

APLIKASI KAMUS BAHASA INDONESIA – JERMAN ONLINE DENGAN MENGGUNAKAN ALGORITMA BOYER-MOORE

**Mhd. Andre Wahyuda Lubis, Mhd. Zulfansyuri Siambaton, Heri Santoso,
Khairuddin Nasution**

Program Studi Teknik Informatika, Fakultas Teknik Universitas Islam Sumatera Utara
andrewahyuda1505@gmail.com; zulfansyuri@ft.uisu.ac.id; herisantoso@uisu.ac.id

Abstrak

Pengaruh teknologi informasi benar adanya, penggunaan smartphone, tablet dan komputer sudah sangat awam dikalangan masyarakat. Begitupula dengan penggunaan internet, masyarakat sudah menggunakan internet seperti menggunakan buku. Seperti halnya dengan kamus, dulu masyarakat menggunakan kamus dengan membuka kamus yang berbentuk buku, seiring dengan zaman dan kemajuan teknologi informasi yang melonjak cepat maka kamus online pun muncul untuk membantu menerjemahkan kata atau kalimat dengan lebih mudah dan cepat. Aplikasi ini menggunakan sarana website, sehingga user akan lebih mudah lagi menggunakan tanpa harus mengurangi penyimpanan smartphone mereka. Penelitian aplikasi ini dilakukan dengan melihat minat dan kesulitan pelajar dalam menggunakan buku kamus. Teknik yang dilakukan dalam penggunaan aplikasi ini adalah pengguna dapat menerjemahkan kata dengan mencari pada search engine yang ada didalam aplikasi ini. Algoritma Boyer-Moore adalah algoritma yang dibutuhkan untuk membuat aplikasi penerjemah/kamus online ini. Metode pencarian string dengan algoritma Boyer-Moore adalah salah satu algoritma yang terbukti lebih efisien. Algoritma ini menggunakan teknik pencocokan string dari kanan ke kiri, metode seperti ini akan mendapatkan informasi yang lebih banyak.

Kata Kunci : *Kamus Online, Website, Penerjemah, Algoritma Boyer-Moore*

I. PENDAHULUAN

Bahasa memiliki peran penting dalam kehidupan manusia karena ia menjadi alat komunikasi yang utama. Sebagai alat komunikasi, bahasa meliputi kata, kumpulan kata, klausa dan kalimat yang diungkapkan secara lisan maupun tulisan. Sementara pengertian bahasa adalah sistem komunikasi manusia yang dinyatakan melalui susunan suara atau ungkapan tulis yang terstruktur untuk membentuk satuan yang lebih besar, seperti morfem, kata, dan kalimat.

Bahasa Jerman adalah bagian bahasa Germanik Barat yang dipakai sebagai bahasa pengantar terutama di kawasan Eropa Tengah. Bahasa ini adalah salah satu bahasa yang lapang dipertuturkan di Eropa dan pernah menjadi bahasa pengantar antarbangsa yang penting sampai awal 100 tahun ke-20. Meskipun sekarang menurun kebutuhannya, bahasa ini sedang lapang dipelajari karena banyak literatur klasik dunia yang ditulis menggunakan bahasa ini.

Dalam kontinum variasi bahasa di Eropa Tengah bagian utara, "bahasa Jerman" adalah semua bahasa yang telah mengalami pergeseran bunyi Germanik kedua. Dengan demikian, ke dalam lingkup ini masuk berbagai bahasa yang dipakai di Jerman selatan, sehingga bahasa Jerman dikenal pula sbg bahasa Jerman Hulu.

Banyaknya orang-orang yang ingin mengetahui Bahasa Jerman membuat banyak kamus serta buku yang dikeluarkan sebagai alat untuk mempelajari bahasa tersebut. Bahkan, pada era ini Bahasa Jerman menjadi salah satu bahasa asing yang diajarkan di

Sekolah Menengah Atas (SMA). Dengan banyaknya kebutuhan dalam mempelajari bahasa asing ini, Kamus Bahasa Jerman merupakan alat yang salah satu alat yang sangat memudahkan orang dalam mempelajari bahasa asing ini. Pencarian kata per kata sangat mudah didapatkan, sehingga tidak sulit untuk mengetahui kata yang ingin diketahui.

Banyak orang yang sudah membuat kamus Bahasa Jerman untuk lebih memudahkan banyak masyarakat mendapatkan alat tersebut. Adapun Kamus Bahasa Jerman tidak hanya ada dalam bentuk buku, tetapi sudah merambah secara digital berbentuk aplikasi. Penggunaan kamus dalam bentuk aplikasi memudahkan serta mempercepat pencarian kata.

Algoritma Boyer-Moore adalah algoritma pencarian string yang paling efektif saat ini. Algoritma yang ditemukan oleh Bob Boyer dan J. Strother Moore ini telah menjadi standar untuk berbagai literatur pencarian string. Algoritma Boyer-Moore akan menyimpan informasi pergeseran untuk melakukan pencarian string. Karakteristik utama dari algoritma Boyer-Moore adalah algoritma ini melakukan pencocokan string mulai dari kanan ke kiri. Dengan karakteristik tersebut, ketidakcocokan saat terjadi perbandingan string akan membuat pergerakan pattern melompat lebih jauh untuk menghindari perbandingan karakter pada string yang diperkirakan gagal. Aplikasi ini dibuat dengan menggunakan metode string matching dan algoritma Boyer Moore, yang di harapkan bisa membantu dalam system pencarian kata yang di inginkan user.

II. TINJAUAN PUSTAKA

2.1 Aplikasi

Aplikasi adalah program yang dibuat oleh pemakai yang ditujukan untuk melakukan suatu tugas khusus (Kadir, 2003).

Menurut Kadir (2008:3) program aplikasi adalah program siap pakai atau program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Aplikasi juga diartikan sebagai penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan atau sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu. Aplikasi software yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi 2 (dua) yaitu:

- a. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
- b. Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

2.3 Kamus Online

Kamus adalah buku acuan yang memuat kata dan ungkapan, biasanya disusun menurut abjad beserta penjelasan tentang makna dan pemakainya.

Online adalah keadaan komputer yang terkoneksi/ terhubung ke jaringan Internet. Sehingga apabila komputer kita *online* maka dapat mengakses internet/ *browsing*, mencari informasi-informasi di internet. Kamus *Online* adalah buku acuan yang memuat kata dan ungkapan yang disusun menurut abjad beserta penjelasannya yang di akses melalui jaringan mendunia.

2.4 Kamus Besar Bahasa Indonesia

Kamus Besar Bahasa Indonesia (KBBI) adalah kamus ekabahasa resmi bahasa Indonesia yang disusun oleh Badan Pengembangan dan Pembinaan Bahasa dan diterbitkan oleh Balai Pustaka. Kamus ini menjadi acuan tertinggi bahasa Indonesia yang baku, karena merupakan kamus bahasa Indonesia terlengkap dan paling akurat yang pernah diterbitkan oleh penerbit yang memiliki hak paten dari Pemerintah Indonesia yang dinaungi oleh Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia.

2.5 Kamus Bahasa Jerman

Bahasa Jerman adalah salah satu bagian dari bahasa Jermanik Barat yang juga banyak digunakan dalam bahasa pengantar Denmark, Afrika, Frisian, dan Inggris hingga Yiddish. Semua bahasa Jerman berasal dan berkembang dari Proto Jermanik yang berada di tahun 500 SM. Hal ini bisa ditelusuri

melalui negara Skandinavia dimana banyak migrasi kala itu terjadi ke wilayah Jerman.

Kamus Bahasa Jerman adalah buku yang berisi kata per kata Bahasa Jerman yang sudah diartikan ke dalam Bahasa Indonesia.

2.6 Website

1. Pengertian Website

Website adalah kumpulan halaman situs yang terdapat dalam sebuah *domain* atau *subdomain* pada jaringan *World Wide Web (WWW)* di Internet.

Website merupakan salah satu *platform* yang paling sering di akses untuk mencari berbagai informasi dan sarana komunikasi. Terlebih saat ini sudah banyak aplikasi dan tutorial cara membuat website sendiri dari nol tanpa *coding* yang membuat proses pembuatan web semakin mudah dan pertumbuhan jumlah *website* selalu meningkat di Indonesia.

2. Jenis – Jenis Website

a. Website Statis

Website statis adalah *website* yang mempunyai tampilan tetap tidak berubah. Konten dari *website* statis juga sama, jarang sekali terjadi update.

Biasanya suatu *website* statis dibuat menggunakan kode HTML oleh seorang developer. Pembuatnya juga harus memahami kode HTML dan pemrograman. *Website* ini dibuat dengan bantuan text editor atau program design *website* seperti Adobe Dream Weaver. Saat ini untuk membuat *website* statis sudah disediakan sarana sendiri seperti gohugo.io dan hexo.io.

Kelebihan *website* statis adalah cepat dan efisien untuk dikembangkan serta disimpan pada web hosting. Sedangkan kekurangannya adalah membutuhkan skill khusus untuk mengubah konten, kurang bermanfaat karena konten menjadi cepat usang (out of date).

b. Website Dinamis

Website dinamis adalah *website* yang kontennya selalu diperbarui secara berkala. Perubahan atau penambahan konten bisa dilakukan oleh seorang developer atau dilakukan oleh beberapa orang yang memiliki akses untuk memperbarui konten. Kebanyakan *website* saat ini berupa *website* dinamis karena pengelolaannya yang mudah.

Keuntungan dari *website* dinamis ini adalah memiliki banyak fitur tambahan, lebih mudah menambah konten baru untuk menambah pengunjung baru dari search engine, dan lebih mudah dikelola bahkan oleh 2 orang atau lebih. Sedangkan kekurangan *website* dinamis adalah membutuhkan waktu dan dana yang lebih banyak untuk melakukan pengembangan serta biaya hosting yang lebih mahal.

2.7 Algoritma

Algoritma adalah sistem kerja komputer memiliki *brainware*, *hardware*, dan *software*. Tanpa salah satu dari ketiga sistem tersebut, komputer tidak akan berguna. Kita akan lebih fokus pada *software* komputer. *Software* terbangun atas susunan program dan *syntax* (cara penulisan/pembuatan program). Untuk menyusun program atau *syntax*, diperlukannya langkah-langkah yang sistematis dan logis untuk dapat menyelesaikan masalah atau tujuan dalam proses pembuatan suatu *software*. Maka, algoritma berperan penting dalam penyusunan program atau *syntax* tersebut.

Pengertian algoritma adalah susunan yang logis dan sistematis untuk memecahkan suatu masalah atau untuk mencapai tujuan tertentu. Dalam dunia komputer, algoritma sangat berperan penting dalam pembangunan suatu *software*. Dalam dunia sehari-hari, mungkin tanpa kita sadari algoritma telah masuk dalam kehidupan kita.

2.8 Algoritma Boyer-Moore

Algoritma *Boyer-Moore* yaitu salah satu algoritma pencarian *string*, dipublikasikan oleh Robert S. Boyer, dan J. Strother Moore pada tahun 1977.

Algoritma ini dianggap sebagai algoritma yang paling efisien pada aplikasi umum. Tidak seperti algoritma pencarian *string* yang ditemukan sebelumnya, algoritma *Boyer-Moore* mulai mencocokkan karakter dari sebelah kanan pattern. Ide dibalik algoritma ini yaitu bahwa dengan memulai pencocokkan karakter dari kanan, dan bukan dari kiri, maka akan banyak informasi yang didapat.

Algoritma Boyer-Moore mulai mencocokkan pattern pada awal teks.

Dari kanan ke kiri, algoritma ini akan mencocokkan karakter per karakter pattern dengan karakter di teks yang selaras, sampai salah satu kondisi berikut dipenuhi:

- Karakter di pattern dan di teks yang dibandingkan tidak cocok (mismatch).
- Semua karakter di pattern cocok. Belakangan algoritma akan memberitahukan penemuan di jabatan ini.
- Algoritma belakangan menggeser pattern dengan memaksimalkan nilai penggeseran *good-suffix* dan penggeseran *bad-character*, lalu mengulangi langkah 2 sampai pattern berada di ujung teks.
- Pseudocode

Berikut yaitu pseudocode algoritma Boyer-Moore pada fase pra-pencarian:

```
procedure preBmBc(
input P : array[0..n-1] of char,
input n : integer,
input/output bmBc : array[0..n-1] of integer
)
```

Deklarasi:

i: integer

Algoritma:

```
for (i := 0 to ASIZE-1)
bmBc[i] := m;
endfor
for (i := 0 to m - 2)
bmBc[P[i]] := m - i - 1;
endfor
```

```
procedure preSuffixes( input P : array[0..n-1] of
char, input n : integer, input/output suff : array[0..n-1] of integer)
```

Deklarasi: f, g, i: integer

Algoritma: suff[n - 1] := n; g := n - 1; for (i := n - 2 downto 0)

```
{ if (i > g and (suff[i + n - 1 - f] < i - g)) suff[i] :=
suff[i + n - 1 - f];
```

```
else if (i < g) g := i; endif f := i; while (g >= 0 and
P[g] = P[g + n - 1 - f]) --g;
```

```
end while
```

```
suff[i] = f - g;
```

```
end if
```

```
end for
```

```
procedure preBmGs( input P : array[0..n-1] of char,
input n : integer, input/output bmBc : array[0..n-1] of integer)
```

Deklarasi: i, j: integer

suff: array [0..RuangAlphabet] of integer
preSuffixes(x, n, suff);

```
for (i := 0 to m-1) bmGs[i] := n
```

```
endfor j := 0
```

```
for (i := n - 1 downto 0)
```

```
if (suff[i] = i + 1) for (j:=j to n - 2 - i)
```

```
if (bmGs[j] = n) bmGs[j] := n - 1 - i
```

```
end if
```

```
end for
```

```
end if
```

```
end for
```

```
for (i = 0 to n - 2)
```

```
bmGs[n - 1 - suff[i]] := n - 1 - i;
```

```
end for
```

Dan berikut yaitu pseudocode algoritma Boyer-Moore pada fase pencarian:

```
procedure Boyer Moore Search( input m, n :
integer, input P : array[0..n-1] of char, input T :
array[0..m-1] of char, output ketemu : array[0..m-1] of boolean)
```

Deklarasi: i, j, shift, bmBcShift, bmGsShift: integer

BmBc : array[0..255] of integer

BmGs : array[0..n-1] of integer

Algoritma : preBmBc(n, P, BmBc) preBmGs(n, P, BmGs)

```
i:=0while (i<= m-n)
```

```
do j:=n-1
```

```
while (j >=0 n and T[i+j] = P[j]) do j:=j-1
```

```
endwhile if(j < 0)
```

```
then ketemu[i]:=true;
```

```
end if
```

```
bmBcShift:= BmBc[chartoint(T[i+j])]-n+j+1
```

```
bmGsShift:= BmGs[j]
```

```
shift:= max(bmBcShift, bmGsShift)
```

```
i:= i+shift
```

Kompleksitas

III. METODOLOGI PENELITIAN

3.1 Instrumen Pembuatan Aplikasi

1. Perangkat Keras

Tabel 1. Perangkat Keras (Hard ware)

Perangkat Keras	Spesifikasi
Processor	Intel Core i3-3217U 1.8 GHz
RAM	2 GB DDR 3
Hardisk	500 GB

2. Perangkat Lunak

Tabel 2. Perangkat Lunak (Soft ware)

Perangkat Lunak	Spesifikasi
Sistem Operasi	Windows 10 64-bit
Text Editor	Microsoft Visual Studio Code
Web Browser	Google Chrome
Bahasa Pemrograman	Javascript,PHP

3.2 Teknik Pengumpulan Data

a. Studi Literatur

Studi literatur atau studi kepustakaan dilakukan dengan mengambil data dari buku, jurnal ataupun artikel mengenai teori dan langkah-langkah dalam pembuatan aplikasi. Sehingga dapat menggunakannya sebagai dasar landasan teori dan perancangan pembuatan aplikasi Kamus Bahasa Indonesia – Jerman.

Langkah-Langkah Algoritma *Boyer-Moore*

1. Buat tabel pergeseran string yang dicari (S) dengan pendekatan Match Heuristic (MH) dan Occurrence Heuristic (OH), untuk menentukan jumlah pergeseran yang akan dilakukan jika mendapat karakter tidak cocok pada proses pencocokkan dengan string (T).
2. Jika dalam proses perbandingan terjadi ketidakcocokkan antara pasangan karakter pada S dan karakter pada T, pergeseran dilakukan dengan memilih salah satu nilai pergeseran dari dua tabel analisa string yang memiliki nilai pergeseran paling besar.
3. Dua kemungkinan penyelesaian dalam melakukan pergeseran S, jika sebelumnya belum ada karakter yang cocok adalah dengan melihat nilai pergeseran hanya pada tabel Occurrence Heuristic, jika karakter yang tidak cocok tidak ada pada S, maka pergeseran adalah sebanyak jumlah karakter pada S; dan jika karakter yang tidak cocok ada pada S, maka banyaknya pergeseran bergantung dari nilai pada tabel.
4. Jika karakter pada teks yang sedang dibandingkan cocok dengan karakter pada S, maka posisi karakter pada S dan T diturunkan sebanyak 1 posisi, kemudian dilanjutkan dengan

pencocokkan pada posisi tersebut dan seterusnya. Jika kemudian terjadi ketidakcocokkan karakter S dan T, maka dipilih nilai pergeseran terbesar dari dua tabel analisa pattern, yaitu nilai dari tabel Match Heuristic dan nilai tabel Occurrence Heuristic dikurangi dengan jumlah karakter yang telah cocok.

5. Jika semua karakter telah cocok, artinya S telah ditemukan di dalam T, selanjutnya geser pattern sebesar 1 karakter.
6. Lanjutkan sampai akhir string T.

Bad-Character Shift (Occurance Heuristic)

Tabel *Occurrence Heuristic* sering disebut juga *Bad-Character Shift*, dimana pergeserannya dilakukan berdasarkan karakter apa yang menyebabkan tidak cocok dan seberapa jauh karakter tersebut dari karakter paling akhir. Untuk menghitung tabel Occurrence Heuristic harus menggunakan langkah-langkah sebagai berikut :

Contoh string: manaman

Panjang: 7 karakter

Posisi :	1	2	3	4	5
	6	7			
String:	M	A	N	A	M
	A	N			
Pergeseran					
(OH)	2	1	0	1	2
	1	0			

1. Lakukan pencacahan mulai dari posisi terakhir string sampai ke posisi awal, dimulai dengan nilai 0 karena terletak pada jarak terakhir, catat karakter yang sudah ditemukan (dalam contoh ini karakter “n”)
2. Mundur ke posisi sebelumnya, nilai pencacah ditambah 1, jika karakter pada posisi ini belum pernah ditemukan, maka nilai pergeserannya adalah sama dengan nilai pencacah (dalam contoh ini, karakter “a” belum pernah ditemukan sehingga nilai pergeserannya adalah sebesar nilai pencacah yaitu 1)
3. Mundur ke posisi sebelumnya, karakter “m” nilai pergeserannya 2
4. Mundur lagi, karakter “a”. Karakter “a” sudah pernah ditemukan sebelumnya sehingga nilai pergeserannya sama dengan nilai pergeseran karakter “a” yang sudah ditemukan paling awal yaitu 1.
5. Begitu seterusnya sampai posisi awal string.
6. Catatan: untuk karakter selain “m, a, n”, nilai pergeseran sebesar panjang string, yaitu tujuh karakter (sepanjang pattern).

Good-Suffix Shift (Match Heuristic)

Tabel Match Heuristic sering disebut juga Good-Suffix Shift, dimana pergeserannya dilakukan berdasarkan posisi ketidakcocokkan karakter yang terjadi. Maksudnya untuk menghitung tabel Match Heuristic, perlu diketahui pada posisi keberapa

terjadi ketidakcocokkan. Posisi ketidakcocokkan itulah yang akan menentukan besar pergeseran.

Untuk menghitung tabel Match Heuristic harus menggunakan langkah-langkah sebagai berikut [10]:
Contoh string: manaman Panjang: 7 karakter

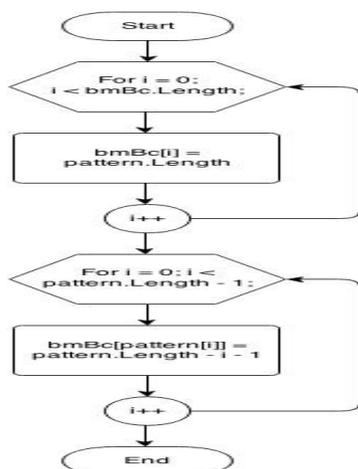
Tabel 3. Tabel Match Heuristic

Posisi	1	2	3	4	5	6	7
String	M	A	N	A	M	A	N
Pergeseran (MH)	4	4	4	4	7	7	1

1. Jika karakter pada posisi 7 bukan “n” maka geser 1 posisi, berlaku untuk semua pattern yang dicari.
2. Jika karakter “n” sudah cocok, tetapi karakter sebelum “n” bukan “a”, maka geser sebanyak 7 posisi, sehingga posisi pattern melewati teks, karena sudah pasti “manambn” bukan “manaman”
3. Jika karakter “an” sudah cocok, tetapi karakter sebelum “an” bukan “m” maka geser sebanyak 7 posisi, sehingga posisi pattern melewati teks, karena sudah pasti “manaban” bukan “manaman”
4. Jika karakter “man” sudah cocok, tetapi karakter sebelum “man” bukan “a” maka geser sebanyak 4 posisi, sehingga posisi pattern berada atau bersesuaian dengan akhiran “man” yang sudah ditemukan sebelumnya, karena bisa saja akhiran “man” yang sudah ditemukan sebelumnya merupakan awalan dari pattern “manaman” yang berikutnya.
5. Jika karakter “aman” sudah cocok, tetapi karakter sebelum “aman” bukan “n” maka geser sebanyak 4 posisi, sehingga posisi pattern berada/bersesuaian dengan akhiran “man” yang sudah ditemukan sebelumnya, karena bisa saja akhiran “man” yang sudah ditemukan sebelumnya merupakan awalan dari pattern “manaman” yang berikutnya.

Selanjutnya sama, pergeseran paling mungkin dan aman dalam tabel *MatchHeuristic* adalah pergeseran sebanyak 4 posisi.

Flowchart

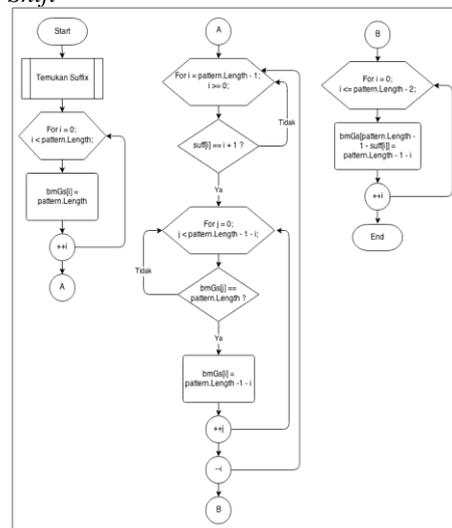


1. Flowchart Perhitungan Tabel *Bad Character Shift*

Gambar 1. Flowchart Perhitungan

Flowchart ini menggambarkan alur kerja yang terjadi pada saat perhitungan tabel *bad-character shift*. Tabel *bad-character shift* berisi nilai-nilai pergeseran yang dapat dilakukan untuk setiap karakter yang terdapat di alfabet, nomor, dan tanda baca yang umum digunakan. Setiap karakter yang ada di *pattern* diberi nilai sesuai dengan ukuran jauhnya karakter tersebut dari karakter paling akhir dari *pattern*. Untuk karakter yang tidak terdapat di dalam *pattern* akan diberi nilai yang sama, yaitu panjang dari *pattern* yang dimasukkan.

2. *Flowchart* Perhitungan Tabel *Good-Suffix Shift*



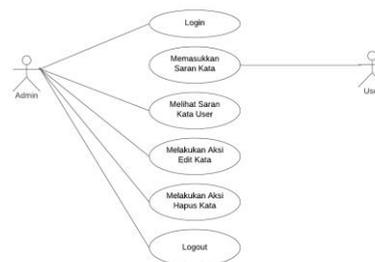
Gambar 2. Flowchart Proses Perhitungan Tabel *Good-Suffix Shift*

Flowchart ini menggambarkan alur yang terjadi pada saat perhitungan tabel *suffix* yang berguna untuk menghitung tabel *good-suffix shift* nantinya. Tabel ini berisi nilai-nilai dari tiap karakter yang ada di *pattern* yang menunjukkan adanya perulangan akhiran (*suffix*) atau tidak dan dimana letak perulangan tersebut sehingga ketika proses perhitungan tabel *good-suffix shift*, dapat diketahui seberapa banyak karakter yang dapat digeser untuk pencocokan karakter selanjutnya.

3.3 Perancangan Sistem

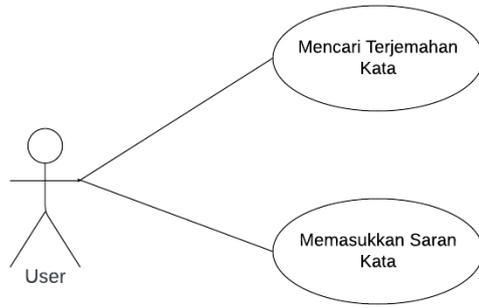
Use Case Diagram

1. *Use Case Diagram Admin*



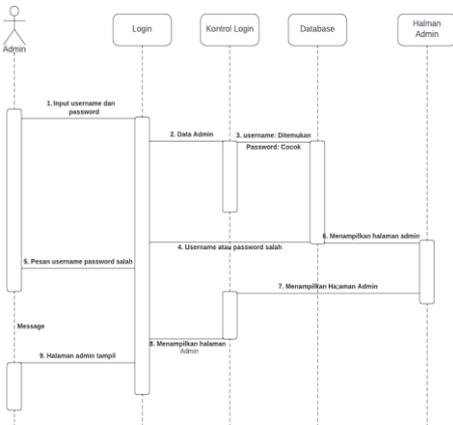
Gambar 3. Use Case Diagram Admin

2. Use Case Diagram User



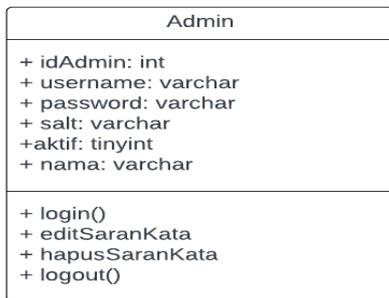
Gambar 4. Use Case Diagram User

3. Sequence Diagram



Gambar 5. Sequence Diagram Admin

4. Class Diagram



Gambar 5. Class Diagram

Perancangan Tabel

1. Tabel Admin

Tabel 4. Tabel Admin	
Nama Kolom	Tipe Data
Id	int(2)
Username	varchar(25)
Password	varchar(225)
Salt	varchar(225)
Aktif	tinyint(1)
Nama	varchar(50)

2. Tabel Saran Kata

Tabel 5. Tabel Saran Kata	
Nama Kolom	Tipe Data
id_admin	int(2)
Kata	varchar(50)
Arti	varchar(50)

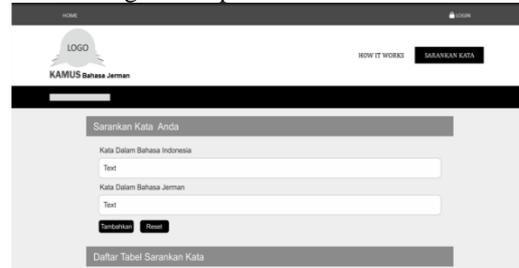
Perancangan Interface (Antarmuka)

1. Perancangan Tampilan Antarmuka User



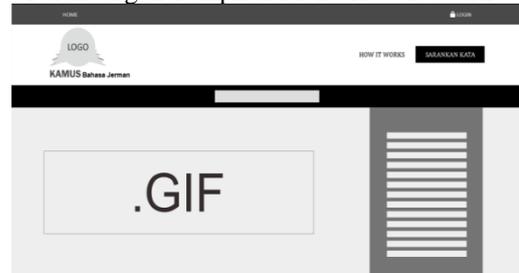
Gambar 6. Perancangan Tampilan Antarmuka User

2. Perancangan Tampilan Menu Saran Kata



Gambar 7. Perancangan Tampilan Menu Saran Kata

3. Perancangan Tampilan Menu How It Work



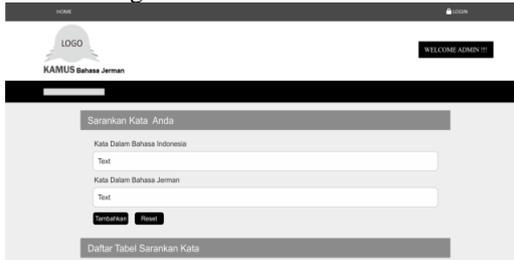
Gambar 8. Perancangan Tampilan Menu How It Work

4. Perancangan Tampilan Menu Login



Gambar 9. Perancangan Tampilan Menu Login

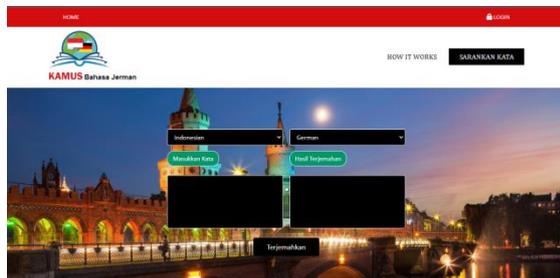
5. Perancangan Halaman Admin



Gamar 10. Perancangan Tampilan Halaman Admin

IV. HASIL DAN PEMBAHASAN

4.1 Implementasi Aplikasi\Tampilan Antarmuka User



Gambar 11. Tampilan Antarmuka User

Pada tampilan awal aplikasi ini, tampilan yang akan ditampilkan adalah seperti gambar diatas. Pada tampilan antarmuka ini, ada beberapa sub menu yang dapat dipilih. Menu login hanya dapat digunakan oleh admin yang sudah didaftarkan sebelumnya dan diinputkan datanya kedalam database admin. Sub menu saran kata memiliki tampilan yang dapat melihat serta menginput saran kata user. Menu saran kata akan memasukkan srnan kata yang diinput oleh user kedalam database saran kata. Sub menu How It Work adalah sub menu yang menampilkan bagaimana cara kerja dari algoritma yang aplikasi ini gunakan.



Gambar 12. Proses Pencarian

Pada halaman ini, user dapat menerjemahkan kata atau kalimat yang ingin dicari dari bahasa Indonesia ke dalam bahasa Jerman maupun sebaliknya. Pada proses penerjemahan ini, program akan mencari arti kata atau kalimat dengan menggunakan algoritma Boyer-Moore, dimana pencarian string akan dilakukan dari kanan ke kiri. Setelah user melakukan aksi pengisian penerjemahan, user harus menekan tombol terjemahkan agar user dapat melihat hasil dari proses penerjemahan ini.



Gambar 13. Proses Penerjemahan

Pada Gambar 13 user akan menerima hasil dari pencarian yang user lakukan. User dapat melihat hasil penerjemahan pada kolom sebelah kanan.

1. Tampilan Menu Sarankan Kata



Gambar 14. Tampilan Menu Sarankan Kata

Pada sub menu sarankan kata, user dapat memberikan saran kata yang tidak dapat ditemukan didalam pencarian kata. User akan diminta memasukkan saran kata dalam Bahasa Indonesia lalu menuliskan arti dari kata yang user masukkan. Setelah itu, user harus menekan tombol tambahkan agar kata dapat masuk kedalam tabel. Jika user salah memasukkan kata, user dapat menekan tombol reset untuk menghapus kata yang dimasukkan

ID	KATA DALAM BAHASA INDONESIA	KATA DALAM BAHASA JERMAN
1	manan	Alam
2	mpj	Mangan
3	ring	Rechnung
4	nkur	Schuler



Gambar 15. Tabel Sarankan Kata

Kata yang sudah *user* tambahkan sebagai sara kata akan otomatis masuk ke dalam tabel sarankan kata. Maka, kata yang sudah user tambahkan dapat user cari didalam menu penerjemah kata.

2. Menu How It Work

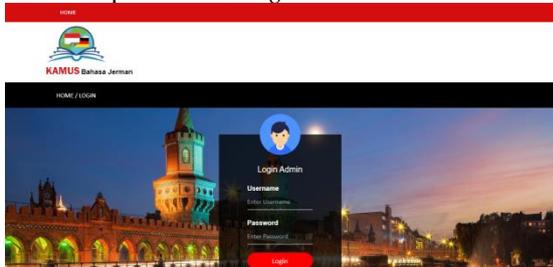


Gambar 16. Tampilan Menu How It Work

Pada tampilan sub menu ini, *user* dapat melihat sekilas pengertian dan cara kerja pencarian *string* algoritma *Boyer-Moore*. Pada tampilan ini, cara kerja algoritma *Boyer-Moore* akan ditampilkan dalam bentuk gambar berjalan yang menjelaskan bahwa algoritma *Boyer-Moore* melakukan pencarian *string* dari kiri ke kanan sampai kata yang dicari memiliki nilai yang sama.

4.2 Tampilan Antarmuka Admin

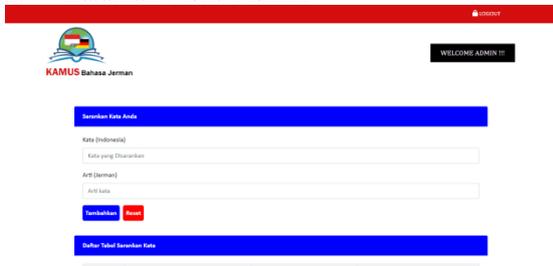
1. Tampilan Menu Login



Gambar 17. Tampilan Menu Login

Menu *login* merupakan menu yang akan menghubungkan admin dengan menu admin. Didalam menu *login* admin ini hanya dapat dilakukan oleh admin yang sebelumnya sudah terdaftar sebagai admin dimana datanya sudah terinput didalam *database* admin.

2. Halaman Menu Admin



Gambar 18. Tampilan Awal Menu Admin

Pada tampilan awal menu admin hampir sama dengan tampilan sub menu sarankan kata. Yang

membedakan adalah admin memiliki kuasa untuk menghapus dan mengedit kata yang salah atau tidak sesuai.

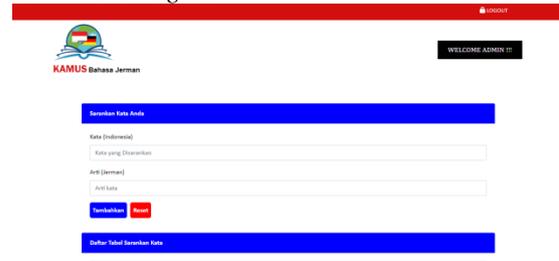
No	Kata (Indonesia)	Arti (Jerman)	Aksi
1	malam	Abend	edit delete
2	pagi	Morgen	edit delete
3	siang	Nachmittag	edit delete
4	malu	Schämen	edit delete



Gambar 19. Tampilan Tabel Pada Menu Admin

Pada tabel menu admin inilah admin dapat melakukan aksi hapus dan edit. Ketika salah satu kata dalam tabel dihapus, maka kata tersebut akan hilang dan tidak dapat dicari juga didalam pencarian. Ketika sebuah kata didalam tabel diedit maka kata dan artian yang akan diedit akan masuk kedalam sarankan kata.

3. Menu Logout



Gambar 20. Tampilan Menu Logout

Ketika masuk ke dalam menu admin, admin dapat melakukan *logout* akun. Ketika *logout* akun dijalankan, maka admin akan kembali ke dalam *login*.

V. KESIMPULAN

Dari hasil pembahasan dapat disimpulkan sebagai berikut:

1. Aplikasi ini merupakan suatu sarana yang dapat mempermudah *user* untuk mencari atau menerjemahkan suatu kata dari bahasa Indonesia ke bahasa Jerman tanpa harus mempunyai sebuah aplikasi. *User* hanya perlu membuka url *website* dari kamus berbasis *online* ini.
2. Pada aplikasi kamus online ini, ada beberapa sub menu dan menu lain didalamnya selain dari menu penerjemahan. Sub menu yang terdapat didalamnya antara lain: sarankan kata dan *how it work*. Menu lain yang ada pada aplikasi ini adalah: *login* dan menu *admin*.

DAFTAR PUSTAKA

[1]. Abdullah, R., 2015, *Web Programming Is Easy*. Jakarta: PT. Elex Media Komputindo.

- [2]. Agung, G., 2000, *Membuat Homepage Interaktif Dengan CGI/Perl*. Jakarta: PT. Elex Media Komputindo.
- [3]. Bekti, H., 2015, *Mahir Membuat Website dengan Adobe Dreamweaver CS6, CSS dan JQuery*. Yogyakarta: ANDI.
- [4]. Hidayat, R., 2010. *Cara Praktis Membangun Website Gratis: Pengertian Website*. Jakarta: PT. Elex Media Komputindo Kompas, Gramedia.
- [5]. Kadir, A., 2008, *Dasar Aplikasi Database Mysql*. Yogyakarta: Andi Offset.
- [6]. Lado, V. H., 2022, September 26). *Pengertian Bahasa, Peran & Fungsi Bahasa secara Umum di Masyarakat*. Retrieved Februari 15, 2023, from *tirto.id*: <https://tirto.id/pengertian-bahasa-peran-fungsi-bahasa-secara-umum-di-masyarakat-gdhW>
- [7]. Lecroq, T. C., 2001, *Handbook of Exact String Matching Algorithm*. ISBN 0-9543006-4-5.
- [8]. Lukmanul, H., 2004, *Website Merupakan Fasilitas Internet*. Jakarta: Gramedia.
- [9]. Permana, A., 2015,. *Rancang Bangun Kamus Online Berbasis Web*. lppm.atmaluhur, 15.
- [10]. Pramana, H. W., 2012, *Aplikasi Inventory Berbasis Access 2003*. Jakarta: PT. Elex Media Komputindo.
- [11]. Rahmanita, E., 2014, *Pencarian String Menggunakan Algoritma Boyer Moore pada Dokumen*. Nero, 12.
- [12]. Santoso, H., 2010, *Aplikasi Web/asp.net + cd*. Jakarta: Elex Media Komputindo.
- [13]. Sholehul, A., 2013, *Gampang dan Gratis Membuat Website Untuk Pemula*. Jakarta: Kunci Komunikasi.
- [14]. Susilowati, Y., 2009, *Modul E-Commerce-Teaching Factory For Students*. Jakarta: Mutiara Publisher.
- [15]. Tech. 2022, Juni 18). *7 Pengertian Website Menurut Ahli, Lengkap Jenis & Fungsinya*. Retrieved Februari 16, 2023, from CNBC Indonesia: <https://www.cnbcindonesia.com/tech/20220618152119-37-348229/7-pengertian-website-menurut-ahli-lengkap-jenis-fungsinya>
- [16]. Widianti, S., 2000, *Pengantar Basis Data*. Yogyakarta: Andi Offset.
- [17]. Zaki, A., & Community, S., 2011, *Kiat Jitu Membuat Website*. Bandung: Elex Media Komputindo.