**PENERAPAN *TEXT MINING* UNTUK PENGELOLAAN IDENTIFIKASI**

**KELUHAN PENGGUNA SECARA OTOMATIS DI**

**POLITEKNIK NEGERI INDRAMAYU**

**Tugas Akhir**

diajukan untuk memenuhi sebagian persyaratan mencapai gelar Ahli Madya jenjang Diploma III

Jurusan Teknik Informatika



Oleh :

**ELBA AYU KURNIA**

**NIM. 1703056**

**JURUSAN TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI INDRAMAYU**

**2020**

# HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh :

Nama : Elba Ayu Kurnia

NIM : 1703056

Program Studi : D3 Teknik Infotmatika

Judul : Penerapan *Text Mining* Untuk Pengelolaan Identifikasi

Keluhan Pengguna di Politeknik Negeri Indramayu

Pembimbing : 1. Iryanto, S.Si.,M.Si. ..................................

2. Alifia Puspaningrum, S.Pd., M.Kom.  ..................................

Telah berhasil dipertahankan dihadapan dewan penguji pada tangal … Agustus 2020 dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Ahli Madya Program Studi Teknik Informatika, Jurusan Teknik Informatika, Politeknik Negeri Indramayu.

DEWAN PENGUJI

Nama Jabatan Tandatangan Tanggal

1. Ketua Penguji ................ ................

2. Sekretaris Penguji ................. ................

3. Anggota ................. ................

Indramayu, ….. Agustus 2020

Ketua Jurusan Teknik Informatika

**Iryanto, S.Si.,M.Si**

NIP. 199008012019031014

# MOTTO

“W*hen you started it, then you have to finished it*. Karena itu adalah salah satu candradimuka untuk membentukmu menjadi pribadi yang bertanggung jawab pada apa yang kamu pilih untuk dirimu sendiri, terhadap orang lain juga, dan apa yang sudah dilakukan orang tua terhadap kalian.”

(Hanum Salsabiela Rais)

# PERNYATAAN KEASLIAN

Saya menyatakan dengan sesungguhnya bahwa Tugas Akhir ini adalah asli hasil karya saya sendiri, dan sepanjang pengetahuan saya tidak terdapat karya atau pendapat yang pernah ditulis atau dipublikasikan oleh orang lain, kecuali yang secara tertulis disebutkan sumbernya dalam naskah dan dalam daftar pustaka.

Indramayu, 7 Agustus 2020 Yang Menyatakan,

ELBA AYU KURNIA

NIM. 1703056

# ABSTRAK

Dalam suatu perguruan tinggi, kepuasan peserta didik (mahasiswa) adalah salah satu hal yang dibutuhkan untuk meningkatkan kualitas atau mutu perguruan tingg itu sendiri. Penanganan komplain saat ini dinilai masih kurang karena bersifat manual. Bukan hal sulit jika proses tersebut dilakukan oleh manusia, hanya saja memerlukan waktu yang tidak sedikit. Oleh karena itu, dengan menggunakan *text mining* dalam tugas akhir ini diharapkan dapat membantu dalam memilah atau mengetahui kategori dari sebuah komplain di Politeknik Negeri Indramayu. Dengan *Text mining* dapat dicari kata-kata yang dapat mewakili isi dari komplain, lalu dianalisis apakah komplain tersebut masuk dalam kategori suatu unit kerja atau tidak, untuk selanjutnya di *broadcast* agar keluhan dapat segera ditinjau. Agar hasil pengukuran tingkat *similaritas* dokumen dengan *keyword* mendapatkan hasil yang optimal maka digunakanlah algoritma *CountVectorizer* yang dapat mengubah fitur teks menjadi sebuah representasi vector*.* Dari data teks yang kita miliki (atau biasa disebut korpus) kita dapat menyusun sebuah *vocabulary*, kita kemudian menjadikan setiap data menjadi representasi *vector* masing - masing untuk setiap kata. Tiap elemen dari *vector* menunjukkan jumlah fitur kata yang ada pada data. Sedangkan untuk pengklasifikasian kategori digunakanlah metode Support Vector Machine (SVM) digunakan untuk mencari hyperplane terbaik dengan memaksimalkan jarak antar kelas. Hyperplane adalah sebuah fungsi yang dapat digunakan untuk pemisah antar kelas. Dalam 2-D fungsi yang digunakan untuk klasifikasi antar kelas disebut sebagai line whereas, fungsi yang digunakan untuk klasifikasi antas kelas dalam 3-D disebut plane similarly, sedangan fungsi yang digunakan untuk klasifikasi di dalam ruang kelas dimensi yang lebih tinggi di sebut hyperplane. Dalam pemodelan klasifikasi, SVM memiliki konsep yang lebih matang dan lebih jelas secara matematis dibandingkan dengan teknik-teknik klasifikasi lainnya. SVM juga dapat mengatasi masalah klasifikasi dan regresi dengan linear maupun non linear. Adapun hasil akurasi model dari sistem identifikasi keluhan secara otomatis menggunakan *text mining* ini sebesar 90%.

Kata kunci : *text mining*, komplain, support vector machine*, count vectorizer.*

## ABSTRACT

In a university, students satisfaction is one thing that is needed to improve the quality of the company or agency itself. Handling complaints is currently considered lacking because it is manual. It's not difficult if the process is done by humans, it's just that it takes a lot of time. Therefore, using text mining in this thesis is expected to help in sorting or knowing the category of a complaint at State Polytechnic of Indramayu. Text mining can search for words that can represent the contents of the complaint, then analyze whether the complaint is included in the category of a work unit or not, for further broadcast so that complaints can be immediately reviewed. So that the results of the measurement of the level of similarity of documents with keywords get optimal results, the CountVectorizer algorithm is used that can convert text features into a vector representation. From the text data that we have (or commonly called a corpus) we can compile a vocabulary, we then make each data into a vector representation each for each word. Each element of the vector indicates the number of word features present in the data. As for the classification of categories, the Support Vector Machine (SVM) method is used to find the best hyperplane by maximizing the distance between classes. Hyperplane is a function that can be used for separating between classes. In 2-D functions used for class classifications are referred to as lines whereas, functions used for class classifications in 3-D are called plane similarly, whereas functions used for classifications in classrooms of higher dimensions are called hyperplanes. In classification modeling, SVM has a more mature and more mathematically clear concept compared to other classification techniques. SVM can also overcome the problem of classification and regression with linear or non linear. The accuracy of the model complaint identification using text mining is 90%.

Keywords: text mining, complaints, support vector machine, count vectorizer

# KATA PENGANTAR

*Alhamdulillahi Rabbil ’Alamin*, puji syukur saya panjatkan kepada Allah *Subhanahu wa Ta’ala* atas rahmat dan karunia-Nya sehingga saya dapat menyelesaikan Tugas Akhir ini dengan tepat waktu.

Tugas akhir ini merupakan salah satu persyaratan memperoleh gelar Ahli Madya pada jenjang D3 Jurusan Teknik Informatika Politeknik Negeri Indramayu. Tugas Akhir ini berjudul ‘Penerapan *Text Mining* Untuk Pengelolaan Identifikasi Keluhan Pengguna Secara Otomatis di Politeknik Negeri Indramayu’.

Tugas Akhir yang dibuat oleh penulis bertujuan untuk mempermudah unit kerja Pusat Pengembangan Pembelajaran dan Penjaminan Mutu Pendidikan (P4MP) Politeknik Negeri Indramayu dalam mengelola komplain, guna meningkatkan mutu atau kualitas Politeknik Negeri Indramayu.

Dalam penyusunan laporan Tugas Akhir ini penulis telah banyak mendapatkan dukungan dari berbagai pihak, oleh sebab itu penulis ingin mengungkapkan rasa terima kasih kepada :

1. Kedua orang tua penulis yang selalu mendoakan dan memberikan semangat kepada penulis agar dapat menyelesaikan Tugas Akhir dengan lancar dan tepat waktu .
2. Bapak Casiman Sukardi, S.T., M.T selaku Direktur Politeknik Negeri Indramayu.
3. Bapak Iryanto,S.Si.,M.Si.,M.Sc selaku ketua jurusan Teknik Informatika Politeknik Negeri Indramayu sekaligus selaku Dosen Pembimbing I .
4. Ibu Alifia Puspaningrum, S.Pd., M.Kom. selaku Dosen Pembimbing II.

Dan semua pihak yang telah membatu penyelesaian Tugas Akhir ini. Semoga Allah SWT senantiasa memudahkan setiap langkah-langkah kita menuju kebaikan dan selalu menganugerahkan kasih sayang-Nya untuk kita semua. Amin.

Demikian apa yang dapat penulis sampaikan. Semoga laporan Tugas Akhir ini dapat bermanfaat bagi kita semua.

Indramayu, 29 Juli 2020

Penulis

# DAFTAR ISI

[HALAMAN PENGESAHAN ii](#_Toc47608831)

[MOTTO iii](#_Toc47608832)

[PERNYATAAN KEASLIAN iv](#_Toc47608833)

[ABSTRAK v](#_Toc47608834)

[ABSTRACT vi](#_Toc47608835)

[KATA PENGANTAR vii](#_Toc47608836)

[DAFTAR ISI viii](#_Toc47608837)

[DAFTAR TABEL xi](#_Toc47608838)

[DAFTAR GAMBAR xii](#_Toc47608839)

[DAFTAR LAMPIRAN xiv](#_Toc47608840)

[BAB I PENDAHULUAN 1](#_Toc47608841)

[1.1 Latar Belakang Masalah 1](#_Toc47608842)

[1.2 Rumusan Masalah 2](#_Toc47608843)

[1.3 Batasan Masalah 2](#_Toc47608844)

[1.4 Tujuan Penelitian 4](#_Toc47608845)

[1.5 Manfaat Penelitian 4](#_Toc47608846)

[1.6 Sistematika Penulisan Laporan 4](#_Toc47608847)

[BAB II LANDASAN TEORI 6](#_Toc47608848)

[2.1 Kecerdasan Buatan 6](#_Toc47608849)

[2.2 *Text Mining* 6](#_Toc47608850)

[2.3 Algoritma *CountVectorizer* 7](#_Toc47608851)

[2.4 Algoritma *Support Vector Machine* 7](#_Toc47608852)

[2.5 *Unified Modeling Languange (UML)* 7](#_Toc47608853)

[2.5 Teknik Pengujian Perangkat Lunak 13](#_Toc47608854)

[BAB III METODOLOGI PENELITIAN 15](#_Toc47608855)

[3.1 Metode Penelitian 15](#_Toc47608856)

[3.2 Pengumpulan Data 16](#_Toc47608857)

[3.2.1 Angket (Kuesioner) 16](#_Toc47608858)

[3.3 Analisis Kebutuhan Sistem 17](#_Toc47608864)

[3.4.3 Kebutuhan *Hardware* 17](#_Toc47608868)

[3.4.3 Kebutuhan *Software* 17](#_Toc47608869)

[3.4 Perancangan Sistem 18](#_Toc47608872)

[3.4.1 Use Case Diagram 18](#_Toc47608873)

[3.4.2 Activity Diagram 21](#_Toc47608880)

[3.4.3 *Class* Diagram 26](#_Toc47608881)

[3.5 Flowchart 26](#_Toc47608887)

[3.5.1 Flowchart User 26](#_Toc47608893)

[3.5.2 Flowchart Admin 28](#_Toc47608898)

[3.5.3 *Flowchart* Superadmin 29](#_Toc47608899)

[3.6 Perancangan ERD (Entity Relationship Diagram) 29](#_Toc47608905)

[3.7 Perancangan Database 31](#_Toc47608912)

[3.7.1 Tabel tb*\_admin* 32](#_Toc47608914)

[3.7.2 Tabel tb\_*complaint* 32](#_Toc47608915)

[3.7.3 Tabel tb\_*raw\_complaint* 33](#_Toc47608924)

[3.7.4 Tabel tb\_status 33](#_Toc47608925)

[3.7.5 Tabel tb\_kategori 34](#_Toc47608926)

[3.7.6 Tabel tb\_token 34](#_Toc47608927)

[3.7.7 Tabel tb\_images 34](#_Toc47608928)

[3.8 Perancangan Desain Antarmuka 35](#_Toc47608935)

[3.8.1 Antarmuka *User* 35](#_Toc47608936)

[3.8.2 Antarmuka Admin 37](#_Toc47608937)

[3.8.3 Antarmuka Superadmin 41](#_Toc47608938)

[BAB IV HASIL DAN PEMBAHASAN 45](#_Toc47608939)

[4.1 Hasil 45](#_Toc47608940)

[4.2 Pembahasan 45](#_Toc47608941)

[4.2.1 Struktur Direktori 45](#_Toc47608942)

[4.2.2 Implementasi *Database* 48](#_Toc47608943)

[4.2.3 Implementasi Desain Antarmuka *(Interface)* 51](#_Toc47608944)

[4.2.4 Implementasi Pembuatan Model 64](#_Toc47608966)

[4.2.5 Pengujian Sistem 68](#_Toc47608967)

[BAB V PENUTUP 96](#_Toc47608968)

[5.1 Kesimpulan 96](#_Toc47608969)

[5.2 Saran 96](#_Toc47608970)

[DAFTAR PUSTAKA 97](#_Toc47608971)

# DAFTAR TABEL

[**Tabel 2.1** Simbol Usecase Diagram 8](#_Toc46995221)

[**Tabel 2.2** Simbol Usecase Diagram (Lanjutan) 9](#_Toc46995222)

[**Tabel 2.3** Simbol Usecase Diagram (Lanjutan) 10](#_Toc46995223)

[**Tabel 2.4** Simbol Class Diagram 11](#_Toc46995224)

[**Tabel 2.5** Simbol Class Diagram (Lanjutan) 12](#_Toc46995225)

[**Tabel 2.6** Simbol Activity Diagram 13](#_Toc46995226)

[**Tabel 3.1** Tabel Kebutuhan Hardware 17](#_Toc46995208)

[**Tabel 3.2** Tabel Kebutuhan Software 17](#_Toc46995209)

[**Tabel 3.3** Penjelasan Use Case Diagram 19](#_Toc46995210)

[**Tabel 3.4** Penjelasan Use Case Diagram (Lanjutan) 20](#_Toc46995211)

[**Tabel 3.5** Penjelasan Use Case Diagram (Lanjutan User) 21](#_Toc46995212)

[**Tabel 3.6** Penjelasan ERD 29](#_Toc46995213)

[**Tabel 3.7** Tabel tb\_admin. 31](#_Toc46995214)

[**Tabel 3.8** Tabel tb\_complaint 31](#_Toc46995215)

[**Tabel 3.9** Tabel tb\_raw\_complaint 32](#_Toc46995216)

[**Tabel 3.10** Tabel tb\_status 32](#_Toc46995217)

[**Tabel 3.11** Tabel tb\_ kategori 33](#_Toc46995218)

[**Tabel 3.12** Tabel tb\_ token 33](#_Toc46995219)

[**Tabel 3.13** Tabel tb\_ images 33](#_Toc46995220)

[**Tabel 4.1** Proses CountVectorizer 65](#_Toc46995196)

[**Tabel 4.2** Black-Box Testing User 69](#_Toc46995197)

[**Tabel 4.3** Black-Box Testing Superadmin 69](#_Toc46995198)

[**Tabel 4.4** Black-Box Testing Admin 69](#_Toc46995199)

# DAFTAR GAMBAR

[**Gambar 2.1** Proses Transformasi Kalimat Menjadi Vector CountVectorizer 7](#_Toc46897396)

[**Gambar 3.1** Metodologi Penelitian 15](#_Toc47608985)

[**Gambar 3.2** Use Case Diagram 18](#_Toc47608986)

[**Gambar 3.3** Activity Diagram Admin 22](#_Toc47608987)

[**Gambar 3.4** Activity Diagram Superadmin 23](#_Toc47608988)

[**Gambar 3.5** Activity Diagram Superadmin (Lanjutan) 24](#_Toc47608989)

[**Gambar 3.6** Activity Diagram Admin 25](#_Toc47608990)

[**Gambar 3.7** Class Diagram 26](#_Toc47608991)

[**Gambar 3.8** Flowchart User 27](#_Toc47608992)

[**Gambar 3.9** Flowchart Admin 28](#_Toc47608993)

[**Gambar 3.10** Flowchart Superadmin 29](#_Toc47608994)

[**Gambar 3.11** Perancangan ERD 31](#_Toc47608995)

[**Gambar 3.12** Antarmuka Beranda (User) 35](#_Toc47608996)

[**Gambar 3.13** Antarmuka Form Keluhan (User) 36](#_Toc47608997)

[**Gambar 3.14** Antarmuka Statistik (User) 37](#_Toc47608998)

[**Gambar 3.15** Antarmuka Rekap Tanggapan (User) 37](#_Toc47608999)

[**Gambar 3.16** Antarmuka Splashscreen (Admin) 38](#_Toc47609000)

[**Gambar 3.17** Antarmuka Login (Admin) 39](#_Toc47609001)

[**Gambar 3.18** Antarmuka Home (Admin) 39](#_Toc47609002)

[**Gambar 3.19** Antarmuka List Keluhan (Admin) 40](#_Toc47609003)

[**Gambar 3.20** Antarmuka Beri Tanggapan / Laporkan (Admin) 41](#_Toc47609004)

[**Gambar 3.21** Antarmuka Splashscreen (Superadmin) 42](#_Toc47609005)

[**Gambar 3.22** Antarmuka Login (Superadmin) 42](#_Toc47609006)

[**Gambar 3.23** Antarmuka Home (Superadmin) 43](#_Toc47609007)

[**Gambar 3.24** Antarmuka List Keluhan (Superadmin) 43](#_Toc47609008)

[**Gambar 3.25** Antarmuka Buatkan Akun (Superadmin) 44](#_Toc47609009)

[**Gambar 3.26** Antarmuka Buatkan Akun (Superadmin) 44](#_Toc47609010)

[**Gambar 4.1** Struktur Direktori Folder Web E-Complaint 44](#_Toc46995035)

[**Gambar 4.2** Struktur Direktori Folder src 45](#_Toc46995036)

[**Gambar 4.3** Struktur Direktori Folder app 45](#_Toc46995037)

[**Gambar 4.4** Struktur Direktori Folder API Ecomplaint 46](#_Toc46995038)

[**Gambar 4.5** Database e-complaint 47](#_Toc46995039)

[**Gambar 4.6** Tabel tb\_admin 48](#_Toc46995040)

[**Gambar 4.7** Tabel tb\_complaint 48](#_Toc46995041)

[**Gambar 4.8** Tabel tb\_images 49](#_Toc46995042)

[**Gambar 4.9** Tabel tb\_kategori 49](#_Toc46995043)

[**Gambar 4.10** Tabel tb\_raw\_complaint 49](#_Toc46995044)

[**Gambar 4.11** Tabel tb\_status 50](#_Toc46995045)

[**Gambar 4.12** Tabel tb\_token 50](#_Toc46995046)

[**Gambar 4.13** Halaman Beranda 51](#_Toc46995047)

[**Gambar 4.14** Halaman Form Keluhan 52](#_Toc46995048)

[**Gambar 4.15** Halaman Statistik 52](#_Toc46995049)

[**Gambar 4.16** Halaman Rekap Tanggapan 53](#_Toc46995050)

[**Gambar 4.17** Halaman Diagram Alir Keluhan 54](#_Toc46995051)

[**Gambar 4.18** Splashscreen 55](#_Toc46995052)

[**Gambar 4.19** Halaman Login 55](#_Toc46995053)

[**Gambar 4.20** Halaman Home 56](#_Toc46995054)

[**Gambar 4.21** Halaman List Keluhan 57](#_Toc46995055)

[**Gambar 4.22** Halaman Detail Keluhan 58](#_Toc46995056)

[**Gambar 4.23** Splashscreen Superadmin 59](#_Toc46995057)

[**Gambar 4.24** Halaman Login Superadmin 59](#_Toc46995058)

[**Gambar 4.25** Tab Kelola Keluhan dan Tab Kelola Admin 60](#_Toc46995059)

[**Gambar 4.26** Halaman List Keluhan 61](#_Toc46995060)

[**Gambar 4.27** Halaman Detail Keluhan. 62](#_Toc46995061)

[**Gambar 4.28** Halaman List Laporan Keluhan 63](#_Toc46995062)

[**Gambar 4.29** Teks Preprocessing 64](#_Toc46995063)

[**Gambar 4.30** Rumus TF/IDF 65](#_Toc46995064)

[**Gambar 4.31** Rumus Multinomial Naive Bayes 66](#_Toc46995065)

[**Gambar 4.32** Hyperlane Yang Memisahkan 2 Kelas 67](#_Toc46995066)

[**Gambar 4.33** Feature Extraction dan Classifier 67](#_Toc46995067)

[**Gambar 4.34** Akurasi Model (TF/IDF & Multinominal Naïve Bayes) 68](#_Toc46995068)

[**Gambar 4.35** Akurasi Model (CountVectorizer & SVM) 68](#_Toc46995069)

# DAFTAR LAMPIRAN

# BAB I PENDAHULUAN

# Latar Belakang Masalah

Setiap perguruan tinggi tentunya akan menyiapkan fasilitas dan layanan yang terbaik bagi setiap mahasiswanya. Namun tidak bisa dipungkiri bahwa pelayanan di suatu perguruan tinggi tidak akan pernah luput dari ketidaksempurnaan. Untuk dapat memperbaiki dan mengembangkan layanan perguruan tinggi, setiap keluhan dan keresahan peserta didik (mahasiswa) harus dapat ditampung sebagai bahan evaluasi dan kemudian ditindaklanjuti.

Perkembangan dan perbaikan layanan di perguruan tinggi sangatlah penting untuk dipantau dan dipelihara. Bila pihak perguruan tinggi telah menyediakan fasilitas dan pelayanan bagi peserta didik namun pelayanan dan fasilitas tersebut tidak dipantau dan dipelihara, maka cepat atau lambat pelayanan dan fasilitas yang telah disediakan akan rusak dan tidak dapat digunakan lagi. Untuk dapat memelihara dan memantau layanan yang ada di perguruan tinggi, tim pengelola layanan perguruan tinggi harus mengetahui sektor-sektor yang perlu diperbaiki dan dikelola (Galih, 2019).

Politeknik Negeri Indramayu adalah salah satu instansi pendidikan (perguruan tinggi) yang masih belum memiliki *platform* khusus untuk menampung keluhan atau pengaduan dari mahasiswanya. Saat ini, sarana penyampaian keluhan dan pengaduan terhadap ketidakpuasan dan permasalahan yang ada di Politeknik Negeri Indramayu (Polindra) masih sangat terbatas yaitu dengan kuesioner saja, dan itupun hanya ditujukan untuk tenaga pengajar saja. Bahkan, tidak sedikit mahasiswa Polindra yang masih bingung harus ke mana untuk mengadukan keluhannya. Hal ini menyebabkan permasalahan yang ada hanya akan menjadi buah bibir di lingkungan kampus dan tak kunjung diproses. Sebagai contoh, fasilitas di salah satu ruang kuliah rusak dan tidak bisa digunakan. Kebanyakan mahasiswa, masih bingung harus melaporkan permasalahan ini ke mana.

Permasalahan ini cukup serius karena apabila pengelola layanan tidak mengetahui kecacatan pada suatu sektor layanan di kampus Polindra, butuh waktu yang lama atau bahkan tidak akan diperbaiki oleh tim pengelola layanan Polindra. Imbasnya, akreditasi akan menurun.

Dari permasalahan di atas, proses penyamampaian keluhan dan pengaduan terhadap ketidakpuasan yang ada di Politeknik Negeri Indramayu (Polindra) dapat dilakukan oleh komputer yang dapat berperan besar dalam mempermudah pekerjaan tim pengelola layanan universitas untuk dapat mengetahui semua keluhan dan aduan setiap mahasiswa. Sebuah aplikasi ­*E-Complaint* yang menggunakan *text mining* dapat dibangun sebagai sarana bagi mahasiswa Polindra agar dapat dengan mudah mengadukan semua permasalahan yang ada di lingkungan kampus Polindra. Dengan menggunakan *text mining* dapat dicari kata-kata yang dapat mewakili isi dari keluhan, lalu dianalisis apakah keluhan tersebut masuk kategori suatu unit kerja tertentu atau tidak. Selanjutnya keluhan akan di *broadcast* kepada unit kerja terkait, sehingga keluhan yang disampaikan tepat sasaran.

Sistem ini ini juga akan sangat berguna bagi para pimpinan di lingkungan Polindra seperti Rektor, Dekan dan Kajur. Sistem ini akan menampilkan informasi semua keluhan yang telah ditampung. Dari informasi ini, pemimpin memiliki dasar untuk menerapkan kebijakan-kebijakan menyangkut fasilitas, layanan dan sarana prasana di lingkungan Politeknik Negeri Indramayu.

# Rumusan Masalah

Berdasarkan latar belakang yang dijabarkan, maka diperoleh suatu rumusan permasalahan yang menjadi dasar pembuatan sistem tersebut, yakni sebagai berikut :

1. Bagaimana membuat sebuah aplikasi komplain yang dapat mengelompokkan komplain secara otomatis berdasarkan kategori atau *class* nya?
2. Algoritma *text mining* apa yang cocok & memiliki akurasi tinggi dalam mengidentifikasi kategori (unit kerja) keluhan yang masuk, agar keluhan didistibusikan dengan tepat sasaran ?

# Batasan Masalah

Untuk memfokuskan pembahasan, dapat diperoleh beberapa batasan masalah,

diantaranya:

1. Data latih di dapat dari responden yang merupakan mahasiswa di Politeknik Negeri Indramayu.
2. Sistem tidak dapat mengidentifikasi data dukung (gambar) yang di inputkan oleh user apabila ternyata hasil identifikasi keluhannya memiliki 2 kategori.
3. *Delimiters* digunakan pada *system* ini untuk memisahkan teks keluhan karena ada kemungkinan dalam 1 inputan keluhan ditujukan pada lebih dari satu kategori unit kerja. Adapun *delimiters* yang digunakan hanya 3 yaitu :
4. Titik (.)
5. Koma (,)
6. Dan
7. *Class* label yang dibuat untuk kategori keluhan hanya empat, yaitu Sarana Prasarana, Tenaga Pengajar, Akademik, dan Keuangan.
8. Pengguna yang memberikan keluhan untuk saat ini hanya ditujukan pada mahasiswa Politeknik Negeri Indramayu.
9. Apabila user menginputkan keluhan yang tidak sesuai dengan keempat kategori keluhan (unit kerja) yang telah dibuat, maka kaluhan tersebut masuk kedalam kategori secara acak (*random*) sesuai dengan prediksi dari model yang telah dibuat. Namun permasalahan ini dapat di *handle* dengan adanya fitur *Report /* Laporkan.

# Tujuan Penelitian

Tujuan dari pembuatan sistem yang dibuat meliputi :

1. Merancang dan membangun sistem pengaduan mahasiswa, untuk mempermudah pencatatan komplain dan mengidentifikasi kategori komplain.
2. Untuk dapat memperbaiki dan mengembangkan layanan di Politeknik Negeri Indramayu, sebagai bahan evaluasi dan untuk kemudian ditindaklanjuti.
3. Sebagai upaya deteksi dini terhadap kelemahan sistem ataupun penyimpangan pelaksanaan mutu di Politeknik Negeri Indramayu.

# Manfaat Penelitian

* + 1. Manfaat bagi penulis :

Menerapkan ilmu pengetahuan yang didapat dari perkuliahan, melatih kreatifitas, menambah wawasan dan pengalaman untuk membuat suatu sistem.

* + 1. Manfaat bagi pengguna :

Mendapatkan keluhan sesuai *role* atau unit kerja yang dipegang, dan keluan yang sudah dianalisis tadi dapat dimonitor atau ditinjau dengan mudah melalui system yang dibuat.

* + 1. Manfaat bagi pembaca :

Sebagai referensi pembaca untuk membuat suatu apikasi yang berkaitan dengan *text mining.*

# Sistematika Penulisan Laporan

Untuk memahami lebih jelas laporan ini, maka materi-materi yang tertera pada laporan tugas akhir ini dikelompokkan menjadi beberapa bab dengan sistematika penyampaian sebagai berikut :

**BAB I PENDAHULUAN**

Pada bab ini berisi pemaparan latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan laporan.

**BAB II LANDASAN TEORI**

Bab ini berisi teori dasar yang menguraikan tentang kecerdasan buatan, *text mining*, metode *CountVectorizer* dan *Support Vector Machine* (SVM), *UML* dan teknik pengujian perangkat lunak.

**BAB III METODOLOGI PENELITIAN**

Bagian ini berisi penjelasan tentang metode penelitian, metode pengumpulan data, analisa data, analisa kebutuhan sistem, *flowchart*, perancangan sistem, perancangan *ERD ,* perancangan *database,* perancangan desain antarmuka dan perancangan *text mining*.

**BAB IV HASIL DAN PEMBAHASAN**

Pada bab ini berisi hasil dan pembahasan tentang implementasi *database*, implementasi desain antarmuka dan pengujian sistem.

**BAB V PENUTUP**

Bab ini berisi kesimpulan dan saran penulis.

# BAB II LANDASAN TEORI

# Kecerdasan Buatan

Pada dasarnya AI adalah suatu pengetahuan yang membuat komputer dapat meniru kecerdasan manusia sehingga komputer dapat melakukan hal-hal yang dikerjakan manusia dimana membutuhkan suatu kecerdasan; misalkan melakukan analisa penalaran untuk mengambil suatu kesimpulan atau keputusan atau penerjemahan dari satu bahasa ke bahasa lain. Kecerdasan buatan merupakan suatu cabang dalam bidang sains komputer Sains yang mengkaji tentang bagaimana untuk melengkapi sebuah komputer dengan kemampuan atau kepintaran seperti manusia. Sebagai contoh, bagaimana komputer bisa belajar sendiri dari pengalaman dan data-data yang telah dikumpulkannya, bagaimana komputer mampu berkomunikasi dan mengucapkan kata demi kata. Dengan kemampuan ini, diharapkan komputer mampu mengambil keputusan sendiri untuk berbagai kasus yang ditemuinya.

(Wicaksono T dkk, 2011)

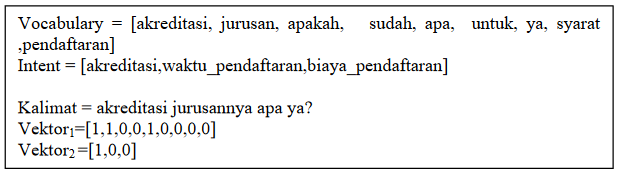
# *Text Mining*

Text mining adalah salah satu bidang khusus dalam data mining yang memiliki definisi menambang data berupa teks dimana sumber data biasanya didapatkan dari dokumen dan tujuannya adalah mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen.

Text mining dapat menganalisa dokumen, mengelompokkan dokumen berdasarkan kata-kata yang terkandung di dalamnya, serta menentukan kesamaan di antara dokumen untuk mengetahui bagaimana mereka berhubungan dengan variabel lainnya. Penerapan yang paling umum dilakukan text mining saat ini misalnya penyaringan spam, analisa sentimen, mengukur preferensi pelanggan, meringkas dokumen, pengelompokan topik penelitian, dan banyak lainnya (Rona, 2018).

# Algoritma *CountVectorizer*

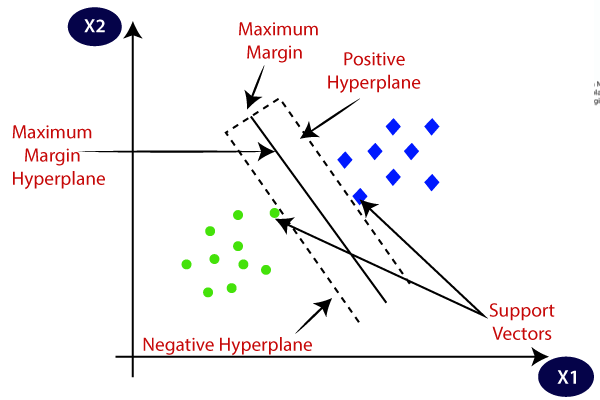
CountVectorizer adalah metode untuk mengubah susuan kata dalam sebuah kalimat menjadi vektor dengan menghitung frekuensi kemunculan kata dalam sebuah kalimat. *CountVectorizer* dapat mengubah fitur teks menjadi sebuah representasi vector*.* Dari data teks yang kita miliki (atau biasa disebut korpus) kita dapat menyusun sebuah *vocabulary*, kita kemudian menjadikan setiap data menjadi representasi *vector* masing - masing untuk setiap kata. Tiap elemen dari *vector* menunjukkan jumlah fitur kata yang ada pada data (Silitonga 2013). Adapun algoritma *CountVectorizer* dapat dilihat pada Gambar 2.1.



**Gambar 2.1** Proses Transformasi Kalimat Menjadi Vector dengan CountVectorizer

# Algoritma *Support Vector Machine*

*Support Vector Machine (SVM)* merupakan sistem pembelajaran terbimbing yang pengklasifikasiannya menggunakan ruang hipotesis berupa fungsi-fungsi linear dalam sebua ruang fitur berdimensi tinggi (Nomleni, 2015). Adapun algoritma *SVM* dapat dilihat pada Gambar 2.2.



**Gambar 2.2** Algoritma Support Vector Machine

# *Unified Modeling Languange (UML)*

*UML (Unified Modeling Language)* adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek”. Pemodelan *(modeling)* sesungguhnya digunakan untuk penyederhanaan permasalahan- permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipahami. (Nugroho,2009)

Berikut penjelasan dari beberapa diagram yang digunakan pada penelitian ini.

* + 1. **Diagram *Use-Case (Usecase Diagram)***

*Usecase diagram* menggambarkan fungsionalitas dari sebuah sistem. Sebuah *usecase* merepresentasikan sebuah interaksi antara *actor* dengan sistem. A*ctor* adalah sebuah entitas manusia yang berinteraksi dengan sistem untuk melakukan pekerjaan tertentu. Simbol *Usecase Diagram* dapat dilihat pada Tabel 2.1

**Tabel 2.1** Simbol Usecase Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| 1 |  | *Actor* | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case* |
| 2 |  | *Depedency* | Hubungan dimana perubahan yang terjadi pada satu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri  (*independent*) |

Sumber : (Nugroho, 2009)

**Tabel 2.2** Simbol Usecase Diagram (Lanjutan)

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| 3 |  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk  (*ancestor)* |
| 4 |  | *Include* | Menspesifikasikan bahwa  *use case* sumber secraa eksplisit |
| 5 |  | *Extend* | Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik  yang diberikan |
| 6 |  | *Association* | Apa yang menghubungjkan antara objek satu dnegan  objek lainnya |
| 7 | System | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas |
| 8 |  | *Use Case* | Deskripsi dari urutan aksi- aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terstruktur  bagi suatu aktor |

Sumber : ((Nugroho, 2009)

**Tabel 2.3** Simbol Usecase Diagram (Lanjutan)

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| 9 |  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-  elemennya (sinergi) |
| 10 | Note | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu  sumber daya komputasi |

Sumber : (Nugroho, 2009)

* + 1. **Diagram Kelas (*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. *Class* memiliki tiga area pokok, yaitu nama (dan *stereotype*), atribut, dan metoda. Atribut dan metoda dapat memiliki salah satu sifat berikut (Nugroho, 2009):

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak- anak yang mewarisinya
3. *Public*, dapat dipanggil oleh siapa saja Simbol *Class Diagram* dapat dilihat pada Tabel 2.4.

**Tabel 2.4** Simbol Class Diagram

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Gambar** | | | **Nama** | **Keterangan** |
| 1 |  | | | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk  (*ancestor)* |
| 2 |  | | | *Nary Association* | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek. |
| 3 |  | Class Name |  | *Class* | Himpunan dari objek-  objek yang berbagi atribut serta operasi yang sama. |
| Attributes |
| Operations |
|  | | |
| 4 |  | | | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-  elemennya (sinergi) |

Sumber : (Nugroho, 2009)

**Tabel 2.5** Simbol Class Diagram (Lanjutan)

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| 5 |  | *Extend* | Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada  suatu titik yang diberikan |
| 6 |  | Association | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri |
| 7 |  | *Association* | Apa yang  menghubungjkan antara objek satu dnegan objek  lainnya |

Sumber : (Nugroho, 2009)

* + 1. **Diagram Aktivitas (*Activity Diagram*)**

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem 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. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu. Simbol *Activity Diagram* dapat dilihat pada Tabel 2.6.

**Tabel 2.6** Simbol Activity Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| 1 |  | *Activity* | Memperlihatkan bagaimana masing masing kelas antarmuka saling berinteraksi satu  sama lain |
| 2 |  | *Action* | State dari sistem yang mencerminkan eksekusi dari suatu aksi |
| 3 |  | *Initial Node* | Bagaimana objek dibentuk atau diawali. |
| 4 |  | *Activity Final Node* | Bagaimana objek  dibentuk dan  dihancurkan |
| 5 |  | *Fork Node* | Satu aliran yang pada  tahap tertentu berubah menjadi beberapa aliran |

Sumber : (Nugroho, 2009)

# Teknik Pengujian Perangkat Lunak

Pengujian pada *text mining* untuk mengidentifikasi kategori keluhan ini menggunakan teknik pengujian *Black-box.* Pengujian *Black-box* merupakan pengujian untuk mengetahui apakah semua fungsi perangkat lunak telah berjalan semestinya sesuai dengan kebutuhan fungsional yang telah didefinisikan. Kasus ini bertujuan untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya. Teknik pengujian ini yaitu melakukan kasus uji dengan mempartisi domain *input* dan *output* program. Metode *Black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan

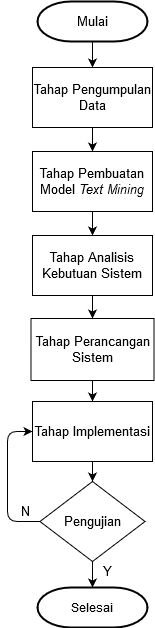
fungsional untuk suatu program. Pengujian ini berusaha menemukan kesalahan dalam kategori fungsi-fungsi yang tidak benar atau hilang, kesalahan *interface*, kesalahan kinerja, dan inisialisasi dan kesalahan terminal (Pressman, 2010).

# BAB III METODOLOGI PENELITIAN

Pada bab metodologi penelitian ini berisikan langkah-langkah yang digunakan dalam pembuatan *text mining* untuk mengidentifikasi keluhan pengguna sebagai tugas akhir. Dengan adanya metodologi ini proses pembuatan aplikasi dapat dipahami oleh pembaca.

## Metode Pelaksanaan

Berdasarkan rumusan masalah dan tujuan penelitian yang ada pada bab I, pembuatan sistem *text mining* untuk mengidentifikasi kategori keluhan ini menggunakan metode pelaksanaan seperti yang terdapat pada Gambar 3.1 berikut.



**Gambar 3.1** Metodologi Pelaksanaan

Metode penelitian pada *text mining* untuk identifikasi keluhanini dimulai dari pengumpulan data, kemudian membuat model *text mining* yang siap pakai, selanjutnya dilakukan analisis kebutuhan sistem, berikutnya perancangan lalu implementasi, dan tahap terakhir adalah pengujian.

## Pengumpulan Data

Salah satu komponen yang penting dalam penelitian adalah proses peneliti dalam pengumpulan data. Kesalahan yang dilakukan dalam proses pengumpulan data akan membuat proses analisis menjadi sulit. Selain itu hasil dan kesimpulan yang akan didapat pun akan menjadi rancu apabila pengumpulan data dilakukan tidak dengan benar.

Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam rangka mencapai tujuan penelitian. Pengumpulan data *text mining* untuk identifikasi keluhanini dibuat dalam bentuk angket (kuisioner).

### **Angket (Kuesioner)**

Kuesioner merupakan metode pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan tertulis kepada responden untuk dijawab. Kuesioner merupakan metode pengumpulan data yang lebih efisien bila peneliti telah mengetahui dengan pasti variabel yag akan diukur dan tahu apa yang diharapkan dari responden. Selain itu kuesioner juga cocok digunakan bila jumlah responden cukup besar dan tersebar di wilayah yang luas.

Berdasarkan bentuk pertanyaannya, kuesioner dapat dikategorikan dalam dua jenis, yakni kuesioner terbuka dan kuesioner tertutup. Kuesioner terbuka adalah kuesioner yang memberikan kebebasan kepada objek penelitian untuk menjawab. Sementara itu, kuesioner tertutup adalah kuesioner yang telah menyediakan pilihan jawaban untuk dipilih oleh objek penelitian. Seiring dengan perkembangan, beberapa penelitian saat ini juga menerapkan metode kuesioner yang memiliki bentuk semi terbuka. Dalam bentuk ini, pilihan jawaban telah diberikan oleh peneliti, namun objek penelitian tetap diberi kesempatan untuk menjawab sesuai dengan kemauan mereka.



## Analisis Kebutuhan Sistem

Dalam pembuatan *text mining* identifikasi keluhan ini terdapat kebutuhan *hardware* dan kebutuhan *software.* Berikut adalah kebutuhan-kebutuhan yang diperlukan :



### **Kebutuhan *Hardware***

Kebutuhan hardware yang diperlukan pada pembuatan sistem *text mining* untuk mengidentifikasi kategori keluhan ini terdapat pada Tabel 3.1.

**Tabel 3.1** Tabel Kebutuhan Hardware

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Jenis *Hardware*** | **Kebutuhan *Hardware*** | **Keterangan** |
| 1 | *Processor* | AMD A8 | Untuk mendukung berjalannya aplikasi  pembuat sistem |
| 2 | *RAM* | 4 GB atau lebih | Minimal |

### **Kebutuhan *Software***

Adapun kebutuhan *software* dalam pembuatan aplikasi sistem identifikasi keluhan secara otomatis menggunakan *text mining* ini terdapat pada Tabel 3.2.

**Tabel 3.2** Tabel Kebutuhan Software

|  |  |  |
| --- | --- | --- |
| **No** | **Jenis Software** | **Kebutuhan Software** |
| 1 | Bahasa *Scripting* | HTML 5, CSS, JavaScript, Python |
| 2 | *Software* Pengolah | VsCode, JupyterLab, Anaconda |
| 3 | Penyimpan Data | Database (MySQL) |
| 4 | Web Browser | Chrome, Firefox, dan lainnya |

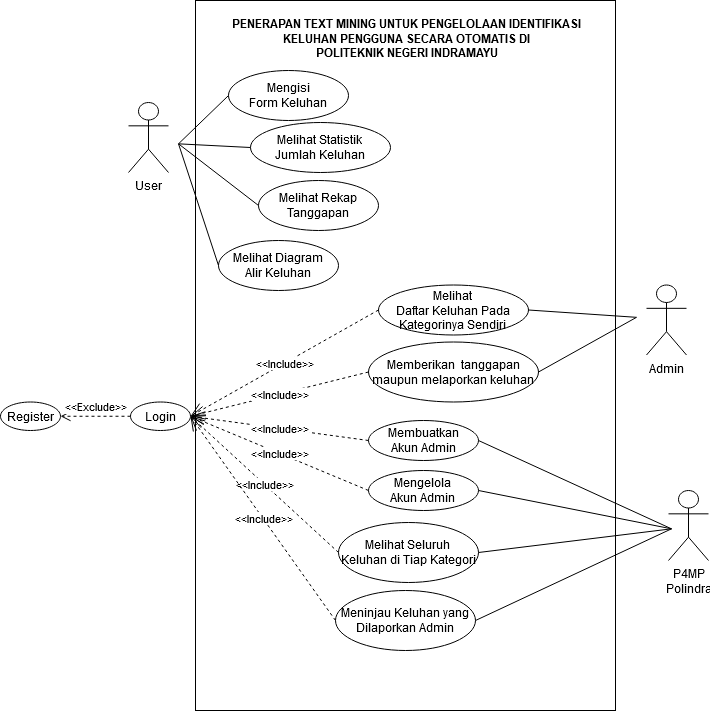


## Perancangan Sistem

Untuk mendapatkan gambaran mengenai sistem yang dibuat, maka dimodelkan dengan menggunakan *functional modelling*. Proses dan data dari sistem dimodelkan dengan *use case diagram, activity diagram,* dan *class diagram.*

### **Use Case Diagram**

Pada Gambar 3.2 merupakan *use case* diagram dari sistem *text mining* untuk mengidentifikasi kategori keluhan secara otomatis*.*



*.*

**Gambar 3.2** Use Case Diagram

Adapun penjelasan dari gambar 3.2 *Use Case Diagram* adalah pada Tabel 3.3 berikut ini :

**Tabel 3.3** Penjelasan Use Case Diagram

|  |  |  |
| --- | --- | --- |
| ***Actor*** | **Nama *Use Case*** | **Keterangan** |
| Admin | *Register* | Sebelum dapat *login* ke system, admin harus registrasi akun terlebih dahulu. Dimana akun tersebut akan dibuatkan oleh *superadmin.* |
| *Login* | Admin dapat masuk ke sistem menggunakan akun yang telah dibuatkan oleh *superadmin* atau P4MP Polindra. |
| Melihat Daftar Keluhan Pada Kategorinya Sendiri | Admin dapat melihat seluru daftar keluhan yang masuk dan teridentifikasi sebagai kategori unit kerja nya. |
| Memberikan Tanggapan Maupun Melaporkan Keluhan | Admin dapat memberikan tanggapannya terhadap keluhan yang masuk. Dan juga dapat melaporkan keluhan ke P4MP apabila isi keluhan tidak sesuai (*spam*). |

**Tabel 3.4** Penjelasan Use Case Diagram (Lanjutan)

|  |  |  |
| --- | --- | --- |
| ***Actor*** | **Nama *Use Case*** | **Keterangan** |
| P4MP  Polindra | *Login* | *Superadmin* (P4MP) dapat masuk ke sistem dengan akun yang sudah terdaftar di *database.* |
| Membuatkan Akun Admin | *Superadmin* (P4MP) dapat membuatkan akun admin, supaya admin bisa *login*  ke sistem. |
| Mengelola Akun  Admin | *Superadmin* (P4MP) dapat mengelola seluruh akun admin, antara lain mengedit atau menghapus akun admin. |
| Meninjau Keluhan yang di Laporkan Admin | Keluhan yang di laporkan admin dapat ditinjau (*accept/decline)* oleh *superadmin* (P4MP). |
| Melihat Seluruh Keluhan  Di Tiap Kategori | *Superadmin* (P4MP) dapat melihat seluruh keluhan yang masuk dari tiap kategori. |
| User | Mengisi Form Keluhan | *User* dapat memberikan keluhanya melalui form keluhan yang ada di *website E-complaint* Polindra. |
| Melihat Statistik Jumlah Keluhan | *User* juga dapat melihat persentase jumlah keluhan yang masuk di setiap kategori. |

**Tabel 3.5** Penjelasan Use Case Diagram (Lanjutan User)

|  |  |  |
| --- | --- | --- |
| ***Actor*** | **Nama *Use Case*** | **Keterangan** |
| *User* | Melihat Rekap  Tanggapan | *User* dapat melihat status tanggapannya di menu Rekap Tanggapan untuk mengetahui apakah tanggapannya sudah di tinjau atau belum. |
| Melihat Diagram  Alir Keluhan | *User* dapat melihat diagram alir *E-complaint* Polindra agar dapat mengetaui bagaimana keluhannya di proses. |

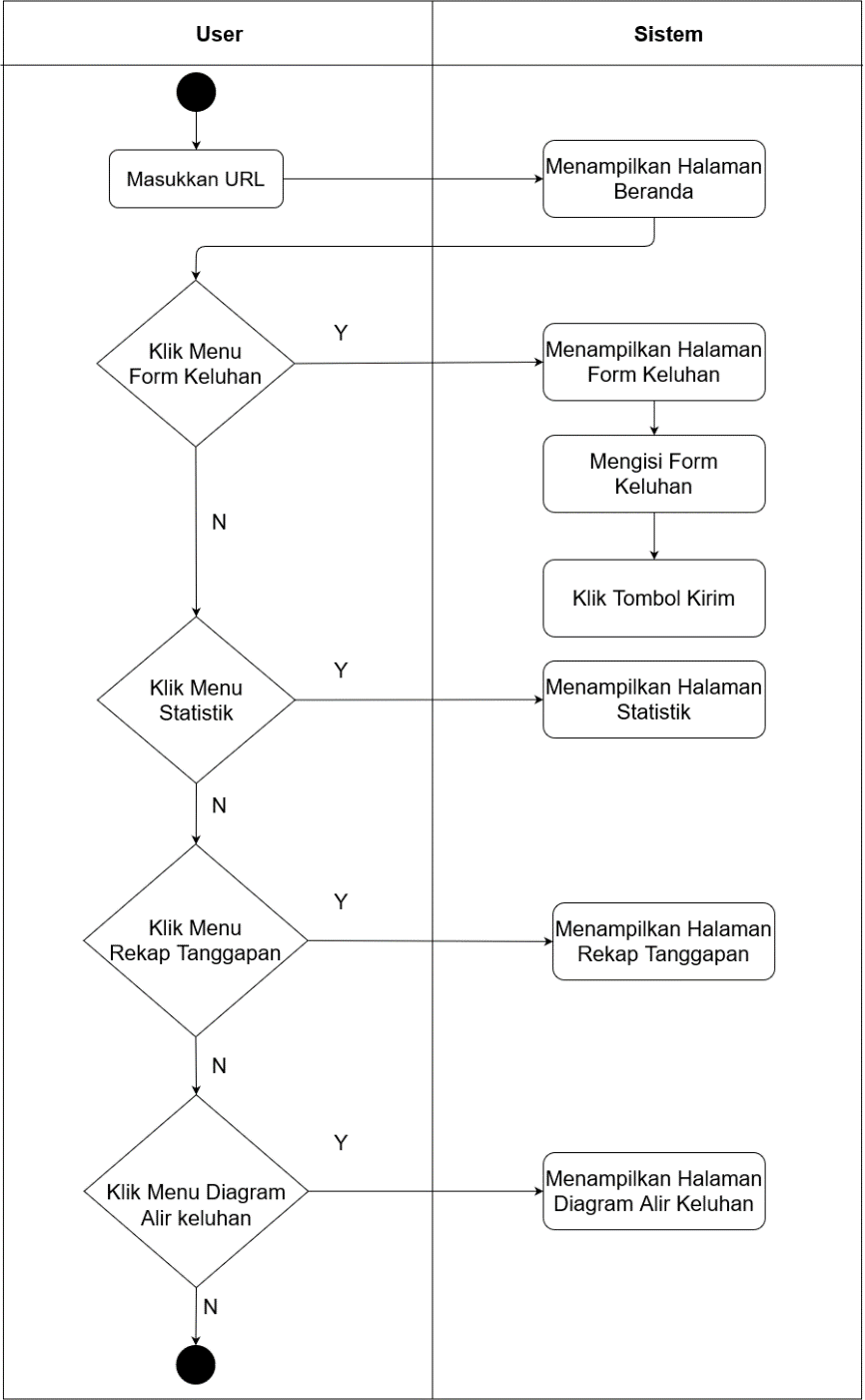


### **Activity Diagram**

*Activity Diagram* merupakan *state* diagram khusus yang menggambarkan berbagai aliran aktivitas di dalam sistem yang sedang dirancang, bagaimana masing- masing aliran berawal dan berakhir. Bahkan mungkin terjadi *decision* didalamnya. *Activity diagram* juga dapat menggambarkan proses parallel yang mungkin terjadi pada beberapa eksekusi dalam sistem.

* + - 1. ***Activity* Diagram *User***

Pada *activity* diagram *user* ini menggambarkan aliran aktivitas yang dilakukan oleh *user* dalam sistem yang dirancang. Adapun diagram aktivitas *user* terdapat pada Gambar 3.3.

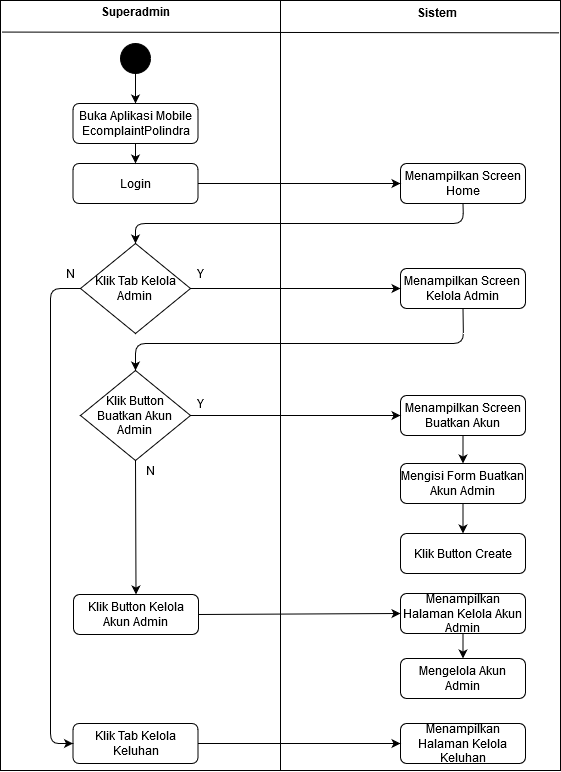


**Gambar 3.3** Activity Diagram Admin

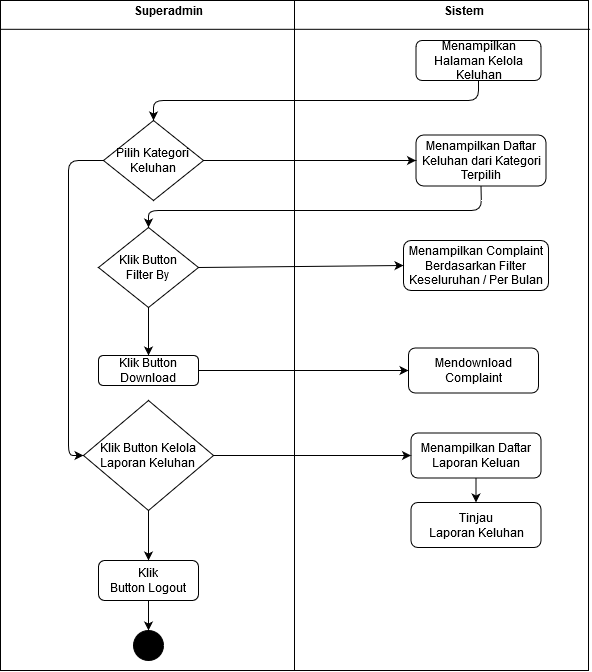
Pada *activity* diagram *user* ini dimulai dengan memasukkan *URL,* lalu *pengguna* akan diarakan pada halaman beranda. User dapat memilih menu *Form Keluhan* untuk menuangkan keluhannya, atau juga dapat memilih menu Statistik untuk dapat mengetahui presentase jumlah keluhan yang masuk di tiap-tiap kategori. Selain itu *user* juga dapat memilih menu Rekap Tanggapan untuk mengetahui tinjauan admin terhadap keluhannya. Dan terakhir user dapat memilih menu Diagram Alir Keluhan untuk dapat mengetahui alur diprosesnya sebuah keluhan yang masuk kedalam system.

* + - 1. ***Activity* Diagram Superadmin**

Pada *activity* diagram superadmin ini menggambarkan aliran aktivitas yang dilakukan oleh superadmin dalam sistem yang dirancang. Adapun *activity* diagram superadmin terdapat pada Gambar 3.4.



**Gambar 3.4** Activity Diagram Superadmin



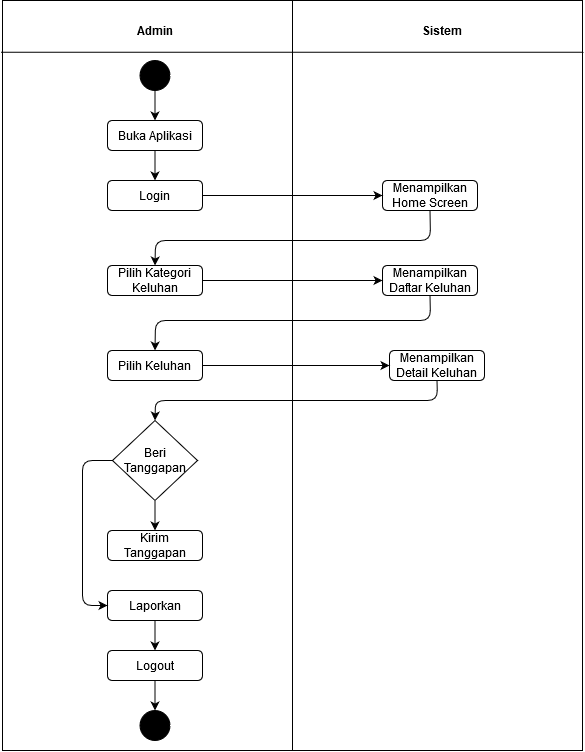
**Gambar 3.5** Activity Diagram Superadmin (Lanjutan)

Pada *activity* diagram superadmin ini dimulai dengan superadmin membuka aplikasi mobile *E-Complaint* Polindra terlebih dahulu. Lalu superadmin diharuskan login terlebih dahulu untuk masuk ke *system*, kemudian system akan mengarahkan ke halaman *Home.* Pada halaman *Home* ini superadmin dapat memilih *tab* Kelola Admin, pada *tab* ini superadmin dapat memilih menu Buatkan Akun Admin, untuk membuatkan akun admin atau menu Kelola Akun Admin untuk mengelola akun admin seperti *edit / delete* akun admin.

Superadmin juga dapat memilih *tab* Kelola Keluhan yang di dalamnya terdapat 2 submenu lagi yaitu memilih Ketegori Keluhan dimana superadmin dapat melihat daftar keluhan dan bisa mendownload nya ke dalam bentuk *PDF.* Dan submenu lainnya yaitu Kelola Laporan Keluhan dimana super admin dapat meninjau keluhan yang di *report* oleh admin.

* + - 1. ***Activity* Diagram Admin**

Pada *activity* diagram admin ini menggambarkan aliran aktivitas yang dilakukan oleh admin dalam sistem yang dirancang. Adapun *activity* diagram admin terdapat pada Gambar 3.6.

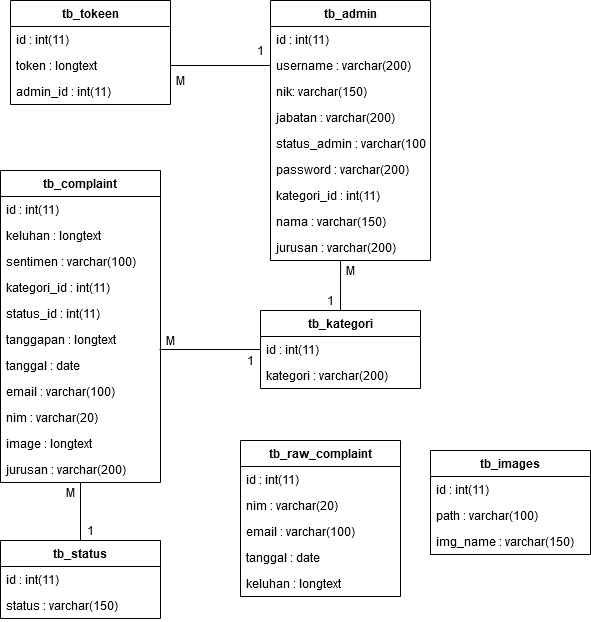


**Gambar 3.6** Activity Diagram Admin

Pada *activity* diagram admin ini dimulai dengan membuka aplikasi mobile *E-complaint* Polindra, lalu admin diharuskan *Login* terlebih dahulu untuk masuk kedalam sistem dengan menggunakan akun yang telah dibuatkan oleh superadmin. Setelah masuk ke halaman *Home*, admin dapat mengklik tombol Kategori Keluhan untuk dapat melihat daftar keluhan. Jika keluhan di klik maka akan muncul *detail* keluhan. Pada *screen* ini *user* dapat memberi tanggapan dengan mengisi *text area* lalu klik tombol Beri Tanggapan, atau admin juga dapat melapurkan keluhan dengan klik tombol Laporkan.

### ***Class* Diagram**

*Class Diagram* merupakan bagan yang menggambarkan jenis-jenis objek dalam sistem dengan berbagai macam relasi yang dimiliki. *Class diagram* juga menjelaskan hubungan antar *class* dalam sebuah sistem yang sedang dibuat dan bagaimana cara mereka saling berkolaborasi untuk mencapai sebuah tujuan. Adapun *class diagram* dari sistem identifikasi keluhan dengan *text mining* ini terdapat pada Gambar 3.7.



**Gambar 3.7** Class Diagram



## Flowchart

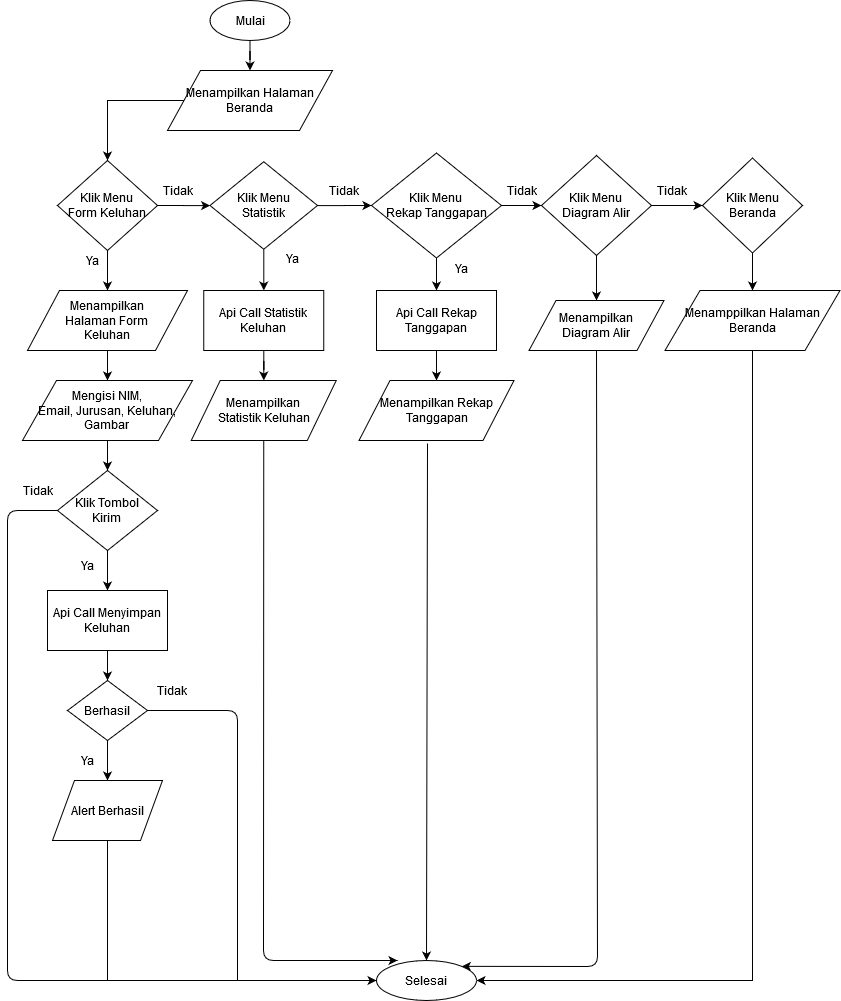
*Flowchart* yang dibuat untuk sistem identifikasi keluhan dengan *text mining* ini terdiri dari *flowchart user*, *flowchart* admin, dan *flowchart* superadmin.



### **Flowchart User**

*Flowchart* ini adalah alur berjalannya sistem dari sisi *user. Flowchart* ini terdapat

pada Gambar 3.8



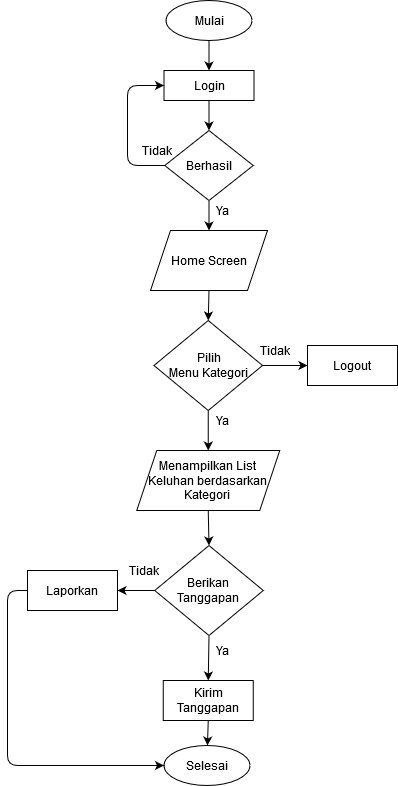
**Gambar 3.8** Flowchart User

*Flowchart user* dimulai dari menampilkan halaman beranda pada *interface,* apabila user memilih menu *form* keluhan maka user akan diarahkan ke halaman yang berisi *form* untuk menuangkan keluhannya. Ketika *user* memilih menu statistik maka sistem akan menampilkan jumlah presentase keluhan yang masuk di tiap-tiap kategori. Jika *user* memilih menu rekap tanggapan maka *user* dapat melihat tanggapan dari keluhan yang pernah diinputkannya. Terakhir, jika *user* memilih menu diagram alir keluhan *user* dapat melihat bagaimana alur sebuah keluhan diproses.



### **Flowchart Admin**

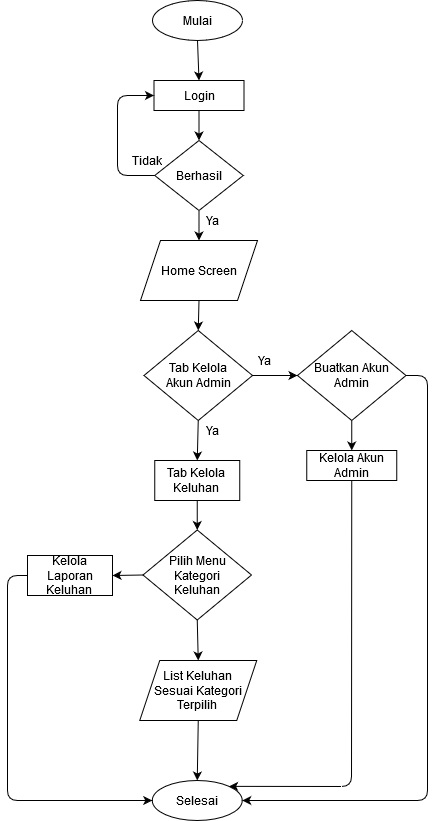
*Flowchart* ini adalah alur berjalannya sistem dari sisi *admin. Flowchart* ini terdapat pada Gambar 3.9.



**Gambar 3.9** Flowchart Admin

*Flowchart* admin dimulai dengan melakukan *login* menggunakan akun yang telah dibuatkan oleh superadmin, kemudian admin diarahkan ke *home screen.* Selanjutnya *user* dapat memilih menu kategori keluhan, kemudian sistem akan menampilkan daftar keluhan di kategori admin tersebut. Ketika admin mengklik salah satu keluhan maka sistem akan menampilkan halaman detail keluhan. Pada halaman ini admin dapat memberikan tanggapan atau melaporkan keluhan.

### ***Flowchart* Superadmin**



**Gambar 3.10** Flowchart Superadmin

Pada *flowchart* superadmin dimulai dengan *login* menggunakan akun yang dibuat langsung di *database,* kemudian superadmin diarahkan ke halaman *home.* Pada halaman ini terdapat 2 *tab,* yaitu kelola akun dan kelola keluhan. Jika user memilih kelola akun maka user dapat membuatkan akun untuk seorang admin maupun mengelola akun admin yang telah terdaftar, misalnya mengedit atau menghapus akun.

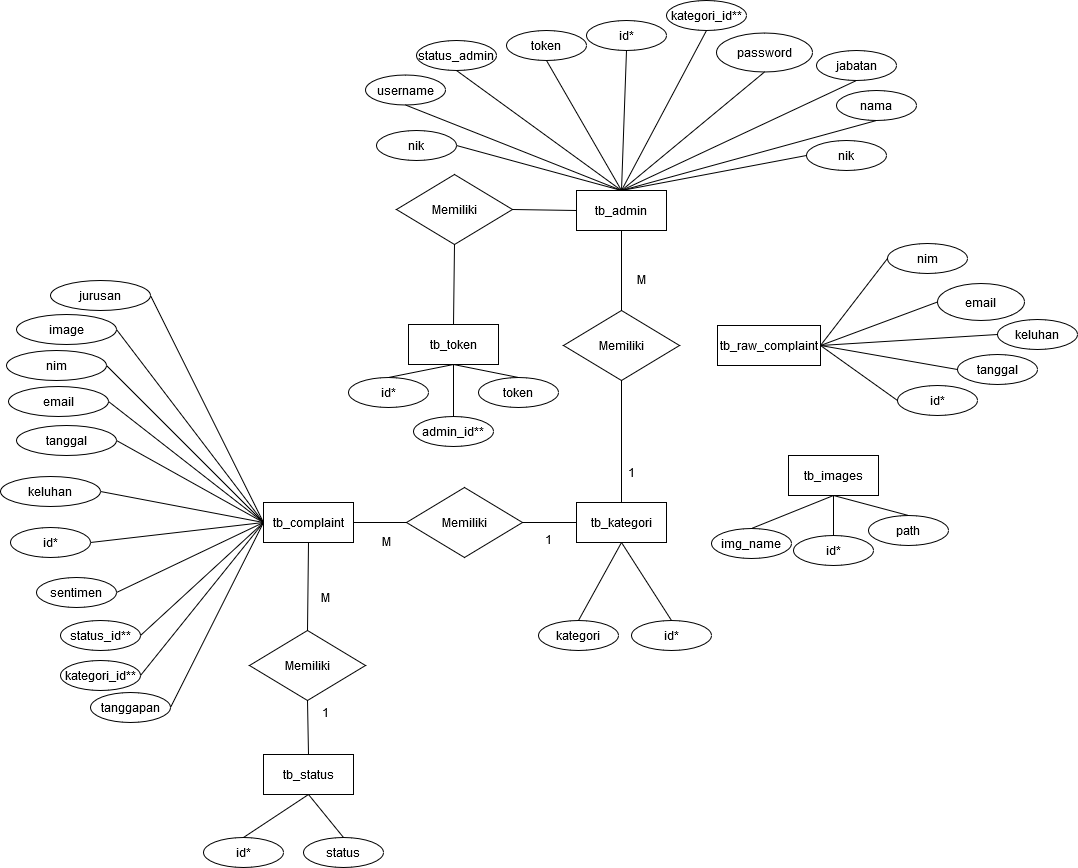


## Perancangan ERD (Entity Relationship Diagram)

Pada Gambar 3.11 merupakan gambar *Entity Relationship Diagram (ERD)* yang digunakan untuk proses Analisa perancangan dari sistem identifikasi keluhan dengan *text mining*. Adapun penjelasan dari Gambar 3.10 adalah pada Tabel 3.6 berikut ini :

**Tabel 3.6** Penjelasan ERD

|  |  |
| --- | --- |
| **Nama Tabel** | **Keterangan** |
| tb\_complaint | Tabel yang digunakan untuk menyimpan data keluhan yang sudah diidentifikasi |
| tb\_raw\_complaint | Tabel yang digunakan untuk menyimpan keluhan yang belum diidentifikasi (untuk keperluan *datasets*) |
| tb\_status | Tabel yang digunakan untuk menyimpan status keluhan |
| tb\_images | Tabel yang digunakan untuk menyimpan data dukung keluhan |
| tb\_token | Tabel yang digunakan untuk menyimpan token di tiap-tiap *device* yang digunakan untuk *login* |
| tb\_kategori | Tabel yang digunakan untuk menyimpan kategori keluhan |
| tb\_admin | Tabel yang digunakan untuk meyimpan data admin dan superadmin |



**Gambar 3.11** Perancangan ERD

Pada sistem identifikasi keluhan dengan *text mining* ini terdiri dari 7 tabel yaitu tb\_*complaint*, tb\_*raw*\_*complaint*, tb\_status, tb\_*images*, tb\_token, tb\_kategori, dan tb\_admin*.*



## Perancangan Database

Berikut ini merupakan tabel dari perancangan sistem identifikasi keluhan dengan *text mining*.



### **Tabel tb*\_admin***

Tabel tb\_admin digunakan untuk meyimpan data admin dan superadmin, terdiri dari *field id, username,* nik, jabatan, status\_admin, *password,* kategori\_id, nama, jurusan. Adapun tabel tb\_admin dapat dilihat pada Tabel 3.7

**Tabel 3.7** Tabel tb\_admin.

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Field** | **Tipe Data** | **Keterangan** |
| 1 | id | Int(11) | Untuk menyimpan id admin |
| 2 | username | Varchar(200) | Untuk menyimpan username admin |
| 3 | nik | Varchar(150) | Untuk menyimpan nik admin |
| 4 | jabatan | Varchar(200) | Untuk menyimpan jabatan admin |
| 5 | status\_admin | Varchar(100) | Untuk menyimpan status admin |
| 6 | password | Varchar(200) | Untuk menyimpan password admin |
| 7 | kategori\_id | Int(11) | Untuk menyimpan id kategori admin |
| 8 | nama | Varchar(150) | Untuk menyimpan nama admin |
| 9 | jurusan | Varchar(200) | Untuk menyimpan jurusan admin |

### **Tabel tb\_*complaint***

Tabel tb\_*complaint* digunakan untuk menyimpan data keluhan yang sudah diidentifikasi, terdiri dari *field id,* keluhan, sentiment, kategori\_id, status\_id, tanggapan, tanggal, *email*, nim*, images,* jurusan. Adapun tabel tb\_*complaint* dapat dilihat pada Tabel 3.8.

**Tabel 3.8** Tabel tb\_complaint

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Field** | **Tipe Data** | **Keterangan** |
| 1 | id | Int(11) | Untuk menyimpan id complaint |
| 2 | keluhan | Longtext | Untuk menyimpan keluhan |
| 3 | sentimen | Varchar(100) | Untuk menyimpan sentimen |
| 4 | kategori\_id | Int(11) | Untuk menyimpan kategori dari keluhan |
| 5 | status\_id | Int(11) | Untuk menyimpan status dari keluhan |
| 6 | tanggapan | Longtext | Untuk menyimpan tanggapan |
| 7 | tanggal | Date | Untuk menyimpan tanggal dibuatnya keluhan |
| 8 | email | Varchar(100) | Untuk menyimpan email dari pengguna |
| 9 | nim | Varchar(20) | Untuk menyimpan nim dari pengguna |
| 10 | image | Longtext | Untuk menyimpan image keluhan |
| 11 | jurusan | Varchar(200) | Untuk menyimpan jurusan dari pengguna |



### **Tabel tb\_*raw\_complaint***

Tabel tb\_*raw\_complaint* digunakan untuk menyimpan keluhan yang belum diidentifikasi (untuk keperluan *datasets*), terdiri dari *field id,* nim, *email,* tanggal, keluhan. Adapun tabel tb\_*raw\_complaint* dapat dilihat pada Tabel 3.9.

**Tabel 3.9** Tabel tb\_raw\_complaint

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Field** | **Tipe Data** | **Keterangan** |
| 1 | id | Int(11) | Untuk menyimpan id keluhan |
| 2 | nim | Varchar(20) | Untuk menyimpan nim dari pengguna |
| 3 | email | Varchar(100) | Untuk menyimpan email dari pengguna |
| 4 | tanggal | Date | Untuk menyimpan tanggal dibuatnya keluhan |
| 5 | keluhan | longtext | Untuk menyimpan keluhan |

### **Tabel tb\_status**

Tabel tb\_statusdigunakan untuk menyimpan status keluhan, terdiri dari *field id,* dan status keluhan. Adapun tabel tb\_statusdapat dilihat pada Tabel 3.11.

**Tabel 3.10** Tabel tb\_status.

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Field** | **Tipe Data** | **Keterangan** |
| 1 | id | Int(11) | Untuk menyimpan id status |
| 2 | status | Varchar(150) | Untuk menyimpan nama status |

### **Tabel tb\_kategori**

Tabel tb\_kategoridigunakan untuk menyimpan kategori keluhan, terdiri dari *field id,* dan kategorikeluhan. Adapun tabel tb\_ kategoridapat dilihat pada Tabel 3.11.

**Tabel 3.11** Tabel tb\_ kategori

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Field** | **Tipe Data** | **Keterangan** |
| 1 | id | Int(11) | Untuk menyimpan id kategori |
| 2 | kategori | Varchar(200) | Untuk menyimpan nama kategori |

### **Tabel tb\_token**

Tabel tb\_tokendigunakan untuk menyimpan token di tiap-tiap *device* yang digunakan untuk *login*, terdiri dari *field id,* token,dan amin\_id. Adapun tabel tb\_ tokendapat dilihat pada Tabel 3.12.

**Tabel 3.12** Tabel tb\_ token

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Field** | **Tipe Data** | **Keterangan** |
| 1 | id | Int(11) | Untuk menyimpan id token |
| 2 | token | Longtext | Untuk menyimpan token device pengguna |
| 3 | admin\_id | Int(11) | Untuk menyimpan id admin |

### **Tabel tb\_images**

Tabel tb\_imagesdigunakan untuk menyimpan data dukung keluhan, terdiri dari *field id, path*,dan *image\_name*. Adapun tabel tb\_ imagesdapat dilihat pada Tabel 3.13.

**Tabel 3.13** Tabel tb\_ images

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Field** | **Tipe Data** | **Keterangan** |
| 1 | id | Int(11) | Untuk menyimpan id gambar |
| 2 | path | Varchar(100) | Untuk menyimpan link gambar |
| 3 | img\_name | Varchar(150) | Untuk menyimpan nama gambar |



## Perancangan Desain Antarmuka

Ada beberapa rancangan desain antarmuka yang dibuat untuk sistem identifikasi keluhan secara otomatis menggunakan *text mining*, berikut merupakan rancangannya :

### **Antarmuka *User***

Antarmuka ini merupakan rancangan *user interface* dari sisi *user*, pada antarmuka *user* ini terdapat menu beranda, *form keluan*, statistik, rekap tanggapan, dan diagram alir keluhan.

* + - 1. **Antarmuka Beranda**

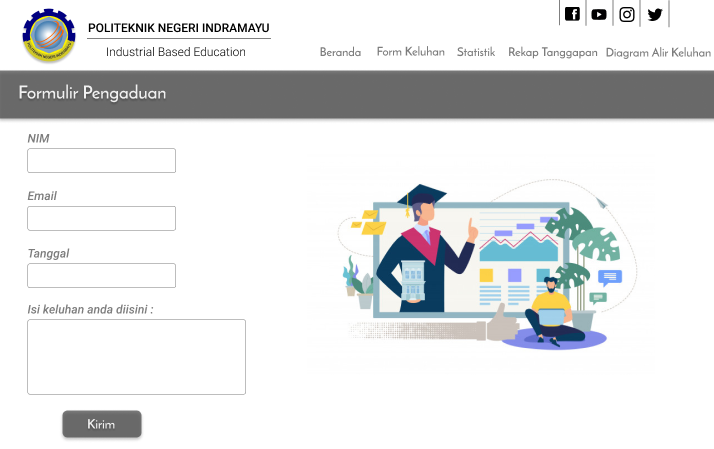
Antarmuka ini ditampilkan ketika *user* baru mengakses *URL website Ecomplaint* Polindra, pada halaman ini berisi informasi singkat mengenai sistem serta terdabat beberapa menu yaitu, *form* keluhan, statistik, rekap tanggapan dan diagram alir keluhan.Adapun rancangan antarmuka beranda dapat dilihat pada Gambar 3.12.



**Gambar 3.12** Antarmuka Beranda (User)

* + - 1. **Antarmuka *Form* Keluhan**

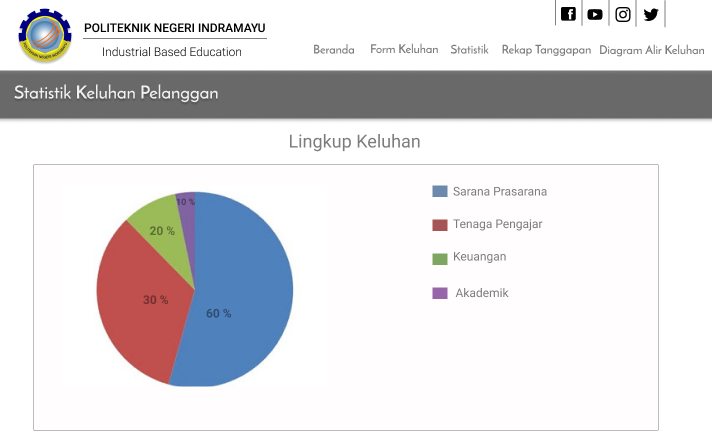
Antarmuka ini ditampilkan ketika memilih menu *form* keluhan, pada halaman ini berisi *form* yang harus diisi ketika user ingin memberikan keluhan. Adapun rancangan antarmuka *form* keluhan dapat dilihat pada Gambar 3.13.



**Gambar 3.13** Antarmuka Form Keluhan (User)

* + - 1. **Antarmuka Statistik**

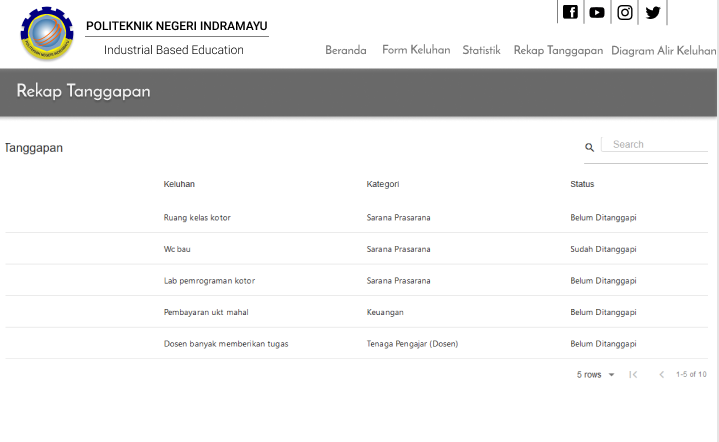
Antarmuka ini ditampilkan ketika memilih menu statistik, pada halaman ini berisi diagram lingkaran yang memberikan persentase jumlah keluhan yang masuk di tiap-tiap kategori. Adapun rancangan antarmuka statistic dapat dilihat pada Gambar 3.14.



**Gambar 3.14** Antarmuka Statistik (User)

* + - 1. **Antarmuka Rekap Tanggapan**

Antarmuka ini ditampilkan ketika memilih menu rekap tanggapan, pada daftar seluruh keluhan yang masuk beserta status tanggapan nya sehingga *user* dapat mengetahui apakan keluhannya sudah ditanggapi atau belum. Adapun rancangan antarmuka rekap tanggapan dapat dilihat pada Gambar 3.15.



**Gambar 3.15** Antarmuka Rekap Tanggapan (User)

### **Antarmuka Admin**

Antarmuka ini merupakan rancangan *user interface* dari sisi admin, pada antarmuka adminini terdapat menu kategori keluhan, berikan tanggapan, dan laporkan keluhan.

* + - 1. **Antarmuka Splashscreen**

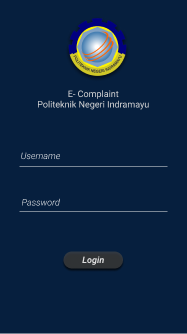
Antarmuka ini ditampilkan sebagai tampilan pertama program yang muncul sementara sebelum masuk ke menu utama. Adapun rancangan antarmuka *splashscreen* dapat dilihat pada Gambar 3.16.

**

**Gambar 3.16** Antarmuka Splashscreen (Admin)

* + - 1. **Antarmuka *Login***

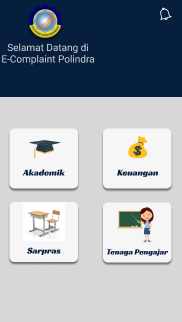
Antarmuka ini ditampilkan sebagai syarat admin utnuk dapat masuk ke dalam sistem dengan memasukan akun yang telah dibuatkan oleh superadmin. Adapun rancangan antarmuka *login* dapat dilihat pada Gambar 3.17.



**Gambar 3.17** Antarmuka Login (Admin)

* + - 1. **Antarmuka *Home***

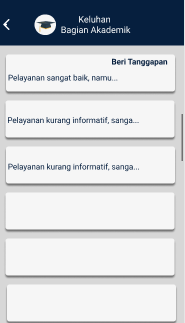
Antarmuka ini ditampilkan setelah admin berhasil melakukan *login*. Pada halaman *home* ini berisi menu kategori keluhan yang dimiliki admin tersebut. Adapun rancangan antarmuka *home* dapat dilihat pada Gambar 3.18.



**Gambar 3.18** Antarmuka Home (Admin)

* + - 1. **Antarmuka *List* Keluhan**

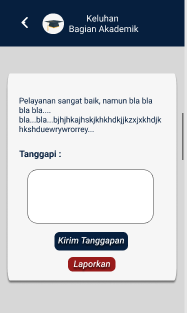
Antarmuka ini ditampilkan ketika admin memilih menu kategori keluhan pada halaman sebelumnnya, halaman ini menampilkan daftar keluhan di kategori admin tersebut. Adapun rancangan antarmuka *list* keluhan dapat dilihat pada Gambar 3.19.



**Gambar 3.19** Antarmuka List Keluhan (Admin)

* + - 1. **Antarmuka Beri Tanggapan / Laporkan**

Ketika keluhan di klik di halaman sebelumnya, maka sistem akan menampilkan detail keluhan pada halaman ini. Selain itu juga terdapat dua *button* untuk memberi tanggapan atau melaporkan keluhan. Adapun rancangan antarmuka beri tanggapan / laporan dapat dilihat pada Gambar 3.20.



**Gambar 3.20** Antarmuka Beri Tanggapan / Laporkan (Admin)

### **Antarmuka Superadmin**

Antarmuka ini merupakan rancangan *user interface* dari sisi superadmin, pada antarmuka superadminini terdapat menu kategori keluhan, *list* keluhan, dan kelola laporan keluhan.

* + - 1. **Antarmuka Splashscreen**

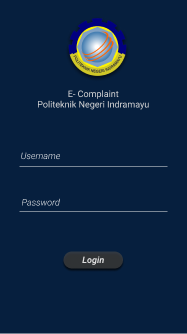
Antarmuka ini ditampilkan sebagai tampilan pertama program yang muncul sementara superadmin belum masuk ke menu utama. Adapun rancangan antarmuka *splashscreen* dapat dilihat pada Gambar 3.21.

**

**Gambar 3.21** Antarmuka Splashscreen (Superadmin)

* + - 1. **Antarmuka *Login***

Antarmuka ini ditampilkan sebagai syarat admin utnuk dapat masuk ke dalam sistem dengan memasukan akun yang telah dibuatkan oleh superadmin. Adapun rancangan antarmuka *login* dapat dilihat pada Gambar 3.22.



Antarmuka Login (Superadmin)

**Gambar 3.22** Antarmuka Login (Superadmin)

* + - 1. **Antarmuka *Home***

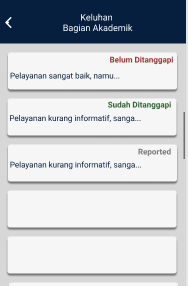
Antarmuka ini ditampilkan setelah superadmin berhasil melakukan *login.* Pada halaman *home* ini terdapat dua *tab* untuk mengelola keluhan atau mengelola akun admin. Adapun rancangan antarmuka *home* dapat dilihat pada Gambar 3.23.



**Gambar 3.23** Antarmuka Home (Superadmin)

* + - 1. **Antarmuka *List* Keluhan**

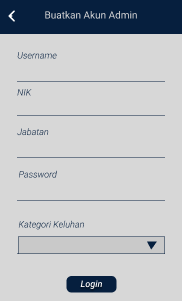
Jika superadmin telah memilih salah satu kategori keluhan di screen sebelumnya maka sistem akan mengarahkan pada *screen* *list* keluhan yang berisi daftar keluhan pada kategori yang dipili tersebut. Adapun rancangan antarmuka *list* keluhan dapat dilihat pada Gambar 3.24.



**Gambar 3.24** Antarmuka List Keluhan (Superadmin)

* + - 1. **Antarmuka Buatkan Akun**

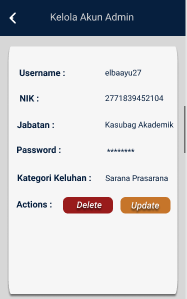
Antarmuka ini ditampilkan ketika superadmin mengklik *button* buatkan akun pada *tab* kelola admin*.* Halaman ini berfungsi untuk membuatkan akun bagi admin baru. Adapun rancangan antarmuka buatkan akun dapat dilihat pada Gambar 3.25.



**Gambar 3.25** Antarmuka Buatkan Akun (Superadmin)

* + - 1. **Antarmuka Kelola Akun**

Antarmuka ini ditampilkan ketika superadmin mengklik *button* kelola akun pada *tab* kelola admin*.* Halaman ini berfungsi untuk mengelola akun admin. Adapun rancangan antarmuka kelola akun dapat dilihat pada Gambar 3.25.



**Gambar 3.26** Antarmuka Buatkan Akun (Superadmin)

# BAB IV HASIL DAN PEMBAHASAN

## Hasil

Penelitian ini menghasilkan sebuah sistem identifikasi keluhan dengan *text mining*. Sistem ini digunakan untuk membantu mempermudah pencatatan komplain dan mengidentifikasi kategori complain di Politeknik Negeri Indramayu.

Penulis dapat menyimpulkan bahwa sistem identifikasi keluhan dengan *text mining* yang telah dibuat adalah sistem yang digunakan sebagai media penjaminan mutu (*quality assurance*) pendidikan tinggi sebagai proses penetapan dan pemenuhan standar mutu pendidikan secara konsisten dan berkelanjutan dimaksudkan agar pelanggan memperoleh kepuasan serta menghasilkan pengembangan berkelanjutan (*continous improvement*) diperguruan tinggi.

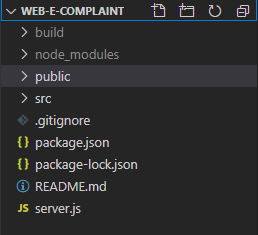
## Pembahasan

Pada bagian ini merupakan tahap pembahasan dari hasil implementasi yang penulis kumpulkan, analisa dan rancang sehingga terciptalah sistem identifikasi keluhan dengan *text mining*.

### **Struktur Direktori**

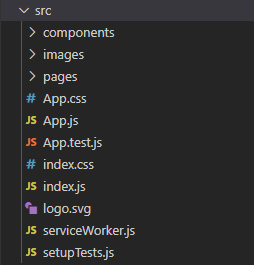
Struktur direktori menjelaskan urutan atau file-file apa saja yang terdapat pada tiap-tiap project dalam membangunsistem identifikasi keluhan dengan *text mining*.

* + - 1. **Struktur Direktori Aplikasi *Ecomplaint* Polindra (*Web*)**

**

**Gambar 4.1** Struktur Direktori Folder Web E-Complaint

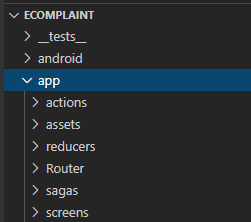
Pada Gambar 4.1 terdapat struktur direktori folder *web e-complaint* yang digunakan untuk membuat sistem identifikasi keluhan dengan *text mining.* Pada folder ini terdapat folder *build, node\_modules, public, dan src.*



**Gambar 4.2** Struktur Direktori Folder src

Pada Gambar 4.2 terdapat struktur direktori folder *src* yang terdiri dari subfolder *components* yang berfungsi sebagai tempat untuk menyimpan file-file komponen ui yang digunakan untuk membuat website Ecomplaint Polindra. Lalu ada subfolder *images* untuk menyimpan *assets* berupa gambar, sedangkan subfolder *pages* berfungsi untuk menyimpan file kode program tiap-tiap *screen* yang memanggil kumpulan komponen di folder *components.*

* + - 1. **Struktur Direktori Aplikasi *Ecomplaint* Polindra (*Mobile*)**

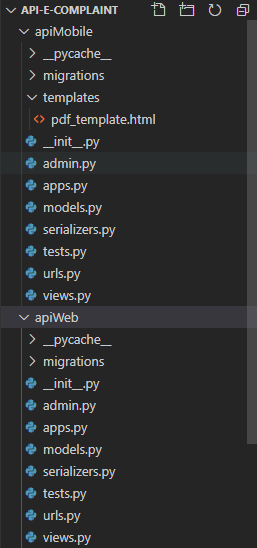
****

**Gambar 4.3** Struktur Direktori Folder app

Pada Gambar 4.3 terdapat stuktur direktori dari folder *app* yang ada pada *project Ecomplaint* (*mobile*). Folder *app* memiliki beberapa subfolder antara lain *actions* merupakan tempat untuk menampung aksi yang berasal dari view (*React Components*), subfolder *assets* yang berisi asset-asset yang berupa gambar untuk keperluan *user interface.*

Subfolder *reducers* sebagai tempat untuk menyimpan *global state* supaya data yang disimpan dapat diakses oleh seluruh *screen,* lalu subfolder *router* untuk membuat *navigation screen,* dan yang terakhir adalah subfolder *screens* sebagai tempat menyimpan tampilan halaman-halaman antarmuka yang terdiri dari banyak komponen.

* + - 1. **Struktur Direktori Aplikasi *Ecomplaint* Polindra (*API*)**

****

**Gambar 4.4** Struktur Direktori Folder API Ecomplaint

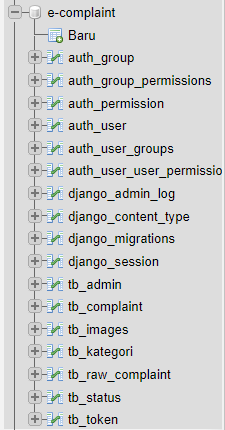
Pada Gambar 4.4 terdapat stuktur direktori dari folder project *API Ecomplaint* dimana *project* ini berfungsi untuk membuat kebutuhan *API* untuk membuat sistem identifikasi keluhan dengan *text mining.* Didalam *project* ini terdapat dua folder yaitu *apiMobile* untuk membuat *API* di aplikasi bagian *mobile* dan *apiWeb* untuk membuat keperluan *API* di aplikasi bagian *web.*

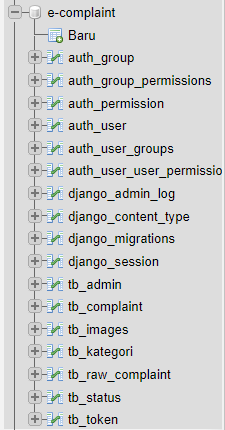
### **Implementasi *Database***

*Database* sistem identifikasi keluhan dengan *text mining* terdiri dari 7 tabel yaitu tabel tb\_*complaint*, tabel tb\_*raw\_complaint,* tabel tb\_status, tabel tb\_*images,* tabel tb\_*token,* tabel tb\_kategori, dan tabel tb\_admin. Berikut implementasi pembuatan *database* untuk aplikasi sistem identifikasi keluhan dengan *text mining.*

**4.2.2.1 *Database e-complaint***

*Database e-complaint* merupakan nama *database* yang dibuat untuk menyimpan tabel-tabel *database* yang telah digunakan. *Database* pakar dapat dilihat pada Gambar 4.5.



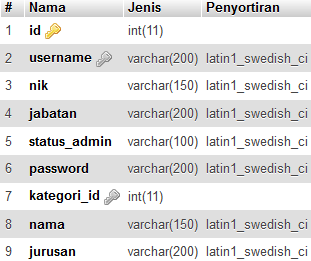


**Gambar 4.5** Database e-complaint

Pada database pakar terdiri dari tabel tb\_*complaint*, tabel tb\_*raw\_complaint,* tabel tb\_status, tabel tb\_*images,* tabel tb\_*token,* tabel tb\_kategori, dan tabel tb\_admin*.*

**4.2.2.2 Tabeltb\_admin**

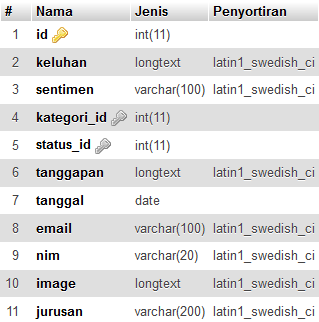
Tabel ini digunakan untuk menyimpan data admin (baik admin maupun superadmin). Tabel tb\_admin dapat dilihat pada Gambar 4.6. Pada tabel tb\_admin terdiri dari *field* id, *username*, nik, jabatan, status\_admin, *password,* kategori\_id, nama, dan jurusan.



**Gambar 4.6** Tabel tb\_admin

**4.2.2.3 Tabeltb\_*complaint***

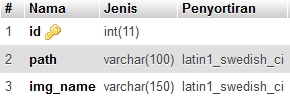
Tabel ini digunakan untuk menyimpan data keluhan yang sudah teridentifikasi kategorinya melalui proses *text mining*. Tabel tb\_*complaint* dapat dilihat pada Gambar 4.7. Pada tabel tb\_*complaint* terdiri dari *field* id, keluhan, sentiment, kategori\_id, status\_id, tanggapan, tanggal, *email*, nim, *image,* dan jurusan.



**Gambar 4.7** Tabel tb\_complaint

**4.2.2.4 Tabeltb\_*images***

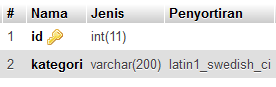
Tabel ini digunakan untuk menyimpan data dukung yang di inputkan oleh *user* sewaktu mengisi *form* keluhan. Tabel tb\_*images* dapat dilihat pada Gambar 4.8. Pada tabel tb\_*images* terdiri dari *field* id, *path,* dan *img\_name*.



**Gambar 4.8** Tabel tb\_images

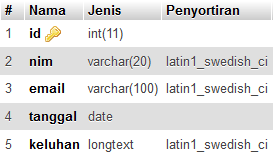
**4.2.2.5 Tabeltb\_kategori**

Tabel ini digunakan untuk menyimpan jenis kategori untuk tiap-tiap keluhan. Tabel tb\_kategori dapat dilihat pada Gambar 4.9. Pada tabel tb\_kategori terdiri dari *field* id*,* dan kategori.



**Gambar 4.9** Tabel tb\_kategori

**4.2.2.6 Tabeltb\_*raw\_complaint***

Tabel ini digunakan untuk menyimpan data keluhan yang masi mentah atau belum diidentifikasi. Tabel tb\_*raw\_complaint* dapat dilihat pada Gambar 4.10. Pada tabel tb\_*raw\_complaint*  terdiri dari *field* id*,* nim, *email,* tanggal, dan keluhan.

**Gambar 4.10** Tabel tb\_raw\_complaint

**4.2.2.7 Tabeltb\_status**

Tabel ini digunakan untuk menyimpan data dari jenis-jenis status tanggapan keluhan. Tabel tb\_status dapat dilihat pada Gambar 4.11. Pada tabel tb\_status terdiri dari *field* id*,* dan status.



**Gambar 4.11** Tabel tb\_status

**4.2.2.8 Tabeltb\_token**

Tabel ini digunakan untuk menyimpan data dari jenis-jenis status tanggapan keluhan. Tabel tb\_token dapat dilihat pada Gambar 4.12. Pada tabel tb\_token terdiri dari *field* id*,* token, dan admin\_id.



**Gambar 4.12** Tabel tb\_token

### **Implementasi Desain Antarmuka *(Interface)***

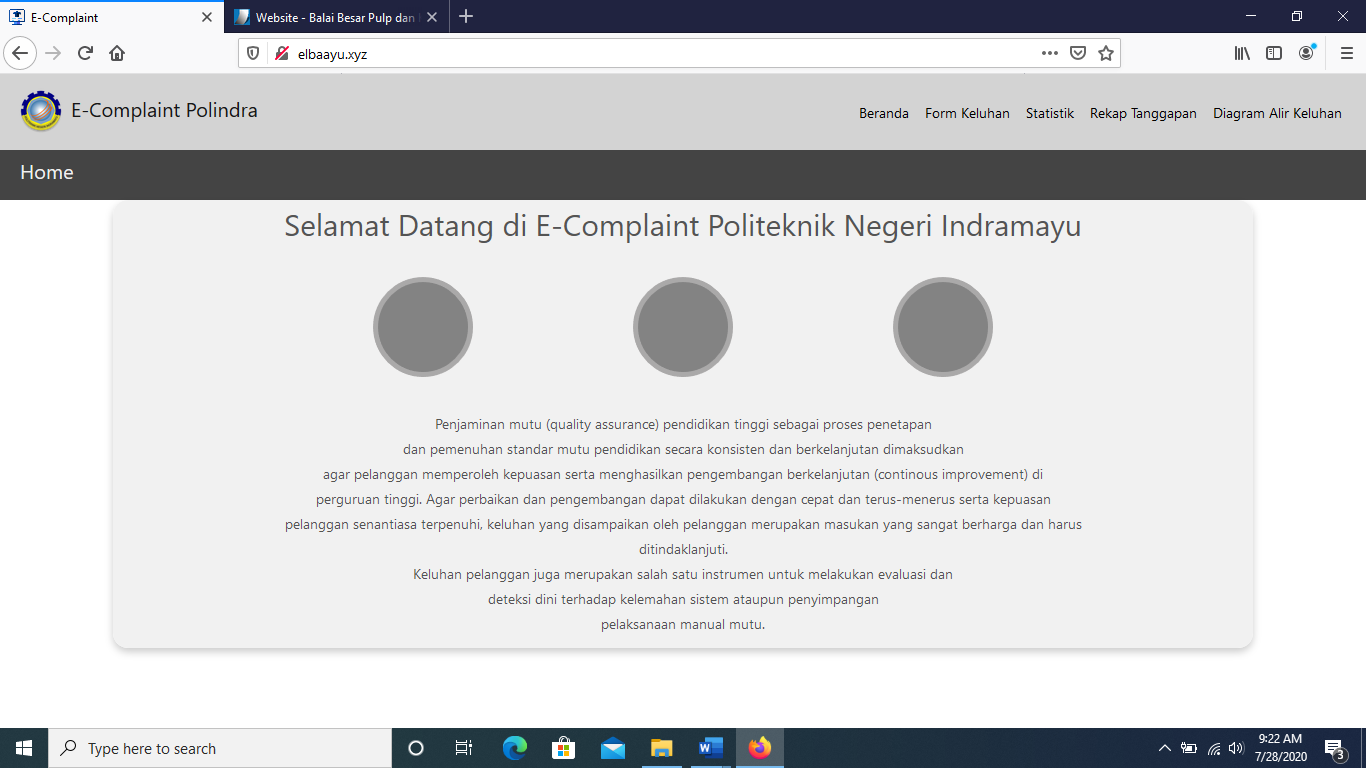
Antarmuka atau *interface* merupakan cara untuk memudahkan pengguna dalam melakukan aktivitas dalam sistem. Terdapat 3 pengelompokan antarmuka, yaitu pertama antarmuka untuk *user* (mahasiswa)*,* antarmuka ini digunakan sebagai media bagi *user* untuk menuangkan keluhannya. Antarmuka kedua yaitu admin, antarmuka ini digunakan admin (ketua sub bagian atau unit kerja) untuk mengelola keluhan yang masuk dari *user,* mulai dari melihat *detail* keluhan, memberikan tanggapan, maupun melaporkan keluhan. Antarmuka ketiga yaitu antarmuka superadmin (*P4MP*), antarmuka ini digunakan oleh *P4MP* untuk mengontrol status keluhan apakan sudah ditanggapi oleh admin atau belum, dan juga dapat meninjau keluhan yang dilaporkan oleh admin. Selain itu superadmin juga dapat memfilter keluhan berdasarkan bulan atau pun mengunduh keluhan jika diperlukan.

* + - 1. **Antarmuka *User***

Antarmuka *user* ditampilkan untuk pengguna (mahasiswa) yang ingin memberikan complaint / keluhan terkait pelayanan di Polindra. Antarmuka *user* ini terdiri dari halaman beranda, halaman *form* keluhan, halaman statistic, halaman rekap tanggapan, halaman diagram alir keluhan.

* + 1. Halaman Beranda

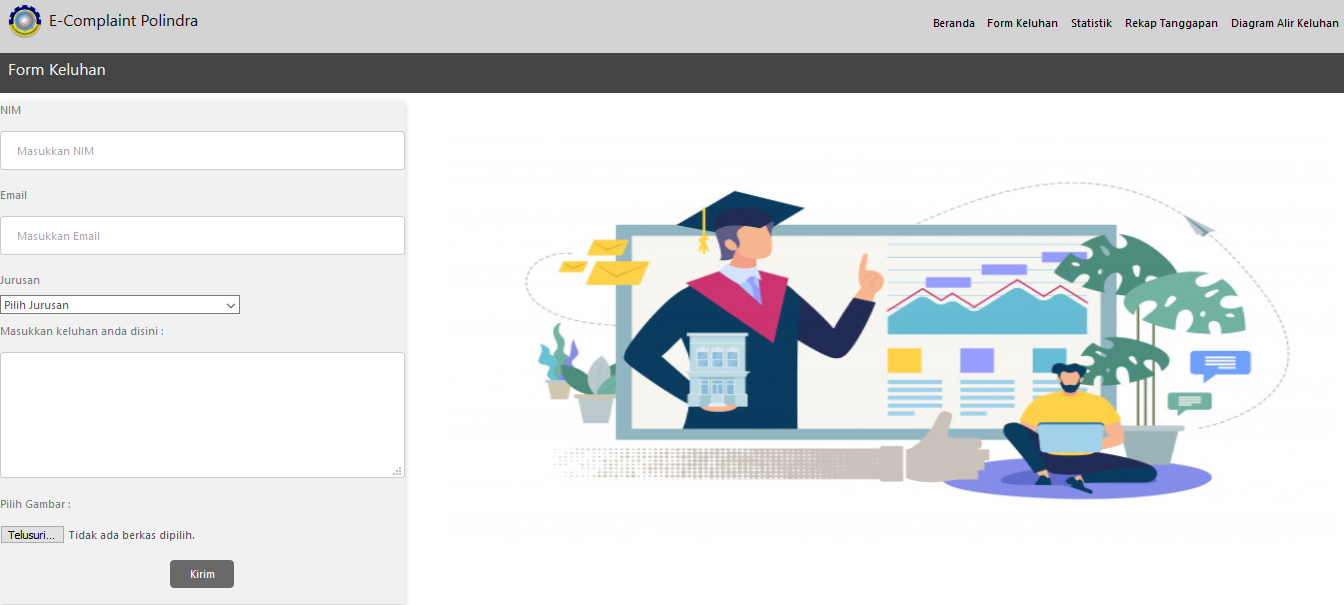
Halaman ini merupakan halaman yang pertama kali ditampilkan setelah *user* memasukan *URL* ke *browser*. Pada halaman ini hanya berisi info singkat mengenai ap aitu aplikasi *Ecomplaint* Polindra. Halaman beranda dapat dilihat pada Gambar 4.13.



**Gambar 4.13** Halaman Beranda

* + 1. Halaman *Form* Keluhan

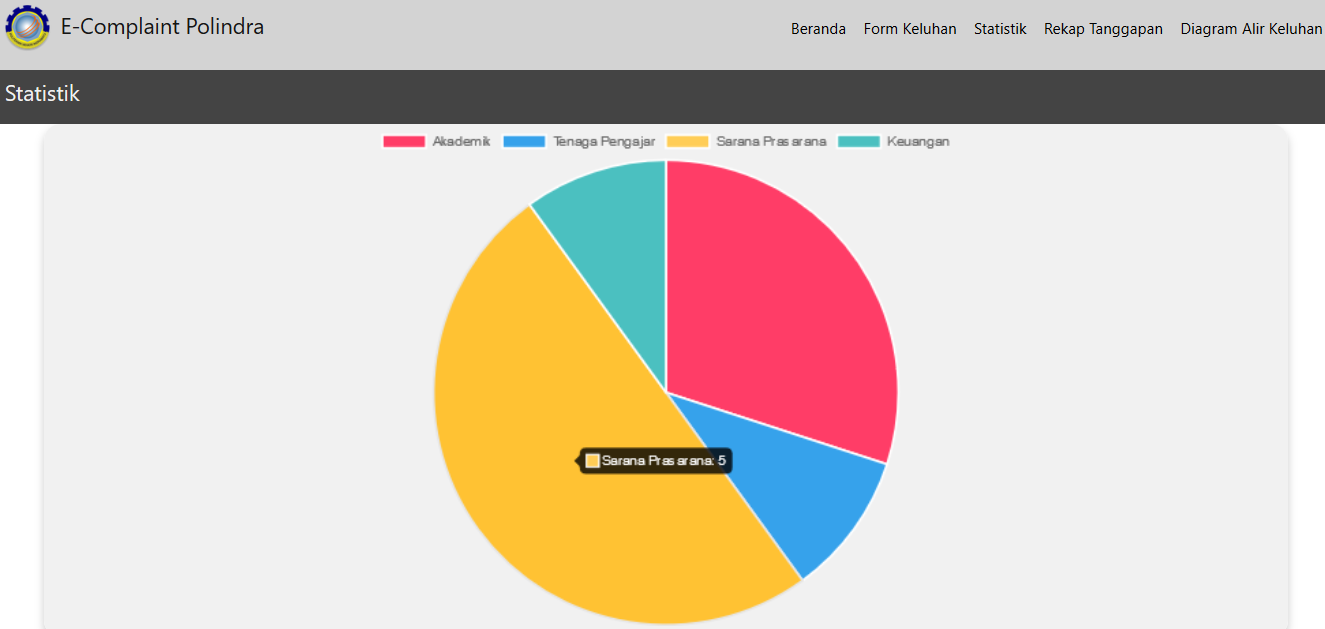
Jika dari halaman beranda *user* memilih menu *form* keluhan, maka halaman *form* keluhan inilah yang akan ditampilkan. Halaman ini merupakan tempat user untuk menuangkan keluhan yang akan disampaikan untuk Politeknik Negeri Indramayu. *Form* yang harus diisi oleh *user* antara lain yaitu nim, *email,* jurusan, kolom keluhan, serta data dukung sebagai bukti keluhan jika ada. Halaman *form* keluhan dapat dilihat pada Gambar 4.14.



**Gambar 4.14** Halaman Form Keluhan

* + 1. Halaman Statistik

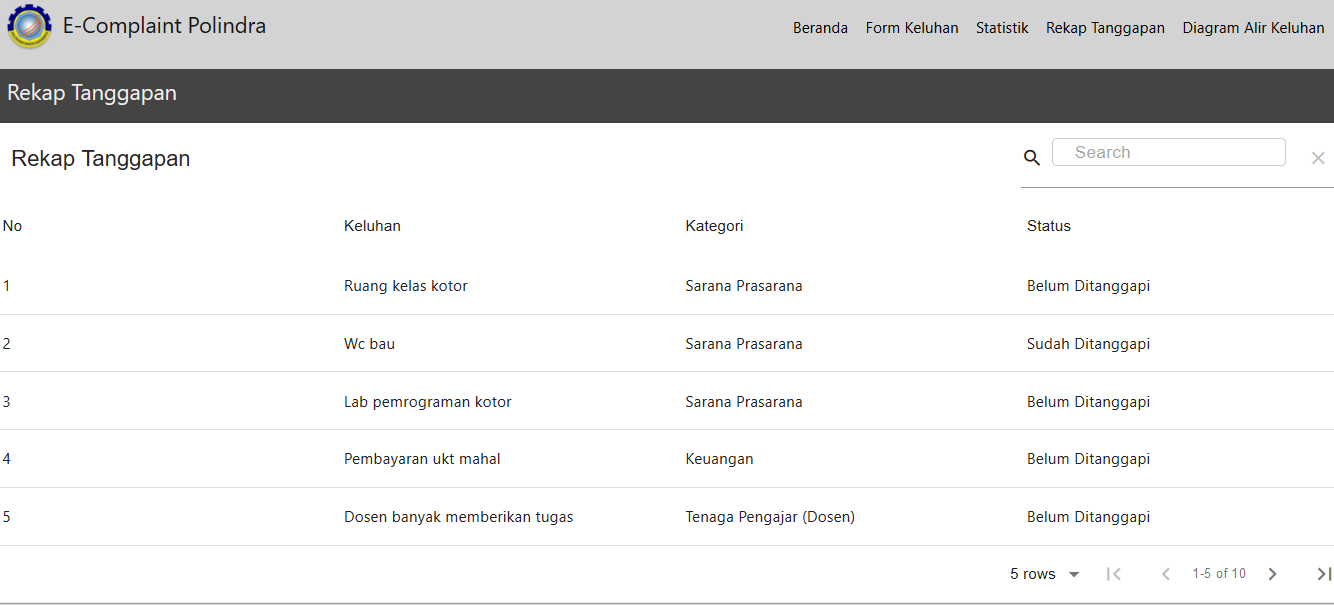
Ketika *user* memilih menu statistik, maka akan muncul halaman statistik ini. Isi dari halaman ini adalah *piechart*  atau diagram lingkaran yang menunjukan persentase keluhan yan g masuk di tiap-tiap kategori. Sehingga *user* dapat mengetahui seberapa banyak keluhan yang masuk pada suatu kategori. Halaman statistik dapat dilihat pada Gambar 4.15.



**Gambar 4.15** Halaman Statistik

* + 1. Halaman Rekap Tanggapan

Pada halaman ini menampilkan data tabel yang berisi rekapan seluruh keluhan yang sudah masuk beserta status keluhannya. Tujuan dari halaman ini adalah agar *user* dapat mengetahui status keluhannya dan apakah keluhannya sudah ditanggapi atau belum. Adapun format dari data tabel tersebut antara lain nomor, keluhan, kategori, dan status keluhan. Halaman rekap tanggapan dapat dilihat pada Gambar 4.16.



**Gambar 4.16** Halaman Rekap Tanggapan

* + 1. Halaman Diagram Alir Keluhan

Halaman ini berisi diagram alir dari bagaimana sebuah keluhan akan di proses oleh sistem. Tujuannya agar user dapat mengetahui bagaimana alur proses dari sistem identifikasi keluhan menggunakan *text mining* ini. Adapun halaman diagram alir keluhan dapat dilihat pada Gambar 4.17.



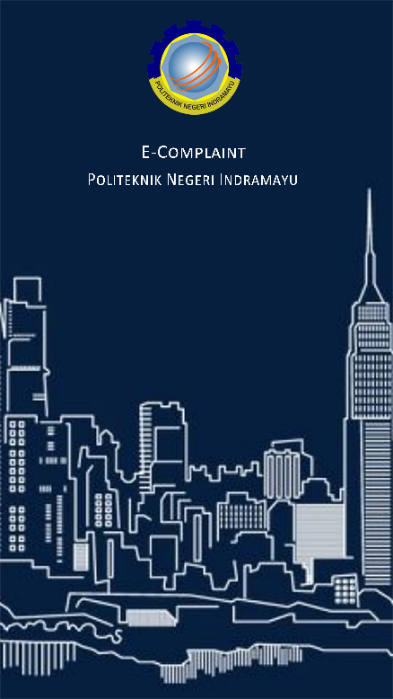
**Gambar 4.17** Halaman Diagram Alir Keluhan

1. 2. 3. 2. **Antarmuka Admin**

Antarmuka adminditampilkan untuk ketua unit kerja (kasubag) agar dapat memberikan tanggapan dari keluhan yang masuk untuk unit kerjanya masing-masing. Antarmuka admin ini terdiri dari beberapa halaman yaitu *spashscreen*, halaman *login,* halaman *home,* halaman *list* keluhan, dan halaman *detail* keluhan. Berikut penjelasan dari masing-masing halaman yang terdapat pada antarmuka admin.

* + 1. *Splashscreen*

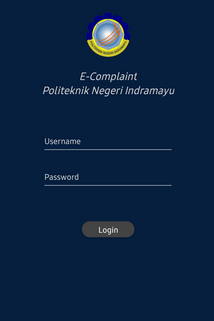
Halaman *splashscreen* ini sebagai tampilan pertama program yang muncul cementara sebelum masuk ke halaman atau menu utama. Adapun *splashscreen* dapat diliat pada Gambar 4.18.



**Gambar 4.18** Splashscreen

* + 1. Halaman *Login*

Halaman *login* ini merupakan halaman untuk masuk ke sistem bagi adminyang telah memiliki akun. Akun admin akan dibuatkan oleh superadmin (P4MP). Pada halaman ini admin harus mengisi *form login* yang telah disediakan oleh sistem, antarmuka *login* dapat dilihat pada Gambar 4.19.



**Gambar 4.19** Halaman Login

* + 1. Halaman *Home*

Halaman *home* ini akan ditampilkan ketika admin berhasil melakukan *login.* Halaman ini berisi menu kategori keluhan milik admin yang *login* tersebut. Adapun tampilan dari halaman *home* dapat dilihat pada Gambar 4.20. Dapat dilihat pada gambar tersebut, admin hanya dapat melihat keluhan di unit kerjanya saja. Jika bukan unit kerjanya, maka tombol kategori keluhan akan *disabled.*



**Gambar 4.20** Halaman Home

* + 1. Halaman *List* Keluhan

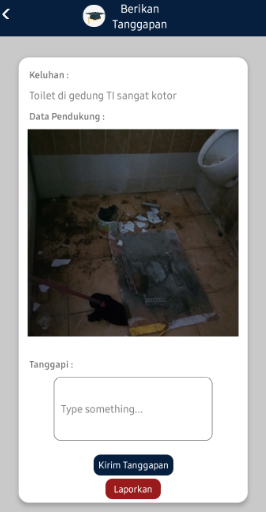
Jika pada halaman *home* sebelumnya admin memilih menu kategori keluhan, maka admin akan diarahkan ke halaman *list* keluhan ini. Pada halaman ini menampilkan daftar keluhan yang ditujukan kepada unit kerja admin tersebut. Adapun tampilan dari halaman *list* keluhan terdapat pada Gambar 4.21.



**Gambar 4.21** Halaman List Keluhan

* + 1. Halaman *Detail* Keluhan

Ketika admin memilih atau mengklik salah satu keluhan pada halaman *list* keluhan sebelumnya, maka sistem akan menampilkan seluruh rincian keluhan pada halaman *detail* keluhan ini. Selain itu pada halaman ini juuga terdapat *text area* untuk admin menuliskan tanggapannya terhadap keluhan tesebut. Serta juga terdapat *button* laporkan untuk melaporkan keluhan yang tidak sesuai. Adapun halaman *detail* keluhan dapat dilihat pada Gambar 4.22.



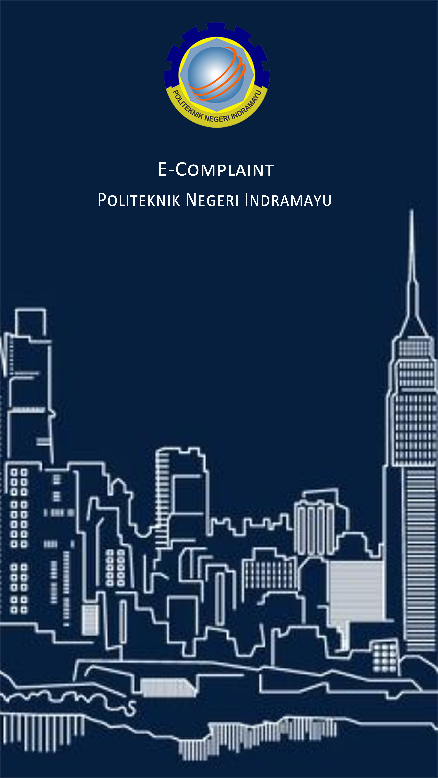
**Gambar 4.22** Halaman Detail Keluhan

1. 2. 3. 3. **Antarmuka Superadmin**

Antarmuka superadmin ditampilkan untuk Pusat Pengembangan Pembelajaran dan Penjaminan Mutu Pendidikan (P4MP) Polindra agar dapat mengontrol seluruh tanggapan dari semua kategori keluhan apakah keluhan sudah ditanggapi oleh admin atau belum. Antarmuka superadmdin ini terdiri dari beberapa halaman yaitu *spashscreen*, halaman *login*, halaman *home*, halaman *list* keluhan, halaman *list* laporan, dan halaman *detail* keluhan . Berikut penjelasan dari masing-masing halaman yang terdapat pada antarmuka superadmin.

1. *Splashscreen* (Superadmin)

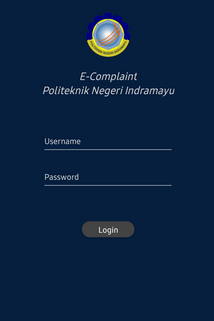
Halaman *splashscreen* ini sebagai tampilan pertama program yang muncul cementara sebelum masuk ke halaman atau menu utama. Adapun *splashscreen* dapat diliat pada Gambar 4.23.



**Gambar 4.23** Splashscreen Superadmin

1. Halaman *Login* (Superadmin)

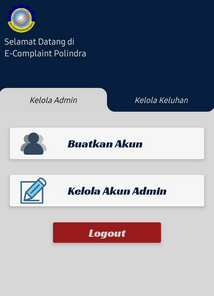
Halaman *login* ini merupakan halaman untuk masuk ke sistem bagi superadmin. Akun superadmin langsung dibuat di *database* sistem. Pada halaman ini superadmin harus mengisi *form login* yang telah disediakan oleh sistem, halaman *login* dapat dilihat pada Gambar 4.24.



**Gambar 4.24** Halaman Login Superadmin

1. Halaman *Home* (Superadmin)

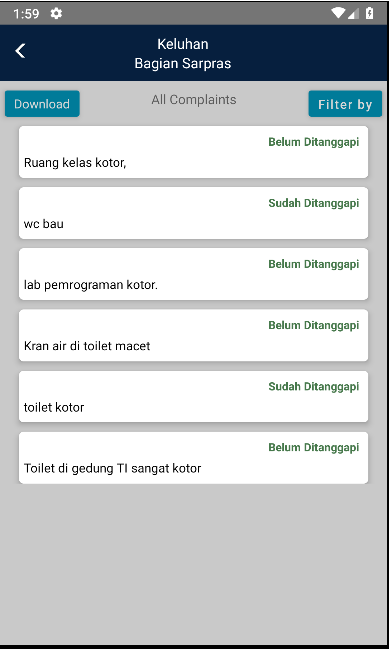
Sistem akan mengarahkan ke halaman *home* jika superadmin berhasil melakukan login. Pada halaman *login* ini terdapat dua *tab,* yang pertama adalah *tab* Kelola Keluhan yang berisi menu kategori keluhan dan kelola laporan keluhan. Sedangkan *tab* kedua yaitu Kelola Akun yang berfungsi untuk mengelola akun admin. Adapun *Tab* Kelola Admin dan *Tab* Kelola Keluhan dapat dilihat pada Gambar 4.25.



**Gambar 4.25** Tab Kelola Keluhan dan Tab Kelola Admin

1. Halaman *List* Keluhan (Superadmin)

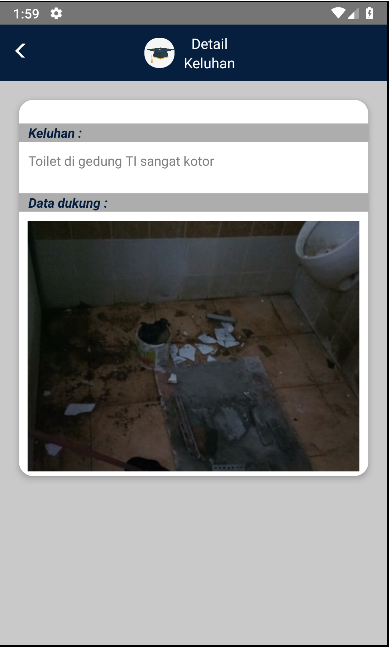
Jika pada saat di halaman *home* superadmin memilih menu salah satu ketegori keluhan maka sistem akan mengarahkan ke halaman *list* keluhan ini. Pada halaman ini berisi seluruh daftar keluhan beserta status tanggapan dari kategori yang dipilih. Selain itu superadmin juga dapat mengunduh data keluhan baik seluruhnya maupun data per bulan. Adapun halaman *list* keluhan dapat dilihat pada Gambar 4.26.



**Gambar 4.26** Halaman List Keluhan

1. Halaman *Detail* Keluhan

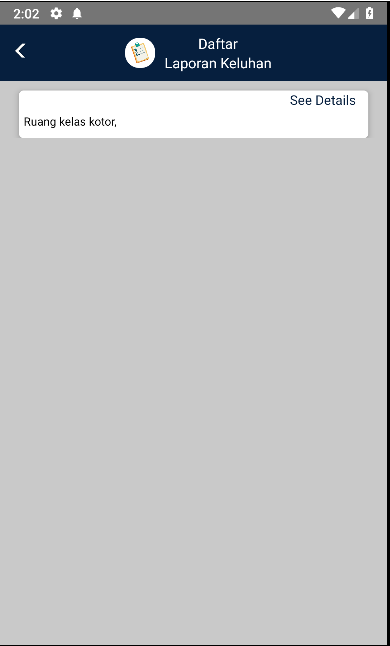
Halaman ini akan ditampilkan ketika superadmin memilih salah satu keluhan pada halaman *list* keluhan. *Detail* keluhan yang ditampilkan antara lain yaitu isi keluhan, tanggapan, serta data dukung keluhan (jika ada). Adapun tampilan dari halaman *detail* keluhan dapat dilihat pada Gambar 4.27.



**Gambar 4.27** Halaman Detail Keluhan.

1. Halaman *List* Laporan Keluhan

Halaman *list* laporan keluhan berisi seluruh daftar laporan yang di *report* atau dilaporkan oleh admin karena bisa saja keluhan yang masuk berisi *spam*, karena sistem yang dibuat belum dapat menangani keluhan *spam* yang masuk maka permasalahan tersebut di *handle* pada fitur laporkan. Adapun tampilan dari halaman *list* laporan keluhan dapat dilihat pada Gambar 4.28.

.

**Gambar 4.28** Halaman List Laporan Keluhan



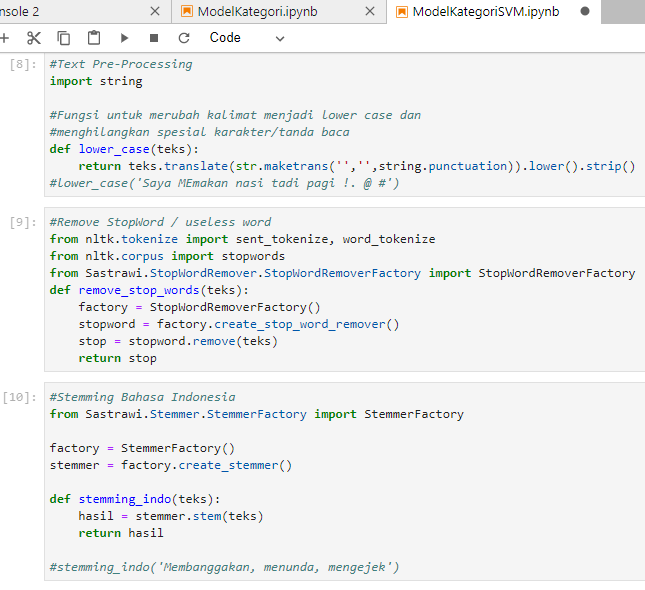
### **Implementasi Pembuatan Model**

Pengklasifikasian teks dalam hal ini keluhan, dilakukan dengan cara mengkategorikan dokumen-dokumen ke dalam satu atau beberapa dari sekelompok topik-topik. Setelah proses pengelompokan telah selesai, maka akan dilakukan proses *filtrasi* sehingga dapat ditentukan kelompok klasifikasi mana yang sesuai. Proses ini disebut dengan model, sedangkan model tersebut dibuat dengan beberapa tahapan atau proses, antara lain teks *preprocessing, feature extraction* atau *vectorizer,* dan *classifier.*

* + - 1. **Teks *Preprocessing***

Teks *preprocessing* merupakan sebuah langkah penting dalam *text mining*. Data ini pada umumnya berupa noise, ukuran yang besar, dan merupakan campuran dari berbagai macam sumber. Teks *preprocessing* sangat diperlukan karena teks yang tidak berkualitas akan menghasilkan kualitas mining yang tidak baik.

Pada penelitian ini penulis menggunakan 3 tahapan teks *preprocessing* yaitu *Lowercase* untuk membuat huruf menjadi nonkapital, *Remove Stopword* untuk mengurangi jumlah kata yang dianggap tidak memiliki makna dalam sebuah dokumen yang nantinya akan berpengaruh dalam kecepatan dan peforma, dan yang terakhir adalah *Stemming* untuk penguraian bentuk dari suatu kata menjadi bentuk kata dasarnya. Teks *preprocessing* dapat dilihat pada Gambar 4.29.

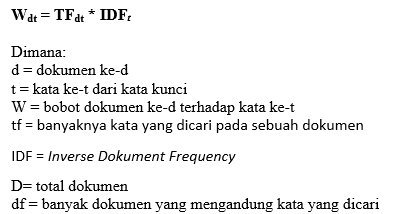


**Gambar 4.29** Teks Preprocessing

* + - 1. ***Feature Extraction***

*Feature extraction* adalah proses pengambilan ciri-ciri yg unik dari data yg akan diolah. Tujuan *feature extraction* diantaranya adalah memperkecil jumlah data, mengambil informasi yg terpenting dari data yg diolah, dan mempertinggi presisi pengolahan. Ada banyak algoritma untuk melakukan proses *feature extraction* antara lain *CountVectorizer* dan *Term Frequency (TF) dan Inverse Document Frequency (IDF)* atau *TF/IDF*.

TF/IDF dalam proses feature extraction menghitung nilai Term Frequency (TF) dan Inverse Document Frequency (IDF) pada setiap token (kata) di setiap dokumen dalam korpus. Secara sederhana, metode TF-IDF digunakan untuk mengetahui berapa sering suatu kata muncul di dalam dokumen. Adapun cara perhitungan atau rumus TF/IDF dapat dilihat pada Gambar 4.30.

******

**Gambar 4.30** Rumus TF/IDF

Sedangkan *CountVectorizer* mengubah susuan kata dalam sebuah kalimat menjadi vektor dengan menghitung frekuensi kemunculan kata dalam sebuah kalimat. Adapun alur kerja *CountVectorizer* dapat dilihat pada Tabel 4.1.

**Tabel 4.1** Proses CountVectorizer

|  |  |
| --- | --- |
| **Korpus** | **Representasi** **Vektor** |
| Kucing itu berwarna putih | Kucing itu berwarna putih :[0,1,0,1,1,0,1] |
| Sapi itu berwarna coklat | Sapi itu berwarna coklat :[1,1,1,1,0,0,0] |
| Kucing memakan coklat putih | Kucing memakan coklat putih : [0,0,1,0,1,1,1] |

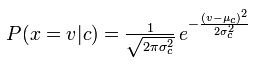
Sebagai contoh kita memiliki 3 buah data teks (korpus), dari data teks yang kita miliki diatas kita dapat menyusun sebuah *vocabulary* yang terdiri dari : anjing, berwarna, coklat, itu, kucing, memakan, putih (7 kata). Kita kemudian menjadikan setiap data menjadi representasi *vector* 7 dimensi (masing — masing untuk setiap kata) Tiap elemen dari *vector* menunjukkan jumlah fitur kata yang ada pada data. Representasinya kurang lebih seperti kolom Representasi Vektor pada Tabel 4.1.

Pada sistem identifikasi keluhan menggunakan *text mining* ini penulis telah mencoba kedua algoritma untuk *feature extraction* tersebut, dan penulis lebih memilih menggunakan algoritma *CountVectorizer* pada proses *feature extraction* karena mendapatkan hasil akurasi 90%, sedangkan ketika menggunakan *TF/IDF* hanya menghasilkan akurasi sebesar 80%. Adapun implementasi dari *feature extraction* pada sistem identifikasi keluhan menggunakan *text mining* dapat dilihat pada Gambar *4.33.*

* + - 1. ***Classifier***

*Classifier* merupakan proses penentuan kategori suatu dokumen teks sesuai dengan karakteristik dari teks tersebut. Dalam prosesnya klasifikasi teks terdiri dari 3 komponen yaitu praproses data, konstruksi pengklasifikasi dan pengkatagorian dokumen. Penelitian mengenai klasifikasi teks telah banyak dikembangkan dengan berbagai macam metode yang secara umum terbagi menjadi 3 kelompok salah satunya adalah klasifikasi teks berbasis statistic seperti *Naive Bayes, K-Nearest Neighbor, Category Center Vector, Support Vector Machine* dan *Maximum Entropy Model*.

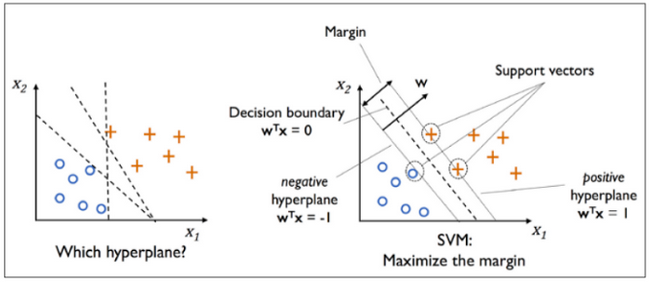
Pada penelitian ini, penulis telah mencoba 2 metode klasifikasi yaitu *Multinominal Naïve Bayes* dan *Support Vector Machine*. ***Multinomial Naive Bayes*** menghitung kemungkinan untuk **dihitung dari suatu kata / token** (variabel acak), *Multinominal* ***Naive Bayes*** menghitung kemungkinan seperti pada Gambar 4.31.



**Gambar 4.31** Rumus Multinomial Naive Bayes

Sedangkan *Support Vector Machine (SVM)* SVM memiliki konsep yang lebih matang dan lebih jelas secara matematis dibandingkan dengan dengan *linear* maupun *non linear*. *SVM* digunakan untuk mencari *hyperplane* terbaik dengan teknik-teknik klasifikasi lainnya. *SVM* juga dapat mengatasi masalah klasifikasi dan regresi memaksimalkan jarak antar kelas.

H*yperplane* adalah sebuah fungsi yang dapat digunakan untuk pemisah antar kelas. Dalam 2-D fungsi yang digunakan untuk klasifikasi antar kelas disebut sebagai *line whereas*, fungsi yang digunakan untuk klasifikasi antas kelas dalam 3-D disebut *plane similarly*, sedangan fungsi yang digunakan untuk klasifikasi di dalam ruang kelas dimensi yang lebih tinggi di sebut *hyperplane*. Adapun cara kerja *hyperplane* dapat dilihat pada Gambar 4.32.



**Gambar 4.32** Hyperlane Yang Memisahkan 2 Kelas

Adapun implementasi dari *classifier* pada sistem identifikasi keluhan menggunakan *text mining* dapat dilihat pada Gambar 4.33.



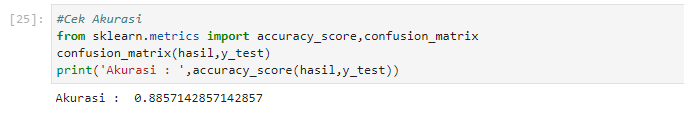
**Gambar 4.33** Feature Extraction dan Classifier

### **Pengujian Sistem**

Setelah semua rancangan telah diimplementasikan tahap berikutnya adalah pengujian. Pengujian pada sistem identifikasi keluhan menggunakan *text mining* ini terdapat pengujian metode yang berisi akurasi model, dan pengujian *Black-Box*.

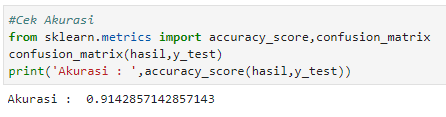
* + - 1. **Akurasi Model**

Dari semua hasil skema pengujian akurasi model, sistem identifikasi keluhan secara otomatis menggunakan *text* mining ini memiliki hasil akurasi sebesar 80% Ketika menggunakan algoritmaa *TF/IDF* untuk proses *feature extraction* dan *Multinominal Naïve Bayes* untuk proses *classifier.* Hasil akurasinya dapat dilihat pada Gambar 4.34.



**Gambar 4.34** Akurasi Model (TF/IDF & Multinominal Naïve Bayes)

Sedangkan tingkat akurasi model lebih tinggi didapatkan dengan menggunakan algoritma *Countvectorizer* untuk proses *feature extraction* dan *Support Vector Machine* untuk proses *classifier,* tingkat akurasi yang diperoleh mencapai 90%. Dari perbandingan hasil akurasi tersebut, maka penulis memutuskan untuk menggunakan kedua algoritma (*CountVectorizer & SVM)* dalam pembuatan sistem identifikasi keluhan secara otomatis menggunakan *text* mining ini*.* Adapun hasil dari akurasi model mengguakan *CountVectorizer* dan *SVM* dapat dilihat pada Gambar 4.35.



**Gambar 4.35** Akurasi Model (CountVectorizer & SVM)

* + - 1. ***Black Box* Testing**

*Black Box testing* merupakan pengujian program yang mengutamakan pengujian terhadap kebutuhan fungsi suatu program. Tujuan dari metode *Black-Box testing* yaitu untuk menemukan kesalahan fungsi pada *program.*

Pengujian dengan menggunakan metode *Black-Box testing* dilakukan hanya dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari *software* dan fungsionalitasnya tanpa mengetahui yang terjadi dalam proses detail, melainkan hanya mengetahui *input* dan *output.* Pengujian *Black Box User,* Superadmin, dan Adminpada sistem ini dapat dilihat pada Tabel 4.2, 4.3, 4.4.

**Tabel 4.2** Black-Box Testing User

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Skenario**  **Pengujian** | **Test Case** | **Hasil Yang**  **Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| ***Form* Keluhan** | | | | | |
| 1 | Mengosongkan kolom pada  *Form* lalu  langsung klik tombol kirim*.* |  | Sistem akan menolak pengisian keluhan dan antarmuka menampilkan pesan *“please fill out this field”* pada kolom  yang tidak  terisi. |  | *Valid* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Skenario**  **Pengujian** | **Test Case** | **Hasil Yang**  **Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 2 | Mengisi seluruh kolom pada *form* keluhan dan klik tombol kirim. |  | Sistem menyimpan data keluhan untuk di identifikasi. |  | *Valid* |
| **Menu Statistik** | | | | | |
| 3 | Memilih menu statistik yang ada pada navbar. |  | Sistem mengarahkan user ke halaman statistik & menampilkan persentase jumlah keluhan |  | *Valid* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Skenario**  **Pengujian** | **Test Case** | **Hasil Yang**  **Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
|  |  |  | Yang ada pada tiap-tiap kategori. |  |  |
| **Menu Rekap Tanggapan** | | | | | |
| 4. | Memilih menu rekap tanggapan yang ada pada navbar. |  | Sistem mengarahkan ke halaman rekap tanggapan yang berisi data tabel dari tindak lanjut tanggapan. |  | *Valid* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Skenario**  **Pengujian** | **Test Case** | **Hasil Yang**  **Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 5. | Klik fitur search dan menuliskan kata kunci (keluhan) yang ingin dicari |  | Sistem menampilkan hasil pencarian sesuai dengan  kata kunci yang dimasukan oleh *user*. |  | *Valid* |
| **Menu Diagram Alir Keluhan** | | | | | |
| 6. | Memilih menu diagram alir keluhan yang ada pada navbar. |  | Sistem menampilkan halaman diagram alir keluhan . |  | *Valid* |

**Tabel 4.3** Black-Box Testing Superadmin

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Skenario**  **Pengujian** | **Test Case** | **Hasil Yang**  **Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| **Menu Login Superadmin** | | | | | |
| 1 | Mengosongkan kolom pada  *form* lalu  langsung klik tombol *login.* |  | Sistem akan menolak login dan akan menampilkan *alert* “Mohon isi semua field !”. |  | *Valid* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Skenario**  **Pengujian** | **Test Case** | **Hasil Yang**  **Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 2 | Mengisi sebagian kolom pada *form login*. |  | Sistem akan menolak login dan akan menampilkan *alert* “Mohon isi semua field !”. |  | *Valid* |
| 3 | Mengisi semua *form login.* |  | *Login validation* berhasil dan akan menampilkan *home* superadmin |  | *Valid* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Skenario**  **Pengujian** | **Test Case** | **Hasil Yang**  **Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 4 | Tidak mengisi seluruh *form* di *screen* buatkan admin dan langsung mengklik *button create* |  | Sistem akan menolak *login* dan akan menampilkan *alert* “Mohon isi semua *field* !”. |  | *Valid* |
| 5 | Mengisi sebagian *form* di *screen* buatkan admin dan langsung mengklik *button create.* |  | Sistem akan menolak *login* dan akan menampilkan *alert* “Mohon isi semua *field* !”. |  | *Valid* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Skenario**  **Pengujian** | **Test Case** | **Hasil Yang**  **Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 6 | Mengisi seluruh *field form* di *screen* buatkan admin. |  | Sistem membuat admin baru dan menampilkan *alert* "Berhasil membuat admin" |  | *Valid* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Skenario**  **Pengujian** | **Test Case** | **Hasil Yang**  **Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 7 | Mengklik *button* *delete* di *screen* kelola akun admin. |  | Akun admin terhapus dan menampilkan *alert* “*Action* berhasil dijalankan”. |  | *Valid* |
| 8 | Mengisi *field* di *screen* kelola akun admin ketika sudah mengklik *button* *update* kemudian *button save* |  | Data akun admin sudah diperbaharui dan menampilkan *alert* “*Action* Berhasil Dijalankan” |  | *Valid* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Skenario**  **Pengujian** | **Test Case** | **Hasil Yang**  **Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 9 | Mengklik *Button Download* di *screen* list keluhan superadmin |  | File pdf seluruh keluhan berhasil di download |  | *Valid* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Skenario**  **Pengujian** | **Test Case** | **Hasil Yang**  **Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 10 | Klik *button filter by* dan memilih bulan dan tahun dari keluhan yang akan ditampilkan |  | Menampilkan *list* keluhan berdasarkan bulan dan tahun yang dipilih |  | *Valid* |
| 11 | Mengklik menu utama kategori keluhan |  | Menampilkan list keluhan berdasarkan kategori |  | *Valid* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Skenario**  **Pengujian** | **Test Case** | **Hasil Yang**  **Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 12 | Klik button kelola laporan keluhan*.* |  | Sistem menampilkan list laporan keluhan |  | *Valid* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Skenario**  **Pengujian** | **Test Case** | **Hasil Yang**  **Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 13 | Klik *button approve* di *screen detail* laporan |  | Sistem akan mengubah status keluhan menjadi *reported* dan menampilkan *screen home* |  | *Valid* |
| 14 | Klik *button decline* di *screen detail* laporan |  | Sistem akan mengubah status keluhan menjadi belum ditanggapi dan menampilkan *screen home* |  | *Valid* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Skenario**  **Pengujian** | **Test Case** | **Hasil Yang**  **Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 15 | Klik *button* kelola akun admin |  | Menampilkan list akun admin |  | *Valid* |
| 16 | Mengklik menu logout |  | Menampilkan *screen login* |  | *Valid* |

**Tabel 4.4** Black-Box Testing Admin

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Skenario**  **Pengujian** | **Test Case** | **Hasil Yang**  **Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| **Menu Login Admin** | | | | | |
| 1 | Mengosongkan kolom pada  *form* lalu  langsung klik tombol *login..* |  | Sistem akan menolak login dan akan menampilkan *alert* “Mohon isi semua field !”. |  | *Valid* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Skenario**  **Pengujian** | **Test Case** | **Hasil Yang**  **Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 2 | Mengisi sebagian kolom pada *form login*. |  | Sistem akan menolak login dan akan menampilkan *alert* “Mohon isi semua field !”. |  | *Valid* |
| 3 | Mengisi semua *form login.* |  | *Login validation* berhasil dan akan menampilkan *home* superadmin |  | *Valid* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Skenario**  **Pengujian** | **Test Case** | **Hasil Yang**  **Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 4 | Mengklik *menu* utama kategori keluhan |  | Menampilkan *list* keluhan berdasarkan kategori |  | *Valid* |
| 5 | Mengklik button kirim tanggapan tanpa mengisi form tanggapan |  | Menampilkan *alert* “*Field* Tanggapan tidak boleh kosong !” |  | *Valid* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Skenario**  **Pengujian** | **Test Case** | **Hasil Yang**  **Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 6 | Mengklik button kirim tanggapan dengan mengisi form tanggapan |  | Sistem menyimpan tanggapan tersebut dan menampilkan alert “Action Berhasil Dilakukan”. |  | *Valid* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Skenario**  **Pengujian** | **Test Case** | **Hasil Yang**  **Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 6 | Klik *button* laporkan |  | Sistem mengubah status keluhan menjadi pending dan menampilkan alert “Action Berhasil Dilakukan” |  | *Valid* |

# BAB V PENUTUP

## Kesimpulan

Kesimpulan yang didapat setelah mengumpulkan, mengelola, menganalisa, mengimplementasi serta melakukan pengujian sistem identifikasi keluhan secara otomatis menggunakan *text mining* ini adalah sebagai berikut :

* + 1. Keluhan dapat dengan mudah didistribusikan pada unit kerja terkait dengan memanfaatkan *text mining* yang mampu mengidentifikasi apakah suatu keluhan ditujukan untuk suatu unit kerja atau bukan.
    2. Berdasarkan hasil pengujian akurasi model yang memiliki persentase sebesar 90%, disimpulkan bahwa algoritma yang paling cocok dalam pembuatan sistem ini adalah *CountVectorizer* untuk proses *Feature Extraction* dan *Support Vector Machine (SVM)* untuk *proses Classifier*.
    3. Hasil pengujian *black-box* sistem identifikasi keluhan secara otomatis menggunakan *text mining* ini berjalan baik, dan telah sesuai dengan kebutuhan sistem.

## Saran

Pada sistem identifikasi keluhan secara otomatis menggunakan *text mining* inimasih terdapat beberapa kekurangan, oleh karena itu penulis menyarankan untuk pengembang selanjutnya agar dapat mempertimbangkan saran yang diberikan agar sistem identifikasi keluhan secara otomatis menggunakan *text mining* ini lebih kompleks dan semakin baik, saran-saran tersebut antara lain sebagai berikut:

1. Memperbaiki tampilan user, baik pengguna (mahasiswa), admin maupun superadmin.
2. Dapat mengklasifikasi data dukung keluhan yang diinputkan oleh user.
3. Dapat dikembangkan ke beberapa metode baru lainnya.

# DAFTAR PUSTAKA

Nugroho, Adi. 2009. *Rekayasa Perangkat Lunak Menggunakan UML & Java*. Yogyakarta: Andi Offset.

Pressman, R.S. 2010. *Software Engineering: A Practitioner’s Approach, 7th Edition.* McGraw-Hill, New York.

Nomleni, Petrix. 2015. Sentiment Analysis Using Support Vector Machine(SVM). Surabaya: Institut Teknologi Sepuluh Nopember.

Wicaksono, T., & Mubtada'i, N. R. (2010). Text Mining untuk Pencarian

Dokumen Bahasa Inggris menggunakan Suffix Tree Clustering. eepis final project.

Dewi, Nesya Rona. 2018. Model *Text Mining* Untuk Identifikasi Keluhan

Pelanggan Produk Perusahaan Perangkat Lunak. Yogyakarta:Universitas Islam Indonesia.

Silitonga, Yosua Riadi. 2013. Analisis Penerapan *Data Mining* Untuk Mendeteksi

Berita Palsu Pada *Social Media* Dengan Mneggunakan SCIKIT LEARN. Jakarta: Universitas Esa Unggul.

Nalendro, Galih. 2019. Sistem Informasi Pengaduan Layanan Universitas.

Yogyakarta:Universitas Islam Indonesia