

APLIKASI QUICKCOUNT PEMILIHAN PRESIDEN RI MENGUNAKAN TEKNOLOGI MOBILE

Juwairiah¹⁾, Dessyanto Boedi Prasetyo²⁾ Fifi Tri Guntari³⁾

^{1,2,3)}Jurusan Teknik Informatika UPN "Veteran" Yogyakarta

Jl. Babarsari no 2 Tambakbayan 55281 Yogyakarta Telp (0274)-485323

e-mail : juwai_riah@yahoo.com

Abstrak

Permasalahan yang sering terjadi dalam Pemilu adalah karena biasanya perhitungan suara secara manual membutuhkan waktu yang lama, bisa mencapai beberapa minggu atau bahkan lebih dari sebulan. Hal ini dapat menyebabkan kesimpangsiuran dan ketidakpastian informasi siapa yang menang; siapa yang berhak duduk dikursi DPR (dalam pemilu legislatif) atau siapa yang menjadi presiden (dalam pemilu presiden). Penggunaan handphone dapat dimanfaatkan sebagai media pengiriman data perhitungan suara dari tiap tempat pemungutan suara (TPS) yang tersebar di seluruh Indonesia. Perhitungan suara ini yang dinamakan Quick Count merupakan salah satu metode yang berguna untuk mempercepat proses pemungutan suara dengan memilih sejumlah TPS yang secara acak (random). Pengumpulan data hasil perhitungan suara dilakukan oleh relawan di setiap TPS yang sudah ditentukan sebelumnya melalui pemantauan langsung saat pemungutan dan perhitungan suara di seluruh TPS yang ada. Pemantau(relawan) mencatat informasi, termasuk hasil perhitungan suara yang ada, dan melaporkan hasil tersebut ke pusat pengumpulan suara data (server) melalui handphone. Server dapat melakukan perhitungan suara total yang masuk, sehingga prediksi hasil pemilu dapat diketahui secara cepat oleh masyarakat melalui web.

Keyword : Pemilu, TPS sampel, Quickcount, handphone, server

1. PENDAHULUAN

Handphone merupakan salah satu *wireless device* yang sudah tidak asing lagi di kalangan masyarakat karena sudah menjadi kebutuhan bagi setiap orang. Sebagai perangkat komunikasi, *handphone* menawarkan banyak keunggulan. Bentuknya yang kecil dan ringan serta kepraktisannya yang mudah dibawa kemana saja membuat *handphone* begitu digemari dan mengalami perkembangan yang pesat. Beberapa *fitur* yang tersedia dalam perangkat ini menawarkan berbagai kemudahan bagi pengguna *handphone*. *Handphone* bukan hanya berfungsi sebagai alat komunikasi tapi juga sebagai pemberi dan pengiriman informasi. *Handphone* juga dapat memberikan fasilitas koneksi ke halaman *web* melalui akses GPRS (*General Packet Radio System*).

Dengan adanya beberapa keunggulan yang ditawarkan oleh *handphone* tersebut maka banyak sekali aplikasi yang bisa dibuat dengan menggunakan perangkat ini. Dalam permasalahan perhitungan suara Pemilu yang biasanya membutuhkan waktu beberapa hari atau bahkan beberapa minggu, penggunaan *wireless device* ini dapat dimanfaatkan sebagai media pengiriman data perhitungan suara dari tiap tempat pemungutan suara (TPS) yang tersebar di seluruh Indonesia. Perhitungan suara ini yang dinamakan *Quick Count* merupakan salah satu metode yang berguna untuk memantau proses pemungutan suara, pengumpulan data hasil perhitungan suara dilakukan oleh ratusan bahkan ribuan relawan melalui pemantauan langsung saat pemungutan dan perhitungan suara di seluruh TPS yang ada. Pemantau mencatat informasi, termasuk hasil perhitungan suara yang ada, dan melaporkan hasil tersebut ke pusat pengumpulan suara data (*Server*) melalui *handphone*.

Quick count adalah perhitungan secara cepat hasil pemilihan umum dengan menggunakan Tempat Perhitungan Suara (TPS) sampel. Dengan *quick count*, hasil perhitungan suara biasa diketahui dua sampai tiga jam setelah perhitungan suara di TPS ditutup. Kecepatan ini bisa didapat karena dalam *quick count* tidak menghitung suara dari semua TPS, cukup dengan sampel TPS saja. Jika penarikan sampel dilakukan dengan benar, prosedur pencatatan dilakukan dengan tepat, meski hanya memakai sampel TPS, hasil *quick count* hampir sama dengan hasil Pemilu.

Kecepatan dan kepastian hasil siapa yang menang dalam Pemilu penting diketahui secepat mungkin oleh publik. KPU/KPUK biasanya akan mengumumkan hasil Pemilu beberapa hari atau bahkan beberapa minggu kemudian. Selama proses itu, tensi politik akan terus tinggi. *Quick Count* berguna untuk mendinginkan tensi politik itu agar publik bisa mulai kehidupan sehari-hari kembali.

Oleh karena itu maka perlu dibuat aplikasi *quick count* pemilihan umum berbasis *mobile* atau lebih dikenal J2ME (*Java2 Micro Edition*), sistem yang dibuat bersifat *client-server*. Dengan adanya aplikasi ini relawan dapat mengakses aplikasi untuk mengirim data hasil perhitungan suara pemilu dan melihat hasil perhitungan suara secara *real time*. Sedangkan masyarakat umum dapat juga mengakses prediksi perhitungan suara pemilu melalui *web* tanpa harus menunggu lama untuk mengetahui perhitungan hasil suara. Salah satu kemudahan yang ditawarkan perangkat *handphone* adalah layanan GPRS. Fasilitas tersebut memungkinkan pengguna dapat mengakses halaman *web* dalam *handphone* untuk mendapatkan informasi yang diinginkan.

2. TINJAUAN PUSTAKA

Penelitian aplikasi *quick count* pemilu sudah diterapkan oleh Lingkaran Survei Indonesia (LSI) dalam memprediksi hasil perhitungan suara. Aplikasi *quick count* yang digunakan oleh Lingkaran Survei Indonesia adalah berbasis SMS *gateway*. Kendala yang sering terjadi adalah ketika Relawan berganti nomor ponsel, maka Relawan tidak bisa mengirim data hasil perhitungan suara. Relawan diharuskan untuk mengingat *format sms* dalam pengiriman data hasil perhitungan suara.

Dalam penelitian ini, metodologi pengembangan sistem yang digunakan adalah *Guidelines for Rapid APPLication Engineering* (GRAPPLE). Aplikasi yang akan dibangun adalah *quick count* berbasis *mobile* untuk Pemilihan Presiden Republik Indonesia. Aplikasi yang dibuat merupakan pengiriman hasil perhitungan suara yang bersifat *client-server* yang diakses melalui GPRS (*embedded* pada hp) yang dapat mengirim data hasil perhitungan suara ke *server*.

Berikut dapat dilihat perbandingan cara perhitungan yang dilakukan oleh KPU/KPUD dan perhitungan yang dilakukan oleh Lingkaran Survei Indonesia untuk mencapai hasil akhir dari proses perhitungan suara suatu pemilu.

Tabel 1. Perbandingan proses perhitungan suara KPU/KPUD dengan LSI

PROSES PERHITUNGAN SUARA	KPU/KPUD	LSI
Jumlah TPS	Semua TPS yang ada	TPS yang dipilih secara <i>random</i>
Petugas	KPU/KPUD	Relawan LSI
Parameter	Berdasarkan rekaman data yang telah dihitung dari TPS, lalu Kelurahan, Kecamatan, Kabupaten, Propinsi dan Pusat	Berdasarkan data yang diinput oleh relawan yang dikirim ke server pusat.
Waktu	Beberapa hari bahkan beberapa minggu setelah proses perhitungan suara selesai	2-3 jam setelah proses perhitungan suara selesai

3. Quick Count

Aspek paling penting dalam *quick count* adalah penarikan sampel TPS. Jika TPS yang diambil tidak *representative*, bisa dipastikan prediksi *quick count* akan gagal dalam memprediksi perolehan suara sesungguhnya dalam Pemilu. Unit analisis (pengamatan) *quick count* adalah TPS. Meski demikian perhitungan awal didasarkan pada jumlah pemilih. Karena itu penentuan jumlah TPS diawali terlebih dahulu dari perhitungan sampel pemilih. Dari sampel pemilih itu baru bisa diprediksikan beberapa banyak TPS yang akan diambil.

Penentuan besar sample berkaitan dengan seberapa jauh kita menginginkan ketelitian dari suatu sampel. Menentukan besar sample tergantung kepada 4 hal berikut : keragaman (*variasi*) dari populasi, batas kesalahan sampel yang dikehendaki (*sampling error*), interval kepercayaan (*confidence interval*) dan jumlah populasi.

3.1. Menentukan Jumlah TPS (Tempat Pemungutan Suara) Sampel

Untuk memahami langkah-langkah Quickcount, diberikan contoh sebagai berikut :
 Diketahui total pemilih berjumlah 306.280 orang dan tersebar di 1.254 TPS. *Quick count* dirancang menggunakan *sampling error* 1%, dengan tingkat kepercayaan 99%. Proporsi populasi beragam (50:50). Rata-rata pemilih per TPS 244. Dengan demikian dapat dilakukan teknik pemilihan TPS sebagai berikut :

1. Menentukan jumlah sampel pemilih

$$n = \frac{Z^2 \cdot [p(1-p)] \cdot N}{Z^2 \cdot [p(1-p)] + (N-1) \cdot E^2} \dots\dots\dots(1)$$

Keterangan :

- n = Besar sampel pemilih
- Z = Mengacu kepada nilai distribusi normal. Jika tingkat kepercayaan yang dipakai 90%, nilai z adalah: 1,65. Tingkat kepercayaan 95%, nilai z adalah 1,96. Sedangkan tingkat kepercayaan 99%, nilai z adalah 2,58.
- P(1-p) = *Variasi* populasi. *Variasi* populasi di sini dinyatakan dalam bentuk proporsi. Proporsi dibagi kedalam dua bagian dengan total 100% (atau 1). Jika populasi diasumsikan heterogen, maka p = 0,5
- E = Kesalahan sampel yang dikehendaki (*sampling error*). Misalnya : 2% atau 0,02.
- N = Jumlah populasi

Contoh :

$$n = \frac{(2,58)^2 \cdot [0,5(1 - 0,5)] \cdot 306280}{(2,58)^2 \cdot [0,5(1 - 0,5)] + (306280 - 1) \cdot 0,01^2} = 15.783$$

Maka harus diambil jumlah sampel pemilih sebesar 15.783

2. Menentukan jumlah TPS sampel (t)

$$t = \frac{\text{Jumlah pemilih yang dibutuhkan}}{\text{Rata - rata pemilih per TPS}} \dots\dots\dots(2)$$

Contoh : $t = \frac{15783}{244} = 65$

Maka TPS sampel yang dibutuhkan adalah 65 TPS

3. Memilih secara *random* 65 TPS dari 1.254 TPS

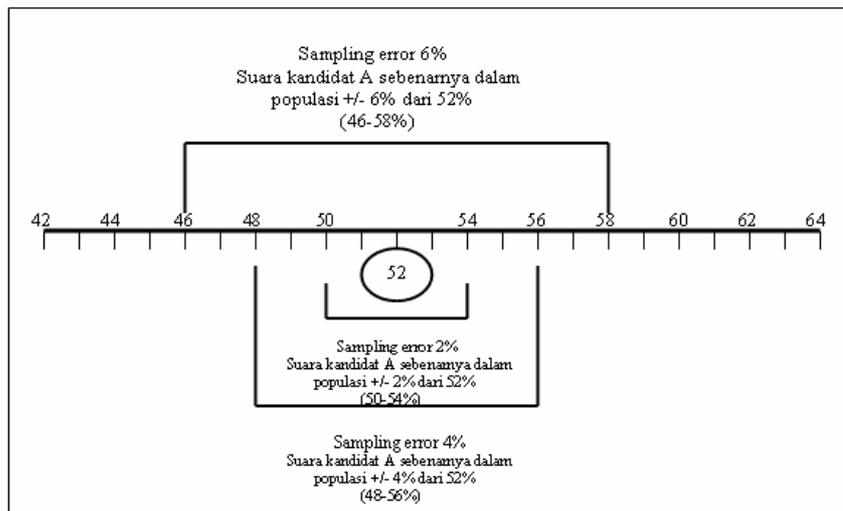
Langkah dalam melakukan acak adalah sebagai berikut :

- a. Menentukan interval sampel
 Interval sampel diperoleh dengan membagi jumlah populasi TPS dengan jumlah TPS sampel. Ini berarti, interval sampel adalah : $1.254 / 65 = 19.29$
 Dibulatkan menjadi 19
- b. Memilih secara *random* sampel awal / pertama
 Sampel pertama dipilih dengan melakukan acak (*random*) angka 1 hingga 19. Sebagai ilustrasi, misalnya ditemukan angka 3. Dengan demikian, angka 3 ini menjadi sampel awal. TPS dengan urutan nomor 3 menjadi sampel awal (pertama) *quick count*.
- c. Memilih secara acak (random) sampel selanjutnya
 Sampel kedua dan seterusnya bisa diperoleh dengan menambah secara sistematis 19. Sampel kedua misalnya adalah 22 (3 + 19), sampel ketiga 41 (22 + 19), dan seterusnya hingga ditemukan 67 TPS sebagai sample, yaitu : TPS 3, TPS 22, TPS 41, TPS 60, TPS 79, dan seterusnya.

3.2. Tingkat Kesalahan Yang Dikehendaki (*Sampling Error*)

Karena *quick count* memakai sampel, selalu ada kesalahan (*sampling error*). Semakin besar sampel, semakin kecil angka *sampling error*. Sebaliknya semakin kecil sampel, semakin besar angka *sampling error*. Karena *sampling error* menunjukkan derajat akurasi, penyelenggaraan *quick count* harus menentukan terlebih dahulu berapa tingkat kesalahan yang dikehendaki (*sampling error*) dari *quick count* yang akan dilakukan. Pilihan *sampling error*, menentukan derajat ketelitian dari *quick count*. Semakin kecil *sampling error*, semakin akurat hasil *quick count* dalam memprediksi populasi (nilai parameter).

Karena tujuan *quick count* Pemilu adalah meramal perolehan suara dari kandidat, maka *sampling error* yang dipakai haruslah sekecil mungkin. Hanya dengan *sampling error* yang kecil, prediksi bisa dilakukan dengan lebih akurat dan presisi. Paling tidak *sampling error* yang dipakai maksimal 1%. Jika memungkinkan, gunakan *sampling error* yang lebih kecil, misalnya 0,5%



Gambar 1. *Sampling error*

3.3. Tingkat Kepercayaan

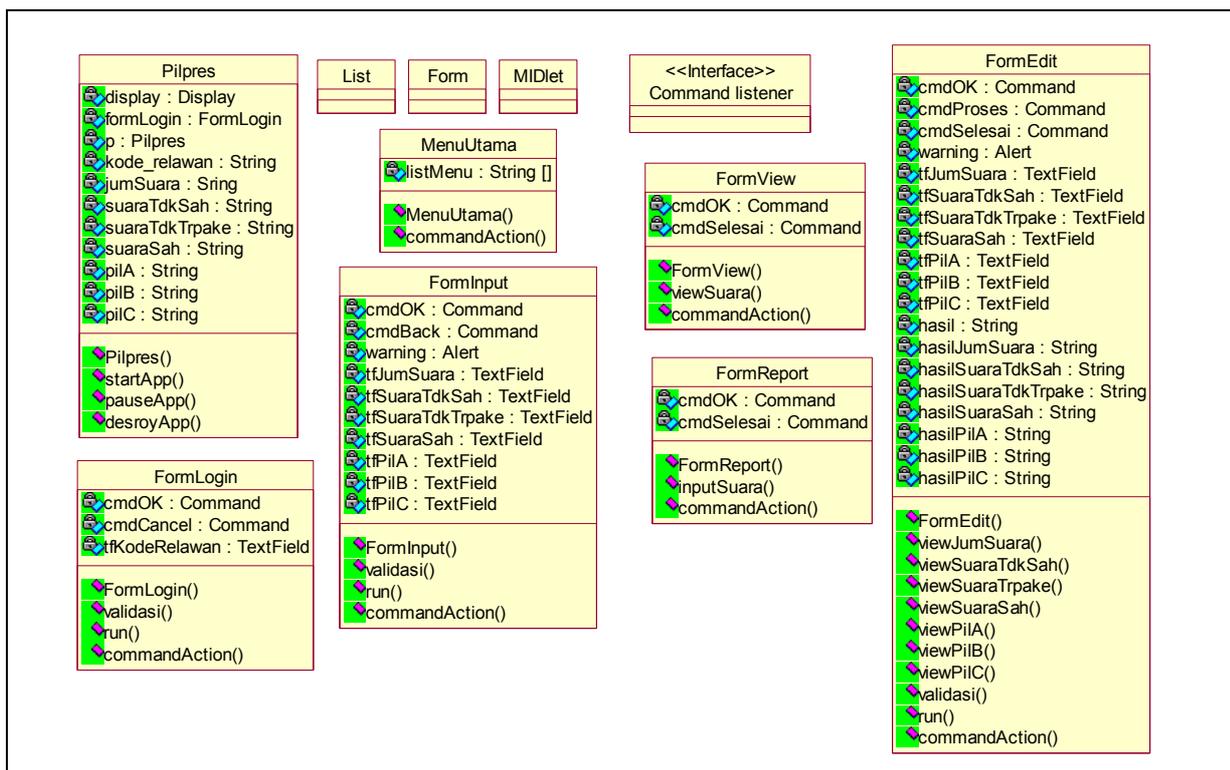
Tingkat kepercayaan berhubungan dengan seberapa besar taksiran atau estimasi dari sampel berlaku untuk populasi. Tingkat kepercayaan seperti dua sisi mata uang dengan *sampling error*. Angka *sampling error* mengestimasi berapa nilai populasi, sementara tingkat kepercayaan memastikan berapa besar estimasi itu berlaku di dalam populasi.

Karena tujuan dari *quick count* Pemilu adalah meramal perolehan suara dari kandidat atau partai, maka sebisa mungkin penyelenggara *quick count* menggunakan tingkat kepercayaan sebesar mungkin, msalnya 90%, 95%, atau 99%. Tingkat kepercayaan yang kerap dipakai dalam *quick count* adalah 99%. Tingkat kepercayaan 99% berarti *probabilitas* kemungkinan hasil sampel sama dengan populasi adalah 99%. Kemungkinan salah adalah sebesar 1%.

4. HASIL DAN PEMBAHASAN

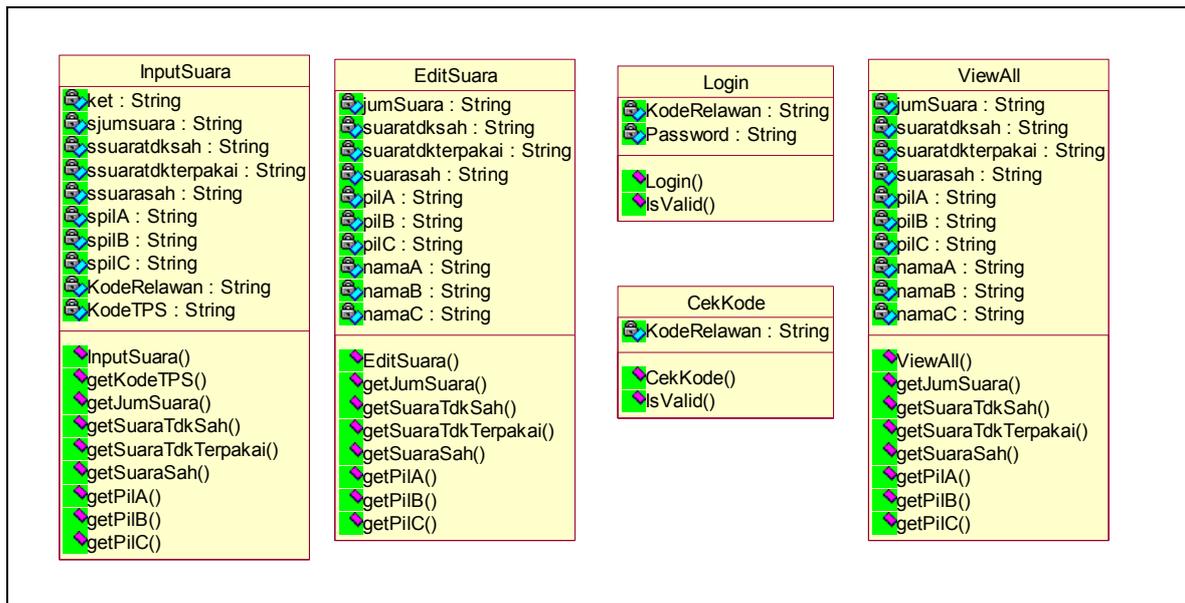
Aplikasi *quick count* pemilihan Presiden Republik Indonesia berbasis *mobile* dibangun menggunakan prinsip *client server* dimana *client* (menggunakan teknologi J2ME) akan mengirimkan data menggunakan *handphone* sedangkan disisi *server* akan menggunakan teknologi JSP. Komunikasi antara *client* dilakukan melalui koneksi *http* via GPRS. Data perhitungan hasil suara yang dikirimkan ke *server* berasal dari data yang diinput oleh relawan, data perhitungan hasil suara yang diinputkan meliputi data jumlah suara, suara tidak sah, suara tidak terpakai, suara sah, serta hasil perolehan suara yang didapat dari tiap kandidat. Admin akan bertanggung jawab terhadap data-data yang menyangkut data capres, data relawan serta data TPS. Sedangkan sistem *database* akan mengelola data perhitungan hasil suara yang diinputkan oleh relawan, sehingga masyarakat umum dapat mengakses halaman web untuk mengetahui prediksi perhitungan hasil suara pemilu dari tiap kandidat.

Objek-objek yang terlibat beserta atribut dan operasi setiap objeknya dibangun dalam diagram *high-level-class*. Diagram *high-level-class* pada *handphone* ini dapat dilihat pada gambar berikut :



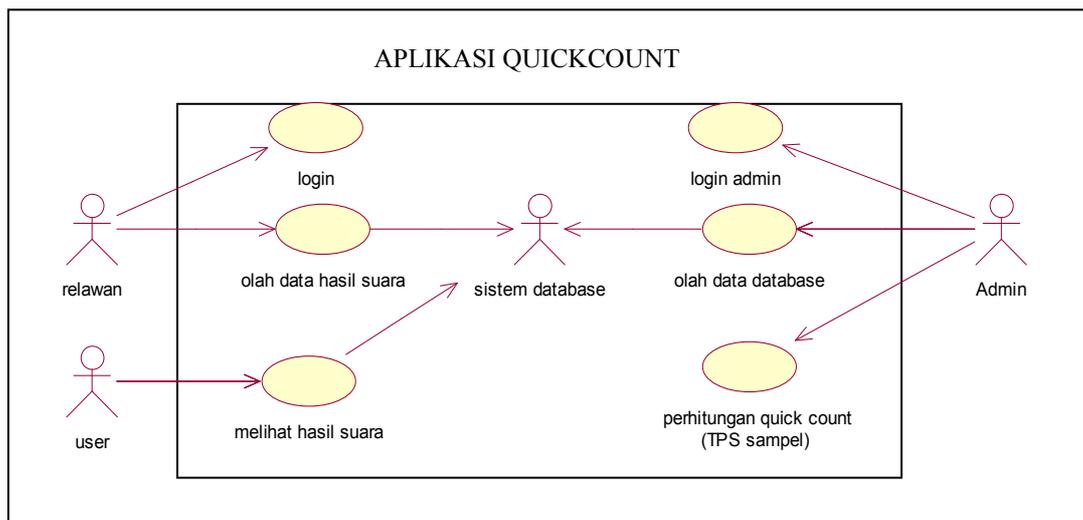
Gambar 2. Diagram *high-level-class* pada *handphone*

Sedangkan gambar diagram *high-level-class* pada *server* dapat dilihat pada gambar berikut ini :



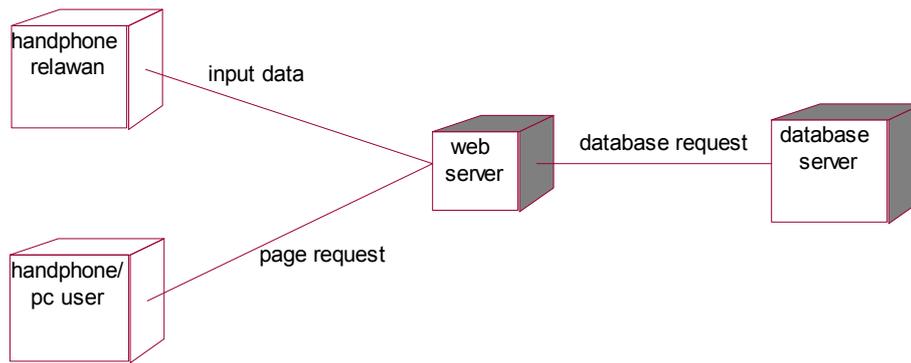
Gambar 3. Diagram *high-level-class* pada *server*

Diagram *use case* nya sebagai berikut :



Gambar 4. Diagram *Use Case*

Aplikasi ini memiliki empat aktor yaitu Relawan, Admin, Sistem *Database* dan *User* (masyarakat umum). Relawan bertindak sebagai pengirim data perhitungan hasil suara yang meliputi jumlah suara, suara tidak sah, suara tidak terpakai, suara sah dan hasil perolehan suara yang didapat dari tiap kandidat dengan mengakses halaman aplikasi pada *handphone*. Admin bertindak sebagai pengolah data : data capres, data relawan, data admin, dan data TPS yang diinputkan kedalam sistem database, untuk sistem database bertindak sebagai pengolah data perhitungan hasil suara yang telah diinputkan oleh relawan dari tiap TPS terpilih. Sedangkan untuk *user*/masyarakat umum bertindak meminta informasi hasil akhir dari prediksi perhitungan hasil suara pemilihan Presiden Republik Indonesia dengan mengakses halaman *web* yang tersedia. Data-data perhitungan hasil suara tiap TPS akan diinputkan oleh relawan dan dikirim ke sistem *database* melalui koneksi *http via GPRS*, kemudian sistem *database* yang akan mengolah data perhitungan hasil suara dari seluruh TPS terpilih. Sebelum melakukan akses terhadap aplikasi *quick count* tersebut relawan diwajibkan untuk melakukan *login* sesuai dengan *username* dan *password* yang telah ditentukan sesuai dengan daerah TPS dimana relawan ditempatkan.



Gambar 5. Deployment diagram

Dalam aplikasi ini, terbagi menjadi 2 kelas, yaitu : kelas-kelas aplikasi pada handphone dan kelas-kelas aplikasi pada server. Kelas utama aplikasi pada handphone yaitu kelas Pilpres yang paling berperan dalam aplikasi. Kelas ini merupakan MIDlet pada aplikasi. Kelas yang berperan dalam proses *login* adalah kelas *Login*. Kelas ini akan melakukan *validasi* dan mengecek apakah *username* dan *password* yang dimasukkan oleh relawan *valid* atau tidak. Kelas-kelas yang berperan dalam antarmuka utama aplikasi yang terdiri dari kelas Pilpres, kelas *Login*, *MenuUtama*, *FormInput*, *FormView*, *FormEdit*, dan *FormReport*. *MenuUtama* yang berperan menampilkan antarmuka utama aplikasi *Quick Count*. Sedangkan kelas yang lain merupakan bagian dari tampilan menu-menu yang ada. *MenuUtama* akan tampil jika relawan telah berhasil melakukan *login*.

Tabel 2. Kelas-kelas Aplikasi pada *handphone*

Kelas	Deskripsi
1. Kelas Pilpres	Kelas ini bertugas untuk menampilkan <i>form login</i> akan ditampilkan setelah muncul alert yang dapat dikatakan sebagai <i>splash screen</i> yang pertama dan setelah itu menampilkan <i>form login</i> , pada kelas inilah yang mengandung <i>method main</i> , dimana aplikasi pertama kali akan mengeksekusi kelas ini.
2. Kelas Login	Kelas ini berfungsi untuk melakukan <i>login</i> dimana <i>username</i> dan <i>password</i> akan dicek/ <i>validasi username</i> dan <i>password</i> dilakukan pada kelas ini. Apabila <i>login</i> sukses maka akan ditampilkan menu utama.
3. Kelas MenuUtama	Kelas ini merupakan kelas yang paling berperan dalam aplikasi, bertugas mengatur <i>link</i> akses dari menu-menu yang tersedia. Terdiri dari tiga menu utama yang dapat diakses untuk keperluan pengolahan data hasil suara.
4. Kelas FormInput	Kelas ini berfungsi untuk melakukan <i>input</i> data yang diinputkan oleh relawan. Data yang diinputkan berupa angka hasil perhitungan akhir dari tiap TPS terpilih.
5. Kelas FormView	Kelas ini berfungsi untuk melihat hasil perhitungan suara secara keseluruhan yang telah diinputkan oleh masing-masing relawan di tiap TPS terpilih. Data yang ditampilkan akan muncul melalui <i>lcd</i> ponsel melalui koneksi <i>via http</i> .
6. Kelas FormEdit	Kelas ini memiliki fungsi ketika relawan melakukan kesalahan dalam proses penginputan data yang telah diinputkan sebelumnya, dalam kelas ini relawan dapat mengedit data dan mengupdatenya langsung.
7. Kelas FormReport	Kelas ini berfungsi untuk menampilkan data yang telah diinputkan sebelumnya oleh relawan yang akan melakukan pengeditan data hasil perhitungan suara.

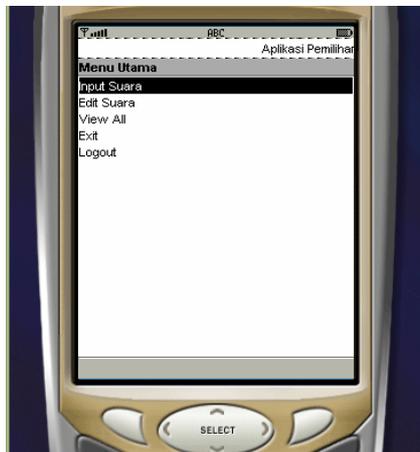
Tampilan Antarmuka di Handphone Untuk Relawan



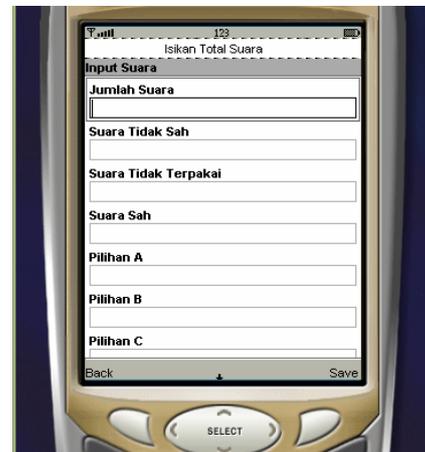
Gambar 6.a Tampilan Splash Screen



Gambar 6.b Tampilan Menu Login Relawan



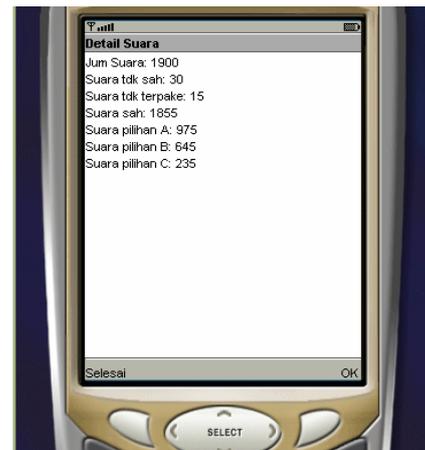
Gambar 6.c Tampilan Menu Utama



Gambar 6.d Tampilan Form Input Suara



Gambar 6.e Tampilan Form Report Suara



Gambar 6.f Tampilan Form View Suara pada Handphone

Sedangkan kelas-kelas aplikasi pada server sebagai berikut :

Tabel 3. Kelas-kelas Aplikasi Pada Server

Kelas	Deskripsi
Kelas Login	Kelas yang melakukan <i>validasi</i> terhadap <i>username</i> dan <i>password</i> pada saat relawan melakukan <i>login</i> . Kelas <i>Login</i> akan mencocokkan data yang dimasukkan oleh relawan apakah sudah sesuai dengan data dalam <i>database</i> .
Kelas CekKode	Kelas <i>CekKode</i> merupakan kelas untuk pengecekan kode relawan yang sebelumnya telah melakukan <i>login</i> terlebih dahulu.
Kelas InputSuara	Kelas ini berfungsi untuk menginputkan data hasil perhitungan suara yang diinputkan oleh relawan.
Kelas ViewAll	Pada kelas <i>ViewAll</i> berfungsi untuk menampilkan seluruh data hasil perhitungan suara yang sudah terinputkan.
Kelas EditSuara	Kelas <i>EditSuara</i> berfungsi untuk <i>mengedit</i> hasil suara yang sebelumnya telah <i>diinputkan</i> oleh relawan, tetapi dapat <i>diupdate</i> melalui kelas <i>EditSuara</i> .

Tampilan Antar Muka Web Untuk Admin



Gambar 7. Tampilan Olah Data Relawan



Gambar 8. Tampilan Olah Data Kandidat



Gambar 9. Tampilan Olah Data TPS

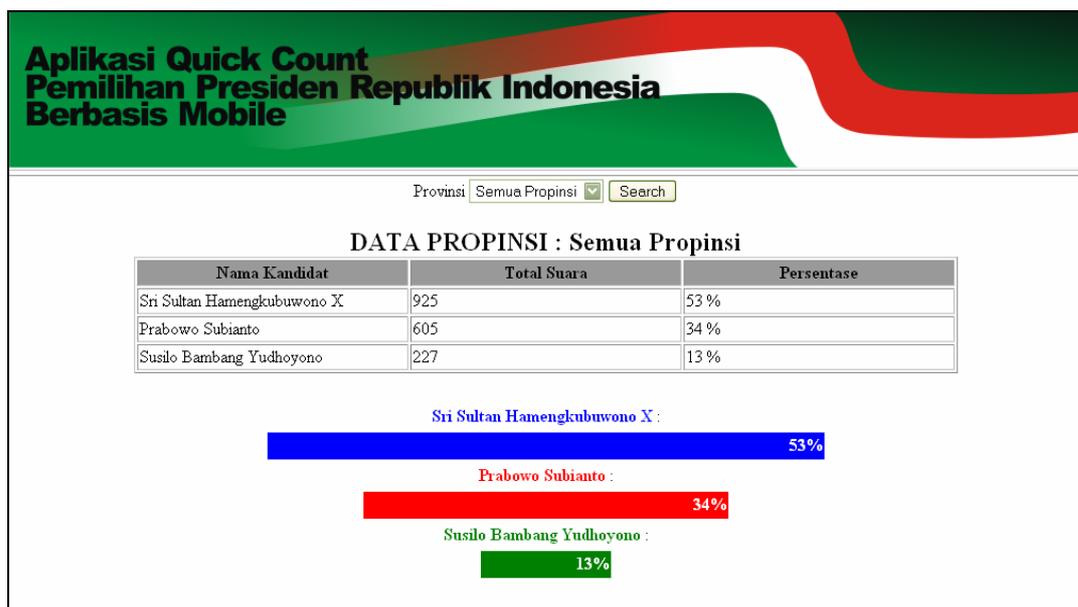


Gambar 10. Tampilan Perhitungan TPS Sampel (Quickcount)



Gambar 11. Tampilan TPS Sampel yang terpilih

Tampilan Antarmuka Untuk User



Gambar 12. Tampilan Antamuka Hasil QuickCount untuk User

5. KESIMPULAN

Dari penelitian yang dilakukan telah berhasil dibuat suatu aplikasi *Quick Count* Pemilihan Presiden RI Berbasis *Mobile*. Aplikasi ini dapat digunakan dari dua sisi yaitu sisi *client* dan sisi *server*. Disisi *client*, aplikasi dapat digunakan untuk proses pengiriman data hasil perhitungan suara yang telah dimasukkan melalui *handphone* dengan menggunakan koneksi GPRS dari tiap TPS terpilih yang diinputkan oleh Relawan. Aplikasi juga dapat digunakan untuk melihat prediksi hasil perhitungan suara oleh user melalui halaman *web*. Sedangkan disisi *server* admin dapat melakukan olah data untuk data Relawan, Kandidat, TPS, Admin, Perhitungan TPS Sampel melalui *web*. Kelebihan dari sistem ini adalah dapat memudahkan Relawan dalam proses pengiriman data hasil suara tanpa harus mengingat *format* SMS seperti pada sistem sebelumnya. Aplikasi ini juga membantu *user* atau masyarakat untuk mengetahui prediksi perhitungan suara Pemilihan Presiden RI secara cepat.

6. DAFTAR PUSTAKA

- Eriyanto, 2006, Panduan Penyelenggaraan *Quick Count*, Penerbit Lingkaran Survei Indonesia, Jakarta.
- Hartanto, Antonious Aditya, 2003, *Java 2 Micro Edition Mobile Interface Device Programming*, Penerbit PT Elex Media Komputindo, Jakarta
- Hartono, Jogyanto, 2000, *Pengenalan Komputer*, Penerbit Andi, Jogjakarta.
- Hermawan, Beny, 2004, *Menguasai Java 2 dan Object Oriented Programming*, Penerbit Andi, Jogjakarta.
- Kadir, Abdul, 2004, *Dasar Pemrograman Web Dinamis Dengan JSP (Java Server Pages)*, Penerbit Andi Offset, Jogjakarta.
- Mehrota, Asha, 1997, *GSM System Engineering*, Artech House, Inc, Boston London.
- Nugroho, Adi, 2005, *Rational Rose Untuk Pemodelan Berorientasi Objek*. Penerbit Informatika Bandung, Bandung.
- Suhendar, dkk, 2002, *Visual Modelling Menggunakan UML dan Rational Rose*, Penerbit Informatika Bandung, Bandung.
- Sunarfrihantono, Bimo, 2003, *PHP dan MySQL untuk Web*, Penerbit Andi, Jogjakarta
- Wicaksono, Adi, 2002, *Pemrograman Aplikasi Wireless dengan Java*, Penerbit PT Elex Media Komputindo, Jakarta.