

1. Latar Belakang

Pertumbuhan *World Wide Web* yang sangat eksplosif membuat sukar menemukan informasi yang sesuai dengan keinginan pemakai. Terlalu banyak server dan halaman yang harus dilihat dan dilakukan secara *online* tetap merupakan tugas yang mengkonsumsi waktu. Hal inilah yang disebut masalah penemuan sumberdaya internet (*internet resource discovery problem*) (Sulastri & Zuliarso, 2010).

Di dalam dunia pendidikan baik itu di sekolah ataupun perkuliahan menunjang prestasi secara akademik dan mengasah kemampuan peserta didik setelah apa yang mereka dapatkan didalam kelas maka perlunya peserta didik tersebut untuk mengikuti berbagai macam lomba yang diadakan oleh banyak instansi baik itu instansi pendidikan, organisasi kemahasiswaan, dan lembaga-lembaga lainnya yang bekerjasama dalam bidang pendidikan, sehingga peserta didik dapat merasakan kompetisi diluar dari kelasnya serta menadapatkan mereka bisa mendapatkan pengalaman ataupun pembelajaran baru dan tentunya dengan mengikuti banyak kompetisi dengan mendapatkan hasil yang baik dapat meningkatkan akreditasi sekolah ataupun universitas tersebut.

Banyaknya *web* yang menyediakan informasi tentang perlombaan membuat peserta didik atau siapapun yang ingin mendapatkan informasi lomba sulit untuk menemukan apa saja *web* yang benar-benar meyediakan informasi tersebut dan juga sulitnya menemukan jenis lomba yang sesuai dengan apa yang mereka inginkan ataupun lomba yang sesuai dengan bidang yang mereka tekuni. Untuk mendapatkan informasi yang sesuai dengan mereka inginkan pastinya membutuhkan waktu yang tidak sebentar. Permasalahan tersebut bisa diselesaikan dengan menggunakan suatu metode *string matching* atau algoritma *Brute Force* yang diterapkan pada sistem computer.

Terdapat beberapa penelitian terkait dengan metode *string matching*, penelitian dengan menggunakan algoritma *Brute Force* telah dilakukan sebelumnya oleh Putri (2016). Pada penelitian tersebut menunjukkan bahwa algoritma *Brute Force* dengan algoritma *Optimal Mismatch* dapat digunakan dan diimplementasikan dengan baik untuk melakukan pencocokan *string* pada aplikasi kamu Bahasa Indoneisa-Sunda. Dimana algoritma *Brute Force* mampu mempersingkat waktu dengan memiliki rata-rata *running time* 14,991416

ms, dibandingkan algoritma *Optimal Mismatch* yang memiliki hasil rata-rata *running time* 31,46832 *ms*.

Penelitian selanjutnya menggunakan algoritma *Brute Force* dilakukan oleh Fenty et al (2014). Pada penelitian tersebut algoritma *Brute Force* diimplementasikan untuk mencocokkan pencarian *string* pada aplikasi kumpulan doa harian yang dikembangkan dengan mengikuti metode pengembangan system *Rapid Application Development* (RAD) yang terdiri dari tiga fase yaitu perencanaan syarat, *workshop design*, dan implementasi. Dimana algoritma *Brute Force* sangat cocok untuk diimplementasikan pada aplikasi tersebut.

Pada penelitian ini penulis mengajukan algoritma *Brute Force* yang merupakan salah satu metode untuk *String Matching*. *String Matching* adalah proses pencarian semua kemunculan *query* yang selanjutnya disebut *pattern* kedalam *string* yang lebih panjang atau teks (Santoso et al, 2012). Algoritma *Brute Force* adalah algoritma mencocokkan *pattern* dengan semua teks antara 0 dan $n-m$ untuk menemukan keberadaan *pattern* dalam teks (Santoso et al, 2012). Dalam mencocokkan *string*, terdapat istilah teks dan *pattern*. Teks merupakan kata yang akan dicari dan dicocokkan dengan *pattern*, sedangkan untuk *pattern* sendiri adalah merupakan kata yang kita inputkan untuk dicocokkan.

Berdasarkan latar belakang diatas, maka penulis mengajukan proposal penelitian dengan judul **SISTEM PENCARIAN INFORMASI LOMBA SECARA ONLINE MENGGUNAKAN ALGORITMA *BRUTE FORCE***.

2. Rumusan Masalah

Begitu banyaknya *web* penyedia informasi perlombaan secara *random* menyebabkan sulitnya peserta didik baik itu siswa dan mahasiswa ataupun orang yang ingin mencari informasi perlombaan sesuai dengan yang diinginkan. Saat ini proses pencarian informasi lomba ini dilakukan secara manual dengan menelusuri satu persatu *web* hasil *search engine* yang tersedia. Oleh sebab itu diperlukannya sebuah sistem untuk menghimpun data informasi lomba yang diambil dari beberapa *web* penyedia informasi lomba lalu dihimpun menjadi satu dalam aplikasi *web* dan diatur sesuai dengan kategorinya masing-masing,

sehingga pencarian informasi lomba yang sesuai dengan yang diinginkan dapat dilakukan secara efektif dan efisien.

3. Batasan Masalah

Pada penelitian ini peneliti membuat batasan masalah untuk mencegah meluasnya ruang lingkup permasalahan dalam penelitian ini. Adapun batasan masalah tersebut, diantaranya yaitu:

1. Data lomba didapat dari hasil *Web Crawling*.
2. Jumlah *sample website* yang digunakan sebagai sumber informasi sebanyak 5.
3. Sistem pencarian informasi lomba ini berbasis *Web*.
4. Kata kunci yang digunakan untuk pencarian berupa 1 frasa yang maksimal terdiri dari 6 kata dan minimal 2 kata.

4. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk pencarian informasi lomba secara *online* dengan menggunakan algoritma *Brute Force*.

5. Manfaat Penelitian

Adapun manfaat penelitian ini diantara lain yaitu:

1. Membantu peserta didik dan masyarakat umum untuk menemukan informasi lomba dengan sistem pencarian.
2. Memberi masukan untuk penelitian lain dalam bidang *string matching* dan *crawling*.

6. Metodologi

Adapun tahapan – tahapan yang akan dilakukan pada penelitian ini adalah :

1. Studi Literatur

Studi literatur dilakukan untuk mengumpulkan berbagai bahan referensi yang berkaitan dengan mengenai *string matching*, algoritma *Brute Force*, dan *crawling*, algoritma *Breadth First Search (BFS)*. Dari berbagai jurnal, artikel, buku dan beberapa sumber referensi lainnya.

2. Pengumpulan Data

Pada tahap ini mempelajari informasi terkait yang mendukung penelitian, penulis mengumpulkan data yang akan digunakan dalam penelitian

3. Analisis Permasalahan

Pada tahap ini penulis melakukan analisis permasalahan dari informasi yang didapat pada tahapan sebelumnya agar didapatkan metode yang tepat untuk mengatasi masalah dalam penelitian ini yaitu sistem pencarian informasi lomba secara *online* berbasis *web*.

4. Perancangan

Pada tahap ini dilakukan perancangan arsitektur sistem yang sesuai dengan analisis permasalahan yang telah dilakukan sebelumnya.

5. Implementasi

Pada tahap ini dilakukan implementasi dari analisis sesuai dengan perancangan system yang sudah dibuat

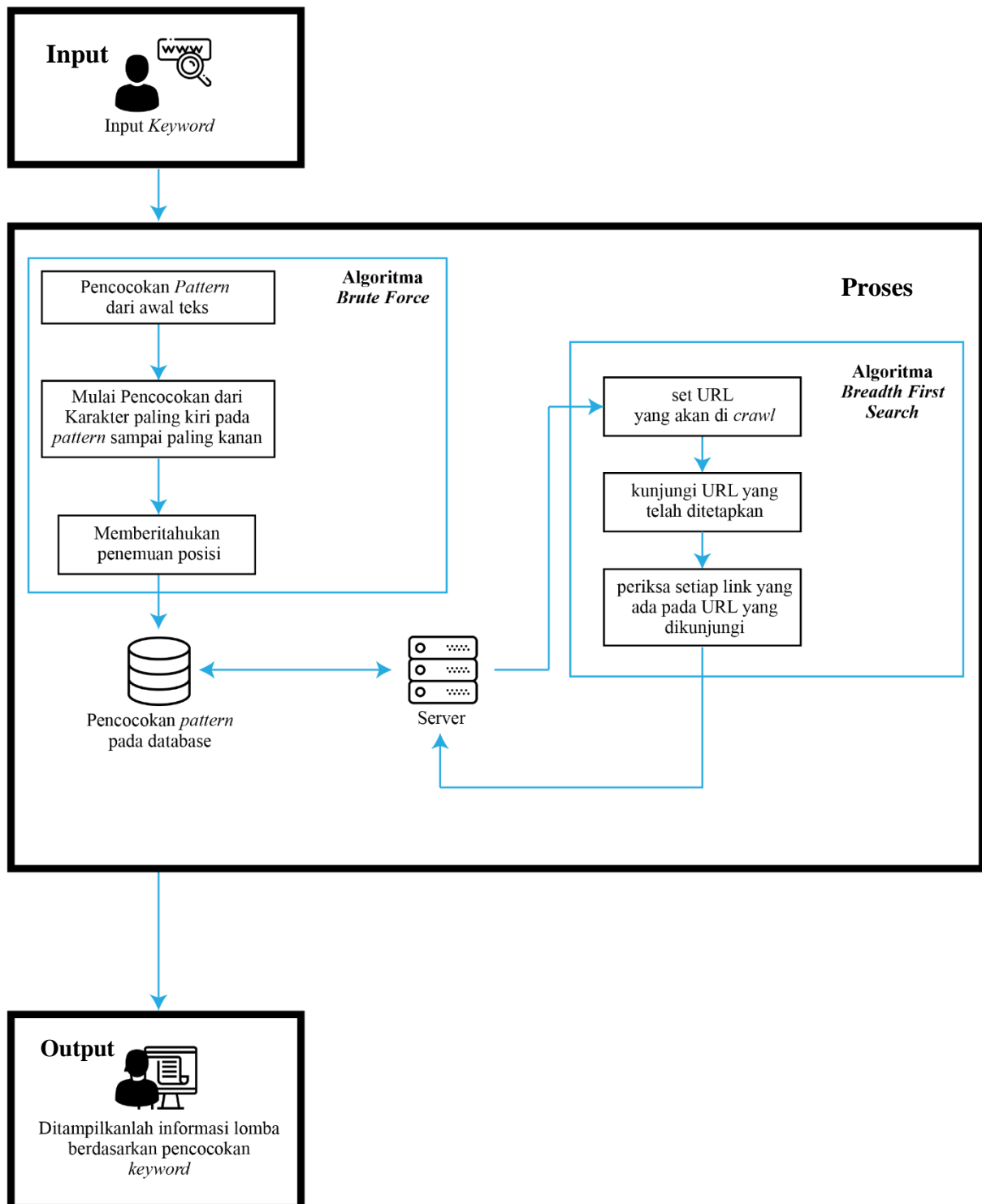
6. Pengujian

Pada tahap ini dilakukan pengujian terhadap *Brute Force* yang telah dibangun

7. Penyusunan Laporan

Pada tahap terakhir, penulis akan menyusun dokumentasi berupa laporan penelitian yang akan memaparkan hasil penelitian yang telah dilakukan.

Pada penelitian ini, arsitektur umum dari perancangan sistem yang akan dibuat adalah sebagai berikut:



Gambar 1. Arsitektur umum

Arsitektur umum pada penelitian yang akan dilakukan terbagi atas 3 yaitu input, proses, dan output :

1. Input

Kata yang diambil adalah dari kata kunci yang diinputkan oleh user pada *form* pencarian yang telah disediakan. Kata kunci tersebut berupa 1 frasa yangmana 1 frasa maksimal 6 kata dan minimal 2 kata.

2. Proses

Tahap ini merupakan tahap dimana kata kunci yang telah diinputkan diproses. Proses yang terjadi pada tahap ini terdiri dari dua bagian yaitu :,.

2.1 Pengolahan Kata Kunci

Pada tahap pengolahan kata kunci yang telah diinputkan sebelumnya oleh pengguna maka kata kunci tersebut diproses menggunakan Algoritma *Brute Force*. Tahap yang dilakukan algoritma tersebut saat melakukan pengolahan kata kunci adalah :

2.1.1 Algoritma *Brute Force* mulai mencocokkan *pattern* dari awal teks.

2.1.2 Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per karakter *pattern* dengan karakter pada teks yang bersesuaian, sampai salah satu kondisi berikut terpenuhi :

2.1.2.a Karakter di *pattern* dan teks yang dibandingkan tidak cocok.

2.1.2.b Semua karakter di *pattern* cocok. Kemudian algoritma akan memberitahukan penemuan pada posisi ini.

2.1.3 Algoritma kemudian terus menggeser *pattern* sebesar satu ke kanan, dan mengulangi langkah ke-2 sampai *pattern* berada di ujung teks

2.2 Pencocokan Kata Kunci pada Database

Setelah melalui tahap pengolahan kata kunci sebelumnya menggunakan algoritma *Brute Force* maka pada tahap ini kata kunci telah diproses akan melakukan pencocokan dengan data lomba yang terdapat pada database.

Database lomba tersebut didapat dari hasil *crawling* yang dilakukan dari tiga *website* yang telah kita tetapkan. Dalam proses *crawling* yang dilakukan menggunakan algoritma *Breadth First Search* (BFS). Algoritma BFS merupakan salah satu algoritma pencarian yang menguji tiap link pada sebuah halaman sebelum memproses kehalaman berikutnya. Jadi, algoritma ini menelusuri tiap link pada halaman pertama pada link pertama dan kemudian menelusuri tiap link pada halaman pertama pada link pertama dan begitu seterusnya sampai setiap level pada link telah dikunjungi.

3. Output

Setelah proses pencocokan kata kunci yang diinput selesai dilakukan, maka sistem dari penelitian ini mengeluarkan output berupa kategori judul lomba dan deskripsi lomba secara singkat yang sesuai atau mendekati dengan kata kunci yang diinputkan sebelumnya.

7. Landasan Teori

7.1 Sistem Informasi

Menurut Budi (2002), Sistem informasi merupakan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi.

7.2 String Matching

String Matching adalah proses pencarian kata (*pattern*) dalam sebuah teks. *String Matching* bisa digunakan untuk menemukan satu *pattern* yang pertama kali ditemukan, atau yang lebih umum menampilkan semua *pattern* yang dapat ditemukan dalam teks. Jenis *string matching* bermacam-macam, dibedakan menurut hasil yang diinginkan (Charras & Lecrog, 1997).

String Matching digunakan secara luas, seperti digunakan dalam pencarian bibliografi, aplikasi tanya jawab, pencocokan DNA, sistem operasi, pengenalan suara, pengolahan citra, aplikasi pengolahan teks, dan pengambilan informasi dari database (Hussain, 2010).

String Matching dirumuskan sebagai berikut :

$$x = x[0 \dots m-1] \dots \dots \dots (1)$$

$$y = y[0 \dots n-1] \dots \dots \dots (2)$$

Keterangan:

x adalah *pattern*

m adalah panjang *pattern*

y adalah teks

n adalah panjang teks

7.3 Web Crawler

Web Crawler adalah program yang menelusuri *World Wide Web* dengan cara yang metodis, otomatis, dan teratur. Istilah lain untuk *web crawler* adalah *ant*, *automatic indexer*, *bots*, *web spider* atau *web robots*.

Web Crawler adalah salah satu jenis bot atau agen perangkat lunak. Secara umum, proses *crawling* dimulai dengan *list URL* yang akan dikunjungi, disebut *seeds*. Kemudian *web crawler* akan mengunjungi URL tersebut satu per satu. Setiap *page URL* yang dikunjungi akan diidentifikasi apakah ada *hyperlink* didalamnya. Jika ada maka akan ditambahkan ke dalam *list URL* yang akan dikunjungi. Ini disebut *crawl frontier*. URL yang didapat dari *crawl frontier* akan dikunjungi secara rekursif dengan beberapa kebijakan tertentu (Shkapyuk et al., 2002).

7.4 Algoritma Brute Force

Menurut Charras & Lecroq (2004), algoritma *Brute Force* digunakan dalam pengecekan pada setiap kedudukan teks dari karakter pertama sampai karakter terakhir. Sesuai pengecekan karakter pertama terjadi, maka proses *shift* dilakukan dengan berpindah tepat satu posisi ke arah kanan (berpindah ke karakter kedua, ketiga dan seterusnya). Algoritma *Brute Force* tidak memerlukan tahap pra-proses. Selama tahap pencarian, perbandingan karakter pada teks dapat selesai pada posisi manapun. Tahap yang dilakukan pada algoritma *Brute Force* saat mencocokkan *string* adalah :

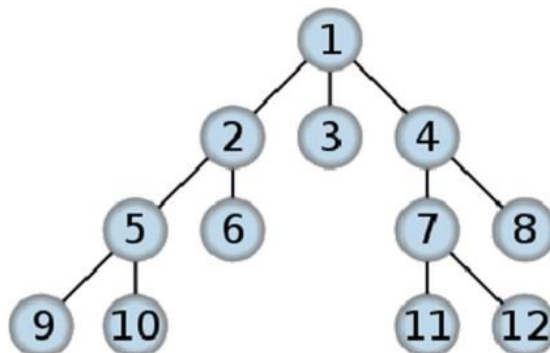
1. Algoritma *Brute Force* mulai mencocokkan *pattern* dari awal teks.

2. Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per karakter *pattern* dengan karakter pada teks yang bersesuaian, sampai salah satu kondisi berikut terpenuhi :
 - a Karakter di *pattern* dan di teks yang dibandingkan tidak cocok.
 - b Semua karakter di *pattern* cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.
3. Algoritma kemudian terus menggeser *pattern* sebesar satu ke kanan, dan mengulangi langkah ke-2 sampai *pattern* berada di ujung teks.

7.5 Algoritma *Breadth First Search* (BFS)

Algoritma *Breadth First Search* atau biasa dikenal dengan nama algoritma pencarian melebar merupakan sebuah teknik umum yang digunakan untuk melakukan traversal pada graf.

Dalam Algoritma BFS dilakukan secara sistematis, artinya biasanya dikunjungi dalam satu arah, misalkan dari simpul paling kiri ke simpul paling kanan. Penelusuran simpul juga dilakukan dalam satu arah terlebih dahulu sebelum mengunjungi simpul pada arah yang lebih tinggi.



Gambar 2. Penelusuran BFS Pencarian Kata

Dalam penerapannya, algoritma BFS memerlukan matriks ketetanggaan $A = [a_{ij}]$ yang berukuran $n \times n$, antrean q untuk menyimpan simpul yang telah dikunjungi, dan table Boolean dikunjungi.

8. Penelitian Terdahulu

Penelitian dengan menggunakan algoritma *Brute Force* telah dilakukan sebelumnya oleh Putri (2016). Pada penelitian tersebut menunjukkan bahwa algoritma *Brute Force* dengan algoritma *Optimal Mismatch* dapat digunakan dan diimplementasikan dengan baik untuk melakukan pencocokan *string* pada aplikasi kamus Bahasa Indonesia-Sunda. Dimana algoritma *Brute Force* mampu mempersingkat waktu dengan memiliki rata-rata *running time* 14,991416 ms, dibandingkan algoritma *Optimal Mismatch* yang memiliki hasil rata-rata *running time* 31,46832 ms.

Penelitian selanjutnya menggunakan algoritma *Brute Force* dilakukan oleh Fenty et al (2014). Pada penelitian tersebut algoritma *Brute Force* diimplementasikan untuk mencocokkan pencarian *string* pada aplikasi kumpulan doa harian yang dikembangkan dengan mengikuti metode pengembangan sistem *Rapid Application Development* (RAD) yang terdiri dari tiga fase yaitu perencanaan syarat, *workshop design*, dan implementasi. Dimana algoritma *Brute Force* sangat cocok untuk diimplementasikan pada aplikasi tersebut.

Pada penelitian selanjutnya menggunakan algoritma *Brute Force* dilakukan oleh Mohsen et al (2017). Pada penelitian tersebut algoritma *Brute Force* diimplementasikan pada *Smart Street Lighting* (SmSL). SmSL ini dibuat untuk dijadikan sebagai aplikasi hirarki dimana aplikasi tersebut dapat menentukan penjadwalan, mematikan, dan mengatur tingkat keredupan setiap kutub berdasarkan yang diusulkan dari ketentuan setiap hari dan juga menransmisikan titik pengontrol untuk mengontrol kutub lokal melalui *Power Line Communications* (PLC). Dimana algoritma *Brute Force* dapat diimplementasikan untuk menyelesaikan masalah seperti meminimalisasi energi dalam kelompok pencahayaan. Algoritma *Brute Force* secara umum sebagai pemecah masalah yang terdiri dari penghitungan secara sistematis semua kandidat apakah setiap kandidat tersebut terdapat masalah.

Penelitian selanjutnya menggunakan algoritma *Brute Force* dilakukan oleh Seongyong et al (2009). Pada penelitian tersebut algoritma *Brute Force* diimplementasikan untuk memeriksa muatan paket dari jaringan dan mendeteksi dari serangan jaringan jahat yang terstruktur pada sebuah perangkat keras yang terhubung dengan jaringan. NIDS (*Network*

Intrusion Detection System) adalah sebuah pencocokan yang memeriksa konten dalam muatan suatu jaringan, namun kecepatan kabel jaringan dan implementasi dengan biaya yang minimal diperlukan pendukung *string matching* yaitu menggunakan algoritma *Brute Force*. Arsitektur algoritma *Brute Force* sangat efisien untuk diimplementasikan pada sebuah aplikasi atau perangkat keras

Penelitian selanjutnya menggunakan algoritma *Brute Force* dilakukan oleh Abdeen, A.R (2011). Pada penelitian tersebut algoritma *Brute Force* diimplementasikan untuk meningkatkan pencarian *string*.

Penelitian terdahulu yang telah dipaparkan akan diuraikan secara singkat pada tabel berikut:

Tabel 1. Penelitian Terdahulu

No	Tahun	Peneliti	Judul Penelitian
1	2017	Mohsen Mahoor et al	A Hierarchical Smart Street Lighting System with Brute Force Energy Optimization
2	2009	Seongyong et al.	A Hardware-Efficient Multi-Character String Matching Architecture Using Brute-Force Algorithm
3	2016	Putri et al.	Analisis dan Implementasi Algoritma Brute-Force Dengan Optimal Mismatch pada Aplikasi Kamus Bahasa Indonesia
4	2011	Abdeen, A.R.	An Algorithm for Searching String Based on Brute Force Algorithm

5	2014	Fenty et al.	Implementasi Algoritma Brute Force dan Fitur Location Based Service (LBS) Pada Aplikasi Kumpulan Doa Harian Berbasis Android
---	------	--------------	--

DAFTAR PUSTAKA

- Abdeen, A.R. 2011. *An Algorithm for String Searching Based on Brute Force Algorithm*. *IJCSNS International Journal Of Computer Science and Network Security*. 11(7): 24-27.
- Arifin, A. Z., Budianto, Lili, S.. 2003. Perancangan dan Pembuatan Perangkat Lunak Penelusur Web (Web Crawler) Menggunakan Algoritma Pagerank. Teknik Informatika. Institut Teknologi Sepuluh November. Surabaya.
- Astuti W . 2007. Analisis *String Matching* pada judul skripsi dengan Algoritma *Knuth-Morris Praat* (KMP). *ILKOM Jurnal Ilmiah* 9.
- Astuti, K. D., Yanuar Firdaus A.W., Warih Maharani. 2009. *Analisis Performansi Web Crawler Berbasis Backlink, Breadth First Search, dan Pagerank*. Bandung. Universitas Telkom.
- Charras, C. & Lecroq, T. 1997. *Handbook of Exact String – Matching Algorithms*. Prancis.
- Charras, C. & Lecroq, T. 2004. *Handbook of Exact String – Matching Algorithms*. King College Publications: Guernsey.
- Choudhary, Suryakant, Mustafa Emre Dincturk, Seyed M. Mirtaheri, Ali Moosavi, Gregor von Bochmann, Guy-Vincent Jourdan, Iosif Viorel Onut. 2012. *Crawling Rich Internet Applications: The State of the Art*. CASCON.
- Mirtaheri, Sayed M, Mustafa Emre Dincturk, Salman Hooshmand, Gregor V. Bochmann, Guy-Vincent Jourdan. 2013. *A Brief History of Web Crawlers*. CASCON.
- Munir, R. 2004. *Strategi Algoritmik, Algoritma Pencarian String (String Matching)*. Departemen Teknik Informatika Institut Teknologi Bandung. Bandung.

- Putri, Dwi S. 2016. *Analisis dan Implementasi Algoritma Brute Force Dengan Optimal Mismatch pada Aplikasi Kamus Bahasa Indonesia - Sunda*. Skripsi. Universitas Sumatera Utara
- Sarno, Riyanto. 2012. *Semantic Search* Pencarian Berdasarkan Konten. *JURNAL SISFOTEK GLOBAL*. Andi. Yogyakarta.
- Sulastri & Eri Zuliarso. 2010. Aplikasi *Web Crawler* Berdasarkan *Breadth First Sarch* dan *Back-Link*. Jurnal Teknologi Informasi DINAMIK. Fakultas Teknologi Informasi, Unibersitas Stikubank Semarang.
- Sutedjo, Budi Dharma Oetomo. 2002. *Perancangan & Pengembangan Sistem Informasi*. Yogyakarta: Penerbit Andi.