# Implementasi Algoritma Secure Hash Algorithm-1 untuk Pengamanan Application Programming Interface dengan Konsep Hash-Based Message Authentication Code Pada Aplikasi Sistem Tryout Berbasis Web dan Android

**Dewa Fakha Shiva[(1)], Muhammad Ainur Rony[(2)]**
[(1)(2)]Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
Jl. Raya Ciledug, Petukangan Utara, Jakarta Selatan 12260
Telp. (021) 5853753, Fax. (021) 5853752
E-mail : dewafakhashiva@gmail.com[1)], ainur.rony@gmail.com[2)]

***Abstrak***

*Ujian tryout sering kali dilakukan untuk mengukur seberapa baik siswa menerima pelajaran yang diberikan oleh guru. Dari hasil tersebut dapat dilakukan analisa seberapa besar siswa tersebut akan berhasil menjawab soal ujian yang nantinya akan diberikan. Ujian tryout mengalami beberapa perubahan, mulai dari ujian berbasis kertas, hingga saat ini ujian berbasis komputer. Ujian tryout akan menjadi lebih mudah jika dilakukan dengan menggunakan smartphone, seperti Android. Karena untuk siswa yang tidak memiliki komputer atau pihak sekolah yang belum bisa memfasilitasi untuk memberikan komputer untuk ujian akan lebih mudah untuk memberikan smartphone sebagai fasilitas untuk ujian tryout. Jika ujian dilakukan melalui smartphone, data soal ujian akan mengalami perubahan dalam cara pengiriman. Data soal ujian akan dikirimkan melalui API (Application Programming Interface) dan akan lebih baik digunakan pengamanan API dengan menggunakan SHA-1 (Secure Hash Algorithm-1), yang nantinya data soal ujian akan dilakukan proses hashing agar tidak dapat terbaca secara jelas, dan dilakukan pengiriman dari server ke smartphone dan dilakukan autentikasi untuk memastikan data soal tersebut sama atau tidak dengan soal ujian yang dikirimkan dari server. Data soal bisa diinputkan melalui web dan bisa diterima melalui Android.*

**Kata kunci:** SHA1, Pengamanan API, HMAC, Web, Android.

## 1. PENDAHULUAN

Sistem ujian di Indonesia masih banyak yang menggunakan kertas dan masih serba manual. Dengan kemajuan teknologi membuat adanya perkembangan dengan sekolah yang menggunakan ujian berbasis komputer yang mana ujiannya dilakukan secara manual dan pengecekannya dilakukan dengan menggunakan komputer. Beberapa waktu belakangan ini muncul lagi perkembangan yang dilakukan dari dinas pendidikan dengan ujian yang berbasis komputer dimana siswa akan melakukan ujian dengan menggunakan komputer dan juga dicek langsung dengan komputer. Dengan ujian yang dilakukan dikomputer ini, menjadikan ujian dan pengecekan hasil ujian menjadi lebih cepat. Untuk melakukan latihan ujian atau *tryout* juga akan dilakukan dengan menggunakan komputer. Akan lebih mudah dan fleksibel untuk para siswa melakukan ujian *tryout* dengan menggunakan *smartphone* masing-masing. Karena dengan semakin canggih teknologi yang ada memungkinkan ujian *tryout* dapat dilakukan pada *smartphone*.

Pada model pengiriman data yang digunakan adalah menggunakan API (*Application Programming Interface*) dan untuk keamanan akan dilakukan saat pengiriman data ini pada API. Pada penelitian ini, algoritma yang akan digunakan adalah SHA-1 dengan dikombinasikan dengan konsep HMAC (*Hash-based Message Authentication Code*) untuk melakukan *hashing* pada proses pengiriman dari *server* ke *client* Android dengan menggunakan API (*Application Programming Interface*) yang di dalamnya terdapat proses *login*, pengiriman dan penerimaan data ujian. Berdasarkan penelitian terdahulu, belum ditemukan penggunaan SHA-1 dengan dikombinasikan dengan HMAC untuk melakukan *hashing* pada data ujian yang akan dikirim dan digunakan pada sistem ujian di Indonesia.

Berdasarkan dari latar belakang yang telah dipaparkan sebelumnya, telah beberapa kali disinggung masalah keamanan data dari segi pengiriman dan penerimaan.

Maka dari itu, tujuan dari penelitian ini adalah menambah sistem pengamanan API pada *website* sistem *tryout* dan juga pada aplikasi android dengan mengimplementasikan algoritma *hashing Secure Hash Algorithm-1* (SHA-1) dipadukan dengan konsep *Hash-based Message Authentication Code* (HMAC).

Dalam penelitian ini metode penelitian yang digunakan dengan model *Software Development Life Cycle* (SDLC) adalah *Prototyping*. Tahapan SDLC dengan metode *Prototyping* meliputi tahapan analisis masalah, literatur review, analisis kebutuhan, pengumpulan data, perencanaan aplikasi, pengkodean, pengujian.

## 2. METODE PENELITIAN

### 2.1. Metode Penelitian

Dalam penelitian tugas akhir ini, metode penelitian yang penulis lakukan, antara lain :

- Analisis Masalah
  Pada tahap ini, analisis masalah sangat dibutuhkan untuk mengetahui masalah yang terdapat pada proses belajar yang kemudian akan diangkat menjadi topik penelitian.
- Literatur Review
  Literatur dibutuhkan sebagai landasan teori mengenai masalah yang akan diteliti. Literatur yang kami gunakan berupa jurnal-jurnal, buku yang berhubungan dengan topik penelitian yang akan dijadikan referensi dalam karya ilmiah.
- Analisis Kebutuhan
  Dalam tahap ini diperlukan adanya analisis terhadap kebutuhan sistem, data-data yang diperlukan terkait dengan penelitian ini.
- Pengumpulan Data
  Pada tahap ini, data yang digunakan adalah data sample soal yang dibuat sendiri oleh penulis agar dapat melanjutkan proses pengujian program yang dibuat.
- Perancangan Aplikasi
  Pada tahap ini, penulis membuat rancangan dalam bentuk mock-up yang dapat memvisualisasikan serta merepresentasikan aplikasi yang akan dibuat.
- Pengkodean
  Pada tahap ini, penulis menerjemahkan rancangan aplikasi yang telah dibuat ke dalam bahasa pemrograman. Bahasa pemrograman yang digunakan untuk membuat aplikasi ini adalah bahasa pemrogramanan Java dan juga PHP.
- Pengujian
  Setelah tahap pengkodean selesai, maka dilakukan pengujian terhadap aplikasi yang sudah dibuat.

Lembaga Pemerintah dan Sekolah memiliki peranan penting sebagai pelaksana ujian yang selalu diadakan tiap tahunnya. Mulai dari menyediakan soal yang akan diujikan, memastikan soal tersebut sampai dengan aman ke tiap lokasi ujian hingga melakukan pemeriksaan hasil ujian. Dengan semakin berkembangnya teknologi, sudah sewajarnya banyak kemajuan yang juga ikut mengisi bagian dalam sisi pendidikan. Seperti beralihnya ujian dengan menggunakan kertas menjadi ujian yang berbasis komputer. Dengan ada beberapa dana sekolah yang cukup terbatas akan lebih baik jika dapat mengurangi pengeluaran untuk melakukan ujian. Seperti halnya ujian yang menggunakan komputer, akan lebih mudah dan murah untuk bisa melakukan ujian dengan menggunakan smartphone Android dan juga diiringi dengan keamanan yang melindungi soal ujian yang harus diujikan kepada para siswa.

Oleh karena itu, diperlukan penelitian untuk mengetahui cara yang tepat untuk mengembangkan aplikasi untuk melakukan ujian pada smartphone Android dan melakukan pengiriman data dari pusat dan diterima oleh pihak sekolah untuk dilakukan proses ujian oleh para siswa dan juga penyuluhan cara penggunaannya agar meminimalisir terjadinya kesalahan.

Dari analisis masalah yang telah diuraikan di atas, diperlukan penelitian untuk mengetahui metode yang tepat untuk mengirim data soal ujian ke dalam smartphone agar ujian menjadi lebih mudah dan murah namun juga data soal ujian tetap aman untuk dikirimkan. Solusi yang bisa digunakan adalah dengan menggunakan algoritma hashing untuk melakukan autentikasi agar hanya yang berhak dan memiliki izin saja yang bisa melakukan akses kepada soal tersebut. Algoritma hashing yang bisa digunakan salah satunya adalah Secure Hash Algorithm-1 (SHA-1) dengan dikombinasikan dengan konsep Hash-based Message Authentication Code (HMAC) agar menjadi lebih aman karena untuk penggunaan hanya SHA-1 saja akan kurang efektif mengingat kurang amannya SHA-1 karena sudah mengalami beberapa serangan yang bisa ditembus, maka dengan menggabungkan SHA-1 dengan HMAC akan meminimalisir kekurangan yang sebelumnya dimiliki oleh SHA-1.

### 2.2. Spesifikasi Basis Data

Berikut adalah rincian struktur dari tabel-tabel yang digunakan dalam database aplikasi ini, yaitu :

Tabel 1. Mata Pelajaran

| Nama Field | Tipe Data | Panjang | Keterangan |
|---|---|---|---|
| id | int | 11 | Id dari tiap data |
| kode_mapel | varchar | 6 | Kode dari tiap mapel |
| nama_mapel | varchar | 25 | Nama dari tiap mapel |

Tabel 2. Soal

| Nama Field | Tipe Data | Panjang | Keterangan |
|---|---|---|---|
| id | int | 11 | Id dari tiap data |
| kode_soal | varchar | 6 | Kode dari tiap soal |
| kode_mapel | varchar | 25 | Kode dari tiap mapel |

| | | | |
|---|---|---|---|
| nama_mapel | varchar | 25 | Nama dari tiap mapel |
| soal | text | 255 | Soal yang diujikan |
| choice_A | varchar | 50 | Pilihan jawaban A |
| choice_B | varchar | 50 | Pilihan jawaban B |
| choice_C | varchar | 50 | Pilihan jawaban C |
| choice_D | varchar | 50 | Pilihan jawaban D |
| jawaban_ benar | varchar | 50 | Jawaban benar dari soal |

Tabel 3. Users

| **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
|---|---|---|---|
| id | int | 10 | Id dari tiap user |
| name | varchar | 255 | Nama dari user |
| email | varchar | 50 | Email dari user |
| password | varchar | 255 | Password dari user |
| remember _token | text | 100 | Kode yang disandikan |
| created_ at | timestamp | 20 | Waktu pembuatan user |
| updated_ at | timestamp | 20 | Waktu update terakhir user |

### 2.3. HMAC

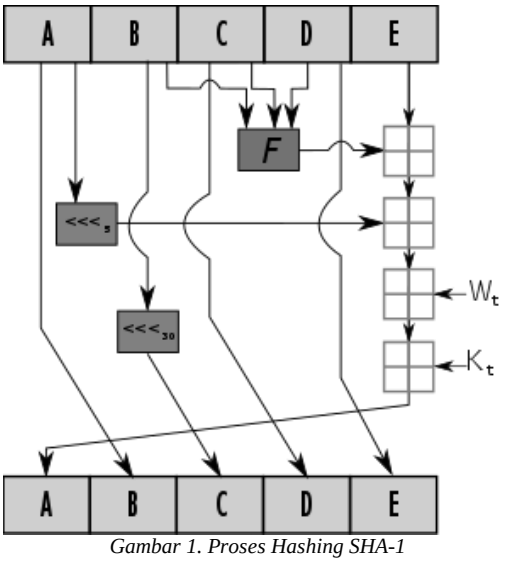
HMAC adalah singkatan dari *Hash-based Message Authentication Code*. HMAC adalah teknik MAC yang memanfaatkan fungsi hash terhadap pesan dan kemudian mengenkripsi pesan tersebut dengan sebuah kunci privat. MAC sendiri adalah teknik autentikasi pesan dengan membandingkan nilai authentication tag yang telah dihitung oleh pengirim dengan authentication tag yang dihitung sendiri oleh penerima [1].

### 2.4. SHA-1

*Secure Hash Algorithm*-1 atau yang sering dikenal dengan SHA-1 adalah salah satu fungsi *hash* yang sudah sangat dikenal. *Hash* sendiri adalah suatu teknik "klasik" dalam ilmu komputer yang banyak digunakan dalam praktek enkripsi. Hash merupakan suatu metode yang secara langsung mengakses *record-record* dalam suatu tabel dengan melakukan transformasi aritmatik pada suatu *input* dari *user* yang biasanya merupakan bentuk *string* [2].

Dengan banyaknya fungsi *hash* yang saat ini sudah tersedia, SHA-1 menjadi salah satu fungsi *hash* yang cukup banyak digunakan dalam beberapa *platform* seperti *Google* atau pun *Firefox*. Ada beberapa yang masih menggunakan SHA-1, ada pula yang sudah meninggalkan SHA-1 dan beralih kepada teknologi terbaru yang dirasa dapat melindungi lebih baik dari SHA-1.

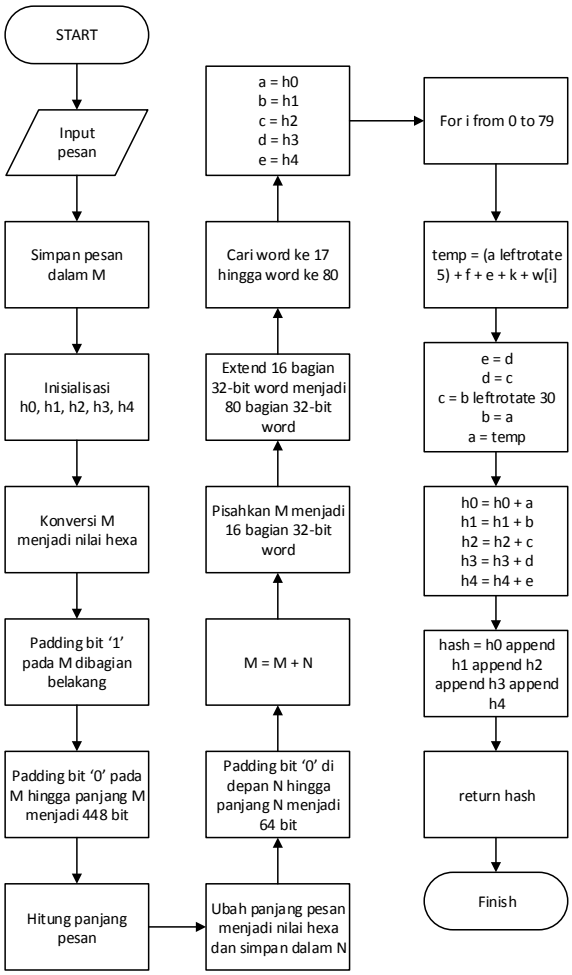
Berikut ini adalah gambaran atau skema pada proses *hashing* yang dilakukan pada Algoritma SHA-1




*Gambar 1. Proses Hashing SHA-1*

### 2.5. Flowchart Algoritma SHA-1

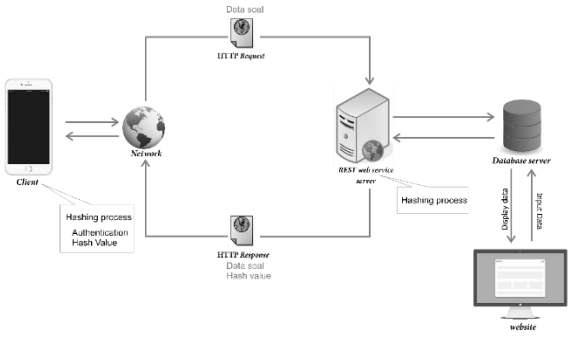
Pada bagian ini, skema dalam bentuk gambar sebelumnya dijabarkan dalam bentuk *flowchart*, yang akan lebih menjelaskan proses *hashing* yang dilakukan oleh SHA-1.




*Gambar 2. Flowchart Algoritma SHA-1*

## 2.6. Arsitektur Aplikasi

Gambar di bawah ini menjelaskan bagaimana arsitektur dari aplikasi sistem *tryout* yang dibuat. Mulai dari *input* yang dilakukan dari sisi *website* hingga *input*an yang dilakukan tampil pada *smartphone* yang digunakan.




*Gambar 3. Arsitektur Aplikasi*

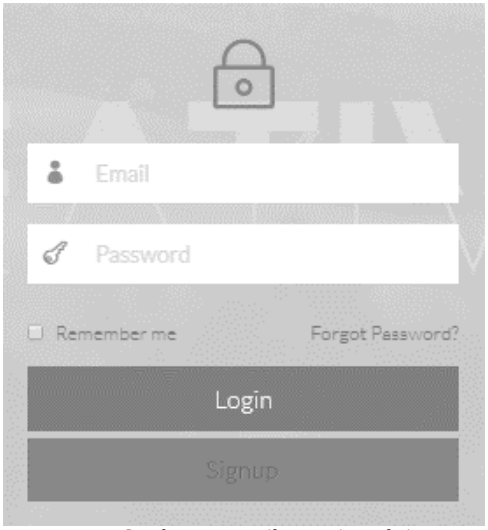
## 3. HASIL DAN PEMBAHASAN

### 3.1. Tampilan Layar Website

Pada bagian tampilan layar ini, akan dijelaskan bagaimana tampilan dari aplikasi sistem tryout dari sisi website.

#### 3.1.1. Tampilan Login

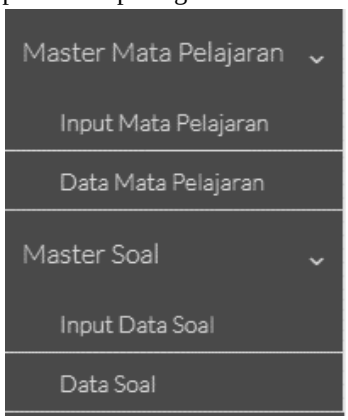
Berikut ini adalah tampilan layar yang digunakan untuk login pada sisi website. Disini diharuskan untuk mengisi email dan password untuk bisa masuk ke dalam website dan melakukan kegiatan di dalamnya.




*Gambar 3. Tampilan Login Website*

#### 3.1.2. Tampilan Menu Dashboard

Berikut ini adalah beberapa menu yang tersedia pada tampilan website. Terdapat 2 menu utama dan 2 dari masing-masing menu memiliki submenu. Detail dari menu dapat dilihat pada gambar berikut.




*Gambar 4. Tampilan Menu Dashboard Website*

#### 3.1.3. Tampilan Input Mata Pelajaran

Berikut ini adalah form input mata pelajaran yang digunakan untuk mengisi mata pelajaran yang sebelumnya belum ada pada mata pelajaran yang akan diujikan.


*Gambar 5. Tampilan Input Mata Pelajaran*

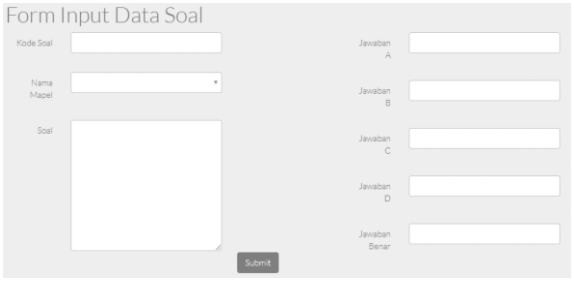
### 3.1.4. Tampilan Daftar Mata Pelajaran

Berikut ini daftar mata pelajaran yang telah berhasil diinputkan, akan muncul pada daftar di bawah ini seperti pada gambar yang terlihat.


*Gambar 6. Tampilan Daftar Mata Pelajaran*

### 3.1.5. Tampilan Input Soal

Berikut ini tampilan untuk input soal yang nantinya akan diujikan pada smartphone.




*Gambar 7. Tampilan Input Soal*

### 3.1.6. Tampilan Bank Soal

Berikut ini adalah hasil soal yang berhasil diinputkan untuk diujikan nantinya pada sistem tryout dismartphone.

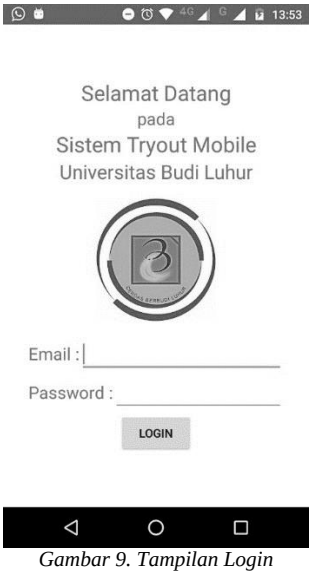

*Gambar 8. Tampilan Bank Soal*

## 3.2. Tampilan Layar Android

Berikut ini adalah tampilan layar aplikasi sistem tryout yang akan ditampilkan pada sisi mobile.

### 3.2.1. Tampilan Login

Berikut ini adalah tampilan login pada sisi mobile. Disini diharuskan mengisi email dan password.




*Gambar 9. Tampilan Login*

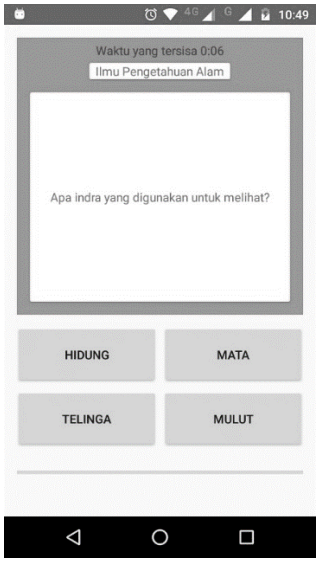
### 3.2.2. Tampilan Daftar Mata Pelajaran

Setelah berhasil login, akan menuju halaman daftar mata pelajaran yang sebelumnya telah berhasil diinput pada sisi website.


*Gambar 10. Tampilan Mata Pelajaran*

### 3.2.3. Tampilan Soal

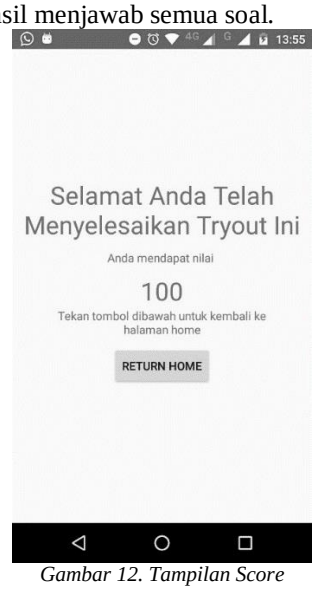
Pilih salah satu mata pelajaran yang tersedia, setelah itu akan dilakukan proses hashing. Proses hashing disini akan dilakukan karena dari sisi mobile akan mengambil data soal dari mata pelajaran yang dipilih. Jika hasil hashing yang diberikan dari sisi website dan hashing yang diproses dari sisi mobile memberikan hasil yang sama, maka dinyatakan data soal tersebut adalah data yang valid dan akan berhasil ditampilkan seperti pada gambar berikut.




*Gambar 11. Tampilan Soal*

### 3.2.4. Tampilan Score

Pada kondisi score ini akan menampilkan hasil akhir dari berapa banyak soal yang berhasil dijawab dalam bentuk score. Kondisi score muncul jika siswa berhasil menjawab soal sebelum waktu yang diberikan habis, atau waktu yang diberikan habis tanpa berhasil menjawab semua soal.




*Gambar 12. Tampilan Score*

### 3.3. Pengujian Fungsi SHA-1

Berikut ini adalah salah satu pengujian fungsi yang dilakukan dari sisi mobile. Pengujian tidak dapat dilakukan dari kedua sisi, karena untuk setiap kali dilakukan pengiriman soal, akan melakukan pengacakan. Maka dari itu, tidak akan sama hasil hash dari pengujian dari sisi website dan juga dari sisi mobile.

```
f5a4fc7e955d556f1f8445250f847672ab06d7af
Matematika.1+1=.1.2.3.4.2Matematika.9/0 =.~.+._0.
Matematika.1+1=.1.2.3.4.2Matematika.9/0 =.~.+._0.
f5a4fc7e955d556f1f8445250f847672ab06d7af
```

*Gambar 13. Pengujian Fungsi SHA-1*

Dapat dilihat, pada baris pertama adalah hasil hash atau *hash value* yang dihasilkan dari sisi website. Baris kedua adalah data yang diatur sedemikian rupa sehingga membuat pola yang dibuat sama dengan yang akan diproses pada sisi mobile. Pada baris ketiga, data yang diterima diolah menjadi sebuah pola yang mana akan sama dengan baris kedua. Pada baris keempat memberikan *hash value* yang diolah dari data pada baris ketiga. Dapat dilihat pada gambar 13 menunjukkan bahwa pada baris pertama dan baris keempat menghasilkan *hash value* yang sama. Karena jika ada perubahan data pada saat pengiriman, akan terlihat pada hasil akhir dari sisi mobile yang akan menunjukkan kegagalan penarikan data karena data yang diterima sudah berbeda dengan data yang dikirimkan.

### 3.4. Pengujian Aplikasi

Penulis telah melakukan pengujian aplikasi yang diberikan kepada 21 responden. Berikut ini adalah beberapa tanggapan yang diberikan saat responden melakukan pengujian aplikasi.

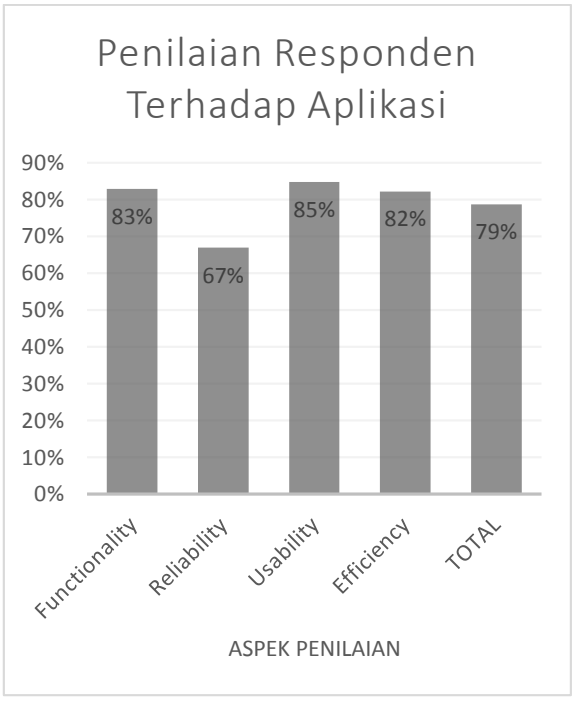
Tabel 4. Tabel Skor Responden

| Aspek Penilaian | Skor Responden | | | | |
|---|---|---|---|---|---|
| | 5 | 4 | 3 | 2 | 1 |
| | SS | S | R | TS | STS |
| Functionality | 16 | 41 | 7 | - | - |
| Reliability | 3 | 28 | 22 | 8 | 2 |
| Usability | 15 | 22 | 5 | - | - |
| Efficiency | 20 | 30 | 13 | - | - |
| **Jumlah** | 54 | 121 | 47 | 8 | 2 |

Tabel 5. Tabel Skor Aktual

| Aspek Penilaian | Skor Aktual | | | | | Total Skor Aktual | Skor Ideal | % |
|---|---|---|---|---|---|---|---|---|
| | 5 | 4 | 3 | 2 | 1 | | | |
| | SS | S | R | TS | STS | | | |
| Functionality | 80 | 164 | 21 | - | - | 265 | 320 | 83% |
| Reliability | 15 | 112 | 66 | 16 | 2 | 211 | 315 | 67% |
| Usability | 75 | 88 | 15 | - | - | 178 | 210 | 85% |
| Efficiency | 100 | 120 | 39 | - | - | 259 | 315 | 82% |
| **TOTAL** | 270 | 484 | 141 | 16 | 2 | 913 | 1.160 | 79% |

Keterangan :
SS = Sangat Setuju
S = Setuju
R = Ragu-ragu
TS = Tidak Setuju
STS = Sangat Tidak Setuju




*Gambar 14. Diagram Akumulasi Penilaian Responden*

### 3.5. Evaluasi Program

Kelebihan Program :

- Hanya orang yang diperbolehkan saja yang bisa menggunakan API Sistem *Tryout* ini dengan tujuan untuk pengamanan data yang dikirimkan oleh API ke Android.
- Aplikasi ini sudah kompatibel dengan perangkat Android pada umumnya terlebih khusus minimal spesifikasi OS Android Jelly Bean.
- Proses *hashing* tidak memakan waktu yang lama.

Kekurangan Program :

- Tidak bisa menampilkan soal dengan tulisan yang terlalu banyak dan pilihan jawaban yang juga terlalu banyak atau terlalu panjang, karena keterbatasan visual yang mampu ditampilkan dari masing-masing smartphone.
- Banyak hal yang masih harus diatur secara manual dari segi koding dan belum bisa diatur secara dinamis dari sisi website.

## 4. KESIMPULAN

Pada proses pembuatan dan pengujian aplikasi untuk Tugas Akhir ini, berdasarkan perancangan, pembuatan, serangkaian ujicoba dan analisis program dapat ditarik kesimpulan, yaitu :

- Dengan adanya aplikasi ini, pelaksaan ujian tryout dapat dilakukan dengan lebih mudah dan lebih efisien.
- Aplikasi ini dapat melakukan hashing dari sisi website dan mobile secara tepat dan cepat.

## 5. DAFTAR PUSTAKA

[1] Ramadhany, Taufik. (2006) 'Keyed-hash Message Authentication Code (HMAC)', 1.

[2] Widaseta, Muharram Huda. (2009) 'Perkembangan Enkripsi Fungsi Hash pada SHA (Secure Hash Algorithm)', *Jurnal Ilmu Komputer dan Teknologi Informasi*, 3(2), 1-7.