**PROPOSAL**

### KOREKSI KESALAHAN EJAAN KATA KUNCI PADA PENCARIAN KAMUS PSIKOLOGI DENGAN ALGORITMA *JARO WINKLER DISTANCE* BERBASIS *MOBILE*

Diajukan Untuk Memenuhi

Salah Satu Syarat Memperoleh Gelar Sarjana Teknik



**ERICK HERDIAWAN**

**E1E1 16 009**

### JURUSAN TEKNIK IN*FORM*ATIKA

### FAKULTAS TEKNIK

### UNIVERSITAS HALU OLEO

### KENDARI

### 2020

## DAFTAR ISI

**HALAMAN PERSEMBAHAN...............................** Error! Bookmark not defined. **HALAMAN PERNYATAAN ..................................** Error! Bookmark not defined.

**HALAMAN PERSEMBAHAN............................................................................ v**

**INTI SARI............................................................................................................. ix**

**ABSTRACT ........................................................................................................... x**

**KATA PENGANTAR .......................................................................................... xi**

**DAFTAR ISI ....................................................................................................... xiii**

**DAFTAR TABEL................................................................................................ xv**

**DAFTAR GAMBAR .......................................................................................... xvi**

### BAB I PENDAHULUAN .................................................................................... 17

1.1. Latar Belakang........................................................................................ 17

1.2. Rumusan Masalah .................................................................................. 18

1.3. Batasan Masalah ..................................................................................... 19

1.4. Tujuan Penelitian .................................................................................... 19

1.5. Manfaat Penelitian .................................................................................. 19

1.6. Sistematika Penulisan ............................................................................. 19

1.7. Tinjauan Pustaka .................................................................................... 20

### BAB II LANDASAN TEORI ............................................................................. 22

2.1. Kamus ..................................................................................................... 22

2.2. Psikologi ................................................................................................. 23

2.3. *Typographical Error* .............................................................................. 23

2.4. *Spelling Correction* ................................................................................ 24

2.5. Algoritma *Jaro Winkler Distance* .......................................................... 24

2.6. *Speech To Text* (STT) ............................................................................. 28

2.7. *Precision* ................................................................................................. 28

2.8. *Flowchart* ............................................................................................... 28

2.9. *Unified Modeling Language* (UML) ...................................................... 30

2.10. *Ionic Framework* .................................................................................... 36

2.11. *Atom Text Editor*..................................................................................... 36

2.12. *Typescript* ............................................................................................... 36

2.13. JSON....................................................................................................... 37

2.14. Data Kamus ............................................................................................ 38

2.15. RUP ........................................................................................................ 38

### BAB III METODOLOGI PENELITIAN ......................................................... 41

3.1. Waktu dan Tempat Penelitian ................................................................ 41

3.2. Metode Pengumpulan Data .................................................................... 41

3.3. Metode Pengembangan Sistem............................................................... 42

3.4. Analisis Kebutuhan ................................................................................ 44

4.5. Analisis Perancangan Sistem .................................................................. 45

### DAFTAR PUSTAKA .......................................................................................... 72

[DAFTAR TABEL](#_Toc107124)

[Tabel 2.1. Simbol-Simbol *Flowchart* 29](#_Toc107125)

[Tabel 2.2. Simbol-simbol *Use case Diagram* 31](#_Toc107126)

[Tabel 2.3. Simbol-simbol *Class Diagram* 32](#_Toc107127)

[Tabel 2.4. Simbol-simbol *Activity Diagram* 33](#_Toc107128)

[Tabel 2.5. Simbol-simbol *Sequence Diagram* 34](#_Toc107129)

[Tabel 3.1. *Gantt Chart* Waktu Penelitian 41](#_Toc107130)

[Tabel 3.2. Spesifikasi Perangkat Keras 44](#_Toc107131)

[Tabel 3.3. Spesifikasi Perangkat Lunak 45](#_Toc107132)

[Tabel 3.4. Keterangan *Use Case Diagram* 48](#_Toc107133)

[Tabel 3.5. Rancangan Pengujian Aplikasi 54](#_Toc107134)

**DAFTAR GAMBAR**

Gambar 2.1 Flowchart *Jaro Winkler Distance* ..................................................... 28

Gambar 2.2 Proses Iteratif RUP ............................................................................. 39

Gambar 4.1 *Flowchart* Sistem .............................................................................. 46

Gambar 4.2 *Flowchart* Algoritma *Jaro Winkler Distance* .................................... 47

Gambar 4.3 *Use Case Diagram* Kamus Psikologi ................................................ 48

Gambar 4.4 *Activity Diagram* Pencarian Kata ...................................................... 49

Gambar 4.5 *Activity Diagram* *Voice Search* ......................................................... 50

Gambar 4.6 *Activity Diagram* Koreksi Kata ......................................................... 51

Gambar 4.7 *Class Diagram* Kamus Psikologi ...................................................... 52

Gambar 4.8 *Sequence Diagram* Kamus ................................................................ 52

**BAB I**

**PENDAHULUAN**

#### 1.1. Latar Belakang

Kamus adalah sejenis buku rujukan yang menerangkan makna kata-kata. Pada umumnya orang membuka kamus untuk mengetahui makna atau arti sebuah kata yang belum diketahuinya. Fungsi utama sebuah kamus adalah menjelaskan makna atau arti sebuah kata (Kurniasih, 2014). Kamus biasanya identik dengan istilah-istilah popular yang menerangkan kata-kata ilmiah pada bidang ilmu tertentu yang jarang ditemui dalam kehidupan sehari-hari, misalnya kamus komputer dan kamus psikologi seperti yang akan penulis paparkan yakni kamus psikologi.

Kamus psikologi adalah kamus yang mencakup istilah-istilah psikologi, termasuk di dalamnya mengenai istilah kelainan mental, istilah atas perasaan manusia dan lain lain. Dalam kamus psikologi, in*form*asi yang disampaikan harus jelas. Menurut Widianingsih (dalam Prasetyo et al., 2018) bahwa dalam unsur bahasa ragam tulis, in*form*asi yang disampaikan secara tertulis harus jelas. Widianingsih menambahkan, bahwa dalam bahasa ragam tulis unsur-unsur bahasa yang dipergunakan harus lengkap. Jika ada unsur-unsur itu tidak lengkap, maka ada kemungkinan in*form*asi yang disampaikan tidak dapat dipahami secara tepat.

Beberapa jenis kamus yang yang ada di perpustakaan atau yang beredar di beberapa toko buku dirasa masih kurang memenuhi kebutuhan manusia. Kamus identik dengan bentuk buku tebal yang sulit dibawa kemana-mana dan rentannya kesalahan penulisan kata, mengingat kamus dengan bentuk buku tebal dibuat secara manual. Beberapa faktor yang menyebabkan terjadinya kesalahan pengetikan adalah letak huruf pada keyboard yang berdekatan, kesalahan karena slip pada tangan atau jari, kesalahan yang disebabkan oleh ketidaksengajaan (Rochmawati & Kusumaningrum, 2016). Salah satu fitur yang dapat digunakan untuk mendeteksi kesalahan dan memberikan sugesti kata yang benar adalah fitur *spelling corrector* atau *spelling checker* atau *spelling suggestion*. Fitur ini berfungsi sebagai pendeteksi kesalahan dan memberikan panduan bagi penggunanya dengan mendanai kata-kata yang tidak terdaftar dalam kamus suatu bahasa tertentu. Fitur

ini juga disertai dengan sugesti kata yang berfungsi menyediakan rekomendasi kata-kata yang mendekati kata yang dimaksud (Sujaini & Nyoto, 2016).

Menurut Gueddah, Yousfi, & Belkasmi (dalam Prasetyo et al., 2018) menjelaskan bahwa pemeriksaan ejaan terdiri dari perbandingan antara kata yang salah dengan daftar kata pada basis data dan menyarankan kata-kata yang mirip dengan kata yang salah dengan menghitung kemiripan jarak antara kata-kata tersebut. Untuk mengurangi kesalahan ejaan yang ada pada kamus saat ini, maka diperlukan kamus elektronik yang disuguhi dengan fitur *autocorrect*. Selain efisien, kamus ini juga dapat membantu pengguna yang salah mengetikkan istilah yang ingin mereka cari dengan menampilkan rekomendasi kata yang mendekati dengan kata tersebut.

Dalam penelitian ini, fitur *autocorrect* dibangun dengan menggunakan Algoritma *Jaro Winkler Distance.* Rochmawati dan Kusumaningrum (2016) mengujicobakan 50 kata yang salah sebagai masukan untuk proses identifikasi kesalahan pengetikan dan saran perbaikan berdasarkan *algoritma Hamming Distance, Levenshtein Distance, Damerau Levenshtein Distance* dan *Jaro-Winkler Distance*. Hasilnya menunjukkan bahwa algoritma *Jaro-Winkler Distance* memiliki nilai tertinggi dibandingkan ketiga algoritma yang lain dengan nilai MAP 0,87 yang terbagi ke dalam empat jenis kesalahan penulisan yaitu jenis kesalahan penghapusan huruf 0,92, jenis kesalahan penambahan huruf 0,90, jenis kesalahan penggantian huruf 0,70 dan jenis kesalahan penukaran huruf 0,95. Dari penelitian tersebut dapat disimpulkan bahwa algoritma *Jaro-Winkler Distance* merupakan algoritma terbaik dalam pengkoreksian kata. Atas dasar inilah peneliti mengangkat judul ***“*Koreksi Kesalahan Ejaan Kata Kunci pada Pencarian Kamus Psikologi dengan Algoritma *Jaro Winkler Distance* Berbasis *Mobile”*.**

#### 1.2. Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dipaparkan, maka rumusan masalah dalam penelitian ini adalah bagaimana mengimplementasikan Algoritma *Jaro Winkler Distance* untuk mengoreksi kesalahan ejaan pada saat mencari kata di Kamus Psikologi.

#### 1.3. Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut :

1. Menggunakan Algoritma *Jaro Winkler Distance* sebagai metode pengoreksian kesalahan pencarian kata pada kamus psikologi.
2. Sistem mengoreksi kata yang salah dengan memberikan saran kata yang mendekati kata *input*an.
3. Sistem menggunakan penyimpanan dalam *form*at JSON.
4. Dibangun menggunakan *framework* *ionic*.
5. Dibangun berbasis *mobile*.
6. Pencarian dapat menggunakan suara.

#### 1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengimplementasikan Algoritma *Jaro Winkler Distance* untuk mengoreksi kesalahan ejaan pada pencarian kata di Kamus Psikologi.

#### 1.5. Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini untuk memberikan solusi bagi pengguna yang sering kali salah dalam mengetikkan istilah ketika ingin mencari kata pada kamus elektronik dan sering kali mereka tidak mengetahui penulisan yang tepat dari istilah yang dicari. Dalam kamus ini, sistem akan secara otomatis memberikan kata saran yang mendekati dengan kata *input*an.

#### 1.6. Sistematika Penulisan

Sistematika penulisan tugas akhir digunakan untuk menjelaskan penulisan per bab sebagai berikut:

### BAB I PENDAHULUAN

Merupakan bab pendahuluan yang menguraikan latar

belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

### BAB II LANDASAN TEORI

Membahas mengenai dasar-dasar teori pendukung yang berhubungan dengan masalah yang diambil dan program aplikasi yang akan digunakan dalam pembangunan sistem.

### BAB III METODE PENELITIAN

Membahas mengenai waktu dan tempat penelitian, metode pengumpulan data, metode pembangunan sistem yang digunakan dalam penelitian tugas akhir ini.

### BAB IV ANALISIS DAN PERANCANGAN

Bab ini akan membahas tentang analisis dan perancangan dari aplikasi yang akan dibuat, dengan menggunakan desain UML (*Unified Modelling Language*) serta desain *user interface*.

### BAB V IMPLEMENTASI DAN PEMBAHASAN

Dalam bab ini akan dikaji mengenai implementasi hasil perancangan aplikasi yang dibuat serta melakukan pengujian terhadap sistem.

### BAB VI PENUTUP

Bab ini berisi kesimpulan dari program yang telah dibuat serta saran yang diperlukan untuk pengembangan program berikutnya.

#### 1.7. Tinjauan Pustaka

Berdasarkan penelitian dilakukan oleh Agung Prasetyo, Wiga Maulana Baihaqi, Iqbaluddin Syam Had (2018) tentang penerapan algoritma *Jaro-Winkler* *Distance* untuk fitur *autocorrect* dan *spelling suggestion* dapat disimpulkan bahwa fitur yang telah dibuat dapat menangani kesalahan penulisan ejaan kata pada penulisan naskah bahasa Indonesia. Dari hasil pengujian terhadap 60 kata yang terdiri dari berbagai kesalahan penulisan ejaan, fitur *autocorrect* dan *spelling suggestion* dapat menangani kesalahan penulisan ejaan pada 49 kata dengan baik. Sedangkan pada kata lainnya terjadi kesalahan dalam menampilkan saran ejaan.

Yeny Rochmawati dan Retno Kusumaningrum (2015) menyimpulkan bahwa perbandingan algoritma pencarian *string* dalam metode *Approximate String Matching* pada kedua eksperimen menunjukkan bahwa algoritma *Jaro Winkler Distance* memiliki nilai tertinggi dibandingkan ketiga algoritma yang lain (*Hamming Distance, Levenshtein Distance, Damerau Levenshtein Distance*) dengan nilai MAP 0,87 yang terbagi ke dalam empat jenis kesalahan penulisan yaitu jenis kesalahan penghapusan huruf 0,92, jenis kesalahan penambahan huruf 0,90, jenis kesalahan penggantian huruf 0,70 dan jenis kesalahan penukaran huruf 0,95. Yulianingsih, 2017 menyimpulkan bahwa pada algoritma *Jaro-Winkler* semakin besar nilai skor yang dimiliki maka kemiripan semakin tinggi. Pada algoritma *Jaro-Winkler* memiliki kompleksitas waktu *quadratic runtime complexity* yang sangat efektif pada *string* pendek dan dapat memperoleh hasil lebih cepat dari algoritma *Levenshtein Distance*. Berdasarkan penelitian-penelitian yang telah dipaparkan di atas, maka akan dilakukan penelitian lebih lanjut dengan judul **“Koreksi Kesalahan Ejaan Kata Kunci pada Pencarian Kamus Psikologi Dengan Algoritma *Jaro Winkler Distance* Berbasis *Mobile”.***

### BAB II

### LANDASAN TEORI

#### 2.1. Kamus

Dua nama besar yang mengawali penyusunan kamus adalah Samuel Johnson (1709-1784) dan Noah Webster (1758-1843). Johnson, ahli bahasa dari Inggris, membuat *Dictionary of the English Language* pada tahun 1755, yang terdiri atas dua volume. Di Amerika, Webster pertama kali membuat kamus *An American Dictionary of the English Language* pada tahun 1828, yang juga terdiri atas dua volume. Selanjutnya, pada tahun 1884 diterbitkan *Oxford English Dictionary* yang terdiri atas 12 volume.

Kamus ekabahasa pertama di Indonesia merupakan kamus bahasa Melayu yang ditulis oleh Raja Ali Haji, berjudul Kitab Pengetahuan Bahasa, yaitu Kamus Loghat Melayu-Johor-Pahang-Riau-Lingga penggal yang pertama. Kamus ini terbit pada abad ke-19. Kitab Pengetahuan Bahasa sebenarnya bukan kamus murni namun merupakan kamus ensiklopedia untuk keperluan pelajar.

Kamus bisa ditulis dalam satu atau lebih dari satu bahasa. Dengan itu kamus bisa dibagi menjadi beberapa jenis yaitu:

1. Kamus Ekabahasa

Kamus ini hanya menggunakan satu bahasa (misal dalam satu kamus hanya terdapat bahasa Indonesia saja atau bahasa Inggris saja). Kata-kata (entri) yang dijelaskan dan penjelasannya adalah terdiri dari bahasa yang sama. Contohnya Kamus Besar Bahasa Indonesia.

1. Kamus Dwibahasa

Kamus ini menggunakan dua bahasa, yakni kata masukan daripada bahasa yang dikamuskan diberi padanan atau pemberian takrifnya dengan menggunakan bahasa yang lain. Contohnya: Kamus Inggris-Indonesia.

1. Kamus Aneka Bahasa

Kamus ini sekurang-kurangnya menggunakan tiga bahasa atau lebih.

Misalnya, kata Bahasa Melayu Bahasa Inggris dan Bahasa Mandarin secara

serentak. Contoh kamus aneka bahasa ialah Kamus Melayu-Cina-Inggris Pelangi susunan Yuen Boon Chan pada tahun 2004.

Kamus adalah sejenis buku rujukan yang menerangkan makna kata-kata, yang berfungsi untuk membantu seseorang mengenal perkataan baru. Selain menerangkan maksud kata, kamus juga mungkin mempunyai pedoman sebutan, asal-usul (etimologi) sesuatu perkataan dan juga contoh penggunaan bagi sesuatu perkataan. Untuk memperjelas kadang kala terdapat juga ilustrasi di dalam kamus.

Kata kamus diserap dari bahasa Arab *qamus*, dengan bentuk jamaknya *qawamis*. Kata Arab itu sendiri berasal dari kata Yunani *okeanos* yang berarti 'lautan'. Sejarah kata itu jelas memperlihatkan makna dasar yang terkandung dalam kata kamus, yaitu wadah pengetahuan, khususnya pengetahuan bahasa, yang tidak terhingga dalam dan luasnya. Dewasa ini kamus merupakan *khazanah* yang memuat perbendaharaan kata suatu bahasa, yang secara ideal tidak terbatas jumlahnya. Setiap kebudayaan besar di dunia bangga akan kamus bahasanya. Dalam kenyataannya kamus itu tidak hanya menjadi lambang kebanggaan suatu bangsa, tetapi juga mempunyai fungsi dan manfaat praktis (Renditia, 2005).

#### 2.2. Psikologi

Dikutip dari situs Kompasian.com, psikologi berasal dari bahasa Yunani yaitu *psyche* yang artinya jiwa dan *logos* artinya ilmu. Jadi secara etimologi psikologi artinya ilmu yang mempelajari jiwa, baik mengenai macam-macam gejalanya, proses maupun latar belakangnya.

Berikut beberapa penjelasan psikologi menurut ahli:

1. Edwin G. Boring dan Herbert S. Langfeld.

Psikologi adalah ilmu yang mempelajari tentang hakekat manusia.

1. Garden Murphy

Psikologi adalah ilmu yang mempelajari respon tentang mahluk hidup dengan lingkungannya.

#### 2.3. *Typographical Error*

Menurut Naradhipa et al dalam (Fahma, 2018)bahwa *Typographical error* merupakan kesalahan yang terjadi pada saat proses mengetik teks dan dapat mengubah arti dari suatu kata bahkan arti dari suatu kalimat. Istilah ini mencakup kesalahan karena kegagalan mekanis atau slip tangan atau jari, dan juga timbul akibat ketidaktahuan penulis seperti kesalahan ejaan. *Typographical error* dapat disebabkan oleh, misalnya jari menekan dua tombol *keyboard* yang berdekatan secara bersamaan.

*Typgraphical error* ini bervariasi mulai dari kesalahan ketik biasa sampai kesalahan dalam tatanan bahasa yang digunakan atau bahkan pengertian dari kata tersebut. Kesalahan-kesalahan tersebut dikategorikan ke dalam 2 jenis yaitu *nonword error* dan *real-word error*.

#### 2.4. *Spelling Correction*

*Spelling correction* adalah proses mendeteksi dan memberikan saran untuk kata-kata yang salah eja di dalam suatu teks. Sedangkan *spelling corrector* merupakan fitur atau aplikasi yang akan melakukan proses tersebut. Fitur ini mencari kata-kata yang salah berdasarkan data korpus yang digunakan aplikasi. Sedangkan saran kata diberikan dengan perhitungan algoritma yang juga digunakan oleh aplikasi. Pemeriksa ejaan terbagi menjadi dua jenis, yaitu : pemeriksa kesalahan yang bukan kata dan pemeriksa kesalahan kata yang sebenarnya. Pemeriksa kesalahan yang bukan kata berfokus pada penanganan kata yang salah eja yang disebabkan oleh kesalahan tipografi. Sedangkan pemeriksa kesalahan kata yang sebenarnya ditekankan pada penanganan kesalahan penempatan kata dalam sebuah kalimat (Sujaini & Nyoto, 2016).

#### 2.5. Algoritma *Jaro Winkler Distance*

Algoritma *Jaro Winkler Distance* merupakan varian dari *Jaro Distance metric* yaitu sebuah algoritma untuk mengukur kesamaan antara dua *string*, biasanya algoritma ini digunakan di dalam pendeteksian duplikat. Semakin tinggi *Jaro Winkler Distance* untuk dua *string* maka semakin mirip dengan *string* tersebut. Skor normalnya seperti 0 menandakan tidak ada kesamaan, dan 1 adalah sama persis. Dasar dari algoritma ini memiliki tiga bagian: (1) Menghitung panjang *string*. (2) Menemukan jumlah karakter yang sama di dalam dua *string*. (3) Menemukan jumlah transposisi. Pada algoritma *Jaro* digunakan Persamaan 1 untuk menghitung jarak (dj) antara dua *string* yaitu 𝑠1dan 𝑠2 dimana :

*m*: jumlah karakter yang sama persis

𝑠1: panjang *string* 1 𝑠2: panjang *string* 2 *t*: jumlah transposisi

Jarak teoritis dua buah karakter yang dianggap sama dapat dikatakan benar jika tidak melebihi batas seperti yang tercantum pada Persamaan 2.

*Jaro Winkler Distance* menggunakan *prefix scale* (*p*) yang memberikan tingkat penilaian yang lebih dan *prefix length* (*ℓ*) yang menyatakan panjang awalan yaitu panjang karakter yang sama dengan *string* yang dibandingkan sampai ditemukan ketidaksamaan, jumlah karakter yang sama sebelum ditemukan ketidaksamaan adalah maksimal 4. Bila *string* 𝑠1dan 𝑠2 yang dibandingkan, maka *Jaro Winkler Distance*-nya ditentukan dengan Persamaan 3, dimana (Rochmawati & Kusumaningrum, 2016):  *dj*: Jaro Distance untuk *string* 𝑠1dan 𝑠2  *ℓ:* panjang prefiks umum di awal *string* (panjang karakter yang sama sebelum ditemukan ketidaksamaan max 4) *p*: konstanta *scaling factor* (p=0,1)

*dj* = 1( 𝑚 + 𝑚 + 𝑚−𝑡) (1)

3 |𝑠1| |𝑠2| 𝑚 jarak max = (max⁡(|𝑠1|,|𝑠2|)) – 1 (2)

2

*dw* = dj + (*ℓp*(1-dj)) (3)

Misalkan pengguna berniat menulis “delirium” tetapi terjadi kesalahan sehingga tertulis “dellirium”. Tidak ditemukan kata “dellirium” pada daftar kata di *database* sehingga kata “dellirium” akan dihitung jarak kedekatannya dengan semua kata yang terdaftar di *database* menggunakan algoritma *Jaro-Winkler Distance*. Berikut perhitungan manual nilai *distance*-nya :

𝑠1 = dellirium⁡dan 𝑠2 = delirium

Perhitungan range uji (jarak max) :

= () – 1 = 3,5 untuk index uji didepan dan belakang

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **D** |  | **E** | **L** | **L** | **I** | **R** | **I** | **U** | **M** |
|  |  |  |  |  |  |  |  |  |  |
| **D** |  | **E** | **L** | **I** | **R** | **I** | **U** | **M** |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **D** | **E** | **L** | **L** | **I** | **R** | **I** | **U** | **M** |
|  |  |  |  |  |  |  |  |  |
| **D** | **E** | **L** | **I** | **R** | **I** | **U** | **M** |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **D** | **E** | **L** | **L** | **I** | **R** | **I** | **U** | **M** |
|  |  |  |  |  |  |  |  |  |
| **D** | **E** | **L** | **I** | **R** | **I** | **U** | **M** |  |

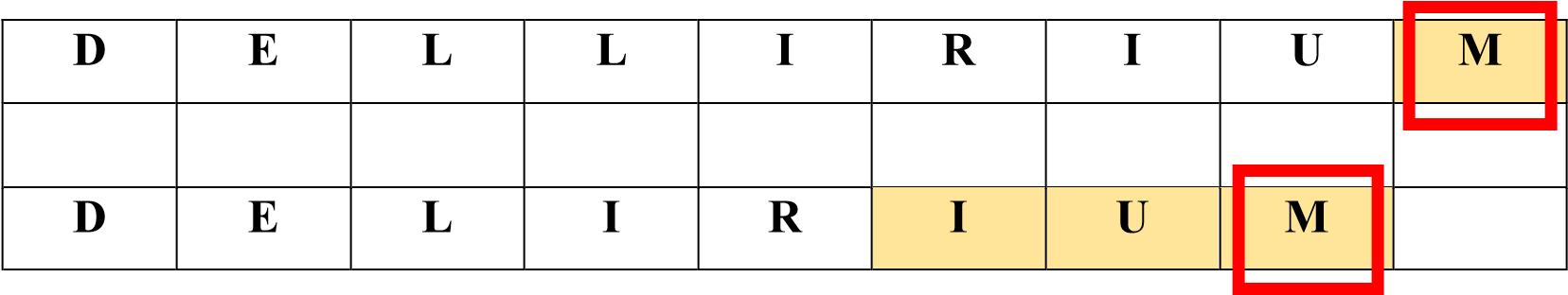
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **D** | **E** | **L** | **L** | **I** | **R** | **I** | **U** | **M** |
|  |  |  |  |  |  |  |  |  |
| **D** | **E** | **L** | **I** | **R** | **I** | **U** | **M** |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **D** | **E** | **L** | **L** |  | **I** | **R** | **I** | **U** | **M** |
|  |  |  |  |  |  |  |  |  |  |
|  |
| **D** | **E** | **L** | **I** | **R** | | **I** | **U** | **M** |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **D** | **E** | **L** | **L** | **I** |  | **R** | **I** | **U** | **M** |
|  |  |  |  |  |  |  |  |  |  |
| **D** | **E** | **L** | **I** | **R** |  | **I** | **U** | **M** |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **D** | **E** | **L** | **L** | **I** | **R** |  | **I** | **U** | **M** |
|  |  |  |  |  |  |  | |  |  |
| **D** | **E** | **L** | **I** | **R** | **I** | **U** | | **M** |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **D** | **E** | **L** | **L** | **I** | **R** | **I** |  | **U** | **M** |
|  |  |  |  |  |  |  |  | |  |
|  |
| **D** | **E** | **L** | **I** | **R** | **I** | **U** | **M** | |  |



𝑚 = 8

𝑠1 = 9

𝑠2 = 8

𝑡 = 0

Sehingga nilai *Jaro distance*-nya adalah : *dj* = 1( 𝑚 + 𝑚 + 𝑚−𝑡); dj = 1 (8 + 8 + 8−0) = 0,96

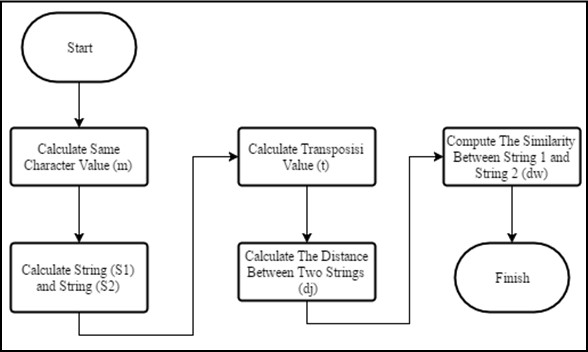
3 |𝑠1| |𝑠2| 𝑐 3 9 8 8

Bila diperhatikan susunan 𝑠1 dan 𝑠2 dapat diketahui nilai *prefix length* (ℓ

)=3 karena karakter yang sama sebelum ditemukan kesalahan adalah karakter ‘d’, ‘e’, dan karakter ‘l’. Nilai konstanta *scaling factor p*=0,1. Maka nilai *Jaro Winkler Distance* adalah : 0.2

*dw* = *dj* + (*p*(1− *dj*)) **;** *dw* = 0,96 + (3x 0,1(1−0,96)) = 0,972

Oleh karena itu nilai *distance* untuk *Jaro Winkler* pada kata yang salah yaitu dellirium dan kata kamus yang menjadi target yaitu delirium adalah 0,972.



### Gambar 2.1. *Flowchart Jaro Winkler Distance* (Brinardi Leonardo dan Seng Hansun, 2017)

#### 2.6. *Speech To Text* (STT)

Prasanthi (dalam Nugroho, 2019) menjelaskan bahwa STT merupakan satu bentuk teknologi yang bisa mengenali ucapan manusia untuk kemudian dikonversikan dalam bentuk teks. Teknologi ini sudah mulai diadopsi oleh *Apple* dalam bentuk aplikasi *SIRI* yang terbenam dalam *Iphone* maupun *Google Voice* yang mengembeded aplikasi tersebut dalam *smartphone* berbasis *Android*. Konversi ucapan dari suara ke teks sangat menguntungkan dan digunakan di berbagai bidang aplikasi yaitu sangat berguna bagi siswa tuna 39 INFOKAM Nomor I Th. XV/MARET/2019 rungu dan orang-orang cacat fisik.

#### 2.7. *Precision*

*Precision* adalah pengujian untuk mengetahui baik tidaknya metode yang digunakan pada penelitian ini dengan membandingkan antara jumlah dokumen relevan yang ditemukan dengan jumlah keseluruhan dokumen yang ditemukan.

Berikut *form*ula untuk perhitungan *precision* dijabarkan pada persamaan (4) :

*Precision* = 𝑡𝑝 (Suka Parwita & Winarko, 2015) (4)

𝑡𝑝+𝑓𝑝

#### 2.8. *Flowchart*

*Flowchart* atau diagram alir merupakan sebuah diagram dengan simbolsimbol grafis yang menyatakan aliran algoritma atau proses berjalannya program, *Flowchart* adalah suatu diagram yang berupa simbol-simbol dan dapat menunjukkan alur data serta operasi yang terjadi pada suatu sistem. Bagan alur digunakan sebagai alat bantu komunikasi dan dokumentasi (Jogiyanto HM., 2005). Bagan alur sistem digambarkan dengan menggunakan simbol-simbol yang tampak pada tabel 2.2.

# Tabel 2.1. Simbol-Simbol *Flowchart*

|  |  |  |  |
| --- | --- | --- | --- |
| **NO.** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1. |  | *Terminator* | Permulaan atau akhir program. |
| 2. |  | *Garis Alir (Flow Line)* | Arah aliran program. |
| 3. |  | *Preparation* | Proses inisialisasi/ pemberian harga awal. |
| 4. |  | *Proses* | Proses perhitungan / proses pengolahan data. |
| 5. |  | *Input / output data* | Proses *input*/*output* data, parameter, in*form*asi. |
| 6. |  | *Predefined*  *Process* | Permulaan sub program/ proses menjalankan sub program. |
| 7. |  | *Decision* | Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya. |
| 8. |  | *On Page Connector* | Keluar atau masuk dari bagian lain *flowchart* khususnya |
| 9. |  | *Off Page*  *Connector* | Keluar atau masuknya dari bagian lain *flowchart* khususnya halaman lain. |

#### 2.9. *Unified Modeling Language* (UML)

Menurut Sukamto & Shalahuddin (dalam Sunarti, 2016) menjelaskan bahwa UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML (*Unified Modelling Language*) menyediakan serangkaian gambar dan diagram yang sangat baik. Beberapa diagram memfokuskan diri pada ketangguhan teori *object-oriented* dan sebagian lagi memfokuskan pada detail rancangan dan konstruksi. Semua dimaksudkan sebagai sarana komunikasi antar *team programmer* maupun dengan pengguna. Menurut Widodo & Herlawati (dalam Zahara, 2014) menegaskan bahwa UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk:

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

UML mendefinisikan diagram-diagram sebagai berikut:

##### 2.6.1. *Use case Diagram*

Menurut Sukamto & Shalahuddin (dalam Sunarti, 2016) menjelaskan bahwa *use case* atau diagram *use case* merupakan pemodelan untuk kelakuan *(behavior)* sistem in*form*asi yang akandibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktordengan sistem in*form*asi yang akan dibuat.Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagaiberikut:

# Tabel 2.2. Simbol-simbol *Use case Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Simbol** | **Nama** | **Deskripsi** |
| 1 |  | *Actor* | Orang, proses, atau sistem lain yang berinteraksi dengan sistem in*form*asi yang akan dibuat di luar sistem in*form*asi yang akan dibuat itu sendiri. |
| 2 |  | *Use case* | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. |
| 3 |  | *Association* | Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor. |
| 4 | <<extend>> | *Extend* | Relasi *use case* tambahan ke sebuah *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu. |
| 5 |  | *Generalization* | Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. |
| 6 | <<include>> | *Include* | Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini. |

#### 2.6.2. *Class Diagram*

Menurut Sukamto & Shalahuddin (dalam Sunarti, 2016) bahwa *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Adapun simbolsimbol yang digunakan dalam *class diagram* adalah sebagai berikut:

# Tabel 2.3. Simbol-simbol *Class Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Deskripsi** |
| 1 |  | *Class* | Kelas pada stuktur sistem. |
| 2 |  | *Interface* | Sama dengan konsep *interface* dalam pemrograman berorientasi objek. |
| 3 |  | *Association* | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*. |
| 4 |  | *Directed association* | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain. |
| 5 |  | *Generalization* | Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus). |
| 6 |  | *Dependency* | Relasi antar kelas dengan makna kebergantungan antar kelas. |
| 7 |  | *Aggregation* | Relasi antar kelas dengan makna semua bagian *(whole-part)*. |

#### 2.6.3. *Activity Diagram*

Menurut Sukamto & Shalahuddin (dalam Sunarti, 2016) menjelaskan bahwa *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Adapun simbol-simbol yang digunakan dalam *activity diagram* adalah sebagai berikut:

# Tabel 2.4. Simbol-simbol *Activity Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Simbol** | **Nama** | **Deskripsi** |
| 1 |  | Status awal | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| 2 |  | Aktivitas | Aktivitas yang dilakukan sistem, biasanya diawali dengan kata kerja. |
| 3 |  | Decision | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| 4 |  | *Join* | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| 5 |  | Status akhir | Status akhir yang dilakukan sebuah sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| 6 | |  | | --- | |  | |  | | *Swimlane* | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

#### 2.6.4. *Sequence Diagram*

Menurut Sukamto & Shalahuddin (dalam Sunarti, 2016) bahwa diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. *Sequence diagram* menunjukkan urutan *event* kejadian dalam suatu waktu. Komponen *sequence* diagram terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progress vertikal. Simbol-simbol yang digunakan dalam *sequence diagram* adalah:

# Tabel 2.5. Simbol-simbol *Sequence Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Simbol** | **Nama** | **Keterangan** |
| 1 | atau   |  | | --- | | nama aktor | | Aktor atau  Tanpa Waktu Aktif | Orang, proses, atau sistem lain yang berinteraksi dengan sistem in*form*asi yang akan dibuat di luar sistem in*form*asi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gamabr orang, belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor. |
| 2 |  | Garis hidup atau *lifeline* | Menyatakan kehidupan suatu objek. |
| 3 |  | Objek | Menyatakan objek yang berinteraksi pesan. |

|  |  |  |  |
| --- | --- | --- | --- |
| 4 |  | Waktu Aktif | Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. |
| 5 | <<create>> | Pesan tipe *create* | Menyatakan suatu objek membuat objek lain, arah panah mengarah pada ojek yang dibuat. |
| 6 | 1:nama\_metode() | Pesan tipe *call* | Arah panah mengarah pada objek yang memiliki operasi atau metode, karena ini memanggil operasi atau metode maka operasi atau metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi. |
| 7 | 1:masukan | Pesan tipe *send* | Menyatakan bahwa suatu objek mengirimkan data atau masukan atau in*form*asi ke objek lainnya, arah panah mengarah pada objek yang dikirimi. |
| 8 | 1:keluaran | Pesan tipe *return* | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian. |
| 9 | <<destroy>> | Pesan tipe *destroy* | Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada |
|  |  |  | *destroy.* |

#### 2.10. *Ionic Framework*

*Framework Ionic* adalah sekumpulan teknologi yang dikembangkan untuk membangun aplikasi mobile *hybrid* yang *powerful*, cepat, mudah dan juga memiliki tampilan yang menarik. *Ionic* menggunakan *AngularJS* sebagai *framework* berbasis *web* dan menggunakan *Cordova* untuk membangun aplikasi *mobile*. *Ionic* *Framework* merupakan *framework* HTML5 yang masih baru, dirilis 2 tahun lalu. *Framework* ini membantu Anda dalam mengembangkan aplikasi *mobile* dengan teknologi *web* seperti HTML, CSS dan *Javascript*. *Ionic platform* menggunakan lisensi *opensource*, boleh digunakan oleh siapapun untuk membuat aplikasi *free* ataupun komersial dengan *Ionic*. *Ionic* memanfaatkan *AngularJS* untuk implementasi logikanya. Jika menggunakan *jQuery* terkenal lambat di *mobile* sedangkan *Angular* menawarkan per*form*a dan respon cepat serasa aplikasi *native* (Rofiq & Putri, 2017).

#### 2.11. *Atom Text Editor*

*Atom* adalah sebuah *text editor* yang memiliki lisensi *open source* yang tersedia untuk *macos, Linux,* dan *Microsoft Windows* dengan dukungan *plug-in* yang ditulis di Node.js, dan *embedded Git Control*, yang dikembangkan oleh GitHub. *Atom* adalah aplikasi desktop yang dibuat menggunakan teknologi web. Sebagian besar paket yang diperluas memiliki lisensi perangkat lunak gratis dan dibuat untuk komunitas dan dipelihara (Aryani et al., 2015).

#### 2.12. *Typescript*

*Typescript* adalah bahasa pemrograman berbasis *Javascript* yang menambahkan fitur konsep pemrograman OOP. Bahasa pemrograman ini menawarkan *class*, *module* dan *interface*, sehingga memudahkan *developer* dalam mengembangkan aplikasi yang kompleks menggunakan konsep OOP klasik (Abdulloh, 2018).

#### 2.13. JSON

JSON (*Java Script Object Notation*) adalah *form*at pertukaran data yang bersifat ringan, disusun oleh Douglas Crockford. Fokus JSON adalah pada representasi data di *website*. JSON dirancang untuk memudahkan pertukaran data pada situs dan merupakan perluasan dari fungsi-fungsi *javascript*. Contoh *form*at JSON :

[{"id": "0", "kata" : "Abasement", "makna" : "Perbandingan secara tidak menyenangkan, entah dibandingkan dengan orang lain atau dengan norma di sekitar."},

{"id": "1", "kata" : "Aborsi", "makna" : "Upaya penggagalan kehamilan dan menggugurkan janin yang dikandung."},

{"id": "2", "kata" : "Abnormal", "makna" : "(perilaku) Menyimpang dengan norma di masyarakat, tidak biasa, luar biasa."},

{"id": "3", "kata" : "Abstraksi", "makna" : proses memahami kesimpulan atau makna yang tidak terlihat dari suatu benda atau fenomena. Dilakukan untuk membentuk konsep pemikiran."},

{"id": "4", "kata" : "Ackerman, Nathan Ward", "makna" : Seorang psikolog dan terapis, penggagas bidang ilmu psikologi keluarga."},

{"id": "5", "kata" : "Adaptasi", "makna" : Perilaku menyesuaikan diri seseorang/suatu organisme dengan lingkungan."},

{"id": "6", "kata" : "ADHD", "makna" : "Gangguan berupa kurangnya perhatian dan perilaku hiperaktif, lebih besar dari yang sewajarnya pada masa tumbuh kembang."},

{"id": "7", "kata" : "Adler, Alfred", "makna" : "Dokter, psikolog, dan terapis, penggagas ilmu psikologi individualistik, teori urutan kelahiran, inferioritas, dan terapi berbasis konseling."},

{"id": "8", "kata" : "Adlerian", "makna" : "Pengikut aliran psikologi milik Alfred Adler."},

{"id": "9", "kata" : "Adopsi", "makna" : "Praktik yang dilakukan orang dewasa terhadap anak, yang mengambil hak asuh dan peran sebagai orang tua."},

{"id": "10", "kata" : "Afeksi", "makna" : "Istilah psikologi untuk ekspresi dan emosi yang dapat diamati. Afeksi termasuk gerak tubuh, nada bicara, ekspresi wajah, tertawa, dan menangis."}]

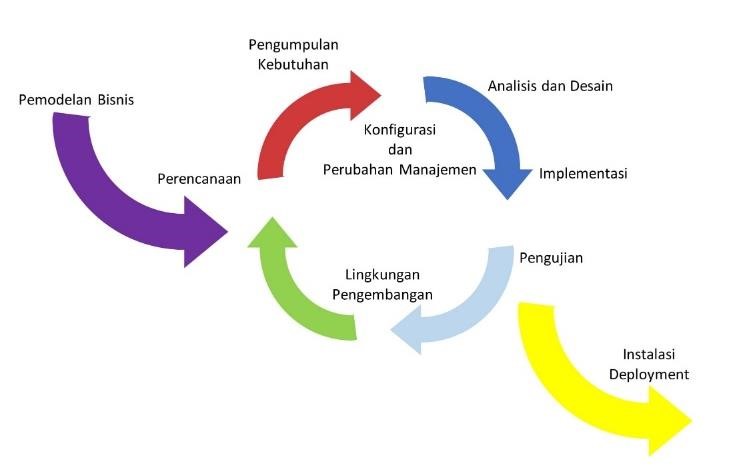
#### 2.14. Data Kamus

Data kamus yang digunakan diambil dari buku Kamus Psikologi yang ditulis oleh Arthur S. Reber & Emily S. Reber yang diterbitkan oleh Pustaka Pelajar serta dari website https://psikologihore.com/kamus-psikologi-bahasa-indonesia/.

Total kata berjumlah 632 kata.

#### 2.15. RUP

Menurut Siahaan (dalam Nofriansyah, 2015) bahwa, RUP (*Rational Unified Process*) adalah salah satu kerangka kerja untuk melakukan proses rekayasa kebutuhan. Tujuan utama standar RUP (*Rational Unified Process*) adalah untuk memastikan bahwa perangkat lunak yang sampai pada pengguna adalah perangkat lunak yang berkualitas baik. (Sukamto dan Shalahuddin, 2013), RUP *(Rational Unified Process*) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*). RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik (*well defined*) dan penstrukturan yang baik (*well structured*). RUP menyediakan pendefinisian struktur yang baik untuk alur hidup proyek perangkat lunak.



### Gambar 2.2. Proses Iteratif RUP (Rosa A. S dan M.Shalahuddin, 2013)

RUP (*Rational Unified Process*) memiliki empat buah tahap fase yang dapat dilakukan pula secara iteratif. Berikut ini penjelasan untuk setiap fase pada RUP (*Rational Unified Process*) :

1. *Inception* (permulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modeling*)dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*). Berikut adalah tahap yang dibutuhkan pada tahap ini :

* 1. Memahami ruang lingkup dari proyek termasuk pada biaya, waktu, kebutuhan, resiko dan lain sebagainya.
  2. Membangun kasus bisnis yang dibutuhkan.

1. *Elaboration* (perluasan atau perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem.

1. *Construction* (konstruksi)

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pasa kode program.

1. *Transition* (transisi)

Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh *user*. Aktifitas pada tahap ini termasuk pada pelatihan *user*, pemelliharaan dan pengujian sistem.

### BAB III

### METODOLOGI PENELITIAN

#### 3.1. Waktu dan Tempat Penelitian

##### 3.1.1. Waktu

Waktu pelaksanaan penelitian tugas akhir dilaksanakan mulai dari bulan Januari 2020 sampai dengan Desember 2020. Rincian kegiatan dapat dilihat pada Tabel 3.1 berikut:

# Tabel 3.1 *Gantt Chart* Waktu Penelitian

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Uraian | Waktu (2020) | | | | | | | | | | | | | | | | | | | | | | | |
| Januari | | | | Maret | | | | Mei | | | | Juli | | | | November | | | | Desember | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | *Inception* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | *Elaboration* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | *Construction* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | *Transition* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

#### 3.1.2. Tempat Penelitian

Adapun tempat penelitian tugas akhir ini adalah di Jurusan Psikologi, Fakultas Psikologi, Universitas Halu Oleo.

#### 3.2. Metode Pengumpulan Data

1. Kajian Pustaka

Kajian pustaka adalah metode pengumpulan data yang dilakukan dengan cara melakukan pengumpulan data yang diperoleh dari Kamus Psikologi yang ditulis oleh Arthur S. Reber & Emily S. Reber dan juga diperoleh dari *website* https://psikologihore.com/kamus-psikologi-bahasa-indonesia/.

41

42

1. Kuisioner

Kuesioner adalah salah satu metode pengumpulan data yang dilakukan dengan cara menyediakan daftar pertanyaan seputar penelitian yang akan dijawab oleh responden guna melakukan penilaian terhadap penelitian. Kuesioner ini digunakan untuk melakukan penilaian terhadap aplikasi.

#### 3.3. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam sistem ini adalah metode *Rational Unified Process*. Dalam metode ini, terdapat lima tahap pengembangan perangkat lunak, yaitu :

1. *Inception*

Tahap ini berfokus pada pemodelan proses bisnis yang dibutuhkan (*business modelling*) dan mendapatkan kebutuhan sistem yang akan dibuat (*requirements*). *Business modelling* dilakukan dengan wawancara penelitian untuk mendapatkan model bisnis. Dari model bisnis tersebut, kita lakukan analisa sehingga didapatkan kebutuhan sistem (*requirments*).

1. *Elaboration*

Tahap ini berfokus pada perancangan arsitektur sistem (*design*). Sistem dirancang berdasarkan hasil pada tahap *inception* (proses bisnis dan kebutuhan sistem). Data yang didapat akan dianalisa terlebih dahulu kemudian dilanjutkan dengan proses desain sistem. Perancangan sistem dibuat melalui diagram UML (*Unfied Modeling Lenguage*) yang meliputi *flowchart, use case, activity, class* dan *sequence.*  Hal ini bertujuan untuk memilah antara kebutuhan yang akan dibuat dan yang tidak.

1. *Construction*

Fase konstruksi terfokus pada pengembangan perangkat lunak baik komponen utama maupun fitur-fitur pendukung dengan melakukan sederet iterasi. Di setiap iterasi terdapat proses analisa, desain, implementasi, dan pengujian. Hasil yang diharapkan dari fase ini adalah sebuah produk perangkat lunak yang siap digunakan oleh *end-user*.

1. *Transition*

Tahap ini berfokus pada tahap *deployment* atau instalasi. Tahap ini menghasilkan produk yang siap digunakan oleh pengguna. Pengguna yang dimaksud adalah mahasiswa psikologi yang kesulitan mencari arti dari suatu istilah psikologi.

**3.4.** **Analisis Kebutuhan**

##### 3.4.1. Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang menyangkut sistem yang dibuat melalui perancangan. Perancangan sistem terdiri dari perancangan alur sistem, perancangan antarmuka, perancangan aktivitas sistem, dan perancangan *database* sistem. Perancangan tersebut dibuat melalui bahasa pemodelan UML(*Unified Modeling Lenguage*).

##### 3.4.2. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah kebutuhan yang diperlukan seorang *development* sistem dalam membangun sistem yang akan dibuat. Kebutuhan ini terdiri dari kebutuhan perangkat keras dan perangkat lunak yang akan digunakan, sehingga dapat ditentukan kompabilitas sistem yang dibangun terhadap sumber daya yang ada.

##### 3.4.2.1.Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan untuk membangun aplikasi kamus psikologi, sebagai berikut :

# Tabel 4.1 Spesifikasi Perangkat Keras

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Perangkat** | **Spesifikasi** |
| 1. | *PC* | *Lenovo* |
| 2. | *Processor* | *Intel Core i7-7700 3.60Ghz* |
| 3. | *Monitor* | *14 inci* |
| 4. | *Memori* | *RAM 8 GB* |
| 5. | *Smartphone* | *Xiaomi Redmi 5 Ram 4 GB* |

#### 3.4.2.2.Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk membangun aplikasi kamus psikologi serta membantu jalannya perangkat keras pada sistem, sebagai berikut :

44

# Tabel 4.2 Spesifikasi Perangkat Lunak

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Perangkat** |  | **Spesifikasi** |
| 1. | *Operating System* | *Windows 10* |  |
| 2. | *Text Editor* | *Atom* |  |
| 3. | *Framework* | *Ionic* |  |
| 4. | *Android OS* | *Nougat* |  |

#### 3.5 Analisis Perancangan Sistem

Perancangan sistem yang akan dibangun terdiri atas perancangan *Flowchart* dan perancangan UML serta perancangan *user interface.*

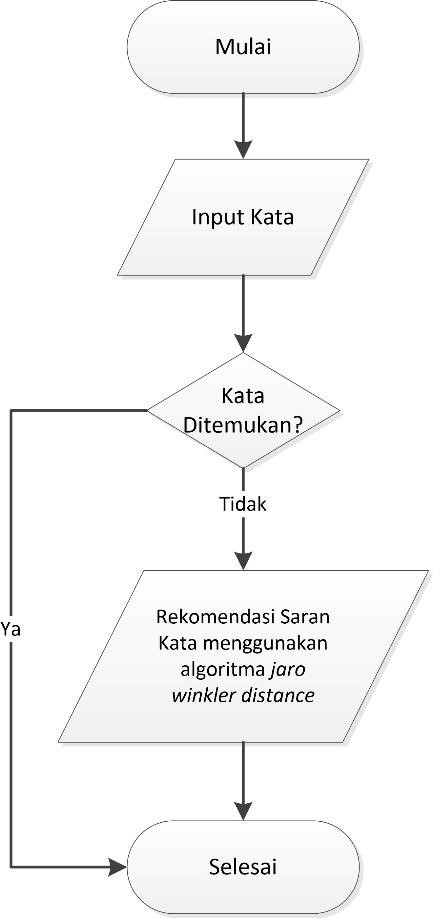
##### 3.5.1. Perancangan *Flowchart*

*Flowchart* adalah simbol-simbol pekerjaan yang menunjukkan bagan aliran proses yang saling terhubung. Perancangan *flowchart* yang akan dibangun terdiri atas *flowchart* sistem, *flowchart* metode *Jaro Winkler Distance*.

##### 3.5.1.1.*Flowchart* Sistem

Setelah menganalisis sistem, maka didapatkan *flowchart diagram* sistem pada aplikasi kamus psikologi ini yang dapat dilihat pada Gambar 4.1. Adapun alur kerja *flowchart diagram* sistem adalah sebagai berikut : 1) *User* memasukkan kata yang ingin dicari misalnya delirium.

1. Jika kata yang di*input* cocok dengan kata yang ada di *database*, maka sistem akan menampilkan keterangan dari kata tersebut.
2. Jika kata yang di*input* tidak cocok misalnya *user* meng*input*kan kata dellirium, maka sistem akan memberikan rekomendasi kata yang mendekati dengan kata *input*an yaitu katanya adalah delirium.
3. Kemudian *user* memilih beberapa saran kata yang direkomendasikan oleh sistem.

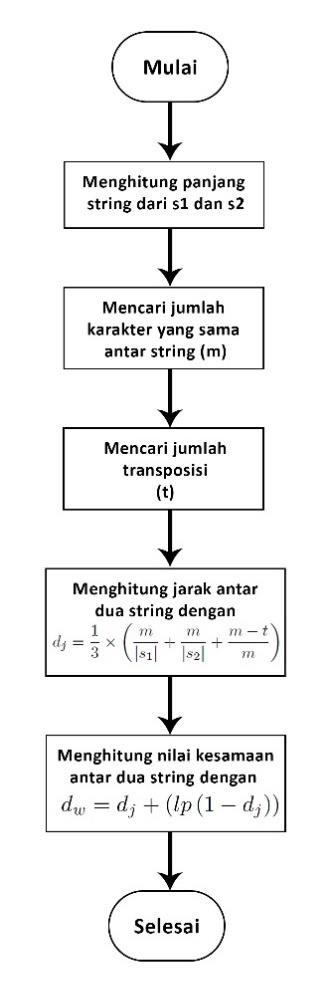


### Gambar 4.1 *Flowchart* Sistem

#### 3.5.1.2.*Flowchart* Algoritma *Jaro Winkler Distance*

Gambar 4.2. menunjukkan alur kerja *flowchart* diagram metode *Jaro Winkler Distance* adalah sebagai berikut:

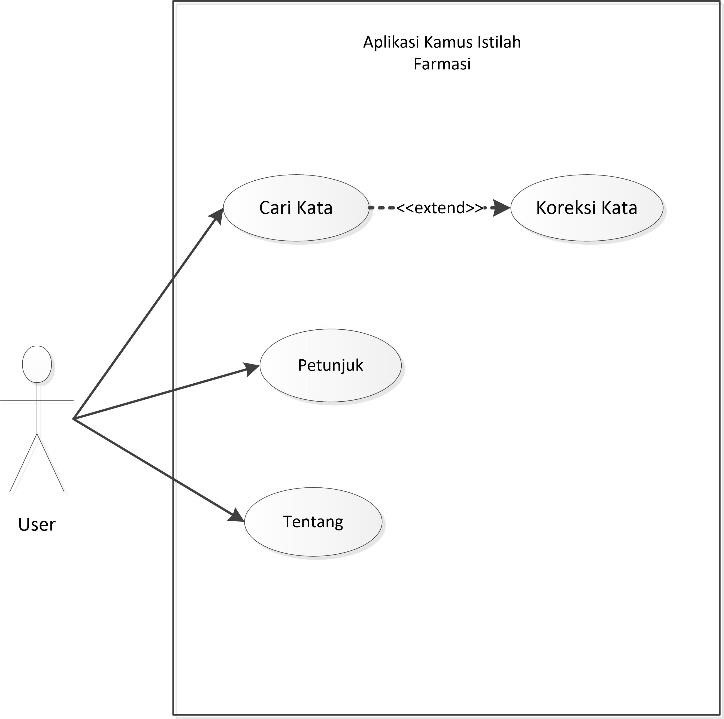
1. Pertama *user* memasukkan kata kunci yang akan dicari.
2. Sistem melakukan proses pencocokkan karakter yang di*input*kan dengan karakter yang ada di penyimpanan kata.
3. Sistem melakukan proses perhitungan panjang *string* 1 (s1) dan *string* 2 (s2).
4. Lalu sistem kemudian mencari jumlah karakter yang sama antar dua *string* tersebut.
5. Kemudian sistem mencari jumlah transposisi atau jumlah karakter yang tertukar (t).
6. Setelah itu, sistem menghitung jarak antara dua *string* (dj).
7. Lalu, sistem akan menghitung nilai kesamaan antara *string* 1 (s1) dan *string* 2 (s2).



**Gambar 4.2 *Flowchart* Algoritma *Jaro Winkler Distance***

#### 3.5.2. Analisis Perancangan *Unified Modeling Language* (UML) 4.2.2.1.*Use Case Diagram*

*Use case diagram* adalah sebuah diagram yang dapat merepresentasikan interaksi yang terjadi antara *user* dengan sistem. *Use case diagram* ini mendeskripsikan siapa saja yang menggunakan sistem dan bagaimana cara mereka berinteraksi dengan sistem. *Use case diagram* dari sistem yang akan dibangun dapat ditunjukkan pada Gambar 4.3.

Aplikasi Kamus Istilah Psikologi

**Gambar 4.3. *Use Case Diagram* Kamus Psikologi**

# Tabel 4.3. Keterangan *Use Case Diagram*

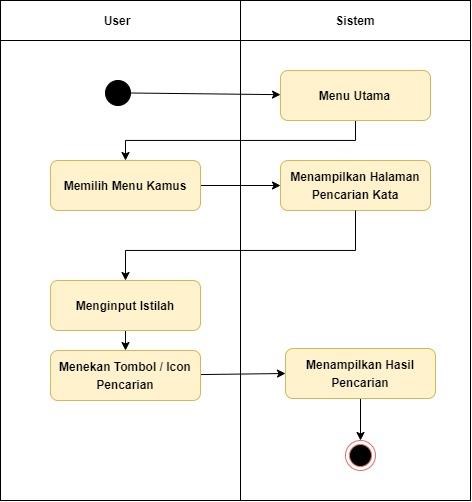
|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| *User* memilih tombol pencarian | Sistem akan menampilkan menyiapkan *form* pencarian. |
| *User* memilih menu petunjuk | Sistem akan menampilkan menu petunjuk bantuan penggunaan aplikasi. |
| *User* memilih menu tentang | Sistem akan menampilkan menu tentang aplikasi. |
| *User* memasukkan kata salah | Sistem akan merekomendasikan beberapa kata saran |

#### 3.5.2.2.*Activity Diagram*

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* *diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Berikut ini adalah *activity diagram* yang akan menggambarkan alir aktivitas sistem.

### 1. *Activity Diagram* Pencarian Kata

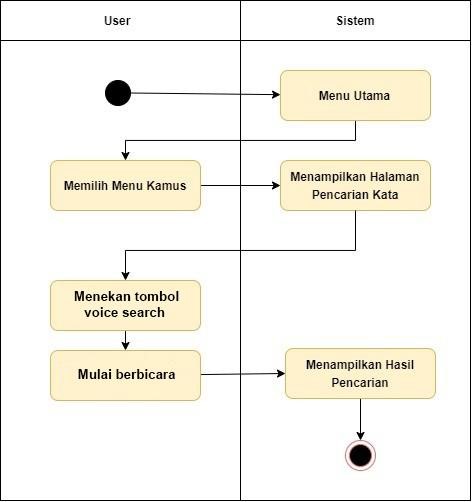
Gambar 4.4 menampilkan aktivitas *user* ketika mencari kata, dan kata yang di*input*kan merupakan kata yang benar. Setelah *user* masuk pada halaman menu utama, *user* memilih menu kamus, lalu sistem akan menampilkan daftar istilah dan *form* pencarian. Dari *form* pencarian, *user* dapat meng*input*kan istilah yang ingin dicari dan menekan ikon *search.* Jika kata yang di*input*kan benar maka sistem akan menampilkan kata, dan detail kata.



**Gambar 4.4. *Activity Diagram* Pencarian Kata**

### *2. Activity Diagram* *Voice Search*

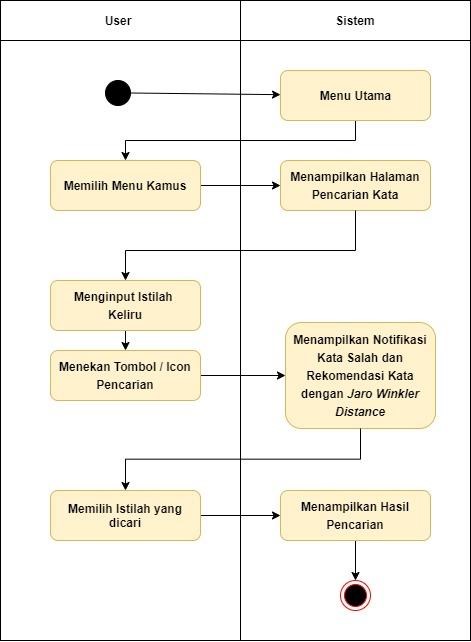
Gambar 4.5 menampilkan aktivitas *user* ketika meng*input*kan kata menggunakan suara. Setelah *user* masuk ke *form* pencarian kata, *user* menekan tombol berbentuk *mic* lalu mulai mengucapkan kata yang ingin dicari. Maka sistem akan menampilkan hasil pencarian.



**Gambar 4.5. *Activity Diagram Voice Search***

### 3. *Activity Diagram* Koreksi Kata

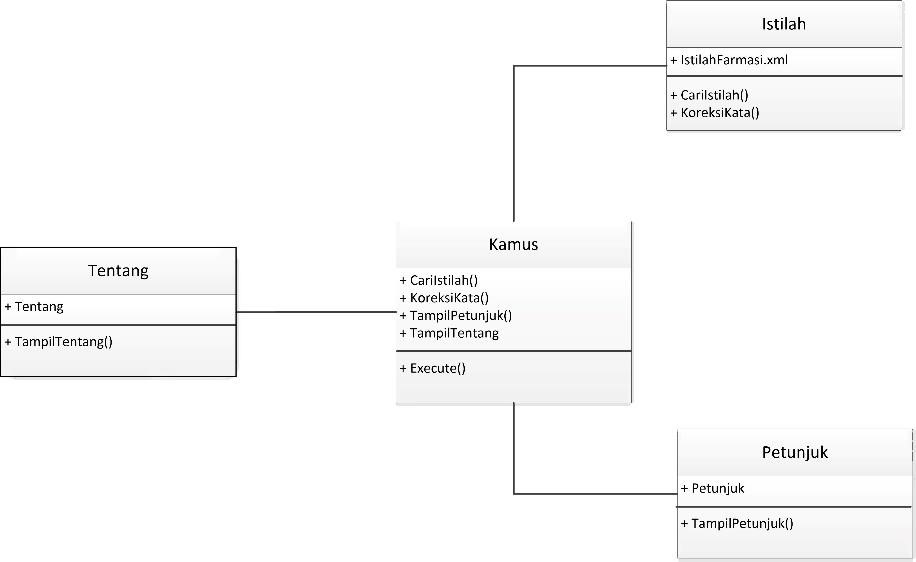
Gambar 4.5 menampilkan aktivitas *user* ketika meng*input*kan kata yang salah. Setelah *user* masuk ke *form* pencarian kata, *user* meng*input*kan kata dengan ejaan yang salah. Maka sistem akan menampilkan rekomendasi kata benar. Untuk melihat detail istilah *user* harus memilih kata terlebih dahulu.



### Gambar 4.6. *Activity Diagram* Koreksi Kata

#### 3.5.2.3. *Class Diagram*

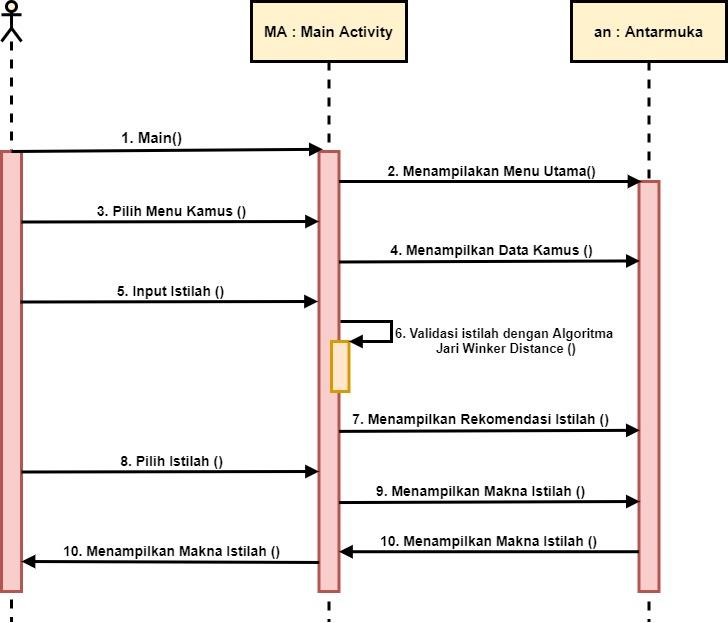
*Class diagram* merupakan diagram yang selalu ada dipemodelan sistem berorientasi objek. *Class diagram* menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. Berikut ini adalah *class diagram* sistem.



### Gambar 4.7. *Class Diagram* Kamus Psikologi

#### 3.5.2.4. *Sequence Diagram* Kamus

*Sequence Diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem yang digambarkan terhadap waktu. Berikut ini adalah *sequence diagram* yang akan menggambarkan interkasi antar objek dan sistem.



**Gambar 4.8. *Sequence Diagram* Kamus**

Gambar 4.7. adalah *diagram* *sequence* yang menjelaskan bagaimana *user* mencari istilah. Ketika *user* memasuki halaman utama, sistem menampilkan *form* pencarian. Kemudian *user* akan meng*input*kan istilah yang dicari. Lalu sistem akan mencocokkan istilah *input*an dengan istilah yang ada di gudang kata. Jika kata cocok, sistem akan menampilkan keterangan dari istilah yang dicari. Namun, jika kata keliru, maka sistem akan memberikan rekomendasi kata yang mirip dengan kata yang dicari.

#### 3.5.2.5. Metode Pengujian

Pengujian merupakan metode yang dilakukan untuk menjelaskan mengenai pengoperasian perangkat lunak yang terdiri dari perangkat pengujian, metode pengujian dan pelaksanaan pengujian.

Pengujian program ini menggunakan metode *black box* . Pengujian *black box* merupakan pengujian program berdasarkan fungsi dari program. Tujuan dari metode *black box* ini adalah untuk menemukan kesalahan fungsi pada program. Pengujian dengan metode ini dilakukan dengan cara memberikan sejumlah *input* pada program aplikasi yang kemudian diproses sesuai dengan kebutuhan fungsionalnya untuk melihat apakah program aplikasi menghasilkan keluaran yang diinginkan dan sesuai dengan fungsi dari program tersebut. Apabila dari masukan yang diberikan proses menghasilkan keluaran yang sesuai dengan kebutuhan fungsionalnya, maka program aplikasi yang bersangkutan telah benar, tetapi jika keluaran yang dihasilkan tidak sesuai dengan kebutuhan fungsionalnya, maka masih terdapat kesalahan pada program aplikasi.

Pengujian dilakukan dengan mencoba semua kemungkinan yang terjadi dan pengujian dilakukan berulang-ulang. Jika dalam pengujian ditemukan kesalahan, maka akan dilakukan penelusuran dan perbaikan untuk memperbaiki kesalahan yang terjadi. Jika telah selesai melakukan perbaikan, maka akan dilakukan pengujian kembali. Pengujian dan perbaikan dilakukan secara terus menerus hingga diperoleh hasil yang terbaik.

# Tabel 4.4. Rancangan Pengujian Aplikasi

|  |  |  |
| --- | --- | --- |
| **Kelas Uji** | **Detail Uji** | **Jenis Pengujian** |
| Pengujian Menu | Home | *Black Box* |
| Petunjuk | *Black Box* |
| Tentang | *Black Box* |
| Pengujian Pencarian | Pencarian Kata Benar | *Black Box* |
| Pengujian *Voice Search* | Pencarian dengan suara | *Black Box* |
| Pengujian Metode *Jaro*  *Winkler Distance* | Pencarian Kata Keliru | *Black Box* |

**DAFTAR PUSTAKA**

Abdulloh, R. (2018). *Mudah Membuat Aplikasi Android dengan Ionic 3*.

Aryani, D., Setiadi, A., & Alfiah, F. (2015). Aplikasi Web Pengiriman Dan

Penerimaan Sms Dengan Gammu Sms Engine Berbasis Php. *CCIT Journal*, *8*(3), 174–190. https://doi.org/10.33050/ccit.v8i3.340

Fahma, A. I. (2018). Identifikasi Kesalahan Penulisan Kata ( Typographical Error ) pada Dokumen Berbahasa Indonesia Menggunakan Metode N-gram dan Levenshtein Distance. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *2*(1), 53–62.

Jogiyanto HM. (2005). *Analisis dan Desain Sistem Informasi : Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. 15–48.

Kurniasih, ika. (2014). *Analisis lema Kamus Bahasa Indonesia untuk Pelajar*. 147. http://eprints.uny.ac.id/17291/1/Ika Kurniasih 10210141004.pdf

Nofriansyah, M. (2015). APLIKASI MONITORING AKTIVITAS LABORATORIUM KOMPUTER MAHASISWA JURUSAN

MANAJEMEN IN*FORM*ATIKA POLITEKNIK NEGERI SRIWIJAYA.

*Journal of Chemical Information and Modeling*, *53*(9), 1689–1699. https://doi.org/10.1017/CBO9781107415324.004

Nugroho, K. (2019). Implementasi Sistem Speech To Text Berbasis Android Menggunakan APP Inventor Speech Recognizer. *Infokam*, *1*, 38–43.

Prasetyo, A., Baihaqi, W. M., & 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), 435. https://doi.org/10.25126/jtiik.201854780

Renditia, Y. (2005). APLIKASI KAMUS DWIBAHASA INDONESIA - INGGRIS BERBASIS ANDROID. *Choice Reviews Online*, *43*(02), 43-0657-

43–0657. https://doi.org/10.5860/choice.43-0657

Rochmawati, Y., & Kusumaningrum, R. (2016). Studi Perbandingan Algoritma Pencarian String dalam Metode Approximate String Matching untuk

Identifikasi Kesalahan Pengetikan Teks. *Jurnal Buana Informatika*, *7*(2), 125–

134. https://doi.org/10.24002/jbi.v7i2.491

Rofiq, M., & Putri, S. I. (2017). Perancangan Sistem Pemesanan Rumah Sakit di

Kota Malang Menggunakan Ionic Framework berbasis Mobile Phone. *Jurnal*

*Ilmiah Teknologi Informasi Asia*, *11*(2), 171.

https://doi.org/10.32815/jitika.v11i2.210

Sujaini, H., & Nyoto, R. D. (2016). *Peter Norvig dan Spelling Checker BK-Trees pada Kata Berbahasa Indonesia*. *5*(1), 1–5.

Suka Parwita, W. G., & Winarko, E. (2015). Hybrid Recommendation System Memanfaatkan Penggalian Frequent Itemset dan Perbandingan Keyword.

*IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *9*(2),

167. https://doi.org/10.22146/ijccs.7545

Sunarti. (2016). Rancang Bangun Sistem Peminjaman Pada Koperasi Hortina Direktorat Jenderal Hortikultura. *Journal on Computer and Information Technology*, *1*(1), 42–50.

Yulianingsih. (2017). IMPLEMENTASI ALGORITMA JARO-WINKLER DAN LEVENSTEIN DISTANCE DALAM PENCARIAN DATA PADA

DATABASE. *Jurnal String Vol. 2 No. 1 Agustus 2017*, *53*(9), 1689–1699. https://doi.org/10.1017/CBO9781107415324.004

Zahara, F. A. (2014). IMPLEMENTASI E-LIBRARY PADA JURUSAN MANAJEMEN IN*FORM*ATIKA POLITEKNIK NEGERI SRIWIJAYA.

*Journal of Chemical Information and Modeling*, *53*(9), 1689–1699.

https://doi.org/10.1017/CBO9781107415324.004