

# PENERAPAN REDIS DALAM SISTEM E-OFFICE DESA SUTERA

Riyadi J. Iskandar<sup>1</sup>, Kartono<sup>2</sup>, Gandhi<sup>3</sup>

<sup>1,2,3</sup> Fakultas Teknologi Informasi, Universitas Widya Dharma, Pontianak  
e-mail: <sup>1</sup>riyadijiskandar@gmail.com, <sup>2</sup>kartono1102@gmail.com, <sup>3</sup>gandhybong@gmail.com

## Abstract

*Administration is something important in government system because if there is no administration then there will be a lot of thing happen like tracking and message will be uncoordinated and those data became hard to find. One important aspect in administration system is speed and that is the reason why author perform testing on administration system in sutera village named nusANTARA and implementing redis strategy in nusANTARA system so the system can be more efficient in handling user. The author uses experimental research designs to make research, while the data collection method used is the study of literature containing scientific books, theses, scientific papers, scientific journals, and other sources that related to the research topic. The system analysis technique used is object-oriented technique with Unified Modelling Language (UML) modeling. The design application uses the Laravel framework and MySQL database management system. This research produce a system that already implemented with redis strategy and compare result between old system and new system. The conclusion of this research is nusANTARA system cannot handle user more than 500 user when test take place which can be solve with implementing redis strategy that can reduce a down server time of nusANTARA.*

**Keywords:** *Laravel, Redis, Jmeter*

## Abstrak

Administrasi adalah hal yang penting dalam sistem pemerintahan karena jika tidak ada administrasi maka hal-hal seperti pencatatan dan surat-surat akan menjadi tidak teratur menyebabkan banyak data susah dicari. Pada sistem administrasi kecepatan juga menjadi sebuah hal yang harus di perhatikan yang daripada itu maka penulis melakukan pengujian pada sistem administrasi di desa sutera bernama nusANTARA dan mengimplementasikan strategi redis pada sistem nusANTARA agar sistem bisa lebih efisien dalam menangani user. Penulis menggunakan desain penelitian eksperimental untuk membuat penelitian, sedangkan metode pengumpulan data yang digunakan adalah studi literatur yang memuat buku-buku ilmiah, skripsi, karya ilmiah, jurnal ilmiah, serta sumber-sumber lain yang berkaitan dengan topik penelitian. Teknik analisis sistem yang digunakan adalah teknik berorientasi objek dengan permodelan Unified Modelling Language (UML). Aplikasi perancangan menggunakan framework Laravel, Redis dan sistem manajemen basis data MySQL. Penelitian ini menghasilkan sistem yang sudah terimplementasi dengan strategi redis dan hasil perbandingan antara sistem yang sudah terimplementasi redis dan sistem yang sedang berjalan. Kesimpulan dari penelitian ini adalah sistem nusANTARA tidak bisa handle user lebih dari 500 user pada saat percobaan berlangsung yang diatasi dengan mengimplementasikan strategi redis ini, yang bisa membuat down server time dari nusANTARA menjadi berkurang.

**Kata Kunci:** *Laravel, Redis, Jmeter*

## 1. PENDAHULUAN

Administrasi dalam arti sempit adalah kegiatan yang meliputi: catat-mencatat, surat-menyurat, pembukuan ringan, ketik-mengetik, agenda, dan sebagainya yang bersifat teknis ketatausahaan. Sistem administrasi juga di pakai di kantor – kantor pemerintahan di kota sampai dengan di desa.

Sistem administrasi di desa-desa banyak yang tergolong tidak rapi dan tidak tercatat secara lengkap. Sistem ini juga menyebabkan banyak kepala desa yang harus standby di kantor desa untuk melakukan tugas mereka. Hal ini juga membuat layanan desa menjadi lamban serta tidak efisien.

Di salah satu desa di Kalimantan Barat yaitu desa sutera CV. turbin dan Perum LKBN Antara sudah membangun sebuah aplikasi E-office yang bernama nusANTARA dimana penulis adalah bagian dari pengembang aplikasi. Aplikasi ini sudah dipakai pada desa sutera dari November 2019 dan sudah banyak membantu masyarakat dalam melakukan pembuatan surat administrasi.

Dalam kurun waktu pemakain aplikasi pada desa sutera ini penulis menemukan beberapa hal yang bisa dikembangkan dari aplikasi nusANTARA ini dimana adanya laporan dari user bahwa performa aplikasi yang melambat ketika diakses oleh banyak user atau yang disebut Peak hours yang membuat server bekerja tidak maksimal dari seharusnya.

Setelah menerima laporan tersebut penulis pun melakukan percobaan dengan mengecek maksimal koneksi dengan aplikasi JMeter yang menunjukkan bahwa user yang dapat di handle oleh server tidak bisa melampaui angka 149 user sekaligus.

## 2. METODOLOGI PENELITIAN

### 2.1 Rancangan Penelitian, Teknik Pengumpulan Data, Teknik Analisis Sistem, Teknik Perancangan Aplikasi.

#### 2.1.1 Rancangan Penelitian

Penulis menggunakan desain penelitian Eksperimental yaitu melakukan percobaan dan pengujian dengan cara mempelajari literatur-literatur yang berhubungan dengan materi perancangan aplikasi.

#### 2.1.2 Metode Pengumpulan Data

Teknik pengumpulan data yang digunakan penulis adalah dengan mencari literatur atau sumber pustaka yang berkaitan dengan perangkat lunak yang dibangun dan membantu mempertegas teori-teori yang ada serta memperoleh data yang sesungguhnya. Literatur dapat berupa jurnal ilmiah, buku, *e-book*, karya ilmiah, skripsi yang berkaitan dengan topik penelitian.

#### 2.1.3 Teknik Analisis Sistem

Teknik analisis sistem yang digunakan penulis adalah teknik berorientasi objek. Alat pemodelan sistem yang digunakan adalah *Unified Modeling Language* (UML), yang berperan untuk membantu memvisualisasikan prosedur dan aliran data yang terdapat pada perancangan dan implementasi strategi redis dalam sistem e-office.

#### 2.1.4 Teknik Perancangan Aplikasi

Teknik perancangan aplikasi yang digunakan penulis dalam penelitian ini adalah menggunakan bahasa pemrograman web sebagai bahasa pemrograman utama, PHP sebagai *backend*, dan MySQL untuk manajemen basis data.

### 2.2 Landasan Teori

#### 2.2.1 Redis

*Redis is an in-memory remote database that offers high performance, replication, and a unique data model to produce a platform for solving problems.* <sup>[1]</sup> *Redis is a data structure server with an in-memory dataset for speed. It is called a data structure server and not simply a key value store because Redis implements data structures allowing keys to contain binary safe strings, hashes, sets and sorted sets, as well as lists. This combination of flexibility and speed makes Redis the ideal tool for many applications.* <sup>[2]</sup>

#### 2.2.2 Web Server

*Web server* adalah *software* yang terletak di *server* untuk menerima *request* berupa halaman *web* dari klien (*browser*) dan mengirim respon berupa halaman *website*. <sup>[3]</sup> *Web server* merupakan *software* yang memberikan layanan data, berfungsi menerima permintaan HTTP atau HTTPS dari client yang dikenal dengan browser web dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML. <sup>[4]</sup>

#### 2.2.3 Model View Controller (MVC)

*Model-View-Controller* atau disingkat MVC adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan antara data (Model) dari tampilan (View) dan cara bagaimana memprosesnya (Controller). <sup>[5]</sup> MVC adalah metode pembuatan *software* yang memisahkan antara alur aplikasi dengan tampilan. Intinya MVC meminimalisir *coding* PHP pada halaman *web* karena telah dipisahkan menjadi Controller dan View. <sup>[6]</sup>

#### 2.2.4 GIT

*Git* is an open-source version control system known for its speed, stability, and distributed *collaboration* model. <sup>[7]</sup> *Git* is a version control tool. *Version control* is a useful tool to manage a set of files to track changes over a period of time. It allows users to compare file changes, restore prior versions, merge conflicting versions together, and more. <sup>[8]</sup>

#### 2.2.5 CSS

*Cascading Style Sheet* atau lebih singkatnya CSS merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. <sup>[9]</sup> *CSS (Cascading Style Sheet)* is responsible for describing to your browser how the HTML elements should be displayed. *CSS* defines how everything appears. It defines colours, sizes, positions and much more. *CSS* can be very powerful and can change the whole look of a website in just one line of code. <sup>[10]</sup>

#### 2.2.6 JavaScript

JavaScript adalah bahasa *script client side* yang populer di Internet dan dapat bekerja di sebagian besar *web browser* populer seperti Internet Explorer (IE), Mozilla Firefox, Netscape dan Opera. <sup>[11]</sup> Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap bahasa HTML dengan mengizinkan pengeksekusian perintah-perintah di sisi *user*, yang artinya di sisi browser bukan di sisi server web. <sup>[12]</sup>

#### 2.2.7 Hypertext Preprocessor (PHP)

*Hypertext Preprocessor* atau lebih singkatnya PHP adalah Bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML. Dengan PHP, kita dapat membuat *website* menjadi dinamis. <sup>[13]</sup> PHP (*Hypertext Preprocessing*) merupakan bahasa pemrograman yang biasa digunakan untuk membuat halaman HTML. <sup>[14]</sup>

2.2.8 Laravel

Laravel merupakan web *application framework* berbasis PHP yang *open source*, menggunakan konsep Model-View-Controller (MVC).<sup>[15]</sup> *Laravel is rapid application development framework. That means it focuses on a shallow (easy) learning curve and on minimizing the steps between starting a new app and publishing.*<sup>[16]</sup>

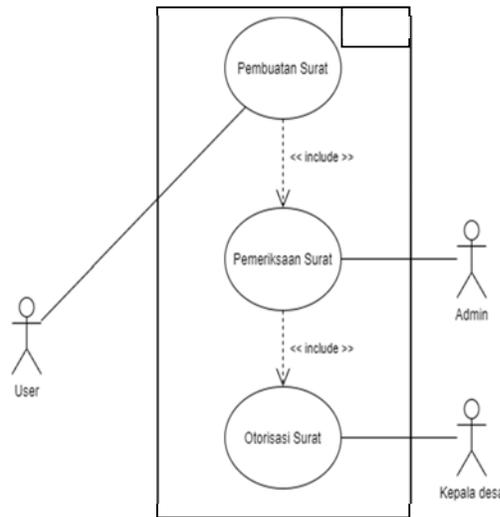
2.2.9 Unified Modeling Language (UML)

Unified Model Language (UML) adalah sebuah ‘bahasa’ yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak.<sup>[17]</sup> UML (*Unified Modeling Language*) adalah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berbasis objek.<sup>[18]</sup>

**3. HASIL DAN PEMBAHASAN**

3.1 Perancangan Unified Modeling Language (UML)

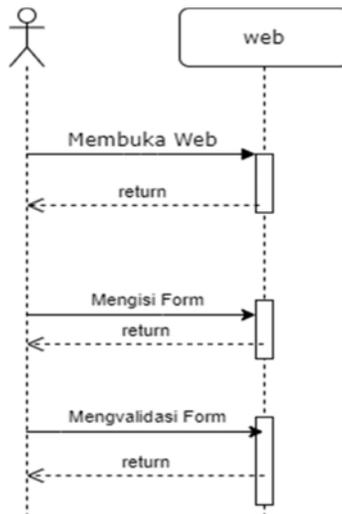
3.1.1 Use Case Diagram



Gambar 1. Use Case Diagram

Diagram use case tersebut memiliki tiga aktor yaitu user, admin, kepala desa. Saat hendak melakukan pembuatan surat user harus membuat surat lewat website dan submit form, yang setelah itu user akan melakukan validasi data saat akan melakukan pencetakan surat kepada admin. Jika data sudah valid maka admin akan melakukan pencetakan surat dan meminta tanda tangan pada kepala desa.

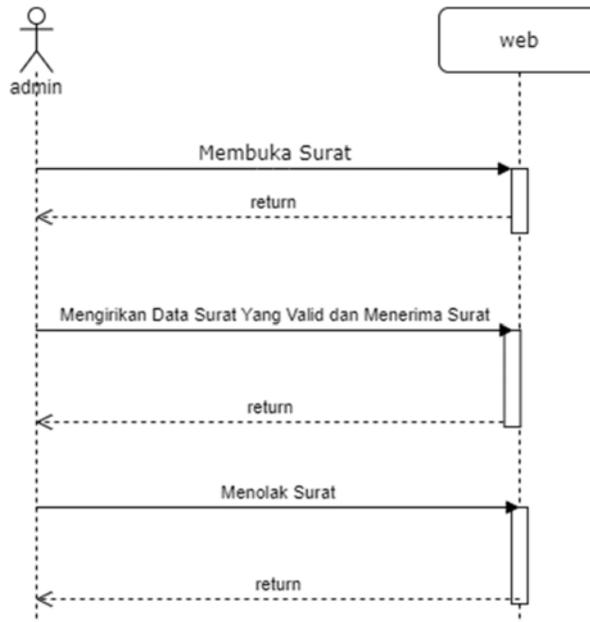
3.1.2 Diagram Sequence Pembuatan Surat



Gambar 2. Diagram Sequence Pembuatan Surat

Pada gambar 2 adalah sequence diagram pembuatan surat. Pada diagram tersebut, user akan membuka sistem setelah itu user harus memilih surat apa yang akan dibuat jika sudah memilih, sistem akan mengarahkan user ke halaman dimana form surat berada. User harus mengisi form yang ada dan sistem akan melakukan validasi dari data yang masuk jika data yang masuk sudah valid maka data akan dimasukan ke dalam database. Tetapi jika data tidak valid maka data akan dikembalikan kepada user untuk diperbaiki

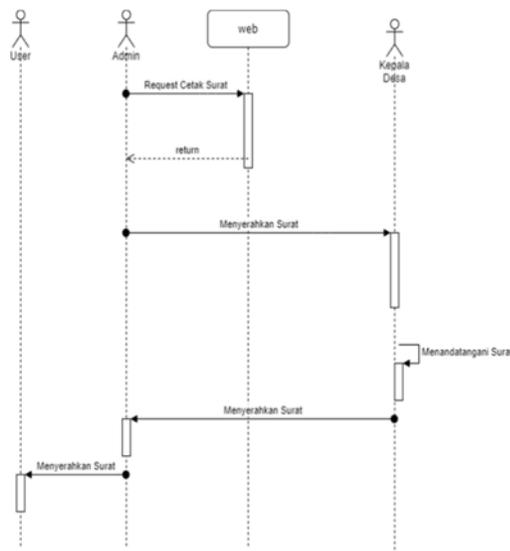
3.1.3 Diagram Sequence Pemeriksaan Surat



Gambar 3. Diagram Sequence Pemeriksaan Surat

Pada gambar 3 adalah diagram sequence pembuatan surat. Pada diagram tersebut, user admin akan membuka sistem, setelah membuka sistem user admin akan melakukan pengecekan pada data surat yang telah diterima. Antara lain data diri pendaftar serta poin-poin yang diperlukan untuk membuat surat yang dimaksud jika ada bagian surat yang salah dan dapat diperbaiki, user admin akan memperbaiki data surat tersebut dengan data yang benar setelah itu data akan disubmit ke server dan dimasukan ke dalam database. Tetapi jika data tidak bisa di perbaiki atau data diri orang yang bersangkutan tidak dapat menunjukkan kepentingan atau kebenaran akan pembuatan surat maka surat akan ditolak dengan cara user admin menekan tolak pada form surat yang akan menandai surat sebagai surat yang ditolak pada database.

3.1.4 Diagram Sequence Otorisasi Surat



Gambar 4. Diagram Sequence Otorisasi Surat

Pada gambar 4 adalah diagram sequence otorisasi surat. Pada diagram tersebut, admin akan mencetak surat yang sudah dipilih dan menyerahkan surat kepada kepala desa ataupun kepada pejabat yang bersangkutan jika kepala desa tidak ada ditempat untuk ditanda tangan, setelah tanda tangan kepala desa akan menyerahkan surat ke admin dan setelah itu surat akan diberikan kepada user yang bersangkutan.

3.2 Ringkasan hasil request

3.2.1 Ringkasan Request 100 user



Gambar 5. Ringkasan Request

Gambar 5 menjelaskan beberapa poin yang dapat dilihat seperti apdex(Application Performance Index), apdex adalah sebuah standard terbuka untuk mengukur performa perangkat lunak dalam komputasi yang tujuannya adalah mengubah pengukuran menjadi pengetahuan yang dalam tentang kepuasan pengguna dengan cara menggunakan teknik penganalisaan yang seragam, apdex memiliki rentan nilai dari 0 sampai 1, 0 adalah tidak puas dan 1 adalah sangat puas. Pada tabel terdapat dua value yaitu toleration threshold dan frustration threshold yang berfungsi sebagai batasan untuk mendapatkan satisfied count dan tolerating count dimana hal tersebut digunakan untuk mendapatkan apdex. ringkasan dari uji coba yang sudah dilakukan dengan 6 halaman berbeda yang menunjukkan 100 persen keberhasilan dari 100 user yang mengakses halaman tersebut

3.2.2 Statistik Testing 100 user

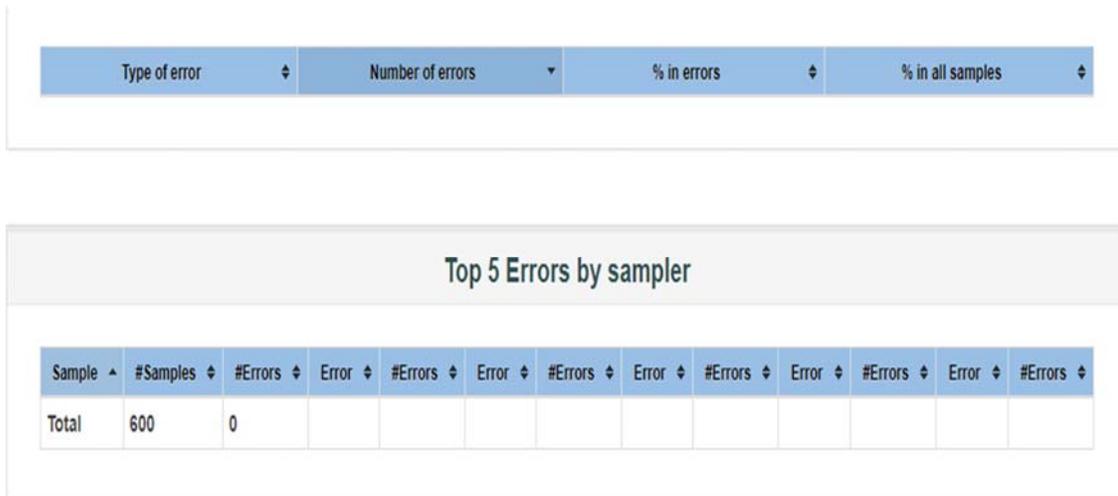
Tabel 1. Statistik Testing

Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
	Label	#Samples	KO	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received
Total	600	0	0.00%	1850.22	84	20695	1456.00	3669.80	4766.10	6413.56	23.40	329.80	2.91
Domisili	100	0	0.00%	1731.66	196	20264	1311.00	2557.50	2883.60	20165.26	4.54	56.94	0.58
Home	100	0	0.00%	1300.31	87	20695	963.00	2140.10	2409.90	20544.43	4.73	51.06	0.55
IMB	100	0	0.00%	1339.75	279	7940	1212.50	1973.50	2571.60	7902.61	4.47	69.72	0.55
Kuasa Tanah	100	0	0.00%	1688.54	89	5828	1603.00	2494.70	2569.25	5803.49	4.48	79.15	0.56
Pajak	100	0	0.00%	1610.27	84	5971	1481.50	2656.40	3118.05	5959.09	4.55	62.35	0.57
Usaha	100	0	0.00%	3430.80	259	7551	3428.50	5647.30	5922.60	7539.66	4.57	65.21	0.58

Statistik pada tabel 1 menunjukkan bahwa total 600 users yang dibagi menjadi 6 halaman yang menghasilkan 100 users per halaman yang memiliki persentase keberhasilan 100 persen tanpa ada user yang gagal mengakses website. Pada tabel diatas kolom dapat dilihat ada beberapa kolom seperti samples yang menunjukkan

total sample, KO (knock out) yang menunjukkan banyak request yang gagal, error % yang menunjukkan presentase kegagalan, pada tabel tersebut disajikan waktu response lengkap dengan detil sampai waktu per detik.

3.2.3 Statistik Error 100 user



Gambar 6. Statistik Error

Gambar 6 menunjukkan bahwa error tidak di temukan di request ini. Dari tabel tersebut dapat diambil juga informasi seperti error yang paling sering muncul, persentase, hingga jumlah error yang muncul. Dibawah informasi tersebut terdapat tabel yang berisikan 5 error yang paling sering muncul. Pada table tersebut dapat dilihat juga tipe dari error yang didapat dan juga total error dari error tersebut yang dikelompokkan menjadi satu jenis error yang sama.

3.2.4 Ringkasan Request 500 user



Gambar 7. Ringkasan Request

Gambar 7 menjelaskan beberapa poin yang dapat dilihat seperti apdex(Application Performance Index), apdex adalah sebuah standard terbuka untuk mengukur performa perangkat lunak dalam komputasi yang tujuannya adalah mengubah pengukuran menjadi pengetahuan yang dalam tentang kepuasan pengguna dengan cara menggunakan teknik penganalisaan yang seragam, apdex memiliki rentan nilai dari 0 sampai 1, 0 adalah tidak puas dan 1 adalah sangat puas. Pada tabel terdapat dua value yaitu toleration threshold dan frustration threshold yang berfungsi sebagai batasan untuk mendapatkan satisfied count dan tolerating count dimana hal tersebut digunakan untuk mendapatkan apdex. menunjukkan ringkasan dari tes yang sudah dilakukan dengan 6 halaman berbeda yang menunjukkan 73.63 persen keberhasilan dengan 26.37 persen yang gagal mengakses halaman dari 500 user yang mengakses halaman tersebut.

3.2.5 Stastistik Testing 500 user

Tabel 2 menunjukkan bahwa total 3000 samples (user) yang dibagi menjadi 6 halaman yang menghasilkan 500 user per halaman yang memiliki persentase kegagalan berkisar 42 persen sampai 94 persen tingkat keberhasilan mengakses website. Pada tabel diatas kolom dapat dilihat ada beberapa kolom seperti samples yang menunjukkan total sample, KO (knock out) yang menunjukkan banyak request yang gagal, error % yang menunjukkan presentase kegagalan, pada tabel tersebut disajikan waktu response lengkap dengan detil sampai waktu per detik.

Tabel 2. Statistik Testing

Requests	Executions			Response Times (ms)						Throughput	Network (KB/sec)		
	Label ^	#Samples ↕	KO ↕	Error % ↕	Average ↕	Min ↕	Max ↕	Median ↕	90th pct ↕		95th pct ↕	99th pct ↕	Transactions/s ↕
Total	3000	791	26.37%	5003.74	23	305871	3113.00	12720.50	15758.00	30862.55	8.70	92.76	1.08
Domisili	500	294	58.80%	8849.33	42	50460	7034.50	16857.00	23027.20	34346.78	7.84	41.58	1.01
Home	500	126	25.20%	2592.02	57	22551	2167.50	5264.20	6249.55	8681.16	21.56	175.89	2.51
IMB	500	109	21.80%	6394.87	29	304019	4175.00	10600.00	19105.20	25931.49	1.47	18.05	0.18
Kuasa Tanah	500	31	6.20%	5986.86	26	305871	2948.50	14864.10	20888.35	33994.34	1.48	24.49	0.19
Pajak	500	167	33.40%	2900.30	35	25316	2517.00	5044.80	5992.50	21191.69	8.19	75.71	1.03
Usaha	500	64	12.80%	3299.09	23	43097	3107.50	5971.70	6833.70	20610.81	1.48	18.45	0.19

Type of error ↕	Number of errors ↕	% in errors ↕	% in all samples ↕
502/Bad Gateway	1815	100.00%	60.50%

Top 5 Errors by sampler												
Sample ^	#Samples ↕	#Errors ↕	Error ↕	#Errors ↕	Error ↕	#Errors ↕	Error ↕	#Errors ↕	Error ↕	#Errors ↕	Error ↕	#Errors ↕
Total	3000	1815	502/Bad Gateway	1815								
Domisili	500	395	502/Bad Gateway	395								
Home	500	308	502/Bad Gateway	308								
IMB	500	253	502/Bad Gateway	253								
Kuasa Tanah	500	353	502/Bad Gateway	353								
Pajak	500	327	502/Bad Gateway	327								
Usaha	500	179	502/Bad Gateway	179								

Gambar 8. Statistik Error

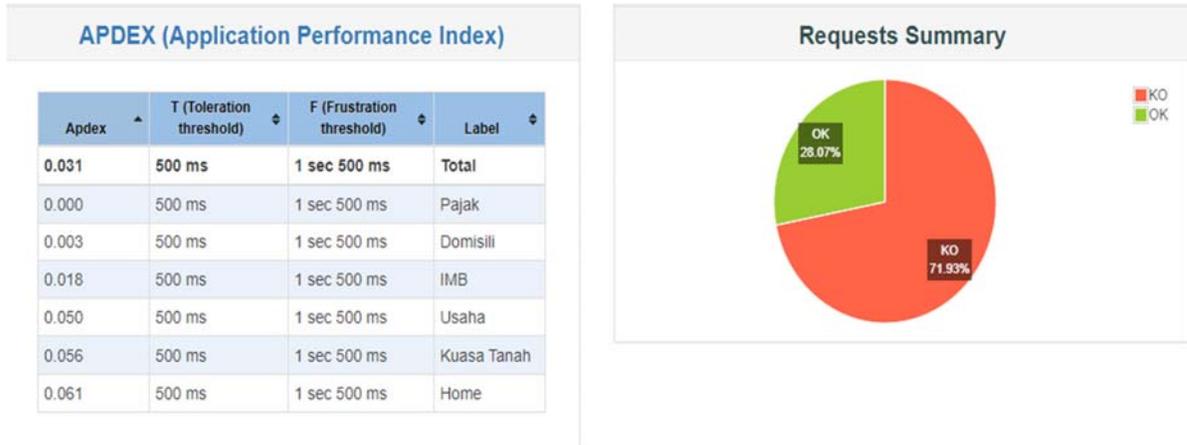
3.2.6 Statistik Error 500 user

Gambar 8 menunjukkan bahwa error yang paling sering ditemukan adalah 502/bad gateway dengan kemunculan hingga 60.5 persen user yang mendapatkan kode error ini. Dari tabel tersebut dapat diambil juga informasi seperti error yang paling sering muncul, persentase, hingga jumlah error yang muncul. Dibawah informasi tersebut terdapat tabel yang berisikan 5 error yang paling sering muncul. Pada table tersebut dapat dilihat juga tipe dari error yang didapat dan juga total error dari error tersebut yang dikelompokkan menjadi satu jenis error yang sama.

3.2.7 Ringkasan Request 1000 user

Gambar 9 mejelaskan beberapa poin yang dapat dilihat seperti apdex(Application Performance Index), apdex adalah sebuah standard terbuka untuk mengukur performa perangkat lunak dalam komputasi yang tujuannya adalah mengubah pengukuran menjadi pengetahuan yang dalam tentang kepuasan pengguna dengan cara

menggunakan teknik penganalisaan yang seragam, apdex memiliki rentan nilai dari 0 sampai 1, 0 adalah tidak puas dan 1 adalah sangat puas. Pada tabel terdapat dua value yaitu toleration threshold dan frustration threshold yang berfungsi sebagai batasan untuk mendapatkan satisfied count dan tolerating count dimana hal tersebut digunakan untuk mendapatkan apdex. menunjukkan ringkasan dari test yang sudah dilakukan dengan 6 halaman berbeda yang menunjukkan 28.07 persen user berhasil dan 71.93 persen user gagal untuk mengakses halaman dari 1000 user yang mengakses halaman tersebut.



Gambar 9. Ringkasan Request

3.2.8 Statistik Testing 1000 user

Tabel 3. Statistik Testing

Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
	Label	#Samples	KO	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received
Total	6000	4316	71.93%	5456.35	21	52007	2307.00	21069.90	22847.00	25177.96	105.02	488.66	13.07
Domisili	1000	812	81.20%	2801.05	52	34640	2379.50	4052.10	5456.55	33716.61	27.62	80.63	3.56
Home	1000	705	70.50%	2226.89	286	33651	1600.00	3301.60	4985.90	30463.66	29.68	101.88	3.45
IMB	1000	728	72.80%	6514.54	21	37479	2592.00	19754.00	21066.65	22483.37	18.09	88.52	2.24
Kuasa Tanah	1000	536	53.60%	2582.05	46	32318	2041.00	4074.90	4784.25	11299.35	19.00	165.75	2.39
Pajak	1000	888	88.80%	16358.44	53	52007	21434.00	23667.90	24849.85	32296.87	18.06	41.24	2.27
Usaha	1000	647	64.70%	2255.11	39	9267	1968.00	4129.70	4597.15	6475.05	18.10	102.52	2.28

Tabel 3 menunjukkan bahwa total 6000 samples (user) yang dibagi menjadi 6 halaman yang menghasilkan 1000 user per halaman yang memiliki persentase kegagalan berkisar 24.3 persen sampai 50.6 persen tingkat keberhasilan mengakses website. Pada tabel diatas kolom dapat dilihat ada beberapa kolom seperti samples yang menunjukkan total sample, KO (knock out) yang menunjukkan banyak request yang gagal, error % yang menunjukkan presentase kegagalan, pada tabel tersebut disajikan waktu response lengkap dengan detail sampai waktu per detik.

3.2.9 Statistik Error 1000 user

Gambar 10 menunjukkan bahwa error yang paling sering ditemukan adalah 502/bad gateway dengan kemunculan hingga 62.98 persen user yang mendapatkan kode error ini. Dari tabel tersebut dapat diambil juga informasi seperti error yang paling sering muncul, persentase, hingga jumlah error yang muncul. Dibawah informasi tersebut terdapat tabel yang berisikan 5 error yang paling sering muncul. Pada table tersebut dapat dilihat juga tipe dari error yang didapat dan juga total error dari error tersebut yang dikelompokkan menjadi satu jenis error yang sama.

Errors			
Type of error	Number of errors	% in errors	% in all samples
502/Bad Gateway	4313	99.93%	71.88%
Non HTTP response code: java.net.SocketException/Non HTTP response message: Connection reset	3	0.07%	0.05%

Top 5 Errors by sampler											
Sample	#Samples	#Errors	Error	#Errors	Error	#Errors	Error	#Errors	Error	#Errors	Error
Total	6000	4316	502/Bad Gateway	4313	Non HTTP response code: java.net.SocketException/Non HTTP response message: Connection reset	3					
Domisili	1000	812	502/Bad Gateway	812							
Home	1000	705	502/Bad Gateway	705							
IIMB	1000	728	502/Bad Gateway	728							
Kuasa Tanah	1000	536	502/Bad Gateway	533	Non HTTP response code: java.net.SocketException/Non HTTP response message: Connection reset	3					
Pajak	1000	888	502/Bad Gateway	888							
Usaha	1000	647	502/Bad Gateway	647							

Gambar 10. Statistik Error

3.2.10 Ringkasan Komparasi

Tabel 4 .Ringkasan Komparasi

Jumlah User	Redis	Persentase keberhasilan	Persentase kegagalan
100	tidak	100%	0%
100	ya	100%	0%
500	tidak	39.50%	60.50%
500	ya	73.63%	26.37%
1000	tidak	37.02%	62.98%
1000	ya	28.07%	71.93%

Pada paparan tabel 4 dapat disimpulkan bahwa redis di jumlah user 500 mengalami peningkatan presentase keberhasilan yang naik dari 39.50 persen menjadi 76.63 persen yaitu naik sebesar 37.13 persen. Sedangkan persentase keberhasilan turun di jumlah user 1000 dari 37.02 persen menjadi 28.07 persen yaitu sebesar 8.95 dan persentase keberhasilan di user 100 tidak berubah yaitu di angka 100 persen.

Setelah melihat angka penurunan gambar 14, penulis mempertanyakan kenapa angka persentase bisa turun di kasus jumlah 1000 user dan penulis menemukan bahwa memory server yang dipakai oleh penulis tidak

mampu meng-handle user sebanyak itu dengan redis sehingga RAM yang dipakai habis yang menyebabkan terjadi error seperti ini.

### 3.2.11 Error Log Server

```

2020/07/12 10:56:34 [error] 10656#10656: *315815 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:34 [error] 10656#10656: *315817 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:34 [error] 10656#10656: *315819 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:34 [error] 10656#10656: *315821 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:34 [error] 10656#10656: *315823 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:34 [error] 10656#10656: *315825 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:34 [error] 10656#10656: *315827 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:34 [error] 10656#10656: *315829 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316055 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316057 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316059 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316061 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316063 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316065 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316067 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316069 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316071 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316073 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316075 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316077 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316079 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316081 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316083 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316085 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316087 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316089 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316091 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316093 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316095 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316097 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316147 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316149 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316151 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316153 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316155 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316157 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316159 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr
2020/07/12 10:56:36 [error] 10656#10656: *316161 connect() to unix:/var/run/php/php7.2-fpm.sock failed (11: Resource temporarily unavailable) while connecting to upstr

```

Gambar 11 . Error Log Server

Error log yang penulis temukan berisi seperti gambar 11 yang jika dibaca menyatakan resource temporary unavailable yang diartikan bahwa memory yang diperlukan untuk memproses request yang masuk sudah tidak cukup lagi.

## 4. KESIMPULAN

Berdasarkan dari proses analisa dari yang dilakukan oleh penulis dengan menggunakan 100, 500, dan 1000 user. Penulis menemukan bahwa kegagalan user dalam mengakses muncul pada 500 dan 1000 user yang mencapai 62.98 persen.

Setelah penulis melakukan analisa pada sistem berjalan, penulis melakukan implementasi redis pada sistem berjalan dan analisa pada aplikasi sistem terintegrasi redis yang menunjukkan pada data pengguna 100, 500, dan 1000. Penulis menemukan bahwa ada peningkatan sampai 73.63 persen dengan erro yang meningkat pada 1000 user yang disebabkan oleh load yang tak dapat di handle lagi oleh server dan dengan ini maka penulis menarik beberapa kesimpulan sebagai berikut:

- Adanya peningkatan limit user setelah implementasi redis dari 39.5% ( 198 users ) menjadi 73.63% ( 368 users ) pada percobaan dengan 500 user yang peningkatannya hampir 100% dari sebelum implementasi redis dan setelah implementasi redis di lakukan.
- Server yang dipakai sekarang memiliki limit akses user yang rendah yaitu di angka 198 user dan jika diakses bersamaan akan membuat server down.
- RAM pada server tidak mencukupi untuk dipakai jika user sudah melebihi 368 user yang juga akan menyebabkan server down itu juga sudah dibantu dengan melakukan setup redis pada server.

## 5. SARAN

Setelah melakukan Perancangan dan Implementasi Strategi Redis dalam Sistem E-office, peneliti menyadari bahwa ada bagian-bagian dari server yang masih bisa di tingkatkan, antara lain:

- Upgrade kapasitas sever seperti CPU, RAM, maupun Storage
- Harus di bersihkannya cache yang ada di databases dan file yang tidak digunakan secara berkala.
- Mengurangi pengulangan coding yang berulang-ulang.

## UCAPAN TERIMA KASIH

Penulis menyadari bahwa penelitian ini tidak akan terwujud tanpa ada dukungan dari berbagai pihak yang mendukung dan terus menyemangati penulis. Penulis juga mengapresiasi ucapan terima kasih yang sebesar-besarnya kepada pihak yang telah mendukung proses penyelesaian penelitian ini.

---

**DAFTAR PUSTAKA**

- [ 1 ] Carlson, Dr. Josiah L. (2013). *Redis In Action*. Manning Publications. Co. 20 Baldwin Road.
- [ 2 ] Macedo, Tiago. Fred Oliveira (2011). *Redis Cookbook*. O'Reilly Media. Sebastopol
- [ 3 ] Maharani, Meilan Anastasia. (2013). *Trik Menguasai PHP + jQuery Berbasis Linux & Windows*. Lokomedia. Yogyakarta.
- [ 4 ] Novianta, Muhammad Andang dan Emy Setyaningsih. (Agustus 2015). "Sistem Informasi Monitoring Kereta Api Berbasis Web Server Menggunakan Layanan GPRS." *Jurnal Momentum*. Vol.17, no. 2: hal. 61.
- [ 5 ] Naista, David. (2017). *Bikin Framework PHP Sendiri dengan OOP & MVC*. Cetakan Kedua. Lokomedia. Yogyakarta.
- [ 6 ] Wahana Komputer. (2014). *Mudah Membuat Aplikasi SMS Gateway dengan CodeIgniter*. PT Elex Media Komputindo. Jakarta.
- [ 7 ] Wahana Komputer. (2014). *Mudah Membuat Aplikasi SMS Gateway dengan CodeIgniter*. PT Elex Media Komputindo. Jakarta.
- [ 8 ] Naista, David. (2017). *Bikin Framework PHP Sendiri dengan OOP & MVC*. Cetakan Kedua. Lokomedia. Yogyakarta.
- [ 9 ] Wahana Komputer. (2014). *Mudah Membuat Aplikasi SMS Gateway dengan CodeIgniter*. PT Elex Media Komputindo. Jakarta
- [ 10 ] Hodson, Ryan. (2012). *Git Succinctly*. Syncfusion. Morrisville.
- [ 11 ] Booth, Joseph D. (2016). *GitHub Succinctly*. Syncfusion. Morrisville.
- [ 12 ] Naista, David. (2017). *Bikin Framework PHP Sendiri dengan OOP & MVC*. Cetakan Kedua. Lokomedia. Yogyakarta.
- [ 13 ] Hidayatullah, Priyanto dan Jauhari Khairul Kawistara. (2017). *Pemrograman Web Edisi Revisi*. Informatika. Bandung.
- [ 14 ] Naista, David. (2017). *Bikin Framework PHP Sendiri dengan OOP & MVC*. Cetakan Kedua. Lokomedia. Yogyakarta.
- [ 15 ] Wood, Kenny. (2018). *Confident Web Design*. Kogan Page. London.
- [ 16 ] Naista, David. (2017). *Bikin Framework PHP Sendiri dengan OOP & MVC*. Cetakan Kedua. Lokomedia. Yogyakarta.
- [ 17 ] Putra, Chandra Anugrah. (2018). *Merancang Web Menggunakan Dreamweaver*. Teknosain. Yogyakarta.
- [ 18 ] Naista, David. (2017). *Bikin Framework PHP Sendiri dengan OOP & MVC*. Cetakan Kedua. Lokomedia. Yogyakarta.
- [ 19 ] Nugraha, Teguh. (2013). *Website Development Fundamental: Fitur, Layout, dan Operasional Lebih Maju*. Nuansa Cendekia. Bandung.
- [ 20 ] Naista, David. (2017). *Bikin Framework PHP Sendiri dengan OOP & MVC*. Cetakan Kedua. Lokomedia. Yogyakarta.
- [ 21 ] Stauffer, Matt. (2017). *Laravel: Up and Running*. O'Reilly Media. Sebastopol.
- [ 22 ] Anwar, Saipul. Yasin Efendi, Rushendra Rustam, Andrew. (2016). "Perancangan Sistem Informasi Pendaftaran Mahasiswa Baru dan Pengisian Kartu Rencana Studi. (KRS) Amik Wahana Mandiri Berbasis Web Mobile." *Jurnal Sistem Informasi*. Vol. 9, no. 1: hal. 75.
- [ 23 ] Sukanto, Rosa A., dan M. Shalahuddin. (2018). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Informatika. Bandung.