

Rancangan sistem Crowdsorce Entry data C1:

-Initial version by Ruli Manurung 14.07.11

Dokumen ini merupakan catatan ringkas mengenai rancangan sistem yang ingin dikembangkan untuk men-crowdsorce entry data scan form C1 Pilpres 2014. Targetnya sih ini bisa dikembangkan secepatnya, namun rasanya lebih baik kita spend more time untuk mengembangkan sistem yang rancangannya tepat dan tidak mengorbankan aspek tertentu untuk mengejar tanggal 22 Juli. Bahkan kalau hasilnya selesai setelah tanggal 22 Juli, sistem ini masih bermanfaat karena akan menjadi credible alternative ke hasil resmi KPU.

Sementara ini (11 Juli 2014) sudah ada 2 sistem yang dikembangkan dengan sangat cepat, yaitu:

https://docs.google.com/forms/d/1Z7uM-u9uiZy9XRjxITEnbStf5FGMFZTd27Q6rHqi17o/viewform?edit_requested=true

<http://digger-soft.com/pemiluc1/index.php>

Namun sistem-sistem ini masih memiliki kekurangan beberapa fungsionalitas yang cukup penting, utamanya adalah accountability dan verifiability. Tidak ada cara mudah untuk mendeteksi apabila ada kesalahan dalam data entry, dan kalau ada, siapa yang melakukannya (supaya bisa ditindaklanjuti).

1. FUNGSIONALITAS:

1.1. Crowdsourcing entry data form C1

Ini merupakan fungsionalitas utama dari sistem yang ingin dikembangkan, yaitu memfasilitasi user untuk melakukan entry data angka-angka dari citra hasil scan form C1. Beberapa requirement yang perlu diperhatikan:

1.1.1. Penentuan TPS mana yang perlu di-entry oleh user *tidak* dilakukan oleh user itu sendiri, namun ditentukan oleh sistem. Jadi, sistem akan langsung menyuguhkan citra scan form C1 dan user tinggal memasukkan angka pada form yang telah disediakan. Penentuan form C1 yang perlu dikerjakan ini dilakukan secara random, namun berdasarkan fungsi prioritas tertentu (lihat penjelasan selanjutnya di bawah). Mengapa ditentukan sistem? Alasannya sbb.:

1.1.1.1 Kalau user diminta untuk menentukan sendiri TPS mana yang mau diisi, kecenderungannya adalah user akan menilai TPS di mana ia memilih, atau TPS dekat rumah/neighbourhood, dst. Tidak ada yang salah dengan ini, namun akibatnya mungkin banyak TPS yang tidak akan diisi oleh user, terutama TPS di daerah-daerah pelosok di

mana banyak penduduknya tidak punya akses Internet dengan mudah, padahal kita menargetkan coverage 100%.

1.1.1.2 Kalau user diminta untuk menentukan sendiri TPS yang ingin dikerjakan, hal ini menyebabkan sistem rawan untuk dimanipulasi, karena pihak yang memang bermaksud untuk sengaja manipulasi bisa kolaborasi dengan rekan untuk saling memperkuat suara yang salah. Tapi kalau pembagian TPS-nya random, sulit untuk mengkoordinasikan “serangan” seperti ini.

1.1.1.3 Kalau user user diminta untuk menentukan sendiri, akan butuh waktu yang cukup lama untuk harus drilldown Provinsi > Kabupaten > Kecamatan > Kelurahan > TPS, padahal kita mau proses yang sangat streamline dan efisien (lihat poin berikutnya)

1.1.2. Proses harus secepat dan senyaman mungkin. Ketika user mulai bekerja, langsung tampilkan scan hal. 3 dan 4 diambil dari website KPU, lalu buat form dengan field-field yang bersesuaian, lalu user tinggal memasukkan angka, pastikan dengan menekan Tab user bisa langsung pindah dari textarea ke textarea. Lalu ada tombol “Next” untuk lanjut ke TPS berikutnya (dan “Finished” kalau sudah capek/bosan untuk sesi berikut?). Dari sisi “streamline” sistem, bisa dicek ‘kompetitor’ berikut: <http://digger-soft.com/pemiluc1/index.php>

1.2 User authentication dengan Facebook/Twitter

Berbeda dengan sistem yang lain, kita mengharuskan user login dengan Facebook/Twitter. Tujuannya adalah supaya pengisian data tidak anonim, sehingga: (i) proses pengisian data menjadi akuntabel, (ii) pengguna merasa harus lebih bertanggungjawabkan isi data yang dimasukkan karena kita akan laporkan identitas siapa yang mengisi data yang mana, (iii) bisa kita gunakan untuk identifikasi user yang secara konsisten memasukkan data salah, lalu kita ban/block, (iv) bisa digunakan untuk membuat semacam gamification, mis. laporkan “Hall of Fame” crowdsourcers, etc.

1.3. Browse results: tentunya kita ingin mengimplementasikan fungsi di mana user bisa melihat hasil rekap entry data, namun untuk mencegah polemik, penampilan data ini harus disertai disclaimer mengenai data yang belum diverifikasi. Misalnya untuk TPS yang baru di-entry satu crowdsourcer, dibedakan warna/icon-nya dengan TPS yang sudah ada ‘second opinion’ dan klop, dan dibedakan lagi dengan TPS yang sudah ada ‘second opinion’ dan malah beda. Harusnya strukturnya mengikuti hirarki Dapil (seperti di website KPU).

1.4. Share to Facebook/Twitter: setelah user menyelesaikan input data untuk satu sesi, berikan opsi kepada user untuk men-share hal tersebut di sosmed (Facebook/Twitter), dengan harapan sistem ini bisa diketahui lebih banyak orang (*make it viral!*). Melihat animo masyarakat di sosmed terhadap Pilpres 2014, secara teoritis kita harusnya bisa mendapatkan jumlah crowdsourcer yang sangat banyak. Kita bisa coba kontak timses kedua kubu untuk mempromosikan halaman ini kepada relawan/pendukung. Contoh teks: “Saya baru saja

membantu mengisi data untuk N buah TPS! Anda pun bisa membantu, yuk kita kawal proses penghitungan suara Pilpres 2014!”.

1.5. Gamification: secara periodik kita bisa hitung tabel siapa user yang paling banyak sudah melakukan data entry, dan maintain “Hall of Fame” / “High score”. Ide: kita juga bisa maintain “Hall of Shame”, isinya akun-akun yang di-ban karena terbukti sengaja melakukan kesalahan data entry.

1.6. Perlu disediakan mekanisme untuk me-“raise flag” bahwa form C1 bermasalah, claim terjadi dispute/konflik, dst. Harus ditindaklanjuti oleh user admin.

1.7. Fungsionalitas untuk mengunggah dokumentasi pribadi, mis. seseorang yang foto langsung di TPS, sehingga data tambahan ini bisa ditampilkan bersebelahan dengan data riil dari KPU.

2. DATA:

Ada dua sumber data yang kita perlu dapatkan dari website KPU:

2.1 Data mengenai wilayah Propinsi > Kabupaten/Kota > Kecamatan > Kelurahan/Desa > TPS. Ini bisa diperoleh dari API di dapil.kpu.go.id dan tps.kpu.go.id. Saya juga sudah dapatkan file .sql.zip untuk dapil/wilayah, tinggal di-import ke MySQL. Di http://digger-soft.com/pemiluc1/pemilu_daerah.csv.zip juga ada (saya belum cek isinya, mungkin formatnya agak berbeda).

2.2 Data scan form C1. Idealnya sih kita tidak perlu crawl lagi datanya, tapi website yang dikembangkan langsung “tembak ke server” dengan contoh URL sebagai berikut: <http://scanc1.kpu.go.id/viewp.php?f=000000400104.jpg> -- masalahnya adalah namafile XXXXX.jpg tidak ada pemetaan langsung berdasarkan kode dapil/wilayah. Sepertinya ini disengaja supaya sulit untuk dicrawl :(Alternatifnya, kita bisa download langsung file gabungan berupa ZIP, mis: http://pilpres2014.kpu.go.id/c1.php?cmd=download&tps=8&kel_id=25826 -- nah untuk ini, nilai kel_id dan tps sama dengan ID di dapil.kpu.go.id dan tps.kpu.go.id

Selain itu, ada data yang kita perlu maintain sendiri:

2.3 Data entry dari user:

- Facebook/Twitter/Google Accounts ID
- Timestamp
- FlagRaised
- Pemilih terdaftar dalam DPT (L/P/J)
- Pemilih terdaftar dalam DPTb (L/P/J)
- Pemilih terdaftar dalam DPTk (L/P/J)
- Pemilih khusus tambahan (DPKTb) / pengguna identitas (L/P/J)

- Jumlah Surat Suara diterima
- Jumlah Surat Suara dikembalikan
- Jumlah Surat Suara tidak terpakai
- Jumlah Surat Suara digunakan
- Jumlah Suara sah
- Jumlah Suara tidak sah
- Jumlah Suara sah dan tidak sah
- Jumlah suara untuk Prabowo-Hatta
- Jumlah suara untuk Jokowi-JK
- Jumlah total suara
- Jumlah suara tidak sah

3. MODUL:

Ini hanya usulan bagaimana kira-kira kita bisa breakdown sistem supaya bisa dikerjakan secara paralel oleh beberapa pihak:

Fase persiapan:

- 3.1 Modul data crawler (jika dibutuhkan?)
- 3.2 Modul database

Fase berikutnya:

- 3.3 Modul authentication dengan Facebook / Twitter
- 3.4 Modul crowdsourcing
- 3.5 Modul result browser
- 3.6 Modul admin / conflict resolution
- 3.7 Modul share to Facebook / Twitter

4. FUNGSI PRIORITAS:

Dalam meng-assign TPS untuk dikerjakan, sistem memilih sebagai berikut:

- Selama masih ada TPS yang belum dikerjakan sama sekali, utamakan. Jika ada >1, pilih secara acak.
- Selama masih ada TPS yang disputed (sudah dikerjakan 2 crowdsourcer dan hasilnya beda), utamakan. Jika ada >1, pilih secara acak.
- Selama masih ada TPS yang baru dikerjakan oleh satu orang, utamakan. Jika ada >1, pilih secara acak.

Catatan: tulisan ini dibuat dengan sangat cepat dan dalam kondisi ngantuk :-) Mohon masukan / saran / kritikan, dan mohon bantuan siapa yang bisa mengimplementasikannya :-)