Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

Team project ©2017
Dony Pratidana S. Hum | Bima Agus Setyawan S. IIP

BAB II
LANDASAN TEORI

2.1 Kamus

KBBI (Kamus Besar Bahasa Indonesia) menyebutkan bahwa kamus adalah buku acuan yang memuat kata dan ungkapan, biasanya disusun menurut abjad berikut keterangan tentang makna, pemakaian atau terjemahannya. Kamus istilah adalah kamus yang memuat istilah dengan makna konsepnya berasal dari suatu bidang ilmu (Harfianti, 2010).

Menurut Dr. Imel Ya'qub (dalam Hakim, 2013), kamus dapat dibedakan menjadi 8 macam, antara lain:

1. Kamus bahasa

   Merupakan kamus yang secara khusus membahas lafal atau kata-kata dari sebuah bahasa dan dilengkapi dengan pemakaian kata-kata tersebut. Kamus bahasa hanya membuat satu bahasa, sehingga biasanya pemaknaan kata hanya menyebut sinonim ataupun definisi kata tersebut.

2. Kamus terjemah

   Kamus terjemah atau disebut juga kamus campuran atau bilingual memadukan dua bahasa untuk menentukan titik temu makna dari kosakata. Kamus terjemah memuat kata-kata asing yang kemudian dijelaskan satu persatu dengan mencari makna yang sama dan disesuaikan dengan Bahasa nasional atau bahasa pemakai kamus.

UNIVERSITAS MULTIMEDIA NUSANTARA

Rancang bangun aplikasi..., Wiwi Clarissa, FTI UMN, 2019

3. Kamus tematik

Merupakan kamus yang kata-katanya terhimpun di dalam kamus yang disusun secara tematik berdasarkan topik-topik tertentu yang mempunyai makna sebidang.

4. Kamus derivatif

Disebut juga dengan istilah kamus etimologis, yaitu sebuah kamus yang membahas asal usul sebuah kata, sehingga kamus derivatif berfungsi untuk menginformasikan asal usul kosakata.

5. Kamus evolutif

Merupakan kamus yang lebih memprioritaskan sejarah perkembangan makna dari sebuah kata, bukan lafalnya. Kamus evolutif memberikan informasi tentang perluasan makna, perubahannya, sebab-sebab perubahan makna dan sebagainya.

6. Kamus spesialis

Merupakan kamus yang menghimpun kata-kata yang ada dalam satu bidang atau disiplin ilmu tertentu. Contohnya yaitu kamus kedokteran, kamus pertanian, kamus musik dan sebagainya.

7. Kamus informatif

Merupakan kamus yang mencakup segala hal termasuk sejarah pengguna bahasa, tokoh, dan sebagainya, atau disebut juga dengan ensiklopedia. Di mana menjelaskan sebuah kata yang tidak hanya sekedar membahas makna dan derivasi dari sebuah kata, tapi juga mencakup segala informasi lain diluar makna leksikon, seperti sejarah, biografi, peta, dan lain sebagainya.

UNIVERSITAS MULTIMEDIA NUSANTARA

Rancang bangun aplikasi..., Wiwi Clarissa, FTI UMN, 2019

8. Kamus visual

Merupakan kamus yang menjelaskan makna kata yang lebih menonjolkan gambar dari kata yang dimaksud daripada sebuah istilah yang definitif.

2.2 Kesalahan Penulisan Kata (Typographical Error)

Kesalahan penulisan kata (*typographical error*) adalah kesalahan yang dibuat pada saat proses mengetik teks. Istilah ini mencakup kesalahan karena kegagalan mekanis atau slip tangan atau jari, dan kesalahan yang timbul akibat ketidaktahuan penulis, seperti kesalahan ejaan. Kesalahan tipografi dapat disebabkan oleh jari yang menekan dua tombol *keyboard* yang berdekatan secara bersamaan. Kesalahan dalam penulisan kata dapat mengubah arti kata atau bahkan arti dari kalimat.

Kesalahan penulisan kata dapat dikategorikan menjadi dua jenis yaitu *non-word error* dan *real-word error*. *Non-word error* adalah kesalahan penulisan kata dimana kata yang tertulis tidak terdapat di dalam kamus atau tidak memiliki arti tertentu. *Real-word error* adalah kesalahan penulisan kata dimana kata tersebut terdapat di dalam kamus, namun kata tersebut bukanlah kata yang dimaksud dalam kalimat (Naradhipa, et al., 2011).

2.3 Damerau-Levenshtein Distance

Algoritma *Damerau-Levenshtein Distance* dikembangkan oleh Frederick J. Damerau. Algoritma *Damerau-Levenshtein Distance* merupakan algoritma pengembangan dari algoritma *Levenshtein Distance*. *Damerau-Levenshtein Distance* adalah suatu pengukuran (metrik) yang dihasilkan melalui perhitungan jumlah perbedaan yang terdapat pada dua *string*.

Rancang bangun aplikasi..., Wiwi Clarissa, FTI UMN, 2019

Algoritma *Damerau-Levenshtein Distance* menentukan jumlah minimum operasi yang dibutuhkan untuk mengubah satu *string* menjadi *string* lain, dimana operasi yang digunakan sama dengan *Levenshtein Distance* yaitu *insertion*, *deletion*, dan *substitution* namun dengan penambahan operasi *transposition* antara dua karakter (Jupin, Shi, & Obradovic, 2012). Damerau menyatakan bahwa sekitar 80% dari kesalahan penulisan adalah hasil dari keempat operasi tersebut. Contoh penggunaan operasi edit pada algoritma Damerau-Levenshtein Distance adalah sebagai berikut.

1. Operasi *Insertion*

   Operasi *insertion* merupakan operasi penyisipan dengan menyisipkan karakter pada indeks tertentu untuk menyamakan *string* sumber dan *string* target. Misalnya ada dua variabel, yaitu S = "sittin" dan T = "sitting". Proses *insertion* dilakukan dengan penambahan karakter 'g' pada posisi 7.

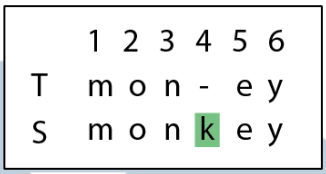
	1	2	3	4	5	6	7
T	s	i	t	t	i	n	g
S	s	i	t	t	i	n	-
							g

Gambar 2.1 Proses Insertion

2. Operasi *Deletion*

   Operasi *deletion* merupakan operasi penghapusan dengan menghapus karakter pada indeks tertentu untuk menyamakan *string* sumber dan *string* target. Misalnya ada dua variabel, yaitu S = "monkey" dan T = "money". Proses *deletion* dilakukan dengan menghapus karakter 'k' pada posisi 4.

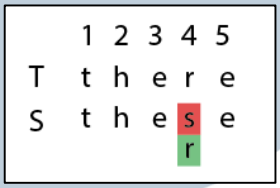
Rancang bangun aplikasi..., Wiwi Clarissa, FTI UMN, 2019



Gambar 2.2 Proses Deletion

3. Operasi *Substitution*

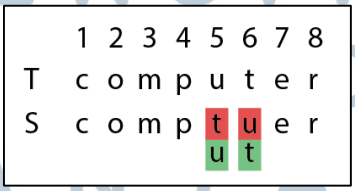
Operasi *substitution* merupakan operasi penggantian dengan mengganti karakter pada indeks tertentu untuk menyamakan *string* sumber dan *string* target. Misalnya ada dua variabel, yaitu S = “these” dan T = “there”. Proses *substitution* dilakukan dengan mengganti karakter ‘s’ pada posisi 4 menjadi karakter ‘r’.



Gambar 2.3 Proses Substitution

4. Operasi *Transposition*

Operasi *transposition* merupakan operasi penukaran dengan menukar karakter berurutan pada indeks tertentu untuk menyamakan *string* sumber dan *string* target. Misalnya ada dua variabel, yaitu S = “computer” dan T = “comptuer”. Pada posisi 5 dan 6, dilakukan penukaran karakter berurutan ‘t’ dan ‘u’ pada *string* sumber untuk menyamakan *string* tersebut terhadap *string* target.



Gambar 2.4 Proses Transposition

Rancang bangun aplikasi..., Wiwi Clarissa, FTI UMN, 2019

Langkah-langkah untuk algoritma *Damerau-Levenshtein Distance* adalah sebagai berikut (Pahdi, 2016).

1. Inisialisasi m sebagai panjang karakter dari s dan n sebagai panjang karakter dari t.
2. Buat sebuah matriks d dengan $m + 1$ baris dan $n + 1$ kolom
3. Isi baris pertama dengan $0..n$ dan isi kolom pertama dengan $0..m$
4. Periksa setiap karakter dari s terhadap t.

   Jika $s[i] = t[j]$ maka $cost = 0$.

   Jika $s[i] \neq t[j]$ maka $cost = 1$.
5. Isi nilai dari setiap sel $d[i, j]$ baris per baris dengan rumus:

$$d[i, j] = \min(x, y, z) \quad \ldots (2.1)$$

Keterangan:

- $d[i, j]$: sel yang merupakan pertemuan kolom j dengan baris i pada matriks d.
- x: adalah nilai yang terdapat di sel atas dari posisi sel sekarang ditambah dengan 1 (satu) atau dapat dirumuskan:

$$x = d[i - 1, j] + 1 \quad \ldots (2.2)$$

- y: nilai yang terdapat di sel sebelah kiri dari posisi sel sekarang ditambah 1 (satu) atau dapat dirumuskan:

$$y = d[i, j - 1] + 1 \quad \ldots (2.3)$$

- z: nilai yang terdapat di sel sebelah atas kiri dari posisi sel sekarang (arah barat laut) ditambah nilai *cost* dan dapat dirumuskan:

$$z = d[i - 1, j - 1] + cost \quad \ldots (2.4)$$

Rancang bangun aplikasi..., Wiwi Clarissa, FTI UMN, 2019

Jika $i > 1$ dan $j > 1$ dan $s[i] = t[j - 1]$ dan $s[i - 1] = t[j]$ yang mana artinya setelah kedua kata dibandingkan terdapat karakter yang dapat ditransposisikan, maka isi nilai sel d[i, j] dengan rumusan berikut:

$$d[i, j] = \min(d[i, j], d[i - 2, j - 2] + cost) \quad \ldots (2.5)$$

6. Setelah langkah iterasi di atas selesai, maka jarak *edit* akan ditemukan pada sel $d[n, m]$ yaitu sel pada pojok kanan baris terakhir.

Pseudocode dari algoritma *Damerau-Levenshtein Distance* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Pseudocode Algoritma Damerau-Levenshtein Distance (Setiadi, 2013)

Algoritma Damerau-Levenshtein Distance
function damerauLevenshteinDistance(input s : array[1..m] of char, input t : array[1..n] of char) → integer {function to compute Damerau-Levenshtein distance between two strings using Damerau-Levenshtein algorithm}
DECLARATION i, j : integer cost : integer d : array [0..m][0..n] of integer

ALGORITHM
for i ← 1 to m do { source prefixes initialization }
  d[i][0] ← i
endfor
for j ← 1 to n do { target prefixes initialization }
  d[0][j] ← j
endfor
{ using Damerau-Levenshtein Algorithm to check }
for i ← 1 to n do
  for j ← 1 to m do
  if (s[i] == t[j]) then
  cost ← 0
  else
  cost ← 1
  endif

Tabel 2.1 Pseudocode Algoritma Damerau-Levenshtein Distance (Lanjutan)

Algoritma Damerau-Levenshtein Distance

d[i][j] ← minimum (
d[i-1][j] + 1, { deletion }
d[i][j-1] + 1, { insertion }
d[i-1][j-1] + cost { substitution }
)
if (i > 1 and j > 1 and s[i] == t[j-1] and s[j-1] == t[i])
then
d[i][j] ← minimum (
d[i][j],
d[i-2][j-2] + cost { transposition }
)
endif
endfor
endfor
→ d[m][n] { return results }

2.4 Filter and Verify Method

Pada tahun 90-an, metode "*filter and verify*" diperkenalkan untuk mengurangi perbandingan data dalam perhitungan *edit distance*. Penelitian tentang metode ini masih sangat aktif. *Filter* dapat membuat sistem menjadi lebih efisien dengan membuang perbandingan yang tidak diperlukan. Salah satu metode yang paling umum adalah *length filtering* (Gravano dalam Jupin, Shi, & Obradovic, 2015), dimana selisih dari panjang dua *string s* dan *t* tidak boleh lebih besar dari *k*. Algoritma dari *length filtering* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Algoritma Length Filtering

Algoritma Length Filtering
Algorithm: LengthFilter(*s*, *t*) **Input**: *s*, *t*: strings of characters **Output**: Boolean **Begin** **if** $abs\big((\lvert s\rvert - \lvert t\rvert)\big) > k$: **return** FALSE **else: return** TRUE **end-if** **end**

Rancang bangun aplikasi..., Wiwi Clarissa, FTI UMN, 2019

Tabel 2.3 merupakan Pseudocode Filtering Saran Kata dimana kata yang telah dihitung nilai *edit distance*-nya akan dimasukkan ke daftar saran kata apabila nilai *edit distance* tidak melebihi 50% dari panjang kata kunci yang dimasukkan oleh *user*.

Tabel 2.3 Pseudocode Filtering Saran Kata

Pseudocode Filtering Saran Kata
function searchSuggestionFilter(input word: JSONObject, keyword: string)
DECLARATION searchSuggestion : array of JSONObject

ALGORITHM
if (word.editDistance <= keyword.length*0.5)
  searchSuggestion.push(word)
endif

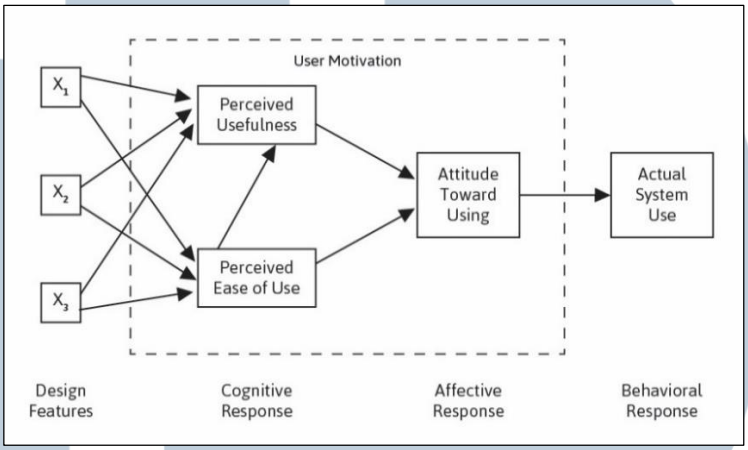
2.5 Technology Acceptance Model (TAM)

Technology Acceptance Model (TAM) diperkenalkan oleh Fred D. Davis pada tahun 1989 sebagai instrumen untuk memprediksi kemungkinan teknologi baru diadopsi dalam suatu kelompok (Tang & Chen, 2011).

Technology Acceptance Model dapat diilustrasikan pada Gambar 1. Menurut model ini, sikap pengguna terhadap penggunaan sistem yang diberikan dianggap menjadi penentu utama apakah ia benar-benar menggunakannya atau tidak. Sikap terhadap penggunaan dipengaruhi dari dua variabel, yaitu *perceived usefulness* dan *perceived ease of use*. *Perceived usefulness* adalah sejauh mana seorang individu percaya bahwa menggunakan sistem tertentu akan meningkatkan kinerja pekerjaannya. *Perceived ease of use* adalah sejauh mana seorang individu percaya bahwa menggunakan sistem tertentu akan bebas dari upaya fisik dan mental. *Perceived usefulness* (manfaat yang dirasakan) juga dipengaruhi oleh *perceived ease of use* (kemudahan penggunaan), karena semakin mudah suatu

Rancang bangun aplikasi..., Wiwi Clarissa, FTI UMN, 2019

sistem digunakan maka sistem tersebut akan semakin bermanfaat. Fitur desain secara langsung memengaruhi *perceived usefulness* dan *perceived ease of use* (Davis, 1985).



Gambar 2.5 Technology Acceptance Model (Davis, 1985)

Awalnya Davis menggunakan sebanyak 14 indikator (*initial scale items*) dalam mengukur *perceived usefulness* dan *perceived ease of use*. Namun setelah beberapa kali melakukan uji coba, diperoleh hasil berupa 6 indikator pengukuran yang lebih baik dan lebih praktis. Tabel 2.4 adalah indikator pengukuran untuk variabel *perceived usefulness* dan *perceived ease of use*.

Tabel 2.4 Indikator Perceived Usefulness dan Perceived Ease of Use (Davis, 1989)

Scale Items
Usefulness 1. Work More Quickly 2. Job Performance 3. Increase Productivity 4. Effectiveness 5. Makes Job Easier 6. Useful
Ease of Use 1. Easy to Learn 2. Controllable 3. Clear & Understandable 4. Flexible 5. Easy to Become Skillful 6. Easy to Use

Rancang bangun aplikasi..., Wiwi Clarissa, FTI UMN, 2019

2.6 Skala Likert

Uji coba dalam penelitian ini menggunakan kuesioner. Kuesioner adalah sejumlah pertanyaan tertulis yang digunakan untuk memperoleh informasi dari responden dalam arti laporan tentang pribadinya atau hal-hal yang ia ketahui (Arikunto, 2013). Kuesioner ini menggunakan skala Likert sebagai pengukuran variabelnya. Skala Likert digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Dengan skala Likert, maka variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikan sebagai titik tolak untuk menyusun *item-item* instrument yang dapat berupa pertanyaan atau pernyataan (Sugiyono, 2012). Skor dari skala Likert dapat dilihat pada Tabel 2.5.

Tabel 2.5 Skor Skala Likert

Skor	Jawaban
1	Sangat Tidak Setuju
2	Tidak Setuju
3	Netral
4	Setuju
5	Sangat Setuju

Persentase skor pada pertanyaan kuesioner dapat dihitung dengan menggunakan Rumus 2.6.

$$\text{Persentase Skor} = \frac{(SS*5)+(S*4)+(N*3)+(TS*2)+(STS*1)}{(5*Jumlah\ Responden)} * 100\% \quad \ldots (2.6)$$

Selanjutnya persentase skor yang telah diperoleh kemudian dimasukkan ke dalam skala persentase pada Tabel 2.6.

UNIVERSITAS MULTIMEDIA NUSANTARA

Tabel 2.6 Skala Persentase Skor

Persentase Skor	**Jawaban**
0% – 20%	Sangat Tidak Setuju
21% – 40%	Tidak Setuju
41% – 60%	Netral
61% – 80%	Setuju
81%-100%	Sangat Setuju

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA