



UNIVERSITAS INDONESIA

Laporan Tugas

Mata Kuliah:  
Infrastruktur Teknologi Informasi

## **Analisis Requirement Sistem KPU di Indonesia**

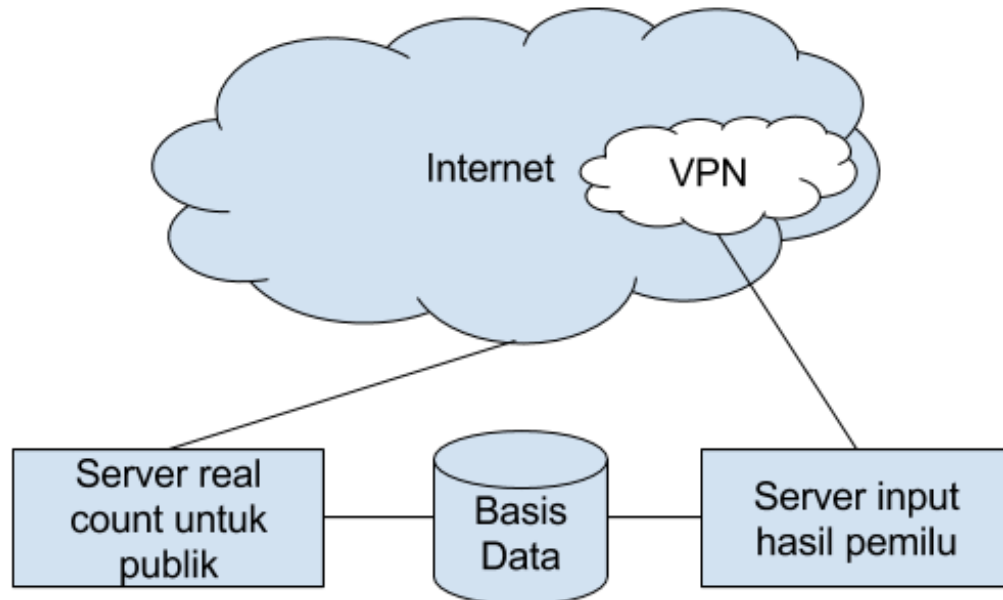
Oleh:

Hendro (1606965064)  
Riza Febriyanto (1606965360)  
Rizka Yuliana (1606965386)

FAKULTAS ILMU KOMPUTER  
PROGRAM STUDI MAGISTER TEKNOLOGI INFORMASI  
JAKARTA  
APRIL 2017

## 1. Latar Belakang

Sebagai salah satu perwujudan transparansi pemilihan, Komisi Pemilihan Umum (KPU) membuat sebuah sistem untuk menampilkan rekapitulasi *real count* dari Tempat Pemungutan Suara (TPS). Hasil dari TPS diinput dari beberapa titik melalui koneksi VPN ke server KPU. Hasil *real count* tersebut selanjutnya dapat langsung diakses ke publik melalui *website real count* KPU. Mengingat pentingnya pemilu di suatu negara, dimana hal tersebut berkaitan dengan pemimpin masyarakat dimasa depan maka animo masyarakat untuk melihat hasil pemilu sangat tinggi. Arsitektur sistemnya sebagai berikut:



Gambar1.1 Arsitektur Sistem KPU

## 2. Kondisi Studi Kasus

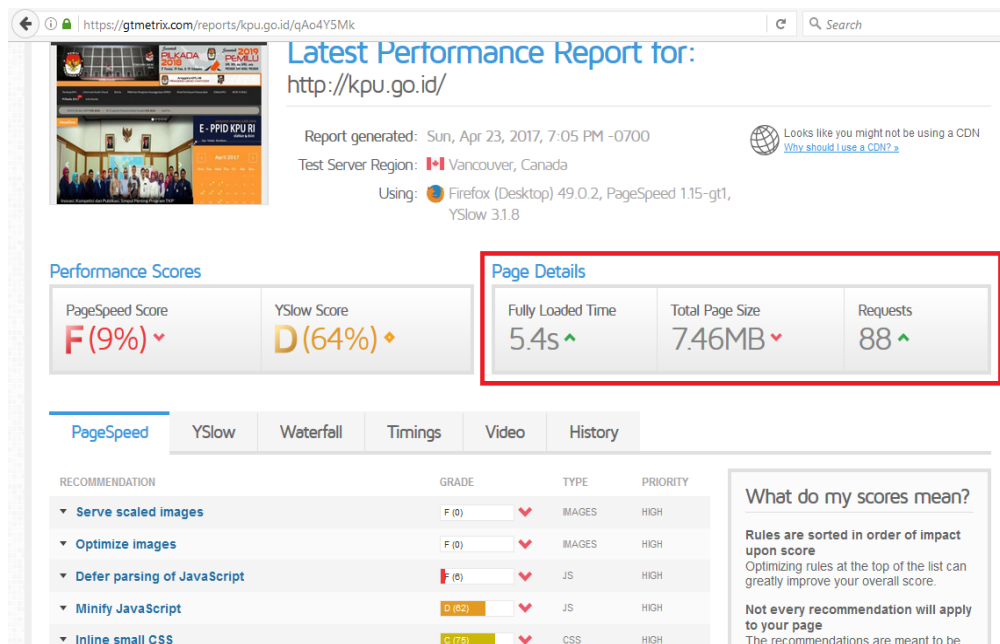
- a. Kondisi umum
  - Jumlah TPS di Indonesia: 106.253 TPS
  - Data diinput selama kurang lebih 3 hari,
- b. Kondisi Infrastruktur Jaringan
  - Saat ini ada dua link *Internet Service Provider (ISP)*, *bandwidth* masing-masing ISP 100 Mbps (total 200 Mbps).
  - Data yang diinput ke dalam sistem adalah 1.2 MB per TPS.
- c. Kondisi Infrastruktur Server/Aplikasi
  - Sistem *real count* ini menggunakan 2 buah server fisik yang divirtualisasi.
  - Satu server fisik digunakan untuk server basis data dan server *image*.
  - Satu server fisik digunakan untuk server *Real count* dan server input hasil.

- Setiap server mendapatkan alokasi memori sebesar 4 GB.
- Satu koneksi ke server aplikasi rata - rata menggunakan memori sebesar 10 MB.
- Satu koneksi ke server basis data rata - rata menggunakan memori sebesar 5 MB.

### 3. Analisis permasalahan studi kasus

#### a. Infrastruktur Jaringan

- Sesuai dari penjelasan studi kasus bahwa penggunaan *bandwidth* belum digunakan secara penuh, hal ini terjadi karena sebuah server hanya mampu menangani 15 transaksi per *second*.
- Jumlah TPS adalah 106.253, data yang akan diupload adalah sebesar 1,2 MB dengan waktu upload maksimal 3 hari dan asumsi proses upload dilakukan selama jam kerja (8 jam perhari).
- Dengan menggunakan data di atas dan melakukan distribusi waktu yang merata, maka dalam sehari total TPS yang melakukan upload data adalah 35.418 TPS per hari, atau 4.427 TPS per jam atau 74 TPS per menit.
- Dari data upload sebesar 1.2 MB per TPS per menit dikali 74 maka bandwidth yang digunakan untuk proses upload adalah 88.8 MB per menit atau 1.48 MBps untuk 74 TPS atau 0.02 MBps per TPS.
- Hasil analisis dengan menggunakan *tools gtmatrix* (<https://gtmetrix.com>), *webpage site kpu.go.id* memiliki ukuran 7,46 MB dengan rata-rata waktu akses adalah 5,4 detik, dengan demikian rata-rata bandwidth yang digunakan dalam sebuah akses ke *website kpu.go.id* adalah 7,46 MB/5,4 s atau 1,38 MBps. Berikut tampilan hasil analisis pada *gtmetrix*.



Gambar 3.1 Tampilan hasil analisis pada *gtmetrix*

## b. Infrastruktur Server

Dari penjelasan studi kasus, disebutkan bahwa saat ini server hanya mampu menangani 15 transaksi *per second* walaupun sudah menggunakan *Random Access Memory* (RAM) sebesar 4 GB. Dengan menggunakan asumsi 1 GB RAM akan digunakan untuk *operating system*, maka untuk menjalankan transaksi setiap server masih punya 3 GB alokasi RAM. Dengan menggunakan data dari studi kasus bahwa pada server setiap transaksi akan menggunakan 10 MB RAM, maka seharusnya server dapat melayani  $3 \text{ GB} \times 1024 / 10 \text{ MB}$  atau 307 transaksi.

Perbedaan antara kemampuan server yang seharusnya (307 transaksi *per second*) dan data transaksi server yang sebenarnya (15 transaksi *per second*) mengindikasikan adanya permasalahan di infrastruktur server.

Permasalahan tersebut dapat disebabkan oleh beberapa hal:

### 1) Keamanan

- Adanya serangan *Network and Application Distributed Denial of Services* (DDOS) yang menyebabkan server tidak dapat memiliki *resource* untuk melayani transaksi yang masuk secara maksimal.
- Saat ini ternyata *website* [kpu.go.id](http://kpu.go.id) masih menggunakan *http* dan belum menggunakan skema *https* yang menggunakan proses *encrypt/decrypt* data sehingga kemungkinan terjadi serangan yang dapat memperlambat akses.

### 2) Performa

Perbedaan performa server dalam memproses jumlah transaksi bisa disebabkan oleh beberapa hal, yaitu sebagai berikut:

- *I/O bottleneck*, *I/O bottleneck* dapat disebabkan oleh:

#### a) *Virtual Storage*

*Virtual environment* merupakan sebuah jaringan yang kompleks dari *resource* yang *independent*. Terdapat beberapa komponen yang harus diperhitungkan, seperti *Virtual machine* (VM), aplikasi yang berjalan di VM, *external storage*, dan lain sebagainya. Jika VM akan menjalankan aplikasi yang memiliki proses I/O yang cukup berat, akan memberikan beban kepada *disk*. Hal ini akan menyebabkan proses lain yang berhubungan dengan *storage* yang sama dengan VM juga akan mengalami kendala performa karena terdapat perebutan *resource*. Dan jika VM kehabisan *physical memory* VM akan membatasi *disk*. Hal ini akan menyebabkan *input/output per second* (IOPS) ke *storage* sistem semakin besar. Jika banyak *user* mencoba untuk mengakses data secara bersamaan ke aplikasi yang ada di dalam VM, *storage I/O* akan terkena efeknya dan menyebabkan tidak semua *request* dari *user* dapat dijalankan.

#### b) *I/O Response Time* yang tinggi

Permasalahan ini terdapat pada *layer storage*. Jika perangkat penyimpanan membutuhkan waktu merespon permintaan I/O, maka ini mengindikasikan

adanya *bottleneck* pada *layer storage*. Perangkat penyimpanan yang sibuk juga bisa menjadi alasan mengapa waktu respon lebih tinggi.

c) Desain *Storage* yang kurang baik

Hal paling mendasar yang menyebabkan *storage bottleneck* adalah desain *storage* yang kurang baik. Hal ini dapat dilihat dari *storage* sistem tidak dapat menangani seluruh proses yang ingin dijalankan oleh *user*, yang secara langsung mempengaruhi jumlah transaksi yang dapat ditangani. Permasalahan ini dapat disebabkan penggunaan level *Redundant Array of Independent Disk (RAID)* yang tidak tepat, tidak adanya *load balancer* yang menangani proses ke *storage*, dan penggunaan *cache array* kecil.

3) Akses VPN yang tidak baik

Dikarenakan semua akses unggah data ke KPU menggunakan VPN, maka perlu adanya pengelolaan VPN yang baik agar proses input data menjadi lancar.

4) Ukuran *webpage* yang terlalu besar

Dari hasil analisis menggunakan *tools gtmatrix* nilai *page speed* ke *website kpu.go.id* adalah F atau buruk. Hal ini disebabkan oleh beberapa komponen website yang memiliki ukuran yang besar sehingga memberatkan akses *website*.

The screenshot shows a GTmetrix performance report for the website <http://kpu.go.id/>. The report was generated on Sun, Apr 23, 2017, at 7:05 PM -0700. The test server region is Vancouver, Canada. The browser used is Firefox (Desktop) 49.0.2, and the PageSpeed version is 1.15-gt1, YSlow version is 3.1.8.

**Performance Scores:**

- PageSpeed Score: **F (9%)**
- YSlow Score: **D (64%)**

**Page Details:**

- Fully Loaded Time: 5.4s
- Total Page Size: 7.46MB
- Requests: 88

**Recommendations Table:**

RECOMMENDATION	GRADE	TYPE	PRIORITY
Serve scaled images	F (0)	IMAGES	HIGH
Optimize images	F (0)	IMAGES	HIGH
Defer parsing of JavaScript	F (6)	JS	HIGH
Minify JavaScript	D (62)	JS	HIGH
Inline small CSS	C (78)	CSS	HIGH

**What do my scores mean?**

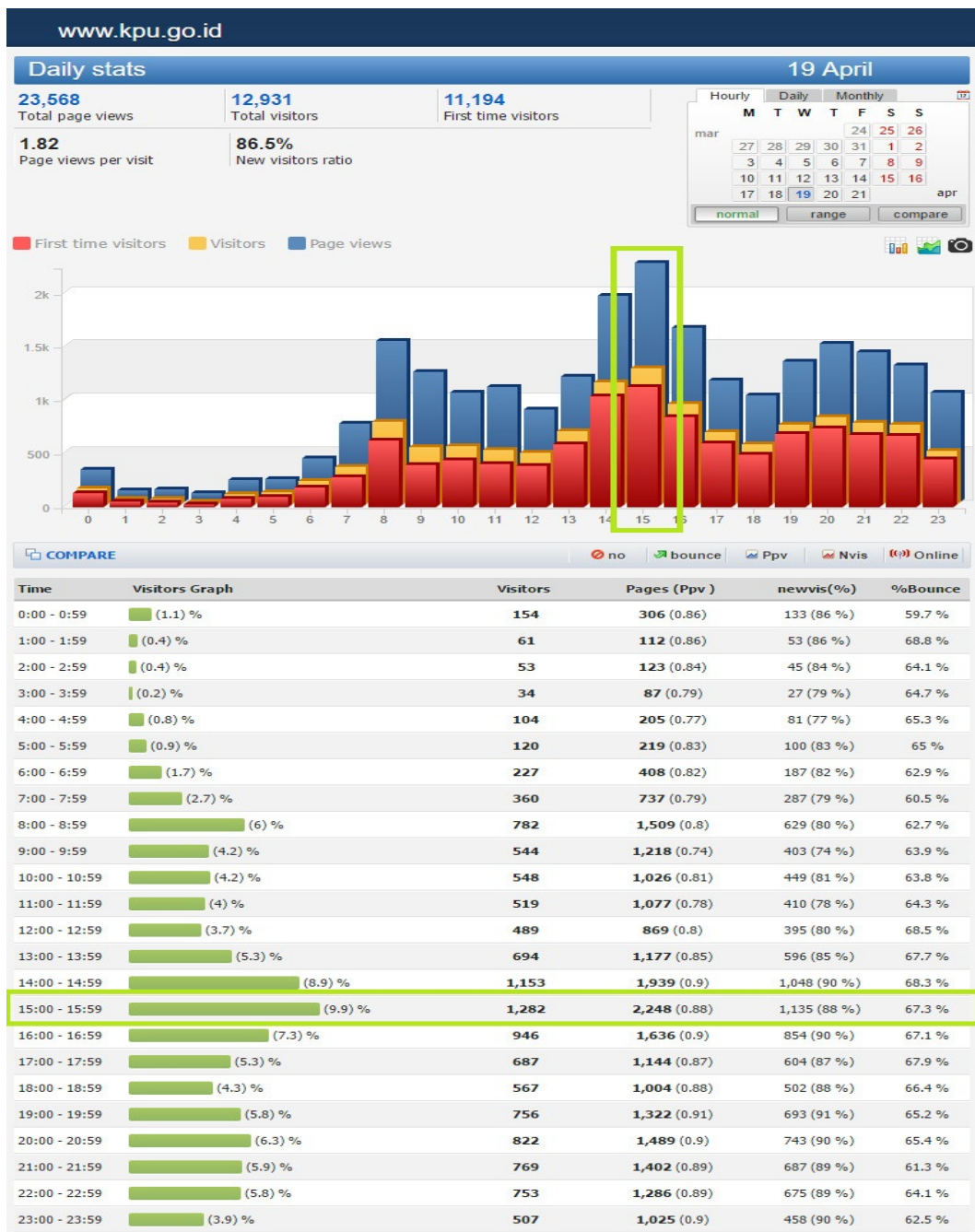
Rules are sorted in order of impact upon score. Optimizing rules at the top of the list can greatly improve your overall score.

Not every recommendation will apply to your page. The recommendations are meant to be

Gambar 3.2 Tampilan Nilai *Page Speed Website KPU* pada *gtmetrix*

#### 4. Asumsi-asumsi yang digunakan

- Akses TPS yang melakukan unggah data hanya dibuka 8 jam perhari (jam kerja) dan akses *website real count* dibuka 24 jam.
- Ukuran RAM yang bisa digunakan per server akan dipotong 1 GB yang akan dialokasikan untuk menjalankan *Operating System*.
- Jumlah pengguna maksimal dalam satu jam yang akan mengakses web real count adalah 1.500 pengguna. Hal ini berdasarkan data statistik web KPU pada pilkada DKI Jakarta 2017 maksimum jumlah pengguna maksimal adalah 1.282 (data 19 april 2017, 15.00-15.39) ditambahkan margin +/- 20% untuk *error*. Dengan asumsi koneksi yang dilakukan oleh pengguna tersebar merata, maka akses per menit adalah 25 pengguna per menit.



Gambar 4.1 Data Jumlah Pengguna Web pada Pilkada DKI 2017

## 5. Rekomendasi

### a) Infrastruktur Server

- Melakukan kompresi *file image* untuk mempercepat proses akses *website*.
- Untuk dapat melayani 1.500 pengguna yang mengakses web *real count*, dengan data rata-rata utilisasi server per akses adalah 10 MB, jumlah RAM yang dibutuhkan adalah (1.500 pengguna x 10 MB) untuk alokasi akses server ditambah Alokasi RAM untuk OS atau 15 GB + 1 GB untuk pengguna atau 16 GB RAM untuk satu server *real count*.

- Setiap server punya pair unit (HA) untuk memastikan aplikasi akan selalu bisa diakses, dan melakukan *load balance* untuk mengoptimalkan kinerja server.

b) Infrastruktur Jaringan

- Jalur internet yang digunakan dipisahkan antara jalur internet untuk unggah data TPS dan jalur internet untuk *real count* namun saling *failover* jika terjadi permasalahan.
- Pembagian beban *link* ini bisa dilakukan dengan menggunakan *link balancer*.

c) Infrastruktur Keamanan

- Perlu digunakan *firewall* yang memiliki kemampuan untuk menangani serangan *Network* dan *Application DDoS*, *Network Firewall* dan *Application Firewall*.
- Menambahkan keamanan dengan protokol *https* menggunakan *certificate* dan *key* dari *authorized partner*.
- Proses enkripsi dan dekripsi dengan menggunakan *certificate* dan *key* dilakukan oleh perangkat khusus sehingga tidak membebani server.
- Antara *web/application server* dan basis data server diberikan parameter keamanan dengan menambahkan *firewall* terkait dengan nilai dari data.
- Akses user internal KPU ke dalam *data center* dipisahkan dengan akses *public internet* dan TPS.

d) Storage

- Mengubah tipe RAID agar sesuai dengan beban kerja.
- Upgrade ke *cache* yang lebih besar. Memiliki *cache* yang besar berarti *disk* akan meningkatkan operasi *write and read* dan *I/O bottleneck* yang lebih rendah.
- Menggunakan *in-memory database* seperti *redis*, sehingga untuk data - data yang sering diakses tidak perlu mengakses ulang ke basis data.