

# MODUL SCD DATAWAREHOUSE

Slowly Changing Dimensions (SCD) adalah konsep dalam data warehouse yang digunakan untuk mengelola perubahan data pada tabel dimensi dari waktu ke waktu. Dimensi seperti pelanggan, produk, atau wilayah dapat mengalami perubahan (misalnya, pelanggan pindah alamat), dan SCD membantu melacak atau menangani perubahan tersebut agar data historis tetap akurat dan relevan untuk analisis.

## SCD Tipe 1 (Overwrite / Timpa Data Lama)

SCD Type 1 tidak menyimpan sejarah perubahan. Nilai lama akan **ditimpa** dengan yang baru. Cocok untuk data yang tidak memerlukan jejak historis, seperti email atau alamat pengguna yang tidak perlu dilacak perubahannya.

## IMPLEMENTASI KODE

#Pembuatan database

```
CREATE DATABASE data_warehouse;
```

# Pembuatan Tabel utama untuk menyimpan data karyawan

```
CREATE TABLE karyawan (
```

```
    id_karyawan INT PRIMARY KEY,
```

```
    nama VARCHAR(100),
```

```
    jabatan VARCHAR(100),
```

```
    tanggal_lahir DATE,
```

```
    email VARCHAR(100),
```

```
    telepon VARCHAR(100)
```

```
);
```

# Pembuatan tabel staging (tempat sementara) untuk menampung data baru yang masuk

```
CREATE TABLE staging_karyawan (
    id_karyawan INT,
    nama VARCHAR(100),
    jabatan VARCHAR(100),
    tanggal_lahir DATE,
    email VARCHAR(100),
    telepon VARCHAR(100)
);
```

# Data Awal pada tabel utama

```
INSERT INTO karyawan VALUES
```

```
(1001, 'Andi Saputra', 'Supervisor', '1990-05-01', 'andi_lama@example.com',
'081234567000'),
```







```
(1002, 'Budi Santoso', 'Staf IT', '1992-03-15', 'budi@example.com', '081234567891');
```

# Data Perubahan pada staging

```
INSERT INTO staging_karyawan VALUES
```

```
(1001, 'Andi Saputra', 'Manajer', '1990-05-01', 'andi@example.com', '081234567890');
```

Hasil Tabel:

		id_karyawan	nama	jabatan	tanggal_lahir	email	telepon
<input type="checkbox"/>	 Edit  Copy  Delete	1001	Andi Saputra	Supervisor	1990-05-01	andi_lama@example.com	081234567000
<input type="checkbox"/>	 Edit  Copy  Delete	1002	Budi Santoso	Staf IT	1992-03-15	budi@example.com	081234567891

# Insert atau update data secara otomatis (overwrite)

Penjelasan : Kode ini menyalin data dari staging\_karyawan ke karyawan. Jika id\_karyawan belum ada, data ditambahkan; jika sudah ada, data lama langsung

diperbarui tanpa menyimpan riwayat. Teknik ini mencerminkan **SCD Type 1** dalam proses ETL untuk menjaga data tetap terbaru.

```
INSERT INTO karyawan (id_karyawan, nama, jabatan, tanggal_lahir, email, telepon)
SELECT id_karyawan, nama, jabatan, tanggal_lahir, email, telepon
FROM staging_karyawan
ON DUPLICATE KEY UPDATE
    nama = VALUES(nama),
    jabatan = VALUES(jabatan),
    tanggal_lahir = VALUES(tanggal_lahir),
    email = VALUES(email),
    telepon = VALUES(telepon);
```

Hasil :

	id_karyawan	nama	jabatan	tanggal_lahir	email	telepon
<input type="checkbox"/> Edit Copy Delete	1001	Andi Saputra	Manajer	1990-05-01	andi@example.com	081234567890
<input type="checkbox"/> Edit Copy Delete	1002	Budi Santoso	Staf IT	1992-03-15	budi@example.com	081234567891

## SCD Tipe 2 Menyimpan Riwayat Perubahan (Full History)

SCD Tipe 2 menyimpan setiap versi data. Biasanya menggunakan tanggal\_mulai, tanggal\_akhir, dan aktif sebagai indikator riwayat.

### IMPLEMENTASI KODE

# SCD 2 : Menyimpan histori data perubahan

# Tabel utama barang (dimensi historis)

```
CREATE TABLE barang (
```

```
    id_barang INT,
```

```
nama VARCHAR(100),
harga FLOAT,
id_kategori INT,
lokasi VARCHAR(100),
tanggal_mulai DATE,
tanggal_akhir DATE,
PRIMARY KEY (id_barang, tanggal_mulai)
);
```

# Pembuatan tabel staging (tempat sementara) untuk menampung data baru yang masuk

```
CREATE TABLE staging_barang (
```

```
id_barang INT,
nama VARCHAR(100),
harga FLOAT,
id_kategori INT,
lokasi VARCHAR(100),
tanggal_mulai DATE,
tanggal_akhir DATE
```

```
);
```

# Data Awal

```
INSERT INTO barang VALUES
```

```
(2001, 'Kemeja Polos', 120000, 5, 'Rak Tengah', '2024-01-01', NULL);
```

# Data perubahan di staging (harga dan lokasi berubah)

```
INSERT INTO staging_barang VALUES
```

(2001, 'Kemeja Polos', 150000, 5, 'Rak Depan', '2024-06-01', NULL);

Hasil Tabel:

	id_barang	nama	harga	id_kategori	lokasi	tanggal_mulai	tanggal_akhir
<input type="checkbox"/> Edit  Copy  Delete	2001	Kemeja Polos	120000		5 Rak Tengah	2024-01-01	NULL

# Tandai record lama sebagai tidak aktif

UPDATE barang AS b

JOIN staging\_barang AS s ON b.id\_barang = s.id\_barang

SET b.tanggal\_akhir = s.tanggal\_mulai

WHERE b.tanggal\_akhir IS NULL

AND b.tanggal\_mulai < s.tanggal\_mulai;

# Tambahkan versi terbaru jika belum ada

INSERT INTO barang (id\_barang, nama, harga, id\_kategori, lokasi, tanggal\_mulai, tanggal\_akhir)

SELECT s.id\_barang, s.nama, s.harga, s.id\_kategori, s.lokasi, s.tanggal\_mulai, s.tanggal\_akhir

FROM staging\_barang AS s

LEFT JOIN barang AS b

ON s.id\_barang = b.id\_barang

AND s.tanggal\_mulai = b.tanggal\_mulai

WHERE b.id\_barang IS NULL;

Penjelasan: Kode ini menerapkan **SCD Type 2** untuk mencatat riwayat perubahan data. Pertama, data lama ditandai tidak aktif jika ada versi baru dengan tanggal lebih baru.

Lalu, versi baru ditambahkan ke tabel jika belum ada. Teknik ini menjaga agar perubahan data tetap tercatat lengkap dalam proses ETL.

Hasil :

				id_barang	nama	harga	id_kategori	lokasi	tanggal_mulai	tanggal_akhir
<input type="checkbox"/>	 Edit	 Copy	 Delete	2001	Kemeja Polos	120000		5 Rak Tengah	2024-01-01	2024-06-01
<input type="checkbox"/>	 Edit	 Copy	 Delete	2001	Kemeja Polos	150000		5 Rak Depan	2024-06-01	NULL

# OLAP Query

SELECT

nama,

harga,

lokasi,

tanggal\_mulai,

tanggal\_akhir

FROM barang

WHERE id\_barang = 2001

ORDER BY tanggal\_mulai;

SCD Tipe 3 Menyimpan Versi Sekarang dan Sebelumnya

SCD Tipe 3 menyimpan hanya **satu perubahan sebelumnya**. Biasanya digunakan 2 kolom: **kolom sekarang dan kolom sebelumnya**, misalnya: kategori\_lama, kategori\_baru.

IMPELMENTASI KODE

# SCD 3 : Simpan nilai sekarang & sebelumnya

# Tabel utama diskon

CREATE TABLE diskon (

id\_diskon INT PRIMARY KEY,

```
nama VARCHAR(100),  
musiman BOOLEAN,  
diskon_sekarang INT,  
diskon_sebelumnya INT  
);
```

# Tabel staging diskon

```
CREATE TABLE staging_diskon (  
id_diskon INT,  
nama VARCHAR(100),  
musiman BOOLEAN,  
diskon_baru INT  
);
```




# Data Awal

```
INSERT INTO diskon VALUES  
(3001, 'Promo Member', FALSE, 15, NULL),  
(3002, 'Diskon Lebaran', TRUE, 30, NULL);
```

# Data perubahan masuk ke staging

```
INSERT INTO staging_diskon VALUES  
(3001, 'Promo Member', FALSE, 25);
```

Hasil Tabel:

				id_diskon	nama	musiman	diskon_sekarang	diskon_sebelumnya
<input type="checkbox"/>	 Edit	 Copy	 Delete	3001	Promo Member	0	15	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	3002	Diskon Lebaran	1	30	NULL

# Update diskon jika sudah ada dan diskon berubah

UPDATE diskon AS d

JOIN staging\_diskon AS s ON d.id\_diskon = s.id\_diskon

SET

d.diskon\_sebelumnya = d.diskon\_sekarang,

d.diskon\_sekarang = s.diskon\_baru

WHERE d.diskon\_sekarang <> s.diskon\_baru;

# Insert data baru jika belum ada di tabel diskon

INSERT INTO diskon (id\_diskon, nama, musiman, diskon\_sekarang, diskon\_sebelumnya)

SELECT s.id\_diskon, s.nama, s.musiman, s.diskon\_baru, NULL







FROM staging\_diskon AS s

LEFT JOIN diskon AS d ON d.id\_diskon = s.id\_diskon

WHERE d.id\_diskon IS NULL;

Penjelasan : Kode ini menerapkan **SCD Type 3**, yaitu menyimpan nilai saat ini dan sebelumnya. Jika id\_diskon sudah ada dan nilai diskon berubah, maka diskon lama disimpan di diskon\_sebelumnya, dan diskon baru masuk ke diskon\_sekarang. Jika id\_diskon belum ada, data baru langsung ditambahkan.

Hasil:

				id_diskon	nama	musiman	diskon_sekarang	diskon_sebelumnya
<input type="checkbox"/>	 Edit	 Copy	 Delete	3001	Promo Member	0	25	15
<input type="checkbox"/>	 Edit	 Copy	 Delete	3002	Diskon Lebaran	1	30	NULL