

IMPLEMENTASI ALGORITMA PENCOCOKAN *STRING KNUTH- MORRIS-PRATT* DALAM APLIKASI PENCARIAN DOKUMEN *DIGITAL* BERBASIS ANDROID

Maya Rossaria¹, Boko Susilo², Ernawati³

^{1,3}Program Studi Teknik Informatika, Fakultas Teknik, Universitas Bengkulu.
Jl. WR. Supratman Kandang Limun Bengkulu 38371A INDONESIA
(tel: 0736-341022; fax: 0736-341022)

¹mayarossaria59@gmail.com

²bksusilo@gmail.com

³w_ier_na@yahoo.com

Abstrak: Pada penelitian ini dibangun sebuah aplikasi yang bertujuan untuk mencari dokumen yang berasal dari Android dan mengaplikasikannya dengan menggunakan algoritma pencocokan *string* sebagai salah satu cara untuk menemukan dokumen digital yang terdapat pada Android. aplikasi ini dibangun berbasis Android dengan menggunakan algoritma Pencocokan *string Knuth-Morris-Pratt* sebagai algoritma dalam aplikasi pencarian dokumen, dan dibangun dengan menggunakan bahasa pemrograman JAVA dengan IDE ECLIPSE JUNO, analisis perancangan sistem ini menggunakan *Unified Modeling Language (UML)*. Dapat disimpulkan bahwa aplikasi ini dapat melakukan pencarian dokumen *digital* yang terdapat dalam Android dengan menggunakan algoritma *Knuth-Morris-Pratt*. Hasil pencarian yang ditampilkan berupa dokumen-dokumen yang tersedia dalam Android dan informasi mengenai jumlah dari dokumen yang tersedia dalam Android tersebut, serta menunjukkan bahwa algoritma *Knuth-Morris-Pratt* bisa digunakan dalam aplikasi pencarian dokumen pada Android.

Kata Kunci: Android, Dokumen *Digital*, Algoritma pencocokan *string Knuth-Morris-Pratt*, Eclipse IDE

Abstract : In this research, an application is developed as a mean to search documents from Android and to apply it by using String Matching Algorithm as one of the ways to find a digital document on Android. This application is built on Android-based by using Knuth-Morris-Pratt string matching algorithm as one in the document retrieval application and applied with JAVA programming language with Eclipse JUNO IDE. This system design analysis uses Unified Modeling Language (UML). It can be concluded that this application is able to run digital document retrieval on Android by using Knuth-Morris-Pratt algorithm. The displayed result of retrieval is of the available document on the Android and give an information about how many document on Android, and also to prove that Knuth-Morris-Pratt string matching

algorithms can be used in document retrieval application on Android.

Keywords: *Android, Digital Document, Knuth-Morris-Pratt String Matching Algorithms, Eclipse IDE*

I. PENDAHULUAN

Sistem operasi Android juga mempunyai beberapa *tools* atau alat tambahan untuk membaca dan menemukan dokumen *digital* yang digunakan, salah satunya adalah dengan penggunaan *file manager*. Tetapi alat tambahan tersebut sering terjadi kesalahan dalam pengoperasiannya. Permasalahan yang paling sering muncul adalah tidak terbacanya dokumen *digital* tersebut di dalam *smartphone*, sedangkan jika *smartphone* dihubungkan dengan komputer, dokumen di dalam *smartphone* terbaca lengkap.

Pada proses pencarian dokumen *digital* dalam sistem operasi Android terbilang sangat sulit, dikarenakan banyaknya *file* dan dokumen di dalam Sistem Operasi Android, sehingga pencarian tersebut menjadi cukup lama jika ingin mencari suatu dokumen. Sistem operasi Android mempunyai sistem kerja mencari satu per satu di dalam sebuah *folder* dalam sistem operasi Android dan menampilkannya. Selain itu, hasil pencarian yang didapat dari pencarian folder satu per satu tidak tepat, karena semua dokumen yang mempunyai unsur nama sama dengan dokumen tersebut, maka akan muncul di pencarian dokumen dan perlu beberapa tambahan waktu untuk menyaring ulang dokumen yang akan dicari.

Dalam pencarian dokumen *digital* dibutuhkan alat tambahan yang dapat membantu proses pencarian dokumen yang lebih tepat. Untuk pencarian dokumen *digital*, dibutuhkan algoritma

pencocokan string yang akan membantu proses pencarian dokumen. Algoritma *Knuth-Morris-Pratt* merupakan suatu algoritma pencarian *String* untuk mencari teks berdasarkan urutan dari kiri ke kanan. Algoritma *Knuth-Morris-Pratt* akan mencocokkan *pattern* atau susunan kata yang akan dicari dari kiri ke kanan pada awal teks dan kemudian menggeser susunan kata sampai susunan kata tersebut berada di ujung teks.

Menurut Sara Basee dalam bukunya yang berjudul "*Computer Algorithm Third Edition : Solution to Selected Exercise*", penjelasan singkat mengenai algoritma *Knuth-Morris-Pratt* yaitu Algoritma *Knuth-Morris-Pratt* dilakukan dengan cara menghitung fungsi pinggiran dari pola terlebih dulu dan kemudian akan dilakukan perbandingan antara pola dan elemen pertama dari kalimat, jika tidak sesuai, maka perbandingan tidak dilakukan pada elemen kedua, namun tergantung dari nilai yang akan dikeluarkan oleh fungsi pinggiran tersebut

Berdasarkan uraian di atas, maka penulis tertarik untuk membuat suatu rancang bangun aplikasi pencarian dokumen *digital* dengan menggunakan algoritma *Knuth-Morris-Pratt* dengan filterisasi untuk pencocokan *string* dari dokumen tersebut. Dari penjabaran di atas, penulis mengangkat judul "*Implementasi Algoritma Pencocokan String Knuth-Morris-Pratt Dalam Aplikasi Pencarian Dokumen Digital Berbasis Android*".

II. LANDASAN TEORI

A. Dokumen Digital

Dokumen digital merupakan setiap informasi elektronik yang dibuat, diteruskan, dikirimkan,

diterima, atau disimpan dalam bentuk *analog*, *digital*, elektromagnetik, optikal, atau sejenisnya, yang dapat dilihat, ditampilkan dan/atau didengar melalui komputer atau sistem elektronik[1].

Ada beberapa keunggulan dari dokumen *digital*, yaitu sebagai berikut :

- a. Dokumen *digital* dapat dikirim dan cepat untuk dipindahkan
- b. Tidak menentukan media cetak berupa kertas atau media cetak lainnya
- c. Dapat dipindahkan ke media cetak sewaktu-waktu
- d. Dapat lebih mudah di indeks, berbeda dengan dokumen yang menggunakan media cetak, beberapa dokumen *digital* dapat di indeks menjadi satu untuk memudahkan pencocokan dokumen
- e. Dapat digunakan di berbagai macam *platform*.

Dengan adanya dokumen dalam bentuk *digital*, maka memungkinkan untuk mendistribusikan sebuah dokumen melalui media web. Dokumen *digital* bentuknya dapat bermacam-macam dan setiap *platform* memiliki jenis dokumen yang berbeda-beda. Namun, untuk dapat mendistribusikan melalui media web ada beberapa format standar yang digunakan, beberapa diantaranya yang sering digunakan adalah *Portable Document Format* (PDF), *Hypertext Markup Language* (HTML), *eXtensible Markup Language* (XML), *eXtensible Hypertext Markup Language* (XHTML) dan lain sebagainya[1].

B. Algoritma Pencocokan String

Algoritma *string matching* adalah sebuah algoritma yang digunakan dalam pencocokkan suatu pola kata tertentu terhadap suatu kalimat atau

teks panjang. Algoritma *string matching* sendiri dapat dilakukan dengan beberapa cara tertentu, antara lain cara *Brute Force* dan cara *Knuth-Morris-Pratt* (KMP) [2].

C. Algoritma Knuth-Morris-Pratt

Algoritma *Knuth-Morris-Pratt* dikembangkan oleh D. E. Knuth, bersama dengan J. H. Morris dan V. R. Pratt. Algoritma *Knuth-Morris-Pratt* merupakan pengembangan dari algoritma pencarian *string* sebelumnya, yaitu algoritma *Brute Force*. Algoritma *Brute-Force* merupakan algoritma dasar yang paling sederhana dalam menyelesaikan persoalan pencocokan *string* yang melakukan pencarian pada setiap posisi di dalam teks antara 0 dan n-m, dimana n adalah panjang teks/banyaknya nama *file* yang tersimpan di komputer dan m adalah panjang karakter dari suatu *pattern* (kata yang akan dicari)[2].

Secara sistematis, langkah-langkah yang dilakukan algoritma *Knuth-Morris-Pratt* pada saat mencocokkan *string*[2] :

1. Masukkan *Query* kata yang akan dicari.
Dengan permisalan
P=*Pattern* atau pola susunan kata yang dijadikan sebagai contoh atau pola teks yang akan dicari
T=Teks atau judul dokumen
2. Algoritma Knuth-Morris-Pratt mulai mencocokkan *pattern* atau pola susunan kata yang dijadikan sebagai contoh pada awal teks.
3. Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per karakter *pattern* atau pola susunan kata yang dijadikan sebagai contoh dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi :

- a. Karakter di *pattern* atau pola susunan kata yang dijadikan sebagai contoh dan di teks yang dibandingkan tidak cocok (*mismatch*).
 - b. Semua karakter di *pattern* atau pola susunan kata yang dijadikan sebagai contoh cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.
4. Algoritma kemudian menggeser *pattern* atau pola susunan kata yang dijadikan sebagai contoh berdasarkan tabel next, lalu mengulangi langkah no. 2 sampai *pattern* atau pola susunan kata yang dijadikan sebagai contoh berada di ujung teks.

D. Sistem Operasi Android

Sistem Operasi Android adalah sistem operasi berbasis *Linux* yang dipergunakan sebagai pengelola sumber daya perangkat keras, baik untuk ponsel, *smartphone* dan juga PC tablet. Secara umum Sistem Operasi Android adalah platform yang terbuka (*Open Source*) bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh berbagai piranti bergerak [3].

Android versi 4.0

Semenjak kehadirannya pada 9 Maret 2009, sistem operasi Android telah hadir dengan versi 1.1, yaitu sistem operasi yang sudah dilengkapi dengan pembaruan estetis pada aplikasinya, seperti *voice search* atau pencarian dengan menggunakan fitur suara, pengiriman pesan dengan Gmail, dan pemberitahuan email [4].

Perangkat Lunak Yang Mendukung

1. *Java J2ME*

Java2 Micro Edition atau yang biasa disebut J2ME adalah lingkungan pengembangan yang

didesain untuk meletakkan perangkat lunak Java pada barangelektronik beserta perangkat pendukungnya. Pada J2ME, jika perangkat lunak berfungsi baik pada sebuah perangkat maka belum tentu juga berfungsi baik pada perangkat yang lainnya. J2ME membawa Java ke dunia informasi, komunikasi, dan perangkat komputasi selain perangkat komputer desktop yang biasanya lebih kecil dibandingkan perangkat komputer desktop. J2ME biasa digunakan pada telepon selular, pager, *Personal Digital Assistants* (PDA) dan sejenisnya[3].

2. *Android SDK*

Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Saat ini disediakan Android SDK sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java[3].

3. *Eclipse IDE*

Eclipse adalah sebuah pengembangan lingkungan yang teintegrasi yang dapat membantu untuk membangun aplikasi-aplikasi dalam banyak bahasa.

4. *Android Development Tools*

ADT adalah salah satu alat yang di desain untuk pengembangan aplikasi Android. ADT memungkinkan Eclipse untuk digunakan dalam membuat aplikasi Android baru, membuat *User Interface*, menambahkan komponen berdasarkan tipe Android yang digunakan, *debug* aplikasi, dan pemaketan aplikasi.

E. *Unified Modeling Language (UML)*

Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Pada perkembangan teknik pemrograman berorientasi objek munculah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang akan dibangun, yaitu *Unified Modeling language* (UML), yang dibangun karena adanya kebutuhan pemodelan visual untuk memesifikasi, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak.

UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek[5].UML dikelompokkan menjadi 3 bagian , yaitu :

1. *Structure Diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior Diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction Diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antarsubsystem pada suatu sistem.

III. METODOLOGI PENELITIAN

A. Teknik Pengumpulan Data

Teknik yang digunakan dalam pengumpulan data adalah angket atau kuisisioner dan dokumen. Berikut ini merupakan penjelasan mengenai teknik yang digunakan dalam penelitian ini.

1. Angket atau Kuisisioner

Angket atau Kuesioner adalah teknik pengumpulan data dengan menyerahkan atau mengirimkan daftar pertanyaan untuk diisi sendiri oleh responden. Dalam penelitian survei pemakaian kuisisioner merupakan hal penting untuk mengumpulkan data. Kuisisioner yang digunakan adalah kuisisioner berbentuk *check list* atau sebuah daftar dimana responden atau pengguna Android membubuhkan tanda *check* atau tanda (\surd) pada kolom yang sesuai dan kuisisioner skala bertingkat, yaitu sebuah pernyataan diikuti oleh kolom-kolom yang menunjukkan tingkatan, misalnya mulai dari sangat baik sampai ke tidak baik. Responden atau pengguna Android dapat memberikan jawaban sangat baik atau tidak baik sesuai dengan kinerja dari aplikasi pencarian dokumen *digital* tersebut. angket atau kuisisioner dapat dilihat pada Lampiran C.

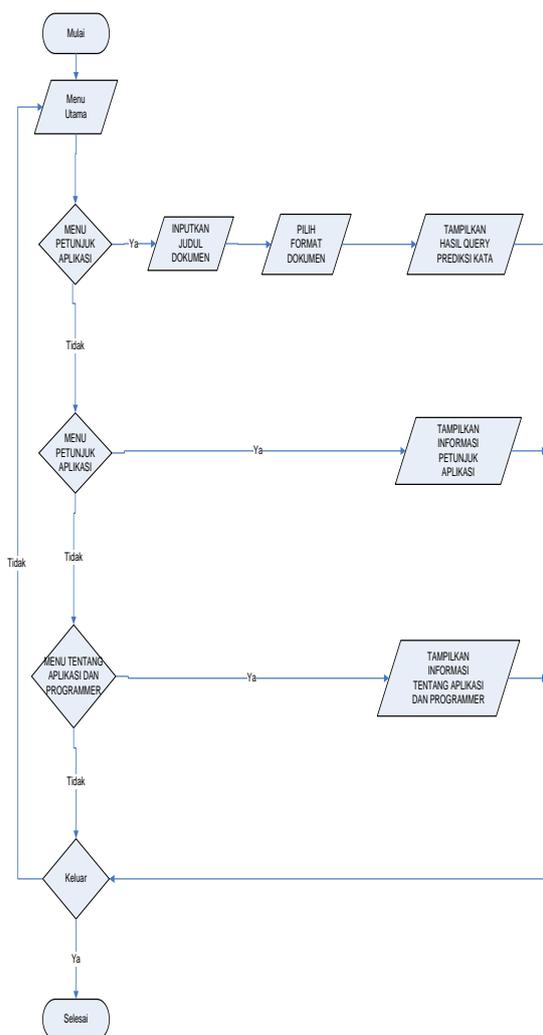
2. Dokumentasi.

Dokumentasi dilakukan dengan cara mencari data atau dokumen *digital* dari Android yang berasal (yang di-unggah) dari internet seperti *e-Mail* yang berbentuk dokumen dengan format *.docx* , *.doc* , *.pdf* , *.xls* , *.xlsx* , *.txt* , *.ppt* , dan *.pptx*,

IV. ANALISIS DAN PERANCANGAN

A. Cara Kerja sistem

Secara garis besar cara kerja sistem yang dibangun ditampilkan dalam Gambar 1 berikut.



Gambar 1. Diagram Alir Antarmuka Pengguna dengan Sistem

Keterangan Diagram Alir

1. Aplikasi dimulai dengan cara pengguna terlebih dahulu memasukkan teks judul yang akan dicari. Lalu terdapat menu pilihan petunjuk aplikasi yang menampilkan tentang tata cara penggunaan aplikasi dan menu pilihan tentang aplikasi dan programmer yang menampilkan informasi mengenai *programmer* atau pembuat aplikasi.
2. Setelah pengguna memasukkan teks judul yang akan dicari, pengguna juga akan memilih format dokumen *digital* yang akan dicari.

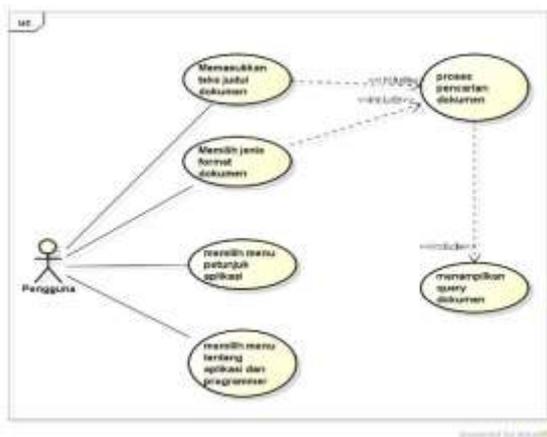
Pilihannya yaitu “Semua Dokumen”, yaitu menampilkan semua format, “.docx” dan “.doc” untuk pilihan *microsoft word*, “.xlsx” dan “.xls” untuk pilihan *microsoft excel*, “.ppt” dan “.pptx” untuk pilihan *microsoft power point*, “.pdf” untuk pilihan *adobe reader* dan “.txt” untuk ekstensi dari file *Text Document*.

3. Setelah semua data yang diperlukan telah dimasukkan maka sistem akan melakukan pencarian dan akan menampilkan hasil pencarian secara bertahap sampai selesai.
4. Pada menu “Petunjuk Aplikasi”, akan ditampilkan beberapa informasi mengenai tata cara penggunaan aplikasi.
5. Apabila pengguna ingin membuka menu “Tentang Aplikasi dan Programmer” maka sistem akan menampilkan informasi tentang pembuat atau *programmer* dari aplikasi yang digunakan. Jika tidak maka kerja sistem selesai.
6. Apabila pengguna tidak ingin membuka menu aplikasi dan tidak memilih menu, maka aplikasi akan selesai.

B. Perancangan Model UML (*Unified Modeling Language*)

Perancangan aplikasi yang mengimplementasikan algoritma *Knuth-Morris-Pratt* pada aplikasi pencarian dokumen *digital* berbasis Android menggunakan UML (*Unified Modeling Language*), seperti berikut :

1. *Use Case Diagram*



Gambar 2. Usecase Diagram

Pada aplikasi pencarian dokumen *digital* yang akan dibangun, terdapat interaksi antara pengguna dan sistem. Pada aplikasi ini, pencarian dokumen *digital* membutuhkan satu pengguna untuk menjalankan aplikasi tersebut.

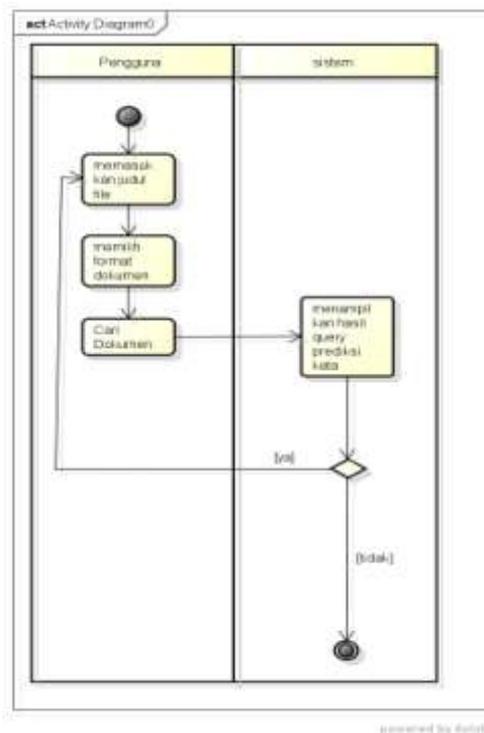
Tabel 1 Interaksi Pengguna dan interaksi sistem.

Aktor	Interaksi
Pengguna	1. Pengguna memasukkan nama file yang akan dicari.
	2. Pengguna memilih format dokumen yang akan dicari
	tentang aplikasi dan programmer, berisi tentang informasi mengenai pembuat aplikasi pencarian dokumen <i>digital</i> .
	petunjuk aplikasi, berisi tentang tata cara penggunaan aplikasi.

2. Activity Diagram

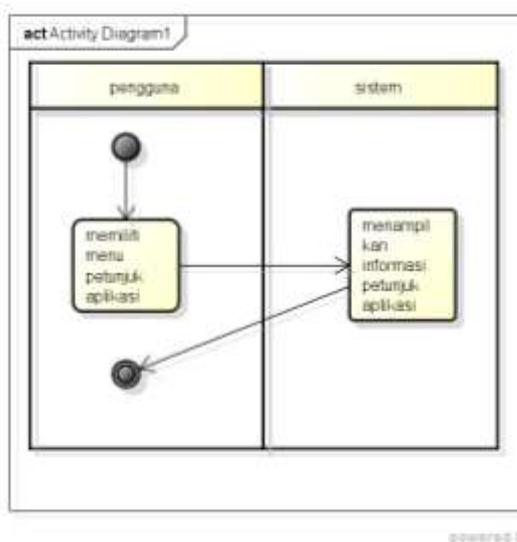
Pada UML (*Unified Modeling Language*), *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses. *Activity Diagram* pada sistem ini terbagi menjadi 3 diagram, yaitu:

a. *Activity Diagram* Pencarian Dokumen
Activity Diagram pada pilihan menu “Pencarian Dokumen” ditunjukkan dalam Gambar 3 berikut.



Gambar 3. Activity Diagram Pencarian Dokumen
 Pada Gambar 3 menunjukkan keseluruhan dari kerja sistem. Pada aplikasi, pencarian diawali dari pengguna terlebih dahulu memasukkan judul teks yang akan dicari, memilih format dokumen dan tempat penyimpanan dari dokumen tersebut. Setelah semua data telah dimasukkan, maka pengguna akan menunggu hasil prediksi kata dari judul teks yang telah dimasukkan sebelumnya. Setelah hasil prediksi kata ditampilkan, maka proses pencarian selesai dilakukan.

b. *Activity Diagram* Petunjuk Aplikasi
Activity Diagram pada pilihan menu “Petunjuk Aplikasi” ditunjukkan dalam Gambar 4 berikut.

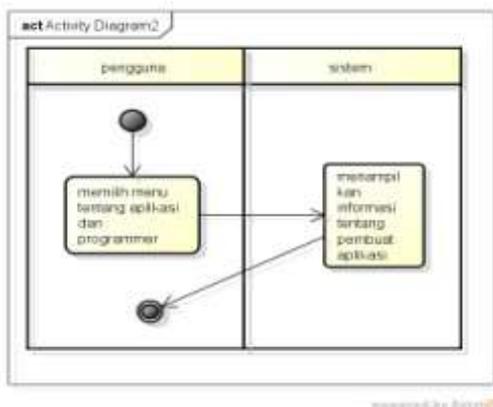


Gambar 4. Activity Diagram Petunjuk Aplikasi

Pada Gambar 4 merupakan Activity Diagram pada menu Petunjuk Aplikasi. Diawali dari pengguna memilih menu pilihan “petunjuk aplikasi” dimana pengguna dapat mengetahui tata cara atau penggunaan aplikasi tersebut. Setelah informasi petunjuk aplikasi ditampilkan, maka proses selesai.

c. Activity Diagram Tentang Aplikasi Dan Programmer

Activity Diagram pada pilihan menu “Tentang Aplikasi dan Programmer” ditunjukkan dalam Gambar 5 berikut.

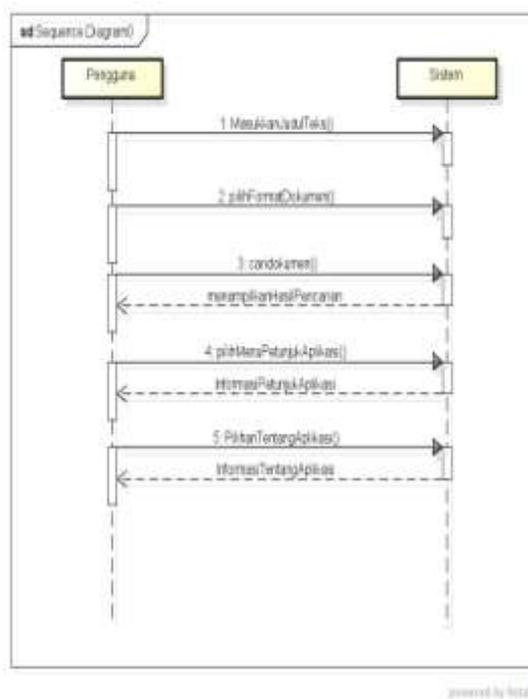


Gambar 5. Activity Diagram Tentang aplikasi dan Programmer

Pada Gambar 5 merupakan Activity Diagram pada menu Tentang Aplikasi dan Programmer. Diawali dari pengguna memilih menu pilihan “tentang Aplikasi dan programmer” dimana pengguna mendapat informasi mengenai pembuat atau programmer aplikasi tersebut. Setelah informasi mengenai tentang aplikasi ditampilkan, maka proses selesai.

3. Sequence Diagram

Berikut merupakan gambaran Sequence Diagram dari aplikasi pencarian dokumen digital berbasis Android yang dijabarkan dalam Gambar 6 berikut.



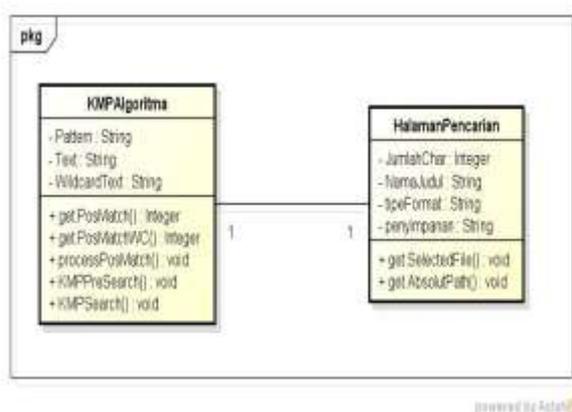
Gambar 6. Sequence Diagram Pencarian Dokumen

Pada diagram diatas, adalah interaksi antara pengguna dengan sistem. Pengguna terlebih dahulu memasukkan judul teks dari dokumen yang dicari, memilih tempat penyimpanan dokumen dan tipe dokumen yang akan dicari. Sistem akan menampilkan hasil pencarian dokumen yang

sebelumnya telah di masukkan oleh pengguna berupa tampilan prediksi dokumen.

4. Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Pada sistem ini disetiap menu yang ditampilkan memiliki operasi dan atribut-atribut yang mengikuti. Adapun class diagram untuk aplikasi pencarian dokumen digital dijabarkan dalam Gambar 7 berikut.



Gambar 7. Class Diagram Pencarian Dokumen

Pada Gambar 7, komponen pada class diagram terdiri dari Class, attributes dan Operations. Class pada Gambar 7 adalah KMPAlgoritma dan halamanPencarian. Class pada KMPAlgoritma menampilkan beberapa atribut dan method didalamnya. Atribut pada class KMPAlgoritma adalah Pattern, Text, dan wildcard dengan tipe data String dikarenakan pada teks, pattern dan wildcard bertipe karakter dan tidak angka. Method pada class KMPAlgoritma adalah GetPostMatch, GetPostMatchWC, ProcessOnMatch, KMPpreSearch dan KMPSearch.

Pada Class HalamanPencarian mempunyai atribut dan method pada prosesnya. Atribut pada class halamanPencarian adalah JumlahChar yang mempunyai tipe data Integer, NamaJudul, tipeFormat dengan tipe data String.

Method pada Class HalamanPencarian adalah GetSelectedFile dan GetAbsolutPath.

V. PEMBAHASAN

A. Implementasi Antarmuka

Pada tahapan implementasi antarmuka ini, sistem akan diimplementasikan menggunakan bahasa pemrograman Java J2ME dengan menggunakan Eclipse IDE. Berikut tampilan hasil implementasi sistem :

1. Halaman Depan Aplikasi

Halaman depan aplikasi ini merupakan halaman aplikasi yang akan tampil pertama kali pada saat aplikasi dijalankan ketika memilih ikon aplikasi. Pada halaman depan aplikasi ini menampilkan nama dari aplikasi, logo folder dan progress bar.



Gambar 8. Tampilan Halaman Depan Aplikasi di Smartphone

2. Halaman Menu Utama Aplikasi

Setelah tampilan halaman depan selesai, maka tampilan selanjutnya adalah halaman menu utama aplikasi. Pada halaman menu utama ini terdapat empat pilihan menu, yaitu menu pencarian dokumen, menu pencarian keseluruhan dokumen, menu petunjuk aplikasi dan menu tentang aplikasi dan programmer.

Dan tampilan menu utama ketika di jalankan di dalam *smartphone* adalah sebagai berikut



Gambar 9. Tampilan Menu Utama Di *Smartphone*

3. Halaman Menu Pencarian Dokumen

Halaman pencarian dokumen merupakan pencarian utama yang terdapat dalam aplikasi ini. Menu Pencarian Dokumen merupakan menu pencarian yang menggunakan Algoritma *Knuth-Morris-Pratt* dan menggunakan beberapa pilihan untuk pencarian dokumen ini, seperti pilihan dokumen yang akan dicari dengan beberapa pilihan ekstensi dari dokumen.



Gambar 10. Hasil pencarian Dokumen

Pada Gambar 10 terdapat tempat untuk pengisian judul teks yang akan dicari, dengan menggunakan *edittext* dalam eclipse. Lalu ada beberapa pilihan dalam mencari dokumen, dengan menggunakan ekstensi dari dokumen. Pada pilihan pencarian dengan tipe dokumen, terdapat pilihan semua dokumen, yang mempunyai arti bahwa semua dokumen akan terbaca, dan jika memilih salah satu dari beberapa ekstensi tersebut, maka hanya dokumen dengan ekstensi itu saja yang terbaca. Jika semua pilihan telah dipilih dan di isi maka aplikasi akan mencari kecocokan judul yang sama dan menampilkannya dalam sebuah daftar atau *list* dan menampilkan juga informasi mengenai jumlah dokumen yang terbaca.

4. Halaman Petunjuk Aplikasi

Halaman petunjuk aplikasi merupakan menu yang terdapat dalam aplikasi pencarian dokumen ini. Halaman petunjuk aplikasi ini berisikan tata cara penggunaan dari aplikasi pencarian dokumen tersebut. Halaman petunjuk aplikasi ini dapat membantu pengguna dalam memahami dalam menggunakan menu-menu yang terdapat di aplikasi ini.

Dan tampilan menu utama ketika di jalankan di dalam *smartphone* adalah sebagai berikut



Gambar 11. Tampilan Halaman Petunjuk Aplikasi di Smartphone

5. Halaman Tentang Aplikasi dan Programmer

Halaman tentang aplikasi dan programmer merupakan halaman aplikasi yang berisikan informasi mengenai aplikasi pencarian dokumen dan pembuat aplikasi tersebut.

Dan tampilan menu utama ketika di jalankan di dalam smartphone adalah sebagai berikut.



Gambar 12. Tampilan Halaman Tentang Aplikasi dan Programmer di Smartphone

B. Pengujian Sistem Secara Manual

Pada sebuah pencarian dokumen, pengguna android ingin mencari dokumen yang akan ia baca.

Pengguna mencari judul yang akan dicari adalah TULISAN, dan *pattern* yang akan dicari adalah LISAN. Pengguna ingin mencari dokumen secara tepat, dan memilih algoritma untuk pencarian dari teks tersebut.

Penyelesaian dari contoh kasus di atas jika menggunakan algoritma *Knuth-Morris-Pratt* adalah sebagai berikut:

1. Terlebih dahulu menentukan *pattern* dan teks yang akan dicari. Dalam kasus ini, *pattern* dan teks yang akan dicari adalah sebagai berikut :

P = LISAN

T = TULISAN

2. Selanjutnya menentukan fungsi pinggiran dari *pattern* dan teks yang akan dicari. Fungsi pinggiran didefinisikan sebagai ukuran awalan terpanjang dari *pattern*. Fungsi pinggiran tersebut akan dijabarkan dalam Tabel 2 berikut.

Tabel 2 Fungsi Pinggiran kasus pencarian string

j	0	1	2	3	4	5	6
p(j)	T	U	L	I	S	A	N
b(j)	0	1	3	0	0	0	0

3. Lakukan peng-*index*-an atau memberikan nilai indeks ke *pattern* dan teks. Adapun nilai indeks *pattern* dan teks akan dijabarkan dalam tabel 3 dan tabel 4 berikut.

Tabel 3 Nilai Index Untuk teks

INDEKS	0	1	2	3	4	5	6
t(j)	T	U	L	I	S	A	N

Tabel 4 Nilai Index Untuk Pattern

INDEKS	0	1	2	3	4
P(j)	L	I	S	A	N

4. Cara perhitungan pergeseran dari algoritma *Knuth-Morris-Pratt* adalah:

- a. Bandingkan ujung kiri teks dan ujung kiri pada *Pattern*.
Pada Ujung kiri teks dan ujung kiri *pattern* terjadi ketidakcocokan, tetapi, pada nilai indeks ke 3, terjadi kecocokan. Teks dengan karakter indeks pertama yaitu T sedangkan *pattern* dengan karakter indeks ketiga yaitu L.
- b. Karena terjadi ketidakcocokan, maka lakukan pergeseran *pattern* P dengan jumlah pergeseran sesuai dengan nilai pinggiran *pattern* P yang cocok. Pada kasus ini, karakter pada *pattern* dan teks yang menemukan kesamaan ada di indeks 2-6, maka panjang kesesuaian teks adalah 5 ($l=5$).
- c. Nilai pinggiran terpanjang dari *pattern* P yang menemukan kecocokan adalah P[2...6], dimana dalam fungsi pinggiran sebelumnya, nilai *pattern* pada indeks urutan 3 dalam fungsi pinggiran adalah 3 ($b(3)=3$).
- d. Setelah ditentukan nilai fungsi pinggiran dan panjang dari kecocokan teks dan *pattern*, maka pergeseran karakter dilakukan dengan cara :

$$\text{Nilai Pergeseran} = l - b \dots\dots\dots(5.1)$$

Keterangan :

Nilai pergeseran : Nilai yang akan digeser tiap karakternya

l = Panjang kecocokan karakter antara *pattern* dan teks

b = Nilai dalam fungsi pinggiran

Maka, nilai pergeseran yang dilakukan adalah

$$\begin{aligned} \text{Nilai Pergeseran} &= l - b \\ \text{Nilai pergeseran} &= 5 - 3 = 2 \end{aligned}$$

KARAKTER

Jadi, *pattern* P digeser sebanyak 2 karakter ke kanan. Pergeseran tersebut akan dijabarkan dalam Tabel 5berikut.

Tabel 5 Hasil pergeseran Karakter teks Dan *Pattern*

Indeks Teks T	0	1	2	3	4	5	6
Teks	T	U	L	I	S	A	N
Indeks Pattern P	PERGESERAN		0	1	2	3	4
Pattern P			L	I	S	A	N

VI. KESIMPULAN

Berdasarkan analisa perancangan sistem, implemantasi, dan pengujian sistem, maka dapat disimpulkan bahwa penelitian ini telah berhasil dibangun,yang dapat melakukan proses pencarian dokumen dengan menggunakan pergeseran dari Fungsi Pinggiran pada algoritma *Knuth-Morris-Pratt* dan pencarian yang dapat dilakukan dengan menggunakan fungsi *Wildcard* atau karakter string yang berfungsi menggantikan peran karakter string alphabet 26 huruf. Aplikasi pencarian dokumen digital berbasis Android dengan menggunakan algoritma *Knuth-Morris-Pratt* ini juga mampu membaca dan menemukan dokumen dengan ekstensi yang sama dan judul yang di inputkan mengandung karakter yang sama tetapi beda penulisan.

VII. SARAN

Berdasarkan hasil penelitian, pengujian, implementasi serta pembahasan mengenai aplikasi pencarian dokumen dengan menggunakan Algoritma *Knuth-Morris-Pratt*, maka untuk pengembangan penelitian selanjutnya penulis menyarankan sebagai berikut.:

1. Aplikasi ini bisa *Case Sensitive* pada proses pemasukkan judul teks.
2. Aplikasi ini bisa membuka dokumen yang telah dicari.

3. Aplikasi ini bisa menyampaikan hasil pencarian dokumen secara informatif dan lengkap, seperti menyampaikan *path* atau tempat dokumen tersebut.

REFERENSI

- [1] Haryanto, B. (2009). *Sistem Operasi*. Bandung: Informatika.
- [2] Basee, S. (2000). *Computer Algorithms : Introduction To Design And Analysis*.
- [3] Safaat, N. (2012). *Pemrograman Aplikasi Mobil Smartphone dan Tablet PC Berbasis Android* . Bandung: Informatika.
- [4] Priawan, M. (2013). *Teknologi, Smartphone, dan Android*.
- [5] Rosa, A. (2011). *Rekayasa Perangkat Lunak*. Bandung: Modula.