

ARSITEKTUR *SEARCH ENGINE*

MATERI PERKULIAHAN : *INFORMATION RETRIEVAL SYSTEM KE-6*

Disusun Oleh :

Nama : Nuning Kurniasih, S.Sos., M.Hum.

NIP. 197606252000122001

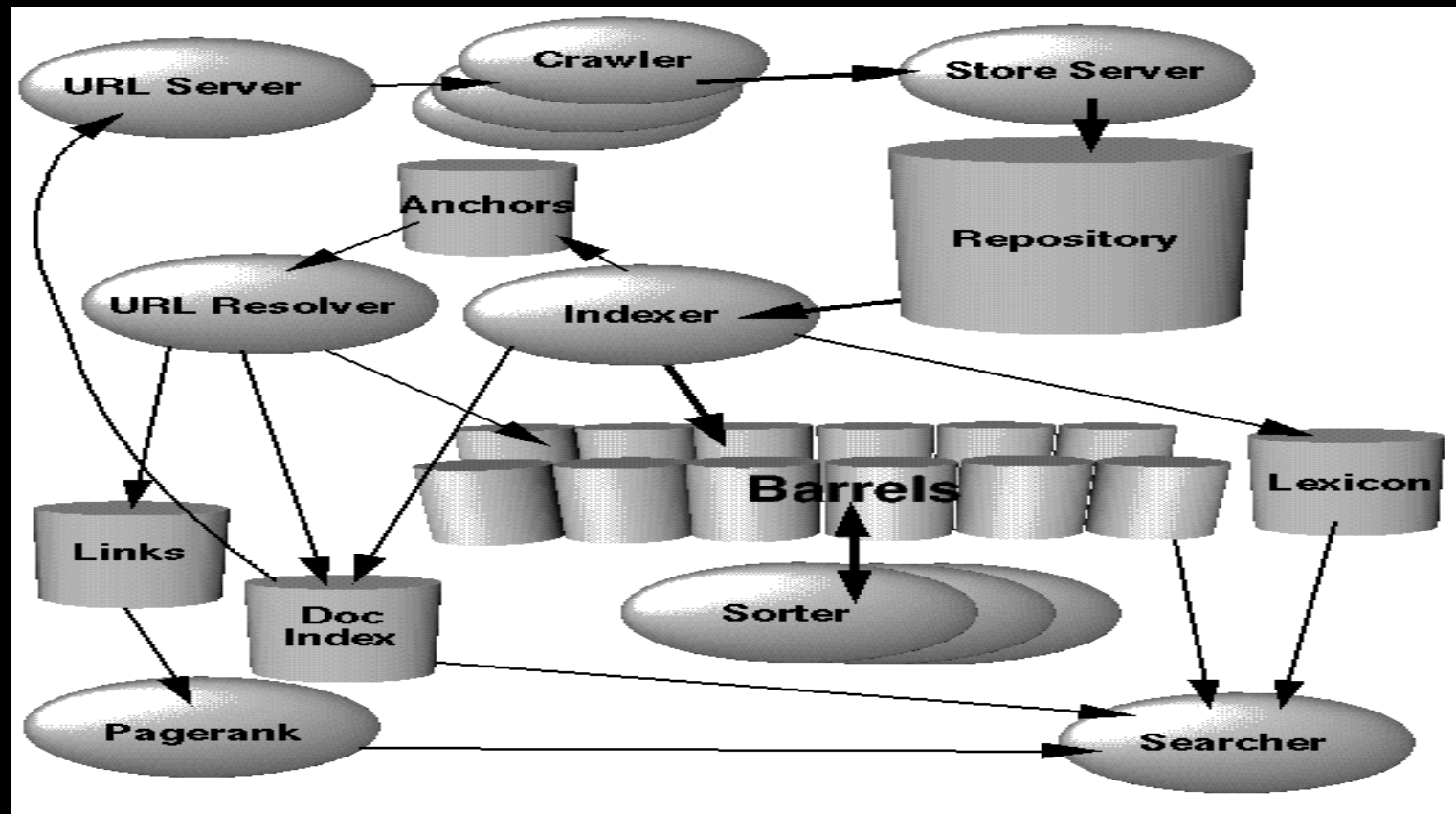
Departemen Ilmu Informasi dan Perpustakaan

Fakultas Ilmu Komunikasi

Universitas Padjadjaran

Ditulis Pertama Tahun 2005, Revisi Januari 2014

ARSITEKTUR GOOGLE



Sergey Brin dan Lawrence Page.

ARSITEKTUR GOOGLE

- *Crawler* : sebuah program (dikenal sebagai jaring atau robot) yang mengunjungi *web sites* dan membaca halaman dan informasi lainnya untuk membuat entri-entri dalam indeks search engine.
- *Anchor* (Jangkar) dalam HTML ditandai dengan istilah, frase, gambar, atau objek informasi lainnya sebagai bagian dari:
 - a. Target *link* hipertext dalam dokumen, atau
 - b. Sebuah referensi (*link* yang dapat dipilih) untuk sebuah target.
- *Pagerank* : alogaritma yang dipergunakan oleh google untuk meranking *website* di dalam *search engine*-nya.

STRUKTUR DATA UTAMA

1. *Big Files*

BigFiles adalah file virtual yang mencakup sejumlah sistem *file* yang tersimpan secara otomatis. Paket *BigFile* juga menangani alokasi dan dealokasi deskriptor *file* serta mendukung pilihan kompresi.

2. *Repository*

Repository berisi HTML penuh pada setiap halaman web. Setiap halaman dikompres. Pemilihan teknik kompresi merupakan pertukaran diantara kecepatan dan rasio kompresi.

3. *Document Index*

Indeks dokumen menjaga informasi setiap dokumen dengan menggunakan ISAM (*Index sequential access mode*), diurutkan berdasarkan identitas (ID) dokumen.

STRUKTUR DATA UTAMA

4. *Lexicon*

Ada beberapa jenis *lexicon*. Saat ini *lexicon* berisi lebih dari 14 juta kata tanpa memuat kata-kata yang jarang dipergunakan dan disimpan di dalam memori mesin. Adapun implementasinya dengan cara :

- a. Mendaftar kata-kata.
- b. Membagi poin-poin dalam tabel.

5. *Hit List*

Daftar *hit* berhubungan dengan daftar kejadian pada sebuah kata tertentu dalam sebuah dokumen, termasuk informasi posisi, huruf dan kapital. Ada 2 macam hit :

- a. *Fancy hits*, termasuk apa yang terjadi pada url, judul, rangka teks atau meta tag.
- b. *Plain hits*, berisi kapitalisasi bit, ukuran huruf, 12 bit posisi kata dalam sebuah dokumen.

STRUKTUR DATA UTAMA

6. *Forward Index*

Forward index sebenarnya telah tersimpan. Ini disimpan di sejumlah kotak berdasarkan isi dokumen tertentu dan diurutkan berdasarkan ID dokumen.

7. *Inverted Index*

Berisi kotak yang sama dengan *forward index*, kecuali yang telah diproses oleh penyortir.

Sergey Brin dan Lawrence Page

PROSES KERJA *SEARCH ENGINE*

1. Akuisisi Teks :
2. Transformasi Teks
3. Penciptaan Indeks
4. Antarmuka Pengguna
5. Ranking
6. Evaluasi

CartereGe, 2011.

PROSES KERJA *SEARCH ENGINE*

1. Akuisisi Teks :

- a. *Crawler* mengidentifikasi dan memperoleh dokumen untuk *search engine* serta menjaganya tetap *up - to - date*.
- b. *Feeds : Real-time* aliran dokumen, seperti *web feeds* untuk berita, blog, video, radio, tv dengan RSS sebagai standar yang dapat menyediakan dokumen XML baru untuk *search engine*.
- c. Konversi dokumen yang berisi teks dan format metadata seperti HTML, XML, Word, PDF, dll ke XML, mengkonversi encoding teks untuk bahasa-bahasa yang berbeda dengan menggunakan standar pengkodean seperti UTF - 8

PROSES KERJA *SEARCH ENGINE*

1. Akuisisi Teks :

d. Penyimpanan data dokumen

- Penyimpanan teks, metadata (informasi mengenai dokumen seperti jenis dan tanggal penciptaan) dan konten lainnya seperti link dan jangkar teks.
- Menyediakan akses yang cepat terhadap isi dokumen untuk komponen *search engine* seperti daftar hasil generasi.
- Menggunakan Sistem *Database Realtime*, lebih sederhana, lebih efisien dalam sistem penyimpanan untuk dokumen dalam jumlah besar.

PROSES KERJA SEARCH ENGINE

2. Transformasi Teks :

a. Parser

Parser adalah sebuah program yang menerima input dalam bentuk instruksi sumber program yang berurutan, komen-komen online interaktif, tag *markup* atau antar muka lainnya dan membaginya ke dalam kata benda untuk objek atau kata kerja untuk metode, dll.

Memproses token teks dalam dokumen secara berurutan untuk mengenali elemen struktur seperti judul, link, tajuk, dsb.

Tokenizer mengenali kata di dalam teks. Isu-isu dalam kapitalisasi, tanda hubung, apostrof, karakter non alfa, pemisahah, dll, harus diperhatikan.

Bahasa Markup seperti HTML, XML, biasanya menggunakan struktur yang spesifik, tag menggunakan elemen dokumen tertentu, parser dokumen menggunakan sintaks dari Bahasa markup atau format lainnya untuk mengidentifikasi struktur.

PROSES KERJA SEARCH ENGINE

2. Transformasi Teks :

- b. *Stopping*, dengan menghilangkan kata-kata seperti “and”, “or”, “the”, “in”
- c. *Stemming*, dengan mengelompokkan kata-kata yang berasal dari asal kata yang sama, seperti “computer”, “computers”, “computing”, “compute”
- d. *Analisis Link*, dengan membuat penggunaan *link* dan jangkar teks dalam halaman web dan menganalisis *link* untuk mengidentifikasi popularitas dan informasi komunitas, seperti dengan *PageRank*. Jangkar teks dapat secara signifikan meningkatkan representasi halaman yang ditunjuk oleh link serta berdampak secara signifikan pada penelusuran web.

PROSES KERJA *SEARCH ENGINE*

2. Transformasi Teks :

e. Mengekstrak Informasi

Mengidentifikasi kelas-kelas dari indeks *term* yang penting bagi beberapa aplikasi, antara lain diberi nama sebagai pengenalan identitas kelas seperti orang, lokasi, perusahaan, tanggal, dsb.

f. *Classifier*

Mengidentifikasi kelas, berhubungan dengan metadata dari dokumen, seperti memberikan label dokumen, topik, tingkat membaca, sentimen, genre.

PROSES KERJA *SEARCH ENGINE*

3. Penciptaan Indeks :

- a. Statistik dokumen, mengumpulkan jumlah dan posisi kata serta fitur lainnya dengan menggunakan ranking alogaritma.
- b. Pembobotan (*Weighting*), bobot komputer untuk indeks *term* menggunakan ranking alogaritma, seperti pembobotan *tf.idf* (kombinasi Antara frekuensi *term* dalam dokumen dan kebalikan frekuensi dokumen di dalam koleksi).

PROSES KERJA *SEARCH ENGINE*

3. Penciptaan Indeks :

- c. Inversi, sebagai inti dari proses pengindeksan dan konversi informasi *term*-dokumen ke *term*-dokumen untuk pengindeksan.
- d. Distribusi Indeks. Mendistribusikan dokumen, *term* dan replika. melalui sejumlah computer dan/atau site.

PROSES KERJA *SEARCH ENGINE*

4. Antarmuka Pengguna :

a. Input Query,

Dengan menyediakan antarmuka dan parse untuk Bahasa query. Bahasa query digunakan untuk mendeskripsikan query yang lebih kompleks dan hasil transformasi query, seperti bahasa query Boolean, Indri dan Galago, mirip dengan Bahasa SQL yang digunakan di dalam aplikasi database. Bahasa query IR mencakup isi dan spesifikasi struktur, tetapi fokus pada isi.

PROSES KERJA *SEARCH ENGINE*

4. Antarmuka Pengguna :

b. Transformasi Query

Meningkatkan penanda query baik sebelum maupun setelah penelusuran, mencakup teknik transformasi informasi teks yang digunakan untuk dokumen.

Memeriksa ejaan dan saran query untuk menyediakan alternatif bagi query asli. Perluasan query dan relevansi umpan balik memodifikasi query asli dengan *term* tambahan.

c. *Output* Hasil

Mengkonstruksi tampilan dokumen ter ranking untuk sebuah query. Menghasilkan bagian terkecil untuk memperlihatkan bagaimana query cocok dengan dokumen.

PROSES KERJA *SEARCH ENGINE*

5. Ranking:

a. *Scoring*

Menghitung skor untuk dokumen dengan menggunakan ranking algoritma., merupakan komponen pokok dari *search engine*. Bentuk dasar dari skor adalah

$$f(Q, D) = \sum_{t \in Q} q_t d_t$$

q_t dan d_t adalah bobot *term* dokumen untuk *term* t .

PROSES KERJA *SEARCH ENGINE*

5. Ranking:

b. Optimalisasi Kinerja

Mendisain ranking algoritma untuk pengolahan yang efisien.

Term-at-a time vs. document-at-a time processing
Optimalisasi aman vs tidak aman

c. Distribusi

Memproses query dalam lingkungan yang terdistribusi.

Broker query mendistribusikan query dan mengumpulkan hasil.

Caching adalah bentuk penelusuran terdistribusi.

PROSES KERJA *SEARCH ENGINE*

5. Evaluasi:

a. *Logging*

Logging query dan interaksi pengguna penting dalam meningkatkan efektivitas dan efisiensi penelusuran. *Query logs* dan klik per tayang data digunakan untuk menyaranakan query, mengecek ejaan, menyembunyikan query, ranking, penelusuran iklan dan komponen lainnya.

b. Analisis Ranking , mengukur dan menyematkan efektivitas.

c. Analisa Kinerja, mengukur dan menyematkan efisiensi sistem.

CartereGe, 2011.

Terima Kasih

Contact Me @nuningkurniasih

