BUSINESS INTELLIGENCE

Extract Transform Load (ETL)

FAST: FOCUS, ADAPTIVE, SINCERE, TRUSTWORTHY

JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

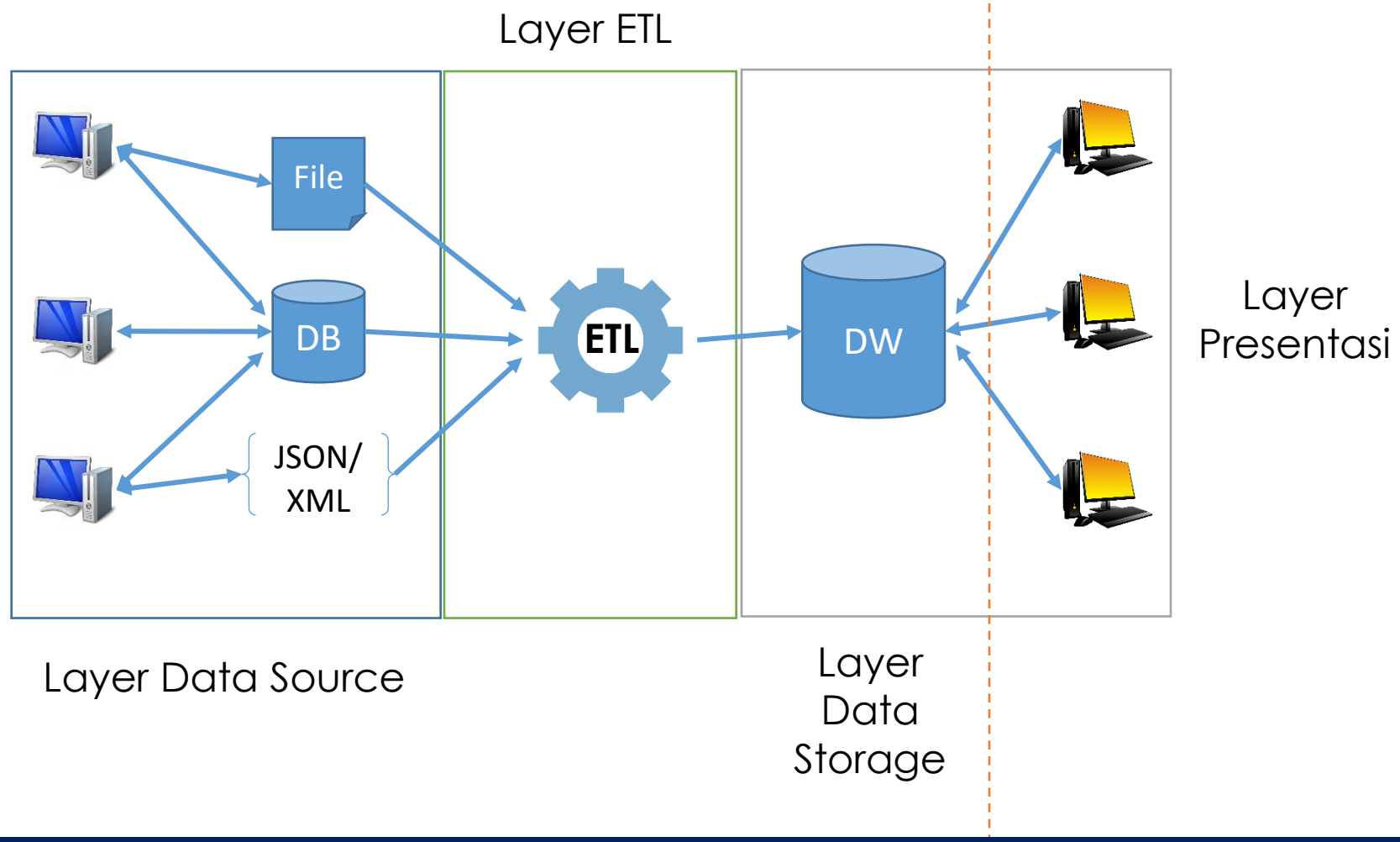
Konsep ETL

- ETL merupakan serangkaian proses yang terdiri dari:

Extract	Transform	Load
• Ekstraksi data dari data source	• Mengubah format data	• Memuat data ke destination

- Perlu dicatat bahwa ETL tidak hanya dari OLTP ke DW

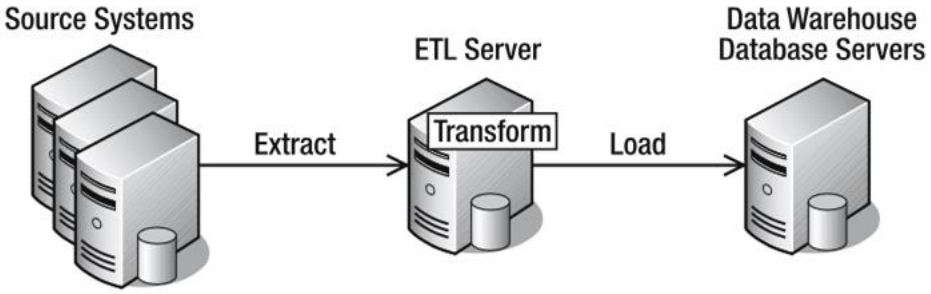
Konfigurasi Minimal

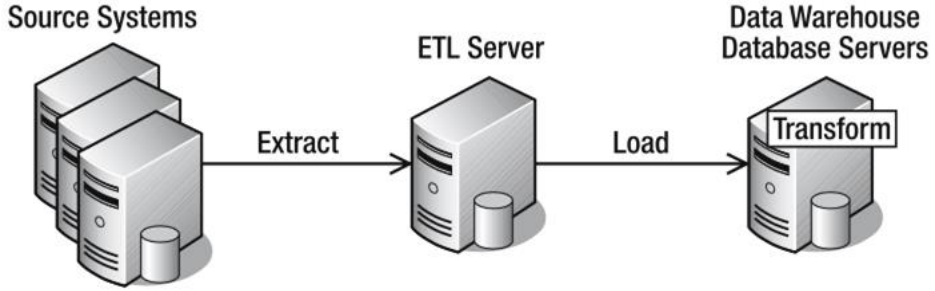


Bagian 1

Arsitektur

ETL Vs ELT


Source Systems
ETL Server
Data Warehouse
Database Servers
Extract
Transform
Load


Source Systems
ETL Server
Data Warehouse
Database Servers
Extract
Load
Transform

PENDEKATAN 1 - ETL

- Menarik data dari data source, meletakkannya di staging area, dilakukan transformasi kemudian dimuat ke data warehouse;

  atau

- Menarik data dari data source, melakukan transformasi di memori (tanpa staging) dan langsung dimuat di data warehouse

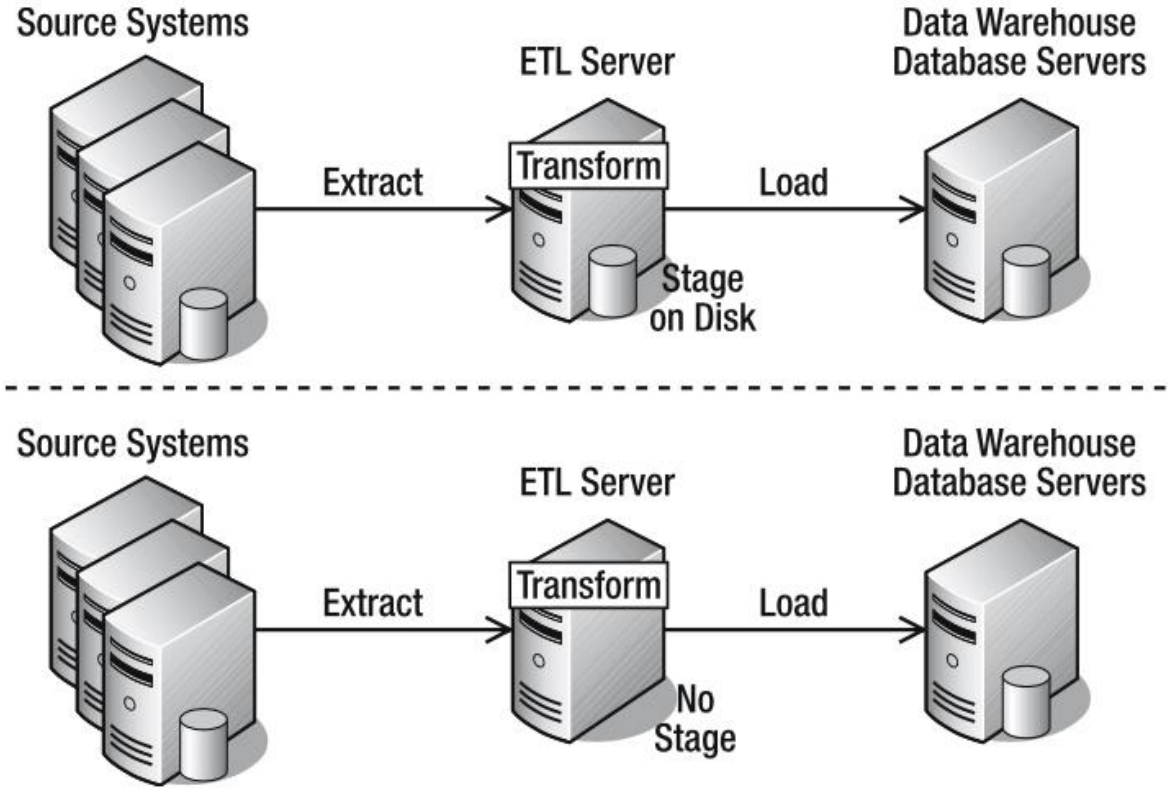


Figure 7-1. *To stage on disks or do transformation in memory*

STAGING

- Tempat sementara di mana data dengan bermacam format dari berbagai data source dikumpulkan sebelum menjalani proses transformasi.
- Tujuan:
  - Berlaku sebagai buffer antara raw data dan data warehouse untuk mencegah direct dumping data ke data warehouse
  - Safety net jika terjadi kegagalan proses ETL → menyediakan titik rollback
  - Mengurangi computational load di DW
  - Memastikan proses ETL berjalan dengan lancar dan efisien

PENDEKATAN 2 - ELT

- Pada ELT, proses transformasi dilakukan setelah data dimuat ke data warehouse.

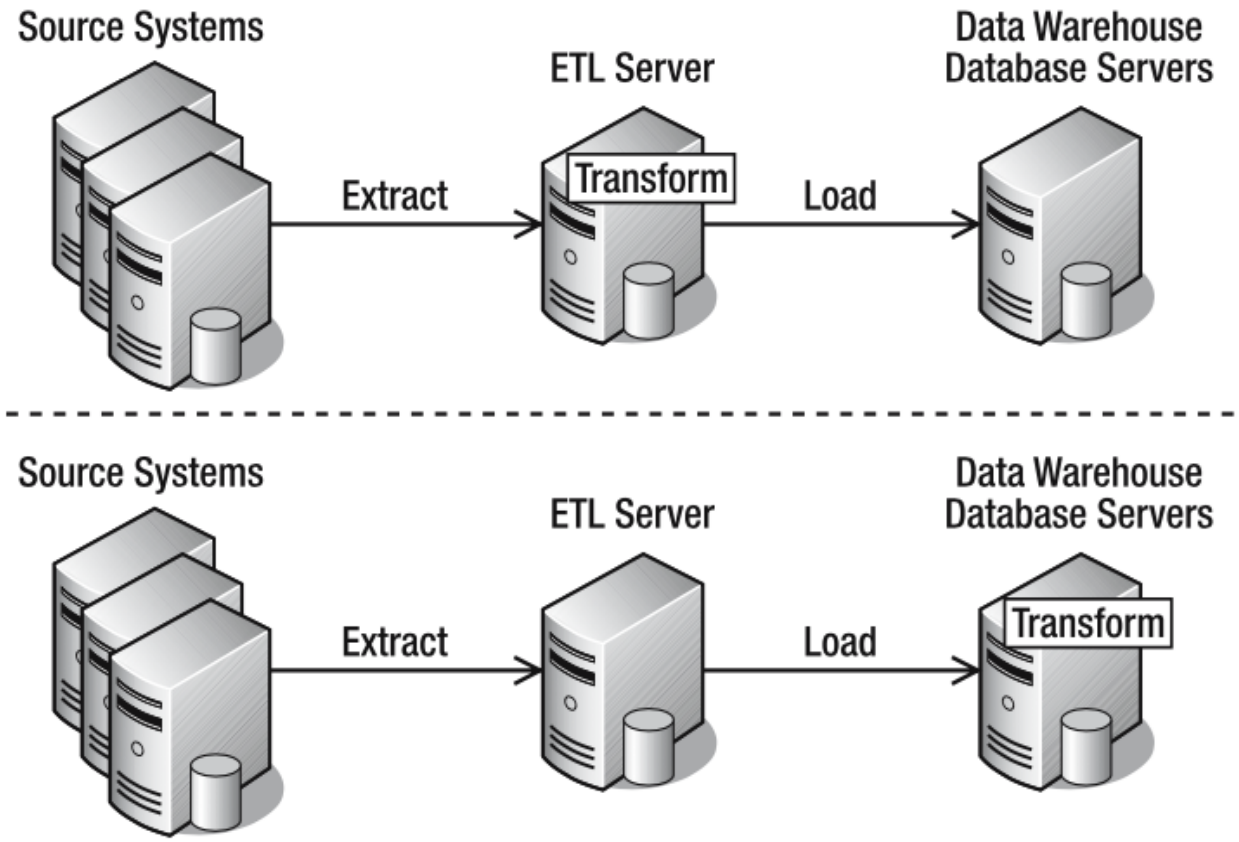


Figure 7-2. *ETL and ELT: choices of where to perform the transformations*

Contoh ETL VS ELT

- Misalnya kita ingin menyimpan data penjualan per kategori di DW

- **ETL**
  1. Extract: Tarik data dari OLTP database
  2. Transform: Hitung total penjualan per kategori dengan Pentaho
  3. Load: Masukkan hasil ke tabel summary di DW

- **ELT**
  1. Extract: Tarik data dari OLTP database
  2. Load: Masukkan *raw data* langsung ke DW
  3. Transform: Gunakan SQL di DW untuk bikin tabel summary

ETL VS ELT

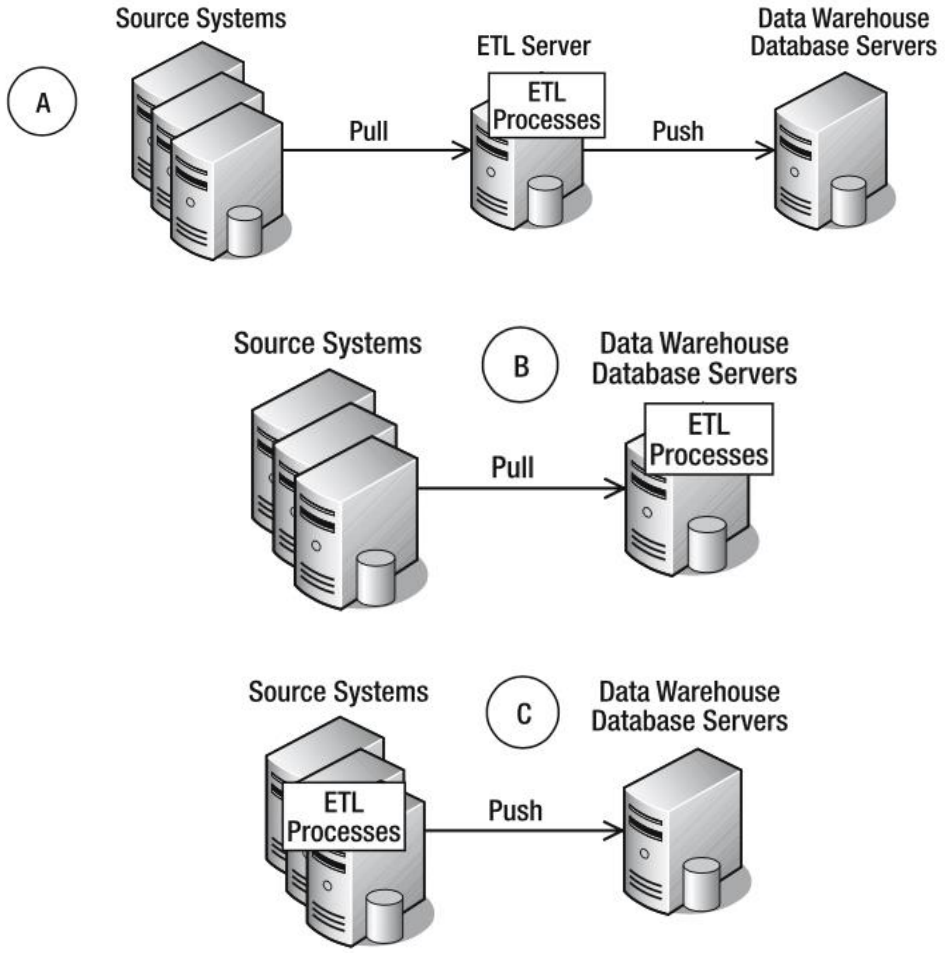
- **ETL**
  1. Menggunakan on-premise data warehouse
  2. Perlu validasi & pembersihan data sebelum masuk ke warehouse
  3. Infrastruktur belum siap untuk menyimpan data mentah di DW
  4. Perlu transformasi kompleks

- **ELT**
  1. Memakai cloud data warehouse (BigQuery, Snowflake, Redshift)
  2. Volume data sangat besar dan ingin memanfaatkan kekuatan compute DWH
  3. Ingin memelihara raw data di warehouse untuk fleksibilitas analitik
  4. Transformasi dapat dilakukan langsung dengan SQL

Hybrid ETL + ELT

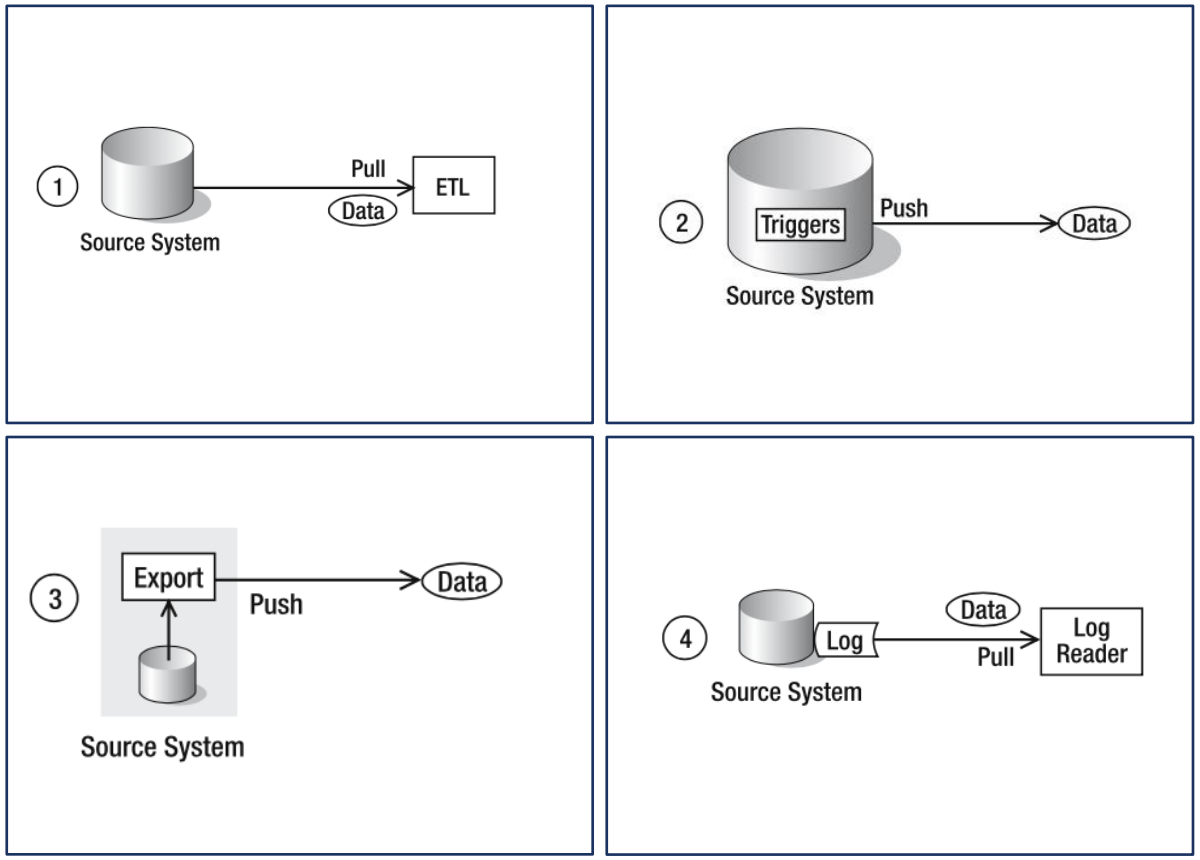
- Dalam praktik nyata, banyak organisasi menggunakan **gabungan ETL dan ELT**:
  - ETL → untuk pembersihan awal (standardisasi format, encoding, dll.)
  - ELT → untuk transformasi logika bisnis (join, aggregasi, normalisasi)

JENIS ETL BERDASARKAN DI MANA ETL DIPROSES



- Menurut tempat dimana engine ETL diletakkan terdiri dari 2 jenis:
  - **Dedicated** Architecture → ETL diletakkan di 1 mesin **sendiri** terpisah dari data source maupun data storage; penting jika mempertimbangkan keamanan, kompleksitas, dan skalabilitas (A)
  - **Shared** Architecture → ETL diletakkan dalam 1 mesin **bersama** dengan data source dan/atau data storage. (B & C)

BERDASARKAN SIAPA YANG MENGEKSTRAKSI DATA



1. Destination driven (PULL) → ETL menarik data dari data source secara periodik (daily, hourly, dst)
2. Source driven (PUSH) → menggunakan trigger
3. Manual extraction → lakukan export kemudian dijadikan input untuk proses ETL. Umumnya digunakan untuk analisis adhoc atau uji coba.
4. Menggunakan log reader. Log reader membaca file log untuk mengidentifikasi perubahan data, menarik data yang berubah.

Extract

METODE EKSTRAKSI

- Ada beberapa metode yang bisa digunakan

1. Full extraction
2. Incremental extraction
3. Real-time extraction
4. Fixed range extraction

FULL EXTRACTION

- Seluruh data pada data source diambil setiap kali proses ekstraksi dilakukan
- Digunakan dalam kondisi:
  - Tidak ada timestamp atau cara lain untuk mendeteksi perubahan data
  - Volume data tidak terlalu besar atau sistem mendukung pemrosesan skala besar
- Kelebihan: sederhana, tidak perlu mentracking perubahan
- Kekurangan: boros waktu dan sumber daya, ada resiko duplikasi

INCREMENTAL EXTRACTION

- Hanya memproses data yang berubah sejak ekstraksi terakhir
- Digunakan jika data source menyediakan change data capture (CDC)
- Kelebihan: efisien
- Kekurangan: butuh tahap khusus untuk memfilter data yang berubah

CHANGE DATA CAPTURE (CDC)

- **Timestamp-based CDC**
  - Mengambil data berdasarkan kolom last_updated atau modified_date
  - Paling sederhana
- **Log-based CDC**
  - Membaca perubahan dari log transaksi database (seperti binlog di MySQL atau SQLServer CDC)
  - Paling efisien, karena tidak perlu query tabel langsung
- **Trigger-based CDC**
  - Menggunakan trigger di database untuk mencatat perubahan ke tabel log
  - Lebih mudah diterapkan jika log tidak tersedia, tapi bisa membebani sistem

LOG-BASED CDC

	__$start_lsn	__$seqval	__$operation	__$update_mask	CustomerId	FirstName	LastName	Amount_purchased
1	0x0000002500000 5B80004	0x0000002500000 5B80003	2	0x0F	4	Ameena	Lalani	50000
2	0x0000002500000 5C00005	0x0000002500000 5C00002	1	0x0F	2	Shawn	McGuire	30000
3	0x0000002500000 5C80003	0x0000002500000 5C80002	4	0x04	3	Amy	Clarkson	40000
4	0x0000002500000 C080003	0x0000002500000 C080002	4	0x04	3	Amy	Blacksmith	40000
5	0x0000002500000 C100003	0x0000002500000 C100002	2	0x0F	5	Sponge	Bob	5000
6	0x0000002500000 C180003	0x0000002500000 C180002	2	0x0F	6	Donald	Duck	6000

REAL-TIME EXTRACTION

- Ekstraksi dilakukan setiap kali terdapat perubahan pada data source
- Digunakan jika data warehouse digunakan untuk real-time analytics seperti live dashboard
- Kelebihan: data langsung tersedia secara up-to-date
- Kekurangan: lebih kompleks secara teknis, membebani data source

FIXED RANGE EXTRACTION

- Data diambil berdasarkan rentang waktu tertentu yang telah ditetapkan (fixed), terlepas dari apakah data itu sudah pernah diambil sebelumnya atau tidak.
- Berdasarkan rentang waktu yang tetap, misalnya 1 Januari – 31 Maret
- Tidak bergantung waktu ekstraksi terakhir
- Umumnya dilakukan untuk:
  - Back-fill → memproses data historis
  - Re-processing → misalnya ada bug dalam proses transformasi
  - Validasi atau uji coba

PRINSIP PROSES EKSTRAKSI

- Data consistency
- Efficiency
- Data quality → lakukan validasi awal
- Auditability dan logging → simpan log (waktu, jumlah baris, berhasil/gagal)
- Scalability → hindari hardcode, gunakan parameter
- Scheduling → sebaiknya tidak dijalankan secara manual, gunakan scheduler

DATA CONSISTENCY

- Pastikan data yang diekstrak akurat dan sinkron dengan data source
- Jika proses ekstraksi memakan waktu lama, data bisa berubah di tengah proses → hasil ekstraksi tidak konsisten

EFFICIENCY

- Tujuannya adalah meminimalkan gangguan terhadap data source
- Sangat penting karena biasanya data source merupakan sistem yang sedang aktif digunakan oleh pengguna
  - Bisa menyebabkan lock atau blocking pada tabel → mengganggu transaksi lain.
  - Dalam kasus ekstraksi real-time, latency meningkat → memperlambat respon aplikasi utama.

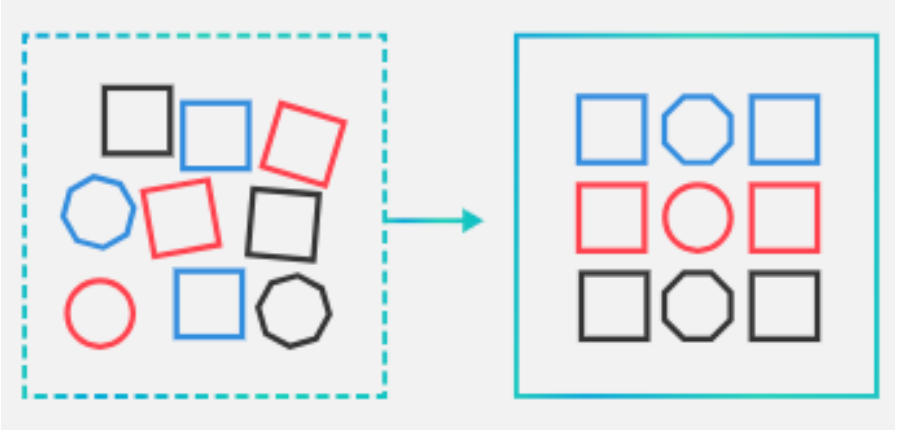
EFFICIENCY (2)

- Strategi:
  - Batching → ambil data dalam batch, tidak sekaligus
  - Gunakan database replication → replikasi data ke database sekunder lalu ekstraksi dilakukan dari sana
  - Lakukan ekstraksi di luar jam sibuk (off-peak hours)
  - Gunakan incremental extraction
  - Menggunakan Change Data Capture (CDC) → sehingga tidak membaca tabel utama secara langsung
  - Indexing & Query Optimization → pastikan kolom yang digunakan untuk filter (misalnya last_updated) sudah di-index, agar ekstraksi cepat dan efisien.

Transform

Definisi Transformasi Data

- Transformasi data adalah proses mengubah data dari suatu format, struktur, atau nilai ke bentuk lain agar sesuai untuk dianalisis, disimpan, atau digunakan dalam sistem target



Tujuan Transformasi Data

- Improving data quality
- Compliance: sesuai dengan database schema pada destination

Sifat Transformasi Data

Proses transformasi data dapat bersifat:

- Konstruktif, dimana data ditambahkan, disalin atau direplikasi
- Destruktif, dimana record dan field dihapus
- Estetika, dimana nilai-nilai tertentu dibakukan, atau
- Struktural, yang mencakup kolom yang diganti namanya, dipindahkan, dan digabungkan

Proses Transformasi Data

- Discovery: identifikasi dan memahami data dalam format sumber aslinya dan tipe data yang perlu diubah
- Mapping: memahami pada tingkat dasar, terkait pemetaan data yang akan dimodifikasi atau digabungkan.
- Code generation: kode yang diperlukan untuk menjalankan proses transformasi dibuat pada langkah ini menggunakan platform atau alat transformasi data.
- Execution: data akhirnya diubah ke dalam format yang dipilih dengan bantuan kode.
- Review: data yang diubah dievaluasi untuk memastikan konversi memberikan hasil yang diinginkan dalam hal format data.

Teknik Transformasi Data (1) Revising

- Normalisasi
- Konversi format → mengubah tipe data untuk memastikan kompatibilitas
- Penataan key
- Deduplikasi → mengidentifikasi dan menghapus duplikat
- Validasi data → menghapus atau memperbaiki data yang tidak valid atau konsisten
- Kolom yang berulang atau tidak digunakan dapat dihapus untuk meningkatkan performance dan readability

Teknik Transformasi Data (2) Manipulation

- Derivasi → menghasilkan data baru dari data yang sudah ada, melalui perhitungan, transformasi logis, atau fungsi tertentu
- Agregasi → menggabungkan banyak nilai menjadi satu nilai ringkasan
- Pivoting → konversi nilai kolom menjadi baris dan sebaliknya
- Menyortir, mengurutkan, dan mengindeks data
- Normalisasi → mengubah nilai agar berada dalam skala umum atau proporsional

Teknik Transformasi Data (3) Splitting

- Pemisahan data menjadi beberapa bagian untuk analisis granular.
- Pemisahan melibatkan pembagian satu kolom dengan beberapa nilai menjadi kolom terpisah dengan masing-masing nilai tersebut → memungkinkan pemfilteran berdasarkan nilai tertentu.

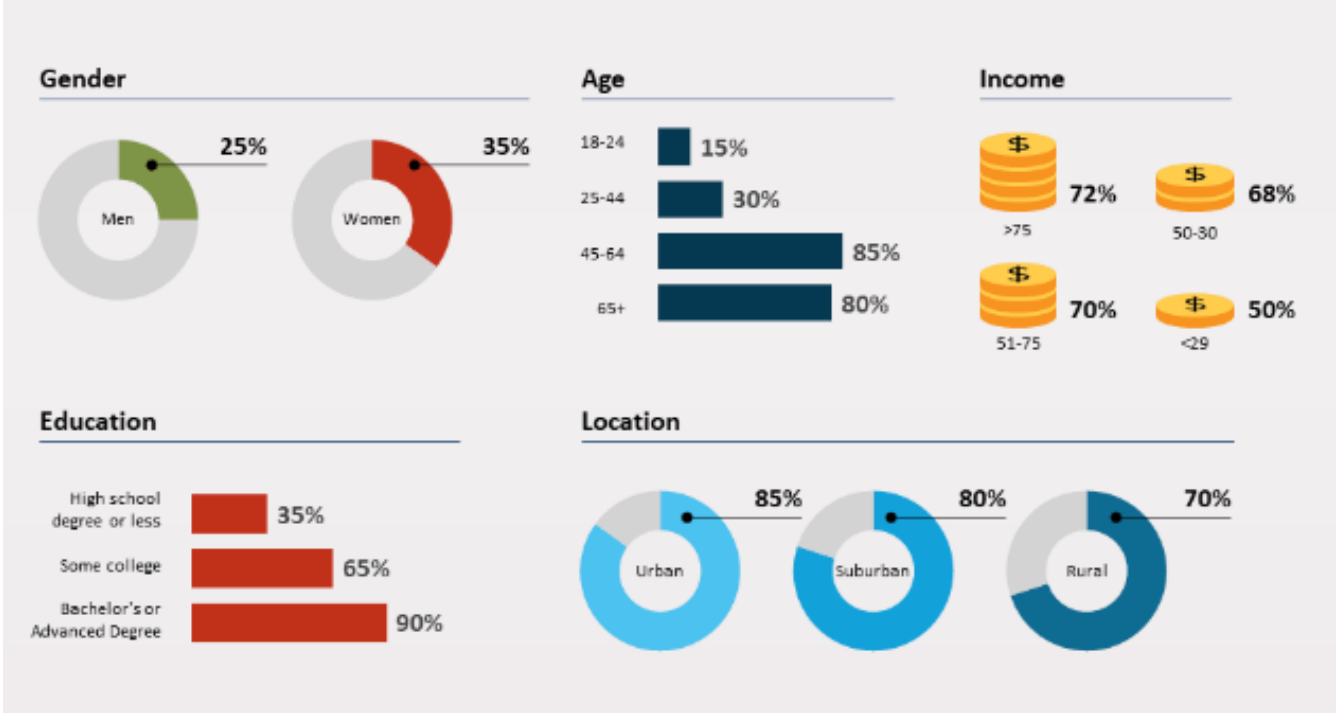
Teknik Transformasi Data (4) Combining/Integrating

- Proses menggabungkan dua atau lebih set data berdasarkan kunci atau relasi untuk membentuk satu kesatuan informasi yang lebih kaya dan utuh
- Biasanya dilakukan dengan JOIN, UNION, atau proses lookup.

Teknik Transformasi Data (6) Generalization

- Proses mengubah data dari bentuk spesifik (detail) menjadi bentuk yang lebih umum atau agregat → menyederhanakan data

- Salah satu cara yang dapat dilakukan → **discretization** (Mengubah data numerik kontinu menjadi kategori atau interval terbatas (diskrit))



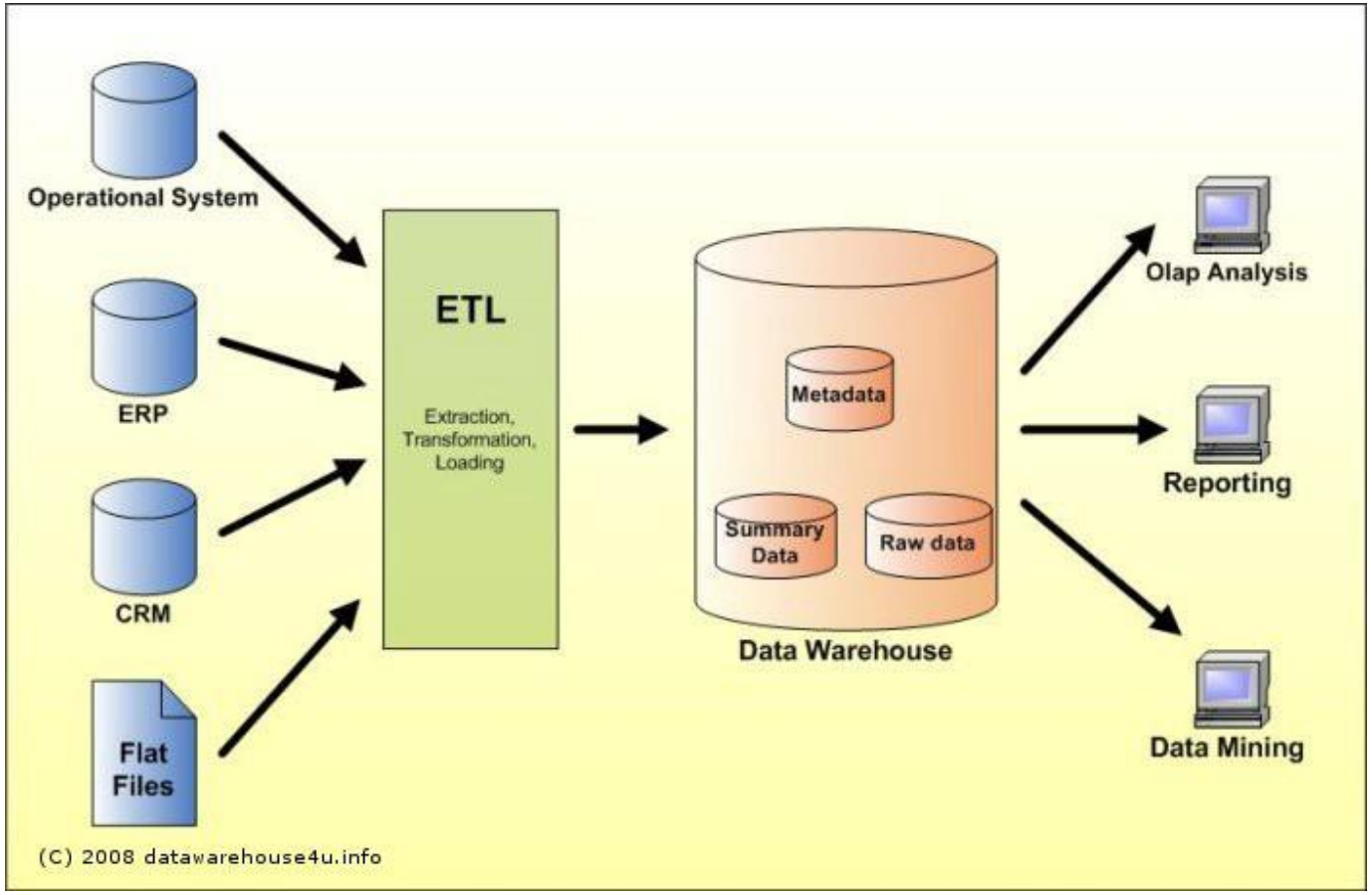
Tantangan dalam Data Transformation

- **Errors and inconsistency:** transformasi yang buruk dapat menghasilkan data yang cacat, atau migrasi yang gagal dan merusak data.
- **Resource intensive:** Proses transformasi memerlukan banyak resource. Beban komputasi yang sangat besar timbul ketika transformasi dilakukan di data warehouse lokal, yang akibatnya memperlambat operasi lainnya.
- **High cost of implementation:** Proses transformasi data adalah proses yang mahal. Tergantung pada kompleksitas proses bisnis, infrastruktur, dan perangkat lunak yang digunakan

https://www.tibco.com/glossary/what-is-data-transformation

Load

LOAD



Bagian dari proses ETL (Extract, Transform, Load) yang merupakan proses memasukkan data dalam jumlah besar yang sudah disesuaikan formatnya ke dalam tabel-tabel yang ada di dalam data warehouse.

Strategi Data Loading

1. **Initial Load / Full Load**
   Initial Load adalah data loading yang hanya dilakukan 1 kali di awal implementasi Data Warehouse. Data yang dimuat adalah semua data yang terkumpul pada data source sejak transaksi pertama hingga saat dilakukannya data loading

2. **Incremental Load**
   Incremental Load adalah data loading ke data warehouse yang dilakukan secara periodik, bisa setiap hari, setiap minggu, ataupun setiap bulan. Data yang dimuat adalah data yang terkumpul pada data source sejak proses load yang terakhir.

3. **Full Refresh**
   Full Refresh adalah strategi pemuatan data dengan cara mengosongkan satu atau lebih tabel pada data warehouse dan memuat data baru pada tabel-tabel tersebut.

Strategi Extraction + Strategi Loading

- Ada kombinasi fleksibel antara metode extraction dan load, tergantung pada kebutuhan dan desain ETL.
- Incremental Extraction + Incremental Load
  - Ambil hanya data yang berubah (misalnya pakai kolom last_updated).
  - Load hanya baris baru atau berubah ke warehouse.
  - Ideal untuk data besar dan sering berubah.
- Incremental Extraction + Full Load
  - Ambil data baru, lalu hapus semua isi tabel target dan isi ulang dengan data baru.
  - Mungkin dilakukan jika struktur data berubah drastis atau untuk staging table.

Strategi Extraction + Strategi Loading

- Full Extraction + Incremental Load
  - Ambil seluruh data sumber.
  - Gunakan logika deduplikasi atau upsert (insert/update) saat load.
  - Berguna saat source system tidak mendukung tracking perubahan.
- Full Extraction + Full Load
  - Seluruh data di-extract dan tabel target di-truncate lalu reload.
  - Mahal dari segi waktu dan sumber daya.
  - Cocok hanya untuk dataset kecil atau tabel lookup.

Loading Dimension Table

- Deteksi data baru atau berubah
- Lakukan transformasi dan cleansing
- Kelola Slowly Changing Dimension (SCD)
- Generate surrogate key
- Load ke dimension table (insert/update)

Loading Fact Table

- Ambil data transaksi
- Cek bahwa semua foreign key-nya valid (lookup ke dimensi)
- Tambahkan ID dimensi (surrogate key)
- Transformasi numerik (jika perlu: agregasi, rounding, dsb.)
- Insert ke fact table (append)