.get((client_addr, ip))

    class TLSServer():
        def __init__(self):
            self.data = {}

        def populate_from_file(self, exp_file):
            check_file('{}.certs'.format(exp_file))
            with open('{}.certs'.format(exp_file)) as fp:
                self.data = json.load(fp)

        def connect(self, ip, server_name):
            if ip not in self.data:
                raise TLSError()

            if server_name is None:
                return self.data[ip]['non_sni']

            if server_name not in self.data[ip]['sni']:
                return {'error': 'server_name not found from dump'}

```

```

        return self.data[ip]['sni'][server_name]

class TLSConnectRequest():
    def __init__(self, client, server_addr, server_name):
        self.client = client
        self.server_addr = server_addr
        self.server_name = server_name

class ProxyServer():
    def __init__(self, dns_server, tls_server, experimenter):
        self.dns_server = dns_server
        self.tls_server = tls_server
        self.experimenter = experimenter
        self.ident_method = None

    def __identify_dns_rev_ptr(self, request):
        res = self.dns_server.reverse_lookup(request.server_addr)
        if res[0] is not None:
            return res[0]

        return request.server_addr

    def __identify_dns_rev_client(self, request):
        res = self.dns_server.get_domain_from_history(request.client.address,
request.server_addr)
        if res is not None:
            return res

        return request.server_addr

    def __identify_sni(self, request):
        if request.server_name is not None:
            return request.server_name

        return request.server_addr

    def __identify_cert_cn(self, request):
        cert = self.tls_server.connect(request.server_addr,
request.server_name)

        common_name = cert.get('common_name')
        if common_name is None:
            return request.server_addr

        if common_name.startswith('*.'):
            common_name = common_name[2:]

        return common_name

    def connect(self, request):
        if self.ident_method is None:
            self.experimenter.ident_result = None

        if self.ident_method == 'dns_rev_ptr':
            self.experimenter.ident_result =
self.__identify_dns_rev_ptr(request)

        if self.ident_method == 'dns_rev_client':
            self.experimenter.ident_result =
self.__identify_dns_rev_client(request)

        if self.ident_method == 'sni':
            self.experimenter.ident_result = self.__identify_sni(request)

```

```

        if self.ident_method == 'cert_cn':
            self.experimenter.ident_result = self._identify_cert_cn(request)

class Client():
    def __init__(self, address, dns_server, proxy_server):
        self.address = address
        self.dns_server = dns_server
        self.dns_cache = {}
        self.proxy_server = proxy_server

    def request(self, server_name, timestamp, sni):
        dns_resp = self.dns_cache.get(server_name, (None, None))
        if dns_resp == (None, None) or dns_resp[1] < timestamp:
            resp = self.dns_server.lookup(server_name, self.address,
timestamp)
            if len(resp) == 0:
                raise DNSError()

        dns_resp = (resp[0][0], resp[0][1])
        self.dns_cache[server_name] = dns_resp

        req = TLSConnectRequest(self, dns_resp[0], None)
        try:
            ipaddress.ip_address(server_name)
        except ValueError:
            req.server_name = server_name

        if sni == False:
            req.server_name = None

        self.proxy_server.connect(req)

    def reset(self):
        self.dns_cache = {}

class Experimenter():
    def __init__(self, exp_file):
        self.exp_file = exp_file

        print('Populating DNS server...', file=sys.stderr)
        self.dns_server = DNSServer()
        self.dns_server.populate_from_file(self.exp_file)

        print('Populating TLS server...', file=sys.stderr)
        self.tls_server = TLSServer()
        self.tls_server.populate_from_file(self.exp_file)

        self.proxy_server = ProxyServer(self.dns_server, self.tls_server,
self)
        self.client = Client('127.0.0.1', self.dns_server, self.proxy_server)

        self.ident_result = None

    def run(self):
        for sni in [True, False]:
            for method in ['dns_rev_ptr', 'dns_rev_client', 'sni',
'cert_cn']:
                print('Running {} method{}...'.format(method, ' with SNI' if
sni else ''), file=sys.stderr)
                with open(self.exp_file) as flog:
                    with open('{}.{}.{}.explog'.format(self.exp_file, method,
'sni' if sni else 'non_sni'), 'w') as fexplog:

```

```

        self.proxy_server.ident_method = method

    for i, line in enumerate(flog):
        line = line.strip().split()

        timestamp = float(line[0])
        client_addr = line[2]
        server_name = line[6][:-4]

        self.client.address = client_addr
        try:
            self.client.request(server_name, timestamp,
sni)
            print(line[0], client_addr, server_name,
self.ident_result, file=fexplog)
        except DNSError:
            pass

        self.client.reset()

def main():
    check_file(sys.argv[1])

    experimenter = Experimenter(sys.argv[1])
    experimenter.run()

main()

```

Program 4-8. Pengumpul delapan log simulator *transparent web proxy*

```

#!/usr/bin/python3 -u

import sys
import os
import sqlite3

methods = ['dns_rev_ptr', 'dns_rev_client', 'sni', 'cert_cn']
results = {}
server_ips = {}
domains = {}
mutual_domains = {}
clients = {}
visit_count = {}

def check_file(file):
    if not os.path.isfile(file):
        print("File path {} does not exist".format(file), file=sys.stderr)
        sys.exit(1)

def prepare_results():
    print('Preparing results storage ...', file=sys.stderr)
    for method in methods:
        results[method] = {}

        for sni in [True, False]:
            results[method][sni] = {}

def calc_ip_domain(exp_name):
    print('Calculating info from {}.hosts ...'.format(exp_name),
file=sys.stderr)

```

```

check_file('{}.hosts'.format(exp_name))

with open('{}.hosts'.format(exp_name)) as fp:
    for i, line in enumerate(fp):
        line = line.strip().split()
        domain = line[0]
        ip = line[1]

        if domain not in server_ips:
            server_ips[domain] = []

        server_ips[domain].append(ip)

        if ip not in domains:
            domains[ip] = []

        domains[ip].append(domain)

        if domain not in clients:
            clients[domain] = set()

        if domain not in visit_count:
            visit_count[domain] = 0

        for method in methods:
            for sni in [True, False]:
                results[method][sni][domain] = {
                    'tp': 0,
                    'fp': 0,
                    'fn': 0
                }

for domain in server_ips:
    if domain not in mutual_domains:
        mutual_domains[domain] = set()

    for ip in server_ips[domain]:
        for md in domains[ip]:
            if md != domain:
                mutual_domains[domain].add(md)

def summarize(exp_name):
    for sni in [True, False]:
        for method in methods:
            exp_file = '{}.{}.{}.explog'.format(exp_name, method, 'sni' if
sni else 'non_sni')
            print('Analyzing {} ...'.format(exp_file))

            check_file(exp_file)

            with open(exp_file) as fp:
                for i, line in enumerate(fp):
                    line = line.strip().split()

                    timestamp = float(line[0])
                    client_addr = line[1]
                    truth = line[2]
                    guess = line[3]

                    clients[truth].add(client_addr)

                    visit_count[truth] += 1

```

```

        if truth == guess:
            results[method][sni][truth]['tp'] += 1
        else:
            results[method][sni][truth]['fn'] += 1

        if guess in results[method][sni]:
            results[method][sni][guess]['fp'] += 1

def export_to_sqlite(file_name):
    if os.path.isfile('{}.expdb'.format(file_name)):
        os.remove('{}.expdb'.format(file_name))

    conn = sqlite3.connect('{}.expdb'.format(file_name))

    # Create tables
    conn.execute('''
CREATE TABLE `domains` (
    `domain` TEXT NOT NULL,
    `server_ips` INTEGER NOT NULL,
    `mutual_domains` INTEGER NOT NULL,
    `visit_count` INTEGER NOT NULL,
    `clients` INTEGER NOT NULL,
    PRIMARY KEY(`domain`)
)
    ''')

    conn.execute('''
CREATE TABLE `results` (
    `method` TEXT NOT NULL,
    `sni` TEXT NOT NULL,
    `domain` TEXT NOT NULL,
    `tp` INTEGER NOT NULL,
    `fp` INTEGER NOT NULL,
    `fn` INTEGER NOT NULL,
    FOREIGN KEY(`domain`) REFERENCES `domains`(`domain`),
    PRIMARY KEY(`method`, `sni`, `domain`)
)
    ''')

    print('Dumping domains ...', file=sys.stderr)
    for domain in server_ips:
        if len(clients[domain]) == 0:
            continue

        conn.execute(
            '''INSERT INTO `domains` VALUES (?, ?, ?, ?, ?, ?)''',
            (
                domain,
                len(server_ips[domain]),
                len(mutual_domains[domain]),
                visit_count[domain],
                len(clients[domain])
            )
        )

    for sni in [True, False]:
        for method in methods:
            print('Dumping {} result{} ...'.format(method, ' with SNI' if sni
else ''), file=sys.stderr)
            for domain in server_ips:
                if len(clients[domain]) == 0:
                    continue

```

```

        tp = results[method][sni][domain]['tp']
        fp = results[method][sni][domain]['fp']
        fn = results[method][sni][domain]['fn']

        conn.execute(
            '''INSERT INTO `results` VALUES (?, ?, ?, ?, ?, ?, ?)''',
            (
                method,
                'sni' if sni else 'non_sni',
                domain,
                tp,
                fp,
                fn
            )
        )

        conn.commit()
        conn.close()

def main():
    exp_name = sys.argv[1]

    prepare_results()
    calc_ip_domain(exp_name)
    summarize(exp_name)
    export_to_sqlite(exp_name)

main()

```

Program 4-8. Bot Telegram untuk notifikasi eksperimen

```

#!/usr/bin/python3 -u

from urllib.parse import urlencode
from urllib.request import urlopen
import sys

token = [token from Telegram API]
chat_room = [personal Telegram chat room ID]
message = sys.argv[1]

endpoint = 'https://api.telegram.org/bot' + token + '/sendMessage'
body = urlencode({ 'chat_id': chat_room, 'text': message }).encode('ascii')

result = urlopen(endpoint, body).read()

```

Program 4-9. *Shell script* untuk menjalankan seluruh rangkaian eksperimen

```

#!/bin/bash

dir=hotspot

do
    cd /home/saiful/ta/$dir
    ..../bot.py3 "Started processing ${dir} log."

```

```

..../bot.py3 "Getting relevant entries from ${dir} log..."
..../filter-https-443-200.py3 < access-${dir}.log > https-access-${dir}.log
..../bot.py3 "Finished getting relevant entries from ${dir} log."

..../bot.py3 "Filtering hostnames from ${dir} log..."
..../unique.awk < https-access-${dir}.log > hostnames-access-${dir}.log
..../bot.py3 "Finished filtering hostnames from ${dir} log."

..../bot.py3 "Checking hostnames status from ${dir} log ..."
..../check.py3 < hostnames-access-${dir}.log > check-access-${dir}.log
..../bot.py3 "Finished checking hostnames status from ${dir} log."

..../bot.py3 "Filtering inactive hostnames from ${dir} log ..."
..../filter-inactive-log.py3 check-access-${dir}.log < https-access-
${dir}.log > clean-access-${dir}.log
..../bot.py3 "Finished filtering inactive hostnames from ${dir} log."

..../bot.py3 "Looking up DNS entries from ${dir} log ..."
awk '$2 == "ok" { print $1 }' < check-access-${dir}.log |
..../dns_lookup.py3 clean-access-${dir}.log
..../bot.py3 "Finished looking up DNS entries from ${dir} log."

..../bot.py3 "Getting TLS certificates from ${dir} log ..."
..../cert_lookup.py3 clean-access-${dir}.log
..../bot.py3 "Finished getting TLS certificates from ${dir} log."

..../bot.py3 "Starting experiment on ${dir} log ..."
..../experiment.py3 clean-access-${dir}.log
..../bot.py3 "Finished experiment on ${dir} log."

..../bot.py3 "Analyzing ${dir} experiment result ..."
..../analyzer.py3 clean-access-${dir}.log
..../bot.py3 "Finished analzing ${dir} experiment result."

..../bot.py3 "Finished processing ${dir} log."
done

```

LAMPIRAN 5
IMPLEMENTASI KONFIGURASI SKALA LAB
PADA LINGKUNGAN GNS3

Konfigurasi 5-1. sites-available/default pada FreeRadius

```
server default {
    listen {
        type = auth
        ipaddr = *
        port = 0
        limit {
            max_connections = 16
            lifetime = 0
            idle_timeout = 30
        }
    }
    listen {
        ipaddr = *
        port = 0
        type = acct
        limit {
        }
    }
    listen {
        type = auth
        ipv6addr = :: # any. ::1 == localhost
        port = 0
        limit {
            max_connections = 16
            lifetime = 0
            idle_timeout = 30
        }
    }
    listen {
        ipv6addr = ::#
        port = 0
        type = acct
        limit {
        }
    }
    authorize {
        filter_username
        preprocess
        chap
        mschap
        digest
        suffix
        eap {
            ok = return
        }
        files
        -sql
        -ldap
        expiration
        logintime
        pap
    }
    authenticate {
        Auth-Type PAP {
            pap
        }
        Auth-Type CHAP {
            chap
        }
        Auth-Type MS-CHAP {
            mschap
        }
    }
}
```

```

    }
    mschap
    digest
    eap
}
preacct {
    preprocess
    acct_unique
    suffix
    files
}
accounting {
    detail
    unix
    radutmp
    sradutmp
    -sql
    exec
    attr_filter.accounting_response
}
session {
}
post-auth {
    update {
        &reply: += &session-state:
    }
    -sql
    exec
    remove_reply_message_if_eap
Post-Auth-Type REJECT {
    -sql
    attr_filter.access_reject
    eap
    remove_reply_message_if_eap
}
}
pre-proxy {
}
post-proxy {
    eap
}
}

```

Konfigurasi 5-2. clients.conf pada FreeRadius

```

client localhost {
    ipaddr = 127.0.0.1
    proto = *
    secret = testing123
    require_message_authenticator = no
    nas_type      = other      # localhost isn't usually a NAS...
    limit {
        max_connections = 16
        lifetime = 0
        idle_timeout = 30
    }
}
client localhost_ipv6 {
    ipv6addr      = ::1
    secret        = testing123
}

```

```

}
client mikrotik {
    ipaddr      = 172.16.16.1
    secret      = rahasia
}

```

Konfigurasi 5-3. users pada FreeRadius

```

user1 Cleartext-Password := "user1"
user2 Cleartext-Password := "user2"
user3 Cleartext-Password := "user3"
...
user97 Cleartext-Password := "user97"
user98 Cleartext-Password := "user98"
user99 Cleartext-Password := "user99"

DEFAULTFramed-Protocol == PPP
    Framed-Protocol = PPP,
    Framed-Compression = Van-Jacobson-TCP-IP
DEFAULTHint == "CSLIP"
    Framed-Protocol = SLIP,
    Framed-Compression = Van-Jacobson-TCP-IP
DEFAULTHint == "SLIP"
    Framed-Protocol = SLIP

```

Bagian dicetak tebal dibuat dengan *shell script* dialek Bash berikut:

```

touch radiuspassfile
for i in {1..99}; do echo "user${i} Cleartext-Password := \"user${i}\"" >>
radiuspassfile; done

```

Konfigurasi 5-4. rules.v4 untuk modul *kernel iptables*

```

# Generated by iptables-save v1.6.0 on Sat May 26 10:21:21 2018
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
COMMIT
# Completed on Sat May 26 10:21:21 2018
# Generated by iptables-save v1.6.0 on Sat May 26 10:21:21 2018
*nat
:PREROUTING ACCEPT [8:1032]
:INPUT ACCEPT [8:1032]
:OUTPUT ACCEPT [9:686]
:POSTROUTING ACCEPT [9:686]
:NO_PROXY - [0:0]
-A PREROUTING -p tcp -m tcp --dport 80 -j NO_PROXY
-A PREROUTING -p tcp -m tcp --dport 443 -j NO_PROXY
-A PREROUTING -p tcp -m tcp --dport 80 -j REDIRECT --to-ports 3129
-A PREROUTING -p tcp -m tcp --dport 443 -j REDIRECT --to-ports 3130
-A NO_PROXY -d 172.16.16.2/32 -j ACCEPT
COMMIT
# Completed on Sat May 26 10:21:21 2018

```

Konfigurasi 5-5. htpasswd pada Squid

```
user1:$apr1$.Qa3ITwo$CeTHW.Kc/UrM0omyKQU1r/
user2:$apr1$aYkbki/L$ZiyXtvtoXokkEET5hCP8h.
user3:$apr1$BFKWJFn/$43WlbD5HqJtZ59E.ADC7q1
...
user97:$apr1$tm7UPD7r$6D0mKzzDaRuMufZr74cfi/
user98:$apr1$b6mESiKL$zVMrdfQCokEaUs71UW6xt0
user99:$apr1$mpEWhzQU$BS4AXO.8fUYDXXP/05iS20
```

Dibuat dengan perintah *shell script* dialek Bash berikut:

```
touch htpasswd
for i in {1..99}; do htpasswd -b htpasswd user${i} user${i}; done
```

Program 5-6. *Shell script* untuk kompilasi Squid dan pembuatan sertifikat TLS

```
# Kompilasi Squid
sudo apt update

mkdir squid
cd squid

sudo apt install packaging-dev debian-keyring devscripts build-essential
fakeroot debhelper dh-autoreconf cdbs equivs
sudo apt build-dep squid
sudo apt install libdbi-perl libssl1.0-dev
apt source squid

dget -x http://http.debian.net/debian/pool/main/s/squid/squid_4.1-1.dsc
cd squid-4.1/
vi debian/rules
patch debian/rules < ../rules.patch
dch --local ~bpo9+ --distribution stretch-backports "Rebuild for stretch-
backports."

dpkg-buildpackage -us -uc -b

sudo apt install squid-langpack

sudo dpkg --install squid-common_4.1-1~bpo9+1_all.deb
sudo dpkg --install squid_4.1-1~bpo9+1_armhf.deb

sudo apt-mark hold squid squid-common

# Pembuatan sertifikat TLS
openssl genrsa -des3 -out server.key.pem 1024
openssl req -new -key server.key.pem -out server.csr
openssl rsa -in server.key.pem -out server.key
openssl x509 -req -days 365 -in server.csr -signkey server.key -out
server.crt

sudo mkdir -p /etc/squid/ssl
cat server.key | sudo tee /etc/squid/ssl/server.key
cat server.crt | sudo tee /etc/squid/ssl/server.crt
```

Berkas 5-7. rules.patch untuk kompilasi Squid dengan SNI

```
--- rules      2018-02-12 04:00:18.000000000 +0700
+++ rules.new  2018-05-20 16:52:14.514860382 +0700
@@ -28,6 +28,9 @@
@@
--enable-delay-pools \
--enable-cache-digests \
--enable-icap-client \
+
--enable-ssl \
+
--enable-ssl-crtd \
+
--with-openssl \
--enable-follow-x-forwarded-for \
--enable-auth-
basic="DB,fake,getpwnam,LDAP,NCSA,NIS,PAM,POP3,RADIUS,SASL,SMB" \
--enable-auth-digest="file,LDAP" \
```

Konfigurasi 5-8. squid.conf pada Squid

```
acl SSL_ports port 443
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl CONNECT method CONNECT

acl clients_net src 192.168.72.0/24
acl mgmt_net src 192.168.88.0/24

auth_param basic program /usr/lib/squid3/basic_ncsa_auth /etc/squid/htpasswd
acl basic_login proxy_auth REQUIRED

external_acl_type ext_login ttl=0 %>a /home/pi/lookup.sh
acl radius_login external ext_login

acl step1 at_step SslBump1
acl step2 at_step SslBump2
acl step3 at_step SslBump3

http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localhost manager
http_access deny manager
http_access allow localhost
http_access allow clients_net radius_login
http_access allow mgmt_net radius_login
http_access deny all

http_port 3128
http_port 3129 intercept
https_port 3130 intercept ssl-bump tls-cert=/etc/squid/ssl/server.crt tls-key=/etc/squid/ssl/server.key generate-host-certificates=off

ssl_bump peek step1
ssl_bump splice step2
```

```

access_log daemon:/var/log/squid/access-transparent.log squid

coredump_dir /var/spool/squid
refresh_pattern ^ftp:      1440    20%    10080
refresh_pattern ^gopher:   1440    0%     1440
refresh_pattern -i (/cgi-bin/|\.?) 0  0%    0
refresh_pattern .          0     20%    4320

```

Program 5-9. lookup.sh untuk komunikasi Squid dengan FreeRadius

```

#!/bin/bash

while read -a line; do
    user=$(radwho -r | grep -e ",${line[0]}\"$' | grep -o "^[^,]\+")
    if [ -z $user ]; then
        echo "ERR"
    else
        echo "OK user=${user}"
    fi
done

```

Konfigurasi 5-10. Public Router

```

# jul/19/2018 03:58:16 by RouterOS 6.41.4
# software id =
#
#
#
/interface ethernet
set [ find default-name=ether1 ] name=ether1-ISP
set [ find default-name=ether2 ] name=ether2-public
/interface wireless security-profiles
set [ find default=yes ] supplicant-identity=MikroTik
/ip address
add address=172.16.16.1/24 interface=ether2-public network=172.16.16.0
/ip dhcp-client
add dhcp-options=hostname,clientid disabled=no interface=ether1-ISP
/ip dns
set allow-remote-requests=yes
/ip firewall nat
add action=masquerade chain=srcnat out-interface=ether1-ISP
/system identity
set name=InternetGW

```

Konfigurasi 5-11. Private Router

```

# jul/19/2018 03:58:21 by RouterOS 6.41.4
# software id =
#
#
#
/interface ethernet
set [ find default-name=ether1 ] name=ether1-server

```

```

set [ find default-name=ether2 ] name=ether2-client
/interface wireless security-profiles
set [ find default=yes ] supplicant-identity=MikroTik
/ip hotspot profile
add hotspot-address=192.168.72.1 login-by=http-chap name=hsprof1 use-radius=\ 
    yes
/ip pool
add name=pool_client ranges=192.168.72.3-192.168.72.254
/ip dhcp-server
add address-pool=pool_client disabled=no interface=ether2-client lease-time=\ 
    1h name=dhcp-client
/ip hotspot
add address-pool=pool_client idle-timeout=none interface=ether2-client name=\ 
    server1 profile=hsprof1
/ip firewall connection tracking
set enabled=yes
/ip address
add address=192.168.99.1/24 interface=ether1-server network=192.168.99.0
add address=192.168.72.1/24 interface=ether2-client network=192.168.72.0
/ip dhcp-client
add disabled=no interface=ether1-server
/ip dhcp-server network
add address=192.168.72.0/24 dns-server=192.168.99.2 gateway=192.168.72.1
/ip dns
set servers=192.168.99.2
/ip firewall filter
add action=passthrough chain=unused-hs-chain comment=\ 
    "place hotspot rules here" disabled=yes
/ip firewall mangle
add action=mark-routing chain=prerouting dst-port=80 in-interface=\ 
    ether2-client new-routing-mark=to-proxy passthrough=yes protocol=tcp
add action=mark-routing chain=prerouting dst-port=443 in-interface=\ 
    ether2-client new-routing-mark=to-proxy passthrough=yes protocol=tcp
/ip firewall nat
add action=passthrough chain=unused-hs-chain comment=\ 
    "place hotspot rules here" disabled=yes
/ip hotspot walled-garden
add comment="place hotspot rules here" disabled=yes
/ip hotspot walled-garden ip
add action=accept disabled=no dst-address=192.168.99.2 !dst-address-list \
    !dst-port !protocol !src-address !src-address-list
/ip route
add distance=1 gateway=192.168.99.3 routing-mark=to-proxy
/radius
add address=192.168.99.4 secret=secret service=hotspot
/system identity
set name=IntranetGW

```

LAMPIRAN 6
IMPLEMENTASI KONFIGURASI SKALA LAB
PADA LINGKUNGAN *DEPLOYMENT*

Konfigurasi 6-1. MikroTik RouterOS pada lingkungan *deployment*

```
# jul/07/2018 20:09:51 by RouterOS 6.42.3
# software id = 9PWD-ENIJ
#
# model = RouterBOARD 750G r3
# serial number = 6F3807182149
/interface ethernet
set [ find default-name=ether1 ] name=ether1-ISP
set [ find default-name=ether2 ] name=ether2-server
set [ find default-name=ether3 ] name=ether3-client
set [ find default-name=ether5 ] name=ether5-mgmt
/interface wireless security-profiles
set [ find default=yes ] supplicant-identity=MikroTik
/ip hotspot profile
set [ find default=yes ] html-directory=flash/hotspot
add hotspot-address=192.168.72.1 html-directory=flash/hotspot login-by=\http-chap name=hsprof-client use-radius=yes
add hotspot-address=192.168.88.1 html-directory=flash/hotspot login-by=\http-chap name=hsprof-mgmt use-radius=yes
/ip pool
add name=pool_mgmt ranges=192.168.88.2-192.168.88.254
add name=pool_client ranges=192.168.72.2-192.168.72.254
/ip dhcp-server
add address-pool=pool_mgmt authoritative=after-2sec-delay disabled=no \
interface=ether5-mgmt lease-time=1h name=server-mgmt
add address-pool=pool_client authoritative=after-2sec-delay disabled=no \
interface=ether3-client lease-time=1h name=server-client
/ip hotspot
add address-pool=pool_client disabled=no idle-timeout=none interface=\ether3-client name=server-client profile=hsprof-client
add address-pool=pool_mgmt idle-timeout=none interface=ether5-mgmt name=\server-mgmt profile=hsprof-mgmt
/ip address
add address=192.168.88.1/24 interface=ether5-mgmt network=192.168.88.0
add address=172.16.16.1/24 interface=ether2-server network=172.16.16.0
add address=10.10.62.15/24 interface=ether1-ISP network=10.10.62.0
add address=192.168.72.1/24 interface=ether3-client network=192.168.72.0
add address=10.10.194.248/24 interface=ether1-ISP network=10.10.194.0
add address=10.10.193.248/24 interface=ether1-ISP network=10.10.193.0
add address=10.10.192.248/24 interface=ether1-ISP network=10.10.192.0
add address=10.10.195.248/24 interface=ether1-ISP network=10.10.195.0
add address=10.10.196.248/24 interface=ether1-ISP network=10.10.196.0
add address=10.10.248.248/24 interface=ether1-ISP network=10.10.248.0
/ip dhcp-client
add dhcp-options=hostname,clientid disabled=no interface=ether1-ISP
/ip dhcp-server network
add address=192.168.72.0/24 dns-server=192.168.72.1 gateway=192.168.72.1
add address=192.168.88.0/24 dns-server=192.168.88.1 gateway=192.168.88.1
/ip dns
set allow-remote-requests=yes
/ip firewall filter
add action=passthrough chain=unused-hs-chain comment="place hotspot rules here" \
disabled=yes
add action=drop chain=forward in-interface=ether3-client out-interface=\ether1-ISP
add action=drop chain=forward disabled=yes in-interface=ether5-mgmt \
out-interface=ether1-ISP
/ip firewall mangle
add action=mark-routing chain=prerouting dst-port=80,443 hotspot=\from-client,auth in-interface=ether3-client new-routing-mark=to-proxy \
passthrough=yes protocol=tcp
```

```

add action=mark-routing chain=prerouting disabled=yes dst-port=80,443 \
    in-interface=ether5-mgmt new-routing-mark=to-proxy passthrough=yes \
    protocol=tcp
/ip firewall nat
add action=passthrough chain=unused-hs-chain comment="place hotspot rules
here" \
    disabled=yes
add action=masquerade chain=srcnat out-interface=ether1-ISP
add action=dst-nat chain=dstnat dst-port=2222 in-interface=ether1-ISP
protocol=\
    tcp to-addresses=172.16.16.2 to-ports=22
/ip hotspot walled-garden
add comment="place hotspot rules here" disabled=yes
/ip hotspot walled-garden ip
add action=accept disabled=no dst-address=172.16.16.0/24 !dst-port !protocol
\
    !src-address
add action=accept disabled=no dst-address=192.168.88.1 !dst-port !protocol \
    !src-address
add action=accept disabled=no dst-address=192.168.72.1 !dst-port !protocol \
    !src-address
/ip route
add distance=1 gateway=172.16.16.2 routing-mark=to-proxy
add check-gateway=ping distance=1 gateway=10.10.62.1
add check-gateway=ping distance=1 gateway=10.10.194.1
add check-gateway=ping distance=1 gateway=10.10.248.1
add check-gateway=ping distance=1 gateway=10.10.195.1
add check-gateway=ping distance=1 gateway=10.10.196.1
add check-gateway=ping distance=1 gateway=10.10.193.1
add check-gateway=ping distance=1 gateway=10.10.192.1
/radius
add address=172.16.16.2 secret=rahasia service=hotspot
/system clock
set time-zone-name=Asia/Jakarta
/system routerboard settings
set silent-boot=no

```