

Tugas Besar III IF2211 Strategi Algoritma

Pencarian Berita pada *News Aggregator* dengan Algoritma Pencocokan String

Batas pengumpulan : Jumat, 28 April 2017 jam 10 pagi.

Arsip pengumpulan : - CD yang berisi *Source* dan *Exe* program disertai *readme.txt*
- Laporan (*hard copy*)

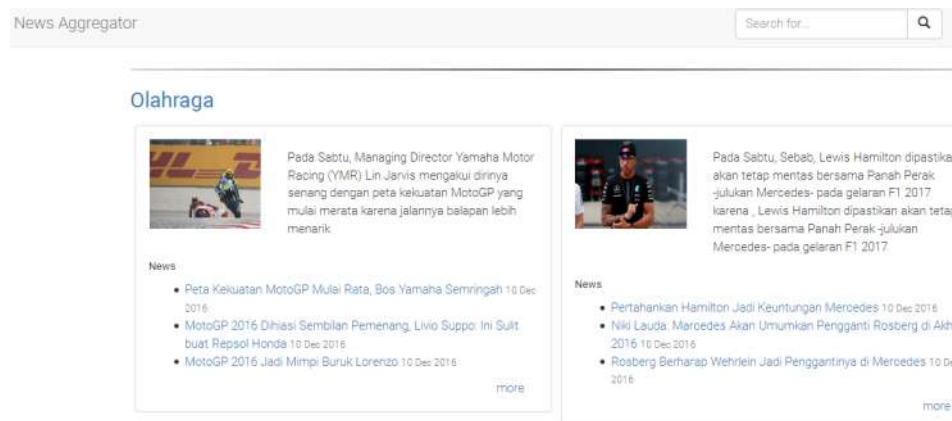
Tempat pengumpulan : Di atas loker Lab IRK

Deskripsi tugas :

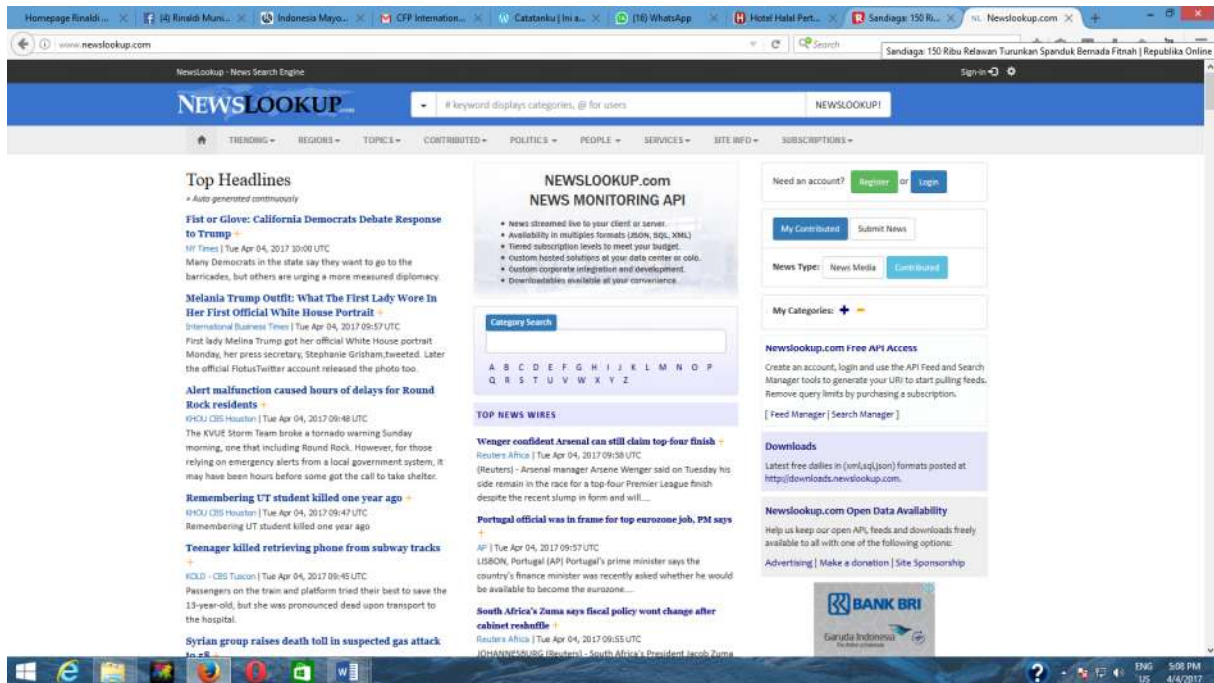
Sistem agregasi berita (*news aggregator*) dikembangkan untuk membantu pembaca berita dengan mengumpulkan informasi berita dari berbagai sumber dan menyajikannya dalam satu tempat. Dengan sistem ini, pembaca tidak perlu mencari berita sendiri, dan aplikasi dapat mengambil berita sesuai kebutuhan dari pembaca (Lasica, 2003). Pencarian berita merupakan salah satu fitur yang terdapat pada sistem ini.



Gambar 1. News aggregator dari Google



Gambar 2. Prototipe News aggregator dari Lab GAIB ITB



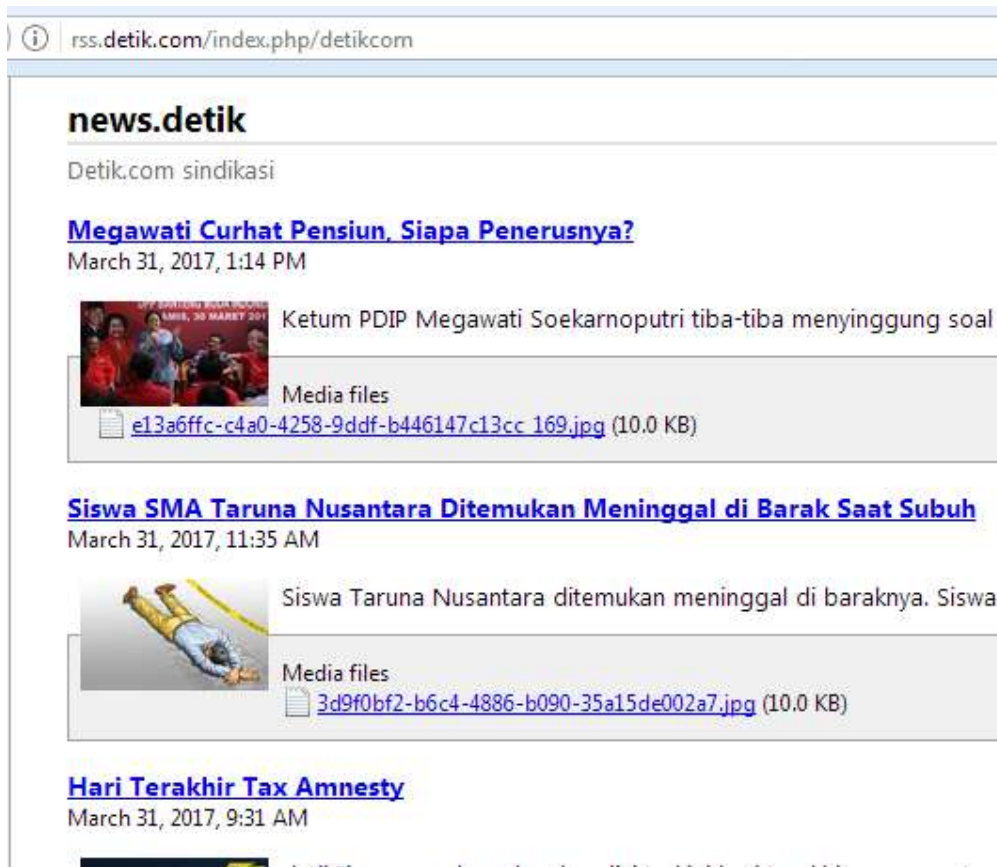
Gambar 3. News aggregator dari situs www.newslookup.com

Algoritma pencocokan *string (pattern)* Knuth-Morris-Pratt (KMP) dan Algoritma Boyer-Moore merupakan algoritma yang lebih baik daripada *brute force*. Pada Tugas Besar III kali ini Anda diminta membuat aplikasi sederhana pencarian berita pada *news aggregator* dengan kedua algoritma tersebut, plus menggunakan *regular expression (regex)*. Teks yang akan Anda proses adalah kumpulan berita berbahasa Indonesia. Pengguna aplikasi ini akan memberikan masukan berupa keyword pencarian, dan menghasilkan daftar berita yang diurut berdasarkan tanggal berita.

Pencocokan *string* yang anda buat adalah *exact matching* (untuk KMP dan BM) jadi artikel berita yang diproses mengandung string yang tepat sama dengan *keyword* yang dimasukkan oleh pengguna. Sedangkan bila menggunakan *regex* maka tidak selalu *exact matching*. Pencarian juga tidak bersifat *case sensitive*, jadi huruf besar dan huruf kecil dianggap sama (hal ini dapat dilakukan dengan menganggap seluruh karakter di dalam *pattern* dan teks sebagai huruf kecil semua atau huruf kapital semua).

Kumpulan berita diambil secara otomatis menggunakan *crawler* berbasis *RSS (rich site summary* atau *really simple syndication)* dari situs berita daring berbahasa Indonesia. Saat membaca *RSS* dengan *XML parser*, informasi yang dibutuhkan berupa judul, tanggal berita, dan *URL* berita. Berikut daftar *RSS* yang dapat digunakan:

<http://rss.detik.com/index.php/detikcom>
<http://tempo.co/rss/terkini>
<http://rss.vivanews.com/get/all>
<http://www.antaraneews.com/rss/terkini>



Gambar 4. Isi RSS detik.com jika dilihat dari browser

Untuk setiap *URL*, unduh artikelnya dan lakukan *parsing*. Artikel berupa file HTML dan tidak hanya mengandung konten berita, tetapi masih mengandung *header*, *footer*, iklan, dan tambahan informasi pada situs berita tersebut. Untuk itu, dilakukan *parsing* HTML untuk mendapatkan hanya teks konten berita dan foto yang terkait berita tersebut. Salah satu library *html parser* yang dapat digunakan misalnya adalah <https://jsoup.org/> untuk Java, *RSS parser* untuk .NET (<http://stackoverflow.com/questions/684507/rss-parser-in-net>) dan *XML parser* dalam Bahasa C# (<http://stackoverflow.com/questions/642293/how-do-i-read-and-parse-an-xml-file-in-c>),

Spesifikasi program :

1. Aplikasi pencarian berita pada *news aggregator* yang anda buat merupakan aplikasi berbasis web yang menerima *keyword* pencarian, misalnya “Informatika ITB”. Tampilan antarmuka pengguna-komputer kira-kira seperti Gambar 5 di bawah ini:

My Search App

Keyword:

Algoritma :

- ☐ Boyer-Moore
- ☐ KMP
- ☐ Regex

1. <url1>
<Kalimat yang mengandung keyword>

2. <url2>
<Kalimat yang mengandung keyword>

...

[Perihal](#)

Gambar 5. Contoh antarmuka aplikasi sederhana pencarian berita

Perihal : link ke halaman tentang program dan pembuatnya

Anda dapat menambahkan menu lainnya, gambar, logo, dan sebagainya

2. Pencarian berita menggunakan hasil implementasi algoritma KMP, Boyer-Moore, dan *Regex* dengan menggunakan Bahasa C#. Pencocokan string dilakukan tidak hanya pada judul berita, tetapi juga isi di dalam berita tersebut.
3. Kumpulan berita yang akan diproses diambil secara otomatis berbasis RSS.

Data Uji

Data uji yang digunakan dapat anda tentukan sendiri, minimal terdapat 20 berita yang diproses. Berita dapat berbahasa Indonesia atau Inggris.

Lain – lain :

1. Anda dapat menambahkan fitur fungsional lain yang menunjang program yang anda buat (unsur kreatifitas diperbolehkan/dianjurkan).

2. Program berbasis *web* dan dapat dikembangkan dengan salah satu kakas: PHP, JSP (*Java Server Pages*), atau ASP.
3. Program implementasi Boyer-Moore dan KMP menggunakan Bahasa C# dan dapat dikembangkan dengan kakas C# dalam *platform* .NET.
4. Akses konten berita dapat menggunakan library *XML* parser dan *html parser* yang ada.
5. Tugas dikerjakan per kelompok dengan jumlah anggota maksimal 3 orang dan tidak boleh sama dengan anggota kelompok sebelumnya.
6. Program harus modular dan mengandung komentar yang jelas.
7. Mahasiswa harus membuat program sendiri kecuali library *XML* dan *HTML parser*, tetapi belajar dari contoh-contoh program serupa yang sudah ada tidak dilarang (tidak boleh mengkopi *source code* dari program orang lain).
8. Keterlambatan pengumpulan akan mengurangi nilai.
9. Program disimpan di dalam *folder* StrAlgo3-xxxxx. Lima digit terakhir adalah NIM anggota terkecil. Didalam *folder* tersebut terdapat tiga folder bin, src dan doc yang masing-masing berisi :
 - a. Folder *bin* berisi *executable file* (exe)
 - b. Folder *src* berisi *source code* dari program
 - c. Folder *test* berisi data uji.
 - d. Folder *doc* berisi dokumentasi program dan *readme*

Folder ini disimpan dalam bentuk CD untuk dikumpulkan bersama berkas laporan dimasukan kedalam amplop coklat.
10. Semua pertanyaan menyangkut tugas ini harus dikomunikasikan melalui milis agar dapat dicermati oleh semua peserta kuliah IF2211 (milis IF2211@students.if.itb.ac.id).
11. Demo program akan dilaksanakan pada tanggal yang dimumkan oleh asisten. Peserta mengisi jadwal demo yang disediakan pada saat pengumpulan tugas.
12. Tiap anggota harus memahami proses pembuatan program, karena akan ada pertanyaan-pertanyaan yang harus dijawab per individu.
13. Pada saat demo, asisten akan memanggil per kelompok sesuai jadwal yang telah diisi sebelumnya. Kelompok yang tidak berkepentingan dilarang masuk. Demo dilakukan di Lab IRK.
14. **Bonus 1** (nilai maksimal 10): jika judul berita mengandung kata tempat yang menunjukkan lokasi, maka gunakan *Google Map API* untuk menuju ke lokasi yang dimaksudkan.
15. **Bonus 2** (nilai maksimal 10): Setiap kelompok membuat video aplikasi yang mereka buat kemudian mengunggahnya ke *Youtube*. Pada waktu demo aplikasi di depan asisten, mahasiswa mengakses video *Youtube* tersebut dan memutarnya di depan asisten sebelum memulai demo.

Isi laporan :

Cover: *Cover* laporan ada foto anggota kelompok (foto bertiga) yang paling keren. Foto ini menggantikan logo “gajah” ganessa.

Bab 1: Deskripsi masalah (dapat meng-copy paste file tugas ini)

Bab 2: Dasar teori (ringkas)

Bab 3: Analisis Pemecahan Masalah. Langkah-langkah pemecahan masalah ada di sini beserta contoh ilustrasi.

Bab 4: Implementasi dan pengujian. Bab ini berisi:

- a. Spesifikasi teknis program, termasuk di dalamnya struktur data atau kelas objek yang didefinisikan, fungsi dan prosedur (*header* fungsi dan prosedur saja, tidak perlu *source code*), antarmuka, dan lain-lain yang dianggap perlu.
- b. Eksperimen/pengujian dengan contoh-contoh *query*.
- c. Analisis hasil pengujian.

Bab 5: Kesimpulan dan saran (hasil yang dicapai, saran pengembangan).

Tuliskan juga referensi (buku, web), yang dipakai/diacu di dalam Daftar Referensi.

Keterangan laporan :

1. Laporan ditulis dalam bahasa Indonesia yang baik dan benar, tidak perlu panjang tetapi tepat sasaran dan jelas.
2. Laporan tidak perlu memakai *cover* mika dan dijilid. Cukup dibuat agar laporan tidak akan tercecer bila dibaca.
3. Laporan boleh menggunakan kertas rius, boleh bolak-balik, boleh dalam satu halaman kertas terdapat dua halaman tulisan asalkan masih terbaca.
4. Identitas per halaman harus jelas (misalnya : halaman, kode kuliah).

Penilaian :

1. Kebenaran program (40%) : program mampu berjalan sesuai dengan spesifikasi yang diberikan.
2. Demo – pemahaman Anda dalam pembuatan program (30%)
3. Laporan (20%)
4. *Interface, feature-feature* program, dan unsur kreativitas (20%)

-selamat mengerjakan-