

**APLIKASI PERBAIKAN KATA DALAM BAHASA
DAERAH GORONTALO MENGGUNAKAN
ALGORITMA LEVENSHTTEIN DISTANCE**

Oleh

ADITYA S HUMALANGGI

T31 16 089

SKRIPSI



PROGRAM SARJANA

TEKNIK INFORMATIKA

UNIVERSITAS ICHSAN GORONTALO

GORONTALO

2023

**APLIKASI PERBAIKAN KATA DALAM BAHASA
DAERAH GORONTALO MENGGUNAKAN
ALGORITMA LEVENSHTAIN DISTANCE**

Oleh

ADITYA S HUMALANGGI

T31 16 089

SKRIPSI

**Untuk memenuhi salah satu syarat ujian
Guna memperoleh gelar Sarjana
Program Studi Teknik Informatika**



**PROGRAM SARJANA
TEKNIK INFORMATIKA
UNIVERSITAS ICHSAN GORONTALO
GORONTALO
2023**

PERSETUJUAN SKRIPSI
APLIKASI PERBAIKAN KATA DALAM BAHASA
DAERAH GORONTALO MENGGUNAKAN
ALGORITMA LEVENSHTTEIN DISTANCE

Oleh
ADITYA S HUMALANGGI
T3116089

SKRIPSI

Untuk memenuhi salah satu syarat ujian guna memperoleh gelar sarjana program
studi Teknik Informatika, ini telah disetujui oleh tim pembimbing

Gorontalo, Mei 2023

Pembimbing I

Pembimbing II

Hastuti Dalai,M.Kom

NIDN : 0918038803

Kartika Chandra Pelangi,M.Kom

NIDN : 0916038304

PENGESAHAN SKRIPSI

APLIKASI PERBAIKAN KATA DALAM BAHASA DAERAH GORONTALO MENGGUNAKAN ALGORITMA LEVENSHTTEIN DISTANCE

(Studi Kasus : Universitas Ichsan Gorontalo)

Oleh

ADITYA S HUMALANGGI

T3116089

Diperiksa oleh Panitia Ujian Strata Satu (S1)

Universitas Ichsan Gorontalo

Gorontalo, Mei 2023

- 
1. Ketua Penguji
Yasin Aril Mustofa, M.Kom
 2. Anggota
Sudirman S. Panna, M.Kom
 3. Anggota
Muis Nanja, M.Kom
 4. Anggota
Hastuti Dalai, M.Kom
 5. Anggota
Kartika Chandra Pelangi, M.Kom

Mengetahui

Dekan Fakultas Ilmu Komputer

Ketua Program Studi

Irvan Abraham Salihi, S.Kom, M.Kom
NIDN. 09028028101

Sudirman S.Panna, M.Kom
NIDN. 0924038205

PERNYATAAN SKRIPSI

Dengan ini saya menyatakan bahwa:

1. Karya tulis (Skripsi) saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (Sarjana) baik di Universitas Ichsan Gorontalo maupun di perguruan tinggi lainnya.
2. Karya tulis (Skripsi) saya ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan dari Tim Pembimbing.
3. Dalam karya tulis (Skripsi) saya ini tidak terdapat karya atau pendapat yang telah dipublikasikan orang lain, kecuali secara tertulis dicantumkan sebagai acuan/sitasi dalam naskah dan dicantumkan pula dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma-norma yang berlaku di Universitas Ichsan Gorontalo.

Gorontalo, April 2023
Yang Membuat Pernyataan,

Aditya S Humalangi

ABSTRACT

ADITYA SHUMALANGGI . T31160 89 . APPLICATION OF WORD CORRECTION IN REGIONAL LANGUAGES OF GORONTALO USING LEVENSHTein DISTANCE

Gorontalo language or Hulondalo language is a regional language used by residents of Gorontalo Province. The use of regional languages has experienced a decrease in use in everyday communication. Moreover, if in writing there are several words or spellings that cannot be understood and there are several wrong words in writing or mistakes when typing which causes the words or spelling to change. Preserving regional languages is very necessary in the midst of increasingly rapid technological developments. To simplify the word search process, in this study the researchers designed an Android- based word improvement application in Gorontalo using the Levenshtein Distance Algorithm. This algorithm is considered as the most efficient algorithm in general applications. The Gorontalo word repair application is designed to perform a word search function in Gorontalo and display the correct word in Gorontalo. The results of this study were obtained from the process of matching the words inputted by the user with the words in the database and displaying the appropriate and correct word results.

Keywords: Correction of Gorontalo Words, Typo, Android, Levenshtein Distance

ABSTRAK

ADITYA S HUMALANGGI. T3116089. APLIKASI PERBAIKAN KATA DALAM BAHASA DAERAH GORONTALO MENGGUNAKAN LEVENSHTTEIN DISTANCE

Bahasa Gorontalo atau bahasa Hulondalo merupakan bahasa daerah yang digunakan oleh penduduk Provinsi Gorontalo. Penggunaan bahasa daerah telah mengalami penurunan penggunaan dalam bahasa komunikasi sehari-hari. Terlebih lagi jika dalam tulisan terdapat beberapa kata atau ejaan yang belum bisa dipahami dan terdapat beberapa kata yang salah dalam menulis ataupun kesalahan saat mengetik yang menyebabkan kata atau ejaan dapat berubah. Melestarikan bahasa daerah sangat perlu ditengah perkembangan teknologi yang semakin pesat. Untuk mempermudah proses pencarian kata maka dalam penelitian ini peneliti merancang sebuah aplikasi perbaikan kata dalam bahasa Gorontalo yang berbasis *Android* dengan menggunakan Algoritma Levenshtein Distance. Algoritma ini dianggap sebagai algoritma yang paling efisien pada aplikasi umum. Aplikasi Perbaikan kata dalam bahasa gorontalo yang dirancang dapat melakukan fungsi pencarian kata dalam Bahasa Gorontalo dan dapat menampilkan kata yang benar dalam bahasa Gorontalo. Hasil penelitian ini didapatkan dari proses pencocokan kata yang diinput user dengan kata didalam database dan menampilkan hasil kata yang sesuai dan benar.

Kata Kunci: *Perbaikan Kata Bahasa Gorontalo, Typo, Android, Levenshtein Distance*

KATA PENGANTAR

Alhamdulillah, penulis dapat menyelesaikan skripsi ini dengan Judul: **“Aplikasi Perbaikan Kata Dalam Bahasa Daerah Gorontalo Menggunakan Algoritma Levenshtein Distance”**, untuk memenuhi salah satu syarat penyusunan Skripsi Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.

Penulis menyadari sepenuhnya bahwa usulan penelitian ini tidak mungkin terwujud tanpa bantuan dan dorongan dari berbagai pihak, baik bantuan moril maupun materil. Untuk itu, dengan segala keikhlasan dan kerendahan hati, penulis mengucapkan banyak terima kasih dan penghargaan yang setinggi-tingginya kepada:

1. Ibu Dr. Juriko Abdusammad, M.Si, selaku Ketua Yayasan Pengembangan Ilmu Pengetahuan dan Teknologi(YPIPT) Ichsan Gorontalo;
2. Bapak Dr.Abdul Gaffar La Tjokke, M,Si, selaku Rektor Universitas Ichsan Gorontalo;
3. Bapak Irvan Abraham Salihi, S.Kom, M.Kom, selaku Dekan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
4. Bapak Sudirman Melangi, S.Kom, M.Kom, selaku Pembantu Dekan I Bidang Akademik Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
5. Ibu Irma Surya Kumala Idris, S.Kom, M.Kom, selaku Pembantu Dekan II Bidang Administrasi Umum dan Keuangan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
6. Sudirman S. Panna, S.Kom, M.Kom, selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
7. Ibu Hastuti Dalai,M.Kom, selaku Pembimbing II;
8. Ibu Kartika Chandra Pelangi,M.Kom, selaku Pembimbing II;
9. Bapak dan Ibu Dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan berbagai disiplin ilmu kepada penulis;
10. Kedua Orang Tua saya yang tercinta (Ayah Harun D Djunu Dan Ibu Rabia Abdullah), atas segala kasih sayang, jerih payah dan doa restunya dalam membesarkan dan mendidik penulis;

11. Rekan-rekan seperjuangan yang telah banyak memberikan bantuan dan dukungan moral yang sangat besar kepada penulis;
12. Untuk Game Arena Of Valor dan Anime yang telah menemani dan menghibur penulis;
13. Kepada semua pihak yang ikut membantu dalam penyelesaian Skripsi ini yang tak sempat penulis sebutkan satu-persatu.

Semoga Allah SWT melimpahkan balasan atas jasa-jasa mereka kepada kami. Penulis menyadari sepenuhnya bahwa apa yang telah dicapai ini masih jauh dari kesempurnaan dan masih banyak terdapat kekurangan. Oleh karena itu, penulis sangat mengharapkan adanya kritik dan saran yang konstruktif. Akhirnya penulis berharap semoga hasil yang telah dicapai ini dapat bermanfaat bagi kita semua, Aamiin.

Gorontalo, Mei 2023

Penulis

DAFTAR ISI

PERSETUJUAN SKRIPSI.....	ii
PENGESAHAN SKRIPSI.....	iii
PERNYATAAN SKRIPSI.....	iv
ABSTRACT	v
ABSTRAK	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Identifikasi masalah.....	3
1.3 Rumusan masalah.....	3
1.4 Tujuan penelitian	3
1.5 Manfaat Penelitian.....	3
1.5.1 Manfaat Teoritis	3
1.5.2 Manfaat Praktis	3
BAB II LANDASAN TEORI	4
2.1 Tinjauan studi	4
2.2 Tinjauan Pustaka	7
2.2.1 <i>Typographical Error</i>	7
2.2.2 Kesalahan penulisan	7
2.2.3 <i>Spelling correction</i>	7
2.2.4 <i>Levenshtein distance</i>	8
2.2.5 Android	9
2.2.6 <i>Android Software Development Kit Manajer (SDK)</i>	9
2.2.7 Analisis sistem	9
2.2.8 Desain sistem	10
2.2.9 Perangkat lunak pendukung.....	15

2.2.10 Pengujian Sistem.....	15
2.3 Kerangka pikir	17
BAB III METODE PENELITIAN	18
3.1 Jenis, Metode, Subjek, Objek, Waktu Dan Lokasi Penelitian.....	18
3.2 Pengumpulan Data	18
3.3 Pengembangan Sistem.....	18
3.3.1 Analisis sistem	20
3.3.2 Desain Sistem.....	20
3.3.3 Kontruksi Sistem.....	21
3.3.4 Pengujian Sistem.....	21
BAB IV HASIL PENELITIAN.....	22
4.1 Hasil Pengumpulan Data	22
4.1.1 Penerapan Metode.....	23
3.1 Hasil Pengembangan Sistem	28
3.2 <i>Activity Diagram Login</i>	28
3.3 <i>Activity Diagram Dashboard</i>	24
3.4 <i>Activity Diagram Data Kata</i>	24
3.5 <i>Activity Diagram Data User</i>	25
3.6 <i>Sequence Diagram Login Admin</i>	25
3.7 <i>Sequence Diagram Dashboard</i>	26
3.8 <i>Sequence Diagram Data Kata</i>	26
4.10 <i>Sequence Diagram User</i>	27
4.11 Arsitektur Sistem.....	27
4.12 <i>Interface Design</i>	28
4.12.1 Mekanisme User	28
4.12.2 Mekanisme Navigasi <i>Home</i>	28
4.12.3 Mekanisme <i>Login</i>	29
4.12.4 Mekanisme <i>Input Data Kata</i>	29
4.12.5 Mekanisme Edit Data Kata	29
4.12.6 Mekanisme Tambah Data User.....	30
4.12.7 Mekanisme Edit Data User	30
4.12.8 Mekanisme <i>Output</i>	31

4.13 Data Desain	31
4.13.1 Struktur Data.....	31
4.14 Hasil Pengujian Sistem.....	32
4.14.1 Pengujian <i>White Box</i>	32
4.14.2 <i>Flowchart Diagram</i>	34
4.14.3 Pengujian <i>White Box</i>	35
4.14.4 Pengujian CC Pada Pengujian <i>White Box</i>	36
4.14.5 Pengujian CC Pada Pengujian <i>White Box</i>	36
4.14.6 Pengujian <i>Black Box</i>	36
BAB V HASIL DAN PEMBAHASAN	39
5.1 Pembahasan Sistem	39
5.1.1 Tampilan Halaman <i>Login</i>	39
5.1.2 Tampilan Halaman Dashboard Admin	40
5.1.3 Tampilan Halaman Data Kata.....	40
5.1.4 Tampilan Halaman Tambah Data Kata	41
5.1.5 Tampilan Halaman Edit Data Kata.....	41
5.1.6 Tampilan Halaman Data User.....	42
5.1.7 Tampilan Halaman Tambah Data User	43
5.1.8 Tampilan Halaman Edit Data User.....	44
5.1.9 Tampilan Halaman User	45
5.1.10 Tampilan Data Profil.....	45
5.1.11 Tampilan Data Kata	46
BAB VI KESIMPULAN DAN SARAN.....	47
6. 1 Kesimpulan.....	47
6. 2 Saran	47
DAFTAR PUSTAKA	48
.....	50

DAFTAR GAMBAR

Gambar 3. 1	Sistem yang diusulkan	19
Gambar 4. 1	Use Case Diagram Aplikasi Perbaikan Kata Bahasa Gorontalo	28
Gambar 4. 2	Activity Diagram Login.....	28
Gambar 4. 3	Activity Diagram Dashboard.....	24
Gambar 4. 4	Activity Diagram Data Kata	24
Gambar 4. 5	Activity Diagram Data User	25
Gambar 4. 6	Sequence Diagram Login Admin	25
Gambar 4. 7	Sequence Diagram Dashboard.....	26
Gambar 4. 8	Sequence Diagram Data Kata.....	27
Gambar 4. 9	Sequence Diagram User	27
Gambar 4. 10	Mekanisme Navigasi <i>Home</i> Admin.....	28
Gambar 4. 11	Mekanisme Navigasi <i>Home</i> User	28
Gambar 4. 12	Mekanisme <i>Login</i>	29
Gambar 4. 13	Mekanisme <i>Input</i> Data kata.....	29
Gambar 4. 14	Mekanisme Edit Data Kata.....	29
Gambar 4. 15	Mekanisme Tambah Data User	30
Gambar 4. 16	Mekanisme Edit Data User.....	30
Gambar 4. 17	Mekanisme <i>Output</i> Perbaikan Kata.....	31
Gambar 4. 18	<i>Flowchart</i>	34
Gambar 4. 19	<i>Flowgraph</i>	35
Gambar 5. 1	Tampilan Halaman Login.....	39
Gambar 5. 2	Tampilan Halaman Dashboard Admin	40
Gambar 5. 3	Tampilan Halaman Data Kata	40
Gambar 5. 4	Tampilan Halaman Data Kata	41
Gambar 5. 5	Tampilan Halaman Edit Data Kata.....	41
Gambar 5. 6	Tampilan Halaman Data User	42
Gambar 5. 7	Tampilan Halaman Tambah Data User	43
Gambar 5. 8	Tampilan Halaman Edit Data User.....	44
Gambar 5. 9	Tampilan Home User.....	45
Gambar 5. 10	Halaman Data Profil	45
Gambar 5. 11	Halaman Data Kata.....	46

DAFTAR TABEL

Tabel 2. 1 Format tinjauan studi.....	4
Tabel 2. 2 Kata salah dan kemungkinan kata	8
Tabel 2. 3 simbol – simbol Use case diagram	10
Tabel 2. 4 Simbol - Simbol Activity diagram	12
Tabel 2. 5 Simbol - Simbol Sequence diagram	13
Tabel 2. 6 Simbol - Simbol Class Diagram	14
Tabel 2. 7 Perangkat Pendukung	15
Tabel 4. 1 Hasil Pengumpulan Data	22
Tabel 4. 2 Mekanisme User	28
Tabel 4. 3 Tabel Kata	31
Tabel 4. 4 Tabel User	32
Tabel 4. 5 Tabel Pengujian <i>Black Box</i>	37

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di Indonesia terdapat beberapa provinsi yang terpencar dari Sabang sampai Merauke, yang masing-masing memiliki rumpun bahasa tersendiri. Indonesia terdiri dari berbagai suku, budaya, agama, dan bahasa Daerah [1]. Bahasa daerah termasuk salah satu unsur yang memiliki peranan penting dalam kebudayaan, antara lain menjadi kebanggaan suatu daerah, identitas daerah, dan media penghubung untuk komunikasi di keluarga dan masyarakat. Hal ini diperkuat dengan adanya Undang-Undang Dasar 1945 mengenai pendidikan dan kebudayaan pasal 32 ayat 2 yang dinyatakan bahwa bahasa daerah termasuk dalam kekayaan budaya nasional yang dihormati dan dipelihara oleh negara [2]. Bahasa Gorontalo merupakan salah satu dari sembilan daerah adat di Indonesia, juga memiliki bahasa daerah sebagai ciri khas daerah yang disebut bahasa Gorontalo. Sebagai bahasa daerah, tentu saja bahasa ini menjadi alat komunikasi warga masyarakat Gorontalo, dari anak-anak hingga orang tua [3].

Bahasa Gorontalo atau bahasa hulondalo merupakan bahasa daerah yang digunakan oleh penduduk provinsi gorontalo. Sekarang ini banyak masyarakat Gorontalo dikalangan orang tua sudah jarang menggunakan Bahasa Gorontalo untuk berkomunikasi sehari – hari. Terlebih lagi dikalangan muda sudah tidak lagi menggunakan bahasa daerah Gorontalo, penyebab utama menghilangnya bahasa daerah Gorontalo dikarenakan pengaruh bahasa pergaulan yang bebas, hal ini seperti yang terlihat dikalangan generasi muda sekarang. Generasi muda sekarang saat berkomunikasi sehari-hari hanya menggunakan Bahasa pergaulan yang bebas, mereka malu menggunakan Bahasa daerah mereka masing-masing. Adapun sebagian dari mereka hanya mampu memahami apa yang diucapkan orang lain, tetapi sangat sulit menggunakannya secara lisan ataupun tulisan. Terlebih lagi jika dalam tulisan terdapat beberapa kata atau ejaan yang belum bisa dipahami dan terdapat beberapa kata yang salah dalam menulis ataupun kesalahan saat mengetik yang menyebabkan kata atau ejaan dapat berubah.

Kesalahan pengetikan yang sering terjadi dapat menyebabkan kata atau ejaan berubah menjadi tidak sesuai. Beberapa faktor yang menyebabkan terjadinya kesalahan pengetikan adalah letak huruf pada *keyboard* yang berdekatan, kesalahan karena *slip* pada tangan atau jari, kesalahan yang disebabkan oleh ketidak sengajaan. Oleh karena itu diperlukan suatu sistem yang mampu mengidentifikasi kesalahan pengetikan agar sesuai dengan penulisan yang benar. Sehingga kesalahan-kesalahan penulisan dapat dihindari. Salah satu pendekatan yang dapat dilakukan untuk mengidentifikasi kesalahan pengetikan adalah metode pencocokan string [4] .

Adapun beberapa hasil penelitian sebelumnya tentang aplikasi perbaikan kata yang sudah dilakukan seperti, hasil penelitian Braddley, Fachrurrozi, Novi yaitu kesalahan penulisan bisa saja terjadi tanpa disengaja, hal ini berpengaruh kepada pembaca karena informasi yang didapat salah. Sistem pengoreksi ejaan kata secara otomatis mampu mengurangi tingkat kesalahan penulisan [5].

Dan hasil penelitian dari rochmawati dan kusumaningrum yaitu Kesalahan pengetikan mengakibatkan kata atau ejaan berubah menjadi kata yang digunakan tidak sesuai. Hal tersebut dapat ditangani dengan mengembangkan sistem untuk mengidentifikasi kesalahan pengetikan. Metode *approximate string matching* merupakan salah satu metode yang banyak diterapkan untuk mengidentifikasi kesalahan pengetikan dengan berbagai jenis algoritma pencarian string [4] .

Sekarang ini perkembangan teknologi semakin meningkat, dengan meningkatnya teknologi tersebut dapat menyelesaikan dan memudahkan pekerjaan manusia. Khususnya dalam perkembangan *Smartphone* yang menanamkan *system android* dan memiliki daya tarik tersendiri kepada penggunanya, karena memiliki beragam fitur yang ditampilkan memenuhi kebutuhan. *Android* merupakan salah satu *flatfrom open source* yang memberikan kebebasan pada *developer* untuk mengembangkan sebuah aplikasi tersendiri [6] .

Berdasarkan pemaparan dan penjelasan diatas, maka penulis mengambil judul “ **Aplikasi perbaikan kata dalam bahasa daerah gorontalo** ” yang insyaallah dapat membantu masyarakat dan pelajar gorontalo dalam memahami penulisan yang benar dalam bahasa daerah gorontalo.

1.2 Identifikasi masalah

Dari uraian diatas, maka identifikasi masalah yang diperoleh pada pengembangan aplikasi perbaikan kata dalam bahasa daerah gorontalo adalah masyarakat gorontalo terutama kalangan muda yang kurang memahami penulisan bahasa daerah gorontalo.

1.3 Rumusan masalah

Berdasarkan penjelasan diatas, maka rumusan masalah dalam penelitian ini yaitu, bagaimana cara agar masyarakat gorontalo terutama kalangan muda dapat memahami penulisan yang benar dalam Bahasa Gorontalo.

1.4 Tujuan penelitian

Berdasarkan rumusan masalah, maka tujuan dalam penelitian ini yaitu, diharapkan penelitian ini dapat membantu para masyarakat Gorontalo terutama kalangan muda agar mudah memahami penulisan yang benar dalam Bahasa gorontalo.

1.5 Manfaat Penelitian

Berdasarkan tujuan penelitian yang hendak ingin dicapai, maka penelitian ini harus mempunyai manfaat dalam pendidikan baik secara langsung maupun tidak langsung. Adapun manfaat penelitian ini adalah sebagai berikut :

1.5.1 Manfaat Teoritis

Penelitian ini diharapkan agar dapat mengembangkan ilmu pengetahuan dibidang teknologi informasi dan sebagai referensi pada penelitian selanjutnya yang berhubungan dengan aplikasi *mobile*.

1.5.2 Manfaat Praktis

Dapat menambah wawasan dan pengalaman dalam pembuat aplikasi mobile dan sebagai bahan pertimbangan dalam menyusun proposal / skripsi untuk memperoleh gelar sarjana.

BAB II

LANDASAN TEORI

2.1 Tinjauan studi

Tabel 2. 1 Format tinjauan studi

NO	PENELITI	JUDUL	TAHUN	METODE	HASIL
1	Winardi, Agung	Aplikasi Perbaikan Ejaan Pada Karya Tulis Ilmiah Di Program Studi Teknik Informatika Dengan Menerapkan Algoritma Levensthei n Distance	2017	<i>Levensthei n-Distance</i>	Sistem yang telah dibuat dalam penelitian ini adalah sistem yang dibuat dapat melakukan koreksi dan pemberian saran perbaikan kesalahan ejaan pada karya tulis ilmiah dengan menggunakan algoritma <i>Levenshtein Distance</i>

2	Hadi, S Sunda, A Margareta, M	Perangkat Lunak Identifikasi dan Koreksi Kesalahan Berbahasa Indonesia Pada Karya Ilmiah	2016	algoritma Nazief dan Andriani	Hasil pengujian menunjukkan bahwa akurasi proses pengecekan Bahasa Indonesia dengan menggunakan algoritma Nazief dan Adriani masih memiliki beberapa kekurangan, panjang kata dan kerumitan kata juga sangat berpengaruh terhadap waktu dalam proses pengecekan
3	Agustian, Sedy Suryaningru m, Kristien Margi	Aplikasi koreksi kesalahan penulisan kata dalam bahasa inggris dengan menggunak	2019	<i>Rabin - karp</i>	diperoleh hasil bahwa kata yang merupakan kata singkatan akan menghasilkan hasil dan nilai yang berbeda daripada kata

		an algoritma rabin - karp			yang bukan merupakan kata singkatan. Hasil untuk kata kedua dari kata singkatan akan menghasilkan nilai yang berbeda dikarenakan kata tersebut menghasilkan kata yang berbeda atau tidak sama dengan yang seharusnya.
4	Braddley, Muhammad Omar Fachrurrozi, Muhammad Novi, Yusliani	Pengoreksian ejaan Kata berbahasa indonesia menggunakan algoritma levensthein distance	2017	<i>Levensthein-Distance</i>	Pengujian akurasi sebesar 100% dan waktu 23 mili detik pada operasi penghapusan karakter, hasil 96% dan waktu 5 mili detik pada operasi pengubahan karakter dan hasil 93% dan

					waktu 88 mili detik untuk penambahan karakter.
--	--	--	--	--	--

2.2 Tinjauan Pustaka

2.2.1 *Typographical Error*

Kesalahan tipografi adalah kesalahan Terjadi saat mengetik teks Bahkan bisa mengubah arti kata Kata dalam satu kalimat termasuk Kegagalan yang disebabkan oleh kerusakan mekanis atau selip Tangan atau jari, ada konsekuensinya Ketidaktahuan penulis seperti kesalahan ejaan. Kesalahan tipografi mungkin karena, Misalnya dua jari Tombol keyboard yang berdekatan pada waktu bersamaan [7].

2.2.2 Kesalahan penulisan

Kesalahan penulisan merupakan suatu *human error* (kesalahan manusia) yang cukup sering dilakukan oleh seseorang dalam melakukan sebuah penulisan. Kesalahan penulisan sendiri dibagi menjadi dua yakni kesalahan penulisan yang dilakukan secara sengaja dan tidak sengaja. Untuk kesalahan penulisan yang tidak disengaja atau biasa disebut dengan kesalahan tipografi (*typographical error*), sedangkan untuk kesalahan penulisan yang dilakukan dengan sengaja karena kesalahan tersebut bersifat konsisten yang dilakukan secara berulang-ulang seperti kata “barang disingkat brg”, “bagus disingkat bgs”, serta beberapa kata lain yang sering dituliskan secara tidak baku [8].

2.2.3 *Spelling correction*

Spelling Correction adalah sistem yang dapat mengoreksi kesalahan pengejaan. Kesalahan yang mungkin terjadi karena kurang atau lebihnya karakter, atau juga dapat terjadi karena terdapat karakter yang tidak disesuaikan. Contoh kesalahan yang sering terjadi di tunjukan pada tabel 2.2 kata salah dan kemungkinan kata [5].

Tabel 2. 2 Kata salah dan kemungkinan kata

Kata yang salah	Kemungkinan kata	Penyebab kesalahan
Kiash	Kisah,kasih	Kesalahan urutan ‘i,a,s’
	Kias	Terdapat huruf ‘h’
Keuar	Keluar,	Kurang huruf ‘l’
Temn	Teman	Kurang huruf ‘a’
	Tema,temu	Menghapus huruf ‘n’ dan menggantinya dengan huruf ‘a’ dan ‘u’

2.2.4 *Levensthein distance*

Jarak Levenshtein adalah Ukur (metrik) untuk menghitung kuantitas Perbedaan antara dua kata menentukan jarak antara dua kata berdasarkan operasi perubahan minimum yang mengubah kata A ke kata B. Algoritma ini dapat melakukan tiga jenis operasi, yaitu:

1. Operasi pengubahan karakter
Operasi menukar sebuah karakter dengan karakter lain contohnya : “yang” menjadi “yan”. Karakter ‘m’ diganti dengan ‘n’.
2. Operasi penambahan karakter
Operasi menambah karakter kedalam sebuah kata contohnya : “sya” menjadi “saya”. Penambahan karakter tidak bisa dilakukan diawal, diakhir atau disisipkan dalam kata.
3. Operasi penghapusan karakter
Operasi penghapusan karakter dilakukan dengan menghilangkan sebuah karakter dalam kata contohnya : “lagio” menjadi “lagi” dengan menghilangkan kata terakhir “o” [5] .

2.2.5 Android

Android adalah sistem operasi berbasis Linux untuk perangkat seluler seperti ponsel pintar. Untuk tujuan ini, Android adalah sistem operasi untuk perangkat seluler, yang mencakup sistem operasi, middleware, dan aplikasi. Awalnya, Android adalah sebuah perusahaan, tetapi diakuisisi oleh Google pada tahun 2000. Setelah diterima, sistem Android dikembangkan hanya untuk kebutuhan internal Google dan belum memperoleh lisensi open source [9].

2.2.6 Android Software Development Kit Manajer (SDK)

Software Development Kit (SDK) manager merupakan tools bagi para programmer yang ingin mengembangkan aplikasi berbasis Google Android. Android SDK mencakup seperangkat alat pengembangan yang komprehensif. Android SDK terdiri dari debugger, libraries, handset emulator, dokumentasi, contoh kode, dan tutorial. Saat ini Android sudah mendukung arsitektur x86 pada Linux (distribusi Linux apapun untuk desktop modern), Mac OS X 10.4.8 atau lebih, Windows XP atau Vista. Persyaratan mencakup JDK, Apache Ant dan Python 2.2 atau yang lebih baru, dengan ini pengembang dapat menggunakan teks editor untuk mengedit file Java dan XML serta menggunakan peralatan command line untuk menciptakan, membangun, melakukan debug aplikasi Android dan pengendalian perangkat Android misalnya, reboot, menginstal paket perangkat lunak dengan jarak jauh [10] .

2.2.7 Analisis sistem

Analisis sistem dapat didefinisikan sebagai Sistem lengkap diuraikan menjadi suatu komponen, dengan maksud untuk identifikasi dan evaluasi masalah, kesempatan, hambatan dan antisipasi kebutuhan guna membuat rekomendasi perbaikan [10]. Pengertian lain dari analisis sistem adalah mengamati suatu sistem yang telah bekerja hingga dapat menemukan bagian mana yang bekerja dengan bagus dan tidak, kemudian mencatat keperluan apa saja yang harus dipenuhi saat adanya sistem yang baru. Inti dari pengertian analisis berdasarkan pendapat ahli yang telah disebutkan adalah suatu tahapan untuk menemukan permasalahan dalam sistem sehingga dapat menentukan solusi yang tepat untuk permasalahan tersebut [11].

2.2.8 Desain sistem

Desain sistem terbagi menjadi dua bagian yaitu desain umum dan desain rinci. Desain sistem umumnya persiapan dari desain terinci. Desain umum merupakan identifikasi komponen sistem informasi yang dirancang secara rinci. Sedangkan desain rinci bertujuan untuk pemrogram komputer dan ahli teknik lainnya yang akan diimplementasikan ke dalam sistem. Pengertian lain dari desain sistem adalah upaya untuk membangun sebuah sistem yang memberikan kepuasan akan spesifikasi kebutuhan fungsional, memenuhi target, memenuhi kebutuhan secara performansi maupun penggunaan sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu dan perangkat [11] .

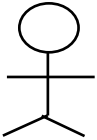
2.2.8.1 Unified Modelling Language (UML)

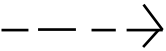
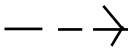
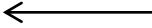
UML adalah salah satu tool permodelan sistem berorientasi objek yang digunakan untuk memperlihatkan banyak pandangan terhadap suatu sistem yang akan dibangun. UML terbagi menjadi beberapa model diagram, dimana setiap diagram menggambarkan masing-masing aktifitas dari system yang berjalan pada aplikasi. Adapun diagram yang akan dipergunakan oleh penulis yaitu terdiri dari Use case Diagram, Sequence Diagram, Activity Diagram, dan Class Diagram [12].

1. Use case diagram

Use case diagram yaitu salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor, use case diagram juga dapat men-deskripsikan tipe interaksi antara si pemakai sistem dengan sistemnya.

Tabel 2. 3 simbol – simbol Use case diagram

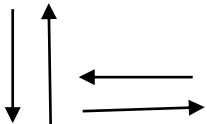
Simbol	Nama	Keterangan
	Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan sebuah use case
	Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan

		mempengaruhi sebuah elemen yang bergantung padanya elemen yang tidak mandiri
	Generalization	Hubungan dimana suatu objek anak berbagi perilaku dan struktur data dari suatu objek yang ada di atasnya objek induk
	Include	Menspesifikasi bahwa <i>use case</i> sumber secara eksplisit
	Extend	Menspesifikasi 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	Menspesifikasi paket yang menampilkan sistem secara terbatas
	Use case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
	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 komputasi

2. Activity diagram

Activity diagram yaitu salah satu jenis diagram pada UML yang dapat memodelkan proses-proses apa saja yang ada pada sebuah sistem.

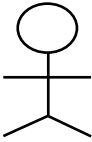
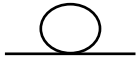
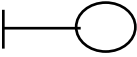
Tabel 2. 4 Simbol - Simbol Activity diagram

Simbol	Nama	Keterangan
	Initial State	Proses dimulai pertama kali didalam activity
	Activity	Aktivitas yang terjadi didalam activity
	Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
	Decision	Menggambarkan cabang suatu keputusan
	Line connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya
	Final State	Proses terakhir didalam activity

3. Sequence diagram

Sequence diagram yaitu salah satu jenis diagram pada UML yang menjelaskan interaksi objek yang berdasarkan urutan waktu, sequence diagram juga dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada use case diagram.

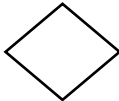
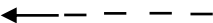
Tabel 2. 5 Simbol - Simbol Sequence diagram

Simbol	Nama	Keterangan
	Actor	Menggambar orang yang sedang berinteraksi dengan sebuah sistem
	Entity Class	Menggambarkan hubungan yang akan dilakukan
	Boundary Class	Menggambarkan sebuah gambaran dari form yang ada
	Control Class	Menggambarkan penghubung antara boundary dengan tabel
	A Focus of Control & A Life Line	Menggambarkan tempat mulai dan berakhirnya sebuah message atau pesan
	A message	Menggambarkan sebuah pengiriman pesan

4. Class diagram

Class diagram yaitu salah satu jenis diagram pada UML yang dapat digunakan untuk menampilkan paket-paket maupun kelas-kelas yang ada pada suatu sistem yang nantinya dapat digunakan. Jadi diagram ini dapat memberikan sebuah gambaran mengenai sistem maupun relasi-relasi yang terdapat pada sistem tersebut.

Tabel 2. 6 Simbol - Simbol Class Diagram

Simbol	Nama	Keterangan
	Generalization	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari suatu objek yang ada di atasnya objek induknya (ancestor)
	Nary Association	untuk menghindari asosiasi dengan lebih dari 2 objek
	Class	Himpunan dari berbagai objek-objek yang berbagi atribut serta operasi yang sama
	Collaboration	Dekskripsi dari urutan aksi aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
	Realization	Sebuah operasi yang dapat benar – benar dilakukan oleh suatu objek
	Dependency	Sebuah hubungan dimana perubahan yang dapat terjadi pada suatu elemen mandiri(independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
	Association	Sebuah objek yang dapat menghubungkan antara objek satu dengan objek lainnya

2.2.9 Perangkat lunak pendukung

Tabel 2. 7 Perangkat Pendukung

NO	Perangkat Lunak	Kegunaan
1	Xampp	Aplikasi untuk membuat data base.
2	Android Studio	Tools untuk membuat Bahasa pemrograman mobile application atau android
3	Jdk	merupakan perangkat lunak yang digunakan untuk manajemen dan membangun berbagai aplikasi Java
4	Php	Bahasa pemrograman untuk membuat web service nya

2.2.10 Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui apakah perangkat yang ada pada sistem sudah dapat diaplikasikan dengan baik dan benar sebelum dilakukan pemasaran.

2.2.10.1 Black box testing

Pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program untuk mengetahui apakah fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Metode BlackboxTesting merupakan salah satu metode yang mudah digunakan karena hanya memerlukan batas bawah dan batas atas dari data yang diharapkan, Estimasi banyaknya data uji dapat dihitung melalui banyaknya field data entri yang akan diuji, aturan entri yang harus dipenuhi serta kasus batas atas dan batas bawah yang memenuhi. Dan dengan metode ini dapat diketahui jika fungsionalitas masih dapat menerima masukan data yang tidak diharapkan maka menyebabkan data yang disimpan kurang valid [13].

2.2.10.2 White box testing

Pengujian perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi masukan dan keluaran yang sesuai dengan spesifikasi kebutuhan. Merawat program bisa dilakukan dengan menyederhanakan source code program sehingga apabila diuji menggunakan White Box Testing lagi, akan menghasilkan Node, Edgeds dan Test Cases yang lebih sedikit dibandingkan dengan pengujian sebelumnya [13].

Rumus untuk menentukan *Cyclomatic Complexity* adalah:

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

$$V(G) = P + 1$$

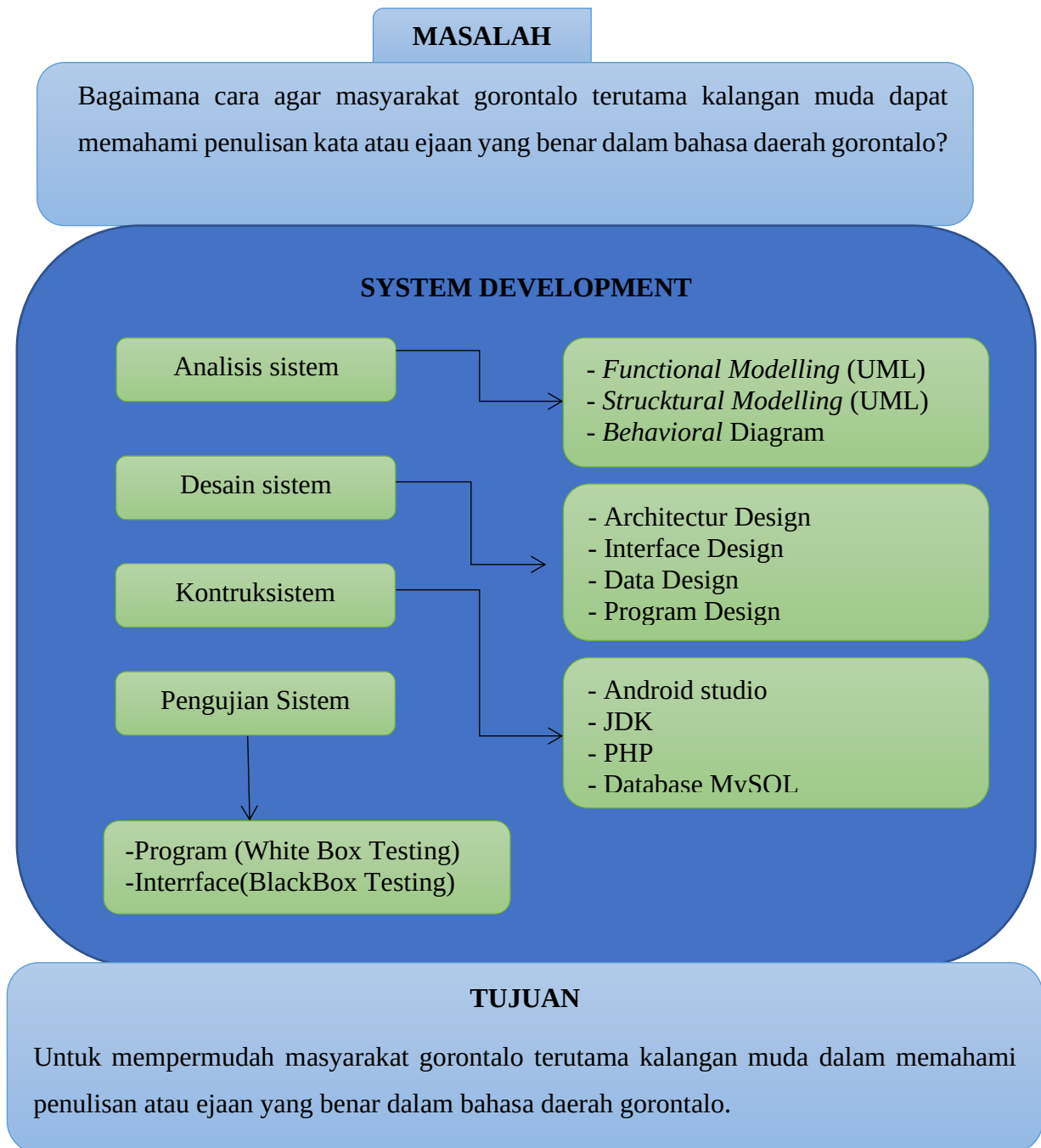
Keterangan:

E= Jumlah edge atau jalur grafik yang ditandakan dengan gambar panah

N= Jumlah node atau simpul grafik alir yang ditandakan dengan gambar lingkaran

V(G)= Besaran kuantitatif terhadap *cyclomatic complexity*

2.3 Kerangka pikir



BAB III

METODE PENELITIAN

3.1 Jenis, Metode, Subjek, Objek, Waktu Dan Lokasi Penelitian

Di pandang dari penerapan tingkat penerapannya maka, penelitian ini merupakan penelitian terapan.

Penelitian ini menggunakan metode penelitian studi kasus. Dengan demikian jenis penelitian ini adalah penelitian *deskriptif*. **Subjek penelitian ini adalah Aplikasi Perbaikan Kata Dalam Bahasa Daerah Gorontalo Menggunakan Algoritma Levenshtein Distance.** Penelitian ini dimulai dari bulan desember 2019 s/d maret 2020

3.2 Pengumpulan Data

Untuk mengumpulkan data digunakan 2 (dua) jenis data yaitu data primer dan sekunder. Data primer yaitu data yang berasal dari penelitian lapangan dan data sekunder berasal dari penelitian kepustakaan.

1. Penelitian data primer

Metode pengumpulan data yang digunakan dalam penelitian ini adalah Studi Literatur dimana pengumpulan data dilakukan dengan cara mengumpulkan literatur, jurnal, paper yang ada kaitannya dengan judul penelitian.

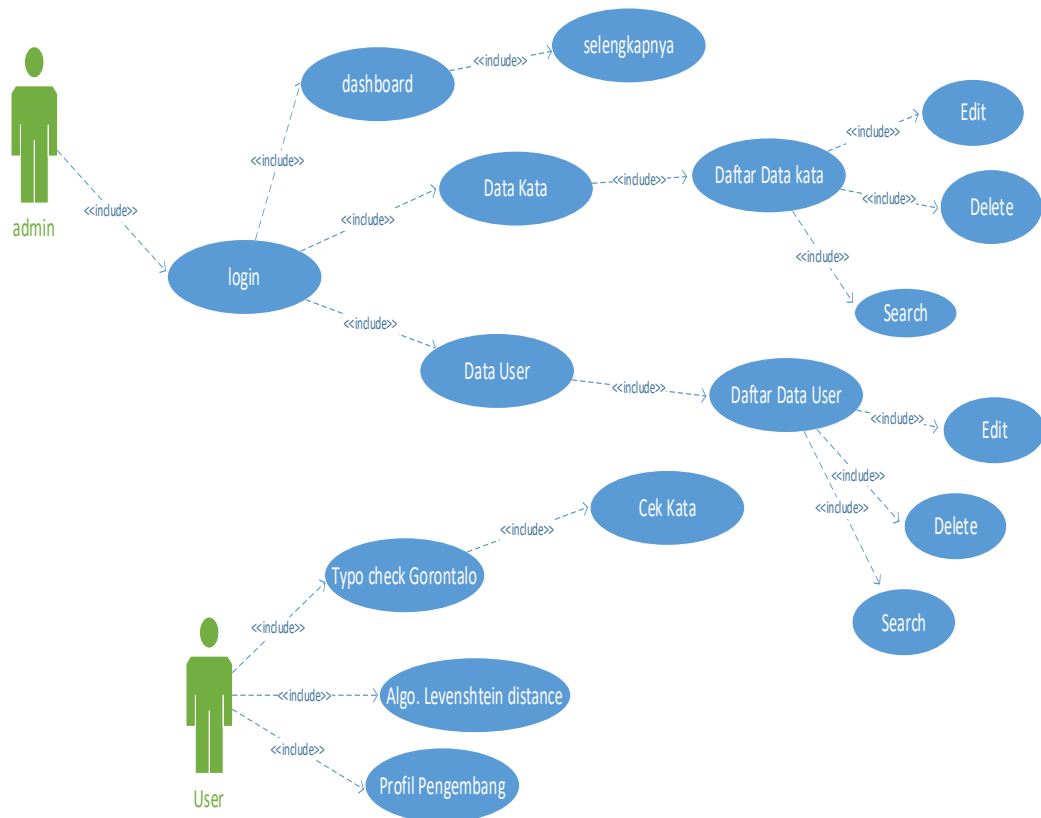
2. Penelitian data sekunder (Kepustakaan)

Metode kepustakaan ini diperlukan untuk mendapatkan data sekunder dengan tujuan melengkapi data primer. Data sekunder didapatkan dari pengkajian kepustakaan yang berisi dasar-dasar teori. Metode kepustakaan ini digunakan oleh analisis sistem dengan cara mengambil contoh dokumen-dokumen yang berhubungan dengan materi penelitian seperti yang ada pada jurnal-jurnal dan buku.

3.3 Pengembangan Sistem

Pengembangan sistem merupakan suatu analisa yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem dan desain sistem dalam merancang

sebuah aplikasi, pada penelitian ini peneliti menggunakan desain sistem bentuk UML dari mulai sistem dibangun sampai dengan di implementasikan.



Gambar 3. 1 Sistem yang diusulkan

3.3.1 Analisis sistem

Analisis sistem menggunakan pendekatan berorientasi pada *Unified /*.

Modeling Language (UML).

- a). *Functional Modelling*, menggunakan alat bantu UML, dalam bentuk:
 - *Use Case Diagram*
 - *Activity Diagram*
- b). *Structural Modelling*, menggunakan alat bantu UML, dalam bentuk:
 - *Class Diagram*
- c). *Behavioral Modelling*, menggunakan alat bantu UML, dalam bentuk:
 - *Sequence Diagram*

3.3.2 Desain Sistem

Desain sistem menggunakan pendekatan berorientasi objek digambarkan dalam bentuk :

- a). Architecture Design, menggunakan alat bantu UML, dalam bentuk :
 - Model dari sistem adalah *client server*
 - Spesifikasi hardware dan software yang direkomendasikan adalah:
 1. Sistem operasi: windows 10
 2. Prosesor dengan kecepatan minimal 2,0Hz
 3. Memori : 4 GB
 4. *Hardisk free space* 30 GB
- b). Interface Design, menggunakan alat bantu UML, dalam bentuk:
 - Mekanisme User
 - Mekanisme Navigasi
 - Mekanisme Input (*Form*)
 - Mekanisme Output (*report*)
- c). Data Design
 - SQL

- Struktur Data
 - Database Diagram
- d). Program Design
- Class
 - Attributes
 - Method
 - Event

3.3.3 Kontruksi Sistem

Pada tahap ini dapat menerjemahkan hasil pada tahap analisis dan desain ke dalam sebuah kode – kode program computer kemudian dapat membangun sistemnya. Alat bantu yang digunakan pada tahap ini adalah android studio, subline text, jdk, xampp. Dengan Bahasa pemrograman PHP dan Android dan Java. Alat bantu database yang digunakan adalah MYSQL.

3.3.4 Pengujian Sistem

A. White Box Testing

Software yang telah direkayasa kemudian diuji dengan metode White Box Testing pada kode program proses penerapan metodenya/modelnya. Kode program tersebut dibuatkan *flowchart* programnya kemudian dipetakan kedalam bentuk *flowgraph* (bagian alir kontrol) yang tersusun dari beberapa node dan edge, Berdasarkan *flowgraph*, ditentukan jumlah *Region* dan *Cyclomatic Complexity* (CC). Apabila independent path = $V(G) = (CC) = \text{region}$, dimana setiap *path* hanya dieksekusi sekali dan sudah benar, maka sistem dinyatakan efisien dari segi kelayakan logika pemrograman.

B. Black Box Testing

Selanjutnya software diuji pula dengan metode Black Box Testing yang fokus pada keperluan fungsional dari *software* dan berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya: (1) fungsi-fungsi yang salah atau hilang; (2) kesalahan *interface*; (3) kesalahan dalam struktur data atau akses data basis eksternal; (4) kesalahan *peforma*; (5) kesalahan inisialisasi dan terminasi. Jika sudah tidak ada kesalahan-kesalahan tersebut, maka sistem dinyatakan efisien dari segi kesalahan komponen-komponen sistem.

BAB IV

HASIL PENELITIAN

4.1 Hasil Pengumpulan Data

Berdasarkan hasil pengumpulan data, diperoleh data sekunder sebagai berikut:

Tabel 4. 1 Hasil Pengumpulan Data

No	Bahasa Gorontalo
1	mominggulo
2	molotongo
3	momilohu
4	mohumoyongo
5	motuluhu
6	modiilo
7	monga
8	molamelo
9	morijiki
10	moluwango
...	...
140	hula
141	tei

4.1.1 Penerapan Metode

Rumus levenshtein distance :

1. Penghapusan : $D(s,t) = \min D(s-1,t) + 1$
2. Penyisipan : $D(s,t) = \min D(s,t-1) + 1$
3. Penggantian : $D(s,t) = \min D(s-1, t-1) + 1$, jika $s(j) \neq t(i)$
4. Tidak ada perubahan : $D(s,t) = \min D(s-1, t-1) + 1$, jika $s(j) = t(i)$

Keterangan : s = string sumber

t = string target

s(j) = karakter string sumber ke-j

t(i) = karakter string target ke -i

D = jarak edit levenshtein distance

Langkah – langkah algoritma levenshtein distance :

1. Inisialisasi
 - A. hitung panjang S dan T
 - B. Buat Matriks berukuran $0 \dots M$ sebagai baris dan $0 \dots N$ sebagai kolom
 - C. Inisialisasi kolom pertama dengan $0 \dots N$
 - D. Inisialisai kolom pertama dengan $0 \dots M$
2. Proses
 - A. Periksa $S[i]$ untuk $1 < i < N$
 - B. Periksa $S[i]$ untuk $1 < j < M$
 - C. Jika $S[i] = T[j]$, maka entrinya adalah nilai yang terletak tepat pada diagonal atas sebelah kiri, yaitu $D[i,j] = D[i-1, j - 1]$
 - D. Jika $S[i] \neq T[j]$, maka entrinya adalah $D[i,j]$, minimum dari :
 - Nilai yang terletak tepat di atasnya , ditambah satu, yaitu $D[i,j - 1] + 1$
 - Nilai yang terletak tepat di kirinya , ditambah satu, yaitu $D[i - 1,j] + 1$
 - Terletak tepat pada di diagonal atas sebelah kirinya , ditambah satu, yaitu $D[i - 1,j - 1] + 1$
3. Hasil entri matriks pada baris ke-i dan kolom-j, yaitu $D[i,j]$
 Ulangi langkah 2 hingga entri $D[m,n]$ ditemukan.

Contoh Kasus :

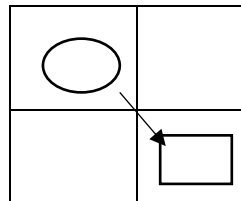
S (Sumber) : AHALY

T (Target) : AHALI

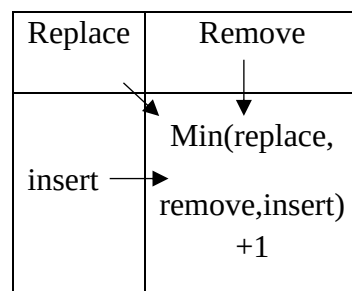
S(i) : String Sumber

T(j) : String Target

Jika $S(i) = T(j)$ maka diagonalnya diturunkan



Jika $S(j) \neq T(j)$ maka nilai minimum dari replace, remove, insert di tambah 1



	S(i)	A	H	A	L	Y
T(j)	0	1	2	3	4	5
A	1					
H	2					
A	3					
L	4					
I	5					

Pada tabel di atas diketahui bahwa string sumber S(i) memiliki 5 karakter dan string target T(j) memiliki 5 karakter. Selanjutnya karakter ke-1 pada masing-masing string dibandingkan dan diketahui bahwa isi karakter ke-1 pada masing-masing string sama, maka nilai diagonalnya diturunkan.

	S(i)	A
T(j)	0	1
A	1	0

Selanjutnya perhitungan jarak pada karakter ke-2 string sumber dengan karakter ke-1 pada string target tidak sama, maka mencari nilai minimum dari ketiga angka diagonal kemudian ditambahkan satu. Diisi dengan nilai 1 seperti tabel dibawah ini:

	S(i)	A	H
T(j)	0	1	2
A	1	0 + 1 → 1	

Selanjutnya perhitungan jarak pada karakter ke-3 string sumber dengan karakter ke-1 pada string target tidak sama, maka mencari nilai minimum dari ketiga angka diagonal kemudian ditambahkan satu. Diisi dengan nilai 2 seperti tabel dibawah ini:

	S(i)	A	H	A
T(j)	0	1	2	3
A	1	0	1 + 1 → 2	

Selanjutnya perhitungan jarak pada karakter ke-4 string sumber dengan karakter ke-1 pada string target tidak sama, maka mencari nilai minimum dari ketiga angka diagonal kemudian ditambahkan satu. Diisi dengan nilai 3 seperti tabel dibawah ini:

	S(i)	A	H	A	L
T(j)	0	1	2	3	4
A	1	0	1	2+1 → 3	

Selanjutnya perhitungan jarak pada karakter ke-5 string sumber dengan karakter ke-1 pada string target tidak sama, maka mencari nilai minimum dari ketiga angka diagonal kemudian ditambahkan satu. Diisi dengan nilai 4 seperti tabel dibawah ini:

	S(i)	A	H	A	L	Y
T(j)	0	1	2	3	4	5
A	1	0	1	2	3+1 → 4	

Selanjutnya perhitungan jarak pada karakter ke-1 string sumber dengan karakter ke-1 pada string target tidak sama, maka mencari nilai minimum dari ketiga angka diagonal kemudian ditambahkan satu. Diisi dengan nilai 4 seperti tabel dibawah ini:

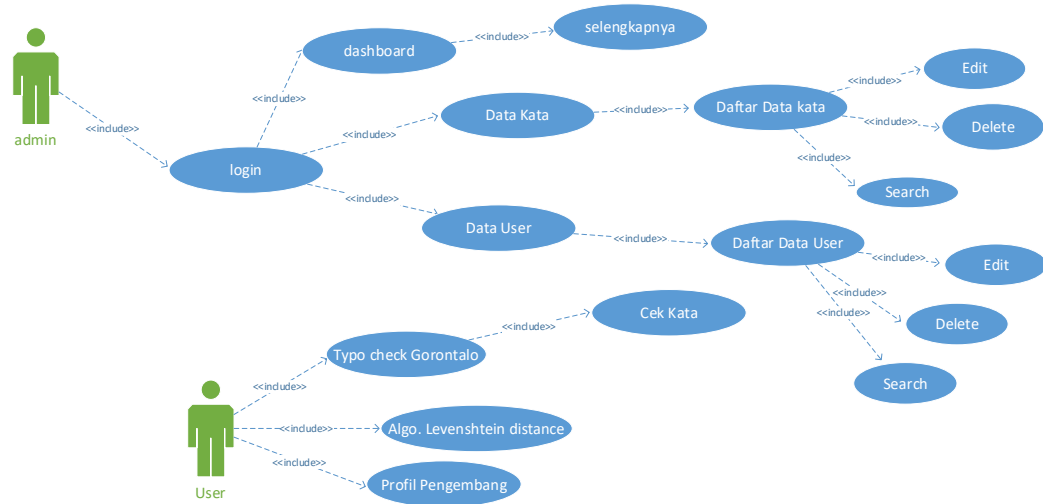
	S(i)	A
T(j)	0	1
A	1	0+1 ↓ 1
H	2	

Selanjutnya karakter ke-2 pada masing-masing string dibandingkan dan diketahui bahwa isi karakter ke-2 pada masing-masing string sama, maka nilai diagonalnya diturunkan. Diisi dengan nilai 0 dan begitu seterusnya seperti tabel dibawah ini :

	S(i)	A	H	A	L	Y
T(j)	0	1	2	3	4	5
A	1	0(+1)	1	2(+1)	3(+1)	4
H	2	1	0(+1)	1(+1)	2(+1)	3
A	3	2(+1)	1(+1)	0(+1)	1(+1)	2
L	4	3(+1)	2(+1)	1(+1)	0(+1)	1
I	5	4	3	2	1	1

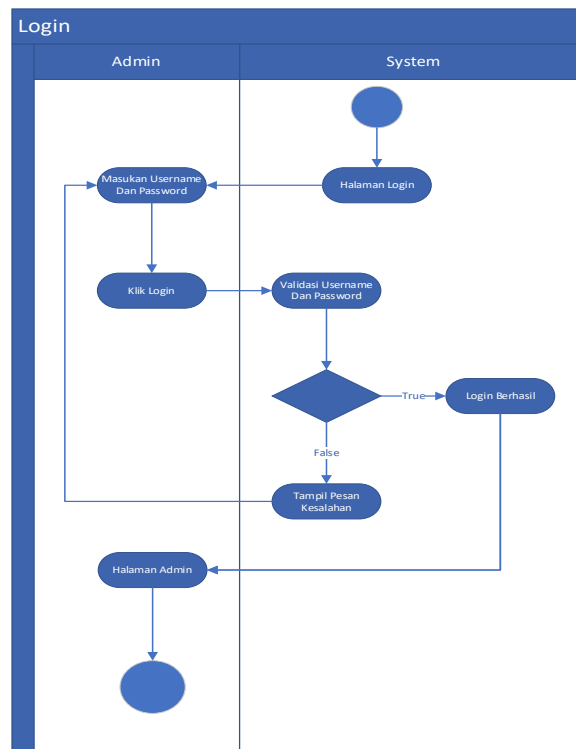
Jadi diketahui bahwa jarak levenshtein distance antara kata 'AHALY' dan kata 'AHALI' adalah 1 yaitu, Mensubtitusikan Y dengan I (AHALY $\rightarrow$ AHALI).

3.1 Hasil Pengembangan Sistem



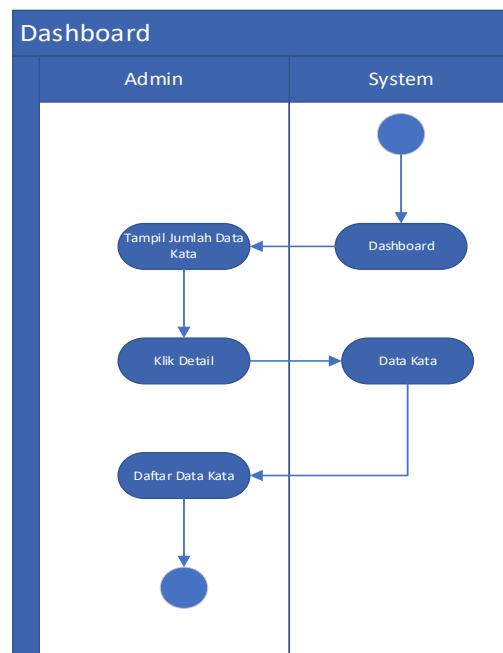
Gambar 4. 1 Use Case Diagram Aplikasi Perbaikan Kata Bahasa Gorontalo

3.2 Activity Diagram Login



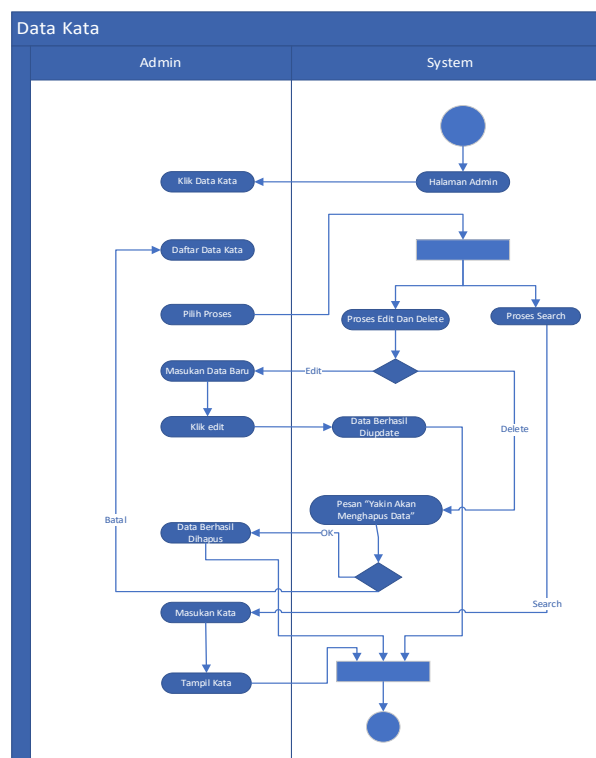
Gambar 4. 2 Activity Diagram Login

3.3 Activity Diagram Dashboard



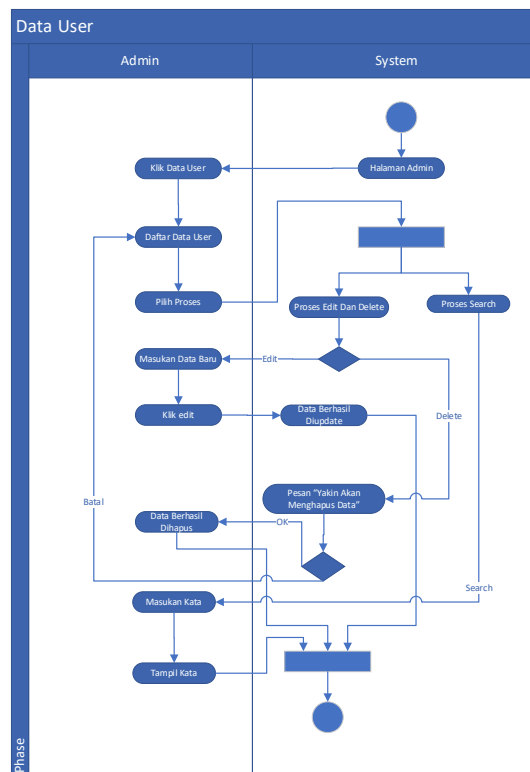
Gambar 4. 3 Activity Diagram Dashboard

3.4 Activity Diagram Data Kata



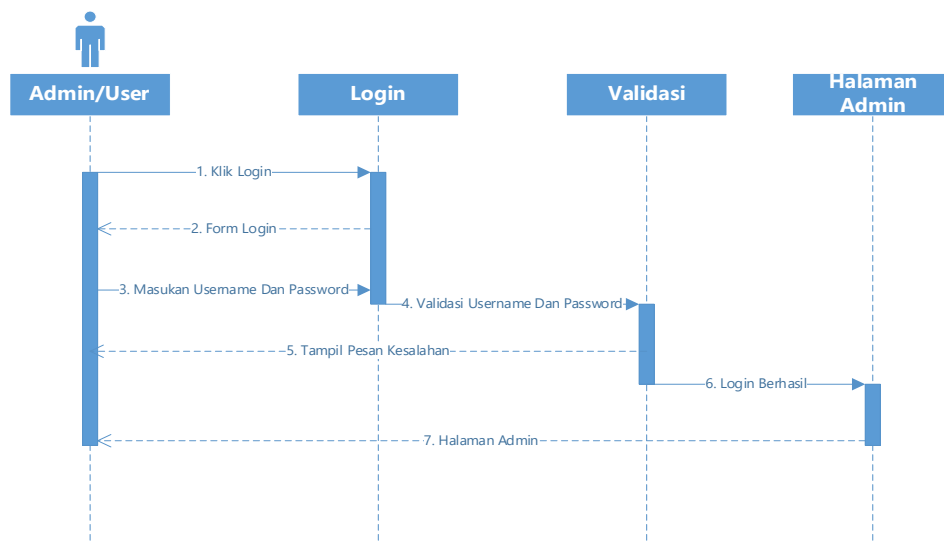
Gambar 4. 4 Activity Diagram Data Kata

3.5 Activity Diagram Data User



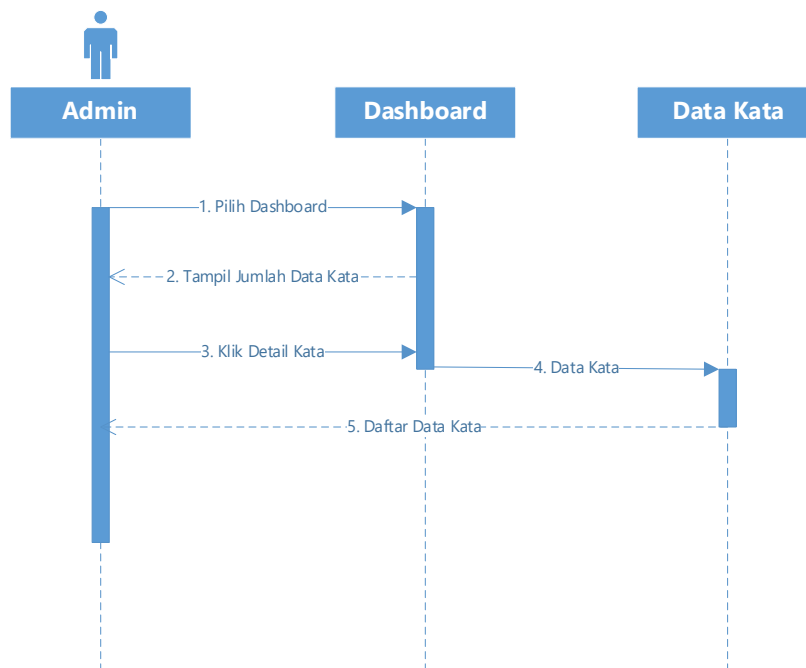
Gambar 4. 5 Activity Diagram Data User

3.6 Sequence Diagram Login Admin



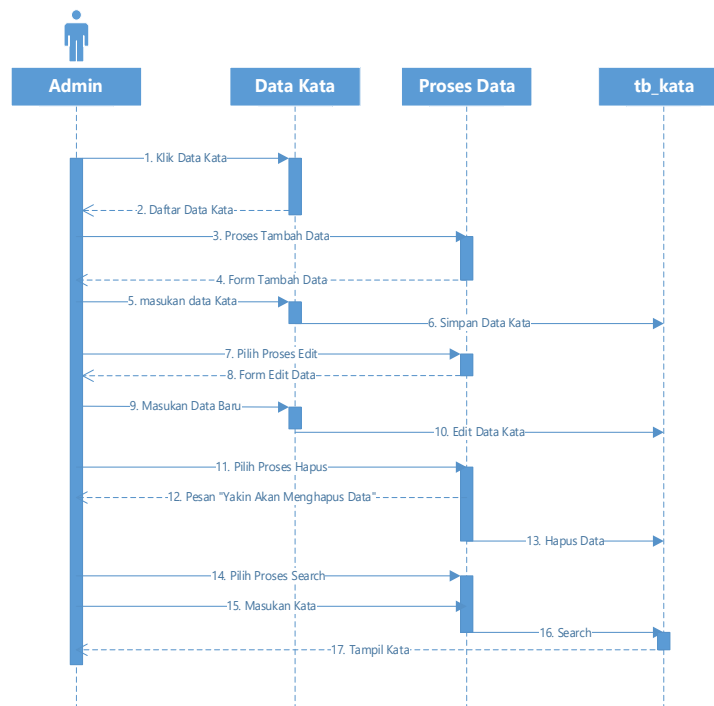
Gambar 4. 6 Sequence Diagram Login Admin

3.7 Sequence Diagram Dashboard



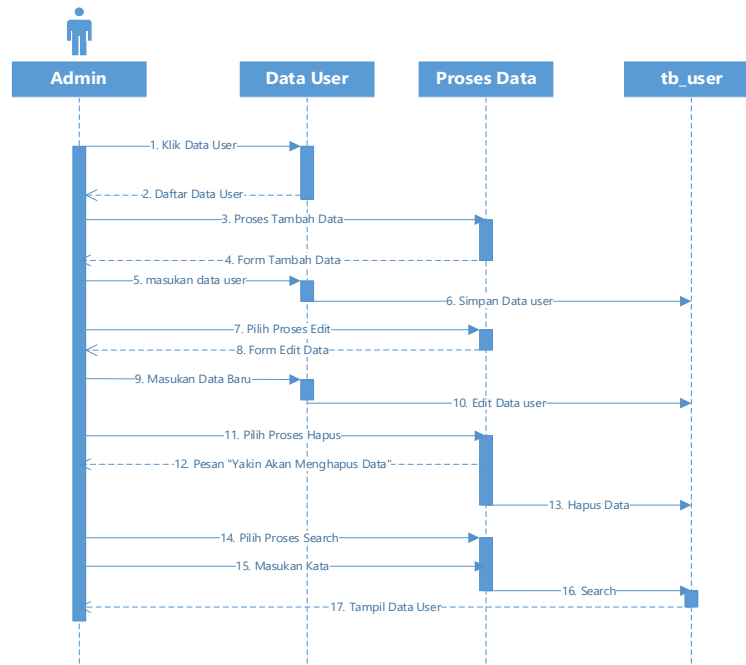
Gambar 4. 7 Sequence Diagram Dashboard

3.8 Sequence Diagram Data Kata



Gambar 4. 8 Sequence Diagram Data Kata

4.10 Sequence Diagram User



Gambar 4. 9 Sequence Diagram User

4.11 Arsitektur Sistem

Untuk kinerja sistem yang optimal, sebaiknya gunakan perangkat keras dan perangkat lunak sebagai berikut:

1. *Processor* : Minimal Core i5
2. *RAM* : 2GB
3. *VGA* : 16 Bit
4. *Hardisk* : 500GB
5. *Operating System* : Windows 8
6. *Tools* : Android Studio

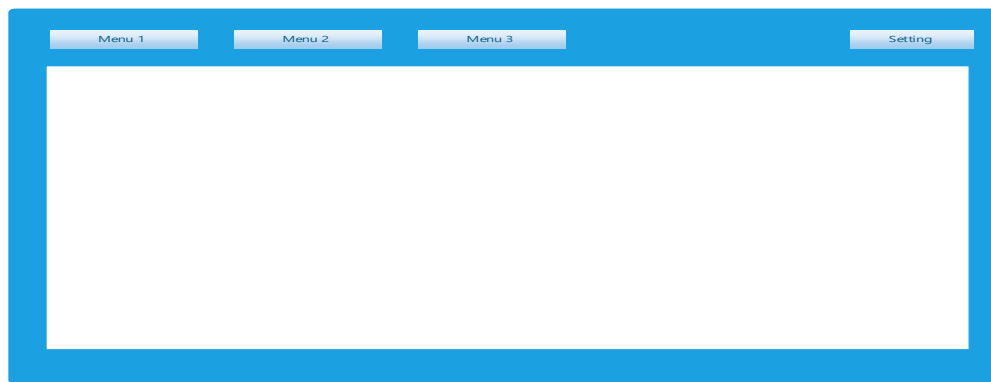
4.12 Interface Design

4.12.1 Mekanisme User

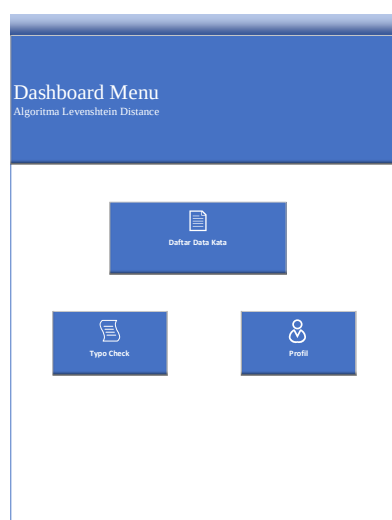
Tabel 4. 2 Mekanisme User

User	Kategori	Akses Input	Akses Output
User Admin	User Administrator	- All	Typo Check All

4.12.2 Mekanisme Navigasi *Home*

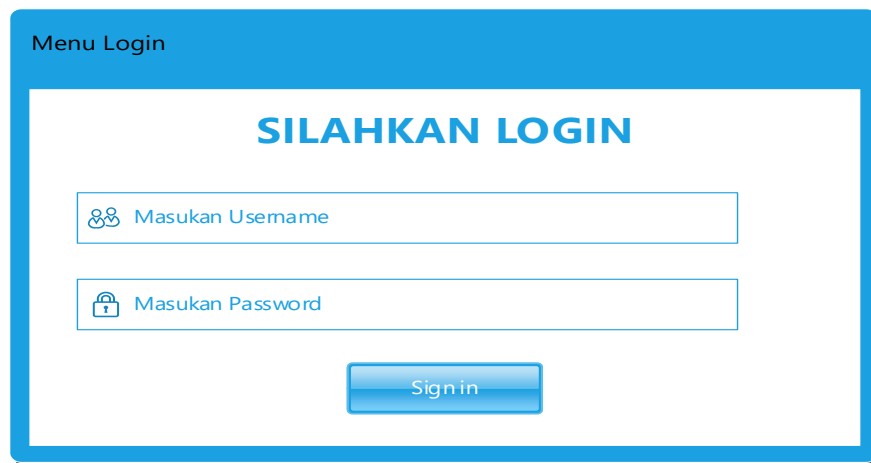


Gambar 4. 10 Mekanisme Navigasi *Home* Admin



Gambar 4. 11 Mekanisme Navigasi *Home* User

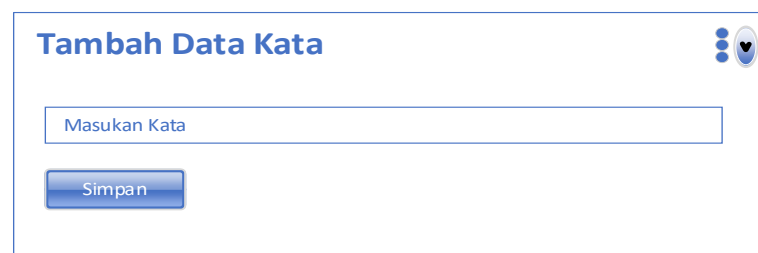
4.12.3 Mekanisme Login



The screenshot shows a login interface with a blue header bar labeled "Menu Login". Below the header, the text "SILAHKAN LOGIN" is displayed in blue. There are two input fields: the first is labeled "Masukan Username" with a user icon, and the second is labeled "Masukan Password" with a lock icon. A blue "Sign in" button is positioned below the password field.

Gambar 4. 12 Mekanisme Login

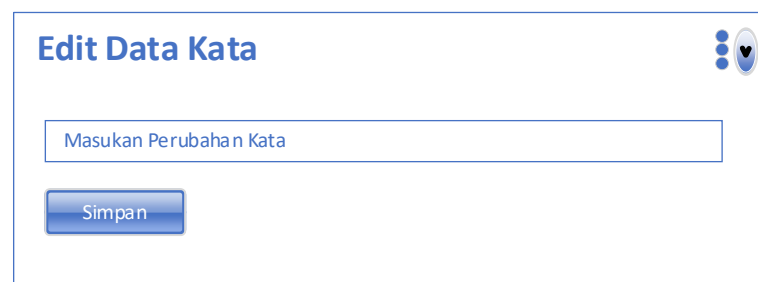
4.12.4 Mekanisme Input Data Kata



The screenshot shows a form titled "Tambah Data Kata" in blue. It features a text input field labeled "Masukan Kata" and a blue "Simpan" button below it. In the top right corner, there is a menu icon consisting of three vertical dots and a circular arrow icon.

Gambar 4. 13 Mekanisme Input Data kata

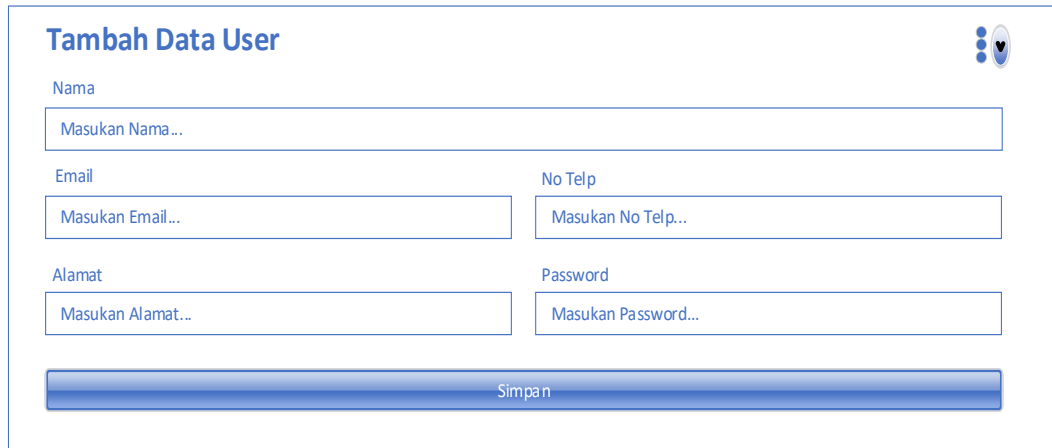
4.12.5 Mekanisme Edit Data Kata



The screenshot shows a form titled "Edit Data Kata" in blue. It features a text input field labeled "Masukan Perubahan Kata" and a blue "Simpan" button below it. In the top right corner, there is a menu icon consisting of three vertical dots and a circular arrow icon.

Gambar 4. 14 Mekanisme Edit Data Kata

4.12.6 Mekanisme Tambah Data User



Tambah Data User

Nama

Masukan Nama...

Email

Masukan Email...

No Telp

Masukan No Telp...

Alamat

Masukan Alamat...

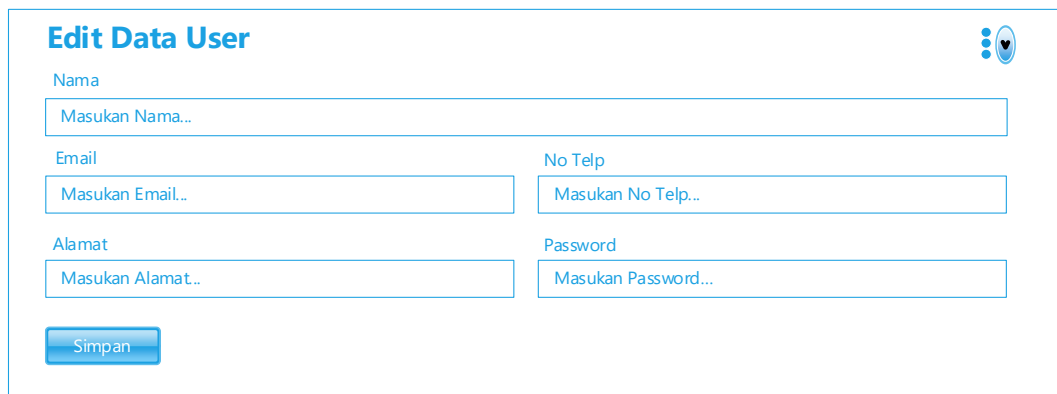
Password

Masukan Password...

Simpan

Gambar 4. 15 Mekanisme Tambah Data User

4.12.7 Mekanisme Edit Data User



Edit Data User

Nama

Masukan Nama...

Email

Masukan Email...

No Telp

Masukan No Telp...

Alamat

Masukan Alamat...

Password

Masukan Password...

Simpan

Gambar 4. 16 Mekanisme Edit Data User

4.12.8 Mekanisme *Output*

Gambar 4. 17 Mekanisme *Output* Perbaikan Kata

4.13 Data Desain

4.13.1 Struktur Data

Tabel 4. 3 Tabel Kata

Nama File : kata Primary key : id Media : hardisk fungsi : menyimpan data kata struktur data :				
No	Field Name	Type	Width	keterangan
1.	Id	int	5	Id kata
2.	Kata	Varchar	100	Data kata

Tabel 4. 4 Tabel User

Nama File : user Primary key : id Media : hardisk fungsi : menyimpan data user struktur data :				
No	Field Name	Type	Width	keterangan
1.	Id	int	11	Id User
2.	nama	Varchar	200	Nama User/Admin
3.	email	Varchar	100	Email User/Admin
4.	telp	varchar	100	No Telpon User/Admin
5.	alamat	varchar	100	Alamat User/Admin
6	pass	varchar	100	Password User/Admin

4.14 Hasil Pengujian Sistem

4.14.1 Pengujian *White Box*

\$pecah = explode(" ", \$input); 1

\$hasil= \$pecah[\$y]; 1

foreach(\$names as \$name) { 2

\$lev = levenshtein(\$hasil, \$name); 3

if (\$lev == 0) { 4

\$closest = \$name; 5

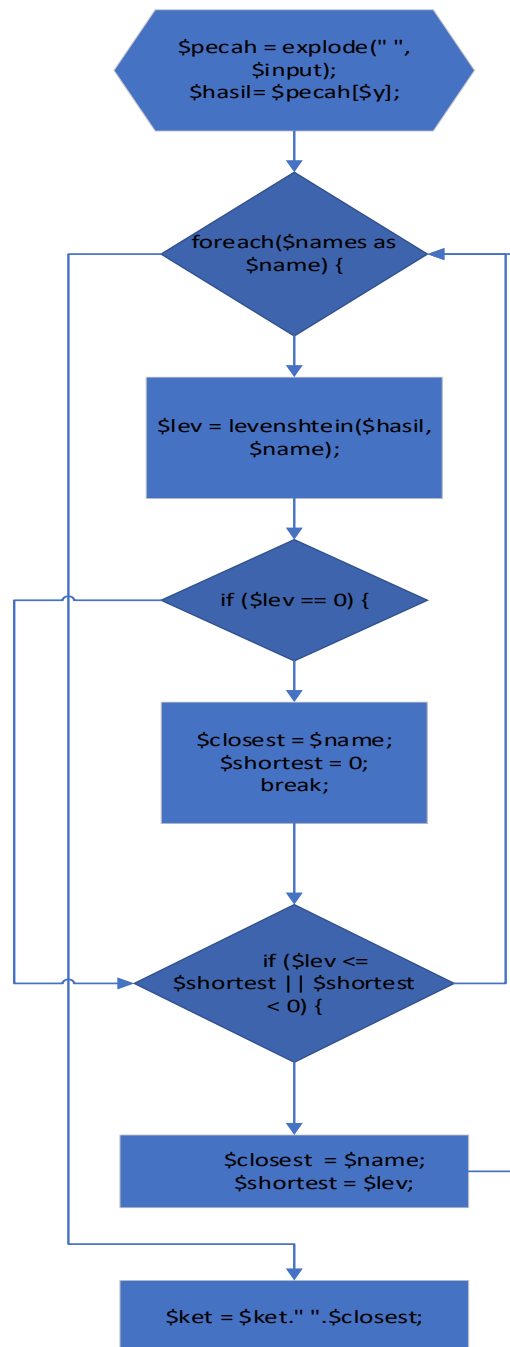
\$shortest = 0; 5

```

        break;.....5
    }
    if ($lev <= $shortest || $shortest < 0) {.....6
        $closest = $name;.....7
        $shortest = $lev; .....7
    }
}
$ket = $ket." ".$closest; .....8

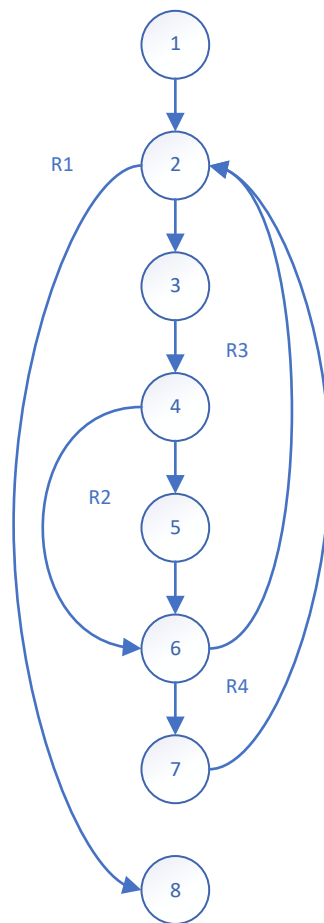
```

4.14.2 Flowchart Diagram



Gambar 4. 18 Flowchart

4.14.3 Pengujian *White Box*



Gambar 4. 19 *Flowgraph*

4.14.4 Pengujian CC Pada Pengujian *White Box*

Dari <i>Flowgraph</i> tersebut, didapatkan :		
Diketahui	$Edge(E)$	= 10
	$Node(N)$	= 8
	$Predicate\ Node(P)$	= 3
	$Region(R)$	= 4
Rumus : $V(G) = E - N + 2$ dan $V(G) P + 1$		
Penyelesaian :	$V(G) = 10 - 8 + 2$	= 4
	$V(G) = 3 + 1$	= 4
	(R1, R2, R3, R4)	

4.14.5 Pengujian CC Pada Pengujian *White Box*

Tabel 4. 5 Basis Path

NO	Path	Ket
1	1 – 2 – 8	OK
2	1 – 2 – 3 – 4 – 6 – 2 – 8	OK
3	1 – 2 – 3 – 4 – 5 – 6 – 2 – 8	OK
4	1 – 2 – 3 – 4 – 5 – 6 – 7 – 2 – 8	OK

Ketika aplikasi dijalankan, maka terlihat bahwa semua basis path yang dihasilkan telah dieksekusi satu kali. Berdasarkan ketentuan tersebut dari segi kelayakan *software*, sistem ini telah memenuhi syarat.

4.14.6 Pengujian *Black Box*

Pengujian *Black Box* dilakukan untuk memastikan bahwa suatu *event* atau masukan akan menjalankan proses yang tepat dan menghasilkan *Output* Sesuai dengan rancangan. Untuk Contoh pengujian terhadap beberapa proses memberikan hasil sebagai berikut :

Tabel 4. 6 Tabel Pengujian *Black Box*

Input/Event	Fungsi	Hasil Yang Diharapkan	Hasil Uji
Klik Menu login	Login ke halaman admin	Tampil form login	Sesuai
Klik Menu Registrasi	Halaman Registrasi	Tampil form Registrasi	Sesuai
Masukkan Username Salah, klik Login	Validasi email	Tampil pesan “Login Gagal! Username Dan Password Anda Salah!”	Sesuai
Masukkan Password Salah, klik Login	Validasi password	Tampil pesan “Login Gagal! Username Dan Password Anda Salah!”	Sesuai
Masukkan email dan password benar, klik login	Validasi email dan password	Tampil halaman Admin	Sesuai
Klik menu dashboard	Menampilkan dashboard	Tampil Dashboard	Sesuai
Klik tombol Detail kata pada dashboard	Menampilkan data kata	Tampil Data Kata	Sesuai
klik menu Data Kata	Menampilkan data kata	Tampil data kata	Sesuai
Klik Tambah data pada menu Data Kata	Menambahkan data kata	Tampil form Tambah Data Kata	Sesuai
Masukkan data kata, klik simpan	Menambahkan data kata	Tampil pesan “Data berhasil diinput!”	Sesuai
Klik Edit pada menu Data Kata	Mengubah data kata	Tampil form Edit Data Kata	Sesuai
Masukkan data kata yang baru, klik Edit	Mengubah data kata	Tampil pesan “Data berhasil diupdate!”	Sesuai
Klik Hapus pada menu Data Kata	Menghapus data kata	Tampil pesan “Yakin akan menghapus data?”	Sesuai
Klik menu Data user	Menampilkan data user	Tampil Data User	Sesuai
Klik edit pada menu Data User	Mengubah data user	Tampil form Edit User	Sesuai

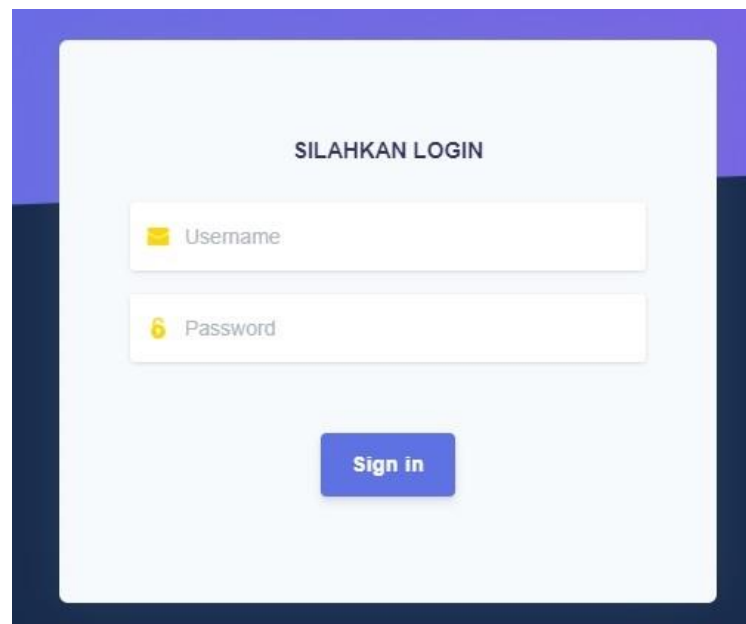
Input/Event	Fungsi	Hasil Yang Diharapkan	Hasil Uji
Masukkan perubahan data user, klik Simpan	Mengubah data user	Tampil pesan “Data berhasil di update”	Sesuai
Klik Delete pada menu data user	Menghapus data user	Tampil pesan “Yakin Akan Menghapus data?”	Sesuai
Klik Logo instagram	Menampilkan instagram	Tampil instagram Developer	Sesuai
Klik Logo facebook	Menampilkan facebook	Tampil facebook Developer	Sesuai
Klik Profil Pengembang	Menampilkan data pembuat aplikasi	Tampil data Tentang Pengembang	Sesuai
Klik Typo Check	Menampilkan Perbaikan kata	Tampil perbaikan kata	Sesuai

BAB V

HASIL DAN PEMBAHASAN

5.1 Pembahasan Sistem

5.1.1 Tampilan Halaman *Login*

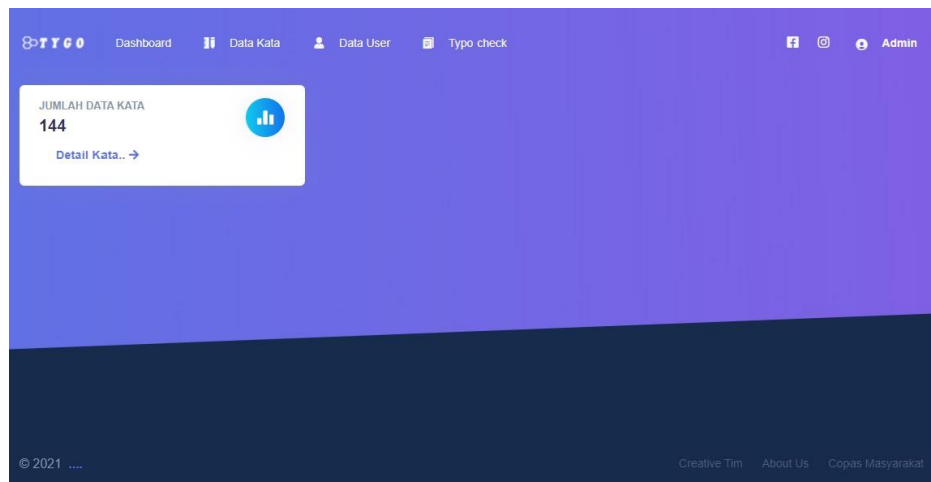


The image shows a login interface with a light blue background and a dark blue border. At the top, the text "SILAHKAN LOGIN" is centered. Below it are two white input fields. The first field is labeled "Username" and has a yellow envelope icon to its left. The second field is labeled "Password" and has a yellow lock icon to its left. Below these fields is a blue button with the text "Sign in" in white.

Gambar 5. 1 Tampilan Halaman *Login*

Halaman ini digunakan untuk masuk ke halaman Admin, dimulai dengan memasukkan Username dan Password, Klik Login untuk melanjutkan proses.

5.1.2 Tampilan Halaman Dashboard Admin



Gambar 5. 2 Tampilan Halaman Dashboard Admin

Halaman ini merupakan halaman dashboard dari admin, data yang ditampilkan berupa Jumlah Data Kata. Untuk melihat detail data kata, klik detail kata.

5.1.3 Tampilan Halaman Data Kata

Daftar Data Kata

+ Tambah Data

Show 10 entries

Search:

NO	KATA	ACTION	
1	mominggulo	Edit	Delete
2	molontongo	Edit	Delete
3	momilohu	Edit	Delete
4	motuluhi	Edit	Delete
5	modiilo	Edit	Delete
6	monga	Edit	Delete
7	molamelo	Edit	Delete
8	morijiki	Edit	Delete
9	moluwango	Edit	Delete
10	moliupo	Edit	Delete

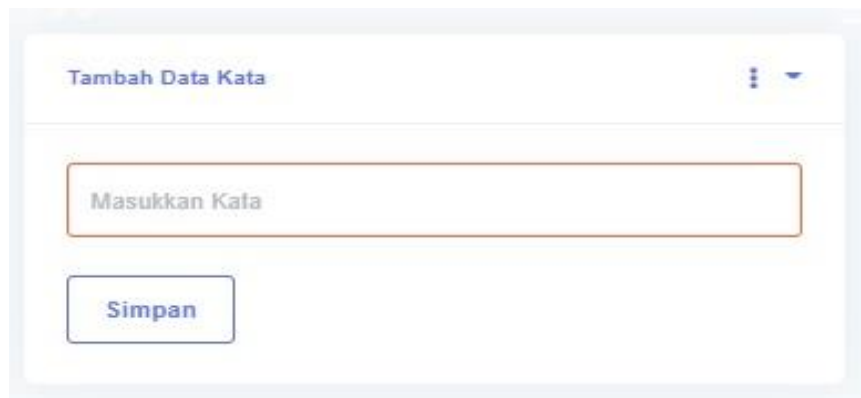
Showing 1 to 10 of 144 entries

Previous12345...15Next

Gambar 5. 3 Tampilan Halaman Data Kata

Halaman ini digunakan untuk menampilkan detail data kata. Terdapat button Edit untuk mengubah data kata, button Hapus untuk menghapus data kata, kolom search untuk mencari kata, dan button Tambahkan Data untuk menambah data Kata.

5.1.4 Tampilan Halaman Tambah Data Kata



Gambar 5. 4 Tampilan Halaman Data Kata

Halaman ini digunakan untuk menambahkan data kata. Hanya dengan memasukan kata kemudian klik Simpan untuk melanjutkan proses penyimpanan data kata.

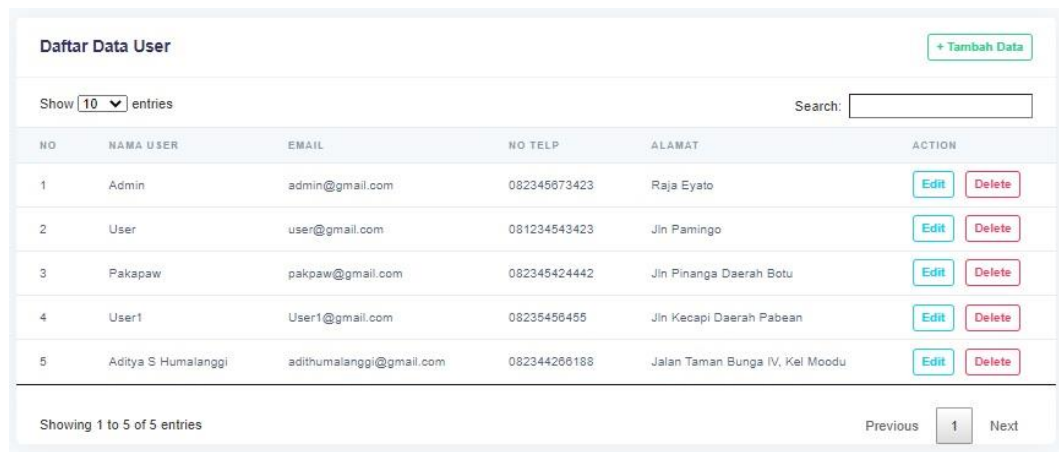
5.1.5 Tampilan Halaman Edit Data Kata



Gambar 5. 5 Tampilan Halaman Edit Data Kata

Halaman ini digunakan untuk mengubah data kata, masukkan perubahan kata. Klik button simpan untuk melanjutkan proses pengubahan data kata.

5.1.6 Tampilan Halaman Data User



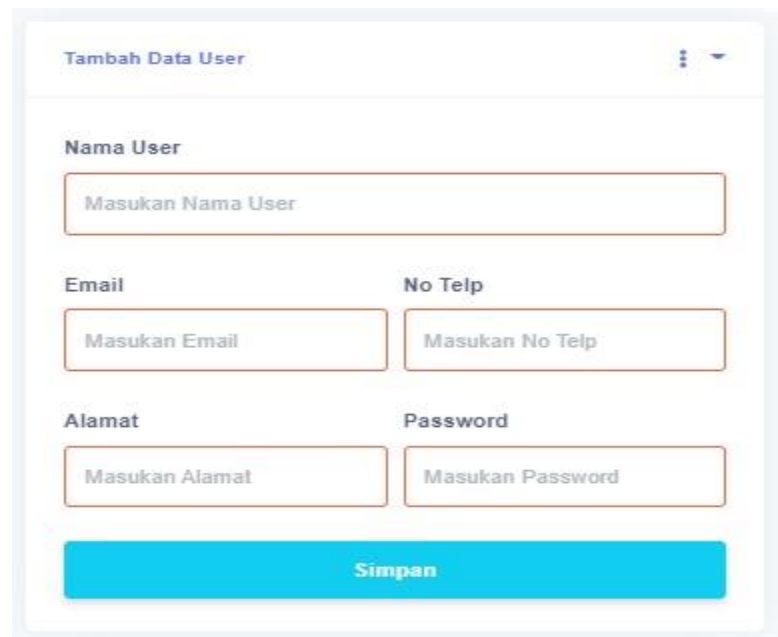
The screenshot shows a web interface titled "Daftar Data User". At the top right is a green button labeled "+ Tambah Data". Below the title, there is a "Show 10 entries" dropdown and a "Search:" input field. The main content is a table with 6 columns: NO, NAMA USER, EMAIL, NO TELP, ALAMAT, and ACTION. The table contains 5 rows of user data. Each row has "Edit" and "Delete" buttons in the ACTION column. At the bottom, there is a pagination bar showing "Showing 1 to 5 of 5 entries" and "Previous 1 Next".

NO	NAMA USER	EMAIL	NO TELP	ALAMAT	ACTION
1	Admin	admin@gmail.com	082345873423	Raja Eyato	Edit Delete
2	User	user@gmail.com	081234543423	Jln Pamingo	Edit Delete
3	Pakpaw	pakpaw@gmail.com	082345424442	Jln Pinanga Daerah Botu	Edit Delete
4	User1	User1@gmail.com	08235456455	Jln Kecapi Daerah Pabean	Edit Delete
5	Aditya S Humalanggi	adithumalanggi@gmail.com	082344266188	Jalan Taman Bunga IV, Kel Moodu	Edit Delete

Gambar 5. 6 Tampilan Halaman Data User

Halaman ini digunakan untuk menampilkan daftar data user, terdiri dari Nama User, Email, No. Telp, dan Alamat, serta terdapat button Tambahkan data User untuk menambahkan data user, button edit untuk mengubah data user, dan button Delete untuk menghapus data user.

5.1.7 Tampilan Halaman Tambah Data User



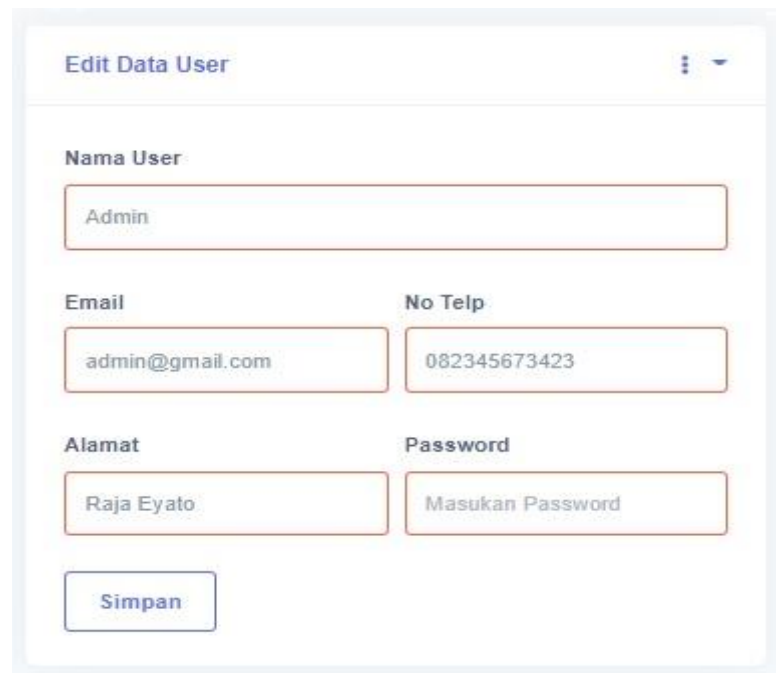
The screenshot shows a web form titled "Tambah Data User" with a menu icon in the top right corner. The form contains five input fields: "Nama User" (Masukan Nama User), "Email" (Masukan Email), "No Telp" (Masukan No Telp), "Alamat" (Masukan Alamat), and "Password" (Masukan Password). A blue "Simpan" button is at the bottom.

Tambah Data User	
Nama User	
Masukan Nama User	
Email	No Telp
Masukan Email	Masukan No Telp
Alamat	Password
Masukan Alamat	Masukan Password
Simpan	

Gambar 5. 7 Tampilan Halaman Tambah Data User

Halaman ini digunakan untuk menambahkan data user. Dimulai dengan memasukkan Nama User, Email, Password, Nomor Telepon, dan Alamat. Untuk melanjutkan proses penambahan data user, klik button Simpan.

5.1.8 Tampilan Halaman Edit Data User

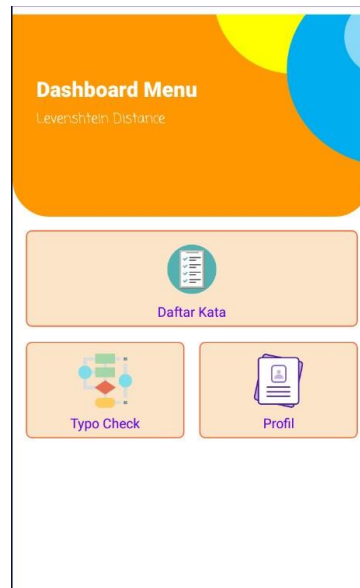


The screenshot shows a web form titled "Edit Data User". The form is organized into a grid-like structure. At the top, there is a header bar with the title "Edit Data User" and a small menu icon. Below the header, the form contains several input fields. The first field is labeled "Nama User" and contains the text "Admin". The second field is labeled "Email" and contains "admin@gmail.com". The third field is labeled "No Telp" and contains "082345673423". The fourth field is labeled "Alamat" and contains "Raja Eyato". The fifth field is labeled "Password" and contains the placeholder text "Masukan Password". At the bottom of the form, there is a button labeled "Simpan".

Gambar 5. 8 Tampilan Halaman Edit Data User

Halaman ini digunakan untuk mengubah data user, dimulai dengan memasukkan perubahan data user. Klik button Simpan untuk melanjutkan proses pengubahan data User.

5.1.9 Tampilan Halaman User



Gambar 5. 9 Tampilan Home User

5.1.10 Tampilan Data Profil



Gambar 5. 10 Halaman Data Profil

5.1.11 Tampilan Data Kata



ID	KATA
1	mominggulo
2	molontongo
3	momilohu
5	motuluhu
6	modililo
7	monga
8	molamelo
9	morijiki
10	moluwango
11	moliupo
12	molamito
13	modelito
14	mohuntapo
15	modengeto
16	mongilu
17	mobisala
18	molololo
19	modungohu

Gambar 5. 11 Halaman Data Kata

BAB VI

KESIMPULAN DAN SARAN

6. 1 Kesimpulan

Berdasarkan pembahasan pada bab – bab sebelumnya maka dapat disimpulkan sebagai berikut.

1. Dengan adanya aplikasi ini masyarakat terutama kalangan muda dapat mengetahui kata yang benar dalam Bahasa daerah Gorontalo serta Sistem mampu menghasilkan kata yang benar dalam bahasa daerah gorontalo .

6. 2 Saran

Agar sistem ini dapat bermanfaat dengan baik untuk sekarang maupun akan datang, maka penulis memberikan saran, sebagai berikut :

1. Menambahkan fitur speech to text dalam bahasa daerah gorontalo agar lebih mudah mengenali kata dalam bahasa daerah gorontalo
2. Untuk mempermudah masyarakat atau kalangan muda lebih memahami bahasa gorontalo diperlukan fitur pencarian kalimat agar lebih efisien

DAFTAR PUSTAKA

- [1] M. Nurlaila, “Pengaruh Bahasa Daerah (Ciacia) Terhadap Perkembangan Bahasa Indonesia Anak Usia 2 Sampai 6,” *Retorika*, vol. 9, no. 1, pp. 90–163, 2016.
- [2] A. Ernawati and A. Saifudin, “Perancangan Aplikasi Pembelajaran Bahasa Jawa Berbasis Android,” *J. Inform. Univ. Pamulang*, vol. 3, no. 4, p. 249, 2018, doi: 10.32493/informatika.v3i4.2281.
- [3] J. Jenderal and S. No, “Pembelajaran Bahasa Daerah Gorontalo Pada Anak Usia Dini Learning Gorontalo Local Language,” vol. 12, no. 1, pp. 81–88, 2017.
- [4] Y. Rochmawati and R. Kusumaningrum, “Studi Perbandingan Algoritma Pencarian String dalam Metode Approximate String Matching untuk Identifikasi Kesalahan Pengetikan Teks,” *J. Buana Inform.*, vol. 7, no. 2, pp. 125–134, 2016, doi: 10.24002/jbi.v7i2.491.
- [5] M. O. Braddley, M. Fachrurrozi, and Y. Novi, “Pengoreksian Ejaan Kata Berbahasa Indonesia Menggunakan Algoritma Levensthein Distance,” *Pros. Annu. Res. Semin.*, vol. 3, no. 1, pp. 167–171, 2017.
- [6] A. N. Arrasyid and M. S. Said, “Aplikasi Kamus Bahasa Daerah Tolaki Berbasis Android,” *Simtek J. Sist. Inf. dan Tek. Komput.*, vol. 1, no. 1, pp. 62–68, 2016, doi: 10.51876/simtek.v1i1.9.
- [7] B. Indonesia, M. Metode, and N. Levenshtein, “Identifikasi Kesalahan Penulisan Kata (Typographical Error) pada Dokumen Identifikasi Kesalahan Penulisan Kata (Typographical Error) pada Dokumen Berbahasa Indonesia Menggunakan Metode N-gram dan Levenshtein Distance,” no. February 2018, 2017.
- [8] Y. M. Febrianti and A. W. Widodo, “Analisis Sentimen Pada Ulasan ‘

- Lazada ' Berbahasa Indonesia Menggunakan K-Nearest Neighbor (K-NN) Dengan Perbaikan Kata Menggunakan Jaro Winkler Distance,” vol. 2, no. 10, pp. 3689–3698, 2018.
- [9] D. Yunita Purnama Sari, “ISSN 2252-9063 Kumpulan Artikel Mahasiswa Pendidikan Teknik Informatika (KARMAPATI) ISSN 2252-9063 Kumpulan Artikel Mahasiswa Pendidikan Teknik Informatika (KARMAPATI) Volume 2 , Nomor 6 , Agustus 2013,” vol. 2, pp. 86–95, 2019.
- [10] Kosidin and R. N. Farizah, “Pemodelan Aplikasi Mobile Reminder Berbasis Android,” *Semin. Nas. Teknol. Inf. dan Komunikas 2016*, vol. 2016, no. Sentika, pp. 271–280, 2016.
- [11] D. T. Hernandhi, E. S. Astuti, and S. Priambada, “Desain Sistem Informasi Pemasaran Berbasis Website Untuk Promosi,” *J. Adm. Bisnis*, vol. 55, no. 1, pp. 1–10, 2018.
- [12] M. B. Miswanto, “Aplikasi Pembelajaran Anti Korupsi Bagi Anak Remaja Berbasis Android,” *Informatika*, vol. III, no. 1, pp. 117–128, 2016, [Online]. Available: <https://ejournal.bsi.ac.id/ejurnal/index.php/ji/article/view/332/301>.
- [13] W. N. Cholifah, S. M. Sagita, and S. Knowledge, “PENGUJIAN BLACK BOX TESTING PADA APLIKASI ACTION & STRATEGY BERBASIS ANDROID,” vol. 3, no. 2, pp. 206–210, 2018.



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS ICHSAN GORONTALO

FAKULTAS ILMU KOMPUTER

SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001

Jl. Achmad Nadjamuddin No. 17 Telp (0435) 829975 Fax (0435) 829976 Gorontalo

SURAT KETERANGAN PENELITIAN

Nomor : 147/FIKOM-UIG/SKP/I/2023

Yang bertanda tangan dibawah ini :

N a m a : Irvan Abraham Salihi, M.Kom
 N I D N : 0928028101
 Jabatan : Dekan Fakultas Ilmu Komputer

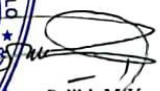
Dengan ini Menerangkan bahwa :

N a m a Mahasiswa : Aditya Supratama Humalanggi
 N I M : T3116089
 Program Studi : Teknik Informatika

Bahwa yang bersangkutan benar-benar telah melakukan penelitian tentang "Aplikasi Perbaikan Kata Dalam Bahasa Daerah Gorontalo Menggunakan Algoritma Levenstein Distance (studi kasus Prodi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo) " Guna untuk menyelesaikan Studi pada Program Studi Teknik Informatika Fakultas Ilmu Komputer, dan bersangkutan telah menyelesaikan penelitian tersebut pada TGL 06 Februari Januari 2023 sesuai dengan waktu yang telah di tentukan.

Demikian Surat Keterangan ini dibuat dan digunakan untuk seperlunya.

Gorontalo, 06 Januari 2023



Irvan Abraham Salihi, M.Kom
 NIDN. 0928028101

Lampiran 1 : Data yang dikumpulkan

Kode Program Php

Halaman Index

```
<ul class="navbar-nav mr-auto">
<li class="nav-item">
<a href="index.php" class="nav-link">
<span class="nav-link-inner--text">Dashboard</span>
</a>
</li>
<li class="nav-item">
<a href="data_kata.php" class="nav-link" >
<i class="ni ni-ruler-pencil"></i>
<span class="nav-link-inner--text">Data Kata</span>
</a>
</li>
<li class="nav-item">
<a href="data_user.php" class="nav-link">
<i class="ni ni-single-02"></i>
<span class="nav-link-inner--text">Data User</span>
</a>
</li>
<li class="nav-item">
<a href="typo.php" class="nav-link">
<i class="ni ni-single-copy-04"></i>
<span class="nav-link-inner--text">Typo check</span>
</a>
</li>
</ul>
<hr class="d-lg-none" />
<ul class="navbar-nav align-items-lg-center ml-lg-auto">
<li class="nav-item">
<a class="nav-link nav-link-icon"
href="https://web.facebook.com/adhyt.blanck" target="_blank"
data-toggle="tooltip" data-original-title="Like us on Facebook">
<i class="fab fa-facebook-square"></i>
<span class="nav-link-inner--text d-lg-none">Facebook</span>
</a>
</li>
<li class="nav-item">
<a class="nav-link nav-link-icon"
href="https://www.instagram.com/adityahumlanggi/"
target="_blank" data-toggle="tooltip" data-original-
title="Follow us on Instagram">
<i class="fab fa-instagram"></i>
<span class="nav-link-inner--text d-lg-none">Instagram</span>
</a>
</li>

<!-- tmpa profil -->
<ul class="navbar-nav align-items-center ml-auto ml-md-0 ">
<li class="nav-item dropdown">
<a class="nav-link pr-0" href="#" role="button" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false">
<div class="media align-items-center">
```

```

<i class="ni ni-circle-08"></i>
<div class="media-body ml-2 d-none d-lg-block">
<span class="mb-0 text-sm font-weight-bold"><?php echo
$_SESSION['nama']; ?></span>
</div>
</div>
</a>
<div class="dropdown-menu dropdown-menu-right ">
<div class="dropdown-header noti-title">
<h6 class="text-overflow m-0">Welcome!</h6>
</div>
<a href="profil.php" class="dropdown-item">
<i class="ni ni-single-02"></i>
<span>My profile</span>
</a>
<a href="#" class="dropdown-item">
<i class="ni ni-settings-gear-65"></i>
<span>Settings</span>
</a>
<div class="dropdown-divider"></div>
<a href="logout.php" class="dropdown-item">
<i class="ni ni-user-run"></i>
<span>Logout</span>
</a>
</div>
</li>
</ul>
</div>
</div>
</nav>
<!-- tmpa profil end -->
</ul>
</div>
</div>
</nav>
<!-- Main content -->
<div class="main-content">
<!-- Header -->
<div class="header bg-gradient-primary py-7 py-lg-8 pt-lg-9">
<div class="container">
<div class="header-body text-center mb-7">
<div class="row justify-content-center">

</div>
</div>
<div class="separator separator-bottom separator-skew zindex-
100">
<svg x="0" y="0" viewBox="0 0 2560 100"
preserveAspectRatio="none" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<polygon class="fill-default" points="2560 0 2560 100 0
100"></polygon>
</svg>
</div>
</div>

<!-- Page content -->
<div class="container mt--9 pb-5">

```

```

<!-- awal -->
<div>

<!-- Card stats -->
<div class="row">
<div class="col-xl-4 col-md-6">
<div class="card card-stats">

<!-- Card body -->
<div class="card-body">
<div class="row">
<div class="col">
<h5 class="card-title text-uppercase text-muted mb-0">Jumlah
Data Kata</h5>
<span class="h2 font-weight-bold mb-0"><?php echo"$temu
";?></span>
</div>
<div class="col-auto">
<div class="icon icon-shape bg-gradient-info text-white rounded-
circle shadow">
<i class="ni ni-chart-bar-32"></i>
</div>
</div>
</div>
<p class="mt-0 mb-0 text-sm">
<span class="btn btn-link text-success"><a
href="data_kata.php">Detail Kata.. <i class="fa fa-arrow-
right"></i></a></span>
</p>
</div>
</div>
</div>
</div>
</div>
</div>

```

Typo check

```

<?php
    // input misspelled word
    $db_host      = "localhost";
    $db_username  = "root";
    $db_password  = "";

    //connect to mysqli database (Host/Username/Password)
    $connection = mysqli_connect($db_host, $db_username,
$db_password) or die("Error " . mysqli_error());
    //select MySQLi dabatase table
    $db = mysqli_select_db($connection, "goro") or
die("Error " . mysqli_error());
    $sql = mysqli_query($connection, "SELECT * FROM kata");
    while($row = mysqli_fetch_array($sql)) {
        $names[] = $row['nama'];
    }
}

```



```

    }

    $shortest = -1;
    $input = $_GET['input'];

    $x = str_word_count($input);
    $x = $x - 1;
    $ket = "";

    for ($y=0; $y<=$x; $y++) {

        $pecah = explode(" ", $input);

        $hasil= $pecah[$y];
        // echo "$y <br>";
        // echo "$hasil <br>";

        foreach ($names as $name) {

            $lev = levenshtein($hasil, $name);
            if ($lev == 0) {

                $closest = $name;
                $shortest = 0;

                break;
            }

            if ($lev <= $shortest || $shortest < 0) {

                $closest = $name;
                $shortest = $lev;
            }
        }
        // echo "$closest <br>";

        $ket = $ket." ".$closest;
    }
?>
<div class="col-lg-6">
<div class="form-group">
<label class="form-control-label" for="input-
email"></label>
<textarea rows="5" class="form-control"
placeholder="Hasil Perbaikan...!!!"><?php
    if ($shortest == 0) {
        echo "Exact match found : $ket";
    } else {
        echo "$ket";
    }
?>

```

Kode Pemrograman Android

Registrasi API

```
public interface RegistrasiAPI {  
    @GET("tampil_all_kata.php")  
    Call<ResponseKata> tampilKata();  
  
    @FormUrlEncoded  
    @POST("tambah_kata.php")  
    Call<ResponseKata> tambahKata(  
        @Field("nama") String nama  
    );  
  
    @FormUrlEncoded  
    @POST("hapus_kata.php")  
    Call<ResponseKata> hapusKata(  
        @Field("id") String id  
    );  
  
    @FormUrlEncoded  
    @POST("detail_kata.php")  
    Call<ResponseKata> detailKata(  
        @Field("id") String id  
    );  
  
    @FormUrlEncoded  
    @POST("update_kata.php")  
    Call<ResponseKata> updateKata(  
        @Field("id") String id,  
        @Field("nama") String nama  
    );  
  
    @FormUrlEncoded  
    @POST("typo_check.php")  
    Call<ResponseKata> typoCheck(  
        @Field("nama") String nama  
    );  
}
```

Retro Sever

```
public class RetroServer {  
    public static final String baseURL =  
        "http://192.168.45.133/start1/api/";  
    private static Retrofit retro;  
  
    public static Retrofit koneksiRetrofit(){  
        if (retro == null){  
            retro = new Retrofit.Builder()  
                .baseUrl(baseURL)  
                .addConverterFactory(GsonConverterFactory.create())  
                .build();  
        }  
    }  
}
```

```

    }
    return retro;
}
}

```

Detail Kata

```

public class detail_kata extends AppCompatActivity {

    private FloatingActionButton fabTambahKata;
    private Toolbar toolbar;
    private SwipeRefreshLayout srlData;
    private RecyclerView rvData;

    private RecyclerView.Adapter adData;
    private RecyclerView.LayoutManager lmData;
    private List<ModelKata> listKata = new ArrayList<>();
    private String idKata, kata;

    @SuppressWarnings("RestrictedApi")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail_kata);

        toolbar = findViewById(R.id.toolbar);

        //        setSupportActionBar(toolbar);
        //        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        //        getSupportActionBar().setDefaultDisplayHomeAsUpEnabled(true);

        rvData = findViewById(R.id.rv_detail_kata);
        srlData = findViewById(R.id.srl_data);

        fabTambahKata = findViewById(R.id.fab_tambah_data);

        fabTambahKata.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new
Intent(detail_kata.this, tambah_kata.class));
            }
        });

        lmData = new
LinearLayoutManager(this, LinearLayoutManager.VERTICAL, false);
        rvData.setLayoutManager(lmData);

        srlData.setOnRefreshListener(new
SwipeRefreshLayout.OnRefreshListener() {
            @Override
            public void onRefresh() {

```

```

        srlData.setRefreshing(true);
        tampilData();
        srlData.setRefreshing(false);
    }
    });
}

public void tampilData() {
    RegistrasiAPI regData =
RetroServer.koneksiRetrofit().create(RegistrasiAPI.class);
    Call<ResponseKata> tmpData = regData.tampilKata();

    tmpData.enqueue(new Callback<ResponseKata>() {
        @Override
        public void onResponse(Call<ResponseKata> call,
Response<ResponseKata> response) {
            listKata = response.body().getResult();

            adData = new
AdapterKata(detail_kata.this, listKata);
            rvData.setAdapter(adData);
            adData.notifyDataSetChanged();
        }

        @Override
        public void onFailure(Call<ResponseKata> call,
Throwable t) {
            Toast.makeText(detail_kata.this, "Gagal
Terhubung dengan server : " + t.getMessage(),
Toast.LENGTH_SHORT).show();
        }
    });
}

@Override
protected void onResume() {
    super.onResume();
    tampilData();
}

@Override
public void onBackPressed() {
    finish();
    super.onBackPressed();
}

@Override
public boolean onSupportNavigateUp() {
    onBackPressed();
    return super.onSupportNavigateUp();
}
}

```

Main Activity

```
public class MainActivity extends AppCompatActivity implements
View.OnClickListener {

    ConstraintLayout btnDetail, btnCheck, btnProfil;
    Dialog dialog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        btnDetail = findViewById(R.id.btn_detail);
        btnCheck = findViewById(R.id.btn_check);
        btnProfil = findViewById(R.id.btn_profil);

        dialog = new Dialog(this);

        btnDetail.setOnClickListener(this);
        btnCheck.setOnClickListener(this);
        btnProfil.setOnClickListener(this);
    }

    @Override
    public void onClick(View view) {
        switch (view.getId()) {
            case R.id.btn_detail:
                // Toast.makeText(this, "Detail Kata!",
                Toast.LENGTH_SHORT).show();
                startActivity(new
                Intent(getApplicationContext(), detail_kata.class));
                break;
            case R.id.btn_check:
                // Toast.makeText(this, "Button Check!",
                Toast.LENGTH_SHORT).show();
                startActivity(new
                Intent(getApplicationContext(), TypoCheck.class));
                break;
            case R.id.btn_profil:
                // Toast.makeText(this, "Button Profil!",
                Toast.LENGTH_SHORT).show();
                openDialogProfile();
                break;
        }
    }

    private void openDialogProfile() {
        dialog.setContentView(R.layout.open_dialog);
        dialog.getWindow().setBackgroundDrawable(new
        ColorDrawable(Color.TRANSPARENT));

        ImageView ivClose = dialog.findViewById(R.id.iv_close);

        ivClose.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
```

```

        dialog.dismiss();
    }
});

    dialog.show();
}
}

```

Tambah Kata

```

public class tambah_kata extends AppCompatActivity {

    private Toolbar toolbar;
    private EditText nama;
    private Button btnTambah;

    private String txtNama;

    @SuppressWarnings("RestrictedApi")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.tambah_kata);

        toolbar = findViewById(R.id.toolbar);

        //        setSupportActionBar(toolbar);
        //        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        //        getSupportActionBar().setDefaultDisplayHomeAsUpEnabled(true);

        nama = findViewById(R.id.et_kata);
        btnTambah = findViewById(R.id.btn_tambah);

        btnTambah.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View view) {
                txtNama = nama.getText().toString();
                if (txtNama.trim().equals("")) {
                    nama.setError("Kata harus di isi !");
                    nama.requestFocus();
                } else {
                    createData();
                }
            }

            private void createData() {
                RegistrasiAPI ardData =
                RetroServer.koneksiRetrofit().create(RegistrasiAPI.class);
                Call<ResponseKata> simpanData =
                ardData.tambahKata(txtNama);

                simpanData.enqueue(new Callback<ResponseKata>()

```

```

{
    @Override
    public void onResponse(Call<ResponseKata>
call, Response<ResponseKata> response) {
        int kode = response.body().getKode();
        String pesan =
response.body().getPesan();

        Toast.makeText(tambah_kata.this, "Kode =
"+kode+" | Pesan = "+pesan, Toast.LENGTH_SHORT).show();
        finish();
    }

    @Override
    public void onFailure(Call<ResponseKata>
call, Throwable t) {
        Toast.makeText(tambah_kata.this, "Gagal
menghubungi server = "+t.getMessage(),
Toast.LENGTH_SHORT).show();
    }
}

});
}

@Override
public void onBackPressed() {
    finish();
    super.onBackPressed();
}

@Override
public boolean onSupportNavigateUp() {
    onBackPressed();
    return super.onSupportNavigateUp();
}
}

```

Typo Check

```

public class TypoCheck extends AppCompatActivity {

    private Toolbar toolbar;

    private EditText etCheck, etHasil;
    private String kataCheck;
    private Button btnCheck;
    @SuppressWarnings("RestrictedApi")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.typo_check);

        toolbar = findViewById(R.id.toolbar);
    }
}

```

```

//          getSupportActionBar(toolbar);
//          getSupportActionBar().setDisplayHomeAsUpEnabled(true);
//
getSupportActionBar().setDefaultDisplayHomeAsUpEnabled(true);

        etCheck = findViewById(R.id.et_check);
        etHasil = findViewById(R.id.et_hasil);
        btnCheck = findViewById(R.id.btn_check);

        btnCheck.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                kataCheck = etCheck.getText().toString();
                if (kataCheck.trim().equals("")) {
                    etCheck.setError("Kata yang dicari masih
kosong !");

                    etHasil.setText("");
                    etCheck.requestFocus();
                } else {
                    cariKata(kataCheck);
                }
            }
        });
    }

    private void cariKata(String kata) {
        RegistrasiAPI regAPI =
RetroServer.koneksiRetrofit().create(RegistrasiAPI.class);
        Call<ResponseKata> tampilKata = regAPI.typoCheck(kata);

        tampilKata.enqueue(new Callback<ResponseKata>() {
            @Override
            public void onResponse(Call<ResponseKata> call,
Response<ResponseKata> response) {
                int kode = response.body().getKode();
                if (kode == 1) {
                    etHasil.setText(response.body().getHasil());
                } else {
                    Toast.makeText(TypoCheck.this,
response.body().getPesanan(), Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<ResponseKata> call,
Throwable t) {
                Toast.makeText(TypoCheck.this, "Gagal
menghubungi Server : "+t.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        });
    }

    @Override
    public void onBackPressed() {
        finish();
        super.onBackPressed();
    }

```



```

    }

    @Override
    public boolean onSupportNavigateUp() {
        onBackPressed();
        return super.onSupportNavigateUp();
    }
}

```

Ubah Kata

```

public class UbahKata extends AppCompatActivity {

    private Toolbar toolbar;
    EditText etNama;
    Button btnUbah;

    String idKata, nama;

    @SuppressWarnings("RestrictedApi")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.ubah_kata);

        toolbar = findViewById(R.id.toolbar);

        //        setSupportActionBar(toolbar);
        //        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        //        getSupportActionBar().setDefaultDisplayHomeAsUpEnabled(true);

        etNama = findViewById(R.id.et_kata);
        btnUbah = findViewById(R.id.btn_ubah);

        Intent detailKata = getIntent();
        etNama.setText(detailKata.getStringExtra("nama"));
        idKata = detailKata.getStringExtra("id");

        btnUbah.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                nama = etNama.getText().toString();
                ubahKata(idKata, nama);
            }
        });
    }

    private void ubahKata(String idKata, String nama) {
        RegistrasiAPI ardData =
        RetroServer.koneksiRetrofit().create(RegistrasiAPI.class);
        Call<ResponseKata> ubahData = ardData.updateKata(idKata,
        nama);
        ubahData.enqueue(new Callback<ResponseKata>() {

```

```

        @Override
        public void onResponse(Call<ResponseKata> call,
Response<ResponseKata> response) {
            Toast.makeText(UbahKata.this,
response.body().getPesan(), Toast.LENGTH_SHORT).show();
            startActivity(new
Intent(getApplicationContext(), detail_kata.class));
            finish();
        }

        @Override
        public void onFailure(Call<ResponseKata> call,
Throwable t) {
            Toast.makeText(UbahKata.this, "Gagal Menghubungi
Server : "+t.getMessage(), Toast.LENGTH_SHORT).show();
        }
    });
    // Toast.makeText(UbahKata.this, "Data : "+ idKata+" |
"+nama, Toast.LENGTH_SHORT).show();
}

    @Override
    public void onBackPressed() {
        finish();
        super.onBackPressed();
    }

    @Override
    public boolean onSupportNavigateUp() {
        onBackPressed();
        return super.onSupportNavigateUp();
    }
}

```

● 29% Overall Similarity

Top sources found in the following databases:

- 29% Internet database
- Crossref database
- 4% Submitted Works database
- 4% Publications database
- Crossref Posted Content database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	123dok.com	Internet	3%
2	media.neliti.com	Internet	3%
3	eprints.polsri.ac.id	Internet	2%
4	andi.ddns.net	Internet	2%
5	scribd.com	Internet	1%
6	es.scribd.com	Internet	1%
7	repository.ittelkom-pwt.ac.id	Internet	1%
8	repository.stmikroyal.ac.id	Internet	1%

9	github.com Internet	1%
10	jurusan.tik.pnj.ac.id Internet	<1%
11	fti.uajy.ac.id Internet	<1%
12	lorca.act.uji.es Internet	<1%
13	ejurnal.unisan.ac.id Internet	<1%
14	repository.uin-suska.ac.id Internet	<1%
15	docobook.com Internet	<1%
16	kc.umn.ac.id Internet	<1%
17	adoc.pub Internet	<1%
18	ejournal-binainsani.ac.id Internet	<1%
19	repository.uncp.ac.id Internet	<1%
20	repository.president.ac.id Internet	<1%

21	widuri.raharja.info	<1%
	Internet	
22	library.polmed.ac.id	<1%
	Internet	
23	repository.upbatam.ac.id	<1%
	Internet	
24	repository.unmuhjember.ac.id	<1%
	Internet	
25	soengsouy.com	<1%
	Internet	
26	join.if.uinsgd.ac.id	<1%
	Internet	
27	repository.atmaluhur.ac.id	<1%
	Internet	
28	ejurnal.ung.ac.id	<1%
	Internet	
29	eprints.umk.ac.id	<1%
	Internet	
30	etd.umy.ac.id	<1%
	Internet	
31	hiperaktif.org	<1%
	Internet	
32	jom.fti.budiluhur.ac.id	<1%
	Internet	

33	core.ac.uk Internet	<1%
34	eprints.poltektegale.ac.id Internet	<1%
35	researchgate.net Internet	<1%

DAFTAR RIWAYAT HIDUP



Nama : Aditya S Humalangi
Tempat,Tgl Lahir : Gorontalo, 20 Maret 1998
Pekerjaan : Mahasiswa
Email : aditpakpaw@gmail.com

Riwayat Pendidikan :

1. Tahun 2010, menyelesaikan Pendidikan di Sekolah Dasar Negeri 69 Kota Timur Kota Gorontalo.
2. Tahun 2013, menyelesaikan Pendidikan di Sekolah Menengah Pertama Muhammadiyah 1 Gorontalo.
3. Tahun 2016, menyelesaikan Pendidikan di Sekolah Menengah Kejuruan Negeri 3 Gorontalo, Kota Gorontalo.
4. Tahun 2016, telah diterima menjadi Mahasiswa di Perguruan Tinggi Swasta Universitas Ichsan Gorontalo