
PENERAPAN SEARCH ENGINE MENGGUNAKAN ALGORITMA BOYER MOORE PENGEMBANGAN APLIKASI BERBASIS WEB PERUSAHAAN FARMASI (STUDI KASUS : PT GALENIKA FARMASINDO)

Harjono Padmono Putro¹⁾ Rudi Hendarto²⁾

¹⁾Harjono Padmono Putro
Sekolah Tinggi Teknologi Informasi NIIT I-Tech
Jl. Asem 2 No.22, Cipete – Jakarta Selatan
<http://www.i-tech.ac.id>
harjono@i-tech.ac.id

²⁾Rudi Hendarto
Sekolah Tinggi Teknologi Informasi NIIT I-Tech
Jl. Asem 2 No.22, Cipete – Jakarta Selatan
<http://www.i-tech.ac.id>
email@i-tech.ac.id

ABSTRAKSI

Kemunculan berbagai *search engine* memiliki peranan yang besar dalam proses pencarian informasi, dimana pemilihan dan penggunaan algoritma yang benar dalam pembangunan *search engine* merupakan langkah yang sangat penting dalam pembangunan *search engine* ini. Kompleksitas data yang ada dalam dunia bisnis farmasi harus ditangani dengan menggunakan komputerisasi dan membutuhkan model atau alat pencarian data.

Search engine ini dibangun dengan menggunakan algoritma *Boyer-Moore String Search*, dimana proses pembangunan *search engine* ini menggunakan *Unified Modelling Language* (UML), seperti : *Usecase* Diagram, *Activity* Diagram dan *Deployment* Diagram. *Search engine* ini dibangun dengan menggunakan Bahasa Pemrograman PHP dan terkoneksi ke basisdata MySQL, dimana *search engine* ini akan ditempatkan dalam website PT.Galenika FarmaSindo yang telah dipergunakan.

Keyword : *Information Retrieval, Boyer Moore, Search Engine*

PENDAHULUAN

Berbagai jenis *search engine* yang telah ada saat ini, seperti : Altavista.com, Excite.com, Yahoo.com, AskJeeves.com, bing.com, dan Google.com merupakan suatu hal yang sangat menarik untuk dicermati perilakunya dalam kemampuannya untuk menghasilkan berbagai dokumen, file dan web yang diinginkan oleh pengguna *search engine*. Seiring dengan perkembangan teknologi internet saat ini, maka *search engine* memiliki peranan yang besar dalam proses pencarian informasi, oleh karena itu kekuatan *search engine* dalam menyajikan informasi terletak dalam kecepatan dan kedekatan antara informasi yang dicari dengan informasi yang ditemukan oleh *search engine*.

Kecepatan dan kedekatan informasi yang dicari dengan kebutuhan informasi pengguna *search*

engine merupakan hal penting, dimana pemilihan dan penggunaan algoritma yang benar dalam pembangunan *search engine* merupakan langkah yang sangat penting dalam pembangunan *search engine* ini. Berbagai algoritma yang sering dipakai dalam membangun *search engine*, seperti : *Sequential Search*, *Index Sequential Search*, *Binary Search* dan *Fibonacci Search* memiliki metodanya sendiri dalam melakukan pencarian berdasarkan keywordnya sehingga akan dapat menghasilkan informasi yang berbeda dengan menggunakan keyword yang sama.

Bisnis dibidang farmasi membutuhkan dan memiliki data yang sangat besar, sehingga pengelolaan data dalam mencari informasi dengan cepat dan tepat untuk kebutuhan bisnis sangatlah penting. Kompleksitas data yang ada dalam dunia

bisnis farmasi harus ditangani dengan menggunakan komputerisasi dan membutuhkan model atau alat pencarian data untuk menjadi informasi yang baik. Perusahaan farmasi PT. Galenika Farmasindo sudah memiliki website, dan akses data internal perusahaan sudah berjalan dengan baik, tetapi pencarian informasi untuk kebutuhan perusahaan dan konsumen belum tersedia

RUMUSAN MASALAH :

1. Bagaimana cara *Search Engine* yang dikembangkan akan diintegrasikan ke dalam web yang sudah terlebih dahulu di implementasikan.
2. Bagaimana cara *Search Engine* akan dikembangkan dengan menggunakan algoritma *Boyer Moore*.
3. Bagaimana cara *Search Engine* akan dikembangkan dengan menggunakan bahasa PHP.
4. Bagaimana cara *Search Engine* ini hanya berfungsi untuk mencari data dan informasi yang bersumber dari basisdata internal perusahaan

BATASAN MASALAH

Laporan tugas akhir ini dibatasi hanya pada pembahasan :

1. *Search Engine* yang dikembangkan akan diintegrasikan ke dalam web yang sudah terlebih dahulu di implementasikan.
2. *Search Engine* akan dikembangkan dengan menggunakan algoritma *Boyer Moore*.
3. *Search Engine* akan dikembangkan dengan menggunakan bahasa PHP.

Search Engine ini hanya berfungsi untuk mencari data dan informasi yang bersumber dari basisdata internal perusahaan

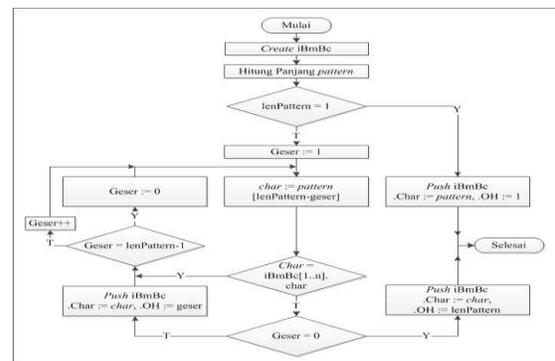
Temu kembali Informasi (*Information Retrieval*) :

Information Retrieval System atau Sistem Temu Balik Informasi merupakan bagian dari *computer science* tentang pengambilan informasi dari dokumen-dokumen yang didasarkan pada isi dan konteks dari dokumen-dokumen itu sendiri. Menurut Kowalski. J Gerald, 2002 di dalam bukunya “*Information Storage and Retrieval Systems Theory and Implementation*”, sistem temu balik informasi adalah suatu sistem yang mampu melakukan penyimpanan, pencarian, dan pemeliharaan informasi. Informasi dalam konteks ini dapat terdiri dari teks (termasuk data numerik dan tanggal),

gambar, audio, video, dan objek multimedia lainnya. Tujuan dari sistem IR adalah memenuhi kebutuhan informasi pengguna dengan me-retrieve semua dokumen yang mungkin relevan, pada waktu yang sama me-retrieve sesedikit mungkin dokumen yang tidak relevan. Sistem IR yang baik memungkinkan pengguna menentukan secara cepat dan akurat apakah isi dari dokumen yang diterima memenuhi kebutuhannya. Agar representasi dokumen lebih baik, dokumen-dokumen dengan topik atau isi yang mirip dikelompokkan bersama-sama

Algoritma Boyer-Moore :

Algoritma Boyer Moore termasuk algoritma *string matching* yang paling efisien dibandingkan algoritma-algoritma *string matching* lainnya. Karena sifatnya yang efisien, banyak dikembangkan algoritma *string matching* dengan bertumpu pada konsep algoritma *Boyer Moore*, beberapa di antaranya adalah algoritma *Turbo BM* dan algoritma *Quick Search*. Algoritma *Boyer Moore* menggunakan metode pencocokan *string* dari kanan ke kiri yaitu men-scan karakter *pattern* dari kanan ke kiri dimulai dari karakter paling kanan. Algoritma *Boyer Moore* menggunakan dua fungsi *shift* yaitu *good-suffix shift* dan *bad-character shift* untuk mengambil langkah berikutnya setelah terjadi ketidakcocokan antara karakter *pattern* dan karakter teks yang dicocokkan.



AGILE MODEL :

Kata **Agile** berarti bersifat cepat, ringan, bebas bergerak, waspada. Kata ini digunakan sebagai kata yang menggambarkan konsep model proses yang berbeda dari konsep model-model proses yang sudah ada. Konsep Agile Software Development dicituskan oleh Kent Beck dan 16 rekannya dengan menyatakan bahwa Agile Software Development adalah cara membangun software dengan melakukannya dan membantu orang lain membangunnya sekaligus.

Dalam Agile Software Development interaksi dan personel lebih penting dari pada proses dan alat,

software yang berfungsi lebih penting daripada dokumentasi yang lengkap, kolaborasi dengan klien lebih penting dari pada negosiasi kontrak, dan sikap tanggap terhadap perubahan lebih penting daripada mengikuti rencana. Namun demikian, sama seperti model proses yang lain, Agile Software Development memiliki kelebihan dan tidak cocok untuk semua jenis proyek, produk, orang dan situasi. Agile Software Development memungkinkan model proses yang toleransi terhadap perubahan kebutuhan sehingga perubahan dapat cepat ditanggapi. Namun di sisi lain menyebabkan produktifitas menurun.

Whitebox Testing :

White box testing didasarkan pada *internal logic* dan struktur dari code. *Whitebox testing* juga biasa disebut *glass*, *structural*, *open box*, dan *clear box testing*. Tes – tes yang ditulis berdasarkan *whitebox testing strategy* mencakup gabungan dari penulisan *code*, *branches*, *paths*, *statement*, dan *internal logic* dari *code*, dan lainnya.

Selain itu untuk mengimplementasikan *whitebox testing*, *tester* harus menyesuaikan *code* dan (kunci) *hence* yang dibutuhkan untuk memahami / mengerti pengkodean dan *logic internal working* dari *code*. Pengujian *Whitebox* juga membutuhkan tester untuk memeriksa *code* dan menemukan unit atau *atau chunk* mana dari code yang *malfunctioning*. Keuntungan dari *whitebox testing*

- a. Pengetahuan tentang *internal coding structure* merupakan prasyarat, karena akan lebih mudah mengetahui tipe dari *input* atau data mana yang dapat membantu pengujian aplikasi secara efektif.
- b. Dengan *whitebox testing* kita dapat mengoptimalisasi kode
- c. *Whitebox testing* juga membantu menghapus kelebihan baris *code* yang dapat menyebabkan kerusakan atau cacat yang tersembunyi.

Sedangkan kekurangan dari *whitebox testing*, seperti :

- a. Pengetahuan tentang *internal coding structure* merupakan prasyarat, oleh karena itu, *skill* atau kemampuan dari tester sangat berpengaruh dalam *whitebox testing*, dan ini dapat meningkatkan biaya.
- b. Dan hampir tidak mungkin untuk melihat setiap *byte code* untuk menemukan *hidden errors*, yang menimbulkan masalah dan kegagalan *output* pada aplikasi.

BlackBox Testing :

Pengujian *Black Box* atau disebut juga sebagai Pengujian Fungsional atau tingkah laku dan

pengujian kotak tertutup mengambil perspektif eksternal dari objek pengujian. Pengujian-pengujian tersebut dapat bersifat tidak fungsional ataupun fungsional, meskipun pada umumnya fungsional. Jenis pengujian di bawah strategi ini secara total memfokuskan pada pengujian untuk kebutuhan dan kemampuan aplikasi.

Metoda pengujian ini dapat digunakan untuk semua tingkat pengujian perangkat lunak, pengujian unit, pengintegrasian, pengujian sistem dan penerimaan. Semakin tinggi tingkatannya, maka semakin rumit dan semakin besar kotaknya, sehingga untuk menyederhanakannya diperlukanlah pengujian *Black Box*.

Website :

Website merupakan sebuah halaman berisi informasi yang dapat dilihat jika komputer anda terkoneksi dengan internet. Dengan adanya website, semua orang di dunia bisa mendapatkan dan mengelola informasi dengan berbagai sumber yang tersedia di internet. Website sendiri saat ini bisa memuat berbagai macam media, mulai dari teks, gambar, suara, bahkan video.

Penemu website adalah Sir Timothy John Tim Berners-lee, sedangkan website yang tersambung dengan jaringan pertama kali muncul pada tahun 1991. Awalnya Tim menciptakan website dengan tujuan untuk mempermudah arus pertukaran dan memperbarui informasi kepada sesama peneliti di CERN, tempat dia bekerja. Pada tanggal 30 April 1993, CERN menginformasikan bahwa WWW dapat digunakan gratis oleh semua orang. Website ditulis atau secara dinamik dikonversi menjadi HTML (Hyper Text Markup Language) dan diakses melalui sebuah program yang biasa disebut browser. Halaman web dapat dilihat dan diakses melalui jaringan komputer dan internet, sedangkan perangkat yang bisa mengakses website antara lain personal computer, laptop, PDA, maupun handphone atau tablet PC. Berdasarkan pengoperasiannya, secara mendasar website dibagi menjadi dua jenis website, pertama *static* dan website *dynamic*

Website static adalah website yang memiliki halaman front end, yaitu halaman yang dapat dilihat oleh pengunjung website. Karena fasilitas yang sangat terbatas, isi dari halaman website static bersifat tetap atau tidak berubah. Untuk mengganti sebuah halaman web static harus dilakukan secara manual dan harus mengganti semua kode – kode HTML yang merupakan unsur utama dari website tersebut. Website static biasa digunakan untuk membuat *company profile* (profil perusahaan), yaitu

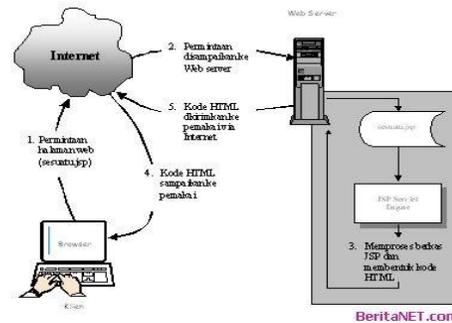
jenis website pengumuman berupa brosur online yang sangat sederhana dan tidak bisa diubah atau dimodifikasi

Website *dynamic* adalah website yang dapat diubah atau di-*update*. Dalam website *dynamic* biasanya terdapat dua halaman, yaitu halaman *front end* dan *back end*. Halaman *front end* merupakan halaman yang dapat diakses semua user, sedangkan halaman *back end* merupakan halaman yang hanya bisa diakses oleh admin yang bersangkutan. *Back end* biasa disebut dengan CMS (*Content Management System*) atau dalam WordPress biasa disebut dengan halaman *Dashboard*. Fungsi dari halaman *back end* adalah untuk mengatur *front end*. Sebagai contoh untuk pengaturan isi artikel, pengaturan tampilan *front end*, hingga tambahan untuk menghias *front end*. Beberapa contoh jenis CMS yang bisa digunakan adalah WordPress, Joomla, PHPNuke, dan lain-lain.

Bahasa Pemrograman PHP :

PHP (Personal Home Page) merupakan bahasa pemrograman aplikasi web dinamis yang bekerja pada sisi server dan memungkinkan interaksi dengan berbagai tipe RDBMS (Relation Database Management System) seperti Oracle, MySQL, IBM DB2, Microsoft SQL Server, PostgreSQL, InnoDB, SQLite, dan lain sebagainya. Dengan dukungan sistem terbuka yang sangat baik menjadikan PHP sebagai salah satu bahasa pemrograman web yang paling populer dan mendukung kemampuan cross-platform, diantaranya adalah mampu beroperasi pada lingkungan sistem Unix, Linux, Windows, dan Mac OS X.

Prinsip kerja pengaksesan berkas PHP dapat dilihat pada gambar 2.24. pengaksesan berkas PHP dimulai dengan permintaan (*request*) pengguna terhadap berkas PHP melalui protokol HTTP. Permintaan dikirim ke web server akan mengambil berkas PHP dan akan diproses (*interpret*) oleh PHP *Interpreter*. PHP *Interpreter* akan melakukan pemrosesan kode-kode PHP, dan kemudian membentuk blok-blok kode HTML yang terbentuk selanjutnya dikirim kembali ke web browser sehingga dapat diproses. Hasil proses berupa halaman web yang diminta oleh pengguna.



Gambar 2.24 Prinsip Kerja Pengaksesan Berkas PHP

MySQL :

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial

Profil Perusahaan :

PT. Galenika Farmasindo didirikan untuk menjadi wadah berkarya bagi profesi apoteker yang tergabung dalam naungan badan hukum (perusahaan) untuk mengaplikasikan segenap kemampuan dalam ber-inovasi dan ber-improvisasi dalam bidang yang menjadi kewenangan profesi apoteker dalam menjalankan praktek kefarmasian dengan berpegang pada konsep pengelolaan profesi sekaligus menerapkan aspek SYARIAH MUDHARABAH dalam menjalankan unit bisnis terkait.

Berdasar komitmen untuk adanya keterikatan antara unit bisnis yang di kelola maka PT. Galenika Farmasindo dibagi berdasarkan 3 rangkaian unit bisnis yang saling terkait antara lain : Produksi, Marketing, dan Distribusi, Outlet / Sarana pelayanan ke Farmasian. Visi kami ke depan berupa keras menjalankan seluruh unit bisnis ini dengan baik

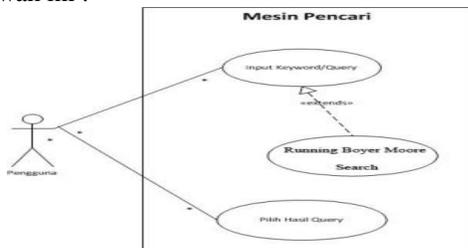
sehingga tujuan kami menuju perusahaan yang menjadi memiliki link kuat dari hulu ke hilir.

Identifikasi Masalah :

Permasalahan utama dalam sebuah Mesin Pencari (*search engine*) adalah menemukan informasi yang sesuai dengan *keyword* masukan dalam sebuah mesin pencari atau keterdekatan informasi antara *query* yang diberikan oleh pengguna terhadap mesin pencari tersebut. Tentunya semakin banyak kedekatan informasi yang ditemukan oleh mesin pencari dengan *query* yang dimasukkan oleh pengguna, maka mesin pencari tersebut semakin baik.

Use Case Diagram :

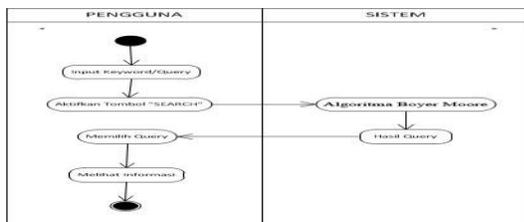
Use Case Diagram yang berfungsi sebagai gambaran tentang hubungan antara pengguna sistem (*user*) dengan sistem dapat dilihat dalam gambar dibawah ini :



Gambar 4.2. Use Case Diagram Mesin Pencari

Activity Diagram :

Dalam mendeskripsikan setiap langkah yang berjalan dalam sistem membutuhkan activity diagram, adapun *activity diagram* yang dihasilkan adalah sebagai berikut :

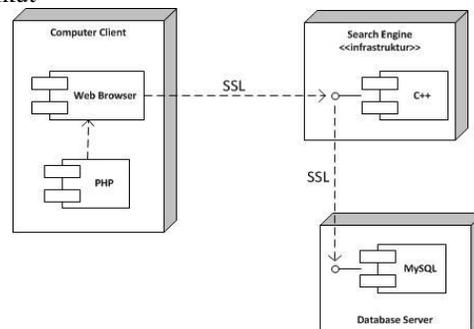


Activity Diagram diatas memberikan gambaran bahwa proses untuk menggunakan Mesin Pencari adalah dengan memasukkan sebuah *keyword* atau *query*, setelah itu pengguna dapat menekan tombol "SEARCH" untuk mulai mengaktifkan Mesin Pencari yang dibangun menggunakan algoritma *brute-force search*. Proses pencarian kecocokan

antara *keyword* atau *query* dengan hasilnya akan dapat terlihat dengan munculnya informasi-informasi dalam bentuk *link*, sehingga pengguna dapat memilih informasi-informasi tersebut dengan memilih salah satu dari informasi yang sesuai dengan kebutuhannya.

Deployment Diagram :

Dalam implementasi Mesin Pencari (*search engine*) ini, dibutuhkan sebuah deskripsi bagaimana hubungan antara perangkat keras dan perangkat lunak dibutuhkan agar aplikasi Mesin Pencari ini dapat berjalan sesuai dengan tujuannya. Adapun *deployment diagram* yang terbentuk adalah sebagai berikut



Gambar 4.4. Deployment Diagram

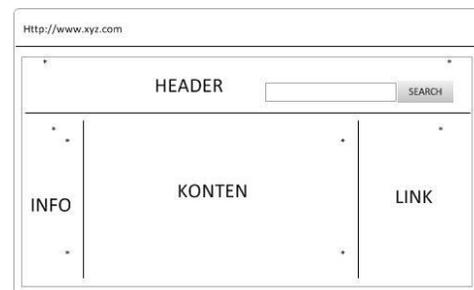
Perancangan Antarmuka Sistem :

Antarmuka (*interface*) yang dibutuhkan sesuai dengan hasil analisa diatas hanya ada 1 (satu) antarmuka, yaitu antarmuka Mesin Pencari sebagai berikut :



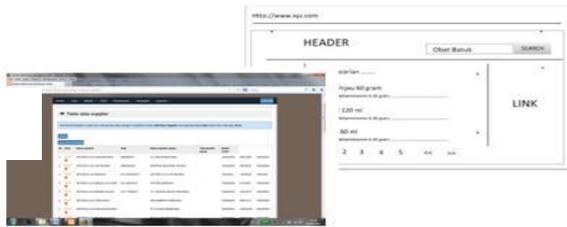
Gambar 4.5. Antarmuka Mesin Pencari

Mesin Pencari ini akan ditempelkan di website PT. XYZ yang telah selesai dibuat, dimana posisi Mesin Pencari ini akan diletakkan diposisi kanan atas, seperti gambar dibawah ini :



Gambar 4.6. Posisi Mesin Pencari di Web

Adapun tampilan hasil proses dari Mesin Pencari diharapkan seperti gambar dibawah ini

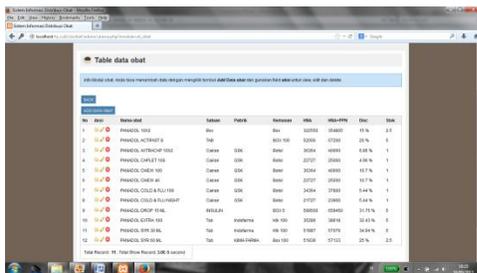


Gambar 4.7. Tampilan Hasil Mesin Pencari

Mendasarkan pada gambar 4.7. diatas, maka pengguna dapat memilih query mana yang paling sesuai dengan informasi yang dibutuhkan, sehingga dengan mendapatkan informasi ini, pengguna dapat tertarik untuk membeli produknya.

Interface :

Tampilan-tampilan antar muka yang terdapat dalam sistem, antara lain:



Pengujian kotak Putih (whitebox) :

Pengujian kotak hitam berfokus pada persyaratan fungsionalitas perangkat lunak. Berikut ini merupakan beberapa pengujian yang dilakukan dengan kotak putih (whitebox).

```

Sres = 0;
function boyer_moore(Spattern, Stext)
{
    Sn = strlen(Stext);
    Sm = strlen(Spattern);

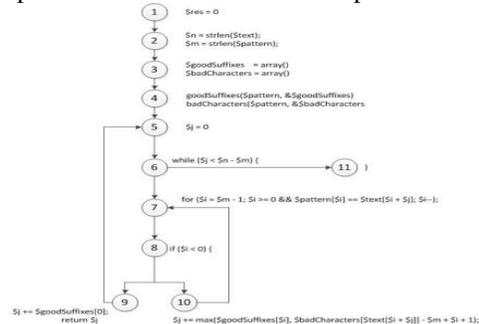
    SgoodSuffixes = array();
    SbadCharacters = array();

    goodSuffixes(Spattern, &SgoodSuffixes);
    badCharacters(Spattern, &SbadCharacters);

    Sj = 0;
    while (Sj < Sn - Sm) {
        for (Si = Sm - 1; Si >= 0 && Spattern[Si] == Stext[Si + Sj]; Si--);
        if (Si < 0) {
            // note that if the substring occurs more
            // than once into the text, the algorithm will
            // print out each position of the substring
            // echo Sj;
            Sj = SgoodSuffixes[0];
            tstatus Sj;
        } else {
            Sj = max(SgoodSuffixes[Si],
                SbadCharacters[Stext[Si + Sj]] - Sm + Si + 1);
        }
    }
}
    
```

Gambar 5.30 Pengujian kotak putih (Whitebox)

Step Pertama : Membuat Flow Graph dari code



Gambar 5.31 Flow Graph Algoritma Boyer Moore

Step Kedua : Menghitung Kompleksitas Skilomatis nya

Dari Flow Graph diatas terdapat :

1. Node = 11
2. Edge = 12
3. Predicate Node = 2

Sehingga Komplexitas Skilomatisnya adalah :

$$\begin{aligned}
 1. \quad V(G) &= E - N + 2 \\
 &= 12 - 11 + 2 \\
 &= 3 \\
 2. \quad V(G) &= P + 1 \\
 &= 2 + 1 \\
 &= 3
 \end{aligned}$$

Pengujian Kotak Hitam (blackbox) :

Pengujian kotak hitam berfokus pada persyaratan fungsionalitas perangkat lunak. Berikut ini merupakan beberapa pengujian yang dilakukan dengan kotak hitam (blackbox).

Button search	pilih	Menjalankan algoritma boyer moore dan mencari keyword yang di input oleh user lalu menampilkan.
---------------	-------	---

Tabel 5.1 Skenario Pengujian Tampilan Awal

Analisa Hasil :

Hasil analisa setelah melakukan pengujian secara keseluruhan perangkat lunak adalah:

1. Aplikasi search engine dengan menggunakan algoritma booyer moore sudah dapat berjalan dengan baik, dan dapat dibuktikan dengan menemukan keyword yang di input oleh user.

2. Mampu memfilter data yang dicari oleh user yang akan ditampilkan dengan menggunakan algoritma boyer moore.

Kesimpulan :

Search engine yang di letakan didalam web aplikasi telah selsai di bangun, hal ini terlihat dari beberapa faktor :

1. Search engine yang menggunakan algoritma boyer moore telah berhasil menemukan informasi yang sesuai dengan queri masukan untuk search engine tersebut.
2. Search engine ini di bangun menggunakan bahasa pemrograman php.

Search engine mampu mencari informasi dari database yang dibangun menggunakan MySQL

Saran :

Saran yang dapat diberikan untuk aplikasi ini, antara lain:

1. Penggunaan algoritma boyer moore untuk membangun search engine akan dapat lebih cepat mencari informasi sesuai queri masukan dengan memodifikasi menggunakan algoritma lainnya.
2. Search engine yang dibangun ini jauh akan lebih baik di ujicobakan dalam kasus pencarian informasi dengan jumlah data yang besar.

Pencarian informasi dalam jumlah data yang besar di mungkinkan untuk menambahkan algoritma lain untuk mendapatkan hasil yang lebih baik.

DAFTAR PUSTAKA

- [GJK,2002] Kowalski. J Gerald, 2002, "*Information Storage and Retrieval Systems Theory and Implementation*", Kluwer Academic Publishers, New York
- [FBW,1992] William B. Frakes dan Ricardo Baeza-Yates, 1992, "*Information Retrieval Data Structures & Algorithms*", Prentice-Hall, New Jersey
- [RP,1997] Pressman, Roger S., Ph.d., 1997. "*Rekayasa Perangkat Lunak*", McGraw – Hill. [SWA,2004] Ambler W. Scott, Miller Randy, 2004, "*Agile Model – Driven Development with UML 2.0*", Cambridge University Press, United State of America.
- [YUS,2002] Yusuf Norazah, Konsin Mohd Zarita, 2002, "*Struktur Data & Algoritma dalam Bahasa Turbo Pascal*", University Teknologi Malaysia, Kuala Lumpur - Malaysia
- [NIIT,2001] NIIT , 2001 , "*Introduction To Unified Modelling Language*".
- [SUP,2010] Supardi, Yuniar, "*Web My Profile dengan Joomla 1.5.x*", Jakarta: Elex Media Komputindo, 2010.
- [SUS,2007] Susanto, Budi , 2007 , "*Administrasi Pemrograman Database Oracle 10G XE*", PT Elex Media Komputindo, Jakarta.
- [HUD,2010]. Huda, Miftakhul, "*Membuat Aplikasi Database*", Jakarta: Elex Media Komputindo, 2011!