The Design and Architecture of kawalpemilu.org

(after the fact)

Ainun Najib, Felix Halim, Andrian Kurniady, Ilham Kurnia, Fajran Iman Rusadi

Last updated: 9 August 2014

Background

The Indonesian presidential election 2014 was held on 9 July. The quick count results on that day were inconclusive (see the summary by Prof. Merlyna Lim). To help clear the confusion and fear of vote manipulations, **kawalpemilu.org** is set up. This solution was only possible because all the polling station (**TPS**) data (**C1 scans**) were openly available by the national election commission (**KPU**) for public scrutiny and verification.

Our Goal

To guard the election by organizing the C1 scans released by the KPU, digitize it, and make it easily accessible for the public to see the real count of the aggregated votes from the polling stations to the village, district, regency, province, up to the national level and provide a way to report back any anomaly on any level to get it verified and fixed.

Technical Challenges

- Over 470,000 C1 scans needs to be digitized manually. The process cannot be automated since the existing OCR tools do not work well enough to recognize handwriting. Other problems include upside down and slanted scans, blurry scans, and the tally boxes being occupied by multiple digits written vertically.

- Accurate digitization of the C1 scans via crowdsourcing has its own challenge. We quickly learned from other crowdsourcing efforts [1] that **verifying** the data entered by volunteers can be a difficult task by itself.

- Only 12 days left until the official announcement by KPU. We needed to design the data-entry to be fast, convenient, and “addictive” as possible to meet the deadline. The challenge here is that sometimes KPU website was not accessible for some period of time. Thus, we needed to design the data-entry system to be unaffected by the availability of the KPU website. That is, hot-linking the C1 scan images may not work.

- KPU website may become unavailable at any time. We had a concern that after we went public, KPU may stop serving the C1 scans to public (for any reason) and our effort would grind to a halt. With this concern, we needed to download and cache as many C1 scans as possible to guarantee availability before we went public. Additionally, we can use this resource to later identify if KPU make changes to their C1 scans.

Our Design

Our system is designed to put an emphasis on the following factors: simplicity, speed for data entry, accuracy and authenticity of the provided information, and availability. Furthermore, we empower the viewers to report errors in the data entry process or in the C1 scans themselves.

- **Simplicity**. We only show what's necessary: navigation to drill down from national level to the TPS level and provides data entry side-by-side with the C1 scan. Also, viewers should be able to go back and forward while navigating (achieved by using HTML5 History API on our single-page AJAX-based site).

- **Shareability**. We would like the viewers to be able to share exactly what they see on our site, especially the particular geographical location the viewer drilled down into. This is a typical stumbling block for a single-page AJAX-based website like ours, but can be solved by implementing an hash-based URL solution which allowed viewers to share e.g. http://www.kawalpemilu.org/#0.78203.79629. We believe this shareability factor later contributed to the viral growth of the site's traffic, because viewers can share information most pertinent to them (e.g. vote count for their hometown) to their friends. Another benefit is it makes it easier for viewers (and also us the developers) to share and report errors found.

- **Speed**. We'd like to have a very quick data entry (e.g., ~5 seconds per TPS) thus the server needs to be very responsive and the user interface must be friendly enough. That is, it supports quick entry using keyboard (via tab shortcut) and automatically jump to the next available C1 scan form when submitted. See the **data entry demo**.

- **Accuracy**. We want the volunteers to be very focused on the accuracy of the numbers they entered to avoid double work of fixing it later. We automatically compute the total legal votes ("suara sah") and the user can use this as a means to double check their input on the first two fields and fix if the sum is incorrect. Sometimes when the sum from the C1 scans itself is incorrect, the entry is flagged and the sum can be temporarily mismatch. It can be verified fixed later by other volunteers.

- **Authentication**. We deliberately put the trusted volunteer's name on the entry that they input, linking it to their Facebook account. This keeps them honest, no matter which candidate they are favoring to. We do not want anonymous users to do the data entry since it could severely damage the quality and would make it very hard to find and fix later as other crowdsourcing effort experienced (e.g., they need to ask another volunteer to re-enter the same TPS for sanity check, which slowdown the progress by at least twice). With authentication in place, we found very few errors made by the trusted volunteers. The names of the volunteers are not made public. They are disclosed only among the trusted volunteers.

- **Addictive**. We would like the users to get addicted to data entry. We've thought about gamification where one can maintain a rank list, but it may backfire since the user may change focus on speed rather than accuracy. After playing around with the design, we think that the users will get a small reward when they finished entering data for a particular village. They can roll up the hierarchy and see who is winning. We think that this good enough to keep the user motivated to continue digitizing the next village.

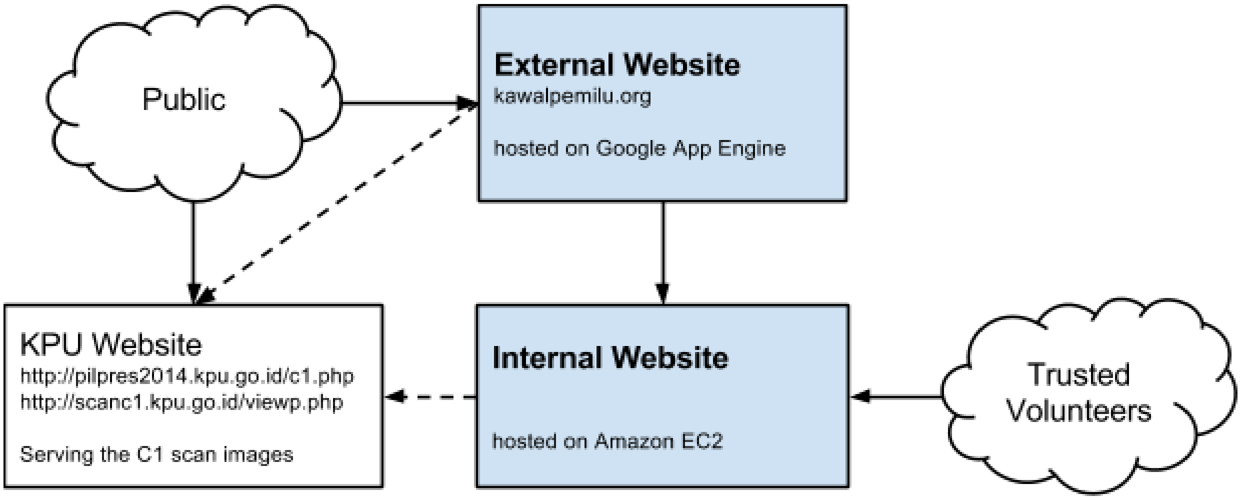
- **Security**. Due to the lack of available development time, we do not have advanced security features other than authentication for data entry and input sanitation. Thus it is important that we kept the data entry system private among the trusted volunteers. We assume that the trusted volunteers we recruited would not try to crack the data-entry system. We used a separate application for publishing the data for untrusted users (i.e., the public). We deployed a read-only Google App Engine (**GAE**) application for the public facing website (i.e., the kawalpemilu.org). GAE has a built in security features and is robust enough to handle the most common DDoS attack (by automatically spinning up additional instances when the load goes up).

  The original public version relied on a shared hosting server in Dreamhost to host the index.html page. Due to the heavy load and DDoS attacks, this was later moved to GAE (as part of the same application) using the custom domain feature. We do have contingency plan in case the internal website was exposed to public and compromised. We backup the crowdsourced data periodically and at anytime we can host another (secret) internal website easily within one hour.

- **Error Reporting**. This feature is unique to kawalpemilu.org. We used the same crowdsourcing strategy for error reporting. We provide the public facing website a way to submit an error report with notes using Google Forms into a Google Spreadsheet. For the (secret) internal website, the volunteers can just click the TPS that contains either data entry error or C1 scan errors. The error reports are aggregated to the upper level up to the national level so that one can easily track which regions needs further investigation. This helps the KPU to track and fix the errors. The group of the trusted volunteers (some of them were KPU employees) can repair the data as shown in this **fixing errors demo**.

Implementation Details

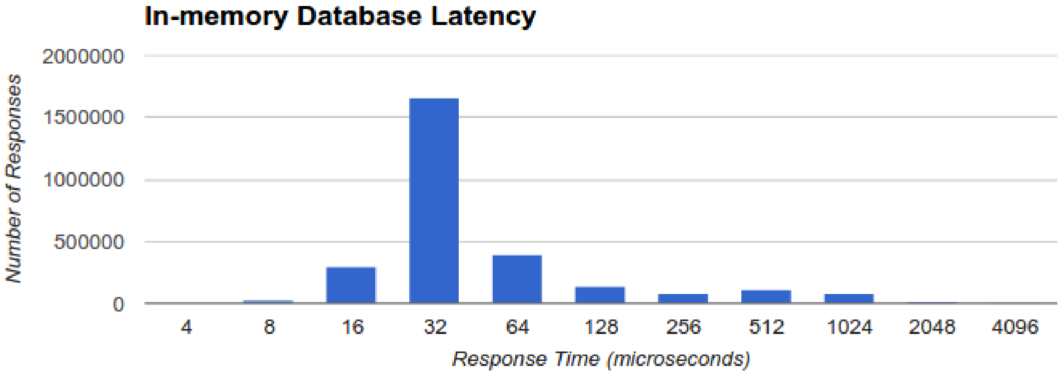
The system is divided into two parts: the internal website that is accessible only to the trusted volunteers and the external website that is accessible to the public.


Public
External Website
kawalpemilu.org
hosted on Google App Engine
KPU Website
http://pilpres2014.kpu.go.id/c1.php
http://scanc1.kpu.go.id/viewp.php
Serving the C1 scan images
Internal Website
hosted on Amazon EC2
Trusted Volunteers

In addition, we also developed an offline version of the external website that works with a data dump, allowing users to view the data dump without having to connect to the external website.

The Internal Website

The internal website consists of two servers: api-server and auth-server. The api-server is where the data is stored. We implement our own in-memory database in C++ hooked up with the http-server built on top of libuv. This provides a very fast response time. Below is the sample of latency for the API calls as recorded from our server (hosted in Amazon EC2 small instance):



The auth-server is where requests that needs authentication is proxied to. We use nodejs and passportjs to authenticate the requests and (if authorized) will forward the requests to the api-server. Requests that do not need authentication can go straight to the api-server.

The C1 form is structured such that its fourth page contains the data we need, namely the vote tally of the first candidate, the vote tally of the second candidate, the total of valid votes, and the total of invalid votes. Therefore, we concentrate the input interface on this fourth page. The C1 scans of the fourth page from the KPU website (scanc1.kpu.go.id/viewp.php) were downloaded offline (around 70GB) and cropped around the “Jumlah” (sum) column into smaller image (around 10KB) to allow fast display of the snippets and avoid the problems when KPU website is down (for any reason). Sometimes, the C1 scans are uploaded incorrectly (e.g., flipped upside down, the page order is wrong, and the fourth page is not uploaded). To handle this case, we also provide a link to all four pages (later on we were made aware that a fifth page exists for several ten thousands of C1 scans).

The Public Website

The public website initially hosted on Dreamhost and later moved to Google App Engine (**GAE**) due to DDoS attacks. It uses memcache to cache and proxy the APIs exposed by the internal website (with ~5 minute cache expiry). This allows concurrent development of the website independent from the development of the data being passed to the website. As a drawback, ordinary users cannot easily mirror this website, as all data have to be retrieved from the app engine. In fact, this dynamic property was used to clone the website without any attribution to kawalpemilu.org (e.g., http://gen-kawalpemilu.com/), while producing traffic to the app engine. Fortunately, the impact was not dramatic (in terms of cost) and could be resolved by adding a blacklist based on HTTP referral and origin headers.

The public website also hosts DA1, DB1, and DC1 recapitulation data provided by KPU. Originally the data were retrieved from a similar website (http://rekapda1.herokuapp.com/) which scraped the data from KPU website. However, due to changes in the website that broke our retrieval code, we decided to scrape the KPU website directly. The scraping code consists of a couple of python scripts that periodically download and parse the recapitulation data pages. The data are then combined and served as an API to be consumed by the public website.

The Extract Page

To allow standalone mirrors of the public facing website, we developed a separate page, called the *extract* page, that incorporates a dump of the data (in JSON format) all in one single HTML file. This format is chosen such that it is simple enough to be used by non-technical users (nothing needed to be set up, only a browser is needed to open the extract). The data format used by the dynamic site cannot be directly used, as it translates to a bigger extract page (the final extract page is 19 MB, needing a few seconds to load). We had to transform the data from the internal server and the recapitulation data scraped from the KPU website to make it more compact and at the same time easier to work with.

The extract page provides yet another option to safeguard the data integrity and quickly rebuild the original website, should the need arise. Additionally, the static data contained in the extract page allows the development of other stand alone applications such as the data visualization and quick count simulation applications. We can also show how the collected data evolves by using the dumps contained in the revisions of the extract page.

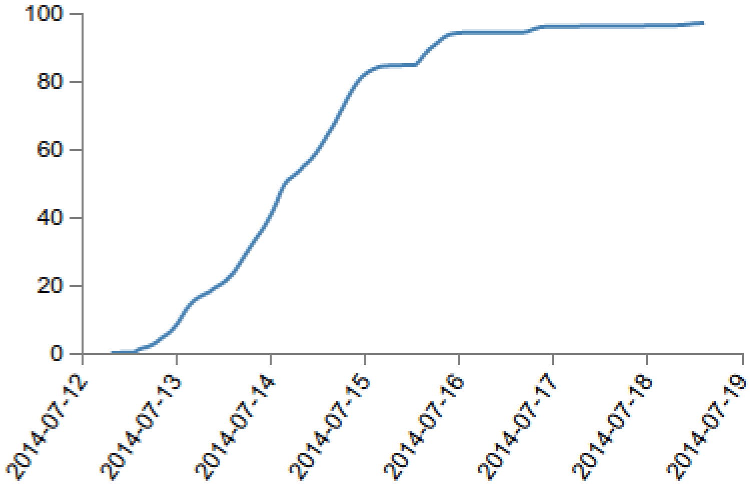
The extract page does not contain the C1 scan snippets and the jQuery library, so viewing the full information on a polling station still requires an internet connection.

The extract page was initially released twice a day. As the data changed more slowly, we released the page only once a day.

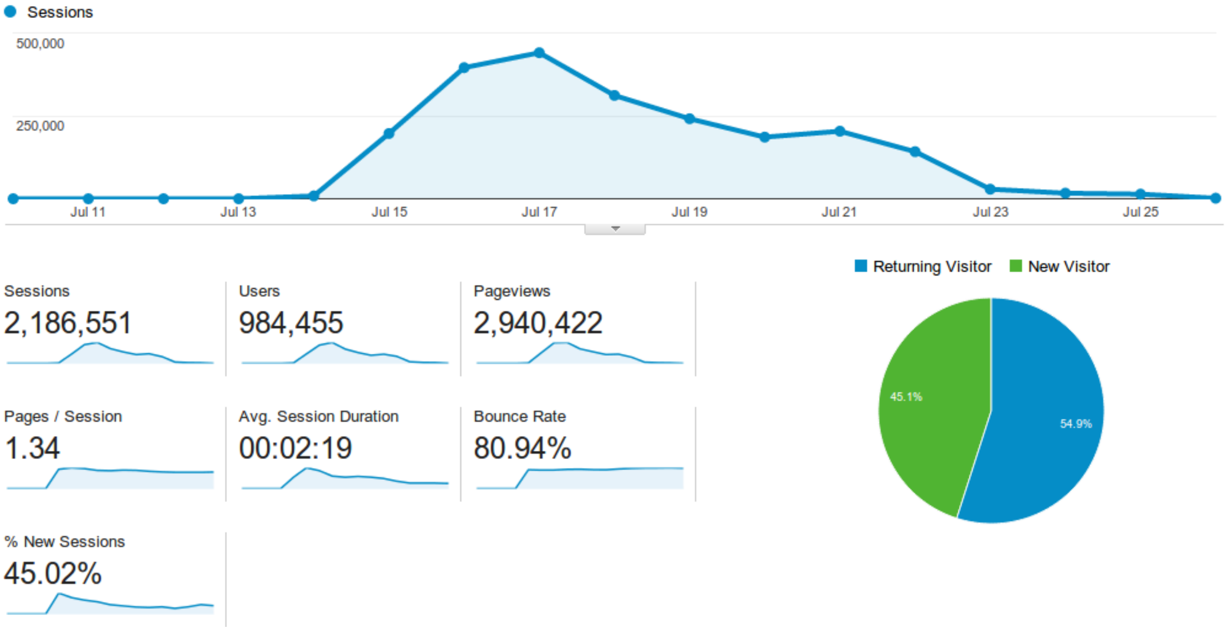
Statistics

- How many volunteers (exactly), their background, how they were recruited:
    - Recruitment of volunteers was done using a secret facebook group. Volunteers who were added into the secret group are encouraged to invite their trustworthy facebook friends into the group. It started from 5 facebook users, all Ainun's friends, to 700 users in the span of 4 days, friends of friends ... of friends.
    - From the 700 users, 431 contributed at least a single entry. Several users digitized C1 scans of more than 7000 polling stations.
    - The volunteers come from very diverse backgrounds: from a secondary school student (who is not even eligible to vote yet) to a professor, from housewives to professionals of various industries, from those who can work on kawalpemilu.org full time to those who only have few minutes to spare - yet still eager to join and contribute.
- How many person hours were needed to develop and monitor the site (up to 22 July)?
    - Felix spent 20 hours over two days developing the internal website and 10 hours spreaded over the week on bug fixing and export features.
    - Andrian spent 5 hours coding and many hours on-call monitoring.
    - Ilham spent 30 hours on coding and 20 hours monitoring.
    - Fajran spent 10 hours coding scripts for scraping the recapitulation data and scripts for preparing data for the extract page.
- The costs in $$
    - The internal website costs $10.88 per month.
    - The public website costs around $35, including the bandwidth cost incurred during the DDoS attempts.
    - The domain (kawalpemilu.org) costs for $9.95 for a year's registration.
- How many errors the crowdsourcing effort uncovered (both from the internal site and the public site)
    - We discovered around 10 thousands errors (either C1 scan errors or data entry error).
    - The data entry errors have been fixed mostly and now around 6 thousands errors regarding C1 scans (for any reasons).
    - Until the public announcement of the national level recapitulation, KPU still had not uploaded around 10 thousands C1 scans.

- It took 3 days to reach digitize 80% of the C1 scans and 6 days to reach 97%.



- The number of views to the public website:



Sessions
2,186,551

Users
984,455

Pageviews
2,940,422

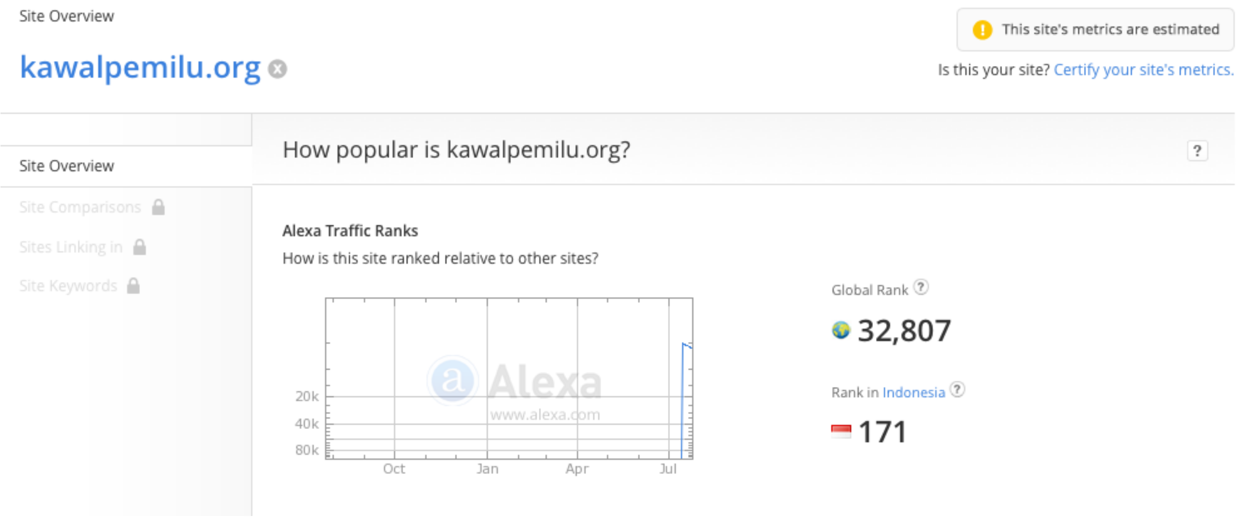
Pages / Session
1.34

Avg. Session Duration
00:02:19

Bounce Rate
80.94%

% New Sessions
45.02%

Alexa estimated on 26-07-2014 that kawalpemilu.org ranks 171 in Indonesia in terms of traffic, above groupon.co.id, path.com, kontan.co.id, bbc.co.uk, airasia.com, and rumah.com, among others.



Site Overview

kawalpemilu.org

This site's metrics are estimated

Is this your site? Certify your site's metrics.

Site Overview

Site Comparisons

Sites Linking in

Site Keywords

How popular is kawalpemilu.org?

Alexa Traffic Ranks

How is this site ranked relative to other sites?

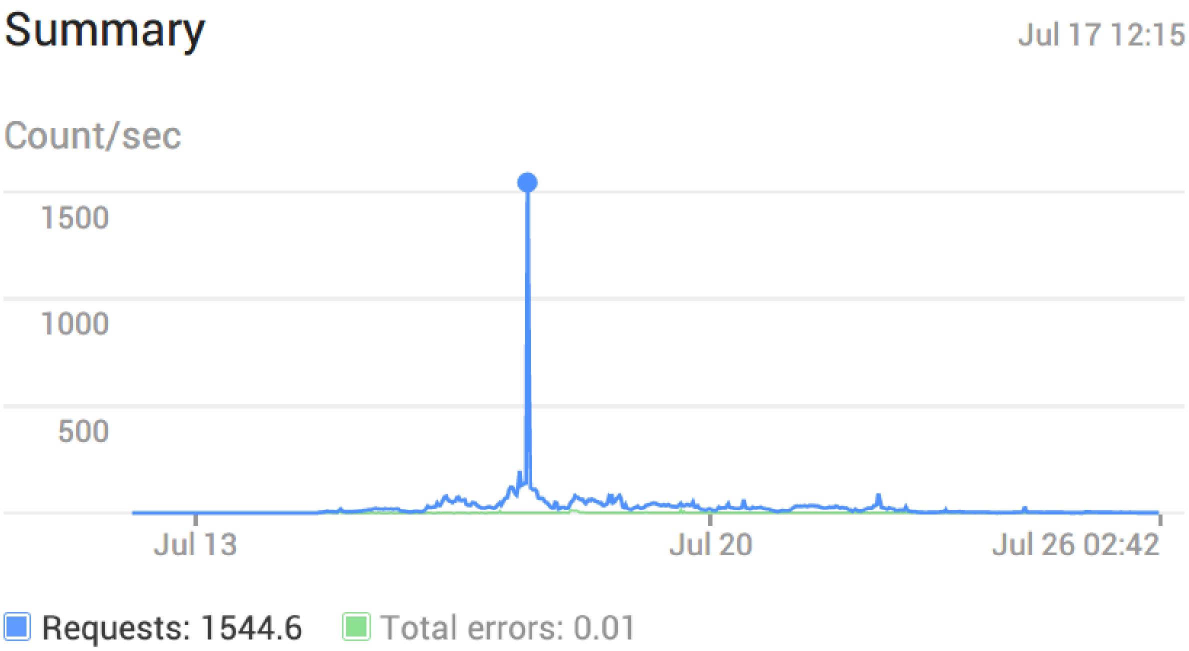
Global Rank

32,807

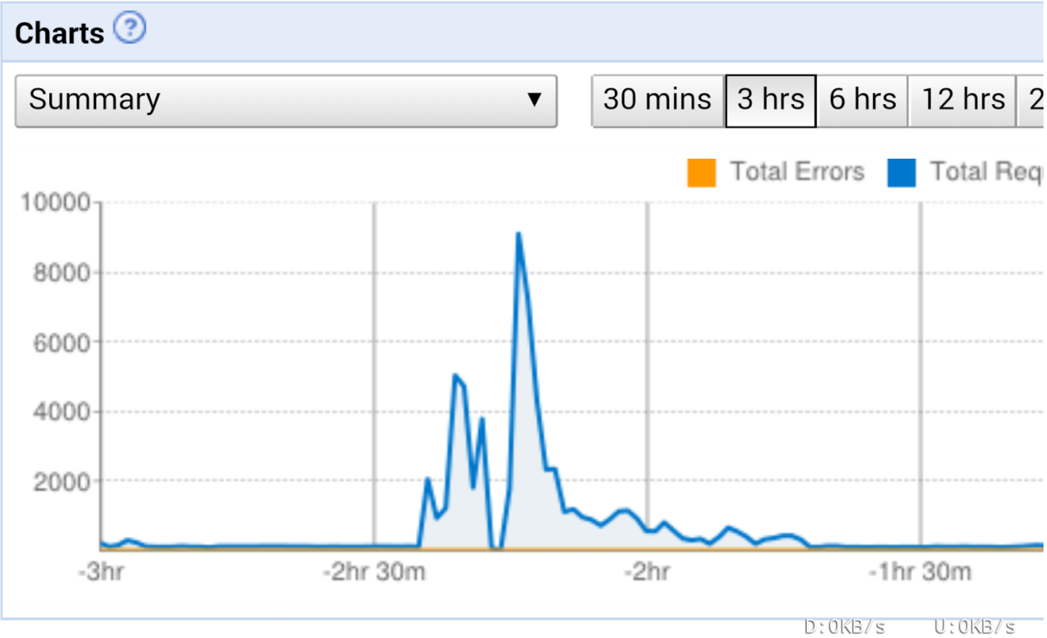
Rank in Indonesia

171

- How stark were the DDoS attacks?
  The figure below shows the requests kawalpemilu.org received since its launch. There were several attack with the most significant attack happened on July 17.



The following figure depicts a more detailed account on the worst attack as it occurred. At its peak, the website was flooded by 9000 requests per second. There was a brief period of unavailability due to the delay in AppEngine spinning up new instances to handle the increased traffic, but this resolved itself shortly afterwards.



- How many C1 scans are digitized, how many C1 scans have errors, the response of KPU to error reports, etc.

Our hope for the future

- Improve the last mile Internet speed in Indonesia. We wouldn't have finished developing the website if we were in Jakarta, for example. We couldn't have downloaded the 70GB C1 scans in 2 days.
- Improve IT education in Indonesia. So far, we see again and again that any field touched by IT can function significantly more efficient and effective. We hope the next election will utilize IT better, by providing a transparent, secure and robust site that is open where anyone can verify its correctness and thus avoid possibilities of cheating as much as possible.
- Provide a place for professional programmers in Indonesia to grow and contribute back to Indonesia. The developers of kawalpemilu.org are all Indonesians, living from outside Indonesia: Singapore, USA, Australia, Germany and the Netherlands, and the majority are working for foreign companies.
- Now we have high quality training data (thanks to the trusted volunteers) that can be used to improve research on handwriting recognition for digits. When the handwriting recognition is mature enough, we may not need to crowdsource as much. Instead the crowdsourcing effort may be geared towards corner cases where the handwriting recognition technique does not perform well.

Acknowledgement

We would like to thank the trusted volunteers for giving their precious time to help in digitizing, and verifying the C1 scans. This shows that when we combine our efforts, we can achieve great things together.

About the Authors

The authors are all Indonesians, staying in different countries, communicating via the Facebook chat:

- **Ainun Najib** (in Singapore), the inventor, the volunteer coordinator and the spokesperson.
- **Felix Halim** (in California, USA) designed and developed the internal data-entry website.
- **Andrian Kurniady** (in Sydney, Australia) set up a public facing website (kawalpemilu.org) which takes read-only data from the internal website and works with:
- **Ilham Kurnia** (in Kaiserslautern, Germany) and
- **Fajran Iman Rusadi** (in Amsterdam, Netherlands) integrated and maintained the votes from KPU website for the DA1, DB1, DC1, and vote counts in foreign countries, and developed a mirrorable “extract” version of the view-only site.

References

1. Rancangan System Crowdsource Entry data C1, Ruli Manurung.
2. Why KawalPemilu.org should get a UX Award, Sulistyawati.
3. How Papua Voted, Cillian Nolan.

Videos

- Ainun Najib interview in Metro TV.
- Wawancara Ainun Najib: kawalpemilu.org 'Kehendak Ilahi', Global Indonesian Voices.
- https://www.youtube.com/watch?v=71ojjt7rnmc&feature=youtu.be

News

- Indonesian techies, Ben Bland.
- Concerned Voters Crowdsource Ballot Count, Lily Yulianti Farid.
- Guardians of Indonesia’s votes, Ulma Haryanto.
- Widodo Leads Indonesia Race in Website Tally of Actual Votes, Berni Moestafa, Brian Leonal and Rieka Rahadiana.
- Indonesia's Jokowi leads with 80 percent of votes counted: private group, Kanupriya Kapoor.
- Technology for Transparency, Inaya Rakhmani.
- Open election data + mass interaction = Indonesian public as watchdog, Ruli Manurung.

- With the Election of Joko Widodo, Indonesia Writes a New Chapter, Yenni Kwok.
- Voluntary watchdog initiative displays democracy at its finest, (The Jakarta Post).
- Crowdsourcing, Jurus Jitu Awasi Pemilu, Oik Yusuf (in Indonesian).
- Elezioni in crowdsourcing, l’esempio modello dell’Indonesia, Fabio Chiusi (in Italian).