**PENERAPAN ALGORITMA *BOYER MOORE* PADA APLIKASI *FONT ITALIC* UNTUK PENDETEKSI BAHASA LATIN**

**S K R I P S I**

Jurusan Informatika

Program Studi Sarjana Informatika

Oleh:

**Mia Helmalia Haldy**

NIM D1041141061



**FAKULTAS TEKNIK**

**UNIVERSITAS TANJUNGPURA**

**PONTIANAK**

**2021**

# HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Mia Helmalia Haldy

NIM : D1041141061

Menyatakan bahwa skripsi yang berjudul ”Penerapan Algoritma *Boyer Moore* pada Aplikasi *Font Italic* Untuk Pendeteksi Bahasa Latin” tidak terdapat karya yang pernah diajukan untuk memperoleh gelar sarjana di suatu perguruan tinggi manapun. Sepanjang sepengetahuan saya, tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam Daftar Pustaka.

Demikian pernyataan ini dibuat dengan sebenar-benarnya. Saya sanggup menerima konsekuensi akademis dan hukum dikemudian hari apabila penyataan yang dibuat ini tidak benar.

Pontianak,15 Juli 2021

Mia Helmalia Haldy

NIM D1041141061

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,

RISET, DAN TEKNOLOGI

UNIVERSITAS TANJUNGPURA

**FAKULTAS TEKNIK**

Jalan Prof. Dr. H. Hadari Nawawi Pontianak 78124

Telepon (0561) 740186 Faximili (0561) 740186

Email : [ft@untan.ac.id](mailto:ft@untan.ac.id) Website : [http://teknik.untan.ac.id](http://teknik.untan.ac.id/)



# HALAMAN PENGESAHAN

PENERAPAN ALGORITMA *BOYER MOORE* PADA APLIKASI *FONT ITALIC* UNTUK PENDETEKSI BAHASA LATIN

Jurusan Informatika

Program Studi Sarjana Informatika

Oleh :

Mia Helmalia Haldy

NIM. D1041141061

Telah dipertahankan di depan Penguji Skripsi pada tanggal 15 juli 2021 dalam sidang secara daring (*online*) dan diterima sebagai salah satu persyaratan untuk memperoleh gelar sarjana.

Susunan Penguji Skripsi :

|  |  |  |
| --- | --- | --- |
| Dosen Pembimbing Utama | : Dr. Herry Sujaini, S.T., M.T. | (NIP.196806291997021001) |
| Dosen Pembimbing Kedua | : Tursina, S.T., M.Cs. | (NIP.197801152002122003) |
| Dosen Penguji Utama | : Dr. Arif Bijaksana P.N. S.T., M.T. | (NIP.197208081998021002) |
| Dosen Penguji Kedua | : Rudy Dwi Nyoto, S.T., M.Eng. | (NIP.197803302005011002) |
|  |  |  |

Wakil Dekan Bidang Akademik,

Dr.-Ing. Ir. Slamet Widodo, M.T.,IPM.  
NIP. 196712231992031002

Pontianak, 15 Juli 2021

Dekan,

Dr.rer.nat. Ir. R. M. Rustamaji, M.T.,IPU.

NIP. 196801161994031003

Halaman Persembahan

**اارَّحِيم الرَّحْمَنِ اﷲِ بِسْــــــــــــــــــمِ**

Skripsi ini saya persembahkan kepada dua orang hebat dalam hidup saya mamak saya **Halimah** dan bapak saya **Edi Hendra,**serta adik saya **Bella Maulia Haldy**

yang telah mengisi dunia saya dengan begitu banyak kebahagiaan. Terimakasih untuk semua cinta yang telah kalian berikan, maaf tidak bisa memberikan skripsi ini lebih cepat.

Skripsi ini juga saya persembahkan untuk suami saya **Raden** **Sandy Widyatama** dan anak saya **Raden** **Azril Atharrayhan**, yang telah mendukung, menemani, dan menguatkan saya dalam pengerjaan skripsi ini.

Tak lupa saya persembahkan skripsi ini untuk Emon, Nuy, Febri, Alan, Jejep dan kawan-kawan informatika 2014, terimakasih atas segala bantuan pengerjaan skripsi ini, serta segala pelajaran dan kenangan yang telah diberikan selama ini.

Terakhir, saya persembahkan skripsi ini untuk diri saya sendiri. Terimakasih masih memilih untuk tetap berjuang dan tidak menyerah.

**Terimakasih!**

**الْعَالَمِينَ رَبِّ لِلَّهِ الْحَمْدُ**

# KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah Subhanaahu Wa Taálaa atas berkat rahmat-Nya sehingga penelitian dan penulisan skripsi dengan judul “Penerapan Algoritma *Boyer Moore* pada Aplikasi *Font Italic* Untuk Pendeteksi Bahasa Latin” ini dapat diselesaikan.

Melalui penelitian ini, ucapan terimakasih saya sampaikan kepada pihak-pihak yang telah membantu saya baik berupa materil maupun dukungan moril yaitu kepada Bapak **Dr. Herry Sujaini, S.T., M.T.**, selaku Dosen Pembimbing Pertama, yang telah memberikan banyak bimbingan, saran dan masukan yang membangun dalam pembuatan skripsi ini, dan kepada Ibu **Tursina, S.T., M.Cs.**, selaku Dosen Pembimbing Kedua sekaligus Dosen Pembimbing Akademik saya yang telah memberikan banyak bimbingan, membantu dalam menyelesaikan penelitian ini. Kepada bapak **Dr. Arif Bijaksana Putra Negara, S.T., M.T.** dan bapak **Rudy Dwi Nyoto, S.T., M.Eng.**, selaku dosen penguji pertama dan dosen penguji pendamping yang telah memberikan saran serta masukkan untuk perbaikan skripsi ini.

Tentunya saya sadari terdapat banyak kekurangan dalam penyusunan skripsi ini, oleh karena itu kritik dan saran saya butuhkan. Semoga skripsi ini bermanfaat bagi perkembangan ilmu pengetahuan.

Pontianak,15 Juli 2021

Penulis,

Mia Helmalia Haldy

# ABSTRAK

Bahasa Latin adalah salah satu dari bahasa kuno Semenanjung Italia, mula-mula dipertuturkan oleh bangsa Latin Italia di wilayah Latium pada zaman Romawi Kuno. Pemakaian bahasa Latin juga banyak terdapat pada karya ilmiah, salah satu pemakaiannya yaitu penamaan nama ilmiah atau ungkapan bahasa Latin. Penulisan sebuah karya ilmiah jika menurut pada aturan penulisan Bahasa Indonesia PEUBI (Pedoman Umum Bahasa Indonesia) yang baik dan benar maka setiap bahasa asing (bahasa Latin) harus dimiringkan. Metode pencocokan string dengan *Boyer Moore* sebagai algoritmanya dapat mencari pada *database* data secara cepat sangat cocok untuk mempermudah mencetak miring kata atau istilah bahasa latin. Pengujian dengan 25 kalimat Latin memberikan tingkat Ketepatan menemukan kata Latin menggunakan algoritma *boyer moore* sebesar 0,981 dengan rata-rata waktu pendeteksian 0.11 kata perdetik . Penelitian ini bisa dikatakan berhasil sesuai dengan tujuan yaitu menghasilkan sebuah aplikasi yang dapat mengimplementasikan algoritma *boyer moore* agar dapat mendeteksi kata berbahasa Latin yang kemudian mencetak miring kata bahasa Latin tersebut.

Kata kunci: Pendeteksi Kata bahasa Latin, mencetak miring kata ,*Boyer moore,*Karya Ilmiah

# ABSTRACT

Latin is one of the ancient languages ​​of the Italian Peninsula, first spoken by Italian Latins in the Latium region in Ancient Rome. The use of Latin is also widely found in scientific works, one of which is the naming of scientific names or Latin expressions. The writing of a scientific paper if according to the rules of writing Indonesian PEUBI (General Guidelines for Indonesian Language) is good and correct, then every foreign language (Latin) must be italicized. The string matching method with Boyer Moore as the algorithm can search the database data quickly, which is very suitable to make it easier to italicize Latin words or terms. Testing with 25 Latin sentences gives an accuracy level of finding Latin words using the *Boyer Moore* algorithm of 0.981 with an average detection time of 0.11 words per second. This research can be said to be successful in accordance with the goal of producing an application that can implement the Boyer Moore algorithm in order to detect Latin words which then italicize the Latin words.

Keyword: Latin word detector, Italiccized word, *Boyer Moore,* scientific works.

**DAFTAR ISI**

[HALAMAN PERNYATAAN.......... ii](#_Toc77318247)

[HALAMAN PENGESAHAN....... iii](#_Toc77318248)

[HALAMAN PERSEMBAHAN...................... iv](#_Toc77318249)

[KATA PENGANTAR.................... v](#_Toc77318250)

[ABSTRAK........................................ vi](#_Toc77318251)

[ABSTRACT..................................... vii](#_Toc77318252)

[DAFTAR ISI................................. viii](#_Toc77318253)

[DAFTAR TABEL................................... xi](#_Toc77318254)

[DAFTAR GAMBAR.............................. xiii](#_Toc77318255)

DAFTAR LAMPIRAN..........................................................................................xv

BAB I [PENDAHULUAN......................... 1](#_Toc77318256)

[1.1 Latar Belakang 1](#_Toc77318257)

[1.2 Rumusan Masalah 2](#_Toc77318258)

[1.3 Tujuan Penelitian 2](#_Toc77318259)

[1.4 Pembatasan Masalah 2](#_Toc77318260)

[1.5 Sistematika Penulisan Skripsi 3](#_Toc77318261)

BAB II [TINJAUAN PUSTAKA................... 4](#_Toc77318262)

[2.1 Kajian Terkait 4](#_Toc77318263)

[2.2 *Natural Language Processing* (NLP) 6](#_Toc77318264)

[2.3 Bahasa Latin 6](#_Toc77318265)

[2.4 Pedoman Umum Ejaan Bahasa Indonesia (PUEBI) 7](#_Toc77318266)

[2.4.1 Huruf Miring 7](#_Toc77318267)

[2.5 Pencocokan String 9](#_Toc77318268)

[2.6 *Boyer Moore* 9](#_Toc77318269)

[2.7 Alat Bantu Perancangan 9](#_Toc77318270)

[2.7.1 *Flowchart* 9](#_Toc77318271)

[2.7.2 *Data Flow Diagram(DFD)* 11](#_Toc77318272)

[2.7.3 *Entity Relationship Diagram (ERD)* 14](#_Toc77318273)

[2.8 *Python* 15](#_Toc77318274)

[2.9 Pengujian Perangkat Lunak 15](#_Toc77318275)

[2.8.1 *Black-Box* 16](#_Toc77318276)

[2.8.2 *Precision* dan *Recall* 16](#_Toc77318277)

[2.8.3 *F-Measure* 18](#_Toc77318278)

[2.8.4 Pengujian Durasi pada deteksi kata Latin 18](#_Toc77318279)

BAB III [METODOLOGI PENELITIAN 19](#_Toc77318280)

[3.1 Spesifikasi Perangkat Keras dan Perangkat Lunak 19](#_Toc77318281)

[3.1.1 Perangkat keras 19](#_Toc77318282)

[3.1.2 Perangkat lunak 19](#_Toc77318283)

[3.2 Metode Penelitian 20](#_Toc77318284)

[3.2.1 Analisis Masalah 20](#_Toc77318285)

[3.2.2 Pengumpulan Data 21](#_Toc77318286)

[3.2.3 Desain dan Perancangan Sistem 21](#_Toc77318287)

[3.2.4 Implementasi Algoritma *Boyer Moore* 33](#_Toc77318288)

[3.2.5 Pengujian Sistem 36](#_Toc77318289)

[3.2.6 Analisis Hasil Pengujian 42](#_Toc77318290)

[3.2.7 Penarikan Kesimpulan 42](#_Toc77318291)

BAB IV [IMPLEMENTASI DAN HASIL PENGUJIAN 43](#_Toc77318292)

[4.1 Implementasi 43](#_Toc77318293)

[4.1.1 Antarmuka Halaman Masuk 43](#_Toc77318294)

[4.1.2 Tampilan Halaman Dashboard Admin 44](#_Toc77318295)

[4.1.3 Tampilan Halaman Tambah Akun 44](#_Toc77318296)

[4.1.4 Tampilan Halaman Perbarui Data Akun 44](#_Toc77318297)

[4.1.5 Tampilan Halaman Tambah Kosa Kata Bahasa Latin 45](#_Toc77318298)

[4.1.6 Tampilan Halaman perbarui Data Kosa Kata Bahasa Latin 45](#_Toc77318299)

[4.1.7 Tampilan Halaman Dashboard Mahasiswa 46](#_Toc77318300)

[4.1.8 Tampilan Halaman Tambah Dokumen 46](#_Toc77318301)

[4.1.9 Tampilan Halaman Perbarui Dokumen 47](#_Toc77318302)

[4.1.10 Tampilan Halaman Hasil Pemeriksaan Dokumen 47](#_Toc77318303)

[4.2 Hasil Pengujian 48](#_Toc77318304)

[4.2.1 Pengujian *Black Box* 48](#_Toc77318305)

[4.2.1.1pengujian halaman Login 48](#_Toc77318306)

[4.2.1.2 pengujian halaman Tambah Akun pengguna 49](#_Toc77318307)

[4.2.1.3 pengujian halaman hapus akun pengguna 49](#_Toc77318308)

[4.2.1.4 pengujian halaman tambah dokumen 49](#_Toc77318309)

[4.2.1.5 pengujian halaman ubah dokumen 49](#_Toc77318310)

[4.2.1.6 pengujian halaman download dokumen 50](#_Toc77318311)

[4.2.1.7 pengujian halaman detail pemeriksaan dokumen 50](#_Toc77318312)

[4.2.1.8 pengujian halaman hapus dokumen 50](#_Toc77318313)

[4.2.1.9 pengujian halaman tambah kosa kata 50](#_Toc77318314)

[4.2.1.10 pengujian halaman ubah kosa kata 50](#_Toc77318315)

[4.2.1.11 pengujian halaman hapus kosa kata 51](#_Toc77318316)

[4.2.1.12 pengujian halaman logout 51](#_Toc77318317)

[4.2.2 Pengujian *Precision*, *Recall* dan *F-Measure* 51](#_Toc77318318)

[4.2.3 Pengujian Durasi pada deteksi kata Latin 52](#_Toc77318319)

[4.3 Analisis Hasil Pengujian 55](#_Toc77318320)

BAB V [KESIMPULAN DAN SARAN............ 57](#_Toc77318321)

[5.1 Kesimpulan 57](#_Toc77318322)

[5.2 Saran 57](#_Toc77318323)

[DAFTAR PUSTAKA.......................... 58](#_Toc77318324)

[LAMPIRAN A TABEL PERBANDINGAN A-1](#_Toc77318325)

aftar Tabel

[Tabel 2. 1Kajian Penelitian Terkait 5](#_Toc77323374)

[Tabel 2. 2 Simbol Flowchart 10](#_Toc77323375)

[Tabel 2. 3 simbol-simbol Data flow Diagram 12](#_Toc77323376)

[Tabel 2. 4 Simbol-simbol Entity Relationship Diagram Chen 14](#_Toc77323377)

[Tabel 2. 5 Variabel untuk Perhitungan *Precision* dan *Recall* 17](#_Toc77323378)

[Tabel 3. 1 Spesifikasi Tabel Users.………………………………………………29](#_Toc77323930)

[Tabel 3. 2 Spesifikasi Tabel Kata 29](#_Toc77323931)

[Tabel 3. 3 Spesifikasi Tabel Dokumen 29](#_Toc77323932)

[Tabel 3. 4 Spesifikasi Tabel Media 30](#_Toc77323933)

[Tabel 3. 5 Contoh cara kerja Algoritma *Boyer Moore* 36](#_Toc77323934)

[Tabel 3. 6 Skenario Pengujian Halaman *Login* 37](#_Toc77323935)

[Tabel 3. 7 skenario pengujian halaman tambah akun pengguna 37](#_Toc77323936)

[Tabel 3. 8 skenario pengujian halaman hapus akun pengguna 38](#_Toc77323937)

[Tabel 3. 9 skenario pengujian halama tambah dokumen 38](#_Toc77323938)

[Tabel 3. 10 skenario pengujian halaman ubah dokumen 38](#_Toc77323939)

[Tabel 3. 11 skenario pengujian halaman download dokumen 39](#_Toc77323940)

[Tabel 3. 12 skenario pengujian halaman detail pemeriksaan dokumen 39](#_Toc77323941)

[Tabel 3. 13 skenario pengujian halaman hapus dokumen 39](#_Toc77323942)

[Tabel 3. 14 skenario pengujian halaman tambah kosa kata 39](#_Toc77323943)

[Tabel 3. 15 skenario penguijian halaman ubah kosa kata 40](#_Toc77323944)

[Tabel 3. 16 skenario pengujian halman hapus kosa kata 40](#_Toc77323945)

[Tabel 3. 17 skenario pengujian halaman logout 40](#_Toc77323946)

[Tabel 3. 18 Pengujian Durasi pada Deteksi Kata Latin 41](#_Toc77323947)

[Tabel 4. 1 Hasil pengujian halaman Login………………………………………48](#_Toc77323953)

[Tabel 4. 2 Hasil pengujian Tambah Akun pengguna 49](#_Toc77323954)

[Tabel 4. 3 Hasil pengujian halaman hapus akun pengguna 49](#_Toc77323955)

[Tabel 4. 4 pengujian halaman tambah dokumen 49](#_Toc77323956)

[Tabel 4. 5 Hasil pengujian halaman ubah dokumen 49](#_Toc77323957)

[Tabel 4. 6 Hasil pengujian halaman download dokumen 50](#_Toc77323958)

[Tabel 4. 7 Hasil pengujian halaman detail pemeriksaan dokumen 50](#_Toc77323959)

[Tabel 4. 8 Hasil pengujian halaman hapus dokumen 50](#_Toc77323960)

[Tabel 4. 9 Hasil pengujian halaman tambah kosa kata 50](#_Toc77323961)

[Tabel 4. 10 Hasil pengujian halaman ubah kosa kata 50](#_Toc77323962)

[Tabel 4. 11 Hasil pengujian halaman hapus kosa kata 51](#_Toc77323963)

[Tabel 4. 12 Hasil pengujian halaman logout 51](#_Toc77323964)

[Tabel 4. 13 Hasil Pengujian *Precision,Recall, dan f-measure* 51](#_Toc77323965)

[Tabel 4. 14 Hasil Pengujian Durasi pada Deteksi Kata Latin 53](#_Toc77323966)

Daftar Gambar

[Gambar 3. 1 Diagram Alir Penelitian……………………...……………………..20](#_Toc77323620)

[Gambar 3. 2Arsitektur Sistem…………….. 22](#_Toc77323621)

[Gambar 3. 3Diagram Alir Sistem…………. 23](#_Toc77323622)

[Gambar 3. 4 Diagram Konteks Sistem 24](#_Toc77323623)

[Gambar 3. 5 Diagram Overview Sistem 24](#_Toc77323624)

[Gambar 3. 6 Diagram Rinci 1.0 25](#_Toc77323625)

[Gambar 3. 7 Diagram Rinci 2.0 26](#_Toc77323626)

[Gambar 3. 8 Diagram Rinci 3.0 27](#_Toc77323627)

[Gambar 3. 9 Diagram Rinci 4.0 27](#_Toc77323628)

[Gambar 3. 10 *Entity Relationship Diagram(ERD)* 28](#_Toc77323629)

[Gambar 3. 11 Hubungan Antar Tabel 31](#_Toc77323630)

[Gambar 3. 12Rancangan Antarmuka Halaman Login 31](#_Toc77323631)

[Gambar 3. 13Rancangan Antarmuka Halaman Dashboard Admin 32](#_Toc77323632)

[Gambar 3. 14Rancang Antarmuka Halaman Kosa Kata 32](#_Toc77323633)

[Gambar 3. 15 Rancangan Antarmuka Halaman Dokumen 33](#_Toc77323634)

[Gambar 3. 16 Contoh kasus 1 34](#_Toc77323635)

[Gambar 3. 17 Contoh Kasus 2 34](#_Toc77323636)

[Gambar 3. 18 Contoh kasus 3 35](#_Toc77323637)

[Gambar 4. 1Tampilan Halaman Masuk……………….…………………………43](#_Toc77323785)

[Gambar 4. 2Tampilan Halaman Dashboard Admin 44](#_Toc77323786)

[Gambar 4. 3 Halaman Tambah Akun 44](#_Toc77323787)

[Gambar 4. 4 Halaman Ubah Data 45](#_Toc77323788)

[Gambar 4. 5 Halaman Tambah Kosa Kata Bahasa Latin 45](#_Toc77323789)

[Gambar 4. 6 Halaman perbarui Data Kosa Kata Bahasa Latin 46](#_Toc77323790)

[Gambar 4.7 Tampilan Halaman Dashboard Mahasiswa 46](#_Toc77323791)

[Gambar 4. 8 Tampilan Halaman Tambah Dokumen 47](#_Toc77323792)

[Gambar 4. 9 Tampilan Halaman Perbarui Dokumen 47](#_Toc77323793)

[Gambar 4. 10 Tampilan Halaman Hasil Pemeriksaan Dokumen 48](#_Toc77323794)

[Gambar 4.11 hasil pengujian 53](#_Toc77323795)

[Gambar 4.12 hasil pengujian 1 54](#_Toc77323796)

[Gambar 4. 13 hasil pengujian 2 54](#_Toc77323797)

[Gambar 4. 14 hasil pengujian 3 55](#_Toc77323798)

Gambar 4. 15 Hasil pengujian Penulisan Yang seharusnya tidak miring………..56

**DAFTAR LAMPIRAN**

Lampiran A Tabel Perbandingan.........................................................................A-1

**BAB I**

# PENDAHULUAN

* 1. **Latar Belakang**

Bahasa Latin adalah salah satu dari bahasa kuno Semenanjung Italia, mula-mula dipertuturkan oleh bangsa Latin Italia di wilayah Latium pada zaman Romawi Kuno. Seperti sebagian besar bahasa-bahasa Eropa, bahasa Latin juga merupakan turunan dari bahasa ProtoIndo-Eropa purba. Dipengaruhi bahasa Etruska dan menggunakan abjad Yunani sebagai dasarnya, bahasa Latin ini menjelma menjadi apa yang sekarang dikenal bahasa Latin di Semenanjung Italia. Pemakaian bahasa Latin juga banyak terdapat pada karya ilmiah, salah satu pemakaiannya yaitu penamaan nama ilmiah atau ungkapan bahasa Latin. Penulisan sebuah karya ilmiah jika menurut pada aturan penulisan Bahasa Indonesia PEUBI (Pedoman Umum Bahasa Indonesia) yang baik dan benar maka setiap bahasa asing (bahasa Latin) harus dimiringkan.

Penulisan karya ilmiah biasanya menggunakan aplikasi pengolah kata Microsoft word. Pada Microsoft word untuk membuat kata tercetak miring harus menggunakan *tools* yang ada di *toolbar*, hal ini berakibat memperlambat waktu dalam penulisan jika harus mengubah satu per satu kata-kata berbahasa asing (bahasa Latin) tersebut menjadi cetak miring dan memiliki kemungkinan terlewatnya kata atau istilah asing (bahasa Latin) yang harus tercetak miring. Seiring dengan perkembangan teknologi, hal ini dapat menjadi sebuah alternatif dan inovasi, dalam mempermudah mencetak miring kata atau istilah bahasa Latin pada pengerjaan penulisan karya ilmiah, salah satunya yaitu dengan penyediaan aplikasi.

Aplikasi yang disediakan dapat menerapkan metode pencocokan string dengan algoritma boyer moore. Menurut Efendi (2012) ide utama dari algoritma boyer moore adalah dengan melakukan pencocokan dari paling kanan string yang dicari. Dengan menggunakan algoritma ini, secara rata-rata proses pencarian akan lebih cepat dibandingkan dengan proses pencarian lainnya. Penerapan algoritma boyer moore untuk menemukan kata atau istilah bahasa Latin dan kemudian otomatis mencetak miring kata tersebut, dapat mengurangi kekeliruan dalam memiringkan kata atau istilah dalam bahasa Latin, dan dapat dijadikan sebagai metode untuk di implementasikan ke dalam sistem.

Berdasarkan uraian tersebut maka dibutuhkan aplikasi untuk mencetak miring sebuah kata berbahasa Latin secara otomatis, yang dapat memanfaatkan metode pencocokan string dengan Boyer Moore sebagai algoritmanya, penelitian ini diharapkan dapat mengurangi kekeliruan dan terlewatnya mencetak miring kata berbahasa Latin pada karya ilmiah, sehingga dapat menghasilkan karya ilmiah yang sesuai dengan kaidah bahasa Indonesia yang baik dan benar.

* 1. **Rumusan Masalah**

Adapun hal yang dapat dijadikan sebagai rumusan masalah dari penelitian ini adalah bagaimana menerapkan algoritma boyer moore pada aplikasi *font italic* sebagai pendeteksi bahasa Latin.

* 1. **Tujuan Penelitian**

Berdasarkan latar belakang dan rumusan masalah yang di bahas diatas, maka tujuan dari penelitian ini adalah untuk menghasilkan aplikasi yang mampu mengimplementasikan algoritma boyer moore agar dapat mendeteksi Bahasa Latin dalam suatu kalimat Bahasa Indonesia, untuk kemudian dicetak miring.

* 1. **Pembatasan Masalah**

Beberapa hal yang menjadi batasan dalam penelitian ini adalah sebagai berikut.

1. Data yang akan diproses dalam sistem hanya berupa dokumen Microsoft Word (.docx).
2. Sistem dibangun berbasis web.
3. Memiringkan kata hanya berfokus pada kata bahasa Latin.
4. Sistem tidak memperbaiki kesalahan penulisan pada kata bahasa Latin.
   1. **Sistematika Penulisan Skripsi**

Sistematika penulisan penelitian ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan. Sistematika laporan tugas akhir ini disusun dalam 5 (lima) bab yang terdiri dari Bab I Pendahuluan, Bab II Tinjauan Pustaka, Bab III Metodologi Penelitian dan Perancangan Sistem, Bab IV Hasil Perancangan dan Analisis Sistem, serta Bab V Penutup.

**Bab I Pendahuluan** adalah bab yang berisi latar belakang penelitian, perumusan masalah, tujuan penelitian, pembatasan masalah, dan sistematika penulisan.

**Bab II Tinjauan Pustaka** adalah bab yang berisi uraian sistematis tentang hasil-hasil penelitian yang didapat oleh peneliti terdahulu dan landasan teori yang ada hubungannya dengan penelitian yang akan dilakukan berupa informasi mengenai NLP (*Natural Language Prosecessing*), huruf miring, Bahasa Latin, Algoritma Boyer Moore, PEUBI, pencocokan string, dan metode pengujian yang akan digunakan.

**Bab III Metodologi Penelitian** adalah bab yang berisi tentang bahan penelitian, alat yang dipergunakan, metode penelitian yang berisi diagram alir penelitian, perancangan sistem, perancangan antarmuka serta rencana pengujian dengan menggunakan pengujian akurasi, pengujian *precision*, *recall* dan *f-measure*, dan pengujian *black box*.

**Bab IV Hasil dan Analisis** adalah bab yang berisi penjelasan mengenai implementasi pada sistem, *screenshoot* tampilan antarmuka sistem yang sudah jadi, serta analisis hasil uji coba. Setiap bagian sistem yang ditampilkan akan dilakukan analisis terlebih dahulu untuk mengarah kepada suatu kesimpulan.

**Bab V Penutup** adalah bab yang berisi kesimpulan dari penelitian yang telah dilakukan dan saran/ rekomendasi untuk perbaikan, pengembangan atau kesempurnaan/ kelengkapan penelitian yang telah dilakukan.

**BAB II**

# TINJAUAN PUSTAKA

* 1. **Kajian Terkait**

Beberapa penelitian terkait dengan penelitian ini diantaranya sebagai berikut.

Dhea Dwi Putri (2019), dalam penelitiannya yang berjudul “Implementasi *Fuzzy String Matching* pada Aplikasi *Font Italic* untuk Deteksi Kata Bahasa Inggris” menjelaskan bahwa penelitian ini menggunakan metode *fuzzy string matching* dalam melakukan pencocokan *string* pada teks dokumen dengan menggunakan algoritma *Jaro-Winkler Distance* untuk menghitung nilai *distance* (kemiripan kata). Tujuan dari penelitian ini adalah untuk mempermudah pengguna dalam memiringkan kata berbahasa asing khususnya bahasa Inggris. Aplikasi yang dibangun berbasis dekstop.

Ichsan Taufik, dkk. (2017), melakukan penelitian dengan judul “Implementasi *Fuzzy Search* untuk Pendeteksi Kata Asing pada Dokumen Microsoft Word”. Penelitian ini mempermudah pengguna untuk mendetaksi kata bahasa asing dalam sebuah karya ilmiah untuk kemudian memiringkannya. Penelitian ini menggunakan metode *fuzzy search* dengan algoritma *Levenshtein Distance* untuk menghitung nilai distance (kemiripan kata). Percobaaan pengujian dilakukan menggunakan teks yang sedikit, sedang, dan banyak menghasilkan akurasi dengan nilai rata-rata 89,6%.

Rohmat Indra Borman dan Agus Pratama (2016), melakukan penelitian yang berjudul “Penerapan *String Matching* dengan *Algoritma Boyer Moore* pada Aplikasi *Font Italic* untuk deteksi Kata Asing”. Peneitian ini mempermudah pengguna untuk mendeteksi bahasa asing dalam sebuah karya ilmiah untuk kemudian memiringkannya. Penelitian ini menjelaskan Menerapkan algoritma *Boyer Moore* untuk pencocokan teks, kata Asing yang dimiringkan berfokus pada bahasa Inggris, serta aplikasi yang dibangun berbasis dekstop.

Tabel 2. 1Kajian Penelitian Terkait

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Penulis | Judul | Keterangan |
| 1. | Dhea Dwi Putri (2019) | Implementasi *Fuzzy String Matching* pada Aplikasi *Font Italic* untuk Deteksi Kata Bahasa Inggris. | * penelitian ini menggunakan metode *fuzzy string matching* dalam melakukan pencocokan string pada teks dokumen dengan menggunakan algoritma *Jaro-Winkler Distance* untuk menghitung nilai distance (kemiripan kata). * Aplikasi berbasis Desktop. |
| 2. | Ichsan Taufik, Izma Dewi Aishia, Jumadi (2017) | Implementasi *Fuzzy Search* untuk Pendeteksi Kata Asing pada Dokumen *Microsoft Word.* | * Penelitian ini menggunakan algoritma *Levenshtein Distance* untuk menghitung nilai distance (kemiripan kata). * Percobaaan pengujian dilakukan menggunakan teks yang sedikit, sedang, dan banyak menghasilkan akurasi dengan nilai rata-rata 89,6%. |
| 3. | Rohmat Indra Borman, Agus Pratama (2016) | Penerapan *String Matching* dengan Algoritma Boyer Moore pada Aplikasi *Font Italic* untuk Deteksi Kata Asing | * Menerapkan algoritma *Boyer Moore* untuk pencocokan teks. * Kata Asing yang dimiringkan berfokus pada Bahasa Inggris. * Aplikasi berbasis dekstop. |
| 4. | Son Thanh Phan, Thang Tat Vu, dan Mai Chi Luong | *Extracting MFCC, F0 feature in Vietnamese HMM-based speech synthesis* | Sistem dibangun menggunakan *toolkit* HTS dan MLSA filter untuk mensintesa suara. |

* 1. ***Natural Language Processing* (NLP)**

Simulasi *Natural Language Processing* (NLP) atau Pemrosesan Bahasa Alami merupakan salah satu cabang ilmu Kecerdasan Buatan (*Artificial Intelligence*). NLP adalah sebuah bagian dari penelitian dan aplikasi yang mengkaji bagaimana komputer dapat digunakan untuk mengerti dan memanipulasi bahasa alami yang berupa teks atau ucapan untuk hal-hal yang berguna (Nugraha, 2014).

Arman dalam Lisangan (2013) mengemukakan bahwa terdapat beberapa alasan yang menyulitkan NLP, yaitu masalah ambiguity atau makna ganda dan jumlah kosa kata (*vocabulary*) yang besar dan berkembang dari waktu ke waktu. Berdasarkan alasan tersebut, NLP tidak mempedulikan bagaimana suatu kalimat dimasukkan ke dalam komputer tetapi menyalin informasi dari kalimat tersebut.

* 1. **Bahasa Latin**

Bahasa Latin adalah bahasa asli kota Roma dan dataran Latian (Latium). Latin termasuk salah satu cabang bahasa Italis dari bahasa Indo-European, sehingga berhubungan erat dengan bahasa Yunani, Etruria dan Celtic (Susanto, A. E, 2013). Bahasa ini berawal dari dialek bahasa masyarakat yang tinggal di sekitar sungai Tiber sejak sekitar 600 SM.

Meski memiliki nama lokal untuk masing-masing negara, organisme (flora dan fauna) memiliki satu nama yang pasti dikenal. Nama ilmiah flora maupun fauna mayoritas menggunakan bahasa Latin. Penting untuk memberikan nama ilmiah untuk tiap organisme yang eksis di Bumi. Hal ini dinamakan binomial nomenclature. Anda pasti akrab dengan nama-nama ilmiah seperti Homo sapiens, Tyrannosaurus rex, Boa constrictor, atau Geranium arboreum.

Penamaan organisme menggunakan bahasa Latin dimulai oleh ahli fisika dan biologi asal Swedia, Carolus Linnaeus, pada abad ke-18. Alasan Linnaeus menggunakan dua suku kata untuk nama ilmiahnya adalah agar organisme tersebut mudah untuk diidentifikasi. Mengutip situs Sciencing, Senin (25/11/2019), penggunaan nama ilmiah mengesampingkan fakta bahwa organisme memiliki nama berbeda yang diberikan oleh penduduk di sebuah wilayah. Bahasa Latin digunakan karena tidak berasal dan tidak digunakan di negara manapun di dunia. Selain Latin, bahasa Yunani Kuno juga kerap digunakan untuk penamaan organisme. Begitu pula bahasa-bahasa kuno lainnya seperti Aztec, Mongolia, dan Xhosa. Nama ilmiah selalu terdiri dari nama genus pada kata pertama, ditulis kapital pada huruf awal. Kemudian, dilanjutkan dengan nama spesies yang spesifik dan tidak menggunakan capital. Nama ilmiah selalu ditulis miring (jika diketik) dan digaris bawahi (jika ditulis tangan).

* 1. **Pedoman Umum Ejaan Bahasa Indonesia (PUEBI)**

Pedoman Umum Ejaan Bahasa Indonesia (PUEBI) merupakan edisi terbaru dari Ejaan yang Disempurnakan (EYD). Oleh sebab itu, ketika muncul kali pertama, di sampul sudah tertera edisi keempat. Pedoman Umum EBI ini dikeluarkan berdasarkan Peraturan Menteri Pendidikan dan Kebudayaan Republik Indonesia Nomor 50 Tahun 2015 tanggal 26 November 2016. Dengan adanya EBI terbaru ini, maka EYD dinyatakan sudah tidak berlaku lagi. Walaupun begitu, isi dari EBI tidak jauh berbeda dengan EYD edisi tiga. EBI edisi terbaru ini bisa menjadi pedoman dasar bagi semua orang untuk menggunakan bahasa Indonesia secara baik dan benar. Tak dimungkiri, pesatnya perkembangan teknologi informasi, membuat orang lebih leluasa menggunakan bahasa sebagai alat komunikasi. Di sinilah, terkadang orang melakukan kesalahan dalam berbahasa. Karena seringnya penggunaan bahasa yang salah, hal itu sudah menjadi kebiasaan dan dianggap sebagai sebuah kebenaran. Untuk itu, perlu adanya referensi tentang aturan penggunaan bahasa Indonesia dengan baik dan benar, terutama dalam teknis penulisan. PUEBI bisa menjadi rujukan dasar bagi semua lapisan masyarakat tentang dasar aturan penggunaan bahasa Indonesia.

1. **Huruf Miring**

Huruf yang tercetak miring dalam terminologi tipografi disebut italik. Huruf italik ini biasanya digunakan untuk memberikan penekanan pada sebuah kata. Di samping itu, huruf-huruf ini juga dipakai untuk menunjukkan istilah atau kata yang berasal dari bahasa asing. Dalam hal ini huruf bercetak miring pada umumnya dipakai pada pengutipan judul buku, nama koran atau media pers. Selain itu, huruf miring juga biasa digunakan untuk menegaskan kata atau bagian tertentu dalam kalimat atau penulisan kata-kata yang bukan merupakan bahasa Indonesia seperti istilah bahasa Inggris, bahasa ilmiah, dan bahasa daerah. Adapun tata cara penggunaan huruf miring diantaranya yaitu:

1. Huruf miring dipakai untuk menuliskan judul buku, nama majalah, atau nama surat kabar yang dikutip dalam tulisan, termasuk dalam daftar pustaka.

Misalnya:

* Saya sudah membaca buku *Salah Asuhan* karangan Abdoel Moeis.
* Majalah *Poedjangga Baroe* menggelorakan semangat kebangsaan.
* Berita itu muncul dalam surat kabar *Cakrawala*.
* Pusat Bahasa. 2011. *Kamus Besar Bahasa Indonesia Pusat Bahasa*. Edisi Keempat (Cetakan Kedua). Jakarta: Gramedia Pustaka Utama

1. Huruf miring dipakai untuk menegaskan atau mengkhususkan huruf, bagian kata, kata, atau kelompok kata dalam kalimat.

Misalnya:

* Huruf terakhir kata *abad* adalah d.
* Dia tidak *diantar*, tetapi mengantar.
* Dalam bab ini *tidak* dibahas pemakaian tanda baca.
* Buatlah kalimat dengan menggunakan ungkapan *lepas tangan*.

1. Huruf miring dipakai untuk menuliskan kata atau ungkapan dalam bahasa daerah atau bahasa asing.

Misalnya:

* + Upacara *peusijuek* (tepung tawar) menarik perhatian wisatawan asing yang berkunjung ke Aceh.
  + Nama ilmiah buah manggis ialah *Garcinia mangostana.*
  + *Weltanschauung* bermakna 'pandangan dunia'.
  + Ungkapan *bhinneka tunggal ika* dijadikan semboyan negara Indonesia.

Catatan:

(1) Nama diri, seperti nama orang, lembaga, atau organisasi, dalam bahasa asing atau bahasa daerah tidak ditulis dengan huruf miring.

(2) Dalam naskah tulisan tangan atau mesin tik (bukan komputer), bagian yang akan dicetak miring ditandai dengan garis bawah.

(3) Kalimat atau teks berbahasa asing atau berbahasa daerah yang dikutip secara langsung dalam teks berbahasa Indonesia ditulis dengan huruf miring.

* 1. **Pencocokan String**

*String matching* atau Pencocokan string merupakan bagian penting dari sebuah proses pencarian string *(string searching*) dalam sebuah dokumen (Siragih, 2013). Hasil dari pencarian string dalam dokumen tergantung dari teknik atau cara pencocokan string yang digunakan. Pencocokan string secara garis besar dapat dibedakan menjadi dua yaitu sebagai berikut :

1. *Exact String Matching*, merupakan pencocokan string secara tepat dengan susunan karakter dalam string yang dicocokan memiliki jumlah maupun urutan karakter dalam string yang sama.
2. *Inexact String Matcing* atau *fuzzy string matcing* merupakan pencocokan string secara samar, maksudnya pencocokan string dimana string yang dicocokan memiliki kemiripan dimana keduanya memiliki susuan karakter yang berbeda (mungkin jumlah atau urutannya) tetapi string-string tersebut memiliki kemiripan baik kemiripan tekstual / penulisan *(approximate string matching*) atau kemiripan ucapan *(phonetic string matching*).
   1. ***Boyer Moore***

Menurut Edward Rompah pada artikelnya yang membahas tentang algoritma *Boyer-Moore*, algoritma *Boyer-Moore* dipublikasikan oleh Robert S. Boyer, dan J. Strother Moore pada tahun 1977. Ide utama dari algoritma ini adalah dengan melakukan pencocokan dari paling kanan string yang dicari. Dengan menggunakan algoritma ini, secara rata-rata proses pencarian akan lebih cepat dibandingkan dengan proses pencarian lainnya. Ide dibalik algoritma ini adalah bahwa dengan memulai pencocokan karakter dari kanan, dan bukan dari kiri, maka akan lebih banyak informasi yang didapat (Mufthy, 2011).

* 1. **Alat Bantu Perancangan** 
     1. ***Flowchart***

*Flowchart* menggambarkan tahapan proses suatu sistem, program *flowchart* menggambarkan urutan-urutan instruksi dari suatu program komputer, oleh karena itu *flowchart* yang dihasilkan dapat bervariasi antara suatu program dengan yang lainnya (Suyanto, 2004).

*Flowchart* menurut Jogiyanto (2005) adalah bagan yang menunjukkan alir di dalam program atau prosedur sistem secara logika dapat didefinisikan sebagai bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Setiap pengolahan dalam *flowchart* terbagi atas 3 bagian utama, yaitu *input*, proses pengolahan dan *output*. *Flowchart* merupakan gambaran dalam bentuk diagram alir dari algoritma-algoritma dalam suatu program yang menyatakan arah alur program tersebut. Flowchart berupa bagan dengan simbol-simbol tertentu seperti pada Tabel 2.2 yang menggambarkan urutan proses secara merinci dan hubungan antara suatu proses (intruksi) dengan proses lainnya.

**Tabel 2. 2** **S**imbol Flowchart

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Simbol** | **Nama** | **Fungsi** |
| 1. |  | *Terminator* | Awal / Akhir Program |
| 2. |  | Garis Alir *(Flow Line)* | Arah aliran program |
| 3. |  | Proses | Proses pengolahan data |
| 4. |  | *Input/Output* Data | Proses *input/output data* |
| 5. |  | *Decision* | Perbandingan pernyataan dan penyeleksian data |

Flowchart terdiri dari beberapa jenis, yaitu:

a) Bagan Alir Sistem (*System Flowchart*)

*System Flowchart* merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan urutan dari prosedur-prosedur yang ada dalam sistem. Bagan alir sistem ini menunjukkan apa yang dikerjakan di dalam sistem.

b) Bagan Alir Dokumen (*Document Flowchart*)

Bagan Alir dokumen atau bagan alir formulir merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya.

c) Bagan Alir Skematik (*Schematic* *Flowchart*)

*Schematic Flowchart* merupakan bagian alir yang mirip dengan bagan alir sistem, yaitu menggunakan prosedur di dalam sistem. Perbedaannya adalah bagan alir sistematik selain menggunakan simbol-simbol alir sistem, juga menggunakan gambar-gambar komputer dan peralatan lainnya yang digunakan.

d) Bagan Alir Program (*Program Flowchart*)

Bagan alir program merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derivikasi bagan alir sistem. Bagan ini terdiri dari dua macam yaitu bagan alir logika program (*program logic flowchart*) dan bagian alir program komputer tehrinci (*detailed computer program flowchart*).

Bagan alir logika program digunakan untuk menggambarkan tiap-tiap langkah di dalam program komputer secara logika yang dipersiapkan oleh *system analyst*, sedangkan bagan alir program terinci yang dipersiapkan oleh progarammer.

e) Bagan Alir Proses (*Process Flowchart*)

Bagan alir proses merupakan bagan alir yang banyak digunakan di teknik industri. Bagan ini berguna bagi *system anlyst* untuk menggambarakan proses dari suatu prosedur.

* + 1. ***Data Flow Diagram(DFD)***

*Entity relation diagram* (ERD) merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. Menurut Rosa dan Shalahuddin (2013), ERD adalah pemodelan awal basis data yang dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional.

Data Flow Diagram atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengatur dari masukan (input) dan keluaran (output). DFD tidak sesuai untuk memodelkan sistem yang menggunakan pemrograman berorientasi objek (Sukamto & Shalahuddin, 2014).

Berikut merupakan simbol-simbol Data Flow Diagram berdasarkan model yang dibuat oleh Edward Yourdon dan Tom DeMarco :

Tabel 2. 3 simbol-simbol Data flow Diagram

|  |  |  |
| --- | --- | --- |
| **Nama** | **Simbol** | **Keterangan** |
| Proses |  | Menggambarkan proses atau fungsi atau prosedur. Pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang seharusnya menjadi fungsi atau prosedur di dalam kode program |
| Data Store |  | Menggambarkan *data store* atau storage. Pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel basis data. |
| Entitas |  | Menggambarkan suatu entitas luar (external entity) atau masukan (input) atau keluaran (output) atau orang yang memakai atau berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan |
| Aliran Data |  | Menggambarkan suatu aliran data. Yaitu data yang di kirm antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (input) atau keluaran (output) |

Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan DFD menurut Sukamto dan Shalahuddin :

1. Membuat DFD Level 0 atau sering disebut juga Context Diagram DFD Level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD Level 0 digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.

2. Membuat DFD Level 1 DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. DFD Level 1 merupakan hasil breakdown DFD Level 0 yang sebelumnya sudah dibuat.

3. Membuat DFD Level 2 Modul-modul pada DFD Level 1 dapat di breakdown menjadi DFD Level 2. Modul mana saja yang harus di breakdown lebih detail tergantung pada tingkat kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu untuk di breakdown lagi. Untuk sebuah sistem, jumlah DFD Level 2 sama dengan jumlah modul pada DFD Level 1 yang di breakdown.

4. Membuat DFD Level 3 dan seterusnya. DFD Level 3, 4, 5 dan seterusnya merupakan breakdown dari modul pada DFD Level di atasnya. Breakdown pada level 3, 4 dan 5 dan seterusnya aturannya sama persis dengan DFD Level 1 atau Level 2.

* + 1. ***Entity Relationship Diagram (ERD)***

Entity Relationship Diagram (ERD) adalah pemodelan awal basis data yang akan dikembangkan berdasarkan teori himpunan dalam bidang matematika untuk pemodelan basis data relasional. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen). Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow’s Foot, dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen (Sukamto dan Shalahuddin,2014).

Berikut adalah simbol-simbol yang digunakan yang digunakan dalam ERD dengan menggunakan notasi Chen :

Tabel 2. 4 Simbol-simbol Entity Relationship Diagram Chen

|  |  |  |
| --- | --- | --- |
| **Nama** | **Simbol** | **Keterangan** |
| Entitas | nama\_entitas | Entitas merupakan data inti yang akan disimpan. Bakal tabel pada basis data. Benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer. Penamaan entitas biasanya lebih ke kata benda dan bukan merupakan nama tabel. |
| Atribut |  | Field atau kolom data yang butuh disimpan dalam suatu entitas. |
| Atribut Multinilai / *Multivalue* |  | Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu. |
| Relasi | nama\_relasi | Relasi yang menghubungkan antar entitas. Biasanya diawali dengan kata kerja. |
| Asosiasi |  | Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaian. |

* 1. ***Python***

*Python* adalah bahasa pemograman komputer multiguna dan optimal untuk produktifitas, keterbacaan kode, dan kualitas perangkat lunak. *Python* merupakan bahasa pemograman yang open source, portable dan memiliki banyak kelebihan. *Python* merupakan bahasa pemograman untuk tujuan umum (*general purpose*) yang lebih umum diterapkan pada peran scripting. *Python* merupakan bahasa scripting berbasis objek (*object-oriented*). *Python* banyak digunakan karena *Python* memiliki banyak library sehingga mempermudah implementasi pada suatu perancangan perangkat lunak (Mark Lutz, 2013).

* 1. **Pengujian Perangkat Lunak**

Pengujian yang dilakukan dalam penelitian ini sebagai tahap implementasi untuk menguji tingkat minimal kesalahan dan keakuratan perangkat lunak yang dirancang. Adapaun pengujian yang akan digunakan dalam penelitian yaitu pengujian sistem menggunakan pengujian *black box*, *Recall* (Perolehan), *Precision* (ketepatan) , dan *F-Measure*.

1. ***Black-Box***

Metode *Black-Box* yaitu data pengujian dipilih berdasarkan spesifikasi masalah tanpa memperhatikan detail internal dari program (Rosa, 2009), untuk memeriksa apakah program dapat berjalan dengan benar. Pemilihan data pengujian paling tidak harus dipilih dengan cara berikut (Rosa, 2009):

* 1. *Extreme values*, banyak program error pada suatu batas range dari aplikasi.
  2. *Ilegal values*, yaitu suatu data atau nilai yang tidak diperbolehkan maupun data yang tidak berguna.
  3. *values*, yaitu data yang mudah diperiksa.
  4. *Typical realistic value*, yaitu mencoba program dengan data pengujian untuk melihat bagaimana program melakukannya. Data ini harus cukup sederhana sehingga hasilnya dapat dihitung secara manual.

Beberapa teknik pengujian secara *Black box* antara lain sebagai berikut:

* 1. *Requirement Testing* adalah spesifikasi kebutuhan yang terasosiasi dengan perangkat lunak (*input, output, fungsi, performansi)* diidentifikasi pada tahap spesifikasi kebutuhan dan desain. Requirement testing melibatkan pembuatan kasus uji untuk setiap spesifikasi kebutuhan yang terkait dengan program (Rosa, 2009).
  2. *Performance Testing* adalah mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari sisi acuan kebutuhan misalnya: aliran data, ukuran pemakaian memori, kecepatan eksekusi dan lain-lain. Untuk mencari tahu beban kerja atau kondisi konfigurasi program dan dapat digunakan untuk menguji batasan lingkungan program (Rosa, 2009).
  3. *Scenario Testing* adalah pengujian yang realistis, kredibel dan memotivasi stakeholder, tantangan untuk program dan mempermudah pengujian untuk melakukan evaluasi. Pengujian ini menyediakan koombinasi variabel-variabel dan fungsi yang sangat berarti daripada kombinasi buatan yang didapatkan dengan pengujian domain atau desain pengujian kombinasi (Simarmata, 2010).

1. ***Precision* dan Recall**

Rozi (2012) menjelaskan bahwa, *precision* adalah rasio jumlah dokumen relevan yang ditemukan dengan total jumlah dokumen yang ditemukan oleh sistem. *Recall* adalah rasio jumlah dokumen relevan yang ditemukan kembali dengan total jumlah dokumen dalam kumpulan dokumen yang dianggap relevan. R*ecall* (perolehan) berhubungan dengan kemampuan sistem untuk memanggil dokumen yang relevan. Sedangkan *precision* (ketepatan) berkaitan dengan kemampuan sistem untuk tidak memanggil dokumen yang tidak relevan. Berikut tabel variabel yang digunakan pada perhitungan *precision* dan *recall*, dapat dilihat pada Tabel 2.5.

Tabel 2. 5 Variabel untuk Perhitungan *Precision* dan *Recall*

|  |  |  |  |
| --- | --- | --- | --- |
|  | Relevan | Not Relevan | Total |
| Retrieved | A | b | a+b |
| Not Retrieved | C | d | c+d |
| Total | a+c | b+d |  |

Persamaan (2.2) dan (2.3) untuk *precision* dan *recall* sebagai berikut.

*Precision* = =𝑃 𝑟𝑒𝑙𝑒𝑣𝑎𝑛|𝑑𝑖𝑡𝑒𝑚𝑢𝑘𝑎𝑛) (2.2)

*Recall* = = 𝑃 (𝑑𝑖𝑡𝑒𝑚𝑢𝑘𝑎𝑛|𝑟𝑒𝑙𝑒𝑣𝑎𝑛) (2.3)

Kemudian dapat disederhanakan ke bentuk Persamaan (2.4) dan (2.5).

*Precision* = (2.4)

*Recall* = (2.5)

Terdapat dua hal penting yang biasanya digunakan sebagai acuan dalam mengukur keefektifan suatu sistem temu kembali informasi yaitu perolehan (*recall*) dan ketepatan (*precision*). Keefektifan suatu sistem temu kembali informasi dinilai berdasarkan teori yang dicetuskan oleh Lancaster (1991) dalam Pendit (2008) yaitu relevan dan tidak relevan, jadi efektifitas temu kembali informasi dibedakan menjadi efektif jika nilai diatas 50% dan tidak efektif jika nilai dibawah nilai 50%. Selain itu, suatu sistem temu kembali dinyatakan efektif apabila hasil penelusuran mampu menunjukkan ketepatan (*precision*) yang tinggi sekalipun perolehannya (*recall*) rendah (Rowley 1992) dalam Hasugian (2006).

1. ***F-Measure***

*F-Measure* merupakan salah satu perhitungan evalusasi dalam informasi temu kembali yang mengkombinasikan recall dan precission. Nilai recall dan Precission pada suatu keadaan dapat memiliki bobot yang berbeda. Ukuran yang menampilkan timbal balik antara Recall dan Precission adalah F-Measure yang merupakan bobot harmonic mean dan reall dan precission. Persamaan (2.3) untuk menghitung nilai F sebagai berikut.

F = 2 . (2.3)

1. **Pengujian Durasi pada deteksi kata Latin**

Pengujian durasi bertujuan untuk menghitunng berapa lama berlangsungnya sistem dalam memproses data untuk dapat mendeteksi kata Latin dalam sebuah dokumen.

**BAB III**

# METODOLOGI PENELITIAN

* 1. **Spesifikasi Perangkat Keras dan Perangkat Lunak**

Alat yang digunakan dalam penelitian ini berupa perangkat lunak dan perangkat keras. Adapaun alat yang digunakan adalah sebagai berikut:

* + 1. **Perangkat keras**

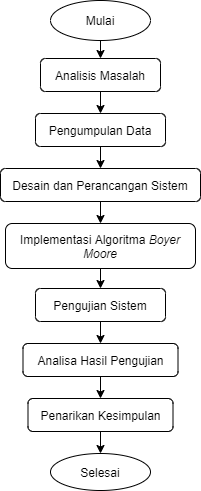
Pada dasarnya, perangkat keras adalah komponen komputer yang dapat dilihat secara langsung atau berbentuk nyata dan berfungsi untuk mendukung proses komputerisasi yang ada. Dalam penelitian ini, perangkat keras yang digunakan yaitu:

1. Laptop Asus A456U
2. Prosessor Intel(R) Core i5
3. RAM 4 GB
4. *Harddisk Drive* 1 TB
5. LCD Monitor 14 inchi
6. *Mouse*
   * 1. **Perangkat lunak**

Berbeda dengan perangkat keras, perangkat lunak adalah program komputer yang berfungsi sebagai sarana interaksi antara pengguna dan perangkat keras. Adapun perangkat lunak yang digunakan adalah sebagai berikut:

1. Sistem Operasi windows 10
2. *Visual Studio Code*
3. PHP 8.0
4. *Xampp*
5. *Database My*SQL
6. *Python* 3.7.2
7. *Google Chrome*
   1. **Metode Penelitian**

Metode penelitian merupakan beberapa tahapan yang dirancang dan dijadikan sebagai panduan dalam melakukan sebuah penelitian. Metode penelitian yang dilakukan digambarkan pada diagram alir penelitian pada Gambar 3.1.



Gambar 3. 1 Diagram Alir Penelitian

* + 1. **Analisis Masalah**

Analisis masalah dilakukan untuk menemukan masalah-masalah yang ada sehingga peneliti dapat merancang dan membangun aplikasi sesuai dengan yang dibutuhkan. Pemberian cetak miring pada kata Bahasa Latin dalam laporan ilmiah menggunakan *tools replace* *pada microsoft word* terkadang memperlambat waktu dalam penulisan jika harus mengubah satu per satu kata-kata berbahasa asing (bahasa Latin) tersebut menjadi cetak miring dan memiliki kemungkinan terlewatnya kata atau istilah asing (bahasa Latin) yang harus tercetak miring. Sehingga dalam hal ini, perancangan dan pengembangan aplikasi dibutuhkan guna menangani masalah tersebut. Pada pengembangan aplikasi, peneliti menentukan algoritma untuk proses pencocokan string dan menemukan kata atau istilah Bahasa Latin yang kemudian kata tersebut dicetak miring. Sehingga dari permasalahan yang ada, algoritma yang digunakan yaitu algoritma *boyer moore*.

* + 1. **Pengumpulan Data**

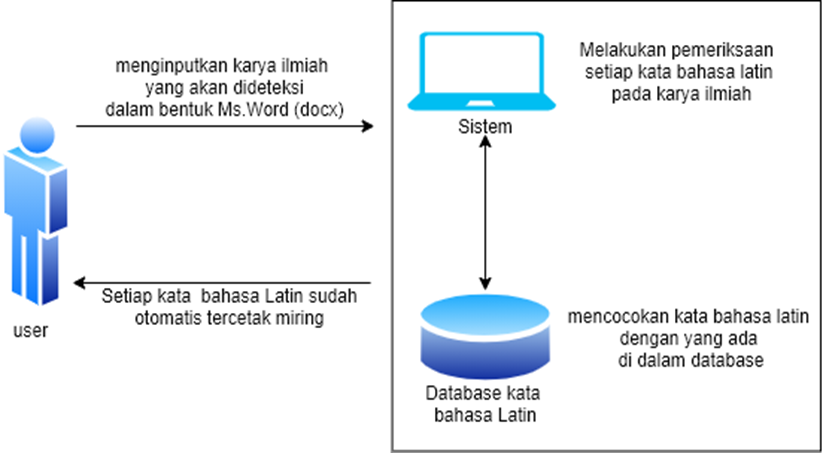
Pengumpulan data diperlukan guna untuk kebutuhan pengembangan aplikasi dan pengujian. Data yang diperlukan pada pengembangan aplikasi adalah data kata-kata Bahasa Latin yang digunakan sebagai database. Database ini diperlukan untuk proses pencocokan string dan pencarian kata antara kata latin dalam dokumen dan kata latin dalam database. Data kata yang dikumpulkan berjumlah 33313 kata, yang didapat dari *http://kyle-p-johnson.com*. Sedangkan data yang diperlukan untuk pengujian adalah berupa laporan ilmiah atau jurnal yang telah sesuai dengan aturan penulisan Bahasa Indonesia PEUBI (Pedoman Umum Bahasa Indonesia) yang mana setiap bahasa asing (bahasa Latin) telah dicetak miring. Data pengujian ini nantinya digunakan dalam pengujian akurasi yaitu pengujian *precision*, *recall* dan *f-measure*.

* + 1. **Desain dan Perancangan Sistem**

Sistem yang dibangun adalah sebuah aplikasi yang digunakan untuk memberikan cetak miring pada bahasa asing (Bahasa Latin) pada laporan ilmiah. Arsitektur sistem merupakan gambaran secara umum mengenai hubungan antar komponen yang terlibat dalam sistem yang dibangun. Sistem terdiri dari pengguna, aplikasi website dan database. Halaman web dapat diakses melalui browser dengan bantuan jaringan internet. Perancangan sistem dilakukan dengan menggunakan alat bantu sebagai berikut:

1. **Arsitektur Sistem**

Berikut arsitekstur sistem penelitian pada Gambar 3.2.

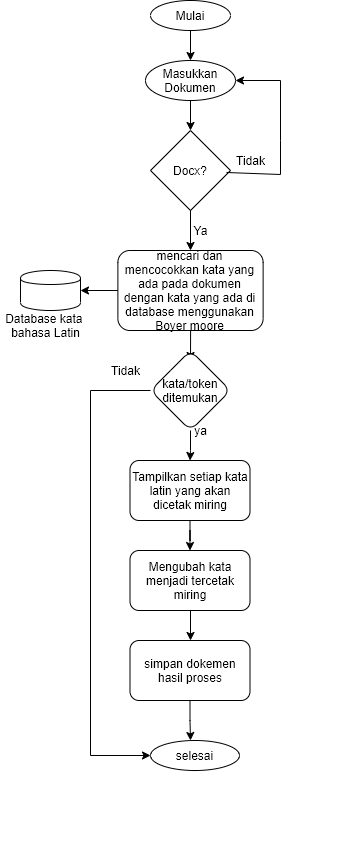


Gambar 3. 2 Arsitektur Sistem

*User* pengguna mengunggah dokumen ilmiah yang ingin diperiksa penulisannya dalam bentuk file Microsoft Word (.docx). Kemudian sistem akan mulai melakukan pencarian kata. Setiap kata pada teks tersebut kemudian diperiksa ketersediaannya pada database menggunakan algoritma *boyer moore*, jika kata terdapat pada database maka kata tersebut dianggap benar dan akan ditampilkan daftar kata Latin yang berhasil terdeteksi untuk kemudian dicetak miring. Sistem yang dibangun terdiri dari dua bagian yaitu *front-end* dan *back-end*. Bagian *front-end* merupakan bagian yang dapat dilihat dan dilakukan oleh pengguna yaitu meng-klik tombol untuk proses penginputan dokumen ilmiah, proses pencetakan miring pada kata latin dan proses pengunduhan dokumen ilmiah yang telah selesai diproses. Bagian *back-end* merupakan bagian yang mengolah teks pada dokumen dengan kondisi kata latin belum tercetak miring menjadi kata latin yang telah dicetak miring.

1. **Diagram Alir Sistem**

Adapun diagram alir proses pada *front-end* sampai *back-end* digambarkan dengan diagram alir pada Gambar 3.3.



Gambar 3. 3 Diagram Alir Sistem

Dari diagram pada Gambar 3.3, dapat dijelaskan sebagai berikut:

1. Proses dimulai.
2. Pengguna memasukkan dokumen dengan format *.docx.*
3. Sistem memverifikasi ekstensi dokumen apakah berupa .*docx.* jika ekstensi sesuai maka proses akan dilanjutkan ke tahap pencarian kata menggunakan algoritma *boyer moore,* dan jika tidak sesuai maka pengguna akan diarahkan untuk memasukkan dokumen ulang.
4. Sistem melakukan pemeriksaan setiap kata/token apakah terdapat pada database kata bahasa Latin atau tidak menggunakan algoritma *boyer moore*.
5. Jika kata terdapat pada *database* kata bahasa Latin maka akan dilanjutkan ke tahap menampilkan kata latin yang akan dicetak miring, jika kata tidak terdapat pada *database* kata bahasa Latin maka proses selesai.
6. sistem menampilkan hasil pemeriksaan setiap kata yang terdapat pada *database* kata bahasa Latin yang akan dicetak miring.
7. Sistem melakukan perubahan kata yang terdapat pada *database* kata bahasa Latin menjadi tercetak miring.
8. Sistem menyimpan dokumen hasil proses, yang setiap kata latinnya sudah tercetak miring.
9. Proses selesai.
10. **Data Flow Diagram(DFD)**
    1. **Diagram Konteks Sistem**

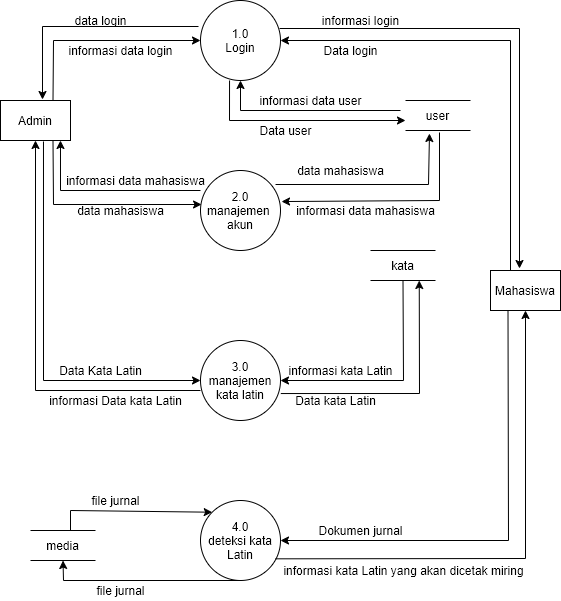
Diagram konteks adalah diagram yang memberikan gambaran umum terhadap kegiatan yang berlangsung dalam sistem. Diagram konteks dapat dilihat pada gambar 3.4



Gambar 3. 4 Diagram Konteks Sistem

* 1. **Diagram OverView Sistem**

Diagram overview sistem merupakan gambaran rinci dari proses yang dilakukan pada diagram konteks sistem. Arus data dapat dilihat dengan jelas pada diagram tersebut. Proses-proses yang terjadi dapat dilihat pada Gambar 3.5.



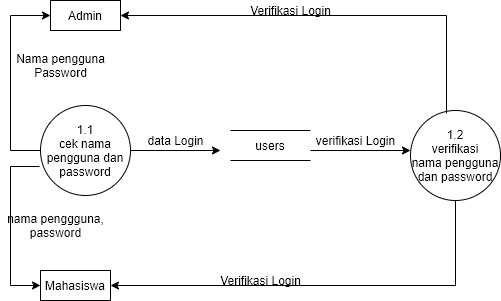
Gambar 3. 5 Diagram Overview Sistem

Berdasarkan gambar 3.5 dapat dijelaskan proses yang terjadi pada diagram overview sistem sebgai berikut :

1. Proses 1.0 *Login*, yaitu proses yang harus dilakukan oleh *admin* dan *user* dengan memasukan *username* dan *password*. Setelah proses *input* *username* dan *password* valid, maka *admin* dan *user* dapat mengakses sistem sesuai fungsinya.
2. Proses 2.0 Manajemen Akun, yaitu proses yang dilakukan oleh *admin* untuk mengelola data *user*.
3. Proses 3.0 Manajemen Kosa Kata, yaitu proses yang dilakukan oleh *admin* untuk mengelola data kosa kata.
4. Proses 4.0 Deteksi Kata Latin, yaitu proses yang dilakukan oleh user untuk melakukan pendeteksian kata Latin yang terdapat pada dokumen yang diunggah oleh user untuk mendapatkan daftar kata Latin unutuk kemudian dicetak miring dari sistem.
   1. **Diagram Rinci Sistem**

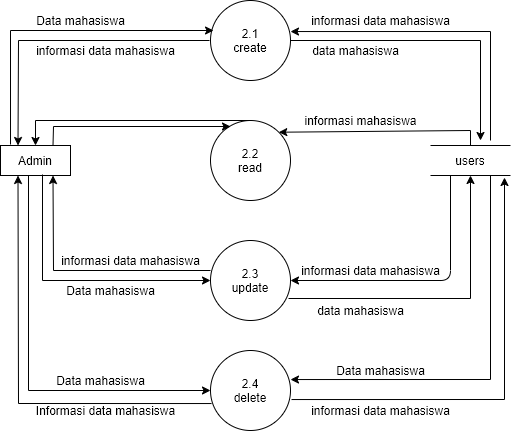
Diagram rinci sistem merupakan diagram yang menguraikan lebih lanjut mengenai proses dari diagram overview, yang memperlihatkan arus data masuk dan arus data keluar. Berdasarkan diagram overview pada gambar 3.6 terdapat beberapa model diagram rinci yaitu sebagai berikut:

* 1. Proses 1.0 adalah login yang terdiri dari 2 proses rinci :



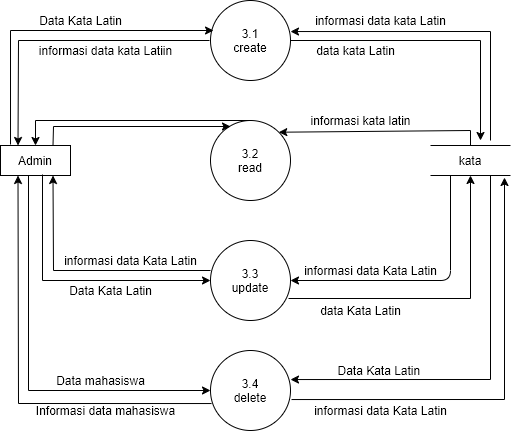
Gambar 3. 6 Diagram Rinci 1.0

1. Proses 1.0 pada proses ini admin atau mahasiswa memasukkan nama pengguna dan password
2. Proses 2.0 pada proses ini sistem sistem memverifikasi nama pengguna yang dimasukkan oleh users
   1. Proses 2.0 adalah manajemen akun yang terdiri dari 4 proses rinci :



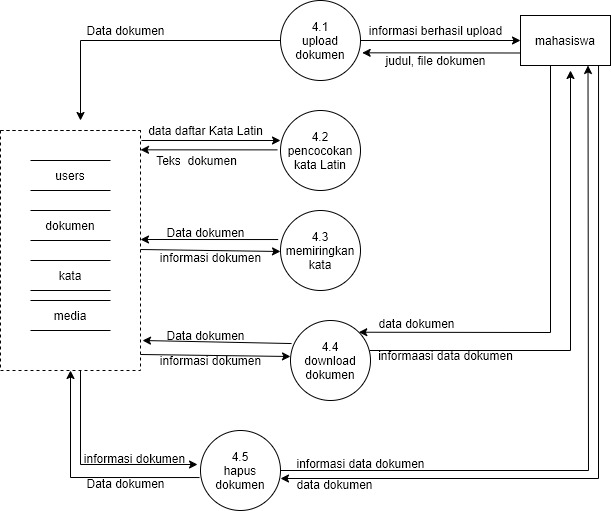
Gambar 3. 7 Diagram Rinci 2.0

1. Proses 2.1 create, pada proses ini *admin* membuat akun yang akan digunakan oleh mahasiswa pada aplikasi.
2. Proses 2.2 read, pada proses ini *admin* dapat melihat seluruh akun yang telah didaftarkan ke dalam aplikasi.
3. Proses 2.3 update, pada proses ini *admin* dapat melakukan perubahan data pada akun-akun mahasiswa yang telah didaftarkan sebelumnya.
4. Proses 2.4 delete, pada proses ini *admin* dapat menghapus akun mahasiswa yang telah terdaftar sebelumnya.
   1. Proses 3.0 adalah manajemen kata Latin yang terdiri dari 4 proses rinci :



Gambar 3. 8 Diagram Rinci 3.0

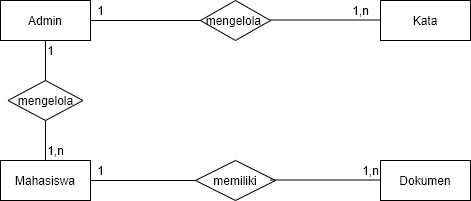
1. Proses 3.1 *create*, pada proses ini *admin* menambahkan data kosa kata Latin pada aplikasi.
2. Proses 3.2 *read*, pada proses ini *admin* dapat melihat seluruh data kosa kata Latin yang telah ditambahkan ke dalam aplikasi.
3. Proses 3.3 update, pada proses ini *admin* dapat melakukan perubahan data Kosa kata Latin yang telah ditambahkan sebelumnya.
4. Proses 3.4 delete, pada proses ini *admin* dapat menghapus kosa kata Latin yang telah ditambahkan sebelumnya.
   1. Proses 4.0 adalah deteksi Kata Latin yang terdiri dari 5 proses rinci:



Gambar 3. 9 Diagram Rinci 4.0

1. Proses 4.1 *upload* dokumen jurnal, pada proses ini mahasiswa dapat mengunggah dokumen jurnal yang akan diproses dalam *form*at file .docx.
2. Proses 4.2 cari deteksi kata Latin, pada proses ini sistem memeriksa ketersediaan kata-kata dari dokumen yang telah diunggah sebelumnya yang terdapat pada database. Jika kata terdapat pada databse, maka kata tersebut akan muncul pada daftar Kata Latin yang akan dimiringkan oleh sistem.
3. Proses 4.3 memiringkan kata, pada proses ini kata-kata Latin yang terdeteksi akan dibuat menjadi tercetak miring oleh sistem .
4. Proses 4.6 *download* dokumen, pada proses ini mahasiswa dapat mengunduh dokumen yang telah diunggah sebelumnya.
5. Proses 4.7 hapus dokumen, pada proses ini mahasiswa dapat menghapus dokumen yang telah diunggah sebelumnya.
6. **Entity Relationship Diagram (ERD)**

Entity Relationship Diagram (ERD) merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. Secara umum basis data yang akan dibangun memiliki hubungan sebagai berikut.



Gambar 3. 10 *Entity Relationship Diagram(ERD)*

1. **Spesifikasi Basis Data**

Berikut ini daftar spesifikasi tabel yang digunakan dalam perancangan aplikasi *font italic* untuk pendeteksi bahasa Latin :

a. Spesifikasi Tabel Users

Nama Database : skripsi

Nama Tabel : users

Keterangan : Tabel yang berisi data user, yaitu admin dan mahasiswa.

Tabel 3. 1 Spesifikasi Tabel Users

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Field | Tipe Data | Null | Key |
| Name | varchar (255) | No | Primery Key |
| username | varchar (255) | No |  |
| Level | varchar (255) | No |  |
| Password | Varchar(255) | No |  |
| Remember\_Token | Varchar(100) | No |  |
| created\_at | timestamp | Yes |  |
| updated\_at | timestamp | Yes |  |

b. Spesifikasi Tabel Kata

Nama Database : skripsi

Nama Tabel : kata

Keterangan : Tabel yang berisi data kosa kata Latin.

Tabel 3. 2 Spesifikasi Tabel Kata

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Field | Tipe Data | Null | Key |
| Isi | varchar (255) | No | primary key |

c. Spesifikasi Tabel Dokumen

Nama Database : skripsi

Nama Tabel : Dokumen

Keterangan :Tabel yang berisi data dokumen yang diunggah ke database

Tabel 3. 3 Spesifikasi Tabel Dokumen

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Field | Tipe Data | Null | Key |
| Id | Bigint (10) | no | primary key |
| Username | varchar (255) | no |  |
| Nama | varchar (255) | no |  |
| created\_at | timestamp | yes |  |
| updated\_at | timestamp | yes |  |

d. Spesifikasi Tabel Media

Nama Database : Skripsi

Nama Tabel : media

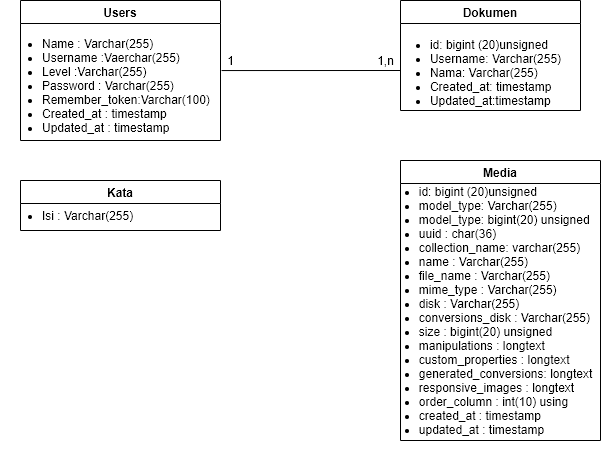
Keterangan : Tabel yang berisi data file yang diunggah ke *database*

Tabel 3. 4 Spesifikasi Tabel Media

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Field | Tipe Data | Null | Key |
| Id | bigint (20) | no | primary key |
| model\_type | varchar (255) | no |  |
| model\_id | bigint (20) | no |  |
| Uuid | char (36) | yes |  |
| collection\_name | varchar (255) | no |  |
| Name | varchar (255) | no |  |
| file\_name | varchar (255) | no |  |
| mime\_type | varchar (255) | yes |  |
| Disk | varchar (255) | no |  |
| conversions\_disk | varchar (255) | yes |  |
| Size | bigint (20) | no |  |
| manipulations | Longtext | no |  |
| custom\_properties | Longtext | no |  |
| generated\_conversions | Longtext | no |  |
| responsive\_images | Longtext | no |  |
| order\_column | int (10) | yes |  |
| created\_at | Timestamp | yes |  |
| updated\_at | Timestamp | yes |  |

1. **Relasi Antar Tabel**

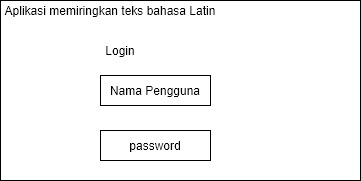
Relasi antar tabel merupakan gambaran hubungan antar tabel yang dipergunakan dalam perancangan sistem. Relasi antar tabel implementasi algoritma boyer moore pada aplikasi *font italic* untuk pendeteksi bahasa Latin dapat dilihat pada Gambar 3.11 berikut.

****

Gambar 3. 11 Hubungan Antar Tabel

1. **Rancangan Antarmuka** 
   1. **Rancangan Antarmuka Halaman Login**

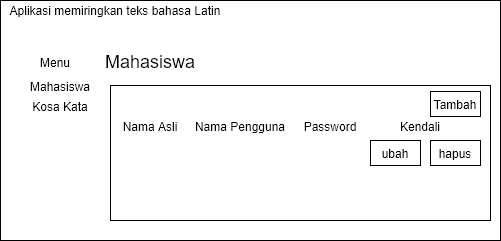
Halaman *login* adalah halaman yang digunakan oleh *user* yaitu admin dan mahasiswa untuk masuk ke dalam sistem. Pada halaman ini *user* harus mengisi data yang diperlukan pada *form login* agar dapat mengakses sistem. Perancangan antarmuka halaman *login* dapat dilihat pada gambar 3.12 berikut.



Gambar 3. 12 Rancangan Antarmuka Halaman Login

* 1. **Rancangan Antarmuka Halaman Dashboard Admin**

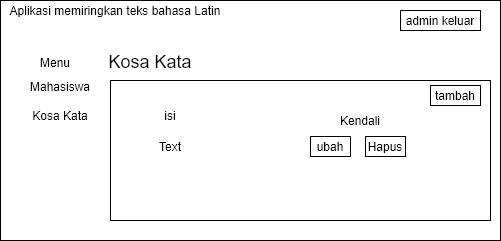
Halaman *dashboard* admin adalah halaman yang dapat diakses oleh *user* apabila berhasil *login* sebagai admin. Halaman ini berisi fitur-fitur yang dapat dijalankan oleh admin. Perancangan antarmuka halaman *dashboard* admin dapat dilihat pada gambar 3.13 berikut



Gambar 3. 13 Rancangan Antarmuka Halaman Dashboard Admin

* 1. **Rancang Antarmuka Halaman Kosa Kata**

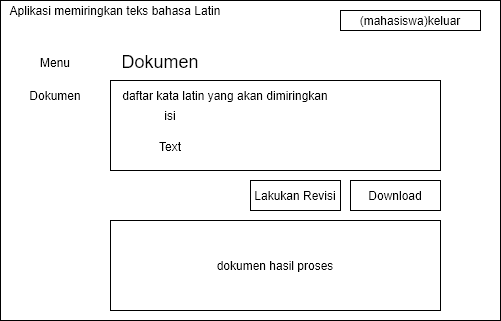
Pada tampilan rancangan halaman kosa kata admin dapat informasi daftar kosa kata bahasa latin yang ada dalam database, admin dapat mengubah,menambah,dan menghapus kosa kata. Berikut gambar tampilan rancangan halaman kosa kata admin dapat dilihat pada Gambar 3.14



Gambar 3. 14 Rancang Antarmuka Halaman Kosa Kata

* 1. **Rancangan Antarmuka Halaman Dokumen**

Pada tampilan rancangan tampilan *utama* mahasiswa yaitu Dokumen. Pada halaman ini pengguna dapat melihat daftar kata latin yang akan dimiringkan, dokumen hasil proses yang sudah dimiringkan kat latinnya. Serta pengguna juga dapat melakukan revisi jika ingin ada kata latin yang dianggap tidak perlu dimiringkan. Pengguna juga dapat mendownload dokumen hasil dari proses memiringkan kata dalam bentuk .docx. Berikut gambar tampilan rancangan halaman utama mahasiswa dapat dilihat pada Gambar 3.15



Gambar 3. 15 Rancangan Antarmuka Halaman Dokumen

* + 1. **Implementasi Algoritma *Boyer Moore***

*Booyer-Moore* merupakan salah satu Algortima *Pattern Matching* yang cukup terkenal. Algoritma *Boyer moore* secara umum mempunyai 2 teknik dalam pencarian katanya yaitu:

1. *The Looking-Glass Technique*

*The Looking-Glass Technique* yang maksudnya pencocokan string ini dilakukan’seolah-olah’ seperti terbalik. pada *boyer moore* pencocokan string dimulai dari karakter terakhir pada kata kunci(P), namun untuk pencocokan tidak dimulai dari akhir teks(T) ,pemeriksaan pada teks(T) tetap dimulai dari awal, hanya nilai karakter diawal menyesuaikan panjang string dari kata kunci(P).

Contoh : T : saya mahasiswa

P : siswa

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T | s | a | y | a |  | m | a | h | a | s | i | s | w | a |
| P | s | i | s | w | a |  |  |  |  |  |  |  |  |  |

T

2. *The Character-Jump Technique*

*Character-jump Technique* dilakukan ketika terjadi ketidakcocokan pada pencocokan string. Maka algoritma boyer moore akan melakukan lompatan atau bisa dianalogikan dengan pergeseran dengan posisi tertentu. Untuk menentukan lompatan, terdapat 3 kasus yang harus diperhatikan , yaitu:

1. Kasus 1 :

Misal

T = ..xa..??

P = xcba



Gambar 3. 16 Contoh kasus 1

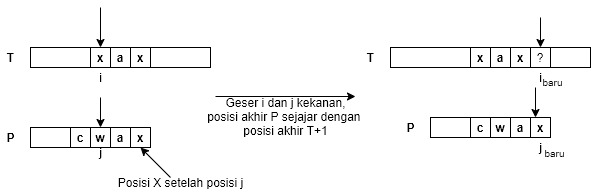
Pada gambar 3.16 terjadi ketidakcocokan pada T(i) dan pada P(J), dan karakter yang menyebabkan ketidak cocokan pada teksnya adalah karakter X, kemudian kita mencari dari kata kinci(P) adakah X muncul pada disebelah kiri dari indeks P(j). jadi yang dilakukan jika menjumpai kasus 1 adalah ‘seolah-olah’ melakukan pergeseran kata kunci(P) kekanan agar posisi karakter X di T(i) sejajar dengan posisi kemunculan terakhir karakter X di kata kunci(P). Perlu diingat bahwa pada algoritma *boyer moore* pemeriksaan berikutnya indeks ‘j’ selalu dimulai pada indeks terakhir yang kita geser.

1. Kasus 2:

Misal

T = ..xax..??

P = cwax



Gambar 3. 17 Contoh Kasus 2

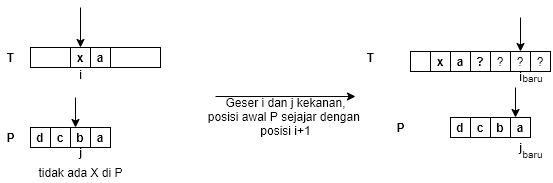
Pada tabel 3.17 dapat dilihat ketidakcocokan terjadi pada T(i) dan P(j) dan karakter pada teks(T) yang menyebabkan terjadi ketidakcocokan adalah karakter X. pada kasus yang kedua ini terdapat karakter X pada kata kunci(P),tetapi pada posisi dengan indeks yang lebih besar dari pada J, jadi posisinya tidak lagi pada kiri J seperti kasus yang pertama. Jika ditemukan kasus yang seperti ini maka dilakukan pergeseran kata kunci(P) satu karakter ke kanan, agar posisi indeks terakhir P sejajar dengan posisi akhir T sebelumnya + 1. Karena diketahui sebenarnya ada karakter X pada P hanya sudah terlewat pemeriksaannya. Untuk pemeriksaan berikutnya perlu diingat bahwa untuk kata kunci(P) selalu dimulai dari indeks yang terakhir.

1. Kasus 3:

Misal

T = xa

P = dcba



Gambar 3. 18 Contoh kasus 3

Pada gambar 3.18 dapat dilihat ketidak cocokan terjadi pada T(i) dan P(j), dan karakter pada teks(T) yang menyebabkan ketidakcocokan adalah karakter X, pada kasus 1 dan 2 masih ditemukan karakter X pada kata kunci(P) , sedangkan untuk kasus 3 ini karakter X pada Teks(T) yang menyebabkan ketidakcocokan tidak pernah muncul pada kata kunci(P) . jika kasus ini terjadi maka dilakukan pergeseran sedemikian sehingga posisi pertama kata kunci(P) sejajar dengan indeks i+1, karena sebetulnya diketahui bahwa karakter X tidak akan pernah muncul di P oleh karena itu pergeseran dilakukan langsung kekanan untuk memeriksa karakter berikutnya pada teks.

Contoh cara kerja algoritma *Boyer Moore* adalah sebagai berikut :

T : Nama Latin jeruk adalah Citrus

P :Citrus

Tabel 3. 5 Contoh cara kerja Algoritma *Boyer Moore*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n | a | m | a |  | l | a | t | i | n |  | j | e | r | u | k |  | a | d | a | l | a | h |  | c | i | t | r | u | s |
| c | i | t | r | u | s |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | c | i | t | r | u | s |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | c | i | t | r | u | s |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | c | i | t | r | u | s |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | c | i | t | r | u | s |

Keterangan :

|  |  |
| --- | --- |
|  | Tidak Cocok(*missmatch*) |
|  | cocok(*match*) |

Pada tabel 3.5 dapat dilihat terjadi ketidakcocokan karakter L pada teks(T) dengan karakter S pada kata kunci (P) . Ternyata karakter L pada teks(T) tidak pernah muncul pada kata kunci(P) maka ini seperti kasus 3, berarti dilakukan pergeseran hingga posisi karakter pertama pada kata kunci(P) sejajar dengan karakter L pada Teks(T) + 1.pemeriksaan selanjutnya dimulai dari indeks terakhir pada kata kunci(P) yang sudah digeser. Terjadi lagi huruf di kanan tidak ada kecocokan dengan n karakter maka m karakter atau kata kunci(P) akan menggeser sejauh m karakter untuk mencocokkan kembali huruf yang dimaksud, sehingga menjadi cocok.

* + 1. **Pengujian Sistem**

Pengujian sistem adalah sebuah tahap mengevaluasi sistem yang dibangun. Terdapat dua jenis objek penilaian yaitu kinerja sistem dan. Pengujian terhadap kinerja sistem akan menggunakan pengujian *black box*,juga menggunakan pengujian *precision, recall* dan *f-measure* ,serta pengujian Durasi dalam pendeteksian kata latinnya.

* + - 1. **Pengujian *Black Box***

Untuk mengetahui aplikasi berjalan dengan baik atau tidak, maka perlu dilakukan pengujian sistem. Adapun pengujian sistem yang akan digunakan adalah pengujian dengan metode *black* *box* *testing*. Pengujian dengan metode *black* *box* *testing* adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Hasil pengujian akan dirangkum ke dalam beberapa tabel antara lain sebagai berikut:

* 1. Pengujian Halaman *Login*

Pada rancangan pengujian halaman *login* dilakukan proses *input* di halaman *login*. Berikut rancangan pengujian halaman *login* dapat dilihat pada Tabel 3.6

Tabel 3. 6 Skenario Pengujian Halaman *Login*

|  |  |  |  |
| --- | --- | --- | --- |
| **Fungsi** | **Contoh Fungsi** | **Hasil Eksekusi** | **Keterangan** |
| Fungsi *login* | Nama Pengguna dan *password* kosong |  |  |
| mengosongkan salah satu kolom |  |  |
| id pegawai salah |  |  |
| *password* salah |  |  |
| Nama Pengguna dan *password* benar |  |  |

* 1. Pengujian Halaman Tambah Akun Pengguna

Pada Perancangan pengujian Halaman Tambah Akun Pengguna dilakukan proses *input* data seperti nama asli, nama pengguna, kata sandi. Berikut rancangan pengujian tambah akun pengguna dapat dilihat pada tabel 3.7

Tabel 3. 7 skenario pengujian halaman tambah akun pengguna

| **Input** | **Contoh data** | | **Hasil eksekusi** | **Keterangan** |
| --- | --- | --- | --- | --- |
| Data kosong | Nama Asli |  |  |  |
| Nama Pengguna |  |
| Kata Sandi |  |
| Salah satu data kosong | Nama Asli |  |  |  |
| Nama Pengguna |  |
| Kata Sandi |  |
| Data  Benar | Nama Asli |  |  |  |
| Nama Pengguna |  |
| Kata Sandi |  |

* 1. Pengujian Halaman Hapus Akun Pengguna

Pada perancangan pengujian halaman hapus akun pengguna dilakukan eksekusi mengklik tombol hapus maka akan timbul pesan konfirmasi. Berikut skenario rancangan pengujian pada halaman hapus akun pengguna dapat dilihiat pada tabel 3.8

Tabel 3. 8 skenario pengujian halaman hapus akun pengguna

| **Input** | **Contoh data** | | **Hasil eksekusi** | **Keterangan** |
| --- | --- | --- | --- | --- |
| Klik tombol “Hapus” |  |  |  |  |

* 1. Pengujian Halaman Tambah Dokumen

Pada Perancangan pengujian Halaman Tambah dokumen dilakukan proses input data seperti nama file, dan dokumen yang ingin diproses. Berikut rancangan pengujian tambah dokumen dapat dilihat pada tabel 3.9

Tabel 3. 9 skenario pengujian halama tambah dokumen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Contoh data** | | **Hasil eksekusi** | **Keterangan** |
| Data kosong | Nama |  |  |  |
| Berkas |  |
| Data salah | Nama |  |  |  |
| Berkas |  |
| Data benar | Nama |  |  |  |
| Berkas |  |

* 1. Pengujian Halaman Ubah Dokumen

Pada Perancangan pengujian Halaman ubah dokumen dilakukan proses input dokumen baru untuk mengganti dokumen lama jika diperlukan. Berikut skenario rancangan pengujian ubah dokumen dapat dilihat pada tabel 3.10

Tabel 3. 10 skenario pengujian halaman ubah dokumen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Contoh data** | | **Hasil eksekusi** | **Keterangan** |
| Data kosong | Nama |  |  |  |
| Berkas |  |
| Data salah | Nama |  |  |  |
| Berkas |  |
| Data benar | Nama |  |  |  |
| Berkas |  |

* 1. Pengujian Halaman Download Dokumen

Pada Perancangan pengujian Halaman download dokumen dilakukan eksekusi dengan mengklik tombol download. Berikut skenario rancangan pengujian download dokumen dapat dilihat pada tabel 3.11

Tabel 3. 11 skenario pengujian halaman download dokumen

| **Input** | **Contoh data** | | **Hasil eksekusi** | **Keterangan** |
| --- | --- | --- | --- | --- |
| Klik tombol “Download” |  |  |  |  |

* 1. Penguian Halaman Detail Pemeriksaan Dokumen

Pada Perancangan pengujian Halaman detail pemeriksaan dokumen dilakukan eksekusi dengan mengklik tombol detail. Berikut skenario rancangan pengujian detail pemeriksaan dokumen dapat dilihat pada tabel 3.12

Tabel 3. 12 skenario pengujian halaman detail pemeriksaan dokumen

| **Input** | **Contoh data** | | **Hasil eksekusi** | **Keterangan** |
| --- | --- | --- | --- | --- |
| Klik tombol “Detail” |  |  |  |  |

* 1. Pengujian Halaman Hapus Dokumen

Pada Perancangan pengujian Halaman hapus dokumen dilakukan eksekusi dengan mengklik tombol hapus. Berikut skenario rancangan pengujian hapus dokumen dapat dilihat pada tabel 3.13

Tabel 3. 13 skenario pengujian halaman hapus dokumen

| **Input** | **Contoh data** | | **Hasil eksekusi** | **Keterangan** |
| --- | --- | --- | --- | --- |
| Klik tombol “Hapus” |  |  |  |  |

* 1. Pengujian Halaman Tambah Kosa Kata

Pada Perancangan pengujian Halaman Tambah kosa kata dilakukan proses input data, Berikut rancangan pengujian tambah data kosa kata dapat dilihat pada tabel 3.14

Tabel 3. 14 skenario pengujian halaman tambah kosa kata

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Contoh data** | | **Hasil eksekusi** | **Keterangan** |
| Data kosong | isi |  |  |  |
| Data salah | isi |  |  |  |
| Data benar | isi |  |  |  |

* 1. Pengujian Halaman Ubah Kosa Kata

Pada Perancangan pengujian Halaman ubah kosa kata dilakukan proses input data kosa kata baru untuk mengganti kosa kata lama jika diperlukan. Berikut skenario rancangan pengujian ubah kosa kata dapat dilihat pada tabel 3.15

Tabel 3. 15 skenario penguijian halaman ubah kosa kata

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Contoh data** | | **Hasil eksekusi** | **Keterangan** |
| Data kosong | isi |  |  |  |
| Data salah | isi |  |  |  |
| Data benar | isi |  |  |  |

* 1. Pengujian Halaman Hapus Kosa Kata

Pada Perancangan pengujian Halaman hspus kosa kata dilakukan proses eksekusi dengan mgklik tombol hapus. Berikut skenario rancangan pengujian hapus kosa kata dapat dilihat pada tabel 3.16

Tabel 3. 16 skenario pengujian halman hapus kosa kata

| **Input** | **Contoh data** | | **Hasil eksekusi** | **Keterangan** |
| --- | --- | --- | --- | --- |
| Klik tombol “Hapus” |  |  |  |  |

* 1. Pengujian Halaman Logout

Pada Perancangan pengujian Halaman logout dilakukan eksekusi mengklik tombol logout. Berikut skenario rancangan pengujian halaman logout dapat dilihat pada tabel 3.17

Tabel 3. 17 skenario pengujian halaman logout

| **Input** | **Contoh data** | | **Hasil eksekusi** | **Keterangan** |
| --- | --- | --- | --- | --- |
| Klik  tombol “Logout” |  |  |  |  |

* + - 1. **Pengujian *Precision*, *Recall* dan *F-Measure***

Pengujian *precision* dilakukan untuk mengukur tingkat relevansi atau ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Pengujian *precision* dilakukan dengan rumus:

*Precision* = (3.2)

Pengujian *recall* dilakukan untuk mengukur tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Pengujian *recall* dilakukan dengan rumus:

*Recall* = (3.3)

Setelah mendapatkan nilai *precision* dan *recall*, selanjutnya dilakukan pengujian *F-measure* untuk mengukur tingkat akurasi dari hasil prediksi.

F= 2 . (3.4)

Keterangan:

a = Kata Latin yang tercetak miring oleh sistem

b = Kata Latin yang tidak tercetak miring oleh sistem

c = Kata Latin yang penulisannya tidak perlu dicetak miring, namun menjadi tercetak miring oleh sistem.

Pengujian ini dilakukan dengan membandingkan kalimat asli dengan kalimat yang di hasilkan sistem.

* + - 1. **Pengujian Durasi pada Deteksi Kata Latin**

Pengujian durasi ini bertujuan untuk menghitung berapa lama berlangsungnya sistem dalam memproses data untuk dapat mendeteksi kata latin dalam sebuah dokumen.pengujian ini dilakukan dengan menggunakan menguji sebuah dokumen pada sistem. Hasil pengujian akan dirangkum dalam tabel 3.18 yaitu sebagai berikut :

Tabel 3. 18 Pengujian Durasi pada Deteksi Kata Latin

|  |  |  |  |
| --- | --- | --- | --- |
| No | Jumlah Kata | Waktu  (detik) | Jumlah kata Latin yang terdeteksi |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |

* + 1. **Analisis Hasil Pengujian**

Pada tahap ini, hasil pengujian sistem akan dianalisis secara keseluruhan untuk mempermudah penarikan kesimpulan.

* + 1. **Penarikan Kesimpulan**

Kesimpulan dirumuskan berdasarkan tahapan-tahapan yang telah dilakukan sebelumnya apakah sistem yang dirancang dan dibangun dapat berjalan baik sesuai dengan yang diharapkan.

**BAB IV**

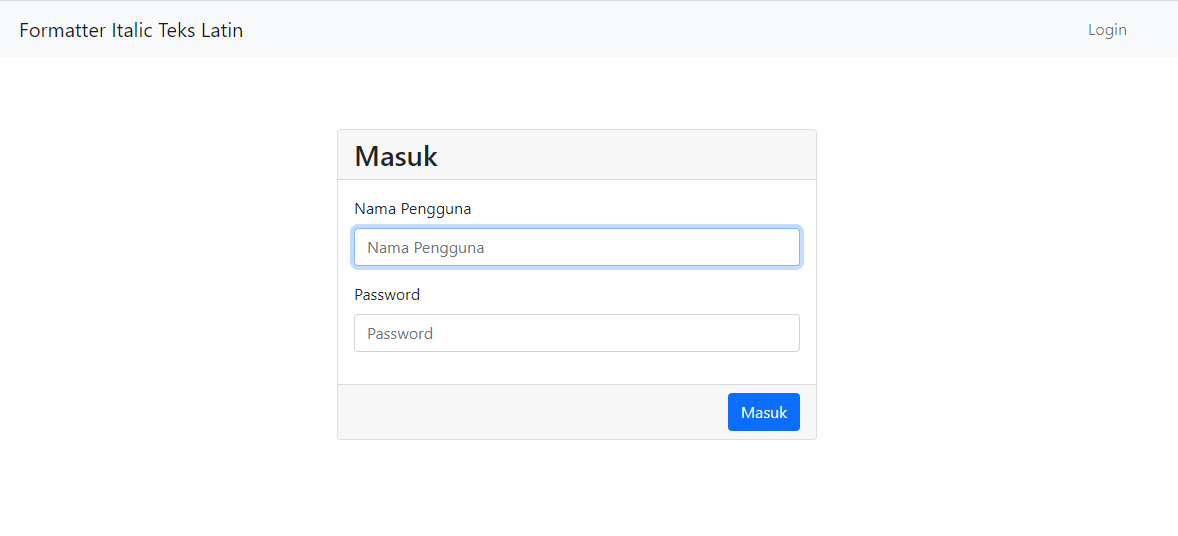
# IMPLEMENTASI DAN HASIL PENGUJIAN

* 1. **Implementasi**

Aplikasi yang dirancang merupakan aplikasi yang dapat membantu pengguna memiringkan kata bahasa Latin pada penulisan dijurnal maupun karya ilmiah agar meminimalisir terlewatnya kata yang harus tercetak miring khususnya kata berbahasa Latin.Adapun hasil implementasinya sebagai berikut:

* + 1. **Antarmuka Halaman Masuk**

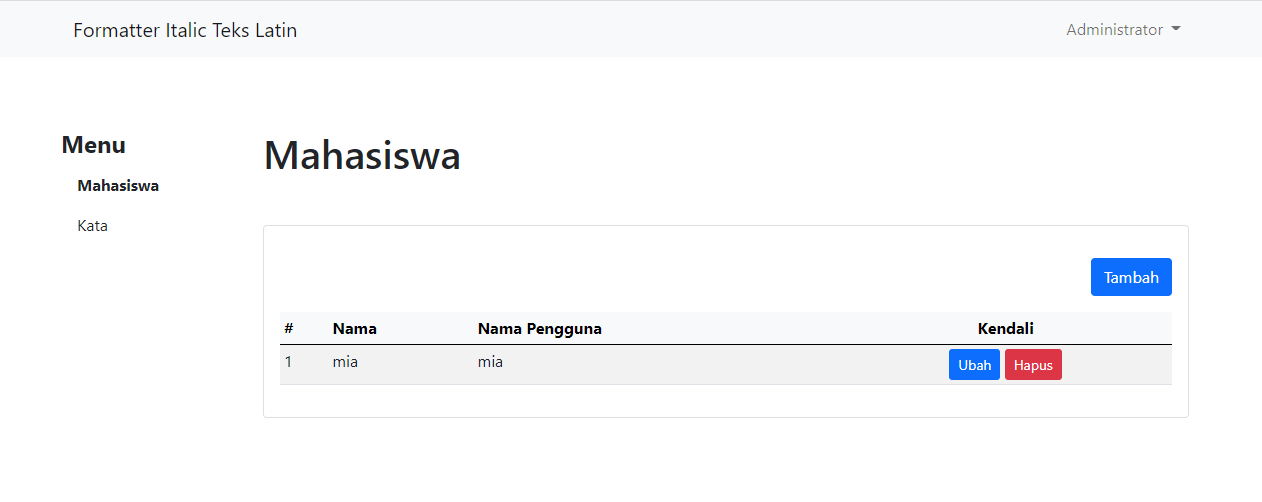
Antarmuka halaman masuk adalah halaman pertama yang muncul ketika *user* mengakses *web* aplikasi. Pengguna harus terlebih dahulu mengisi nama pengguna dan *password* pada *Field* yang sudah disediakan. Jika nama pengguna dan *password* sesuai dengan yang terdapat dalam *database*, maka akan diteruskan ke halaman *dashboard* pengguna. Sedangkan jika nama pengguna dan *password* tidak sesuai, maka akan tampil pesan *error*. Tampilan antarmuka halaman Masuk dapat dilihat pada gambar 4.1



Gambar 4. 1 Tampilan Halaman Masuk

* + 1. **Tampilan Halaman Dashboard Admin**

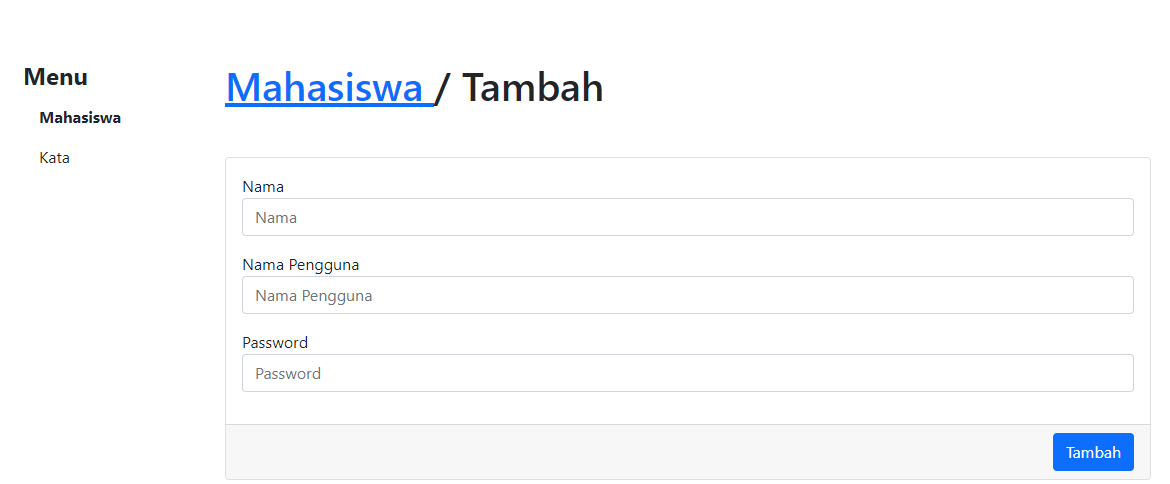
Halaman dashboard admin merupakan halaman yang akan ditampilkan setelah *user* berhasil *login* sebagai admin. Halaman ini menampilkan seluruh data akun mahasiswa yang telah didaftarkan oleh admin di aplikasi. Tampilan antarmuka halaman dashboard admin dapat dilihat pada gambar 4.2.

****

Gambar 4. 2 Tampilan Halaman Dashboard Admin

* + 1. **Tampilan Halaman Tambah Akun**

Halaman tambah akun adalah halaman yang hanya dapat diakses oleh admin. Halaman ini dapat digunakan admin untuk menambahkan data akun mahasiswa, yang dapat digunakan oleh mahasiswa untuk login kedalam aplikasi. Tampilan antarmuka halaman tambah akun dapat dilihat pada 4.3

****

Gambar 4. 3 Halaman Tambah Akun

* + 1. **Tampilan Halaman Perbarui Data Akun**

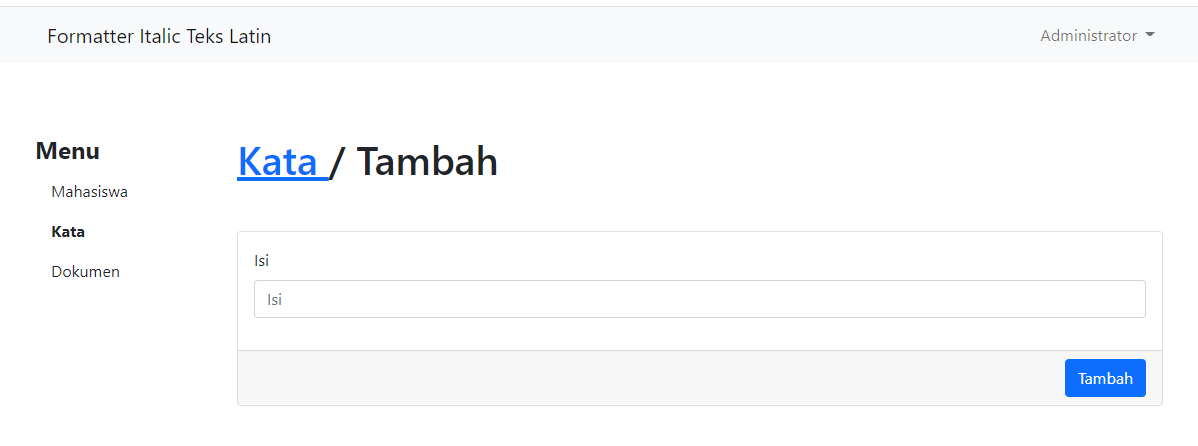
Halaman ubah data akun adalah halaman yang hanya dapat diakses oleh admin. Halaman ini digunakan admin untuk memperbaharui data akun mahasiswa yang sudah ada jika diperlukan. Tampilan halaman perbaharui data akun dapat dilihat pada gambar 4.4



Gambar 4. 4 Halaman Ubah Data

* + 1. **Tampilan Halaman Tambah Kosa Kata Bahasa Latin**

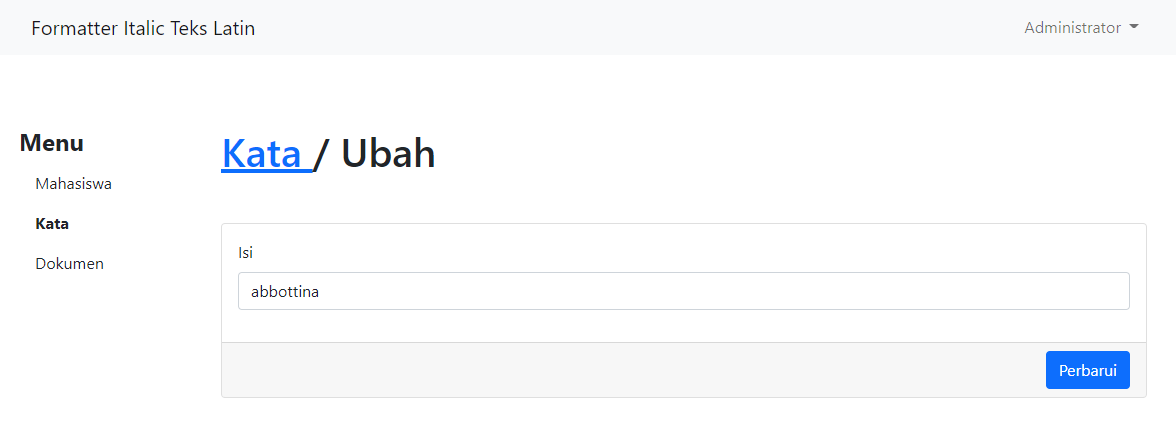
Halaman tambah kata adalah halaman yang hanya dapat diakses oleh admin. Halaman ini dapat digunakan admin untuk menambahkan data kosa kata bahasa Lati. Tampilan antarmuka halaman tambah kosa kata bahasa Latin dapat dilihat pada 4.5

****

Gambar 4. 5 Halaman Tambah Kosa Kata Bahasa Latin

* + 1. **Tampilan Halaman perbarui Data Kosa Kata Bahasa Latin**

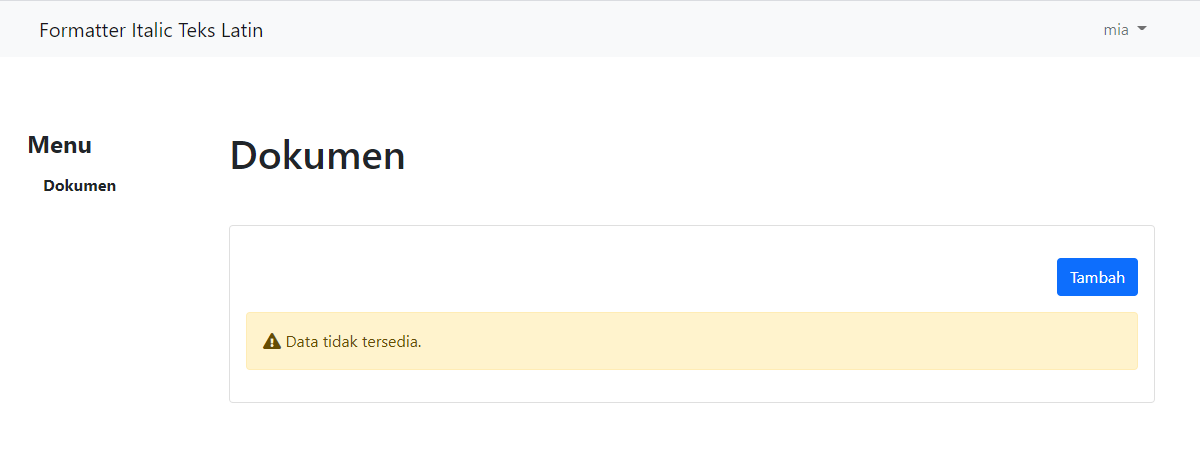
Halaman perbarui data kosa kata bahasa latin adalah halaman yang dapat diakses melalui dashboard admin. Ubah data kosa kata bahasa Latin digunakan oleh admin untuk memperbarui data kosa kata bahasa latin jika diperlukan. Tampilkan antarmuka halaman ubah data kosa kata bahasa latin dapat dilihat pada gambar 4.6



Gambar 4. 6 Halaman perbarui Data Kosa Kata Bahasa Latin

* + 1. **Tampilan Halaman Dashboard Mahasiswa**

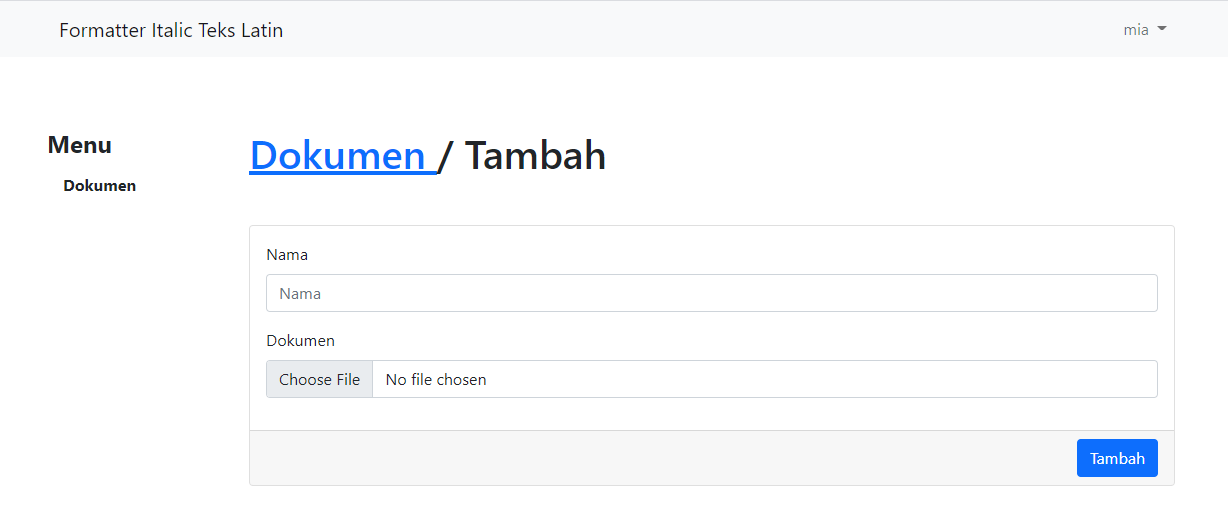
Halaman dashboard mahasiswa adalah halaman yang ditampilkan setelah user berhasil login sebagai mahasiswa. Halaman ini menampilkan seluruh dokumen yang pernah diunggah oleh mahasiswa yang bersangkutan ke aplikasi. Tampilan antarmuka halaman dashboard mahasiswa dapat dilihat pada gambar 4.7



Gambar 4.7 Tampilan Halaman Dashboard Mahasiswa

* + 1. **Tampilan Halaman Tambah Dokumen**

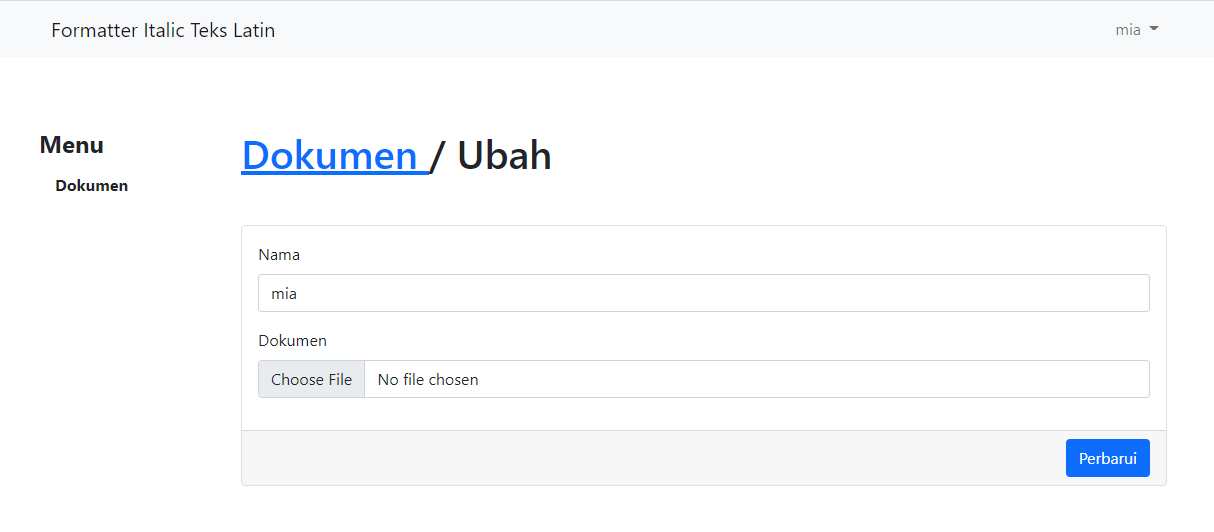
Halaman tambah dokumen adalah halaman yang dapat diakses melalui diakses dashboard mahasiswa. Halaman ini dapat digunakan oleh mahasiswa untuk mengunggah dokumen yang ingin diperiksa pada aplikasi. Tampilan antarmuka halaman tambah dokumen dapat dilihat pada gambar 4.8



Gambar 4. 8 Tampilan Halaman Tambah Dokumen

* + 1. **Tampilan Halaman Perbarui Dokumen**

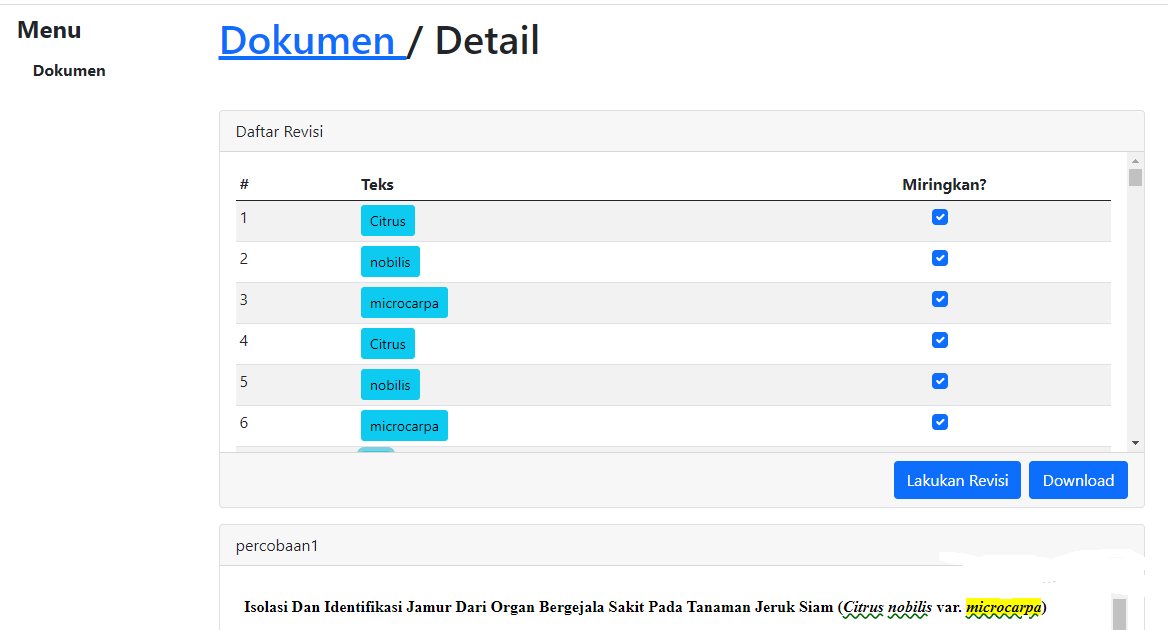
Halaman perbarui dokumen adalah halaman yang dapat diakses melalui dashboard mahasiswa. Halaman ini dapat digunakan oleh mahasiswa untuk mengubah data dokumen yang telah diunggah sebelumnya. Tampilan antarmuka halaman perbarui data dokumen dapat dilihat pada gambar 4.9



Gambar 4. 9 Tampilan Halaman Perbarui Dokumen

* + 1. **Tampilan Halaman Hasil Pemeriksaan Dokumen**

Halaman pemeriksaan dokumen adalah halaman yang menampilkan hasil proses memiringkan kata berbahasa Latin oleh aplikasi pada dokumen yang telah diunggah oleh mahasiswa. Tampilan halaman hasil pemeriksaan dokumen dapat dilihat pada gambar 4.10



Gambar 4. 10 Tampilan Halaman Hasil Pemeriksaan Dokumen

* 1. **Hasil Pengujian**
     1. **Pengujian *Black Box***

Pengujian *black box* bertujuan untuk menguji fungsionalitas dari sistem. Pengujian *black box* berdasarkan aksi dari pengguna. Sistem memiliki beberapa jenis aksi berupa tombol pada halaman detail berita website sehingga pengujian *black box* dilakukan dengan melihat hasil proses dari aksi yang dilakukan oleh pengguna.

* + - 1. **pengujian halaman Login**

**Tabel 4. 1** Hasil pengujian halaman Login

|  |  |  |  |
| --- | --- | --- | --- |
| **Fungsi** | **Contoh Fungsi** | **Hasil Eksekusi** | **Keterangan** |
| Fungsi *login* | Nama pengguna dan *password* kosong | Tidak berhasil | Muncul pesan, *“*Bidang isian wajib diisi.” |
| Nama pengguna salah dan *password* salah | Tidak berhasil | Muncul pesan, “Identitas tersebut tidak cocok dengan data kami.” |
| Nama pengguna dan *password* benar | Berhasil | Di arahkan ke halaman *Dashboard* |

* + - 1. **pengujian halaman Tambah Akun pengguna**

**Tabel 4. 2** Hasil pengujian Tambah Akun pengguna

| **Input** | **Contoh data** | | **Hasil eksekusi** | **Keterangan** |
| --- | --- | --- | --- | --- |
| Data kosong | Nama Asli |  | Tidak berhasil | Muncul Pesan “Bidang isian wajib diisi.” |
| Nama Pengguna |  |
| Kata Sandi |  |
| Salah satu data kosong | Nama Asli |  | Tidak berhasil | Muncul Pesan “Bidang isian wajib diisi.” |
| Nama Pengguna | Ahmad |
| Kata Sandi | \*\*\*\*\*\*\*\* |
| Data  Benar | Nama Asli | Ahmad Khusnan | Berhasil | Di arahkan ke halaman *Dashboard* |
| Nama Pengguna | Ahmad |
| Kata Sandi | \*\*\*\*\*\*\*\* |

* + - 1. **pengujian halaman hapus akun pengguna**

Tabel 4. 3 Hasil pengujian halaman hapus akun pengguna

| **Input** | **Contoh data** | | **Hasil eksekusi** | **Keterangan** |
| --- | --- | --- | --- | --- |
| Klik tombol “Hapus” |  |  | Berhasil | Muncul peringatan konfirmasi tindakan “Apakah Anda yakin ingin melakukan tindakan ini?” |

* + - 1. **pengujian halaman tambah dokumen**

Tabel 4. 4 pengujian halaman tambah dokumen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Contoh data** | | **Hasil eksekusi** | **Keterangan** |
| Data kosong | Nama |  | Tidak berhasil | Muncul Pesan “Bidang isian wajib diisi.” |
| Berkas |  |
| Data salah | Nama | Jurnal Mipa | Tidak berhasil | Muncul Pesan “Berkas harus dalam format .docx” |
| Berkas | File berekstensi .pdf |
| Data benar | Nama | Jurnal Mipa | Berhasil | Diarahkan ke halaman hasil pemeriksaan dokumen dan muncul pesan “Data berhasil ditambahkan.” |
| Berkas | File berekstensi .docx |

* + - 1. **pengujian halaman ubah dokumen**

Tabel 4. 5 Hasil pengujian halaman ubah dokumen

| **Input** | **Contoh data** | | **Hasil eksekusi** | **Keterangan** |
| --- | --- | --- | --- | --- |
| Klik tombol “Perbarui” |  |  | Berhasil | Muncul proses *loading* “Dokumen sedang diproses.” |

* + - 1. **pengujian halaman download dokumen**

Tabel 4. 6 Hasil pengujian halaman download dokumen

| **Input** | **Contoh data** | | **Hasil eksekusi** | **Keterangan** |
| --- | --- | --- | --- | --- |
| Klik tombol “Download” |  |  | Berhasil | Muncul *pop-up download file* di *browser* atau aplikasi *download manager* |

* + - 1. **pengujian halaman detail pemeriksaan dokumen**

Tabel 4. 7 Hasil pengujian halaman detail pemeriksaan dokumen

| **Input** | **Contoh data** | | **Hasil eksekusi** | **Keterangan** |
| --- | --- | --- | --- | --- |
| Klik tombol “Detail” |  |  | Berhasil | Diarahkan ke halaman hasil pemeriksaan dokumen |

* + - 1. **pengujian halaman hapus dokumen**

Tabel 4. 8 Hasil pengujian halaman hapus dokumen

| **Input** | **Contoh data** | | **Hasil eksekusi** | **Keterangan** |
| --- | --- | --- | --- | --- |
| Klik tombol “Hapus” |  |  | Berhasil | Muncul peringatan konfirmasi tindakan “Apakah Anda yakin ingin melakukan tindakan ini?” |

* + - 1. **pengujian halaman tambah kosa kata**

Tabel 4. 9 Hasil pengujian halaman tambah kosa kata

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Contoh data** | | **Hasil eksekusi** | **Keterangan** |
| Data kosong | Isi |  | Tidak berhasil | Muncul pesan “bidang isian wajib diiisi” |
| Data salah | Isi | Citrus, | Tidak berhasil | Muncul pesan “Format isi hanya boleh mengandung huruf” |
| Data benar | Isi | Citrus | Berhasil | Diarahkan ke halaman daftar kosa kata dan muncul pesan “Data berhasil ditambahkan.” |

* + - 1. **pengujian halaman ubah kosa kata**

**Tabel 4. 10** Hasil pengujian halaman ubah kosa kata

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Contoh data** | | **Hasil eksekusi** | **Keterangan** |
| Data kosong | isi |  | Tidak berhasil | Muncul pesan “bidang isian wajib diiisi” |
| Data salah | isi |  | Tidak berhasil | Muncul pesan “Format isi hanya boleh mengandung huruf” |
| Data benar | isi |  | Berhasil | Diarahkan ke halaman daftar kosa kata dan muncul pesan “Data berhasil diubah.” |

* + - 1. **pengujian halaman hapus kosa kata**

**Tabel 4. 11** Hasil pengujian halaman hapus kosa kata

| **Input** | **Contoh data** | | **Hasil eksekusi** | **Keterangan** |
| --- | --- | --- | --- | --- |
| Klik tombol “Hapus” |  |  | Berhasil | Muncul peringatan konfirmasi tindakan “Apakah Anda yakin ingin melakukan tindakan ini?” |

* + - 1. **pengujian halaman logout**

**Tabel 4. 12** Hasil pengujian halaman logout

| **Input** | **Contoh data** | | **Hasil eksekusi** | **Keterangan** |
| --- | --- | --- | --- | --- |
| Klik tombol “Keluar” |  |  | Berhasil | Diarahkan ke halaman Masuk |

* + 1. **Pengujian *Precision*, *Recall* dan *F-Measure***

Pengujian *precision*, *recall* dan *f-measure* bertujuan untuk mendapatkan informasi hasil pencarian dokumen yang relevan dengan dokumen asli yang ingin dibandingkan. Pengujian *precison*, *recall* dan *f-measure* dilakukan untuk melihat persamaan kata Latin yang tercetak miring pada kalimat yang ada dalam dokumen asli berupa jurnal yang telah terpublikasi dengan kata Latin yang tercetak miring hasil proses dari aplikasi. Jumlah kalimat yang diuji yaitu sebanyak 25 kalimat. Tabel hasil pengujian *precision, recall,* dan *f-measure* dapat dilihat pada Tabel 4.13. Untuk hasil lengkap pengujian dapat dilihat di lampiran A.

Tabel 4. 13 Hasil Pengujian *Precision,Recall, dan f-measure*

|  |  |  |  |
| --- | --- | --- | --- |
| **Kalimat ke-** | **(a)** | **(b)** | **(c)** |
| 1 | 3 | 0 | 0 |
| 2 | 1 | 0 | 0 |
| 3 | 3 | 0 | 0 |
| 4 | 1 | 0 | 0 |
| 5 | 1 | 0 | 0 |
| 6 | 1 | 0 | 0 |
| 7 | 2 | 0 | 0 |
| 8 | 1 | 0 | 0 |
| **Kalimat**  **Ke-** | **(a)** | **(b)** | **(c)** |
| 9 | 1 | 0 | 0 |
| 10 | 1 | 0 | 0 |
| 11 | 1 | 0 | 0 |
| 12 | 1 | 0 | 0 |
| 13 | 1 | 0 | 0 |
| 14 | 6 | 1 | 0 |
| 15 | 1 | 1 | 1 |
| 16 | 2 | 0 | 0 |
| 17 | 1 | 0 | 0 |
| 18 | 1 | 0 | 0 |
| 19 | 3 | 1 | 0 |
| 20 | 1 | 0 | 0 |
| 21 | 6 | 0 | 0 |
| 22 | 6 | 4 | 0 |
| 23 | 2 | 0 | 0 |
| 24 | 1 | 0 | 0 |
| 25 | 5 | 1 | 0 |
| **Total** | **53** | **8** | **1** |

|  |  |
| --- | --- |
| Precision **( )** | 0,981 |
| Recall **( )** | 0,869 |
| F-Measure **( 2. )** | 0,922 |

Keterangan :

Nilai a : jumlah kata Latin yang tercetak miring .

Nilai b : jumlah kata Latin yang tidak tercetak miring.

Nilai c : jumlah kata Latin yang penulisannya tidak perlu tercetak miring,namun menjadi tercetak miring.

Tabel 4.13 menunjukkan bahwa untuk memiringkan kata berbahasa Latin menghasilkan nilai *precision* sebesar 0.981, nilai *recall*  sebesar 0.869 dan 0.922 untuk *F-measure*.

* + 1. **Pengujian Durasi pada deteksi kata Latin**

Dari proses pencarian kata Latin tersebut dilakukan pengujian durasi untuk mengetahui berapa lama sistem mendeteksi kata Latin dengan menerapkan algoritma *boyer moore* pengujian ini dilakukan dengan menggunakan menguji sebuah dokumen pada sistem*.* Pada pengujian ini dilakukan dengan menggunakan 5 buah dokumen dimana setiap dokumen memiliki jumlah kata yang berbeda,dengan jumlah kosa kata Latin di database sebanyak 33313 kata. Sehingga menghasilkanHasil pengujian Durasi pada deteksi kata Latin dapat dilihat pada tabel 4. 14.

Tabel 4. 14 Hasil Pengujian Durasi pada Deteksi Kata Latin

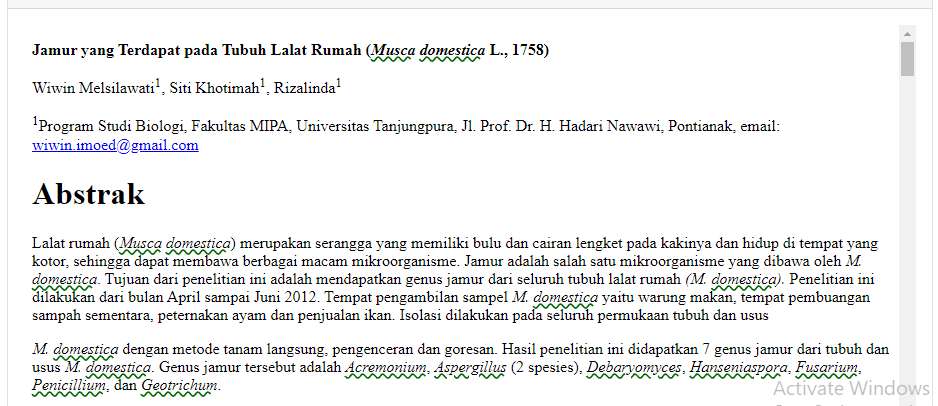
|  |  |  |  |
| --- | --- | --- | --- |
| No | Jumlah Kata | Waktu  (detik) | Jumlah kata Latin yang terdeteksi |
| 1 | 4.070 | 40 | 92 |
| 2 | 2.572 | 20 | 282 |
| 3 | 3.209 | 24 | 108 |
| 4 | 2.120 | 16 | 31 |
| 5 | 3.401 | 34 | 165 |
| Total | 15.372 | 134 | 678 |

Rata-Rata =

**=** = 0.11

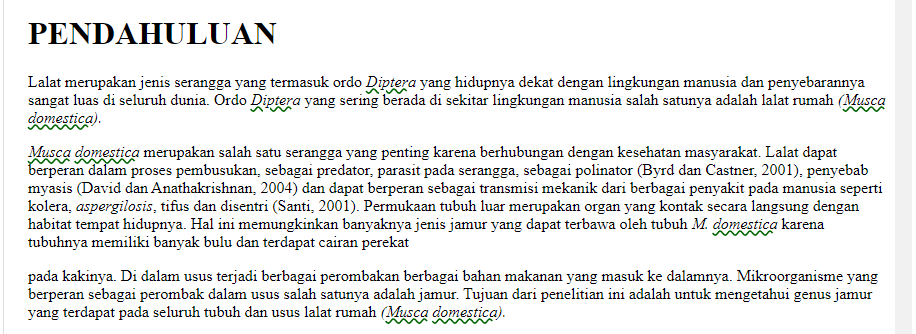
Dari tabel 4.14 menunjukan bahwa untuk menemukan kata berbahasa Latin dalam dokumen memiliki rata-rata membutuhkan waktu 0.11 detik perkata.

Berikut ini merupakan hasil *screenshoot* pengujian menggunakan sebuah dokumen pada sistem. Setiap kata yang bergaris bawah hijau merupakan kata Latin yang berhasil dideteksi dan dimiringkan hurufnya oleh sistem.



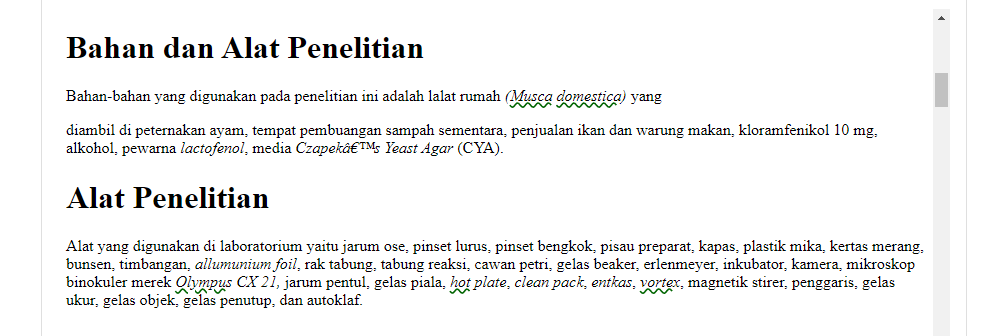
Gambar 4.11 hasil pengujian

Pada pengujian diatas kata Latin yeng berhasil dideteksi oleh sistem dan berhasil dicetak miring adalah *Musca Domestica, acremonium, aspergillus*, *Debaryomyces*, *Hanseniaspora*, *Fusarium*, *Penicillium*, dan *Geotrichum.*

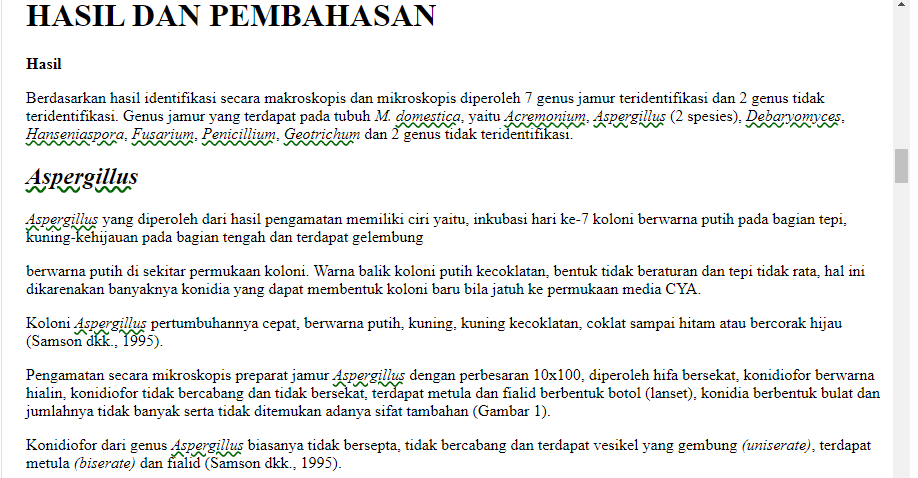


Gambar 4.12 hasil pengujian 1

Pada pengujian diatas kata Latin yeng berhasil dideteksi oleh sistem dan berhasil dicetak miring adalah *Diptera, musca domestika.*



Gambar 4. 13 hasil pengujian 2



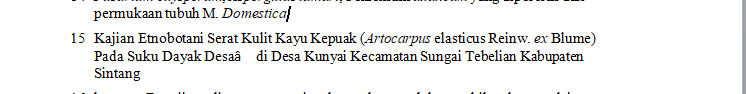
Gambar 4. 14 hasil pengujian 3

Pada pengujian diatas kata Latin yeng berhasil dideteksi oleh sistem dan berhasil dicetak miring adalah  *domestica, Acremonium, Aspergillus, Debaryomyces*, *Hanseniaspora*, *Fusarium*, *Penicillium*, *Geotrichum.*

* 1. **Analisis Hasil Pengujian**

Analisis hasil pengujian penerapan algoritma *boyer moore* pada aplikasi *font italic* untuk pendeteksi bahasa Latin dapat disimpulkan sebagai berikut :

1. Hasil pengujian menggunakan metode *black box* bahwa saat dilakukan input data salah satu data kosong atau keseluruhan data kosong, akan menampilkan notifikasi kesalahan pada aplikasi, sehingga kegagalan pada aplikasi dapat ditangani.
2. Berdasarkan Pengujian *precision*, *recall* dan *f-measure* dilakukan dengan menghitung tingkat akurasi kata Latin yang tercetak miring pada kalimat yang ada dalam dokumen asli berupa jurnal yang telah terpublikasi dengan kata Latin yang tercetak miring hasil proses dari aplikasi. Hasil pengujian dengan menghitung nilai *precision*, *recall* dan *f-measure* dilihat pada Tabel 4.13. Nilai *precision* dan *recall* yang tinggi karena data kata berbahasa latin yang sesuai dengan data training. Kalimat jurnal yang diinputkan sesuai dengan kata berbahasa Latin yang terdapat didatabase.
3. Berdasarkan pengujian durasi deteksi kata Latin pada tabel 4.14 mendapatkan hasil rata-rata sistem dalam mendeteksi kata Latin yaitu 0.11 detik perkata, lamanya sistem dalam mendeteksi kata Latin bergantung pada banyaknya jumlah kata dalam sebuah dokumen dan banyaknya kata Latin yang terdeteksi.
4. Berdasarkan pengujian yang telah dilakukan terdapat masalah dalam mencetak miring kata, yaitu terdapat beberapa aturan penamaan ilmiah yang tidak perlu dicetak miring dalam penulisannya.



Gambar 4. 15 Hasil pengujian Penulisan Yang seharusnya tidak miring

1. Proses implementasi algoritma *boyer moore* pada aplikasi *font italic* untuk mendeteksibahasa Latinsudah dapat dikatakan cukup berhasil dan dapat berjalan sesuai dengan yang diharapkan.

**BAB V**

# KESIMPULAN DAN SARAN

* 1. **Kesimpulan**

Berdasarkan hasil analisis dan pengujian terhadap penerapan algoritma *boyer moore* pada aplikasi *font italic* untuk mendeteksi bahasa Latin, maka dapat ditarik kesimpulan sebagai berikut:

1. Berdasarkan pengujian *black box*, aplikasi dapat menjalankan setiap proses yang ada pada aplikasi dengan baik sesuai dengan jenis aksi yang telah definisikan. Nilai valid pada setiap skenario menunjukkan bahwa fungsional pada sistem telah berjalan sesuai dengan yang diharapkan.
2. Berdasarkan pengujian *precision*, *recall* dan *f-measure*, dengan nilai *precision* sebesar 0,981, nilai *recall* sebesar 0,869 dan nilai *f-measure* sebesar 0,922. Nilai *precision* yang berada diatas 0,5 atau 50% dapat dinyatakan sudah efektif.
3. Berdasarkan hasil pengujian durasi pada deteksi kata Latin dengan algoritma *boyer moore,*  durasi bekerja sistem selain dipengaruhi oleh banyaknya jumlah kata yang ada pada teks, juga dipengaruhi oleh jumlah kata Latin yang berhasil di deteksi oleh sistem.
4. Berdasarkan hasil pengujian sistem,terdapat beberapa kata yang penulisannya tidak harus dicetak miring.
   1. **Saran**

Adapun beberapa hal yang perlu ditambahkan dalam pengembangan aplikasi ini agar menjadi lebih baik adalah sebagai berikut:

* 1. Aplikasi dikembangkan dengan deteksi kata dengan mengkobinasikan bahasa yang dideteksi,misalnya seperti bahasa Latin dengan bahasa inggris.
  2. Aplikasi dikembangkan dengan adanya aturan penamaan ilmiah dalam pendeteksian kata latin, seperti nama ilmiah mana yang harus dimiringkan dengan yang tidak perlu dimiringkan.
  3. Aplikasi dapat memperbaiki penulisan bahasa Latin.
  4. Aplikasi sangat bergatung pada kuantitas dan kualitas kosa kata bahasa

latin,sehingga diperlukan pembaharuan pada kosa kata bahasa Latin.

# DAFTAR PUSTAKA

A. S., Rosa dan Shalahuddin, M. 2013. *Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek*. Informatika. Bandung.

Borman, R. I., & Pratama, A. (2016). *Penerapan String Matching dengan Algoritma Boyer Moore pada Aplikasi Font Italic untuk Deteksi Kata Asing.* Bandar Lampung: Jurnal TEKNOINFO STMIK Teknorat.

Dwiputri, D. (2019*). Implementasi Fuzzy String Matching pada Aplikasi Font Italic untuk Deteksi Kata Bahasa Inggris*. Palembang: Universitas Sriwijaya.

Halim, Y. A. (2016). *IMPLEMENTASI BOYER-MOORE PADA APLIKASI PENCARIAN RUMUS MATEMATIKA DAN FISIKA.* Jakarta: Universitas Bunda Mulia.

Jogiyanto, H. (2005). Analisis dan Desain Sistem Informasi. Yogyakarta: Andi Offset.

Kristanto, A. (2008). *Perancangan Sistem Informasi dan Aplikasinya.* Yogyakarta: Gava Media.

Lisangan, E. A. (2013). *Natural Language Processing dalam memperoleh informasi akademik mahasiswa Universitas ATMAJAYA Makassar*. Makassar: Universitas Atmajaya.

Nugraha, A. T. (2014). Prediksi Jeda Dalam Ucapan Kalimat Bahasa Indonesia dengan Hidden Markov Model. *Jurnal Sistem dan Teknologi Informasi (JUSTIN)*, Vol 2, No.3.

*Penjelasan Pedoman Umum Ejaan Bahasa Indonesia (PEUBI*). http://badanbahasa.kemdikbud.go.id/lamanbahasa/sites/default/files/PUEBI.pdf. (Diakses 8 Januari 2021)

Rahmat Arfianto, “*Pengertian Flowchart Dan Jenis – Jenisnya*,” 2014. https://rahmatarifianto.wordpress.com/2014/11/20/pengertian-flowchart-dan-jenis-jenisnya/.

Rozi, I. F. (2012). Implementasi Opinion Mining (Analisis Sentimen) untuk Ekstraksi Data Opini Publik Pada Perguruan Tinggi. *Jurnal Teknik Elektro Universitas Brawijaya*

Sukamto, A., & Shalahuddin, M. (2014). *Rekayasa Perangkat Lunak Struktur dan Berorientasi Objek.* Bandung: Informatika.

Susanto, A. E. (2013). *Sejarah Bahasa Latin.* Magelang: SMA Seminari Mertoyudan.

Suyanto, M. (2004). *Aplikasi Desain Grafis Untuk Periklanan.* Yogyakarta: Andi Offset

Taufik, I., Aishia, I. D., & Jumadi. (2017). *Implementasi Fuzzy Search untuk Pendetaksi Kata Asing pada Dokumen Microsoft Word*. Bandung: Jurnal Informatika.

# LAMPIRAN A TABEL PERBANDINGAN

| No | Kalimat Asli  (jurnal) | Kalimat Uji | Kalimat  Sistem |
| --- | --- | --- | --- |
| 1. | Isolasi Dan Identifikasi Jamur Dari Organ Bergejala Sakit Pada Tanaman Jeruk Siam (*Citrus nobilis* var*. microcarpa*) | Isolasi Dan Identifikasi Jamur Dari Organ Bergejala Sakit Pada Tanaman Jeruk Siam (Citrus nobilis var. microcarpa) | Isolasi Dan Identifikasi Jamur Dari Organ Bergejala Sakit Pada Tanaman Jeruk Siam (*Citrus* *nobilis* var. *microcarpa*) |
| 2. | jenis-jenis jamur yang dapat diisolasi dari organ sakit pada tanaman jeruk siam (*C. nobilis*) | jenis-jenis jamur yang dapat diisolasi dari organ sakit pada tanaman jeruk siam (C. nobilis) | jenis-jenis jamur yang dapat diisolasi dari organ sakit pada tanaman jeruk siam (C. *nobilis*) |
| 3. | Gejala sakit pada tanaman jeruk yang ditimbulkan oleh jamur mulai terlihat pada umur yang berbeda yaitu *Fusarium* sp, *Phytopthtora* sp, dan *Diploidia* sp | Gejala sakit pada tanaman jeruk yang ditimbulkan oleh jamur mulai terlihat pada umur yang berbeda yaitu Fusarium sp, Phytopthtora sp, dan Diploidia sp | Gejala sakit pada tanaman jeruk yang ditimbulkan oleh jamur mulai terlihat pada umur yang berbeda yaitu *Fusarium* sp, *Phytopthtora* sp, dan *Diploidia* sp |
| 4. | Jamur *Elsinoe* sp ditemukan pada organ yang menunjukkan gejala penyakit kudis pada tanaman jeruk | Jamur Elsinoe sp ditemukan pada organ yang menunjukkan gejala penyakit kudis pada tanaman jeruk | Jamur *Elsinoe* sp ditemukan pada organ yang menunjukkan gejala penyakit kudis pada tanaman jeruk  . |
| 5. | Penyakit embun jalaga disebabkan oleh jamur *Capnodium* sp | Penyakit embun jalaga disebabkan oleh jamur Capnodium sp | Penyakit embun jalaga disebabkan oleh jamur *Capnodium* sp |
| 6. | Penyakit kudis disebabkan oleh jamur *Sphaceloma* sp (Semangun, 2000) | Penyakit kudis disebabkan oleh jamur Sphaceloma sp (Semangun, 2000) | Penyakit kudis disebabkan oleh jamur *Sphaceloma* sp (Semangun, 2000) |
| 7. | Jamur yang Terdapat pada Tubuh Lalat Rumah (*Musca domestica* L., 1758) | Jamur yang Terdapat pada Tubuh Lalat Rumah (Musca domestica L., 1758) | Jamur yang Terdapat pada Tubuh Lalat Rumah (*Musca* *domestica* L., 1758) |
| 8. | Tujuan dari penelitian ini adalah mendapatkan genus jamur dari seluruh tubuh lalat rumah *(M. Domestica)* | Tujuan dari penelitian ini adalah mendapatkan genus jamur dari seluruh tubuh lalat rumah (M. Domestica) | Tujuan dari penelitian ini adalah mendapatkan genus jamur dari seluruh tubuh lalat rumah (M. *domestica*). |
| 9. | Genus jamur tersebut adalah *Acremonium* | Genus jamur tersebut adalah Acremonium | Genus jamur tersebut adalah *Acremonium* |
| 10. | Lalat merupakan jenis serangga yang termasuk ordo *Diptera* | Lalat merupakan jenis serangga yang termasuk ordo Diptera | Lalat merupakan jenis serangga yang termasuk ordo *Diptera* |
| 11 | Pengamatan secara mikroskopis preparat jamur *Geotrichum* sp | Pengamatan secara mikroskopis preparat jamur Geotrichum sp | Pengamatan secara mikroskopis preparat jamur *Geotrichum* sp |
| 12 | genus *Schizosaccharomyces* reproduksinya dengan cara pembelahan sel | genus Schizosaccharomyces reproduksinya dengan cara pembelahan sel | genus *Schizosaccharomyces* reproduksinya dengan cara pembelahan sel |
| 13 | Pengamatan secara mikroskopis preparat jamur *Aspergillus* dengan perbesaran | Pengamatan secara mikroskopis preparat jamur Aspergillus dengan perbesaran | Pengamatan secara mikroskopis preparat jamur *Aspergillus* dengan perbesaran |
| 14 | *Fusarium oxysporum, Aspergillus tamari, Penicillum axalicum* yang diperoleh dari permukaan tubuh *M. Domestica* | Fusarium oxysporum, Aspergillus tamari, Penicillum axalicum yang diperoleh dari permukaan tubuh M. Domestica | *Fusarium* *oxysporum*, *Aspergillus* *tamari*, Penicillum *axalicum* yang diperoleh dari permukaan tubuh M. *Domestica* |
| 15 | Kajian Etnobotani Serat Kulit Kayu Kepuak (*Artocarpus elasticus* Reinw*.* ex Blume) Pada Suku Dayak Desa’di Desa Kunyai Kecamatan Sungai Tebelian Kabupaten Sintang | Kajian Etnobotani Serat Kulit Kayu Kepuak (Artocarpus elasticus Reinw. ex Blume) Pada Suku Dayak Desa’di Desa Kunyai Kecamatan Sungai Tebelian Kabupaten Sintang | Kajian Etnobotani Serat Kulit Kayu Kepuak (*Artocarpus* elasticus Reinw. *ex* Blume) Pada Suku Dayak Desaâ™di Desa Kunyai Kecamatan Sungai Tebelian Kabupaten Sintang |
| 16 | kampus Brawijaya dimana renovasi gedung rektorat telah menghilangkan serak jawa (*Tyto alba*) | kampus Brawijaya dimana renovasi gedung rektorat telah menghilangkan serak jawa (Tyto alba) | kampus Brawijaya dimana renovasi gedung rektorat telah menghilangkan serak jawa (*Tyto* *alba*) |
| 17 | akses keluar masuk *T. alba* untuk memelihara anak-anaknya | akses keluar masuk T. alba untuk memelihara anak-anaknya | akses keluar masuk T. *alba* untuk memelihara anak-anaknya |
| 18 | *Passeriformes* adalah ordo dengan jumlah jenis dan jumlah famili paling banyak dijumpai | Passeriformes adalah ordo dengan jumlah jenis dan jumlah famili paling banyak dijumpai | *Passeriformes* adalah ordo dengan jumlah jenis dan jumlah famili paling banyak dijumpai |
| 19 | Dua jenis burung yang termasuk dalam kategori melimpah yaitu *Collocalia fuciphaga* dan *Passer montanus* | Dua jenis burung yang termasuk dalam kategori melimpah yaitu Collocalia fuciphaga dan Passer montanus | Dua jenis burung yang termasuk dalam kategori melimpah yaitu Collocalia *fuciphaga* dan *Passer* *montanus* |
| 20 | Jenis *P. goiavier* adalah jenis yang umum ditemukan disemua lokasi penelitian | Jenis P. goiavier adalah jenis yang umum ditemukan disemua lokasi penelitian | Jenis P. *goiavier* adalah jenis yang umum ditemukan disemua lokasi penelitian |
| 21 | Jenis yang umum ditemukan di lokasi X yaitu *Muscicapa dauurica, Pycnonotus goiavier,* dan *Zosterops palpebrosus* | Jenis yang umum ditemukan di lokasi X yaitu Muscicapa dauurica, Pycnonotus goiavier, dan Zosterops palpebrosus | Jenis yang umum ditemukan di lokasi X yaitu *Muscicapa* *dauurica*, *Pycnonotus* *goiavier*, dan *Zosterops* *palpebrosus* |
| 22 | frekuensi tertinggi (100%), yaitu: *Passer montanus, Pycnonotus aurigaster, Pycnonotus goiavier, Collocalia fuciphaga,* dan *Anthreptes malacensis* | frekuensi tertinggi (100%), yaitu: Passer montanus, Pycnonotus aurigaster, Pycnonotus goiavier, Collocalia fuciphaga, dan Anthreptes malacensis | frekuensi tertinggi (100%), yaitu: *Passer* *montanus*, *Pycnonotus* aurigaster, *Pycnonotus* *goiavier*, Collocalia *fuciphaga*, dan Anthreptes malacensis |
| 23 | *Centropus sinensis* memiliki kombinasi pakan yang paling beragam | Centropus sinensis memiliki kombinasi pakan yang paling beragam | *Centropus* *sinensis* memiliki kombinasi pakan yang paling beragam |
| 24 | *Columbidae* memiliki jumlah jenis yang paling banyak di antara famili lainnya | Columbidae memiliki jumlah jenis yang paling banyak di antara famili lainnya | *Columbidae* memiliki jumlah jenis yang paling banyak di antara famili lainnya |
| 25 | jenis yang paling sedikit dijumpai adalah Paruh-kodok Tanduk (*Batrachostomus* *cornutus*), Bubut Besar (*Centropus* *sinensis*), dan Opior Jawa (*Lophozosterops javanicus*) | jenis yang paling sedikit dijumpai adalah Paruh-kodok Tanduk (Batrachostomus cornutus), Bubut Besar (Centropus sinensis), dan Opior Jawa (Lophozosterops javanicus) | jenis yang paling sedikit dijumpai adalah Paruh-kodok Tanduk (*Batrachostomus* *cornutus*), Bubut Besar (*Centropus* *sinensis*), dan Opior Jawa (Lophozosterops *javanicus*) |