

WEB INFORMATION RETRIEVAL

Pertemuan 2

Arsitektur Sistem Web IR

Crawler • Repository • Indexer • Query Processor • Ranking • Result Presentation

FROM THE WEB TO THE SERP



Capaian Pertemuan 2

Apa yang harus mampu dijelaskan mahasiswa setelah perkuliahan?

Sub-CPMK 2

Mampu menjelaskan arsitektur Web IR dan alur kerja komponen crawler, repository, indexer, query processor, ranking, serta antarmuka hasil pencarian. (C3, A3)

Indikator

Mahasiswa mampu memetakan komponen, menjelaskan fungsi tiap lapisan, dan menunjukkan aliran data dari web hingga SERP.

Bukti Belajar

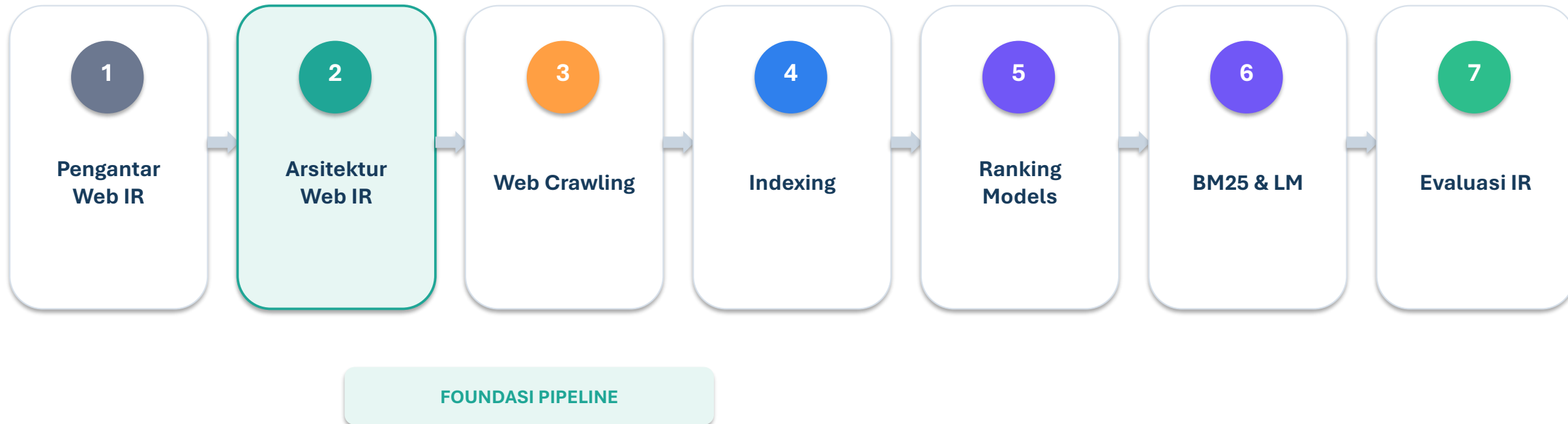
Diagram pipeline Web IR, diskusi studi kasus arsitektur mesin pencari, dan Quiz 1.

TARGET PRAKTIS

- Membedakan document pipeline dan query pipeline.
- Menjelaskan hubungan crawler → repository → indexer → index.
- Menjelaskan query processing → candidate retrieval → ranking → result presentation.
- Mengidentifikasi bottleneck dan failure point pada arsitektur pencarian sederhana.

Posisi Pertemuan 2 dalam Semester

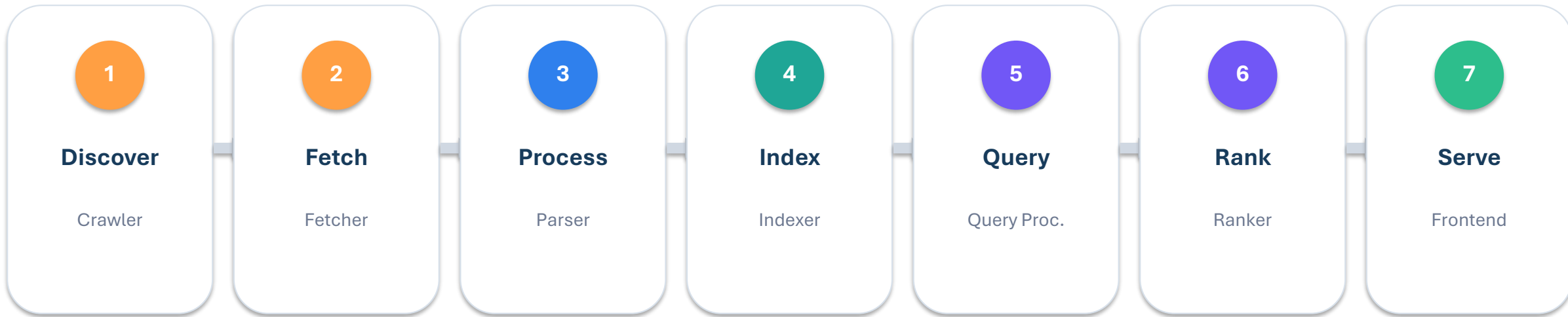
Hari ini kita memperdalam “mesin” di balik proses pencarian web.



P1 memahami masalah pencarian. P2 memahami arsitektur. P3–P7 mempelajari algoritma pada setiap komponen.

Dari “Search Lifecycle” ke Arsitektur

Pertemuan 1 memberi gambaran besar; sekarang kita buka kotak hitamnya.



Pertanyaan inti arsitektur:

Komponen apa yang mengerjakan tiap tahap, data apa yang dipertukarkan, dan bagaimana memastikan pencarian tetap cepat serta relevan?

01

Arsitektur Besar Web IR

Memisahkan pekerjaan berat “sebelum query” dan pekerjaan latensi-rendah “saat query”.

Offline Side vs Online Side

Dua jalur kerja dengan karakteristik waktu dan beban yang berbeda.

DOCUMENT PIPELINE / OFFLINE SIDE

- Menemukan dan mengambil dokumen web
- Membersihkan & mengekstrak konten
- Membangun / memperbarui indeks
- Boleh memproses secara batch / near-real-time

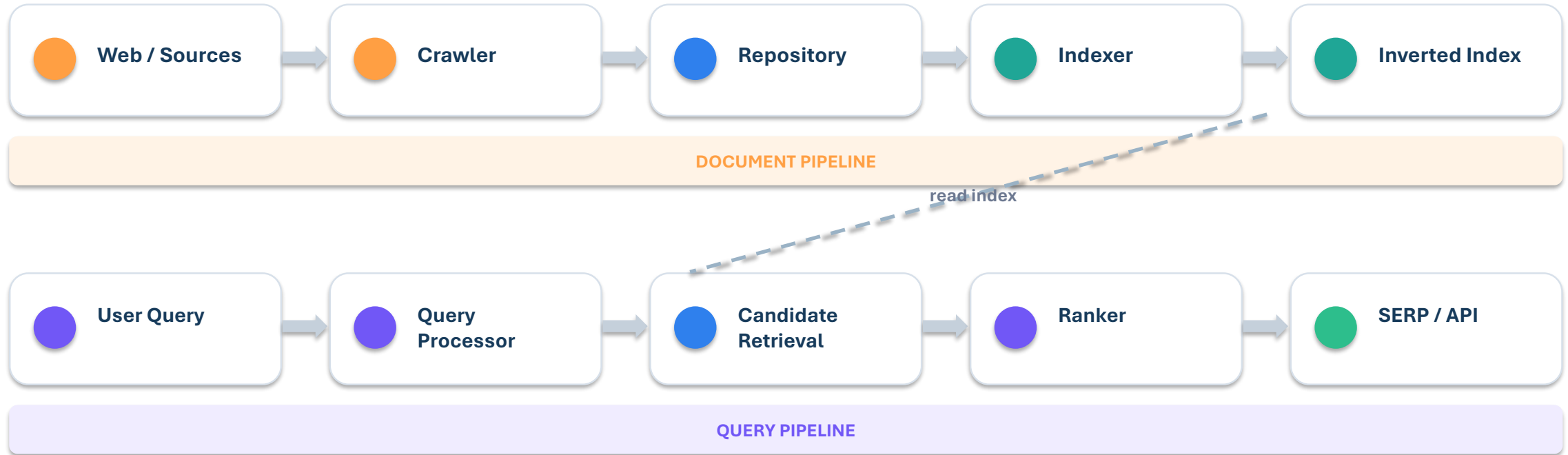
QUERY PIPELINE / ONLINE SIDE

- Menerima query pengguna
- Memproses & memahami query
- Mengambil kandidat dari indeks
- Ranking, snippet, filter, dan menyajikan hasil

DESIGN PRINCIPLE: kerja mahal dipindahkan ke offline; online harus cepat, konsisten, dan terukur.

Peta Komponen Web Information Retrieval

Satu arsitektur sederhana yang menghubungkan document pipeline dan query pipeline.



Repository = raw/processed documents • Index = search-optimized representation

Apa yang Mengalir di Antara Komponen?

Arsitektur menjadi jelas ketika kita melihat bentuk datanya.

URL**`https://example.com/product/123`**

Crawler menerima & menjadwalkan URL

HTML**`<title>...</title><body>...`**

Fetcher / parser memperoleh konten

Document**`{id, title, body, links, metadata}`**

Repository menyimpan representasi dokumen

Tokens**`[sepatu, lari, pria, hitam]`**

Indexer membuat unit pencarian

Postings**`sepatu → [D1, D5, D8]`**

Index menyimpan mapping term → dokumen

Query**`"sepatu lari hitam"`**

Query processor membentuk query internal

Ranked list**`D5 > D1 > D8`**

Ranker menghasilkan urutan kandidat

SERP**`title + snippet + URL + facets`**

Frontend menyajikan hasil

02

Crawler & Repository

Mengubah web yang terus berubah menjadi koleksi dokumen yang dapat diproses.

Apa Tugas Web Crawler?

Crawler bukan “search engine”; ia hanya membangun pasokan dokumen.

1 Discover

Menemukan URL baru dari seed list, sitemap, dan link yang ditemukan pada halaman.

2 Fetch

Mengambil halaman melalui HTTP/HTTPS dengan memperhatikan status code, timeout, dan kebijakan akses.

3 Parse & Extract

Mengekstrak teks, metadata, canonical URL, dan outbound links untuk diproses lebih lanjut.

OUTPUT UTAMA CRAWLER

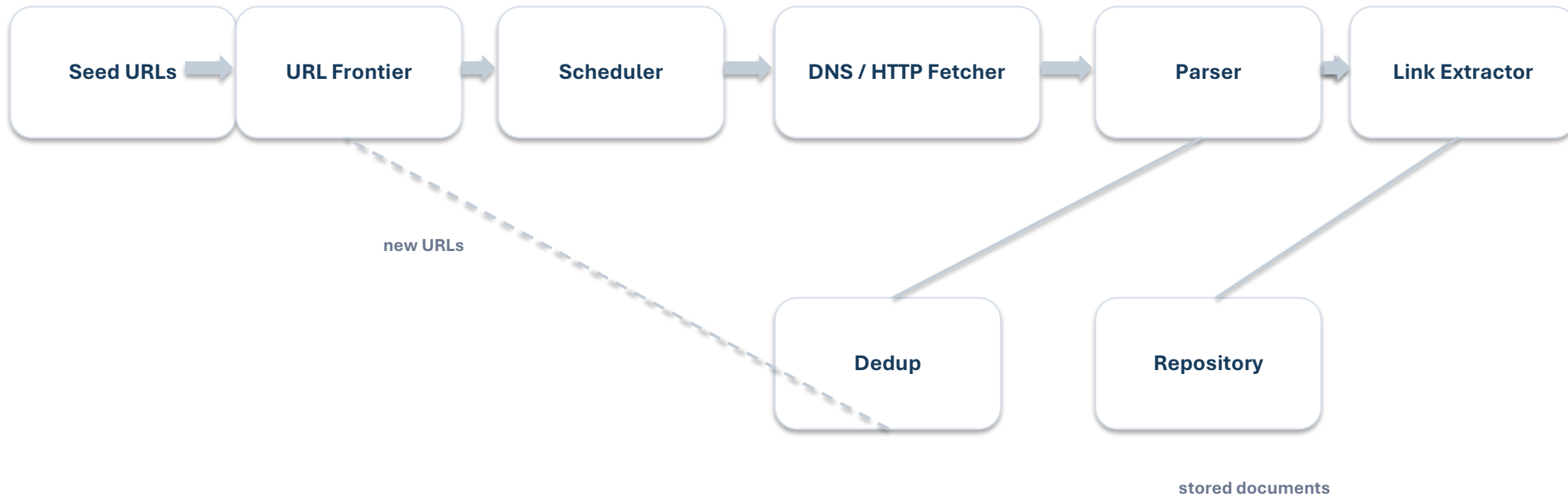
- Dokumen yang berhasil diambil
- URL baru untuk frontier
- Metadata crawl: status, timestamp, content type
- Sinyal duplikasi / canonical URL

Design concern

**Coverage • Freshness • Politeness •
Throughput • Duplicate control**

Anatomi Crawler Sederhana

Komponen internal yang membuat crawling terkontrol dan dapat diulang.



FRONTIER menentukan “apa berikutnya”; **scheduler** menentukan “kapan & ke host mana”.

URL Frontier & Scheduling

Antrian URL bukan sekadar FIFO.

Prioritas yang dapat dipakai

- Seed / sitemap importance
- Host politeness & crawl delay
- Freshness: kapan terakhir diambil
- Estimated change rate
- URL depth / path pattern
- Page importance / authority signal

Contoh scheduling

news.id

/home

/politik

/tech

shop.id

/p/18

/p/22

campus.ac.id

/adm

/news

Tujuan: throughput tinggi tanpa membanjiri satu host.

Raw Content, Parsed Content, dan Canonical Document

Sebelum indexing, sistem perlu memastikan dokumen memiliki identitas yang konsisten.

Raw Repository

HTML/JSON asli, HTTP headers, timestamp, status code. Berguna untuk reprocessing tanpa crawl ulang.

Parsed Document

Title, body text, links, language, metadata, structured fields.

Canonical Document

Satu identitas utama untuk mengurangi duplikasi URL / konten.

Contoh duplikasi URL

`https://shop.id/product?id=55&utm_source=ig`

`https://shop.id/product?id=55`

`https://shop.id/product/55`



Canonical ID: PRODUCT-55

Satu dokumen → satu entitas pencarian.

03

Indexer & Search Index

Mengubah dokumen menjadi struktur yang dioptimalkan untuk pencarian cepat.

Apa Tugas Indexer?

Indexer adalah “pabrik” yang membuat dokumen siap dicari.



HASIL: indeks yang jauh lebih cepat dicari daripada membaca seluruh dokumen satu per satu.

Inverted Index: Jantung Search Engine Klasik

Dari “dokumen berisi term” menjadi “term menunjuk dokumen”.

FORWARD VIEW

D1	web search engine
D2	search ranking model
D3	web crawler search



INVERTED VIEW

web	→	D1, D3
search	→	D1, D2, D3
ranking	→	D2
crawler	→	D3

Query “web ranking” cukup membuka postings untuk dua term.

Dictionary, Postings, dan Statistics

Index menyimpan lebih dari sekadar daftar document ID.

DICTIONARY

crawler

df=1

index

df=2

ranking

df=1

retrieval

df=2

web

df=2

POSTINGS untuk term “web”

D1

tf=1

pos=[1]

→ enables scoring & phrase/proximity queries

D3

tf=1

pos=[1]

→ enables scoring & phrase/proximity queries

Statistik umum: document frequency (df), term frequency (tf), positions, field, document length.

Batch Indexing vs Near-Real-Time Indexing

Seberapa cepat perubahan web terlihat pada hasil pencarian?

Batch

Index dibangun ulang / diperbarui secara berkala. Sederhana, efisien, tetapi freshness lebih rendah.

Incremental

Dokumen baru/berubah ditambahkan ke segmen index tanpa rebuild total.

Near Real-Time

Segmen kecil dibuat dan dipublikasikan cepat; kemudian digabung melalui merge/compaction.

Trade-off desain



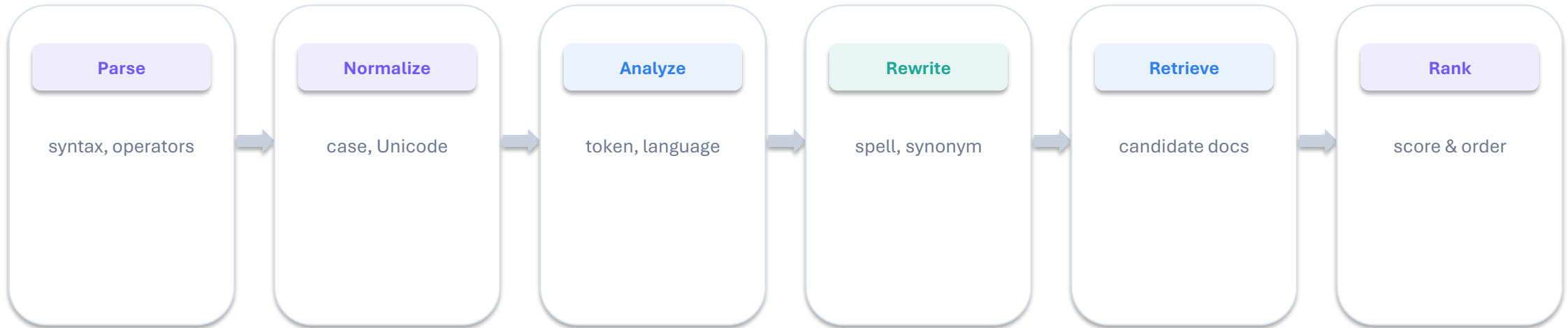
04

Query Processor & Retrieval

Saat pengguna mengetik query, sistem harus merespons dalam milidetik–detik.

Apa yang Terjadi pada Query Pengguna?

Query teks perlu diterjemahkan menjadi representasi yang dapat dicocokkan dengan index.



Contoh: **"web retrival"** → spell correction → **"web retrieval"** → retrieve candidates → rank

Query Understanding & Rewriting

Tujuannya bukan mengubah maksud pengguna, tetapi meningkatkan kemungkinan menemukan dokumen yang relevan.



Catatan: rewriting perlu diawasi agar tidak memperkenalkan bias atau menggeser intent.

Candidate Retrieval: Jangan Ranking Semua Dokumen

Tahap pertama mempersempit miliaran dokumen menjadi kandidat yang masuk akal.

KOLEKSI / INDEX

1.000.000 dokumen

Candidate generation

mengutamakan recall & efisiensi

10.000 lexical candidates

1.000 rerank candidates

Ranking / reranking

mengoptimalkan urutan relevansi

10 hasil SERP

Ranking Layer: Mengurutkan Kandidat

Pertemuan 5–6 akan membahas model ranking; hari ini fokus pada posisi ranker dalam arsitektur.

Input

Candidate documents + query + document statistics + optional behavioral/context signals.

Scoring

Model menghitung relevance score / utility score untuk setiap kandidat.

Output

Daftar dokumen terurut beserta skor yang kemudian diperkaya dengan snippet / metadata.

Signal families

Term match

Field importance

Freshness

Authority / link

Behavioral

Business rules

Result Presentation / Search API

Ranking bagus tetap membutuhkan penyajian hasil yang jelas dan cepat.

web information retrieval architecture

Arsitektur Web IR — Overview

Crawler, indexer, query processor, ranking dan serving layer.

Search Engine Architecture

Document pipeline dan query pipeline pada mesin pencari.

Information Retrieval System

Struktur index, candidate retrieval dan ranking results.

Snippet

Cuplikan konteks query dari dokumen.

Facets / Filters

Kategori, tanggal, lokasi, tipe dokumen.

Cache & API

Mempercepat respons dan melayani banyak client.

Latency Budget pada Query Pipeline

Setiap komponen online menghabiskan sebagian waktu respons.



Ilustrasi konseptual — bukan angka baku.



Bottleneck

Index lambat, network, model ranking berat, cache miss, atau fan-out terlalu besar.



Mitigasi

Caching, sharding, pre-compute, candidate pruning, parallelism, timeout.



SLO

Tetapkan target latency, availability, freshness, dan quality.

Monitoring Arsitektur Search

Search engine yang baik harus dapat diobservasi, bukan hanya “jalan”.

Crawler

fetch rate, error rate, queue size, freshness

Indexer

docs/min, indexing lag, segment size, merge time

Query

QPS, p50/p95/p99 latency, timeout

Quality

CTR, zero-result rate, precision proxy, abandonment

Operational metrics ≠ relevance metrics. Keduanya harus dipantau bersama.

Failure Modes yang Sering Terjadi

Pahami gejala, kemungkinan sumber masalah, dan komponen yang perlu diperiksa.

GEJALA	HIPOTESIS	CEK KOMPONEN
Hasil tidak menemukan konten baru	Crawling / indexing lag	Crawler → Indexer
Query lambat hanya pada jam sibuk	Overload / fan-out / cache	Query Serving
Banyak hasil duplikat	Canonicalization / dedup lemah	Repository / Indexer
Zero result meningkat	Analyzer mismatch / index issue	Query Proc. / Index
Ranking terasa tidak relevan	Candidate/ranker/signal issue	Retrieval / Ranker

Kasus: Search Portal Akademik

Universitas ingin menyatukan pencarian berita, dokumen akademik, pengumuman, dan repositori.

Situasi

- Sumber tersebar di 5 situs / aplikasi
- Format: HTML, PDF, metadata repositori
- Konten berubah setiap hari
- Pengguna: mahasiswa, dosen, staf
- Masalah: hasil duplikat & sulit menemukan dokumen resmi

Tugas kelompok — 12 menit

- Gambar pipeline Web IR yang diperlukan.
- Tentukan apa yang masuk repository dan index.
- Jelaskan bagaimana PDF dan HTML diproses.
- Tentukan 3 metric operasional untuk monitoring.
- Identifikasi 2 failure mode dan mitigasinya.

OUTPUT: 1 diagram + 5 keputusan desain

Kasus: Arsitektur Search Marketplace

Produk harus segera dapat dicari setelah seller mengubah stok, harga, atau judul.

Requirement 1

Update stok dan harga tampil dalam pencarian < 1 menit.

Requirement 2

Query puncak tinggi saat kampanye flash sale.

Requirement 3

Search mendukung filter kategori, lokasi, harga, rating.

Diskusikan keputusan arsitektur:

- Apakah perlu crawling? Atau ingestion via event/API?
- Batch, incremental, atau near-real-time indexing?
- Field apa yang masuk inverted index vs filter store?
- Di mana caching paling efektif?
- Bagaimana menjaga search tetap available saat update index?

Cek Pemahaman — Arsitektur Web IR

Pilih jawaban paling tepat. Diskusikan alasan, bukan hanya huruf jawabannya.

1. Komponen yang bertanggung jawab menemukan URL baru adalah...

A Repository B Crawler C Ranker D Query Processor

2. Struktur yang menghubungkan term ke daftar dokumen disebut...

A URL Frontier B Inverted Index C Cache D Sitemap

3. Komponen online yang menafsirkan dan menormalisasi query adalah...

A Query Processor B Fetcher C Repository D Parser

4. Mengapa candidate retrieval dipisahkan dari ranking?

A agar UI lebih cantik B agar semua dokumen diranking C efisiensi & latency D menghapus index

Pembahasan Jawaban

Gunakan penjelasan arsitektur, bukan menghafal istilah.

1

B — Crawler

Crawler menemukan dan mengambil halaman serta mengekstrak URL baru.

2

B — Inverted Index

Mapping term → postings memungkinkan pencarian cepat.

3

A — Query Processor

Query diparse, dinormalisasi, dianalisis, dan dapat direwrite.

4

C — Efisiensi & latency

Ranking mahal; kandidat dipersempit dulu agar online pipeline tetap cepat.

Lima Ide Utama Pertemuan 2

Jika lima hal ini sudah jelas, arsitektur dasar Web IR sudah terbentuk.

- 1 Pisahkan offline dan online** Document pipeline membangun index; query pipeline harus cepat.
- 2 Crawler memasok dokumen** Frontier, scheduler, fetcher, parser, dedup, repository.
- 3 Indexer mengubah dokumen** Text processing → dictionary/postings → search index.
- 4 Query processor membuka jalan** Normalize/rewrite → candidate retrieval → rank.
- 5 Serving harus observable** Latency, freshness, availability, quality dipantau bersama.

ARCHITECTURE = COMPONENTS + DATA FLOW + PERFORMANCE + FAILURE HANDLING

Persiapan Pertemuan 3 — Web Crawling

Pada pertemuan berikutnya kita memperdalam crawler secara teknis dan etis.

Pelajari

URL frontier, BFS/priority crawling, politeness, robots.txt, sitemap, canonical URL.

Latihan

Buka robots.txt dari 2 situs publik dan catat aturan Allow/Disallow yang terlihat.

Pertanyaan

Bagaimana crawler mencari banyak halaman tanpa membebani server tujuan?

Tugas dari Pertemuan 2

Buat diagram pipeline Web IR untuk satu kasus pilihan: portal kampus, marketplace, berita, atau perpustakaan digital. Tandai setiap komponen dan data yang mengalir.

Referensi Utama

Bacaan untuk memperkuat konsep arsitektur search engine dan information retrieval.

- Manning, C. D., Raghavan, P., & Schütze, H. — Introduction to Information Retrieval. Cambridge University Press.
- Croft, W. B., Metzler, D., & Strohman, T. — Search Engines: Information Retrieval in Practice. Pearson.
- Baeza-Yates, R., & Ribeiro-Neto, B. — Modern Information Retrieval: The Concepts and Technology behind Search.
- **RPS Web Information Retrieval OBE — Modul 2: Arsitektur Web IR; document pipeline; query pipeline; komponen mesin pencari.**

PERTEMUAN 2 SELESAI • QUIZ 1 • LANJUT: WEB CRAWLING