

Diterima : February 01, 2021
Disetujui : February 05, 2021
Diterbitkan: February 24, 2021

**Conference on Management, Business,
Innovation, Education and Social Science**
<https://journal.uib.ac.id/index.php/combindes>

Analisa dan Perancangan Proxy Server dan Catching SSL menggunakan Squid

Haeruddin¹, Vincent²

Email korespondensi : haeruddin@uib.ac.id, 1731062.vincent@uib.edu

^{1, 2}Fakultas Ilmu komputer. Universitas Internasional Batam, Batam, Indonesia

Abstrak

Internet telah menjadi sarana pertukaran informasi dan komunikasi yang dapat menunjang produktivitas secara luas, hampir semua instansi terhubung ke layanan internet. Namun dikarenakan alokasi budget yang minim terhadap layanan internet hasil yang diterima tidak maksimal. *Proxy server* merupakan perantara yang berfungsi sebagai memenuhi permintaan *client* dengan menggunakan *squid* di *ubuntu server* dapat memberikan performa akses internet yang lebih baik kepada *client*. Studi ini menggunakan metode *NDLC (Network Development Life Cycle)* untuk melakukan perancangan *proxy server* dan implementasi *squid* di *ubuntu server*.

Kata Kunci : *Internet, NDLC, Proxy, Squid, Ubuntu server,*

Pendahuluan

Saat ini infrastruktur dan pengembangan teknologi sudah berkembang dengan pesat salah satunya adalah transisi evolusi industry 3.0 ke 4.0. Hal ini tidak akan berkembang pesat jika tidak ada regulasi dan kebijakan dari pemerintah, Salah satu terpentingnya adalah terkoneksi ke *internet*. *Internet* merupakan jaringan yang menghubungkan satu Bersama lainnya untuk pertukaran informasi dan komunikasi serta penunjang produktivitas dalam instansi maupun perusahaan (Wibawanto, 2018). Setiap perusahaan memiliki Bandwidth yang berbeda dikarenakan kebutuhan yang telah di sediakan oleh ISP (*Internet service provider*).

ISP merupakan layanan yang menyediakan internet kepada *client* sehingga ketika *client* mengakses ke sebuah website maka *IP address* tersebut akan muncul di *webserver* tersebut. Banyak *client* mencoba mengakses ke situs yang sama dikarenakan alasan yang berbeda dan situs yang di akses beragam dimulai dari *web/URL* yang bersifat dinamis dan frekuensi yang tinggi seperti : Berita, Forum, *Social Media* ataupun bersifat statis seperti : *Company profile*. Mengakses ke *website* yang sama secara berulang dapat menyebabkan *bandwidth* yang pakai oleh *client* lain tidak maksimal karena ketika *client* mengakses ulang terhadap situs yang sama maka *router* akan *capture* apa yang di *request* oleh *client* dan di *forward* ke tujuan (Lubis et al., 2020). *Proxy server* merupakan sebuah *server* yang berfungsi untuk melayani *request* yang diberikan oleh *client*, *proxy* ini juga dapat meneruskan, menolak

dan memantau dari setiap yang di *request* oleh *client* juga dapat memberikan keamanan privasi dalam mengakses ke *internet*.(R & Marisa, 2017)

Squid merupakan perangkat lunak yang dapat meningkatkan performa akses ke internet. *Squid* ini berfungsi *cache* dan *release* SSL yang pernah diakses oleh *client* sehingga ketika *client* *request* ulang tidak perlu *forward* ke tujuan *website* melainkan ke *squid* karena *squid* menyimpan *logfiles* yang pernah di akses oleh *client* namun jika terjadinya *update* terhadap *website* yang di tuju maka *squid* akan *forward* ke tujuan serta *cache* URL tersebut.(Kurniawan et al., 2020)

Tujuan yang ingin dicapai dari penelitian ini adalah memberikan performa akses *internet* lebih baik dan lebih cepat terhadap situs yang sama.

Tinjauan Pustaka

Penelitian yang dilakukan oleh (Khasanah, 2017) yang berjudul "**Performa Kecepatan Akses Internet Dengan Squid Proxy Server pada Ubuntu server 1.10**" yang diawali oleh penggunaan *internet* yang tidak dimbangi dengan penambahan *bandwidth* alhasil ingin mengakses ke situs *www.yahoo.com* membutuhkan 70 detik oleh karena itu dirancang *proxy server* diubuntu *server* dan juga *squid* untuk meningkatkan performa akses *internet*. Penelitian ini juga membandingkan performa akses *internet* sebelum menggunakan *proxy* dan sesudah hasilnya menunjukan efektifitas rata-rata 87.875%

Penelitian yang dilakukan oleh (Subandri & Hanadwiputra, 2018) yang berjudul "**Penerapan Teknologi Cache Server Berbasis Iot Dengan Raspberry Pi3 Menggunakan Metode Forward Chaining (Studi Kasus Smk Binakarya Mandiri 2 Kota Bekasi)**" yang diawali oleh jumlah pemakaian internet sebesar 1200 siswa dan 100 guru dengan kecepatan 10MB mengakibatkan akses internet tidak stabil dan putus-putus. Penelitian ini merancang *proxy server* didalam sistem operasi *Raspberry Pi 3 Model B* dan *squid* untuk *cache* yang disambungkan dengan mikrotik dengan metode *Forward Chaining*. Pengujian *caching* ini dilakukan dengan 3 kondisi dimana : *caching* di lakukan pada saat *proxy server* kosong, B saat *caching* pada *proxy server* terisi dan C pada saat kondisi dimana *caching* dilakukan di mikrotik dan melakukan testing performa menggunakan *wireshark* untuk *capture* waktu yang dibutuhkan sebelum dan sesudah menggunakan *proxy*. Hasilnya *proxy server* dapat membantu menyimpan *cache*.

Penelitian yang dilakukan oleh (Sirait & Siahaan, 2019) yang berjudul "**Analisis Kinerja Windows Server Dan Linux Server Terhadap Respons Time Squid Proxy**" diawali dengan ingin mengetahui sistem operasi apa yang dapat memberikan performa yang baik. penelitian ini membandingkan *respond time* dari *windows server* dan *linux server* dengan menggunakan *squid*. Pengujian ini dilakukan 10kali kesitus *server*. Hasilnya sistem operasi linux memberikan *response time* lebih besar dibandingkan dengan windows server dan penggunaan *squid* sebagai *proxy server* dapat meningkatkan performa akses internet.

Penelitian yang dilakukan oleh (Suroso, 2019) yang berjudul "**Optimasi Bandwidth Hits Local Up To Dengan Transparent Proxy 2.7 Lusca Menggunakan Metode Network Development Life Cycle**" diawali dengan akses internet di STMIK Bani saleh yang tidak stabil akibat dari penguasaan *bandwidth* individu , Perancangan *proxy server* ini menggunakan metode NDLC dan sistem operasi Linux OpenSUSE 12.1 juga menimplementasikan *filtering web* menggunakan *proxy* serta pengujian *reponse time* menggunakan *squid*. Hasilnya pengujian

respons time sebelum menggunakan *transparent proxy* adalah 90 detik setelah menggunakan *proxy* adalah 24 detik sehingga selisih *respons time* adalah 57 detik (73.33%)

Penelitian yang dilakukan oleh (Aminuddin & HS, 2018) yang berjudul "**Perancangan Proxy Server Dengan Menggunakan Squid di CV.Nusantara Geotech Makassar**" yang diawali dengan kesulitan dalam mengatur *bandwidth* yang diakses oleh kantor dan belum tersusun rapi serta beberapa akses internet berkonten spam, porno yang mengganggu aktivitas kantor. Perancangan ini menggunakan proxy server squid dengan membatasi penggunaan *bandwidth* dengan fitur *delay pool* dan *ACL (Access control list)* di *proxy server* berdasarkan *word, forbidden, IP* dan *url*. Hasilnya *user* dapat mengakses lebih cepat dan menyimpan layanan *cache* pada *proxy server* sehingga dapat mengurangi waktu akses tersebut.

Berikut merupakan penelitian yang dijadikan acuan oleh penulis sebagai berikut :

Tabel 1 : Tinjauan Pustaka

Teknologi		
Peneliti	Tahun	Kesimpulan
Khasanah & Fata Nidaul	2017	Penggunaan <i>proxy server Squid</i> pada ubuntu server dapat meningkatkan efisiensi
Subandri & Hanadwiputra, Sabar	2018	Penggunaan <i>Wireshark</i> sebagai menangkap situs yang diakses dan pengujian <i>respons time</i> terhadap 3 kondisi : A. Saat <i>proxy</i> kosong, B saat <i>proxy</i> terisi, C, Saat <i>proxy</i> di <i>caching</i> di mikrotik
Sirait, Parulian & Siahaan, R.Fanry	2019	Pengujian <i>repons time</i> menggunakan Linux server dan squid memberikan <i>respons time</i> yang lebih besar dibandingkan windows server
Suroso, Amat	2019	Pengujian perancangan menggunakan NDLC dan filtering web menggunakan <i>proxy</i> dan pengujian <i>time respons</i> menggunakan squid
Aminuddin, Rosihan & HS, Moch. Apriyadi	2018	Pengujian <i>delay pool</i> dan <i>ACL</i> di <i>proxy server</i> untuk membatasi hak akses user dan menggunakan squid untuk menyimpan <i>cache</i> di <i>proxy server</i> dapat mengurangi waktu akses terhadap situs..

Berdasarkan penelitian (Aminuddin & HS, 2018; Khasanah, 2017; Sirait & Siahaan, 2019; Subandri & Hanadwiputra, 2018; Suroso, 2019) penulis dapat menyimpulkan bahwa penerapan *proxy server squid* dapat meningkatkan performa terhadap situs yang diakses selain itu menggunakan *Wireshark* sebagai menangkap *traffic* yang lewat untuk menguji waktu yang dibutuhkan untuk mengakses sebuah *web* dan juga melakukan beberapa kondisi dalam pengujian. Hasilnya *respond time* terhadap situs yang diakses jauh lebih maksimal.

Metodologi Penelitian

Jenis penelitian ini adalah terapan, dengan menggunakan metode *NDLC (Network Development Life Cycle)* untuk melakukan perancangan tujuan perancangan NDLC ini memudahkan peneliti dalam mengklasifikasi alur yang harus dilalui, Menurut (Sujadi & Mutaqin, 2017) penggunaan NDLC sangat membantu dalam menggambarkan jaringan antar komputer lainnya. Berikut merupakan tahapan dari NDLC.

1. *Analysis*

Analysis yang dilakukan berupa bagaimana mengoptimalkan performa akses *internet* terhadap situs yang sering diakses oleh client sehingga dapat mengurangi waktu yang dibutuhkan untuk mengakses ke situs tersebut. Selain itu dilakukan juga 2 analisa kebutuhan berupa:

1. Hardware

Adapun dibutuhkan beberapa perangkat pendukung untuk melakukan perancangan berupa

Tabel 2. Kebutuhan Hardware

Hardware	
Jenis	Keterangan
Komputer/ proxy server	<i>Sistem Manufacture</i> : <i>Asus</i> <i>Processor</i> : Intel® Core™ i3-4030U <i>CPU</i> : @1.9GHz, <i>Memory</i> : 10.00 GB RAM <i>Harddisk SSD</i> : 500 GB
Router	<i>Router</i> : ZTE ZXHN F609 <i>Product Code</i> : GPON ONT Deskripsi Produk : 4 slot port LAN
Switch	<i>D-link 16 Port DGS 1016A</i>
Komputer /Client	<i>Sistem Manufacture</i> : <i>Asus</i> <i>Processor</i> : Intel® Core™ i3-2100 <i>CPU</i> : @3.1GHz, <i>Memory</i> : 4.00 GB RAM <i>Harddisk</i> : 500 GB

2. *Software*

Adapun dibutuhkan beberapa perangkat lunak yang dibutuhkan untuk melakukan perancangan berupa:

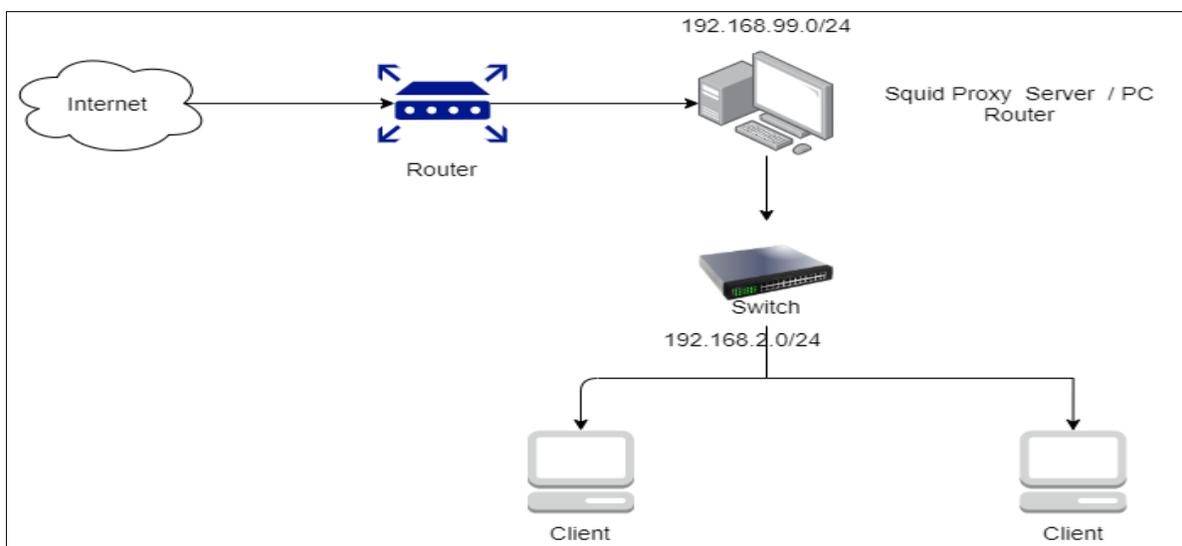
Tabel 3. Kebutuhan Software

Software	
<i>VMware</i>	Perangkat lunak yang digunakan untuk memvirtualisasi pada <i>OS</i> yang berbeda

<i>Winscp</i>	Perangkat lunak yang digunakan untuk meremote <i>SSH server</i> di IP <i>Ubuntu Server</i>
<i>Mozilla Fire Fox</i>	Perangkat lunak yang digunakan untuk melakukan ujicoba <i>Browsing</i> terhadap yang di implementasi
<i>Squid</i>	Perangkat lunak yang digunakan untuk meng <i>cache</i> dan menyimpan <i>SSL</i> yang di <i>request</i> oleh <i>client</i>
<i>Wireshark</i>	Perangkat lunak yang digunakan untuk mencapture traffic yang lewat dan menguji respons time terhadap situs yang di akses.

2. Design

Design yang dilakukan berupa merancang topologi yang akan dibangun.



Gambar 1. Topologi jaringan

3. Simulation

Simulation yang dilakukan berupa uji coba sebelum melakukan tahap implementasi. simulasi ini dilakukan dengan tujuan mengetahui apabila terjadinya *error* sehingga dapat di diagnosa *error* tersebut, jika tidak terjadi *error* maka implementansi dapat dilakukan.

4. Implementation

Implementation yang dilakukan berupa menimplementasikan dan konfigurasi pada *proxy server* menggunakan ubuntu server versi 14.04.06 di VMware dan melakukan instalasi serta konfigurasi di *squid* versi 3.5.12 setelah itu dilakukan uji coba *cache meta data* terhadap sebuah situs untuk mengetahui jika *squid* sudah berfungsi dengan baik.

5. Monitoring

Monitoring yang dilakukan berupa melakukan pengujian respons *time* menggunakan *wireshark* sebagai *tools* dan *brower mozilla firefox* sebagai *platfrom broswer* dengan dua kondisi seperti melakukan *capture traffic* terhadap situs yang sudah ditentukan ketika *proxy* belum diisi oleh *squid* dan melakukan *capture traffic* setelah diisi oleh *squid* terhadap situs yang dicoba. Berikut merupakan daftar situs yang akan dicoba respons time

No	URL	Tipe
1	www.Kontan.co.id	HTTPS
2	www.detik.com	HTTPS
3	www.Cnbcindonesia.com	HTTPS
4	www.amazon.com	HTTPS
5	www.kaskus.co.id	HTTPS

Tabel 4. Situs pengujian

6. *Management*

Management yang dilakukan berupa membandingkan pengujian respons *time* sebelum dan sesudah *proxy* di *cache* oleh *squid* dan melihat perbandingan perbedaan tersebut lalu disimpulkan.

Hasil dan Pembahasan

1. Konfigurasi *Proxy Server* ubuntu & *squid*

1. Tahapan awal merupakan instalasi ubuntu server dan melakukan konfigurasi *hostname*, *username* dan *password* supaya dapat melakukan *login* didalam ubuntu serta melakukan pengaturan zona waktu dan alokasi penyimpanan disk di ubuntu.



Gambar 1. Instalasi Ubuntu Server

2. Tahapan berikutnya mengetik *syntac ifconfig* untuk mengetahui *IP address* didalam ubuntu.Selanjutnya adalah melakukan instalasi dan konfigurasi pada *squid* sehingga dapat mencache situs yang diakses oleh client

```

root@ubuntu:/home/vincent# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:9c:2e:35
          inet addr:192.168.99.141  Bcast:192.168.99.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe9c:2e35/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:25 errors:0 dropped:0 overruns:0 frame:0
          TX packets:31 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2794 (2.7 KB)  TX bytes:3102 (3.1 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@ubuntu:/home/vincent#
root@ubuntu:/home/vincent# _

```

Gambar 2. Tampilan dalam ubuntu

3. Tahapan berikutnya melakukan *configure* pada squid tujuan konfigurasi tersebut untuk mengaktifkan fitur dari squid tersebut dan melakukan penempatan *file* pada *squid*.

```

root@ubuntu:/home/vincent/squid-3.5.12# ./configure --prefix=/usr --bindir=/usr/bin --sbindir=/usr/s
bin --libexecdir=/usr/lib/squid --sysconfdir=/etc/squid --localstatedir=/var --libdir=/usr/lib --wit
h-default-user=proxy --with-logdir=/var/log/squid --with-pidfile=/var/run/squid.pid --enable-ssl-crt
d --with-openssl

```

Gambar 3. Configure pada squid

4. Tahapan berikutnya melakukan konfigurasi di dalam squid.conf seperti *memasukan IP network* dari ubuntu dan *localhost* sehingga *squid* dapat berfungsi dan dilakukan penentuan *port* selain itu juga memberikan hak akses kepada *localhost* dan *localnet* dari ubuntu serta melakukan *replica SSL Certificate* sehingga *squid* dapat mencache *meta data* dari *web HTTPS..*

```

/etc/squid/squid.conf - root@192.168.99.141 - Editor - WinSCP
acl localnet src 10.0.0.0/8 # RFC1918 possible internal network
acl localnet src 172.16.0.0/12 # RFC1918 possible internal network
acl localnet src 192.168.0.0/16 # RFC1918 possible internal network
acl localnet src fc00::/7 # RFC 4193 local private network range
acl localnet src fe80::/10 # RFC 4291 link-local (directly plugged) machines

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

```

Gambar 4. Konfigurasi Squid.conf

```

http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports

http_access allow localhost manager
http_access deny manager

http_access allow localnet
http_access allow localhost
http_access deny all

https_port 3127 ssl-bump generate-host-certificates=on dynamic_cert_mem_cache_size=4MB cert=/etc/squid/ssl_cert/myCA.pem

cache_dir aufs /etc/squid/cache 10000 100 256

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

acl step1 at_step SslBump1
acl step2 at_step SslBump2
acl step3 at_step SslBump3

ssl_bump peek step1 all

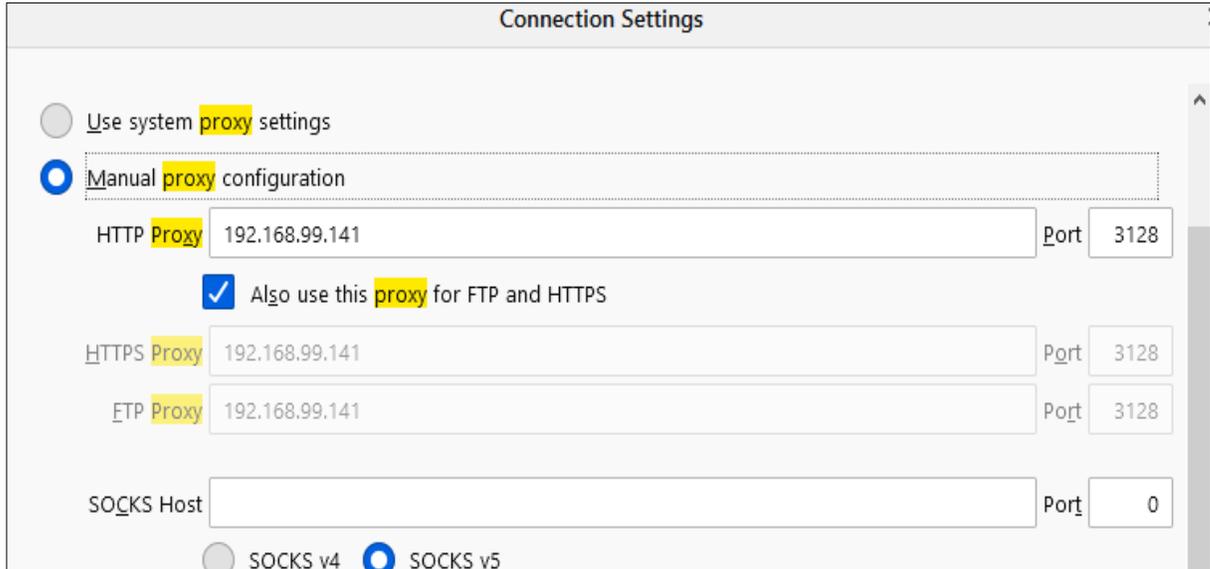
ssl_bump bump all]
# END

```

Gambar 5. Konfigurasi lanjutan Squid.conf

2. Pengujian *cache squid* & *respons time*

- Langkah pertama yaitu memasukan *Ip address proxy server* dan *port* kedalam *browser Mozilla Firefox*



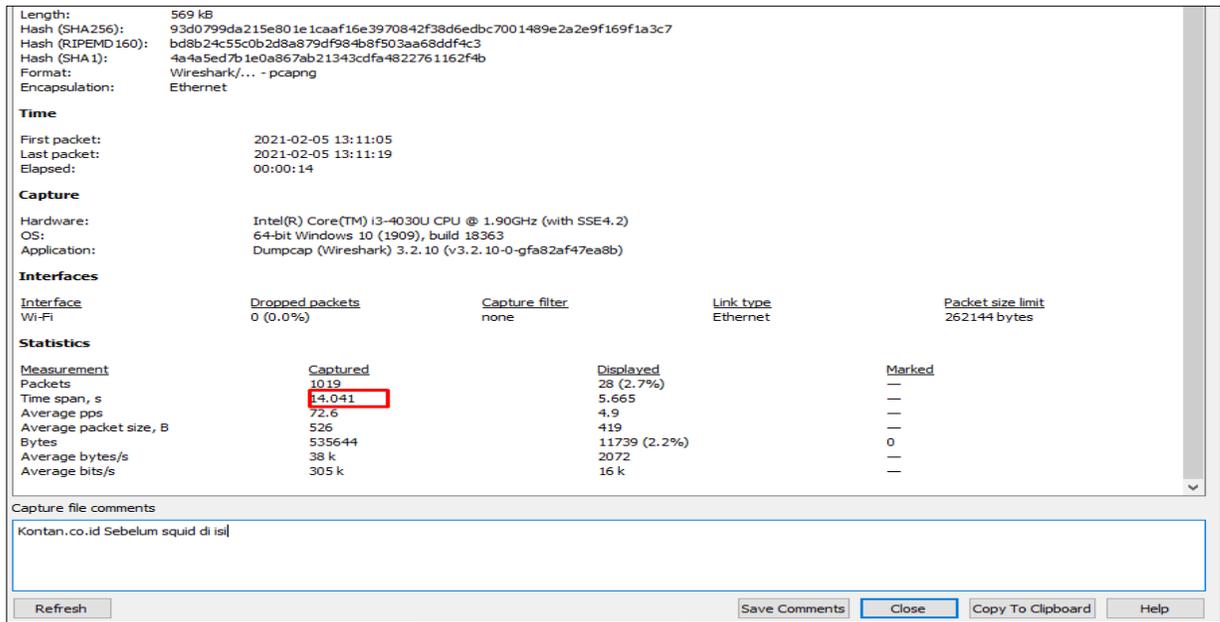
Gambar 6. Konfigurasi Web browser

- Langkah selanjutnya merupakan melakukan uji coba akses ke salah satu situs sehingga *Squid* dapat men*cache* situs tersebut



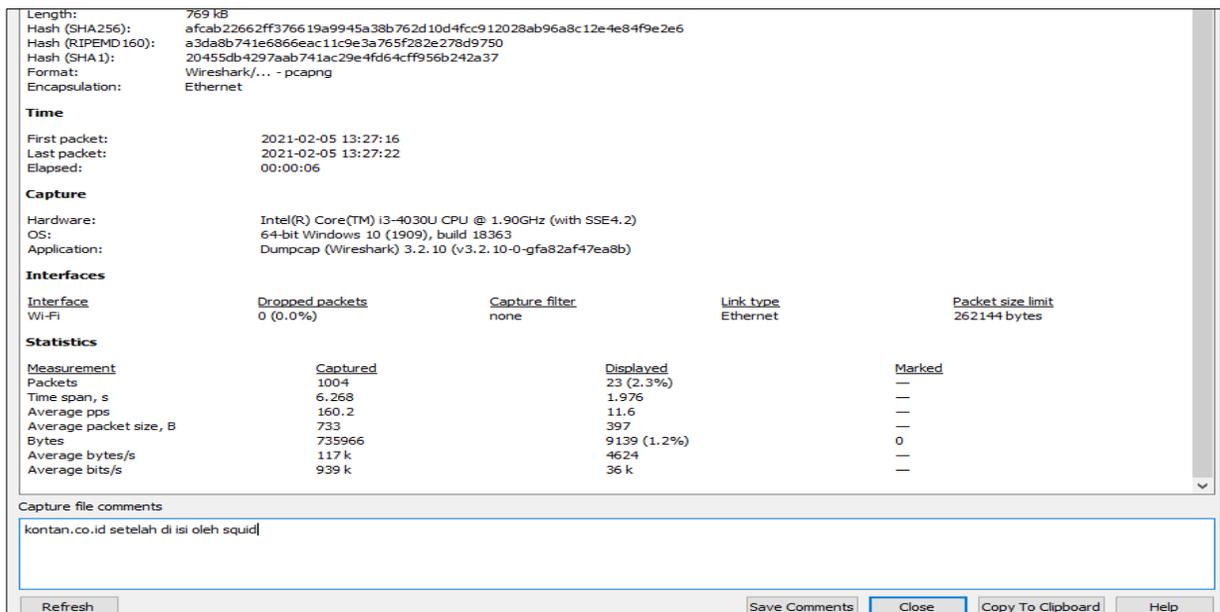
Gambar 7. Pengujian cache

- Langkah selanjutnya merupakan melakukan pengujian respons time ke situs *kontan.co.id* sebelum dan sesudah *proxy* terisi oleh *meta data* yang di *cache* oleh *squid* serta dicapture menggunakan *wireshark*.



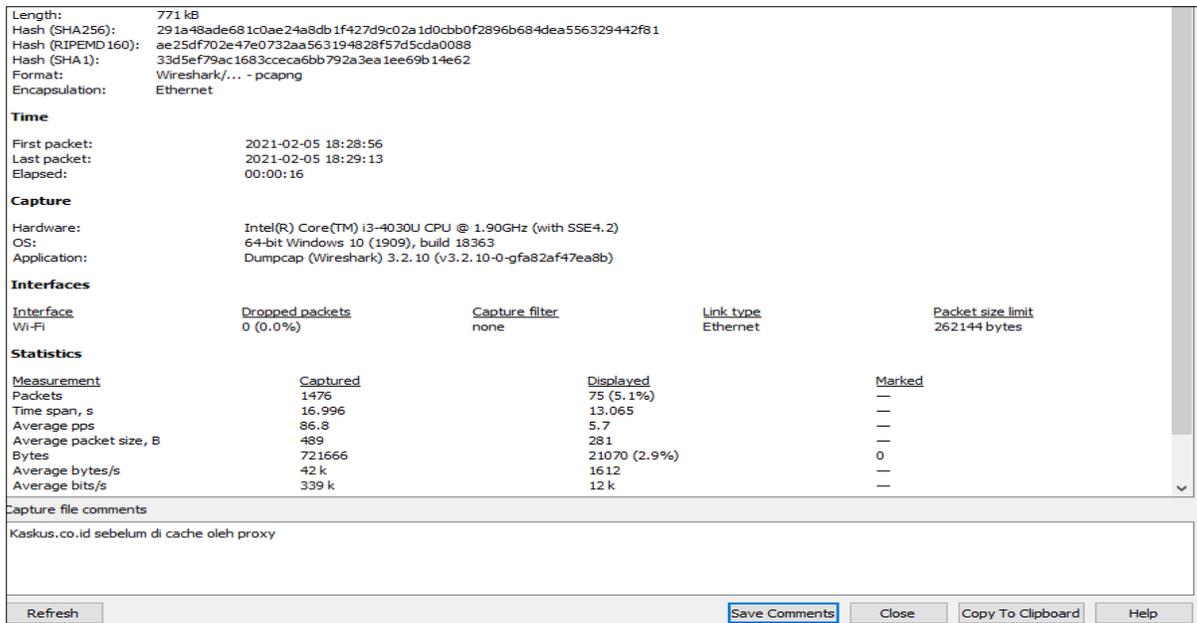
Gambar 8. Pengujian situs kontan sebelum dicache oleh proxy

- Pada Gambar 8. dilakukan pengujian menggunakan *wireshark* terhadap situs kontan sebelum dicache oleh *squid* . waktu yang dibutuhkan untuk mengakses ke kontan adalah 14.041 detik.



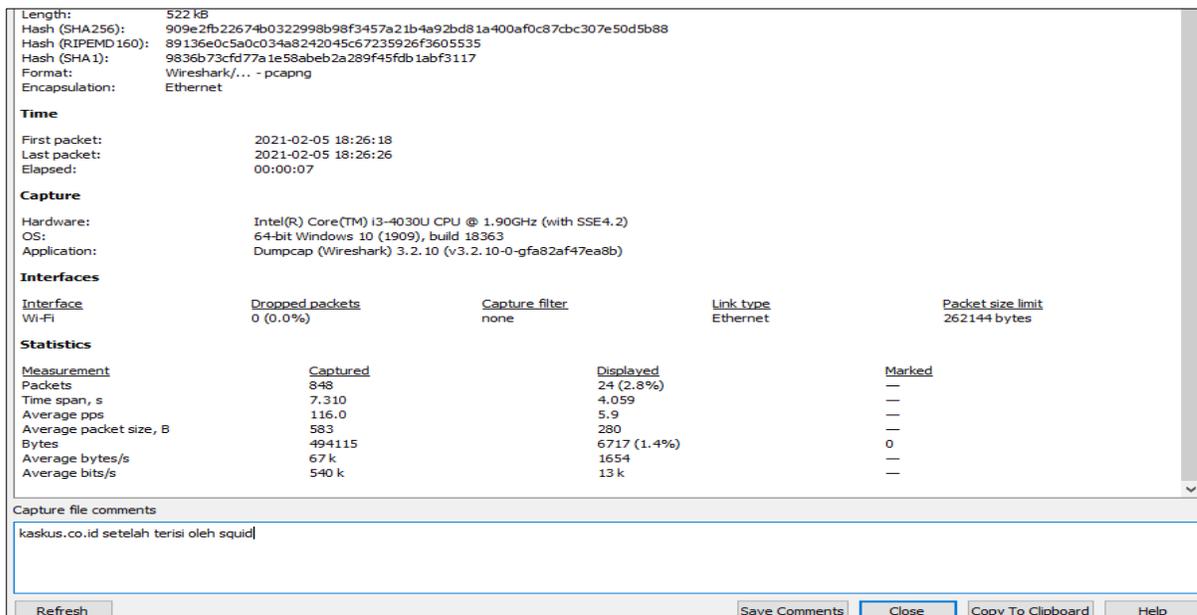
Gambar 9. Pengujian situs kontan setelah dicache oleh proxy

- Pada Gambar 9. Diatas dilakukan pengujian menggunakan *wireshark* terhadap situs kontan setelah dicache oleh *squid* waktu yang dibutuhkan untuk mengakses ke situs tersebut adalah 6.268 detik.
- Langkah selanjutnya merupakan pengujian repons *time* terhadap situs kaskus sebelum dan sesudah dicache oleh *proxy*.



Gambar 10. Pengujian situs kaskus sebelum dicache oleh proxy

7. Pada Gambar 10 dilakukan pengujian menggunakan *wireshark* terhadap situs kaskus sebelum dicache oleh *squid*. Waktu yang dibutuhkan untuk mengakses ke situs tersebut adalah 16.996 detik.



Gambar 11. Pengujian situs kaskus setelah dicache oleh proxy

8. Pada Gambar 11 diatas dilakukan pengujian menggunakan *wireshark* terhadap situs kaskus stelah dicache oleh *squid*. Waktu yang dibutuhkan untuk mengakses ke situs tersebut adalah 7.310 detik.
9. Langkah selanjutnya merupakan pengujian repons time terhadap situs detik sebelum dan sesudah di cache oleh proxy

Length: 2321 kB
 Hash (SHA256): b8ee80ca52e807a9671348421fc1592cb310adea53c4fb3b4a8af1b0d404b766
 Hash (RIPEMD 160): ac11fa822a9a71c22d9c80c251a3142665b4bdea
 Hash (SHA 1): 318f6fb199b81b4179ea54e7a5fca330932ae51
 Format: Wireshark/... - pcapng
 Encapsulation: Ethernet

Time
 First packet: 2021-02-05 19:15:03
 Last packet: 2021-02-05 19:15:18
 Elapsed: 00:00:14

Capture
 Hardware: Intel(R) Core(TM) i3-4030U CPU @ 1.90GHz (with SSE4.2)
 OS: 64-bit Windows 10 (1909), build 18363
 Application: Dumpcap (Wireshark) 3.2.10 (v3.2.10-0-gfa82af47ea8b)

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
Wi-Fi	0 (0.0%)	none	Ethernet	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	2620	16 (0.6%)	—
Time span, s	14.440	8.230	—
Average pps	184.3	1.9	—
Average packet size, B	836	502	—
Bytes	2231344	8028 (0.4%)	0
Average bytes/s	154 k	975	—
Average bits/s	1236 k	7803	—

Capture file comments
 detik.com sebelum di isi oleh proxy

Refresh Save Comments Close Copy To Clipboard Help

Gambar 12. Pengujian situs detik sebelum di cache oleh proxy

10. Pada Gambar 12 diatas dilakukan pengujian menggunakan *wireshark* terhadap situs detik sebelum di cache oleh *squid*. Waktu yang dibutuhkan untuk mengakses ke situs tersebut adalah 14.440 detik.

Length: 276 kB
 Hash (SHA256): 675de351e8d6b24de4a552abb2ed76c278c23c67456a4153434aff2beed16f85
 Hash (RIPEMD 160): 189e0b8efc400a9505b36e0f9c58421f2180016c
 Hash (SHA 1): 643b63e27428b3f512a86bec4d956883cde4a0b1
 Format: Wireshark/... - pcapng
 Encapsulation: Ethernet

Time
 First packet: 2021-02-05 19:13:16
 Last packet: 2021-02-05 19:13:21
 Elapsed: 00:00:05

Capture
 Hardware: Intel(R) Core(TM) i3-4030U CPU @ 1.90GHz (with SSE4.2)
 OS: 64-bit Windows 10 (1909), build 18363
 Application: Dumpcap (Wireshark) 3.2.10 (v3.2.10-0-gfa82af47ea8b)

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
Wi-Fi	0 (0.0%)	none	Ethernet	262144 bytes

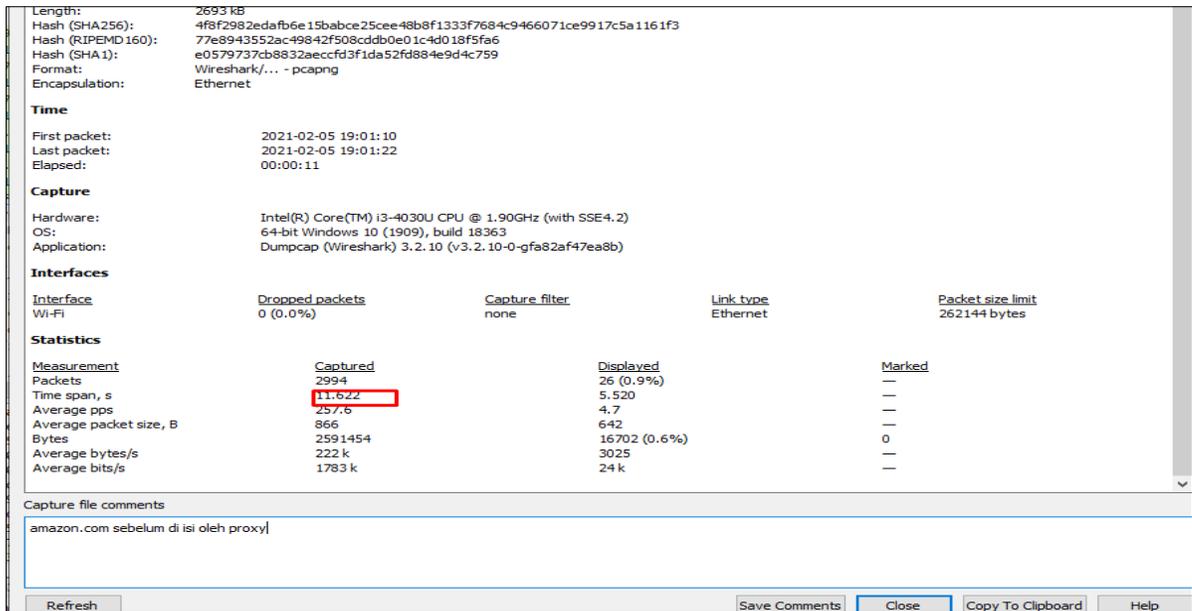
Statistics

Measurement	Captured	Displayed	Marked
Packets	484	6 (1.2%)	—
Time span, s	5.134	1.791	—
Average pps	94.3	3.4	—
Average packet size, B	538	612	—
Bytes	260406	3672 (1.4%)	0
Average bytes/s	50 k	2050	—
Average bits/s	405 k	16 k	—

Capture file comments
 detik.com sesudah proxy server terisi|

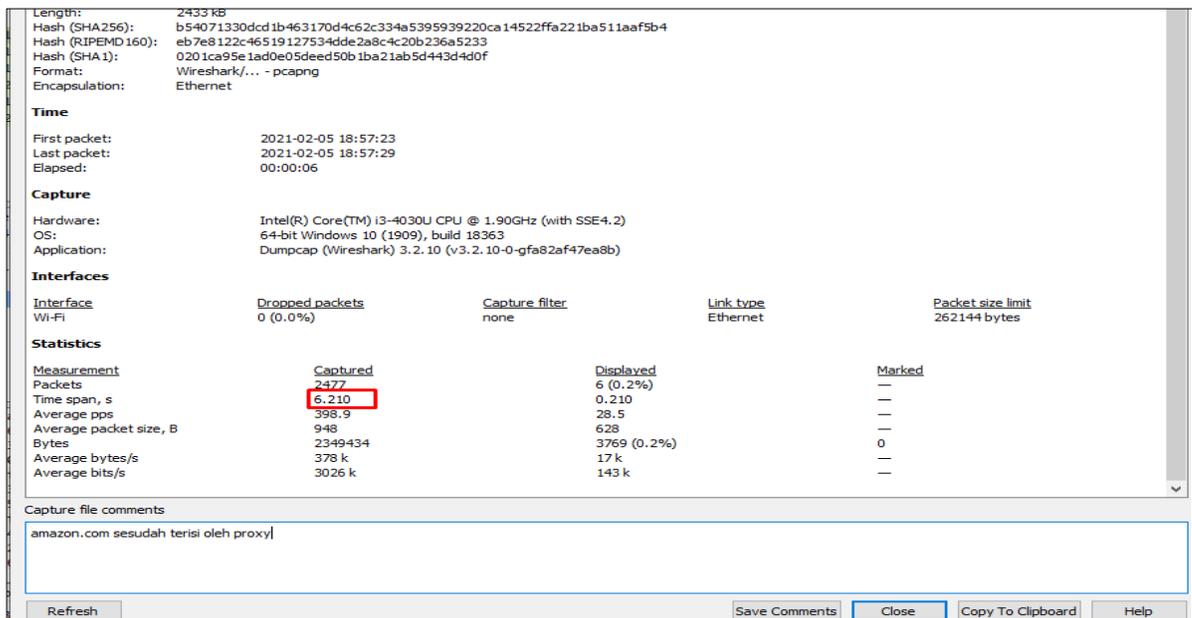
Gambar 13. Pengujian situs detik sesudah di cache oleh proxy

11. Pada Gambar 13 diatas dilakukan pengujian menggunakan *wireshark* terhadap situs detik setelah di cache oleh *squid*. Waktu yang dibutuhkan untuk mengakses ke situs tersebut adalah 5.134 detik.
12. Langkah selanjutnya merupakan pengujian respons *time* terhadap situs *amazon* sebelum dan sesudah dicache oleh *proxy*.



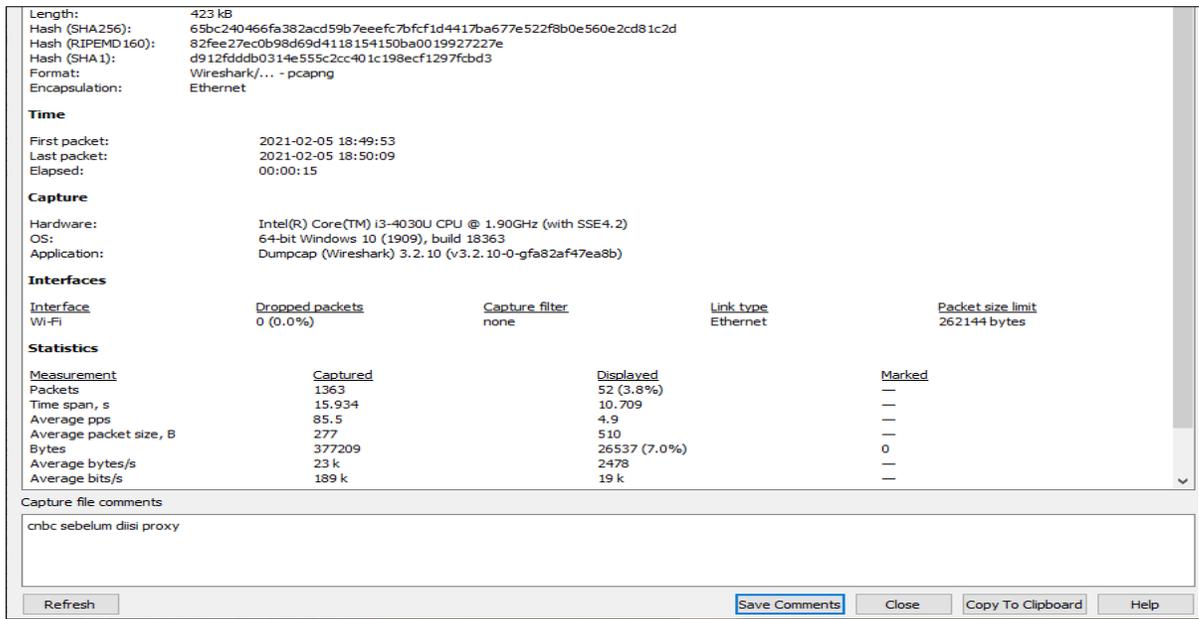
Gambar 14. Pengujian situs amazon sebelum di *cache* oleh *proxy*

13. Pada Gambar 14 diatas dilakukan pengujian menggunakan *wireshark* terhadap situs amazon sebelum di *cache* oleh *squid*. Waktu yang dibutuhkan untuk mengakses ke situs tersebut adalah 11.622 detik.



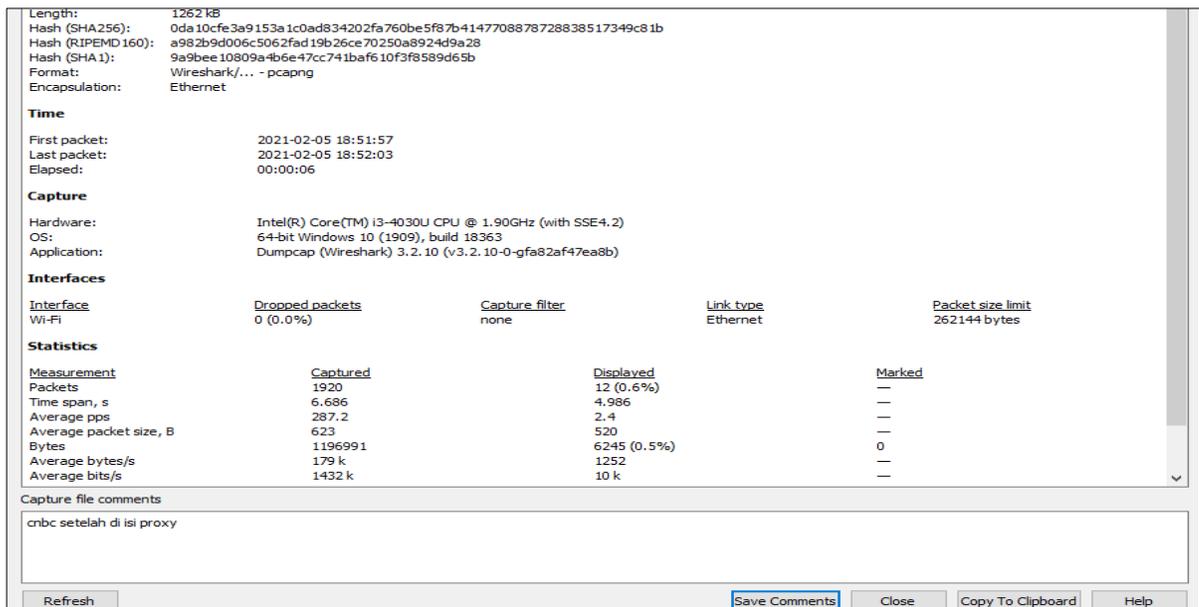
Gambar 15. Pengujian situs amazon setelah di *cache* oleh *proxy*

14. Pada Gambar 15 diatas dilakukan pengujian menggunakan *wireshark* terhadap situs *amazon* setelah dicache oleh *squid*. Waktu yang dibutuhkan untuk mengakses ke situs tersebut adalah 6.210
15. Langkah Selanjutnya merupakan pengujian situs terhadap *cnbcindonesia* sebelum sesudah di *cache* oleh *proxy*



Gambar 16. Pengujian situs cncbindonesia sebelum di *cache* oleh *proxy*

16. Pada Gambar 16 diatas dilakukan pengujian menggunakan *wireshark* terhadap situs cncbindonesia sebelum di *cache* oleh *squid*. Waktu yang dibutuhkan untuk mengakses situs tersebut adalah 15.934 detik.



Gambar 17. Pengujian situs cncbindonesia setelah di *cache* oleh *proxy*

17. Pada Gambar 17 diatas dilakukan pengujian menggunakan *wireshark* terhadap situs cncb indonesia sebelum di *cache* oleh *squid*. Waktu yang dibutuhkan untuk mengakses situs tersebut adalah 6.686 detik.

Kesimpulan

Tabel 5. Hasil perbandingan respons time

URL	Sebelum di <i>cache</i>	Sesudah di <i>cache</i>	Selisih	Efisiensi
Kontan.co.id	14.041	6.268	7.773	55.4%.
Kaskus.co.id	16.996	7.310	9.686	56.9%
Detik.com	14.440	5.134	9.306	64.4%
Amazon.com	11.622	6.210	5.412	46.5%
Cnbcindonesia.com	15.934	6.686	4.936	58.%
Rata-rata				56.45%

Berdasarkan Tabel 5.diatas rumus efisiensi dan selisih di dapat dari :

$$Ef = \frac{(\text{Sebelum di } \textit{cache} - \text{Sesudah di } \textit{cache})}{(\text{Sebelum di } \textit{cache})} \times 100\%$$

$$\text{Selisih} = (\text{Sebelum di } \textit{cache} - \text{Sesudah di } \textit{cache})$$

Pada Tabel 5 diatas terdapat selisih diantara *URL* yang sebelum di *cache* dan sesudah di *cache* , selisih tersebut menunjukkan angka efisiensi, Angka efisiensi menunjukkan 56.45 % artinya dapat mengurangi waktu respons *time* dengan rata-rata 56.45%. Disimpulkan bahwa pengujian repons *time* terhadap situs menggunakan *proxyserver squid* sebagai *cache meta* data dapat memberikan hasil respons waktu yang lebih baik dibandingkan tidak.

Daftar Pustaka

- Aminuddin, R., & HS, M. A. (2018). Perancangan Proxy Server Dengan Menggunakan Squid di CV.Nusantara Geotech Makassar. *Jurnal INSTEK (Informasi Sains Dan Teknologi)*, 2(2), 151–160.
- Khasanah, F. N. (2017). Performa Kecepatan Akses Internet Dengan Squid Proxy Server pada Ubuntu server 10.10. *INFORMATIC FOR EDUCATORS AND PROFFESIONALS: Journal of Informatics*, 2(1), 11–18.
- Kurniawan, H., Irawan, J. D., & Ariwibisono, F. (2020). Implementasi Squid Proxy Pada Mikrotrik dan Monitoring Traffic Jaringan Berbasis Website. *JATI(Jurnal Mahasiswa Teknik Informatika)*, 4(2), 136–143.
- Lubis, M. D. S., Hasannudin, D., Efendi, J., Wijono, L., & Sufiani, M. (2020). Membangun Router Pada Jaringan Komputer Menggunakan Ubuntu OS. *JTIK(Teknik Informatika Kaputama)*, 4(2), 111–125.
- R, A. I., & Marisa, F. (2017). Membangun Proxy Server Sebagai Penyaring Konten dan Manajemen Akses Jaringan internet Pada PT.Indomarine Surabaya. *Jurnal Teknologi Dan Manajemen Informatika*, 3(2).
- Sirait, P., & Siahaan, R. F. (2019). Analisis Kinerja Windows Server dan Linux Server Terhadap Respons Time Squid Proxy. *Journal Of Informatic Pelita Nusantara*, 4(2), 41–46.
- Subandri, & Hanadwiputra, S. (2018). Penerapan Teknologi Cache Server Berbasis Iot Dengan Raspberry Pi3 Menggunakan Metode Forward Chaining (Studi Kasus Smk Binakarya Mandiri 2 Kota Bekasi). *Kilat*, 7(2), 169–177.
- Sujadi, H., & Mutaqin, A. (2017). Rancang Bangun Arsitektur Jaringan Komputer Teknologi Metropolitan Area Network (MAN) Dengan Menggunakan Metode Network Development

Life Cycle (Ndlc) (Studi Kasus : Universitas Majalengka). *J-Ensitem*, 4(01).

Suroso, A. (2019). Optimasi Bandwidth Hits Local Up To Dengan Transparent Proxy 2.7 Lusca Menggunakan Metode Network Development Life Cycle. *Journal Informatics, Science & Technology*, 9(2), 26–34.

Wibawanto, A. (2018). Penggunaan Internet dalam Perpustakaan. *Pustakaloka*, 10(2), 191–203.