**BAB IV**

**ANALISIS DAN DESAIN SISTEM**

Analisis dan perancangan sistem memerlukan tahapan yang sistematis untuk menghasilkan aplikasi yang bbaik dan bersesuaian dengan kegunaan dan tujuannya. Tahap awal dari analisis adalah menganalisis kebutuhan-kebutuhan sistem mulai dari kebutuhan pengguna, kebutuhan non-fungsional, dan kebutuhan fungsional. Sedangkan untuk tahap perancangan aplikasi yaitu perancangan proses transmisi data, proses kriptografinya dan proses antarmuka.

1. **Analisa Sistem**

Analisis sistem (*system analysis*) dapat didefinisikan sebagai penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasikan permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikan. Analisis dapat juga diartikan memahami sistem pemikiran yang kompleks dengan memecahnya ke dalam unsur-unsur yang lebih sederhana sehingga hubungan antar unsur-unsur itu menjadi jelas.

1. **Analisis Sistem Berjalan**

Pada penelitian sebelumnya telah dibangun sebuah sistem pengamanan basis data berbasis web dengan menggunakan algoritma RC4. Dalam sistem tersebut hanya mengenkripsi kolom password saja, sedangkan isi basis data yang lainnya tidak di enkripsi, juga belum terdapat pengukuran waktu enkripsi/dekripsi basis data, jadi tidak dapat diketahui seberapa berpengaruh panjang kunci dan panjang data terhadap proses enkripsi.

Berdasarkan penelitian dan pengamatan langsung pada model konvensional arsitektur aplikasi pengaman basis data berbasis web tersebut adalah sebagai berikut:



**Gambar 4.1** Analisis sistem berjalan

Gambar 4.1 diatas memperlihatkan model konvensional arsitektur pengamanan basis data berbasis web pada penelitian Aditya Eka Arifyanto dengan menggunakan algoritma RC4 dengan panjang kunci yang tidak diketahui. Sistem ini hanya mengamankan kolom password saja sedangkan isi basis data lainnya tidak.

Di penelitian kali ini penulis mencoba mengenkripsi keseluruhan isi basis data, dan menambahkan perhitungan waktu enkripsi/dekripsi basis data berdasarkan panjang karakter kunci ditambah panjang karakter isi basis data. Sehingga dapat diketahui seberapa berpengaruh panjang karakter kunci ditambah panjang karakter isi basis data terhadap waktu enkripsi/dekripsi basis data.

1. **Analisis Sistem Usulan**

Sistem yang akan dibangun pada penelitian kali ini adalah pengamanan basis data menggunakan algoritma kriptografi bersifat *stream cipher* yaitu RC4. Lebih jelasnya dapat dilihat pada gambar berikut ini:



**Gambar 4.2** Metode pengamanan RC4 Usulan

|  |  |
| --- | --- |
| Enkripsi | Dekripsi |
|  | |

**Gambar 4.3** Diagram alir sistem usulan

1. **Analisis Metode Pengamanan**

Algoritma RC4 memiliki dua fase, *setup* kunci dan pengenkripsian. Pada bagian ini akan di jelaskan proses pengaamanan mulai dari *setup* kunci sampai proses pengenkripsian.

1. ***Setup* kunci**

*Setup* untuk kunci adalah fase pertama dan yang paling sulit dalam algoritma ini. Dalam setup S-bit kunci (S merupakan panjang dari kunci), kunci enkripsi digunakan untuk menghasilkan variabelenkripsi yang menggunakan dua buah *array*, *state* dan kunci, dan sejumlah-S hasil dari operasi penggabungan. Operasi penggabungan ini terdiri dari pemindahan (*swapping*) *byte*, operasi modulo, dan rumus lain. Operasi modulo merupakan proses yang menghasilkan nilai sisa dari satu pembagian. Sebagai contoh, 11 dibagi 4 adalah 2 dengan sisa pembagian 3, begitu juga jika tujuh modulo empat maka akan dihasilkan nilai tiga.



**Gambar 4.4** Proses *setup* kunci RC4

Contoh perhitungan maual *setup* kunci RC4:

Pertama menginisialisasi *array* S 4 bit sehingga terbentuk *state*-*array* S dan *state*-*array* K sebagai berikut:

*Array* S *Array* K

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 1 | 2 | 3 |

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | 5 | 7 | 3 |

Inisialisasi i dan j dengan 0 kemudian dilakukan KSA agar tercipta *state-array* yang acak. Penjelasan iterasi lebih lanjut dapat dijelaskan sebagai berikut:

* **Iterasi 1**

i = 0

j = (0 + S[0] + K [0 mod 4]) mod 4

= (0 + 0 + 2) mod 4 = 2

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | 1 | 0 | 3 |

*Swap* (S[0],S[2])

Hasil *array* S

* **Iterasi** **2**

i = 1

j = (2 + S[1] + K [1 mod 4]) mod 4

= (2 + 1 + 5) mod 4 =0

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 0 | 3 |

*Swap* (S[1],S[0])

Hasil *array* S

* **Iterasi 3**

i = 2

j = (0 + S[2] + K [2 mod 4]) mod 4

= (0 + 0 + 7) mod 4 = 3

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 0 |

*Swap* (S[2],S[3])

Hasil *array* S

* **Iterasi 4**

i = 3

j = (3 + S[3] + K [3 mod 4]) mod 4

= (3 + 0 + 3) mod 4 = 2

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 0 | 3 |

*Swap* (S[3],S[2])

Hasil *array* S

Setelah melakukan KSA, akan dilakukan PRGA. PRGA akan dilakukan sebanyak 4 kali dikarenakan plainteks yang akan dienkripsi berjumlah 4 karakter. Hal ini disebakan karena dibutuhkan 1 kunci dan 1 kali pengoperasian XOR untuk tiap-tiap karakter pada plainteks. Berikut adalah tahapan penghasilan kunci enkripsi dengan PRGA.

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 0 | 3 |

*Array* S

Inisialisasi

i = 0

j = 0

* **Iterasi** **1**

i = (0 + 1) mod 4 = 1

j = (0 + S[1]) mod 4 = (0+ 2) mod 4 = 2

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 0 | 2 | 3 |

swap (S[1],S[2])

K1 = S[(S[1]+S[2]) mod 4] = S[2 mod 4] = 2

K1 = 00000010

* **Iterasi** **2**

i = (1 + 1) mod 4 = 2

j = (2 + S[2]) mod 4 = (2+ 2) mod 4 = 0

swap (S[2],S[0])

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | 0 | 1 | 3 |

K2 = S[(S[2]+S[0]) mod 4] = S[3 mod 4] = 3

K2 = 00000011

* **Iterasi 3**

i = (2 + 1) mod 4 = 3

j = (0 + S[3]) mod 4 = (0+ 3) mod

4 = 3

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 0 | 2 | 3 |

swap (S[3],S[3])

K3 = S[(S[3]+S[3]) mod 4] = S[6 mod 4] = 2

K3 = 00000010

* **Iterasi 4**

i = (3 + 1) mod 4 = 0

j = (3 + S[0]) mod 4 = (3+ 1) mod 4 = 0

swap (S[0],S[0])

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 0 | 2 | 3 |

K1 = S[(S[0]+S[0]) mod 4] = S[2 mod 4] = 2

K1 = 00000010

Jadi kunci untuk enkripsi/dekripsi adalah:

|  |  |  |  |
| --- | --- | --- | --- |
| K1 | K2 | K3 | K4 |
| 00000010 | 00000011 | 00000010 | 00000010 |

1. **Proses enkripsi dan dekripsi**

Setelah menemukan kunci untuk tiap karakter, maka dilakukan operasi XOR antara karakter pada plaintext dengan kunci yang dihasilkan. Berikut adalah tabel ASCII untuk tiap-tiap karakter pada plainteks yang digunakan.



**Gambar 4.5** Proses enkripsi RC4

Proses dekripsi pada metode RC4 merupakan kebalikan dari proses enkripsi, dimana pada proses enkripsi operasi XOR dilakukan antara plainteks dengan kunci untuk mendapatkan cipherteks, sedangkan pada proses dekripsi operasi XOR dilakukan antara cipherteks dengan kunci untuk mendapatkan plainteks.



**Gambar 4.6** Proses dekripsi RC4

Contoh perhitungan manual proses enkripsi dan dekripsi:

**Tabel 4.1** Proses XOR kunci dengan plaintext pada enkripsi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | H | A | L | O |
| Plaintext | 01001000 | 01000001 | 01001100 | 01001111 |
| Key | 00000010 | 00000011 | 00000010 | 00000010 |
| Ciphertext | 01001010 (L) | 01000010 (B) | 01001110 (N) | 01101101 (M) |

**Tabel 4.2** Proses XOR kunci dengan ciphertext pada dekripsi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | L | B | N | M |
| Ciphertext | 01001010 | 01000010 | 01001110 | 01101101 |
| Key | 00000010 | 00000011 | 00000010 | 00000010 |
| Plaintext | 01001000 (H) | 0100000 (A) | 01001100 (L) | 01001111 (O) |

1. **Desain Sistem**

Tahaap ini dilakukan untuk membangun rancangan sistem pengamanan basis data berdasarkan hasil analisa. Himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (*Unified Modeling Language*) akan dijelaskan berikut beserta Desain *Interface*nya.

1. **Desain Model dengan UML**

*Unified Modeling Language* merupakan salah satu alat bantu yang dapat digunakan dalam mendesain program dan bahasa pemrograman yang berorientasi objek.

1. ***Functional* *Requirment***

Pada tahap ini akan dituliskan sistem yang akan dibuat dengan semua fungsi yang diperlukan.

|  |
| --- |
| ***Functional* *Requirment***  Membuat sistem pengamanan basis data dengan menggunakan algoritma RC4. Aplikasi ini dapat melakukan hal berikut:   1. Melakukan pemilihan basis data (.sql) 2. Membangkitkan kunci RC4 3. Melakukan proses enkripsi secara keseluruhan pada plainteks basis data 4. Melakukan perhitungan waktu pengenkripsian basis data 5. Melakukan proses dekripsi secara keseluruhan pada cipherteks basis data 6. Melakukan perhitungan waktu pengdekripsian basis data |

1. **Usecase Diagram**

**Gambar 4.7** Usecase Diagram

1. **Activity Diagram**



**Gambar 4.8** Activity diagram proses pilih basis data



**Gambar 4.9** Activity diagram enkripsi basis data



**Gambar 4.10** Activity diagram dekripsi basis data



**Gambar 4.11** Activity diagram hitung waktu enkripsi basis data



**Gambar 4.12** Activity diagram hitung waktu dekripsi basis data



**Gambar 4.13** Activity diagram Simpan plainteks/cipherteks basis data

1. **Sequence Diagram**

****

**Gambar 4.14** Proses pilih plainteks basis data



**Gambar 4.15** Proses pilih cipherteks basis data

****

**Gambar 4.16** Proses enkripsi basis data



**Gambar 4.17** Proses dekripsi basis data



**Gambar 4.18** Proses hitung waktu enkripsi basis data



**Gambar 4.19** Proses hitung waktu dekripsi basis data



**Gambar 4.20** Proses simpan plainteks basis data



**Gambar 4.21** Proses simpan cipherteks basis data

1. **Desain User Interface**

Pada penelitian ini akan dibangun sebuah sistem untuk pengamanan basis data dengan *user interface* yang dapat di ilustrasikan dengan gambar modelnya.

1. **Desain *interface* menu utama**



**Gambar 4.22** Desain form menu utama

Desain menu utama untuk pengamanan basis data, di form ini terdapat deskripsi singkat algoritma RC4 yaitu gambar cara kerja algoritma dan sejarah algoritma. pada sebelah kanan atas terdapat tombol menu home untuk kembali ke menu utama, tombol menu about untuk membuka form biodata singkat pembuat program, tombol menu enkripsi untuk membuka form enkripsi dan tombol menu dekripsi untuk membuka form dekripsi.

1. **Desain *interface* untuk form *input***



**Gambar 4.23** Desain form enkripsi *input*



**Gambar 4.24** Desain form dekripsi *input*

Desain antar muka untuk form *input* enkripsi dan dekripsi, di form ini terdapat tombol browse unntuk meng*input* file plainteks/ cipherteks basis data (.sql), dan *teks box* untuk memasikan kunci.

1. **Desain *interface* form *output***



**Gambar 4.25** Desain form enkripsi *output*



**Gambar 4.26** Desain form dekripsi *output*

Desain antar muka untuk form *output* enkripsi/ dekripsi, di form ini terdapat dua *list box* yaitu untuk menampilkan plainteks dan cipherteks setelah terenkripsi/ dekripsi, *teks box* kunci untuk menampilkan kunci, waktu kecepatan enkripsi/ dekripsi, *teks box* untuk memasukan nama file sebelum di simpan, dan tombol simpan untuk menyimpan file plainteks/ cipherteks basis data.