



Pertemuan 2 : Lingkungan Basis Data

Basis Data

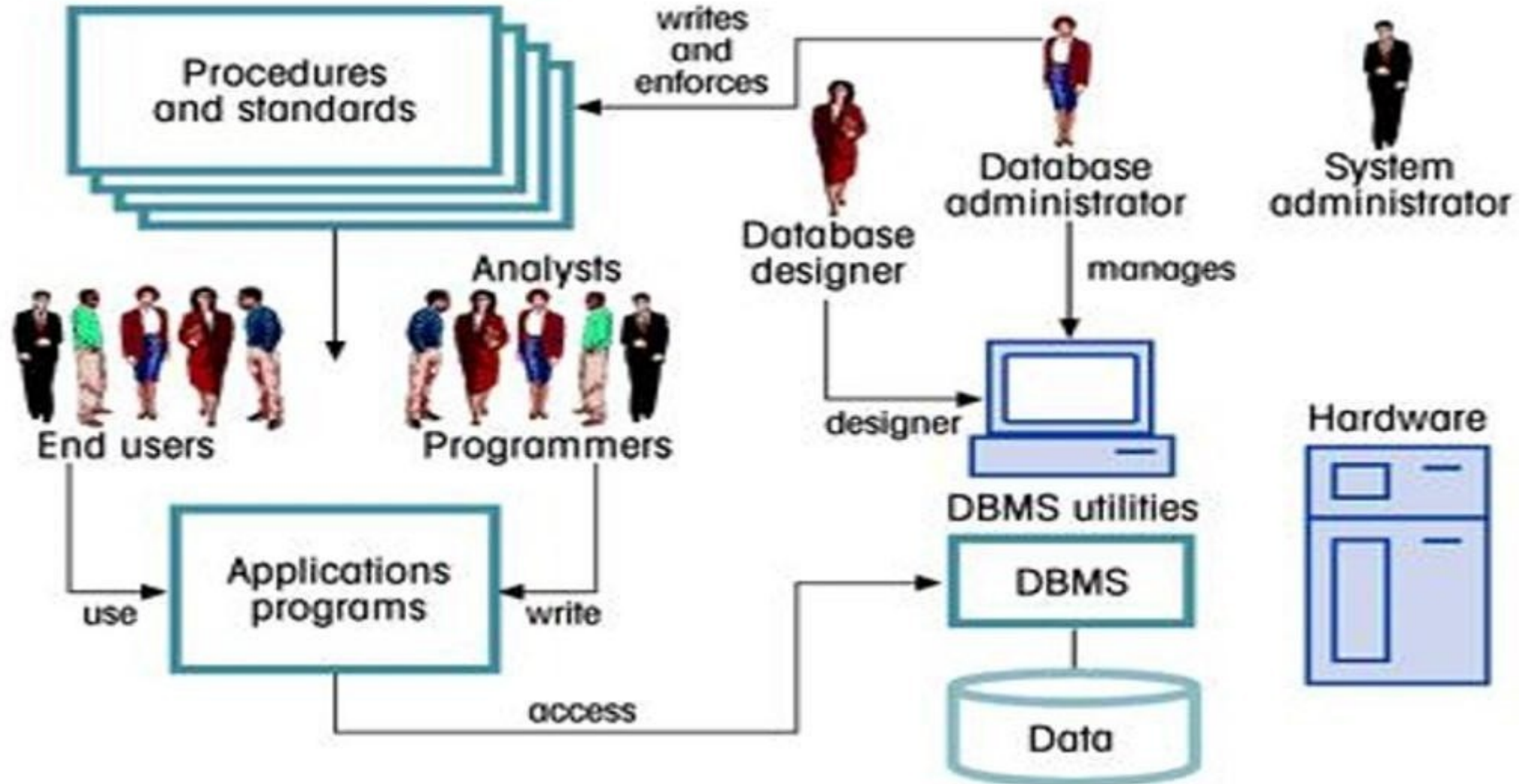
Agung Nugroho, M.Kom
Teknik Informatika – S1
Fakultas Teknik
Universitas Pelita Bangsa



Lingkungan Basis Data

Pertemuan 2

Lingkungan Basis Data



Lingkungan Basis Data

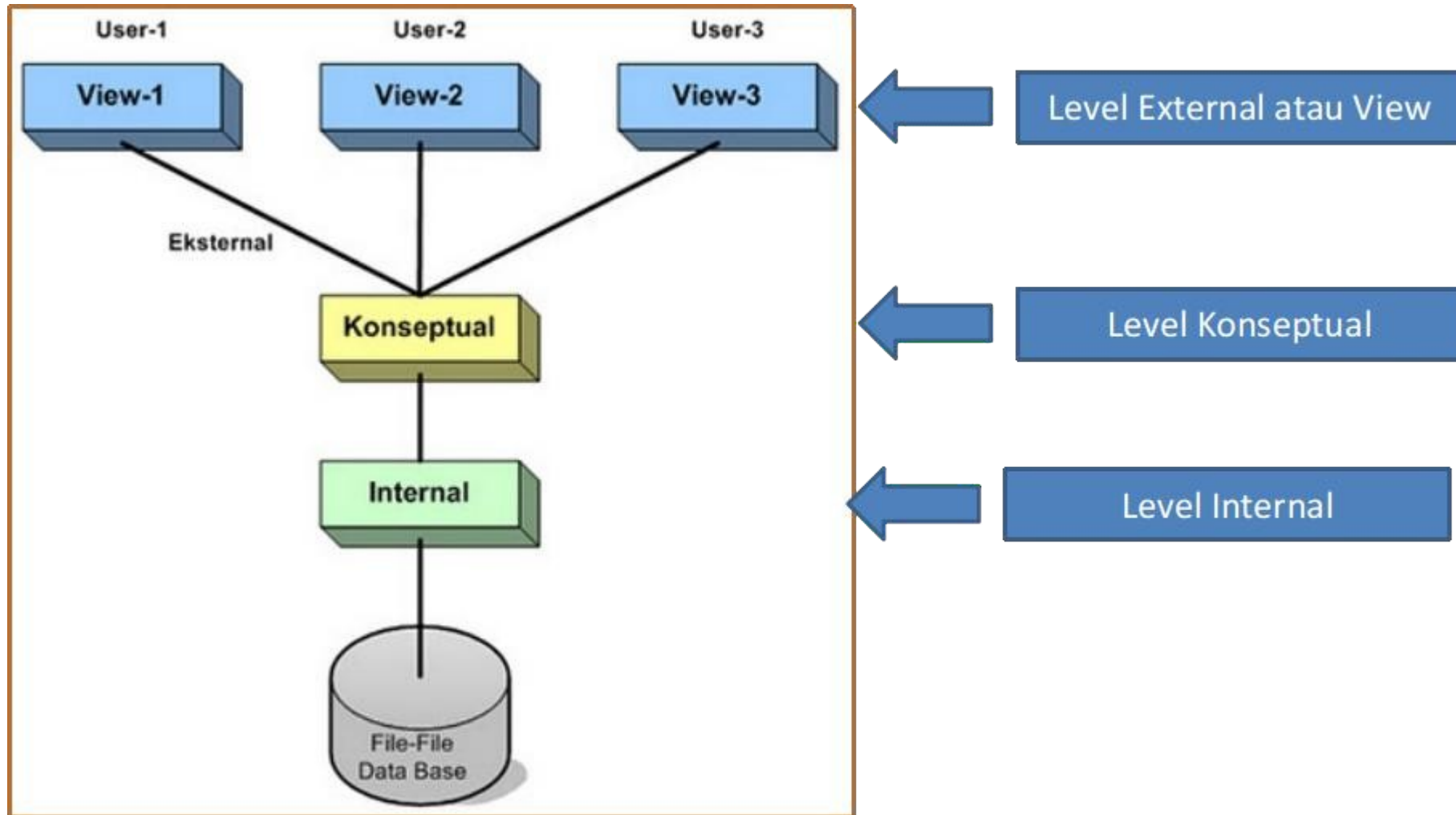
- Lingkungan basis data merupakan sebuah habitat di mana terdapat basis data untuk bisnis.
- Pengguna memiliki alat untuk mengakses data untuk keperluan bervariasi, seperti data mining, manipulasi data, dll
- Pengguna tidak punya hak mengakses data, baik secara fisik maupun logis.
- Tujuan utama dari sistem basis data adalah menyediakan pemakai melalui suatu pandangan abstrak mengenai data, dengan menyembunyikan detail dari bagaimana data disimpan dan dimanipulasikan.

Arsitektur Basis Data

Pada tahun 1975, badan standarisasi nasional Amerika ANSI-SPARC (American National Standards Institute – Standards Planning and Requirements Committee) menetapkan tiga level abstraksi dalam database, yaitu:

- Level Eksternal (external level) atau Level Pandangan (view level)
- Level Konseptual (conceptual level)
- Level Internal (internal level) atau Level Fisik (physical level)

Arsitektur Basis Data



Level Eksternal

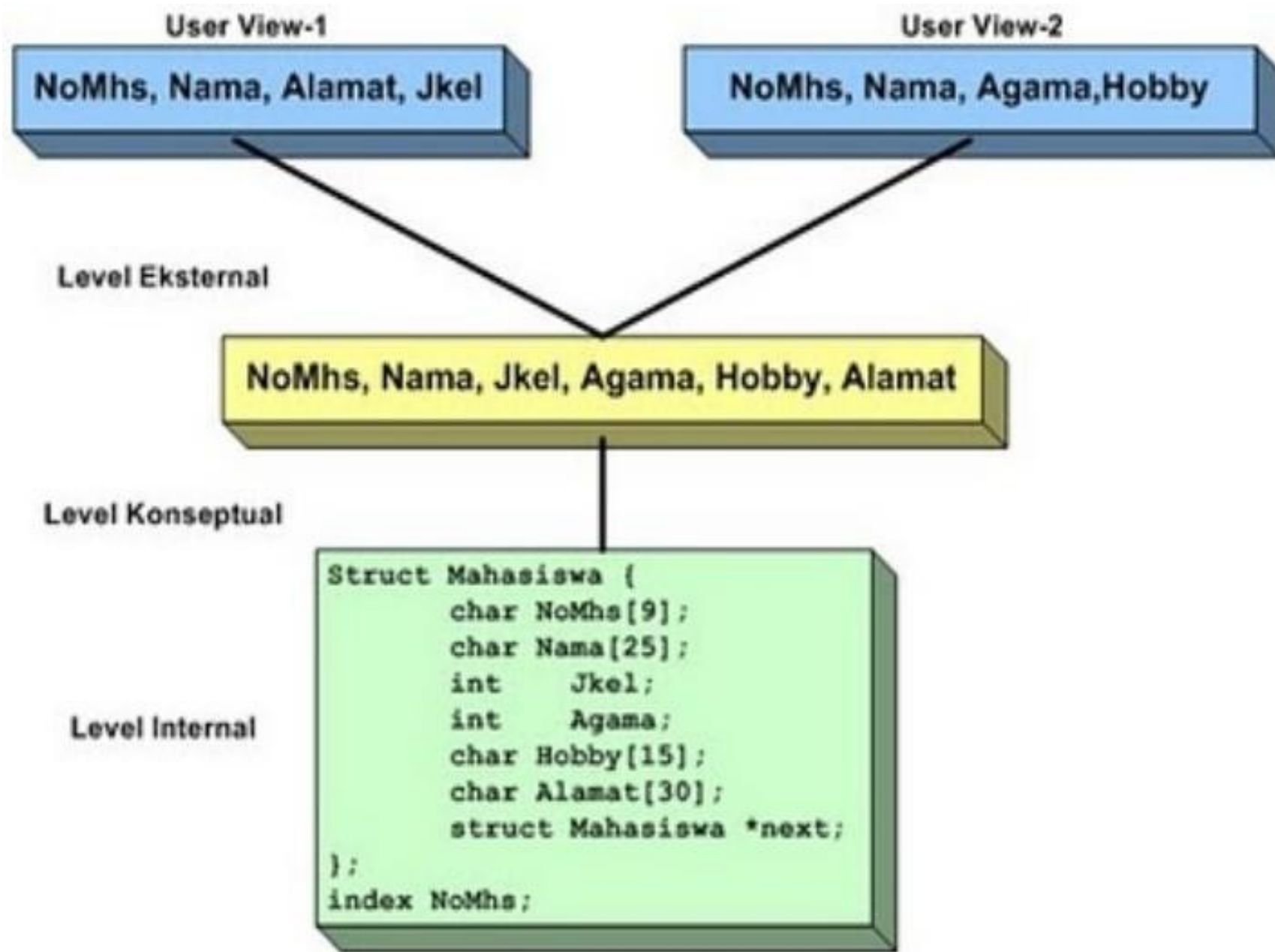
- Level Eksternal adalah level yang berhubungan langsung dengan pengguna database.
- Pada level ini pengguna (user) hanya bisa melihat struktur data sesuai dengan keperluannya sehingga setiap user bisa memiliki pandangan (view) yang berbeda dari user lainnya.
- Pada level ini pula dimungkinkan pandangan user berbeda dengan representasi fisik dari data, misalkan untuk data hari secara fisik data direkam dalam bentuk kode (1, 2, 3, dst) sedang user melihat data dalam bentuk teks nama hari (Ahad, Senin, Selasa, ...).
- Data yang dilihat oleh user seakan-akan berasal dari satu file, secara fisik mungkin diambil dari beberapa file yang berelasi.

Level Konseptual

- Level Konseptual adalah level dari para administrator database, pada level ini didefinisikan hubungan antar data secara logik, sehingga diperlukan struktur data secara lengkap.
- Para administrator database memahami bagaimana satu view dijabarkan dari beberapa file data, demikian pula pada saat perancangan database mereka dapat saja membagi data menjadi beberapa file agar dapat diakses dan disimpan secara efisien.

Level Internal

- Level Internal adalah level dimana data disimpan secara fisik dalam bentuk kode, teks, angka, bit.
- Pada level ini didefinisikan alokasi ruang penyimpanan data, deskripsi data dalam penyimpanan, kompresi data (agar lebih hemat), dan enkripsi data (agar lebih aman).



Data independence

Konsep dari level-level tersebut akan menambah pengertian mengenai kebebasan data/data independence.

Data independence dapat dibagi menjadi dua bagian :

1. Physical data independence.
2. Logical data independence.

Physical data independence

- Bahwa *internal schema* dapat diubah oleh DBA tanpa mengganggu *conceptual schema*.
- Dengan kata lain *physical data independence* menunjukkan kekebalan *conceptual schema* terhadap perubahan internal schema.

Logical data independence

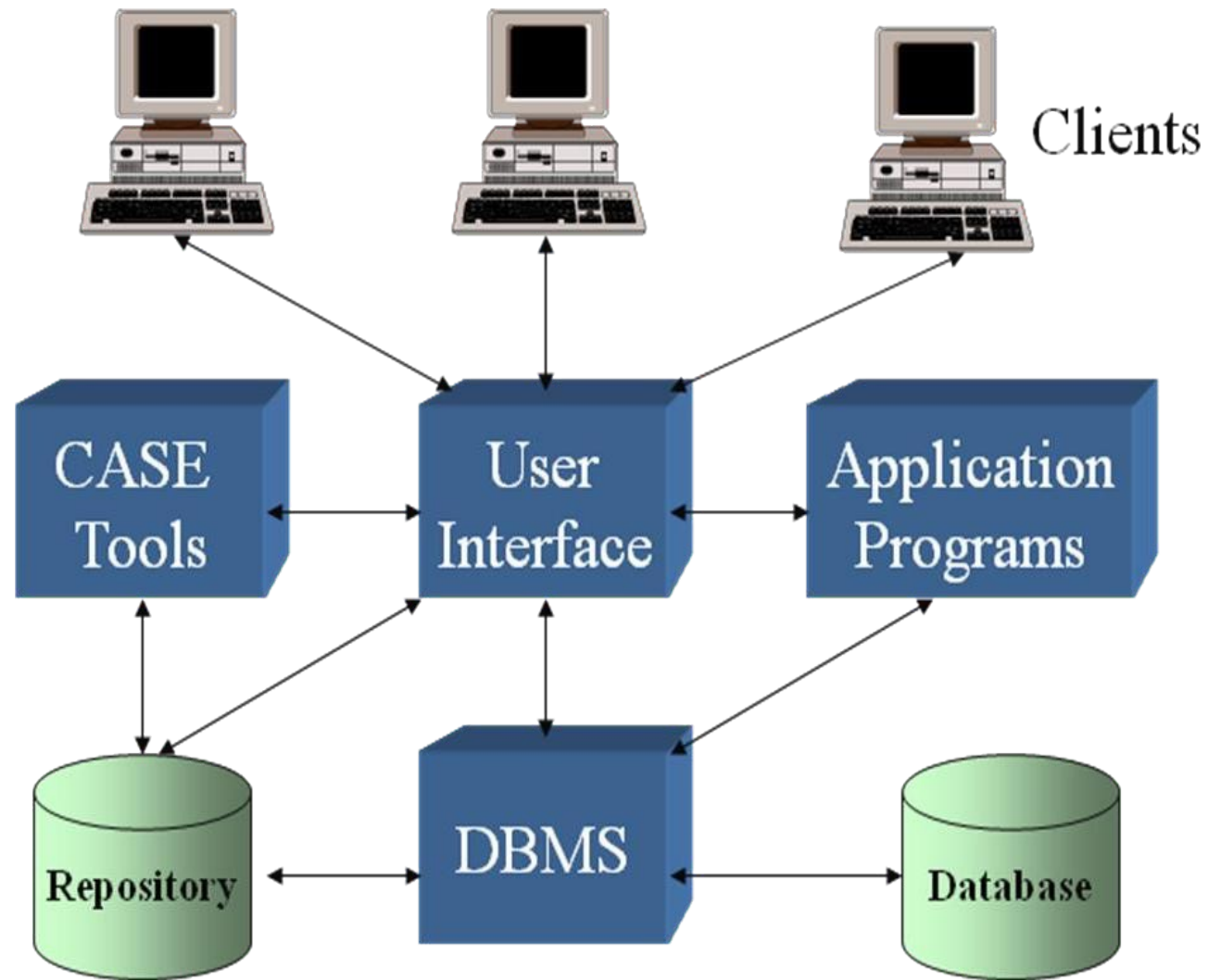
- Bahwa *conceptual schema* dapat diubah oleh DBA tanpa mengganggu *external schema*.
- Dengan kata lain *logical data independence* menunjukkan kekebalan *external schema* terhadap perubahan *conceptual schema*.

Konsep Data Indenpendence

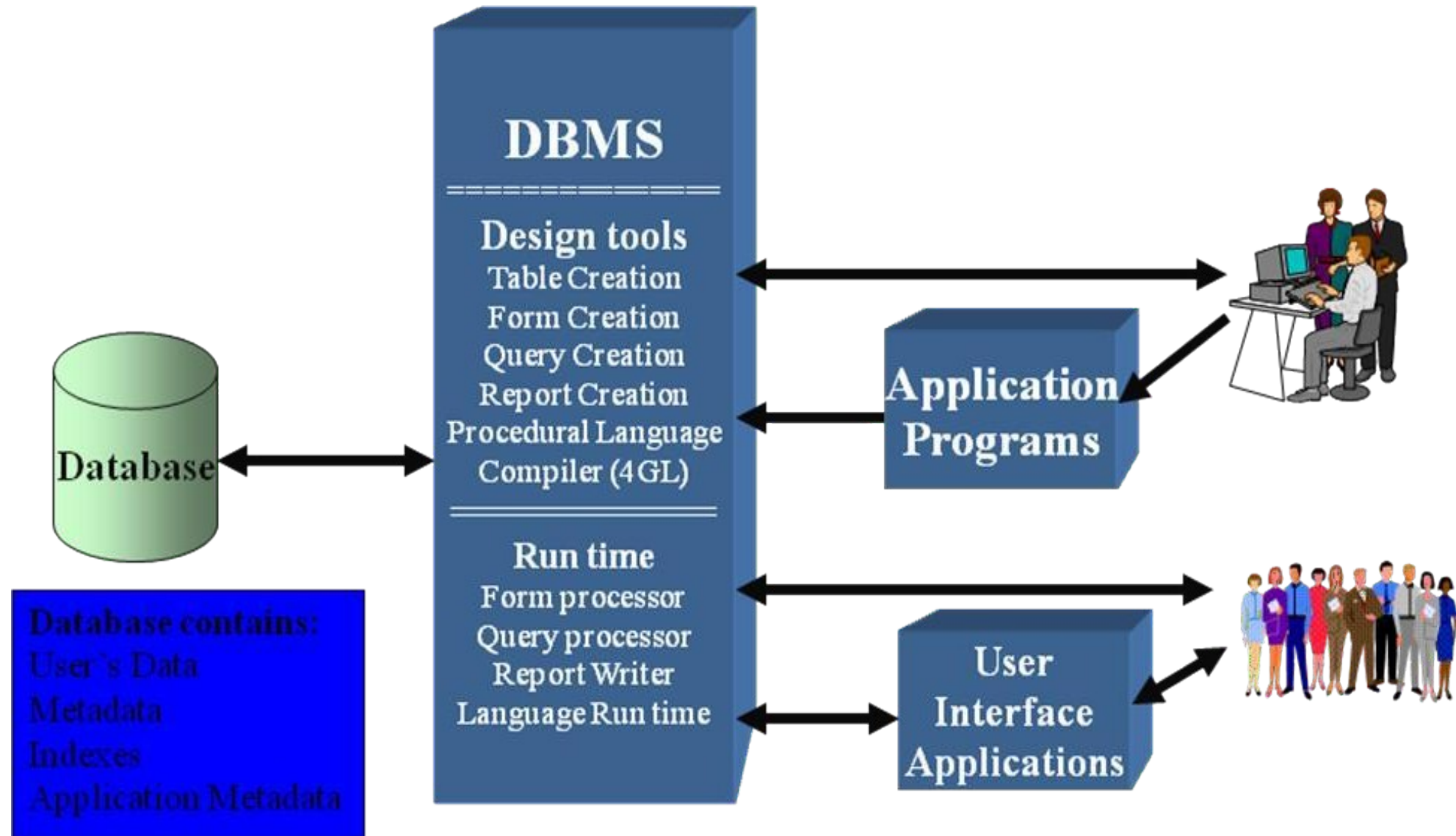
Prinsip data independence adalah salah satu hal yang harus diterapkan di dalam pengelolaan sistem basis data dengan alasan-alasan sbb:

- DBA dapat mengubah isi, lokasi, perwujudan dalam organisasi basis data tanpa mengganggu program-program aplikasi yang sudah ada.
- Pabrik / agen peralatan / software pengolahan data dapat memperkenalkan produk-produk baru tanpa mengganggu program-program aplikasi yang sudah ada.
- Untuk memindahkan perkembangan program-program aplikasi
- Memberikan fasilitas pengontrolan terpusat oleh DBA demi keamanan dan integritas data dengan memperhatikan perubahan-perubahan kebutuhan pengguna.

Lingkungan Basis Data



Komponen Basis Data



DBMS

Untuk mengelola data base diperlukan suatu perangkat lunak yaitu DBMS (Database Managemen System)

- **C.J. Date** : DBMS merupakan software yang menghandel seluruh akses pada database untuk melayani kebutuhan user.
- **S. Attre** : DBMS adalah software, hardware, firmware dan procedure-procedure yang manage database. Firmware adalah software yang telah menjadi modul yang tertanam pada hardware (ROM).
- **Gordon C. Everest** : DBMS adalah manajemen yang efektif untuk mengorganisasi sumber daya data.

Pengertian DBMS

DBMS adalah suatu system perangkat lunak yang memungkinkan user (pengguna) untuk membuat, memelihara, mengontrol, dan mengakses database secara praktis dan efisien.

Dengan DBMS, user akan lebih mudah mengontrol dan memanipulasi data yang ada

Komponen DBMS

- **File Manager:** mengelola ruang dalam disk dan struktur data yang dipakai untuk merepresentasikan informasi yang tersimpan dalam disk.
- **Database Manager:** menyediakan interface antara data low-level yang ada di basis data dengan program aplikasi dan query yang diberikan ke sistem.
- **Query Processor**, yang menterjemahkan perintah-perintah dalam query language ke perintah low-level yang dapat dimengerti oleh database manager.
- **DML Precompiler**, yang mengkonversi perintah DML yang ditambahkan dalam sebuah program aplikasi prosedur normal dalam bahasa induk.
- **DDL Compiler**, yang mengkonversi perintah-perintah DDL ke dalam sekumpulan tabel yang mengandung metadata. Tabel-tabel ini kemudian disimpan dalam kamus data.

Fungsi DBMS

- Pendefinisian data yang meliputi penentuan tipe, struktur dan batasan data yang akan disimpan dalam basis data.
- Kontruksi data yang meliputi proses penyimpanan data dalam database yang pengendaliannya diatur oleh DBMS
- Manipulasi data merupakan fungsi untuk menampilkan data, mengubah data serta menampilkan data yang ada dalam bentuk laporan
- Keamanan dan integritas data.
- Menyediakan data dictionary

Bahasa yang digunakan DBMS

- *Data sub language* adalah subset bahasa yang dipakai untuk operasi manajemen basis data.
- Secara umum maka setiap pengguna basis data memerlukan bahasa yang dipakai sesuai tugas dan fungsinya
- Dalam basis data secara umum dikenal 2 *data sub language*:
 - Data Definition Language (DDL)
 - Data Manipulation Language (DML)
 - Procedural DML
 - Non Procedural

Data Definition Language (DDL)

- Bahasa yang digunakan untuk mendefinisikan struktur atau kerangka dari basis data, di dalamnya termasuk record, elemen data, kunci elemen, dan relasinya
- Berupa perintah-perintah yang digunakan untuk menjelaskan objek dari database.
- Dengan kata lain DDL digunakan untuk mendefinisikan kerangka database.

Data Manipulation Language (DML)

- Bahasa yang digunakan untuk menjabarkan pemrosesan dari basis data, fasilitas ini diperlukan untuk memasukkan, mengambil, mengubah data.
- Berupa perintah-perintah yang digunakan untuk mengoperasikan atau memanipulasi isi database.
- Adapun perintah-perintah pada DML diantaranya :
Select, Insert, Update dan Delete.

Data Manipulation Language (DML)

Ada dua jenis DML, yaitu:

- **Procedural DML**

Digunakan untuk mendefinisikan data yang diolah dan perintah yang akan dilaksanakan.

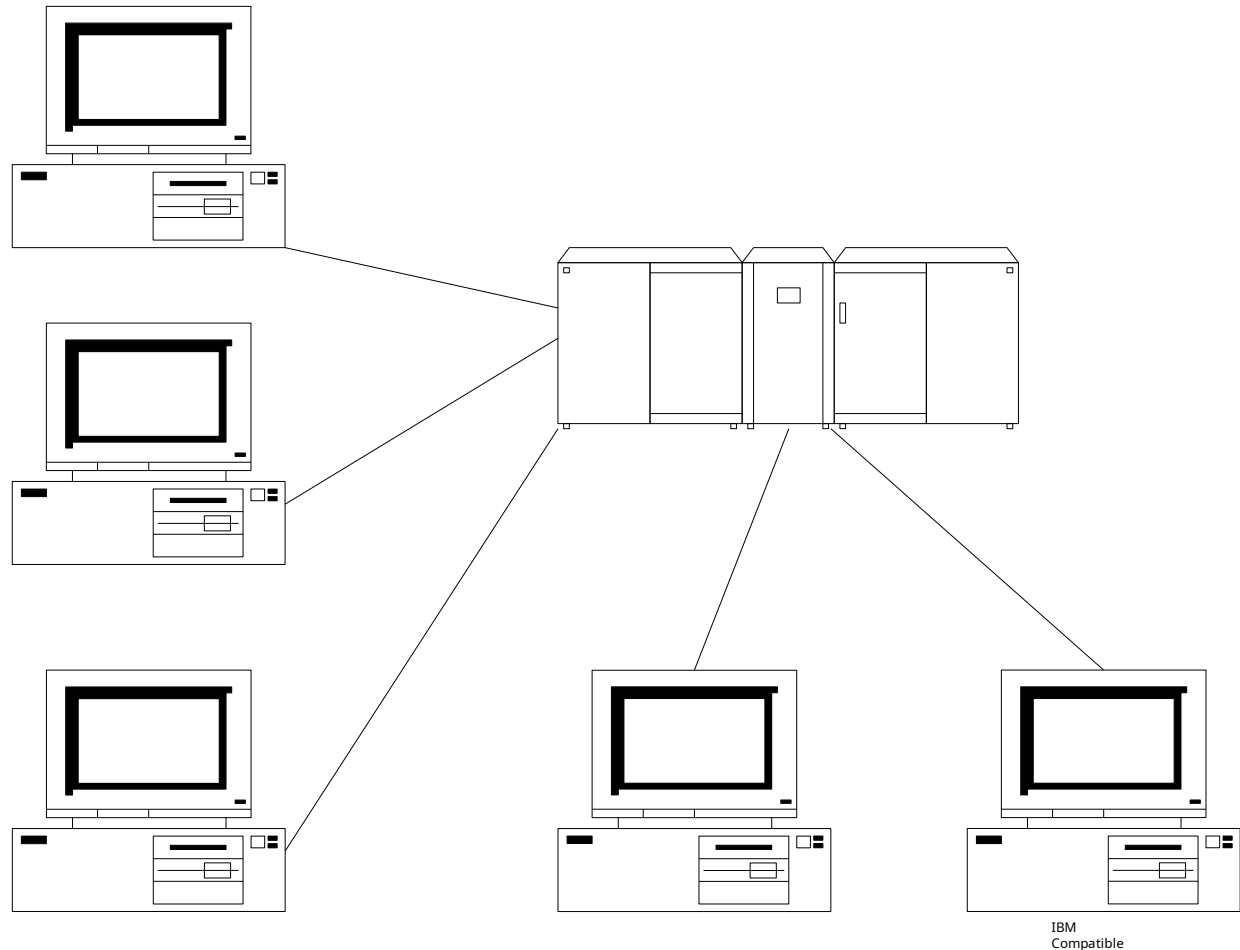
- **Non Procedural DML**

Digunakan untuk menjabarkan data yang diinginkan tanpa menyebutkan bagaimana cara pengambilannya

Arsitektur DBMS Multi User

Teleprocessing

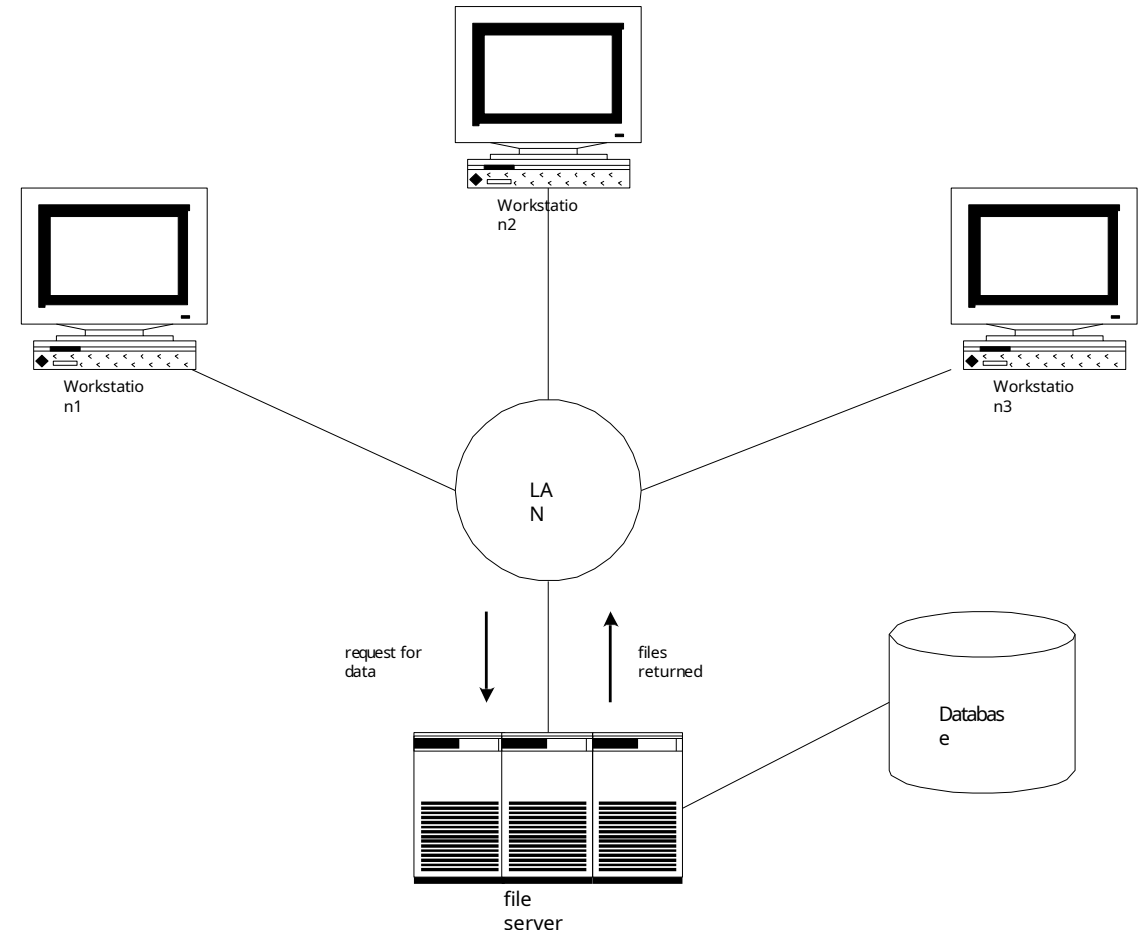
Arsitektur tradisional untuk sistem multi user adalah teleprocessing, dimana satu komputer dengan sebuah CPU dan sejumlah terminal seperti pada gambar disamping.



Arsitektur DBMS Multi User

File-Server

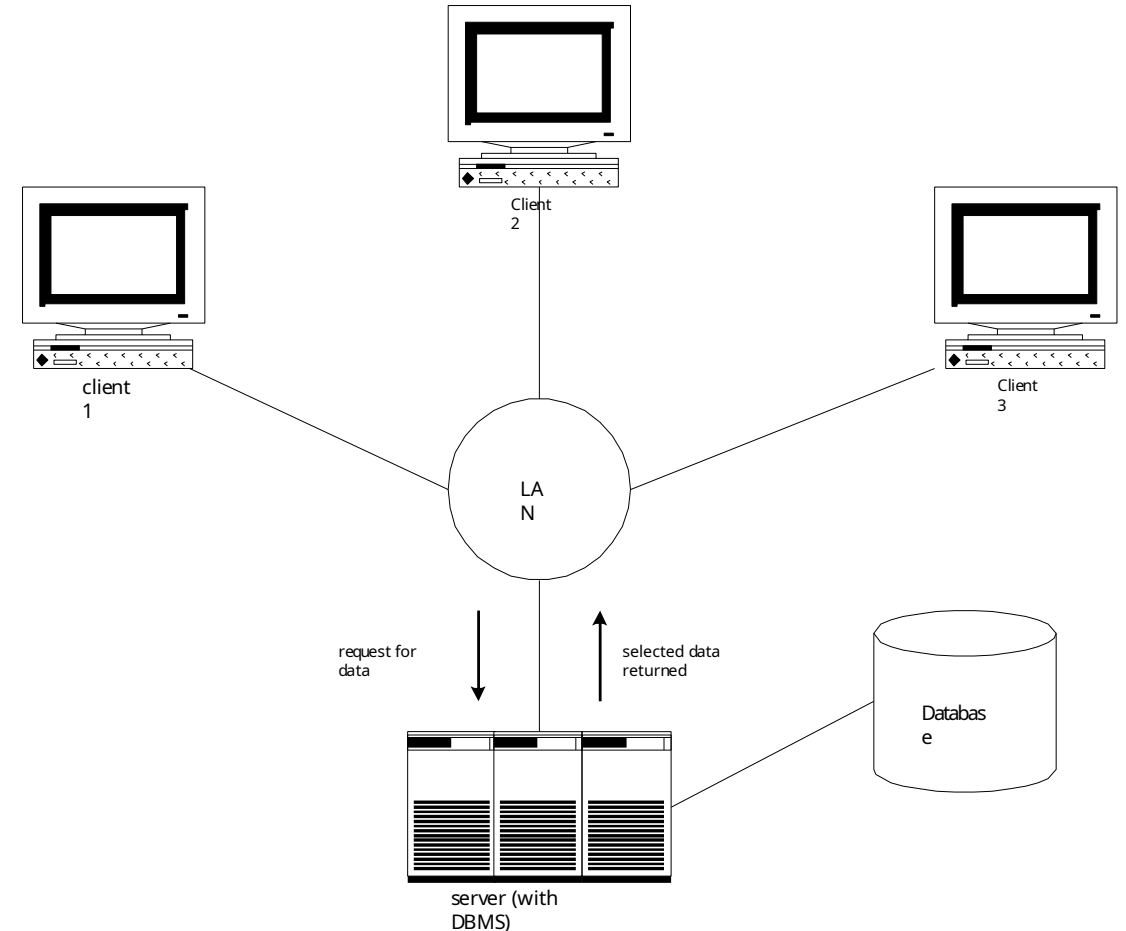
- Proses didistribusikan ke dalam jaringan, sejenis LAN (Local Area Network).
- File server mengendalikan file yang diperlukan oleh aplikasi dan DBMS.
- Meskipun aplikasi dan DBMS dijalankan pada masing-masing workstation, tetapi tetap meminta file dari file server jika diperlukan.



Arsitektur DBMS Multi User

Client-Server

- Client-server menunjukkan cara komponen software berinteraksi dalam bentuk sistem.
- Sesuai dengan namanya, ada sebuah pemroses client yang membutuhkan sumber dan sebuah server yang menyediakan sumbernya.



DATA DICTIONARY

- Data dictionary adalah tempat penyimpanan informasi yang menggambarkan data dalam database.
- Data dictionary biasa disebut juga dengan metadata atau data mengenai data.

Berisi tentang:

- nama-nama user yang mempunyai wewenang untuk menggunakan DBMS
- nama-nama item data yang ada dalam database
- jenis dan ukuran item data
- batasan untuk masing-masing item data

Model Data

Kumpulan perangkat konseptual untuk menggambarkan data, hubungan data, semantik data dan batasan data.

Beberapa Jenis Model Data:

- Model data File datar (*Flat-file data model*)
- Model data Hirarki (*Hierarchical data model*)
- Model data Jaringan (*Network data model*)
- Model data Relasional (*Relational data model*)
- Model data Keterhubungan Entitas (*Entity Relationship data model*)
- Model data Berorientasi Objek (*Object Oriented data model*)

Model data File datar

- Data flat-file terdiri dari satu atau lebih file yang dapat dibaca, yang secara normal berbentuk format file text.
- Informasi pada suatu flat-file disimpan sebagai fields, dengan fields-nya memiliki panjang konstan atau panjang bervariasi yang dipisahkan beberapa karakter (delimiter).

Contoh 1: Flat-file Model Data

- Model data flat-file dengan panjang fields-nya konstan.

1 2 3 4	5	6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5	6	7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
0 1 2 3		Mulyono		Progdi TI S1
1 2 3 4		Arifin		Progdi TI S1
2 3 4 5		Tyas Catur P.		Progdi TI S1
3 4 5 6		Ifan Riska		Progdi TI S1
4 5 6 7		Ayu Pertiwi		Progdi TI S1

Contoh 2: Flat-file Model Data

- Model data flat-file dengan panjang fields-nya bervariasi

0123; Mulyono; Progdi TI-S1

1234; Max Tetelepta; Progdi TI-S1

2345; Tyas Catur P.; Progdi TI-S1

3456; Ifan Riska; Progdi TI-S1

4567; Ayu Pertiwi; Progdi TI-S1

5678; Etika Kartika; Progdi TI-S1

6789; Anthoni Suteja; Progdi TI-S1

7890; Fikri Budiman; Progdi TI-S1

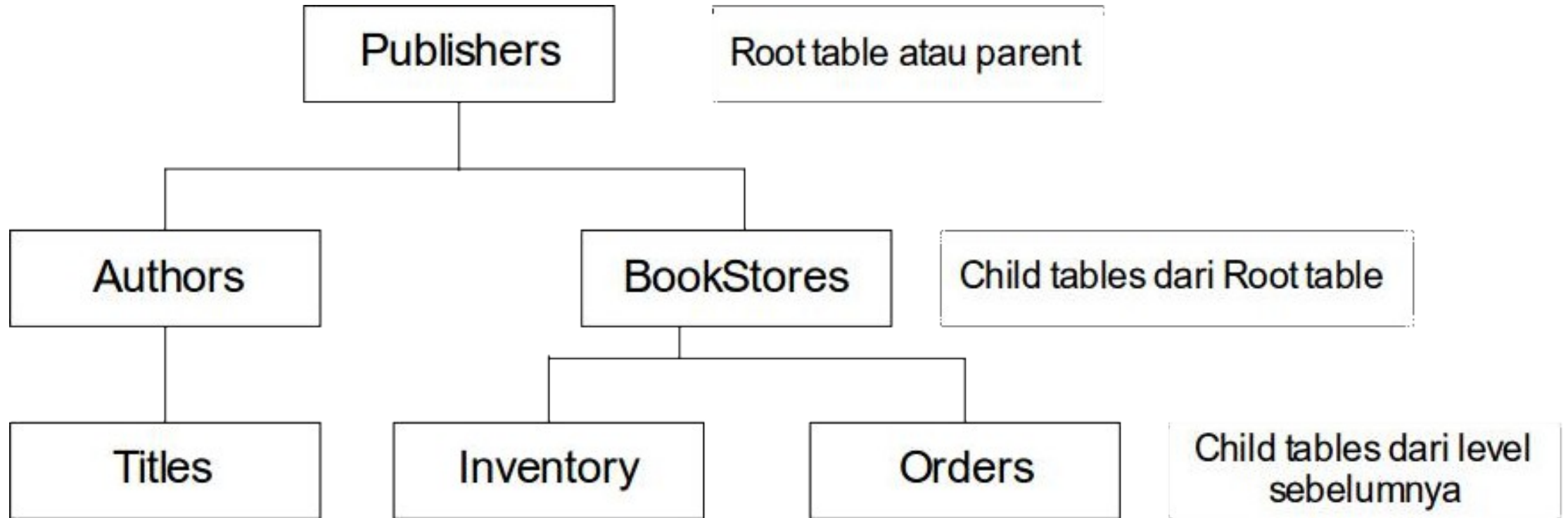
Kelemahan model data flat-file

- Flat-file tidak menggunakan struktur data yang dengan mudah dapat direlasikan
- Sulit untuk mengatur data secara efisien dan menjamin akurasi
- Lokasi fisik fields data dengan file harus diketahui
- Program harus dikembangkan untuk mengatur data

Model Data Hirarki

- Basis data Hirarki satu tingkat di atas basis data flat-file, dalam hal ini kaitanya dengan kemampuan untuk menemukan dan memelihara relasi antar kelompok data
- Arsitektur model data hirarki berdasarkan konsep hubungan parent/child
- Pada model data hirarki, suatu root table atau parent table berada apa struktur yang paling atas, terhubung ke child table yang dihubungkan dengan data

Contoh: Hirarki Model Data



Model Data Hirarki

Kelebihan basis data hirarki dibandingkan flat-file:

- Data dapat dengan cepat dilakukan retrieve
- Integritas data mudah dilakukan pengaturan

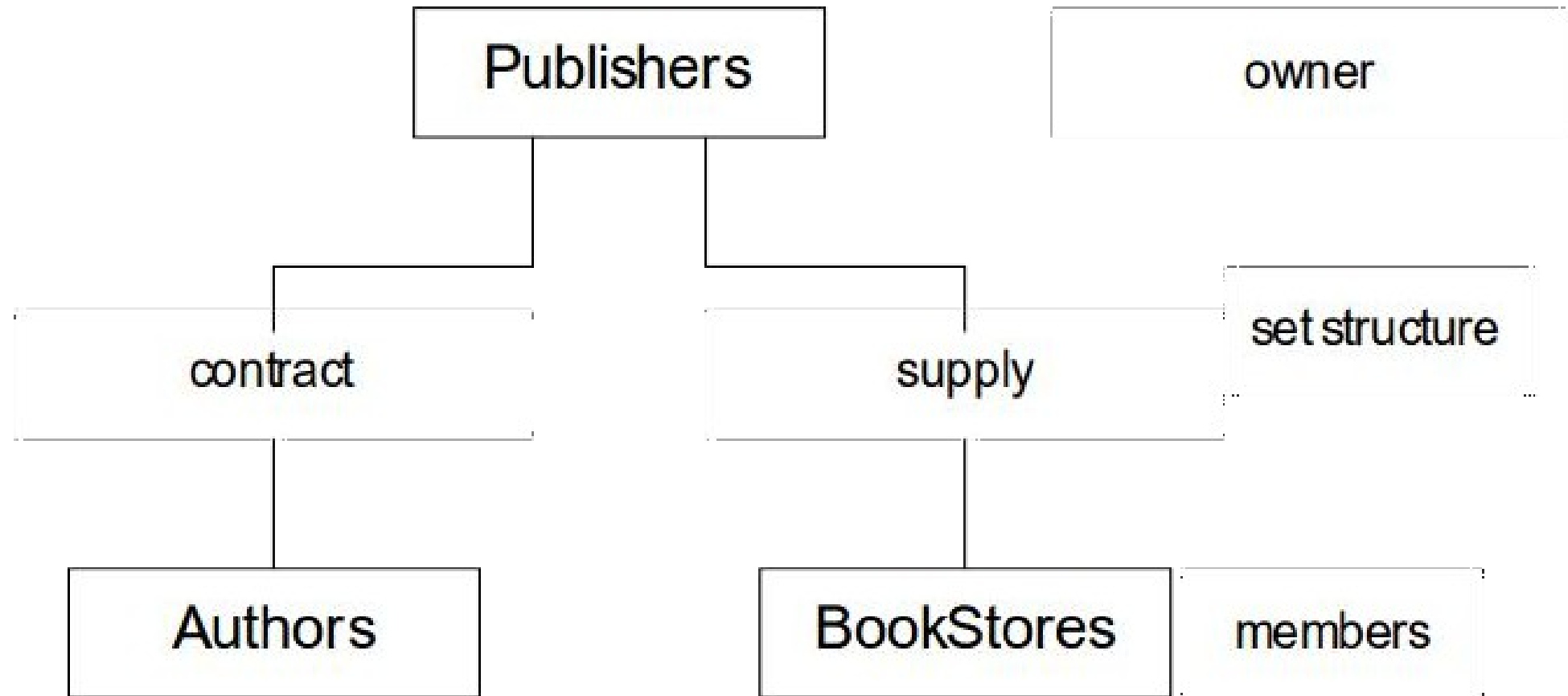
Kelemahan basis data hirarki dibandingkan flat-file:

- Pengguna harus sangat familiar dengan struktur basis data
- Terjadi redudansi data

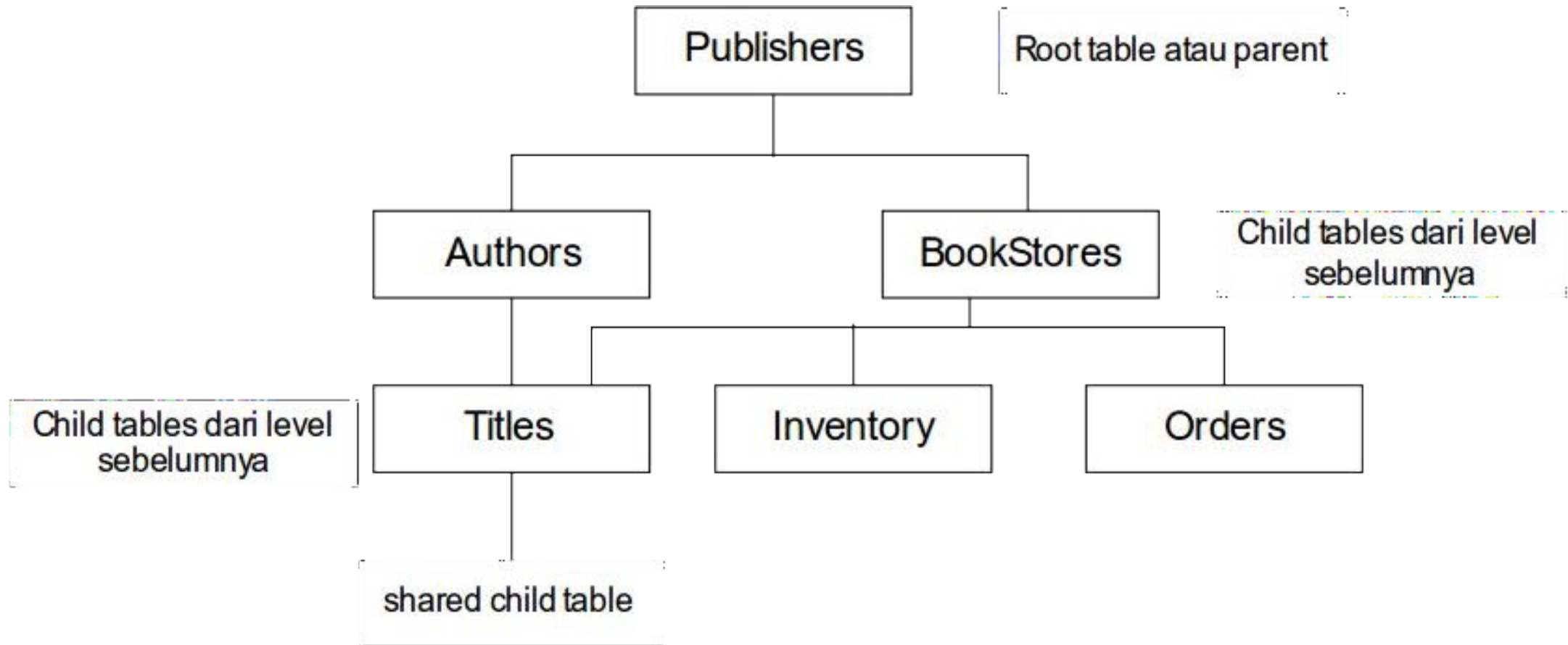
Model Data Jaringan

- Model basis data jaringan merupakan perbaikan dari model basis data hirarki, yaitu dengan menambahkan kemampuan root table untuk melakukan share relationships dengan child tables.
- Dalam hal ini child table dapat memiliki banyak root table dan untuk melakukan akses terhadap child table, tidak dibutuhkan lagi untuk mengakses root table terlebih dahulu.

Contoh 1: Model Data Jaringan



Contoh 2: Model Data Jaringan



Model Data Jaringan

Kelebihan model data jaringan:

- Data lebih cepat diakses
- User dapat mengakses data dimulai dari beberapa tabel
- Mudah untuk memodelkan basis data yang kompleks
- Mudah untuk membentuk query yang kompleks dalam melakukan retrieve data.

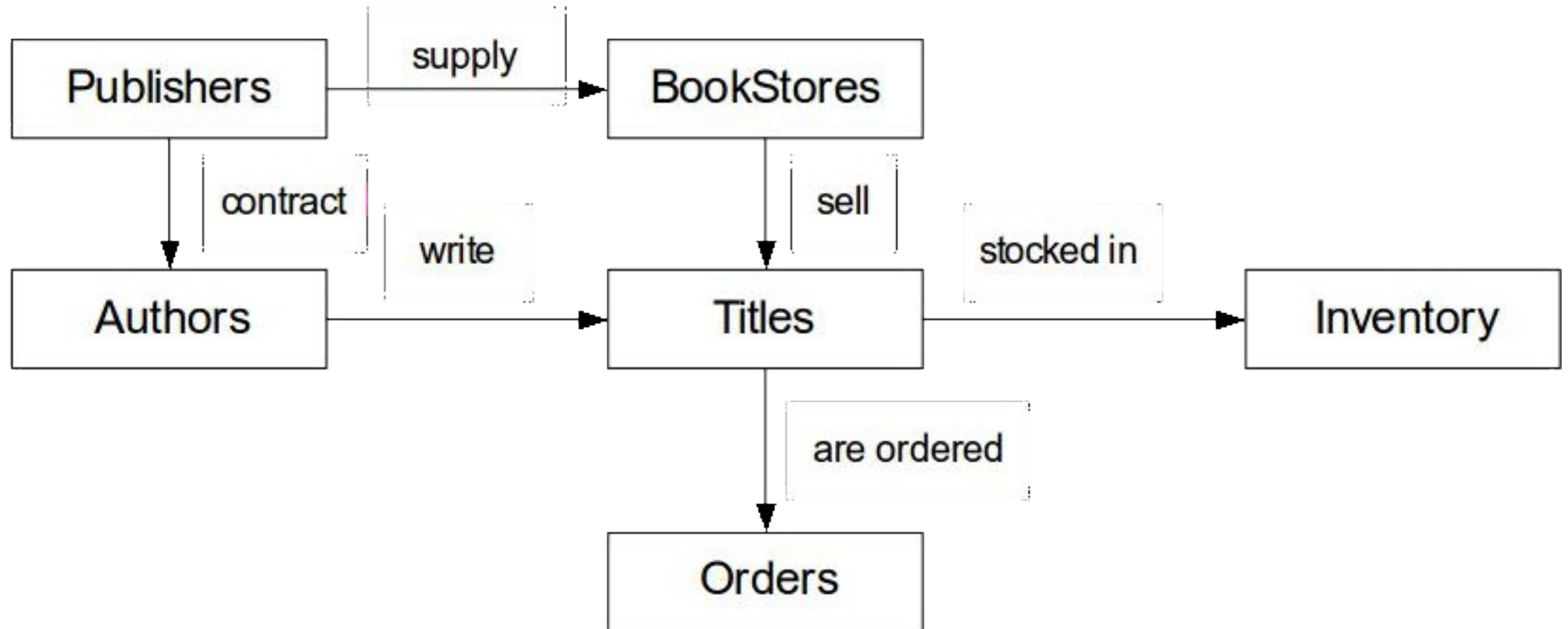
Kelemahan basis data jaringan:

- Struktur basis datanya tidak mudah untuk dilakukan modifikasi
- Perubahan struktur basis data yang telah didefinisikan akan mempengaruhi program aplikasi yang mengakses basis data
- User harus memahami struktur basis data.

Model Data Relasional

- Model basis data relasional merupakan model basis data yang paling populer banyak digunakan sekarang ini
- Unit utama yang disimpan pada basis data adalah berbentuk tabel atau kelompok data yang saling berhubungan
- Tabel terdiri baris dan kolom, baris adalah merepresentasikan tuple atau record pada tabel, dan kolom merepresentasikan fields pada tabel
- Tabel dapat berhubungan dengan tabel yang lain dengan menggunakan kunci

Contoh: Model Data Relasional



Model Data Relasional

Kelebihan basis data relasional:

- Data sangat cepat diakses
- Struktur basis data mudah dilakukan perubahan
- Data direpresentasikan secara logik, user tidak membutuhkan bagaimana data disimpan.
- Mudah untuk membentuk query yang kompleks dalam melakukan retrieve data
- Mudah untuk mengimplementasikan integritas data
- Data lebih akurat
- Mudah untuk membangun dan memodifikasi program aplikasi
- Telah dikembangkan Structure Query Language (SQL).

Model Data Relasional

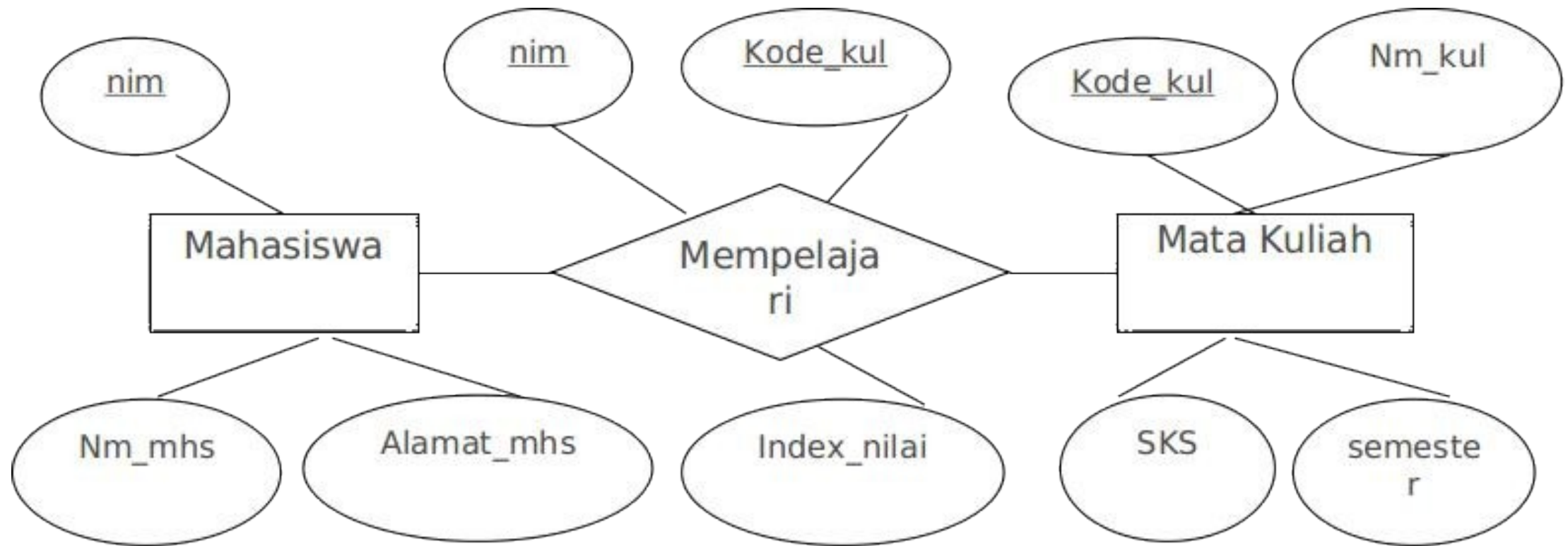
Kelemahan basis data relasional:

- Kelompok informasi/tables yang berbeda harus dilakukan joined untuk melakukan retrieve data
- User harus familiar dengan relasi antar tabel
- User harus belajar SQL.

Model data Keterhubungan Entitas (ER-D)

- Menjelaskan hubungan antar data dalam sistem basis data berdasarkan suatu persepsi bahwa real world terdiri dari obyek-obyek dasar yang mempunyai hubungan relasi antara obyek-obyek tersebut
- Relasi antara obyek dilukiskan dengan menggunakan simbol-simbol grafis tertentu

Contoh: Model Data ERD



Keterangan simbol ERD



: menunjukkan obyek dasar/entitas
(entity)



: menunjukkan relasi



: menunjukkan atribut dari obyek
dasar/entitas

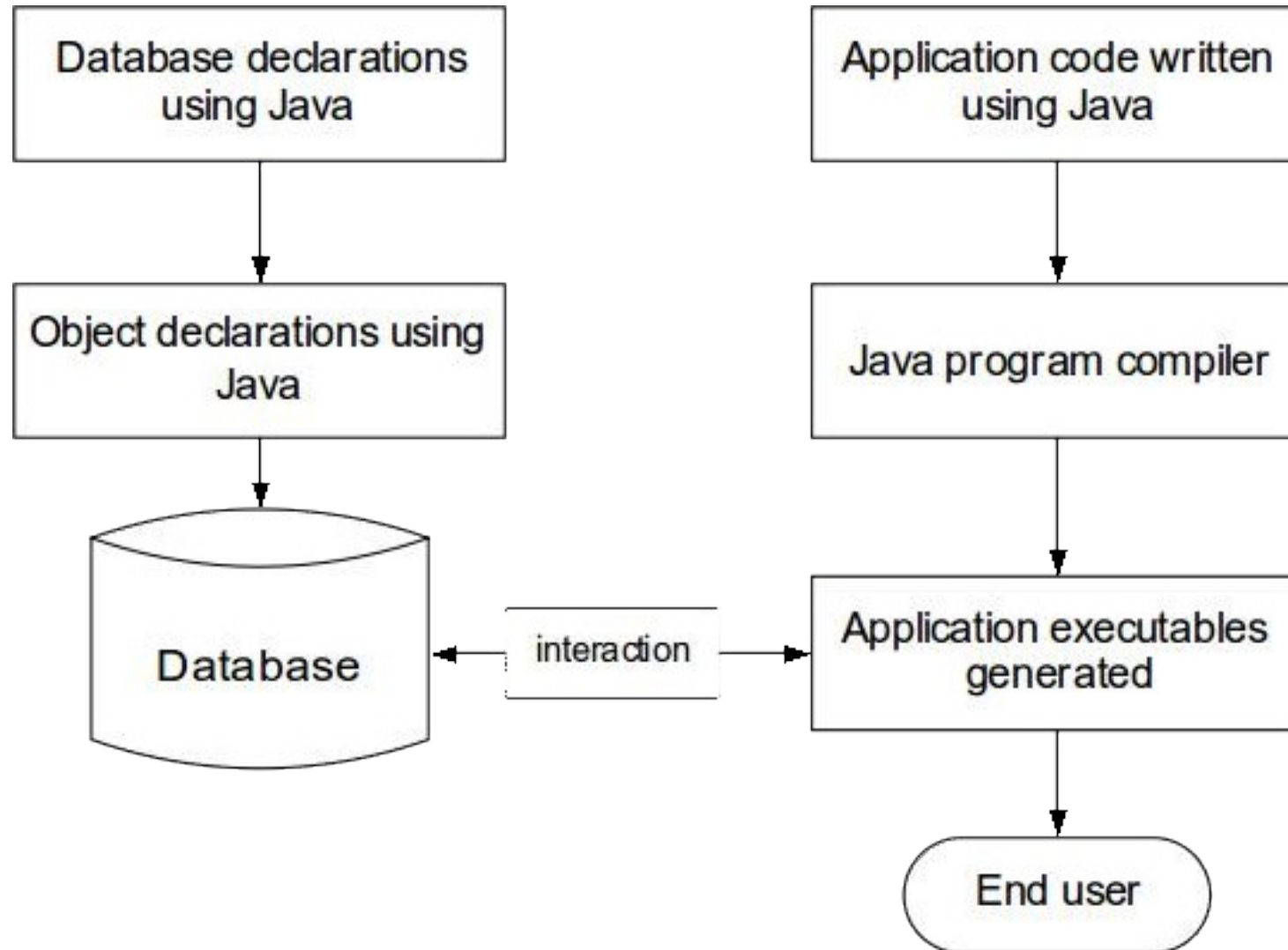


: menunjukkan adanya relasi/link

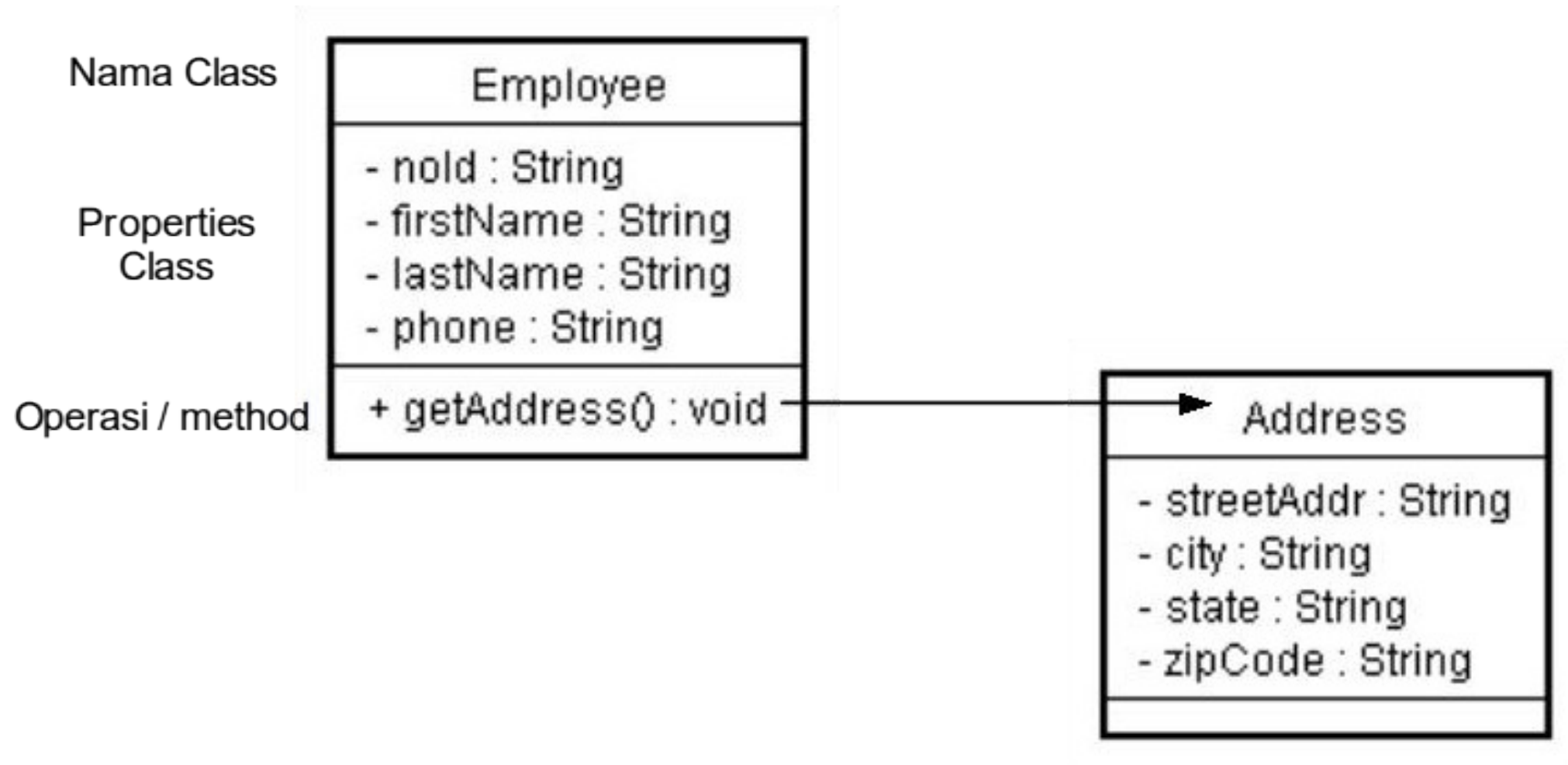
Model Data Berorientasi Obyek

- Model basis data berorientasi objek adalah suatu model basis data, dimana data didefinisikan, disimpan, dan diakses menggunakan pemrograman berorientasi objek.
- Basis data berorientasi objek didefinisikan dengan menggunakan bahasa pemrograman berorientasi objek, yaitu bahasa Java.
- Aplikasi End user juga di bangun dengan menggunakan bahasa berorientasi objek.
- Object database management system digunakan untuk membuat link antara basis data dan aplikasi.

Contoh: Model Data Berorientasi Obyek



Relasi pada basis data berorientasi obyek



Model Data Berorientasi Obyek

Kelebihan basis data berorientasi objek:

- Programmer hanya dibutuhkan memahami konsep berorientasi objek untuk mengkombinasikan konsep berorientasi objek dengan storage basis data relasional
- Objek dapat dilakukan sifat pewarisan dari objek yang lain
- Secara teoritis mudah untuk mengatur objek
- Model data berorientasi objek lebih kompatibel dengan tools pemrograman berorientasi objek.

Model Data Berorientasi Obyek

Kelemahan basis data berorientasi objek:

- User harus memahami konsep berorientasi objek, karena basis data berorientasi objek tidak dapat bekerja dengan metoda pemrograman tradisional



Question?

Tugas

Buat resume penjelasan dari materi pembelajaran pertemuan ini dalam bentuk video presentasi yang diunggah di youtube.

Ketentuan:

- Video menampilkan wajah presenter
- Durasi video 5-10 menit
- Submit link youtube pada kolom tugas di eCampus.
- Waktu pengerjaan selama satu pekan.



Terimakasih

Agung Nugroho
agung@pelitabangsa.ac.id