

## **BAB III**

### **ANALISIS DAN PERANCANGAN**

#### **III.1. Analisis**

Sebelum merancang sebuah sistem, perlu dilakukan analisis terlebih dahulu. Analisis sistem adalah proses menentukan kebutuhan sistem, apa yang harus dilakukan sistem untuk memenuhi kebutuhan untuk membangun sistem. Dengan adanya analisis sistem, sistem yang dirancang akan lebih baik dan memudahkan pengembang sistem dalam perbaikan apabila pada kemudian hari ditemukan kesalahan atau kekurangan. Berikut ini dilakukan analisis terhadap hal-hal yang berhubungan dengan sistem.

##### **III.1.1 Analisis Masalah**

Berbagai bentuk informasi berupa data dan berkas sering kali kita temui apabila mengoperasikan sebuah komputer ataupun telepon genggam. Yang menjadi persoalan adalah bagaimana kita bisa mengetahui keaslian, keamanan dan keutuhan dari data tersebut. Secara kasat mata, sangat susah untuk membedakan jika terjadi suatu keanehan di dalam data tersebut. Hal-hal seperti ketidakutuhan data, perubahan data, bahkan yang lebih parah berupa tersisipnya virus di dalam file dapat mengganggu kinerja seseorang terhadap file tersebut. Untuk mengetahui berkas yang diterima masih asli atau sudah terjadi perubahan di dalamnya, metode MD5 dapat digunakan yang merupakan salah satu solusi untuk menangani

persoalan kerahasiaan, autentikasi, keutuhan dan non-repudiation dalam mengirim atau menerima informasi/data digital.

### III.1.2 Analisis Spesifikasi

Dalam pembuatan aplikasi Android ini dibutuhkan spesifikasi dasar, antara lain:

- a. Menggunakan *Intel processor dual-core*.
- b. Menggunakan RAM 2 GB.
- c. Menggunakan *Windows 7*.
- d. Program aplikasi dibuat menggunakan *Eclipse Android SDK*.
- e. *Eclipse Android SDK* membutuhkan JDK (*Java Development Kit*) versi 1.7.
- f. Metode *checksum* yang digunakan adalah MD5 (*Message Digest 5*).
- g. Data yang diuji adalah teks dan juga *file* berekstensi *.exe*.

Nantinya aplikasi yang dibuat ini dapat berjalan di dalam emulator dengan komputer yang memiliki spesifikasi sebagai berikut:

- a. Menggunakan *intel processor dual-core* atau yang lebih tinggi.
- b. Sistem membutuhkan Sistem Operasi *Windows 7* atau *Windows 8* yang *compatible* dengan berbagai aplikasi yang ada.
- c. Memori minimal 1 GB untuk *Windows 7*, dan 1GB untuk *Windows 8*.

Sedangkan untuk pengujian langsung pada perangkat Android, spesifikasi yang dibutuhkan pada perangkat Android tersebut antara lain:

- a. Perangkat Android bersistem operasi minimal ICS (*Ice Cream Sandwich*) versi 4.0.
- b. Memiliki RAM minimal 512 MB.

### **III.2.Strategi Pemecahan Masalah**

Setelah melihat masalah-masalah yang ditemukan serta mengingat betapa pentingnya kebutuhan akan pengecekan data atau informasi, penulis memutuskan untuk merancang dan membuat sebuah program aplikasi *checksum* yang menggunakan metode MD5 (*Message Digest 5*). MD5 dimanfaatkan dalam berbagai aplikasi keamanan, dan umumnya digunakan untuk menguji integritas sebuah file. MD5 sering digunakan karena *output* yang dihasilkannya bersifat 'one way hash'. Berapapun string yang dibangkitkan, hasilnya tetap sepanjang 32 karakter. Keluaran dari MD5 berupa 4 buah blok yang masing-masing 32-bit yang mana akan menjadi 128-bit yang biasa disebut nilai hash.

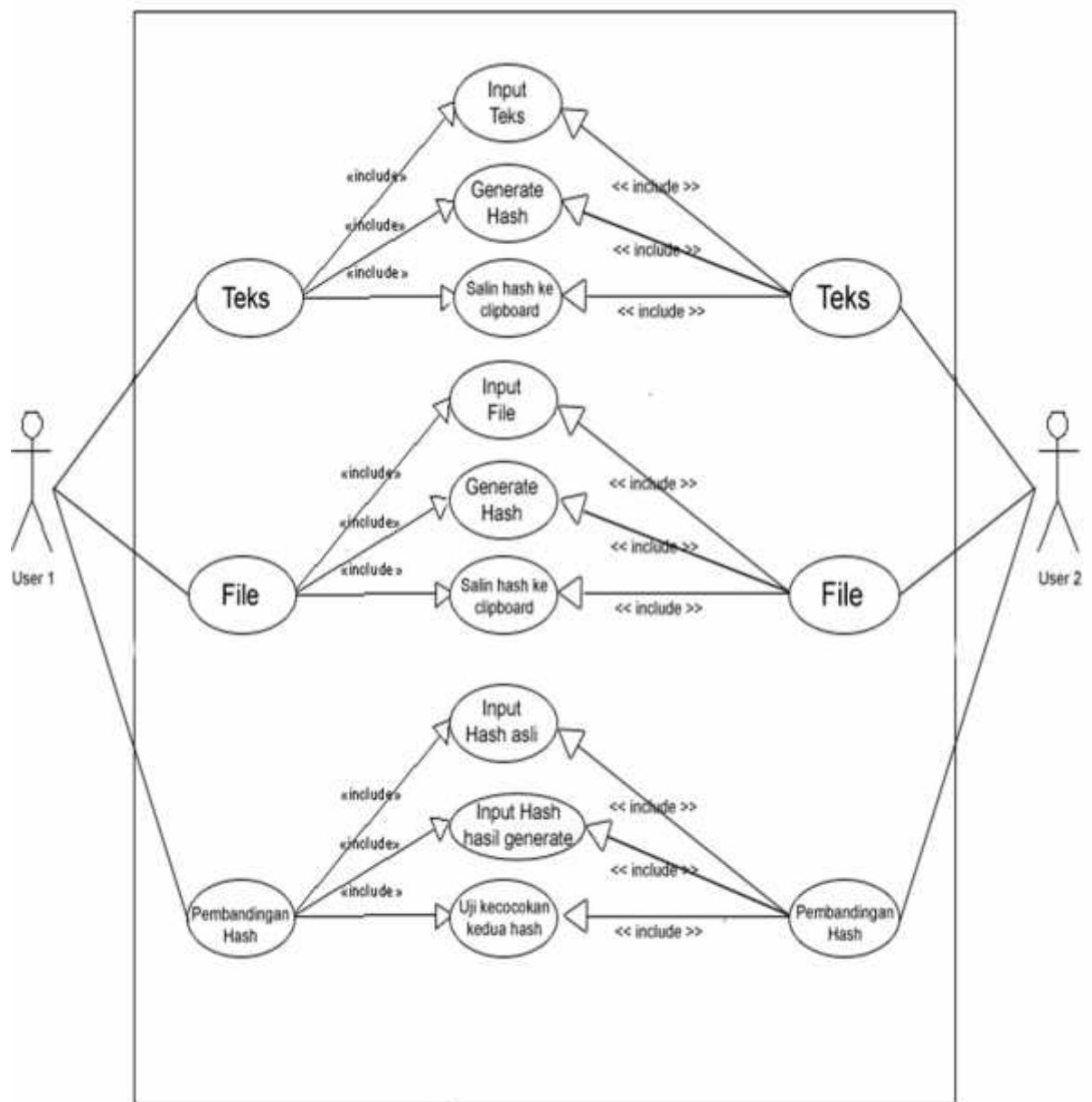
### **III.3. Perancangan**

Tahap perancangan perangkat lunak menggunakan rancangan layar dan metode UML (*Unified Modelling Language*), UML sendiri meliputi *use case*, *sequence diagram*, dan *activity diagram*.

#### **III.3.1 Use Case Diagram**

Kegiatan interaksi antara aktor terhadap sistem ditunjukkan pada *use case diagram*, Aktor yang terlibat dalam kegiatan tersebut adalah *user*. *User* memiliki

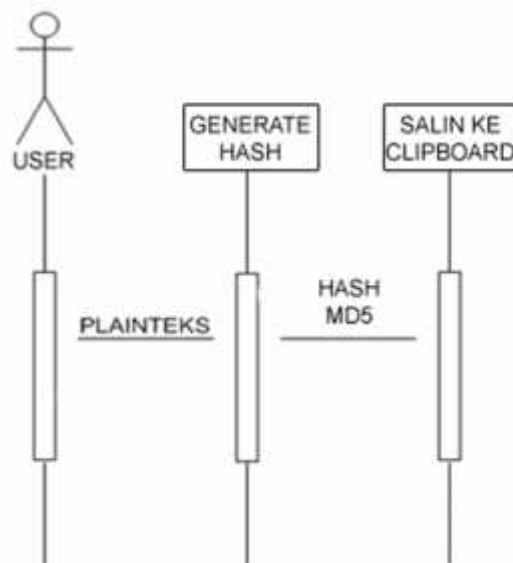
tiga *use case*, yaitu *form* pembangkit hash MD5 untuk data berbentuk teks, *form* pembangkit hash MD5 untuk data berbentuk file berekstensi .exe, serta *form* untuk perbandingan hash asli dengan hash dari data yang diinput oleh user. *Use case* ini kemudian memiliki beberapa *use case* lagi. *Use case diagram* perangkat lunak yang dibangun terlihat pada gambar III.1 berikut.



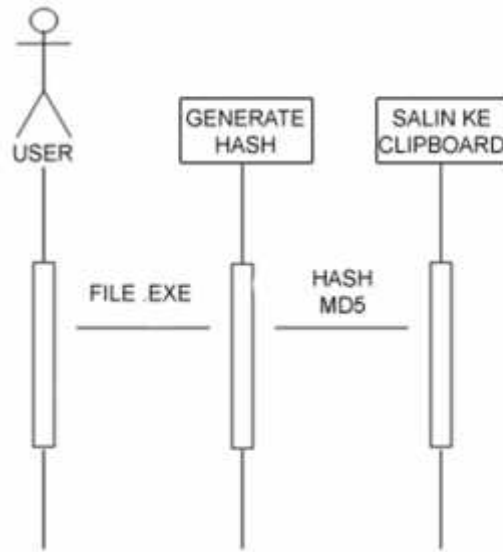
**Gambar III.1.** Use Case Aplikasi Checksum Hash MD5

### III.3.2 Sequence Diagram

Proses yang dilakukan oleh tiap aktor digambarkan pada *sequence diagram* pada gambar III.2 dan gambar III.3 sebelum melakukan proses pembangkitan/generate hash MD5, user harus menginput *plainteks* ataupun file yang ingin diuji. Setelah menginput teks ataupun *file* yang akan diuji, selanjutnya adalah proses pembangkitan hash dari data masukan *user* menggunakan metode *checksum* MD5. Proses ini akan menghasilkan 32 karakter unik dari data tersebut. Kemudian setelah proses pembangkitan hash berhasil, hash tersebut bisa disimpan ke dalam clipboard untuk selanjutnya bisa dibandingkan dengan hash asli.



**Gambar III.2** *Sequence Diagram* proses pembangkitan hash MD5 untuk data berbentuk teks



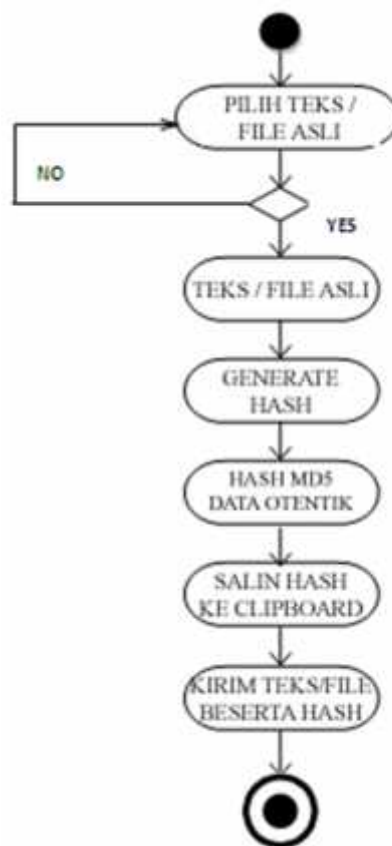
**Gambar III.3** *Sequence Diagram* proses pembangkitan hash MD5 untuk data berbentuk *file .exe*



**Gambar III.4** *Sequence Diagram* proses perbandingan hash MD5

### III.3.3 Activity Diagram

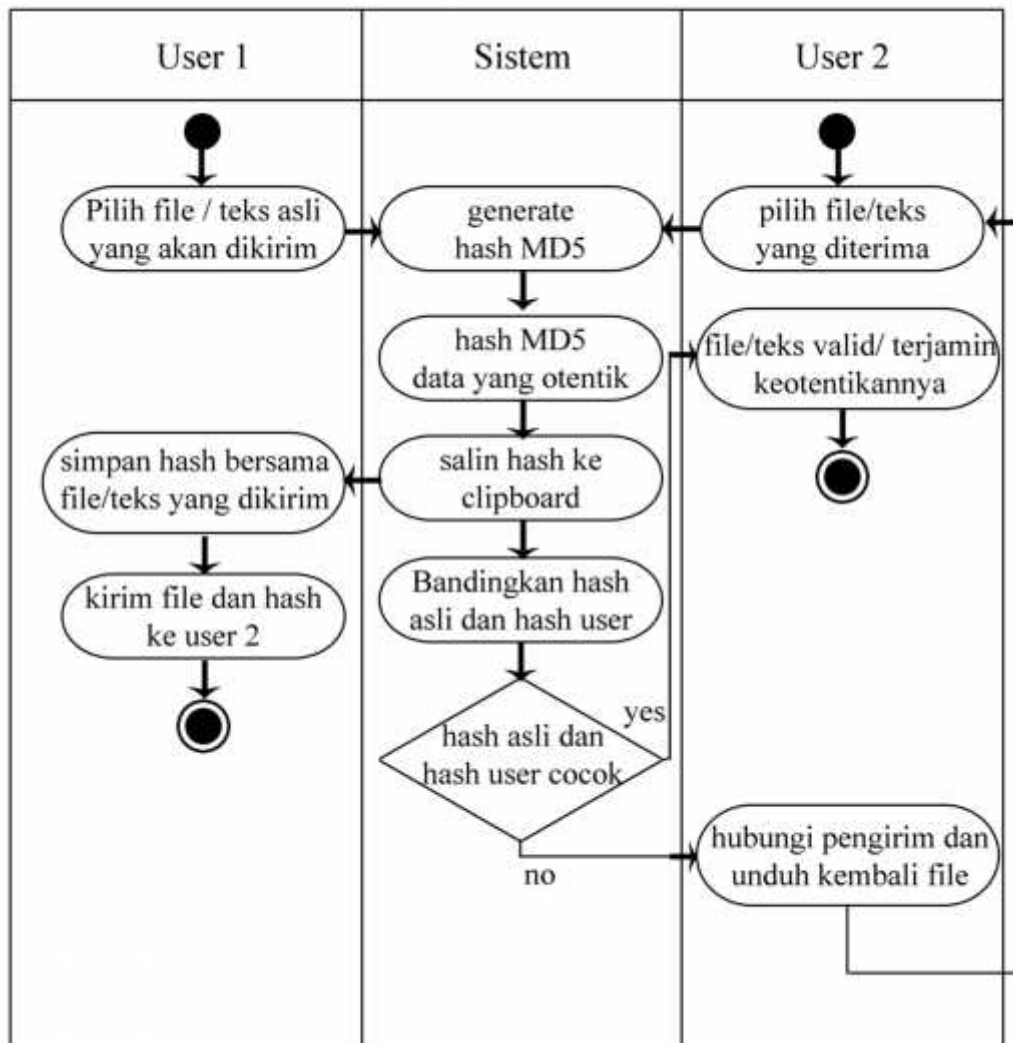
*Activity diagram* merupakan diagram yang menggambarkan seluruh aktivitas sistem secara terperinci yang melibatkan pengirim dan penerima. Proses dimulai dengan pemilihan teks maupun *file* yang akan diuji oleh *user*. Selanjutnya *user* memasukkan data tersebut untuk memproses pembangkitan hash menggunakan metode *checksum* MD5. Proses generate/pembangkitan hash akan menghasilkan *hash* MD5 berupa 32 karakter unik dari data yang diuji coba. Selanjutnya hash MD5 tersebut bisa kita simpan ke *clipboard* agar nantinya hash ini bisa dibandingkan dengan hash otentik/asli yang biasa diberikan pemilik asli data.



**Gambar III.5.** Activity Diagram pembangkitan dan perbandingan *hash* MD5

Dari perbandingan tersebut bisa kita lihat, apabila hash cocok berarti data sampai di tangan *user* dengan aman tanpa kerusakan selama proses pengiriman dari pemilik asli data ke tangan *user*. Bila hash tidak cocok, maka keterjaminan integritas data bisa dipertanyakan, karena kemungkinan yang terjadi pada data tersebut bisa jadi telah diubah oleh orang lain, rusak/terjadi *corrupt* saat pengiriman data, maupun adanya kemungkinan tersisip virus, sehingga mengakibatkan adanya perbedaan hash.

Perubahan data sedikit saja tentu akan mempengaruhi hash yang dibangkitkan dari suatu data, sehingga metode *checksum hash* sangat berguna untuk memastikan/memverifikasi data yang secara kasat mata tidak dapat dibedakan antara yang asli dengan data yang diterima *user*. Proses checksum hash MD5 setidaknya membutuhkan dua pelaku, yaitu aktor pertama sebagai pengirim/pemilik asli file, dan aktor kedua sebagai penerima file yang berhak. Adapun prosesnya dapat digambarkan dengan tabel berisi *activity diagram* seperti terlihat pada tabel III.1. berikut:

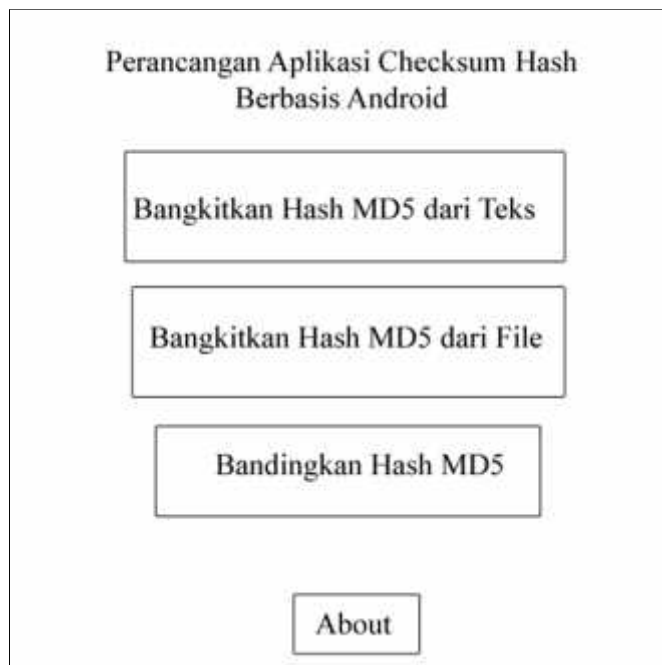


**Tabel III.1.** Activity Diagram Pembangkitan dan Perbandingan Hash MD5

Antar Dua User

### III.3.4 Rancang Layar

#### A. Antarmuka *Form* Utama



**Gambar III.7.** Tampilan *Form* Utama

Rancangan *form* ini dibuat sebagai *form* utama dimana di *form* ini akan menjadi tampilan awal. *Form* ini memuat 3 tombol utama fungsi hash MD5 yang masing-masingnya bisa dipilih dengan cara menyentuh *button* yang diinginkan, ditambah satu tombol “*about*” yang berisi informasi singkat mengenai penulis serta deskripsi aplikasi.

### B. Antarmuka *Form* Hitung *Hash* Dari Teks

HASH DARI TEKS	HASH DARI FILE	BANDINGKAN HASH
Masukkan teks yang akan diuji:		
<i>TextBox</i>		
<input type="button" value="Bangkitkan Hash MD5"/> <input type="button" value="Bersihkan Teks"/> <input type="button" value="Salin Hash ke Clipboard"/>		
Checksum Hash MD5:		
<i>TextView</i>		

**Gambar III.8.** Tampilan *Form* Hitung *Hash* Dari Teks

Rancangan *form* ini dibuat untuk melakukan proses pembangkitan hash MD5 yang berasal dari teks yang diinput oleh *user*.

### C. Antarmuka *Form* Hitung *Hash* Dari *File*

HASH DARI TEKS	HASH DARI FILE	BANDINGKAN HASH
<input type="button" value="Buka File"/> <input type="button" value="Bangkitkan Hash MD5"/> <input type="button" value="Salin Hash ke Clipboard"/>		
File Anda:		
<i>TextView</i>		
Checksum Hash MD5:		
<i>TextView</i>		

**Gambar III.9.** Tampilan *Form* Hitung *Hash* Dari *File*

Rancangan *form* ini dibuat untuk melakukan proses pembangkitan hash MD5 yang berasal dari *file* yang diinput oleh *user*.

#### D. Antarmuka *Form* Perbandingan Hash Otentik dengan Hash Data User

HASH DARI TEKS	HASH DARI FILE	BANDINGKAN HASH
Masukkan hash MD5 otentik/asi:		
<input type="text"/>		
<input type="button" value="Tempel"/>	<input type="button" value="Bersihkan"/>	
Masukkan hash MD5 milik user:		
<input type="text"/>		
<input type="button" value="Tempel"/>	<input type="button" value="Bersihkan"/>	
<input type="button" value="Bandingkan Hash MD5"/>		
Hasil:		
<input type="text"/>		

**Gambar III.10.** Tampilan *Form* Perbandingan Hash Otentik dengan Hash Data User

Rancangan *form* ini dibuat untuk melakukan proses perbandingan hash MD5 yang berasal dari *file* yang diinput oleh *user* dengan hash MD5 asli.