

**PEMBUATAN APLIKASI *KOREKSI OTOMATIS*
MENGUNAKAN ALGORITMA
LEVENSHTEIN DISTANCE**

TUGAS AKHIR

Diajukan untuk Memenuhi Salah Satu Syarat Kelulusan
Program Pendidikan Sarjana

Oleh
Alfaro Johnatan Herawan
2017130001



**JURUSAN INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER–LIKMI
BANDUNG
2022**

**PEMBUATAN APLIKASI *KOREKSI OTOMATIS*
MENGUNAKAN ALGORITMA
LEVENSHTEIN DISTANCE**

Oleh
Alfaro Johnatan Herawan
2017130001

Bandung, 12 Juli 2022
Menyetujui,

Maria Christina, S.Kom., M.Kom.
Ko-Pembimbing

Dhanny Setiawan, S.T.,M.T.
Pembimbing

Dhanny Setiawan, S.T.,M.T.
Ketua Jurusan

**JURUSAN INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER-LIKMI
BANDUNG
2022**

ABSTRAK

Pada tahun 2022, penggunaan algoritma sering digunakan dalam kehidupan sehari-hari seperti menyelesaikan permasalahan perhitungan, membuat simulasi, membuat kemungkinan yang akan terjadi, menyelesaikan perhitungan yang rumit ataupun menyederhanakan sebuah rumus yang rumit. Dalam dunia komputer, algoritma merupakan instruksi-instruksi yang digunakan untuk memecahkan suatu permasalahan. Salah satunya adalah algoritma *auto correct*, yang memiliki fungsi untuk memprediksi kata yang benar dari sebuah kata.

Auto correct merupakan fitur yang berfungsi sebagai pengoreksi kata secara otomatis jika pengguna melakukan kesalahan pengetikkan. Hal ini diperlukan agar pengguna dapat mengurangi waktu dalam melakukan pengoreksian kata secara manual. Sehingga dibuatlah penelitian berjudul "PEMBUATAN APLIKASI KOREKSI OTOMATIS MENGGUNAKAN ALGORITMA LEVENSTEIN DISTANCE" yang memiliki fitur koreksi otomatis, menyimpan data, memilih kata yang direkomendasikan dan memilih kata yang diprediksi selanjutnya.

Hal yang terdapat dalam penelitian ini adalah penggunaan diagram seperti *Use Case Diagram*, *Sequence Diagram* dan *Class Diagram*. Serta lampiran pengujian aplikasi di berbagai browser.

Kata kunci : *Auto correction*, Algoritma Levenstein Distance, Website, Diagram

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas segala rahmat dan berkat-Nya sehingga penulis dapat menyelesaikan laporan penelitian yang berjudul “Pembuatan Aplikasi Koreksi Otomatis Menggunakan Algoritma Levenstein Distance” ini tepat pada waktunya.

Adapun tujuan dari penulisan dari makalah ini adalah untuk memenuhi tugas akhir pada bidang studi Teknik Informatika di Sekolah Tinggi Manajemen Informatika dan Komputer LIKMI, Bandung. Selain itu, makalah ini juga bertujuan untuk menjelaskan bagaimana cara kerja aplikasi yang telah dibuat kepada para pembaca.

Dalam penyusunan penelitian ini, penulis menyadari bahwa tidak lepas bantuan dan dorongan baik secara langsung ataupun tidak langsung dari berbagai pihak sehingga penulis dapat menyelesaikan tugas akhir. Untuk itu penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada yang terhormat:

1. Bapak Dhanny Setiawan, S.T.,M.T. selaku dosen pembimbing dan ketua jurusan program studi S1 Informatika di STMIK LIKMI yang telah memberi dukungan dan arahan kepada penulis.
2. Ibu Maria Christina, S.Kom., M.Kom. selaku dosen ko-pembimbing yang sudah meluangkan waktu, tenaga dan pikiran untuk membimbing penulis dalam proses pembuatan tugas akhir ini.
3. Orang tua serta keluarga yang telah membantu dalam mendukung dan memberi motivasi selama penulis menyelesaikan tugas akhir ini.
4. STMIK LIKMI yang telah memberikan kesempatan bagi penulis untuk menempuh pendidikan serta seluruh dosen STMIK LIKMI yang telah memberikan ilmu dan pengarahannya untuk memberikan pengajaran yang terbaik.
5. Seluruh teman dan sahabat yang selalu mendukung, membantu dan memberikan motivasi dalam menyelesaikan tugas akhir.
6. Semua pihak yang terlibat secara tidak langsung dalam penyusunan tugas akhir ini dimana penulis tidak dapat menyebutkannya satu-persatu.

Akhir kata dari penulis, penulis meminta maaf apabila terdapat kesalahan dalam penulisan makalah. Oleh karena, penulis mengharapkan segala bentuk kritik dan saran agar pembuatan makalah dapat lebih baik lagi. Sekian dari saya terimakasih telah membaca.

Bandung, 12 Juli 2022

Penulis

DAFTAR ISI

ABSTRAK.....	i
KATA PENGANTAR	ii
DAFTAR ISI.....	iv
DAFTAR GAMBAR	vi
DAFTAR TABEL	vii
DAFTAR SIMBOL	viii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Batasan Masalah	2
1.5 Kegunaan Hasil	2
1.6 Sistematika Penulisan.....	3
BAB II LANDASAN TEORI.....	4
2.1 Rekayasa Perangkat Lunak.....	4
2.1.1 Definisi Rekayasa Perangkat Lunak.....	4
2.1.2 Metodologi Penelitian.....	4
2.2 <i>Autocorrect</i>	6
2.3 Levenshtein Distance.....	7
2.4 Visual Studio Code.....	8
2.5 Website	8
2.6 Bahasa Pemrograman	9
2.6.1 Javascript	9
2.6.2 CSS	10
2.6.3 JQuery.....	10
2.6.4 PHP	10
2.7 XAMPP	11
2.8 MYSQL.....	11
2.9 JSON.....	11
2.10 UML.....	12
BAB III ANALISA DAN PERANCANGAN PERANGKAT LUNAK	17
3.1 Gambaran Umum Penggunaan <i>Auto correct</i>	17
3.2 Spesifikasi Kebutuhan Perangkat Lunak dan Perangkat Keras untuk Aplikasi	18
3.3 Use Case Diagram.....	19

3.4	Skenario Use Case	19
3.4.1	Skenario <i>Auto correct</i>	19
3.4.2	Skenario Mengetikan Kata	21
3.4.3	Skenario Menyimpan Data.....	22
3.5	Activity Diagram	23
3.6	Class Diagram.....	25
3.7	Perancangan Antar Muka	26
BAB IV Perancangan Antar Muka		27
4.1	Tampilan Antar Muka	27
4.2	Pengujian Fungsional.....	30
4.3	Pengujian Kompabilitas.....	31
BAB V Kesimpulan dan Saran		34
5.1	Kesimpulan.....	34
5.2	Saran.....	34
DAFTAR PUSTAKA		35
LAMPIRAN PENGUJIAN		38
Lampiran 1 Pengujian Fungsional		38
Lampiran 2 Pengujian Kompabilitas		44
Lampiran 3 Listing Program		63

DAFTAR GAMBAR

Gambar 3. 1 Use case diagram auto correct	19
Gambar 3. 2 Activity diagram mengetikan kata	23
Gambar 3. 3 Activity diagram menyimpan data	24
Gambar 3. 4 Class diagram mengetikan kata.....	25
Gambar 3. 5 Rancangan antarmuka tampilan awal atau default.....	26
Gambar 4. 1 Tampilan Awal.....	27

DAFTAR TABEL

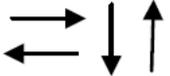
Tabel 3. 1 Skenario normal auto correct.....	19
Tabel 3. 2 Skenario alternatif auto correct.....	20
Tabel 3. 2 Skenario alternatif auto correct.....	20
Tabel 3. 3 Skenario normal mengetikan kata	21
Tabel 3. 4 Skenario alternatif mengetikan kata	21
Tabel 3. 5 Skenario normal menyimpan data	22
Tabel 3. 6 Skenario alternatif menyimpan data	22
Tabel 4. 1 Tabel Tampilan Antar Muka.....	28
Tabel 4. 2 Tabel Pengujian Fungsional	30
Tabel 4. 3 Tabel Pengujian Windows 10	31
Tabel 4. 4 Tabel Pengujian Ubuntu dan Android.....	32

DAFTAR SIMBOL

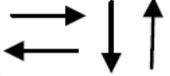
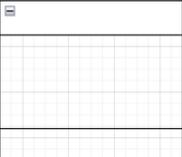
Use Case Diagram

Nama Simbol	Simbol	Keterangan
Use Case		Sebuah kebiasaan atau fungsi utama (<i>key behaviour</i>) yang dilakukan oleh sistem perangkat lunak
Asosiasi		Hubungan antara aktor dengan <i>use case</i> yang menunjukkan adanya interaksi aktor dengan sistem atau campur tangan aktor dalam suatu <i>use case</i>
Aktor		Entitas eksternal yang berhubungan dengan sistem

Acitivity Diagram

Nama Simbol	Simbol	Keterangan
Kondisi Awal		Menunjukkan kondisi awal kegiatan sistem
Kondisi Akhir		Menunjukkan kondisi berakhirnya kegiatan sistem
Action State		Kegiatan yang sistem lakukan
Join		Kegiatan yang digabungkan
Decision		Percabangan dimana sistem harus melakukan salah satu cabang keputusan
Arah aktivitas		Menunjukkan arah sebuah <i>action state</i> ke <i>action state</i> selanjutnya

Class Diagram

Nama Simbol	Simbol	Keterangan
Asosiasi		Menunjukkan relasi yang mengindikasikan bahwa suatu kelas mereferensikan kelas yang lain.
Class		Menunjukkan komponen atas yang berisi nama kelas, komponen tengah untuk mendeskripsikan kualitas kelas, dan komponen bawah yang menyertakan operasi kelas yang ditampilkan dalam format daftar.

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pada tahun 2022, penggunaan algoritma sering digunakan dalam kehidupan sehari-hari seperti menyelesaikan permasalahan perhitungan, membuat simulasi, membuat kemungkinan yang akan terjadi, menyelesaikan perhitungan yang rumit ataupun menyederhanakan sebuah rumus yang rumit. Dalam dunia komputer, algoritma merupakan instruksi-instruksi yang digunakan untuk memecahkan suatu permasalahan. Salah satunya adalah algoritma *auto correct*, *auto correct* berguna untuk memperbaiki kata-kata menjadi lebih baik, contoh aplikasi yang menggunakan fitur *auto correct* adalah Microsoft dan aplikasi android lainnya yang memiliki fitur *auto correct*.

Alasan diperlukan *auto correct* adalah karena sering terjadinya kesalahan pengetikkan dalam melakukan pencarian seperti mencari nama obat yang penyebutannya sulit dan jika terjadi kesalahan pengetikkan, pencarian kata pun menjadi sulit dan akan memakan banyak waktu dan bersikap fatal (Saidah, Ishlah and Rokhmah, 2020). Fungsi lainnya yaitu dapat membantu pengguna untuk memperbaiki kata ketika menggunakan bahasa asing, sehingga salah pengetikkan seperti kosa kata dapat diminimalkan (Wangi and Rosalin, 2019).

Untuk membuat fitur *auto correct* ini, menggunakan bahasa pemrograman PHP dengan algoritma Levenshtein. PHP adalah bahasa pemrograman bagian server yang akan diproses di server untuk pengembangan web. Algoritma Levenshtein digunakan di proyek ini untuk mencari perbedaan antara kata yang dimasukkan oleh pengguna dengan kata yang ada pada basis data.

Pembuatan fitur *auto correct* ini dilakukan karena sering terjadinya kesalahan penulisan sebuah kata sehingga sering terjadi pengulangan ketikan. Karena hal tersebut, waktu sering terbuang karena harus melakukan pengecekan ulang setiap kata atau melakukan penghapusan karakter agar menjadi kata yang benar. Dengan program ini,

diharapkan waktu yang digunakan untuk melakukan pengecekan tiap kata dapat dikurangi dan digantikan dengan penggunaan algoritma ini.

1.2 Rumusan Masalah

Dari uraian di atas, dapat ditentukan dalam berbagai masalah seperti

1. Bagaimana mengimplementasi algoritma Levenshtein pada aplikasi?
2. Bagaimana mengimplementasikan aplikasi koreksi otomatis secara online?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah

1. Memperbaiki pengetikan kata yang salah.
2. Menghemat waktu untuk perbaikan penulisan.
3. Memberikan masukkan kata sesuai yang diketik.
4. Menampilkan kata yang mirip dengan yang diketik.
5. Aplikasi digunakan secara online.

1.4 Batasan Masalah

Terdapat batasan-batasan masalah dalam program antara lain

1. Program ini hanya menggunakan bahasa Indonesia.
2. Kata baru perlu dilakukan pengujian ulang.
3. Kamus data adalah hasil input dari penulis.
4. Dalam pengujian ini menggunakan 1000 kata.
5. Tidak ada pengecekan terhadap kata yang salah.

1.5 Kegunaan Hasil

Kegunaan hasil dari aplikasi perangkat lunak ini:

1. Memudahkan pengguna untuk memperbaiki penulisan.
2. Memudahkan pengguna untuk menghemat waktu untuk perbaikan penulisan.

Kegunaan hasil dari karya tulis tugas akhir ini:

1. Menjelaskan cara kerja koreksi otomatis.

1.6 Sistematika Penulisan

1. BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, tujuan penelitian, batasan masalah, kegunaan hasil, dan sistematika penelitian.

2. BAB II LANDASAN TEORI

Bab ini membahas tentang definisi rekayasa perangkat lunak, metodologi penelitian, model pengembangan, penjelesan tentang *auto correction*, bahasa pemrograman yang digunakan, penjelasan tentang diagram-diagram UML dan perangkat lunak yang digunakan untuk membuat aplikasi.

3. BAB III ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tentang gambaran umum perangkat lunak, gambaran umum simulasi, spesifikasi kebutuhan perangkat lunak, penggunaan algoritma, perancangan *Use Case Diagram*, perancangan Skenario *Use Case*, perancangan *Activity Diagram*, perancangan *Class Diagram* dan perancangan antar muka.

4. BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini membahas tentang spesifikasi kebutuhan perangkat keras, pengujian antarmuka dan pengujian fungsi.

5. BAB V KESIMPULAN

Bab ini membahas tentang kesimpulan dan saran dari penulis untuk tujuan penelitian selanjutnya

BAB II

LANDASAN TEORI

2.1 Rekayasa Perangkat Lunak

2.1.1 Definisi Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah penerapan pendekatan secara sistematis dan disiplin, serta dapat melakukan pengukuran untuk pengembangan, pengoperasian, dan pemeliharaan perangkat lunak (Fairley, 2019).

Sementara menurut Miroslaw Staron rekayasa perangkat lunak memfokuskan pada aspek sistematis dalam cara mengembangkan, merancang, menguji, dan memelihara produk perangkat lunak (Staron, 2020).

Kesimpulan yang dari definisi yang ada, dapat disimpulkan bahwa rekayasa perangkat lunak adalah pendekatan yang memfokuskan pada aspek sistematis, disiplin untuk pengembangan, merancang, menguji dan memelihara perangkat lunak.

2.1.2 Metodologi Penelitian

Dalam pembuatan rekayasa perangkat lunak metode penelitian yang digunakan berdasarkan *object oriented*. Dasar-dasar *object oriented* memberikan landasan teoritis yang kuat untuk analisis, desain, arsitektur, pengkodean, dan pengujian sistem perangkat lunak (Unhelkar, 2017). Hal-hal yang mendasari *object oriented* adalah:

1. *Object*

Menurut Bhuvan Unhelkar pada buku *Software Engineering with UML*, objek merupakan bagian dari kelas yang terdiri dari data dan perilaku yang terkait dengan data pada waktu proses (Unhelkar, 2017). Dan menurut Vaskaran Sarcar pada buku *Interactive Object-Oriented Programming in Java*, objek merupakan bagian dari sebuah kelas dan setiap objek memiliki status, perilaku, dan identitasnya sendiri (Sarcar, 2020).

Hasil yang dapat disimpulkan adalah objek merupakan bagian dari kelas yang mempunyai ciri khas tersendiri seperti status, perilaku atau data tertentu.

2. *Class*

Kelas merupakan bagian dari program perangkat lunak yang merangkum data dengan *function* atau *method* dengan cara yang berbeda untuk membentuk sistem (Sarcar, 2020). Kelas adalah wadah berisi pemodelan suatu objek, mendeskripsikan karakteristik dan objek tersebut (Retnoningsih, Shadiq and Oscar, 2017). Jadi kelas adalah bagian dari program perangkat lunak yang berisi pemodelan suatu objek dan merangkum data dengan *function* atau *method* untuk membentuk sistem yang mendeskripsikan karakteristik dan objek tersebut.

3. *Encapsulation*

Menurut Unhelkar, dalam pemrograman berorientasi objek, penggunaan perizinan untuk kelas sangat diperlukan. Tujuan enkapsulasi setidaknya salah satu seperti menempatkan batasan-batasan pada komponen-komponen suatu object agar tidak dapat diakses secara langsung atau mempengaruhi data dengan metode yang bertindak atas data tersebut. (Unhelkar, 2017)

Dan enkapsulasi mempunyai fungsi untuk menyatukan potongan data yang saling terhubung dengan kode yang lebih berarti (Sarcar, 2020). Sehingga enkapsulasi dapat mengatur batasan pada sebuah *class* agar hanya bisa mengakses objek yang diberi izin pada *class* tersebut.

4. *Abstraction*

Objek yang merupakan entitas asli, perlu diwakili oleh template untuk menentukan karakteristik dan perilakunya. Kumpulan objek yang diklasifikasi akan diabstraksi pada sebuah kelas. Selanjutnya kelas memberikan definisi mendetail dari semua objek untuk dibuat penjelasannya (Unhelkar, 2017). Tujuan utama dari abstraksi adalah untuk hanya menampilkan fitur-fitur penting dan untuk menyembunyikan detail latar belakang implementasi (Sarcar, 2020). Sehingga abstraksi dapat disimpulkan sebagai sebuah penjelasan yang lebih umum dari sebuah kelas agar lebih mudah dimengerti.

5. *Inheritance*

Dalam dunia pemrograman, *inheritance* membuat kelas anak dari kelas induk yang sudah ada. Kelas anak dapat membuat method baru ataupun dapat menggunakan method yang sudah ada dari kelas induknya (Unhelkar, 2017). Kelas di *object oriented* juga berhubungan satu sama lain melalui *inheritance*. Hasil *inheritance* dari kelas yang digeneralisasikan menjadi kelas tingkat yang lebih tinggi atau kelas abstrak (Sarcar, 2020). Sehingga *inheritance* dapat disimpulkan sebagai sebuah penurunan kelas dari kelas induk dimana kelas yang diturunkan mewarisi identitas kelas induknya.

6. *Polymorphism*

Polymorphism adalah kemampuan objek yang dipakai pada saat runtime untuk memahami dan menafsirkan pesan yang dikirim dari objek pemanggil. Pengertian pesan dari suatu objek bergantung pada karakteristik dan definisinya sendiri (Unhelkar, 2017).

Untuk jenisnya *polymorphism* dapat terdiri dari dua jenis yaitu (Sarcar, 2020) *Compile-time polymorphism (Overloading)* dengan fungsi compiler dapat memutuskan sejak awal method mana yang akan dipanggil sesuai situasi yang dibutuhkan dan *Runtime polymorphism (Overriding)* yang berfungsi ketika waktu di kompilasi, method yang dipilih tidak dapat ditentukan langsung. Tergantung method tersebut digunakan oleh apa, sehingga untuk mengetahui hasilnya harus menunggu program dieksekusi.

2.2 **Autocorrect**

Autocorrect adalah suatu sistem yang dapat memeriksa dan memperbaiki kesalahan penulisan kata secara otomatis (Prasetyo, Baihaqi and Had, 2018). *Autocorrect* dapat ditemukan dari berbagai aplikasi seperti Microsoft Word ataupun keyboard yang ada di android. *Autocorrect* adalah fungsi perangkat lunak yang biasanya banyak ditemukan di *smartphone* dan perangkat *web portable* seperti iPod Touch (Madison, 2012). Jadi *autocorrect* adalah sistem perangkat lunak yang berfungsi untuk memeriksa dan

memperbaiki kesalahan penulisan secara otomatis dan banyak ditemukan di *smartphone* dan perangkat *web portable* seperti iPod Touch.

2.3 Levenshtein Distance

Algoritma Levenshtein Distance merupakan algoritma yang dibuat pada tahun 1965 oleh Vladimir Levenshtein (Saidah, Ishlah and Rokhmah, 2020). Algoritma ini berfungsi untuk mencari perbedaan antara kata yang dimasukkan oleh pengguna dengan kata yang ada pada basis data. Perhitungan dilakukan dengan menggunakan tabel matriks, untuk panjang matriks sendiri akan disesuaikan dengan panjang kata yang akan dibandingkan. Untuk cara kerjanya, algoritma Levenshtein Distance menghitung jarak antara kedua string kemudian jumlah minimum akan dicari. Nilai yang ada di pojok kanan bawah merupakan nilai akhir dari jarak kedua String. Semakin kecil nilai tersebut, artinya kedua string tersebut memiliki kesamaan yang semakin besar. Pada Algoritma Levenshtein Distance terdapat tiga macam operasi yaitu operasi pengubahan karakter, penambahan karakter dan penghapusan karakter.

Untuk pseudo-code dapat dilihat dibawah ini.

```
Error ← jumlah error
N ← panjang kata input
M ← panjang kata basis data
If N <= M
    For i ← 1 sampai N
        If input[i] != basis data[i]
            Error++
    Error += M - N
Else
    For i ← 1 sampai M
        If input[i] != basis data[i]
            Error++
    Error += N - M
```

Untuk variabel error memiliki fungsi sebagai penanda berapa banyak jumlah karakter yang berbeda jika dibandingkan, sementara variabel input memiliki arti kata yang diinput oleh pengguna lalu diubah menjadi per karakter begitu juga dengan variabel basis data. Perubahan kata menjadi karakter memiliki fungsi untuk mempermudah melakukan pengecekan tiap karakter. Selanjutnya algoritma akan membandingkan panjang huruf input dengan huruf yang berada di basis data. Jika panjang huruf input lebih panjang dari huruf basis data, maka sistem akan mengulang berdasarkan panjang huruf input yang selanjutnya akan membandingkan huruf yang jika salah maka akan menambah error dan sisa huruf yang berlebihan akan dihitung sebagai error. Sementara jika panjang huruf basis data lebih panjang dari huruf input, maka sistem akan mengulang berdasarkan panjang huruf basis data yang selanjutnya akan membandingkan huruf yang jika salah akan menambah error dan sisa huruf yang berlebihan akan dihitung error.

2.4 Visual Studio Code

Visual Studio Code merupakan perangkat lunak untuk membantu pemrogram dalam mempermudah melakukan coding dalam berbagai bahasa. Fitur yang ada dalam Visual Studio Code antara lain (*Visual Studio Code - Code Editing. Redefined, 2022*):

1. Menyediakan penyelesaian otomatis berdasarkan tipe variabel, fungsi, dan modul yang diimpor.
2. Dapat melakukan debug secara langsung di editor.
3. Bekerja sama dengan git, sehingga mempermudah pengistalan melalui git.
4. Dapat menginstal ekstensi sendiri untuk layanan tambahan.

2.5 Website

Sebelum munculnya website pada tahun awal tahun 1990, para ilmuwan mengalami kesulitan dalam melakukan distribusi data dalam jumlah yang luar biasa banyak ke berbagai ratus ribu komputer. Karena kesulitan tersebut, tim Berners-Lee merancang metode navigasi dengan menggunakan kerangka hyperlink yang dikenal sebagai HTTP atau *Hypertext Transfer Protocol* dan menciptakan bahasa markup yang disebut HTML

atau *Hypertext Markup Language*. Untuk menyatukan hal tersebut, Tim Berners-Lee membuat browser web dan server web pertama di dunia (Nixon, 2021).

Untuk fungsi sendiri, HTTP memiliki fungsi sebagai standar komunikasi yang mengatur permintaan dan tanggapan yang dikirim antara klien dengan server web. Dan tugas server adalah sebagai penerima layanan yang dikirim klien dan mengirim timbal balik yang sesuai dengan kebutuhan klien. Untuk menggunakan HTTP, maka diperlukanlah hosting sebagai perantara agar dapat digunakan oleh pengguna yang disebut domain. Yang jika di analogikan, domain seperti alamat rumah dan hosting sebagai tanah bangunan. Untuk membangun rumah diperlukan berbagai material, untuk rumah bisa dikatakan sebagai website dan material bisa disebut file yang diperlukan untuk membuat website. File yang diperlukan dapat berupa gambar, script, video, email, aplikasi dan basis data yang mendukung untuk pembuatan website.

2.6 Bahasa Pemrograman

Bahasa pemrograman atau yang diistilahkan sebagai bahasa komputer merupakan bahasa berupa instruksi-instruksi dasar yang digunakan untuk memberi perintah kepada komputer. Biasanya bahasa pemrograman digunakan untuk menyelesaikan tugas-tugas tertentu ataupun mengelola data. Dibawah merupakan bahasa pemrograman yang digunakan dalam membuat program ini.

2.6.1 Javascript

Javascript pertama kali muncul pada browser Netscape Navigator pada tahun 1995 (Nixon, 2021). Javascript sendiri memiliki fungsi untuk membentuk interaktif antara pengguna dengan website yang dikunjungi. Hal yang bisa dilakukan seperti, mengubah warna ketika mouse mendekati kata, mengubah isi layar ketika menekan tombol, mengirim data di halaman dan mengirimnya ke halaman lain dan masih banyak lagi.

2.6.2 CSS

CSS atau *Cascading Style Sheets* dapat melakukan perubahan *style* pada sebuah halaman web (Nixon, 2021). Misalnya ingin mengubah tampilan default pada elemen html seperti `<h1>` menjadi memiliki font yang berbeda, ukuran font, warna background, warna font, huruf tebal, huruf miring, margin, dan sebagainya. Dengan CSS, tampilan website dapat menjadi lebih sedap dipandang.

2.6.3 JQuery

Jquery memiliki tingkat kompatibilitas lintas-browser yang sangat tinggi dan juga memiliki akses cepat dan mudah untuk memanipulasi HTML, memiliki fungsi khusus untuk berinteraksi langsung dengan CSS, kemampuan untuk mengontrol *event*, membuat efek dan animasi, dan memiliki fungsi asinkron dengan server web (Nixon, 2021). Secara teknikal, JQuery adalah method panggilan yang browser gunakan dan mengecilkan jumlah baris kode yang diperlukan untuk menyelesaikan sesuatu (Bibeault, De Rosa and Katz, 2015). Jadi JQuery adalah method panggilan yang browser gunakan, yang memiliki tingkat kompatibilitas lintas-browser yang sangat tinggi dan juga memiliki akses cepat dan mudah untuk memanipulasi HTML, bisa berinteraksi dengan CSS, kemampuan untuk mengontrol event, membuat efek dan animasi, fungsi asinkron dengan server web dan dapat mengecilkan jumlah baris kode yang diperlukan untuk menyelesaikan sesuatu.

2.6.4 PHP

PHP merupakan singkatan recursive dari *PHP : Hypertext Preprocessor*. Bahasa PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1994 (Achmad Solichin, 2018). Untuk penulisan dalam program ini akan menggunakan sintaks yang diawali dengan sintaks `<?php` dan diakhir dengan sintaks `?>` untuk mewakili penggunaan bahasa PHP. Setiap satu perintah biasanya ditulis dengan diakhir tanda `(;)`. Case sensitive sangat berpengaruh dalam penggunaan variabel, konstanta, fungsi, dan sebagainya. Akan tetapi dalam penggunaan bahasa pemrograman php, case sensitive tidak berpengaruh.

2.7 XAMPP

XAMPP merupakan platform untuk melakukan pengembangan dengan menggunakan PHP paling populer. XAMPP sendiri adalah perangkat lunak web server gratis dan open source yang memungkinkan pengguna mengupload website di internet dan support dengan MariaDB, PHP, dan Perl. XAMPP adalah distribusi Apache yang sederhana dan ringan yang membuatnya sangat mudah bagi developer untuk menciptakan web server lokal dengan tujuan untuk pengujian (Walia and Gill, 2014). Jadi XAMPP adalah perangkat lunak dengan web server lokal untuk melakukan pengujian bagi para pengguna yang ingin mengupload website di internet dengan support MariaDB, PHP, dan Perl.

2.8 MYSQL

MySQL merupakan sebuah perangkat lunak yang berbasis manajemen basis data SQL atau dalam bahasa Inggris disebut *basis data management system* atau disingkat DBMS yang memiliki keunggulan untuk melakukan lebih dari satu pekerjaan dalam waktu yang bersamaan serta dapat diakses oleh banyak pengguna (Achmad Solichin, 2018). Penggunaan MySQL memiliki kelebihan sendiri seperti gratis, stabil dan tangguh, dapat dipakai oleh berbagai program, keamanan yang baik, memiliki banyak dukungan dari komunitas, kemudahan dalam melakukan manajemen basis data, dan perkembangan perangkat lunak yang cukup cepat (*XAMPP Installers and Downloads for Apache Friends*, 2022).

2.9 JSON

JSON merupakan format data paling populer dalam mengirim permintaan dan tanggapan API, akan tetapi memiliki kekurangan dalam menentukan struktur data yang jelas (Pezoa *et al.*, 2016). Contohnya, hasil output untuk mengeluarkan data yang ada memiliki struktur "id" dan "data". Akan tetapi, untuk output yang tidak memiliki data memiliki struktur "hasil". Keduanya dapat dijalankan dengan normal, tetapi ketika masuk dalam program akan terjadi kebingungan karena key tiap struktur bisa berbeda. Solusinya adalah

harus ada kesepakatan struktur tiap percabangan, sehingga hasil key yang keluar akan sama.

2.10 UML

UML merupakan sebuah pemodelan atau perencanaan dalam membuat rekayasa perangkat lunak melalui diagram. Penggunaan UML memiliki berbagai cara dalam pembuatan rekayasa perangkat lunak, antara lain : (Unhelkar, 2017)

1. Visualisasi

Visualisasi adalah tujuan utama UML karena notasi dan diagramnya menyediakan mekanisme berstandar industri untuk menggambarkan persyaratan, proses, desain solusi, dan arsitektur. Visual ini dibuat menggunakan CASE sebagai alat pemodelan, yang juga memungkinkan berbagi pekerjaan pemodelan dengan tim lain.

2. Spesifikasi

UML memfasilitasi alat untuk melakukan pemodelan. Misalnya, alat untuk pembuatan aktor, use case, kelas, atribut, dan operasi pemberian detail tambahan untuk catatan secara visual. Bantuan ini sangat membantu meningkatkan kualitas dalam pencarian solusi dan membantu menyelesaikan banyak kesalahpahaman antara pengguna dan pengembang.

3. Konstruksi

UML digunakan untuk mengkonstruksi perangkat lunak karena pembuatan kode bergantung pada CASE yang digunakan. Namun, fitur konstruksi UML memiliki aplikasi yang terbatas, terutama ketika kode dibuat, sebagian besar proyek bekerja secara langsung dalam memodifikasi kode daripada desainnya.

4. Dokumentasi

Melakukan dokumentasi dengan UML merupakan penambahan dan rincian dokumentasi untuk persyaratan, arsitektur, desain, rencana proyek, pengujian, dan prototipe dapat meningkatkan spesifikasi dan gambaran visual.

5. Pemeliharaan

membuat model UML yang baik dapat membantu secara signifikan dalam pemeliharaan sistem perangkat lunak secara berkelanjutan. Pembuatan model memungkinkan membuat tampilan jadi lebih mudah untuk sistem, arsitektur, dan desain TI.

Bagian – bagian diagram yang terdapat dalam diagram UML, antara lain :

1. *Use Case Diagrams*

Use case diagrams adalah model untuk persyaratan sistem pada tingkat tinggi. Diagram use case biasanya digunakan untuk memvisualisasikan kasus penggunaan, sektor yang sesuai, dan interaksi. Tampilan model dari use case dapat membantu memahami proses bisnis dan komunikasi dengan orang yang berkepentingan. Sementara untuk spesifikasi dan dokumentasi dari diagram use case dapat memperlihatkan inti dari syarat pemodelan.

Use case diagram dapat menggambarkan perilaku apa saja yang dapat dilakukan oleh aktor melalui tiap case yang ada. Tiap case juga dapat melakukan kolaborasi dengan aktor lain (Nathan & Scobell, 2017). Dengan penggambaran seperti ini, keterlibatan aktor terhadap tiap case ataupun keterlibatan aktor dengan aktor lain melalui case yang sama dapat terlihat lebih jelas.

2. *Activity Diagrams*

Activity diagrams memodelkan alur proses dalam suatu sistem. Pemodelan alur proses ini dapat dilakukan ke tingkat proses bisnis dan bersama *use case*. Kegunaan diagram *activity* dapat menjelaskan tentang tahapan alur proses yang terjadi pada *use case* atau bisnis. Kegunaan penting lainnya dari diagram *activity* adalah kemampuan untuk menunjukkan keterlibatan antara aktivitas. *Activity diagrams* juga membantu dalam memperlihatkan hubungan aktivitas dengan aktor yang terkait dalam sistem. Selain itu, karena kemampuannya untuk menampilkan banyak alur proses, dapat digunakan untuk melihat apa yang terjadi pada sistem secara bersamaan.

3. *Class Diagrams*

Class diagrams mewakili entitas kunci dalam bisnis atau domain teknis yang ditunjukkan dalam bentuk kelas dan juga menunjukkan relasi antar kelas. Dengan class diagram dapat melihat isi tiap kelas dan melihat setiap hubungan antara tiap kelas yang mempermudah bisnis dan teknis.

4. *Sequence Diagrams*

Sequence diagrams dapat menggambarkan sebuah gambaran dan skenario dalam use case. Setiap tahap dalam use case dapat memunculkan diagram sequence sebagai catatan atau narasi. Diagram sequence menggambarkan detail interaksi antara aktor dan sistem atau antara objek dengan objek lain dalam waktu tertentu. Namun, informasi informasi tentang apa yang terjadi sebelum interaksi dimulai dan apa yang terjadi setelah blok waktu terhenti tidak ditampilkan dalam diagram sequence.

5. *Interaction Overview Diagrams*

Interaction overview diagrams memberikan gambaran umum tentang interaksi yang terjadi dalam suatu sistem. Karena interaksi lebih baik digambarkan menggunakan diagram sequence (atau alternatifnya communication), interaction overview diagrams berisi referensi dari diagram sequence. Interaction overview diagrams juga dapat menggambarkan situasi "if-then-else", sehingga diagram ini lebih dekat dengan diagram *activity*.

6. *Communication Diagrams*

Communication diagram menunjukkan kumpulan objek yang saling terhubung, bagaimana mereka terhubung melalui pesan mereka dan urutan pesan tersebut. Diagram communication menunjukkan informasi yang mirip dengan diagram sequence, tetapi cara menampilkannya berbeda. Secara tampilan, diagram communication dapat dianggap sebagai alat untuk menampilkan semua pesan yang dikirim dan diterima dari suatu objek. Informasi ini dapat digunakan untuk memastikan muatan pada suatu objek pada saat runtime.

7. *Object Diagrams*

Diagram *object* menunjukkan pada titik waktu tertentu, struktur berbagai objek, dan hubungan satu sama lain. Jadi ini hanya membentuk “snapshot” dalam memori komputer saat sistem dioperasikan atau rangkaian hubungan antara beberapa objek. Diagram ini tidak dapat menunjukkan perubahan dari waktu ke waktu. Dengan kata lain hanya menunjukkan apa yang terjadi yang dikaitkan dengan hubungan antar objek baik dalam memori utama atau sebagai mekanisme untuk menggambarkan dan mendiskusikan berbagai macam di papan tulis.

8. *State Machine Diagram*

State machine diagram (dapat disebut *statechart*) menunjukkan berbagai kondisi pada suatu objek, use case, atau keseluruhan sistem dapat berada. Dibandingkan *object diagram*, *state machine diagram* menunjukkan semua kemungkinan kondisi untuk suatu objek. Diagram ini juga memiliki kemampuan untuk menggambarkan waktu yang tepat dengan waktu real-time.

9. *Composite Structure Diagrams*

Composite structure diagrams menguraikan objek atau komponen pada waktu proses dan menunjukkan berbagai interface dan realisasi yang terkait dengan objek tersebut.

10. *Component Diagrams*

Component diagrams menunjukkan struktur sistem seperti yang diterapkan, tetapi tidak menunjukkan perilaku sistem dalam operasi dan juga tidak terdapat konsep waktu.

11. *Deployment Diagrams*

Deployment diagrams membuat penorganisasian pemrosesan *node* untuk memungkinkan untuk membagi sistem perangkat lunak dan perangkat keras. Diagram deployment juga berguna untuk membuat keputusan untuk menyambungkan perangkat keras yang terkait. Selain itu, diagram ini dapat memungkinkan digunakan untuk operasional sistem termasuk kemampuan sistem

dalam menangani kecepatan dan volume, lokasi dan keamanan sistem, dan mencari tempat untuk executable untuk diterapkan di seluruh jaringan.

12. *Package Diagrams*

Package diagrams mewakili bagian besar dari hubungan suatu sistem (subsistem). Diagram package adalah kumpulan properti yang saling berhubungan secara logis dengan diagram UML. Diagram ini juga dapat menunjukkan ketergantungan antar paket, namun ketergantungan dalam diagram package tidak wajib.

13. *Timing Diagrams*

Timing diagrams dapat membandingkan keadaan objek baik di dalam ruang masalah atau solusi. Diagram ini digunakan untuk menunjukkan perubahan status pada suatu objek. Sementara tujuan yang sama dicapai melalui state machine diagrams, diagram timing dapat menunjukkan banyak objek dan status yang sesuai pada saat yang sama.

14. *Profile Diagrams*

Metamodel yang mendasari UML memungkinkannya diterapkan ke berbagai situasi. Salah satu cara yang mungkin untuk memperluas UML ke aplikasi proyek adalah *profile diagrams*. *Profile* juga dapat dibuat untuk proyek tertentu seperti pemodelan proses bisnis.

BAB III

ANALISA DAN PERANCANGAN PERANGKAT LUNAK

3.1 Gambaran Umum Penggunaan *Auto correct*

Aplikasi yang dibuat seperti sebuah box yang dimana pengguna akan mengetikan sebuah kata. Ketika pengguna mengetikkan sebuah kata, maka program akan mulai mencocokkan kata yang ditulis oleh pengguna dengan yang ada di basis data. Jika yang diketikkan oleh pengguna adalah spasi dan fitur *auto correct* dinyalakan maka sistem akan merubah otomatis kata yang telah diketikkan oleh pengguna menjadi kata yang direkomendasikan paling atas. Sementara jika pengguna mengetikkan spasi dan fitur *auto correct* dimatikan, maka sistem akan mengabaikan kesalahan apapun yang dituliskan oleh pengguna dan hanya menampilkan kata yang paling mirip dengan kata yang diketikkan pengguna.

Untuk penerapan algoritma Levenshtein, digunakan ketika pengguna sedang mengetikkan sebuah kata, setiap kata yang diketik akan dibandingkan dengan data yang ada pada basis data, selanjutnya kata dari basis data akan dihitung jumlah errornya berdasarkan perbandingan kata yang diketik oleh pengguna dengan kata yang ada pada basis data. Kata yang errornya paling sedikit akan direkomendasikan menjadi yang terutama. Jika terjadi kasus jumlah error yang dibandingkan sama, maka sistem akan mengambil kata yang paling sering diinput atau disimpan dalam basis data.

Aplikasi ini juga menyediakan fitur untuk memprediksi kata selanjutnya setelah pengguna mengetikkan sebuah kata. Untuk memprediksi kata selanjutnya, sistem akan mengambil kata yang direkomendasikan yang terutama. Selanjutnya sistem akan mencari pasangan kata yang sesuai dengan kata yang direkomendasikan tadi dan menampilkannya kedalam list kata yang akan diprediksi selanjutnya. Untuk urutan list yang ditampilkan akan diurutkan berdasarkan seringnya pasangan kata yang diinput atau disimpan dalam basis data.

Setelah pengguna selesai mengetikkan kata, pengguna dapat menyimpannya jika kata yang telah diketik ingin disimpan dalam basis data. Ketika pengguna menekan tombol save, maka sistem akan membagi kalimat yang telah pengguna buat menjadi beberapa kata dan kata yang telah dibagi akan disimpan dalam basis data.

3.2 Spesifikasi Kebutuhan Perangkat Lunak dan Perangkat Keras untuk Aplikasi

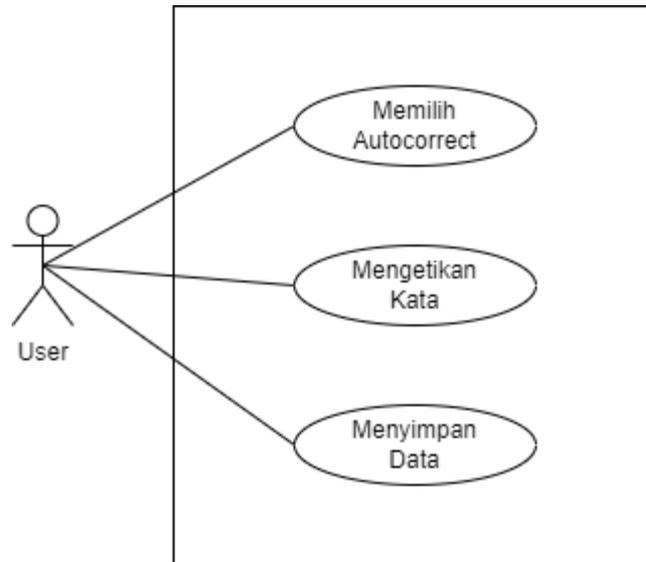
Spesifikasi komputer yang diperlukan agar aplikasi dapat digunakan oleh pengguna, sebagai berikut:

1. RAM : 1 GB
2. Storage : 512 MB
3. OS : Windows 10
4. Processor : Intel I3
5. Browser Chrome versi 101.0.4951.64 (yang direkomendasikan)

Spesifikasi komputer yang diperlukan agar aplikasi dapat digunakan oleh penulis, sebagai berikut:

1. RAM : 8 GB
2. Storage : 1 GB
3. OS : Windows 10
4. XAMPP versi 3.2.4
5. phpMyAdmin versi 10.4.17
6. Visual Studio Code 1.66.2

3.3 Use Case Diagram



Gambar 3. 1 Use case diagram *auto correct*

3.4 Skenario Use Case

3.4.1 Skenario *Auto correct*

Use Case : *Auto correct*

Aktor : Pengguna

Pre-kondisi : Browser telah dibuka

Deskripsi : Setiap kata yang diketik oleh pengguna dapat dikoreksi secara langsung

Tabel 3. 1 Skenario normal *auto correct*

Pengguna	System
1. Memilih <i>auto correct</i>	
	2. Mengeksekusi pilihan pengguna
3. Menggetikkan kata	
	4. Menampilkan rekomendasi kata yang diketik
5. Melihat kata yang diperbaiki dan di rekomendasi	

Post kondisi = Pengguna melihat kata

Skenario Alternatif : *Auto correct* di nyalakan dan pengguna menekan spasi

Use Case : *Auto Correct*

Aktor : Pengguna

Pre-kondisi : Browser telah dibuka

Deskripsi : Pengguna memilih basis data yang ingin digunakan

Tabel 3. 2 Skenario alternatif *auto correct*

Pengguna	System
1. Memilih <i>auto correct</i>	
	2. Mengeksekusi pilihan pengguna
3. Menekan spasi	
	4. Merubah kata sesuai yang direkomendasikan sistem
5. Pengguna melihat kata yang sudah diperbaiki	

Post kondisi = Pengguna melihat kata

Skenario Alternatif : *Auto correct* di matikan

Use Case : *Auto Correct*

Aktor : Pengguna

Pre-kondisi : Browser telah dibuka

Deskripsi : Pengguna memilih basis data yang ingin digunakan

Tabel 3. 3 Skenario alternatif *auto correct*

Pengguna	System
1. Memilih <i>auto correct</i> dimatikan	
	2. Mengeksekusi pilihan pengguna
3. Mengetikkan kata yang salah	
	4. Menampilkan rekomendasi kata yang diketik
5. Pengguna melihat kata yang salah dan kata yang direkomendasikan	

Post kondisi = Pengguna melihat kata

3.4.2 Skenario Mengetikan Kata

Use Case : Mengetikan Kata

Aktor : Pengguna

Pre-kondisi : Browser telah dibuka

Deskripsi : Pengguna mengetikkan sebuah kata

Tabel 3. 4 Skenario normal mengetikan kata

Pengguna	System
1. Mengetik sebuah kata	
	2. Membandingkan setiap huruf dengan dengan huruf di basis data
	3. Mecocokan kata yang paling mirip dari basis data
	4. Menampilkan kata yang paling mirip untuk direkomendasikan menggunakan algoritma Levenshtein
5. Melihat kata yang direkomendasikan	
6. Memilih kata yang diinginkan	
	7. Menampilkan kata yang mirip dengan yang dipilih
8. Melihat kata yang mirip dengan pilihan	

Post-kondisi : Pengguna melihat kata

Skenario Alternatif : Kata sudah benar

Use Case : Mengetikan Kata

Aktor : Pengguna

Pre-kondisi : Browser telah dibuka

Deskripsi : Pengguna mengetikkan sebuah kata

Tabel 3. 5 Skenario alternatif mengetikan kata

Pengguna	System
1. Mengetik sebuah kata	
	2. Membandingkan setiap huruf dengan dengan huruf di basis data
	3. Mecocokan kata yang paling mirip dari basis data
	4. Menampilkan kata yang diketik
5. Melihat kata yang sudah diketik sebelumnya	

Post-kondisi : Pengguna melihat kata

3.4.3 Skenario Menyimpan Data

Use Case : Menyimpan Data

Aktor : Pengguna

Pre-kondisi : Browser telah dibuka

Deskripsi : Pengguna menekan tombol simpan

Tabel 3. 6 Skenario normal menyimpan data

Pengguna	System
1. Menekan tombol simpan	
	2. Memeriksa text
	3. Menyimpan data dari text yang sudah diketik
	4. Memunculan notifikasi "Tersimpan"
5. Melihat notifikasi	

Post kondisi = Pengguna melihat notif

Skenario Alternatif : Text kosong

Use Case : Menyimpan data

Aktor : Pengguna

Pre-kondisi : Browser telah dibuka

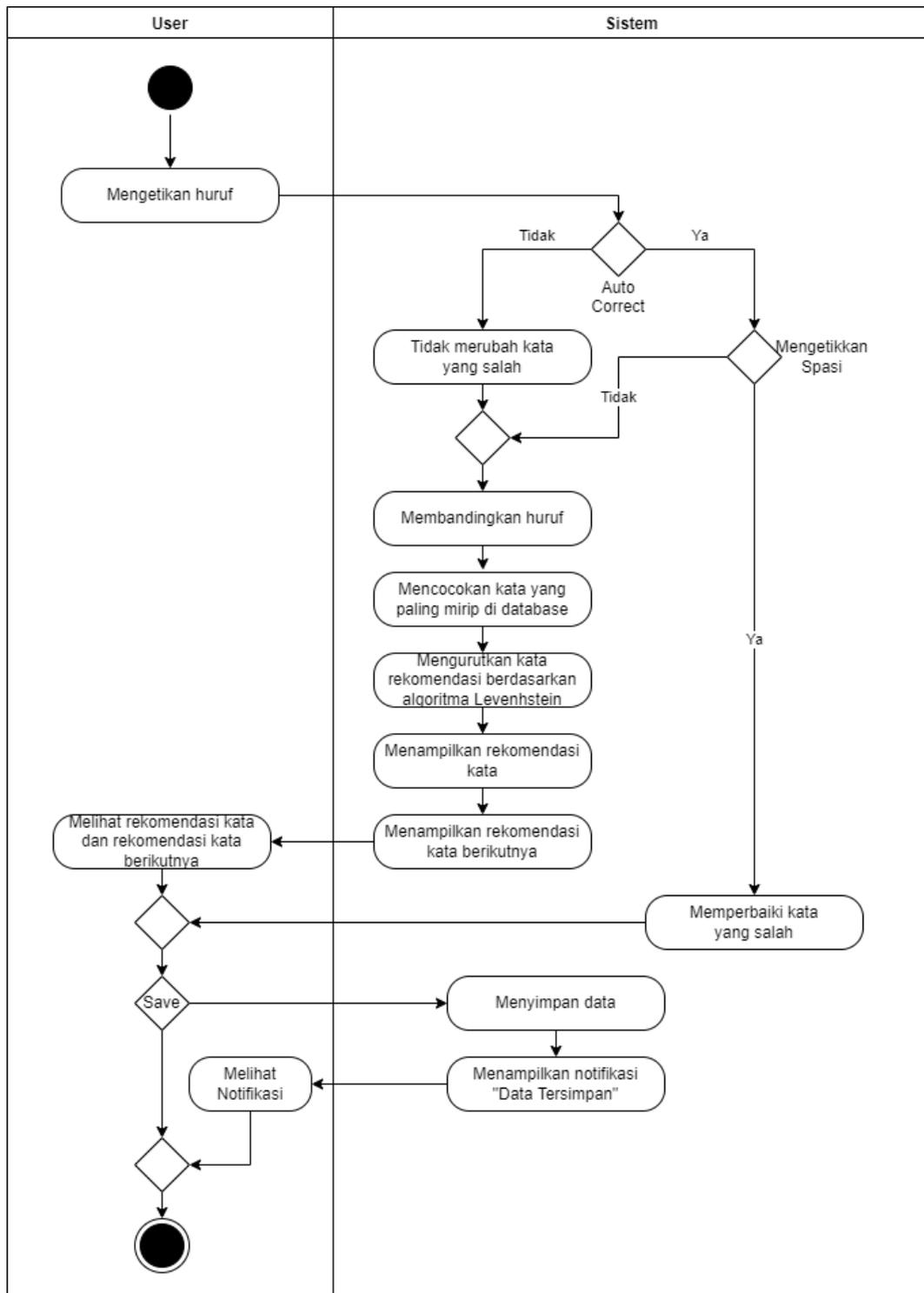
Deskripsi : Pengguna menekan tombol simpan

Tabel 3. 7 Skenario alternatif menyimpan data

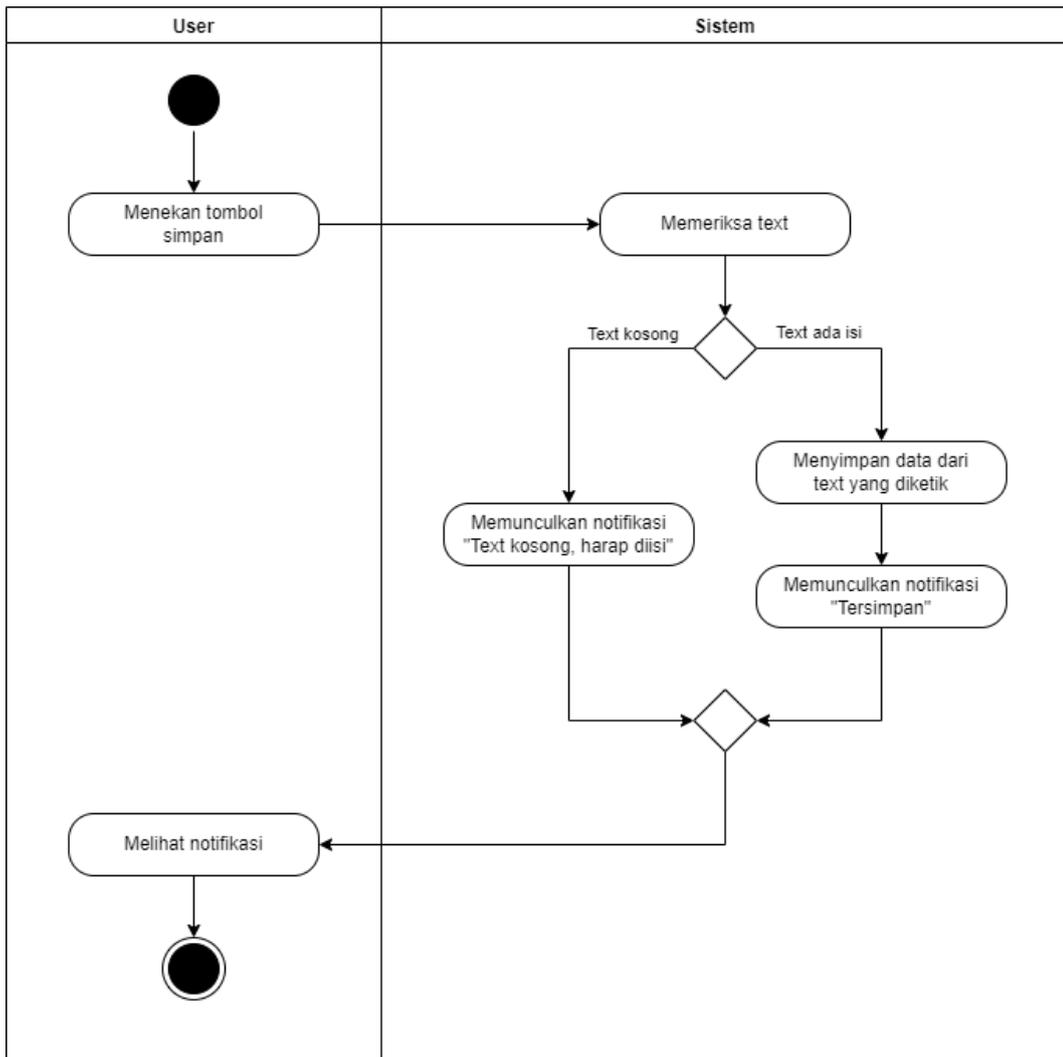
Pengguna	System
1. Menekan tombol simpan	
	2. Memeriksa text
	3. Memunculan notifikasi "Text kosong, harap diisi"
4. Melihat notifikasi	

Post kondisi = Pengguna melihat notif

3.5 Activity Diagram

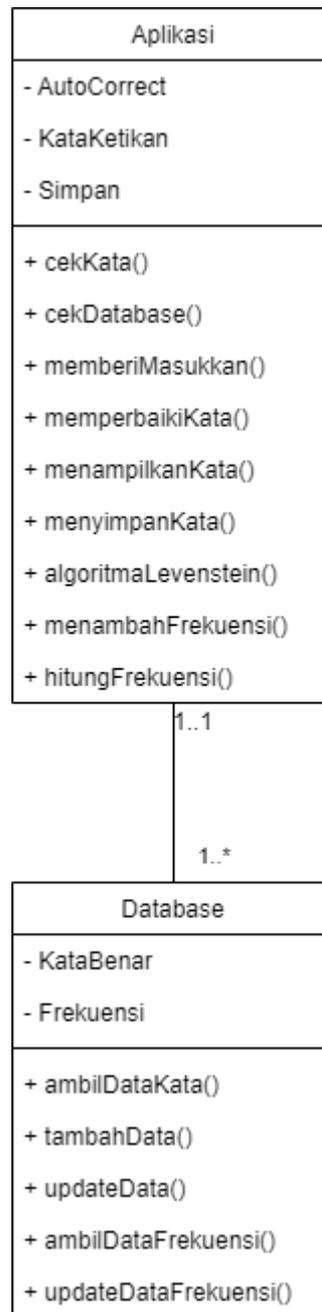


Gambar 3. 2 Activity diagram mengetikan kata



Gambar 3. 3 Activity diagram menyimpan data

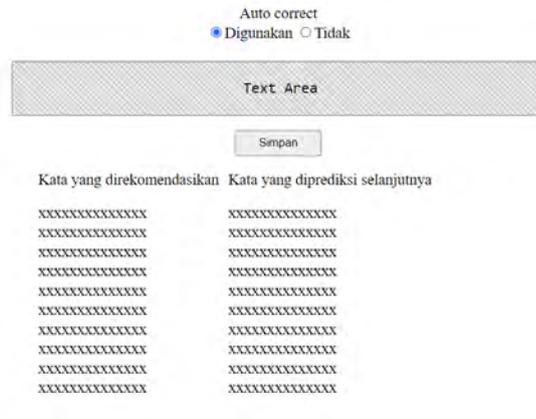
3.6 Class Diagram



Gambar 3. 4 Class diagram mengetikan kata

3.7 Perancangan Antar Muka

Rancangan antarmuka untuk program *auto correct*.



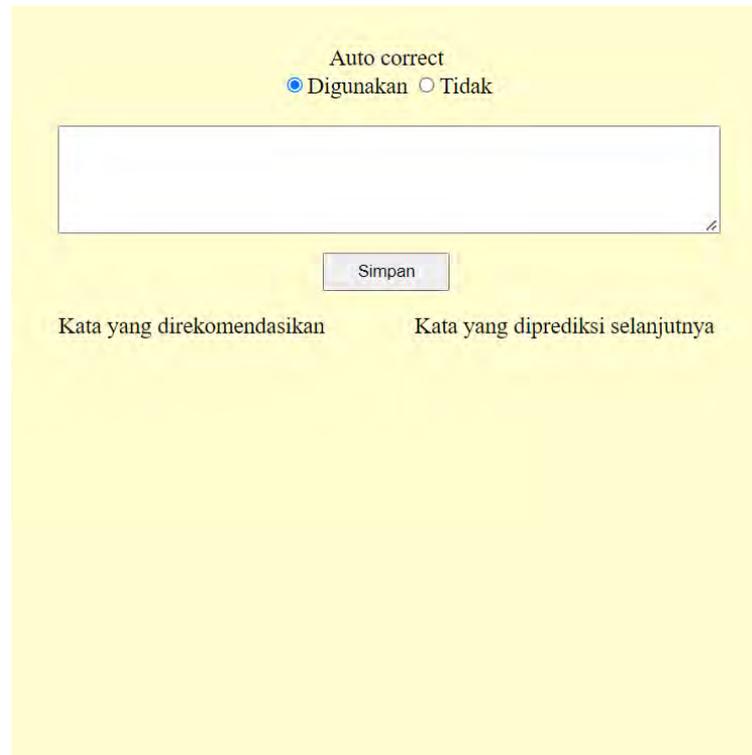
Gambar 3. 5 Rancangan antarmuka tampilan awal atau default

Gambar 3.5 merupakan gambaran dari rancangan antarmuka ketika program pertama kali dijalankan. Fitur auto correct berfungsi sebagai pengoreksi otomatis, jika radio button diposisi digunakan, maka setiap user menekan spasi tulisan akan berubah otomatis. Jika auto correct diposisi tidak, maka tulisan tidak akan berubah. Untuk tombol simpan berfungsi sebagai menyimpan tulisan yang diketik oleh user. Sementara kata yang direkomendasikan akan menampilkan kata yang mirip dengan kata yang diketik oleh user dan kata yang diprediksi selanjutnya akan menampilkan kata yang terhubung dengan kata yang diketik oleh user.

BAB IV

Perancangan Antar Muka

4.1 Tampilan Antar Muka



Gambar 4. 1 Tampilan Awal

Tampilan Awal merupakan tampilan ketika browser pertama kali dibuka. Tampilan awal memiliki beberapa fitur, berikut penjelasan tiap fitur pada tampilan awal:

1. *Auto correct* memiliki fungsi sebagai mengubah kata yang diketik oleh pengguna berubah otomatis sesuai dengan kata yang direkomendasikan paling atas.
2. Text area merupakan tempat untuk pengguna mengetikkan berbagai kata.
3. Tombol simpan memiliki fungsi untuk menyimpan semua kata yang diketik oleh pengguna ke basis data. Semakin banyak data yang tersimpan, maka data yang keluar akan semakin akurat.
4. Kata yang direkomendasikan merupakan list yang menampilkan kata-kata apa saja yang paling mirip dengan kata yang diketik oleh pengguna.
5. Kata yang diprediksi selanjutnya merupakan list yang menampilkan kata-kata apa saja yang cocok dengan kata yang sedang diketik oleh pengguna.

Tabel 4. 1 Tabel Tampilan Antar Muka

No	Gambar	Judul Gambar	Keterangan
1		Tampilan ketika di ketik	Merupakan tampilan ketika pengguna sedang mengetikkan kata. Sistem akan mulai menampilkan kata rekomendasi yang paling mirip sesuai dengan kata yang sedang diketik. Sementara kata yang diprediksi akan mengikuti kata yang direkomendasikan paling atas.
2		Tampilan ketika kata di spasi dan <i>auto correct</i> dinyalakan	Merupakan tampilan ketika pengguna menekan tombol spasi dan <i>auto correct</i> dinyalakan. Sistem akan merubah otomatis sesuai dengan kata yang direkomendasikan paling atas.
3		Tampilan kata di spasi dan <i>auto correct</i> dimatikan	Merupakan tampilan ketika pengguna menekan tombol spasi dan <i>auto correct</i> dimatikan. Sistem tidak melakukan perubahan apapun.
4		Tampilan ketika <i>cursor</i> diarahkan ke kata yang direkomendasikan	Merupakan tampilan ketika pengguna mengarahkan <i>cursor</i> ke kata yang direkomendasikan. Kata yang terkena <i>cursor</i> akan berubah warna.

No	Gambar	Judul Gambar	Keterangan
5		Tampilan ketika kata yang direkomendasikan di klik	Merupakan tampilan ketika pengguna mengklik kata yang di <i>cursor</i> , lalu sistem akan merubah text yang ditulis pengguna menjadi kata yang sesuai dengan yang diklik.
6		Tampilan ketika <i>cursor</i> diarahkan ke kata yang direkomendasikan	Merupakan tampilan ketika pengguna mengarahkan <i>cursor</i> ke kata yang diprediksi. Kata yang terkena <i>cursor</i> akan berubah warna.
7		Tampilan ketika kata yang diprediksi di klik	Merupakan tampilan ketika pengguna mengklik kata yang di <i>cursor</i> , lalu sistem akan menambah text sesuai dengan kata yang diklik.
8		Tombol simpan diklik ketika text masih kosong	Merupakan tampilan ketika pengguna langsung mengklik simpan dengan kondisi text kosong. Maka pengguna akan memberikan notifikasi "Text kosong, harap diisi" agar sistem dapat bekerja dengan benar.

No	Gambar	Judul Gambar	Keterangan
9		Tombol simpan diklik ketika ada text	Merupakan tampilan ketika pengguna sudah mengetikkan kata dan menekan tombol simpan. Sistem akan menampilkan notifikasi "Tersimpan" dan data yang diketik akan tersimpan di basis data.

4.2 Pengujian Fungsional

Tabel 4. 2 Tabel Pengujian Fungsional

Test Case ID	Aktivitas Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Mengetikkan kata benar dan <i>auto correct</i> dinyalakan	Kata tidak berubah ketika pengguna menekan spasi dan sistem menampilkan rekomendasi dan kata yang di prediksi selanjutnya	Kata tidak berubah dan sistem menampilkan rekomendasi dan kata yang di prediksi selanjutnya	Valid
2	Mengetikkan kata salah dan <i>auto correct</i> dinyalakan	Setiap pengguna menekan spasi, kata yang salah akan berubah menjadi kata yang direkomendasikan paling awal	Setiap pengguna menekan spasi, kata yang salah akan berubah menjadi kata yang direkomendasikan paling awal	Valid
3	Mengetikkan kata benar dan <i>auto correct</i> dimatikan	Kata tidak berubah dan sistem menampilkan rekomendasi dan kata yang di prediksi selanjutnya	Kata tidak berubah dan sistem menampilkan rekomendasi dan kata yang di prediksi selanjutnya	Valid
4	Mengetikkan kata salah dan <i>auto correct</i> dimatikan	Kata tidak berubah dan sistem menampilkan rekomendasi dan kata yang di prediksi selanjutnya	Kata tidak berubah dan sistem menampilkan rekomendasi dan kata yang di prediksi selanjutnya	Valid
5	Mengklik kata yang direkomendasikan	Kata yang sedang diketik oleh pengguna berubah sesuai dengan kata yang di klik	Kata yang sedang diketik oleh pengguna berubah sesuai dengan kata yang di klik	Valid
6	Mengklik kata yang diprediksi	Kata yang sedang diketik oleh pengguna langsung ditambahkan sesuai	Kata yang sedang diketik oleh pengguna langsung ditambahkan sesuai	Valid

Test Case ID	Aktivitas Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
		dengan kata yang diklik	dengan kata yang diklik	
7	Menghapus semua text	Tampilan kata rekomendasi dan kata prediksi hilang	Tampilan kata rekomendasi dan kata prediksi hilang	Valid
8	Keluar dari textfield	Tampilan kata rekomendasi dan kata prediksi hilang	Tampilan kata rekomendasi dan kata prediksi hilang	Valid
9	Menghapus kata (maksudnya dari "saya mau makan" jadi "saya mau " =)	Tampilan kata rekomendasi dan kata prediksi hilang	Tampilan kata rekomendasi dan kata prediksi hilang	Valid
10	Menghapus beberapa huruf (maksudnya dari "latihan" jadi "lat")	Tampilan kata rekomendasi dan kata prediksi menyesuaikan sesuai kata yang ada	Tampilan kata rekomendasi dan kata prediksi menyesuaikan sesuai kata yang ada	Valid
11	Menekan tombol simpan ketika textfield kosong	Memunculkan notifikasi "Text kosong, harap diisi"	Memunculkan notifikasi "Text kosong, harap diisi"	Valid
12	Menekan tombol simpan ketika textfield ada isi	Memunculkan notifikasi "Tersimpan"	Memunculkan notifikasi "Tersimpan"	Valid

4.3 Pengujian Kompabilitas

Proses uji coba ini dilakukan untuk mengetahui tingkat kompabilitas program. Uji coba akan dilakukan di berbagai browser. OS yang digunakan yaitu, Windows dan Ubuntu. Untuk Windows 10 versi 19044 akan menggunakan browser Google Chrome, Mozilla Firefox dan Microsoft Edge. Sementara Ubuntu akan menggunakan browser Mozilla Firefox dan Google Chrome. Dan untuk android akan menggunakan merk oppo dengan OS android 12. Berikut tabel pengujian fungsional dari berbagai browser.

Tabel 4. 3 Tabel Pengujian Windows 10

Test Case ID	Aktivitas Pengujian	Google Chrome Versi 101.0.4951.64	Mozilla Firefox Versi 100.0	Microsoft Edge Versi 101.0.1210.39
1	Mengetikkan kata benar dan <i>auto correct</i> dinyalakan	Valid	Valid	Valid
2	Mengetikkan kata salah dan <i>auto correct</i> dinyalakan	Valid	Valid	Valid
3	Mengetikkan kata benar dan <i>auto correct</i> dimatikan	Valid	Valid	Valid

Test Case ID	Aktivitas Pengujian	Google Chrome Versi 101.0.4951.64	Mozilla Firefox Versi 100.0	Microsoft Edge Versi 101.0.1210.39
4	Mengetikkan kata salah dan <i>auto correct</i> dimatikan	Valid	Valid	Valid
5	Mengklik kata yang direkomendasikan	Valid	Valid	Valid
6	Mengklik kata yang diprediksi	Valid	Valid	Valid
7	Menghapus semua text	Valid	Valid	Valid
8	Keluar dari textfield	Valid	Valid	Valid
9	Menghapus kata (maksudnya dari "saya mau makan" jadi "saya mau " =)	Valid	Valid	Valid
10	Menghapus beberapa huruf (maksudnya dari "latihan" jadi "lat")	Valid	Valid	Valid
11	Menekan tombol simpan ketika textfield kosong	Valid	Valid	Valid
12	Menekan tombol simpan ketika textfield ada isi	Valid	Valid	Valid
Presentase				
Sukses		100%	100%	100%
Gagal		0%	0%	0%

Tabel 4. 4 Tabel Pengujian Ubuntu dan Android

Test Case ID	Aktivitas Pengujian	Mozilla Firefox Versi 100.0	Google Chrome Versi 101.0.4951.64	Android 12
1	Mengetikkan kata benar dan <i>auto correct</i> dinyalakan	Valid	Valid	Valid
2	Mengetikkan kata salah dan <i>auto correct</i> dinyalakan	Valid	Valid	Valid
3	Mengetikkan kata benar dan <i>auto correct</i> dimatikan	Valid	Valid	Valid
4	Mengetikkan kata salah dan <i>auto correct</i> dimatikan	Valid	Valid	Valid
5	Mengklik kata yang direkomendasikan	Valid	Valid	Valid
6	Mengklik kata yang diprediksi	Valid	Valid	Valid
7	Menghapus semua text	Valid	Valid	Valid
8	Keluar dari textfield	Valid	Valid	Valid
9	Menghapus kata (maksudnya dari "saya mau makan" jadi "saya mau " =)	Valid	Valid	Valid
10	Menghapus beberapa huruf (maksudnya dari "latihan" jadi "lat")	Valid	Valid	Valid

Test Case ID	Aktivitas Pengujian	Mozilla Firefox Versi 100.0	Google Chrome Versi 101.0.4951.64	Android 12
11	Menekan tombol simpan ketika textfield kosong	Valid	Valid	Valid
12	Menekan tombol simpan ketika textfield ada isi	Valid	Valid	Valid
Presentase				
Sukses		100%	100%	100%
Gagal		0%	0%	0%

Pada tabel pengujian 4.3 dapat disimpulkan bahwa browser Mozilla, Google Chrome dan Microsoft Edge tidak memiliki masalah ketika dijalankan pada OS Windows 10. Untuk tabel 4.4 dapat disimpulkan bahwa browser Mozilla dan Google Chrome tidak memiliki masalah ketika dijalankan pada OS Ubuntu, sementara pada browser Android 12 juga tidak memiliki masalah. Dari penjelasan tersebut dapat disimpulkan bahwa program tidak memiliki masalah di browser yang telah diuji coba.

BAB V

Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan hasil rancangan dan pengujian yang telah dilakukan, maka diperoleh kesimpulan sebagai berikut:

1. Algoritma Levenshtein dapat diimplementasikan sebagai pembandingan antara kata yang diketik oleh pengguna dan kata yang berada dalam basis data.
2. Aplikasi bisa diakses secara online, sehingga bisa berguna bagi banyak orang.
3. Dari hasil pengujian yang dilakukan penulis, aplikasi bisa berjalan dengan baik di OS windows dan Ubuntu dengan browser Microsoft Edge, Google Chrome, dan Mozilla Firefox. Namun di OS android masih belum berjalan sesuai yang diharapkan.

5.2 Saran

Berdasarkan implementasi dan uji coba yang dilakukan, masih terdapat beberapa kekurangan pada website yang dibuat dan berikut pengembangan yang dapat dilakukan:

1. Aplikasi dapat dikembangkan untuk menghitung plagiarisme.
2. Keamanan untuk basis data masih belum maksimal.

DAFTAR PUSTAKA

- Achmad Solichin (2018) 'Pemrograman Web dengan PHP dan MySQL - Google Books', *Penerbit SPARTA*, (January 2005), pp. 1–122.
- Bibeault, B., De Rosa, A. and Katz, Y. (2015) *jQuery in Action*. 3rd edn. Manning; 3rd edition. Available at:
https://books.google.co.id/books?hl=id&lr=&id=ZzkzEAAAQBAJ&oi=fnd&pg=PT19&dq=jQuery+in+Action&ots=fikl3-ceRA&sig=5U54j0rPdCYCAAXiivwky2ZL7Lw&redir_esc=y#v=onepage&q=jQuery in Action&f=false.
- Fairley, R. E. (2019) *Systems Engineering of Software-Enabled Systems, Systems Engineering of Software-Enabled Systems*. doi: 10.1002/9781119535041.
- Madison, J. (2012) *Damn You, Autocorrect!* Ebury Publishing. Available at:
https://books.google.co.id/books?hl=id&lr=&id=hYiD4I8S8SYC&oi=fnd&pg=PT9&dq=autocorrect&ots=EBkqut0n2N&sig=9h2rK3sELklr8F6c9dJSJMn0quE&redir_esc=y#v=onepage&q=autocorrect&f=false.
- Nixon, R. (2021) *Learning PHP, MySQL & JavaScript*. Sixth. United States of America: O'Reilly Media, Inc. Available at:
https://books.google.co.id/books?hl=en&lr=&id=KZw5EAAAQBAJ&oi=fnd&pg=PR2&q=javascript&ots=1MLLe_cP_w&sig=G-okkqm_0DOQxVPjHO_ZFB3AY2g&redir_esc=y#v=onepage&q=javascript&f=false.
- Pezoa, F. et al. (2016) 'Foundations of JSON schema', *25th International World Wide Web Conference, WWW 2016*, pp. 263–273. doi: 10.1145/2872427.2883029.
- Prasetyo, A., Baihaqi, W. M. and Had, I. S. (2018) 'Algoritma Jaro-Winkler Distance: Fitur Autocorrect dan Spelling Suggestion pada Penulisan Naskah Bahasa Indonesia di BMS TV', *Jurnal Teknologi Informasi dan Ilmu Komputer*, 5(4), p. 435. doi: 10.25126/jtiik.201854780.
- Retnoningsih, E., Shadiq, J. and Oscar, D. (2017) 'Pembelajaran Pemrograman Berorientasi Objek (Object Oriented Programming) Berbasis Project Based Learning', *Informatics for Educators and Professionals*, 2(1), pp. 95–104.
- Saidah, H. T., Ishlah, M. S. N. and Rokhmah, N. N. (2020) 'Autocorrect pada Modul Pencarian Drugs e-Dictionary Menggunakan Levenshtein Distance', *Rekayasa Sistem dan Teknologi Informasi*, 4(1), pp. 64–69. Available at:
<https://doi.org/10.29207/resti.v4i1.1401>.
- Sarcar, V. (2020) *Interactive Object-Oriented Programming in Java, Names*.
- Staron, M. (2020) *Action Research in Software Engineering, Action Research in Software Engineering*. doi: 10.1007/978-3-030-32610-4.
- Unhelkar, B. (2017) *Software engineering with UML, Software Engineering with UML*. doi: 10.1201/9781351235181.
- Visual Studio Code - Code Editing. Redefined* (2022). Available at:
<https://code.visualstudio.com/> (Accessed: 18 May 2022).
- Walia, S. and Gill, S. (2014) 'A Framework for Web Based Student Record Management System using PHP', *International Journal of Computer Science and Mobile Computing ISSN 2320-088X*, 3(8), pp. 24–33. Available at:
https://pdfs.semanticscholar.org/f5e1/b5d218431f89d5ecb05e7d2e23892cfaf042.pdf?_ga=2.4884785.364962908.1574432524-89530086.1569769324.
- Wangi, J. and Rosalin, K. (2019) 'Chinese Grammar Auto-Correct Website Chinesegrammarchecker.Com Development Model', *Business Economic*,

- Communication, and Social Sciences*), 1(1), pp. 137–146. Available at: <http://nlp.stanford.edu:8080/parser/index.jsp>.
- XAMPP Installers and Downloads for Apache Friends (2022). Available at: <https://www.apachefriends.org/index.html> (Accessed: 18 May 2022).
- Achmad Solichin (2018) 'Pemrograman Web dengan PHP dan MySQL - Google Books', *Penerbit SPARTA*, (January 2005), pp. 1–122.
- Bibeault, B., De Rosa, A. and Katz, Y. (2015) *jQuery in Action*. 3rd edn. Manning; 3rd edition. Available at: https://books.google.co.id/books?hl=id&lr=&id=ZzkzEAAAQBAJ&oi=fnd&pg=PT19&dq=jQuery+in+Action&ots=fikl3-ceRA&sig=5U54j0rPdCYCAAXiivwky2ZL7Lw&redir_esc=y#v=onepage&q=jQuery in Action&f=false.
- Fairley, R. E. (2019) *Systems Engineering of Software-Enabled Systems, Systems Engineering of Software-Enabled Systems*. doi: 10.1002/9781119535041.
- Madison, J. (2012) *Damn You, Autocorrect!* Ebury Publishing. Available at: https://books.google.co.id/books?hl=id&lr=&id=hYiD4I8S8SYC&oi=fnd&pg=PT9&dq=autocorrect&ots=EBkqut0n2N&sig=9h2rK3sElklr8F6c9dJSJMn0quE&redir_esc=y#v=onepage&q=autocorrect&f=false.
- Nixon, R. (2021) *Learning PHP, MySQL & JavaScript*. Sixth. United States of America: O'Reilly Media, Inc. Available at: https://books.google.co.id/books?hl=en&lr=&id=KZw5EAAAQBAJ&oi=fnd&pg=PR2&q=javascript&ots=1MLLe_cP_w&sig=G-okkqm_0DOQxVPjHO_ZFB3AY2g&redir_esc=y#v=onepage&q=javascript&f=false.
- Pezoa, F. *et al.* (2016) 'Foundations of JSON schema', *25th International World Wide Web Conference, WWW 2016*, pp. 263–273. doi: 10.1145/2872427.2883029.
- Prasetyo, A., Baihaqi, W. M. and Had, I. S. (2018) 'Algoritma Jaro-Winkler Distance: Fitur Autocorrect dan Spelling Suggestion pada Penulisan Naskah Bahasa Indonesia di BMS TV', *Jurnal Teknologi Informasi dan Ilmu Komputer*, 5(4), p. 435. doi: 10.25126/jtiik.201854780.
- Retnoningsih, E., Shadiq, J. and Oscar, D. (2017) 'Pembelajaran Pemrograman Berorientasi Objek (Object Oriented Programming) Berbasis Project Based Learning', *Informatics for Educators and Professionals*, 2(1), pp. 95–104.
- Saidah, H. T., Ishlah, M. S. N. and Rokhmah, N. N. (2020) 'Autocorrect pada Modul Pencarian Drugs e-Dictionary Menggunakan Levenshtein Distance', *Rekayasa Sistem dan Teknologi Informasi*, 4(1), pp. 64–69. Available at: <https://doi.org/10.29207/resti.v4i1.1401>.
- Sarcar, V. (2020) *Interactive Object-Oriented Programming in Java, Names*.
- Staron, M. (2020) *Action Research in Software Engineering, Action Research in Software Engineering*. doi: 10.1007/978-3-030-32610-4.
- Unhelkar, B. (2017) *Software engineering with UML, Software Engineering with UML*. doi: 10.1201/9781351235181.
- Visual Studio Code - Code Editing. Redefined* (2022). Available at: <https://code.visualstudio.com/> (Accessed: 18 May 2022).
- Walia, S. and Gill, S. (2014) 'A Framework for Web Based Student Record Management System using PHP', *International Journal of Computer Science and Mobile Computing ISSN 2320–088X*, 3(8), pp. 24–33. Available at: <https://pdfs.semanticscholar.org/f5e1/b5d218431f89d5ecb05e7d2e23892cfaf042.pdf?>

_ga=2.4884785.364962908.1574432524-89530086.1569769324.

Wangi, J. and Rosalin, K. (2019) 'Chinese Grammar Auto-Correct Website Chinesegrammarchecker.Com Development Model', *Business Economic, Communication, and Social Sciences*, 1(1), pp. 137–146. Available at: <http://nlp.stanford.edu:8080/parser/index.jsp>.

XAMPP Installers and Downloads for Apache Friends (2022). Available at: <https://www.apachefriends.org/index.html> (Accessed: 18 May 2022).

LAMPIRAN PENGUJIAN

Lampiran 1
Pengujian Fungsional

Test Case ID	Aktivitas Pengujian	Hasil yang diharapkan	Hasil Pengujian
1	Mengetikkan kata benar dan <i>auto correct</i> dinyalakan	Kata tidak berubah ketika pengguna menekan spasi dan sistem menampilkan rekomendasi dan kata yang di prediksi selanjutnya	<p>Sebelum</p>  <p>Sesudah</p> 
2	Mengetikkan kata salah dan <i>auto correct</i> dinyalakan	Setiap pengguna menekan spasi, kata yang salah akan berubah menjadi kata yang direkomendasikan paling awal	<p>Sebelum</p>  <p>Sesudah</p> 

Test Case ID	Aktivitas Pengujian	Hasil yang diharapkan	Hasil Pengujian
3	Mengetikkan kata benar dan <i>auto correct</i> dimatikan	Kata tidak berubah dan sistem menampilkan rekomendasi dan kata yang di prediksi selanjutnya	<p>Sebelum</p>  <p>Sesudah</p> 
4	Mengetikkan kata salah dan <i>auto correct</i> dimatikan	Kata tidak berubah dan sistem menampilkan rekomendasi dan kata yang di prediksi selanjutnya	<p>Sebelum</p>  <p>Sesudah</p> 

Test Case ID	Aktivitas Pengujian	Hasil yang diharapkan	Hasil Pengujian
5	Mengklik kata yang direkomendasikan	Kata yang sedang diketik oleh pengguna berubah sesuai dengan kata yang di klik	<p>Sebelum</p>  <p>Sesudah</p> 
6	Mengklik kata yang diprediksi	Kata yang sedang diketik oleh pengguna langsung ditambahkan sesuai dengan kata yang diklik	<p>Sebelum</p>  <p>Sesudah</p> 

Test Case ID	Aktivitas Pengujian	Hasil yang diharapkan	Hasil Pengujian
7	Menghapus semua text	Tampilan kata rekomendasi dan kata prediksi hilang	<p>Sebelum</p>  <p>Sesudah</p> 
8	Keluar dari textfield	Tampilan kata rekomendasi dan kata prediksi hilang	
9	Menghapus kata (maksudnya dari "saya mau makan" jadi "saya mau ")	Tampilan kata rekomendasi dan kata prediksi hilang	<p>Sebelum</p> 

Test Case ID	Aktivitas Pengujian	Hasil yang diharapkan	Hasil Pengujian
			<p>Sesudah</p> 
10	Menghapus beberapa huruf (maksudnya dari "latihan" jadi "lat")	Tampilan kata rekomendasi dan kata prediksi menyesuaikan sesuai kata yang ada	<p>Sebelum</p>  <p>Sesudah</p> 

Test Case ID	Aktivitas Pengujian	Hasil yang diharapkan	Hasil Pengujian
11	Menekan tombol simpan ketika textfield kosong	Memunculkan notifikasi "Text kosong, harap diisi"	
12	Menekan tombol simpan ketika textfield ada isi	Memunculkan notifikasi "Tersimpan"	

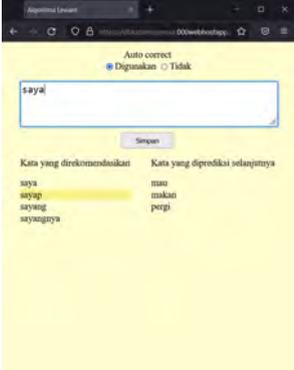
Lampiran 2
Pengujian Kompabilitas

Microsoft Windows 10

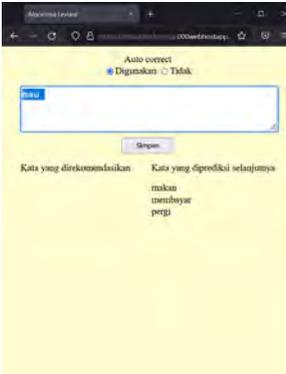
Test Case ID	Google Chrome Versi 101.0.4951.64	Mozilla Firefox Version 100.0	Microsoft Edge Version 101.0.1210.39
1	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 
2	<p style="text-align: center;">Sebelum</p> 	<p style="text-align: center;">Sebelum</p> 	<p style="text-align: center;">Sebelum</p> 

Test Case ID	Google Chrome Versi 101.0.4951.64	Mozilla Firefox Version 100.0	Microsoft Edge Version 101.0.1210.39
	<p style="text-align: center;">Sesudah</p> 	<p style="text-align: center;">Sesudah</p> 	<p style="text-align: center;">Sesudah</p> 
3	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 

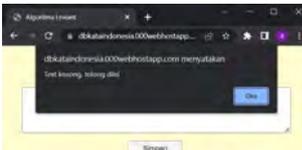
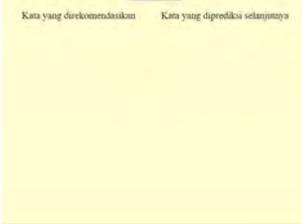
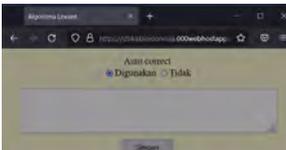
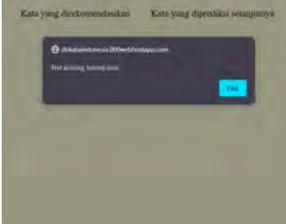
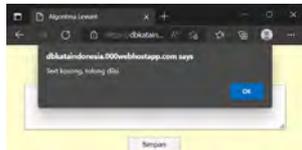
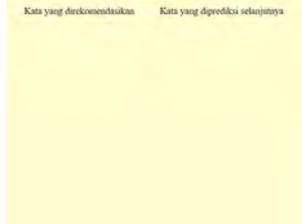
Test Case ID	Google Chrome Versi 101.0.4951.64	Mozilla Firefox Version 100.0	Microsoft Edge Version 101.0.1210.39
4	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 

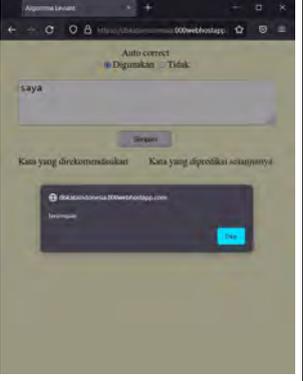
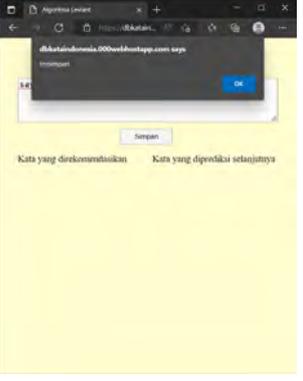
Test Case ID	Google Chrome Versi 101.0.4951.64	Mozilla Firefox Version 100.0	Microsoft Edge Version 101.0.1210.39
5	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 

Test Case ID	Google Chrome Versi 101.0.4951.64	Mozilla Firefox Version 100.0	Microsoft Edge Version 101.0.1210.39
6	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 

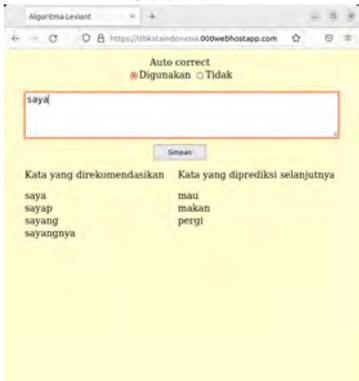
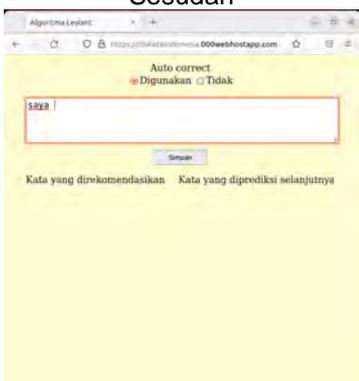
Test Case ID	Google Chrome Versi 101.0.4951.64	Mozilla Firefox Version 100.0	Microsoft Edge Version 101.0.1210.39
7	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 
8			
9	Sebelum	Sebelum	Sebelum

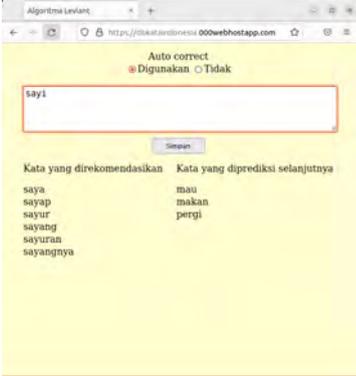
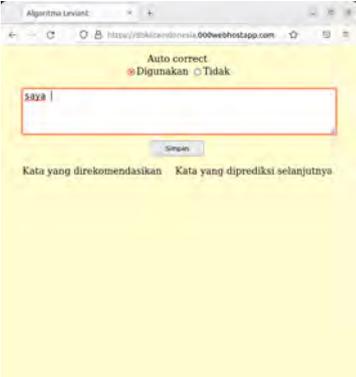
Test Case ID	Google Chrome Versi 101.0.4951.64	Mozilla Firefox Version 100.0	Microsoft Edge Version 101.0.1210.39
	 <p style="text-align: center;">Sesudah</p> 	 <p style="text-align: center;">Sesudah</p> 	 <p style="text-align: center;">Sesudah</p> 

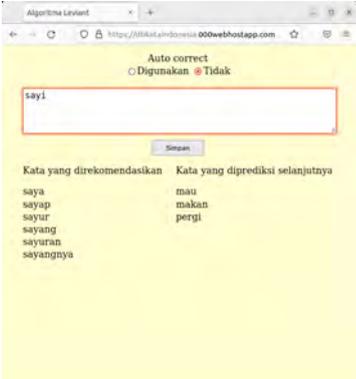
Test Case ID	Google Chrome Versi 101.0.4951.64	Mozilla Firefox Version 100.0	Microsoft Edge Version 101.0.1210.39
10	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 
11	 <p style="text-align: center;">Sesudah</p> 	 <p style="text-align: center;">Sesudah</p> 	 <p style="text-align: center;">Sesudah</p> 

Test Case ID	Google Chrome Versi 101.0.4951.64	Mozilla Firefox Version 100.0	Microsoft Edge Version 101.0.1210.39
12			

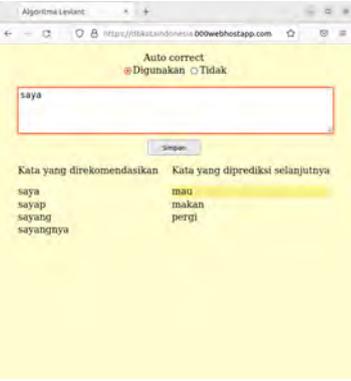
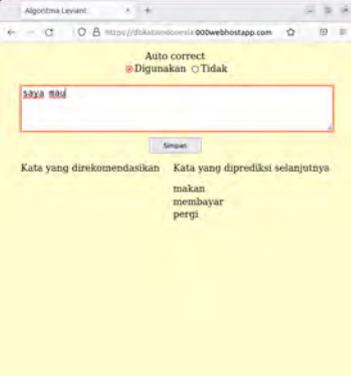
Ubuntu

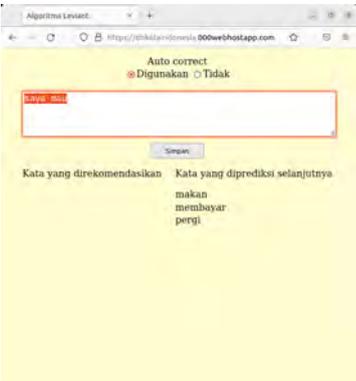
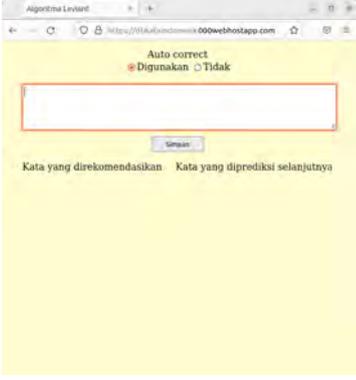
Test Case ID	Mozilla Firefox Versi 100.0	Google Chrome Versi 101.0.4951.64
1	<p>Sebelum</p>  <p>Sesudah</p> 	<p>Sebelum</p>  <p>Sesudah</p> 

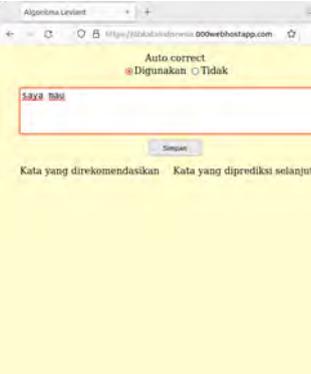
Test Case ID	Mozilla Firefox Versi 100.0	Google Chrome Versi 101.0.4951.64
2	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 
	3	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 

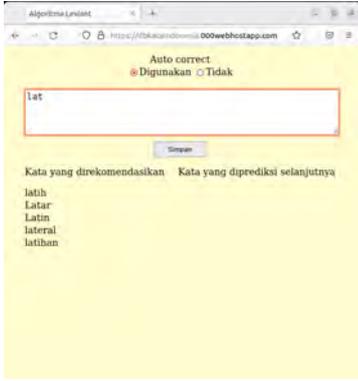
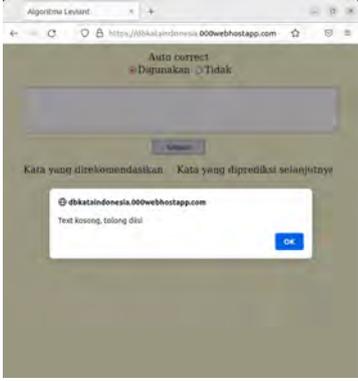
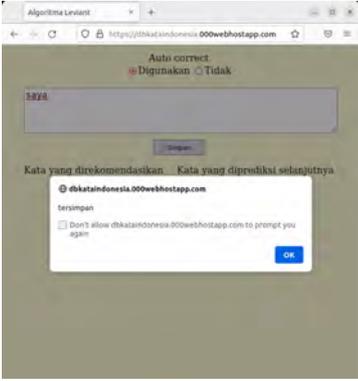
Test Case ID	Mozilla Firefox Versi 100.0	Google Chrome Versi 101.0.4951.64
4	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 

Test Case ID	Mozilla Firefox Versi 100.0	Google Chrome Versi 101.0.4951.64
5	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 

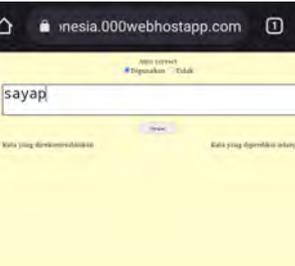
Test Case ID	Mozilla Firefox Versi 100.0	Google Chrome Versi 101.0.4951.64
6	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 

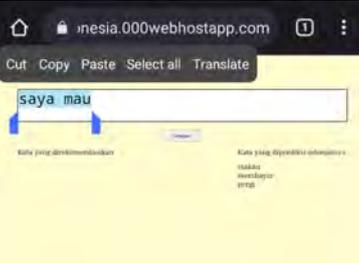
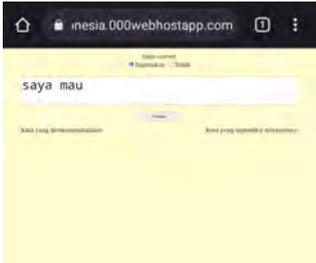
Test Case ID	Mozilla Firefox Versi 100.0	Google Chrome Versi 101.0.4951.64
7	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 
8		
9	Sebelum	Sebelum

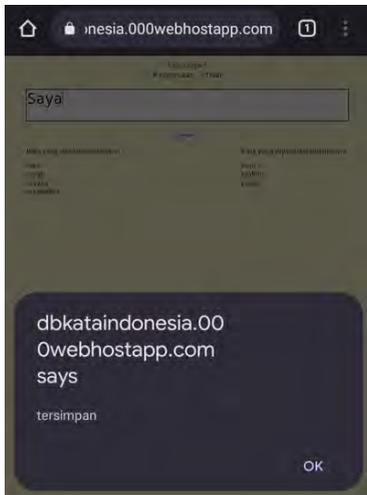
Test Case ID	Mozilla Firefox Versi 100.0	Google Chrome Versi 101.0.4951.64
	 <p>The screenshot shows the Mozilla Firefox browser interface. The address bar contains 'Algoritma Leviant'. The page title is 'Auto correct' with a blue 'Dipunakan' (Used) status. The input field contains 'saya mau makan'. Below the input field is a 'Simpan' (Save) button. Underneath, there are two columns of suggestions: 'Kata yang direkomendasikan' (Recommended words) and 'Kata yang diprediksi selanjutnya' (Predicted next words). The recommended words are 'makan', 'Makanan', 'makanan', 'Makanya', 'makananmu', and 'makanannya'. The predicted words are 'ikan' and 'nasi'.</p> <p style="text-align: center;">Sesudah</p>  <p>The screenshot shows the Mozilla Firefox browser interface after the correction. The address bar contains 'Algoritma Leviant'. The page title is 'Auto correct' with a blue 'Dipunakan' status. The input field now contains 'saya mau'. The 'Simpan' button is still present. The suggestion columns are now empty.</p>	 <p>The screenshot shows the Google Chrome browser interface. The address bar contains 'Algoritma Leviant'. The page title is 'Auto correct' with a red 'Dipunakan' status. The input field contains 'saya mau makan'. Below the input field is a 'Simpan' button. Underneath, there are two columns of suggestions: 'Kata yang direkomendasikan' and 'Kata yang diprediksi selanjutnya'. The recommended words are 'makan', 'makan', 'Makanan', 'Makanya', 'makananmu', and 'makanannya'. The predicted words are 'ikan' and 'nasi'.</p> <p style="text-align: center;">Sesudah</p>  <p>The screenshot shows the Google Chrome browser interface after the correction. The address bar contains 'Algoritma Leviant'. The page title is 'Auto correct' with a red 'Dipunakan' status. The input field now contains 'saya mau'. The 'Simpan' button is still present. The suggestion columns are now empty.</p>

Test Case ID	Mozilla Firefox Versi 100.0	Google Chrome Versi 101.0.4951.64
10	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 	<p style="text-align: center;">Sebelum</p>  <p style="text-align: center;">Sesudah</p> 
11		
12		

Android

Test Case ID	Oppo	
1	<p style="text-align: center;">Sebelum</p> 	<p style="text-align: center;">Sesudah</p> 
2	<p style="text-align: center;">Sebelum</p> 	<p style="text-align: center;">Sesudah</p> 
3	<p style="text-align: center;">Sebelum</p> 	<p style="text-align: center;">Sesudah</p> 
4	<p style="text-align: center;">Sebelum</p> 	<p style="text-align: center;">Sesudah</p> 
5	<p style="text-align: center;">Sebelum</p> 	<p style="text-align: center;">Sesudah</p> 

Test Case ID	Oppo	
6	<p style="text-align: center;">Sebelum</p> 	<p style="text-align: center;">Sesudah</p> 
7	<p style="text-align: center;">Sebelum</p> 	<p style="text-align: center;">Sesudah</p> 
8	<p style="text-align: center;">Sesudah</p> 	
9	<p style="text-align: center;">Sebelum</p> 	<p style="text-align: center;">Sesudah</p> 
10	<p style="text-align: center;">Sebelum</p> 	<p style="text-align: center;">Sesudah</p> 

Test Case ID	Oppo
11	
12	

Lampiran 3
Listing Program

```
index.html
<!DOCTYPE html>
<html>
<head>
  <title>Algoritma Leviant</title>
  <style type="text/css">
    body{
      padding-top: 1%;
      padding-left: 5%;
      padding-right: 6%;
      background-color: rgb(255, 253, 208);
      font-size: large;
    }
    .input{
      text-align: center;
    }
    .grup_radio{
      padding-bottom: 2%;
    }
    .kalimat{
      height: 80px;
      width: 99.80%;
      font-size: larger;
      margin-top: 10px;
    }
    .simpan{
      margin-top: 2%;
      height: 30px;
      width: 100px;
    }
    .output{
      padding-left: 0%;
      padding-right: 0%;
    }
    .judul{
      margin-bottom : 2px;
      margin-top : 0px;
    }
    .list{
      margin-bottom : 2px;
      margin-top : 0px;
    }
    .list:hover{
      background-color: rgb(248, 242, 141);
      cursor: pointer;
    }
    .judulRekomendasi{
      float: left;
    }
    .judulPrediksi{
      float: right;
    }
  </style>
  <script type="text/javascript" src="https://code.jquery.com/jquery-3.5.0.js"></script>
  <script type="text/javascript" src="script.js"></script>
```

```

</head>
<body>
  <div class="input">
    <p class="judul">Auto correct</p>
    <div class="grup_radio">
      <input type="radio" name="pilih" value='ya' checked='checked' />Digunakan
      <input type="radio" name="pilih" value='tdk' />Tidak
    </div>
    <textarea class="kalimat" name="kalimat"></textarea><br>
    <button class="simpan">Simpan</button>
  </div>
  <div class="output">
    <div class="judulRekomendasi">
      <p>Kata yang direkomendasikan</p>
      <div class="rekomendasi">
        <p class="list i0"></p>
        <p class="list i1"></p>
        <p class="list i2"></p>
        <p class="list i3"></p>
        <p class="list i4"></p>
        <p class="list i5"></p>
        <p class="list i6"></p>
        <p class="list i7"></p>
        <p class="list i8"></p>
        <p class="list i9"></p>
      </div>
    </div>
    <div class="judulPrediksi">
      <p>Kata yang diprediksi selanjutnya</p>
      <div class="prediksi">
        <p class="list p0"></p>
        <p class="list p1"></p>
        <p class="list p2"></p>
        <p class="list p3"></p>
        <p class="list p4"></p>
        <p class="list p5"></p>
        <p class="list p6"></p>
        <p class="list p7"></p>
        <p class="list p8"></p>
        <p class="list p9"></p>
      </div>
    </div>
  </div>
</body>
</html>

```

koneksi.php

```

<?php
  $servername = "localhost";
  $basis data = "id18867635_dbkata";
  $penggunaname = "id18867635_root";
  $password = "X\ _5q&HO?G{wvzCV";
  $conn = mysqli_connect($servername, $penggunaname, $password, $basis data);
  if (!$conn) {
    die("Koneksi Tidak Berhasil: " . mysqli_connect_error());
  }
?>

```

```

db.php
<?php
    require_once "koneksi.php";

    if(function_exists($_GET['function'])) {
        $_GET['function']();
    }
    function getData(){
        global $conn;

        $query = "SELECT * FROM `db1`";
        $resultData = $conn->query($query);
        $result = array();

        while($row = $resultData->fetch_assoc()) {
            array_push($result, array(
                'id' => $row['idKata'],
                'kata' => $row['kata'],
                'jumlah' => $row['jumlah']
            ));
        }
        echo json_encode($result);
    }

    function getDataKata(){
        global $conn;
        $kata = $_GET['kata'];
        $query = "SELECT * FROM `db1` WHERE kata LIKE '$kata%' ORDER BY `jumlah`
DESC, `kata` ASC";
        $resultData = $conn->query($query);
        $result = array();

        while($row = $resultData->fetch_assoc()) {
            array_push($result, array(
                'id' => $row['idKata'],
                'kata' => $row['kata'],
                'jumlah' => $row['jumlah']
            ));
        }
        echo json_encode($result);
    }

    function prediksiKata(){
        global $conn;
        $result = array();
        $kata = $_GET['kata'];
        $query = "SELECT `idKata` FROM `db1` WHERE `kata` = '$kata'";
        if(mysqli_num_rows(mysqli_query($conn, $query)) > 0){
            $idKata = mysqli_fetch_array(mysqli_query($conn, $query));

            $query = "SELECT kata FROM `dbkataprediksi` JOIN `db1` ON
dbkataprediksi.idKata2 = db1.idKata WHERE dbkataprediksi.idKata1 = $idKata[0] ORDER
BY dbkataprediksi.jumlah DESC";
            if(mysqli_num_rows(mysqli_query($conn, $query)) > 0){
                $resultData = $conn->query($query);

                while($row = $resultData->fetch_assoc()) {
                    array_push($result, array(

```

```

        'status' => True,
        'pesan' =>'Data tersedia',
        'kata' => $row['kata']
    ));
    }
}
else{
    $result=array(
        'status' => False,
        'pesan' =>'Tidak ada kata yang dapat diprediksi',
        'kata' => ""
    );
}
}
else{
    $result=array(
        'status' => False,
        'pesan' =>'Kata tidak ditemukan',
        'kata' => ""
    );
}
echo json_encode($result);
}

function updatePrediksi(){
    global $conn;
    $respon = "";
    if(empty($_GET['kata1']) && empty($_GET['kata2'])){
        $respon=array(
            'status' => False,
            'pesan' =>'Tidak ada data yang tersimpan'
        );
    }
    else{
        //mengecek apakah semua data sudah ada di basis data, jika sudah data akan
        //diupdate atau ditambah
        $kata1 = $_GET['kata1'];
        $kata2 = $_GET['kata2'];
        $query = "SELECT `idKata` FROM `db1` WHERE `kata` = '$kata1'";
        if(mysqli_num_rows(mysqli_query($conn, $query)) > 0){
            $idKata1 = mysqli_fetch_array(mysqli_query($conn, $query));

            $query = "SELECT `idKata` FROM `db1` WHERE `kata` = '$kata2'";
            if(mysqli_num_rows(mysqli_query($conn, $query)) > 0){
                $idKata2 = mysqli_fetch_array(mysqli_query($conn, $query));

                $query = "SELECT `jumlah` FROM `dbkataprediksi` WHERE `idKata1` =
                $idKata1[0] AND `idKata2` = $idKata2[0]";
                if(mysqli_num_rows(mysqli_query($conn, $query)) > 0){
                    $jumlah = mysqli_fetch_array(mysqli_query($conn, $query));

                    $query = "UPDATE `dbkataprediksi` SET `jumlah` = $jumlah[0]+1 WHERE
                    `idKata1` = $idKata1[0] AND `idKata2` = $idKata2[0]";
                    mysqli_query($conn, $query);
                }
                else{
                    $query = "INSERT INTO `dbkataprediksi` VALUES ($idKata1[0],
                    $idKata2[0], 1)";
                }
            }
        }
    }
}

```

```

        mysqli_query($conn, $query);
    }
    $respon=array(
        'status' => True,
        'pesan' =>'Data tersimpan '
    );
}
else{
    $respon=array(
        'status' => False,
        'pesan' =>'Kata2 tidak terbaca '
    );
}
}
else{
    $respon=array(
        'status' => False,
        'pesan' =>'Kata1 tidak terbaca '
    );
}
}
}
echo json_encode($respon);
}

function updateKata(){
    global $conn;
    $respon = "";
    if(empty($_GET['kata'])){
        $respon=array(
            'status' => False,
            'pesan' =>'Tidak ada data yang tersimpan'
        );
    }
    else{
        $kata = $_GET['kata'];
        //mengecek data ada atau tidak, jika tidak maka data akan ditambahkan
        $query = "SELECT * FROM `db1` WHERE `kata` = '$kata'";
        if(mysqli_num_rows(mysqli_query($conn, $query)) == 0){
            $query = "SELECT MAX(idKata) FROM `db1`";
            $id = mysqli_fetch_array(mysqli_query($conn, $query));

            $query = "INSERT INTO `db1` VALUES ($id[0]+1, '$kata', 1)";
            mysqli_query($conn, $query);
        }
        else{
            $query = "SELECT MAX(jumlah) FROM `db1`";
            $jumlah = mysqli_fetch_array(mysqli_query($conn, $query));

            $query = "UPDATE `db1` SET `jumlah`= $jumlah[0]+1 WHERE `kata` = '$kata'";
            mysqli_query($conn, $query);
        }
        $respon=array(
            'status' => True,
            'pesan' =>'Data terupdate'
        );
    }
}
echo json_encode($respon);

```

```
}  
?>
```

```
script.js  
var kata = "";  
var kataSebelumnya = "";  
var cekFokus = true;  
var cekClick = true;  
var space = false;  
var dbKata = new Array();  
var panjangKalimat = 0  
var cek = true;  
$(document).ready(function(){  
  for (let a = 0; a < 10; a++) {  
    $(".i"+a).click(function(){  
      var kalimat = $(".kalimat").val();  
      var hasil = kalimat.split(" ");  
      kalimat = "";  
      for (let b = 0; b < hasil.length - 1; b++) {  
        kalimat += hasil[b] + " ";  
      }  
      kalimat += $(".i"+a).text();  
      kataSebelumnya = $(".i"+a).text();  
      $(".kalimat").val(kalimat);  
      $.ajax({  
        type: "GET",  
        url:  
        "https://dbkataindonesia.000webhostapp.com/db.php?function=predisiKata&kata=" +  
        kataSebelumnya,  
        success:function(result){  
          var json = JSON.parse(result);  
          if(json.length > 10){  
            for (let b = 0; b < 10; b++) {  
              $(".p"+b).html(json[b]['kata']);  
            }  
          }  
          else if(json.length>0){  
            for (let b = 0; b < json.length; b++) {  
              $(".p"+b).html(json[b]['kata']);  
            }  
          }  
          else{  
            for (let b = 0; b < 10; b++) {  
              $(".p"+b).html("");  
            }  
          }  
        }  
      })  
      kata = "";  
      for (let b = 0; b < 10; b++) {  
        $(".i"+b).html("");  
      }  
      $(".kalimat").focus();  
      cekFokus = true;  
      cekClick = true;  
    })  
    $(".p"+a).click(function(){  
      kataSebelumnya = $(".p"+a).text();
```

```

$.kalimat).val($.kalimat).val() + " " + $(".p"+a).text());
$.ajax({
  type: "GET",
  url:
"https://dbkataindonesia.000webhostapp.com/db.php?function=predisiKata&kata=" +
kataSebelumnya,
  success:function(result){
    var json = JSON.parse(result)
    if(json.length > 10){
      for (let b = 0; b < 10; b++) {
        $(".p"+b).html(json[b]['kata']);
      }
    }
    else if(json.length>0){
      for (let b = 0; b < json.length; b++) {
        $(".p"+b).html(json[b]['kata']);
      }
    }
    else{
      for (let b = 0; b < 10; b++) {
        $(".p"+b).html("");
      }
    }
  }
})
kata = "";
for (let b = 0; b < 10; b++) {
  $(".i"+b).html("");
}
$(".kalimat").focus();
cekFokus = true;
cekClick = true;
})
}

$(".kalimat").focusout(function(){
  if(cekFokus){
    for (let i = 0; i < 10; i++) {
      $(".i"+i).css("display", "none");
      $(".p"+i).css("display", "none");
    }
  }
})
$(".rekomendasi").hover(function(){
  if(!cekClick){
    cekFokus=false;
  }
  for (let i = 0; i < 10; i++) {
    $(".i"+i).css("display", "block");
    $(".p"+i).css("display", "block");
  }
})
$(".prediksi").hover(function(){
  if(!cekClick){
    cekFokus=false;
  }
  for (let i = 0; i < 10; i++) {
    $(".i"+i).css("display", "block");
  }
})

```

```

        $(".p"+i).css("display", "block");
    }
})
$(".kalimat").focusin(function(){
    cekFokus = true;
    for (let i = 0; i < 10; i++) {
        $(".i"+i).css("display", "block");
        $(".p"+i).css("display", "block");
    }
})

$(".simpan").click(function(){
    if($(".kalimat").val() == ""){
        alert("Text kosong, tolong diisi!");
    }
    else{
        var kalimat = $(".kalimat").val();
        var kata1 = kalimat.split(" ");
        for (let i = 0; i < kata1.length; i++) {
            $.ajax({
                url:
                "https://dbkataindonesia.000webhostapp.com/db.php?function=updateKata&kata=" +
                kata1[i]
            })
            if(i < kata1.length-1){
                $.ajax({
                    url:
                    "https://dbkataindonesia.000webhostapp.com/db.php?function=updatePrediksi&kata1=" +
                    kata1[i] + "&kata2=" + kata1[i+1]
                })
            }
        }
        alert("tersimpan");
        kata = "";
        $(".kalimat").val("");
        for (let b = 0; b < 10; b++) {
            $(".i"+b).html("");
            $(".p"+b).html("");
        }
    }
    $(".kalimat").focus();
})
$('.kalimat').on('input',function(e){
    var kalimat = String($(".kalimat").val());
    hasil = kalimat.split(" ");
    kata = hasil[hasil.length-1];
    if(panjangKalimat < kalimat.length){
        cek = true;
    }
    else{
        cek = false;
    }
    // $(".test").html(panjangKalimat + " " + kalimat.length)
    panjangKalimat = kalimat.length;
    if($(".kalimat").val() == ""){
        for (let i = 0; i < 10; i++) {
            $(".i"+i).html("");
            $(".p"+i).html("");
        }
    }
}

```

```

    }
  }
  else if(kalimat[kalimat.length-1] == " "){
    var rekomendasi = $(".i0").text();
    var radioValue = $("input[name='pilih']:checked").val();
    if (radioValue == "ya" && cek) {
      var tmpkalimat = "";
      for (let i = 0; i < hasil.length - 2; i++) {
        tmpkalimat += hasil[i] + " ";
      }
      tmpkalimat += rekomendasi + " ";
      kataSebelumnya = rekomendasi;
      $(".kalimat").val(tmpkalimat);
    }
    kata = ""
    for (let i = 0; i < 10; i++) {
      $(".i"+i).html("");
      $(".p"+i).html("");
    }
  }
  else if($(".kalimat").val() != ""){
    $.ajax({
      type: "GET",
      url:
"https://dbkataindonesia.000webhostapp.com/db.php?function=getDataKata&kata=" +
kata,
      success:function(result){
        var json = JSON.parse(result);
        var panjangArray = 0;
        var tmpKata = new Array();

        if(json.length>0){
          dbKata = new Array();
          for (let i = 0; i < json.length; i++) {
            dbKata.push(String(json[i]['kata']));
          }
        }

        //algoritma leviant
        for (let i = 0; i < dbKata.length; i++) {
          var error = 0;
          var kata2 = dbKata[i];
          if(kata.length <= kata2.length){
            for (let j = 0; j < kata.length; j++) {
              if (kata[j] != kata2[j]) {
                error++;
              }
            }
            error += kata2.length - kata.length;
          }
          else {
            for (let j = 0; j < kata2.length; j++) {
              if (kata[j] != kata2[j]) {
                error++;
              }
            }
            error += kata.length - kata2.length;
          }
        }
      }
    }
  }

```

