

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini menjelaskan tentang pengujian terhadap hasil dari Implementasi Naive Byes Pada Proxy Server Untuk Klasifikasi Pengguna Internet. Tahap pengujian proses uji yang dilakukan adalah melalui cara pencocokan terhadap hasil perhitungan sistem. Tahap uji ini dilakukan dengan cara yang tidak efektif. Dimana pengujian dilakukan dengan meliputi pengujian seperti Analisis Uji Squid, Uji coba pencarian meta data tag, pengujian akurasi confusion matrik, dan Uji Akses secara manual .

6.1 Hasil dan Analisis Uji Squid

1.1.1 Skenario Pengujian

Pada sekanrio pengujian ini menggunakan percobaan yang dilakukan sebanyak 5 skenario. Dalam setiap skenario memiliki jumlah data Url yang berbeda yaitu sebanyak 10,20,40,80,150 data. Pengujian yang dilakukan adalah dengan pengukuran banyaknya jumlah URL yang dilakukan oleh squid server selama berjalan. Selama proses berjalan squid server akan terus menangkap aktifitas penggunaannya dengan perintah khusus pada terminal linux sampai dengan dihentikan proses tersebut. Hasil dari pengumpulan data link akan di samakan secara manual dengan wireshark dan riwayat browser. Nilai dari hasil pengujian berpengaruh pada besar dan kecilnya dari masing masing kebutuhan analisis.

1.1.2 Analisa Kebutuhan

Analisa yang digunakan dalam penelitian ini adalah dengan menggunakan software wireshark yang berfungsi untuk menganalisa kinerja jaringan, dan pada riwayat browser googlechrome dapat menyesuaikan kecocokan diantara squid server dan wireshark tools.

1.1.3 Pengumpulan dan Pengelompokan Data

Pada **tabel 6** percobaan yang dilakukan squid server adalah mengumpulkan data *URL* (Uniform Resource Locator) sehingga dapat dilakukan percobaan pengukuran jumlah kecocokan yang dilakukan oleh aplikasi wireshark dan histori browser client. Berikut adalah pengujian yang dihasilkan dengan perbandingan antara aplikasi wireshark dan histori browser .

Tabel6 Uji Coba dengan aplikasi Wireshark dan Browser

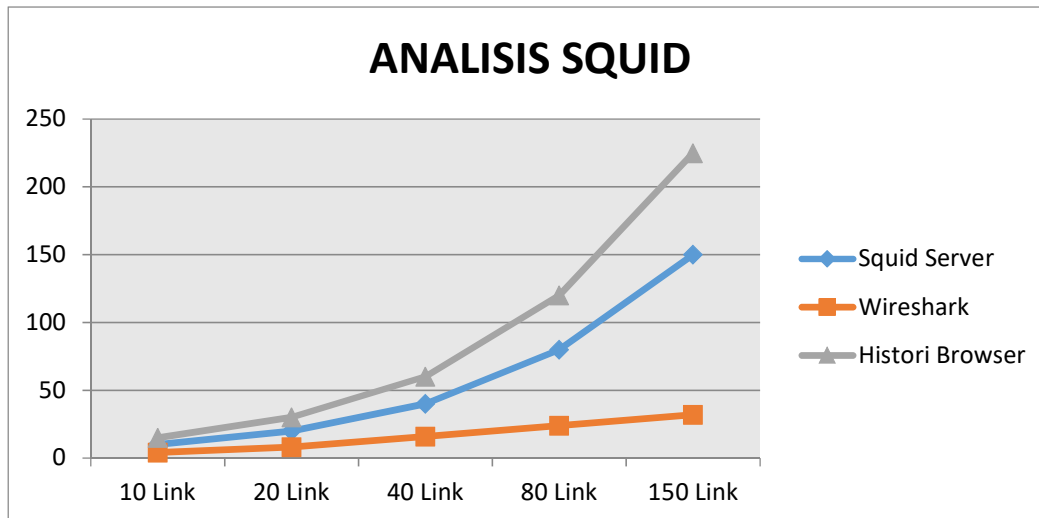
Sumber [Implementasi]

| | Squid Server | Wireshark | Riwayat Browser |
|------------|--------------|-----------|-----------------|
| Jumlah URL | 10 | 4 | 15 |
| | 20 | 8 | 30 |
| | 40 | 16 | 60 |
| | 80 | 24 | 120 |
| | 150 | 32 | 225 |

1.1.4 Analisa dan Penarikan Kesimpulan

Untuk seluruh skenario system squid server berhasil mengumpulkan data dengan total sebanyak 300URL pada wireshark hanya terkumpul sebanyak 84URL. Sedangkan pada Riwayat

Browser mampu mengumpulkan jumlah sebanyak 450 URL dengan jumlah skenario sebanyak 5 kali.



Gambar 6 Hasil Perbandingan Link

Sumber [Implementasi]

Dari **gambar 6** terjadi peningkatan yang berbeda dengan menggunakan pengumpulan data squid server, wireshark dan history browser dimana data yang paling banyak ditangkap adalah pada history browser yang diambil melalui aktifitas client/ pengguna dimana pada browser semua aktifitas akan di rekam dan di simpan kedalam riwayat akses pada komputer client. Pada hasil grafik **gambar 6** disimpulkan nilai pengujian terbaik adalah dengan menggunakan squid server karena data URL yang masuk tidak berulang ulang seperti pengujian History Browser. Hasil dari pola grafik tersebut dapat disimpulkan bahwa proses dalam pencarian informasi metadata deskripsi memiliki nilai integritas yang berbeda dari masing masing pengujian tersebut.

6.2 Hasil dan Analisis Uji Coba Pencarian Metadata Tag

6.2.1 Skenario Pengujian

Skenario pengujian ini adalah dengan menggunakan tools meta tag analyzer sebagai perbandingan untuk uji kecocokan dengan system yang berjalan. Skenario yang digunakan adalah dengan membaca setiap link URL yang tersimpan dan pada setiap URL akan dicari informasi mengenai metadata tag deskripsi riwayat.

6.2.2 Analisa kebutuhan

Analisa yang digunakan dalam penelitian ini adalah dengan menggunakan Tool meta tag analyzer, dan riwayat dimana dapat dilakukan pengukuran besar kecocokan data system server squid.

6.2.3 Pengumpulan dan Pengelompokan Data

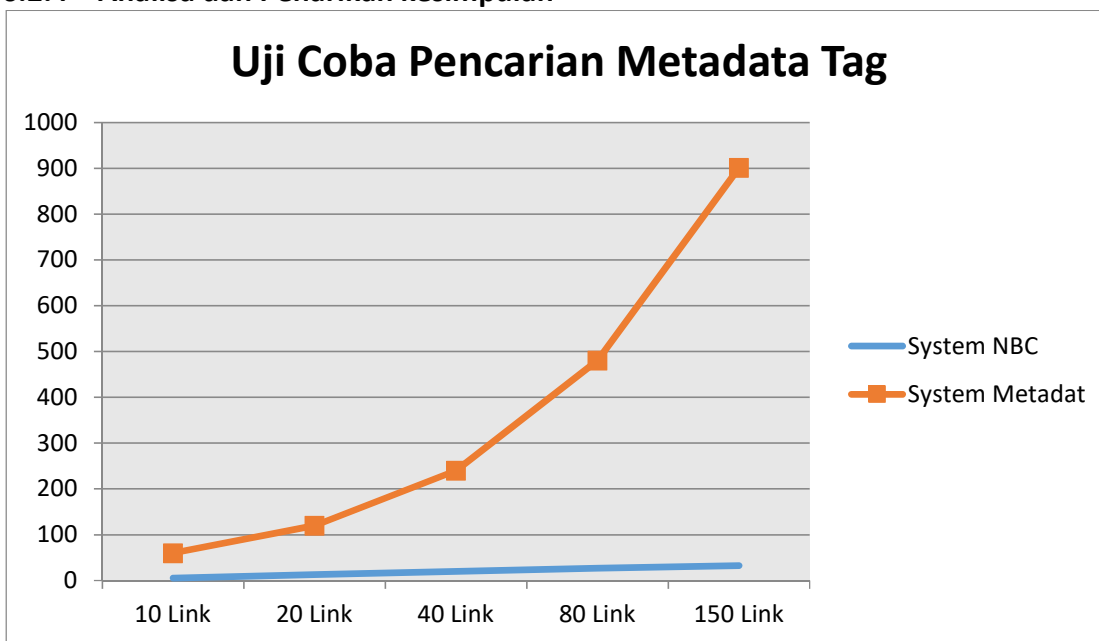
Pada pengujian pencarian metadata tag sama halnya dengan uji coba analisis squid yang terdiri dari 5 tahap skenario uji coba dengan mencari kecepatan untuk mendapatkan informasi metadata deskripsi dari setiap URL pada setiap skenario. Berikut merupakan data yang diperoleh dalam pengujian pencarian metadata tag.

Tabel 6.1 Uji Coba Waktu Pencarian Meta Data Tag
Sumber [Implementasi]

| Jumlah URL | System NBC | Metadata Tag Analyzer | Waktu Total System NBC | Waktu Total Metadata Tag Analyzer |
|------------|------------------|-----------------------|------------------------|-----------------------------------|
| 10 | 0,0551 / Metatag | 6.01 Detik/Metadata | 5.51 Detik | 60.1 detik |
| 20 | 0,1305 /metatag | 6.01 Detik/Metadata | 13.05 Detik | 120.2 detik |
| 40 | 0,2059 /metatag | 6.01 Detik/Metadata | 20.05 Detik | 240.4 detik |
| 80 | 0,2713 /metatag | 6.01 Detik/Metadata | 27.05 Detik | 480.8 detik |
| 150 | 0,3267 /metatag | 6.01 Detik/Metadata | 32.56 Detik | 901.5 detik |

Dari **tabel 6.1** diatas pada pencarian Metadata tag semua *URL* dapat menemukan isi dari informasi Metada tag yaitu deskripsi pada suatu website. pada perbandingan system NBC berhasil mencari informasi metada tag lebih cepat dibandingkan dengan metadata tag analyzer dikarenakan pada system NBC memiliki pencarian secara masal, berbeda dengan metadata tag analyzer yang mampu mencari secara manual.

6.2.4 Analisa dan Penarikan kesimpulan



Gambar 6.2 Hasil Waktu Pencarian Metadata Tag
Sumber [Implementasi]

Pada hasil **gambar 6.2** terlihat system NBC memiliki integritas yang berbeda dalam uji coba kecepatan terhadap system metadata tag dengan menggunakan tools meta tag analyzer. Melihat hasil grafik diatas nilai hasil pengujian bisa berubah karena beberapa faktor seperti : isi informasi metadata deskripsi yang tidak tersedia, dan

kecepatan permintaan dalam yang lambar karena pengaruh koneksi internet yang lemah/ lambat.

6.3 Hasil Pengujian Akurasi

6.3.1 Skenario Pengujian

Sesudah skenario dalam mencari nilai dari confussion matrix , dapat dilakukan penentuan nilai Precision, Reccal, Accuracy, dan Error untuk dapat mengetahui tingkat penentuan akurasi dalam penentuan kategori yang dilakukan oleh perhitungan naive bayes. Pengujian akurasi dilakukan sebanyak 5 skenario.

6.3.2 Analisa kebutuhan

Analisa yang digunakan dalam mengumpukan data adalah dengan menggunakan hasil dari tabel Confusion Matrix dengan posterior pada hasil proses naivebayes

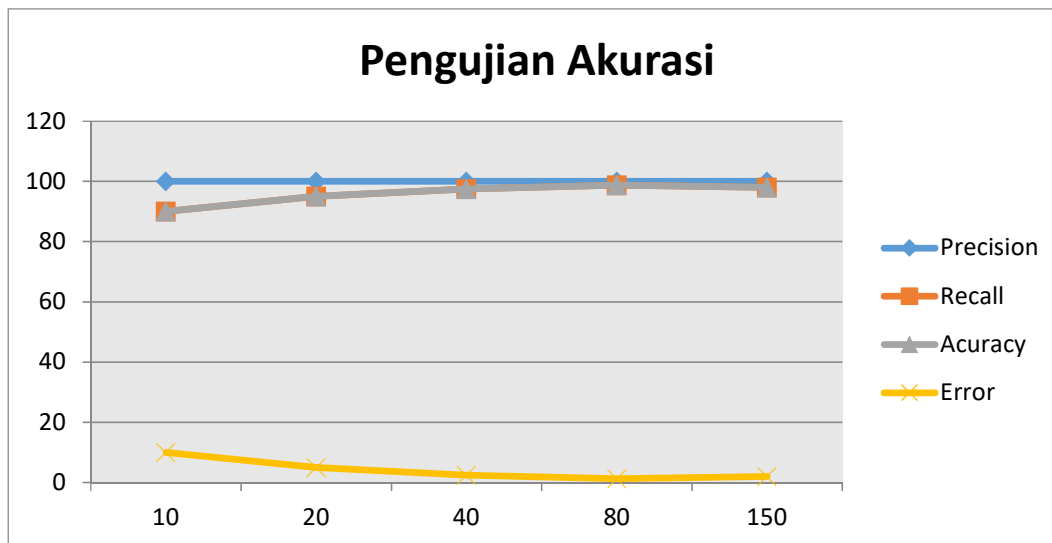
6.3.3 Pengumpulan dan Pengelompokan Data

Dari hasil Tabel Confusion Matrik yang menghasilkan nilai dari jumlah data benar yang benar dan salah serta data salah yang benar dan salah dengan skenario sebanyak 5 kali pada **tabel 6.2** dilakukan pengujian dari precision , reccal, accuracy dan error di bawah ini .

Tabel6.2 Tabel Perhitungan Precision, Reccal, Accuracy, dan Error
Sumber [Implementasi]

| | Precision | Recall |
|------------|--------------------|------------------|
| | Accuracy | Error |
| 10 | 100% Precision | 90% Recall |
| | 90% Accuracy | 10% Error |
| 20 | 100% Precision | 95% Recall |
| | 95% Accuracy | 5% Error |
| 40 | 100% Precision | 97,5% Recall |
| | 97,5% Accuracy | 2,5% Error |
| 80 | 100% Precision | 98,75% Recall |
| | 98,75% Accuracy | 1,25% Error |
| 150 | 100% Precision | 98% Recall |
| | 98% Precision | 2% Error |

6.3.4 Analisa dan Penarik Kesimpulan



Gambar 6.3 Hasil grafik Perbandingan Precision, Reccal, Accuracy, dan Error
Sumber [Implementasi]

Pada grafik **gambar 6.3** dilihat bahwa setiap skenario memiliki nilai Precision, Reccal, Accuracy, dan Error yang berbeda beda yang saling terhimpit dengan garis frekuensi kategori dan frekuensi error yang menghasilkan kategori sangat baik dalam penentuan akurasinya dan hal tersebut dapat diambil kesimpulan bahwa penentuan penetapan akurasi kategori naive bayes memenuhi syarat dari sudut pandang pengujian akurasi yang dihasilkan untuk penentuan system.

6.4 Uji Block

6.4.1 Skenario Pengujian

Dalam sekenario ini dibentuk dengan 5 buah skenario yang diuji dengan cara dilakukan uji akses Ping, Traceroute, Nslookup, Akses Browser, Proxy untuk dapat mengetahui apakah data URL dapat terhubung dengan server website.

6.4.2 Analisa Kebutuhan Hardware dan Software

Pengujian ini dilakukan dengan menggunakan aplikasi CMD/Commend Prompt untuk dapat dilakukan proses uji akses akses Ping, Traceroute, Nslookup. Sedangkan pada akses browser dan proxy menggunakan aplikasi google chrome , dan plugin dari google yaitu aplikasi proxy Browsec untuk dapat dilakukan uji coba akses secara langsung.

6.4.3 Pengumpulan dan Pengelompokan data

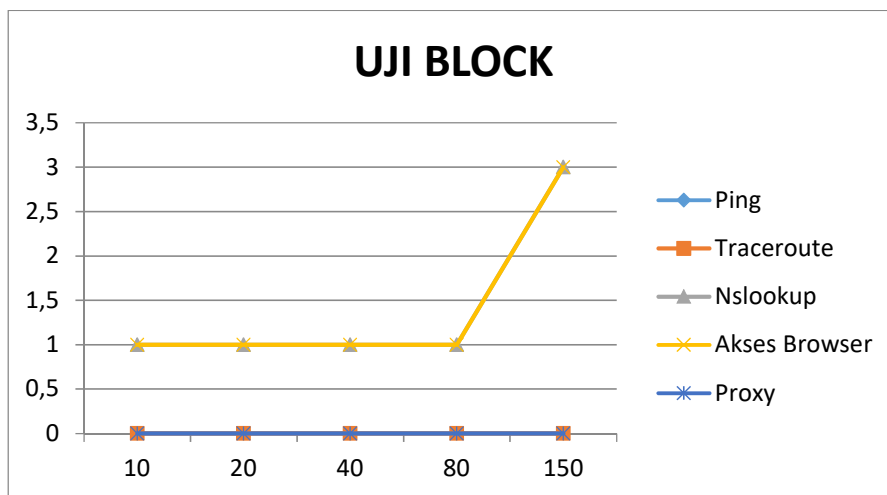
Dari hasil uji Block dengan cara Ping, Traceroute, Nslookup, Akses Browser, Proxy hasil yang ditemukan dengan **tabel 6.7** sesuai berikut.

Tabel 6.3 Uji Block Ping, Traceroute, Nslookup, Akses Browser, Proxy(1)

Sumber: [Implementasi]

| Jumlah URL | Parameter | | | | |
|------------|-----------------------|------------|----------|---------------|-------|
| | Ping | Traceroute | Nslookup | Akses Browser | Proxy |
| | Jumlah Link Terblokir | | | | |
| 10 | 0 | 0 | 1 | 1 | 0 |
| 20 | 0 | 0 | 1 | 1 | 0 |
| 40 | 0 | 0 | 1 | 1 | 0 |
| 80 | 0 | 0 | 1 | 1 | 0 |
| 150 | 0 | 0 | 3 | 3 | 0 |

6.4.4 Analisa dan Penarikan Kesimpulan



Gambar 6.4 Hasil grafik uji block

Sumber [Implementasi]

Jika dilihat pada **gambar 6.5** terhadap percobaan uji akses Ping, Traceroute, Nslookup, Akses Browser, Proxy dapat di tarik sebuah kesimpulan bahwa uji coba pemblokiran yang berhasil keseluruhannya pada Nslookup, Akses. Hal ini terbukti dari **tabel 6.7** yang memiliki hasil yang sama. Hal ini dapat membuktikan jenis link url yang diizinkan oleh system untuk tidak dapat melakukan akses permintaan akses content web.