**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# **IDENTITAS PENGUSUL**

**Nama : IWANG ARYADINATA**

**NRP : 5109100064**

**Dosen Wali** : **Ary Mazharuddin Shiddiqi, S.Kom, M.Comp.Sc**

# **JUDUL TUGAS AKHIR**

**Rancang Bangun Aplikasi dengan Protokol Keamanan Challenge Response untuk Pemungutan Suara pada Perangkat Bergerak**

***Design Build Applications with Challenge Response Security Protocol for Voting on Mobile Devices***

# **URAIAN SINGKAT**

Secara tradisional, sistem pemilihan umum pada suatu organisasi dilakukan dengan cara memberikan suara pada hari tertentu dan tempat tertentu, yang bertentangan dengan mobilitas gaya hidup masyarakat saat ini. Dan juga pemungutan suara masih dilakukan dengan menggunakan kertas suara yang akan menambah biaya dalam proses pemungutan suara yang akan dilakukan. Dengan meluasnya internet akses dapat dilihat sebagai kesempatan untuk menangani masalah mobilitas, yaitu dengan menerapkan *Mobile Voting*.

*Mobile Voting* merupakan penggunaan *mobile phone* untuk melakukan pemungutan suara dalam pemilihan umum. *Mobile Voting* bertujuan untuk meningkatkan partisipasi, menurunkan biaya pemilihan umum berjalan, dan memberikan kesempatan kepada pemilih yang saat tertentu berada jauh dari tempat pemilihan untuk memilih calon.

Aplikasi *Mobile Voting* ini berbasis Android, dikarenakan banyak orang telah menggunakan *device mobile phone* dengan berbasis Android. Oleh karena itu diperlukan Android SDK yang merupakan paket untuk membangun aplikasi Android. Pembangunan aplikasi ini memerlukan *server* dan *client*. *Server* mempunyai basis data (MySQL) untuk menyimpan data dari pemungutan suara yang telah dilakukan. Dan juga memakai *Web Service* sebagai metode komunikasi antara *mobile phone* dengan basis data *server* (MySQL). Saat *mobile phone* berkomunikasi dengan basis data *server* pengiriman data akan dienkripsi agar data tidak dapat dibaca oleh pihak yang tidak bertanggung jawab.

Makalah ini merupakan rincian kebutuhan danimplementasi untuk sistem pemilihan suara dengan protokol *challenge response* pada perangkat bergerak untuk pemilihan ketua organisasi pada sebuah universitas sehingga mahasiswa dapat memberikan suara mereka kapan saja dan dimana saja.

# **4. PENDAHULUAN**

## 4.1 LATAR BELAKANG

Pemungutan suara dan pemilihan umum merupakan hal yang berpengaruh pada masyarakat moderen saat ini. Pemungutan suara telah banyak digunakan mulai dari masyarakat kecil, sampai dengan sebuah negara. Dalam sebuah negara yang menerapkan sistem demokrasi, pemungutan suara digunakan untuk mengambil keputusan dari pilihan yang telah ditentukan, contohnya dalam pemilihan Presiden dan Wakil Presiden. Terdapat beberapa syarat untuk seseorang dapat melakukan pemilihan seperti pemilih merupakan warga negara tersebut dan juga sudah mempunyai cukup umur untuk memilih. Sehingga tidak semua masyarakat dapat memberikan suaranya. Jadi dalam pemungutan suara dibutuhkan prosedur yang dapat menjamin keaslian dan keabsahan dari pemungutan suara yang telah dilakukan [[1](#Ins03)].

Hasil dari pemilihan umum juga dapat memiliki efek positif dan juga efek negatif banyak pada komunitas dan kesejahteraan masyarakat. Pemungutan suara yang pertama kali dilakukan dengan kertas suara yang telah ditandai sebagai bukti kertas suara dapat diterima, kemudian pemilih akan memilih calon dengan mencoblos kertas suara. Pemilih lalu memasukkan ke kotak suara dan panitia akan melakukan penghitungan suara secara manual kemudian hasil dari perhitungan panitia akan dikirimkan ke pusat untuk menggabungkan suara dengan daerah lainnya. Pemilihan dengan metode ini sangat membutuhkan banyak tenaga, waktu dan sumber daya manusia. Disisi lain terdapat juga beberapa hal seperti tidak teliti dalam penghitungan suara dan kecurangan seperti manipulasi kertas suara.

Penggunaan perangkat bergerak untuk memberikan suara dalam pemilihan umum atau disebut juga *Mobile Voting* dapat menjadi solusi untuk mengatasi kelemahan ini. Dengan menggunakan *Mobile Voting* dapat menyederhanakan proses pemilu, meningkatkan partisipasi dari pemilih, mengurangi kesalahan dalam penghitungan dan meminimalkan waktu yang dibutuhkan untuk mengumumkan hasil. Setelah pemilih terdaftar pada saat pemilihan maka pemilih dapat memberikan suaranya hanya dengan *login* di aplikasi *voting* yang telah ditentukan menggunakan *id* dan *password* masing-masing, setelah itu pemilihan calon dapat dilakukan hanya dengan memilih pilihan yang disediakan pada sistem, lalu sistem akan menghitung pilihannya secara digital. Untuk perhitungan total pun dilakukan oleh sistem secara digital menggunakan fungsi penjumlahan [[2](#ZQa07)]. Digunakan sebuah teknik algoritma enkripsi dalam pengiriman data yang dilakukan agar keamanan dan kerahasian data yang dikirimkan tetap terjaga.

## 4.2 RUMUSAN MASALAH

Adapun rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimanakah membuat basis data *server* yang dapat menangani semua kebutuhan dalam pemilihan umum?
2. Bagaimanakah membuat *Web Service* yang dapat menghubungkan *mobile phone* dengan basis data *server*?
3. Bagaimanakah mengimplementasikan algoritma enkripsi RC4 untuk keamanan pengiriman data?
4. Bagaimanakah cara mencegah beberapa akun digunakan untuk melakukan pemilihan pada satu perangkat mobile?

## 4.3 BATASAN MASALAH

Adapun permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, diantaranya sebagai berikut:

1. Aplikasi dibangun dengan berbasiskan *client - server*.
2. Untuk basis data *server* menggunakan basis data (MySQL) dalam penyimpanan data.
3. Untuk *client* berada pada *mobile phone* yang berbasis Android.
4. Aplikasi ini menggunakan enkripsi dengan metode RC4 pada pengiriman data saat komunikasi antar basis data *server* dan *client*.

## 4.4 TUJUAN DAN MANFAAT

Adapun beberapa tujuan dari pembuatan tugas akhir ini adalah:

1. Memanfaatkan internet dalam melakukan pemungutan suara.
2. Merancang dan membangun sistem pemilihan umum berbasis *mobile phone.*
3. Merancang dan membangun sistem pemungutan suara yang aman.

Manfaat yang diharapkan dari Tugas Akhir ini adalah adanya suatu sistem pemilihan umum berbasis *mobile pho*ne agar dapat meningkatkan partisipasi, menurunkan biaya dalam pengadaan pemilihan umum, dan dapat dilakukan dimana saja.

# **5. TINJAUAN PUSTAKA**

## 5.1 ANDROID

Android adalah sistem operasi yang berbasis Linux yang dikembangkan oleh Google untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam perangkat bergerak. Android juga dapat dikembangkan sendiri dengan menggunakan Android SDK (*Software Development Kit*) dengan menggunakan bahasa Java dan Google telah menyediakan banyak *library* untuk pengembang.

Perangkat Android sendiri terdiri dari sistem operasi, *middleware*, dan *key application*. Pengembangan software Android menggunakan bahasa pemrograman Java dan aplikasi Android tidak berjalan langsung diatas kernel sistem operasi namun aplikasi tersebut berjalan diatas Dalvik, *virtual machine* yang khusus diotimasikan untuk perangkat *mobile* [[3](#NSa11)].

## 5.2 ANDROID SDK (Software Development Kit)

Android SDK adalah tools API (*Aplication Programming Interface*) yang diperlukan untuk mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Android SDK tersebut berisi *debugger*, *emulator*, *librar*y, dokumentasi, sampel *source code*, dan tutorial untuk system operasi Android. Sebagai platform aplikasi netral, Android memberi kesempatan bagi semua orang untuk membuat aplikasi yang dibutuhkan, yang bukan merupakan aplikasi bawaan *Handphone* atau *Smartphone* [[4](#Cor13)].

## 5.3 HASH

*Hashing* adalah transformasi aritmatik sebuah *string* dari karakter menjadi nilai yang merepresentasikan *string* aslinya. Menurut bahasanya, *hash* berarti memenggal dan kemudian menggabungkan. *Hashing* digunakan sebagai metode untuk menyimpan data dalam sebuah *array* agar penyimpanan data, pencarian data, penambahan data, dan penghapusan data dapat dilakukan dengan cepat. Ide dasarnya adalah menghitung posisi *record* yang dicari dalam *array*, bukan membandingkan *record* dengan isi pada *array*. Fungsi yang mengembalikan nilai atau kunci disebut fungsi *hash* dan *array* yang digunakan disebut *hash table*. *Hash table* menggunakan struktur data *array* asosiatif yang mengasosiasikan *record* dengan sebuah *field* kunci unik berupa bilangan (*hash*) yang merupakan representasi dari *record* tersebut [[5](#Sya13)].

Secara teori, kompleksitas waktu (T(n)) dari fungsi *hash* yang ideal adalah O(1). Untuk mencapai itu setiap *record* membutuhkan suatu kunci yang unik. Fungsi *hash* menyimpan nilai asli atau kunci pada alamat yang sama dengan nilai *hash*. Pada pencarian suatu nilai pada tabel hash, yang pertama dilakukan adalah menghitung nilai *hash* dari kunci atau nilai aslinya, kemudian membandingkan kunci atau nilai asli dengan isi pada memori yang beralamat nomor *hash*. Dengan cara ini, pencarian suatu nilai dapat dilakukan dengan cepat tanpa harus memeriksa seluruh isi tabel satu per satu [[5](#Sya13)].

Fungsi *Hash* (dilambangkan dengan h(k)) bertugas untuk mengubah k (*key*) menjadi suatu nilai dalam interval [0....X], dimana "X" adalah jumlah maksimum dari *record-record* yang dapat ditampung dalam tabel. Jumlah maksimum ini bergantung pada ruang memori yang tersedia. Fungsi *Hash* yang ideal adalah mudah dihitung dan bersifat *random*, agar dapat menyebarkan semua *key*. Dengan *key* yang tersebar, berarti data dapat terdistribusi secara seragam bentrokan dapat dicegah. Sehingga kompleksitas waktu model *Hash* dapat mencapai O(1), dimana kompleksitas tersebut tidak ditemukan pada struktur model lain. Ada beberapa macam fungsi *hash* yang relatif sederhana yang dapat digunakan dalam penyimpanan basis data [[5](#Sya13)]:

* Metode Pembagian Bersisa (*division-remainder method*)

Jumlah lokasi memori yang tersedia dihitung, kemudian jumlah tersebut digunakan sebagai pembagi untuk membagi nilai yang asli dan menghasilkan sisa. Sisa tersebut adalah nilai hashnya. Secara umum, rumusnya h(k)= k mod m. Dalam hal ini m adalah jumlah lokasi memori yang tersedia pada *array*. Fungsi *hash*tersebut menempatkan *record*dengan kunci K pada suatu lokasi memori yang beralamat h(k). Metode ini sering menghasilkan nilai *hash*yang sama dari dua atau lebih nilai aslinya atau disebut dengan bentrokan. Karena itu, dibutuhkan mekanisme khusus untuk menangani bentrokan yang disebut kebijakan resolusi bentrokan.

* Melipat (*folding*)

Metode ini membagi nilai asli ke dalam beberapa bagian, kemudian menambahkan nilai-nilai tersebut, dan mengambil beberapa angka terakhir sebagai nilai *hash*.

* Transformasi Radiks (*radix transformation*)

Karena nilai dalam bentuk *digital*, basis angka atau *radiks* dapat diganti sehingga menghasilkan urutan angka-angka yang berbeda. Contohnya nilai *desimal* (basis 10) bisa ditransformasikan kedalam *heksadesimal* (basis 16). Digit atas hasilnya bisa dibuang agar panjang nilai *hash*dapat seragam.

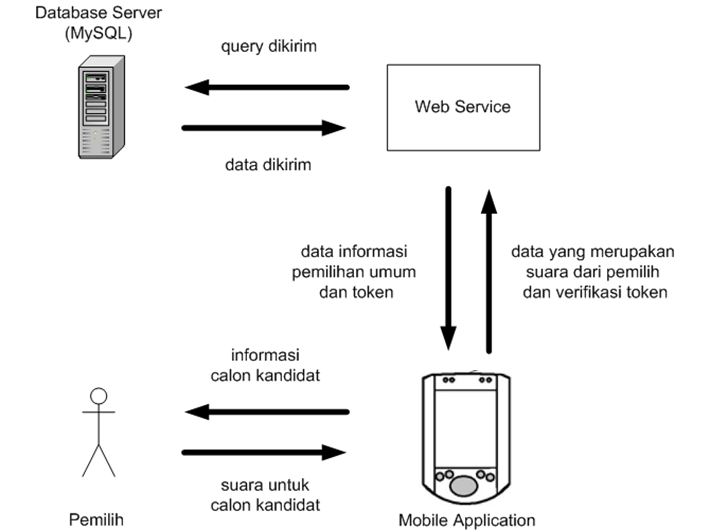
* Pengaturan ulang digit (*digit rearrangement*)

Metode ini mengubah urutan digit dengan pola tertentu. Contohnya mengambil digit ke tiga sampai ke enam dari nilai asli, kemudian membalikkan urutannya dan menggunakan digit yang urut terbalik itu sebagai nilai *hash*. Fungsi *hash*yang bekerja dengan baik untuk penyimpanan pada basis data belum tentu bekerja dengan baik untuk keperluan kriptografi  atau pengecekan kesalahan. Ada beberapa fungsi *hash*terkenal yang digunakan untuk keperluan kriptografi. Diantaranya adalah fungsi *hash message-diggest*, contohnya MD2, MD4, dan MD5, digunakan untuk menghasilkan nilai *hash*dari tanda tangan *digital* yang disebut *message-diggest*. Ada pula *Secure Hash Algorithm* (SHA), sebuah algoritma standar yang menghasilkan *message-diggest* yang lebih besar (60-bit) dan serupa dengan MD4.

## 5.4 TOKEN

Token adalah sebuah alat yang mengesahkan pengguna layanan bahwa pengguna memiliki otentikasi yang valid. Token berawal dari alat pembangkit token yang akan digunakan oleh setiap pengguna. Token bersifat one-life time password atau hanya dipakai sekali. Jadi, setelah token dipakai, pengguna akan menggunakan token yang baru untuk pemakaian selanjutnya.

# **6. METODOLOGI**



Gambar 1. Arsitektur Sistem

Aplikasi ini dibangun pada perangkat komunikasi bergerak agar pengguna dapat memberikan suara pilihan dimana saja dan kapan pun selama waktu yang pemilihan masih ada. Pada Gambar 1 dijelaskan aplikasi ini bekerja dengan menerima *input* berupa pilihan yang merupakan suara dari pemilih, kemudian *input* tersebut dikonversi menjadi sebuah data dan dienkripsi setelah itu akan dikirim ke basis data *server*. Digunakan *Web Service* untuk mengakses basis data yang berada pada *server*. Perangkat komunikasi bergerak akan berkomunikasi dengan *Web Service*, kemudian *Web Service* akan berkomunikasi dengan server. Setelah data diterima oleh *server* maka *server* akan memeriksa apakah pemilih telah memberikan suara karena setiap pemilih hanya mempunyai satu kesempatan untuk memilih kandidat yang ada. Jika pemilih belum memberikan suaranya maka data yang telah diterima *server* akan diolah dan dimasukkan ke basis data. Data yang dikomunikasikan bisa berupa XML (*Extensible Mark-up Language*) maupun JSON (*JavaScript Object Notation*).

Untuk menghindari duplikasi pengiriman data dengan menggunakan *login* yang berbeda di perangkat bergerak yang sama, maka dilakukan pemeriksaan ICCID dari perangkat bergerak yang akan digunakan untuk memberikan suara. Sehingga apabila sebuah nomor telpon telah digunakan untuk memberikan suara, maka nomor telpon tersebut tidak bisa lagi digunakan untuk melakukan pengiriman suara. Dan juga sebelum *server* menerima data berisi suara pemilih, perangkat bergerak milik pemilih akan menerima sebuah nomor *token* yang berfungsi untuk menghindari akun milik pemilih digunakan oleh orang lain.

Untuk menjaga keaslian data dari suara yang telah diberikan, maka data yang telah diterima dan dimasukkan ke *server* akan diubah dengan menggunakan algoritma *hash*. Sehingga suara yang telah diberikan oleh pemilih tidak dapat diubah oleh seseorang yang tidak bertanggung jawab.

Fitur yang akan diimplementasikan antara lain:

1. Informasi Calon Kandidat

Menampilkan informasi kandidat (nama, visi, misi, dan organisasi) yang ikut dalam pemilihan.

1. Vote Calon Kandidat

Menampilkan halaman untuk memilih kandidat.

1. Informasi Hasil Pemilihan Umum

Menampilkan total perhitungan suara yang telah diberikan oleh masyarakat

# **7. JADWAL PEMBUATAN TUGAS AKHIR**

Berikut merupakan jadwal pengerjaan tugas akhir ini:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tahapan** | **2013** | | | | | | | | | | | | | | | |
| **Maret** | | | | **April** | | | | **Mei** | | | | **Juni** | | | |
| **Penyusunan Proposal** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Studi Literatur** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Perancangan Sistem** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Implementasi** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Uji Coba dan Evaluasi** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Penyusunan Buku** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# **8. DAFTAR PUSTAKA**

x

|  |  |
| --- | --- |
| [1] | Institut Pertanian Bogor. (2003, Maret) [Online]. <http://repository.ipb.ac.id/bitstream/handle/123456789/14834/Pendahuluan%20G09mfi.pdf?sequence=11> |
| [2] | Ghassan Z. Qadah, "Computer Standards & Interfaces," *Electronic Voting Systems : Requirements, Design, and Implementation*, vol. 29, pp. 376 - 386, 2007. |
| [3] | N Safaat, *Android : Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Android*, Pertama ed. Bandung, Indonesia, 2011. |
| [4] | Janssen Cory Technopedia. (2013, Maret) Janalta Interactive Inc. [Online]. <http://www.techopedia.com/definition/4220/android-sdk.> |
| [5] | Syazdi Ayhodia N. (2013, Maret) Hashing. [Online]. <http://syazdiayhodian.blogspot.com/2011/06/hashing.html> |

x