KULIAH 11

WEB IR

BAB 13 Baeza-Yates & Ribeiro-Neto

Fakultas Ilmu Komputer – Universitas Indonesia

World Wide Web

- Dikembangkan oleh Tim Berners-Lee pada tahun 1990 di CERN untuk mengorganisasikan dokumen penelitian yang ada di Internet.
  - Mengembangkan protokol jaringan HTTP awal, URL, HTML, dan “web server” yg pertama.
- Browser pertama dikembangkan pada 1992 (Erwise, ViolaWWW).
- Pada 1993, Marc Andreessen dan Eric Bina dari UIUC NCSA mengembangkan browser *Mosaic* dan menyebarluaskannya.

Sejarah Awal Web Search

- 1993, web robots (spiders) pemula dibuat untuk mengumpulkan URL's:
  - Wanderer, ALIWEB (Pengindeks WEB seperti Archie), WWW Worm (mengindeks URL dan judul2 agar bisa dilacak dengan regex)
- 1994:
  - mahasiswa pasca dari Stanford David Filo dan Jerry Yang mulai mengumpulkan web sites yang digemari secara manual menjadi suatu *topical hierarchy* yang disebut Yahoo.

Sejarah Awal Web Search

- WebCrawler - tugas kelas di U Wash. (kemudian menjadi bagian dari Excite dan AOL).
- Lycos – mahasiswa CMU mengindeks sejumlah besar webpages
- Altavista – DEC

- 1998:
  - Larry Page dan Sergey Brin, mahasiswa S3 di Stanford, memulai Google. Kelebihan utamanya adalah penggunaan *link analysis* untuk mengurutkan dokumen yang diperoleh.

Tantangan dari Web untuk IR

- Data yang terdistribusi: Dokumen tersebar pada lebih dari sejuta web servers yang berbeda.
- Data yang mudah berubah: Banyak dokumen berubah atau hilang dengan cepat (mis. *dead links*).
- Volume yang besar: Dokumen berbeda dalam jumlah milyaran.
- Data yang tidak terstruktur dan terulang: Tidak ada struktur yang sama, kesalahan pada HTML, dokumen yang sama hampir 30%.
- Kualitas Data: Tidak ada pengeditan, informasi yg salah, penulisan yg buruk, salah tulis, dll.
- Data yang heterogen: Jenis2 media yang bervariasi (gambar, video), bahasa (100+), set karakter, dll.

Tantangan di Web : Pemakai

- Pemakai sangat bervariasi
  - Perbedaan pendidikan, kebutuhan, kemampuan
- Tidak ahli dalam membuat query
  - Singkat (rata-2 2.35 kata), tanpa operator (80%), kata-2 tdk tepat, sedikit usaha
- Tidak sabar /ahli dalam melihat hasilnya
  - 85% pemakai hanya melihat layar pertama yg dihasilkan search engine
  - 78% pemakai hanya menggunakan satu query
  - Mengikuti *link*

Contoh Search Engines

- Search Engines Umum:
  - Langsung: Google, Altavista, Lycos …
  - Tidak langsung (metasearch): MetaCrawler, DogPile, AskJeeves …
- Direktori bertingkat: Yahoo!
  - Yahoo mengatur secara manual suatu direktori yang sangat besar dari web pages yang terstruktur menurut hierarki.
  - Mis. Arts & Humanities Education
  - Automotive Business

Contoh Search Engines

- Search Engines Khusus:
  - Penemuan homepage: Ahoy
  - Robot untuk belanja: Jango
- Pelacakan dengan contoh:
  - Excite : “more like this”
  - Google: “Googlescout”

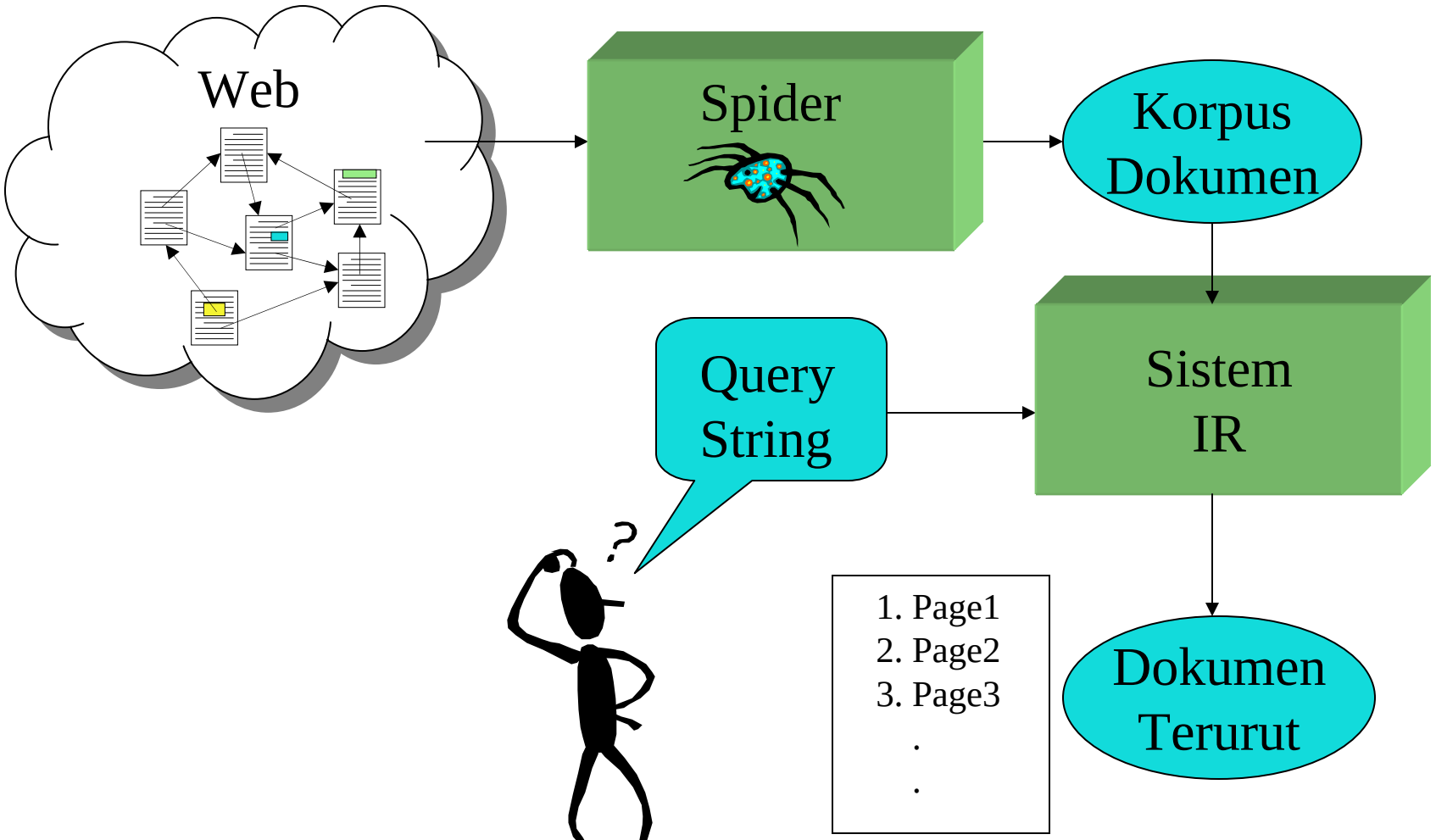
Bagaimana Mengukur Web?

- Teknologi sampling dan checking
  - Sample: ambil satu subset besar dari halaman
  - Cek apakah di-indeks oleh search engine

Arsitektur Search Engine

- Tersentralisasi
  - Lokasi dokumen hanya ada di satu lokasi
- Terdistribusi
  - Lokasi dokumen ada di beberapa lokasi

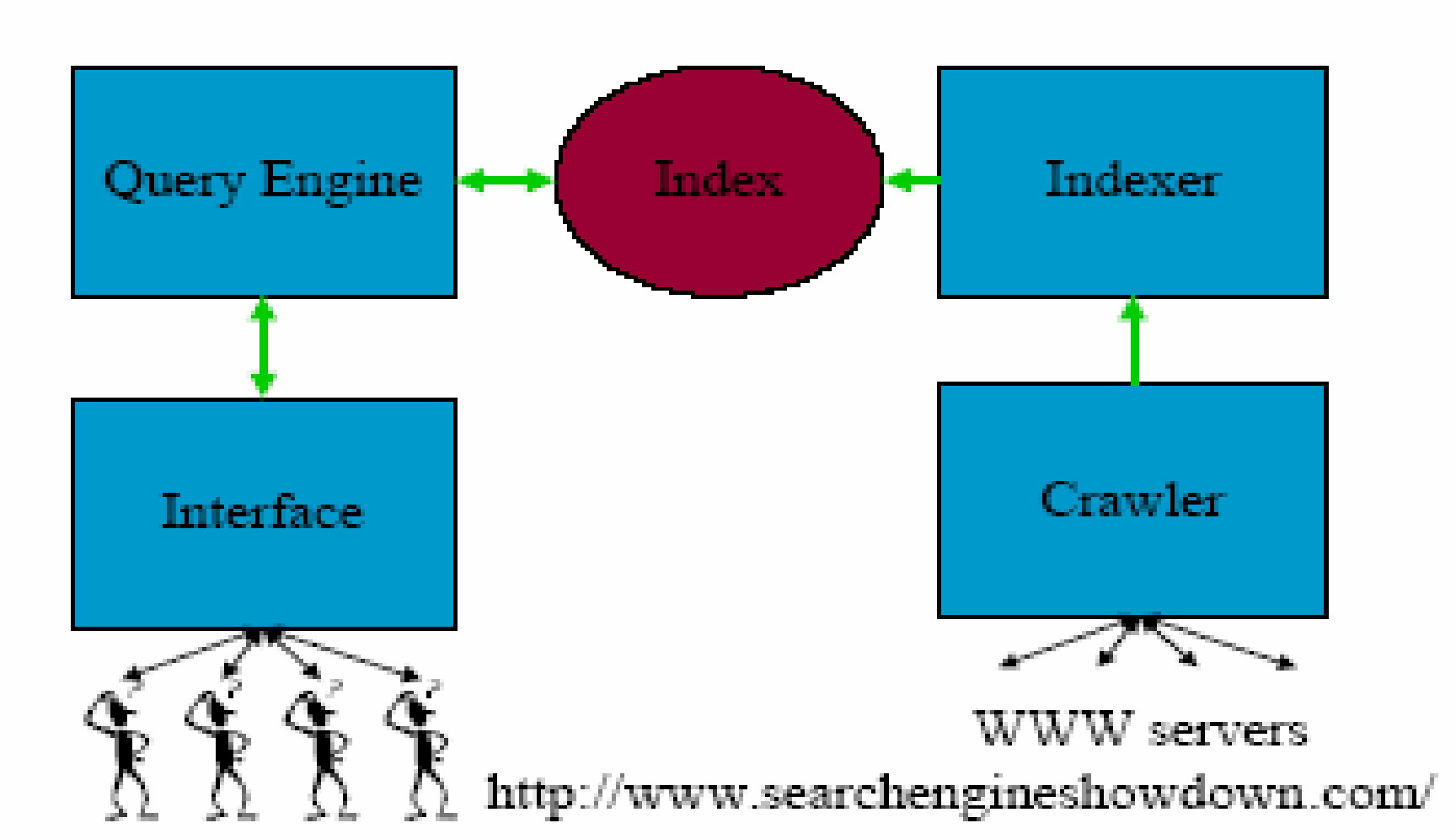
Web Search Dengan IR



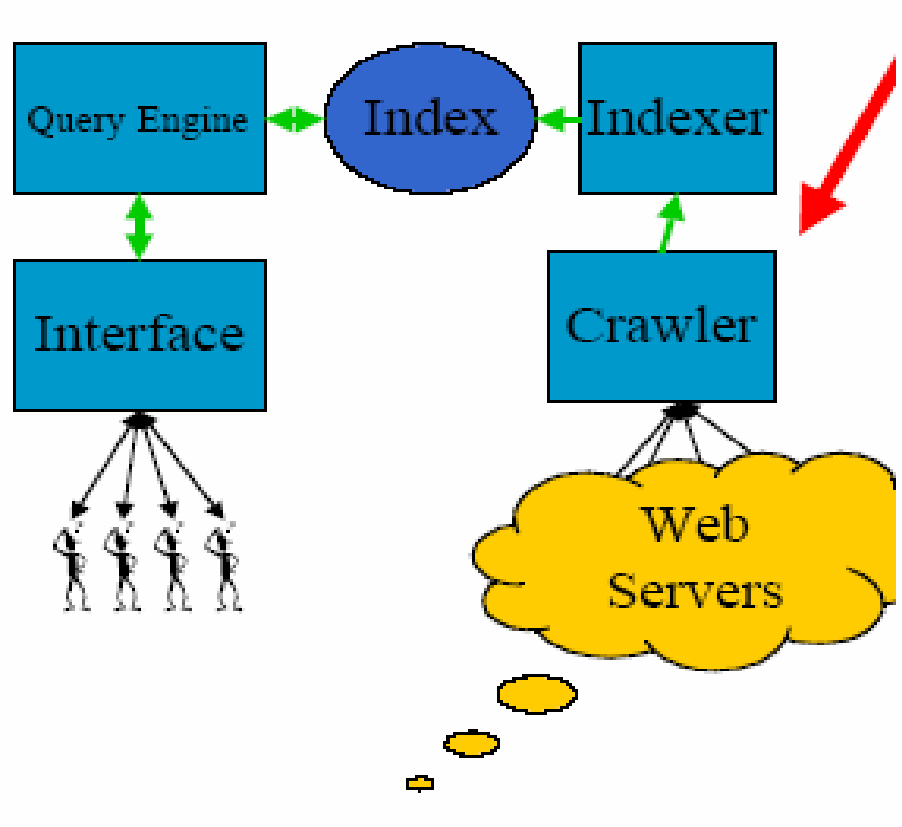
Pelacakan

- Search Engines
  - Cakupannya baik, kualitasnya rendah
- Directories
  - Kualitasnya baik, cakupannya rendah
- Metasearch engines
- Dynamic Search

Search Engine: Arsitektur

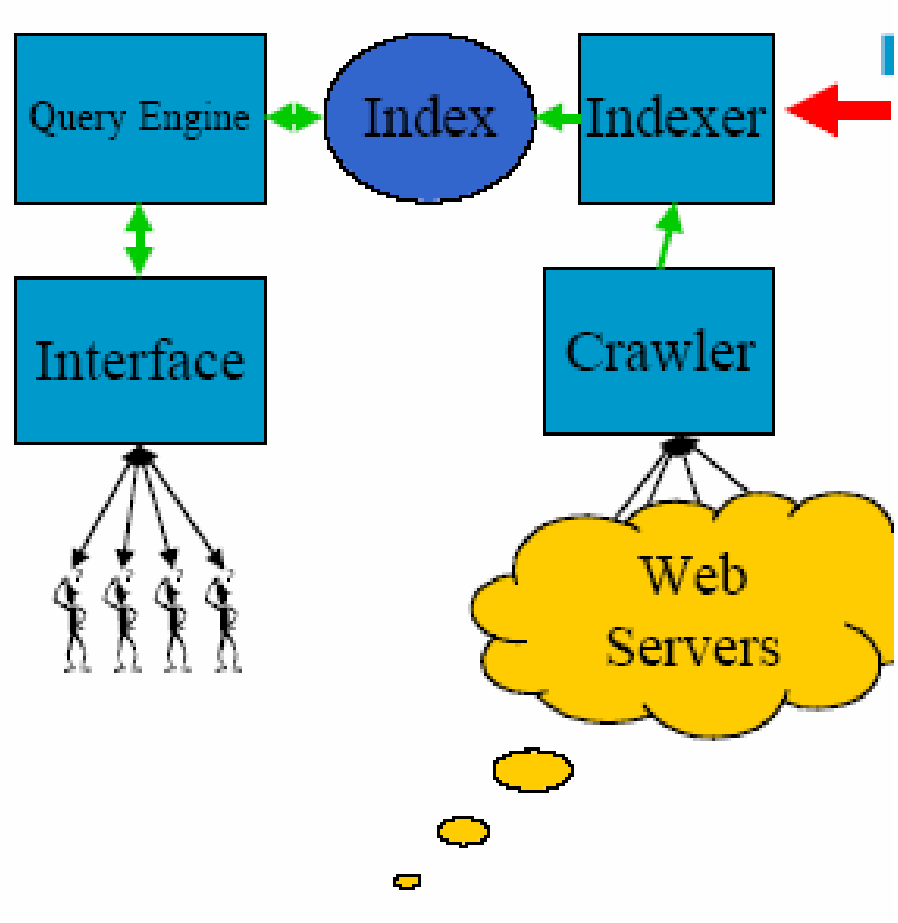


Search Engine: Arsitektur



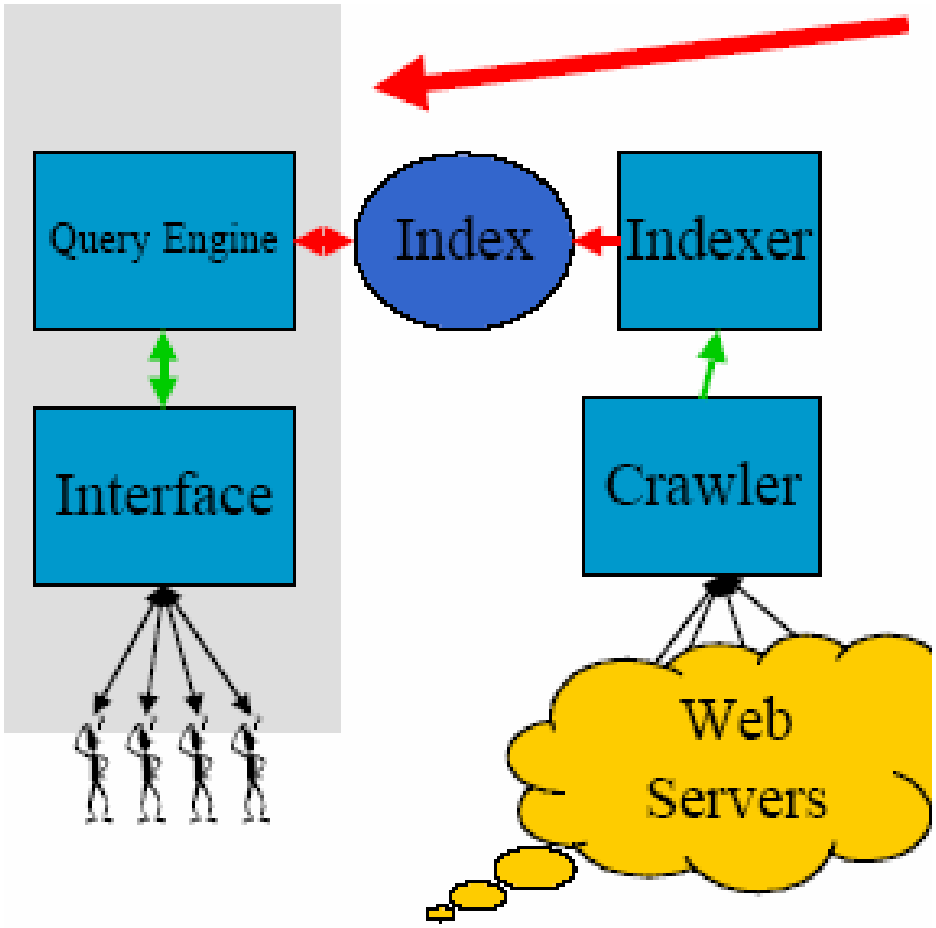
- *Web-crawler (robots, spiders, walkers)* berada di server lokal (tidak berjalan di web) dan mengirim permintaan ke server web lain
- Mengumpulkan *web pages*

Search Engine: Arsitektur



- *Indexer* membuat pengganti atau representasi dok yang di-ingat di *index*

Search Engine: Arsitektur



- Ada bagian yang berkaitan dgn user
- Beberapa search engine menyimpan query yg paling umum & hasilnya di cache.

Search Engine: Crawling dan Masalahnya

- Masalah pada crawler
  - Banyak pekerjaan yg hanya dilakukan sekali atau diulangi
  - Kemana harus pergi?
  - Harus adil terhadap web-page
- Pemecahan
  - Distributed crawling
  - Pengurutan untuk crawling
  - Teknik re-visiting

Search Engines: Indexer dan Masalah Pengindeksan

- Ukuran. Apa yg bisa disimpan?
  - Kata (dan posisi kata di dokumen)
  - Tanggal saat kunjungan
  - Starting page
  - Seluruh dokumen
- Masalah web yang mudah berubah

Search Engines: Masalah Pelacakan

- Bagaimana cara melacak dengan cepat?
  - Indeks yang lebih kecil
    - Stopwords, stemmer
  - Arsitektur terdistribusi

- Bagaimana cara mengurutkan hasil?
  - User tidak sabar
  - Kualitas terbaik harus ada di urutan awal
  - Mengatasi *spam*

Search Engines: Pengurutan Hasil Pelacakan (Ranking)

- Penilaian dokumen menggunakan sejumlah faktor yang ada pada dokumen web:
  - Model Ruang Vektor (*vector space*)
    - TF. IDF
  - Menggunakan *Hyperlinks*
    - Menggunakan links
    - Algoritma Page Rank dari Google
  - Struktur Dokumen
  - Term Proximity
  - Pseudo Relevance Feedback

Search Engines: Ranking …

- Hyperlinks:
  - Suatu webpage dianggap populer bila banyak mempunyai link yang masuk
  - Suatu webpage dianggap sebagai sumber dokumen yang baik bila mempunyai sejumlah link keluar menuju dokumen lain yang baik
  - Menghitung jumlah link yang masuk dan keluar dari suatu webpage
  - Caranya:
    - Menjumlahkan link yang keluar dan yang masuk
    - Atau hanya menghitung jumlah link yang masuk saja

Search Engines: Ranking … Algoritma Page Rank

- Suatu teknik pengurutan dokumen berdasarkan pada analisa link.

- $P(A) = (1-d) + d \sum_{Di..Dn} P(D_i) / C(D_i)$

$C(D_i)$ = jumlah link yang keluar dari page D_i

- d = dampening factor dari 0 – 1, supaya PR tidak 0 untuk page yg tidak mempunyai in-link
- Penghitungan dilakukan secara iteratif, dgn nilai perhitungan sebelumnya sampai tidak ada perubahan yg signifikan

Search Engines: Ranking … Struktur Dokumen

- Yaitu:
  - Judul, headings
  - Anchor text
    - Kuliah Pemrosesan Teks
  - Teks yang dicetak tebal

- Berguna untuk menemukan *name page* tapi tidak berpengaruh pada ad-hoc retrieval

Pelacakan : Directories
Pembuatan Hirarki Dokumen

- Pembuatan hirarki secara manual memerlukan tenaga yang sangat besar, subyektif, dan banyak kesalahan.

- Pembuatan hirarki secara otomatis
  - Bisa menggunakan teknik-teknik pengelompokan (*hierarchical text clustering*) misalnya *Hierarchical Agglomerative Clustering (HAC)*

Pelacakan: Meta Search Engine

- Mengapa harus membuat search engine sendiri?
- Penemuan dari pengukuran web
  - Intersection antara search engine sangat kecil
  - Query yang sama tp hasilnya berbeda
  - Ruang lingkup indeks terbatas pada setiap search engine
- Masalah pengurutan (menggabungkan hasil dari beberapa search engines)

Meta Search Engine

- Contoh:
  - MetaCrawler (13 SE), DogPile (25 SE)
- Webserver mengirim query ke
  - Beberapa search engines, web directories
  - Kumpulkan hasilnya
  - Gabungkan hasilnya (data fusion)
- Tujuan:
  - Ruang lingkup lebih baik, lebih efektif

Meta Search Engine

- Dibagi dalam beberapa fase
  - Pemilihan search engine
    - Tergantung topiknya, query yg lalu, network traffic
  - Pemilihan dokumen
    - Berapa jumlah dokumen dari setiap search engine?
  - Algoritma Penggabungan (*merging algorithm*)
    - Menggunakan posisi urutan, doc retrieval score …

Pelacakan: Term Proximity

- Term proximity dianggap berhasil untuk korpus yang terus bertambah
- Contoh Query: Kecelakaan pesawat terbang
- Bag of words:
  - #combine(kecelakaan pesawat terbang)
  - Pelacakan menggunakan pembobotan yang biasa
- Statistical phrase detection
  - #combine(kecelakaan “pesawat terbang”)
  - Frase sudah dikenali menurut statistik pada koleksi
- Weighted statistical phrases
  - #combine($w_1$kecelakaan w_2“pesawat terbang”)
  - Berikan bobot tersendiri untuk frase yang muncul pada query

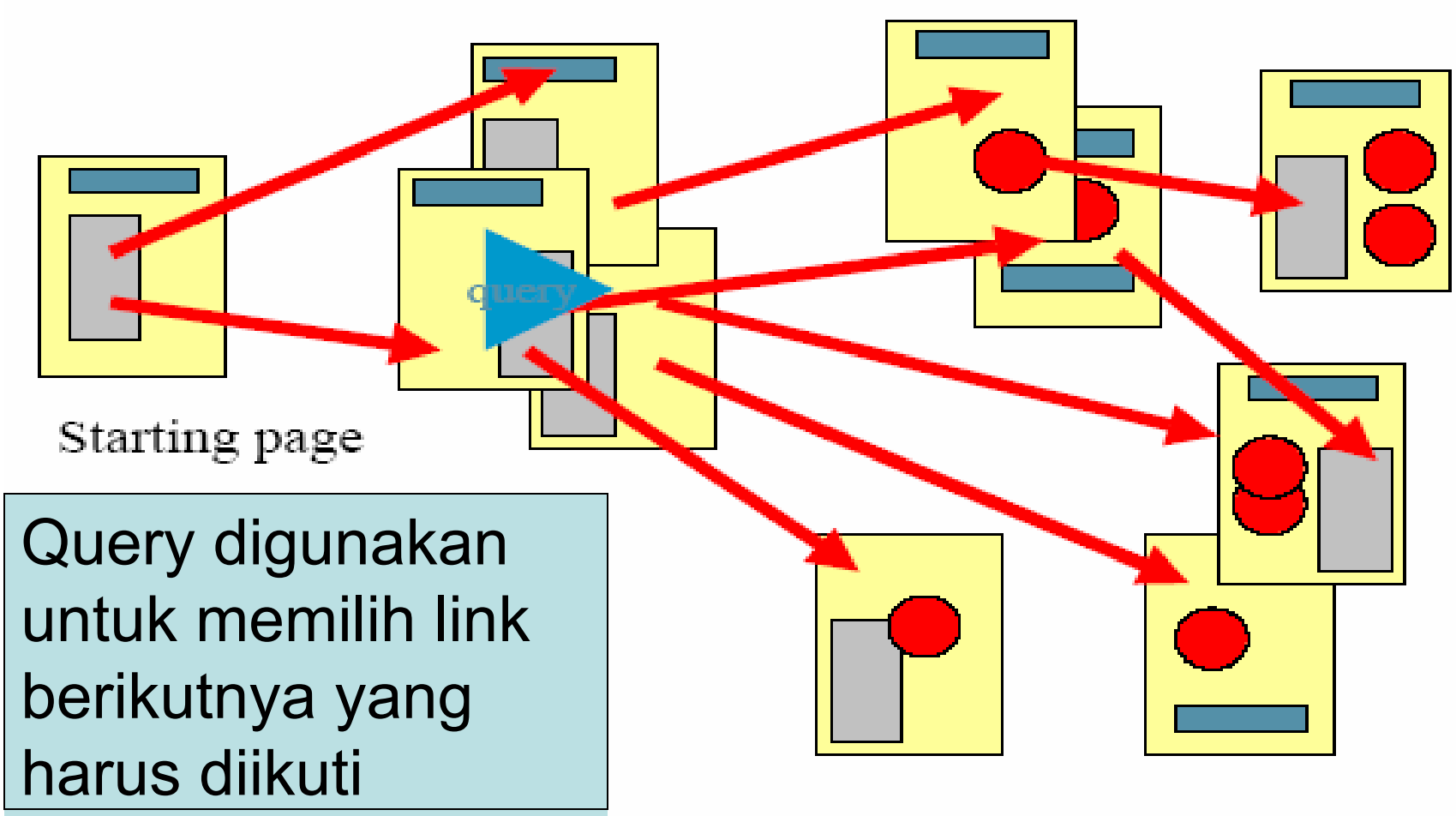
Pelacakan: Dynamic Search

- Idenya tidak mencari informasi yang disimpan pada search engine tapi di web itu sendiri
  - Pelacakan lebih lambat
  - Dapat digunakan di subset dari web yang dinamis dan kecil
- Disebut juga *focus crawling*

Dynamic Search

- Salah satu algoritma yaitu *fish search* yang menggunakan intuisi bahwa dok yg relevan seringkali mempunyai tetangga yg relevan
- Ide utamanya adalah mengikuti link dengan prioritas tertentu:
  - Starting page
  - Query yang digunakan untuk mencari suatu dokumen web

Dynamic Search



Query digunakan untuk memilih link berikutnya yang harus diikuti

WEBIR di CLEF dan TREC

- Menggunakan koleksi yang sangat besar (mis. CLEF mempunyai 10 GB data *EUROGOV* yang terkompres)
- Jenis query:
  - Menemukan Homepage : query adalah suatu *site* yang dicari (mis. Kedutaan Singapura) dan sistem harus memberikan URL dari *homepage* site tsb di (atau dekat) urutan pertama.
  - Menemukan Namepage: query adalah nama page yg bukan homepage (mis. Informasi visa di Indonesia) dan sistem harus memberikan URL dari *homepage* site tsb di (atau dekat) urutan pertama.

Evaluasi

- MRR: Mean Reciprocal Rank dari jawaban pertama yang benar
- Success@1: Proporsi dari query di mana jawaban yang benar pada urutan 1 (jawaban pertama yg dilihat user)
- Success@5: Proporsi dari query di mana 1 atau lebih jawaban ada di top 5 (user tidak perlu menggeser halaman). Mis. Jika jawaban yang benar muncul di 90 pertanyaan dari 225 pada top 5, maka Success@5 = 0.4

Evaluasi

- Succes@10: Mengindikasikan seberapa sering sistem menemukan jawaban pada top 10 (biasanya merupakan halaman pertama dari suatu hasil pelacakan pada web).
- Precision@10
- Recall@1000

Teknik pada WEBIR

- Pemberian bobot dokumen berdasarkan:
  - Isi
  - Judul
  - Anchor text
  - Panjang URL
- Efek penggunaan stemmer
  - Ada yang hasilnya meningkat tapi ada juga yang turun (tergantung koleksi)

Analisa Link dan URL

- Harus menemukan semua link pada suatu dokumen dan ambil URL-nya.
  -
  - <frame src="index.htm">
- Harus melengkapi URL relatif menggunakan URL page yang ada pada saat itu:
  - menjadi http://telaga.cs.ui.ac.id/WebKuliah/textpro/PR3
  - menjadi http://telaga.cs.ui.ac.id/WebKuliah/textpro/Tugas1.html

Analisa Link dan URL

- URL yang direktori path-nya pendek kemungkinan besar merupakan suatu homepage.

- Bobot isi bisa ditambah untuk dokumen yang populer dan mempunyai link masuk (*in-coming links)*, dapat meningkatkan recall dari dokumen ini.

Anchor Text

- Ambil *anchor text* (antara <a> dan) dari tiap link yang mengikutinya.
- *Anchor text* biasanya adalah deskripsi dari dokumen yang ditunjuknya.
- Tambahkan *anchor text* pada isi dari halaman tujuan untuk memberikan tambahan kata indeks yang relevan.
- Bisa digunakan untuk mengindeks dokumen dengan menggunakan semua *anchor text* yang mengacu dokumen tsb.
- Contoh:
  - Evil Empire
  - IBM

Anchor Text (2)

- Menolong bila teks deskripsi pada halaman tujuan tersimpan pada logo gambar (bukan pada teks yg bisa diakses).
- Seringkali *anchor text* is tidak berguna:
  - "click here"
- Bobot dari suatu kata bisa ditambah bila kata tsb. berasal dari *anchor text*.

Kenapa Google Berhasil

- PageRank dikombinasilan dengan teknik-teknik pencocokan teks yang mutakhir
  - Tidak hanya frekuensi kata, isi dokumen ataupun isi dari dokumen yang dirujuk
- Features yang ditangani dengan baik
  - Statik (PageRank, probabilitas spam)
  - Kata (kemunculan, kedekatan, konteks)
  - Spesifik query (klasifikasi query)
- Parameter Tuning
  - Gosip: mereka memberikan bobot secara manual pada parameter
- Data (ukuran indeks dan penggunaan data)
- Penelitian yang signifikan dan banyak (mis. Stemming)

TOPIK PENELITIAN

- Modelling, Querying, Ranking
- Arsitektur Terdistribusi
- Indexing
- Duplikasi Data
- Multimedia
- Browsing
- User Interface

Ringkasan

- Teknik harus disesuaikan untuk memroses dokumen web
  - Analisa link dan struktur dokumen
  - Anchor text
    - Penting untuk mencari name-page
    - Tidak terlalu berpengaruh untuk ad-hoc retrieval
  - Term proximity juga berguna terutama untuk koleksi yang terus bertambah
  - Page Rank hanya merupakan sebagian kecil dari yang dilakukan Google