**BAB IV**

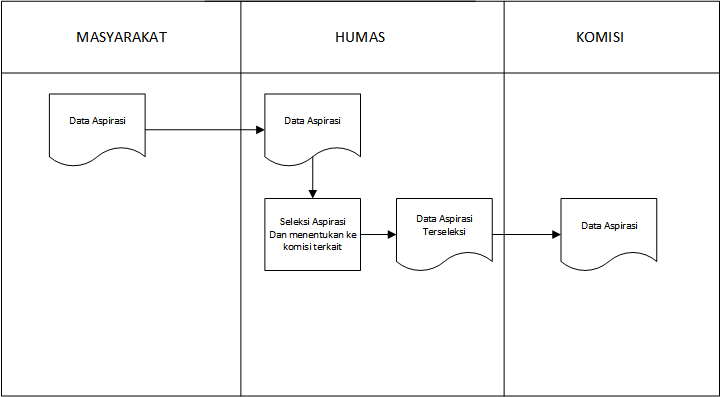
**ANALISIS DAN PERANCANGAN**

* 1. **Analisis Sistem**

Analisis sistem adalah sesuatu hal yang terpenting dalam melakukan sebuah penelitian, analisis sistem adalah menggambarkan bagaimana sistem yang sedang berjalan dan sistem yang baru yang diusulkan. Analisa sistem dapat menguaraikan permasalahan yang sedang terjadi dan bagaiaman mencari solusi atau sistem yang akan diusulkan.

* + 1. **Analisis Sistem Yang Sedang Berjalan**

Analisis sistem adalah salah satu tahapan penting yang digunakan dalam menyelesaikan permasalahan, Hasil dari analisis sistem dapat digunakan dalam pengembangan pada sistem dengan memberikan gambaran sistem yang akan dibuat. Penelitian ini membahas tentang implementasi *text mining* dan *cosine similarity* dalam menentukan ke komisi mana aspirasi dari masyarakat akan di teruskan. Setelah dilakukan analisis dan wawancara dengan bagian kepala bagian Hubungan Masyarakat (humas) di Kantor Dewan Perwakilan Rakyat Daerah (DPRD) ada beberapa permasalahan yang muncul sehingga menyebabkan penelitian ini dilakukan. Permasalahan tersebut antara lain adalah proses penentuan komisi yang tepat oleh bagian humas masih bersifat manual sehingga membutuhkan waktu yang cukup lama dan tentunya akan menambah beban kerja bagian humas



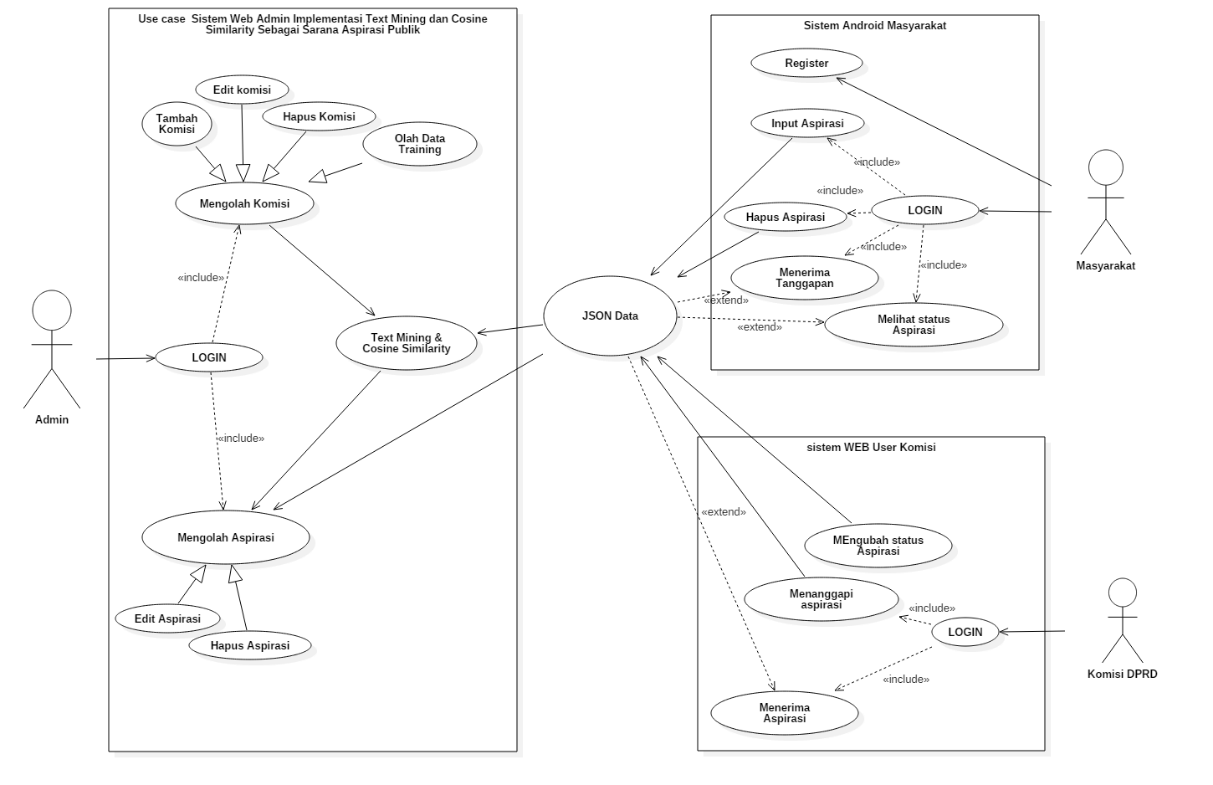
**Gambar 4.1 Sistem Yang Sedang Berjalan**

* 1. **Perancangan Sistem**

Seperti yang telah disebutkan pada bab sebelumnya bahwa alat desain sistem yang akan digunakan adalah UML (*Unfield Modelling Language*) maka pada sub bab ini akan ditampilkan mengenai *case diagram, class* *diagram, sequence diagram* dan *activity diagram* dari aplikasi yang akan dibangun.

* + 1. ***Use Case Diagram***

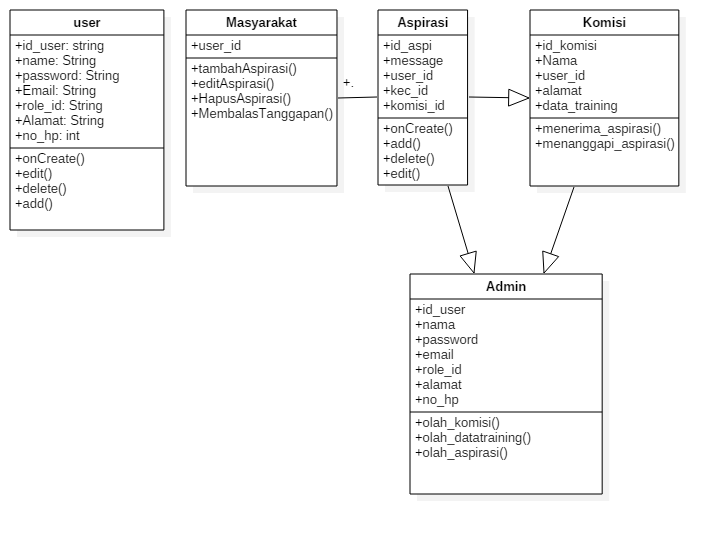
Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Use Case Untuk perangkat lunak yang akan dibangun dijelaskan pada Gambar 4.2 .



Gambar 4. 1 *Use Case* Sistem Aplikasi

* + 1. ***Class Diagram***

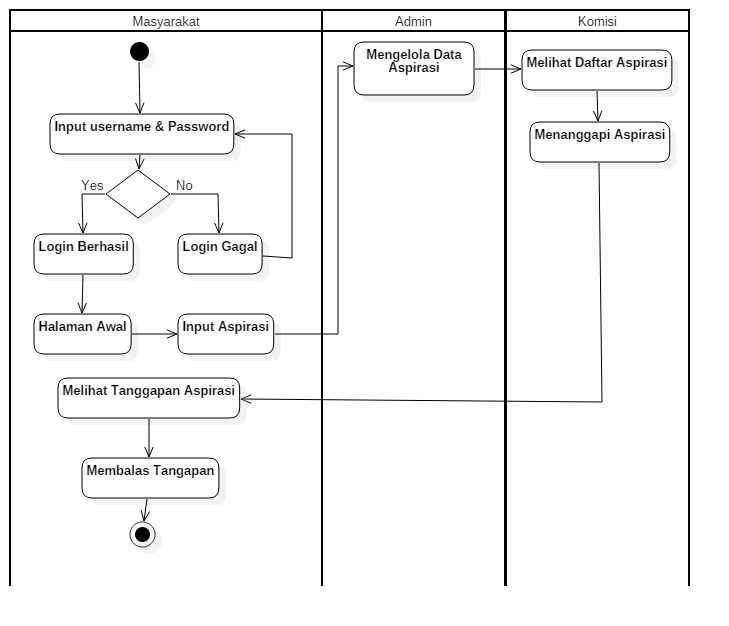
*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class* diagram menggambarkan struktur dan deskripsi *Class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. Gambar 4.3 berikut kami akan menjelaskan bagaimana bentuk *class* diagram pada aplikasi yang akan dibangun.



Gambar 4. 2 *Class Diagram*

* + 1. ***Activity Diagram***

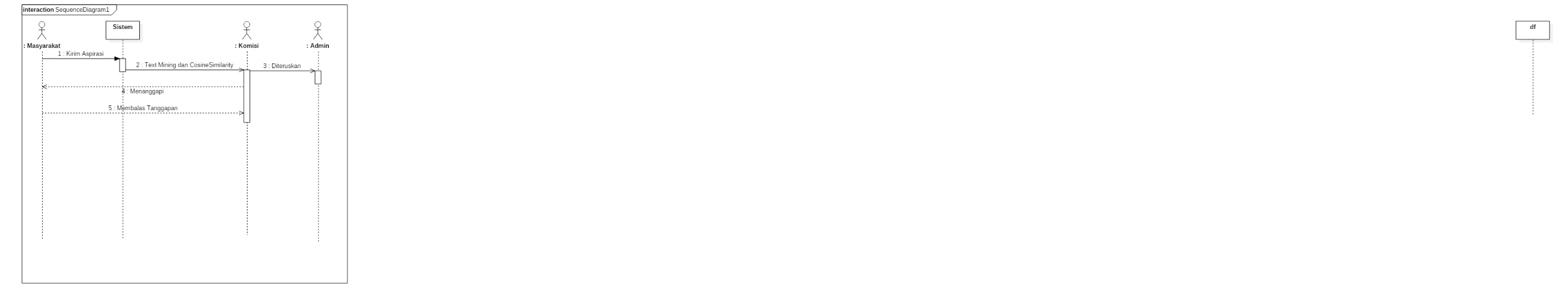
*Activity* diagram menggambarkan berbagai alir aktivitas dalam aplikasi yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.



Gambar 4. 3 *Activity Diagram*

* + 1. ***Sequence Diagram***

*Sequence* *diagram* (diagram urutan) adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, display, dan sebagainya berupa pesan/message. *Sequence Diagram* Untuk perangkat lunak yang akan dibangun dijelaskan pada gambar 4.4.



Gambar 4. 4 *Sequence Diagram*

* 1. **Rancangan Antarmuka**

Sebelum kami membangun sistem, terlebih dahulu yang harus dilakukan adalah dengan merancang secara terperinci desain antarmuka pada sistem. Berikut kami akan menjabarkan desain antarmuka sistem yang akan kami bangun.

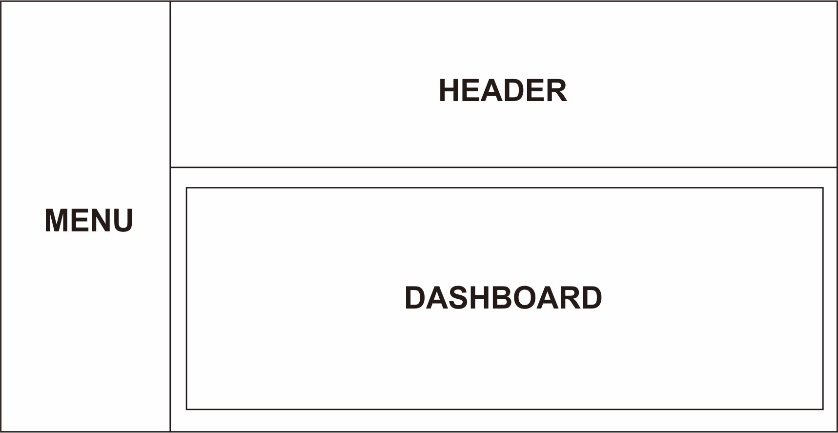
* + 1. **Perancangan *Output***

*Output* merupakan produk dari sistem informasi yang dapat dilihat. *Output* ini dapat berupa hasil yang dikeluarkan dimedia keras (kertas dan lain-lain) dan *output* yang berupa hasil dikeluarkan kemedia lunak (tampilan dilayar).

Bentuk atau format dari output dapat berupa keterangan-keterangan tabel atau grafik. Yang paling banyak dihasilkan adalah *output* yang berbentuk tabel akan tetapi sekarang dengan kemampuan teknologi komputer yang dapat menampilkan *output* dalam bentuk grafik, maka *output* berupa grafik juga mulai banyak dihasilkan.

1. Rancangan *Output* Halaman Utama Admin

Rancangan ini dibuat untuk menggambarkan bentuk menu yang tampil pada aplikasi.



Gambar 4. 5 Halaman Utama Admin

1. Rancangan *Output* Data Daftar Aspirasi

Rancangan ini dibuat untuk menggambarkan bentuk halaman yang tampil pada aplikasi.



Gambar 4. 6 Daftar Aspirasi

1. Rancangan *Output* Tambah Komisi

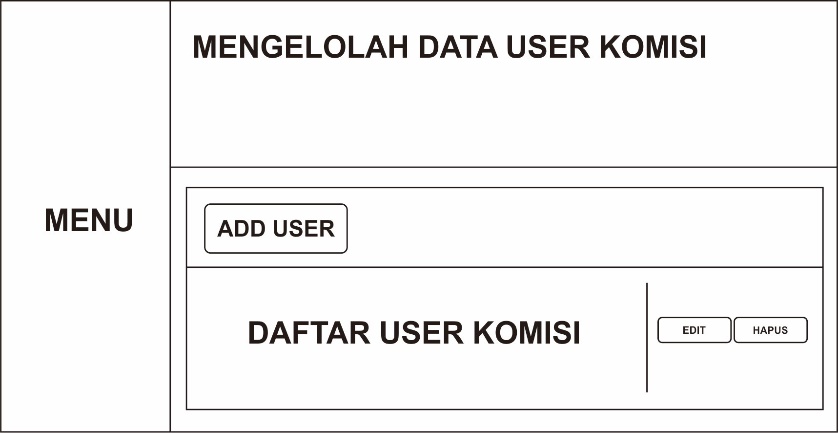
Rancangan ini dibuat untuk menggambarkan bentuk halaman yang tampil pada aplikasi.



Gambar 4. 7 Tambah Komisi

1. Rancangan *Output* Mengolah User Komisi

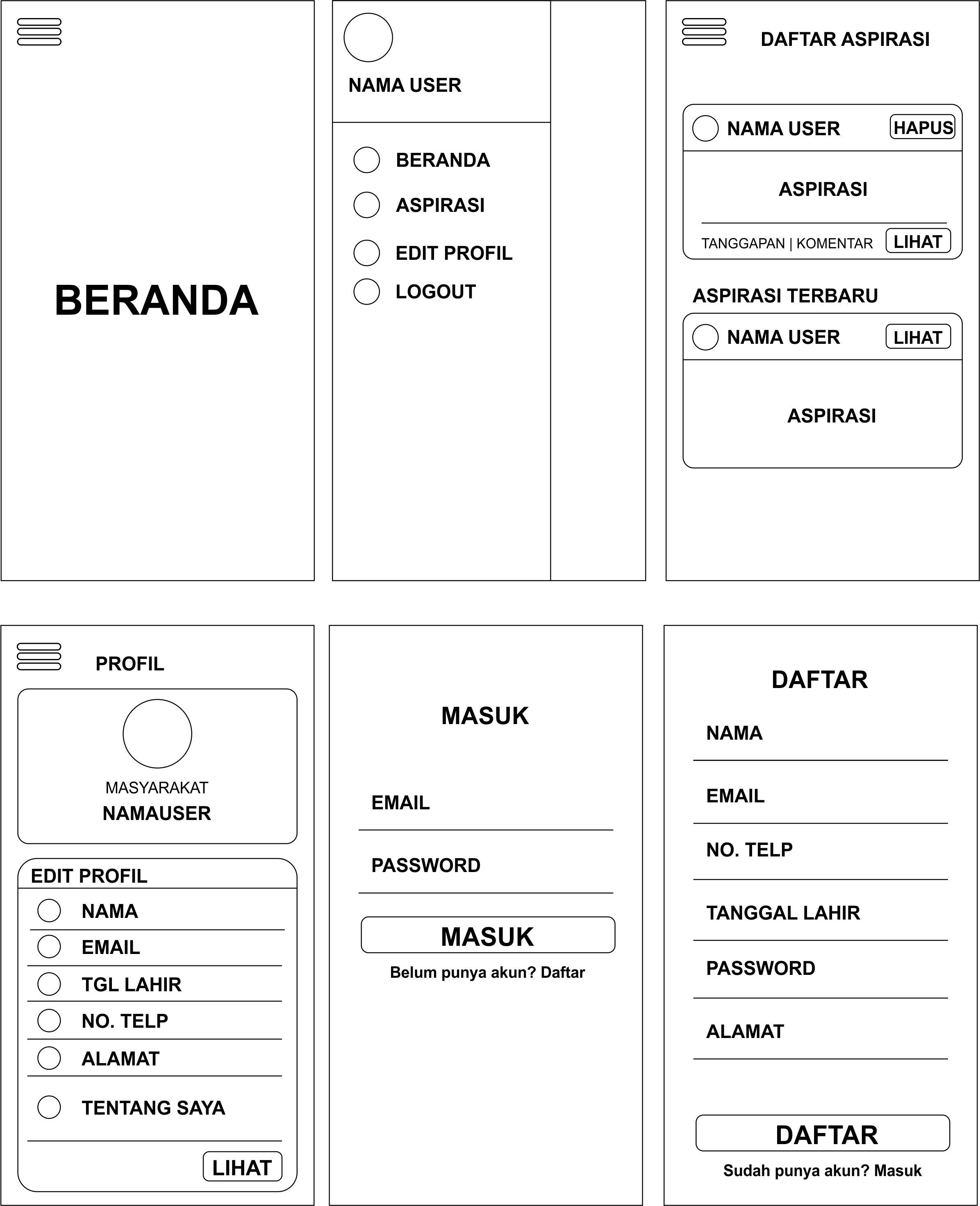
Rancangan ini dibuat untuk menggambarkan bentuk halaman yang tampil pada aplikasi.



Gambar 4. 8 Mengolah *User* Komisi

1. Rancangan *Output* Halaman Beranda *User*

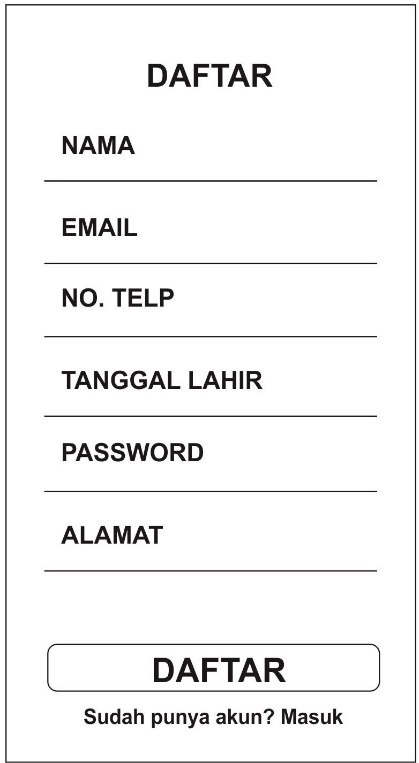
Rancangan ini dibuat untuk menggambarkan bentuk menu yang tampil pada aplikasi.



Gambar 4. 9 Menu Beranda *User*

1. Rancangan *Output* Halaman Daftar

Rancangan ini dibuat untuk menggambarkan bentuk halaman yang tampil pada aplikasi.



Gambar 4. 10 *Output* Halaman Daftar

1. Rancangan *Output* Halaman Login

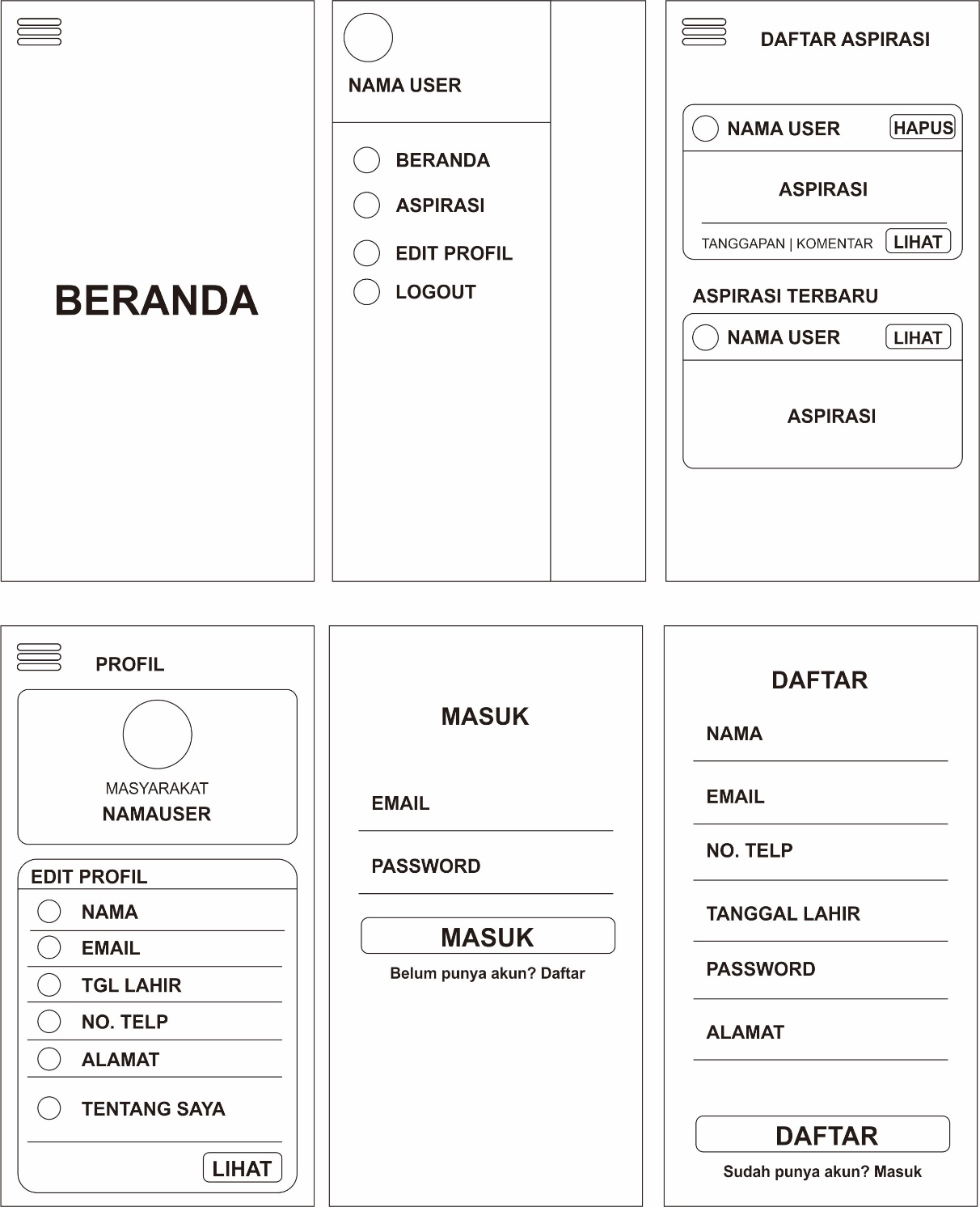
Rancangan ini dibuat untuk menggambarkan bentuk halaman yang tampil pada aplikasi.



Gambar 4. 11 *Output* Halaman Login

1. Rancangan *Output* Menu *Sidebar*

Rancangan ini dibuat untuk menggambarkan bentuk menu yang tampil pada aplikasi.



Gambar 4. 12 Menu *Sidebar*

1. Rancangan *Output* Menu Aspirasi

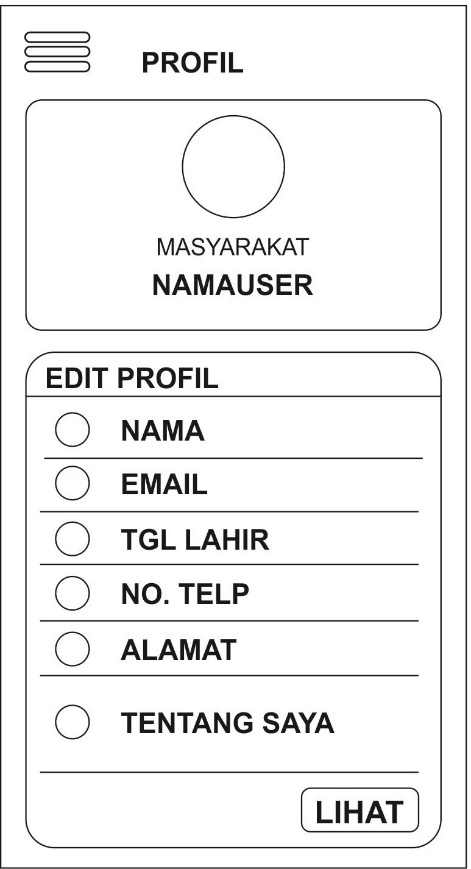
Rancangan ini dibuat untuk menggambarkan bentuk menu yang tampil pada aplikasi.



Gambar 4. 13 Menu Aspirasi

1. Rancangan *Output* Menu Profil

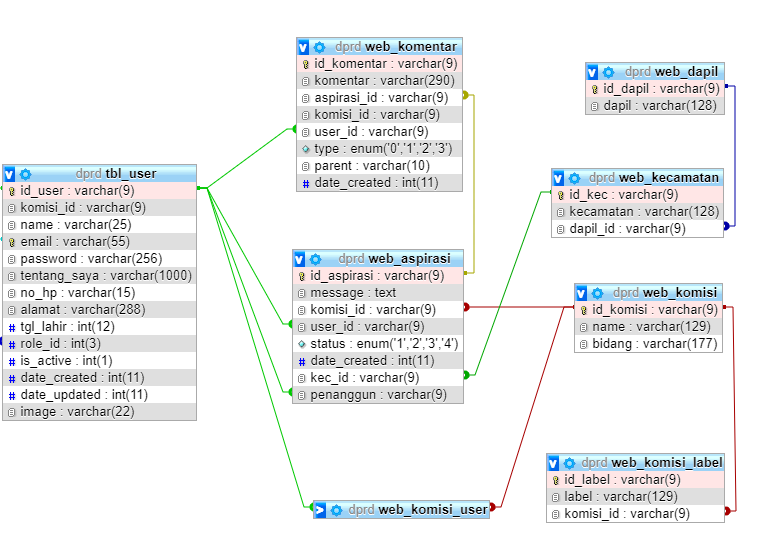
Rancangan ini dibuat untuk menggambarkan bentuk menu yang tampil pada aplikasi.



Gambar 4. 14 *Output* Menu Profil

* 1. **Struktur Basis Data**
     1. **Rancangan Relasi Tabel**

Relasi tabel adalah model konseptual yang mendeskripsikan hubungan antara tabel-tabel yang di bangun untuk keperluan sistem yang di rancang.



Gambar 4. 5 Rancangan Relasi Basis data

* + 1. **Rancangan Tabel**

Adapun tabel yang penulis gunakan dalam rancangan sistem implementasi algoritma *text mining* dan *cosine similarity* sebagai sarana aspirasi publik adalah sebagai berikut :

1. Tabel User

Tabel 4. 1 tbl\_user

|  |  |  |  |
| --- | --- | --- | --- |
| colum | Type | Null | Default |
| Id\_user | String(9) | no |  |
| name | String(25) | No |  |
| email | String(25) | No |  |
| password | String(256) | No |  |
| No\_hp | Int(12) | No |  |
| alamat | String(288) | No |  |
| Tgl\_lahir | Int(11) | no | current\_timestamp() |
| Role\_id | Int(3) | No |  |
| Is\_active | Int(1) | No |  |
| Date\_created | Int(110 | no | current\_timestamp() |
| Image | Varchar(22) | no | Default.png |

1. Table web\_aspirasi

Tabel 4. 2 web\_aspirasi

|  |  |  |  |
| --- | --- | --- | --- |
| colum | Type | Null | Default |
| Id\_aspirasi | String(9) | no |  |
| message | text | No |  |
| Komisi\_id | String(9) | No |  |
| User\_id | String(9) | No |  |
| status | Enum(0,1,2,3,4) | No | 0 |
| Date\_created | Int(110 | No | current\_timestamp() |
| Kec\_id | String (9) | no |  |
| penanggun | String(9) | No | User\_001 |

1. Tabel web\_dapil

Tabel 4. 3 web\_dapil

| colum | Type | Null | Default |
| --- | --- | --- | --- |
| Id\_dapil | String(9) | no |  |
| dapil | String(128) | No |  |

1. Tabel web\_kecamatan

Tabel 4. 4 web\_kecamatan

| colum | Type | Null | Default |
| --- | --- | --- | --- |
| Id\_kecamatan | String(9) | no |  |
| kecamatan | String(128) | No |  |
| Dapil\_id | String(9) | No |  |

1. Tabel web\_komentar

Tabel 4. 5 web\_komentar

|  |  |  |  |
| --- | --- | --- | --- |
| colum | Type | Null | Default |
| Id\_komentar | String(9) | no |  |
| komentar | String(290) | No |  |
| Aspirasi\_id | String(9) | No |  |
| User\_id | String(9) | No |  |
| type | Enum(0,1,2,3,4) | No | 0 |
| Date\_created | Int(11) | No | current\_timestamp() |
| parent | String (9) | no | 0 |

1. Tabel web\_komisi

Tabel 4. 6 web\_komisi

|  |  |  |  |
| --- | --- | --- | --- |
| colum | Type | Null | Default |
| Id\_komisi | varchar (9) | no |  |
| Name | Varchar(129) | No |  |
| Bidang | varchar (117) | No |  |

1. Tabel web\_komisi\_label

Tabel 4. 7 web\_komisi\_label

| colum | Type | Null | Default |
| --- | --- | --- | --- |
| Id\_label | varchar (9) | no |  |
| Label | Varchar(129) | No |  |
| Komisi\_id | varchar (9) | No |  |

1. Tabel web\_komisi\_user

Tabel 4. 8 web\_komisi-user

| colum | Type | Null | Default |
| --- | --- | --- | --- |
| Id\_k\_u | varchar (9) | no |  |
| Dapil\_id | Varchar(9) | No |  |
| Komisi\_id | varchar (9) | No |  |
| jabatan | Varchar(129) | No |  |
| Jumlah\_tugas | Int(2) | No | 0 |

* 1. **Proses Perhitungan**
     1. ***Text Mining***

Pada saat masyarakat mengirim aspirasi , saat itulah proses *text mining* akan bekerja dimana aspirasi tersebut akan di proses dengan tahapan sebagai berikut :

1. *Folding*

Pada tahap ini semua huruf aspirasi akan menjadi huruf kecil dan hanya akan menerima huruf “a” sampai “z” dan membuat angka atau karakter lainnya.Adapun contohnya “Terjadi perdebatan antara masyarakat dengan aparat Pemerintah dikarenakan pembangunan sekolah di atas Tanah Sengketa” akan di ubah menjadi “terjadi perdebatan antara masyarakat dengan aparat Pemerintah dikarenakan pembangunan sekolah di atas tanah sengketa”

1. *Tokenizing*

Tahap *tokenizing* adalah tahap pemotongan string input berdasarkan tiap kata yang menyusunnya. Dimana aspirasi sebelumnya akan di ubah kedalam bentuk array dan hasilnya akan menjadi :

| Array Aspirasi |
| --- |
| terjadi |
| perdebatan |
| antara |
| masyarakat |
| dengan |
| aparat |
| pemerintah |
| dikarenakan |
| pembangunan |
| sekolah |
| di |
| atas |
| tanah |
| sengketa |

1. *Filtering*

Selanjutnya adalah tahap membuang kata kata yang tidak penting atau mengambil kata kata yang penting. pada penelitian ini kami memutuskan untuk menggunakan *algoritma stoplist* yaitu membuang kata kata yang tidak penting seperti kata “dan”, “di” dan lain lain. Maka akan menjadi seperti pada tabel di bawah ini.

| **Array Aspirasi** | **Tahap Tokenizing** |
| --- | --- |
| terjadi | terjadi |
| perdebatan | perdebatan |
| antara | antara |
| masyarakat | masyarakat |
| dengan | dengan |
| aparat | aparat |
| pemerintah | pemerintah |
| dikarenakan | dikarenakan |
| pembangunan | pembangunan |
| sekolah | sekolah |
| di |  |
| atas | atas |
| tanah | tanah |
| sengketa | sengketa |

Proses selanjutnya ialah membuang kata penghubung atau akhir dengan kata lain mencari kata awal/*root* dari tiap kata hasil *filtering* pada tabel diatas.

| **Array Aspirasi** | **Tahap Tokenizing** | **Tahap Stemming** |
| --- | --- | --- |
| terjadi | terjadi | jadi |
| perdebatan | perdebatan | debat |
| antara |  |  |
| masyarakat | masyarakat | masyarakat |
| dengan |  |  |
| aparat | aparat | aparat |
| pemerintah | pemerintah | perintah |
| dikarenakan | dikarenakan | karena |
| pembangunan | pembangunan | bangun |
| sekolah | sekolah | sekolah |
| di |  |  |
| atas | atas | atas |
| tanah | tanah | tanah |
| sengketa | sengketa | sengketa |

1. *Analizyng*

Pada tahap ini merupakan tahap untuk mencari keterhubungan aspirasi diatas dengan menggunakan *Cosine Similarity*  yang akan di bahas pada **4.5.2 *Cosine Similarity*** berikut***.***

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

Tahap ini merupakan lanjutan dari tahap sebelumnya, yaitu untuk menganalisa kecocokan antar tiap kata penyusun dari aspirasi yang di input masyarakat.

Langkah pertama adalah mecari kecocokan kata penyusun dari aspirasi setelah melewati proses *text mining* dengan kata penyusun tiap-tiap table komisi dilanjutkan melakukan uji kemiripan text bisa di lihat pada tabel berikut.

| **Tahap Akhir Text Mining** | **Jumlah Kemunculan kata** | | | | |
| --- | --- | --- | --- | --- | --- |
| **Aspirasi** | **Komisi A** | **komisi B** | **komisi C** | **komisi D** |
| jadi | 1 | 0 | 0 | 0 | 0 |
| debat | 1 | 0 | 0 | 0 | 0 |
| masyarakat | 1 | 1 | 0 | 0 | 1 |
| aparat | 1 | 0 | 0 | 0 | 0 |
| perintah | 1 | 1 | 0 | 1 | 0 |
| karena | 1 | 0 | 0 | 0 | 0 |
| bangun | 1 | 0 | 0 | 0 | 0 |
| sekolah | 1 | 0 | 0 | 0 | 1 |
| atas | 0 | 0 | 0 | 0 | 0 |
| tanah | 1 | 0 | 0 | 0 | 0 |
| sengketa | 1 | 0 | 0 | 0 | 0 |

Berdasarkan rumus *cosine similarity* akan di peroleh hasil sebagai berikut

Maka diperoleh bobot sebagai berikut :

Komisi A : 0.42640143271122

Komisi B : 0

Komisi C : 0.30151134457776

Komisi D : 0.42640143271122

Dari hasil diatas terdapat 2 bobot yang paling besar adalah Komisi A dan Komisi D, berdasarkan pembahasan dari BAB II adalah akan di cari nilai tertinggi kemudian akan dijadikan patokan untuk menentukan nilai *range*  untuk penentuan multi komisi.

Nilai maksimal = 0.42640143271122

Peryaratan untuk menentukan multi komisi ialah harus sama dengan atau lebih besar dari 75% dari nilai maksimal, dari persyaratan tersebut di peroleh hasil yaitu Komisi A dan Komisi D yang memenuhi persyaratan kemudian aspirasi tersebut akan diarahkan ke komisi A dan komisi B.

Dari hasil analisis , semakin banyak data training yang di input pada tiap tiap komisi maka akan semakin baik hasil yang diharapkan.