

# Implementasi *Metode Multistage Random Sampling* untuk Aplikasi *Quick Count* pada Pilkada Kota Palembang Berbasis Java Mobile

Purwita Sari<sup>1</sup>, Lucky Indra Kesuma<sup>2</sup>, Ahmad Rifai<sup>3</sup>

<sup>1</sup> Program Studi Manajemen Informatika, Fakultas Ilmu Komputer, Universitas Sriwijaya

<sup>2</sup> Jurusan Sistem Informas,i Fakultas Ilmu Komputer, Universitas Sjahyakti

<sup>3</sup> Jurusan Sistem Informasi, Fakultas Ilmu Komputer, Universitas Sriwijaya  
Palembang, Indonesia

Email: [wita@ilkom.unsri.ac.id](mailto:wita@ilkom.unsri.ac.id), [luckyindra25@gmail.com](mailto:luckyindra25@gmail.com), [rifai.bae@gmail.com](mailto:rifai.bae@gmail.com)

**Abstrak**—Pilkada diselenggarakan dalam rangka memilih calon pemimpin daerah atau kepala daerah secara langsung oleh rakyat yang dilakukan di daerah provinsi dan kabupaten atau kota *Quick count* adalah proses perhitungan hasil perolehan suara yang terpilih secara random dipuluhan atau raturan TPS. Keberhasilan *quick count* sangat bergantung pada proses penarikan sampel. Metode dalam pengembangan sistem adalah FAST sedangkan metode penarikan sampel yang digunakan adalah metode *Multistage Random Sampling*. Aplikasi ini dibangun bersifat client-server, dimana dari sisi server berbasis web sedangkan di sisi client berbasis java mobile. Aplikasi ini dapat menyajikan informasi hasil pilkada secara *real time*, dapat memprediksi hasil pilkada secara cepat sehingga dapat diperoleh secara akurat mendekati bahkan sama dengan hasil yang sebenarnya.

**Kata Kunci**—*pilkada, quick count, metode sampling*

## I. PENDAHULUAN

Indonesia merupakan negara demokrasi yang melakukan pergantian kepemimpinan secara periodik untuk menjunjung tinggi kedaulatan rakyat. Oleh sebab itu, keinginan aspirasi dan kepentingan rakyat diperjuangkan dengan cara terpilihnya wakil rakyat berdasarkan suara terbanyak[1].

Berdasarkan Undang-Undang Nomor 32 Tahun 2004 tentang Pemerintah Daerah, dalam rangka menyelenggarakan pemilihan calon kepala daerah/pimpinan daerah yang dipilih secara langsung oleh rakyat yang dilaksanakan oleh daerah provinsi dan kabupaten/kota. Kepala daerah dan wakil kepala daerah untuk provinsi adalah gubernur dan wakil gubernur, untuk kabupaten adalah bupati dan wakil bupati, serta untuk kota adalah walikota dan wakil walikota. Masih banyak kekurangan dalam proses dilaksanakannya Pilkada di Indonesia. Seringkali pengumuman kepada publik tidak dapat segera dilaksanakan yang disebabkan oleh dalam menghitung perolehan suara oleh KPU dibutuhkan waktu yang tidak sedikit. Dengan demikian, dapat terjadi peluang ketidakpastian atau kekisringan politik yang dapat menjadi ancaman nasional dalam negara kesatuan Republik Indonesia. Pilkada di Indonesia sangat rentan akan tindak kecurangan, hal ini terjadi karena hasil perolehan suara yang merupakan indikator utama dalam proses dilaksanakannya Pilkada dimanipulasi oleh oknum yang tidak bertanggung jawab [2]. Pilkada yang akan dibahas pada penelitian ini adalah

pilkada Kota Palembang (pemilihan walikota). Namun sebagai studi kasus pembahasan kan ditekankan untuk KPU Kota Palembang dalam mendukung pelaksanaan Pilkada Kota Palembang.

Metode *quick count* merupakan salah satu metode/ cara yang digunakan untuk mengetahui hasil perhitungan suara pada pilkada dengan melakukan teknik pengambilan *sampling*. *Quick count* adalah proses perhitungan hasil perolehan suara yang terpilih secara random dipuluhan atau raturan TPS. *Quick count* bersumber dari fakta bukan berdasarkan pendapat untuk memprediksi hasil Pilkada. Dengan adanya *quick count* dapat dijadikan perkiraan dan data dukung untuk proses verifikasi hasil perhitungan resmi dari KPU. Dengan menggunakan *quick count* lebih efisien dari sisi biaya jika dibandingkan dengan dilakukannya perhitungan manual di seluruh TPS. Banyak sampel membuktikan bahwa *quick count* dapat menghilangkan rasa tidak percaya terhadap cara kerja panitia pemilu atau pilkada dan untuk legalitas atas proses pilkada atau pemilu tersebut[3].

Keberhasilan *quick count* sangat tergantung pada bagaimana proses pengambilan sampel yang mana pada sampel terpilih akan mewakili suara pemilih yang kemudian akan digunakan sebagai dasar estimasi Pilkada. Penarikan sampel yang baik dan benar akan menjadi dasar yang kuat dalam mewakili karakteristik populasi [4].

Metodologi statistik dan penarikan sampel yang ketat serta implementasi secara konsisten di lapangan dapat menghasilkan estimasi *quick count* yang akurat. Melakukan identifikasi terhadap berbagai faktor yang berdampak pada distribusi suara dalam populasi suara pemilih merupakan tolak ukur keberhasilan *quick count*. Apabila Pilkada dilakukan tanpa kecurangan, keakuratan *quick count* dapat dibandingkan dengan hasil resmi KPU. Sebaliknya jika Pilkada dilakukan penuh kecurangan, maka hasil *quick count* yang dapat dikatakan kredibel walaupun hasilnya berbeda dengan hasil resmi KPU. Selain dengan *quick count* biasanya pemantauan juga dilakukan dengan menggunakan metode penarikan sampel secara acak [4].

Dari hasil uraian di atas, dapat diambil kesimpulan bahwa pelaksanaan Pilkada memerlukan sebuah aplikasi hitung cepat atau *quick count* berbasis java mobile yang *real time*. Aplikasi ini dibangun bersifat *client-server*, dimana

dari sisi server berbasis web sedangkan di sisi client berbasis java mobile. Aplikasi ini dapat membantu relawan dalam menyampaikan laporan perhitungan suara Pilkada dalam waktu yang lebih cepat. Selain itu publik dapat juga menerima atau melihat informasi secara *real time* mengenai hasil perhitungan suara melalui web yang sudah disediakan.

Penelitian ini bertujuan menimplementasikan metode multistage random sampling untuk aplikasi quick count pada Pilkada berbasis java mobile pada studi kasus Komisi Pemilihan Umum (KPU) Kota Palembang.

## II. TINJAUAN PUSTAKA

### A. Quick Count

*Quick Count* adalah proses perhitungan hasil perolehan suara yang terpilih secara random dipuluhan atau ratusan TPS[5]. *Quick Count* bersumber dari fakta bukan berdasarkan pendapat untuk memprediksi hasil Pilkada. Dengan demikian, *quick count* sangat berbeda dengan yang dinamakan *exit poll* yaitu survey yang dilakukan segera setelah pemilih meninggalkan TPS [4]. Keberhasilan dilaksanakannya *quick count* dapat terjadi oleh beberapa faktor antara lain[6]:

1. Terdapatnya akses ke TPS, independensi dan kredibilitas, dukungan komunikasi data jaringan dan diakar rumput (*grass root*)
2. Dilakukan pelatihan terhadap semua unsur yang terkait pada kegiatan pilkada.
3. Melakukan *quality control* : (i) dilakukan validasi hasil pencatatan perolehan suara oleh relawan (ii) ketua TPS menandatangani hasil validasi data perolehan suara (iii) melakukan pemantauan untuk memastikan kesiapan proses sudah berjalan dengan lancar (iv) memastikan perolahan data dengan *spotcheck* apakah data sudah benar dan valid (v) memilih secara acak TPS ataupun daerah yang dikunjungi untuk *spotcheck*

Fungsi utama *quick count* adalah sebagai alat kontrol terhadap penyelenggara Pilkada dan melakukan perkiraan perolehan suara Pilkada [7]. Sebagai alat pengontrol, *quick count* dapat mampu mengungkapkan, melaporkan, dan mendeteksi kecurangan atau penyimpangan. Sudah terbukti *quick count* dapat meningkatkan kepercayaan terhadap kinerja penyelenggara dan memberikan legalitas terhadap proses Pilkada. Dalam hal ini, proses perhitungan suara masih lamban walaupun sudah menggunakan teknologi informasi yang canggih dan mahal yang dilakukan oleh KPU. Estimasi hasil sampel perhitungan suara baru dapat diketahui dua minggu setelah dilaksanakannya Pilkada. Sementara itu setelah proses pemungutan suara dilakukan baru dapat dilakukan perhitungan resmi lebih kurang satu bulan. Dengan *quick count* ini, publik dapat mengetahui hasil Pilkada dengan cepat. Penarikan sampel yang baik dan akurat serta presisi dari perhitungan suara yang diperoleh dapat mempengaruhi keberhasilan *quick count*.

### B. Java Mobile

Java adalah salah satu platform bahasa pemrograman

berbasis OOP (*Object Oriented Programming*). Penyelesaian masalah direpresentasikan ke model objek merupakan paradigma OOP. Platform java memiliki tiga buah edisi yang berbeda yaitu *Java 2 Platform Micro Edition (J2ME)*, *Java 2 Platform Standar Edition (J2SE)*, dan *Java 2 Platform Enterprise Edition (J2EE)*.

Lingkungan pengembangan yang didesain untuk meletakkan perangkat lunak Java pada barang elektronik beserta perangkat pendukungnya pada J2ME, jika perangkat lunak berfungsi baik pada sebuah perangkat maka belum tentu juga berfungsi baik pada perangkat yang lainnya[8].

J2EE adalah kelompok dari beberapa API Java, menyediakan kelas dasar dan API yang mendukung pengembangan dan rutin standar untuk aplikasi klien maupun server, termasuk *web browser*.

## III. METODOLOGI PENELITIAN

### A. Metode Pengembangan Sistem

Pengembangan sistem informasi ini dilaksanakan dengan menggunakan metodologi proses pengembangan sistem yang bernama FAST (*Framework for the Application of System Thinking*). FAST adalah suatu pendekatan yang menggabungkan seluruh pendekatan untuk melakukan analisis atas suatu sistem informasi. FAST juga mendefinisikan tahapan untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi, dan kebutuhan yang diharapkan pengguna/user sehingga dapat diusulkan perbaikan. Pengembangan ini bersifat daur hidup karena setelah selesai tahapan implementasi dan pemeliharaan maka sistem tersebut akan memberikan umpan balik ke analisis sistem yang telah dirancang. Sehingga tahapan pengembangan di atas terus-menerus dilakukan penyempurnaan sistem[9].

### B. Metode Penarikan Sampel dalam Quick Count

*Multistage random sampling* adalah metodologi yang digunakan untuk penarikan sampel pada penelitian ini. *Multistage random sampling* merupakan teknik sampling yang dikonstruksikan dari metode sampling acak sederhana yang melalui beberapa tahapan pengambilan sampel secara acak. Setiap anggota populasi memungkinkan mempunyai peluang yang sama untuk terpilih sebagai sampel jika menggunakan teknik tersebut. Kemudian dengan menggunakan sedikit sampel dapat dilakukan pengukuran. Dengan demikian tanpa melibatkan semua anggota populasi, hasil survey dapat digeneralisasikan sebagai representasi populasi. Pada akhirnya informasi statistik yang sangat bermanfaat dapat diperoleh terutama pada masalah yang kompleks[10].

Gabungan antara sampel stratifikasi (*stratified random sampling*) dengan sampel kluster (*cluster sampling*) merupakan teknik dasar yang dilakukan *multistage random sampling*. Heterogenitas dari populasi dari tercermin dalam sampel dengan dilakukannya teknik stratifikasi. Untuk meminimalkan biaya yang timbul karena stratifikasi, maka perlu dikombinasikan dengan teknik kluster. Sampel tidak menyebar sehingga dapat mengurangi biaya untuk menjangkanya walaupun dengan kluster dapat mengurangi membuat cerminan karakteristik populasi[4].

$$t = \frac{\text{Jumlah pemilih yang dibutuhkan}}{\text{Rata-rata pemilih per TPS}} \quad (1)$$

*Multistage random sampling* dibangun dengan mengambil serangkaian sampel acak sederhana secara bertahap. Model pengacakan yang dilakukan secara proporsional berdasarkan besaran populasi jumlah TPS. Dengan metode pengacakan yang bertahap seperti di atas, maka akan didapatkan jumlah sampel yang proporsional sesuai dengan presentase jumlah TPS di daerah tersebut sehingga hasil *quick count* dapat digeneralisasi dan mencerminkan hasil pilkada dengan akurasi tinggi[11].

Dalam menentukan besaran sampel sangat terkait dengan keinginan kita akan ketelitian dari suatu sampel. Menentukan besar sampel tergantung kepada tiga hal berikut yaitu *variability, sampling error, dan confidence interval*[12].

#### a. Menentukan Jumlah TPS Sampel

Berikut ini adalah langkah yang dilakukan dalam proses *quick count* : Berdasarkan hasil penelitian yang diakuakan pemilih tetap berjumlah 354.260 orang dan terbagi di 1.542 TPS. *Sampling error* yang digunakan dalam *quick count* ini sebesar 1% dan 99% untuk tingkat kepercayaan. Jumlah rata-rata pemilih per TPS sebanyak 266. dengan nilai proporsi populasi beragam (50:50). Teknik pemilihan TPS dapat dilakukan sebagai berikut :

##### 1. Menentukan jumlah sampel pemilih

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

Keterangan :

n = Besar sampel pemilih

N = Jumlah populasi

E = Kesalahan sampel yang dikehendaki (*sampling error*). Misalnya : 2% atau 0,02.

P(1-p) = Variasi populasi dalam kasus ini dinyatakan dalam bentuk proporsi. Proporsi dibagi kedalam dua bagian dengan total 100% (atau 1). Jika populasi diasumsikan heterogen, maka  $p = 0,5$

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.

Contoh :

$$n = \frac{(2,58)^2 \cdot [0,5(1-0,5)] \cdot 354260}{(2,58)^2 \cdot [0,5(1-0,5)] + (354260-1) \cdot 0,01^2} = 15.883$$

Jadi jumlah sampel pemilih yang harus diambil sebanyak 15.883

##### 2. Menentukan jumlah TPS sampel (t)

$$t = \frac{\text{Jumlah pemilih yang dibutuhkan}}{\text{Rata-rata pemilih per TPS}} \quad (3)$$

Contoh :

$$t = \frac{15883}{266} = 60$$

Jadi membutuhkan TPS sampel sebanyak 60 TPSTPS dari 1.542 TPS secara random.

##### 3. Menentukan interval sampel

Cara yang dilakukan untuk menentukan nilai interval sampel adalah dengan melakukan pembagian jumlah populasi dengan jumlah TPS sampel. didapatlah interval sampel sebagai berikut :  $1.542 / 60 = 25,7$ , kemudian dilakukan pembulatan menjadi 26.

###### ➤ Memilih secara *random* sampel awal / pertama

Hal pertama yang dilakukan adalah memilih secara random sampel pertama dari angka 1 sampai dengan 26 secara random. Misalnya 4 dijadikan sebagai angka sampel awal TPS. sehingga urutan nomor 4 dijadikan sampel awal/pertama *quick count*.

###### ➤ Memilih secara *random* sampel berikutnya

Sampel selanjutnya dilakukan secara distematis dengan menambahkan nilai interval sampel yaitu 26. Sampel kedua adalah 30 (4 + 26), sampel ketiga 56 (30 + 26), dan seterusnya hingga ditemukan 60 TPS yang dijadikan sample yaitu : TPS 4, TPS 30, TPS 56, dan seterusnya.

##### b. Tingkat Kesalahan (*Sampling Error*)

Dikarenakan *quick count* menggunakan sampel, dapat dipastikan terjadi kesalahan (*sampling error*). Jika besaran sampel semakin kecil maka akan semakin besar *sampling error* dan sebaliknya *sampling error* akan semakin mengecil apabila jumlah sampel semakin besar. *Sampling error* menggambarkan derajat akurasi, di dalam penyelenggaraan *quick count* harus didahului dengan menentukan seberapa besar tingkat kesalahan yang diinginkan dari *quick count* yang akan dilaksanakan. Pilihan *sampling error*, menentukan derajat ketelitian dari *quick count*. Keakuratan hasil *quick count* dalam memprediksi populasi (nilai parameter) ditentukan oleh semakin kecilnya *sampling error*.

Tujuan *quick count* adalah melakukan peramalan terhadap hasil perolehan suara dari kandidat, jadi *sampling error* yang digunakan harus sekecil mungkin. Dengan *sampling error* yang kecil, hasil prediksi akan lebih akurat dan presisi. Nilai *sampling error* yang digunakan maksimal 1%. Jika memungkinkan, gunakan *sampling error* yang lebih kecil, misalnya 0,5%.

##### c. Tingkat Kepercayaan (*confidence interval*)

Besaran estimasi atau taksiran dari sampel yang berlaku untuk populasi sangat berhubungan dengan tingkat kepercayaan. Tingkat kepercayaan diibaratkan dua sisi mata uang dengan *sampling error*. Nilai populasi diestimasi dengan angka *sampling error*, disisi lain besaran estimasi itu berlaku di dalam populasi dipastikan dengan tingkat kepercayaan.

Untuk perolehan suara dari kandidat sebaiknya

penyelenggara *quick count* dapat menggunakan tingkat kepercayaan, msalnya 90%, 95%, atau 99%. Biasanya tingkat kepercayaan yang sering digunakan sebesar 99% dan kemungkinan salah sebesar 1%.

#### IV. HASIL DAN PEMBAHASAN

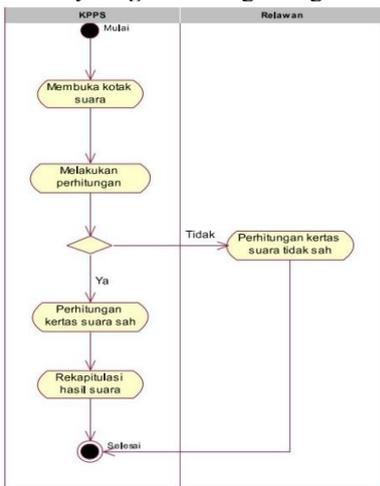
##### A. Scope Definition

Secara umum aplikasi *quick count* akan dibangun dengan tujuan dalam mewujudkan integrasi data, kemudahan pengelolaan, dan kemudahan pengaksesan sehingga dapat mendukung fungsi dalam perhitungan suara dan kelancaran pelaksanaan dan pengiriman data suara yang efektif dan efisien. Dari uraian di atas bahwa aplikasi *quick count* sangat dibutuhkan dalam suatu instansi atau lembaga pemerintah.

##### B. Problem Analysis

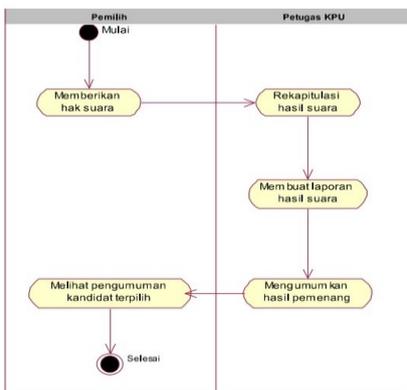
Pada diagram ini akan menjelaskan pemodelan bisnis dengan dilakukannya analisa pada proses bisnis pilkada yang sedang berjalan di KPU Kota Palembang yang kemudian akan diilustrasikan ke dalam bentuk *business activity diagram* yang menampilkan rangkaian proses yang ada pada proses pilkada.

##### a. Business activity diagram Menghitung Hasil Suara



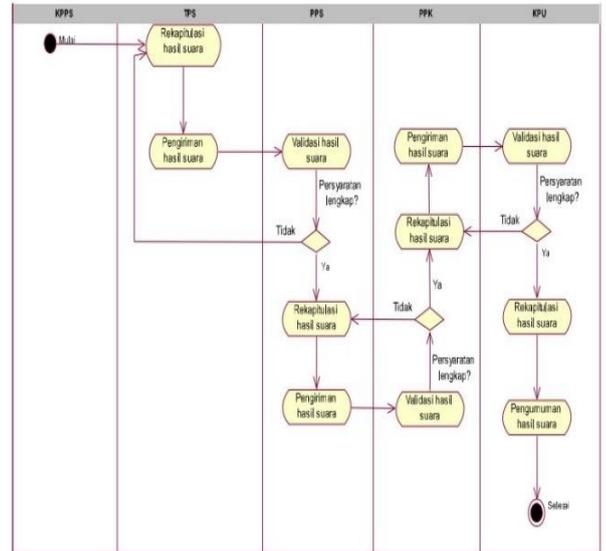
Gambar 1. Business activity diagram menghitung hasil suara

##### b. Business activity diagram melakukan rekapitulasi dan mengirim hasil suara



Gambar 2. Business activity diagram melakukan rekapitulasi dan mengirim hasil suara

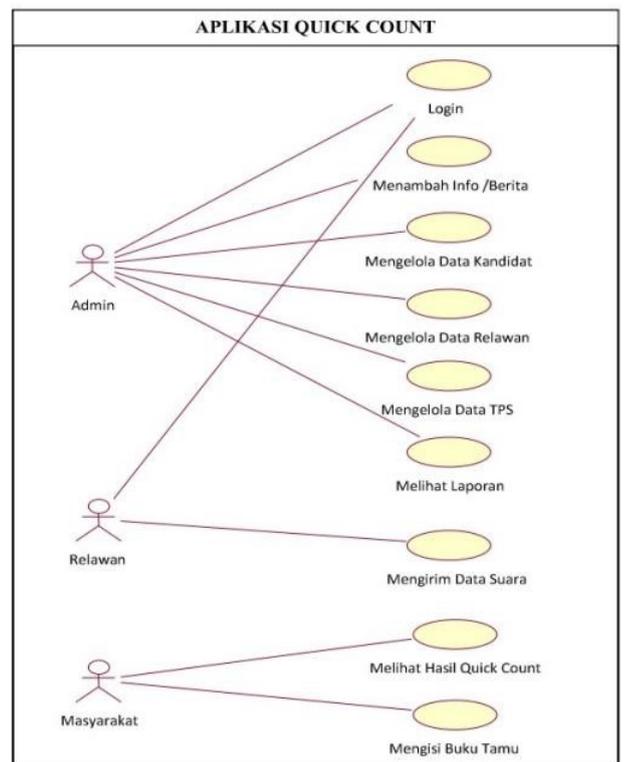
##### c. Business activity diagram melihat pengumuman kandidat terpilih



Gambar 3. Business activity diagram melihat pengumuman kandidat terpilih

##### C. Requirement Analysis

Pemodelan *use case* merupakan cara yang baik untuk dapat menentukan *requirement* dari *software*.



Gambar 4. Use case diagram aplikasi quick count

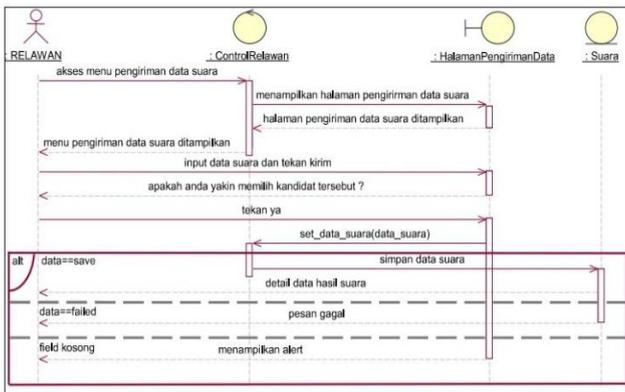
##### D. Logical Design

##### a. Sequence Diagram

Pemodelan dengan *sequence diagram* bertujuan untuk menggambarkan bagaimana proses pesan dapat diterima diantara objek dan *class* serta terkirim dalam suatu rangkaian. Berikut ini *sequence diagram* yang

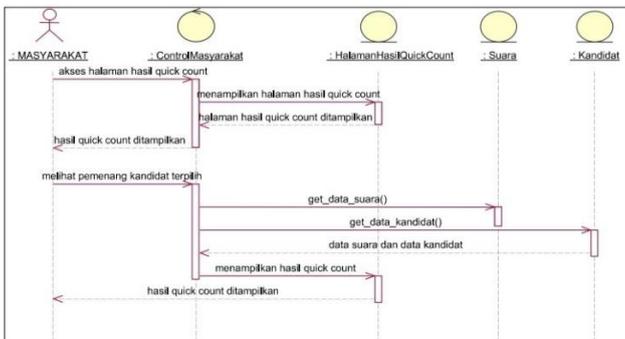
menggambarkan bagaimana objek dapat berinteraksi satu sama lain melalui pesan yang dieksekusi sebuah *use case* atau operasi.

➤ *Sequence Diagram* Mengirim Data Suara



Gambar 5. *Sequence diagram* mengirim data suara

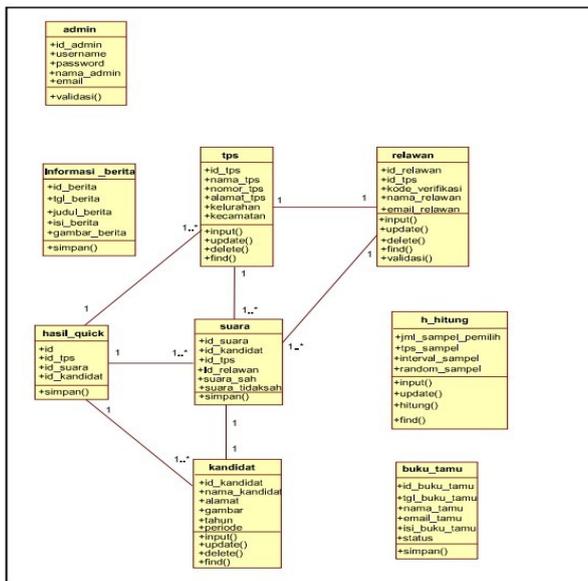
➤ *Sequence Diagram* Melihat Hasil Quick Count



Gambar 6. *Sequence diagram* melihat hasil quick count

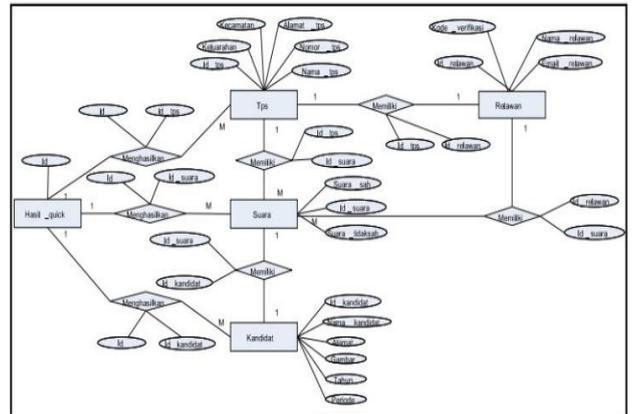
b. *Class Diagram*

*Class diagram* berfungsi untuk menggambarkan bagaimana hubungan antar *class* dalam suatu sistem yang sedang dibuat dan bagaimana caranya agar mereka dapat saling bekerjasama/kolaborasi untuk menghasilkan suatu tujuan. Berikut ini adalah *class diagram* dari aplikasi *quick count* yang akan dibangun.



Gambar 7. *Class diagram* aplikasi *quick count*

c. *Entity Relationship Diagram (ERD)*

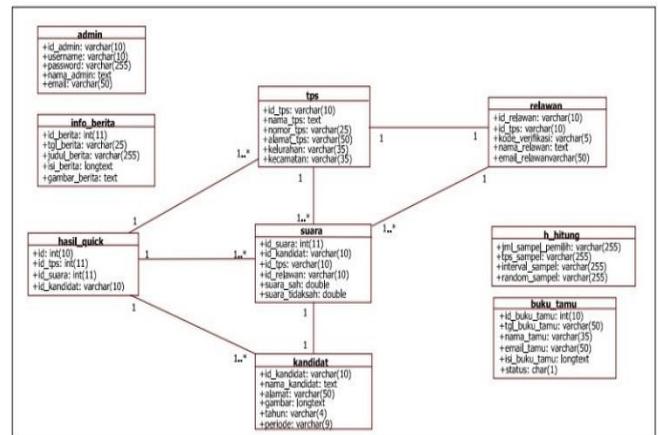


Gambar 8. *ERD* aplikasi *quick count*

E. *Physical Design*

a. *Skema Database*

Berikut ini merupakan skema *database* yang menggambarkan relasi tabel-tabel dari aplikasi *quick count*.



Gambar 9. *Database* aplikasi *quick count*

b. *Interface Java*



Gambar 10. Tampilan menu input kandidat suara

### c. Interface Website



Gambar 11. Tampilan menu hasil *quick count*

## V. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan dan diuraikan, sehingga dapat diambil kesimpulan yaitu aplikasi *quick count* ini menggunakan arsitektur *client server* yang dapat menyajikan informasi hasil pilkada secara *real time*, dapat memprediksikan hasil pilkada secara cepat sehingga dapat diperoleh secara akurat dan dapat dipasitikan sama dengan hasil sebenarnya, dan namun hanya dapat diproses dengan jumlah pemilih minimal 700 pemilih agar dapat melakukan sampel, apabila nilai populasi kurang dari batas minimal maka penarikan sampel tidak dapat diproses oleh aplikasi.

## DAFTAR PUSTAKA

- [1] Hermawan, "Analisis Preferensi Masyarakat Kota Bogor Terhadap Calon Walikota 2009-2014," IPB (Bogor Agricultural University), 2008.
- [2] H. P. Angga Lesvian, Anton Mulyanto, "Kajian Analitis Metode Sampling Yang Tepat Dengan Akurasi Tinggi Untuk Estimasi Pemenang Pemilu Pada Quick Count," Bogor, 2009. [Online]. Available: [https://repository.ipb.ac.id/jspui/bitstream/123456789/19915/10/KAJIAN\\_ANALITIS2.doc](https://repository.ipb.ac.id/jspui/bitstream/123456789/19915/10/KAJIAN_ANALITIS2.doc).
- [3] A. Setiawan, "Penggunaan Resampling Dalam Penggambaran Quick Count," *Jambura J. Probab. Stat.*, vol. 1, no. 1, pp. 1–12, 2020.
- [4] Kismiantini, "Pengumpulan Data dengan Quick Count dan Exit Poll," 2007.
- [5] Ujiyati TP, "Quick count," 2004. <http://www.lp3es.or.id/program/pemilu2004/QCount.htm> (accessed Dec. 27, 2020).
- [6] Sumargo M, "Quick count," 2006. <http://www.beritaipetek.com/zberitaberitaipetek-2006-07-12-Quick-Count.shtml> (accessed Dec. 27, 2020).
- [7] A. I. Adha, "Kebebasan pengumuman hasil hitung cepat (quick count) sebagai bentuk pengawasan dan partisipasi masyarakat dalam penyelenggaraan pemilu (analisis Putusan Mahkamah Konstitusi Nomor 24/PUU-XII/2014)," Fakultas Syariah dan Hukum UIN Syarif Hidayatullah Jakarta, 2017.
- [8] M. Shalahuddin and A. S. Rosa, "Belajar Cepat Pemrograman Perangkat Telekomunikasi Mobile," *Inform. Bandung*, 2008.
- [9] Y. Dharmawan, "Sistem Informasi Efisiensi Penggunaan Tempat Tidur Unit Rawat Inap Dengan Menggunakan Indikator Grafik Barber Johnson Di Rumah Sakit Panti Wilasa Citarum Semarang," program Pascasarjana Universitas Diponegoro, 2006.

- [10] A. Rusgiyono and others, "Perhitungan Cepat (Quick Count) Dengan Metode Multistage Random Sampling (Studi Kasus Pilkada Nanggroe Aceh Darussalam 2006)," *Mathematics and Natural science*, 2008.
- [11] D. Bilgies, "Rancang Bangun Website Sistem Hitung Cepat (Quickcount) Pada Pemilihan Umum," Universitas Brawijaya, 2014.
- [12] J. Juwairiah, B. P. Dessyanto, and T. G. Fifi, "Aplikasi Quickcount Pemilihan Presiden RI Menggunakan Teknologi Mobile," *Telematika*, no. 7, 2011.