

# Administrasi Basis Data

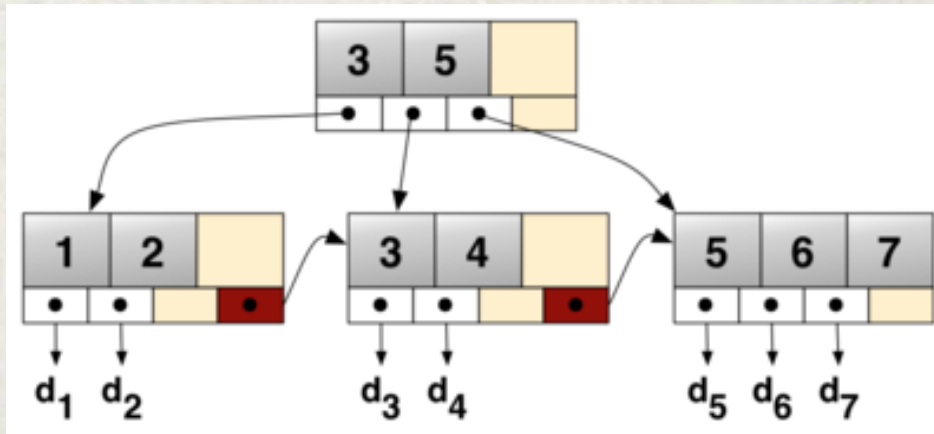
## Index

Yoannita



# Mengenal Index

- Index adalah objek database yang dipakai untuk meningkatkan kecepatan
- Sebuah index terdiri atas sekumpulan page yang dikenal sebagai pohon B+.
- Struktur pohon B+ dapat mempercepat pencarian data.



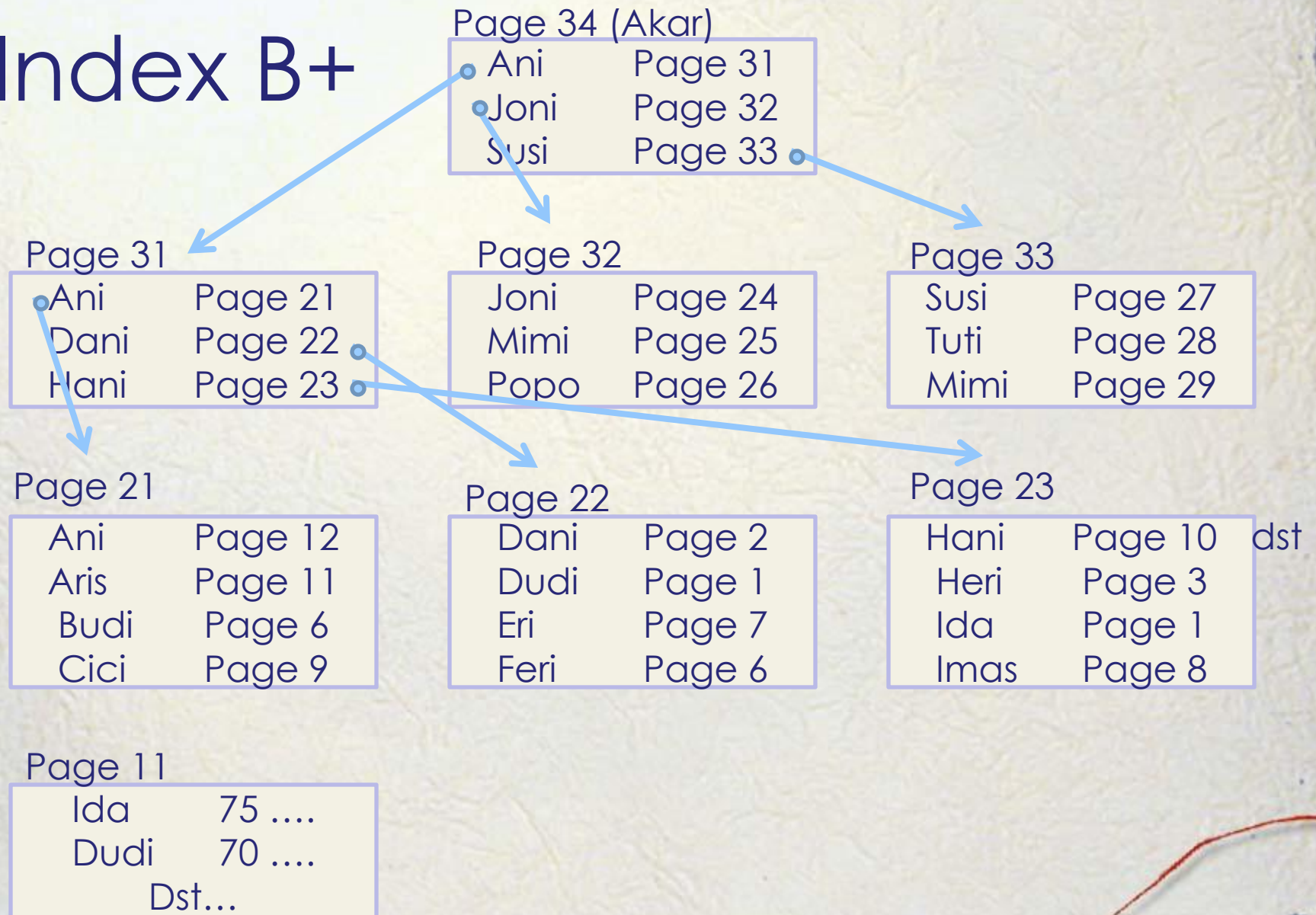
# Mengenal Index

- Misalnya anda akan mencari sebuah data. Anda harus menelusuri pohon B+ untuk mendapatkan sebuah baris lalu berpindah ke data sebenarnya.
- Anda akan mulai mencari dari page akar. Pointer ke page akar akan disimpan di tabel sistem yang bernama **SysIndexes** (untuk index non-clustered ada pada kolom root).
- Page akar berisi entri-entri index yang ada di level di bawahnya. Setiap index dapat memiliki satu atau beberapa level menengah.
- Jadi setiap entri index berisi nilai index dan pointer ke page di bawahnya

# Mengenal Index

- Setiap level index berbentuk list berkait ganda.
- Setiap elemen mempunyai **dua pointer**, satu pointer menunjuk ke page level **sebelumnya** dan satu pointer lagi menunjuk page level **sesudahnya**.

# Contoh Index B+



# Cara Kerja Index (i)

- Misalkan dari gambar sebelumnya kita akan mencari data Dudi.
- Kita mulai dari page **akar**
- Dudi lebih besar dari Ani dan lebih kecil dari Joni, maka kita pakai data pertama pada page akar yaitu Ani pada page 31
- Kita akses page 31, Dudi di antara Dani dan Hani, maka kita ambil Dani page 22
- Jika masih ada level-level menengah, kita akses level-level tersebut sampai kita mencapai level **daun**.

# Cara Kerja Index (i)

- Page 22 adalah page daun, maka kita akses satu per satu. Dudi ada di page 1 (page data). Kita akses page 1 dan ditemukan data dari Dudi.

# Mengenal Index

- Jika anda mengubah data dalam tabel, setiap index pada tabel juga harus diubah. SQL server menjamin konsistensi data antara data dalam tabel dan indexnya.
- **Dengan adanya index, operasi INSERT, UPDATE, dan DELETE menjadi lebih lambat dibandingkan tanpa index.**
- Oleh sebab itu, buatlah index dengan cara yang selektif dan yang memang sangat diperlukan.



# Jenis Index

SQL server mempunyai dua jenis index, yaitu :

- Index clustered
- Index non-clustered

# Jenis Index : Index clustered

- **Mengurutkan data secara fisik dan level daun berisi data sebenarnya.**
- Mengakses data lebih cepat daripada index non-clustered, karena tidak diperlukan proses lookup pada page data.
- Oleh karena data terurut secara fisik, anda hanya dapat mempunyai sebuah index clustered pada sebuah tabel. Oleh karena itu, anda harus memilih kunci index dengan hati-hati.

# Jenis Index : Index clustered

- **Kolom yang anda pilih sebagai kunci index juga dapat menimbulkan masalah besar**
- Misalnya anda meng-index berdasarkan field Nama yang bertipe Char(30). Field tersebut berukuran besar sehingga pencarian pada index clustered akan lama dan juga semua index non-clustered juga akan berisi nilai char(30) pada setiap barisnya.
- Oleh sebab itu, **anda harus memilih kunci yang tidak sering berubah**, karena perubahan pada index clustered akan mengubah semua index non-clustered.

# Jenis Index : Index Non-Clustered

- Index non-clustered sama seperti struktur pohon B+
- Setiap index mempunyai sebuah page akar, satu atau beberapa page level menengah dan sebuah level daun yang berisi sebuah baris untuk setiap baris pada tabel.
- Urutan pembuatan index non-clustered tidaklah berpengaruh.

# Jenis Index : Index Non-Clustered

- Index non-clustered **tidak** mengubah urutan data secara fisik.
- Baris-baris pada level daun diurutkan sesuai dengan kolom yang dipilih sebagai index.
- Setiap baris berisi sebuah pointer ke nomor page dan nomor baris dari data dalam tabel, jika tabel tidak mempunyai index clustered. Jika tabel mempunyai index clustered, setiap baris berisi nilai kunci index clustered.

- Singkatnya, index clustered adalah cara untuk proses pengurutan awal pada data.
- Tabel yang tidak mempunyai index clustered disebut **heap** dan
- index non-clustered adalah struktur index yang terpisah yang tidak mengurutkan data secara langsung.

# Keunikan Index

- Keunikan menentukan apakah nilai duplikasi diperbolehkan.
- Defaultnya index SQL Server bersifat tidak unik (boleh ada duplikasi)
- Index unik lebih baik, karena jika anda mencari sebuah data dan ditemukan, pencarian selesai. Jika index tidak unik, setelah menemukan sebuah data anda harus mencari data lain dengan nilai sama.

# Keunikan Index

- Keunikan index tergantung pada kunci yang dipilih.
- Misalnya jika kita mengindex berdasarkan NPM (Nomor Pokok Mahasiswa), dijamin unik, karena tidak ada dua orang mahasiswa mempunyai NPM sama.
- Apakah index berdasarkan alamat dapat unik? Tidak dapat, mungkin ada dua orang mahasiswa mempunyai alamat sama karena kakak beradik atau tinggal bersama dalam sebuah kos



# Menampilkan informasi Index

- Stored procedure untuk menampilkan informasi index adalah **sp\_help** dan **sp\_helpindex**.

# Membuat Index

- Contoh :
- Membuat index bernama Mahasiswa1 pada tabel Mahasiswa dengan field kunci adalah NPM
  - **CREATE INDEX Mahasiswa1 ON Mahasiswa(NPM)**
- Index yang dibuat adalah index tidak unik dan nonclustered

# Membuat Index

- Contoh (ii):
- Membuat index yang bernama Dosen1 pada tabel Dosen dengan field kunci adalah KodeMK dan KodeDosen.
  - **CREATE UNIQUE INDEX Dosen1 ON Dosen (kodeMK, KodeDosen)**
- Index yang dibuat adalah index unik dan nonclustered

# Option DROP\_EXISTING

- Option Drop\_Existing menyatakan index yang sudah ada dibuang dan dibuat ulang. Sebenarnya proses yang terjadi adalah reorganisasi dari index. Jika anda menghapus dan membuat ulang index clustered, semua index nonclustered akan terpengaruh.
- Contoh : Index Dosen1 sudah ada, kita akan menggantinya berdasarkan NamaDosen. Untuk itu kita harus memakai option DROP\_EXISTING
  - CREATE INDEX Dosen1  
ON Dosen (NamaDosen)  
WITH DROP\_EXISTING

# OPTION IGNORE\_DUP\_KEY

- Option IGNORE\_DUP\_KEY menyatakan bahwa pada saat menjalankan perubahan data pada beberapa baris pada sebuah tabel yang mempunyai index clustered unik, data yang duplikasi akan dibuang dan dilanjutkan dengan perintah yang lain (tidak semua perintah dibatalkan)

```
CREATE UNIQUE INDEX MK1  
ON MK (NamaMK)  
WITH IGNORE_DUP_KEY
```

# Menghapus Index

- Index yang tidak dipakai lagi sebaiknya dihapus

**DROP INDEX Dosen.Dosen1**

- Nama tabel atau nama view harus disertakan pada perintah DROP INDEX. Anda dapat menghapus beberapa index dalam sebuah perintah