

Perancangan Aplikasi Katalog Perpustakaan Dengan Menerapkan Algoritma Boyer Moore

Nabila Rahmi Arifa

Program Studi Teknik Informatika, Universitas Budi Darma Medan, Indonesia

Email: bilaiskandar14@gmail.com

Email Koresponden: bilaiskandar14@gmail.com

Abstrak– Perpustakaan sebagai penyedia informasi yang representative. Artinya kehidupan dan keberadaan perpustakaan dapat menjadi rantai sejarah bagi masa lalu, akar bagi hidup dimasa sekarang ini dan pembimbing untuk melangkah ke masa depan. Pelayanan informasi terdiri dari lima aspek yaitu pengumpulan informasi, pelestarian informasi, pengolahan informasi, pemanfaatan informasi dan penyebarluasan informasi. String matching menggunakan Algoritma Boyer Moore untuk melakukan pencocokan dengan pattern yang akan di cari dari katalog perpustakaan, sehingga pencocokan dalam pemilihan pattern dapat dilihat kecocokannya. Diharapkan melalui Algoritma Boyer Moore dapat menyelesaikan pencocokan dalam pengenalan katalog perpustakaan, sehingga dapat membantu para pelajar, pekerja, dan masyarakat dalam melakukan pencarian katalog perpustakaan.

Keyword : String Matching, Perpustakaan, Algoritma Boyer Moore

Abstract– *Libraries as representative providers of information. This means that the life and existence of a library can be a chain of history for the past, a root for life in the present and a guide for moving into the future. Information services consist of five aspects, namely information collection, information preservation, information processing, information utilization and information dissemination. String matching uses the Boyer Moore Algorithm to match patterns that will be searched for from the library catalog, so that the matches in pattern selection can be seen. It is hoped that the Boyer Moore Algorithm can complete matching in the introduction of library catalogues, so that it can help students, workers and the public in searching the library catalogue.*

Keywords: *String Matching, Library, Boyer Moore Algorithm*

1. PENDAHULUAN

Perpustakaan sebagai salah satu penyedia informasi yang keberadaannya sangat penting terutama di dunia pendidikan dimana perpustakaan sangat berkaitan erat dengan ilmu pengetahuan dan teknologi informasi. Ketiganya saling mendukung satu dengan yang lainnya, perpustakaan memberikan kontribusi terhadap ilmu pengetahuan melalui penyimpanan berbagai informasi dan sejarah perkembangan ilmu pengetahuan, sedangkan teknologi informasi memberikan dukungan pada kemudahan akses dan sistem informasi dalam sebuah perpustakaan.

Museum Negeri Provinsi Sumatera Utara memiliki sebuah perpustakaan yang berisikan buku-buku sejarah kebudayaan dari suku yang ada di Sumatera Utara. Perpustakaan tersebut memiliki koleksi buku yang terbilang banyak, namun perpustakaan tersebut belum memiliki aplikasi yang berisikan daftar buku yang ada dalam perpustakaan tersebut. Daftar buku masih disajikan dalam bentuk buku besar yang kemudian datanya diinputkan dengan menggunakan aplikasi Microsoft Excell. Keterbatasan aplikasi tersebut menyebabkan kesulitan untuk menemukan letak buku. Selain itu pegawai dari perpustakaan tersebut sering mengalami kesulitan dalam menjelaskan ketersediaan buku ataupun menjelaskan isi dari buku tersebut. Untuk menyelesaikan permasalahan tersebut dibutuhkan suatu aplikasi yang dapat menyajikan informasi tentang ketersediaan buku dan informasi tentang isi buku tersebut dalam bentuk katalog. Aplikasi catalog diharapkan dapat mempermudah pengunjung dalam mencari letak dan informasi buku. Untuk mempermudah pencarian buku tersebut maka dibutuhkan algoritma pencocokan string (String Matching). Algoritma string matching yang digunakan dalam penelitian ini adalah algoritma Boyer Moore.

Sistem penelusuran yang akan dibangun memerlukan algoritma pencarian string. Algoritma Boyer-Moore ini dianggap sebagai algoritma yang paling efisien dalam implementasi penelusuran katolog buku pada perpustakaan.

Algoritma boyer Moore adalah salah satu algoritma untuk mencari suatu string di dalam teks, dibuat oleh R.M Boyer dan J.S Moore. Algoritma Boyer Moore melakukan perbandingan dimulai dari kanan ke kiri, tetapi pergeseran window tetap dari kiri kekanan[1]. Jika terjadi kecocokan maka dilakukan perbandingan karakter teks dan karakter pola yang sebelumnya, yaitu dengan sama-sama mengurangi indeks teks dan pola masing-masing sebanyak satu.

Ide pokok dari algoritma ini ialah mengubah perhitungan awal yang berguna untuk mempersiapkan tabel yang akan digunakan. Pada algoritma Boyer Moore biasa, pembacaan awal dilakukan terhadap satu karakter saja, sedangkan pada algoritma BM', pembacaan awal dilakukan terhadap dua karakter sekaligus[2].

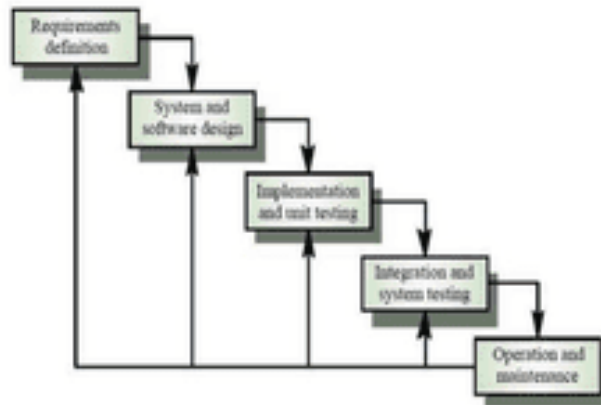
2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini akan dilaksanakan melalui beberapa tahapan pengembangan waterfall [1] sebagai berikut:

1. Analisis data sistem ini bertujuan untuk memberikan gambaran tentang sistem yang diusulkan, menganalisis teknologi yang dibutuhkan.
2. Desain sistem ini bertujuan untuk memberikan gambaran umum tentang sistem yang diusulkan. Desain yang digunakan adalah Diagram Alir (Flowchart).
3. Coding & Testing dalam hal ini difokuskan pada pembuatan program dan penggunaannya. Pada tahap ini bertujuan untuk meletakkan Algoritma Boyer Moore yang di buat agar siap untuk dioperasikan atau dijalankan.
4. Pengujian ini menggunakan metode pengujian whitebox. Pengujian ini digunakan untuk memastikan bahwa semua statemen pada program telah dieksekusi paling tidak satu kali selama pengujian dan bahwa semua kondisi logis telah diuji.
6. Pemeliharaan perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (peripheral atau sistem operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

Setiap tahapan dilakukan secara berurutan mulai dari langkah pertama sampai langkah terakhir, setiap langkah yang telah selesai dikerjakan harus dilakukan pengkajian ulang, tertera dalam gambar 1 berikut :



Gambar 1 Alur dari Penelitian

2.1 String Matching

String matching adalah proses pencarian semua kemunculan query yang selanjutnya di sebut Pattern kedalam string yang lebih panjang (text). Pattern di lambangkan dengan $x=x[0..m-1]$ dan panjangnya adalah m . Teks di lambangkan dengan $y=y[0..n-1]$ dan panjangnya adalah n . kedua string terdiri dari sekumpulan karakter yang disebut alphabet yang di lambangkan dengan Σ dan mempunyai ukuran $\sigma[3]$. Algoritma string matching mempunyai tiga komponen yaitu:

1. Pattern, yaitu deretan karakter yang di cocokkan dengan teks
2. Teks, yaitu tempat pencocokkan pattern di lakukan
3. Alfabet, yang berisi semua simbol yang di gunakan oleh bahasa pada text pattern

String merupakan bagian penting dari sebuah proses pencarian string (string searching) dalam sebuah dokumen. hasil dari pencarian sebuah string dalam dokumen tergantung dari teknik atau cara pencocokkan string yang di gunakan. Algoritma string matching merupakan suatu algoritma yang digunakan untuk memecahkan masalah pencocokkan suatu teks terhadap teks lain[3]. Algoritma string yang digunakan adalah boyer moore.

3.1.1 Klasifikasi String Matching

Secara garis besar teknik string matching dalam pencocokkan string dibedakan menjadi 2 yaitu :

1. Exact string matching
Exact string matching merupakan pencocokkan string secara tepat dengan susunan karakter dalam string yang dicocokkan memiliki jumlah maupun urutan karakter dalam string yang sama. Bagian algoritma ini bermanfaat jika pengguna ingin mencari string dalam dokumen yang sama persis dengan string masukan. Beberapa contoh algoritma exact string matching yang mengemukakan antara lain :
 - a Brute Force
Brute Force membandingkan karakter per karakter sampai ditemukannya pola yang dicari dari awal string sampai dengan akhir string.
 - b Knuth-Morris-Pratt

- Knuth-Morris-Pratt membandingkan karakter per karakter sampai awal string sampai dengan akhir string dari arah kiri ke arah kanan
- c Boyer Moore
Boyer moore melakukan pencocokan karakter yang dimulai dari kanan ke kiri.
2. Inexact string matching
Inexact string matching merupakan pencocokan string secara samar, maksudnya pencocokan string dimana string yang dicocokkan memiliki kemiripan dimana keduanya memiliki susunan karakter yang berbeda tetapi string-string tersebut memiliki kemiripan baik kemiripan tekstual atau penulisan atau kemiripan ucapan (phonetic string matching). Metode Inexact string matching diarahkan untuk mencari nilai dari beberapa string yang mendekati dan tidak hanya menghasilkan cocok atau tidak cocok.

3.2 Algoritma Boyer Moore

Algoritma Boyer Moore adalah salah satu algoritma untuk mencari suatu string di dalam teks, dibuat oleh R.M Boyer dan J.S Moore. Ide utama Algoritma ini adalah mencari string dengan melakukan perbandingan karakter mulai dari karakter paling kanan dari string yang dicari. Kemudian panjang teks dihitung menggunakan Match Heuristic (MH) dan Occurrence Heuristic (OH). Setelah nilai MH dan OH sudah ditemukan, nilai MH dan OH tersebut dibandingkan untuk menentukan lompatan teks. Nilai yang paling besar yang akan digunakan untuk menentukan lompatan teks sehingga membuat hasil dari pencocokan pattern terhadap teks jauh lebih cepat dibandingkan dengan yang lain. Dengan menggunakan Algoritma ini, secara rata - rata proses pencarian akan menjadi lebih cepat jika dibandingkan dengan Algoritma lainnya[4].

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Algoritma Boyer Moore

Implementasi algoritma *boyer moore* pada penelitian ini yaitu untuk mendapatkan hasil pencarian data Museum Negeri Provinsi Sumatera Utara dengan cepat dan tepat. Salah satu masalah pencarian data Museum Negeri Provinsi Sumatera Utara yang akan dilakukan pada penelitian ini adalah pencarian karakter. Pada penelitian ini Boyer Moore diterapkan untuk melakukan pencarian katalog perpustakaan, maka yang menjadi teks dalam penelitian ini adalah nama katalog dari perpustakaan tersebut. Langkah pencarian diuraikan dibawah ini :

Teks : PAMERAN PATUNG

Pattern : PATUNG

Pada tahap ini untuk mencari nilai OH: M-1-i

Keterangan :

i : Nilai index

M : Pattern/karakter (N)

Tabel 1. bmBc

| | | | | | | |
|----------------------------------|---|---|---|---|---|---|
| index | 0 | 1 | 2 | 3 | 4 | 5 |
| Pattern | P | A | T | U | N | G |
| <i>Occurrence Heuristic (OH)</i> | 5 | 4 | 3 | 2 | 1 | 0 |

1. Untuk mendapatkan nilai P yaitu: 6-1-0 maka dapat nilai P adalah 5
2. Untuk mendapatkan nilai A yaitu: 6-1-1 maka dapat nilai A adalah 4
3. Untuk mendapatkan nilai T yaitu: 6-1-2 maka dapat nilai T adalah 3
4. Untuk mendapatkan nilai U yaitu: 6-1-3 maka dapat nilai U adalah 2
5. Untuk mendapatkan nilai N yaitu: 6-1-4 maka dapat nilai N adalah 1
6. Untuk mendapatkan nilai G yaitu: 6-1-5 maka dapat nilai G adalah 0

Tabel 2 Nilai OH dan MH pada *pattern* penyelesaian :

| | | | | | | |
|----------------------------------|---|---|---|---|---|---|
| index | 0 | 1 | 2 | 3 | 4 | 5 |
| Pattern | P | A | T | U | N | G |
| <i>Occurrence Heuristic (OH)</i> | 5 | 4 | 3 | 2 | 1 | 0 |
| <i>MacthHeuristik(MH)</i> | 6 | 6 | 6 | 6 | 6 | 1 |

Tahapan pencocokan *string* untuk pencarian karakter *pattern* PATUNG pada *text* PAMERAN PATUNG adalah sebagai berikut ini :

1. Tahap Pertama

Pada tahap pertama, pencocokan *string pattern* dimulai dengan deretan *string text* yang pertama "ERA ".

| | | | | | | | | | | | | | | |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|
| index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|

| | | | | | | | | | | | | | | |
|---------|---|---|---|---|---|---|---|--|---|---|---|---|---|---|
| Teks | P | A | M | E | R | A | N | | P | A | T | U | N | G |
| Pattren | P | A | T | U | N | G | | | | | | | | |

Pada tahap ini karakter G dengan A tidak sama dengan karakter *pattern* yang lainnya, sehingga bergeser sebanyak jumlah karakter *pattern* yaitu 4 langkah.

2. Tahap Kedua

Pada tahap kedua yaitu pencocokan *string* PATUNG dengan deretan *text* "PA ".

| | | | | | | | | | | | | | | | |
|---------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|---|
| index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | |
| Teks | P | A | | M | E | R | A | N | | P | A | T | U | N | G |
| Pattren | | | | | P | A | T | U | N | G | | | | | |

Pada tahap ini karakter G dengan A tidak sama, dengan karakter *pattern* yang lainnya, sehingga bergeser sebanyak jumlah karakter *pattern* yaitu 4 langkah.

3. Tahap Ketiga

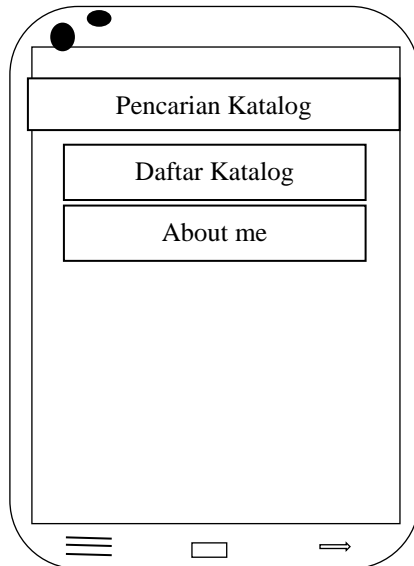
Pada tahap ketiga dilakukan pencocokan *string pattern* dengan deretan *string text* "PATUNG".

| | | | | | | | | | | | | | | |
|---------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|
| index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| Teks | P | A | M | E | R | A | N | | P | A | T | U | N | G |
| Pattren | | | | | | | | | P | A | T | U | N | G |

Pada tahap ini semua karakter *pattern* sesuai dengan karakter *text* yang sejajar dengan karakter *pattern*, sehingga pergeseran karakter *pattern* berhenti.

3.2 Hasil

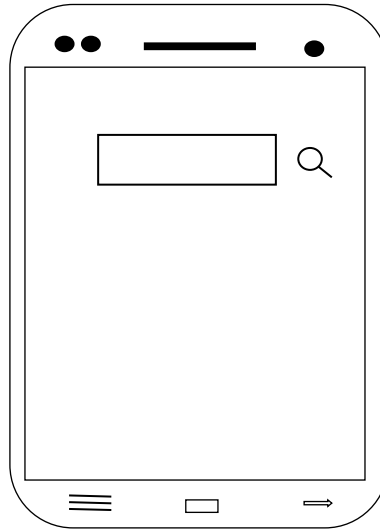
Rancangan antar muka pada tampilan utama adalah tampilan pertama program ketika dijalankan. Di mana pengguna dapat menginput nama rambu yang akan di cari informasinya. di tampilan utama terdapat 3 *button*.



Gambar 2 Tampilan Utama

3.2.1 Pencarian Katalog

Pencarian rambu adalah tampilan ketika user memilih Button daftar pencarian katalog pada tampilan utama. *Form* pencarian rambu berfungsi untuk melakukan pencarian katalog. User dapat melakukan pencarian dengan menulis karakter di tabel pencarian setelah menulis karakter klik tombol maka karakter yang di cari akan muncul.



Gambar 3 Pencarian katalog

Keterangan:

1. *Edittext*, untuk memasukkan nama katalog.
2. *Button*, tombol cari.

3.2.2 Daftar Katalog Perpustakaan

Daftar katalog perpustakaan adalah tampilan ketika user melakukan pencarian pada tampilan pencarian. Dan daftar katalog akan menampilkan gambar atau karakter yang di cari oleh user, dan ini lah tampilan dari Daftar katalog yang sudah dicari oleh user di pencarian di atas



Gambar 4 Daftar Katalog

3.2.3 Tampilan Tentang dan Bantuan

Rancangan *About me* adalah tampilan tentang berisikan tentang aplikasi sedangkan tampilan menampilkan bantuan atau cara penggunaan aplikasi.



Gambar 5 *About me*

4. KESIMPULAN

Berdasarkan hasil akhir pemecahan masalah pencarian Katalog Perpustakaan pada penelitian maka penulis menguraikan beberapa kesimpulan. Adapun kesimpulan-kesimpulan tersebut antara lain sebagai berikut : Dengan menyusun informasi buku perpustakaan dalam bentuk katalog. Menerapkan metode Boyer Moore untuk pencarian informasi buku pada aplikasi katalog buku perpustakaan. Aplikasi Katalog Perpustakaan telah selesai dirancang dengan menggunakan editor eclipse juno dan telah dapat dijalankan untuk mencari katalog perpustakaan..

UCAPAN TERIMAKASIH

Terima kasih disampaikan kepada pihak-pihak yang telah mendukung terlaksananya penelitian ini.

REFERENCES

- [1] K. W. Argakusumah and S. Hansun, "Implementasi Algoritma Boyer-Moore pada Aplikasi Kamus Kedokteran Berbasis Android," *J. Ultim.*, vol. 6, no. 2, pp. 70–78, 2014.
- [2] A. Mushofan, I. T. Bandung, and J. G. Bandung, "Pengembangan Algoritma Boyer Moore," 2014.
- [3] R. Fitri Riyanto Sarwono, Yeni Anistyasari, *SEMANTIC SEARCH*, Andi Offset. Yogyakarta, 2012.
- [4] K. W. Argakusumah and S. Hansun, "Implementasi Algoritma Boyer-Moore pada Aplikasi Kamus Kedokteran Berbasis Android," *J. Ultim.*, vol. 6, no. 2, pp. 70–78, 2014.
- [5] S. S. Wicida, "PENELUSURAN KATALOG PERPUSTAKAAN PADA SMA IT YABIS BONTANG DENGAN ALGORITMA BOYER-MOORE," pp. 15–21, 2008.
- [6] K. W. Argakusumah and S. Hansun, "Implementasi Algoritma Boyer-Moore pada Aplikasi Kamus Kedokteran Berbasis Android," *J. Ultim.*, vol. 6, no. 2, pp. 70–78, 2014.
- [7] Rosa A. S. M. Shalahuddin, *Rekayasa Perangkat Lunak*. Bandung: Informatika Bandung, 2016.
- [8] Murtiwyati and G. Lauren, "JURNAL ILMIAH KOMPUTASI Komputer & Sistem Informasi 1-10," *J. Ilm.*, vol. 12, p. 2,3, 2013.
- [9] S. K. Alfa Satyaputra, M.cs & Eva Maulina Aritonang, *Java for beginners with eclipse 4.2 juno*. Jakarta: PT. Elex Media Komputindo, 2012.