**BAB V**

**HASIL DAN PENGUJIAN**

**5.1 Pengujian *blackbox***

Teknik pengujian *blackbox* berfokus pada domain informasi dari perangkat lunak, dengan melakukan *test case* dengan mempartisi domain *input* dari suatu program dengan cara yang memberikan cakupan pengujian yang mendalam.

Metode ujicoba ini juga memfokuskan pada keperluan fungsional dari software, karena itu ujicoba *blackbox* memungkinkan pengembang software untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsional suatu program.

Adapun rencana pengujian yang akan di uji dengan teknik pengujian *black box* adalah sebagai berikut :

Table 5. 1 rencana pengujian

| No | Komponen Sistem yang di uji | Butir uji | Jenis Pengujian |
| --- | --- | --- | --- |
| 1 | Halaman *Login* | *Input Email* dan *Password* yang benar | *Black box* |
| *Input Email* dan *password* yang salah | *Black box* |
| 2 | Menu Utama admin | Tambah Komisi | *Black box* |
| Tambah data training Komisi | *Black box* |
| Menambah *User* Komisi | *Black box* |
| 3 | Menu Utama *User* Komisi | Melihat daftar aspirasi | *Black box* |
| Melihat detail aspirasi | *Black box* |
| Menanggapi aspirasi | *Black box* |
| 4 | Edit Profile | *Form Update* *Profile* | *Black box* |
| 5 | Ganti *password* | Mengisi *Form update* *password* yang benar | *Black box* |
| Mengisi *form update password* yang salah | *Black box* |
| 6 | Masyarakat Register | Mengisi *Form Register* Masyarakat yang benar | *Black box* |
| Mengisi *form register* yang salah | *Black box* |
| 7 | Masyarakat *input* aspirasi | Proses *Text mining* dan *cosine similarity* | *Black box* |
| 8 | Masyarakat membalas tanggapan | Mengisi *form* balas tanggapan dan mengirim ke komisi penanggap | *Black box* |

**5.2 Pengujian Fungsional**

Pada bagian ini di bahas mengenai pengujian program, proses- proses pada masing- masing modul program berdasarkan dari rencana pengujian yang di bahas sebelumnya:

Table 5. 2 Pengujian Fungsional

| No | Komponen | Hasil Yang Di Harapkan | Hasil Pengujian |
| --- | --- | --- | --- |
| 1 | Halaman Login | User Login dengan memasukkan password yang benar akan menuju ke menu berikutnya | [🗸] diterima  [ ] ditolak |
| User memasukkan password email yang salah maka akan muncul pesan kesalahan | [🗸] diterima  [ ] ditolak |
| 2 | Menu Utama Admin | Berhasil Menambah Komisi | [🗸] diterima  [ ] ditolak |
| Menambahkan data training dan berhasil | [🗸] diterima  [ ] ditolak |
| Menambahkan User pada komisi tertentu dan berhasil | [🗸] diterima  [ ] ditolak |
| 3 | Menu Utama User Komisi | Melihat daftar aspirasi sesuai dengan komisi yang aktif | [🗸] diterima  [ ] ditolak |
| Sukses mengirim tanggapan Aspirasi | [🗸] diterima  [ ] ditolak |
| 4 | Halaman *edit profile* | Sukses mengedit data pribadi user | [🗸] diterima  [ ] ditolak |
| 5 | Ganti Password | Memasukkan *form* inputan yang benar dan menuju ke menu berikutnya | [🗸] diterima  [ ] ditolak |
| Memasukkan *Form* inputan yang salah dan muncul notifikasi salah | [🗸] diterima  [ ] ditolak |
| 6 | Masyarakat Register | Masyarakat mengisi form register yang benar  Dan muncul pesan sukses | [🗸] diterima  [ ] ditolak |
| Masyarakat mengisi form register yang salah dan muncul pesan kesalahan | [🗸] diterima  [ ] ditolak |
| 7 | Masyarakat Input aspirasi | Proses *text mining* dan *Cosine similarity* berjalan dengan benar. | [🗸] diterima  [ ] ditolak |
| 8 | Masyarakat membalas tanggapan User Komisi | Balasan tanggapan berhasil dikirim oleh masyarakat | [🗸] diterima  [ ] ditolak |

**5.2 Pengujian *White Box***

Pada bab ini akan dibahas yang berkaitan dengan penerapan metode *white box* pada sistem implementasi algoritma *text minin*  dan *cosine similarity.*

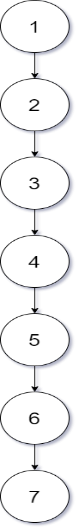
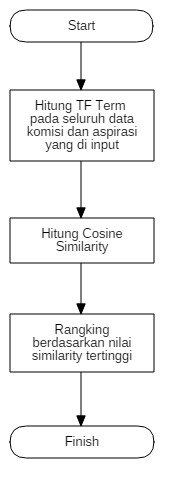
Adapun rencana pengujian yang akan dilakukan dengan metode *white box*  adalah proses penerapan algoritma *text mining* dan *cosine similarity*  yang penulis terapkan dalam penelitian ini.

**5.3.1 *Text mining***

Algoritma dalam melakukan pengimputan aspirasi dengan menggunakan *text mining,* :

1. Input aspirasi pada aplikasi *mobile*
2. *query* inputan di lakukan *case folding* untuk menghilankan karakter yang tidak penting dan mengubah semua huruf menjadi huruf kecil
3. setelah dilakukan *case folding* selanjutnya masuk ke tahap *tokenizing* untuk memotong *string query* setiap kata penyusunnya menjadi *array*
4. tahap selanjutnya adalah *filtering* yaitu untuk mengambil kata kata penting dan membuang kata kata umum yang sering di gunakan
5. tahap selanjutnya ialah tahap *stemming* untuk mencari *root* tiap pada query.

Berdasarkan algoritma diatas dibuat dalam bentuk flowchart sebagai berikut.

**

Dari gambar diatas maka dapat ditentukan *Cyclomatic Complexity* sebagai berikut :

Path = 1-2-3-4-5-6-7

V(G) = E-N + 2

= 6 – 7 + 2

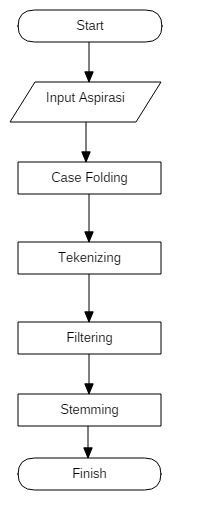
= 1

* + 1. ***Cosine Similarity***

Adapun algoritma pada tahap *cosine similarity* yaitu sebagai berikut :

1. Menghitung TF term pada seluruh data komisi dan query aspirasi yang sudah melalui tahap *text mining*
2. Kemudian dilakukan perhitungan *cosine similarity*
3. Dari hasil perhitungan *cosine similarity*  maka dilakukan peringkinan berdasarka nilai *similarity* tertinggi.

Adapun *flowhchart* dan *flowgrap* dari *cosine similarity* sebagai berikut :

Dari gambar diatas maka dapat ditentukan Cyclomatic Complexity sebagai berikut :

Path = 1-2-3-4-5

V(G) = E-N + 2

= 5 – 4 + 2

= 1

**5.4 Kesimpulan hasil pengujian aplikasi**

Setelah melakukan pengujian implementasi text mining dan cosine similarity sebagai sarana aspirasi publik berbasis mobile (Studi Kasus DPRD makasar). Kesimpulan dari hasil pengujian tersebut adalah baik, dimana semua fungsi aplikasi telah berjalan dengan baik.