# BAB V
# PENGUJIAN DAN ANALISIS

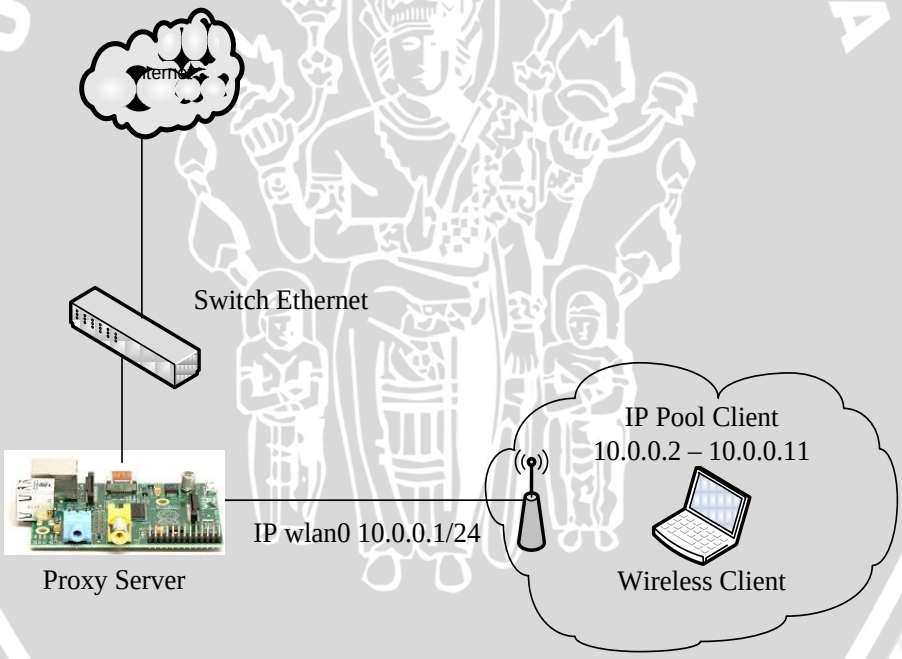
## 5.1 Pengujian

Pengujian Server proxy ini bertujuan untuk mengetahui *filtering* website/*content* kata domain dan *caching*. Pengujian dilakukan dengan 2 (dua) skenario, yaitu pengujian *filtering* dan pengujian *cache*.

### 5.1.1 Pengujian Filtering Website/Content Kata Domain

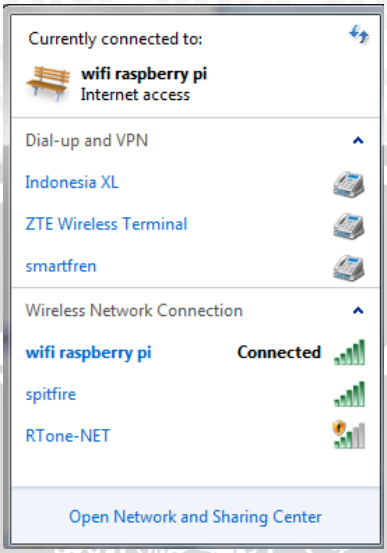
Langkah-langkah pada pengujian *filtering website/content* kata domain dilakukan sebagai berikut.

1. Topologi pengujian *filtering / content* kata domain.




Gambar 5.1 Topologi pengujian *filtering / content* kata domain

2. Client melakukan koneksi internet melalui *access point* "wifi raspberry pi" (Gambar 5.2).




Gambar 5.2 Wireless Connection

3. Terdapat 2 (dua) skenario pengaksesan *filtering website/content* kata domain oleh satu *client*. Skenario pertama *filtering website/content* kata domain adalah *client* mengakses website dan *content* kata domain yang termasuk dalam daftar *blacklist* pada satu jam kerja. Hasil *filtering* dapat dilihat pada tabel 5.1 dan tabel 5.2.

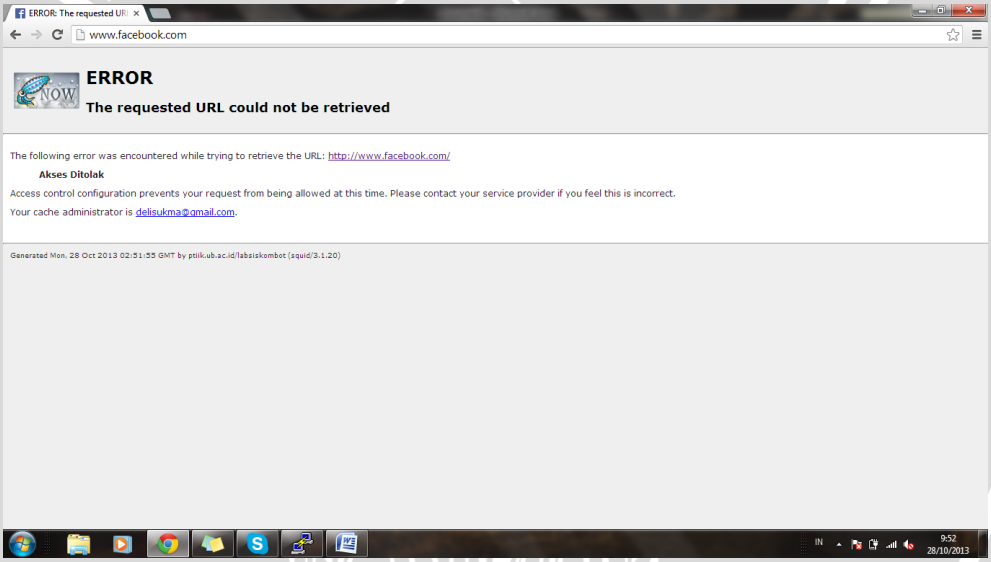
Tabel 5.1 Hasil *filtering website* pada satu jam kerja

| **Waktu** | **Kategori** | **Website** | **Status Filtering** |
|---|---|---|---|
| 08:00 – 11.59 | Social Network | facebook.com | Sukses |
| | | friendster.com | Sukses |
| | | instagram.com | Sukses |
| | | tumblr.com | Sukses |
| | | twitter.com | Gagal |
| | Streaming | youtube.com | Sukses |
| | | mivo.com | Sukses |
| | Portal Download | 4shared.com | Sukses |
| | | fileshare.com | Sukses |
| | | indowebster.com | Sukses |
| | | mediafire.com | Sukses |

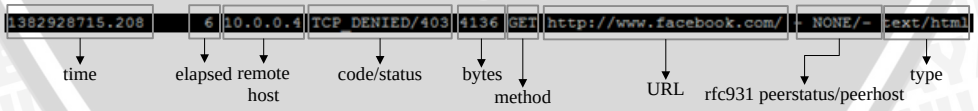
Tabel 5.2 Hasil *filtering* konten kata pada satu jam kerja

| **Waktu** | **Kata** | **Status Filtering** |
|---|---|---|
| **08:00 – 11:59** | Gambling | Sukses |
| | Judi | Sukses |
| | Kotor | Sukses |
| | Porno | Sukses |
| | Porn | Sukses |
| | Seks | Sukses |
| | Sex | Sukses |

Contoh tampilan hasil *filtering website* jam 9:52 untuk facebook.com dapat dilihat pada Gambar 5.2 dan access log facebook.com pada Gambar 5.3.




Gambar 5.3 Tampilan website facebook




Gambar 5.4 Access log facebook

Keterangan Gambar 5.4 sebagai berikut:

Time : Waktu ketika permintaan *completed.*
`1382928715.208` Format yang digunakan adalah "Unix time" dengan resolusi miliseconds.

| | | |
|---|---|---|
| Elapsed<br>`6` | : | Lama *request* dalam miliseconds. Lama *request* ini diperoleh antara waktu accept() dan close() dari soket *client.* |
| Remote host<br>`10.0.0.4` | : | IP address *client.* |
| Code/status<br>`TCP_DENIED/403` | : | Code merupakan *action* dari permintaan yaitu `TCP_DENIED` (Akses permintaan tersebut ditolak) karena facebook.com tidak dapat diakses pada saat jam kerja.<br>Status merupakan `HTTP_reply_code` diambil dari baris pertama dari `HTTP_reply_header`. Status `403` adalah kode HTTP untuk *forbidden*. |
| Bytes<br>`4136` | : | Size *request* dari client. |
| Method<br>`GET` | : | Metode HTTP *request*. |
| URL<br>`http://facebook.com/` | : | Alamat website yang diminta oleh client. |
| Rfc931<br>– | : | Hasil rfc931/ident lookup dari username client. Jika rfc931 / ident lookup dinonaktifkan (default: ‘`ident_lookup off`) maka client login sebagai –. |
| Peerstatus/peerhost<br>NONE/- | : | Peerstatus adalah deskripsi bagaimana dan dimana objek yang di*request* diambil. Peerstatus NONE menjelaskan bahwa objek yang |

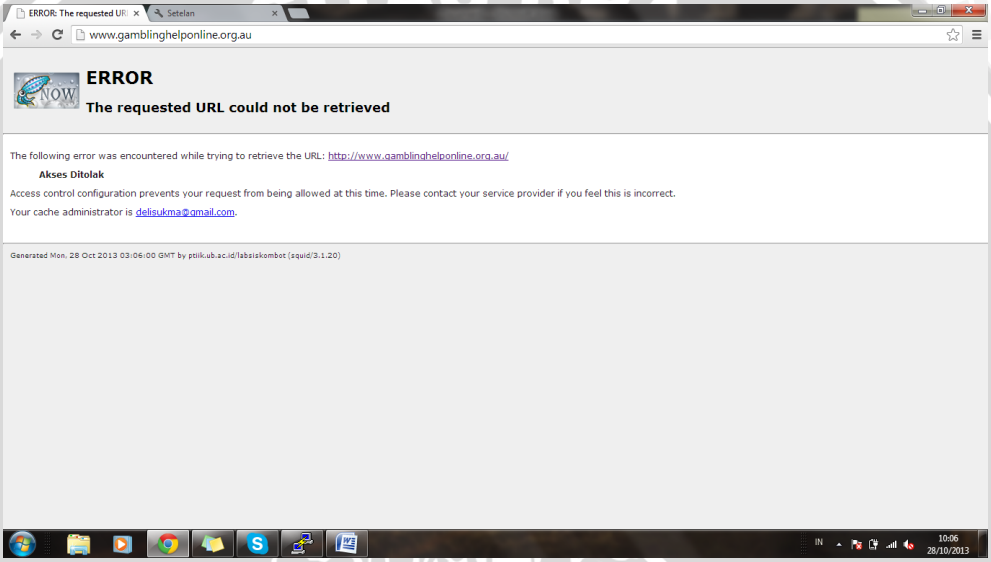
diminta tidak ada didalam cache.

Peerhost adalah *hostname* dari mesin dimana objek berada.

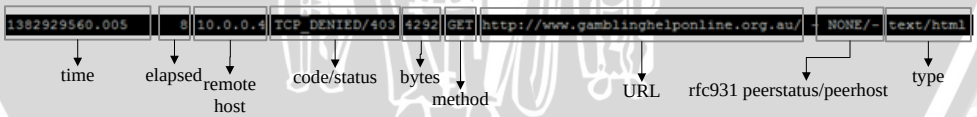
Type : *Content type* dari objek (dari `HTTP_reply_header`)

`text/html`

Contoh tampilan hasil *filtering* konten kata domain jam 10:06 untuk kata "gambling" dapat dilihat pada Gambar 5.5 dan access log *content* untuk kata "gambling" pada Gambar 5.6.




Gambar 5.5 Tampilan konten kata domain




Gambar 5.6 Access log konten kata domain

Keterangan Gambar 5.6 sebagai berikut:

Time : Waktu ketika permintaan *completed.* Format yang digunakan adalah "Unix time" dengan resolusi miliseconds.

`1382929560.005`

Elapsed : Lama *request* dalam miliseconds. Lama *request* ini diperoleh antara

`8`

| | | |
|---|---|---|
| | | waktu accept() dan close() dari soket *client.* |
| Remote host<br>10.0.0.4 | : | IP address *client.* |
| Code/status<br>TCP_DENIED/403 | : | Code merupakan *action* dari permintaan yaitu TCP_DENIED (Akses permintaan tersebut ditolak) karena URL teridentifikasi menggunakan kata terlarang.<br>Status merupakan HTTP_reply_code diambil dari baris pertama dari HTTP_reply_header. Status 403 adalah kode HTTP untuk forbidden. |
| Bytes<br>4292 | : | Size request dari client. |
| Method<br>GET | : | Metode HTTP *request*. |
| URL<br>http://www.gamblinghelponline.org.au | : | Alamat website yang diminta oleh client. |
| Rfc931<br>- | : | Hasil rfc931/ident lookup dari username client. Jika rfc931 / ident lookup dinonaktifkan (default: 'ident_lookup off) maka client login sebagai –. |
| Peerstatus/peerhost<br>NONE/- | : | Peerstatus adalah deskripsi bagaimana dan dimana objek yang di*request* diambil. Peerstatus NONE menjelaskan bahwa objek yang diminta tidak ada didalam cache. |

Peerhost adalah *hostname* dari mesin dimana objek berada.

Type : *Content type* dari objek (dari `HTTP_reply_header`)
`text/html`

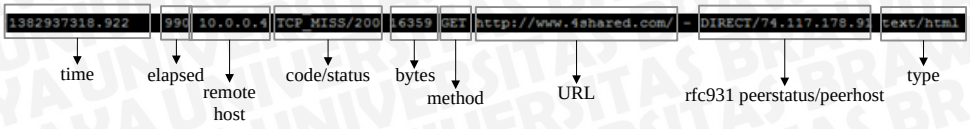
Skenario kedua *filtering website/content* kata domain adalah *client* mengakses satu website dan satu *content* kata yang termasuk dalam daftar *blacklist* pada jam kerja, jam istirahat dan jam pulang kerja. Hasil *filtering* dapat dilihat pada tabel 5.3.

Tabel 5.3 Hasil *filtering* satu website dan *content* kata domain.

| **Website / Kata** | **Waktu** | **Satus Filtering** |
|---|---|---|
| 4shared.com | 08:00 – 11:59 | Sukses |
| | 12:00 – 12:59 | Sukses |
| | 13:00 – 15:59 | Sukses |
| | 16:00 – 07:59 | Sukses |
| Kotor | 08:00 – 11:59 | Sukses |
| | 12:00 – 12:59 | Sukses |
| | 13:00 – 15:59 | Sukses |
| | 16:00 – 07:59 | Sukses |

Contoh tampilan hasil *filtering website* jam 12:16 untuk 4shared.com dapat dilihat pada Gambar 5.7 dan access log 4shared.com pada Gambar 5.8.


Gambar 5.7 Tampilan website 4shared.com



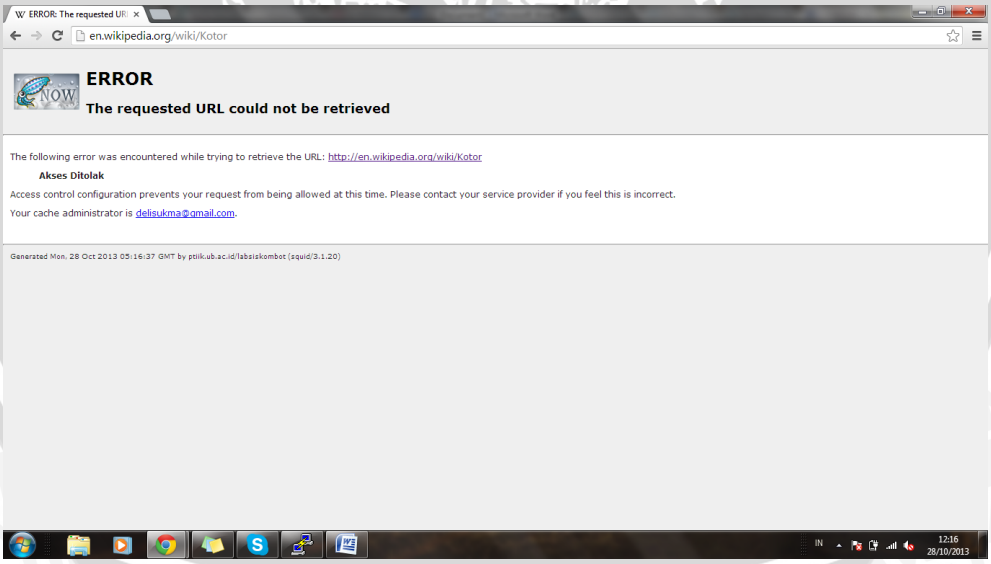

Gambar 5.8 Access log 4shared.com

Keterangan Gambar 5.8 sebagai berikut:

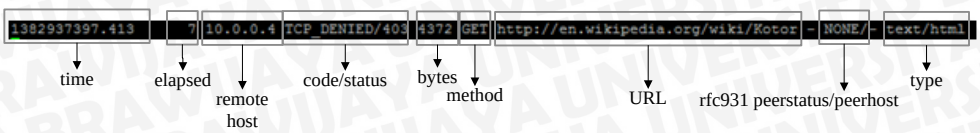
| | | |
|---|---|---|
| Time<br>`1382937318.992` | : | Waktu ketika permintaan *completed.* Format yang digunakan adalah "Unix time" dengan resolusi miliseconds. |
| Elapsed<br>`990` | : | Lama *request* dalam miliseconds. Lama *request* ini diperoleh antara waktu accept() dan close() dari soket *client.* |
| Remote host<br>`10.0.0.4` | : | IP address *client.* |
| Code/status<br>`TCP_MISS/200` | : | Code merupakan *action* dari permintaan yaitu `TCP_MISS` (objek yang diminta tidak ada didalam cache).<br>Status merupakan `HTTP_reply_code` diambil dari baris pertama dari `HTTP_reply_header`. Status `200` adalah kode HTTP untuk OK. |
| Bytes<br>`16359` | : | Size request dari client. |
| Method<br>`GET` | : | Metode HTTP *request*. |
| URL<br>`http://www.4shared.com` | : | Alamat website yang diminta oleh client. |
| Rfc931<br>`-` | : | Hasil rfc931/ident lookup dari username client. Jika rfc931 / ident lookup dinonaktifkan (default: |

| | | |
|---|---|---|
| | | ‘`ident_lookup off`) maka client login sebagai –. |
| Peerstatus/peerhost<br>`DIRECT/74.117.178.91` | : | Peerstatus adalah deskripsi bagaimana dan dimana objek yang di*request* diambil. Peerstatus `DIRECT` menjelaskan bahwa objek yang diminta berada di server asal.<br>Peerhost adalah *hostname* dari mesin dimana objek berada (`74.117.178.91`). |
| Type<br>`text/html` | : | *Content type* dari objek (dari `HTTP_reply_header`) |

Contoh tampilan hasil *filtering* konten kata domain jam 12:16 untuk kata “kotor” dapat dilihat pada Gambar 5.9 dan access log *content* untuk kata “gambling” pada Gambar 5.10.




Gambar 5.9 Tampilan content kata "kotor"




Gambar 5.10 Tampilan access log content kata “kotor”

Keterangan Gambar 5.10 sebagai berikut:

| | | |
|---|---|---|
| Time<br>`1382937397.413` | : | Waktu ketika permintaan *completed.* Format yang digunakan adalah "Unix time" dengan resolusi miliseconds. |
| Elapsed<br>`7` | : | Lama *request* dalam miliseconds. Lama *request* ini diperoleh antara waktu accept() dan close() dari soket *client.* |
| Remote host<br>`10.0.0.4` | : | IP address *client.* |
| Code/status<br>`TCP_DENIED/403` | : | Code merupakan *action* dari permintaan yaitu `TCP_DENIED` (Akses permintaan tersebut ditolak) karena URL teridentifikasi menggunakan kata terlarang.<br>Status merupakan `HTTP_reply_code` diambil dari baris pertama dari `HTTP_reply_header`. Status `403` adalah kode HTTP untuk forbidden. |
| Bytes<br>`4372` | : | Size request dari client. |
| Method<br>`GET` | : | Metode HTTP *request*. |
| URL<br>`http://en.wikipedia.org/wiki/Kotor` | : | Alamat website yang diminta oleh client. |
| Rfc931<br>- | : | Hasil rfc931/ident lookup dari username client. Jika rfc931 / ident lookup dinonaktifkan (default: `'ident_lookup off`) maka client |

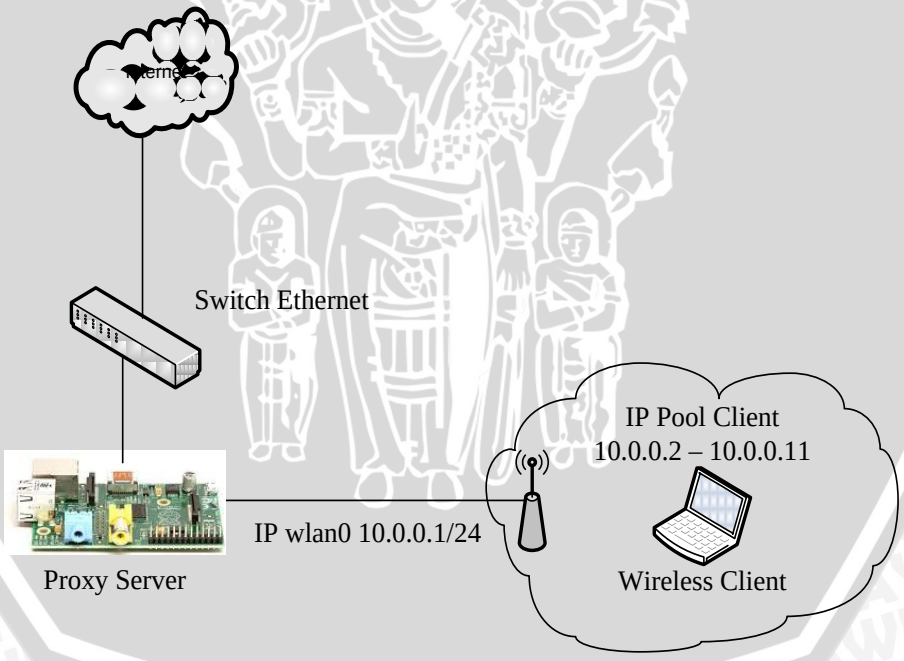
login sebagai –.

| | | |
|---|---|---|
| Peerstatus/peerhost<br>NONE/- | : | Peerstatus adalah deskripsi bagaimana dan dimana objek yang di*request* diambil. Peerstatus NONE menjelaskan bahwa objek yang diminta tidak ada didalam cache.<br>Peerhost adalah *hostname* dari mesin dimana objek berada. |
| Type<br>`text/html` | : | *Content type* dari objek (dari `HTTP_reply_header`) |

### 5.1.2 Pengujian Cache

Skenario pengujian cache sebagai berikut:

1. Topologi pengujian cache.




Gambar 5.11 Topologi pengujian *Cache*

2. Reset access log squid3 pada raspberry pi. Hal ini diperlukan untuk mengetahui *request hit rate*.
3. *Client* melakukan koneksi internet melalui *access point* "wifi raspberry pi".

4. *Client* mengakses tiga website yaitu ub.ac.id, ptiik.ub.ac.id dan teknik.ub.ac.id secara bersamaan. Pengaksesan ketiga website tersebut dilakukan 10 kali oleh satu *client*, berikut waktu pengaksesannya:
   - Pukul 9:00
   - Pukul 10:00
   - Pukul 11:00
   - Pukul 12:00
   - Pukul 13:00
   - Pukul 16:00
   - Pukul 17:00
   - Pukul 18:00
   - Pukul 19:00
   - Pukul 20:00
5. Untuk mengetahui apakah objek tersimpan dalam cache squid3 raspberry pi dapat dilakukan dengan cara menganalisa access log (Gambar 5.10 dan Gambar 5.11).


Gambar 5.12 Request TCP HIT

Keterangan Gambar 5.12 sebagai berikut:

| | | |
|---|---|---|
| Code<br>`TCP_HIT` | : | Code merupakan *action* dari permintaan yaitu `TCP_HIT` (salinan objek berada dalam cache). |
| Status<br>`200` | : | Status merupakan `HTTP_reply_code` diambil dari baris pertama dari `HTTP_reply_header`. Status `200` adalah kode HTTP untuk objek sukses di-*request*. |


Gambar 5.13 Request TCP IMS HIT

Keterangan Gambar 5.13 sebagai berikut:

| | | |
|---|---|---|
| Code<br>`TCP_IMS_HIT` | : | Code merupakan *action* dari permintaan yaitu `TCP_IMS_HIT` (salinan objek secara valid berada |

dalam cache (*fresh*)).

Status 304 : Status merupakan `HTTP_reply_code` diambil dari baris pertama dari `HTTP_reply_header`. Status `304` adalah kode HTTP untuk objek yang tidak termodifikasi.

6. Selain melihat access log, performansi cache untuk mengetahui *request hit rate* dapat dilihat dengan menggunakan calamaris (Gambar 5.14).

```
root@raspberrypi:~# cat access.log | calamaris -a --output-file access_stats.txt
```

Gambar 5.14 *Code* penggunaan calamaris

```
Cache statistics
------------------------------------------------------  ------------  ------
Total amount cached:                                        requests    1189
Request hit rate:                                                  %   55.66
Bandwidth savings:                                              Byte   1344K
Bandwidth savings in Percent (Byte hit rate):                      %    6.75
Average cached object size:                                     Byte    1157
Average direct object size:                                     Byte   20074
Average object size:                                            Byte    9544
------------------------------------------------------  ------------  ------

# Incoming TCP-requests by status
Status                 request  %       sec/req  Byte       %       kB/sec
---------------------- -------  ------  -------  ---------  ------  ------
HIT                       1189   55.66     0.01    1376003    6.75  188.92
 TCP_IMS_HIT              1117   52.29     0.01     436147    2.14   71.89
 TCP_HIT                    72    3.37     0.02     939856    4.61  772.58
MISS                       947   44.34     1.06   19010421   93.25   18.58
TCP_MISS                   940   44.01     1.06   18707185   92.20   18.46
TCP_REFRESH_UNMODIFIED       5    0.23     0.43     106060    0.52   48.56
TCP_REFRESH_MODIFIED         2    0.09     1.34     107176    0.53   38.92
ERROR                        0       0        0          0       0       0
---------------------- -------  ------  -------  ---------  ------  ------
Sum                       2136  100.00     0.47   20386424  100.00   19.78
```

Gambar 5.15 *Plain text output* calamaris

## 5.2 Analisis

Berdasarkan data dari hasil pengujian *filtering* konten didapatkan bahwa daftar website yang termasuk dalam blacklist file kenablok.txt tidak dapat diakses pada jam kerja. Hanya satu dari sebelas list website yang gagal terblok pada saat jam kerja. Website yang gagal terblok adalah twitter.com. Pada saat mengakses twitter.com, secara otomatis langsung *redirect* ke alamat website twitter dengan port https. Website tersebut menggunakan port https/443 (halaman terenkripsi). Website yang menggunakan halaman terenkripsi *history* aksesnya tidak muncul di access log squid3. Hasil pengujian *filtering* konten untuk daftar *content* kata domain yang termasuk dalam blacklist yang tidak boleh diakses terdapat dalam file fullblok.txt berhasil dilakukan *filtering*.

$$\text{Prosentase keberhasilan } \textit{filtering} \text{ berdasarkan tabel 5.1 dan tabel 5.2}$$
$$= \frac{\text{Total jumlah website} - \text{website yang gagal terblok}}{\text{Total jumlah website}} \text{ x } 100\,\%$$
$$= \frac{18-1}{18} \text{ x } 100\% = \mathbf{94.44\%}$$

Berdasarkan data dari hasil pengujian analisa cache menggunakan calamaris terhadap access log squid3 didapatkan bahwa permintaan hit rate total yang terdapat pada server proxy raspberry pi sebesar 1189 *request* (55.66%) dari 2136 *request* dengan ukuran objek cache rata-rata sebesar 1157 Byte dan dapat menghemat bandwidth sebesar 6.75%.

```
# Incoming TCP-requests by host
Host                    request    hit-%   sec/req       Byte    hit-%   kB/sec
----------------------  -------  ------  -------  ---------  ------  ------
10.0.0.3                   2136   55.66     0.47   20386424    6.75   19.78
----------------------  -------  ------  -------  ---------  ------  ------
Sum                        2136   55.66     0.47   20386424    6.75   19.78
```

Gambar 5.16 Data akses oleh *host*