Pengembangan Python API untuk Mendukung Online Bingo Voting

Muhammad Umar Fariz Tumbuan - 13515050
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13515050@std.stei.itb.ac.id

Kode sumber tersedia di: https://github.com/agent-whisper/if4020_bingo_voting.git

***Abstrak*—*Voting* (pemungutan suara) merupakan salah satu cara untuk mendukung penerapan demokrasi. Namun, sistem voting konvensional rentan untuk diserang atau diancam pihak ketiga. *Bingo voting* adalah sebuah sistem *voting* elektronik yang memungkinkan pemungutan suara diadakan secara transparan. Pemilih dapat memastikan suaranya sudah masuk tanpa perlu mengungkapkan pilihannya. Tetapi *bingo voting* belum menjadi sebuah konsep yang umum, sehingga belum banyak implementasi yang dapat digunakan. Sebuah API yang dapat memudahkan implementasi *bingo voting* dapat membantu membuat konsep *bingo voting* menjadi lebih umum.**

Keywords—Bingo voting; e-voting; Python, pedersen commitment scheme.

I. Pendahuluan

Voting (pemungutan suara) merupakan salah satu cara untuk mendukung penerapan demokrasi di sebuah komunitas, organisasi, atau negara. Dalam demokrasi yang ideal, setiap anggota dapat mengeluarkan hak suaranya tanpa dipengaruhi oleh pihak lain. Namun, sistem *voting* konvensional rentan untuk diserang oleh pihak ketiga, baik melalui manipulasi suara yang masuk atau membayar / mengancam para pemilih [1]. Salah satu faktor penyebab kerentanan adalah para pemilih tidak dapat mengecek apakah suara mereka diproses dengan benar tanpa mengeluarkan informasi mengenai pilihan mereka.

Bingo voting adalah sistem *voting* elektronik yang memungkinkan pemungutan suara diadakan secara transparan. *Bingo voting* memungkinkan pemilih untuk mengecek apakah pilihan mereka sudah diproses tanpa perlu mengungkapkan pilihan mereka. Hal ini memungkinkan proses *voting* dapat berjalan dengan lebih transparan, aman, dan adil. Namun saat ini belum ada implementasi *bingo voting* yang dapat digunakan umum. Karena itu pembuatan API untuk *bingo voting* dapat memudahkan implementasinya sehingga *bingo voting* menjadi lebih umum digunakan.

II. Dasar Teori

A. Pedersen Commitment Scheme

Pedersen commitment memungkinkan kita untuk meng-*commit* (mengenkripsi) sebuah pesan, namun dapat mengungkapkan isinya di lain waktu. Skema ini memungkinkan pembuktian nilai sebenarnya dari *commitment* terhadap klaim nilai sesungguhnya.

Skema dimulai dengan menentukan dua nilai prima besar p dan q. Kemudian nilai g dan s ditentunkan secara acak dari rentang $1 <= g, s < q-1$. Lalu sebuah nilai h ditentunkan dengan persamaan 1. Sebuah pesan m dapat di-*commit* dengan menggunakan persamaan 2 untuk menghasil c_1. Nilai r dipilih secara acak dari rentang $1 <= r < q-1$. Untuk membuktikan isi *commitment*, nilai m dan r dipublikasikan untuk digunakan kembali pada persamaan 2, menghasilkan c_2. Jika $c_1 = c_2$, maka *commitment* dan klain nilai m terbukti

$$h = g^s \bmod (q) \quad (1)$$

$$c = g^m h^r \bmod (q) \quad (2)$$

B. Bingo Voting

Bingo voting adalah sistem *voting* elektronik yang memanfaatkan konsep *commitment scheme* dan *random number generator* (rng) [1]. *Bingo voting* terdiri atas tiga fase:

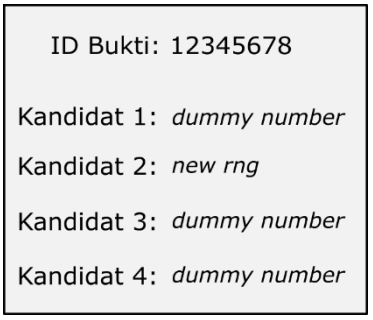
1) Fase pra-pemilihan

Pada fase ini, setiap kandidat diberikan p buat *dummy vote* di mana p adalah jumlah pemilih. *Dummy vote* dibangkitkan menggunakan rng yang dapat dipercaya. Misalkan terdapat k

kandidat, maka secara total akan terdapat $k * p$ buah *dummy vote*. *Dummy vote* kemudian didistribusikan secara acak ke setiap kandidat. Akhirnya, setiap *dummy vote* dienkripsi menggunakan sebuah *commitment scheme*. Hasil *commit* dari setiap *dummy vote* beserta pemiliknya dipublikasikan sebelum *voting* dilakukan.

2) Fase pemilihan

Setiap pemilih mulai memasukkan kandidat pilihannya. Setelah pemilih memberi masukan, sebuah nilai baru dibangkitkan menggunakan rng untuk kandidat yang dipilihnya. Untuk setiap kandidat lain yang tidak dipilih diambil satu *dummy vote* yang dimiliki masing-masing. Pemilih mendapatkan bukti berupa karcis yang berisi angka acak baru untuk kandidat yang dipilih dan sebuah *dummy vote* untuk setiap kandidat lainnya. Karena *dummy vote* dan angka acak yang baru tidak bisa dibedakan, pihak lain tidak dapat mengetahui kandidat mana yang telah dipilih. Ilustrasi karcis diberikan pada Gambar 1.



Gambar 1. Ilustrasi karcis.

3) Fase pasca-pemilihan

Setelah fase pemilihan berakhir, beberapa informasi mulai dipublikasikan pada sebuah papan pengunguman. Informasi tersebut antara lain:

- Hasil akhir pemungutan suara.
- Karcis yang telah dikeluarkan.
- Nilai, *commitment*, dan *secret* dari setiap *dummy vote* yang tidak terpakai.
- Bukti bahwa setiap *commitment* yang tidak dibuka benar-benar digunakan dalam setiap karcis yang telah dikeluarkan.

III. Rancangan API

A. Kelas PedersenBVM

PedersenBVM menyediakan API untuk mengimplementasi server utama dari *bingo voting*. Kelas ini menyimpan konfigurasi, informasi, dan *state* dari proses *voting*. PedersenBVM diinisialisasi dengan masukan konfigurasi. Isi konfigurasi beserta penjelasannya diberikan pada Tabel 1. Menggunakan informasi tersebut, Objek PedersenBVM akan membangkitkan *dummy vote* untuk setiap kandidat. PedersenBVM juga menyediakan fungsi-fungsi untuk melakukan *query* terhadap informasi kandidat, seperti label kandidat, *dummy vote*, *commitment*, dan lain-lain.

Tabel 1. Deskripsi konfigurasi PedersenBVM

No.	Konfigurasi	Deskripsi
1.	security	Mempengaruhi batas atas dari pembangkitan angka untuk *dummy vote*
2.	num_of_voters	Jumlah orang yang akan melakukan *voting*
3.	candidate_labels	Larik yang berisi label untuk setiap kandidat.
4.	ip	alamat ip server tempat PedersenBVM dijalankan
5.	port	nomor *port* dari server tempat PedersenBVM dijalankan.

B. Kelas PedersenBooth

Kelas PedersenBooth menyediakan API untuk mengimplementasi klien tempat pemilih melakukan *voting*. Fungsi yang disediakan kelas ini adalah fungsi mengirim *vote* dan meng-*query* label kandidat. Fungsi hanya dibatasi sampai kedua hal tersebut karena klien dari server tidak perlu mengetahui informasi lain.

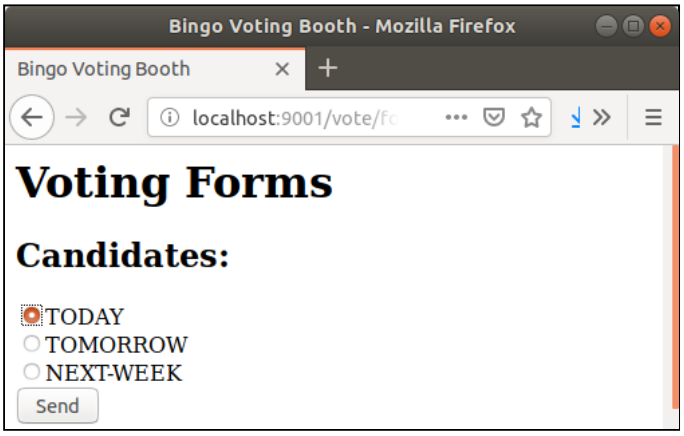
C. Kelas PedersenBoard

Kelas PedersenBoard menyediakan API untuk mengimplementasi klien papan pengunguman. Kelas ini memiliki akses *query* yang lebih dalam dibandingkan kelas PedersenBooth. Informasi yang dapat di-*query* sesuai dengan yang telah dijelaskan sebelumnya. Kelas ini perlu menyimpan informasi uri dari server utama.

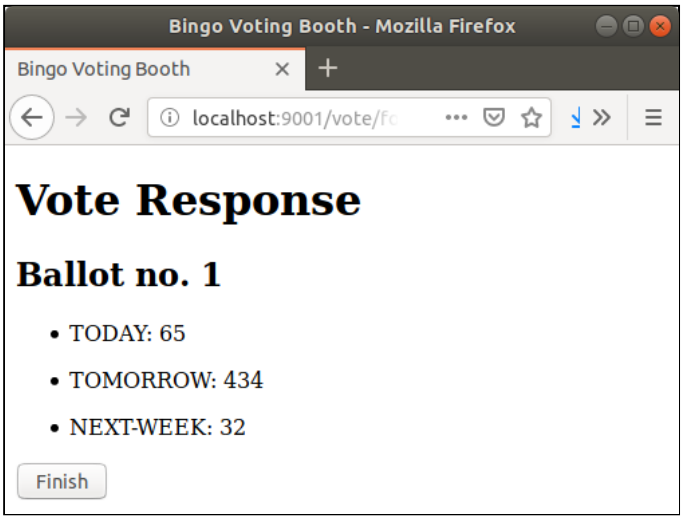
IV. Hasil Implementasi

Tabel 2 menunjukkan uri dari *server* utama yang telah diimplementasi. Gambar 2 menunjukkan hasil implementasi klien tempat pemilihan. Gambar 3 menunjukkan hasil implementasi papan pengunguman. Hasil implementasi tersebut menunjukkan bahwa API yang telah dibuat sudah dapat memberikan fungsi-fungsi dasar yang dibutuhkan dalam mengimplementasi *bingo voting*.

(a)

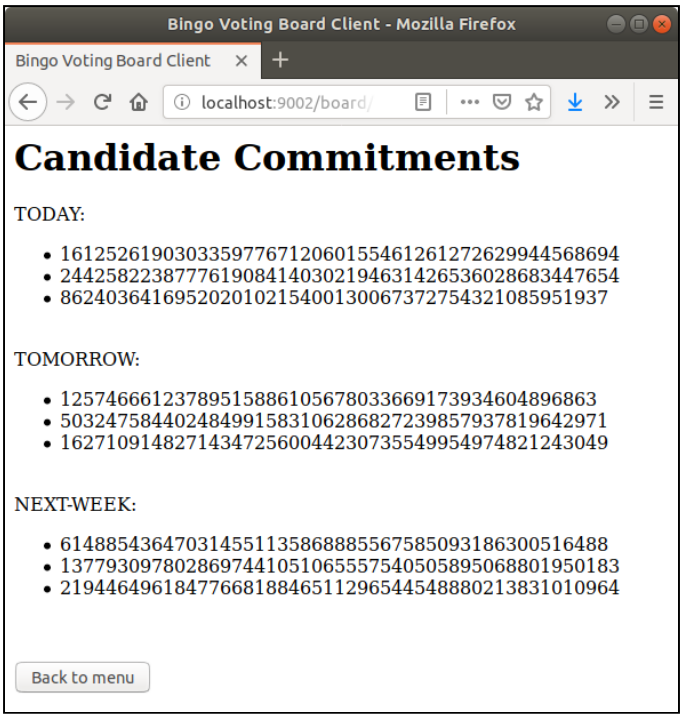


(b)

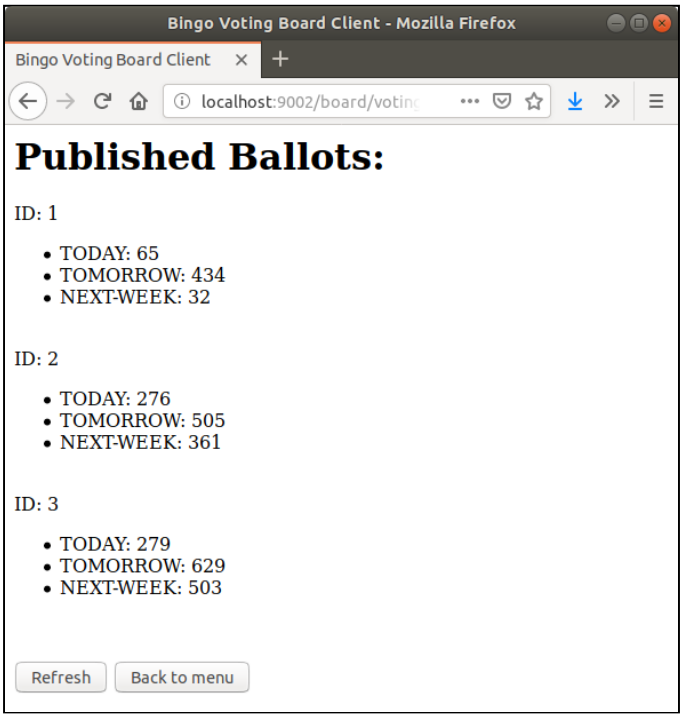


(c)

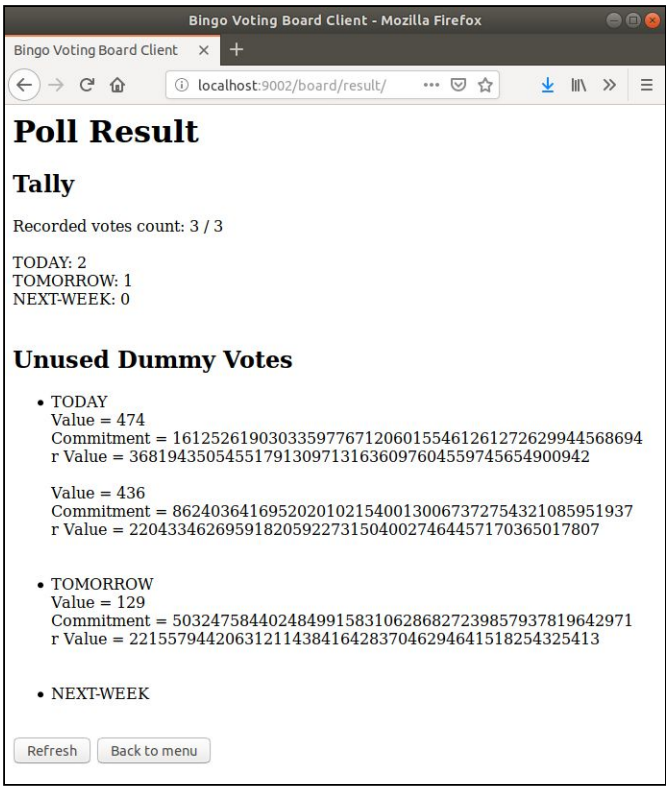
Gambar 2. Implementasi klien tempat pemilihan.



(a)



(b)



(c)

Gambar 3. Implementasi klien papan pengunguman.

V. Kesimpulan dan Saran

Hasil implementasi klien dan server menunjukkan bahwa API yang dibuat sudah dapat digunakan untuk mengimplementasi fungsi-fungsi dasar dari skema *bingo voting*. Namun terdapat banyak perbaikan yang bisa dilakukan, baik dari sisi API ataupun pelaksanaan *bingo voting*. Perbaikan tersebut di antaranya:

1. Menerapkan *digital signature* untuk memastikan transaksi dilakukan oleh server dan klien yang benar.
2. Menerapkan suatu cipher untuk menjaga kerahasiaan informasi selama transaksi dilakukan.
3. Melakukan *bingo voting* di dalam sebuah *virtual private network*.
4. Memfilter alamat *ip* yang tidak dikenal.

Referensi

[1] Jens-Matthias Bohli, Müller-Quade Jörn, dan Stefan Röhrich. Bingo Voting: Secure and Coercion-Free Voting Using a Trusted Random Number Generator. E-Voting and Identity. 2007

[2] Torben Pryds Pedersen. Non-interactive and Information-Theoretic Secure Verifiable Secret Sharing. In Joan Feigenbaum, editor, Advances in Cryptology – CRYPTO ’91: Proceedings, volume 576 of Lecture Notes in Computer Science, pages 129–140. Springer, 1991.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 9 Mei 2019

M. Umar Fariz T.
13515050