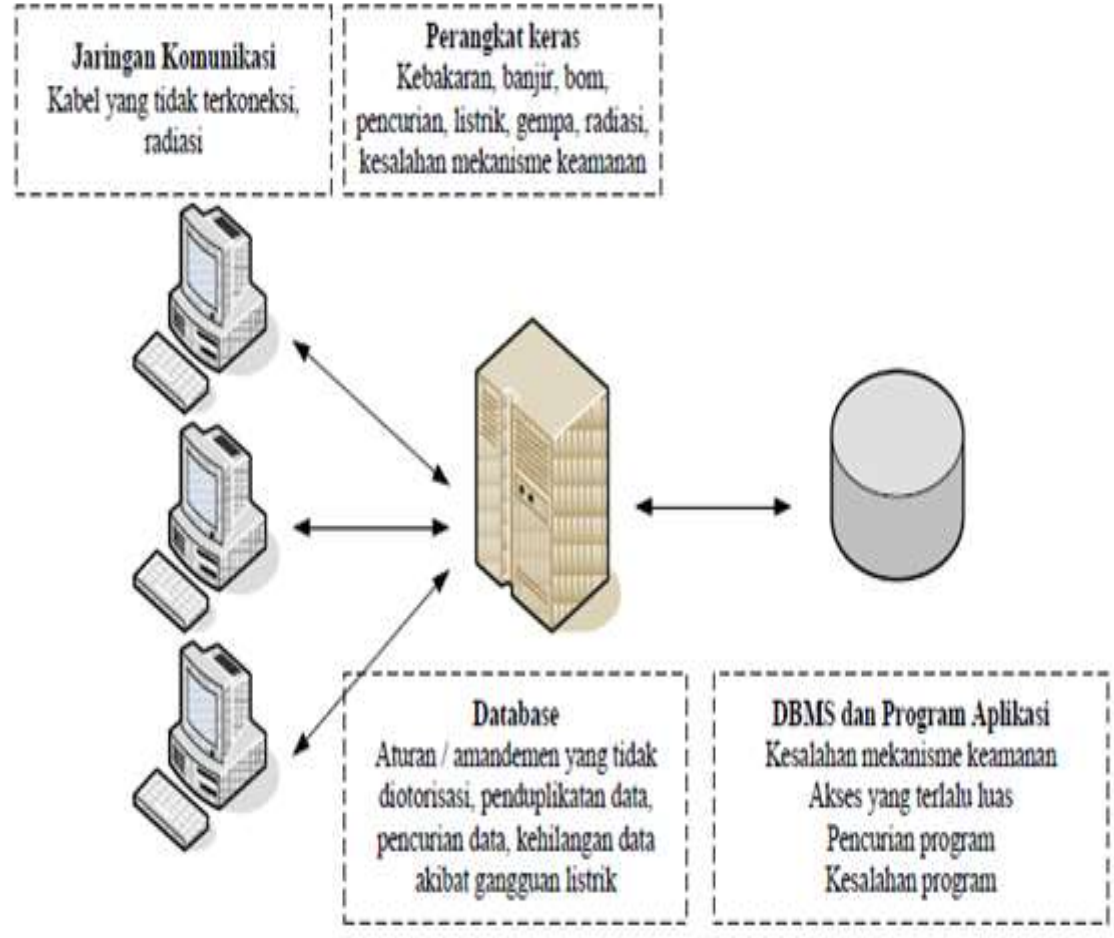
PENGAMANAN BASIS DATA

Mengapa perlu pengamanan basis data?

- Data yang disimpan dalam basis data perlu dilindungi dari akses yang tidak diijinkan.
- Perusakan atau pengubahan data yang merugikan.
- Menimbulkan inkonsistensi data secara tidak sengaja.

Keamanan Basis Data

Keamanan pada basis data merupakan **suatu proteksi terhadap pengrusakan data dan pemakaian data oleh pemakai yang tidak punya kewenangan.**



Penyalahgunaan Basis Data

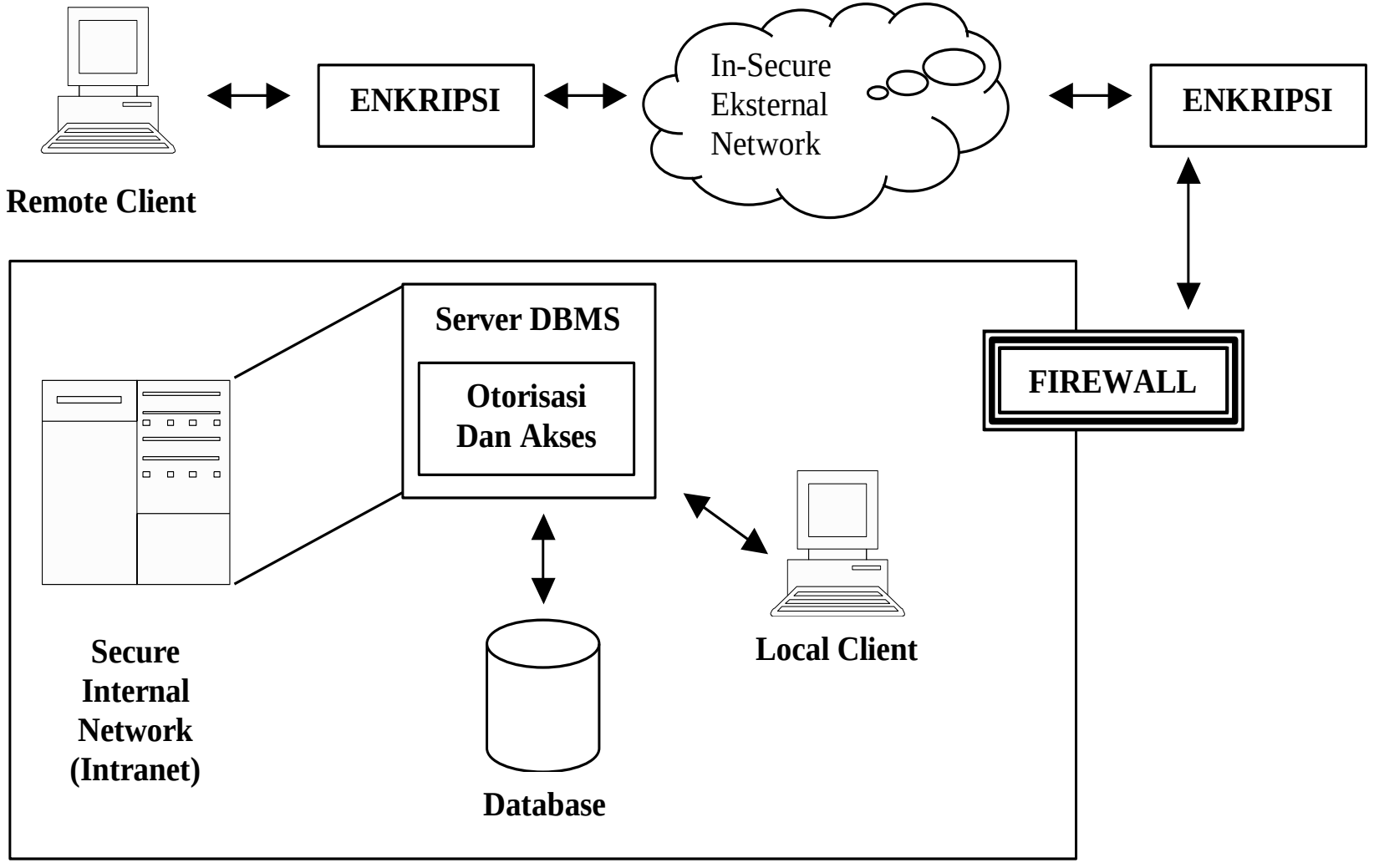
- **Tidak Disengaja**, jenisnya :
  - Kerusakan selama proses transaksi.
  - Inkonsistensi yang disebabkan oleh akses basis data yang konkuren.
  - Inkonsistensi yang disebabkan oleh pendistribusian data pada beberapa komputer.
  - Logika error yang mengancam kemampuan transaksi untuk mempertahankan konsistensi basis data.

- **Disengaja**, jenisnya :
  - Pengambilan data atau pembacaan data oleh pihak yang tidak berwenang (mencuri informasi).
  - Pengubahan data oleh pihak yang tidak diijinkan.
  - Penghapusan data oleh pihak yang tidak diijinkan.

Tingkatan Pada Keamanan Basis Data

- *Physical* ⇒ lokasi-lokasi dimana terdapat sistem komputer haruslah aman secara fisik terhadap serangan *destroyer*.
- *User* ⇒ wewenang *user* harus dilakukan dengan berhati-hati untuk mengurangi kemungkinan adanya manipulasi oleh *user* lain yang otoritas.
- Sistem Operasi ⇒ kelemahan entitas ini memungkinkan pengaksesan data oleh *user* tak berwenang, karena hampir seluruh jaringan sistem basis data berjalan secara *on-line*.
- Sistem Basisdata ⇒ Pengaturan hak pengguna yang baik.

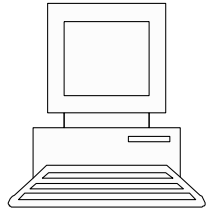
Skema Utama Mekanisme Keamanan Basis Data *on-line*



Aspek untuk dukungan keamanan Basis Data

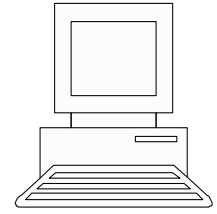
- *Network security* $\Rightarrow$ fokus kepada saluran pembawa informasi.
- *Application security* $\Rightarrow$ fokus kepada aplikasi itu sendiri.
- *Computer security* $\Rightarrow$ fokus kepada keamanan dari komputer (*end system*) yang digunakan.

Batasan *Privillege-user* **untuk** *Access Control* **pada Basis Data**



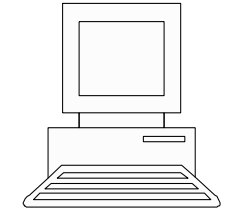
Pengguna Akhir

- Menggunakan hak akses orang lain.
- Melihat & menutup data yang tidak diotorisasi
- Staf tidak di-training
- Pemasukan data yang dilakukan oleh yang tidak berhak.
- Virus
- pemerasan



Programmer / Operator

- Membuat Password.
- Membuat program yang tidak aman
- Staf yang tidak di-training.
- Kebijakan keamanan & prosedur
- Pemogokan staf



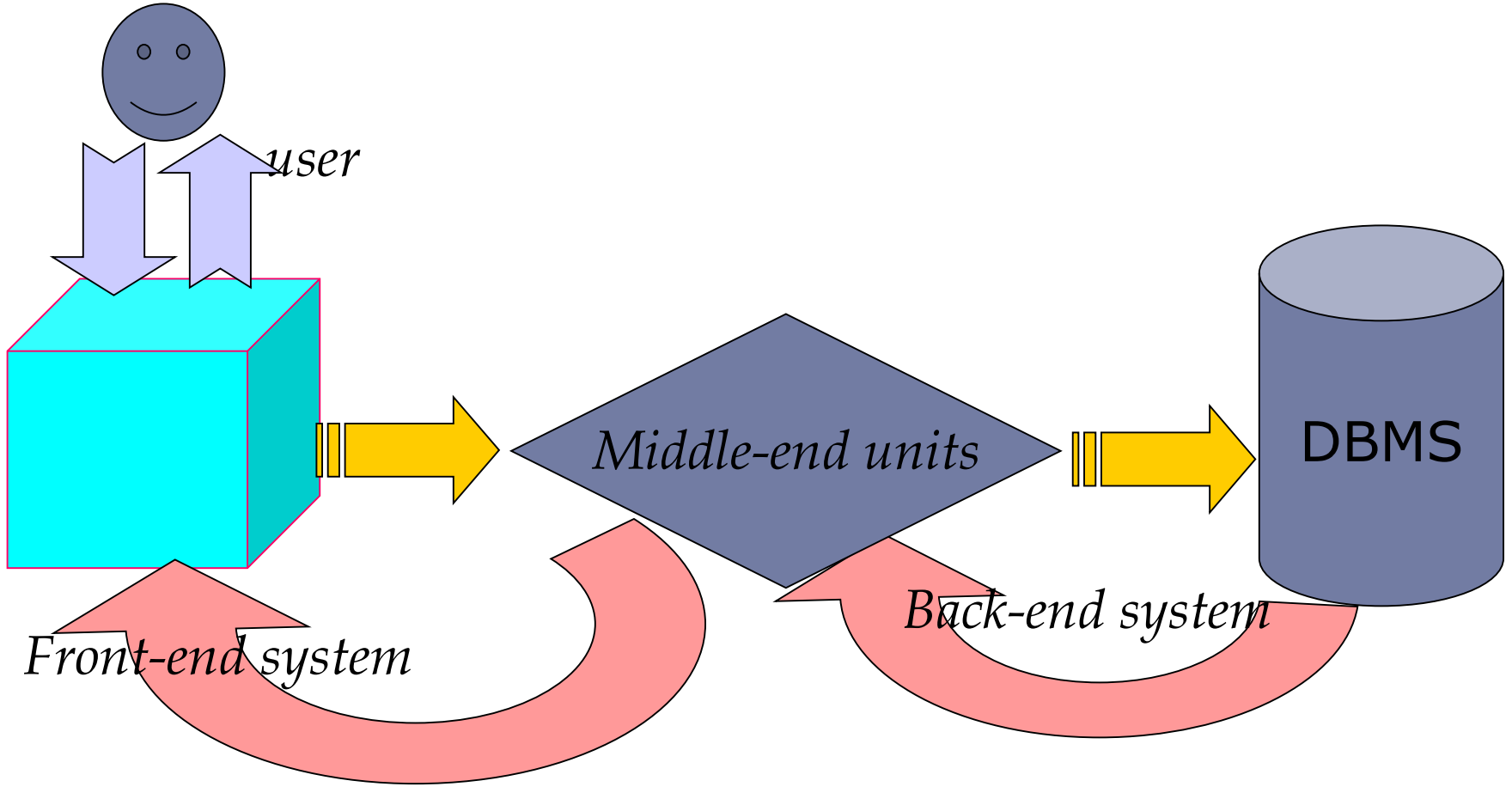
Database Administrator

- Kebijakan keamanan & prosedur

Keamanan pada Basis Data

- Keamanan merupakan suatu proteksi terhadap pengrusakan data dan pemakaian data oleh *user* yang tidak memiliki otoritas.
- Untuk menjaga keamanan Basis Data dibutuhkan:
  - Penentuan perangkat lunak Basis Data *Server* yang handal.
  - Pemberian otoritas kepada *user* mana saja yang berhak mengakses, serta memanipulasi data-data yang ada.

Skema Lapisan pada Basis Data yang dinterkoneksikan dengan aplikasi sistem utama



Alasan dibutuhkan otoritas pada keamanan basis data

- Pemberian wewenang atau hak istimewa (*privilege*) untuk mengakses sistem basis data.
- Kendali otorisasi (kontrol akses) dapat dibangun pada perangkat lunak dengan 2 fungsi :
  - Mengendalikan sistem atau obyek yang dapat diakses
  - Mengendalikan bagaimana *user* menggunakannya
- Sistem administrasi yang bertanggungjawab untuk memberikan hak akses dengan membuat *user account*.

Tabel View pada keamanan basis data

- Merupakan metode pembatasan bagi *user* untuk mendapatkan model basis data yang sesuai dengan kebutuhan pengguna.
- Metode ini dapat menyembunyikan data yang tidak digunakan atau tidak perlu dilihat oleh *user*.

Untuk Beberapa tingkat pengamanan pada Basis Data Relasional

- *Relasi* $\Rightarrow$ *user* diperbolehkan atau tidak diperbolehkan mengakses langsung suatu relasi.
- *View* $\Rightarrow$ *user* diperbolehkan atau tidak diperbolehkan mengakses data yang terdapat pada view.
- *Read Authorization* $\Rightarrow$ *user* diperbolehkan membaca data, tetapi tidak dapat memodifikasi.

Untuk Beberapa tingkat pengamanan pada Basis Data Relasional

- *Insert Authorization* $\Rightarrow$ *user* diperbolehkan menambah data baru, tetapi tidak dapat memodifikasi data yang sudah ada.
- *Update Authorization* $\Rightarrow$ *user* diperbolehkan memodifikasi data, tetapi tidak dapat menghapus data.
- *Delete Authorization* $\Rightarrow$ *user* diperbolehkan menghapus data.

Otorisasi tambahan untuk Modifikasi Data (*Update Authorization*)

- *Index Authorization* $\Rightarrow$ *user* diperbolehkan membuat dan menghapus index data.
- *Resource Authorization* $\Rightarrow$ *user* diperbolehkan membuat relasi-relasi baru.
- *Alteration Authorization* $\Rightarrow$ *user* diperbolehkan menambah/menghapus atribut suatu relasi.
- *Drop Authorization* $\Rightarrow$ *user* diperbolehkan menghapus relasi yang sudah ada.

Contoh perintah menggunakan SQL

GRANT : memberikan wewenang kepada pemakai

Sintaks :

GRANT <priviledge list> **ON** <nama relasi basis data/view> **TO** <pemakai>

Contoh :

GRANT INSERT **ON** Mahasiswa **TO** Ali, Ani

GRANT SELECT,UPDATE (Alamat, NoTelp) **ON** Mahasiswa **TO** Ali, Ani

Contoh perintah menggunakan SQL

REVOKE : mencabut wewenang yang dimiliki oleh pemakai

Sintaks :
REVOKE <priviledge list> **ON** <nama relasi basis data /view> **FROM** <pemakai>

Contoh :

REVOKE INSERT **ON** Mahasiswa **FROM** Ali

REVOKE SELECT,UPDATE (Alamat, NoTelp) **ON** Mahasiswa **FROM** Ali, Ani

Privilege list : READ, INSERT, DROP, DELETE, INDEX,

ALTER, RESOURCE

Back-up **data dan** *Recovery*

- ***Back-up*** : proses secara periodik untuk membuat duplikat dari basis data dan melakukan *logging file* (atau program) ke media penyimpanan eksternal.
- ***Recovery*** **:** merupakan upaya untuk mengembalikan basis data ke keadaaan yang dianggap benar setelah terjadinya suatu kegagalan.

Cara mudah menerapkan keamanan

- Memberi semua pengguna akses maksimum, tetapi kelemahannya adalah adanya kemungkinan kehilangan atau penyalahgunaan data.
- Pengguna basis data harus mempunyai akses yang cukup untuk melaksanakan pekerjaannya. Dengan kata lain, akses yang diberikan kepada pengguna harus sekecil mungkin untuk menghindari masalah.

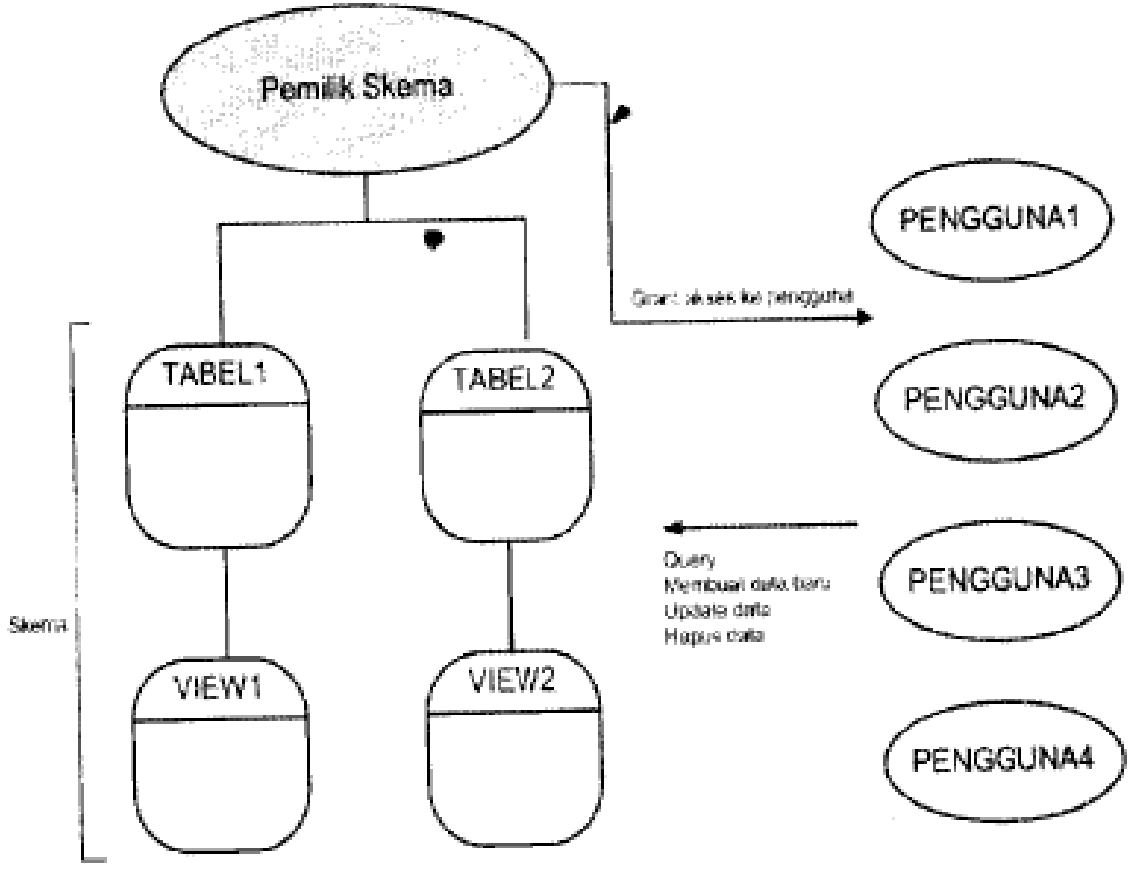
Tingkatan akses ke suatu sistem informasi

- Pengguna Super (root pada Unix, Admin pada Novell)
- Pemilik Basisdata
- Pemilik Skema
- Pengguna Akhir

- Pengguna super adalah *account* pada sistem operasi yang mempunyai *Privilege* paling banyak. Pengguna super memiliki akses ke semua *file* yang disimpan pada sistem.
- Pemilik basis data memiliki akses ke semua *file* yang berhubungan dengan *software* basis data dan *file* data pada suatu sistem. Meskipun pemilik basis data dibatasi pada *file-file* yang berhubungan dengan basis data, tetapi perlu di ingat bahwa sebagian *file* pada sebagian sistem berhubungan dengan basis data.

- Pemilik skema adalah pembuat dan pemilik objek-objek basis data yang digunakan untuk aplikasi pengguna. Pemilik skema mempunyai akses tidak terbatas ke seluruh objek skema dan bertanggungjawab mengontrol akses ke *account* pengguna lainnya.
- Pengguna akhir mempunyai akses paling sedikit meskipun basis data dibuat untuk pengguna akhir.

Hubungan pengguna dengan basis data



Privilege

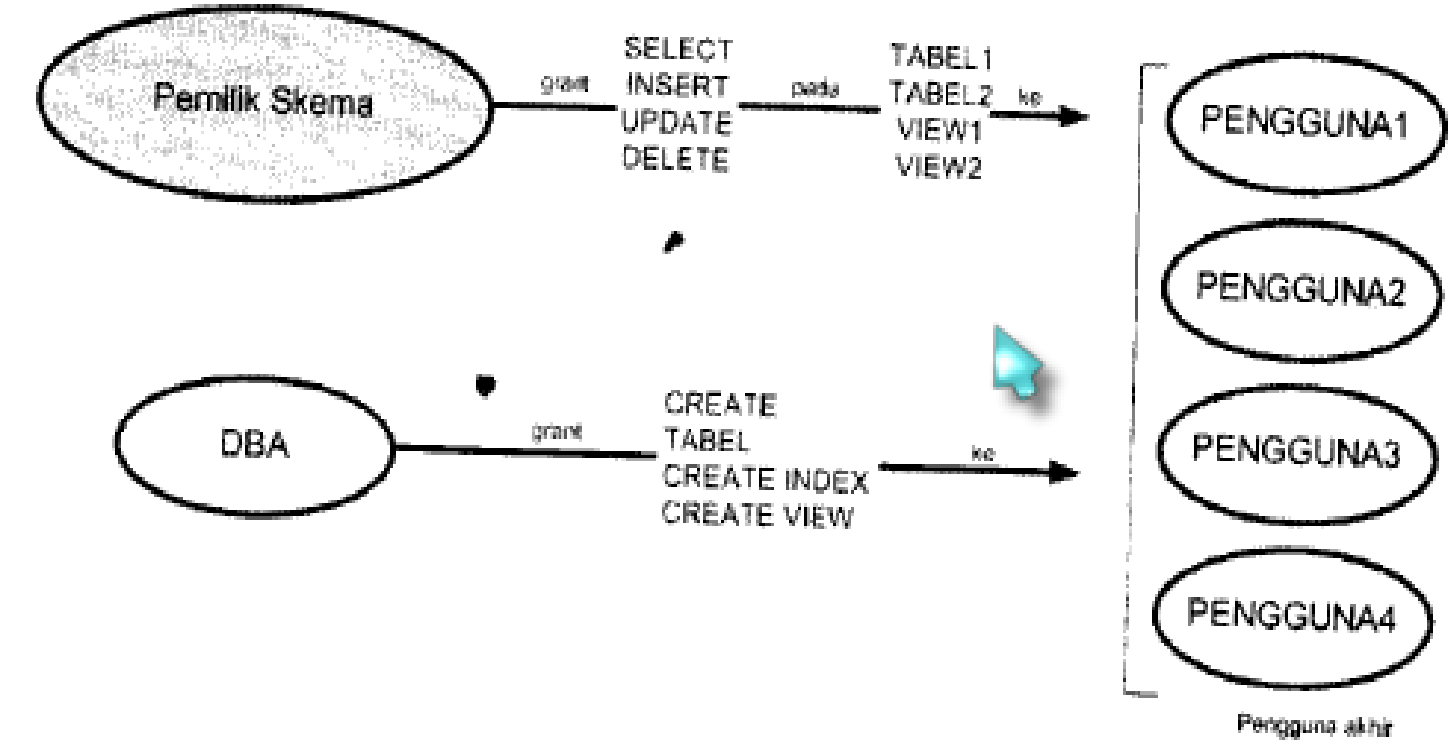
- *Privilege* digunakan untuk mengontrol akses pengguna. *Privilege* terdapat pada tingkatan sistem operasi, basis data dan aplikasi.
- *Privilege* basis data mengontrol akses pengguna dalam lingkungan basisdata seperti manipulasi struktur basis data dan akses ke objek skema.

PRIVILAGE SISTEM

- Sistem terdiri atas hal-hal yang memungkinkan pengguna melakukan tugasnya pada ruang lingkup basis data.
- *Privilege* sistem meliputi kemampuan membuat tabel, menghapus tabel, mengubah struktur tabel, membuat indeks dan view dan memanipulasi *account* pengguna.
- *Privilege* sistem berbeda-beda antara satu perangkat lunak basisdata relasional dengan yang lainnya.

- *Privilege* objek memungkinkan pengguna melakukan tugasnya pada ruang lingkup skema yang meliputi kemampuan mengambil data dari tabel dan memanipulasi data tabel.
- *Privilege* objek :
  - SELECT - memungkinkan data diambil dari table.
  - INSERT - memungkinkan pembentukan baris data baru pada tabel.
  - UPDATE - memungkinkan data yang sudah ada dalam tabel untuk dimodifikasi.
  - REFERENCES - memungkinkan kolom dalam tabel untuk diacu kolom lain (seperti melalui kunci tamu).

Proses pengaturan akses pengguna akhir melalui privilege basis data



TERIMA KASIH