

PENERAPAN FITUR AUTOCOMPLETE DAN ALGORITMA TURBO BOYER MOORE PADA APLIKASI KAMUS KIMIA

Firda Novina¹, Tony Darmanto², Antonius³

^{1,2,3}Informatika, Fakultas Teknologi Informasi Universitas Widya Dharma, Pontianak
e-mail: ¹19421294_firda_n@widyadharm.ac.id, ²tony.darmanto@yahoo.com, ³antoniusok@yahoo.com

Abstract

This research implements the autocomplete feature and the Turbo Boyer Moore algorithm on an Android-based chemical term dictionary application. The autocomplete feature allows users to quickly search for chemical terms by entering only the first few letters, while the Turbo Boyer Moore algorithm is used for efficient string matching. This research uses data collection techniques, object-oriented system analysis, and application design techniques using Android Studio and SQLite. The result is an Android-based chemical term dictionary application that speeds up the search process and display relevant word suggestions. The conclusion that researchers can give from the results of applying the autocomplete feature and the Turbo Boyer Moore algorithm in this chemical term dictionary application is that it can search quickly and accurately, and minimizes time in typing patterns. The author gives advice to readers to be able to add other features such as the autocorrect features, voice recognition and can combine it with other algorithms.

Keywords: Turbo Boyer Moore Algorithm, Autocomplete, Dictionary, Chemical Term Dictionary.

Abstrak

Penelitian ini mengimplementasikan fitur *autocomplete* dan algoritma Turbo Boyer Moore pada aplikasi kamus istilah kimia berbasis Android. Fitur *autocomplete* memungkinkan pengguna untuk dengan cepat mencari istilah kimia dengan memasukkan beberapa huruf pertama saja, sementara algoritma Turbo Boyer Moore digunakan untuk pencocokan *string* yang efisien. Penelitian ini menggunakan teknik pengumpulan data studi literatur, analisis sistem berorientasi objek serta teknik perancangan aplikasi menggunakan Android Studio dan SQLite. Hasilnya adalah sebuah aplikasi kamus istilah kimia berbasis Android yang dapat mempercepat proses pencarian dan menampilkan saran kata yang relevan. Kesimpulan yang dapat peneliti berikan dari hasil penerapan fitur *autocomplete* dan algoritma Turbo Boyer Moore pada aplikasi kamus istilah kimia ini adalah dapat melakukan pencarian dengan cepat dan akurat, serta meminimalkan waktu dalam pengetikan *pattern*. Penulis memberikan saran kepada pembaca agar dapat menambahkan fitur-fitur lain seperti fitur *autocorrect*, *voice recognition* serta dapat mengkombinasikannya dengan algoritma lainnya.

Kata kunci: Algoritma Turbo Boyer Moore, Autocomplete, Kamus, Kamus Istilah Kimia.

1. PENDAHULUAN

Perkembangan teknologi informasi yang pesat berperan penting dalam berbagai bidang kehidupan. Hampir semua manusia saat ini memanfaatkan teknologi untuk mempermudah pekerjaan dan kehidupan sehari-hari. Setiap teknologi informasi memiliki tempat penyimpanan data yang berbeda, namun teks merupakan bentuk utama dari penyimpanan data. Seringkali operasi pada teks melibatkan proses pencarian terhadap kemunculan suatu *string* dan lokasinya. Oleh karena itu, pencarian *string* merupakan salah satu yang diperlukan untuk dapat menemukan data di dalam sebuah penyimpanan.

Algoritma yang digunakan untuk pencarian *string* semakin berkembang dari hari ke hari. Hal ini tentu saja bertujuan untuk mencari atau menemukan pencarian *string* yang seakurat dan secepat mungkin. Hingga saat ini algoritma pencarian *string* terbagi menjadi tiga kategori berdasarkan arah pencocokan *string* yaitu dari kiri ke kanan, kanan ke kiri dan dari arah yang ditentukan secara spesifik.

Algoritma Turbo Boyer Moore merupakan salah satu algoritma yang pencocokan *string*-nya dari arah kanan ke kiri. Algoritma Turbo Boyer Moore merupakan pengembangan dari algoritma Boyer Moore. Pada algoritma ini pencocokan kata dimulai dari karakter terakhir kata kunci menuju karakter awal. Algoritma ini cocok untuk diterapkan dalam aplikasi yang membutuhkan pencarian yang cepat, seperti aplikasi pendukung pembelajaran yaitu aplikasi kamus. Untuk mempermudah dalam melakukan pencarian istilah yang ada dapat menambahkan fitur *autocomplete*. Fitur *Autocomplete* adalah sebuah fitur yang berfungsi untuk menampilkan perkiraan kata yang dimasukkan oleh pengguna tanpa harus mengetikkan keseluruhan kata. Dengan adanya fitur *autocomplete* ini, selain aplikasi akan menjadi lebih

responsif, diharapkan pengguna juga dapat lebih terbantu dalam melakukan pencarian.

Dari beberapa hal tersebut, muncullah ide yang mendorong penulis untuk mengimplementasikan fitur *autocomplete* dan algoritma Turbo Boyer Moore dalam kamus istilah kimia. Kamus istilah kimia ini merupakan kamus yang mencakup istilah-istilah yang ada dalam ilmu kimia. Aplikasi kamus ini dapat digunakan oleh pelajar, mahasiswa atau masyarakat yang memiliki minat dan ketertarikan akan ilmu kimia. Aplikasi kamus istilah kimia ini dapat mempermudah dan mempercepat pengguna dalam melakukan pencarian istilah yang diinginkan karena memanfaatkan algoritma Turbo Boyer Moore dan fitur *autocomplete*.

2. METODE PENELITIAN

2.1 Metode Penelitian

2.1.1 Rancangan Penelitian

Dalam penelitian ini, menggunakan perancangan penelitian deskriptif, dengan mengkaji literatur-literatur untuk implementasi fitur *autocomplete* dan algoritma Turbo Boyer Moore dalam aplikasi kamus istilah kimia.

2.1.1.1 Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan adalah studi literatur, yang dimana penulis mengumpulkan data berdasarkan kajian literatur seperti buku, jurnal, *e-book* dan laporan penelitian, yang berkaitan dengan objek penelitian.

2.1.1.2 Teknik Analisis Perangkat Lunak

Teknik analisis perangkat lunak yang digunakan peneliti adalah teknik yang berorientasi dengan objek. Pemodelan UML (*Unified Modeling Language*) digunakan peneliti untuk menjelaskan alur, prosedur, dan proses kerja aplikasi yang dirancang.

2.1.1.3 Teknik Perancangan Perangkat Lunak

Perangkat lunak yang dirancang menggunakan Android Studio sebagai *code editor*, Java sebagai Bahasa pemrograman, serta menggunakan SQLite sebagai *database*.

2.2 Landasan Teori

2.2.1 Data

Data adalah catatan atas kumpulan fakta dunia nyata yang mewakili objek seperti manusia, barang, hewan, konsep, peristiwa dan sebagainya yang diwujudkan dalam bentuk huruf, angka, simbol, gambar, teks, bunyi atau kombinasinya^[1].

2.2.2 Aplikasi

Aplikasi merupakan *software* yang fungsinya untuk melaksanakan berbagai bentuk pekerjaan maupun tugas-tugas tertentu misalnya seperti penerapan, pemakaian dan juga penambahan data^[2].

2.2.3 Perangkat Lunak

Perangkat lunak merupakan kumpulan instruksi dalam bentuk kode program yang ditulis menurut aturan bahasa pemrograman tertentu, disimpan dan dijalankan pada perangkat keras untuk mengerjakan fungsi tertentu^[3].

2.2.4 Database

Basis Data adalah himpunan data yang saling terhubung dan diorganisasi sedemikian rupa supaya kelak dapat dimanfaatkan kembali secara cepat dan mudah. Basis Data juga merupakan kumpulan data dalam bentuk file/tabel/arsip yang saling berhubungan dan tersimpan dalam media penyimpanan elektronik, untuk kemudahan dalam pengaturan, pemilahan, pengelompokan dan pengorganisasian data sesuai tujuan^[1].

2.2.5 Perancangan Sistem

Perancangan sistem adalah merupakan perancangan yang melibatkan sistem tertentu. Dalam arti sistem yang dirancang adalah sistem yang memang ingin dibuat dan dikendalikan^[4].

2.2.6 Kamus

Kamus adalah daftar alfabetis kata-kata yang disertai dengan arti, lafal, contoh penggunaannya dalam kalimat, dan keterangan lain yang berkaitan dengan kata tadi. Kamus memuat bermacam informasi tentang kata dengan segala aspeknya yang disusun secara alfabetis^[5].

2.2.7 Kimia

Kimia adalah ilmu yang mempelajari mengenai komposisi, struktur, dan sifat zat atau materi dari skala atom (mikroskopik) hingga molekul serta perubahan atau transformasi serta interaksi mereka untuk membentuk materi yang ditemukan sehari-hari^[6].

2.2.8 Autocomplete

Fitur *AutoComplete* merupakan fitur yang akan menampilkan daftar nama fungsi berdasarkan huruf tertentu yang anda ketikkan pada suatu formula yang sama dengan huruf awal nama fungsi^[7]. *Autocomplete* adalah fitur pencarian yang memberikan saran untuk istilah pencarian saat pengguna mengetik teks di kotak pencarian^[8].

2.2.9 Algoritma Turbo Boyer Moore

Algoritma Turbo Boyer Moore merupakan algoritma yang digunakan untuk melakukan proses pencarian *string*. Hingga saat ini algoritma pencarian *string* terbagi atas 3 (tiga) kategori berdasarkan arah pencocokan *string* yaitu kiri ke kanan, kanan ke kiri, dan dari arah yang spesifik. Metode pencocokan dari kanan ke kiri merupakan metode yang dianggap paling efisien dalam praktiknya dan pencocokan *string* dari arah yang ditentukan secara spesifik merupakan algoritma yang memiliki hasil yang paling baik secara teoritis^[9].

2.2.10 Android Studio

Android Studio adalah suatu IDE berbasis JavaIDE yang diciptakan oleh JetBrains kemudian diperkenalkan oleh Google. Awal mula Android Studio diumumkan secara resmi adalah pada bulan Mei 2013. Penggunaan Android Studio telah secara signifikan menggantikan Eclipse yang sering digunakan sebagai IDE dalam pengembangan aplikasi mobile berbasis Android^[10].

2.2.11 UML

UML (Unified Modeling Language) adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan artefak (bagian dari informasi yang digunakan untuk dihasilkan oleh proses pembuatan perangkat lunak, artefak tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi objek^[11].

2.2.12 Black Box

Black box testing merupakan pengujian kualitas perangkat lunak yang berfokus pada fungsionalitas perangkat lunak. Pengujian *black box testing* bertujuan untuk menemukan fungsi yang tidak benar, kesalahan antarmuka, kesalahan pada struktur data, kesalahan performansi, kesalahan inialisasi dan terminasi^[12].

3. HASIL DAN PEMBAHASAN

3.1 Analisis Algoritma Turbo Boyer Moore

Algoritma Turbo Boyer Moore adalah pengembangan dari algoritma Boyer Moore merupakan algoritma yang melakukan pencocokan dimulai dari karakter yang paling kanan hingga karakter paling kiri *pattern*. Algoritma Turbo Boyer Moore mengambil metode pencarian *string* Boyer Moore yaitu *good suffix* dan *bad character suffix*. Akan tetapi, algoritma Turbo Boyer Moore dimungkinkan terjadinya lompatan melewati segmen mengingat faktor dan teks yang cocok dengan akhiran dari *pattern* selama *attempt* terakhir sehingga memanfaatkan teknik *turbo shift*. Penggunaan algoritma Turbo Boyer Moore untuk pencarian *pattern* dalam teks, seperti pada contoh berikut:

Teks : Asam Basa

Pattern : Basa

Tabel 1. Penentuan BmBc dan BmGs

<i>i</i>	0	1	2	3
<i>Pattern</i>	B	A	S	A
BmBc	3	2	1	0
<i>suffix</i>	0	0	0	4
BmGs	3	3	3	1

Rumus:

$$BmBc = m - i - 1$$

$$BmGs = m - 1 - i \text{ suffix}$$

m = panjang *pattern*

i = posisi *pattern* pada teks

i suffix = posisi *suffix* pada index pola

3.1.1 Pola Pertama

Tabel 2. Pergeseran I

A	S	A	M		B	A	S	A
B	A	S	A					

Adanya perbedaan pada indeks M, maka dilakukan perhitungan:

$$Shift \text{ by } BmBc[M] - m + (\text{index bawah} + 1) = 4 - 4 + 4 = 4$$

$$Shift \text{ by } BmGs[3] = 1$$

Geser pola sebanyak empat (4) (yaitu nilai maksimal dari kedua perhitungan)

3.1.2 Pola Kedua

Tabel 3. Pergeseran II

A	S	A	M		B	A	S	A
				B	A	S	A	

Adanya perbedaan pada indeks S, maka dilakukan perhitungan:

$$Shift \text{ by } BmBc[S] - m + (\text{index bawah} + 1) = 1 - 4 + 4 = 1$$

$$Shift \text{ by } BmGs[3] = 1$$

Geser pola sebanyak satu (1) (nilai perhitungan sama)

3.1.3 Pola Pertama

Tabel 4. Pergeseran III

A	S	A	M		B	A	S	A
					B	A	S	A

Pada pola ketiga sudah ditemukan kecocokan, maka tidak perlu dilakukan pergeseran lagi.

3.2 Analisis Fitur Autocomplete

Fitur *autocomplete* berfungsi memberikan rekomendasi sejumlah kata yang serupa dengan kata kunci yang diketikkan *user* pada kotak pencarian. Dengan adanya fitur *autocomplete* dapat mempermudah *user* dalam melakukan pencarian kata, karena *user* tidak perlu mengetikkan kata secara keseluruhan. Apabila kata kunci yang diketikkan *user* tidak terdapat di *database* maka fitur *autocomplete* ini tidak akan bekerja. Jika *user* salah memasukkan kata kuncinya maka tidak akan ada rekomendasi perbaikan kata yang keluar, karena fitur *autocomplete* hanya berfungsi menampilkan kata sesuai dengan *pattern* yang di-*input*-kan.

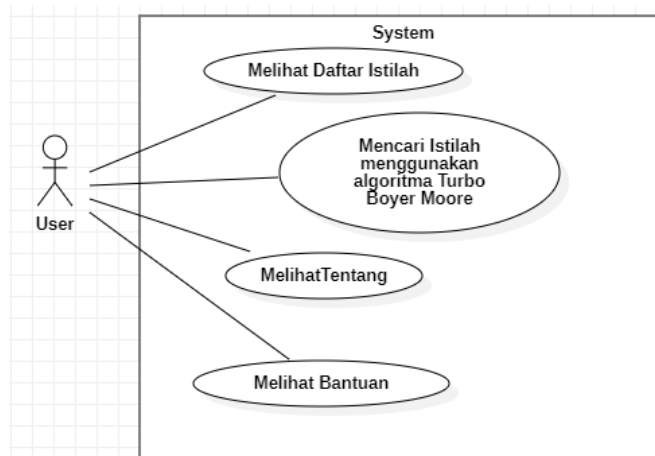
Pada aplikasi kamus istilah kimia, *user* dapat mencari istilah kimia yang terdapat dalam *database*. Proses pengambilan data dilakukan ketika *user* mengetikkan kata kunci (*pattern*) pada kotak pencarian. Sistem akan membandingkan *pattern* yang diketikkan pada kotak pencarian dengan semua kata yang ada pada *database*. Ketika ada kecocokan antara *pattern* dengan data pada *database*, maka akan ditampilkan sebuah daftar (*listview*) yang berisi semua istilah dari *database* yang mengandung *pattern* yang diketikkan. Setelah itu, *user* dapat memilih sebuah istilah yang ingin ditampilkan dari sejumlah istilah pada *listview* yang dihasilkan. Sebaliknya, jika tidak ada kecocokan antara *pattern* yang diketikkan dengan data yang ada pada *database*, maka akan ditampilkan sebuah pesan bahwa kata kunci yang dimasukkan tidak ditemukan.

3.3 Perancangan Unified Modeling Language (UML)

Untuk menggambarkan prosedur, aliran atau proses kerja sistem dari perancangan aplikasi ini penulis menggunakan teknik berorientasi objek dengan Unified Modeling Language (UML) sebagai alat pemodelannya. Diagram UML yang digunakan dalam perancangan aplikasi kamus istilah kimia yaitu diagram *use case* dan *sequence diagram* atau diagram urutan.

3.4 Diagram Use Case

Perancangan diagram *use case* bertujuan untuk menggambarkan model dari aplikasi kamus istilah kimia dimana *user* akan berinteraksi dengan sistem yang dibangun. Diagram *use case* pada perancangan aplikasi kamus istilah kimia dapat dilihat pada gambar 1 berikut ini:



Gambar 1. Use Case Diagram aplikasi

Dari diagram *use case* pada gambar 1, diketahui bahwa terdapat empat kegiatan utama dalam aplikasi kamus istilah kimia ini yaitu:

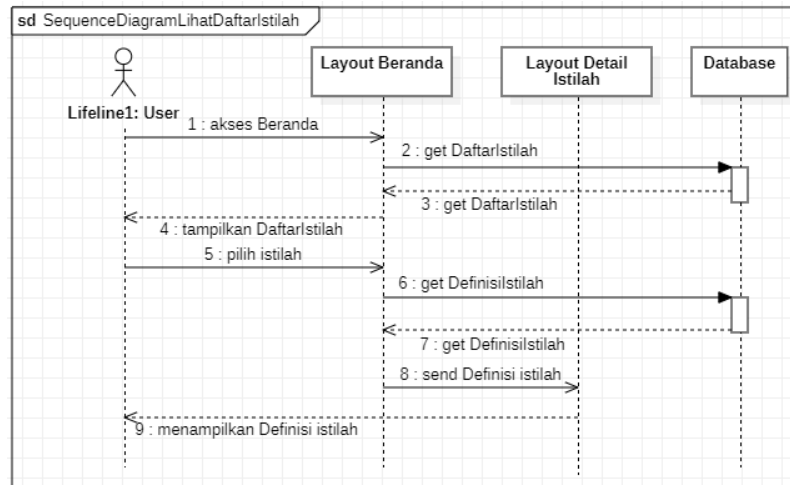
- Daftar Istilah Kimia
- Daftar istilah merupakan menu utama dari aplikasi kamus istilah kimia karena saat *user* masuk dalam aplikasi maka semua daftar istilah yang tersedia akan langsung ditampilkan.
- Pencarian Istilah
- Pencarian istilah merupakan menu yang berfungsi untuk melakukan pencarian istilah berdasarkan *pattern* yang di-*input*-kan ke dalam kotak pencarian dengan menggunakan algoritma Turbo Boyer Moore. Hasil dari *pattern* yang di-*input*-kan akan ditampilkan dalam bentuk *listview*.
- Tentang
- Tentang adalah sebuah halaman yang berisi tentang informasi mengenai *developer* aplikasi, seperti nama, *email*, universitas dan lainnya.

- g. Bantuan
Bantuan adalah sebuah halaman yang berfungsi untuk menampilkan informasi yang dapat membantu pengguna dalam menggunakan aplikasi.

3.5 Diagram Urutan (Sequence Diagram)

Diagram urutan atau *sequence diagram* berfungsi untuk memperlihatkan interaksi-interaksi yang terjadi pada saat menjalankan aplikasi kamus yang disusun berdasarkan urutan atau rangkaian waktu. Berikut ini *sequence diagram* yang digunakan pada perancangan aplikasi:

- a. Diagram Urutan Lihat Daftar Istilah

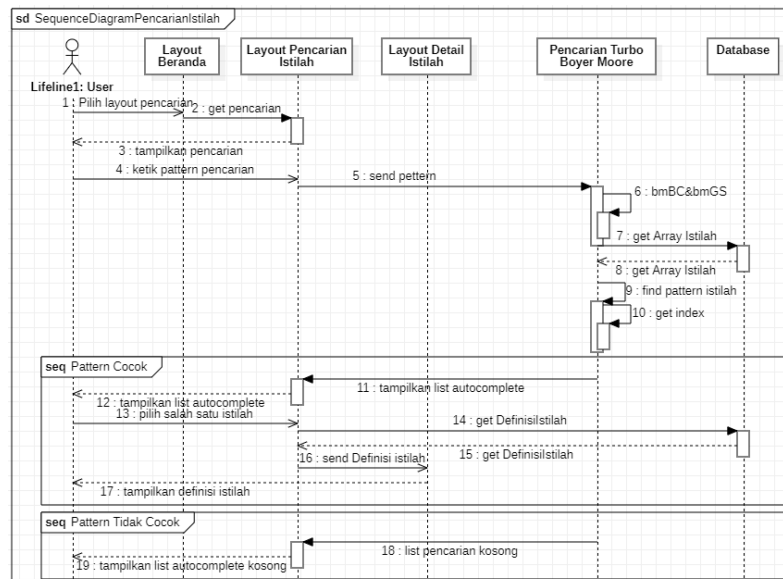


Gambar 2. Diagram Urutan Lihat Daftar Istilah

Dari diagram urutan lihat daftar istilah pada gambar 2 dapat dijelaskan sebagai berikut:

- 1) *User* akan mengakses aplikasi dan masuk pada menu utama atau beranda.
- 2) Sistem akan mengambil seluruh data yang ada pada *database*, kemudian menampilkan daftar istilah yang ada kepada *user*.
- 3) *User* melakukan klik pada istilah yang diinginkan, kemudian sistem akan mengambil definisi istilah dari *database* untuk ditampilkan kepada *user* dalam *layout* Detail Istilah.

- b. Diagram Urutan Pencarian Istilah

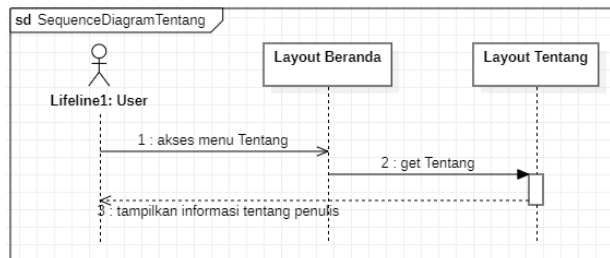


Gambar 3. Diagram Urutan Pencarian Istilah

Dari diagram urutan pencarian istilah pada gambar 3, dapat dijelaskan sebagai berikut:

- 1) *User* akan melakukan klik pada menu pencarian yang terdapat pada menu utama. Sistem akan beralih ke *layout* pencarian istilah dan secara otomatis mengaktifkan kotak pencarian.
- 2) Setelah kotak pencarian aktif, *user* akan memasukkan atau mengetikkan *pattern* yang ingin dicari. Sistem akan mengambil *pattern* yang dimasukkan dan akan melakukan pencocokkan dengan seluruh *string* yang ada pada *database* dengan algoritma Turbo Boyer Moore.
- 3) Jika terdapat kecocokan antara *pattern* dengan *database*, maka hasil pencarian dari *pattern* akan ditampilkan ke dalam sebuah *list* hasil. Kemudian pada *layout* Pencarian akan ditampilkan sejumlah *list* istilah yang sesuai dengan *pattern*. Selanjutnya pengguna dapat memilih istilah dari hasil pencarian yang ditampilkan. Kemudian sistem akan mengambil data dari *database* dan menampilkan istilah dan definisi ke dalam *layout* Detail Istilah.
- 4) Sedangkan jika *pattern* yang dimasukkan tidak cocok dengan data yang ada dalam *database*, maka sistem tidak akan menampilkan list hasil karena hasil tidak ditemukan.

c. Diagram Urutan Tentang

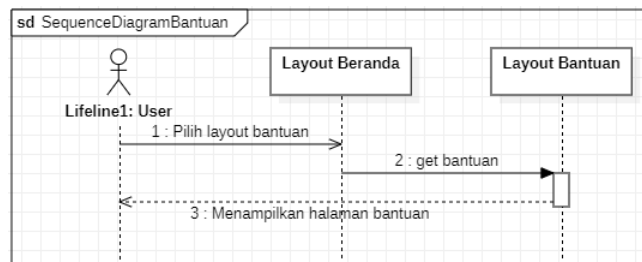


Gambar 4. Diagram Urutan Tentang

Diagram urutan tentang pada gambar 4, dapat dijelaskan sebagai berikut:

- 1) *User* memilih menu tentang dengan melakukan klik dari *layout* beranda, kemudian sistem akan merespon melalui sebuah *layout* tentang.
- 2) *Layout* tentang akan ditampilkan dan berisi informasi tentang penulis atau *developer*.

d. Diagram Urutan Bantuan



Gambar 5. Diagram Urutan Bantuan

Diagram urutan bantuan pada gambar 5, dapat dijelaskan sebagai berikut:

- 1) Pengguna memilih menu Bantuan pada layout Beranda, kemudian sistem akan merespon melalui sebuah layout Bantuan.
- 2) Halaman Bantuan akan ditampilkan, yang berisi petunjuk penggunaan aplikasi. Dengan adanya halaman Bantuan ini, pengguna akan merasa terbantu dalam memahami cara penggunaan aplikasi ini.

3.6 Perancangan Database

Perancangan *database* merupakan suatu proses pembuatan struktur *database* yang diperlukan untuk merancang sebuah aplikasi.

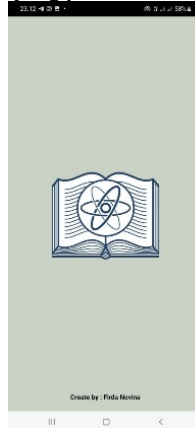
Tabel 5. Struktur Tabel Istilah

Nama Field	Tipe Data	Ukuran	Keterangan
Id_Istilah	Int	10	<i>Auto increment</i>
Istilah	Text	1000	
Definisi	Text	10000	

3.7 Tampilan Antarmuka Aplikasi Kamus Istilah Kimia

3.7.1 Tampilan Splash Screen

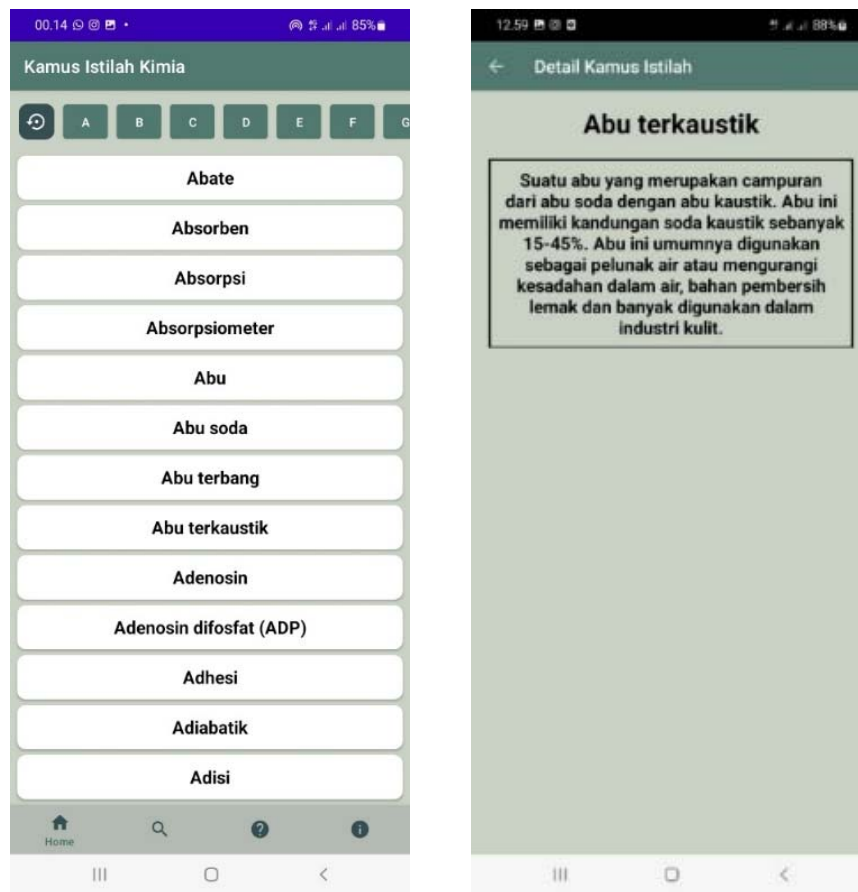
Tampilan *splash screen* adalah tampilan yang pertama kali tampil ketika *user* membuka sebuah aplikasi.



Gambar 6. Tampilan Splash Screen

3.7.2 Tampilan Halaman Utama (Daftar Istilah)

Tampilan halaman utama ini hanya berisi *listview*. Data yang ditampilkan dalam bentuk *listview* diambil dari *database* yang telah tersimpan dalam aplikasi kamus istilah kimia. Semua istilah yang terdapat pada *listview* dapat dipilih oleh *user* sesuai dengan kebutuhan setelah itu akan ditampilkan lagi sebuah halaman yang berisi istilah yang dipilih serta definisi dari istilah tersebut. Terdapat fitur sederet abjad yang dapat dipilih *user* sesuai dengan kebutuhan.



Gambar 7. Tampilan Halaman Utama dan Halaman Detail

3.7.3 Tampilan Halaman Pencarian

Tampilan halaman pencarian ini merupakan sebuah halaman yang digunakan *user* untuk melakukan pencarian istilah yang diinginkan. Pada halaman pencarian ini terdapat sebuah *edittext* yang berfungsi untuk melakukan pencarian. Selain terdapat *edittext* pada halaman pencarian ini juga terdapat *listview* yang memiliki fungsi untuk menampilkan hasil pencarian istilah berdasarkan *pattern* yang di-*input*-kan.



Gambar 8. Tampilan Halaman Pencarian

3.7.4 Tampilan Halaman Tentang

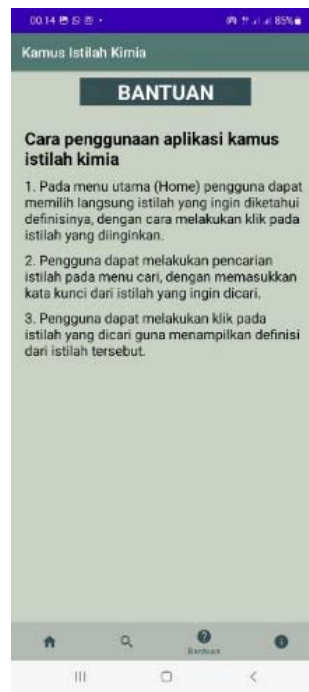
Halaman tentang merupakan sebuah halaman yang memiliki fungsi untuk menampilkan informasi tentang penulis atau *developer*.



Gambar 9. Tampilan Halaman Tentang

3.7.5 Tampilan Halaman Bantuan

Halaman bantuan merupakan sebuah halaman yang memiliki fungsi menampilkan informasi tentang tata cara penggunaan aplikasi kamus istilah kimia, agar pengguna dapat menggunakan aplikasi dengan baik.



Gambar 10. Tampilan Halaman Bantuan

3.8 Pengujian Aplikasi

Pengujian aplikasi merupakan suatu proses eksekusi program yang dilakukan untuk memastikan bahwa aplikasi berfungsi sebagaimana mestinya dan memenuhi standar yang telah ditetapkan. Selain itu, tujuan dari pengujian aplikasi ini adalah untuk mengetahui apakah algoritma yang diterapkan pada aplikasi kamus istilah kimia ini berjalan dengan baik ataupun tidak sesuai dengan rancangan. Pada pengujian ini, penulis menggunakan dua perangkat untuk melakukan pengujian aplikasi.

Samsung Galaxy Tab A8, dengan spesifikasi:

- a. Layar 10,5 inch
- b. Prosesor Octa-core 2GHz
- c. RAM 4 GB
- d. Memori Internal 128 GB
- e. Sistem Operasi Android 13

Vivo S1, dengan spesifikasi:




- a. Layar 6,38 inch
- b. Prosesor Octa-core 2.0 GHz
- c. RAM 4 GB
- d. Memori Internal 128 GB
- e. Sistem Operasi Android 11

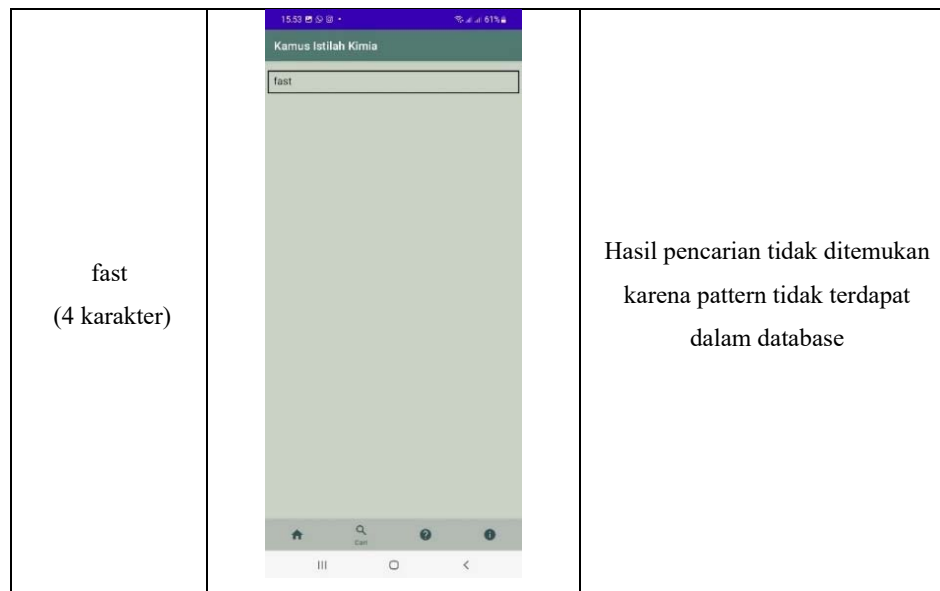
Dalam pengujian perangkat yang berbeda-beda akan dilihat apakah algoritma dan fitur yang diterapkan dalam aplikasi dapat berjalan sesuai dengan yang diharapkan. Penulis akan melakukan pengujian pencarian *pattern* dari beberapa perangkat yang berbeda.

Tabel 6. Pengujian Pencarian Patten

No	Jenis Smartphone	Pattern	Jumlah Istilah Ditemukan	Hasil
1	Samsung Galaxy Tab A8	Sin	25	Ditemukan
2		Bahan	19	Ditemukan
3		Efek	4	Ditemukan
4		Fi	35	Ditemukan
5		Alka	17	Ditemukan
6	Vivo S1	Sin	25	Ditemukan
7		Bahan	19	Ditemukan
8		Efek	4	Ditemukan
9		Fi	35	Ditemukan
10		Alka	17	Ditemukan

Tabel 7. Pengujian Hasil Pencarian

Pattern	Hasil Pencarian	Hasil
<p>f (1 karakter)</p>		<p>Pencarian bekerja dengan pattern minimal 1 karakter / huruf</p>
<p>zi (2 karakter)</p>		<p>Hasil pencarian ditemukan</p>
<p>ngan “ ” (4 karakter + 1 spasi)</p>		<p>Hasil pencarian ditemukan</p>



Dapat disimpulkan bahwa pada pengujian pencarian *pattern* pada perangkat yang berbeda memiliki hasil pencarian yang sama. Pengimplementasian fitur *autocomplete* dan algoritma Turbo Boyer Moore sesuai dengan perancangan yang telah dilakukan.

4. KESIMPULAN

Berdasarkan implementasi dan pengujian fitur *autocomplete* dan algoritma Turbo Boyer Moore pada aplikasi kamus istilah kimia yang telah dilakukan, penulis dapat menarik kesimpulan sebagai berikut:

- Algoritma Turbo Boyer Moore dapat diimplementasikan dalam perancangan fitur *autocomplete* yang berguna untuk menampilkan daftar istilah yang mungkin dimaksudkan oleh *user* dalam pencarian istilah pada aplikasi kamus istilah kimia berbasis Android.
- Dari hasil perbandingan pada beberapa *smartphone*, menunjukkan bahwa aplikasi kamus istilah kimia mendukung berbagai layar.
- Fitur *autocomplete* pada aplikasi istilah kimia ini dapat meminimalkan kesalahan *user* dalam mengetikkan istilah karena dapat menampilkan *suggestion* ketika *user* mengetikkan *pattern*. Pencarian akan ditampilkan apabila data yang dicari terdapat dalam *database*.
- Fitur *autocomplete* pada aplikasi kamus istilah kimia ini tidak akan menampilkan *suggestion* apabila *user* salah mengetikkan *pattern* karena kata tersebut dianggap tidak ditemukan dalam *database*, serta tidak ada rekomendasi kata yang dicari dan perbaikan kata.
- Jumlah maksimal istilah yang dapat ditampilkan dalam hasil pencarian adalah sebanyak jumlah hasil kecocokan antara *pattern* dengan istilah yang terdapat dalam *database*.

5. SARAN

Setelah melakukan implementasi fitur *autocomplete* dan algoritma Turbo Boyer Moore pada aplikasi kamus istilah kimia, penulis menyadari bahwa aplikasi yang telah dihasilkan belum sempurna. Oleh karena itu, penulis memberikan saran agar aplikasi dapat dikembangkan lebih lanjut pada penelitian selanjutnya, yaitu sebagai berikut:

- Untuk pengembangan selanjutnya, diharapkan dapat menerapkan dua fitur sekaligus ke dalam aplikasi, yaitu fitur *autocomplete* (rekomendasi kata) dan *autocorrect* (perbaikan kata).
- Untuk pengembangan selanjutnya, diharapkan dapat melakukan pencarian *string* dalam definisi istilah.

UCAPAN TERIMA KASIH

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena berkat dan karunia-Nya penulis dapat menyelesaikan penulisan penelitian ini dengan baik. Pada kesempatan ini penulis mengucapkan terima kasih kepada Civitas Akademika Fakultas Teknologi Informasi Universitas Widya Dharma Pontianak atas segala dukungan yang diberikan terhadap penelitian ini, kepada rekan-rekan yang telah memberikan bantuan dan saran kepada penulis dalam menyelesaikan penelitian ini, kepada keluarga yang senantiasa selalu mendukung dan kepada teman-teman semua atas semangat dan motivasi yang telah diberikan.

DAFTAR PUSTAKA

- [1] Setyawan, Muhammad Yusril Helmi dan Dinda Ayu Pratiwi. (2020). *Membuat Sistem Informasi Gadai Online Menggunakan Codeigniter Serta Kelola Proses Pemberitahuannya*. Kreatif Industri Nusantara. Bandung.
- [2] Fathoroni, Anisa, Rd. Nuraini Siti Fatonah, Roni Andarsyah dan Noviana Riza. (2020). *Buku Tutorial Sistem Pendukung Keputusan Penilaian Kinerja Dosen Menggunakan Metode 360 Degree Feedback*. CV. Kreatif Industri Nusantara. Bandung.
- [3] Hadiprakoso, Raden Budiarto. (2020). *Rekayasa Perangkat Lunak*. Andi. Yogyakarta.
- [4] Pane, Syafrial Fachri, Mochamad Zamzam, dan Muhamad Diar Fadillah. (2020). *Membangun Aplikasi Peminjaman Jurnal Menggunakan Aplikasi Oracle Apex Online*. Kreatif. Bandung.
- [5] Yusup, Pawit M. (2022). *Ilmu Informasi, Komunikasi, dan Kepustakaan*. Bumi Aksara. Jakarta.
- [6] Eliyarti dan Chichi Rahayu. (Oktober 2019). *Tinjauan Motivasi Berprestasi Mahasiswa Teknik Dalam Perkuliahan Kimia Dasar*. Jurnal Pendidikan Glasser. Vol. 3, no. 2: hal. 198.
- [7] Wicaksono, Yudhy dan Solusi Kantor. (2021). *Aplikasi Kuitansi, Stok Barang, & Penggajian dengan MS Excel*. PT Elex Media Komputindo. Jakarta.
- [8] Irawan, Yosef Ariyanto, Billyanto Hendrik, Antonius, Taufan Abdurrachman, Arie Tunggal dan Bernard Renaldy Suteja. (Agustus 2019). *Pendekatan MapReduce untuk Implementasi Suffix Tree pada AutoComplete Produk dengan Metodologi Agile*. Jurnal Teknik Informatika dan Sistem Informasi. Vol. 5, no. 2: hal. 222.
- [9] Neory, meijini Lucy. (Desember 2020). *Implementasi Algoritma Turbo Boyer Moore Pada Aplikasi Pengobatan dengan Rempah-Rempahan Berbasis Android*. Kajian Ilmiah Informatika dan Komputer. Vol.1, no. 3: hal. 94.
- [10] Rumahordo, Andreas, Pandi Barita Nauli Simangunsong, Syarifah Fadillah Rezky, Richard Parlindungan Simanjuntak, Andy Paul Harianja dan Pristiwanto. (2022). *Rekayasa Perangkat Lunak Aplikasi Donor Plasma Konvalen*. Cattleya Darmaya Fortuna. Deli Serdang.
- [11] Destriana, Rachmat, Syepri Maulana Husein, Nurdiana Handayani dan Aditya Tegar Prahara Siswanto. (2021). *Diagram UML Dalam Membuat Aplikasi Android Firebase “Studi Kasus Aplikasi Bank Sampah”*. Deepublish. Yogyakarta.
- [12] Setiyani, Lila. (2019). *“Pengujian Sistem Informasi Inventory Pada Perusahaan Distributor Farmasi Menggunakan Metode Black Box Testing”*. Techno Xplore (Jurnal Ilmu Komputer dan Teknologi Informasi). Vol. 4, no: 1: hal: 21.