BAB I

PENDAHULUAN

A. Latar Belakang

Bahasa Indonesia adalah sebagai bahasa resmi dalam dunia pendidikan. Sesuai dengan pasal 36 undang-undang dasar 1945 menyatakan bahwa bahasa Indonesia sebagai bahasa negara, mempunyai fungsi sebagai bahasa resmi negara, bahasa pengantar di lembaga-lembaga pendidikan, alat penghubungan dan pembangunan, alat bagi kepentingan menjalankan roda pemerintahan, seni serta teknologi modern. Sayangnya, tidak semua pemakai bahasa menggunakan bahasa Indonesia yang baik dan benar, sesuai dengan konteks pemakaiannya.

Sebagai pemakai bahasa Indonesia, kita wajib mematuhi aturan baku berbahasa, terutama saat kita berada di forum resmi atau saat menyajikan tulisan ilmiah. Kenyataannya, tidak semua penutur bahasa Indonesia mampu berbahasa Indonesia sesuai dengan aturan yang berlaku. Sering kali kita juga kebingungan apakah bahasa yang kita pakai termasuk bahasa baku atau tidak, atau apakah bahasa yang digunakan sudah menggunakan ejaan yang disempurnakan atau belum?. Hal ini tersebut bisa di maklumi, sebab bahasa Indonesia yang kita pakai sekarang pun diserap dari berbagai bahasa , baik bahasa daerah, bahasa asing dan bahasa-bahasa lainnya. Namun demikian, sebagai bangsa yang punya jatidiri, maka perlu adanya pembakuan istilah. Sebab bahasa tidak semata-mata sebagai

alat untuk berkomunikasi, tetapi sudah menjadi identitas suatu bangsa . hal ini sejalan dengan semboyan bahwa “bahasa menunjukkan bangsa.

Menulis merupakan salah satu keterampilan dan kebiasaan manusia dalam kehidupan sehari-hari bahkan dalam dunia pendidikan. Menulis juga salah satu media komunikasi yang di sampaikan secara tidak langsung, baik itu dalam bentuk berita, jurnal, laporan dan tulisan lainnya. Setiap kata dalam satu kalimat memiliki makna yang penting dan sangat berkaitan dengan kalimat lainnya. Tetapi dalam pengetikan ada yang sesuai dengan EYD (Ejaan Yang Dibenarkan) ada juga yang tidak sesuai EYD, dalam bahasa lain bisa disebut bahasa yang baku dan tidak baku (Susilawati, 2019). Seseorang yang mengetahui suatu kata tetapi tidak mampu merangkainya berarti mengetahui makna kata tersebut, dan hal itu bisa menyebabkan kesalahan dalam penulisan dalam kalimat. Pengaruh bahasa asing dan bahasa daerah juga menjadi penyebab munculnya kesalahan dalam penyusunan kalimat.

Kemampuan setiap manusia dalam memeriksa kembali kesalahan yang dibuat berbeda-beda dan membutuhkan ketelitian yang sangat tinggi, maka ditawarkan sebuah aplikasi yang akan mendeteksi kesalahan pengetikan khususnya pada kata nonbaku yang masi banyak digunakan penulis dalam proses pembuatan karya tulis (Hartina & Agustin, 2020).

Berdasarkan uraian di atas, sehingga dari permasalahan tersebut penulis telah membuat aplikasi yang dapat menjadi solusi dari permasalahan yang ada, maka penulis mengangkat judul “Aplikasi Pendeteksi Kata Baku dan Tidak Baku”.

B. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka diperoleh beberapa masalah sebagai berikut:

1. Bagaimana membuat aplikasi pendeteksi kata baku dan tidak baku ?
2. Bagaimana proses dalam mendeteksi kata baku dan tidak baku ?

C. Tujuan Penelitian

Berdasarkan latar belakakang tersebut maka tujuan dilakukan penelitian adalah sebagai berikut :

1. Aplikasi ini di buat untuk mengenali perbedaan antara kata baku dan tidak baku bahasa Indonesia.
2. Untuk mengetahui kata baku dan tidak baku di dalam sebuah kalimat.

D. Batasan Masalah

Penelitian ini memiliki beberapa batasan, sehingga lebih terfokus dengan batasan-batasan yang di buat untuk menghindari adanya pelebaran pokok permasalahan. Adapun beberapa batasan permasalahan sebagai berikut :

1. Kata yang dideteksi adalah kata yang memiliki kata tidak baku yang di kutip dari buku “pedoman kata baku dan tidak baku dilengkapi ejaan yang disempurnakan (EYD)” oleh Ernawati Waridah, S.S.
2. Kata yang di input berjumlah 1.000 kata dan tidak memiliki imbuhan awalan maupun akhiran.
3. Kata yang di deteksi sesuai dengan pencocokan kata yang ada dalam *database*

4. Jika terdapat kata inputan oleh pengguna yang tidak terdapat didalam *database* aplikasi maka akan menampilkan hasil bahwa kata tersebut tidak terdapat di *database* aplikasi
5. Kata terdeteksi yang tidak baku akan diberi tulisan berwarna merah dan menampilkan hasil bahwa kata tersebut adalah kata tidak baku kemudian tampil bentuk kata bakunya serta kelompok katanya
6. Jika kata terdeteksi merupakan kata baku maka akan menampilkan hasil bahwa kata tersebut adalah kata baku serta kelompok katanya
7. Program Aplikasi ini dirancang menggunakan bahasa pemrograman *JavaScript*

E. Manfaat Penelitian

Berikut beberapa manfaat yang diperoleh dalam pembuatan tugas akhir ini adalah sebagai berikut :

1. Untuk Penulis
   a) Mengimplementasikan ilmu yang di peroleh selama dibangku kuliah terutama dalam bidang ilmu yang sesuai dengan tugas akhir ini.
   b) Menambah pengetahuan mengenai kata baku dan tidak baku bahasa Indonesia
2. Untuk Pembaca

   Sebagai media pembelajaran untuk mengetahui mana kata baku dan kata tidak baku bahasa Indonesia.
3. Untuk Akademik

Memberikan sumber acuan (referensi) dan bahan kajian untuk pengembangan aplikasi serta menambah *literature* perpustakaan dan bahan pertimbangan yang berhubungan dengan tugas akhir.

F. Sistematika Penulisan

Sistematika penulisan merupakan penjabaran secara deskriptif tentang hal-hal yang akan ditulis, yang secara garis besar terdiri dari bagian awal, bagian isi dan bagian akhir. Adapun sistematika penulisan dari tugas akhir ini yaitu sebagai berikut:

BAB I PENDAHULUAN

Pada Bab ini menjelaskan tentang latar belakang dari masalah dalam penelitian ini, rumusan masalah, tujuan penelitian, batasan dari masalah, manfaat penelitian dan sistematika penulisan pada tugas akhir ini.

BAB II TINJAUAN PUSTAKA

Pada Bab II berisi tentang penelitian terdahulu, dan juga uraian-uraian teori yang akan digunakan sebagai referensi atau acuan dalam desain dan implementasi mengenai Aplikasi pendeteksi kata baku dan tidak baku .

BAB III METODE PENELITIAN

Pada Bab III berisi tentang waktu penelitian, jenis penelitian, teknik pengumpulan data, alat yang digunakan dalam penelitian, tahap penelitian, metode pengujian, dan desain sistem pada Aplikasi Pendeteksi Kata Baku dan Tidak Baku.

BAB IV HASIL DAN PEMBAHASAN

Pada Bab IV dibahas hasil penelitian yang akan memberikan pembahasan seputar yang terdapat pada Rumusan Masalah dan Tujuan Penelitian pada Aplikasi Pendeteksi Kata Baku dan Tidak Baku .

BAB V PENUTUP

Pada Bab V mencakup kesimpulan dari masing-masing sub-sub bab yang ada serta memberi saran agar dapat bermanfaat untuk pengembangan *system* ini selanjutnya.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Kata Baku dan Tidak Baku

Bahasa Indonesia terdiri dari dua jenis kata, yaitu kata baku dan tidak baku. Kata baku biasanya digunakan pada suatu yang bersifat formal seperti penulisan jurnal, laporan penelitian, dan lain sebagainya. Sedangkan kata tidak baku digunakan pada suasana tidak formal, misalnya berbicara santai dengan teman dekat, *chatting*, komentar atau posting dimedia *social*, dan yang lainnya (Qulub et al., 2020).

2. Android

Android adalah suatu sistem operasi pada *smartphone* atau *tablet* yang mempunyai banyak fitur didalamnya untuk mempermudah kehidupan manusia dan sampai sekarang terus berkembang semakin canggih (Pradana, 2019).

3. JavaScript

Javascript adalah bahasa pemrograman untuk sisi *client* atau *client side*. *Javascipt* adalah bahasa pemrograman yang mendekati bahasa manusia atau bisa dikatakan bahasa tingkat tinggi, maka dari itu *javascript* mudah di pelajari. *Javasript* sendiri tujuannya dibuat untuk memperkaya fitur pada *website* agar lebih dinamis, seperti untuk menampilkan dan menghilangkan objek-objek pada

website kemudian dengan fungsi *javascript* dapat memanggil kembali objek yang di hilangkan tersebut (Marlina & Masnur, 2021).

4. Node.js



Gambar 2.1 Logo *Node.js*

Node.js adalah perangkat lunak yang didesain untuk mengembangkan aplikasi berbasis web dan ditulis dalam sintaks bahasa pemrograman *JavaScript*. Bila selama ini kita mengenal *JavaScript* sebagai bahasa pemrograman yang berjalan di sisi *client/browser* saja, maka *Node.js* ada untuk melengkapi peran *JavaScript* sehingga bisa juga berlaku sebagai bahasa pemrograman yang berjalan di sisi *server*, seperti halnya PHP, *Ruby*, *Perl* dan sebagainya. *Node.js* dapat berjalan di sistem operasi *Windows*, *Mac OS X* dan *Linux* tanpa perlu ada perubahan kode program.

Berbeda dengan bahasa pemrograman sisi *server* pada umumnya yang bersifat sinkronis atau *blocking*, *Node.js* bersifat asinkronis atau *non-blocking* sebagaimana halnya *JavaScript* bekerja. *Node.js* berjalan dengan basis *event* (*event-driven*). Maksud dari *blocking* secara sederhana adalah bahwa suatu kode program akan dijalankan hingga selesai, baru kemudian beralih ke kode program selanjutnya (Nasution, 2019).

5. React native

React Native adalah salah satu *framework* untuk membuat aplikasi *mobile* dengan menggunakan kode *Javascript*. *Framework React Native* memiliki seperangkat komponen bagi *platform* iOS dan android untuk membangun aplikasi *mobile* dengan tampilan yang benar-benar seperti *native*. Dengan menggunakan *Framework React Native*, kita dapat *merender User Interface* untuk *platform* iOS dan Android. *React Native* ini adalah *framework open source*, yang bisa kompatibel dengan *platform* lain seperti *Windows* atau *tvOS* dalam waktu dekat (Wiguna et al., 2018).

Dengan *Framework React Native*, kita tidak membangun "aplikasi *mobile web*", "aplikasi HTML5", atau "aplikasi *hybrid*". Kita membangun aplikasi *mobile native* yang tidak dapat dibedakan dari aplikasi yang dibuat menggunakan *Objective*-C atau Java. *React Native* menggunakan blok bangunan *user interface* dasar yang sama seperti aplikasi iOS dan *Android* biasa. Sehingga kita tinggal memasang blok-blok tersebut menggunakan *Javascript* (Wiguna et al., 2018).

6. SQLite

SQLite adalah sebuah *open source database* yang telah ada cukup lama, cukup stabil, dan sangat terkenal pada perangkat kecil, termasuk *Android. Android* menyediakan *database* relasional yang ringan untuk setiap aplikasi menggunakan *SQLite*. Aplikasi dapat mengambil keuntungan dari itu untuk mengatur *relational database engine* untuk menyimpan data secara aman dan efisien. Untuk *Android, SQLite* dijadikan satu di dalam *Android runtime*, sehingga setiap aplikasi *Android* dapat membuat basis data *SQLite*. Karena *SQLite* cukup mudah untuk digunakan

orang-orang dengan pengalaman lain yang berbasis *database*. Terdapat beberapa alasan mengapa *SQLite* sangat cocok untuk pengembangan aplikasi *Android*, yaitu:

a.) *Database* dengan konfigurasi nol. Artinya tidak ada konfigurasi *database* untuk para *developer*. Ini membuatnya *relative* mudah digunakan.

b.) Tidak memiliki *server*

c.) Tidak ada proses *database SQLite* yang berjalan. Pada dasarnya satu set *libraries* menyediakan fungsionalitas *database*.

d.) *Single-file database*. Ini membuat keamanan *database* secara langsung.

e.) *Open source*. Hal ini membuat *developer* mudah dalam pengembangan aplikasi (Tandra et al., 2017).

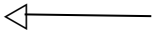
7. *Unified Modelling Language* (UML)

Unified Modelling Language (UML) adalah suatu bentuk metode pemodelan visual yang digunakan dalam analisa dan perancangan sistem berorientasi objek. Fungsi UML sendiri adalah dapat menjadi *blue print* yang lengkap dan detail yang siap untuk diimplementasikan ke dalam koding oleh programmer. Dengan diagram UML, programmer bisa memahami *behavior* (kelakuan) dari *objek* dari segi relasi dan interaksi dengan mudah dan cepat karena telah divisualisasikan dalam bentuk diagram baku (Anak Toraja, 2019). Berikut ini merupakan diagram UML :

a.) Use Case Diagram

Use case diagram adalah salah satu jenis UML untuk menggambarkan hubungan antara *system* dan *actor* yang dapat menunjukkan tipe interaksi yang terjadi dalam *system*.

Tabel 2.1 Simbol *Use case Diagram*

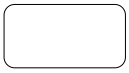
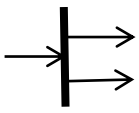
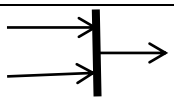
Nama	Simbol	Keterangan
Actor		Menspesifikasi himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
Dependency		Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
Generalisasi		Apa yang menghubungkan antara objek satu dengan objek lainnya
Include		Menspesifikasikan bahwa *use case* sumber secara eksplisit.
Extend		Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
Association		Apa yang menghubungkan antara objek satu dengan objek lainnya
System		Menspesifikasikan paket yang menampilkan sisten secara terbatas
Use case		Deskripsi dari urutan aksi-aksi yang ditampilkan system yang menghasilkan suatu hasil yang teukur bagi suatu *actor*

Nama	Simbol	Keterangan
Collaboration		Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
Note		Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya koputasi.

b.) Activity Diagram

Activity diagram memiliki cara kerja persis seperti *flowchart* (diagram alir). Tujuan dari *activity diagram* adalah menggambarkan proses yang terjadi dala suatu sistem secara teratur, mulai dari awal, keputusan yang bisa terjadi, sampai aktivitas berakhir.

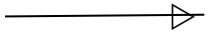
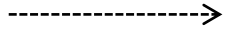
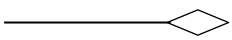
Tabel 2.2 Simbol *Activity Diagram*

Simbol	Keterangan
	*Start Point*
	*End Point*
	*Activities*
	*Fork* (Percabangan)
	*Join* (Penggabungan)
	*Decision*

c. *) Class Diagram*

Class diagram adalah diagram yang menunjukkan kelas-kelas dan hubungan antar kelas dalam suatu *sistem*.

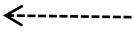
Tabel 2.3 Simbol *Class diagram*

Nama dan *symbol*	Keterangan
Asosiasi / *association*	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*
Asosiasi berarah / *directed association* 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan / *dependency* 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi / *aggregation* 	Relasi antar kelas dengan makna semua-bagian (*whole-part*)

c.) Sequence Diagram

Sequence diagram adalah diagram yang menggambarkan interaksi abyek bersadarkan urutan, baik itu urutan tahapan atau urutan waktu. Objek-objek dalam *Sequence Diagram* diurutkan dari kiri ke kanan. Perlu diketahui, dalam *Sequence Diagram*, dimensi *vertical* digunakan untuk merepresentasikan waktu dan dimensi *horizontal* untuk mererepsentasikan objek-objek yang terkait.

Tabel 2.4 Simbol *Sequence Diagram*

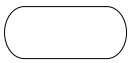
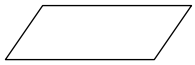
Nama dan *symbol*	Keterangan
	*Actor* adalah representasi dari suatu entitas yang letaknya berada di luar *system* dan bisa berinteraksi dengan *system*. *Actor* di sini bisa berupa manusia, *hardware* atau pun *system* lainnya.
Lifeline	Untuk mengeksekusi obyek selama *sequence*
	Tepi dari sistem, seperti *user interface* atau alat yang dibutuhkan untuk berinteraksi dengan sistem yang lainnya.
	Elemen jaringan komputer yang lainnya adalah untuk mengatur aliran informasi dari sebuah skanario.
	Atau entitas adalah eleme yang bertanggung jawab dalam menyimpan data dan informasi
Message to self 	Menunjukkan hubungan antara obyek terhadap obyek itu sendiri berdasarkan urutan kejadian
Message Entry 	Menunjukkan hubungan antara objek berdasarkan urutan kejadian

(Anak Toraja, 2019)

8. Flowchart

Flowchart (Diagram Alir) atau di sebut *Flowchart* merupakan bagan (*Chart*) yang mengarahkan alir (*flow*) di dalam prosedur atau program sistem secara logika. *Flowchart* adalah cara untuk menjelaskan tahap-tahap pemecahan masalah dengan merepresentasikan symbol-simbol tertentu yang mudah dipahami, mudah digunakan dan standar (Syamsiah, 2019). Bagan alir program memiliki simbol-simbol sebagai berikut :

Tabel 2.5 Simbol *Flowchart*

Gambar	Keterangan
	Titik Terminal, menggambarkan awal dan akhir
	Deklarasi, menggambarkan pendefinisian komponen atau variable yang akan digunakan.
	IO, menggambarkan masukan dan keluaran atau *input/output* sebuah data.
	Proses, menggambarkan sebuah proses

B. Tinjauan Hasil Penelitian Terdahulu

Dalam membangun aplikasi pada penelitian ini, penulis menggunakan beberapa referensi terdahulu yang telah dilakukan. Adapun penelitian terdahulu yang di maksud yaitu :

Penelitian mengenai, "**Implementasi Google Speech Api pada Aplikasi Pengenalan Kata Baku untuk Siswa Tingkat Sekolah Dasar Berbasis Android**". Dalam mata pelajaran bahasa Indonesai kita juga di perkenalkan tentang kata baku, kata baku merupakan bahasa yang sudah sesuai dengan kaidah dan ejaan yang benar pada kamus besar bahasa Indonesia (KBBI). Namun dengan kurangnya minat dalam pembelajaran bahasa Indonesia, bahasa baku mulai dikhawatirkan ketika jarangnya penggunaan bahasa resmi di kehidupan sehari-hari. Padahal, bahasa baku sangat penting dalam kedudukan bangsa. (ABIMANTRA, 2021)

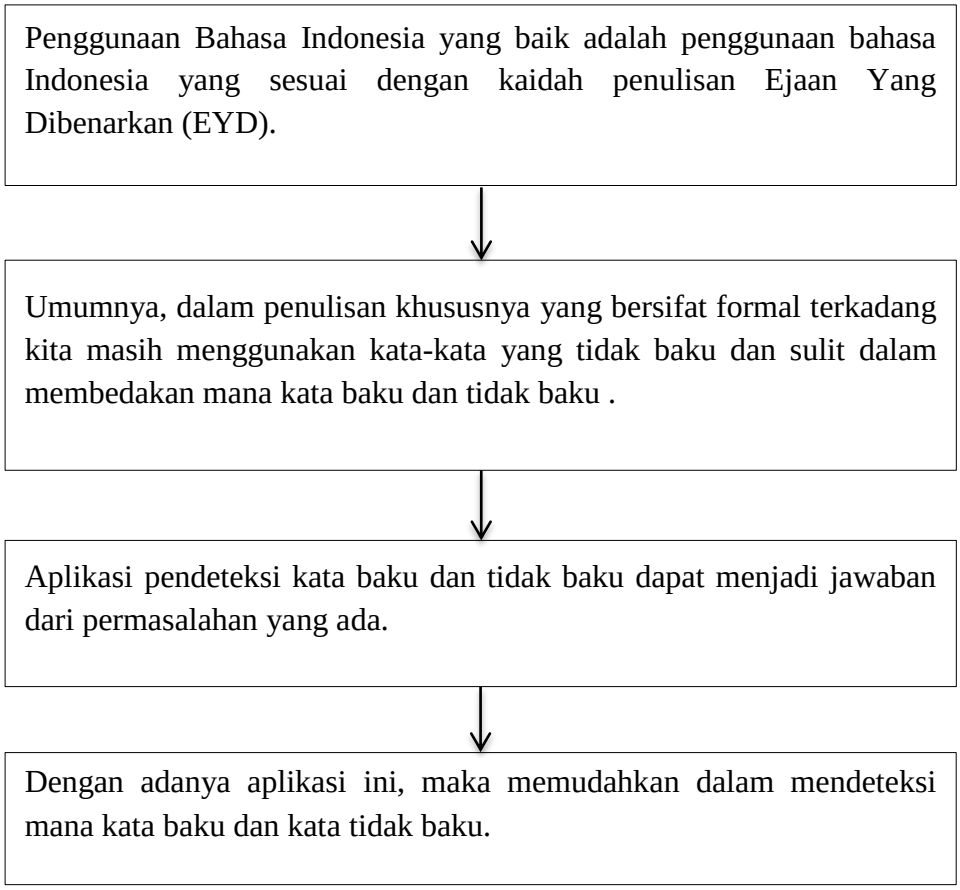
Penelitian selanjutnya "**Aplikasi Pendeteksi Kesalahan Ejaan bahasa Indonesia pada Karya Ilmiah Bidang Ilmu Komputer menggunakan KEBI 1.0 Checher**". Karya ilmiah berbahasa Indonesia di bidang ilmu computer cukup banyak ditemui pengguna kombinasi kata dalam bahasa Indonesia maupun kata dalam bahasa Inggris yang di gunakan untuk menunjukkan suatu nama perangkat, algoritma, metode dan istilah lainnya. Oleh karena itu , penelitian ini berinisiatif untuk mengembangkan aplikasi berbasis web untuk mendeteksi kesalahan ejaan kata tidak baku dan kesalahan pengetikan kata (typo) yang di sebut KEBI 1.0 Checker. (Fahrudin et al., 2021)

Penelitian selanjutnya“ **Pendeteksi Kesalahan Pengetikan Kata Non Baku pada Karya Tulis Menggunakan Metode N-Gram**”. Metode N-gram dapat menghitung persentasi bobot untuk menentukan apakah kata Berbahasa Indonesia akan dideteksi atau tidak. Aplikasi yang dirancang mampu untuk mendeteksi sekitar 70% dari 10% kata yang dideteksi dan 80% keberhasilah dari pengujian karya ilmiah oleh system. (Hartina & Agustin, 2020)

Dari penjelasan ketiga penelitian terkait yang bersifat *online*, penulis telah membuat aplikasi pendeteksi kata baku dan tidak baku yang bersifat *offline* sehingga dapat di gunakan tanpa harus terhubung jaringan internet dan berbasis Android.

C. Kerangka Pikir

Berikut merupakan sebuah garis besar atau alur logika didalam penelitian ini, atau juga gambaran yang berupa konsep yang didalamnya menjelaskan mengenai suatu hubungan antara variabel yang satu dengan variabel lainnya :



Gambar 2. 2 Kerangka pikir

BAB III

METODE PENELITIAN

A. Lokasi dan Waktu

Lokasi penelitian bertempat di laboratorium Teknik Informatika, Universitas Muhammadiyah Parepare, Jl.Jendral Ahmad Yani KM.6 Kota Parepare. Adapun waktu yang digunakan dalam penelitian setelah judul diterima untuk tetap di lanjutkan adalah 6 bulan.

B. Jenis Penelitian

Metode penelitian yang digunakan untuk membangun aplikasi pendeteksian kata baku dan tidak baku adalah menggunakan penelitian analisis *desktriptif* dengan menggunakan *surve.*.

C. Teknik Pengumpulan Data

Berdasarkan sumbernya, data dapat dibedakan menjadi dua, yaitu data primer dan data sekunder, berikut data-data yang di maksud :

1. Data Primer

Dalam memperoleh data-data yang dibutuhkan, maka penulis melakukan pengumpulan data berupa kata baku dan tidak baku yang bersumber dari buku karangan Ernawati Waridah (2014) dengan judul buku “Pedoman Kata Baku dan Tidak Baku” yang akan di inputkan kedalam *database*.

2. Data Sekunder

Dalam penelitian ini yang menjadi sumber data sekunder adalah mengumpulkan data-data maupun informasi yang terkait dengan cara mengumpulkan jurnal, buku-buku, artikel dan bacaan-bacaan yang diperlukan untuk membangun aplikasi pendeteksi kata baku dan tidak baku.

D. Alat dan Bahan Penelitian

Untuk melakukan proses penelitian, maka yang diperlukan adalah alat dan bahan yang digunakan pada proses penelitian, guna untuk mendukung kegiatan penelitian tersebut. Adapun alat dan bahan yaitu sebagai berikut :

1. Alat

a. Kebutuhan *Hardware* (Perangkat keras)

Tabel 3.6 Kebutuhan Perangkat keras

Jenis	Spesifikasi
Laptop	Asus X441M
Processor	Intel Celeron N4000, up to 2.6GHz
Memori	4GB
Hard Disk Drive	1TB
Layar	14 *inch*
Sistem Operasi	Windows 10 PRO 64-bit

b. Kebutuhan Perangkat Lunak (*Software*)

Spesifikasi minimum *software* pengguna aplikasi pendeteksi kata baku dan tidak baku yaitu sebagai berikut :

Tabel 3.7 Kebutuhan perangkat lunak (*Software*)

Jenis	Keterangan
Sistem Operasi	windows 10 (64 bit)
Software	*Node.js, Visual Studio Code*
Bahasa Pemrograman	*JavaScript*
Database	*SQLite*

Jenis	Keterangan
Rancangan Program	*Start* UML

2. Bahan

Bahan yang digunakan dalam penelitian ini adalah Buku Pedoman kata baku dan tidak baku dilengkapi ejaan yang disempurnakan (EYD) oleh Ernawati Waridah, S.S Sebagai acuan di *database* nya.

E. Metode Pengujian

Metode pengujian pada perangkat lunak pada sistem ini menggunakan beberapa pengujian :

1. Blackbox Testing

Blackbox Testing adalah tahap yang di gunakan untuk menguji kelancaran program yang telah di buat. Pengujian ini penting agar tidak terjadi kesalahan alur program yang telah dibuat.

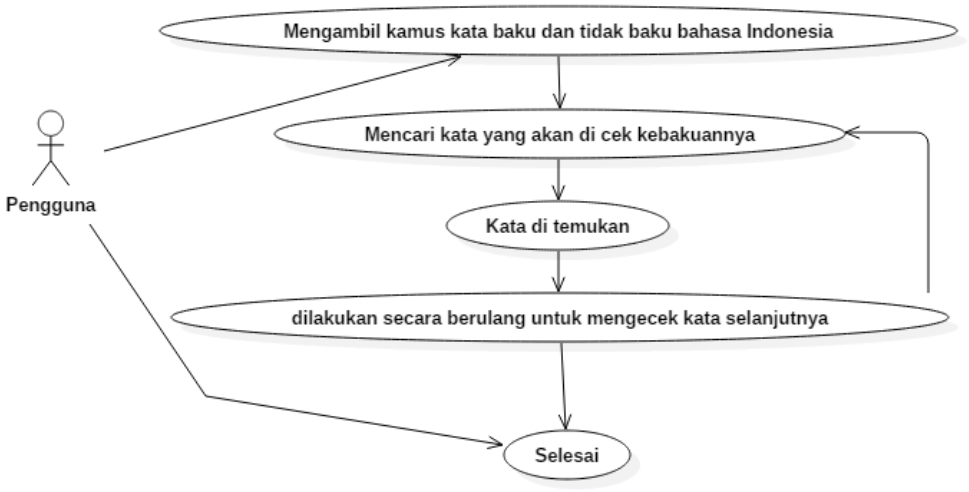
2. Whitebox

Whitebox Testing adalah test bagian dalam dari sebuah *software development*, pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara *procedural* untuk membagi pengujian ke dalam beberapa kasus pengujian. Secara sekilas dapat diambil kesimpulan bahwa *whitebox testing* merupakan petunjuk untuk mendapatkan program yang benar secara 100%.

F. Desain Sistem

1. Sistem Yang Berjalan

Sistem yang berjalan pada Aplikasi Pendeteksi Kata Baku dan Tidak Baku adalah sebagai berikut :

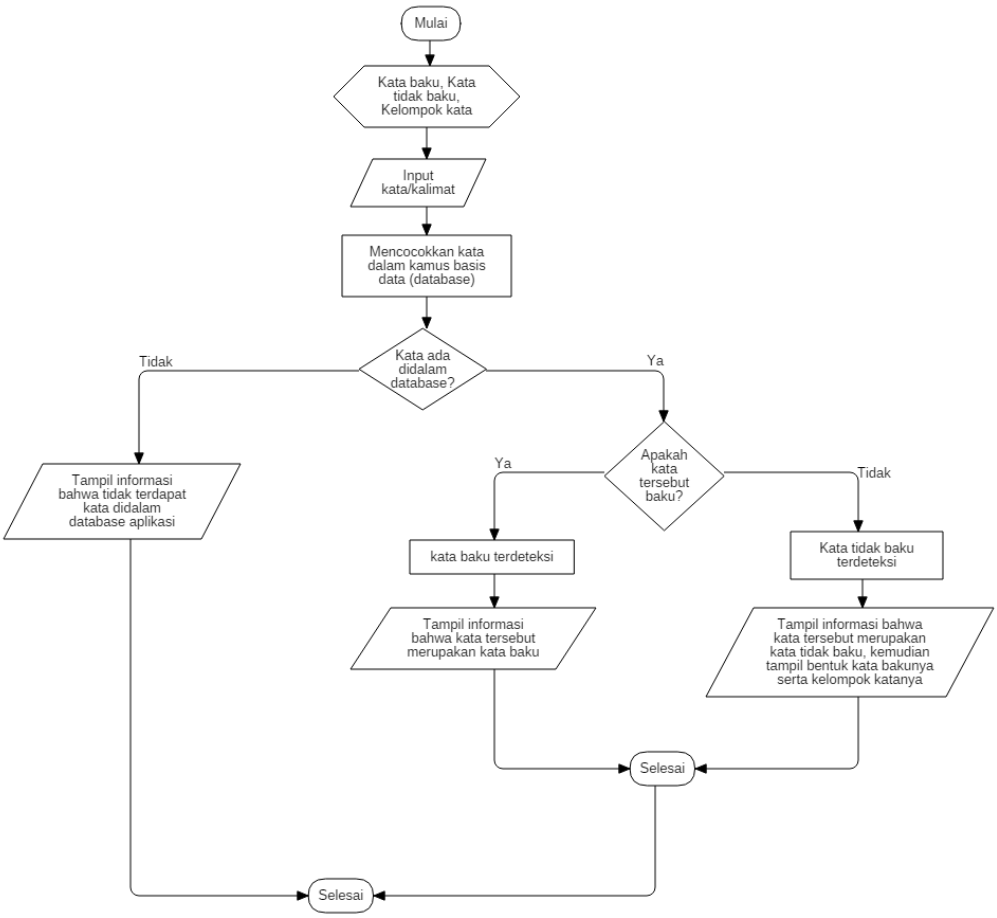


Gambar 3.3 Sistem yang berjalan

Penggunaan kamus bahasa Indonesia sebagai media untuk mengetahui suatu kata, apakah kata tersebut baku atau tidak, dianggap kurang efektif sebab masi dalam pengecekan secara manual, dan membutuhkan waktu yang lama, terlebih jika memiliki kata yang cukup banyak untuk di cek kebakuannya.

2. Sistem Yang Diusulkan

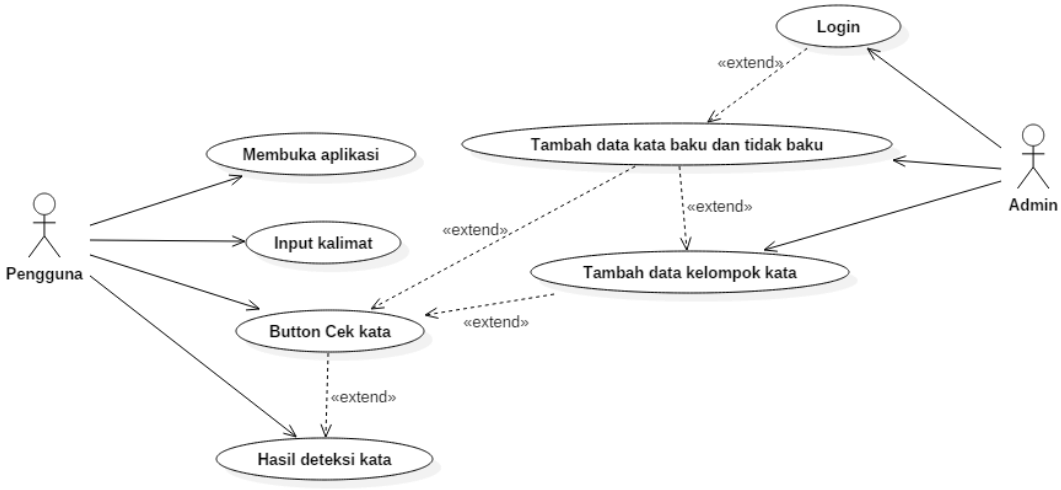
Sistem yang di usulkan pada Aplikasi Pendeteksi Kata Baku dan Tidak Baku adalah sebagai berikut :



Gambar 3. 4 *Flowcart* sistem yang diusulkan

3. *Use case diagram*

Use case diagram merupakan diagram yang menggambarkan hubungan antara aktor dengan *system*. *Use case diagram* bisa mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat.



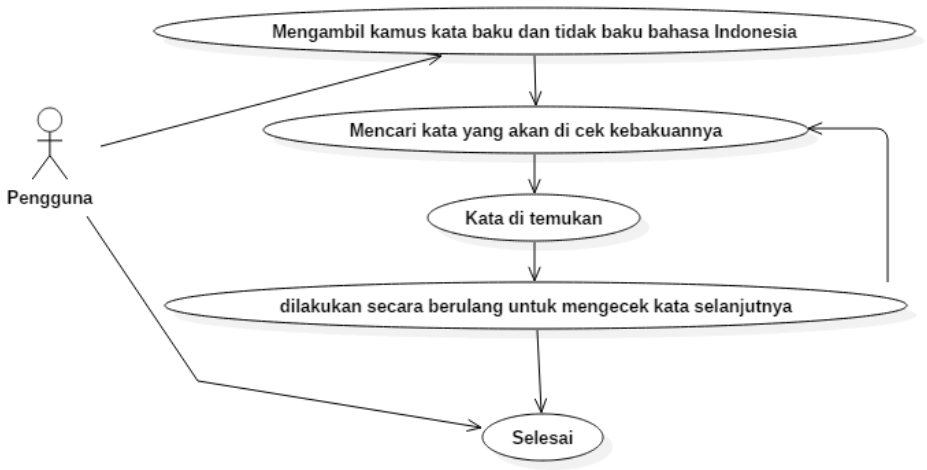
Gambar 3.5 *Use case diagram*

BAB IV

HASIL DAN PEMBAHASAN

A. Sistem Yang Berjalan

Sistem yang berjalan pada Aplikasi Pendeteksi Kata Baku dan Tidak Baku adalah sebagai berikut :

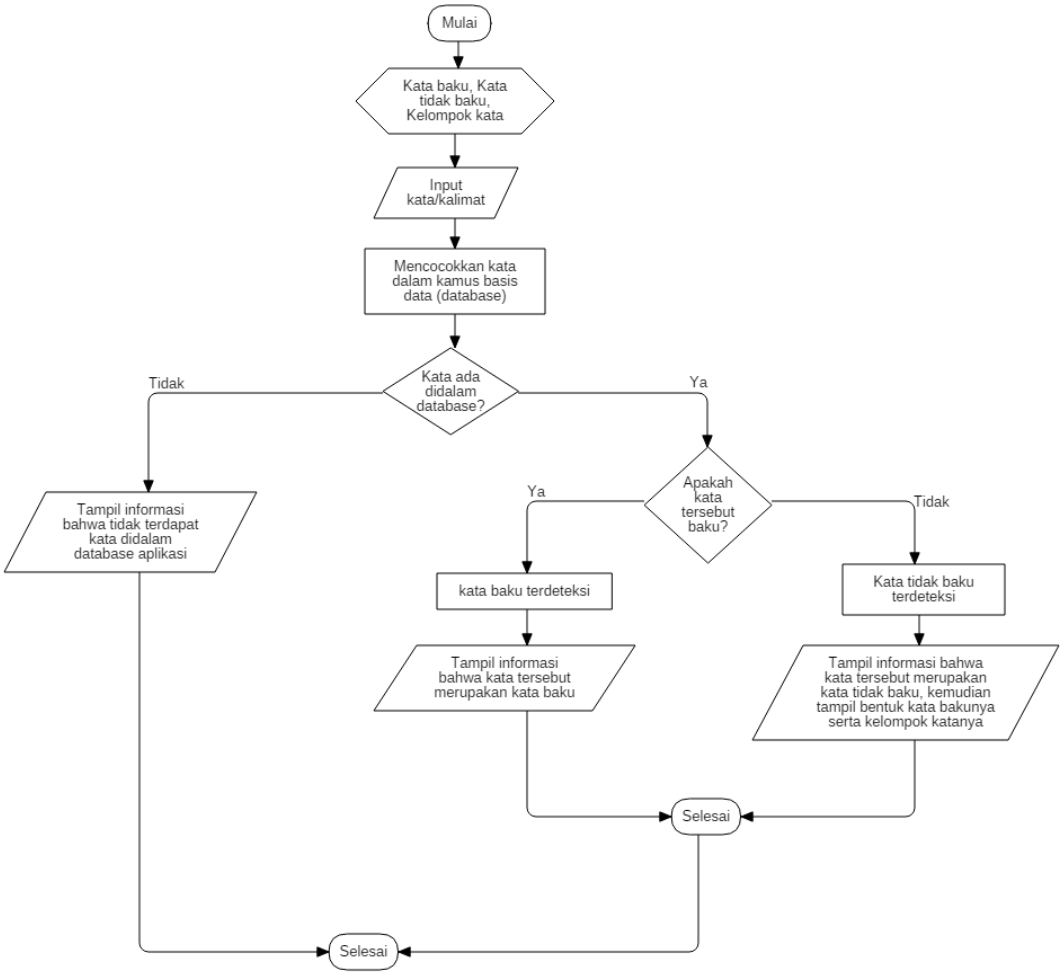


Gambar 4.6 Sistem yang berjalan

Penggunaan kamus bahasa Indonesia sebagai media untuk mengetahui suatu kata, apakah kata tersebut baku atau tidak, dianggap kurang efektif sebab masi dalam pengecekan secara manual, dan membutuhkan waktu yang lama, terlebih jika memiliki kata yang cukup banyak untuk di cek kebakuannya.

B. Sistem Yang di Usulkan

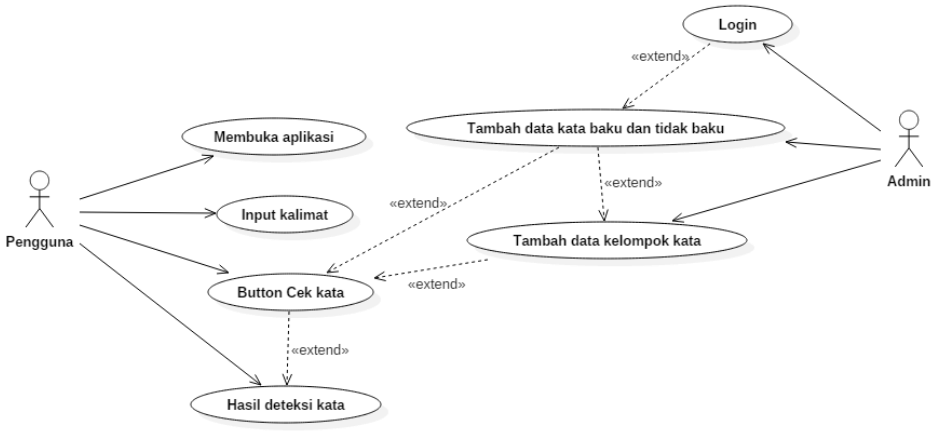
Sistem yang di usulkan pada Aplikasi Pendeteksi Kata Baku dan Tidak Baku adalah sebagai berikut :



Gambar 4.7 *Flowchart* sistem yang diusulkan

1. *Use Case Diagram*

Berikut merupakan gambaran *Usecase diagram* dari Aplikasi Pendeteksi Kata Baku dan Tidak Baku :



Gambar 4. 8 *Use Case* diagram

Pada gambar diatas merupakan perancangan sistem pada Admin yang dibuat menggunakan diagram *usecase*. Berikut penjelasan *usecase* diagram diatas

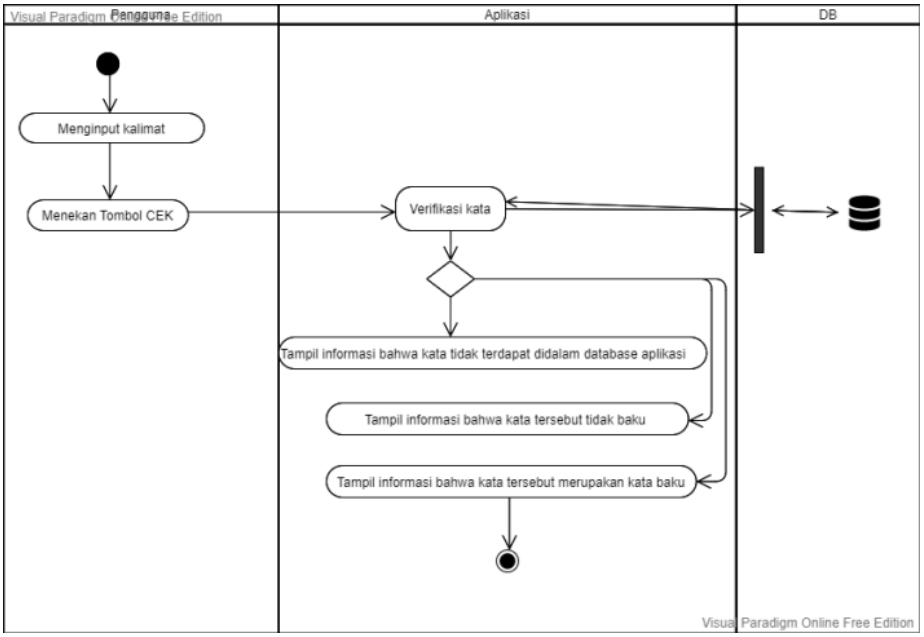
Nama *Use Case*	Deskripsi *Use Case*
Input kalimat	*Use case* ini menjelaskan tentang proses memulai dalam deteksi kata yaitu pengguna akan menginput kata
Button **Periksa Kata Baku dan Tidak Baku**	*Use case* ini menjelaskan setelah pengguna telah menginput kata yang akan dideteksi maka pengguna dapat menekan Button Periksa kata baku dan tidak baku untuk melihat hasil deteksinya

Nama *Use Case*	Deskripsi *Use Case*
Hasil deteksi kata	*Usecase* ini menjelaskan bahwa ketika pengguna telah menekan tombol **cek** maka akan terjadi 3 kondisi yaitu: - Apabila kata yang di input tidak terdapat didalam *database* aplikasi maka akan tampil hasil bahwa kata tersebut tidak terdapat di dalam *database* aplikasi. - Apabila kata yang di input terdapat didalam *database* aplikasi dan merupakan kata tidak baku maka akan tampil hasil bahwa kata tersebut tidak baku kemudian tampil bentuk kata bakunya serta kelompok katanya. - Apabila kata yang di input terdapat didalam *database* aplikasi dan merupakan kata baku maka akan tampil hasil bahwa kata tersebut merupakan kata baku serta tampil kelompok katanya
Login	*Usecase* ini menjelaskan bahwa pada saat Admin ingin menambahkan kata baku dan kata tidak baku serta kelompok kata pada *database*, maka admin harus login terlebih dahulu.
Tambah data kata baku dan tidak baku	*Usecase* ini menjelaskan bahwa setelah admin berhasil login, admin dapat menginput, ,mengedit serta menghapus *database* kata baku dan tidak baku pada aplikasi ini.
Tambah kelompok kata	*Usecase* ini menjelaskan bahwa setelah admin berhasil login, admin dapat menginput, mengedit, serta menghapus database kelompok kata pada aplikasi ini.

2. *Activity Diagram*

Activity Diagram pada Aplikasi Pendeteksi Kata Baku dan Tidak Baku adalah sebagai berikut :

a. *Activity Diagram* pengguna



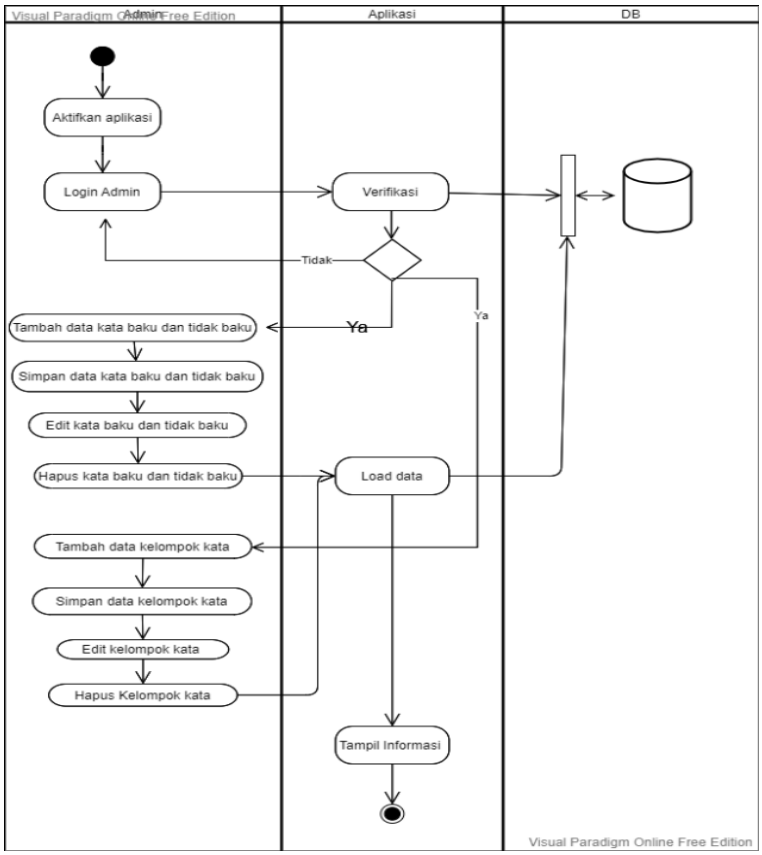
Gambar 4.9 *Activity Diagram* pengguna*/user*

Activity Diagram tersebut menjelaskan bahwa ketika *user* menjalankan aplikasi ini, *user* akan langsung di hadapkan ke halaman utama aplikasi yaitu penginputan kata untuk melakukan pendeteksian kata baku dan tidak baku.

Pada tahap penginputan kata, *user* akan menekan tombol periksa kata untuk melihat hasil dari proses pendeteksi, kemudian sistem melakukan pencocokan didalam *database* aplikasi. Jika aplikasi tidak mendeteksi adanya kata baku maupun kata tidak baku pada kata yang diinput oleh pengguna, maka aplikasi akan menampilkan hasil bahwa kata tersebut tidak terdapat didalam *database* aplikasi. Kemudian jika aplikasi mendeteksi kata didalam *database* yang merupakan kata tidak baku maka akan menampilkan hasil bahwa kata tersebut tidak baku sehingga akan muncul bentuk kata bakunya serta kelompok katanya. Dan jika aplikasi mendeteksi adanya kata baku didalam inputan pengguna, maka

akan menampilkan hasil bahwa kata tersebut merupakan kata baku serta kelompok katanya.

b. *Activity* Diagram Admin



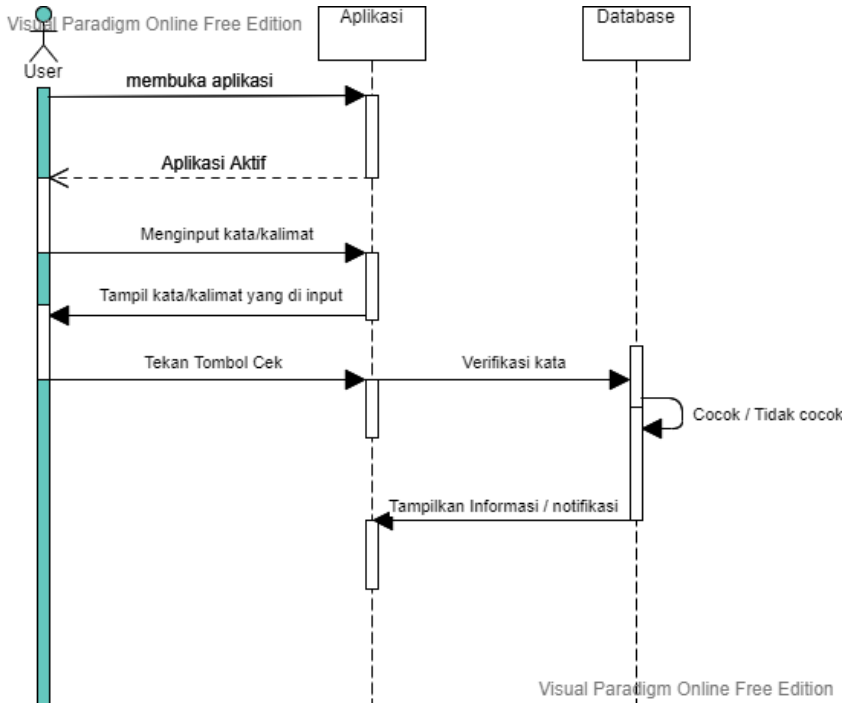
Gambar 4.10 *Activity diagram* pada *Admin*

Activity diagram diatas menjelaskan bahwa ketika Admin ingin menambahkan data kata baku dan tidak baku serta data kelompok kata pada aplikasi ini, maka Admin di harapkan untuk login terlebih dahulu, jika login berhasil admin dapat melakukan input data, *edit*, dan hapus data kata baku dan tidak baku serta data kelompok kata.

3. *Sequence diagram*

Sequence diagram pada Aplikasi Pendeteksi Kata Baku dan Tidak Baku adalah sebagai berikut :

a. *Sequence* Diagram untuk *User*

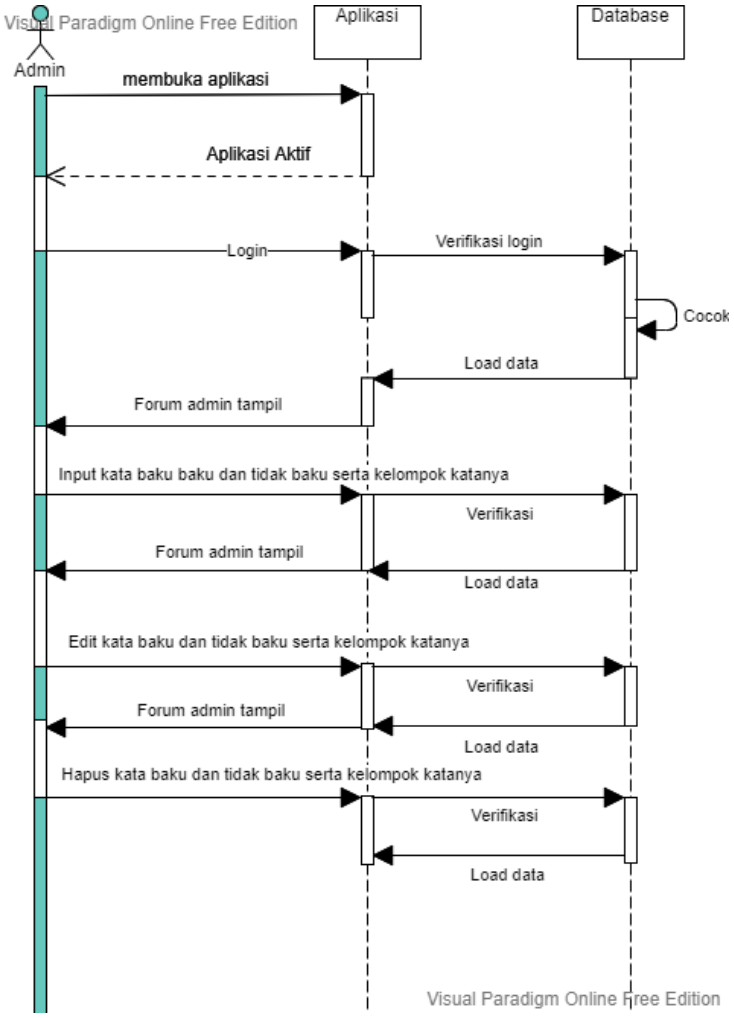


Gambar 4.11 *Sequence diagram user*

Sequence Diagram diatas menjelaskan bahwa untuk menggunakan aplikasi ini pengguna/*user* terlebih dahulu di minta untuk menginput kalimat/kata yang akan dideteksi kemudian pengguna menekan tombol cek, setelah itu sistem akna mencocokkan kata yang di input sesuai dengan pencocokkan yang ada didalam database aplikasi. Jika dalam kalimat tersebut aplikasi tidak mendeteksi adanya kata di dalam *database* maka akan menampilkan informasi bahwa kata tersebut tidak terdapat didalam *database* aplikasi, kemudian jika kata yang di input oleh

pengguna terdapat di dalam *database* maka akan tampil informasi bahwa kata tersebut merupakan kata baku atau jika kata yang di input oleh pengguna merupakan kata tidak baku maka akan tampil informasi bahwa kata tersebut merupakan kata tidak baku yang kemudian di sertakan bentuk kata bakunya dan kelompok katanya.

b. *Sequence* Diagram untuk Admin



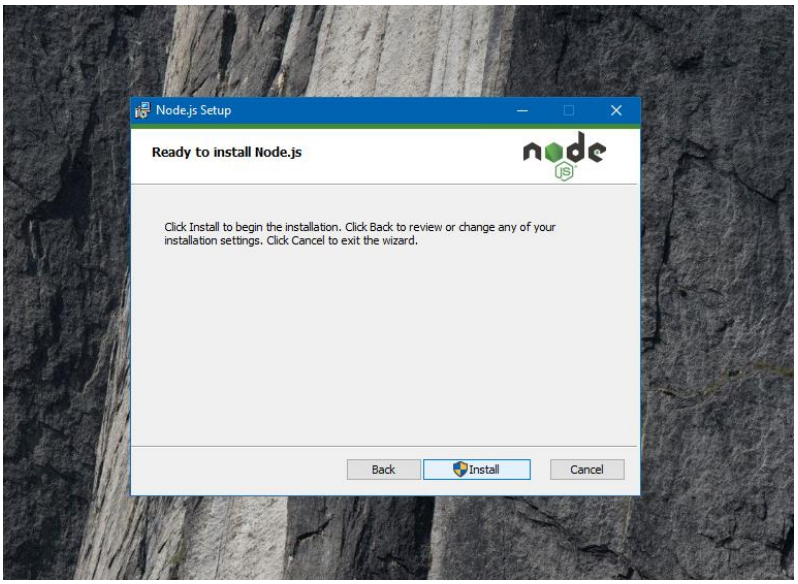
Gambar 4.12 *Sequence* Diagram untuk Admin

Diagram *Sequence* tersebut menjelaskan bahwa untuk menambah data, Admin harus login terlebih dahulu, jika login berhasil, maka akan di alihkan ke halaman *database* aplikasi yang meliputi data kategori dan data kata. Admin dapat menambah/menginput data kategori dan data kata, kemudian admin juga dapat mengedit data, dan menghapus data dari *database* aplikasi.

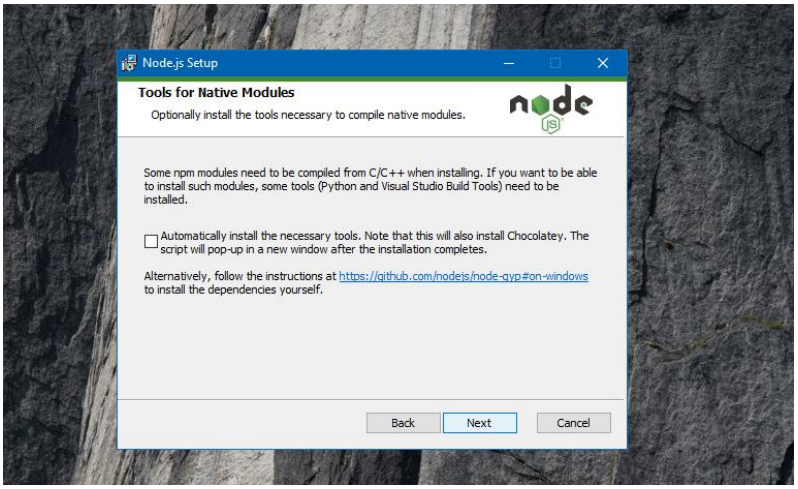
C. Rancangan Sistem

1. Proses Pembuatan Aplikasi

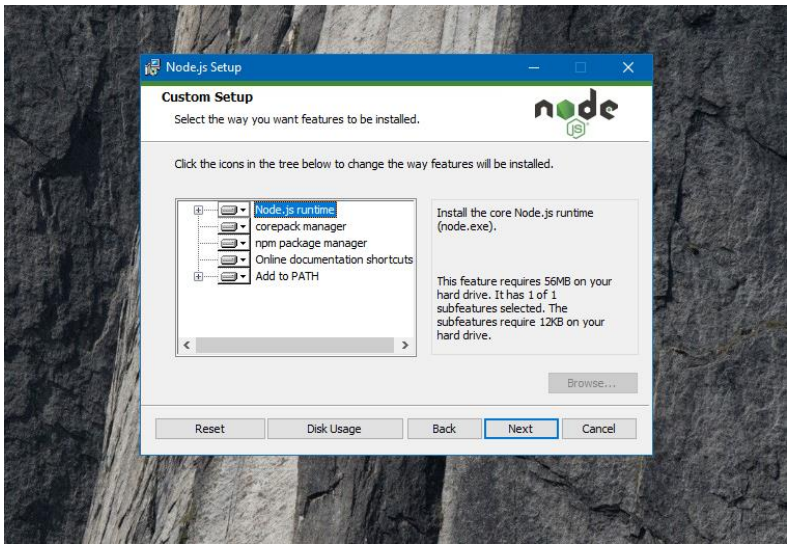
a. Install Node. js

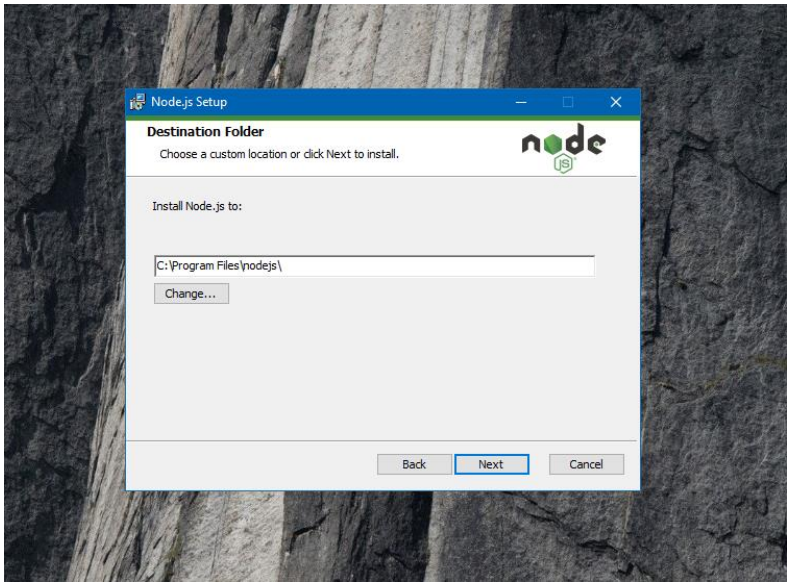


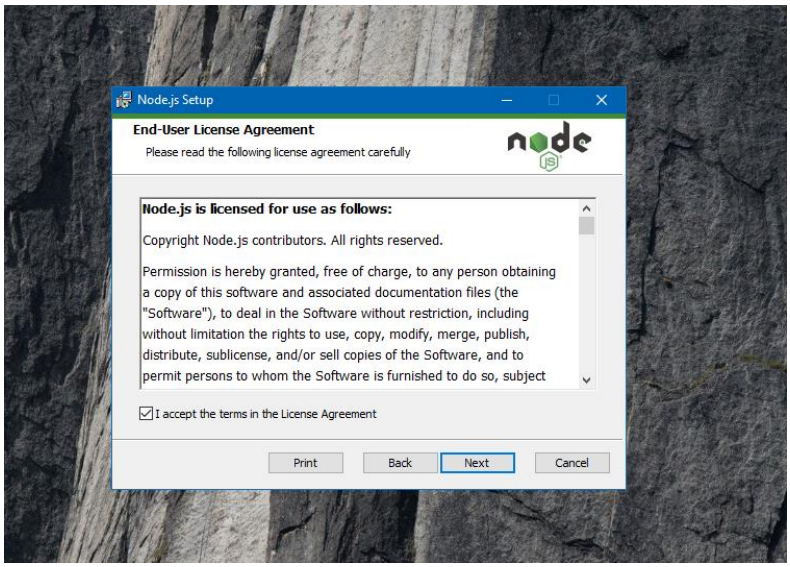
Gambar 4.13 Proses awal *install Node.js*

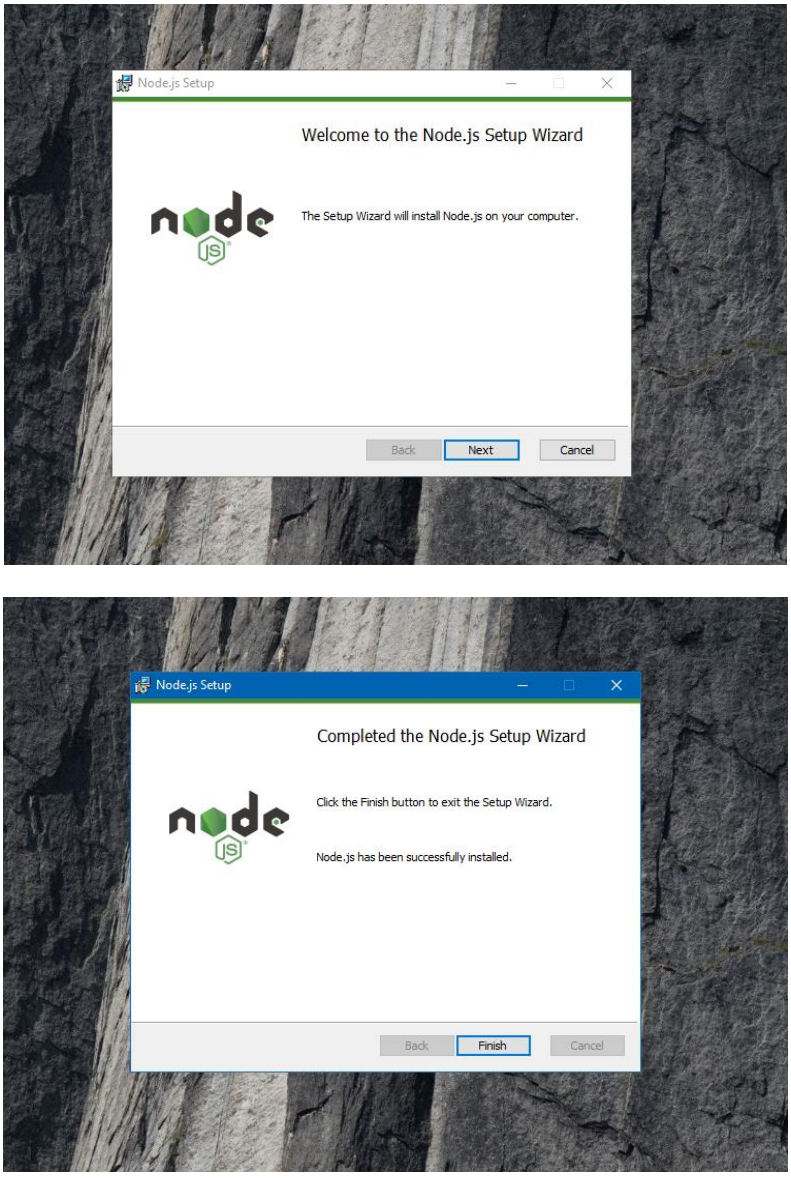


Gambar 4.14 Langkah selanjutnya *Install Node.js*




Node.js Setup
Destination Folder
Choose a custom location or click Next to install.
Install Node.js to:
C:\Program Files\nodejs\
Change...
Back
Next
Cancel

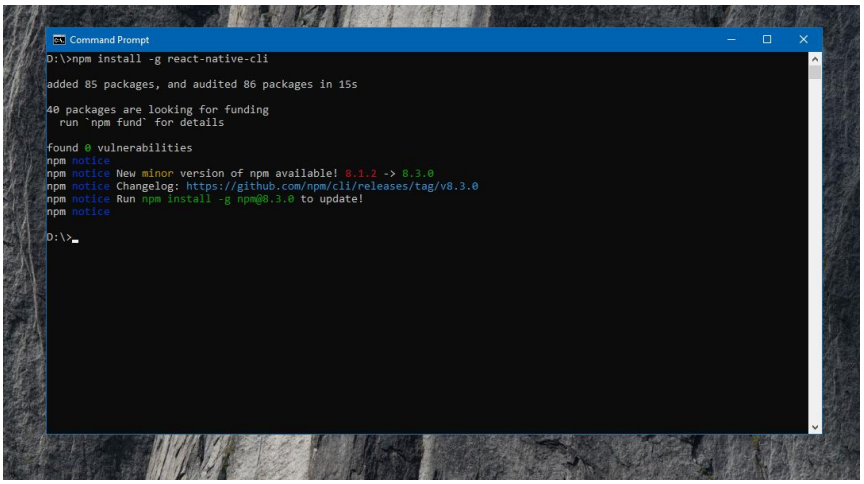

Node.js Setup
End-User License Agreement
Please read the following license agreement carefully
Node.js is licensed for use as follows:
Copyright Node.js contributors. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject
I accept the terms in the License Agreement
Print
Back
Next
Cancel



Gambar 4.15 Proses *Install Node.jc* telah selesai

b. *Install React Native*

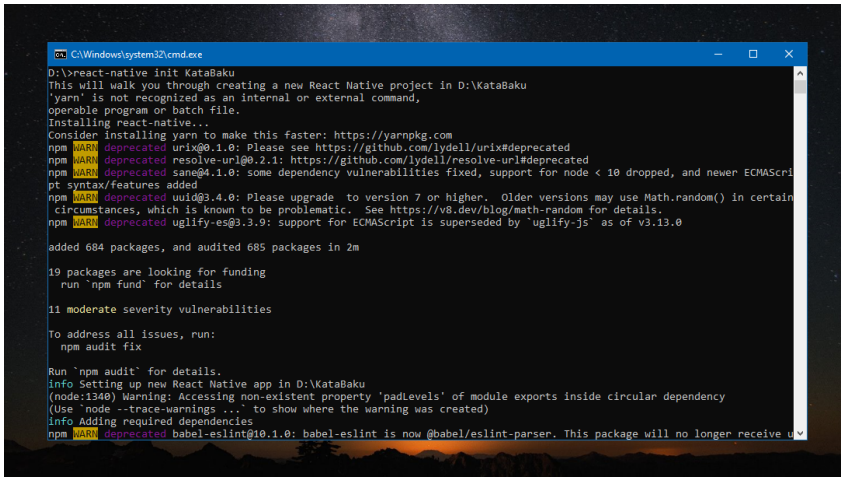
npm install -g react-native-cli



Gambar 4.16 *Install react native*

c. Buat *Project React Native* baru

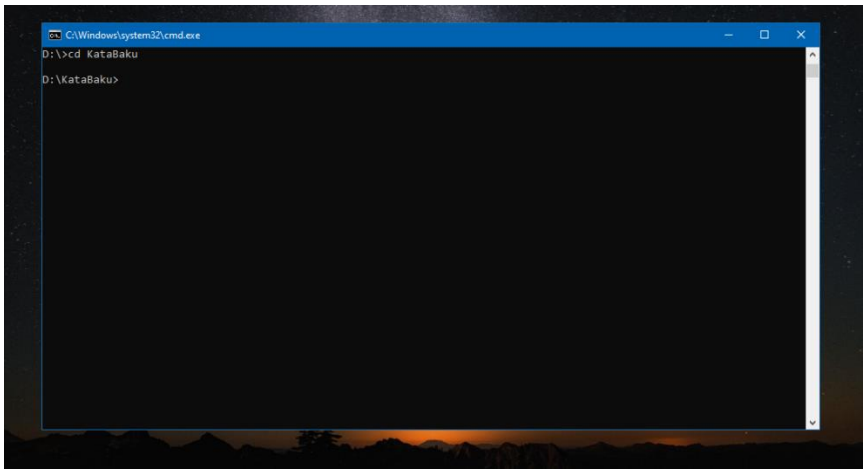
react-native init KataBaku



Gambar 4.17 Buat *Project React Native* baru

d. Pindah ke direktori KataBaku

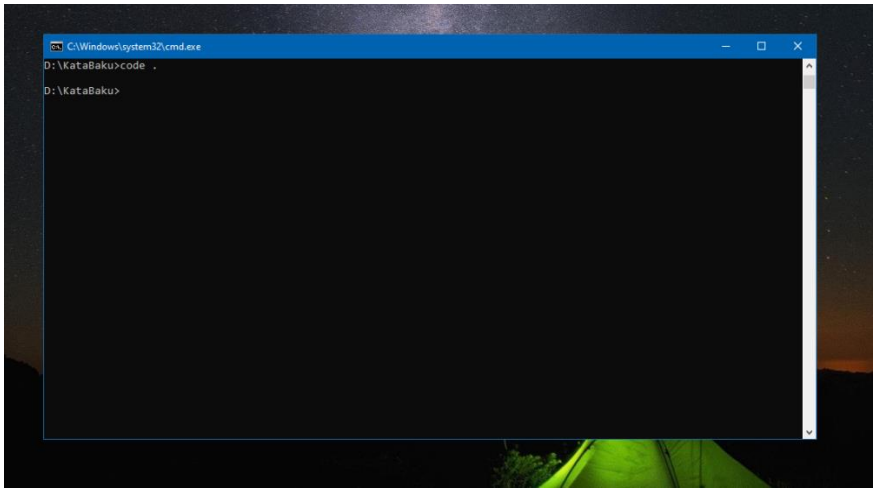
cd KataBaku



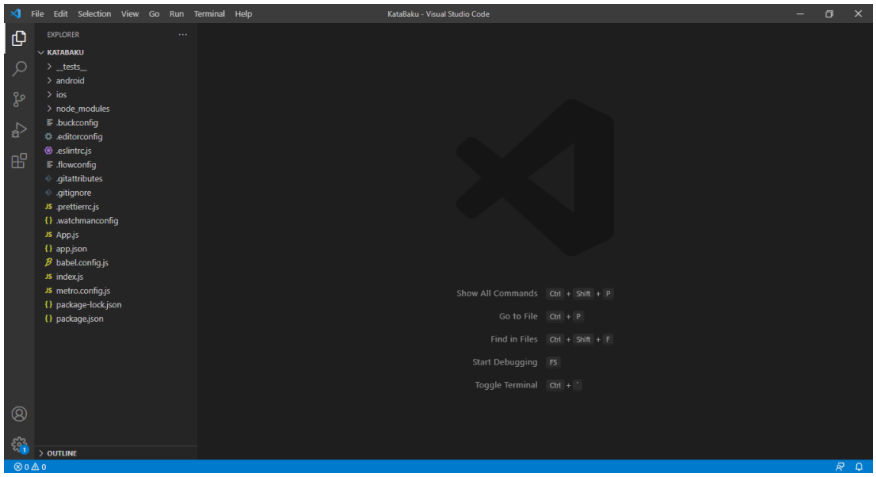
Gambar 4.18 Pindah direktori kata baku

e. Buka direktori dengan kode editor ***code.***

Disini penulis menggunakan *Visual Studio Code* dalam penulisan codingnya.

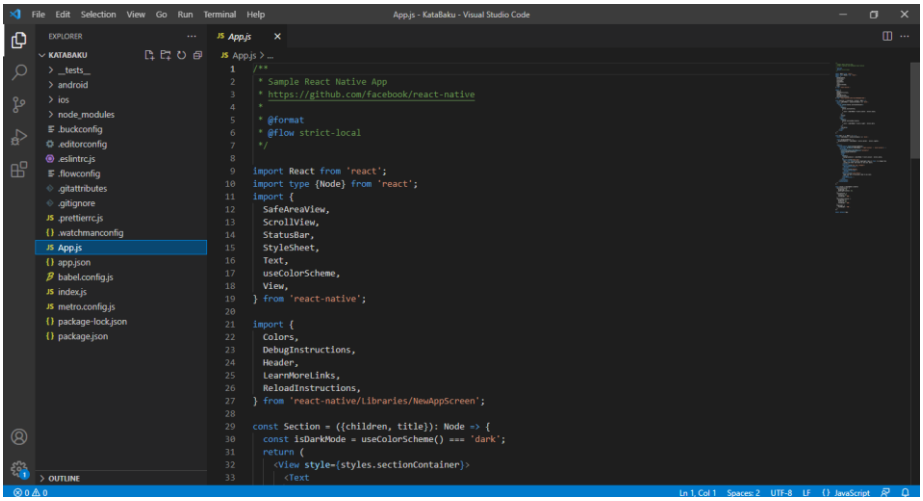


Gambar 4.19 Direktori *code editor*.



Gambar 4.20 Tampilan *Visual Studio code* untuk membuat aplikasi

f. Buka file *App.js* dan mulai pembuatan aplikasi



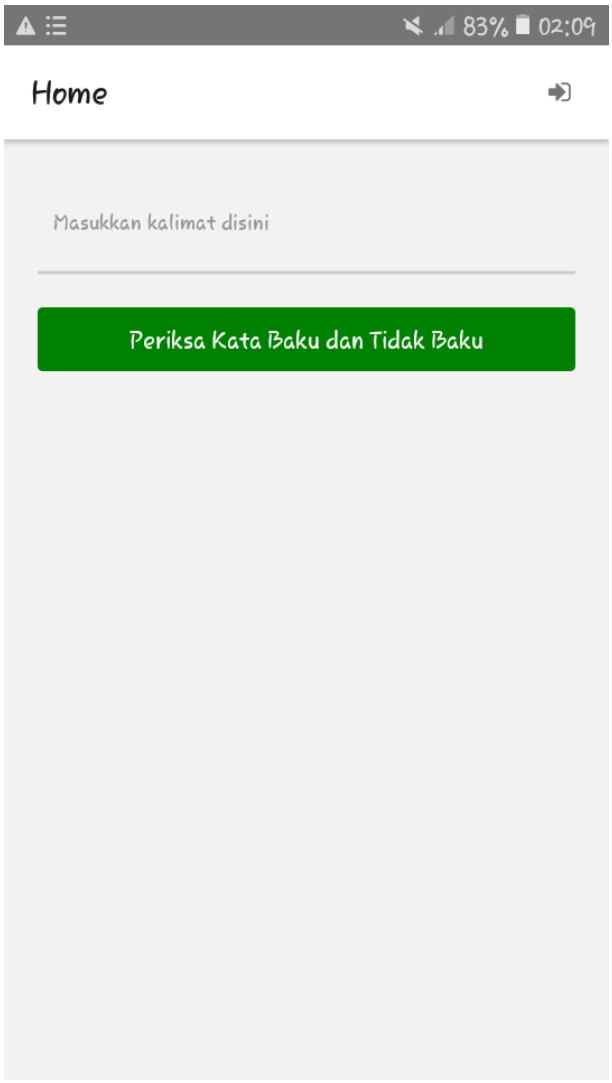
Gambar 4.21 Tampilan file *App.js*

2. Desain Sistem

Dalam proses pembuatan aplikasi ini, penulis menginputkan kata baku dan tidak baku didalam database aplikasi sebagai acuan. Database yang di gunakan di dalam Aplikasi yaitu bersumber dari buku Ernawati Waridah, 2014, Pedoman Kata Baku & Tidak Baku Dilengkapi Ejaan Yang Disempurnakan (EYD),

Bandung, Penerbit Ruang Kata imprint Kawan Pustaka. Berikut tampilan Aplikasi Pendeteksi Kata Baku dan Tidak Baku yaitu sebagai berikut :

a. Tampilan Menu Utama

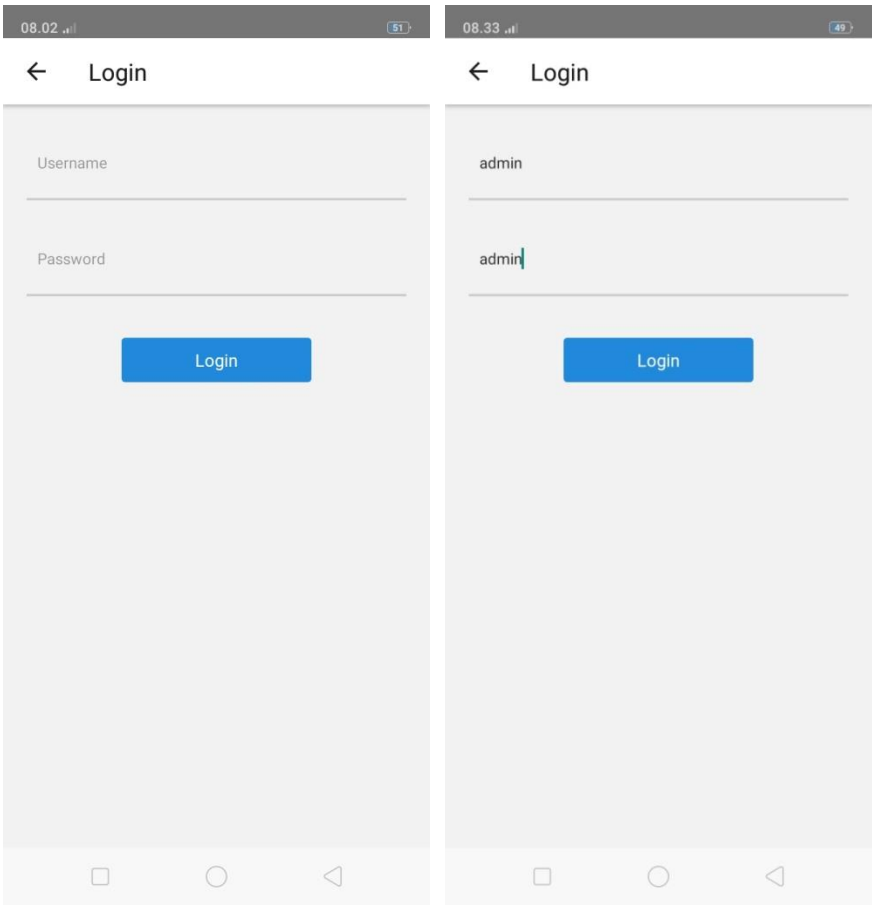


Gambar 4.22 Tampilan Menu Utama Aplikasi

Gambar tersebut merupakan Tampilan Menu Utama ketika pengguna atau *user* pertama kali mengaktifkan aplikasi ini. Di menu Utama, terdapat beberapa tombol yaitu :

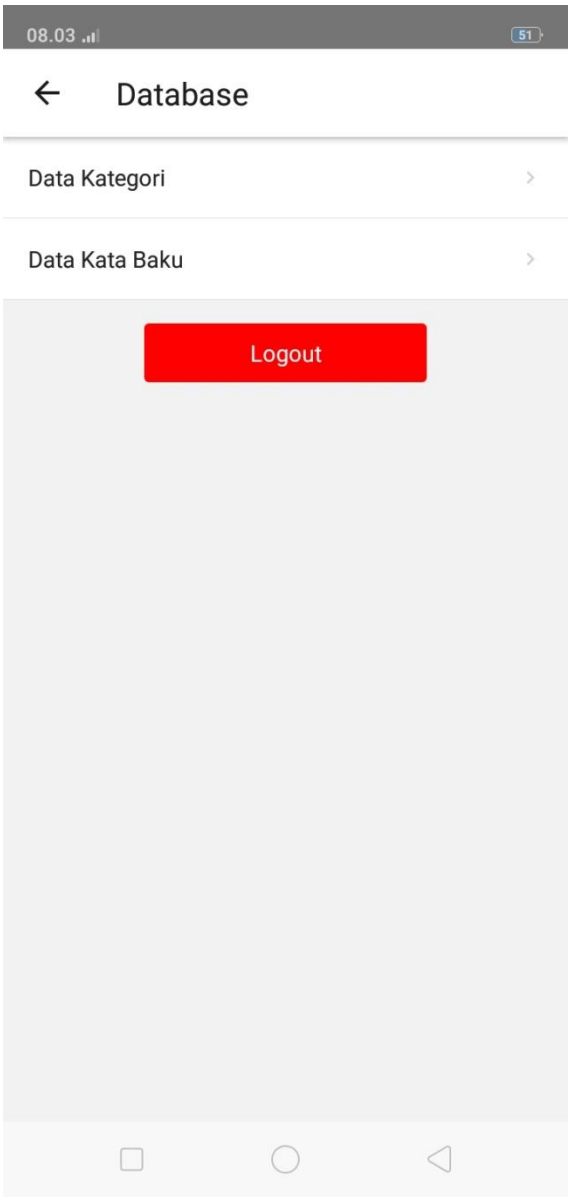
1) Tombol **Periksa Kata Baku dan Tidak Baku** berfungsi untuk melakukan proses pengecekan kata, apakah kata tersebut tidak baku.
2) Tombol panah berfungsi untuk masuk ke login Admin

b. Tampilan Halaman Login Admin



Gambar 4.23 Halaman login Admin

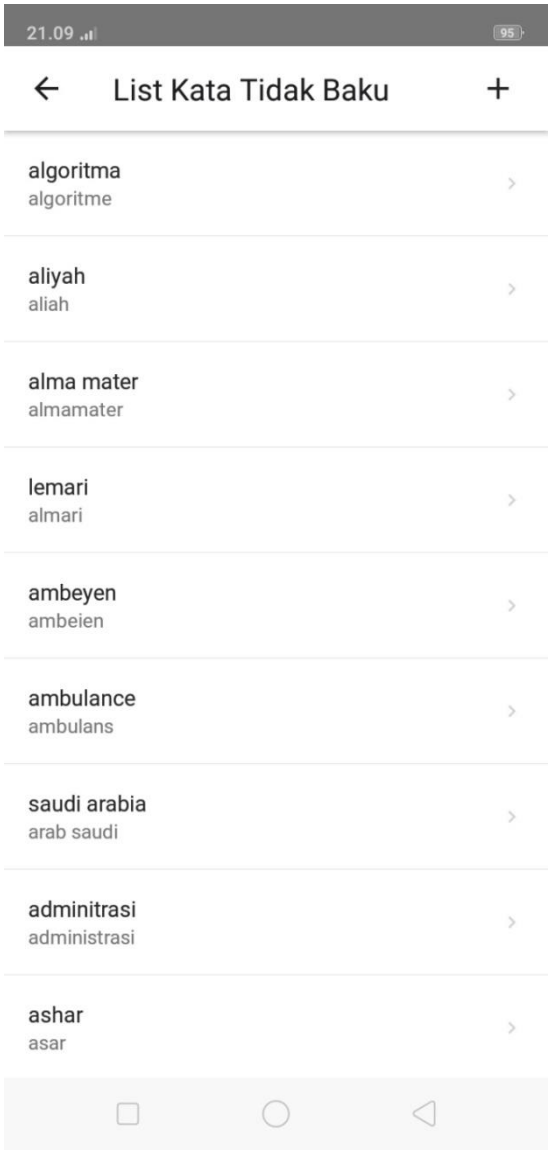
c. Tampilan Halaman *Database*



Gambar 4.24 Halaman Admin *database*

Gambar Tampilan di atas merupakan tampilan dari login Admin yang berhasil yang menampilkan data kategori dan data kata baku. Dan juga terdapat tombol keluar dari halaman admin.

d. Halaman Admin *Database* kata baku dan tidak baku

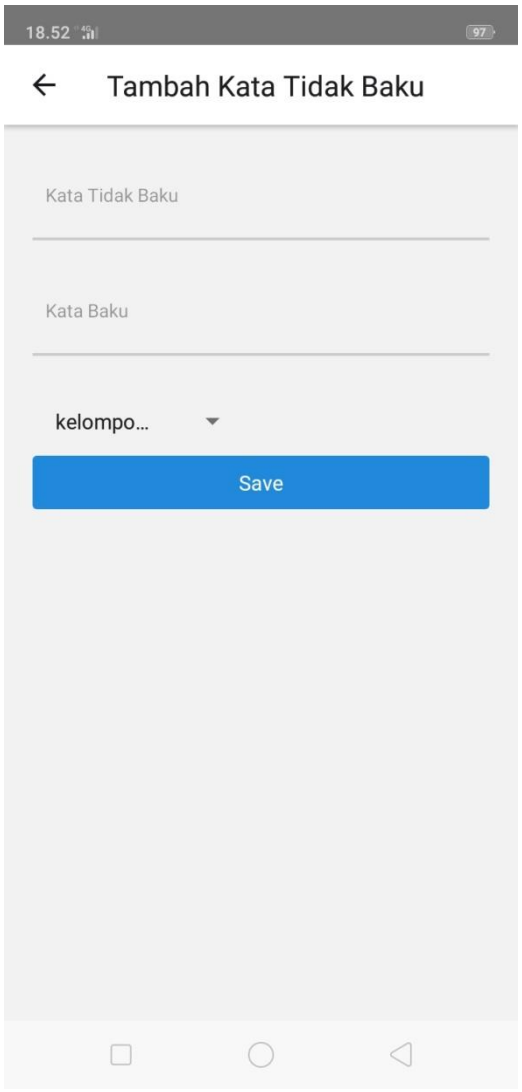


Gambar 4.25 Tampilan awal *database* kata

Gambar diatas merupakan tampilan awal dari data kata baku dan tidak baku, pada tampilan menampilkan list kata baku dan tidak baku, selain itu juga terdapat beberapa tombol diantaranya :

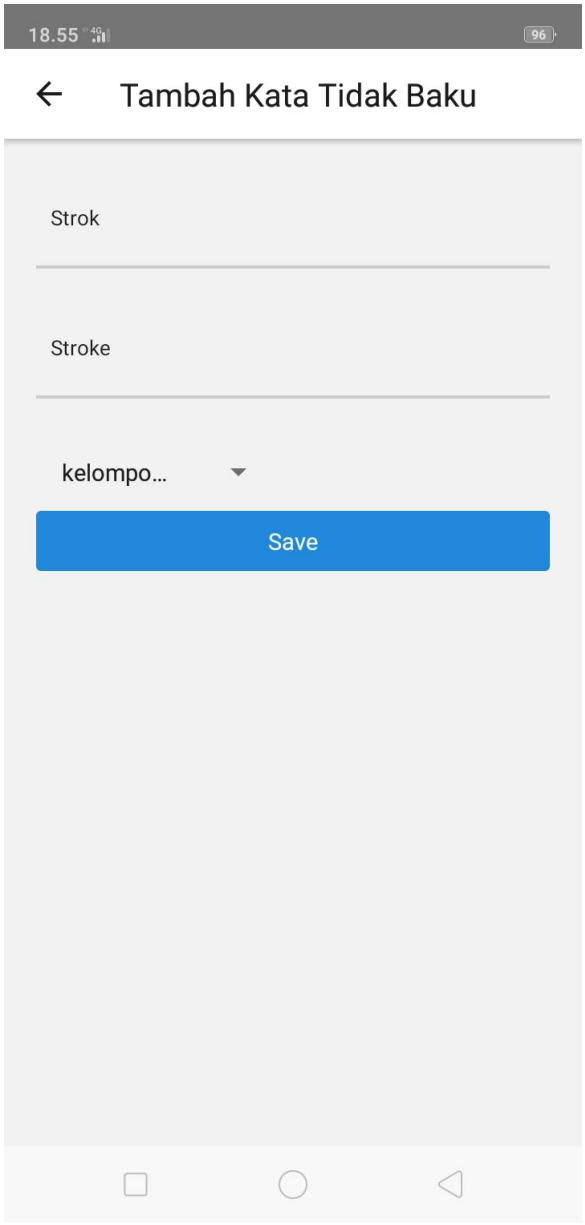
1) Tombol + berfungsi untuk menambahkan kata baku dan tidak baku
2) Tombol > berfungsi untuk melihat detail dari kata tersebut

3) Tombol ← berfungsi untuk kembali ke tampilan sebelumnya

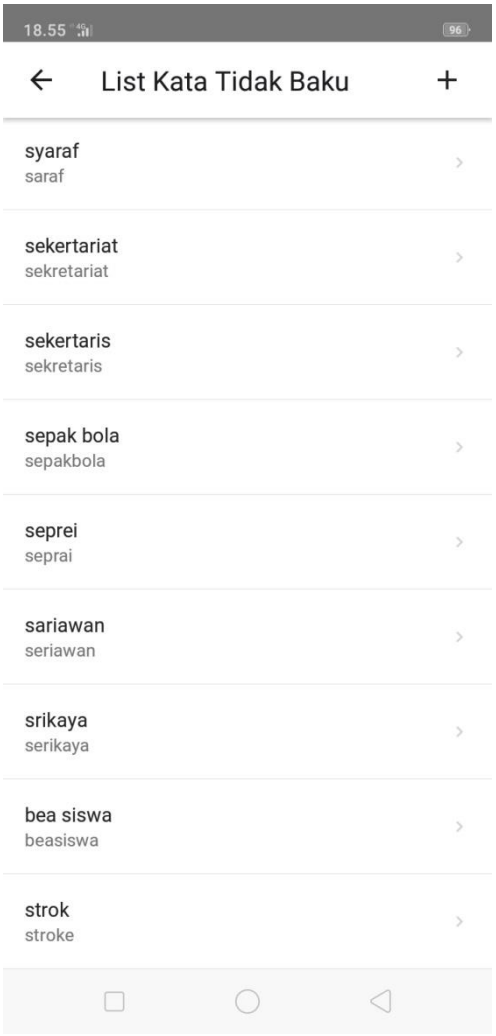


Gambar 4.26 Halaman tambah data kata baku dan tidak baku

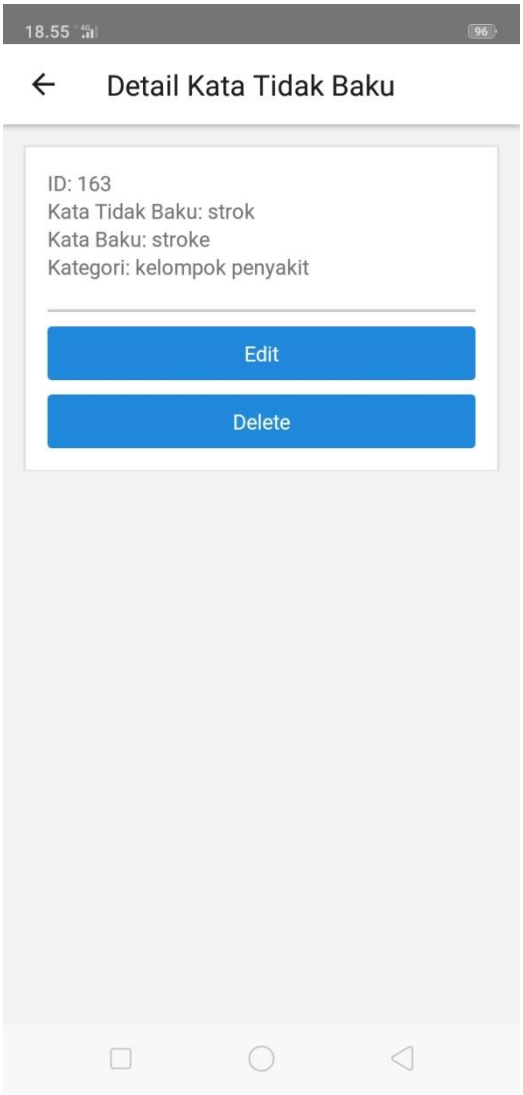
Gambar tampilan di atas merupakan tampilan dari tombol tambah (+) untuk menambah atau menginput data kata baku dan tidak baku.



Gambar 4.27 Tambah data kata baku dan tidak baku yang telah di input

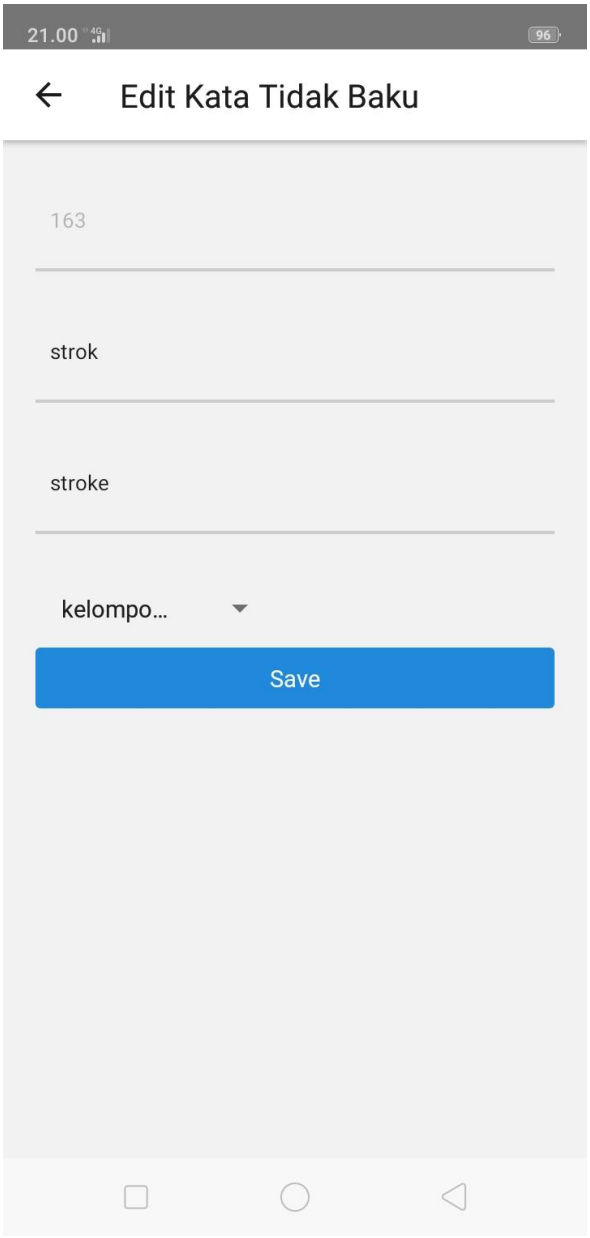


Gambar 4.28 *List* kata baku dan tidak baku yang berhasil di input



Gambar 4.29 detail kata

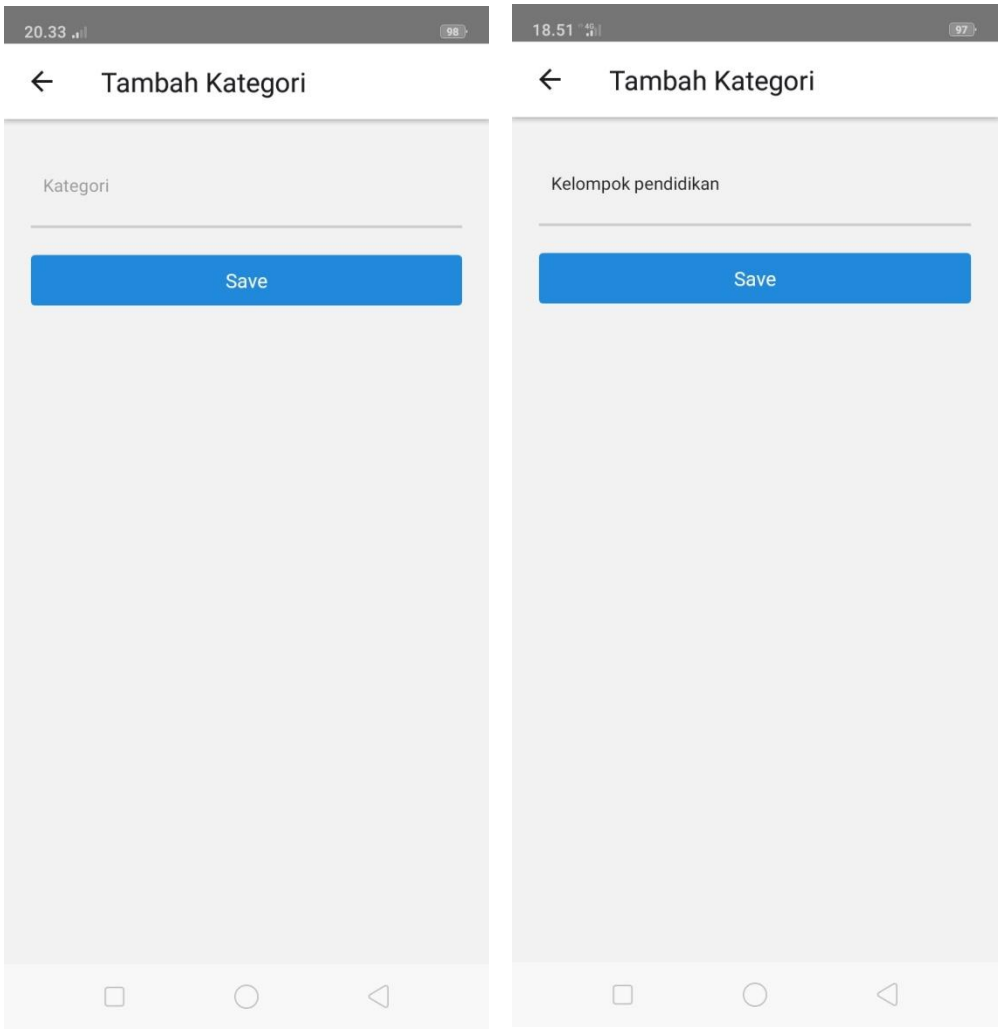
Gambar di atas merupakan tampilan dari tombol detail (>) yang menampilkan keterangan bahwa ini kata baku, kata tidak baku dan kelompok katanya .



Gambar 4.30 Halaman edit kata

Gambar di atas merupakan tampilan dari tombol edit yang menampilkan tampilan untuk mengubah kata baku dan tidak baku.

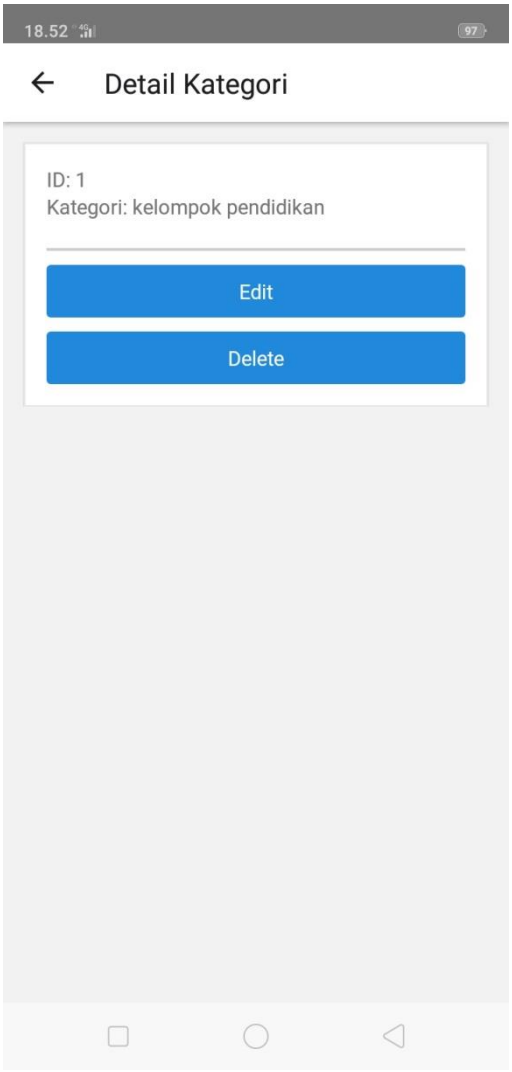
e. Halaman data kategori



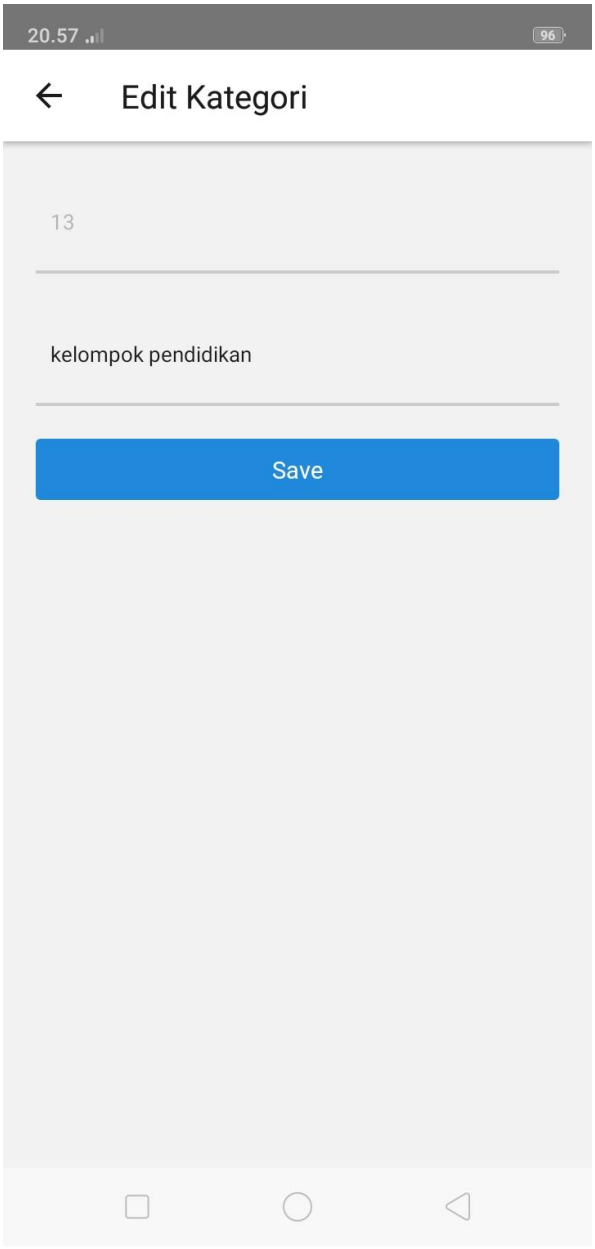
Gambar 4.31 Tampilan halaman tambah kategori yang telah terinput



Gambar 4.32 Tampilan halaman *list* data kategori yang berhasil tersimpan



Gambar 4.33 Tampilan halaman detail kategori

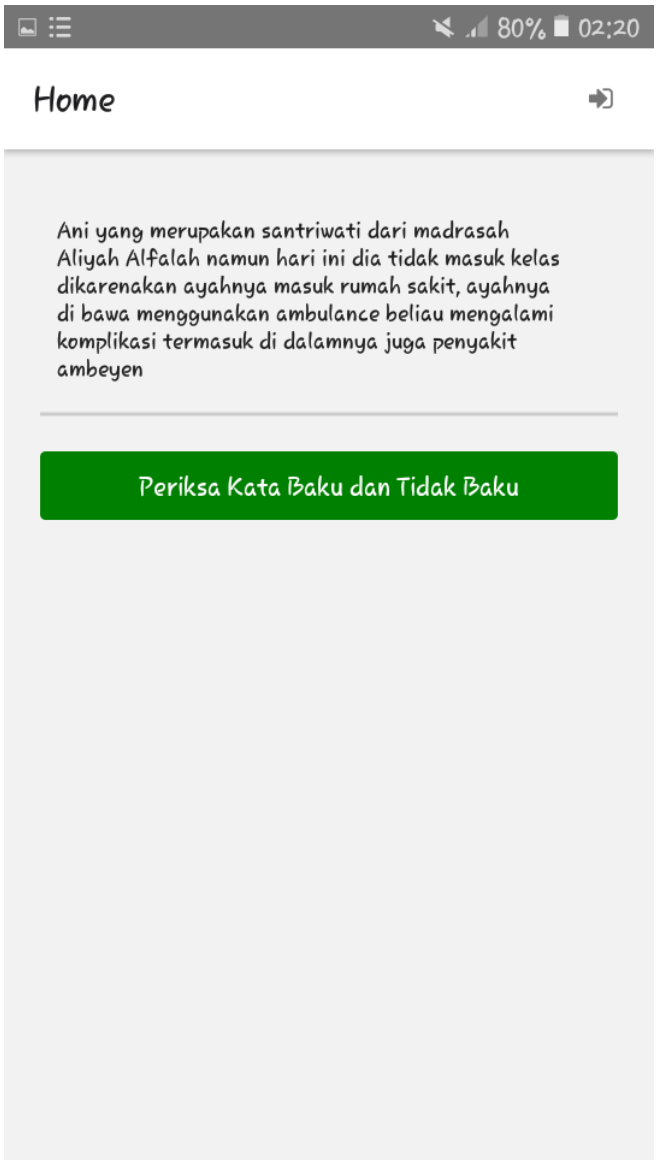


Gambar 4.34 Tampilan halaman edit kategori

Gambar di atas merupakan tampilan dari tombol edit yang menampilkan tampilan untuk mengubah data kategori.

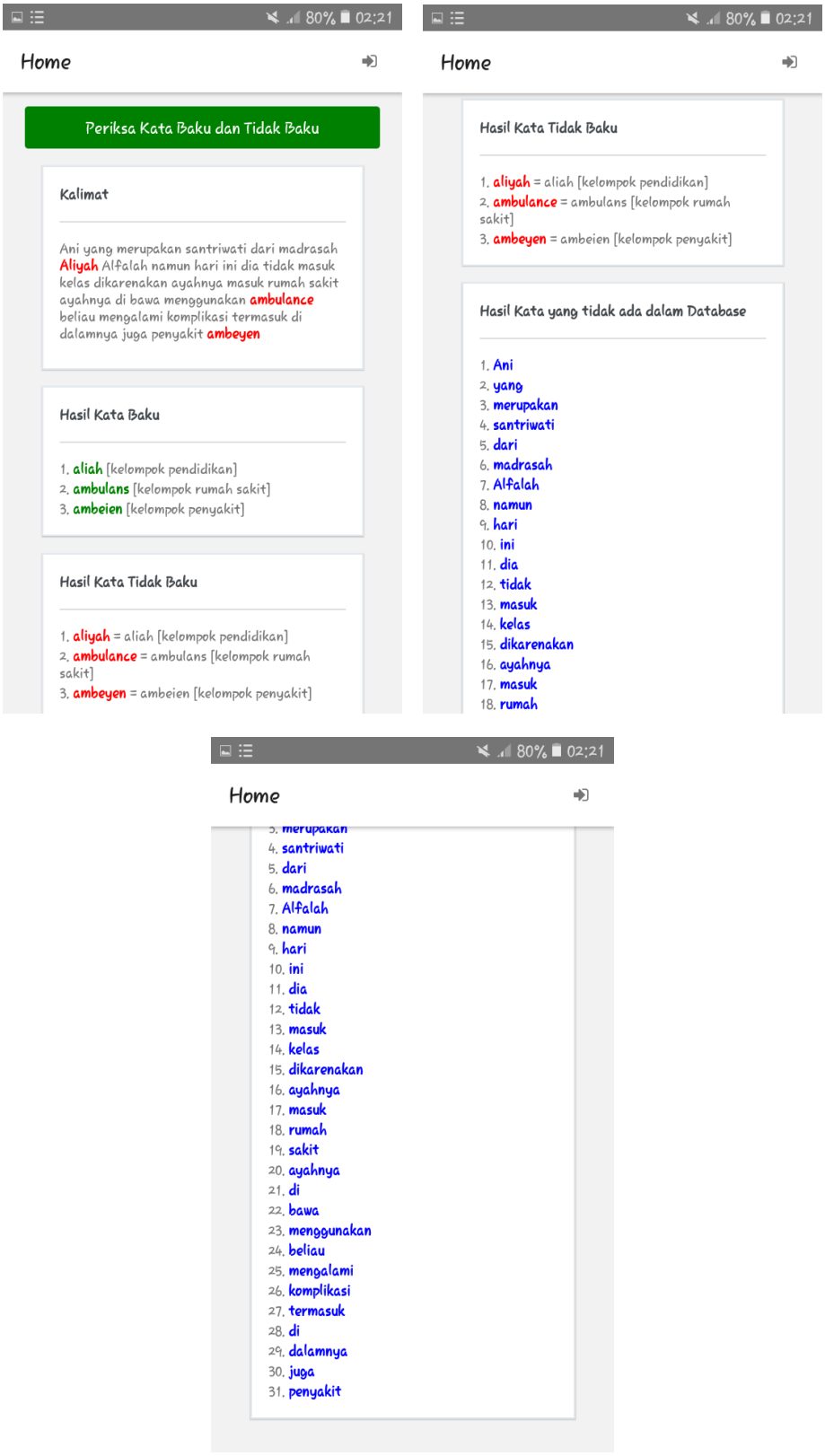
f. Halaman deteksi

Berikut merupakan tampilan dari halaman hasil dari proses deteksi, yaitu sebagai berikut :



Gambar 4.35 Halaman input kalimat

Gambar di atas merupakan input kalimat yang akan di deteksi.

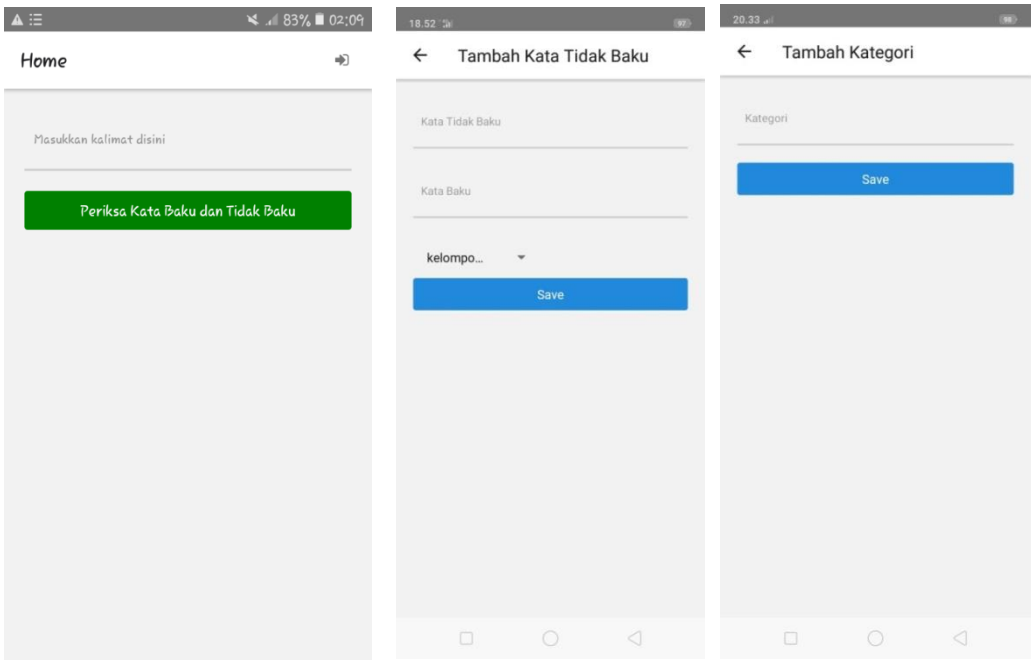


Gambar 4.36 Halaman Proses deteksi berhasil

Dapat kita lihat pada gambar di atas merupakan tampilan dari tombol Periksa Kata Baku dan Tidak Baku, yang menampilkan hasil dari proses deteksi kalimat yang telah di inputkan oleh pengguna, dan pada hasil terdapat tulisan berwarna merah yang merupakan kata tidak baku yang disertakan kata bakunya dan diikuti pengelompokkan katanya. Serta hasil dari kata yang merupakan kata baku dan hasil dari kata yang tidak terdapat didalam *database* aplikasi.

D. Perancangan Input dan Output

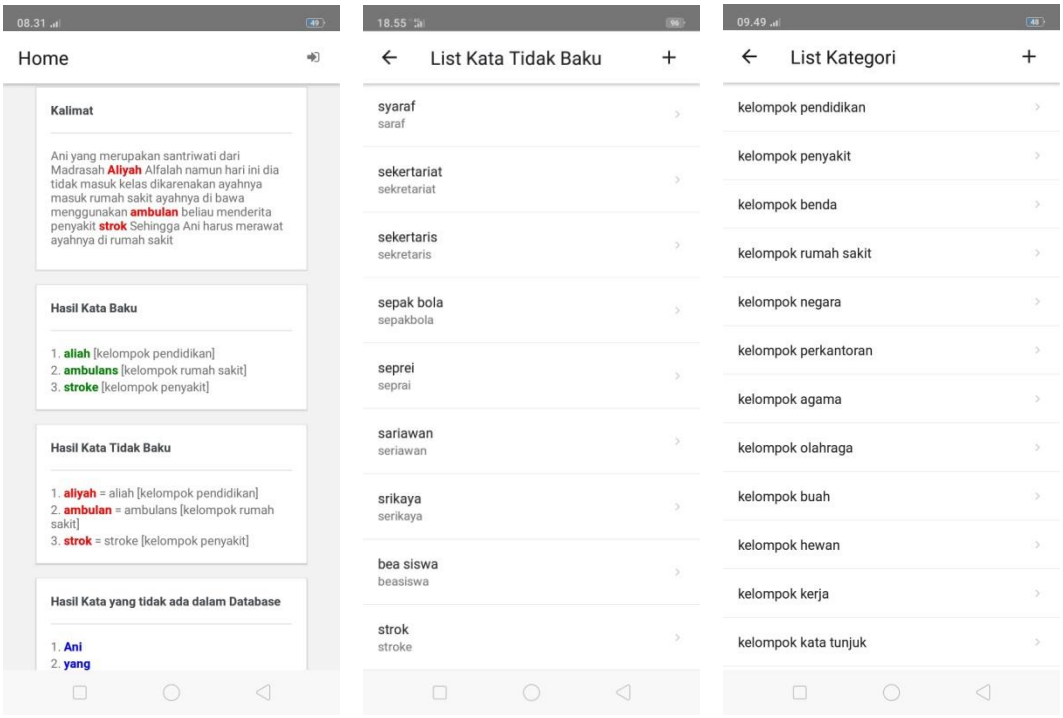
1. Rancangan Input



Gambar 4.37 Rancangan Input

Gambar di atas merupakan kumpulan tempat kata di masukkan atau di input untuk selanjutnya di proses.

2. Rancangan Output



Gambar 4.38 Rancangan *Output*

Gambar di atas merupakan tampilan *Output* dari aplikasi Pendeteksi Kata Baku dan Tidak Baku, yaitu kumpulan data kategori yang berhasil tersimpan dalam bentuk list kata, kumpulan data kata baku dan tidak baku yang berhasil tersimpan dalam bentuk list serta hasil dari proses deteksi yang berjalan.

E. Implementasi

Implementasi sistem adalah sesi pelaksanaan teknologi yang sudah di desain serta siap untuk di operasikan oleh pengguna atau *user*. Pada sesi ini, desain sistem yang sudah di rancang akan diterapkan kedalam bahasa pemrograman yang selanjutnya akan di lakukan pengujian sistem.

Bahasa Pemrograman yang di gunakan untuk membuat dan merancang aplikasi ini adalah bahasa pemrograman *JavaScript*. *JavaSript* sendiri adalah bahasa pemrograman yang mendekati bahasa manusia atau bisa dikatakan bahasa tingkat tinggi, maka dari itu *javascript* mudah di pelajari.

1. Kebutuhan *Hardware*

Spesifikasi *hardware* pengguna aplikasi ini yaitu :

Tabel 4.8 Spesifikasi Perangkat Keras

Jenis	Spesifikasi
Merk	Asus X441M
Processor	*Intel Celeron N4000, up to 2.6GHz*
Memory	4 GB
Hard Disk Drive	1 TB

2. Kebutuhan *Software*

Spesifikasi *Software* atau pengguna aplikasi ini yaitu :

Tabel 4.9 Spesifikasi Perangkat Lunak

Jenis	Spesifikasi
Sistem Operasi	Windows 10 (64 bit)
Software	*Node.js, Visual Studio Code*
Database	*SQLite*
Bahasa Pemrograman	*JavaScript*
Rancangan program	Start UML

3. Perangkat *Android Mobile*

Spesifikasi *Android Mobile* atau *Smartphone* pengguna aplikasi ini adalah :

Tabel 4.10 Spesifikasi *Smartphone*

Jenis	Spesifikasi
Merek	*OPPO A5s*
Processor	*Eight core*
RAM	2GB

4. Kebutuhan *database*

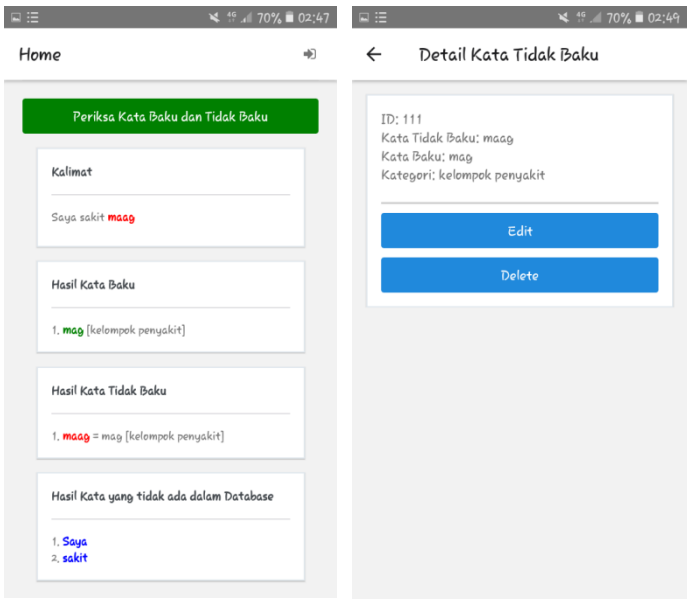
Adapun data kata baku dan tidak baku yang di inputkan di dalam *database* yaitu bersumber dari buku karangan dari Ernawati Waridah, S.S dengan bukunya yang berjudul yaitu Pedoman Kata Baku dan Tidak Baku Dilengkapi Ejaan Yang Disempurnakan.

F. Pengujian Sistem

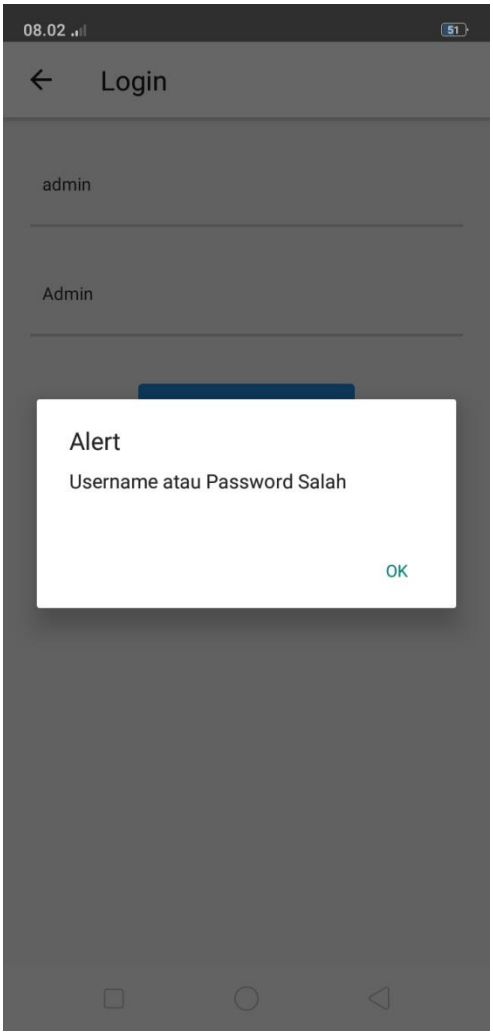
1. Pengujian *Blackbox*

Tabel 4.11 Pengujian *Blackbox* Tombol Periksa kata Baku dan Tidak Baku

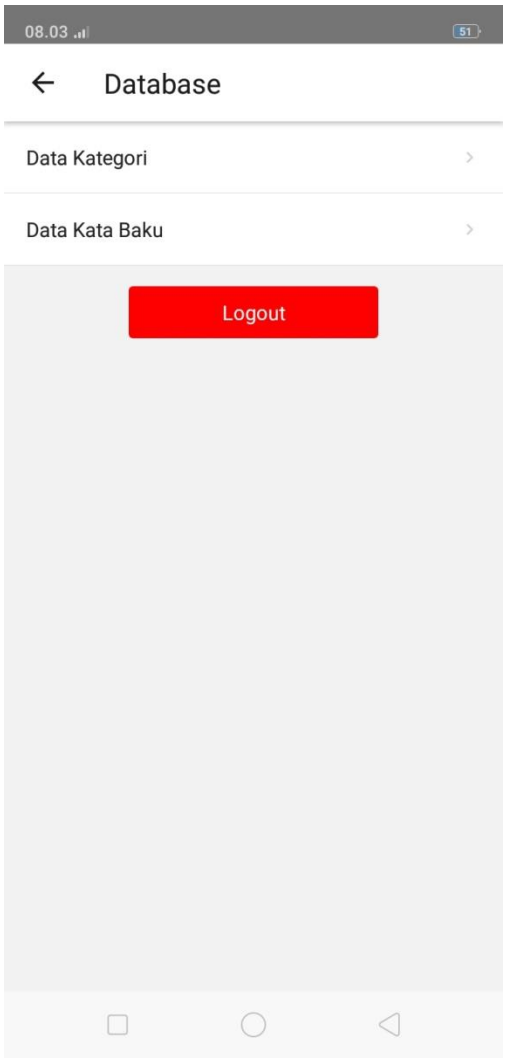
Tes Faktor	Hasil	Keterangan
Proses deteksi kata	✓	Berhasil karena hasil proses deteksi tampil dengan baik, sesuai dengan pencocokan kata yang ada di dalam *database* aplikasi
Tangkapan Layar		



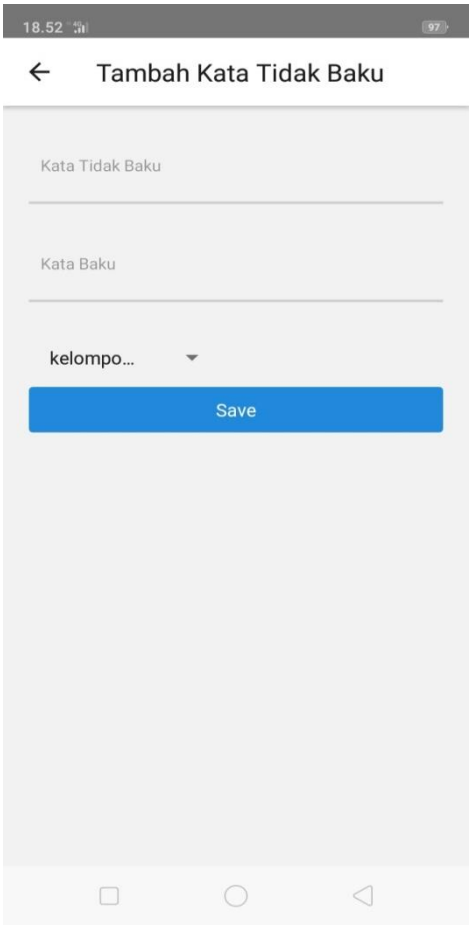
Tabel 4.12 Pengujian *BlackBox* login Admin salah

Tes Faktor	Hasil	Keterangan
Login admin salah	✓	Berhasil, karena tampil pesan *username* atau password salah.
Tangkapan Layar		
		

Tabel 4.13 login Admin berhasil

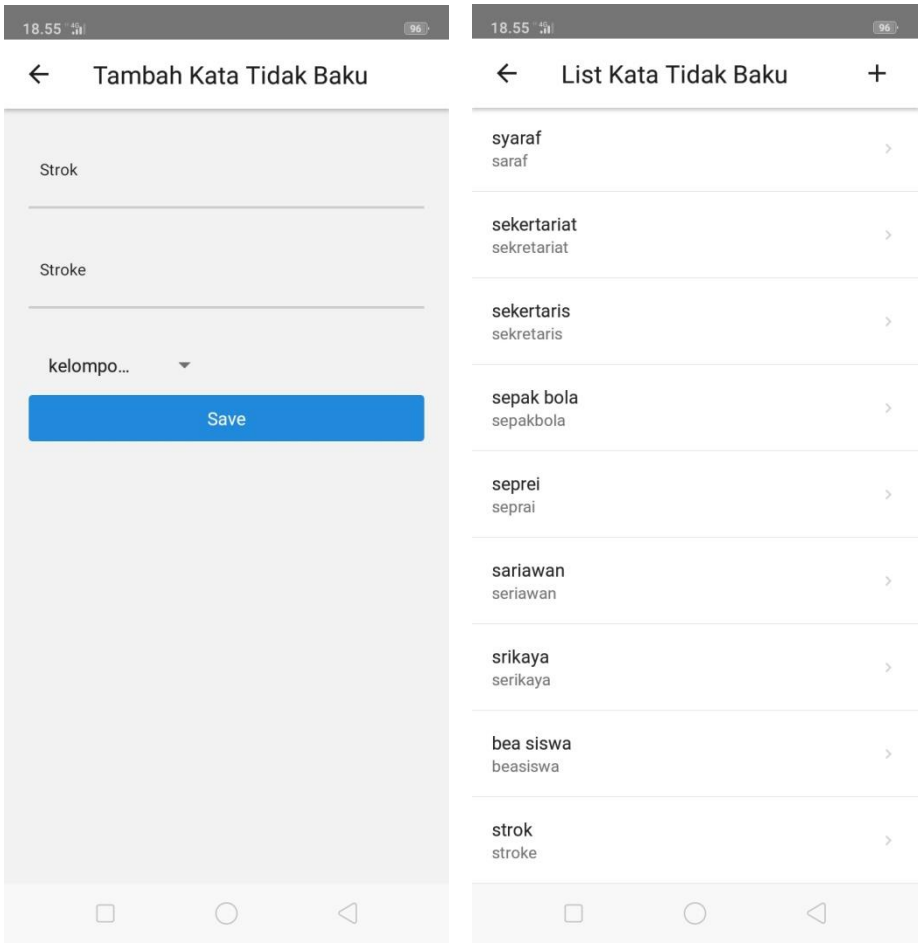
Tes Faktor	Hasil	Keterangan
Login Admin berhasil	✓	Berhasil, karena ketika *admin berhasil login*, maka akan tampil halaman *database*
Tangkapan Layar		
		

Tabel 4.14 Pengujian *BlackBox* Tombol Tambah kata tidak baku

Tes Faktor	Hasil	Keterangan
Tombol Tambah kata tidak baku +	✓	Berhasil, karena ketika *user* menekan tombol tambah pada aplikasi maka akan muncul tampilan tambah kata baku dan tidak baku.
Tangkapan Layar		
		

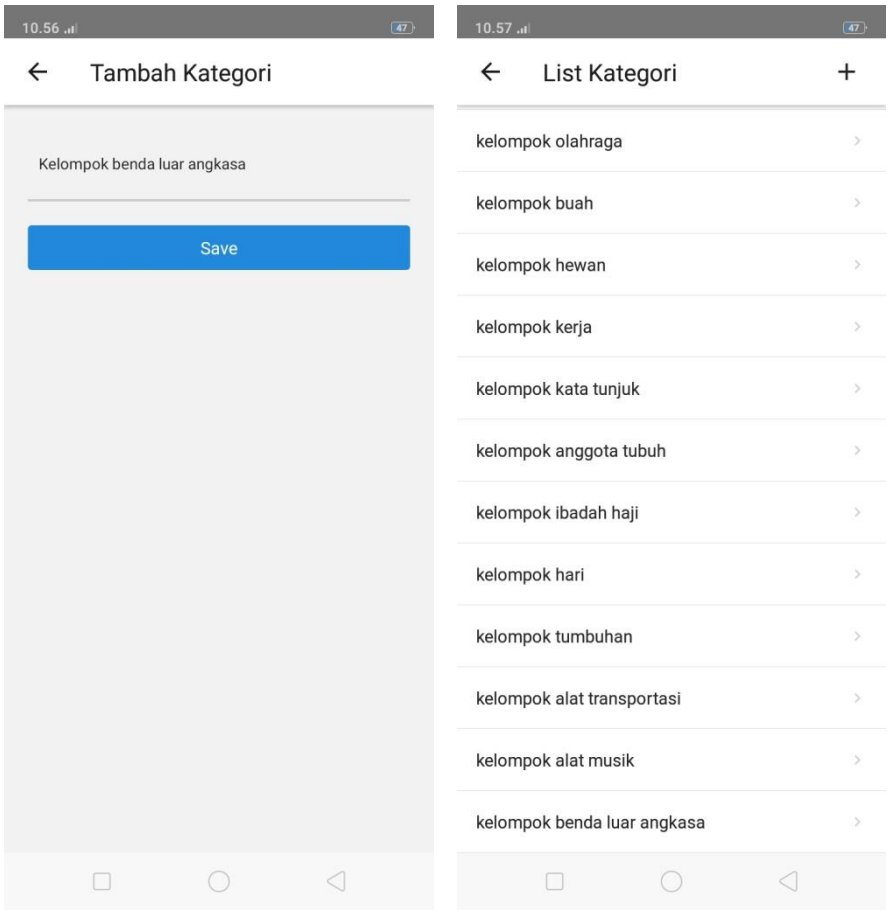
Tabel 4.15 Pengujian *BlackBox* Tombol *Save* kata baku dan tidak baku

Tes Faktor	Hasil	Keterangan
Tombol *Save* data kata baku dan tidak baku	✓	Berhasil karena kata yang di inputkan telah tersimpan di dalam *list* kata.
Tangkapan Layar		



Tabel 4.16 Pengujian BlackBox Tombol Save Data katagori

Tes Faktor	Hasil	Keterangan
Tombol save data kategori	✓	Berhasil karena kata kategori yang di inputkan telah tersimpan didalam list data kategori
Tangkapan Layar		



Tabel 4.17 Hasil pengujian lainnya

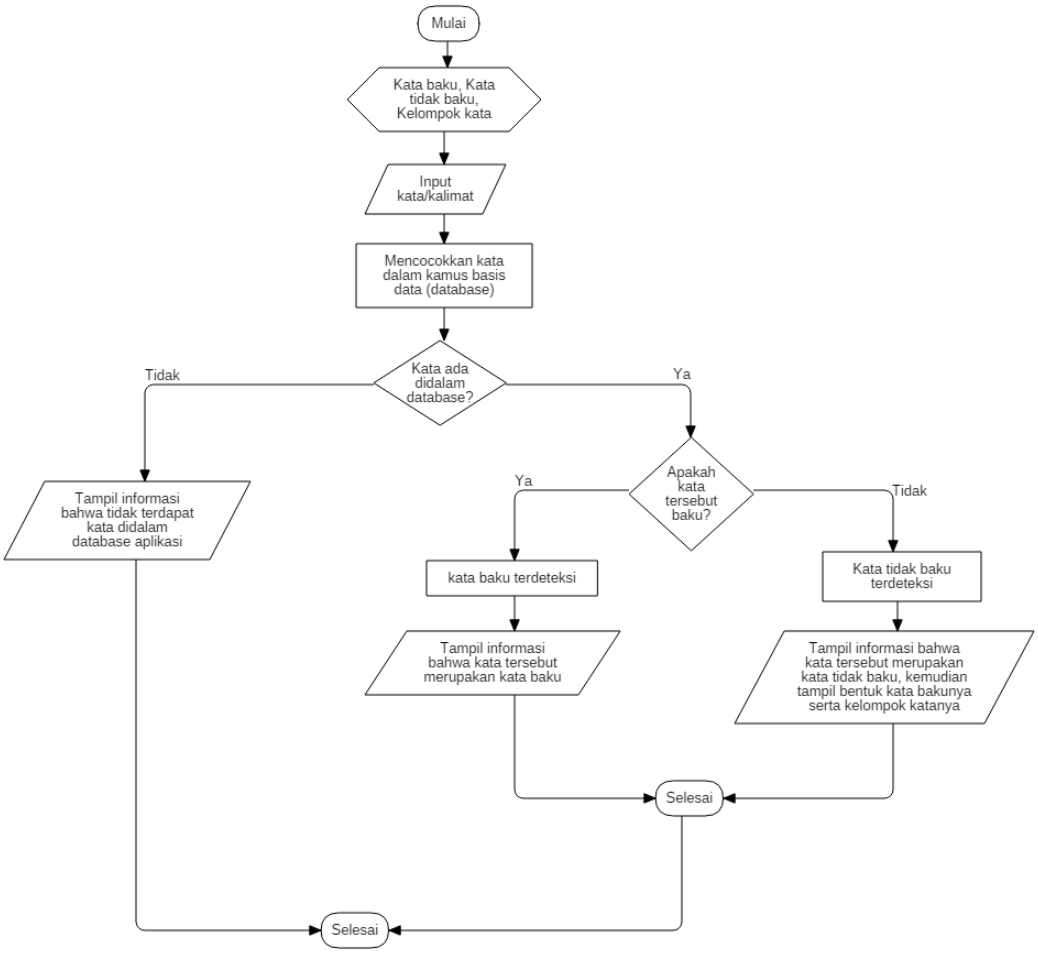
No	Modular	Berhasil	Gagal
1	Tampilan awal aplikasi	✓	-
2	Tombol kembali	✓	-
3	Tampil List data kategori	✓	-
4	Tampil List data kata	✓	-
5	Tampil tanda panah	✓	-
6	Proses tombol login Admin	✓	-
7	Proses tombol *edit*	✓	-
8	Proses tombol *delete*	✓	-
9	Proses tombol *save*	✓	-
10	Proses Tombol keluar	✓	-
Total		10	0

2. Pengujian *Whitebox*

Pengujian *whitebox* dilakukan untuk memastikan bahwa aplikasi dapat berjalan dengan lancar dan tidak terdapat kesalahan dalam logika pemrograman.

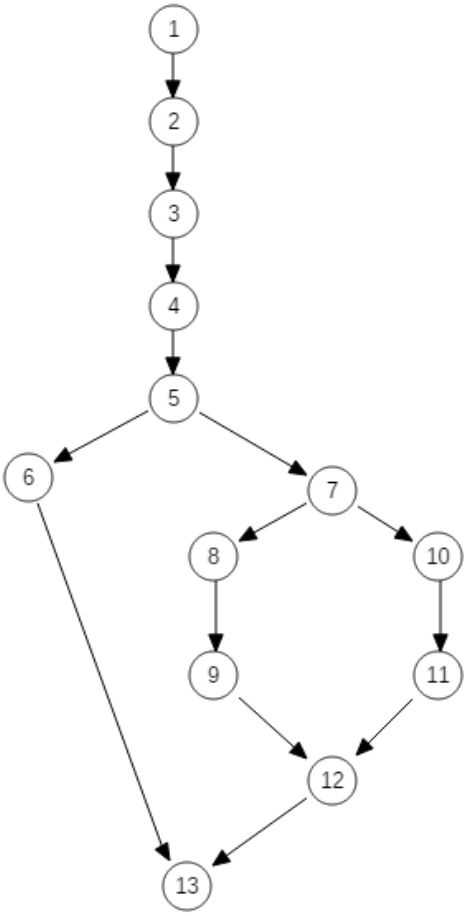
a. Whitebox Aplikasi

Berikut merupakan *Flowchart* dan *Flowgraph* pada Aplikasi Pendeteksi Kata Baku dan Tidak Baku, yaitu sebagai berikut :



Gambar 4.39 *Flowchart* Aplikasi

Dari *Flowchart* yang telah di tampilkan untuk pengujian perangkat lunak, maka ditentukan *flowgraph* sebagai berikut :



Gambar 4.40 *Flowgraph* Tampilan halaman utama

Dari *flowgraph* di atas dapat di lakukan proses perhitungan sebagai berikut :

1) Menghitung *Cyclomatic ComplexMcity* V(G) asal *Egde* dan *Node* :

   Mengunakan rumus : $V(G) = E - N + 2$

   E (edge) = 14

   N (Node) = 13

   P (Predikat Node) = 2

   Penyelesaian : $V(G) = E - N + 2$

   = 14– 13 + 2

   = 3

Predikat Node (N) = P + 1

= 2 + 1

= 3

2) Berdasarkan perhitungan *Cycloatmic Complexcity* dari *flowgraph* di atas mempunyai Region = 3

3) Independentpath pada *flowgraph* di atas ialah :

Path 1 = $1-2-3-4-5-7-10-11-12-13$

Path 2 = $1-2-3-4-5-7-8-9-12-13$

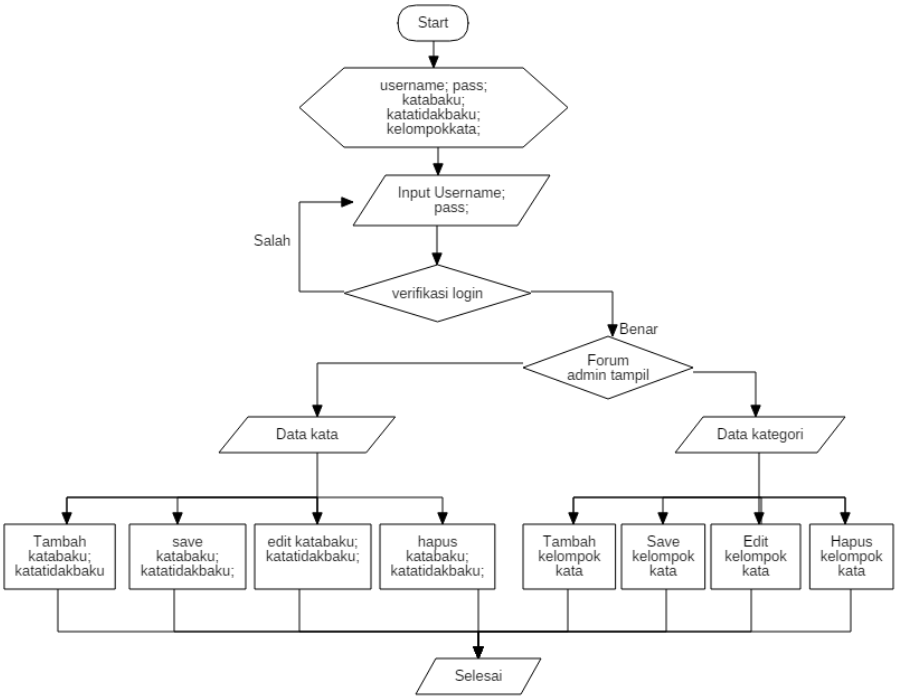
Path 3 = $1-2-3-4-5-6-13$

4) Grafik matriks Aktivitas tammpian halaman

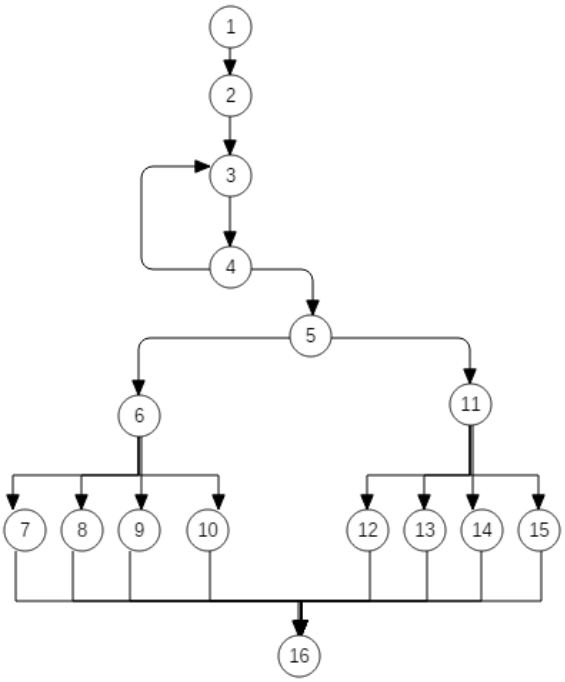
Tabel 4.18 Grafik matriks Aktifitas halaman

	1	2	3	4	5	6	7	8	9	10	11	12	13	E - 1
1		1												$1-1=0$
2			1											$1-1=0$
3				1										$1-1=0$
4					1									$1-1=0$
5						1	1							$2-1=1$
6													1	$1-1=0$
7								1		1				$2-1=1$
8									1					$1-0=0$
9												1		$1-1=0$
10											1			$1-1=0$
11												1		$1-1=0$
12													1	$1-1=0$
13														0
SUM(E+1)											2 + 1 = 3			

b. *Whitebox* Untuk Admin



Gambar 4.41 *Flowchart* Untuk *Form* Admin



Gambar 4.42 *Flograph* Form Admin

1.) Proses perhitungan

Dari gambar *FlowGraph* tersebut dapat dilakukan proses perhitungan sebagai berikut :

a.) Menghitung *Cyclomatic Complexcity* V(G) dari *Egde* dan *Node* :

Dengan Rumus : $\mathrm{V(G)} = \mathrm{E} - \mathrm{N} + 2$

N(*node*) = 16

E(*edge*) = 24

P(*predikat node*) = 9

Penyelesaaian : $V(G) = E - N + 2$

= 24 – 16 + 2

= 10

Predikat Node (N) $= P + 1$

$= 9 + 1$

$= 10$

b) berdasarkan perhitungan *Cyclomatic Complexcity* dari *Flowgraph* diatas memiliki *Region* = 2

c) *Independent Path* pada *flowgraph* diatas adalah :

Path 1 = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 16

Path 2 = 1 + 2 + 3 + 4 + 5 + 6 + 8 + 16

Path 3 = 1 + 2 + 3 + 4 + 5 + 6 + 9 + 16

Path 4 = 1 + 2 + 3 + 4 + 5 + 6 + 10 + 16

Path 5 = 1 + 2 + 3 + 4 + 5 + 6 + 12 + 16

Path 6 = 1 + 2 + 3 + 4 + 5 + 6 + 13 + 16

Path 7 = 1 + 2 + 3 + 4 + 5 + 6 + 14 + 16

Path 8 = 1 + 2 + 3 + 4 + 5 + 6 + 15 + 16

Path 4 = 1 + 2 + 3 + 4 + 3 + 4 + 5 + 6 + 7 + 16

Path 9 = 1 + 2 + 3 + 4 + 5 + 6 + 10 + 16

d) Grafik matriks aktivitas tampilan halaman

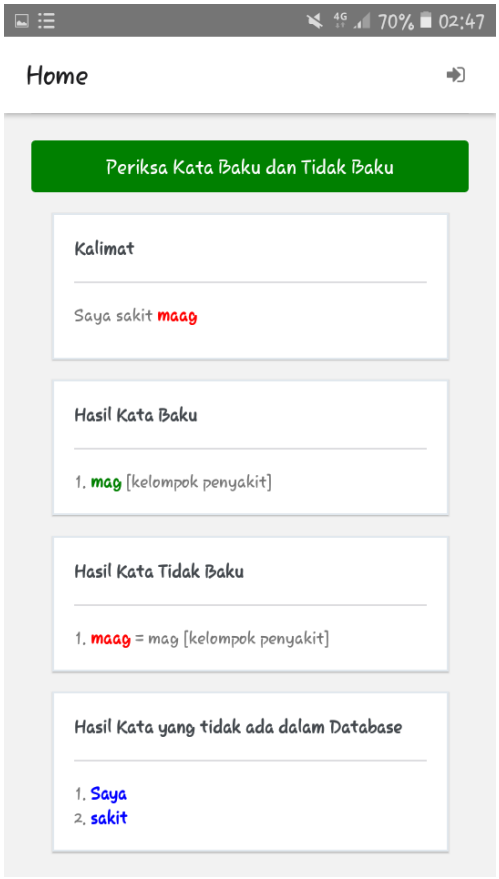
Tabel 4.19 Grafik *matriks* aktifitas halaman

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	E - 1
1		1															1 - 1 = 0
2			1														1 - 1 = 0
3				1													1 - 1 = 0
4			1		1												2 - 1 = 1
5						1					1						2 - 1 = 1
6							1	1	1	1							4 - 1 = 3
7																1	1 - 1 = 0
8																1	1 - 1 = 0
9																1	1 - 1 = 0
10																1	1 - 1 = 0
11												1	1	1	1		4 - 1 = 3
12																1	1 - 1 = 0
13																1	1 - 1 = 0
14																1	1 - 1 = 0
15																1	1 - 1 = 0
16																	1 - 1 = 0
													SUM(E +1)				8 + 1 = 9

G. Pembahasan

Dari hasil pengujian yang telah dilakukan pada antarmuka mesin Linux, pengujian tersebut menunjukkan bahwa fitur program berfungsi sesuai dengan yang diharapkan, input diterima dengan benar serta output yang dihasilkan dengan benar.

Pada pengujian, pengguna menginput kata/kalimat sebanyak mungkin, dalam proses pendeteksian dapat cepat dan juga lambat tergantung dari kata yang diinput oleh pengguna, yang selanjutnya akan di proses untuk mendapatkan hasilnya. Dari proses yang berjalan, kata yang terdeteksi hanya sesuai dengan pencocokkan kata yang terdapat di dalam *database* aplikasi, sehinga jika kata yang di input oleh pengguna terdeteksi ada didalam *database* dan merupakan kata baku maka akan menampilkan hasil tulisan berwarna hijau dan hasil kata yang baku serta tampil kelompok katanya, begitupun jika kata yang di input oleh pengguna terdeteksi ada didalam *database* dan merupakan kata tidak baku maka akan menampilkan hasil tulisan berwarna merah dan hasil kata tidak baku serta bentuk kata bakunya dan juga kelompok katanya, kemudian jika kata yang di input oleh pengguna tidak terdeteksi atau tidak terdapat didalam *database* aplikasi maka tetap akan menampilkan hasil tulisan berwarna biru dan hasil kata tidak terdapat didalam *database* aplikasi.



Gambar 4.43 Hasil pembahasan

Pada gambar di atas menjelaskan bahwa pengguna menginput 3 kata yaitu **Saya**, **Sakit**, **Maag**, dari ketiga kata tersebut saat pengguna menekan tombol Cek, maka akan tampil hasil yaitu :

1. Hasil Kata Baku, dalam hal ini yaitu **mag**
2. Hasil Kata Tidak Baku, dalam hal ini yaitu **maag**
3. Hasil Kata yang tidak ada dalam *Database*, dalam hal ini yaitu **Saya** dan **Sakit.**

Dari pengujian tersebut, sistem aplikasi yang telah di rancang memiliki kekurangan sehingga pada pengembangan kedepannya dapat di kembangkan sesuai dengan kekurangan dari aplikasi yang ada. Pada aplikasi ini penulis hanya

mendeteksi kata yg terdapat didalam *database* aplikasi sesuai dengan panduan yang terdapat pada buku, sehingga dari segi aplikasi hanya mendeteksi sebagain kata saja oleh karenanya penambahan kata pada *database* dapat di lakukan untuk penelitian selanjutnya. Tidak hanya itu, kata yang di gunakan didalam *database* aplikasi merupakan kata-kata dasar saja, dalam artian tidak menggunakan imbuhan awalan maupun akhiran, dari hal tersebut dapat di kembangkan dengan menambahkan imbuhan awalan-akhiran dalam proses pendeteksiannya. Selain dari kekurangan tersebut, dari segi fitur aplikasi juga dapat di tambahkan untuk input suara atau file dokumen sehingga pengguna tidak lagi mengetik secara manual kata kedalam aplikasi.



Gambar 4.44 buku sumber acuan pada *database*

Adapun kata yang penulis gunakan pada aplikasi yaitu sebanyak 1000 kata, yang kemudian dari kata tersebut penulis kelompokkan kedalam beberapa kategori diantaranya yaitu : kelompok pendidikan, kelompok penyakit, kelompok

benda, kelompok rumah sakit, kelompok luar negri, kelompok perkantoran, , kelompok agama, kelompok olahraga, kelompok buah, kelompok hewan, kelompok kerja, kelompok kata tunjuk, kelompok bagian tubuh, kelompok ibadah haji, kelompok hari, kelompok tumbuhan, kelompok transportasi/kendaraan, kelompok music, kelompok benda luar angkasa, kelompok sholat, kelompok bulan hijriah, kelompok bulan masehi, kelompok makanan, kelompok pertandingan, kelompok bunga, kelompok warna, kelompok pemerintahan, kelompok alqur'an, kelompok bangunan, kelompok zat, kelompok sejarah, kelompok bahan makanan, kelompok pernikahan, kelompok bagian kendaraan/transportasi, kelompok kata umum, kelompok gedung/tempat, kelompok gaib, kelompok wisatawan. Berikut contoh penggunaan kata baku dan tidak baku di dalam *database* aplikasi, yaitu sebagai berikut :

Tabel 4.20 Contoh kata baku dan tidak baku serta kelompok katanya

Kata baku	Kata tidak baku	Kelompok kata
aerobik	erobik	Kelompok olahraga
akuntan	accountan	Kelompok perkantoran
ambeien	Ambeyen/ambeiyen	Kelompok penyakit
ambulans	Ambulan/ambulance	Kelompok rumah sakit
gurami	gurameh	Kelompok hewan
Capcai	Cap cai	Kelompok makanan
cengkih	Cengkeh	Kelompok bunga
Cina	China	Kelompok luar negri
Fatihah	Fatikah/fatehah	Kelompok al-quran
Februari	Pebruari	Kelompok bulan masehi
glukose	glukosa	Kelompok zat
hadis	hadist	Kelompok agama
ijazah	ijasah	Kelompok pendidikan
jumat	Jum'at	Kelompok hari
jupiter	yupiter	Kelompok benda luar angkasa
kurma	korma	Kelompok buah
Kerja sama	kerjasama	Kelompok umum
lever	liver	kelompok bagian tubuh

Kata baku	Kata tidak baku	Kelompok kata
Maskawin	Mas kawin	Kelompok pernikahan
oranye	orange	Kelompok warna
pulpen	polpen	Kelompok benda
rajab	rejeb	Kelompok bulan hijriah
rukuk	Ruku’	Kelompok sholat
sai	Sa’i	Kelompok ibada haji
skore	Sekor/score	Kelompok pertandingan
Suling	seruling	Kelompok alat music
teater	theater	Kelompok gedung/tempat

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil analisis, implementasi dan pengujian yang telah dilakukan, maka dapat di ambil kesimpulan bahwa Aplikasi Pendeteksi Kata Baku dan Tidak Baku, sudah dapat mendeteksi keseluruhan kata sesuai dengan pencocokan yang terdapat pada *database* aplikasi. Untuk lebih jelasnya berikut penjelasannya yaitu :

1. Pada tahap pendeteksi kata, pengguna menginput kata/kalimat. Kemudian kata yang diinput akan mendeteksi kata tidak baku dan kata baku sesuai yang terdapat pada *database* aplikasi.
2. Dari hasil terdeteksinya kata tidak baku, akan ditandai dengan tulisan berwarna merah kemudian akan tampil hasil dari kata tidak baku dan bentuk kata bakunya beserta pengelompokkan katanya.
3. Jika kata yang terdeteksi kata baku maka akan tampil hasil bahwa kata tersebut merupakan kata baku serta tampil kelompok katanya
4. Jika kata yang di input oleh pengguna tidak terdapat didalam *database* aplikasi maka akan tampil hasil berupa kumpulan kata yang tidak terdapat didalam *database* aplikasi.
5. Di dalam *database* kata, hanya terdapat 1.000 data kata yang digunakan, kata yang digunakan didalam *database* aplikasi di kelompokkan yaitu

kelompok pendidikan, kelompok penyakit, kelompok benda, kelompok rumah sakit, kelompok perkantoran, kelompok agama, kelompok olahraga, kelompok buah, kelompok hewan, kelompok anggota tubuh, kelompok ibadah haji, kelompok hari, kelompok tumbuhan, kelompok alat transportasi, kelompok musik, dan lain-lain.

6. Aplikasi ini di buat menggunakan *software Node.js* dan *visual studio code* dengan menggunakan bahasa pemrograman *javascript*.
7. Aplikasi ini bersifat *offline*, sehingga kata yang di deteksi pada kalimat terbatas sebab hanya mengikut dari kata yang di input pada *database* saja.

B. Saran

Berdasarkan hasil penelitian dan pembuatan aplikasi pendeteksi kata baku dan tidak baku, penulis dapat memberikan beberapa saran dari penelitian ini adalah sebagai berikut :

1. Agar pengembangan aplikasi ke depan, dalam proses pendeteksian menggunakan imbuhan awalan dan akhiran.
2. Pada penelitian ini dapat dikembangkan dengan menambah *database* kata sehingga pada proses pendeteksian, akan banyak kata yang terbaca oleh aplikasi.
3. Pada penelitian ini, aplikasi yang di rancang dapat di kembangkan dengan menambahkan atau meningkatkan kualitas dari segi fitur aplikasi dan juga metode pendeteksian yang digunakan.

DAFTAR PUSTAKA

ABIMANTRA, M. L. (2021). *Implementasi Google Speech Api pada Aplikasi Pengenalan Kata Baku untuk Siswa Tingkat Sekolah Dsar Berbasis Android*. UNIVERSITAS BUMIGORA.

Anak Toraja. (2019, 06). Retrieved Juli 06, 2021, from Pengertian UML : Jenis-jenis Diagram UML, Simbol dan Contohnya: https://www.anaktoraja.com/2019/06/pengertian-uml-diagram.html

Fahrudin, T. M., Sa'diyah, I., Latipah, L., Illah, I. Z., Lirna, C. C. B., & Acarya, B. S. (2021). Aplikasi Pendeteksi Kesalahan Ejaan Bahasa Indonesia pada Karya Ilmiah Bidang Ilmu Komputer Menggunakan KEBI 1.0 Checker. *Prosiding Seminar Nasional Informatika Bela Negara*, *2*, 64–72.

Hartina, T., & Agustin, A. (2020). Pendeteksi Kesalahan Pengetikan Kata Non Baku pada Karya Tulis Menggunakan Metode N-Gram. *Jurnal Informatika*, *7*(1), 77–84.

Marlina, M., & Masnur, M. (2021). Aplikasi E-Learning Siswa Smk Berbasis Web. *Jurnal Sintaks Logika*, *1*(1), 8–17.

Nasution, S. G. F. (2019). *RANCANG BANGUN SISTEM INFORMASI PERPUSTAKAAN MENGGUNAKAN WEB SERVICE PADA JURUSAN TEKNIK KOMPUTER POLSRI*. Politeknik Negeri Sriwijaya.

Pradana, A. G. (2019). Rancang Bangun Game Edukasi "AMUDRA" Alat Musik Daerah Berbasis Android. *Prosiding Seminar Nasional Teknologi Informasi Dan Komunikasi (SENATIK)*, *2*(1), 49–53.

Qulub, M., Utami, E., & Sunyoto, A. (2020). Stemming Kata Berimbuhan Tidak Baku Bahasa Indonesia Menggunakan Algoritma Jaro-Winkler Distance. *Creative Information Technology Journal*, *5*(4), 254–263.

Susilawati, S. (2019). *Perbandingan akurasi algoritma Levensthein Distance dengan algoritma Cosine Similarity untuk koreksi penulisan dalam kalimat*. UIN Sunan Gunung Djati Bandung.

Syamsiah, S. (2019). Perancangan Flowchart dan Pseudocode Pembelajaran Mengenal Angka dengan Animasi untuk Anak PAUD Rambutan. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, *4*(1), 86–93.

Tandra, P. B., Andjarwirawan, J., & Yulia, Y. (2017). Aplikasi Tablet Interaktif In-Room Guest Services Bebasis Android Pada Hotel Grand Duta Palu. *Jurnal Infra*, *5*(2), 103–106.

Wiguna, P. D. A., Swastika, I. P. A., & Satwika, I. P. (2018). Rancang bangun aplikasi point of sales distro management system dengan menggunakan

framework react native. *Jurnal Nasional Teknologi Dan Sistem Informasi*, *4*(3), 149–159.