

**IMPLEMENTASI ALGORITMA BOYER MOORE
UNTUK KAMUS BAHASA INDONESIA
GORONTALO**

Oleh
ISHAK HELUWA
T31 16 050

SKRIPSI

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



**PROGRAM SARJANA
TEKNIK INFORMATIKA
UNIVERSITAS ICHSAN GORONTALO
GORONTALO
2020**

HALAMAN JUDUL

IMPLEMENTASI ALGORITMA BOYER MOORE
UNTUK KAMUS BAHASA INDONESIA
GORONTALO

Oleh

ISHAK HELUWA

T31 16 050

SKRIPSI

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



PROGRAM SARJANA
TEKNIK INFORMATIKA
UNIVERSITAS ICHSAN GORONTALO
GORONTALO
2020

PERSETUJUAN SKRIPSI

IMPLEMENTASI ALGORITMA BOYER MOORE **UNTUK KAMUS BAHASA INDONESIA** **GORONTALO**

Oleh

ISHAK HELUWA

T31 16 050

SKRIPSI

Untuk memenuhi salah satu syarat ujian
Guna memperoleh gelar sarjana
Program Studi Teknik Informatika,
ini telah disetujui oleh Tim Pembimbing

Gorontalo, 23 Juli 2020

Pembimbing I



Moh. Effendi Lasulika, M.Kom
NIDN. 0929048902

Pembimbing II



Yulianty Lasena, M. Kom
NIDN. 0907078603

PENGESAHAN SKRIPSI

IMPLEMENTASI ALGORITMA BOYER MOORE UNTUK KAMUS BAHASA INDONESIA GORONTALO

Oleh

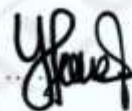
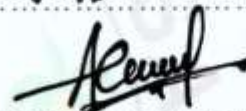
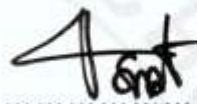
ISHAK HELUWA

T31 16 050

Diperiksa oleh Panitia Ujian Strata Satu (S1)
Universitas Ichsan Gorontalo

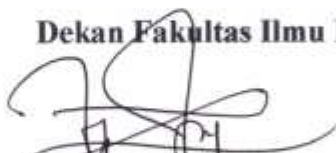
Gorontalo, 06 Agustus 2020

1. Penguji I
Asmaul Husna N, M.Kom
2. Penguji II
Apriyanto Alhamad, M.Kom
3. Penguji III
Maryam Hasan, M.Kom
4. Pembimbing I
Moh. Effendi Lasulika, M.Kom
5. Pembimbing II
Yulianty Lasena, M.Kom



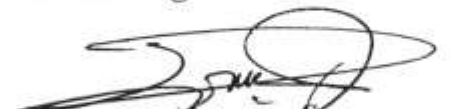
Mengetahui

Dekan Fakultas Ilmu Komputer



Zohrahayaty, M.Kom
NIDN. 0912117702

Ketua Program Studi



Irvan Abraham Salihi, M.Kom
NIDN. 0928028101

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, 23 Juli 2020

Yang Membuat Pernyataan,



Ishak Heluwa

ABSTRACT

The dictionary is a tool used to translate a language. Gorontalo or Hulondalo is a regional language used by residents of Gorontalo Province. The use of regional languages has decreased in use in everyday communication languages. Preserving regional languages is very necessary amid the increasingly rapid technological development. For this reason, the research wants to design an Indonesian Gorontalo dictionary application based on Android using Boyer Moore's Algorithm. This application is designed to perform a word search functionality Indonesia - Gorontalo and to show the meaning of words into the language Gorontalo Indonesian and vice versa. Based on the results of the test table, the search accuracy can be concluded that the level of search accuracy in the Indonesian Gorontalo dictionary application reaches 100% because the amount of data obtained is the same as the data in the database.

Keyword: *Dictionary, Indonesia-Gorontalo Language, Android, Boyer-Moore Algorithm*

ABSTRAK

Kamus merupakan alat bantu yang digunakan untuk menterjemah suatu bahasa. 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. Melestarikan bahasa daerah sangat perlu ditengah perkembangan teknologi yang semakin pesat. Untuk itu penelitian ingin merancang sebuah aplikasi kamus Indonesia Gorontalo yang berbasis *Android* dengan menggunakan Algoritma Boyer Moore. Aplikasi kamus ini yang dirancang dapat melakukan fungsi pencarian kata Indonesia - Gorontalo dan menampilkan arti dari kata bahasa Indonesia kedalam bahasa Gorontalo dan sebaliknya. Berdasarkan tabel hasil pengujian akurasi pencarian dapat disimpulkan bahwa tingkat akurasi pencarian pada aplikasi kamus Indonesia Gorontalo mencapai 100% karena jumlah data yang didapat sama dengan data yang ada didalam *database*.

Kunci: *Kamus, Bahasa Indonesia-Gorontalo, Kamus, Android, Algoritma Boyer-Moore*

KATA PENGANTAR

Alhamdulillah, penulis dapat menyelesaikan usulan penelitian ini dengan Judul: **“Implementasi Algoritma Boyer Moore Untuk Kamus Bahasa Indonesia Gorontalo”**, untuk memenuhi salah satu syarat penyusunan Skripsi Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.

Penulis menyadari sepenuhnya bahwa skripsi 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. Bapak Muhammad Ichsan Gaffar, SE, MAK, 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. Ibu Zohrahayaty, S.Kom, M.Kom, selaku Dekan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
4. Bapak Sudirman S.Panna, 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. Bapak Sudirman Melangi, S.Kom, M.Kom, selaku Pembantu Dekan III Bidang Kemahasiswaan Fakultas Ilmu komputer Universitas Ichsan Gorontalo;
7. Bapak Irvan A. Salihi, S.Kom, M.Kom, selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
8. Bapak Moh. Effendi Lasulika, S.Kom,M.Kom, selaku Pembimbing I, yang telah membantu memberikan arahan dan masukan kepada penulis;

9. Ibu Yulianty Lasena, S.Kom,M.Kom, selaku Pembimbing II, yang telah membantu memberikan arahan dan masukan kepada penulis;
10. Bapak dan Ibu Dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan berbagai disiplin ilmu kepada penulis;
11. Kedua Orang Tua saya yang tercinta, atas segala kasih sayang, jerih payah dan doa restunya dalam membesarkan dan mendidik penulis;
12. Rekan-rekan seperjuangan yang telah banyak memberikan bantuan dan dukungan moril yang sangat besar kepada penulis;
13. Dan kepada istri saya yang tercinta, atas segala kasih sayangnya, yang telah memberikan semangat dan dorongan kepada penulis;
14. 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, 23 Juli 2020

Ishak Heluwa

DAFTAR ISI

HALAMAN JUDUL.....	i
PERSETUJUAN SKRIPSI	ii
PENGESAHAN SKRIPSI	Error! Bookmark not defined.
PERNYATAAN SKRIPSI.....	iii
ABSTRACT.....	v
ABSTRAK	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
DAFTAR LAMPIRAN	xvi
BAB I PEMBAHASAN.....	1
1.1 Latar Belakang Masalah	1
1.2 Identifikasi Masalah	3
1.3 Rumusan Masalah	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
BAB II LANDASAN TEORI	5
2.1 Tinjauan Studi	5
2.2 Tinjauan pustaka.....	7
2.2.1 Pengertian Kamus	7
2.2.2 Kamus <i>Elektronik</i>	7
2.2.3 <i>Boyer Moore</i>	7
2.2.4 Aplikasi	11
2.2.5 <i>Android</i>	11
2.2.6 Android Studio	12
2.2.7 Xampp	12
2.2.8 <i>Java Development Kit (JDK)</i>	12
2.2.9 PHP	12

2.2.10	<i>Software Development Kit (SDK)</i>	13
2.2.11	<i>Android Virtual Device (AVD)</i>	13
2.2.12	Perangkat Lunak Pendukung.....	13
2.2.13	<i>Unified Modelling Language (UML)</i>	14
2.2.14	<i>System Development Life Cycle (SDLC)</i>	18
2.2.15	Analisis Sistem.....	19
2.2.16	Desain Sistem.....	19
2.2.17	<i>Implementasi</i>	20
2.2.18	Pengujian Sistem.....	20
2.3	Kerangka Pikir.....	24
BAB III METODE PENELITIAN.....		25
3. 1	Jenis, Metode, Subjek, Objek, Waktu Dan Lokasi Penelitian.....	25
3. 2	Pengumpulan Data.....	25
3. 3	Pengembangan Sistem.....	27
3. 4. 1	Analisis Sistem.....	28
3. 4. 2	Desain Sistem.....	28
3. 4. 3	Konstruksi Sistem.....	29
3. 4. 4	Pengujian Sistem.....	30
BAB IV HASIL PENELITIAN		31
4.1	Hasil Pengumpulan Data	31
4.1.1	Penerapan Metode	32
4.2	Hasil Pengembangan Sistem	38
4.3	<i>Activity Diagram Login</i>	39
4.4	<i>Activity Diagram Dashboard</i>	39
4.5	<i>Activity Diagram Daftar Kata</i>	40
4.6	<i>Activity Diagram Data User</i>	40
4.7	<i>Sequence Diagram Login Admin</i>	41
4.8	<i>Sequence Diagram Dashboard</i>	41
4.9	<i>Sequence Diagram Daftar Kata</i>	42
4.10	<i>Sequence Diagram User</i>	42
4.11	Arsitektur Sistem.....	43

4.12	<i>Interface Design</i>	43
4.12.1	Mekanisme User.....	43
4.12.2	Mekanisme navigasi <i>Home</i>	43
4.12.3	Mekanisme <i>Login</i>	44
4.12.4	Mekanisme <i>Input</i> data Kata	45
4.12.5	Mekanisme Edit Data Kata	45
4.12.6	Mekanisme Tambah Data User	45
4.12.7	Mekanisme Edit Data User	46
4.12.8	Mekanisme <i>Output</i>	46
4.13	Data Desain	47
4.13.1	Struktur Data	47
4.14	Hasil Pengujian Sistem.....	48
4.14.1	Pengujian <i>White Box</i>	48
4.14.2	<i>Flowchart</i>	49
4.14.3	Pengujian <i>White Box</i>	50
4.14.4	Perhitungan CC pada pengujian <i>White Box</i>	51
4.14.5	Perhitungan CC pada pengujian <i>White Box</i>	51
4.14.6	Pengujian <i>Black Box</i>	52
4.14.7	Pengujian Akurasi Algoritma Boyer Moore	54
BAB V HASIL DAN PEMBAHASAN.....		57
5.1	Pembahasan Sistem	57
5.1.1	Tampilan Halaman <i>Login</i>	57
5.1.2	Tampilan Halaman Dashboard Admin.....	57
5.1.3	Tampilan Halaman Data Kata	58
5.1.4	Tampilan Halaman <i>Export</i> Data Kata <i>File</i> Exel	58
5.1.5	Tampilan Halaman <i>Import</i> Data Kata <i>File</i> Excel	59
5.1.6	Tampilan Halaman Tambah Data Kata.....	59
5.1.7	Tampilan Halaman Edit Data Kata	60
5.1.8	Tampilan Halaman Data User.....	60
5.1.9	Tampilan Halaman Tambah Data User	61
5.1.10	Tampilan Halaman Edit Data User	61

5.1.11	Tampilan Halaman User	62
5.1.12	Tampilan Halaman Data Profil	62
5.1.13	Tampilan Halaman Algoritma Boyer-Moore.....	63
5.1.14	Tampilan Halaman Kamus Indonesia - Gorontalo.....	63
5.1.15	Tampilan Halaman Kamus Indonesia – Gorontalo <i>speech to text</i> ..	64
5.1.16	Tampilan Halaman Kamus Gorontalo - Indonesia.....	64
BAB VI KESIMPULAN DAN SARAN		65
6.1	Kesimpulan.....	65
6.2	Saran	65
DAFTAR PUSTAKA		
LAMPIRAN		

DAFTAR GAMBAR

Gambar 2. 1: <i>Flowchart</i> Boyer Moore.....	11
Gambar 2. 2: Bagan Alir	21
Gambar 2. 3: Aliran Grafik	21
Gambar 2. 4: Kerangka Fikir	24
Gambar 4. 1: <i>Use Case Diagram</i> Halaman Admin	38
Gambar 4. 2: <i>Use Case Diagram</i> Kamus Bahasa Indonesia Gorontalo	38
Gambar 4. 3: <i>Activity Diagram</i> Proses <i>Login</i>	39
Gambar 4. 4: <i>Activity Diagram</i> Dashboard.....	39
Gambar 4. 5: <i>Activity Diagram</i> Daftar Kata	40
Gambar 4. 6: <i>Activity Diagram</i> User.....	40
Gambar 4. 7: <i>Sequence Diagram</i> <i>Login</i> Admin.....	41
Gambar 4. 8: <i>Sequence Diagram</i> Dashboard.....	41
Gambar 4. 9: <i>Sequence Diagram</i> Daftar Kata	42
Gambar 4. 10: <i>Sequence Diagram</i> User.....	42
Gambar 4. 11: Mekanisme Navigasi <i>Home</i> Admin	43
Gambar 4. 12: Mekanisme Navigasi <i>Home</i> User.....	44
Gambar 4. 13: Mekanisme <i>Login</i>	44
Gambar 4. 14: Mekanisme <i>Input</i> data Kata	45
Gambar 4. 15: Mekanisme Edit Data Kata	45
Gambar 4. 16: Mekanisme Tambah Data User.....	45
Gambar 4. 17: Mekanisme Edit Data User	46
Gambar 4. 18: Mekanisme <i>Output</i> Kamus.....	46
Gambar 4. 19: <i>Flowchart</i> Algoritma Boyer Moore.....	49
Gambar 4. 20: Flowgraph Algoritma Boyer Moore.....	50
Gambar 5. 1: Tampilan Halaman <i>Login</i>	57
Gambar 5. 2: Tampilan Halaman Dashboard Admin.....	57
Gambar 5. 3: Tampilan Halaman Data Kata.....	58
Gambar 5. 4: Tampilan Halaman <i>Export</i> Data Kata File Exel	58
Gambar 5. 5: Tampilan Halaman <i>Import</i> Data Kata File Excel.....	59

Gambar 5. 6: Tampilan Halaman Tambah Data Kata.....	59
Gambar 5. 7: Tampilan Halaman Edit Data Kata	60
Gambar 5. 8: Tampilan Halaman Data User	60
Gambar 5. 9: Tampilan Halaman Tambah Data User.....	61
Gambar 5. 10: Tampilan Halaman Edit Data User	61
Gambar 5. 11: Tampilan Home User	62
Gambar 5. 12: Halaman data Profil.....	62
Gambar 5. 13: Halaman Algoritma Boyer Moore	63
Gambar 5. 14: Halaman Kamus Indonesia – Gorontalo	63
Gambar 5. 15 Halaman Kamus Indonesia – Gorontalo <i>speech to text</i>	64
Gambar 5. 16: Halaman Kamus Gorontalo - Indonesia.....	64

DAFTAR TABEL

Tabel 2. 1: Tinjauan Studi.....	5
Tabel 2. 2: <i>Occurrence Heuristic</i>	9
Tabel 2. 3: <i>Match Heuristic</i>	10
Tabel 2. 4: <i>Match Heuristic</i>	10
Tabel 2. 5: <i>Match Heuristic</i>	10
Tabel 2. 6: Pencarian <i>Match Heuristic</i>	11
Tabel 2. 7: Perangkat Lunak Pendukung	13
Tabel 2. 8: Simbol Simbol <i>Use Case Diagram</i>	14
Tabel 2. 9: Simbol Simbol <i>Activity Diagram</i>	16
Tabel 2. 10: Simbol Simbol <i>Diagram Class</i>	17
Tabel 2. 11: Simbol – Simbol <i>Sequence Diagram</i>	18
Tabel 4. 1: Hasil Pengumpulan Data.....	31
Tabel 4. 2: Daftar Pencacah Prefix dan Suffix dari Pattern	34
Tabel 4. 3: Pemberian Nilai Pencacahan.....	34
Tabel 4. 4: Pencarian Nilai Match Heuristic (OH)	34
Tabel 4. 5: Pemberian Nilai (OH) dan (MH)	37
Tabel 4. 6: Pencarian Pencocokan Karakter	37
Tabel 4. 7: Hasil Pencarian Pencocokan Karakter	37
Tabel 4. 8: Mekanisme User	43
Tabel 4. 9: Tabel Kamus	47
Tabel 4. 10: Tabel User.....	47
Tabel 4. 11: Basis Path.....	51
Tabel 4. 12: Tabel Pengujian Black Box	52
Tabel 4. 13: Pengujian Akurasi.....	55

DAFTAR LAMPIRAN

Jadwal Penelitian

Data Yang Dikumpulkan

Surat Keterangan Telah Melakukan Penelitian

Kode Program

Hasil Turnitin

Berita Acara Perbaikan / Revisi Ujian

Daftar Riwayat Hidup

BAB I

PEMBAHASAN

1.1 Latar Belakang Masalah

Bahasa daerah merupakan unsur yang mempunyai peranan penting dalam kebudayaan, antara lain menjadi kebanggaan suatu daerah, identitas daerah, dan media penghubung untuk komunikasi dikeluarga 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 [1].

Setiap daerah mempunyai bahasa komunikasi berbeda - beda yang digunakan untuk melakukan komunikasi dengan orang lain. Untuk memahami bahasa komunikasi yang digunakan oleh setiap daerah, diperlukan suatu alat yang dapat memberikan kemudahan dalam mengartikan kata yang ada, yaitu kamus sehingga perbedaan bahasa tidak menjadi suatu kendala dalam melakukan komunikasi. Sehingga yang bukan penduduk asli Gorontalo atau yang mempunyai keinginan untuk belajar dan berkomunikasi dengan menggunakan bahasa daerah Gorontalo akan lebih mudah mengetahui bahasa daerah Gorontalo.

Bahasa Gorontalo atau bahasa Hulondalo merupakan bahasa daerah yang digunakan oleh penduduk Provinsi Gorontalo. Masyarakat Gorontalo sekarang ini sudah tidak mengenal lagi bahasa daerah Gorontalo, penyebab utamanya dikarenakan pengaruh bahasa pergaulan yang bebas sehingga membuat menghilangnya bahasa daerah Gorontalo, seperti yang terlihat pada kalangan generasi muda [2]. Meskipun selama ini pemerintah telah melakukan berbagai upaya pelestarian bahasa daerah Gorontalo, terutama penggunaannya oleh anak-anak dan remaja, antara lain melalui program siaran bahasa Gorontalo di Radio Republik Indonesia (RRI) Gorontalo, kurikulum muatan lokal bahasa daerah Gorontalo yang diajarkan di sekolah-sekolah, termasuk di lembaga PAUD, namun realita menunjukkan penggunaan bahasa daerah Gorontalo didalam kalangan masyarakat khususnya di kota Gorontalo masih sangat memprihatinkan. Sangat

sedikit ditemukan didalam masyarakat yang menggunakan bahasa daerah Gorontalo dalam komunikasi sehari-hari [3]. Jika Kondisi ini tidak dicarikan solusinya maka dikhawatirkan bahasa daerah Gorontalo akan hilang dengan sendirinya.

Hasil dari suatu penelitian (Fatmah, 1994) menunjukkan semakin rendahnya penguasaan bahasa daerah Gorontalo oleh generasi muda. Sebagian dari mereka hanya mampu memahami apa yang diucapkan orang lain, tetapi sangat sulit menggunakannya secara lisan. Sebagian lagi tidak dapat memahami, serta mampu menggunakannya dalam komunikasi sehari – hari [4].

Hasil penelitian (Baruadi, 2014) juga mengatakan dalam pendapatnya bahwa dewasa ini sangat sedikit warga masyarakat Gorontalo utamanya anak-anak dan remaja, bahkan orang tua khususnya di perkotaan yang menggunakan bahasa Gorontalo dalam komunikasi sehari-hari. Jika kondisi seperti sekarang ini dibiarkan berlarut-larut maka bahasa daerah Gorontalo akan mengalami kepunahan [3]. Hal ini dipertegas kembali dalam sebuah penelitian (Maryam Rahim, 2018) jika dianalisis ternyata tidak jauh berbeda dengan kondisi sekarang, bahkan diduga kondisinya menjadi lebih memprihatinkan. Hal ini didasari oleh pemikiran semakin bervariasinya media yang memberi peluang berkembangnya bahasa gaul di kalangan generasi muda, dan di sisi lain mempersempit peluang penggunaan bahasa daerah Gorontalo [5].

Untuk melestarikan kembali bahasa daerah Gorontalo dan membuat masyarakat Gorontalo kembali mengenal dan memahami lagi bahasa daerah Gorontalo atau penduduk yang berasal dari daerah lain yang ingin mengenal dan memahami bahasa daerah Gorontalo atau ingin berkomunikasi dengan bahasa daerah gorontalo. saya mencoba untuk memberikan solusi dengan membuat aplikasi kamus bahasa Indonesia – Gorontalo berbasis *mobile* dengan menerapkan Algoritma Boyer Moore dalam pencarian kata.

Sekarang ini banyak sekali kamus yang beredar, mulai dari bentuk buku, kamus *elektronik*, aplikasi *desktop*, *web* dan *mobile* kamus. Dari semua itu, masing masing memiliki kelemahan dan juga kelebihan. Contohnya kamus dalam bentuk buku memiliki kelebihan dalam jumlah kosa kata yang banyak, tetapi juga

memiliki kelemahan dalam hal pencarian kosa kata dan arti kata yang membutuhkan waktu yang lama.

Berdasarkan berbagai penelitian terkait tentang aplikasi kamus dengan menerapkan Algoritma *Boyer Moore* Adapun penelitian yaitu tentang aplikasi kamus Bahasa Indonesia – Arab berbasis *android* untuk melakukan pencarian *String* dengan menerapkan Algoritma *Boyer Moore* dengan memberikan hasil yang lebih baik. Algoritma *Boyer Moore* merupakan Algoritma pencarian *String* yang paling efisien dan banyak digunakan dalam pencarian *String* [6]. Algoritma *Boyer Moore* juga merupakan algoritma pencarian *String* mulai dari kanan ke kiri, jika terjadi ketidakcocokan saat perbandingan string maka akan membuat pergerakan pattern melompat lebih jauh untuk menghindari karakter pada string yang diperkirakan gagal [7].

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 *platform open source* yang memberikan kebebasan pada *developer* untuk mengembangkan sebuah aplikasi tersendiri [8].

Berdasarkan pemaparan dan penjelasan diatas, maka penulis mengambil judul “ **Implementasi Algoritma Boyer Moore Untuk Kamus Bahasa Indonesia Gorontalo** ” yang insyaallah dapat membantu masyarakat untuk mengenal dan memahami kembali bahasa daerah Gorontalo. Aplikasi kamus ini melakukan pencarian arti perkata dan dilengkapi dengan *Speech To Text* dimana pengguna aplikasi kamus tinggal mengucapkan kata yang ingin diterjemahkan.

1.2 Identifikasi Masalah

Berdasarkan penjelasan diatas, maka identifikasi masalah dalam penelitian ini yaitu, kurangnya pemahaman masyarakat dalam bahasa daerah Gorontalo.

1.3 Rumusan Masalah

Berdasarkan penjelasan diatas, maka rumusan masalah dalam penelitian ini yaitu, bagaimana cara menerapkan Algoritma *Boyer Moore* untuk mencocokkan kata bahasa Indonesia dan Gorontalo ?

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah, maka tujuan dalam penelitian ini yaitu, diharapkan penelitian ini dapat menerapkan Algoritma *Boyer Moore* dalam melakukan pencocokan kata bahasa Indonesia dan 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. Manfaat Teoritis

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

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: Tinjauan Studi

NO	PENELITI	JUDUL	TAHUN	METODE	HASIL
1	Ernawati, Saifudin	Perancangan aplikasi pembelajaran bahasa jawa berbasis <i>android</i>	2018	<i>Brute force</i> dan <i>Binery Search</i>	dapat menerjemahkan bahasa pemrograman yang dapat di eksekusi dan dimengerti oleh mesin. Dan setelah dilakukan pengujian <i>Black Box</i> aplikasi tersebut sangat efisien.
2	Aziz, Harafani	Aplikasi kamus bahasa betawi berbasis <i>Android</i> menggunakan metode <i>Sequencial Search</i>	2016	<i>Sequencial Search</i>	Lebih efisien dan efektif dalam menterjemahkan bahasa betawi sehingga memudahkan pengguna dalam mempelajari bahasa betawi

NO	PENELITI	JUDUL	TAHUN	METODE	HASIL
3	Fadisar, Darnila, E	Aplikasi kamus bahasa indonesia-arab berbasis <i>Android</i> menggunakan metode <i>Boyer Moore</i>	2017	<i>Boyer Moore</i>	Proses pencarian kata yang diinput dengan kata yang ada dalam database menampilkan hasil terjemahan kata yang sangat optimal dan efisien.
4	Rahmanita	Pencarian string menggunakan algoritma <i>Boyer Moore</i> pada dokumen	2015	<i>Boyer Moore</i>	Semakin panjang kata yang dicari, maka semakin hemat waktu yang ditempuh pada file berekstensi .txt. namun pada file berekstensi .doc dan .pdf semakin panjang kata yang dicari maka akan membutuhkan waktu yang lama

2.2 Tinjauan pustaka

2.2.1 Pengertian Kamus

Kamus merupakan buku yang digunakan untuk rujukan yang digunakan untuk menerangkan makna kata-kata, yang berfungsi untuk membantu dan memberikan kemudahan kepada seseorang untuk mengenal perkataan baru. kamus merupakan alat penunjang yang memuat berbagai macam daftar kosa kata atau merupakan gabungan kata dengan keterangan mengenai berbagai seni maknanya dan penggunaanya dalam bahasa dan biasanya disusun dengan abjad.

2.2.2 Kamus *Elektronik*

kamus *elektronik* diproduksi di Jepang pertamakali pada tahun 1979. Secara fisik kamus *elektronik*, bentuknya hampir sama dengan kamus elektronik sekarang ini [9]. kamus *elektronik* merupakan kamus yang menggunakan bahasa pemrograman aplikasi komputer yang merupakan media perangkat lunak [10]. Sekarang ini kamus *elektronik* telah berkembang pesat dan banyak beredar dan diakses oleh kalangan masyarakat, dari kamus *mobile* aplikasi sampai dengan kamus yang berbasis *web*.

2.2.3 *Boyer Moore*

Robert S. Boyer, dan J. Strother Moore adalah yang mendirikan Algoritma *Boyer Moore* pada tahun 1977. Algoritma *Boyer Moore* merupakan salah satu Algoritma yang digunakan untuk melakukan pencarian *string*. Algoritma ini dianggap sebagai algoritma yang paling efisien. Algoritma *Boyer Moore* melakukan pencarian string dengan mencocokkan karakter dari kanan kekiri dan lebih banyak informasi yang didapat dan juga menggunakan dua fungsi Shift yaitu pergeseran akhiran yang baik dan karakter yang buruk apabila karakter tidak sesuai [11].

proses yang dilakukan Algoritma *Boyer Moore* pada saat melakukan mencocokkan string adalah [6]:

- 1 Algoritma *Boyer Moore* mulai mencocokkan pattern pada awal teks.
- 2 Dari kanan ke kiri, Algoritma ini akan mencocokkan karakter per karakter *pattern* dengan karakter di teks yang bersesuaian, sampai dengan salah satu kondisi berikut dipenuhi:
 - a Karakter di *pattern* dan di teks yang dibandingkan tidak cocok
 - b Semua karakter di *pattern* cocok kemudian algoritma akan memberitahukan penemuan di posisi ini.
- 3 Algoritma kemudian menggeser *pattern* dengan memaksimalkan nilai penggeseran *good suffix* dan penggeseran *bad-character*, lalu mengulangi langkah 2 sampai *pattern* berada di ujung teks.

A. Penerapan Metode *Boyer Moore*

1. Buat tabel pergeseran *pattern* yang dicari (P) dengan pendekatan *Match Heuristic* (MH) dan Occurence Heuristic (OH) untuk menentukan jumlah pergeseran yang dilakukan jika mendapat karakter tidak cocok pada proses pencocokan dengan teks (T)
2. Jika dalam proses perbandingan terjadi ketidakcocokan antara pasangan karakter pada *pattern* dan karakter teks, pergeseran dilakukan dengan memilih salah satu nilai pergeseran dari dua tabel, dan memiliki nilai pergeseran paling besar dari tabel *Match Heuristic* dan *Occurence Heuristic*.
3. Dua kemungkinan penyelesaian dalam melakukan pergeseran *pattern*, Jika karakter yang tidak cocok, tidak ada pada *pattern* maka pergeseran adalah sebanyak jumlah karakter pada *pattern*. dan jika karakter yang tidak cocok, ada pada *pattern*, maka banyaknya pergeseran bergantung dari nilai pada tabel *Match Heuristic* dan *Occurence Heuristic*.
4. Jika karakter pada teks yang sedang dibandingkan cocok dengan karakter pada *pattern*, maka posisi karakter pada *pattern* dan teks diturunkan sebanyak 1 posisi, kemudian lanjutkan dengan pencocokan pada posisi

tersebut dan seterusnya. Jika kemudian terjadi ketidakcocokan karakter *pattern* dan teks, maka pilih nilai pergeseran terbesar dari tabel *match heuristic* dan nilai tabel *occurrence heuristic*.

Cara menghitung tabel *Occurrence Heuristic*

Contoh pattern : NANDA

Panjang pattern : 5 karakter

Tabel 2. 2: *Occurrence Heuristic*

Indeks	0	1	2	3	4
Pattern	N	A	N	D	A
OH					

Langkah-langkah pemberian nilainya adalah sebagai berikut :

1. Lakukan perhitungan, **OH** = (**length -1 -index**) length = panjang karakter = 5 karakter
2. Karakter pertama adalah “N” dengan Index = 0 **OH** = (5 – 1 – 0 = 4) maka nilai karakter “N” = 4
3. Karakter kedua adalah “A” dengan index = 1 **OH** = (5 – 1 – 1 = 3) maka nilai karakter “A” = 3
4. Karakter ketiga adalah “N” dengan index = 2 **OH** = (5 – 1 – 2 = 2) maka nilai karakter “N” = 2
5. Karakter keempat adalah “D” dengan index = 3 **OH** = (5 – 1 – 3 = 1) maka nilai karakter “D” = 1
6. Karakter kelima adalah “A” dengan index = 4 **OH** = (5 – 1 – 4 = 0) maka nilai karakter “A” = 0
7. **OH** = (5 – 1 – 4 = 0) maka nilai karakter “A” = 0

Cara menghitung tabel *Match Heuristic*

Contoh Pattern : NANDA

Panjang Pattern : 5 Karakter

Tabel 2. 3: Match Heuristic

Indeks	0	1	2	3	4
Pattern	N	A	N	D	A
MH					

Langkah – langkah pemberian nilainya adalah sebagai berikut :

1. Berikan nilai 1 (nilai default MH) pada index terbesar, dalam kasus ini adalah “A”.
2. Bandingkan karakter kedua, jika karakter “A” sudah ditemukan, tetapi karakter sebelum “A” bukan karakter “D”, maka check apakah karakter yang sudah cocok pada teks ada pattern berikutnya, jika ada maka sejajarkan karakter tersebut dengan karakter yang sudah cocok, dan beri nilai pergeseran ketabel MH, pada kasus ini “A” sudah cocok.

Tabel 2. 4: Match Heuristic

Teks	N	A	N	D	A
Pattern	N	A	N	D	A

Tabel 2. 5: Match Heuristic

Teks	N	A	N	D	A		
Pattern	1	2	N	A	N	D	A

3. Lanjutkan langkah kedua untuk karakter berikutnya sampai akhir *pattern*, ketidakcocokan *pattern*, maka bandingkan karakter yang sudah cocok dengan karakter yang belum dibandingkan pada *pattern*, jika ada maka sejajarkan karakter tersebut, jika tidak maka pergeseranya sejauh panjang karakter, dan beri nilai pergeserannya ketabel MH.

merupakan *system operasi open source* yang memberikan kebebasan pada *developer* untuk mengembangkan sebuah aplikasi tersendiri [13].

2.2.6 Android Studio

Android studio merupakan sebuah *Integrated Development Environment* (IDE) untuk *platform android*. Aplikasi android studio ini diumumkan pada tanggal 16 Mei 2013 pada konferensi Google I/O oleh Produk Meneger Google, Ellie Powers [6].

2.2.7 Xampp

Xampp adalah sebuah *software* yang berfungsi untuk menjalankan website berbasis PHP dan menggunakan pengolah data MySQL di komputer lokal. Xampp berperan sebagai server web pada komputer lokal. Xampp juga dapat disebut sebagai sebuah Cpanel server *virtual*, yang dapat membantu melakukan *preview* sehingga dapat dimodifikasi website tanpa harus terakses dengan internet atau *online* [14].

2.2.8 Java Development Kit (JDK)

Java Development Kit (JDK) merupakan paket fungsi API dari sebuah perangkat lunak yang mengerjakan bahasa pemrograman berbasis java, yang meliputi *Java Runtime Environment* (JRE) dan *Java Virtual Machine* (JVM) [15].

2.2.9 PHP

PHP merupakan bahasa *scripting server – side*, dimana pemrosesan datanya dilakukan pada sisi server. Sederhananya, serverlah yang akan menerjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada client yang melakukan permintaan [14].

2.2.10 Software Development Kit (SDK)

Software Development Kit (SDK) merupakan tools API (*Application Programming Interface*) yang digunakan untuk memulai proses pengembangan aplikasi *android* yang menggunakan bahasa pemrograman java. *Android SDK* ini terdiri dari *debugger*, *libraries*, *hendset emulator*, dokumentasi, kode contoh dan tutorial. SDK memungkinkan pengembang membuat aplikasi *android* yang mencakup proyek sampel dengan kode sumber, perangkat pengembangan, *emulator* dan perpustakaan yang diperlukan untuk membangun aplikasi *android* [15].

2.2.11 Android Virtual Device (AVD)

Android Virtual Device merupakan *emulator* yang digunakan untuk menjalankan program aplikasi *android* yang kita buat. AVD ini selanjutnya digunakan sebagai tempat untuk test dan menjalankan aplikasi *android* tanpa harus menggunakan perangkat *android* yang sebenarnya. Sebelum menggunakan AVD harus menentukan karakteristiknya, misalkan dalam menentukan versi *android*, jenis dan ukuran layar dan besarnya memori. AVD bisa dibuat sebanyak yang kita inginkan [15].

2.2.12 Perangkat Lunak Pendukung

Tabel 2. 7: Perangkat Lunak Pendukung

No	Perangkat Lunak	Keterangan
1	Eclipse	Sebuah IDE (<i>Integrated Development Environment</i>) untuk mengembangkan sebuah perangkat lunak atau tools dalam membuat aplikasi mobile atau aplikasi <i>android</i> dan dapat dijalankan disemua <i>platform</i>
2	Xampp	Digunakan dalam membuat database atau basis data

No	Perangkat Lunak	Keterangan
3	JDK	JDK (<i>java development kit</i>) merupakan sebuah perangkat lunak yang mengerjakan aplikasi berbasis java
4	PHP	Php adalah sebuah bahasa pemrograman <i>web service</i> yang didesain untuk pengembangan <i>web</i> dan merupakan bahasa pemrograman umum

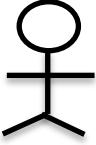
2.2.13 Unified Modelling Language (UML)

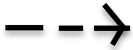
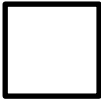
Unified Modelling Language (UML) yaitu standar untuk bahasa grafis dalam pemodelan *software* yang berorientasi objek. Dikembangkan oleh James Rumbaugh, Grady Booch, dan Ivaar Jacobson pada tahun 1990an dengan notasi mereka masing-masing yang kemudian UML akhirnya diresmikan sebagai sebuah standar pemodelan objek oleh *object Management Group* [16]. Adapun diagram yang akan dipergunakan oleh penulis yaitu terdiri dari *Use case Diagram*, *Sequence Diagram*, *Activity Diagram*, dan *Class Diagram*.

A. Use case diagram

Use case diagram merupakan sebuah gambaran *graphical* dari semua actor, use case yang digunakan untuk menjelaskan bagaimana langkah - langkah yang seharusnya dikerjakan oleh sistem dan interaksi antara komponen-komponen tersebut yang memperkenalkan suatu sistem yang akan dibangun [8].

Tabel 2. 8: Simbol Simbol *Use Case Diagram*

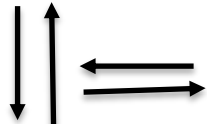
Simbol	Nama	Keterangan
	Actor	Memberikan sfesifikasi himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i>

Simbol	Nama	Keterangan
	Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>)
	Generalization	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>)
	Include	Menspesifikasi bahwa <i>use case</i> sumber secara eksplisit
	Extend	Menspesifikasi bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> 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

B. *Activity diagram*

Diagram ini merupakan tipe khusus dari *diagram state* yang digunakan untuk memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. *Diagram* ini merupakan peranan penting dalam melakukan pemodelan fungsi-fungsi dalam suatu sistem yang dikerjakan dan memberikan tekanan pada aliran antar objek yang terkendali [8].

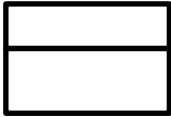
Tabel 2. 9: Simbol Simbol *Activity Diagram*

Simbol	Nama	Keterangan
	Initial State	Proses dimulai pertama kali didalam <i>activity</i>
	Activity	Aktivitas yang terjadi didalam <i>activity</i>
	Action	<i>State</i> 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 <i>activity</i>

C. *Class diagram*

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem [17].

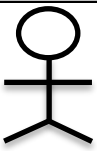
Tabel 2. 10: Simbol Simbol *Diagram Class*

Simbol	Nama	Keterangan
	Generalization	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>)
	Nary Association	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek
	Class	Himpunan dari 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	Operasi yang benar – benar dilakukan oleh suatu objek
	Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
	Association	Objek yang menghubungkan antara objek satu dengan objek lainnya

D. *Sequence diagram*

Sequence diagram mendeskripsikan bagaimana entitas dalam sistem berinteraksi, termasuk pesan yang digunakan saat interaksi. Semua pesan dideskripsikan dalam urutan eksekusi. *Sequence diagram* berhubungan erat dengan *Use Case diagram* [8].

Tabel 2. 11: Simbol – Simbol Sequence Diagram

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

2.2.14 *System Development Life Cycle (SDLC)*

System Development Life Cycle (SDLC) merupakan sebuah metode yang diperlukan untuk mengarahkan dalam pembangunan sebuah perangkat lunak dan untuk menghindari *software crisis* [18]. *System Development Life Cycle (SDLC)* memiliki beberapa tahapan yaitu analisis system, desain system dan implementasi [19].

2.2.15 Analisis Sistem

Analisis sistem merupakan pengembangan sistem tahapan paling awal yang menjadi fondasi untuk memberikan dan menentukan keberhasilan dari sebuah sistem informasi yang akan dihasilkan nantinya. Analisis sistem merupakan sebuah istilah yang dilakukan secara kolektif dan mendeskripsikan fase-fase awal pengembangan [20].

Menurut (Whitten & Bentley, 2017) Analisis sistem merupakan langkah awal dalam pengembangan sistem untuk menentukan kebutuhan, permasalahan yang dapat diatasi dari sebuah sistem yang akan dibangun, dan sistem seperti apa yang akan dibuat nanti [21].

a Fungsi Analisis Sistem

- 1 Membuat keputusan apabila hasil analisis dari sebuah sistem tidak memberikan hasil yang baik atau mempunyai masalah, dan digunakan sebagai dasar untuk memperbaiki sebuah sistem
- 2 Mengetahui ruang lingkup pekerjaannya yang akan ditangani.
- 3 Memahami sistem yang sedang berjalan.
- 4 Mengidentifikasi masalah dan mencari solusi.

2.2.16 Desain Sistem

Desain sistem merupakan penentuan bagaimana cara untuk menyelesaikan apa yang mesti diselesaikan; tahap ini merupakan konfigurasi dari sebuah komponen perangkat lunak dan perangkat keras, sehingga setelah proses instalasi yang telah ditetapkan dari sebuah sistem akan benar - benar memuaskan dari rancang bangun yang telah ditetapkan [22].

a Fungsi Desain Sistem

- 1 Mendesain sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi.

2.2.17 Implementasi

Implementasi merupakan tahap akhir yang dilakukan dalam menjalankan atau melakukan uji coba sebuah sistem yang bertujuan untuk mengevaluasi dan mengetahui fungsi sebuah sistem [18].

a Fungsi Implementasi

- 1 Melakukan kegiatan *spesifikasi* rancangan logikal ke dalam kegiatan yang sebenarnya dari sistem yang akan dibangun atau dikembangkan.
- 2 Mengimplementasikan sistem yang baru.
- 3 Menjamin bahwa sistem yang baru dapat berjalan secara optimal.

2.2.18 Pengujian Sistem

Pengujian sistem merupakan suatu proses yang dilakukan untuk menemukan sebuah kesalahan dari sebuah sistem. Merupakan suatu test yang baik apabila mempunyai kemungkinan untuk menemukan suatu kesalahan yang belum pernah terungkap, dan apabila menemukan atau mengetahui kesalahan yang tidak pernah ditemukan pada awalnya [23]. Dan pengujian ini terdiri dari *Black Box Testing* dan *White Box testing*

A. *Black Box Testing*

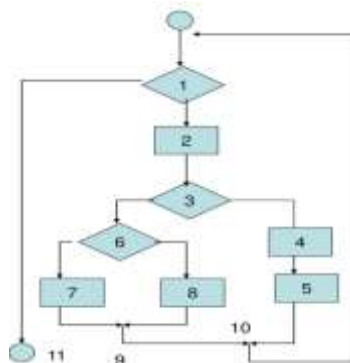
Pengujian *Black box Testing* adalah metode pengujian yang hanya memerlukan batas bawah dan batas atas dari data yang diharapkan dan merupakan metode yang paling mudah digunakan, Estimasi banyaknya data yang diuji dapat dihitung melalui banyaknya *field* data entri yang akan diuji. Metode ini dapat diketahui jika fungsionalitas masih dapat menerima masukan data yang tidak diharapkan maka menyebabkan data yang disimpan kurang valid [24]. Pendekatan pengujian *Black box Testing* adalah metode pengujian di mana data tes berasal dari persyaratan fungsional yang ditentukan tanpa memperhatikan struktur program akhir [25].

B. *White Box Testing*

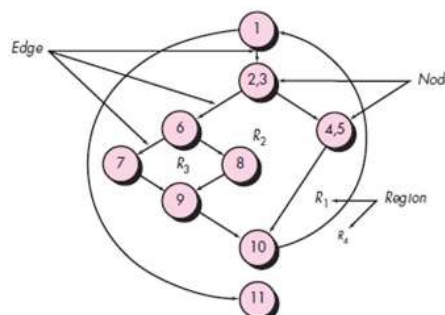
White Box Testing adalah salah satu cara untuk menguji suatu aplikasi dengan cara melihat modul untuk dapat meneliti dan menganalisa kode dari program yang di buat ada yang salah atau tidak. Kalau modul yang telah dihasilkan berupa output yang tidak sesuai dengan yang di harapkan maka akan dikompilasi ulang dan di cek kembali kode - kode tersebut hingga sesuai dengan yang diharapkan [23].

Dengan menggunakan metode uji *White Box*, para pengembang perangkat lunak dapat menghasilkan kasus-kasus uji seperti [26]:

1. Menjamin bahwa seluruh independent paths dalam modul telah dilakukan sedikitnya satu kali.
2. Melakukan seluruh keputusan logikal baik dari sisi benar maupun salah
3. Melakukan seluruh perulangan sesuai batasannya dan dalam batasan operasionalnya, dan
4. Menguji struktur data internal untuk memastikan validitasnya.



Gambar 2. 2: Bagan Alir



Gambar 2. 3: Aliran Grafik

Complexity Cyclomatic merupakan matriks perangkat lunak yang memberikan pengukuran kuantitatif terhadap kompleksitas logika sebuah program. Bila matriks ini digunakan dalam konteks metode pengujian basis path, maka nilai yang terhitung untuk kompleksitas siklomat menentukan jumlah jalur *independen*. Jalur yang melalui program yang mengintroduksi sedikitnya suatu rangkaian prose statemen baru atau kondisi baru merupakan jalur *independen*. Bila dinyatakan dengan terminologi grafik alir, jalur *independen* harus bergerak sepanjang paling tidak satu *edge* yang tidak dilewatkan sebelum jalur tersebut ditentukan.

Contoh serangkaian jalur *independen* untuk grafik alir yang ditunjukkan pada gambar diatas adalah:

Path 1 = 1-11

Path 2 = 1-2-3-4-5-10-1-11

Path 3 = 1-2-3-6-8-9-10-1-11

Path 4 = 1-2-3-6-7-9-10-1-11

Perhatikan bahwa masing-masing jalur baru memperkenalkan sebuah *edge* baru

Path 1-2-3-4-5-10-1-2-3-6-8-9-10-1-11

Tidak dianggap jalur *independen* karena merupakan gabungan dari jalur-jalur yang sudah ditentukan dan tidak melewati beberapa *edge* baru.

Jalur 1,2,3 dan 4 yang ditentukan di atas terdiri dari sebuah basis set untuk grafik alir pada gambar di atas. *Complexity* dapat dihitung dengan tiga cara berikut:

1. Jumlah *region* grafik alir sesuai dengan *cyclomatic complexity*
2. *Cyclomatic complexity* $V(G)$ untuk grafik alir dihitung dengan rumus :

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

Dimana: E = Jumlah *edge* pada grafik alir

N = Jumlah *node* pada grafik alir

3. *Cyclomatic complexity* $V(G)$ juga dapat dihitung dengan rumus :

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

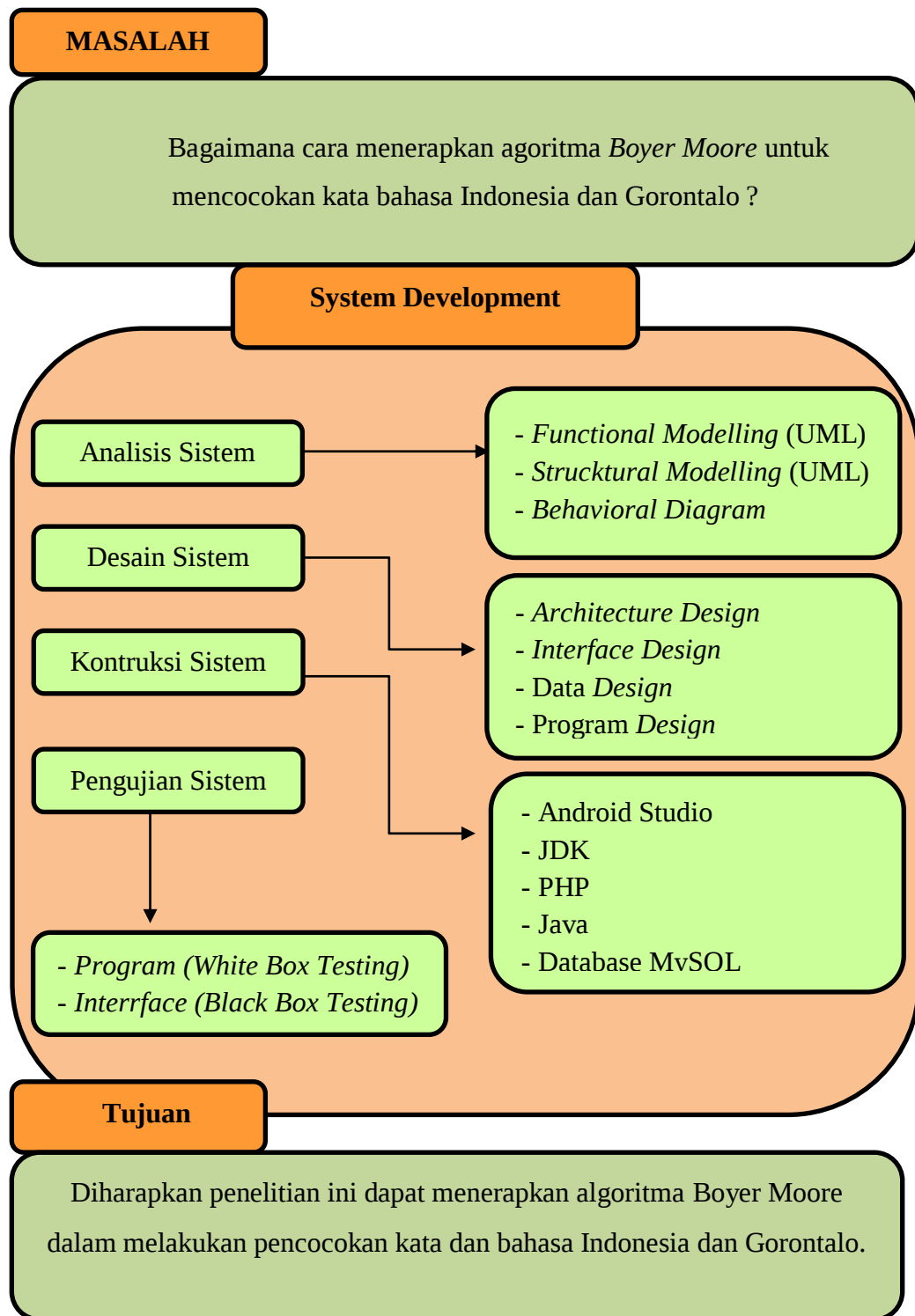
Dimana P = jumlah *predicate node* pada grafik alir

Dari gambar diatas dapat dihitung *cyclomatic complexity* :

1. Grafik alir mempunyai 4 region
2. $V(G) = 11 \text{ edge} - 9 \text{ node} + 2 = 4$
3. $V(G) = 3 \text{ predicate node} + 1 = 4$

Jadi *cyclomatic* untuk grafik alir pada gambar di atas adalah 4 (empat).

2.3 Kerangka Pikir



Gambar 2. 4: Kerangka Fikir

BAB III

METODE PENELITIAN

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

Jika dipandang dari tingkat penerapannya, maka penelitian ini merupakan jenis penelitian terapan.

Penelitian ini menggunakan metode penelitian studi literatur, dimana penelitian melakukan pencarian pengaruh terhadap pengurangan bahasa daerah Gorontalo yang ada di Provinsi Gorontalo ini, dan memberikan pemahaman kembali tentang bahasa daerah Gorontalo. Dengan demikian jenis penelitian ini adalah penelitian *deskriptif*. **Subjek penelitian ini adalah Implementasi Algoritma Boyer Moore Untuk Kamus Bahasa Indonesia - Gorontalo.** Penelitian ini dilakukan pada bulan Februari 2020 yang berlokasi di kantor Bahasa Gorontalo.

3.2 Pengumpulan Data

Dalam memperoleh pengumpulan 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

Untuk memperoleh data primer yang merupakan data langsung dari objek penelitian yaitu bertempat di Kantor Bahasa Gorontalo. Maka dilakukan dengan teknik dokumentasi, wawancara dan observasi.

- a. Observasi merupakan tehnik pengumpulan data dengan melakukan pengamatan secara langsung terhadap objek penelitian untuk meninjau segala permasalahan yang ada pada lokasi yang diteliti demi membantu peneliti dalam menyelesaikan penelitian yang sedang berlangsung, dalam hal ini peneliti mengambil hasil observasi berdasarkan jurnal-jurnal atau referensi terkait.

- b. Teknik wawancara merupakan teknik pengumpulan data melalui proses interaksi antara peneliti dengan pihak responden, biasanya dilakukan dengan proses tanya jawab.
 - c. Dalam studi dokumentasi peneliti masih menggunakan jurnal-jurnal penelitian terkait, literature, atau referensi untuk mendapatkan data
2. Penelitian data sekunder (Kepustakaan)

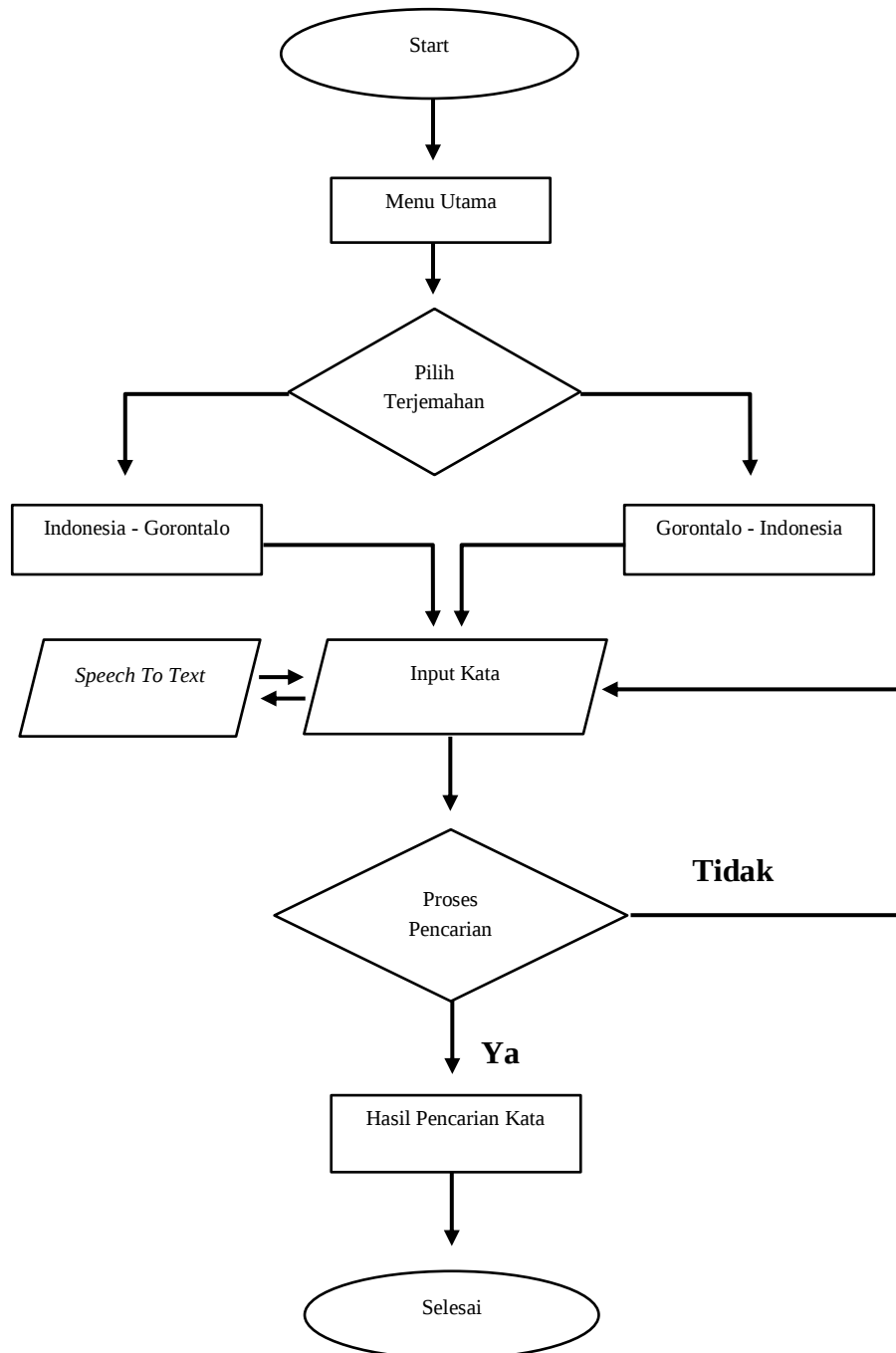
Metode kepustakaan ini digunakan untuk mendapatkan data sekunder dengan tujuan melengkapi data primer. Data sekunder didapatkan dari pengkajian kepustakaan yang berisi dasar - dasar teori. Metode kepustakaan ini digunakan oleh analisis sistem dengan cara mengambil contoh dokumen - dokumen yang berhubungan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

Tabel 3. 1: Atribut Data

No	Nama	Type	Value	Keterangan
1	Input text	Text	Makan	Input
2	Output Text	Text	Hemonga	Output
3	Bahasa Indonesia	Char	Bahasa Indonesia	Input
4	Bahasa Gorontalo	Char	Bahasa Gorontalo	Input

3.3 Pengembangan Sistem

Sistem yang diusulkan dapat digambarkan menggunakan *flowchart* dokumen yang pada gambar di bawah ini



Gambar 3. 1: Flowchart

3. 4. 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. 4. 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 *stand alone*
- Spesifikasi *Hardware* dan *Software* yang digunakan
 1. *Hardware*
 - Komputer
 1. Prosessor Intel(R) Core(TM) i5-4465 CPU @ 3.20GHz
 2. VGA NVIDIA GeForce GT 730
 3. Ram 4 GB
 4. Hard Disk 500GB
 - *Smartphone*
 1. CPU Octa Core 1.4GHz
 2. Ram 3 GB
 3. Rom 32 GB
 4. Layar 5.65 Inch

2. *Software*

1. Sistem Operasi Windows 7 (64 Bit)
2. Sistem Operasi Android 8.0 (Oreo)
3. Android Studio
4. Xampp
5. JDK
6. PHP
7. Java

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. 4. 3 Kontruksi Sistem

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

3. 4. 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 $\text{independent path} = V(G) = (CC) = \text{region}$, 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.

Tabel 3. 2: Rencana Pengujian *Black Box Testing*

No	Kelas Uji	Butir Uji
1	Menu Utama	Pilihan terjemahan Tentang Keluar
2	Pilihan terjemahan	Menampilkan pilihan bahasa
3	Input kata	Menampilkan Hasil Terjemahan
	<i>Speech To Text</i>	Menampilkan perekaman dan <i>convert</i> ke text
4	Tentang	Menampilkan Tentang Aplikasi

BAB IV

HASIL PENELITIAN

4.1 Hasil Pengumpulan Data

Berdasarkan hasil pengumpulan data, diperoleh data primer sebagai berikut:

Tabel 4. 1: Hasil Pengumpulan Data

No	Bahasa Indonesia	Bahasa Gorontalo
1	sepakat	a'aakorowa
2	saling berjauhan	lalaminga
3	disapu	a'aato
4	disapu	a'atalo
5	sapu	a'ato
6	sapu ijuk	aa'ata dunula
7	sapu lidi	aa'ata hiyo
8	sapu	aa'ato
9	acar	aacari
10	adat	aadati
...	...	...
559	tajam	lalito
560	rakus	lalo

4.1.1 Penerapan Metode

1. Buat tabel pergeseran string yang dicari (S) dengan pendekatan *Match Heuristic* (MH) dan *Occurence Heuristic* (OH), untuk menentukan jumlah pergeseran yang akan dilakukan jika mendapat karakter tidak cocok pada proses pencocokan dengan string (T).
2. Jika dalam proses perbandingan terjadi ketidakcocokan antara pasangan karakter pada S dan karakter pada T, pergeseran dilakukan dengan memilih salah satu nilai pergeseran dari dua tabel analisa string, yang memiliki nilai pergeseran paling besar.
3. Dua kemungkinan penyelesaian dalam melakukan pergeseran S, jika sebelumnya belum ada karakter yang cocok adalah dengan melihat nilai pergeseran hanya pada tabel *occurence heuristic*. Jika karakter yang tidak cocok tidak ada pada S maka pergeseran adalah sebanyak jumlah karakter pada S. dan jika karakter yang tidak cocok ada pada S, maka banyaknya pergeseran bergantung dari nilai pada tabel.
4. Jika karakter pada teks yang sedang dibandingkan cocok dengan karakter pada S, maka posisi karakter pada S dan T diturunkan sebanyak 1 posisi, kemudian lanjutkan dengan pencocokan pada posisi tersebut dan seterusnya. Jika kemudian terjadi ketidakcocokan karakter S dan T, maka pilih nilai pergeseran terbesar dari dua tabel analisa pattern yaitu nilai dari tabel *match heuristic* dan nilai tabel *occurence heuristic* dikurangi dengan jumlah karakter yang telah cocok.
5. Jika semua karakter telah cocok, artinya S telah ditemukan di dalam T, selanjutnya geser *pattern* sebesar 1 karakter.
6. Lanjutkan sampai akhir string T.

Contoh Kasus :

Pattern : GORONTALO

Tekst : ABCDGORONTALO

1. Lakukan pencacahan mulai dari posisi terakhir string sampai ke posisi awal, dimulai dengan nilai 0, catat karakter yang sudah ditemukan (dalam contoh ini karakter “O”)
2. Mundur ke posisi sebelumnya, nilai pencacah ditambah 1, jika karakter pada posisi ini belum pernah ditemukan, maka nilai pergeserannya adalah sama dengan nilai pencacah. (dalam contoh ini, karakter “L” belum pernah ditemukan sehingga nilai pergeserannya adalah sebesar nilai pencacah yaitu 2)
3. Mundur ke posisi sebelumnya, karakter “A” nilai pergeserannya 2
4. Mundur ke posisi sebelumnya, karakter “T” nilai pergeserannya 3
5. Mundur ke posisi sebelumnya, karakter “N” nilai pergeserannya 4
6. Mundur ke posisi sebelumnya, karakter “O” sudah pernah ditemukan sebelumnya sehingga nilai pergeserannya sama dengan nilai pergeseran karakter “O” yang sudah ditemukan paling awal yaitu 0
7. Mundur ke posisi sebelumnya, karakter “R” nilai pergeserannya 6
8. Mundur ke posisi sebelumnya, karakter “O” sudah pernah ditemukan sebelumnya sehingga nilai pergeserannya sama dengan nilai pergeseran karakter “O” yang sudah ditemukan paling awal yaitu 0
9. Mundur ke posisi sebelumnya, karakter “G” nilai pergeserannya 8
10. Begitu seterusnya sampai posisi awal string.

Tabel 4. 1: Occurrence Heuristic

Indeks	0	1	2	3	4	5	6	7	8
Pattern	G	O	R	O	N	T	A	L	O
Pergeseran OH	8	0	6	0	4	3	2	1	0

Tabel 4. 2: Daftar Pencacah *Prefix* dan *Suffix* dari *Pattern*

Suffix (Kanan-Kiri)	Prefix	Suffix (Kiri-Kanan)
Null	G	ORONTALO
G	O	RONTALO
GO	R	ONTALO
GOR	O	NTALO
GORO	N	TALO
GORON	T	ALO
GORONT	A	LO
GORONTA	L	O
GORONTAL	O	NULL

Tabel 4. 3: Pemberian Nilai Pencacahan

Compure	GORONTAL	GORONTA	GORONT	GORON	GORO	GOR	GO	G	NULL
Pergeseran	1	2	3	4	5	6	7	8	9

Tabel 4. 4: Pencarian Nilai *Match Heuristic* (OH)

Pattern	G	O	R	O	N	T	A	L	O
---------	---	---	---	---	---	---	---	---	---

Pattern	G	O	R	O	N	T	A	L	O
Nilai MH	-	-	-	-	-	-	-	-	1

MH Selalu Bernilai 1

Pattern	G	O	R	O	N	T	A	L	O
---------	---	---	---	---	---	---	---	---	---

Pattern	G	O	R	O	N	T	A	L	O
Nilai MH	-	-	-	-	-	-	-	9	1

Prefix	L
Suffix	O

Suffix Compure Null
Pergeseran Suffix = 9

Pattern	G	O	R	O	N	T	A	L	O
---------	---	---	---	---	---	---	---	---	---

Pattern	G	O	R	O	N	T	A	L	O
Nilai MH	-	-	-	-	-	-	9	9	1

Prefix	A
Suffix	LO

Suffix Compure Null
Pergeseran Suffix = 9

Pattern	G	O	R	O	N	T	A	L	O
---------	---	---	---	---	---	---	---	---	---

Pattern	G	O	R	O	N	T	A	L	O
Nilai MH	-	-	-	-	-	9	9	9	1

Prefix	T
Suffix	ALO

Suffix Compure Null
Pergeseran Suffix = 9

Pattern	G	O	R	O	N	T	A	L	O
---------	---	---	---	---	---	---	---	---	---

Pattern	G	O	R	O	N	T	A	L	O
Nilai MH	-	-	-	-	9	9	9	9	1

Prefix	N
Suffix	TALO

Suffix Compure Null
Pergeseran Suffix = 9

Pattern	G	O	R	O	N	T	A	L	O
---------	---	---	---	---	---	---	---	---	---

Pattern	G	O	R	O	N	T	A	L	O
Nilai MH	-	-	-	9	9	9	9	9	1

Prefix	O
Suffix	NOTALO

Suffix Compure Null
Pergeseran Suffix = 9

Pattern	G	O	R	O	N	T	A	L	O
---------	---	---	---	---	---	---	---	---	---

Pattern	G	O	R	O	N	T	A	L	O
Nilai MH	-	-	9	9	9	9	9	9	1

Prefix	R
Suffix	ONTALO

Suffix Compure Null
Pergeseran Suffix = 9

Pattern	G	O	R	O	N	T	A	L	O
---------	---	---	---	---	---	---	---	---	---

Pattern	G	O	R	O	N	T	A	L	O
Nilai MH	-	9	9	9	9	9	9	9	1

Prefix	O
Suffix	RONTALO

Suffix Compure Null
Pergeseran Suffix = 9

Pattern	G	O	R	O	N	T	A	L	O
---------	---	---	---	---	---	---	---	---	---

Pattern	G	O	R	O	N	T	A	L	O
Nilai MH	9	9	9	9	9	9	9	9	1

Prefix	G
Suffix	ORONTALO

Suffix Compure Null
Pergeseran Suffix = 9

Langkah-langkah melakukan prosedur

1. Pencocokan dimulai dari index text terkecil atau dari karakter paling kiri
2. Pencocokkan per karakter dimulai dari karakter paling kanan *patern*
3. Setiap kali ditemukan ketidakcocokan karakter, maka ambil nilai OH dengan karakter yang bersesuaian dengan teks, kemudian nilai OH dikurangi dengan nilai kecocokkan yang telah terjadi. Ambil nilai MH dengan indeks karakter yang ditemukan ketidakcocokkan, kemudian bandingkan dengan nilai OH yang telah dioperasikan. Dengan nilai terbesar, geser *patern* ke arah kiri teks dan ulangi pencocokkan.
4. Jika nilai kecocokkan sama dengan panjang *pattern* maka *patern* telah ditemukan pada *pattern* sepanjang jumlah karakter *pattern* ke arah kiri teks untuk melanjutkan pencocokkan selanjutnya.

Penyelesain :

Tabel 4. 5: Pemberian Nilai (OH) dan (MH)

Occurence Heuristic (OH)									
Indeks	0	1	2	3	4	5	6	7	8
Posisi	1	2	3	4	5	6	7	8	9
Karakter	G	O	R	O	N	T	A	L	O
Nilai OH	8	0	6	0	4	3	2	1	0

Match Heuristic (MH)									
Indeks	0	1	2	3	4	5	6	7	8
Posisi	1	2	3	4	5	6	7	8	9
Karakter	G	O	R	O	N	T	A	L	O
Nilai MH	9	9	9	9	9	9	9	9	0

Tabel 4. 6: Pencarian Pencocokan Karakter

Indeks	0	1	2	3	4	5	6	7	8	9	10	11	12
Posisi	1	2	3	4	5	6	7	8	9	10	11	12	13
Teks	A	B	C	D	G	O	R	O	N	T	A	L	O
Pattern	G	O	R	O	N	T	A	L	O				

Nilai OH [N] = 4 banding Nilai MH [O] = 0

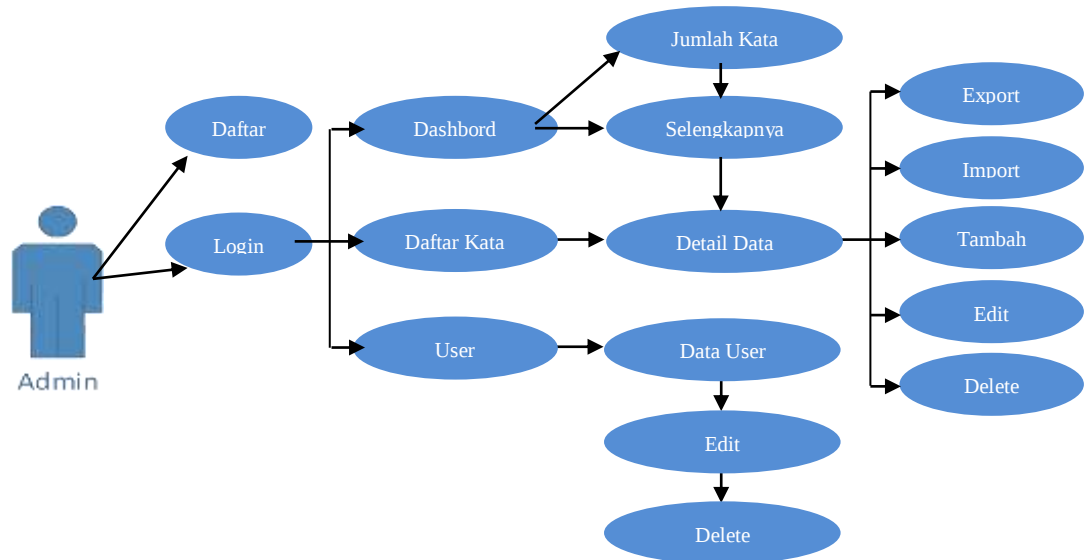
Shif = 4 Pergeseran

Tabel 4. 7: Hasil Pencarian Pencocokan Karakter

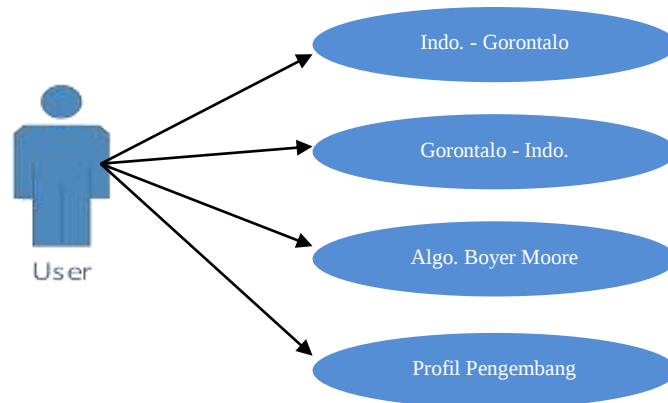
Indeks	0	1	2	3	4	5	6	7	8	9	10	11	12
Posisi	1	2	3	4	5	6	7	8	9	10	11	12	13
Teks	A	B	C	D	G	O	R	O	N	T	A	L	O
Pattern					G	O	R	O	N	T	A	L	O

Hasil Pencarian telah ditemukan pada indeks ke 4.

4.2 Hasil Pengembangan Sistem

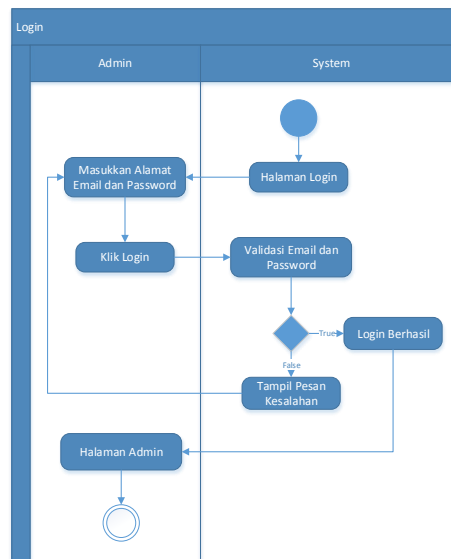


Gambar 4. 1: *Use Case Diagram* Halaman Admin



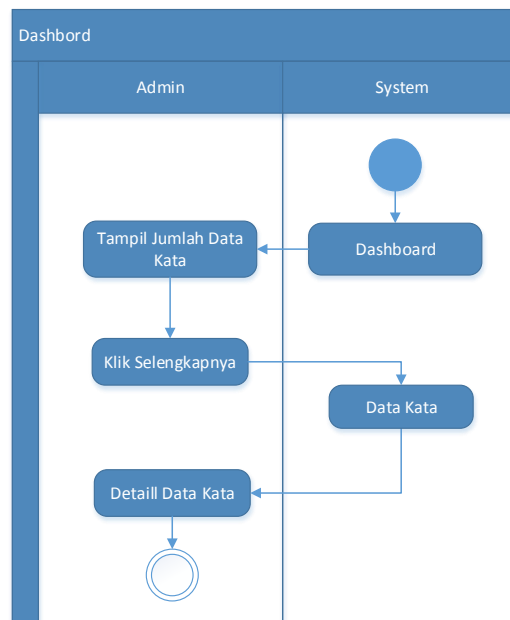
Gambar 4. 2: *Use Case Diagram* Kamus Bahasa Indonesia Gorontalo

4.3 Activity Diagram Login



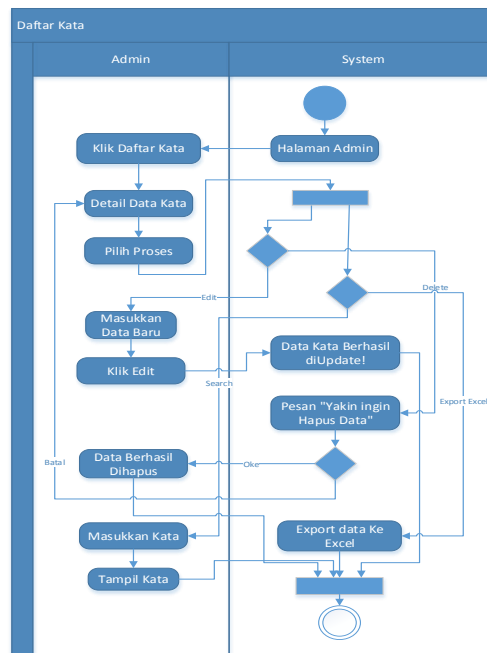
Gambar 4. 3: Activity Diagram Proses Login

4.4 Activity Diagram Dashboard



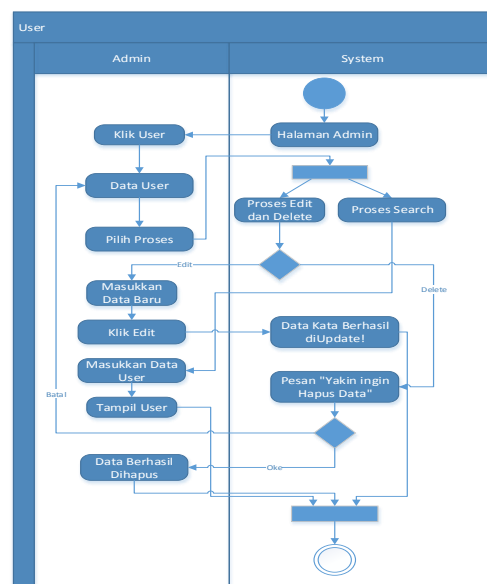
Gambar 4. 4: Activity Diagram Dashboard

4.5 Activity Diagram Daftar Kata



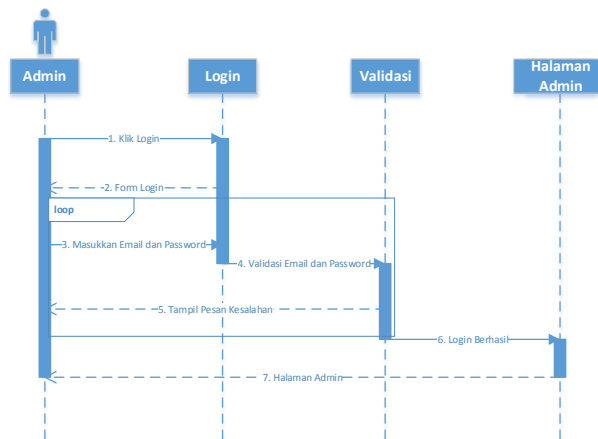
Gambar 4. 5: Activity Diagram Daftar Kata

4.6 Activity Diagram Data User



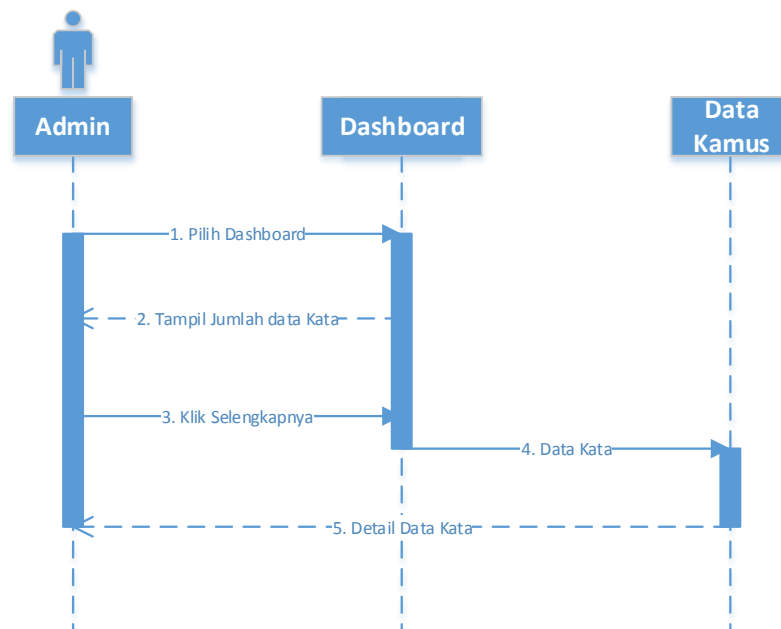
Gambar 4. 6: Activity Diagram User

4.7 Sequence Diagram Login Admin



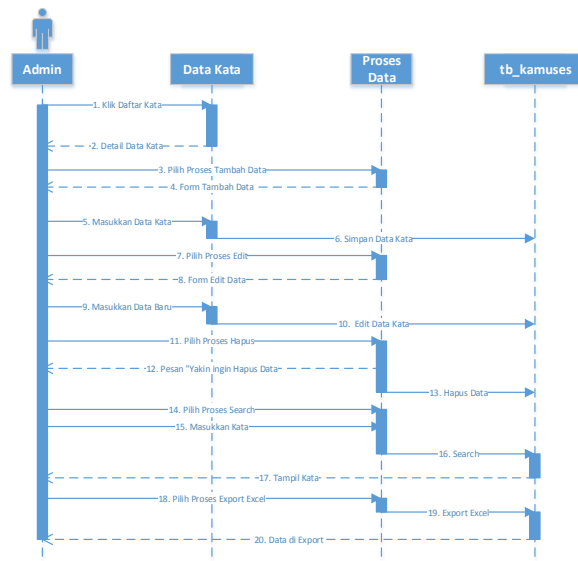
Gambar 4. 7: Sequence Diagram Login Admin

4.8 Sequence Diagram Dashboard



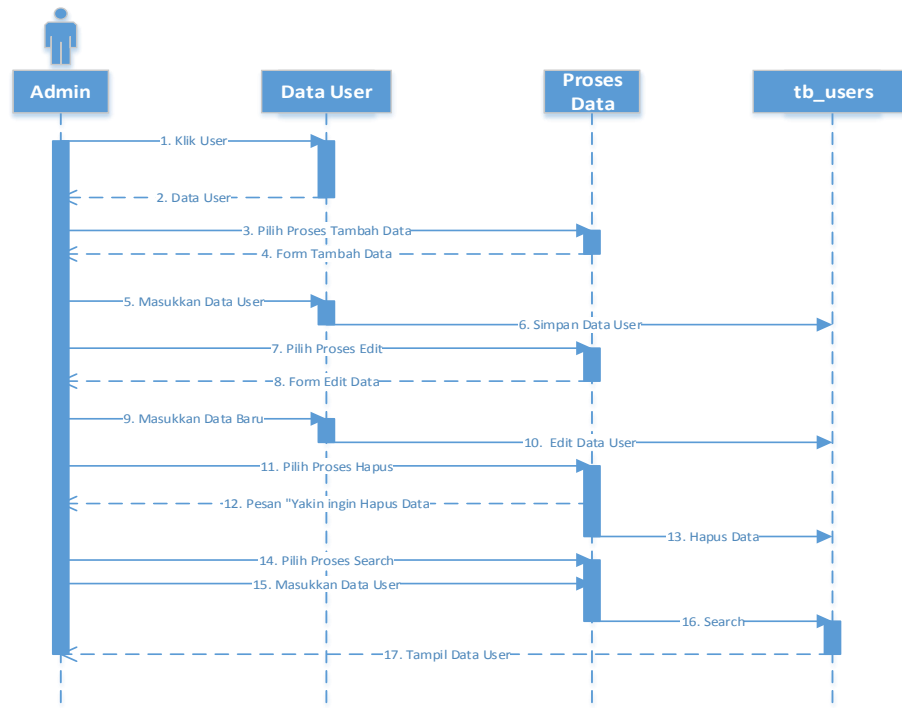
Gambar 4. 8: Sequence Diagram Dashboard

4.9 Sequence Diagram Daftar Kata



Gambar 4. 9: Sequence Diagram Daftar Kata

4.10 Sequence Diagram User



Gambar 4. 10: 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

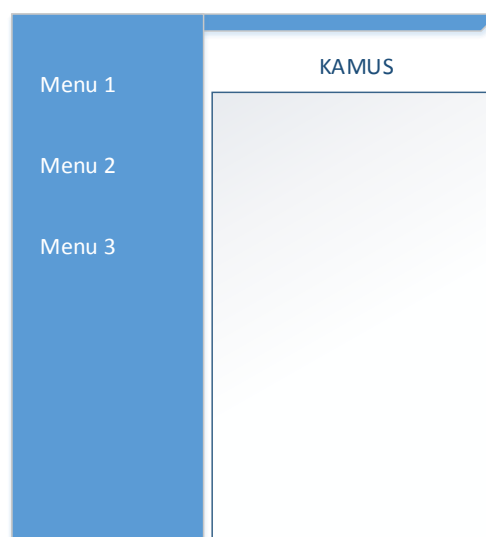
4.12 Interface Design

4.12.1 Mekanisme User

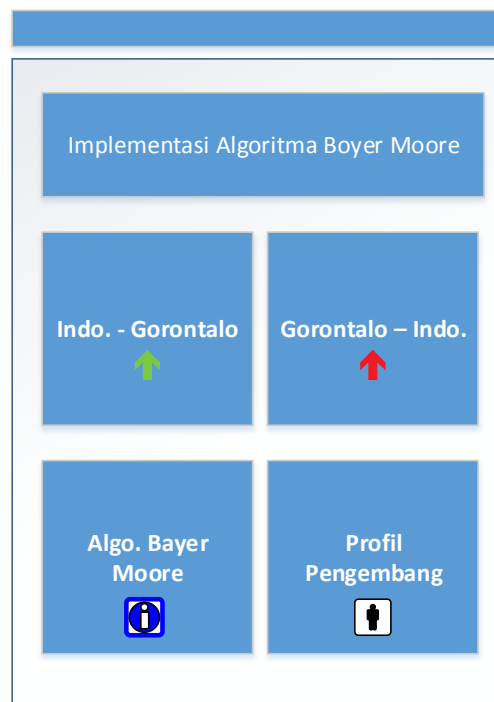
Tabel 4. 8: Mekanisme User

User	Kategori	Akses Input	Akses Output
User	User	-	Kamus
Admin	Administrator	All	All

4.12.2 Mekanisme navigasi *Home*



Gambar 4. 11: Mekanisme Navigasi *Home* Admin



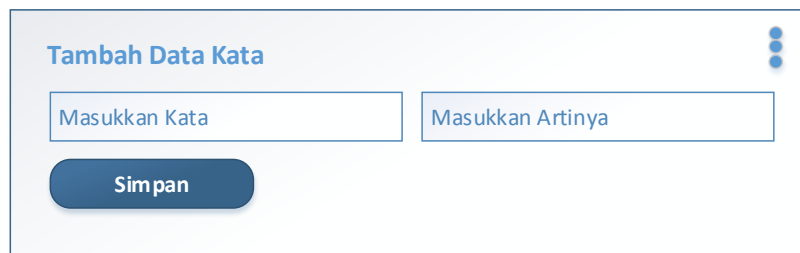
Gambar 4. 12: Mekanisme Navigasi *Home* User

4.12.3 Mekanisme *Login*

The diagram shows the layout of the login page. It has a light gray background with the title "Halaman Login" in blue. Below the title are two rounded rectangular input fields: "Enter Email Adress..." and "Password". Under the password field is a checkbox labeled "Remember Me". At the bottom is a wide, rounded blue button labeled "Login".

Gambar 4. 13: Mekanisme *Login*

4.12.4 Mekanisme *Input* data Kata



Tambah Data Kata

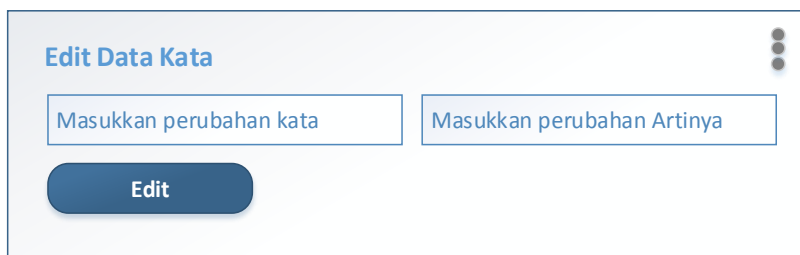
Masukkan Kata

Masukkan Artinya

Simpan

Gambar 4. 14: Mekanisme *Input* data Kata

4.12.5 Mekanisme Edit Data Kata



Edit Data Kata

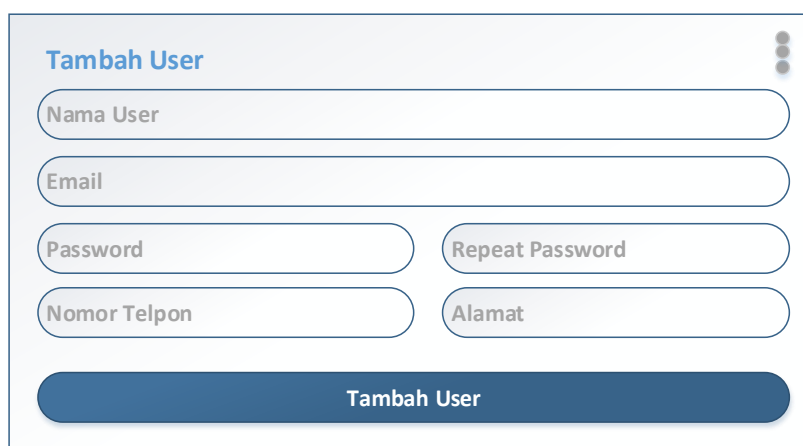
Masukkan perubahan kata

Masukkan perubahan Artinya

Edit

Gambar 4. 15: Mekanisme Edit Data Kata

4.12.6 Mekanisme Tambah Data User



Tambah User

Nama User

Email

Password

Repeat Password

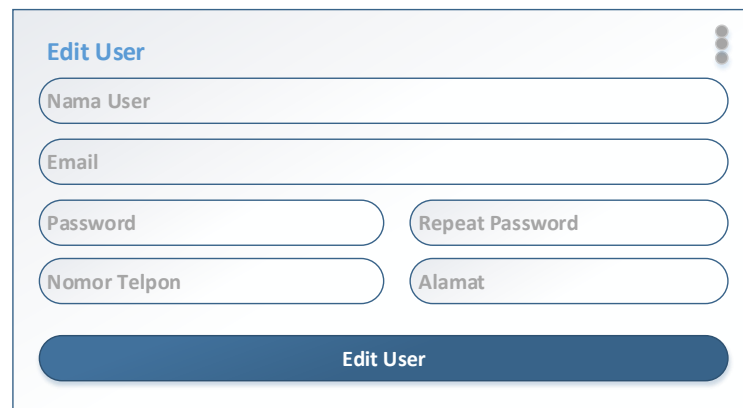
Nomor Telpn

Alamat

Tambah User

Gambar 4. 16: Mekanisme Tambah Data User

4.12.7 Mekanisme Edit Data User



Edit User

Nama User

Email

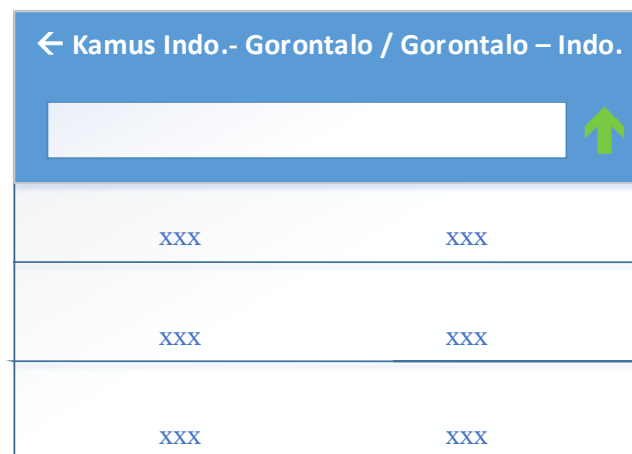
Password Repeat Password

Nomor Telpn Alamat

Edit User

Gambar 4. 17: Mekanisme Edit Data User

4.12.8 Mekanisme *Output*



← Kamus Indo.- Gorontalo / Gorontalo – Indo.

XXX XXX

XXX XXX

XXX XXX

Gambar 4. 18: Mekanisme *Output* Kamus

4.13 Data Desain

4.13.1 Struktur Data

Tabel 4. 9: Tabel Kamus

Nama File : kamuses Primary key : id Media : hardisk fungsi : menyimpan data kamus struktur data :				
No	Field Name	Type	Width	keterangan
1.	Id	Bigint	20	Id kamus
2.	Kata1	Varchar	255	Data kata kamus
3.	Kata2	Varchar	255	Kata kamus
4.	Created_at	Timestamp	-	Waktu pemasukan data
5.	Updated_at	Timestamp	-	Waktu mengubah data

Tabel 4. 10: Tabel User

Nama File : users Primary key : id Media : hardisk fungsi : menyimpan data user struktur data :				
No	Field Name	Type	Width	keterangan
1.	Id	Bigint	20	Id user
2.	Name	Varchar	255	Nama user
3.	Email	Varchar	255	Alamat email user
4.	Email_verified_at	Timestamp	-	Waktu verifikasi email

5.	Password	Varchar	255	Password user
6.	Phone_number	Varchar	255	Nomor telpon
7.	Address	Varchar	255	Alamat user
8.	Remember_token	Varchar	100	Remember user
9.	Created_at	Timestamp	-	Waktu pemasukan data
10.	Updated_at	Timestamp	-	Waktu pengubahan data

4.14 Hasil Pengujian Sistem

4.14.1 Pengujian White Box

```

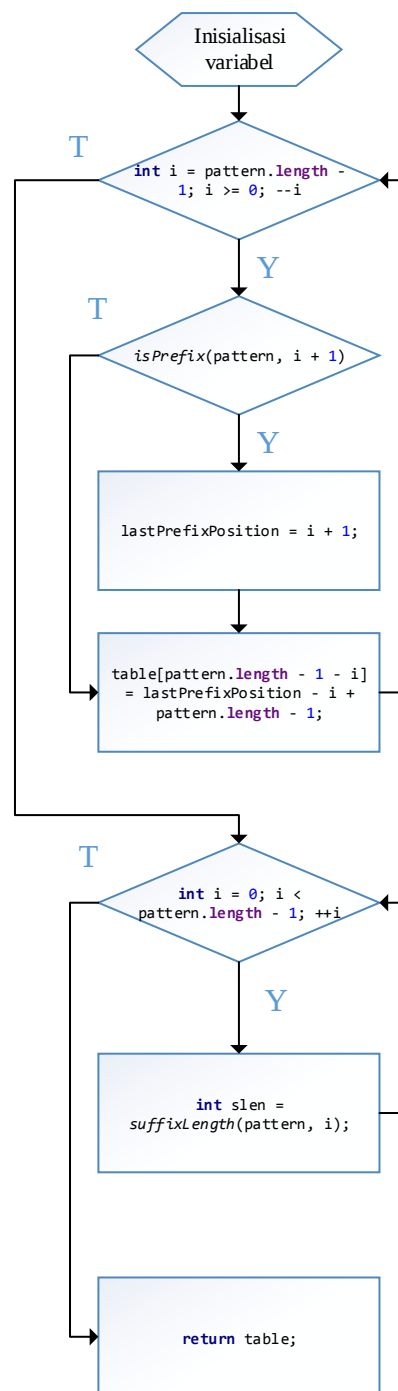
int[] table = new int[pattern.length]; -----1
int lastPrefixPosition = pattern.length; -----1
for (int i = pattern.length - 1; i >= 0; --i){ -----2
    if (isPrefix(pattern, i + 1)) -----3
        lastPrefixPosition = i + 1; -----4
    table[pattern.length - 1 - i] = lastPrefixPosition - i +
pattern.length - 1; -----5
}
for (int i = 0; i < pattern.length - 1; ++i){ -----6
    int slen = suffixLength(pattern, i); -----7
    table[slen] = pattern.length - 1 - i + slen; -----7
}
return table;-----8

```

Keterangan :

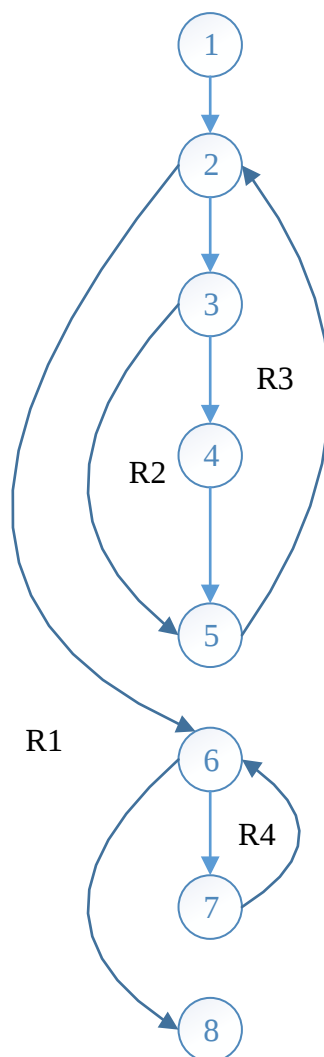
1. Merupakan panjang karakter pattern pada suatu tabel dan penentuan posisi terakhir pattern didalam suatu tabel.
2. Panjang pattern dikurangi 1 dimana dimana “i = panjang pattern”.
3. Isprefix merupakan merupakan merupakan awal dari pattern
4. LastPrefix merupakan akhir pattern
5. Panjang pattern yang ada didalam tabel
6. SuffixLength merupakan panjang karakter dari pattern yang didapat dari lastPrefixPosition

4.14.2 Flowchart



Gambar 4. 19: Flowchart Algoritma Boyer Moore

4.14.3 Pengujian *White Box*



Gambar 4. 20: *Flowgraph* Algoritma Boyer Moore

4.14.4 Perhitungan CC pada pengujian *White Box*

Dari *Flowgraph* tersebut, didapatkan :

Diketahui	$Region(R)$	= 4
	$Node(N)$	= 8
	$Edge(E)$	= 10
	$Predicate Node(P)$	= 3
	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 Perhitungan CC pada pengujian *White Box*

Tabel 4. 11: Basis Path

NO	PATH	KET
1.	1-2-6-8	OK
2.	1-2-3-5-2-...	OK
3.	1-2-3-4-5-2-...	OK
4.	1-2-6-7-2-...	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. 12: Tabel Pengujian *Black Box*

Input/Event	Fungsi	Hasil Yang Diharapkan	Hasil Uji
Masukkan email address salah, klik Login	Validasi email	Tampil pesan kesalahan	Sesuai
Masukkan Password Salah, klik Login	Validasi password	Tampil pesan kesalahan	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 Selengkapnya pada dashboard	Menampilkan data kata	Tampil Data Kata	Sesuai
klik menu Daftar Kata, Detail Data Kata	Menampilkan data kata	Tampil data kata	Sesuai
Klik Tambahkan data pada menu Daftar Kata	Menambahkan data kamus	Tampil form Tambah Data Kata	Sesuai
Masukkan data kata, klik simpan	Menambahkan data kamus	Tampil pesan “Data kata berhasil disimpan!”	Sesuai
Klik Edit pada menu Daftar Kata	Mengubah data kata	Tampil form Edit Data Kata	Sesuai
Masukkan data kata yang baru, klik Edit	Mengubah data kata	Tampil pesan “Data kata berhasil diupdate!”	Sesuai
Klik Hapus pada menu Daftar Kata	Menghapus data kata	Tampil pesan “Yakin akan menghapus data?”	Sesuai
Klik Export Excel	Mengambil	Data kamus terdownload	Sesuai

Input/Event	Fungsi	Hasil Yang Diharapkan	Hasil Uji
pada menu Daftar Kata	data kamus		
Klik menu User, Data user	Menampilkan data user	Tampil Data User	Sesuai
Klik Profil	Menampilkan data profil	Tampil Profile Developer	Sesuai
Klik edit pada menu User	Mengubah data user	Tampil form Edit User	Sesuai
Masukkan perubahan data user, klik Edit User	Mengubah data user	Tampil pesan “Data berhasil di update”	Sesuai
Klik Delete pada menu user	Menghapus data user	Tampil pesan “Yakin ingin hapus data?”	Sesuai
Klik Profil Pengembang	Menampilkan data pembuat aplikasi	Tampil data Tentang Pengembang	Sesuai
Klik Algo. Bayer Moore	Menampilkan data seputar algoritma bayer moore	Tampil data tentang Algoritma Bayer Moore	Sesuai
Klik Indo. – Gorontalo	Membuka kamus Indonesia-Gorontalo	Tampil pencarian kata	Sesuai
Klik Gorontalo – indo.	Membuka kamus Gorontalo – Indonesia	Tampil pencarian kata	Sesuai

4.14.7 Pengujian Akurasi Algoritma Boyer Moore

Pengujian akurasi proses pencarian kata bahasa Indonesia Gorontalo menggunakan algoritma boyer moore. Pengujian ini dilakukan untuk mengetahui tingkat akurasi algoritma boyer moore dengan cara mencocokkan hasil pencarian algoritma boyer moore dengan data yang ada pada *database*

A. *Confusion Matriks*

Pada penelitian ini peneliti menggunakan *confusion matriks* sebagai metode untuk menghitung nilai akurasi *Algoritma Boyer Moore* pada pencarian kata.

		Kelas Sebenarnya	
		True	False
Kelas Prediksi	Positive	True Positive	False Positive
	Negative	True Negative	False Negative

Kriteria

- True Positive Rate disebut juga Sensitivity (TPR) = $TP/(TP+FN)$
- True Negative Rate disebut juga Specivity (TNR) = $TN/(TN+FP)$
- Accuracy = $(TP+TN)/(TP+FP+TN+FN) \times 100\%$

Dimana :

TP = True Positive yaitu klasifikasi yang dari kelas yang positif

FN = False Negative yaitu kesalahan type I

FP = False Positive yaitu kesalahan type II

Tabel 4. 13: Pengujian Akurasi

No	Kata	Jumlah Data	Hasil Pencarian Yang benar (TP)	Data Yang seharusnya Benar	Hasil Pencarian Salah	Data Yang Tidak Terseleksi Dengan Benar (TN)	Keterangan	Akurasi
1	se	560	30	30	0	530	Tepat	100%
2	Sepakat	560	2	2	0	558	Tepat	100%
3	Sa	560	34	34	0	526	Tepat	100%
4	Sarung	560	1	1	0	559	Tepat	100%
5	ma	560	34	34	0	526	Tepat	100%
6	yang	560	13	13	0	547	Tepat	100%
7	kemenyang	560	3	3	0	557	Tepat	100%
8	Sapu	560	6	6	0	554	Tepat	100%
9	saling	560	3	3	0	557	Tepat	100%
10	Apa	560	2	2	0	558	Tepat	100%

$$Akurasi = \frac{True\ Positive\ (TP) + True\ Negative\ (TN)}{Total\ Population} \times 100\%$$

$$Akurasi = \frac{128 + 5472}{5728} \times 100\%$$

$$Akurasi = \frac{5600}{5728} \times 100\% = 100\%$$

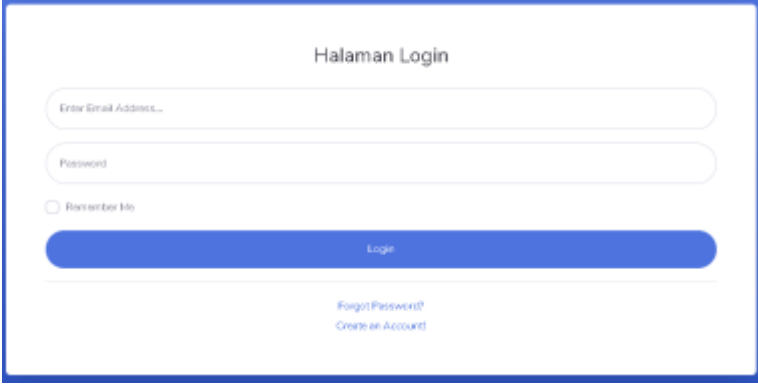
Berdasarkan hasil uji coba akurasi sebesar 100% pada 10 kali uji coba pencarian, karena jumlah data yang didapat sama dengan data yang ada didalam *database*.

BAB V

HASIL DAN PEMBAHASAN

5.1 Pembahasan Sistem

5.1.1 Tampilan Halaman *Login*



The screenshot shows a login form titled "Halaman Login". It contains two input fields: "Enter Email Address..." and "Password". Below the password field is a checkbox labeled "Remember Me". A large blue button labeled "Login" is positioned below the inputs. At the bottom, there are two links: "Forgot Password?" and "Create an Account".

Gambar 5. 1: Tampilan Halaman *Login*

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

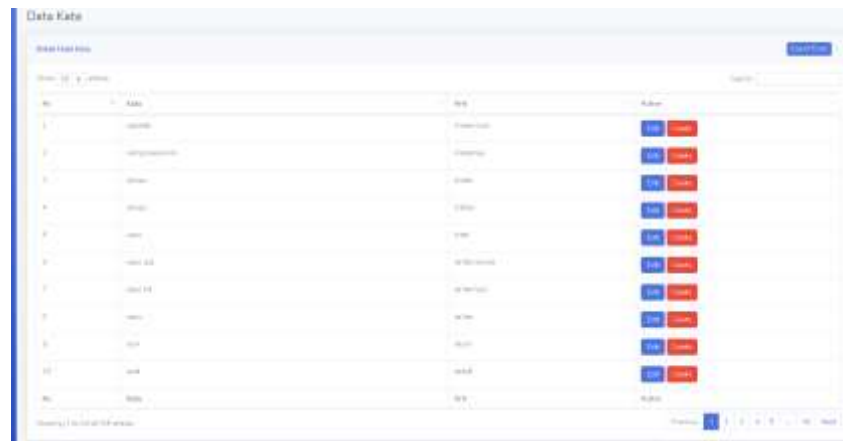
5.1.2 Tampilan Halaman Dashboard Admin



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 button Selengkapnya.

5.1.3 Tampilan Halaman Data Kata

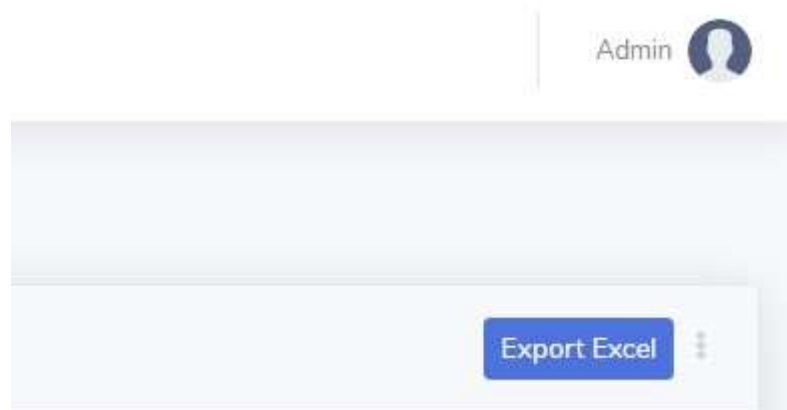


ID	Kata	Arti	Aksi
1	gampang	gampang	Edit Hapus Export Excel
2	gampang	gampang	Edit Hapus Export Excel
3	gampang	gampang	Edit Hapus Export Excel
4	gampang	gampang	Edit Hapus Export Excel
5	gampang	gampang	Edit Hapus Export Excel
6	gampang	gampang	Edit Hapus Export Excel
7	gampang	gampang	Edit Hapus Export Excel
8	gampang	gampang	Edit Hapus Export Excel
9	gampang	gampang	Edit Hapus Export Excel
10	gampang	gampang	Edit Hapus Export Excel

Gambar 5. 3: Tampilan Halaman Data Kata

Halaman ini digunakan untuk menampilkan detail data kata di kamus, terdiri dari Kata Bahasa Indonesia dan Artinya dalam Bahasa Gorontalo. Terdapat button Edit untuk mengubah data kata, button Hapus untuk menghapus data kata, button Export Excel untuk mengunduh data kata, dan button Tambahkan Data untuk menambah data Kata.

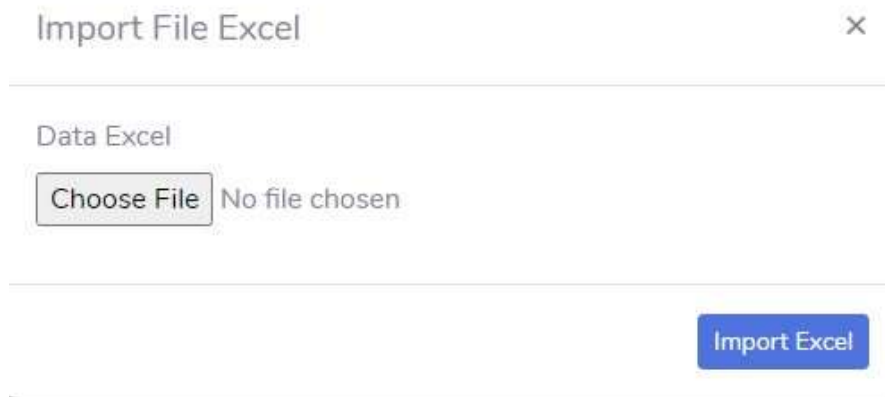
5.1.4 Tampilan Halaman *Export Data Kata File Exel*



Gambar 5. 4: Tampilan Halaman *Export Data Kata File Exel*

Halaman ini digunakan untuk melakukan *export* data kata dalam *file* excel . klik export untuk melakukan *export* data.

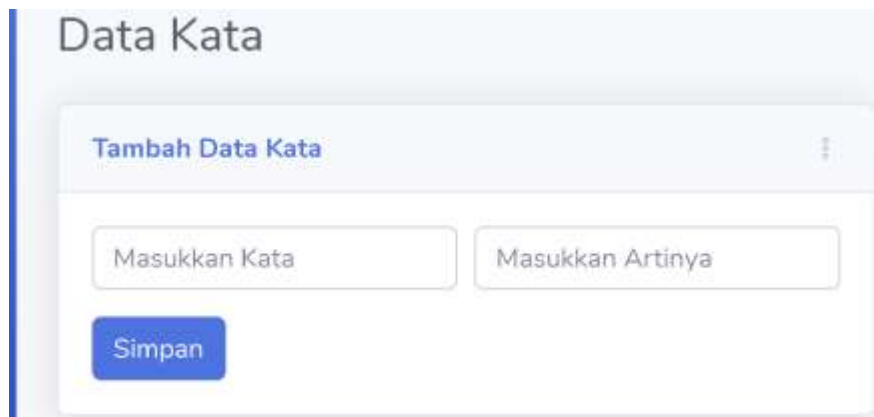
5.1.5 Tampilan Halaman *Import Data Kata File Excel*



Gambar 5. 5: Tampilan Halaman *Import Data Kata File Excel*

Halaman ini digunakan untuk melakukan *import* data kata kamus. Dimulai dengan klik *button choose file* untuk memasukan data kata. Klik *button import excel* untuk melanjutkan proses *import* data Kata.

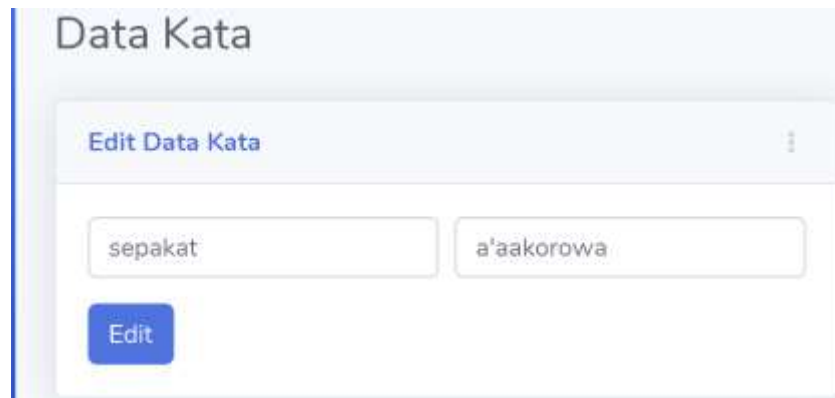
5.1.6 Tampilan Halaman *Tambah Data Kata*



Gambar 5. 6: Tampilan Halaman *Tambah Data Kata*

Halaman ini digunakan untuk menambahkan data kata kamus. Dimulai dengan memasukkan Kata dan Artinya. Klik Simpan untuk melanjutkan proses penyimpanan data Kata.

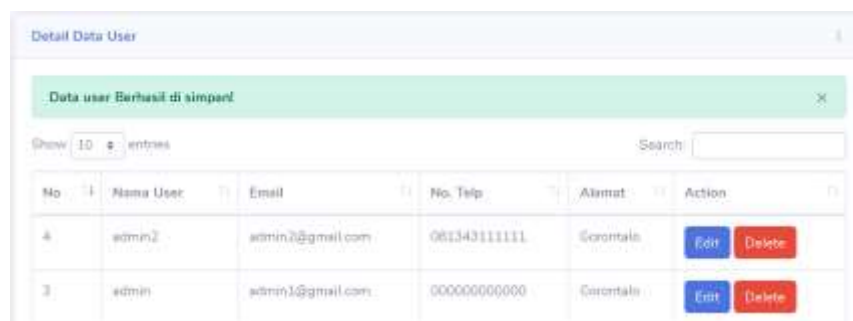
5.1.7 Tampilan Halaman Edit Data Kata



Gambar 5. 7: Tampilan Halaman Edit Data Kata

Halaman ini digunakan untuk mengubah data kata, dimulai dengan memasukkan perubahan kata beserta artinya. Klik button Edit untuk melanjutkan proses pengubahan data Kata.

5.1.8 Tampilan Halaman Data User

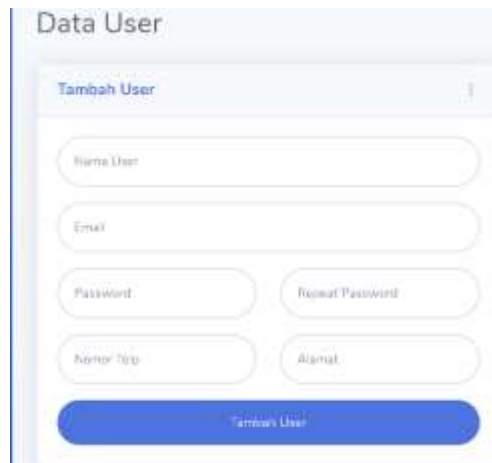


No	Nama User	Email	No. Telp	Alamat	Action
4	admin2	admin2@gmail.com	081343111111	Gorontalo	Edit Delete
3	admin	admin1@gmail.com	000000000000	Gorontalo	Edit Delete

Gambar 5. 8: Tampilan Halaman Data User

Halaman ini digunakan untuk menampilkan detail 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.

5.1.9 Tampilan Halaman Tambah Data User

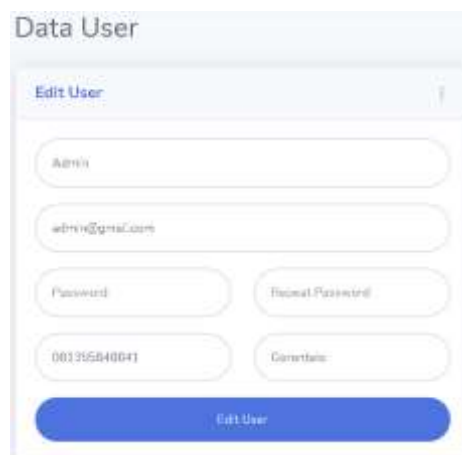


The screenshot shows a mobile application interface for adding a new user. The title bar at the top is light blue and contains the text 'Data User'. Below the title bar, there is a header 'Tambah User' in blue text. The form consists of several input fields: 'Nama User', 'Email', 'Password', 'Repeat Password', 'Nomor Telp', and 'Alamat'. At the bottom of the form is a large blue button labeled 'Tambah User'.

Gambar 5. 9: Tampilan Halaman Tambah Data User

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

5.1.10 Tampilan Halaman Edit Data User

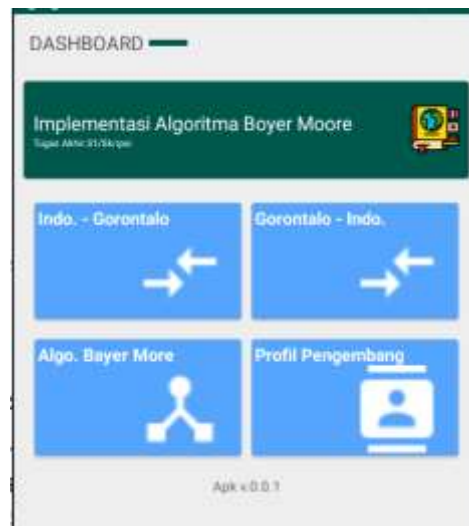


The screenshot shows a mobile application interface for editing an existing user. The title bar at the top is light blue and contains the text 'Data User'. Below the title bar, there is a header 'Edit User' in blue text. The form consists of several input fields: 'Admin', 'Email', 'Password', 'Repeat Password', 'Nomor Telp', and 'Alamat'. At the bottom of the form is a large blue button labeled 'Edit User'.

Gambar 5. 10: Tampilan Halaman Edit Data User

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

5.1.11 Tampilan Halaman User



Gambar 5. 11: Tampilan Home User

Halaman ini merupakan halaman utama dari sisi User, halaman ini dapat menampilkan data kamus Indo. – Gorontalo, kamus Gorontalo – Indo, seputar Algoritma Bayer Moore, dan Profil Pengembang.

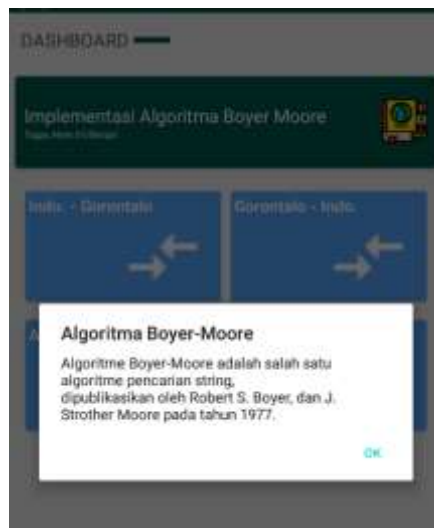
5.1.12 Tampilan Halaman Data Profil



Gambar 5. 12: Halaman data Profil

Halaman ini menampilkan data Tentang pengembang sistem, terdiri dari Nama Profil, NIM, dan Judul Skripsi.

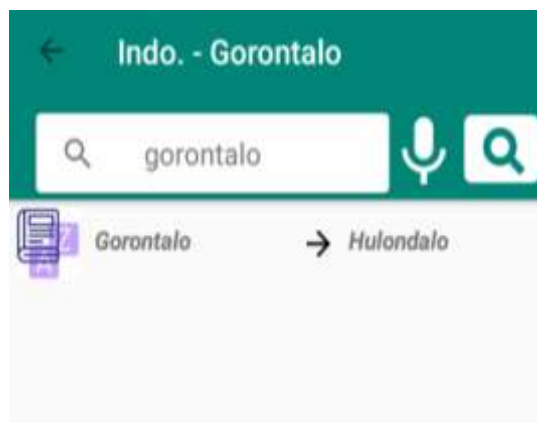
5.1.13 Tampilan Halaman Algoritma Boyer-Moore



Gambar 5. 13: Halaman Algoritma Boyer Moore

Halaman ini menampilkan informasi seputar Algoritma Boyer Moore. Untuk kembali ke Halaman Utama User, klik OK

5.1.14 Tampilan Halaman Kamus Indonesia - Gorontalo



Gambar 5. 14: Halaman Kamus Indonesia – Gorontalo

Tampilan ini digunakan untuk mencari data Kamus Indonesia – Gorontalo, dimulai dengan memasukkan kata yang akan dicari, klik search untuk melanjutkan proses pencarian. Kata yang di cari akan ditampilkan apabila kata ditemukan.

5.1.15 Tampilan Halaman Kamus Indonesia – Gorontalo *speech to text*



Gambar 5. 15 Halaman Kamus Indonesia – Gorontalo *speech to text*

Tampilan ini digunakan untuk mencari data Kamus Indonesia – Gorontalo, dimulai dengan melakukan *speech to text* kata yang akan dicari, klik search untuk melanjutkan proses pencarian. Kata yang di cari akan ditampilkan apabila kata ditemukan.

5.1.16 Tampilan Halaman Kamus Gorontalo - Indonesia



Gambar 5. 16: Halaman Kamus Gorontalo - Indonesia

Tampilan ini digunakan untuk mencari data Kamus Gorontalo - Indonesia, dimulai dengan memasukkan kata yang akan dicari, kemudian klik search untuk melanjutkan proses pencarian. Kata yang di cari akan ditampilkan apabila ditemukan.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan pembahasan pada bab-bab sebelumnya maka dapat disimpulkan sebagai berikut :

1. Algoritma boyer-moore dapat diterapkan untuk melakukan pencocokan kata bahasa Indonesia dan Gorontalo dengan cara melakukan persamaan *string* pada kata yang di *input* oleh user. Apabila kata yang di *input* sesuai dengan data yang ada pada *database* maka data akan ditampilkan beserta dengan artiya.
2. Berdasarkan hasil pengujian akurasi pencarian dapat disimpulkan bahwa tingkat akurasi pencarian pada aplikasi kamus Indonesia Gorontalo mencapai 100% karena jumlah data yang didapat sama dengan data yang ada didalam *database*.

6.2 Saran

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

1. Menambahkan fasilitas pencarian kalimat
2. Menambahkan keterangan dalam penggunaan kata
3. Menambahkan data kata yang lebih banyak lagi.

DAFTAR PUSTAKA

- [1] 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.
- [2] Supriyadi, "Pemertahanan Bahasa-Bahasa Minoritas Di Provinsi Gorontalo Untuk Mengangkat Budaya Lokal," *Semin. Nas. Bhs. Ibu*, vol. 53, no. 9, pp. 73–85, 2016, doi: 10.1017/CBO9781107415324.004.
- [3] Y. J. Wenny Hulukati, Maryam Rahim, "Pengembangan Panduan Untuk Meningkatkan Kompetensi Guru Melaksanakan Pembelajaran Muatan Lokal Bahasa Daerah Gorontalo Bagi Anak Usia Dini," *Lap. Penelit. Akhir Tahun (Tahap 2)*, pp. 1–297, 2017.
- [4] 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.
- [5] Y. J. Wenny Hulukati, Maryam Rahim, "Pengembangan Panduan Untuk Meningkatkan Kompetensi Guru Melaksanakan Pembelajaran Muatan Lokal Bahasa Daerah Gorontalo Bagi Anak Usia Dini," *Lap. Penelit. Akhir Tahun (Tahap 3)*, pp. 1–57, 2018.
- [6] N. Fadisar and S. S. Darnila, E, "Aplikasi Kamus Bahasa Indonesia-Arab Berbasis Android Menggunakan Metode Boyer-Moore," *Ojs.Unimal.Ac.Id*, pp. 55–76, 2017.
- [7] E. Rahmanita, "Pencarian String Menggunakan Algoritma Boyer Moore Pada Dokumen," *J. Ilm. NERO*, vol. 1, no. 1, pp. 15–26, 2014.
- [8] A. N. Arrasyid, "Aplikasi Kamus Bahasa Daerah Tolaki Berbasis Android," vol. 1, no. 1, pp. 62–68, 2016.
- [9] Ali Subhan Afrizal, "Rancang Bangun Aplikasi Dekstop Kamus Indonesia , Inggris Dan Arab Menggunakan Netbeans Dan Mysql," *J. Tek. Inform. Politek. Sekayu*, vol. 1, no. 1, pp. 1–9, 2014.
- [10] Z. D. L. Farah, "Pengaruh Penggunaan Kamus Elektronik (E-Dictionary) Tematik Bergambar Pemrograman Visual Basic 6.0 Terhadap Penguasaan

Kosakata Bahasa Jepang Pada Siswa Kelas X SMK Negeri 1 Prigen Tahun Ajaran 2018/2019,” vol. 8, no. 5, p. 55, 2019.

- [11] A. Darmawan dan Setianingrum, “Implementasi Algoritma Boyer Moore Pada Aplikasi Kamus Istilah Kebidanan Berbasis Web,” *Query*, vol. II, no. April, pp. 53–62, 2018.
- [12] I. Priana and L. Fitriani, “Perancangan Aplikasi Perangkat Lunak Pengelolaan Data Bank Sampah di PT. Inpower Karya Mandiri Garut,” *J. Algoritm.*, vol. 13, no. 2, pp. 407–413, 2017, doi: 10.33364/algoritma/v.13-2.407.
- [13] H. Mulyana, “Aplikasi Mobile Kamus Istilah Komputer Berbasis Android,” *Penelit. Ilmu Komput. Sist. Embed. dan Log.*, vol. 2, no. 1, pp. 27–34, 2014.
- [14] Ivan Alfatih Saputra, “Aplikasi Layanan Bengkel Mobil Berbasis Android di Kota Bandar Lampung,” *Univ. Lampung*, p. 137, 2017.
- [15] E. Maiyana, “Pemanfaatan Android Dalam Perancangan Aplikasi Kumpulan Doa,” *J. Sains dan Inform.*, vol. 4, no. 1, pp. 54–65, 2018, doi: 10.22216/jsi.v4i1.3409.
- [16] A. Maslan, Y. Setiono, and F. Alfazri, “Pengembangan Smart Application Translation Aneka Bahasa Sulawesi Berbasis Android,” *Teknosi*, vol. 02, no. 01, pp. 55–64, 2016.
- [17] K. Kawano, Y. Umemura, and Y. Kano, “Pemodelan UML Sistem Informasi Monitoring Penjualan Dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak),” *Informatika*, vol. 23, no. 2, pp. 201–205, 2016, doi: 10.2135/cropsci1983.0011183x002300020002x.
- [18] Y. S. Dwanoko, “Implementasi Software Development Life Cycle (Sdlc) Dalam Penerapan Pembangunan Aplikasi Perangkat,” *J. Teknol. Inf.*, vol. 7, no. 2, pp. 83–94, 2016.
- [19] C. D. Prabowo, C. Putih, and A. U. Digital, “Pemanfaatan System Development Life Cycle,” vol. 2009, no. semnasIF, pp. 228–232, 2009.
- [20] R. Muhidin, N. F. Kharie, and M. Kubais, “Analisis Dan Perancangan Sistem Informasi Pada Sma Negeri 18 Halmahera Selatan Sebagai Media Promosi Berbasis Web,” *IJIS - Indones. J. Inf. Syst.*, vol. 2, no. 2, pp. 56–

68, 2017, doi: 10.36549/ijis.v2i2.31.

- [21] A. R. Adiguna, M. Saputra Chandra, and F. Pradana, “Analisis dan Perancangan Sistem Informasi Manajemen Gudang pada PT Mitra Pinasthika Mulia Surabaya,” *Pengantar Sist. Inf.*, vol. 2, no. 2, pp. 612–621, 2018, doi: 10.1016/j.humimm.2008.04.008.
- [22] F. Muttaqin, “Analisi dan desain sistem informasi berbasis komputer untuk persediaan barang pada toko bahan bangunan (Studi Kasus pada UD. Sumber Bumi Subur),” *J. Adm. Bisnis S1 Univ. Brawijaya*, vol. 8, no. 1, pp. 1–7, 2014.
- [23] M. S. Mustaqbal, R. F. Firdaus, and H. Rahmadi, “Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN),” vol. I, no. 3, pp. 31–36, 2015.
- [24] W. N. Cholifah, Y. Yulianingsih, and S. M. Sagita, “Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap,” *STRING (Satuan Tulisan Ris. dan Inov. Teknol.*, vol. 3, no. 2, p. 206, 2018, doi: 10.30998/string.v3i2.3048.
- [25] M. Komarudin, “Pengujian perangkat Lunak metode Black box berbasis partitions pada aplikasi sistem informasi di sekolah,” *J. Mikrotik*, vol. o6, no. 3, pp. 02–16, 2016, doi: <http://dx.doi.org/10.1016/j.brat.2012.09.004>.
- [26] M. Nuris, “White box testing pada sistem penilaian pembelajaran,” *Skripsi Tek. Inform.*, pp. 1–102, 2015.

LAMPIRAN

Jadwal Penelitian

[illegible]

Bahasa Indonesia - Gorontalo

No	Bahasa Indonesia	Bahasa Gorontalo
1	sepakat	a'aakorowa
2	saling berjauhan	a'aapanga
3	disapu	a'aato
4	disapu	a'atalo
5	sapu	a'ato
6	sapu ijuk	aa'ata dunula
7	sapu lidi	aa'ata hiyo
8	sapu	aa'ato
9	acar	aacari
10	adat	aadati
11	adil	aadili
12	disukai	aahe'o
13	akhirat	aahirati
14	alam	aalamu
15	aman	aamani
16	halus	aaruti
17	kasihan	aati
18	kasihan sekali	aati olo
19	ayah	aba
20	lamban	abalo
21	pisang batu	abati
22	pisang batu masi muda	abati mayihulalo
23	ungkit	abi
24	membuka	abila
25	pegang	abito

No	Bahasa Indonesia	Bahasa Gorontalo
26	membuka sedikit	abiyolo
27	kurap (penyakit)	abongo
28	pisang yang sudah masak	aboya
29	kelelawar	abuhu
30	adab	adabu
31	nangka yang masi muda	adelo
32	agama	agama
33	minggu	Ahadi
34	akhlak	ahalaki
35	ahli	ahali
36	agen	ahengi
37	kerak	ahengo
38	kerak nasi	ahengo ila
39	kerak hidung	ahengo wulingo
40	kidal	ahingo
41	tenggelm	ahumo
42	ajal	ajali
43	akad	akaji
44	di akad	akajiyolo
45	akal	akali
46	akikah	akiiki
47	setuju	akoro
48	dahak	ala'o
49	alamat	alaamati
50	kemenyang	alama

No	Bahasa Indonesia	Bahasa Gorontalo
51	kemenyang tanah	alama huta
52	kemenyang api	alama tulu
53	layang-layang	alanggaya
54	layang-layang rumah	alanggaya bele
55	layang-layang elang	alanggaya buliya
56	layang bersudut empat	alanggaya molo'u
57	bantal	alanguluwa
58	kunyit	alawahu
59	sumur	ali
60	supaya	alihu
61	arkus	alikusu
62	kupu-kupu	alinguwa
63	kulit	alipo
64	sarung	alipopo
65	alif	alipu
66	kelumpang	alumbango
67	peniti	alumbeneti
68	berburu	alupo
69	amanat	amaanati
70	amal	amali
71	sebelah	ambahu
72	penyakit kulit	ambalo
73	sukun	amo
74	tikar	amongo
75	sedangkan	amula

No	Bahasa Indonesia	Bahasa Gorontalo
76	hingga	anggađu
77	bentak	antingo
78	lebih	apingo
79	anjing	apula
80	rabu	Araba'a
81	kentang	artape
82	ketela	atetela
83	terong babi	atimbu
84	ketupat	atupato
85	samudra	awuhu
86	kayu	ayu
87	jagung yang sudah digiling	ba'a lo binte
88	jejak	ba'ato
89	kepalan tangan	baa'alo
90	bandel	baa'angi
91	balak	baalaki
92	kolam	baalanga
93	kakek	baapu
94	batik	baate
95	bawang	baawangi
96	saling menarik	babaantanga
97	bedak	bada'a
98	dipakaikan bedak	bada'aalo
99	memakai bedak	bada-bada'a
100	nasi gula merah	bajo'e

No	Bahasa Indonesia	Bahasa Gorontalo
101	kerangka	bakakawu
102	pagar	bala
103	pagar	bala
104	kutu sapi	bala'i
105	sedang menyebrang	bala-balango
106	sedang berbaring	bala-balato
107	sedang belajar	balaajari
108	urunan	balaantangi
109	kolam	balangga
110	menyebrang	balango
111	belanja	balanja
112	balsem	balasemu
113	baring	balato
114	belok	bale
115	gelembang	baleyango
116	belimbing	balimbi
117	benar	banari
118	terengah-engah	banga-bangango
119	kutil	bangalo
120	engah	bangango
121	sementara direbus	bangga-banggahu
122	direbuskan	banggahelo
123	direbus	banggahu
124	kasar	bangganga
125	jantan	bangge

No	Bahasa Indonesia	Bahasa Gorontalo
126	kelelewang	banggo
127	angsa	bano
128	ditarik	bantangalo
129	tarik	bantango
130	dangau	bantayo
131	malas	bantila
132	berani	barani
133	kambing	batade
134	batal	batali
135	batrei	batereyi
136	beton	batonu
137	muka	baya
138	rumah	bele
139	belek	beleki
140	itik	bibi'o
141	saling memandang	bibiilohe
142	bengkok sana sini	bibiinggewunga
143	peliharaan	bibiyahu
144	pemberat	biibi
145	bibir	biihu
146	kerikil	biilato
147	dilihat	biilohe
148	sudah disebrangi	bilalango
149	menjadi terengah-engah	bilangango
150	sudah direbus	bilanggahe

No	Bahasa Indonesia	Bahasa Gorontalo
151	betina	bilango
152	sudah ditarik	bilantanga
153	telur busuk	bili
154	bubungan	biliinga
155	dipelihara	biliyaheyi
156	dipelihara	biliyahu
157	melihat-lihat	biloh'e'o-biloh'e'o
158	dilihat	bilohelo
159	menarik	bandanga
160	lihat	bilohi
161	lihat	bilohu
162	hutang	biloli
163	burung puyu	bilu-bilulu
164	beludru	biludu
165	kebiri	bilutuwa
166	binatang	binaatangi
167	terbuka sedikit	binga-bingalo
168	dibuka sedikit	bingalolo
169	bengkok	binggewungo
170	dibengkokkan	binggewungolo
171	lereng	binggi
172	pundak	binggungo
173	istri	bini
174	angkat	binta'o
175	jagung	binte

No	Bahasa Indonesia	Bahasa Gorontalo
176	bicara	bisala
177	keris	bitu'o
178	ubi hutan	bitule
179	sedang dipelihara	biya-biyahu
180	pelihara	biyahu
181	biawak	biyawa'o
182	hati	biyodu
183	gila	biyongo
184	hanya	bo
185	baju	bo'o
186	jerawat	bobira
187	berubah-ubah	bobooli'a
188	bisul	bohe
189	geraham	bohengo
190	arak	bohito
191	arang	bohu
192	pertama	bohuliyo
193	langsar	bohulo
194	patah	boli
195	patahlan	boli'a
196	ubahlah	boli'alo
197	ubah	boli'o
198	diubah	boli'olo
199	terlalu	boti
200	ini	botii

No	Bahasa Indonesia	Bahasa Gorontalo
201	ini	botiya
202	pingsan	boto'o
203	batu	botu
204	naik	botulo
205	tahi	bowuntu
206	babi	boyi
207	itu	boyito
208	lambat	boyu
209	tempurung	bu'awu
210	boneka	bu'i
211	ombak	bu'olo
212	siput	bu'oyo
213	rusa	bu'ulu
214	penyemrangan	bubalanga
215	tempat tidur	bubalata
216	mencuci pakaian	bubo'o
217	tupai	bubudu
218	berani	buheli
219	berat	buheto
220	ikat	buhuto
221	danau	bulalo
222	mengutang	buli
223	belalang	bulita
224	elang	buliya
225	bengkarung	buliya'ata

No	Bahasa Indonesia	Bahasa Gorontalo
226	leher	bulo'o
227	wajan	bulonggo
228	guntur	bulonggodu
229	pinjam	buloto
230	perahu	bulotu
231	ribut	buluhuto
232	peti	buluwa
233	pertama	bungaliyo
234	kikir	bunggili
235	tuli	bungolo
236	paha	bungolopa
237	ditembak	butahulo
238	betis	butiyoto
239	hukum	buto'o
240	tawon	butota
241	kenyang	butuhu
242	cambuk	buubo'o
243	bubur	buuburu
244	burung	buurungi
245	botol	buutulu
246	kumbang	buwahanga
247	kalkulasi	caakari
248	sedang mengakulasi	caakariyolo
249	kotor	caparuni
250	stempel	capu

No	Bahasa Indonesia	Bahasa Gorontalo
251	cat	ceti
252	sarat muatan	ciikari
253	kutang	cole
254	cukup	cukupu
255	besar	da'a
256	dakwa	da'awa
257	kepala kampung	daa'a
258	dai	daa'i
259	daging	daagingi
260	banyak	daata
261	tetapi	debo
262	sambal	dabu-dabu
263	banyak	dadaata
264	jaga	daha
265	jalan	dalalo
266	dalil	dalili
267	damar	damahu
268	besar	damango
269	jambu	dambu
270	jengot	danggu
271	daftar	daputari
272	dataran	datahu
273	tulang rusuk	datala
274	samurai	dawata
275	tidak	de'e

No	Bahasa Indonesia	Bahasa Gorontalo
276	daki	de'i
277	nyala	de'ito
278	tangkap	de'upo
279	sentuh	dedeto
280	laut	deheto
281	jatuh	dehu
282	bawa	delo
283	gigit	dengeto
284	antar	depito
285	dapur	depula
286	muat	detohu
287	jahit	detu
288	hujan	didi
289	lesung	didingga
290	monyet	dihe
291	tidak	diila
292	cium	diilo
293	lidah	dila
294	dinding	dingingo
295	licin	dipulato
296	tidak ada	diya'a
297	langkah	diyambango
298	uang	doyi
299	doa	du'a
300	arang	du'ito

No	Bahasa Indonesia	Bahasa Gorontalo
301	kukuran	dudangata
302	bawaan	dudelo
303	jarum	dudetu
304	dada	duhelo
305	tambah	duhengo
306	duri	duhi
307	siang	dulahu
308	mari	dulo
309	marilah	dulolo
310	dua	duluwo
311	pagi-pagi	dumodupo
312	gigi	dungito
313	papan	dupi
314	durian	duriya
315	sungai	dutula
316	daun	duungo
317	bangau	duwayo
318	nike	duwo
319	kapuk	duyungo
320	tuhan	e'a
321	teringat	e'eela
322	ingat	eela
323	diingat	eelalo
324	ember	eemberi
325	keguguran	eenggila

No	Bahasa Indonesia	Bahasa Gorontalo
326	tuhan	eeya
327	meskipun	eleponu
328	keris	eluto
329	peot	embe
330	peot	embeto
331	dilarang	entelalo
332	larang	entelo
333	pukat	eputo
334	es	esi
335	jaring ikan	eyambu
336	kecewa	gaahari
337	gambar	gaambari
338	gambir	gaambebe
339	gadai	gade
340	digadaikan	gade-gade
341	pegadaian	gadeya
342	gaji	gaji
343	gelang	galangi
344	geledah	galeda
345	gampang	gambang
346	gambus	gambusi
347	genap	ganapu
348	karet	goro
349	gerobak	goroba
350	gorontalo	Hulontalo

No	Bahasa Indonesia	Bahasa Gorontalo
351	gua	gowa
352	konon	haantai
353	hafal	haapali
354	khatam	haatamu
355	kabar	habari
356	hadis	hadisi
357	hadiah	hadiya
358	peniti	haki
359	tegalan	halabolu
360	gergaji	halahadi
361	halal	halale
362	gelas	halati
363	tabiat	hale
364	kamis	Hamisi
365	handuk	hando
366	menumpang	hangato
367	harga	haraga
368	haram	haramu
369	harap	harapu
370	harta	harata
371	kurus	hata
372	hawatir	hawaatiri
373	panjang	haya'o
374	rakit	he'ita
375	ikat	he'odu

No	Bahasa Indonesia	Bahasa Gorontalo
376	tutup	he'uto
377	eja	heda
378	awan	heengo
379	menetes	heheto
380	rampas	hehu
381	udang	hele
382	setuju	helidu
383	keliling	helili
384	sepakat	helumo
385	kering	hengu
386	sedang dilihat	hepo'obilohulo
387	sedang bermain	hepoyitohe
388	sedang dimainkan	hepoyitohu
389	ratus	hetuto
390	goyang	hewulo
391	pindah	heyi
392	hibah	hiba
393	manja	hidi
394	lama	hihewo
395	gilingan	hihilinga
396	rambut	hihitinga
397	beda	hiihede
398	iri	hihiya
399	tiup	hiipo
400	hilap	hilapu

No	Bahasa Indonesia	Bahasa Gorontalo
401	hati	hilawo
402	minta	hile
403	giling	hilingo
404	domba	himba
405	gitar	hitari
406	pasangan hidup	hiyalo
407	sedangkan	hiyambola
408	celah	hiyango
409	ceroboh	hiyanta
410	lidi	hiyo
411	cincin	hu'alimo
412	gunung	hu'idu
413	rumput	hu'oyoto
414	gedung	hudungu
415	pintu	huhebu
416	istirahat	huheli
417	kepiting	huhemo
418	bantal peluk	huheputo
419	cuci	huheto
420	sisir	huheyidu
421	keranda	huhulihe
422	dingin	huhulo
423	gunting	huhuntingo
424	perbuatan	huhutu
425	gula	hula

No	Bahasa Indonesia	Bahasa Gorontalo
426	putik	hula'o
427	bulan	hulalo
428	emas	hulawa
429	benih	hulawu
430	sembrono	huleleto
431	mengigau	hulimamango
432	besar	hulinta
433	duduk	hulo'o
434	bodoh	hulodu
435	cabul	huloto
436	agas	huloto'o
437	langit	hulungo
438	kencur	humopoto
439	gunting	huntingo
440	rotan	hutiya
441	kusta	hutungo
442	kentut	hututo
443	gotong royong	huuyula
444	menguap	huwabu
445	kamar	huwali
446	ketiak	huwange'e
447	lubang	huwango
448	buaya	huwayo
449	mabuk	huwo'o
450	ingus	huwodu

No	Bahasa Indonesia	Bahasa Gorontalo
451	malah	huyi
452	tengah malam	huyi da'a
453	sedang tertawa	hemo I'i
454	tertawa	i'i
455	berkedip	ilailapo
456	geli	i'ilo'o
457	sembelit	i'imbu
458	ibadah	ibaadati
459	ibarat	ibaarati
460	iblis	ibilisi
461	dipanggil	ibodelo
462	panggil	ibodu
463	ikhlas	ihilasi
464	nasi	ila
465	kilat	ilato
466	diingat	ileela
467	kebun	ileengi
468	dilarang	ilentela
469	ilmu	ilimu
470	berhutang	ilobiloli
471	karya	ilomata
472	sebelah	imbihu
473	pegal	imi'ito
474	sampai atau hingga	inggidu
475	islam	isilamu

No	Bahasa Indonesia	Bahasa Gorontalo
476	senin	Isinini
477	setrika	isiriki
478	kami	ito
479	senyum	iyomo
480	jambu	jaabu
481	jahil	jaahili
482	bersenandung	jajaabu
483	zakat	jakati
484	dibayarkan zakat	jakatiyalo
485	jaksa	jakisa
486	mengomel	jalo
487	sedang mengomel	jalo-jalo
488	dimarahi	jalowalo
489	jemaah	jamaa'ati
490	zaman	jamani
491	jangkar	jangga
492	jepang	Japangi
493	jasad	jasadi
494	zaman	jamani
495	jangkar	jangga
496	jepang	Japangi
497	jasad	jasadi
498	jaket	jeketi
499	jin	jini
500	ziarah	jiyara

No	Bahasa Indonesia	Bahasa Gorontalo
501	ya	jo
502	jumat	Juma'ati
503	jubah	jumba
504	jumlah	jumula
505	kebal	kaabali
506	aduk	kaahu
507	jamban	kaakusi
508	kamar	kaamari
509	kampung	kaambungu
510	kandas	kaandasi
511	kapal	kaapali
512	kafir	kaapiru
513	kapur	kaapuru
514	kasur	kaatulu
515	kain	kaayini
516	kebaya	kabaya
517	sobek	kadango
518	gali	kakudu
519	keranjang	kalanji
520	coret, gores	kaluhu
521	keluar	kaluuwari
522	kemeja	kameja
523	kemiri	bindalo
524	comong	kanuta
525	kerja	karaja

No	Bahasa Indonesia	Bahasa Gorontalo
526	kertas	karatasi
527	hitam legam	kasopa
528	ketimun	katimu
529	ketumbar	katuumbali
530	ketua	katuwa
531	jangan	keke
532	kernet	keneti
533	kiblat	kibilati
534	kecil	kiki'o
535	geli	kikilo'o
536	kecil	kikingo
537	kecil	kikino
538	ranjang	koyi
539	cubit	kubingo
540	kudis	kukudu
541	contek	kupe'i
542	Al - Qur'an	kuru'ani
543	berkurban	kurubani
544	kerupuk	kurupu
545	kusir	kusiri
546	kubur	kuuburu
547	kuas	kuwasi
548	laknat	la'anati
549	nyala	la'ito
550	turun	laahu

No	Bahasa Indonesia	Bahasa Gorontalo
551	telanjang	laala
552	sagu	labiya
553	rebus	lahe
554	sedang direbus	lahe-lahe
555	direbus	laheelo

No	Bahasa Indonesia	Bahasa Gorontalo
556	cabut	lahuto
557	kunig	lalahu
558	janur	lale
559	tajam	lalito
560	rakus	lalo



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
KANTOR BAHASA
GORONTALO

Jalan dr.Zainal Umar Sidiki, Tunggulo, Tilongkabila, Bone Bolango, Gorontalo
Telepon/Faksimile 0435-8533588
Pos-el bahasa.gorontalo@kemdikbud.go.id

SURAT KETERANGAN

Nomor: 0343./14.24/KP./2020

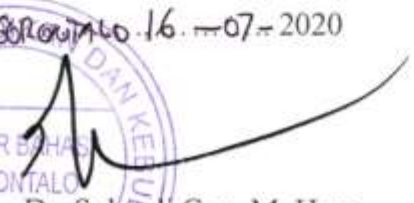
Yang bertanda tangan di bawah ini

nama : Dr. Sukardi Gau, M.Hum.
NIP : 197511292001121001
pangkat dan golongan : Pembina/IVa
jabatan : Kepala
unit organisasi : Kantor Bahasa Gorontalo

dengan ini menerangkan bahwa mahasiswa atas nama **Ishak Heluwa**, pada Fakultas Ilmu Komputer, Universitas Ichsan Gorontalo, benar-benar telah mengadakan penelitian di Kantor Bahasa Gorontalo pada tanggal 10—11 Februari 2020.

Demikian surat keterangan ini dibuat untuk dapat dipergunakan sebagaimana mestinya.

Gorontalo, 16.07.2020



Dr. Sukardi Gau, M. Hum.
NIP197511292001121001

Kode Program Java

1. Activity

▪ Dashboard

```
1. package com.apracha.boyermoorealgorithm.Activity;
2.
3. import androidx.appcompat.app.AlertDialog;
4. import androidx.appcompat.app.AppCompatActivity;
5.
6. import android.content.Intent;
7. import android.os.Bundle;
8. import android.view.View;
9. import android.widget.RelativeLayout;
10.
11. import com.apracha.boyermoorealgorithm.R;
12.
13. public class Dashboard extends AppCompatActivity {
14.
15.     private RelativeLayout mnIndo, mnGorontalo, mnAboutAlgo, mnProfil;
16.     @Override
17.     protected void onCreate(Bundle savedInstanceState) {
18.         super.onCreate(savedInstanceState);
19.         setContentView(R.layout.activity_dashboard);
20.
21.         mnIndo = findViewById(R.id.mn_indo);
22.         mnGorontalo = findViewById(R.id.mn_gorontalo);
23.         mnAboutAlgo = findViewById(R.id.mn_aboutalgo);
24.         mnProfil = findViewById(R.id.mn_profil);
25.
26.         mnProfil.setOnClickListener(new View.OnClickListener() {
27.             @Override
28.             public void onClick(View v) {
29.                 AlertDialog.Builder pesan = new AlertDialog.Builder(Dash
board.this);
30.                 pesan.setTitle("Tentang Pengembang");
31.                 pesan.setMessage("Nama : Ishak Heluwa \n" +
32.                     "NIM : T3116040 \n" +
33.                     "Judul Skripsi : Implementasi Algoritma Boyer Mo
ore Untuk Kamus Indonesia Gorontalo");
34.                 pesan.setPositiveButton("Ok", null);
35.                 pesan.show();
36.             }
37.         });
38.
39.         mnAboutAlgo.setOnClickListener(new View.OnClickListener() {
40.             @Override
41.             public void onClick(View v) {
42.                 AlertDialog.Builder pesan = new AlertDialog.Builder(Dash
board.this);
43.                 pesan.setTitle("Algoritma Boyer-Moore");
44.                 pesan.setMessage("Algoritme Boyer-
Moore adalah salah satu algoritme pencarian string, \n" +
45.                     "dipublikasikan oleh Robert S. Boyer, dan J. Str
other Moore pada tahun 1977.");
46.                 pesan.setPositiveButton("Ok", null);
47.                 pesan.show();
48.             }
49.         });
50.     }
51. }
```

```

49.         });
50.
51.         mnIndo.setOnClickListener(new View.OnClickListener() {
52.             @Override
53.             public void onClick(View v) {
54.                 startActivity(new Intent(getApplicationContext(),IndoGor
ontalo.class));
55.             }
56.         });
57.
58.         mnGorontalo.setOnClickListener(new View.OnClickListener() {
59.             @Override
60.             public void onClick(View v) {
61.                 startActivity(new Intent(getApplicationContext(),Goronta
loIndo.class));
62.             }
63.         });
64.     }
65. }

```

▪ Kamus Indonesia-Gorontalo

```

1. package com.apracha.boyermoorealalgorithm.Activity;
2.
3. import androidx.annotation.Nullable;
4. import androidx.appcompat.app.AppCompatActivity;
5. import androidx.appcompat.widget.Toolbar;
6. import androidx.recyclerview.widget.LinearLayoutManager;
7. import androidx.recyclerview.widget.RecyclerView;
8.
9. import android.annotation.SuppressLint;
10. import android.content.Intent;
11. import android.os.Bundle;
12. import android.speech.RecognizerIntent;
13. import android.view.View;
14. import android.widget.ImageButton;
15. import android.widget.ImageView;
16. import android.widget.TextView;
17. import android.widget.Toast;
18.
19. import com.apracha.boyermoorealalgorithm.Adapter.AdapterIndoGorontalo;
20. import com.apracha.boyermoorealalgorithm.Algo.BoyerMoore;
21. import com.apracha.boyermoorealalgorithm.Api.APIRequestData;
22. import com.apracha.boyermoorealalgorithm.Api.RetroServer;
23. import com.apracha.boyermoorealalgorithm.Model.DataModel;
24. import com.apracha.boyermoorealalgorithm.Model.ResponModel;
25. import com.apracha.boyermoorealalgorithm.R;
26. import com.arlib.floatingsearchview.FloatingSearchView;
27.
28. import java.util.ArrayList;
29. import java.util.List;
30. import java.util.Locale;
31.
32. import retrofit2.Call;
33. import retrofit2.Callback;
34. import retrofit2.Response;
35.
36. public class IndoGorontalo extends AppCompatActivity {
37.

```

```

38.     private static final int REQUEST_CODE_SPEECH_INPUT = 1000 ;
39.     //views from activity
40.
41.     ImageButton mVoiceBtn;
42.
43.     private RecyclerView rvData;
44.     private RecyclerView.Adapter adData;
45.     private RecyclerView.LayoutManager lmData;
46.     private List<DataModel> listData = new ArrayList<>();
47.
48.     private FloatingSearchView fsvSearch;
49.     private ImageView ivGo;
50.
51.     //toolbar
52.     private Toolbar toolbar;
53.
54.     @SuppressWarnings("RestrictedApi")
55.     @Override
56.     protected void onCreate(Bundle savedInstanceState) {
57.         super.onCreate(savedInstanceState);
58.         setContentView(R.layout.indo_gorontalo);
59.
60.         mVoiceBtn = findViewById(R.id.voiceBtn);
61.
62.         //button click to show speech to text dialog
63.         mVoiceBtn.setOnClickListener(new View.OnClickListener() {
64.             @Override
65.             public void onClick(View v) {
66.                 speak();
67.             }
68.         });
69.
70.         toolbar = findViewById(R.id.toolbar);
71.         setSupportActionBar(toolbar);
72.         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
73.         getSupportActionBar().setDefaultDisplayHomeAsUpEnabled(true);
74.
75.         rvData = findViewById(R.id.rv_data);
76.         fsvSearch = findViewById(R.id.fsv_search);
77.         ivGo = findViewById(R.id.iv_go);
78.
79.         lmData = new LinearLayoutManager(this,LinearLayoutManager.VERTICAL,false);
80.
81.         rvData.setLayoutManager(lmData);
82.
83.         fsvSearch.setOnQueryChangeListener(new FloatingSearchView.OnQueryChangeListener() {
84.             @Override
85.             public void onSearchTextChanged(String oldQuery, String newQuery) {
86.                 // searchIndoGorontalo(newQuery);
87.                 if (newQuery.isEmpty()){
88.                     rvData.setAdapter(null);
89.                 }
90.             }
91.         });
92.
93.         ivGo.setOnClickListener(new View.OnClickListener() {
94.             @Override
95.             public void onClick(View v) {

```

```

96.         if (fsvSearch.getQuery().isEmpty()){
97.             rvData.setAdapter(null);
98.         }else{
99.             searchIndoGorontalo(fsvSearch.getQuery().toString())
100.        };
101.        }
102.    });
103. }
104.
105.     private void speak() {
106.         //intent to show speech to text dialog
107.         Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
108.         intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
109.             RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
110.         intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault());
111.         intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Silahkan Berbicara");
112.
113.         //start intent
114.         try {
115.             //in there was some error
116.             //show dialog
117.             startActivityForResult(intent, REQUEST_CODE_SPEECH_INPUT);
118.         }
119.         catch (Exception e) {
120.             //if there was some error
121.             //get message of error and show
122.             Toast.makeText(this, ""+e.getMessage(), Toast.LENGTH_SHORT).show();
123.         }
124.     }
125.
126.     //Receive voice input and hadle it
127.
128.     @Override
129.     protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
130.         super.onActivityResult(requestCode, resultCode, data);
131.         switch (requestCode){
132.             case REQUEST_CODE_SPEECH_INPUT:{
133.                 if (resultCode == RESULT_OK && null!=data){
134.                     //get text array from voice intent
135.                     ArrayList<String> result = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
136.                     //set to text view
137.                     searchIndoGorontalo(fsvSearch.getQuery().toString());
138.                 }
139.                 break;
140.             }
141.         }
142.     }
143.
144.     private void searchIndoGorontalo(String newQuery) {
145.         APIRequestData ardData = RetroServer.koneksiRetrofit().create(APIRequestData.class);

```

```

146.         Call<ResponModel> tampilData = ardData.ardSearchDataIndo(
            newQuery);
147.
148.         tampilData.enqueue(new Callback<ResponModel>() {
149.             @Override
150.             public void onResponse(Call<ResponModel> call, Respon
                se<ResponModel> response) {
151.                 int kode = response.body().getKode();
152.                 String pesan = response.body().getPesan();
153.
154.                 listData = response.body().getData();
155.
156.
157.                 adData = new AdapterIndoGorontalo(IndoGorontalo.t
                    his,listData);
158.                 rvData.setAdapter(adData);
159.                 adData.notifyDataSetChanged();
160.
161.                 if (listData.isEmpty()){
162.                     Toast.makeText(IndoGorontalo.this, "Data belu
                        m tersedia!", Toast.LENGTH_SHORT).show();
163.                 }else{
164.                     for (int i = 0; i < listData.size(); i++){
165.                         BoyerMoore bm = new BoyerMoore();
166.                         int index = bm.findPattern(listData.get(i
                            ).getKata1(),newQuery);
167.                         if (index == -1){
168.                             Toast.makeText(IndoGorontalo.this, "B
                                    oyer Moore tidak dapat menemukan data!", Toast.LENGTH_SHORT).show();
169.                         }else{
170.                             Toast.makeText(IndoGorontalo.this, "B
                                    oyer Moore! Data ditemukan, pada posisi : "+index, Toast.LENGTH_SHORT).
                                        show();
171.                         }
172.                     }
173.                 }
174.
175.                 // Toast.makeText(IndoGorontalo.this, "Hasil Penca
                        rian ! kode : "+kode+" | Pesan : "+pesan, Toast.LENGTH_SHORT).show();
176.             }
177.
178.             @Override
179.             public void onFailure(Call<ResponModel> call, Throwab
                le t) {
180.                 Toast.makeText(IndoGorontalo.this, "Gagal menghub
                        ungi Server : "+t.getMessage(), Toast.LENGTH_SHORT).show();
181.             }
182.         });
183.     }
184.
185.     @Override
186.     protected void onResume() {
187.         super.onResume();
188.         // retrieveData();
189.     }
190.
191.     private void retrieveData() {
192.
193.         APIRequestData ardData = RetroServer.koneksiRetrofit().cr
            eate(APIRequestData.class);
194.         Call<ResponModel> tampilData = ardData.ardShowAllData();

```

```

195.         tampilData.enqueue(new Callback<ResponModel>() {
196.             @Override
197.             public void onResponse(Call<ResponModel> call, Respon
198.                 se<ResponModel> response) {
199.                 int kode = response.body().getKode();
200.                 String pesan = response.body().getPesan();
201.                 // Toast.makeText(MainActivity.this, "Kode : "+kod
202.                 e+" | Pesan : "+pesan, Toast.LENGTH_SHORT).show();
203.                 listData = response.body().getData();
204.                 adData = new AdapterIndoGorontalo(IndoGorontalo.t
205.                     his,listData);
206.                 rvData.setAdapter(adData);
207.                 adData.notifyDataSetChanged();
208.             }
209.         }
210.         @Override
211.         public void onFailure(Call<ResponModel> call, Throwab
212.             le t) {
213.             Toast.makeText(IndoGorontalo.this, "Gagal menghub
214.                 ungi Server", Toast.LENGTH_SHORT).show();
215.             // pbData.setVisibility(View.INVISIBLE);
216.             }
217.         });
218.     }
219.     @Override
220.     public void onBackPressed() {
221.         finish();
222.         super.onBackPressed();
223.     }
224.     @Override
225.     public boolean onSupportNavigateUp() {
226.         onBackPressed();
227.         return super.onSupportNavigateUp();
228.     }
229. }
230.

```

▪ Algoritma boyer-moore

```

1. package com.apracha.boyermoorealgorithm.Algo;
2.
3. import java.io.BufferedReader;
4. import java.io.IOException;
5. import java.io.InputStreamReader;
6.
7. public class BoyerMoore {
8.     /** function findPattern */
9.     public int findPattern(String t, String p) {
10.         char[] text = t.toCharArray();
11.         char[] pattern = p.toCharArray();
12.         int pos = indexOf(text, pattern);
13.         // if (pos == -1)
14.         //     System.out.println("\nNo Match\n");
15.         // else

```

```

16. //          System.out.println("Pattern found at position : "+ pos);
17.
18.     return pos;
19. }
20. /** Function to calculate index of pattern substring */
21. public int indexOf(char[] text, char[] pattern) {
22.     if (pattern.length == 0)
23.         return 0;
24.     int charTable[] = makeCharTable(pattern);
25.     int offsetTable[] = makeOffsetTable(pattern);
26.     for (int i = pattern.length - 1, j; i < text.length; j)
27.     {
28.         for (j = pattern.length - 1; pattern[j] == text[i]; --i, --
29.             if (j == 0)
30.                 return i;
31.
32.         // i += pattern.length - j; // For naive method
33.         i += Math.max(offsetTable[pattern.length - 1 -
34.             j], charTable[text[i]]);
35.         return -1;
36.     }
37.     /** Makes the jump table based on the mismatched character informati
38.     on */
39.     private int[] makeCharTable(char[] pattern) {
40.         final int ALPHABET_SIZE = 256;
41.         int[] table = new int[ALPHABET_SIZE];
42.         for (int i = 0; i < table.length; ++i)
43.             table[i] = pattern.length;
44.         for (int i = 0; i < pattern.length - 1; ++i)
45.             table[pattern[i]] = pattern.length - 1 - i;
46.         return table;
47.     }
48.     /** Makes the jump table based on the scan offset which mismatch occ
49.     urs. */
50.     private static int[] makeOffsetTable(char[] pattern) {
51.         int[] table = new int[pattern.length];
52.         int lastPrefixPosition = pattern.length;
53.         for (int i = pattern.length - 1; i >= 0; --i)
54.         {
55.             if (isPrefix(pattern, i + 1))
56.                 lastPrefixPosition = i + 1;
57.             table[pattern.length - 1 - i] = lastPrefixPosition -
58.                 i + pattern.length - 1;
59.         }
60.         for (int i = 0; i < pattern.length - 1; ++i)
61.         {
62.             int slen = suffixLength(pattern, i);
63.             table[slen] = pattern.length - 1 - i + slen;
64.         }
65.         return table;
66.     }
67.     /** function to check if needle[p:end] a prefix of pattern */
68.     private static boolean isPrefix(char[] pattern, int p) {
69.         for (int i = p, j = 0; i < pattern.length; ++i, ++j)
70.             if (pattern[i] != pattern[j])
71.                 return false;
72.         return true;
73.     }

```

```

71.     /** function to returns the maximum length of the substring ends at
    p and is a suffix */
72.     private static int suffixLength(char[] pattern, int p) {
73.         int len = 0;
74.         for (int i = p, j = pattern.length -
    1; i >= 0 && pattern[i] == pattern[j]; --i, --j)
75.             len += 1;
76.         return len;
77.     }
78. }

```

▪ Server Api Android

```

1. package com.apracha.boyermoorealgorithm.Api;
2.
3. import retrofit2.Retrofit;
4. import retrofit2.converter.gson.GsonConverterFactory;
5. public class RetroServer {
6.
7.     private static final String baseUrl = "http://192.168.43.22:8000/api/";
8.
9.     private static Retrofit retro;
10.
11.     public static Retrofit koneksiRetrofit(){
12.         if (retro == null){
13.             retro = new Retrofit.Builder()
14.                 .baseUrl(baseUrl)
15.                 .addConverterFactory(GsonConverterFactory.create())
16.                 .build();
17.
18.         return retro;
19.     }
20. }

```

Kode Program PHP

1. Login Admin

```
1. @extends('auth.app')
2. @section('title','Halaman Login')
3. @section('content')
4.
5. <div class="row">
6.     <div class="col-lg-12">
7.         <div class="p-5">
8.             <div class="text-center">
9.                 <h1 class="h4 text-gray-900 mb-4">@yield('title')</h1>
10.            </div>
11.            <form class="user" method="POST" action="{{ route('login') }}">
12.
13.                @csrf
14.                <div class="form-group">
15.                    <input type="email" class="form-control form-control-
16.                        user @error('email') is-invalid @enderror" id="email" name="email" aria-
17.                        describedby="emailHelp" placeholder="Enter Email Address...">
18.                    @error('email')
19.                        <span class="invalid-feedback" role="alert">
20.                            <strong>{{ $message }}</strong>
21.                        </span>
22.                    @enderror
23.                </div>
24.                <div class="form-group">
25.                    <input type="password" class="form-control form-control-
26.                        user @error('password') is-
27.                        invalid @enderror" name="password" id="password" placeholder="Password"
28.                    >
29.                    @error('password')
30.                        <span class="invalid-feedback" role="alert">
31.                            <strong>{{ $message }}</strong>
32.                        </span>
33.                    @enderror
34.                </div>
35.                <div class="form-group">
36.                    <div class="custom-control custom-checkbox small">
37.                        <input class="custom-control-
38.                            input" type="checkbox" name="remember" id="remember" {{ old('remember')
39.                                ? 'checked' : '' }}>
40.                        <label class="custom-control-label" for="remember">
41.                            {{ __('Remember Me') }}
42.                        </label>
43.                    </div>
44.                </div>
45.                <button type="submit" class="btn btn-primary btn-user btn-
46.                    block">
47.                    Login
48.                </button>
49.            </form>
50.            <hr>
51.            <div class="text-center">
52.                <a class="small" href="forgot-
53.                    password.html">Forgot Password?</a>
54.            </div>
```

```

46.         <div class="text-center">
47.             <a class="small" href="{{ route('register') }}">Create an Acco
unt!</a>
48.         </div>
49.     </div>
50. </div>
51. </div>
52.
53. @endsection

```

2. Dashboard (Home)

```

1. @extends('layout.app')
2. @section('title', 'Dashboard')
3. @section('subtitle', '')
4. @section('content')
5.
6.     <!-- Page Heading -->
7.     <div class="d-sm-flex align-items-center justify-content-between mb-
4">
8.         <h1 class="h3 mb-0 text-gray-800">@yield('title')</h1>
9.     </div>
10.
11.     <!-- Content Row -->
12.
13.     <div class="row">
14.
15.         <!-- Earnings (Monthly) Card Example -->
16.         <div class="col-xl-3 col-md-6 mb-4">
17.             <div class="card border-left-primary shadow h-100 py-2">
18.                 <div class="card-body">
19.                     <div class="row no-gutters align-items-center">
20.                         <div class="col mr-2">
21.                             <div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Jumlah Data Kata</div>
22.                             <div class="h5 mb-0 font-weight-bold text-gray-800">
23.                                 {{ $jumKata }} Kata
24.
25.                         </div>
26.                         <a href="{{ route('show.kata') }}" class="btn btn-
sm btn-primary">Selengkapnya..</a>
27.                     </div>
28.                     <div class="col-auto">
29.                         <i class="fas fa-book fa-2x text-gray-300"></i>
30.                         {{-- <i class="fas fa-calendar fa-2x text-gray-
300"></i> --}}
31.                     </div>
32.                 </div>
33.             </div>
34.         </div>
35.     </div>
36.
37. </div>
38.
39.
40. @endsection

```



```

47.         </tr>
48.     </thead>
49. <tfoot>
50.     <tr>
51.         <th>No</th>
52.         <th>Kata</th>
53.         <th>Arti</th>
54.         <th>Action</th>
55.     </tr>
56. </tfoot>
57. <tbody>
58.     @php
59.         $no=1;
60.     @endphp
61.     @foreach ($data as $data)
62.     <tr>
63.         <td>{{ $no++ }}</td>
64.         <td>{{ $data->kata1 }}</td>
65.         <td>{{ $data->kata2 }}</td>
66.         <td>
67.             <a href="{{ url('edit.kata.'.$data-
>id) }}" class="btn btn-primary">Edit</a>
68.             <a href="{{ url('delete.kata.'.$data-
>id) }}" onclick="return confirm('Yakin Akan menghapus data?')" class="b
tn btn-danger">Delete</a>
69.         </td>
70.     </tr>
71. @endforeach
72.
73.     </tbody>
74. </table>
75. </div>
76.
77. </div>
78. </div>
79. </div>
80.
81. </div>
82.
83. <!-- Modal -->
84. <div class="modal fade" id="importExcel" tabindex="-
1" role="dialog" aria-labelledby="exampleModallabel" aria-
hidden="true">
85.     <div class="modal-dialog">
86.         <div class="modal-content">
87.             <div class="modal-header">
88.                 <h5 class="modal-
title" id="exampleModallabel">Import File Excel</h5>
89.                 <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
90.                     <span aria-hidden="true">x</span>
91.                 </button>
92.             </div>
93.             <div class="modal-body">
94.                 <form method="POST" action="{{ route('import.kata') }}" enctype=
"multipart/form-data">
95.                     @csrf
96.                     <div class="form-group">
97.                         <label for="exampleFormControlFile1">Data Excel</label>
98.                         <input type="file" class="form-control-
file" id="exampleFormControlFile1" name="data">

```

```
99.         </div>
100.
101.         </div>
102.         <div class="modal-footer">
103.             <button type="submit" class="btn btn-sm btn-
primary">Import Excel</button>
104.         </div>
105.     </form>
106. </div>
107. </div>
108. </div>
109.
110. @endsection
```

Implementasi Algoritma Boyer Moore Pada Kamas Indonesia Gorontalo

by Ishak Heluwa T3116050

Submission date: 14-Jul-2020 10:22AM (UTC+0700)

Submission ID: 1357259074

File name: mentasi_Algoritma_Boyer_Moore_Pada_Kamas_Indonesia_Gorontalo.pdf (2.08M)

Word count: 9160

Character count: 51976

Implementasi Algoritma Boyer Moore Pada Kamas Indonesia Gorontalo

ORIGINALITY REPORT

35%

SIMILARITY INDEX

33%

INTERNET SOURCES

13%

PUBLICATIONS

28%

STUDENT PAPERS

PRIMARY SOURCES

1

ojs.unimal.ac.id

Internet Source

4%

2

www.scribd.com

Internet Source

3%

3

etheses.uin-malang.ac.id

Internet Source

3%

4

webmail.informatika.org

Internet Source

2%

5

[Submitted to Universitas Brawijaya](#)

Student Paper

2%

6

pendidikan.kampung-media.com

Internet Source

2%

7

core.ac.uk

Internet Source

2%

8

amir-saribudin.blogspot.com

Internet Source

2%

9

pt.scribd.com

Internet Source

1%

10

Submitted to LL Dikti IX Turnitin Consortium

Student Paper

1%

11

L.J. LEWIS. "PROBLEMS OF ADAPTATION",
Elsevier BV, 1965

Publication

1%

12

blogpost-tugas.blogspot.com

Internet Source

1%

13

sttgarut.ac.id

Internet Source

1%

14

eprints.dinus.ac.id

Internet Source

1%

15

id.scribd.com

Internet Source

1%

16

ejournal.kopertis10.or.id

Internet Source

1%

17

www.fikom-unisan.ac.id

Internet Source

1%

18

widuri.raharja.info

Internet Source

1%

19

jurnal.unismabekasi.ac.id

Internet Source

1%

Submitted to Universitas Muria Kudus

20

Student Paper

1 %

21

teknosi.fti.unand.ac.id

Internet Source

1 %

22

ejournal.unib.ac.id

Internet Source

<1 %

23

Submitted to iGroup

Student Paper

<1 %

24

Submitted to Universitas Krisnadwipayana -
Faculty of Administration

Student Paper

<1 %

25

Submitted to UIN Sultan Syarif Kasim Riau

Student Paper

<1 %

26

ejournal.catursekti.ac.id

Internet Source

<1 %

27

armanfauzi378.blogspot.com

Internet Source

<1 %

28

media.neliti.com

Internet Source

<1 %

29

www.slideshare.net

Internet Source

<1 %

30

Submitted to Sriwijaya University

Student Paper

<1 %

fr.scribd.com

31

Internet Source

<1 %

32

repository.amikom.ac.id

Internet Source

<1 %

33

Reva Ragam Santika, Auliatunnisa Kamila, M. Ikhsan Abdillah, Samuel Hansen. "Penerapan Metode Profile Matching Dalam Sistem Penunjang Keputusan Pemilihan Karyawan Terbaik Pada Lembaga Penyiaran Publik Televisi Republik Indonesia", INOVTEK Polbeng - Seri Informatika, 2020

Publication

<1 %

34

id.123dok.com

Internet Source

<1 %

Exclude quotes On

Exclude matches < 25 words

Exclude bibliography On



**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS ICHSAN GORONTALO**

FAKULTAS ILMU KOMPUTER

SK MENDIKNAS NOMOR 84/D/O/2001

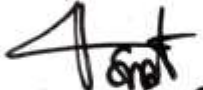
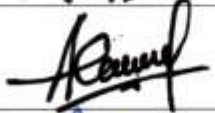
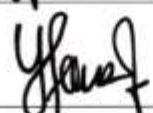
JL. Achmad Nadjamuddin No. 17 Telp. (0435) 829975 Fax (0435) 829976 Gorontalo

Berita Acara Perbaikan/Revisi Ujian

Pada hari ini, Kamis 23-Juli-2020, Pukul 10.00-12.00 Wita. Telah dilaksanakan Ujian mahasiswa Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.

Nama : Ishak Heluwa
Nim : T3116050
Pembimbing I : Moh. Efendi Lasulika, M.Kom
Pembimbing II : Yulianty Lasena, M.Kom
Judul : Implementasi Algoritma Boyer Moore Untuk Kamus Bahasa Indonesia Gorontalo

Oleh Komite Seminar sebagai berikut :

No	Komite Seminar	Status	Tanda Tangan
1	Asmaul Husna N, M.Kom	Ketua	
2	Apriyanto Alhamad, M.Kom	Anggota	
3	Maryam Hasan, M.Kom	Anggota	
4	Moh. Efendi Lasulika, M.Kom	Anggota	
5	Yulianty Lasena, M.Kom	Anggota	

DAFTAR RIWAYAT HIDUP



Nama : Ishak Heluwa
Tempat, Tgl Lahir : Tamboo, 23 Maret 1994
Pekerjaan : Karyawan Swasta
Email : ishakheluwa46@gmail.com

Riwayat Pendidikan:

1. Pada Tahun 2002, Telah Menyelesaikan Pendidikan Taman Kanak-Kanak (TK) Islam Kreatif di desa Tamboo, Kecamatan Bone Pantai, Kabupaten Bone Bolango, Kota Gorontalo
2. Pada Tahun 2008, Telah Menyelesaikan Pendidikan di Sekolah Dasar (SD) Inpres Tamboo dan sekarang telah menjadi Sekolah Dasar (SD) Negeri 9 Bone Pantai, Kecamatan Bone Pantai, Kabupaten Bone Bolango, Kota Gorontalo
3. Pada Tahun 2011, Telah Menyelesaikan Pendidikan di Sekolah Menengah Pertama (SMP) Negeri 1 Bonepantai, Kecamatan Bone Pantai, Kabupaten Bone Bolango, Kota Gorontalo
4. Pada Tahun 2014, Telah menyelesaikan Pendidikan di Sekolah Menengah Kejuruan (SMK) Negeri 1 Bone Pantai, Kecamatan Bone Pantai, Kabupaten Bone Bolango, Kota Gorontalo
5. Pada Tahun 2016, Telah diterima menjadi Mahasiswa di Perguruan Tinggi Swasta Universitas Ichsan Gorontalo, Kecamatan Dulalowo Timur, Kota Gorontalo