

BAB I

PENDAHULUAN

1.1 Latar Belakang

Setiap teknologi maupun alat yang selalu digunakan oleh manusia semakin banyak model dan fungsinya, namun banyak orang yang tidak mengetahui tokoh ilmuwan dibalik alat atau teknologi padahal hampir setiap hari digunakan. Banyak yang melupakan jasa ilmuwan padahal tanpa ada tokoh ilmuwan yang menemukan berbagai macam alat maupun teknologi akan menjadi sulit untuk melakukan pekerjaan karena hampir setiap saat dimanapun dan kapanpun akan selalu membutuhkan alat dan teknologi untuk menunjang pekerjaan yang dilakukan.

Seiring dengan tingkat mobilitas yang tinggi, beberapa tahun terakhir tengah marak perangkat bergerak atau *mobile device*. Salah satu perangkat *mobile* yang paling pesat perkembangannya adalah *smartphone* berbasis *android*, dimana hampir setiap orang memilikinya. *Android* merupakan sistem operasi berbasis *linux* dapat digunakan diberbagai perangkat *mobile*. Hingga saat ini *android* terus berkembang baik secara sistem maupun aplikasinya.[1] Salah satu bentuk aplikasi yang bermanfaat untuk memberikan informasi adalah ensiklopedia. Menurut Kamus Besar Bahasa Indonesia, ensiklopedia adalah buku atau serangkaian buku yang menghimpun keterangan atau uraian tentang berbagai hal ilmu pengetahuan yang disusun menurut abjad atau menurut lingkungan ilmu.

Salah satu komponen yang membentuk sebuah pencarian/pencocokan string pada aplikasi adalah algoritma. Algoritma diciptakan untuk meningkatkan kinerja

suatu aplikasi yang menggungkannya dalam menangani pekerjaan tertentu, karena algoritma adalah urutan langkah-langkah untuk memecahkan suatu masalah.[2]

pencocokan *string* atau pencarian *string* sangat banyak digunakan dalam pemrograman perangkat lunak. Bahkan hampir setiap perangkat lunak mengandung algoritma pencarian *string*. Diperlukan cara yang cepat dalam proses pencocokan *string* yaitu dengan menggunakan algoritma Knuth Morris Pratt (KMP). Langkah-langkah algoritma KMP ini mulai mencocokkan *pattern* pada awal teks, dari kiri ke kanan dan akan mencocokkan karakter per karakter *pattern* dengan karakter teks yang bersesuaian sampai kondisi karakter di *pattern* dan di teks yang dibandingkan tidak cocok, semua karakter di *pattern* cocok, kemudian algoritma akan memberitahukan penemuan diposisi ini.

Karena kurangnya aplikasi yang memberikan informasi tentang penemu alat atau teknologi, membuat banyak orang tidak mengetahui siapa tokoh ilmuwan yang menemukan alat atau teknologi yang sering digunakan, sehingga diperlukan aplikasi dan perangkat *mobile* yang dapat memenuhi tingkat mobilitas yang cukup tinggi. Dengan kemudahan dan perkembangannya yang sangat maju, android menjadi pilihan para pengguna sebagai media komunikasi dan informasi. Dalam merancang aplikasi ensiklopedia dibutuhkan alat pencarian yang bisa memberikan kecepatan dalam pencocokan *string*, dengan menggunakan algoritma Knuth-Morris-Pratt (KMP) ini proses pencocokan tersebut dapat meminimalkan waktu yang secepat mungkin. Dalam mengaplikasikannya, algoritma KMP menyimpan pergeseran *string pattern* yang lebih jauh, tidak hanya satu karakter saja. Dengan Algoritma KMP, waktu pencocokan dapat dikurangi secara signifikan [8]

Dari latar belakang permasalahan di atas, maka tema tersebut menjadi intisari dari skripsi yang di ambil dengan judul “***Implementasi Algoritma Knuth-Morris-Pratt Untuk Optimasi Pencocokan String Pada Aplikasi Ensiklopedia Tokoh Ilmuwan Berbasis Android***”.

1.2 Perumusan Masalah

Merujuk pada latar belakang diatas, maka masalah yang dapat dirumuskan dalam proses pembangunan sistem ini ialah:

1. Bagaimana mengimplementasikan algoritma Knuth-Morris-Pratt untuk optimasi pencocokan *string* pada aplikasi ensiklopedia tokoh ilmuwan berbasis android ?
2. Bagaimana mengetahui perbandingan pencocokan string menggunakan algoritma Knuth-Morris-Pratt dengan sql query pada aplikasi ensiklopedia tokoh ilmuwan berbasis android ?

1.3 Tujuan

Tujuan yang hendak dicapai dalam proses pembuatan skripsi ini adalah:

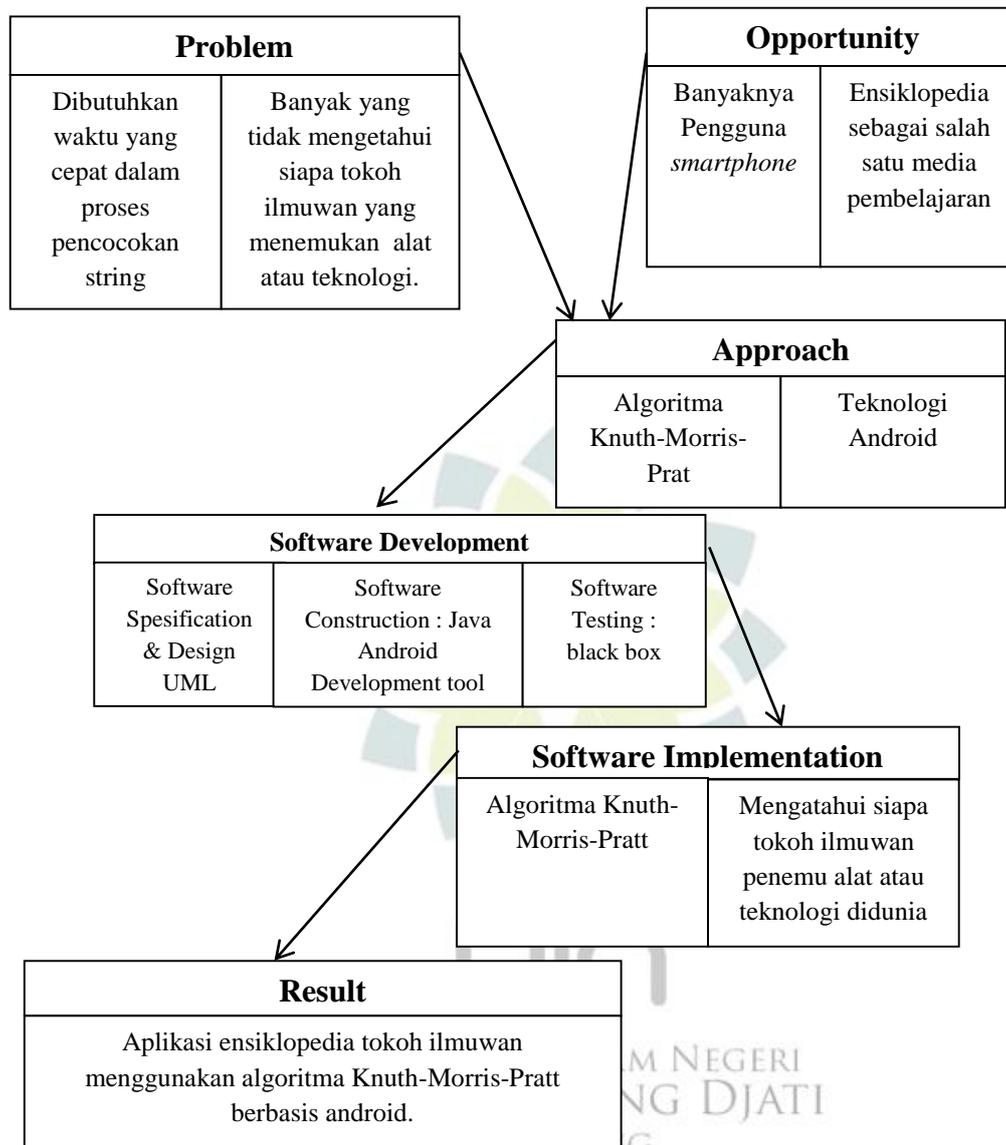
1. Pengimplementasian algoritma Knuth-Morris-Pratt untuk pencocokan *string* pada Aplikasi Ensiklopedia Tokoh Ilmuwan.
2. Merancang media pembelajaran untuk memberikan informasi tentang tokoh ilmuwan serta hasil temuannya, khususnya untuk pelajar atau mahasiswa.

1.4 Batasan Masalah

Untuk mempermudah proses pengerjaan, maka sistem aplikasi yang dirancang akan dibatasi. Adapun batasan masalah yang melingkupi kinerja sistem ini yaitu:

1. Aplikasi hanya berjalan pada *smartphone* bersistem operasi Android.
2. Ensiklopedia yang dirancang yaitu ensiklopedia khusus yang hanya berisi pembahasan tokoh ilmuwan yang berhubungan dengan bidang teknologi dengan data dari buku "*Ensiklopedi Teknologi Penemuan Dan Perkembangannya*" dari jilid 1 sampai jilid 6.
3. Tokoh ilmuwan yang ada dalam database sesuai dengan buku *Database* yang digunakan merupakan penyimpanan secara *offline* menggunakan *SQLite*.
4. *Sql query* sebagai pembanding dengan algoritma KMP untuk proses pencocokan *string* atau pencarian.
5. Aplikasi ini dirancang menggunakan bahasa pemrograman Java pada *platform* Android 4.2.2 dengan menggunakan *IDE Tools Eclipse*.

1.5 Kerangka Pemikiran



Gambar 1.1 kerangka pemikiran penelitian

1.6 State of The Art

Berikut merupakan pembahasan beberapa penelitian sebelumnya yang memiliki konsep yang hampir sama serta sejalan dengan penelitian yang dilakukan. Lalu melihat sejauh mana perbedaan masing-masing penelitian, sehingga masing-masing penelitian mempunyai tema yang original.

Penelitian Utomo, Harjo dan Handoko [4] tentang “Perbandingan Algoritma *String Searching Brute Force*, Knuth-Morris-Pratt, Boyer Moore dan Karp Rabin Pada Teks Alkitab Bahasa Indonesia” meneliti pengaruh panjang pola pada teks yang dicari terhadap waktu pencarian. Panjang pola yang diujikan yaitu 1 sampai 30 karakter.

Dari penelitian ini didapatkan kesimpulan bahwa semakin panjang pola waktu pencariannya tetap (BF), cenderung meningkat (KMP dan KR), dan menurun (BM). Waktu rata – rata (dari 280 pola) yang diperlukan untuk masing – masing algoritma adalah sebagai berikut : BM : 0.92 detik, BF : 0.98 detik, KMP: 0.99 detik, dan KR: 3.46 detik dengan perulangan sebanyak 100 kali. Pada algoritma *Knuth Morris Pratt* waktu pencariannya cenderung sedikit meningkat ketika pola yang dicoba semakin panjang. Hal ini disebabkan pembentukan *table next*-nya. Algoritma *Knuth Morris Pratt* kurang cocok digunakan pada bahasa sehari-hari karena kecil kemungkinan untuk mendapatkan pola yang banyak berulang. Algoritma ini cocok digunakan ketika jumlah *alphabet* kecil, seperti untuk *file* biner yang hanya terdiri dari 2 *alphabet* (0 dan 1).

Penelitian Moch. Yusup Soleh [5] membahas tentang “Implementasi Algoritma KMP dan Boyer-Moore dalam Aplikasi *Search Engine* Sederhana”. Penelitian tersebut berhasil mengkombinasikan 2 buah algoritma *string matching* yaitu algoritma Knuth-Morris-Pratt dan Boyer-Moore didalam sebuah aplikasi pencarian sederhana.

Penelitian Akhtar Rasool dan Nilay Khare [6] membahas paralelisasi algoritma pencocokan string KMP pada arsitektur SIMD (*Single Instruction, Multiple Data*) berbeda: Multicore dan GPGPU (*General Purpose Computing on*

Graphics Processing Units) menghasilkan algoritma KMP dengan paralelisasi sangat meningkatkan efisiensi, cocok jika ukuran teks sangat besar dan jumlah yang cukup dari prosesor yang tersedia. Karakteristik yang paling penting dari algoritma KMP adalah dengan membuat lebih baik menggunakan karakter bit berikutnya, jarak bergerak maksimum dicapai.

Hasil dari semua penelitian yang sudah dipaparkan pada paragraf sebelumnya sudah memberi gambaran aplikasi yang dibuat dan akan sangat membantu dalam menentukan tema yang original. Dapat di ambil ikhtisar dari penelitian-penelitian tersebut dan membandingkannya dengan penelitian yang akan dilakukan, ditampilkan pada Tabel 1.1.

Tabel 1.1 *State of The Art*

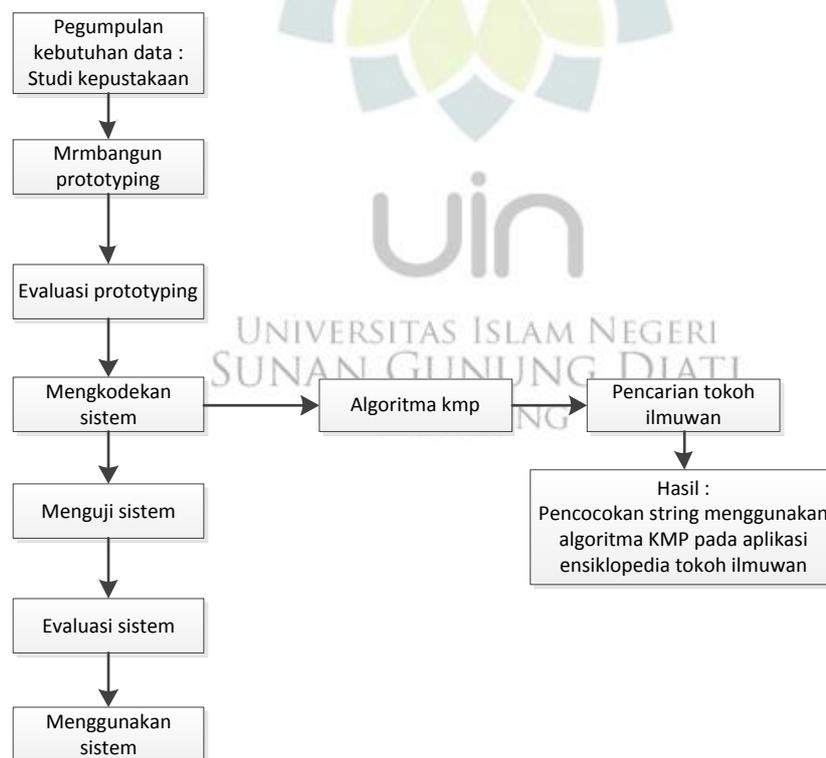
No	Metode	Utomo, Harjo, Handoko (2008)	Moch. Yusup Soleh (2011)	Akhtar Rasool, Nilay Khare (2012)	Irwan Irawan (2014)
1	KMP	✓	✓	✓	✓
2	Teknologi Android	-	-	-	✓
3	Kamus	-	-	-	✓
Membandingkan dengan algoritma lain					
4	Boyer Moore	✓	✓	-	-
5	<i>SQL Query</i>	-	-	-	✓
Bahasa yang digunakan					
6	Java	✓	-	✓	✓
7	PHP	-	✓	-	-

Dari seluruh penelitian menggunakan algoritma KMP, namun penelitian dari Utomo, Harjo dan Handoko (2008), Yusup Soleh (2011), Akhtar Rasool, Nilay Khare (2012) tidak menggunakan teknologi Android seperti penelitian dari Irwan Irawan (2014) dari aplikasi yang dibuat Irwan Irawan memasukan menu kamus untuk memudahkan pengguna agar bisa melihat seluruh daftar data yang ada di dalam aplikasi. Perbandingan algoritma yang dilakukan oleh Utomo,

Harjo, Handoko (2008), Yusup Soleh (2011) menggunakan algoritma lain yaitu Boyer Moore tapi peneliti Akhtar Rasool, Nilay Khare (2012) tidak menggunakan metode lain dalam membandingkan algoritma KMP, Irwan Irawan (2014) tidak menggunakan algoritma lain dalam membandingkan kecepatan pencocokan *string* menggunakan algoritma KMP tetapi membandingkannya dengan menggunakan metode *SQL query*. Bahasa Pemrograman yang digunakan semua peneliti yaitu java kecuali Yusup Soleh (2011) menggunakan bahasa pemrograman PHP.

1.7 Metodologi Penelitian

Guna mendapatkan data yang diperlukan untuk membantu dalam penelitian yang akan dilakukan, maka digunakan metodologi sebagai berikut:



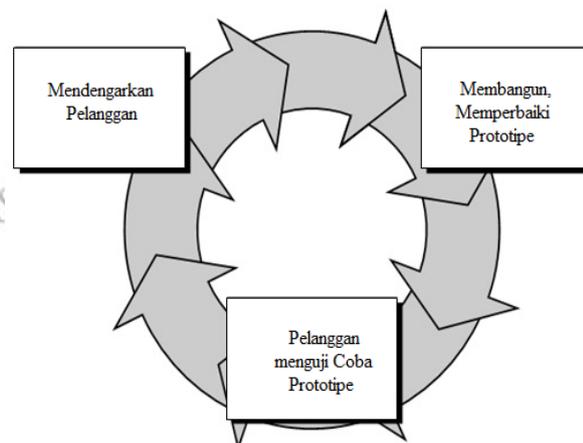
Gambar 1.2 Metodologi Penelitian

1.7.1 Pengumpulan Data

Metode Kepustakaan digunakan untuk mendapatkan konsep-konsep teoritis dengan cara menganalisa data pada literatur (pustaka) dan media lain yang dapat membantu dalam pemecahan masalah.

1.7.2 Metode Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak yang digunakan untuk penelitian ini adalah model prototype. Prototype merupakan teknik analisis data dalam pengembangan perangkat lunak menggunakan model prototipe. Menurut Roger S. Pressman [7] pendekatan prototipe atau *prototyping* paradigma sangat cocok digunakan untuk sistem atau perangkat lunak yang dibangun mengikuti kebutuhan pengguna, metode ini sangat sesuai diterapkan dalam proses perancangan perangkat lunak yang akan dibangun yang menitik-beratkan pada pendekatan aspek desain, fungsi, dan *user-interface*.



Gambar 1.3. Paradigma Prototyping [7]

Dengan model prototipe ini perancang dan pengguna bertemu untuk mendefinisikan secara obyektif keseluruhan perangkat lunak, mengidentifikasi kebutuhan yang diketahui, dan area lebih besar dimana definisi lebih jauh merupakan keharusan kemudian dilakukan perancangan kilat berupa maket atau

prototipe sistem untuk kemudian dievaluasi pengguna untuk menyaring kebutuhan pengembangan perangkat lunak.

Tahapan-tahapan dalam metode *Prototype* adalah sebagai berikut:

a. Pengumpulan kebutuhan

Pelanggan dan pengembang bersama-sama mendefinisikan *format* seluruh perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.

b. Membangun *prototyping*

Membangun *prototyping* dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan. Perancangan sementara berupa rancangan perangkat lunak dengan menggunakan *Unified Modelling Language* (UML).

c. Evaluasi *prototyping*

Evaluasi ini dilakukan oleh pelanggan apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai maka langkah d akan diambil. Jika tidak *prototyping* direvisi dengan mengulangi langkah a, b, dan c.

d. Mengkodekan sistem

Dalam tahap ini, *prototyping* yang sudah di sepakati diterjemahkan ke dalam bahasa pemrograman java.

e. Menguji sistem

Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan *Black Box Testing*.

f. Evaluasi Sistem

Pelanggan mengevaluasi apakah sistem yang sudah jadi sesuai dengan yang diharapkan. Jika ya, langkah g dilakukan; jika tidak, ulangi langkah c dan e.

g. Menggunakan sistem

Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.

Beberapa keuntungan menggunakan model prototipe pada pengembangan perangkat lunak yaitu adanya komunikasi yang baik antara pengembang dan pelanggan, pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan, pelanggan berperan aktif dalam pengembangan sistem, lebih menghemat waktu dalam pengembangan sistem, dan penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya.

Walaupun banyak sekali keuntungan menggunakan model prototipe ini namun terdapat pula beberapa kekurangan. Model pengembangan perangkat lunak prototipe kurang efektif untuk proyek pembangunan perangkat lunak yang besar dan membutuhkan dukungan sistem secara terus menerus karena model ini lebih memperhatikan desain awal dan kebutuhan pengguna tanpa memperhatikan kepastian efisiensi algoritma dan manajemen resiko terhadap produk perangkat lunak yang dibangun.

1.8 Sistematika Penulisan

BAB I PENDAHULUAN

Bab I menguraikan latar belakang, perumusan masalah yang merumuskan berbagai masalah yang diteliti secara lebih jelas, tujuan penelitian yang berisi tentang tujuan dilakukannya penelitian, manfaat penelitian, batasan masalah untuk memberikan batasan yang tegas dan jelas serta sistematika penulisan yang menguraikan urutan penyajian yang digunakan dalam penyusunan skripsi ini.

BAB II LANDASAN TEORI

Bab II membahas tentang landasan teori dari topik penulisan skripsi secara mendalam beserta dengan referensinya.

BAB III ANALISIS DAN PERANCANGAN

Bab III akan menguraikan hasil analisis dan perancangan aplikasi yang akan dibangun.

BAB IV IMPLEMENTASI

Bab IV akan menguraikan implementasi aplikasi yang telah dianalisa dan dirancang sebelumnya.

BAB V PENUTUP

Bab V berisi uraian tentang kesimpulan dan saran terhadap aplikasi yang hendak dibangun dan dikembangkan lebih lanjut.