

ORACLE PL/SQL · PERTEMUAN 04 / 14

SELECT & Filtering Data

Memilih kolom dan baris secara tepat, termasuk data NULL serta batas hasil.

Instalasi & SQL → Mini Project → PL/SQL → Final Project · Oracle XE lokal atau FreeSQL dengan login

Versi PDF materi pertemuan 04. Contoh dan tabel dapat dibaca di PDF. Kuis dapat dijalankan di halaman web: [Buka materi daring](#).

Tujuan pembelajaran

1. Membaca SELECT dengan alias dan DISTINCT.
2. Menerapkan perbandingan, BETWEEN, IN, LIKE, dan IS NULL.
3. Menggabungkan AND, OR, NOT dengan tanda kurung.
4. Mengurutkan hasil serta membatasi secara deterministik memakai FETCH FIRST atau ROWNUM.

Pokok bahasan / praktikum

- SELECT dan alias
- DISTINCT
- WHERE dan operator perbandingan
- BETWEEN
- IN
- LIKE
- IS NULL
- AND OR NOT
- ORDER BY ASC DESC NULLS LAST
- FETCH FIRST
- ROWNUM

Prasyarat: Modul02/03 atau setup indeks sekali dengan seed acuan. Gunakan XE lokal ATAU FreeSQL login pada schema yang sama; tidak ada perubahan data.

Pilih salah satu lingkungan praktikum

Oracle XE: XE lokal dengan SQL Developer: cocok untuk instalasi, koneksi, user/schema, privilege, tablespace serta latihan SQL/PLSQL.

FreeSQL: FreeSQL dengan login: masuk menggunakan akun sendiri dan gunakan schema latihan yang diizinkan untuk SQL/PLSQL. Tidak perlu memasang XE untuk menyelesaikan latihan non-administratif.

Pengumpulan tugas menerima hasil dari salah satu lingkungan. Pastikan objek prasyarat tersedia dan tetap gunakan schema latihan, bukan schema produksi.

AMATI → PREDIKSI → JELASKAN

Mulai dari pertanyaan

1. Apakah urutan penyimpanan selalu sama dengan urutan hasil?
2. Bagaimana mencari pelanggan yang kotanya belum diketahui?
3. Mengapa dua kondisi OR perlu dikelompokkan ketika digabung dengan AND?

Uji pemahaman awal

1. Bagaimana memilih kota yang belum diketahui?

- ☐ kota = NULL
- ☐ kota = kosong
- ☐ kota IS NULL
- ☐ kota <> NULL

2. BETWEEN 12000 AND 15000 berarti apa?

- ☐ Batas tidak ikut
- ☐ Hanya 12000
- ☐ Kedua batas ikut
- ☐ Semua nilai selain batas

3. Bagaimana memilih dua harga tertinggi dengan ROWNUM?

- Batasi sebelum mengurutkan pada query sama
- Urutkan dalam subquery lalu batasi di luar
- Gunakan LIMIT 2
- Tidak perlu ORDER BY

PAHAMI KONSEP · BACA TERSTRUKTUR · TERAPKAN

Teori dan pembahasan

Susun query agar pertanyaan bisnis, kondisi filter dan urutan hasil dapat dibaca dengan jelas. Bedakan nilai kosong, pola teks dan pembatasan jumlah baris.

Konsep inti bab ini

- WHERE hanya mempertahankan kondisi TRUE ; NULL perlu IS NULL / IS NOT NULL .
- Gunakan tanda kurung untuk kelompok AND/OR yang bermakna bisnis.
- Top-N membutuhkan ORDER BY yang jelas sebelum pembatasan hasil.

Urutan pembahasan

01 Proyeksi kolom, alias, dan DISTINCT

02 Perbandingan dan pemilihan rentang

03 Daftar nilai dan pola teks

04 NULL dan logika tiga nilai

05 AND, OR, NOT dan tanda kurung

06 Urutan dan pembatasan yang benar

Gambaran konsep dalam satu tabel

Kebutuhan	Pilihan SQL	Catatan
Nilai tertentu	IN	Daftar nilai non- <code>NULL</code> untuk pengantar
Rentang	BETWEEN	Kedua batas ikut dihitung
Pola teks	LIKE	% banyak karakter; _ satu karakter
Nilai belum tersedia	<code>IS NULL</code>	Jangan memakai <code>= NULL</code>
Urutan dan Top-N	<code>ORDER BY + FETCH FIRST</code>	Gunakan pemutus seri yang konsisten

01 Proyeksi kolom, alias, dan DISTINCT

`SELECT` menentukan kolom atau ekspresi yang ingin dibaca, `FROM` menentukan sumber, dan `WHERE` memilih baris yang memenuhi kondisi.

`SELECT` tidak mengubah tabel. Untuk laporan, sebut nama kolom alih-alih selalu memakai bintang agar hasil stabil dan mudah dipahami. Ekspresi harga * stok menghasilkan nilai persediaan yang diprediksi dari data, bukan kolom permanen baru dalam tabel.

POIN PEMBAHASAN

- Alias memberi label hasil, misalnya nama AS nama_pelanggan.
- Alias dengan spasi dapat memakai kutip ganda; teks nilai tetap memakai kutip tunggal.
- Alias `SELECT` umumnya boleh dipakai pada `ORDER BY` tetapi bukan `WHERE` pada tingkat query yang sama.
- `DISTINCT` membuang duplikasi kombinasi semua kolom yang dipilih, bukan memilih satu baris bebas untuk setiap nama.
- `DISTINCT` kota menunjukkan kota berbeda, termasuk satu kelompok `NULL`.

02 Perbandingan dan pemilihan rentang

WHERE menggunakan =, <>, >, >=, <, dan <= untuk membandingkan nilai.

Operator <> berarti tidak sama. Query harga >= 15000 memilih Kopi dan Teh pada seed, sementara harga < 15000 memilih Gula dan Sabun. Gunakan angka sebagai angka, bukan teks berformat rupiah, agar tipe dan perbandingannya jelas. Prediksi hasil dengan memeriksa empat baris produk sebelum mengeksekusi.

POIN PEMBAHASAN

- BETWEEN batas_bawah AND batas_atas mencakup kedua batas.
- Harga BETWEEN 12000 AND 15000 mencakup Gula dan Teh.
- Untuk DATE yang dapat menyimpan jam, rentang tanggal sampai tengah malam dapat melewati baris pada hari terakhir.
- Pola tanggal >= DATE '2026-10-01' AND tanggal < DATE '2026-10-04' mencakup seluruh tiga hari tanpa bergantung waktu persis tengah malam.

03 Daftar nilai dan pola teks

IN menyederhanakan beberapa perbandingan kesamaan: kota IN ('Subang','Bandung') memilih Ayu dan Bima pada seed.

NOT IN membutuhkan perhatian bila daftar atau subquery mengandung **NULL** karena hasil perbandingan bisa menjadi **UNKNOWN**. Pada modul ini daftar IN memakai literal non-**NULL** agar maknanya langsung terbaca. Jangan mencampurkan angka dan teks dalam daftar lalu berharap konversi selalu berhasil.

POIN PEMBAHASAN

- LIKE mencocokkan pola, bukan persamaan biasa.
- Tanda persen berarti nol atau lebih karakter, sedangkan garis bawah berarti tepat satu karakter. nama LIKE 'A%' mencari awalan A; nama LIKE '_ima' dapat cocok dengan Bima.
- Jika ingin mencari persen sebagai karakter literal gunakan ESCAPE dan karakter pelolos yang dinyatakan.

- Pencocokan huruf mengikuti aturan collation lingkungan; fungsi normalisasi huruf akan dibahas modul05.

04 NULL dan logika tiga nilai

NULL menyatakan nilai belum diketahui atau tidak tersedia, bukan nol dan bukan tulisan **NULL**.

Gunakan **IS NULL** atau **IS NOT NULL**, bukan **= NULL** atau **<> NULL**. Pelanggan Citra memiliki kota dan email **NULL** sehingga query **IS NULL** akan menemukannya.

Perbandingan kota = 'Subang' terhadap **NULL** menghasilkan **UNKNOWN**; **WHERE** hanya meloloskan **TRUE**, bukan **FALSE** atau **UNKNOWN**.

POIN PEMBAHASAN

- Oracle memperlakukan string kosong sebagai **NULL** untuk data karakter biasa.
- Karena itu nama = " bukan cara mencari kosong yang andal.
- **NOT** pada kondisi tak diketahui tetap tak diketahui, sehingga **NOT** (kota = 'Subang') tidak otomatis mencakup pelanggan berkota **NULL**.
- Bila tujuan mencakup **NULL**, tuliskan kondisi kota **<> 'Subang' OR kota IS NULL** secara eksplisit agar makna laporan tidak bergantung asumsi.

05 AND, OR, NOT dan tanda kurung

AND mensyaratkan kedua kondisi benar, **OR** cukup salah satu benar, dan **NOT** menegasikan kondisi dengan tetap mengikuti logika **NULL**.

AND dievaluasi dengan prioritas lebih tinggi daripada **OR**. Query (kota = 'Subang' OR kota = 'Bandung') AND email **IS NOT NULL** mengharuskan semua pelanggan terpilih memiliki email. Tanpa kurung, syarat email hanya melekat pada cabang Bandung menurut prioritas operator.

POIN PEMBAHASAN

- Jangan mengandalkan urutan penulisan kondisi untuk menjamin sebuah fungsi berisiko tidak dipanggil: optimizer dapat mengubah rencana evaluasi.

- Pada latihan, kondisi memakai tipe yang tepat dan ekspresi aman.
- Tuliskan tanda kurung ketika maksud bisnis memiliki kelompok, walaupun aturan prioritas dapat menghasilkan hasil serupa.
- Sebelum query rumit, uji masing-masing kondisi lalu gabungkan sambil membandingkan ID hasilnya.

06 Urutan dan pembatasan yang benar

Tanpa **ORDER BY**, urutan hasil tidak dijamin.

Gunakan **ORDER BY** harga DESC, id_produk ASC agar harga tertinggi muncul lebih dahulu dan ID menjadi pemutus seri. ASC mengurutkan naik, DESC turun; NULLS FIRST atau NULLS LAST mengatur posisi **NULL** secara eksplisit. Mengurutkan tidak mengubah susunan penyimpanan tabel. Alias nilai_persediaan dapat dipakai untuk mengurutkan ekspresi hasil.

POIN PEMBAHASAN

- **FETCH FIRST 2 ROWS ONLY** setelah **ORDER BY** memberi dua produk teratas pada Oracle 19c.
- **ROWNUM** adalah alternatif lama, tetapi batas harus diterapkan di luar subquery yang sudah diurutkan.
- **WHERE ROWNUM <= 2** pada query yang baru mengurutkan setelahnya dapat memilih dua baris yang bukan teratas.
- Jangan memakai **LIMIT**.
- Jumlah terbatas tanpa urutan lengkap tidak menjamin baris yang sama pada setiap eksekusi, terutama saat ada nilai seri.

Alur merancang query

- 1 Tentukan sumber tabel
- 2 Pilih kolom/alias
- 3 Nyatakan kondisi `WHERE`
- 4 Tetapkan `ORDER BY`
- 5 Batasi hasil bila diperlukan

Gunakan alur ini sebagai panduan saat mengikuti praktikum; bukti hasil tetap berasal dari eksekusi Oracle Anda.

AMATI DATA · IKUTI LANGKAH · PAHAMI HASIL

Contoh dan tabel pembahasan

Contoh berikut dibaca langsung, tanpa parameter atau tombol simulasi. Nilai adalah **prediksi ilustratif dari data acuan**, bukan hasil eksekusi pada akun Oracle Anda.

Memilih produk dengan harga minimal 15000

Dari empat produk seed, pilih yang memenuhi batas harga lalu urutkan dari yang paling mahal.

Contoh SQL / PL/SQL

```
SELECT id_produk, nama_produk, harga
FROM produk_demo
WHERE harga >= 15000
ORDER BY harga DESC, id_produk;
```

id_produk	nama_produk	harga
11	Kopi	30000

id_produk	nama_produk	harga
12	Teh	15000

Cara membaca contoh

1. **WHERE** menahan Kopi dan Teh; Gula dan Sabun berada di bawah batas.
2. **ORDER BY** harga DESC menempatkan Kopi lebih dahulu.
3. id_produk menjadi urutan tambahan jika ada harga yang sama.

Yang perlu dipahami: Filter bekerja pada nilai data, bukan posisi baris di layar. Hasil dapat berubah jika seed Anda sudah dimodifikasi.

Video pembelajaran

Video utama — fokus bab ini

SELECT/projection dan WHERE/selection menggunakan Oracle SQL. Menggantikan video utama sebelumnya yang menggunakan T-SQL.

SQL WHERE clause | Oracle SQL fundamentals

DBA Genesis · [Buka video utama di YouTube](#)

Batas cakupan dan panduan menonton: Video ini fokus fondasi filter. Operator BETWEEN/IN/LIKE/IS NULL, ORDER BY dan FETCH FIRST/ROWNUM tetap dipelajari serta diuji melalui contoh Oracle pada halaman.

Video merupakan pendamping, bukan pengganti keseluruhan bab. Pemeriksaan relevansi memakai transkrip yang tersedia, deskripsi pembuat dan bab waktu; bukan klaim seluruh durasi telah ditonton. Versi/editor atau bahasa dapat berbeda. Praktikum tetap mengikuti SQL/PLSQL Oracle pada halaman dan pilihan XE atau FreeSQL login. Jika pemutar dibatasi oleh YouTube, gunakan tautan Buka video.

Praktikum Oracle XE atau FreeSQL

Pilih salah satu lingkungan praktikum

Oracle XE: XE lokal dengan SQL Developer: cocok untuk instalasi, koneksi, user/schema, privilege, tablespace serta latihan SQL/PLSQL.

FreeSQL: FreeSQL dengan login: masuk menggunakan akun sendiri dan gunakan schema latihan yang diizinkan untuk SQL/PLSQL. Tidak perlu memasang XE untuk menyelesaikan latihan non-administratif.

Pengumpulan tugas menerima hasil dari salah satu lingkungan. Pastikan objek prasyarat tersedia dan tetap gunakan schema latihan, bukan schema produksi.

Gunakan [FreeSQL dengan login](#) atau SQL Developer yang terhubung ke XE. [Baca panduan dua jalur praktikum](#). Jalankan kode sesuai urutan dan keterangan tiap langkah.

Output contoh adalah prediksi berdasarkan dataset, bukan bukti bahwa kode telah dijalankan pada akun Anda. Periksa koneksi, hak membuat objek, autocommit dan panel DBMS_OUTPUT. Slash pada baris sendiri adalah pemisah mode skrip klien tertentu, bukan statement PL/SQL.

Gunakan tabel pertemuan02 dan data pertemuan03 atau setup index sekali. Periksa keadaan data bila latihan sebelumnya pernah di-commit. Jangan menjalankan ulang CREATE TABLE/INSERT atau menghapus data secara buta.

Langkah 1 — Kolom, alias, DISTINCT dan perbandingan

```
SELECT id_pelanggan, nama AS nama_pelanggan, kota FROM
pelanggan_demo ORDER BY id_pelanggan;
SELECT DISTINCT kota FROM pelanggan_demo ORDER BY kota NULLS LAST;
SELECT id_produk, nama_produk, harga * stok AS nilai_persediaan
FROM produk_demo WHERE harga >= 15000 AND harga <> 10000
ORDER BY nilai_persediaan DESC, id_produk;
SELECT id_produk, nama_produk FROM produk_demo WHERE harga <
15000;
SELECT id_produk, nama_produk FROM produk_demo WHERE harga > 12000
AND harga <= 30000;
```

Hasil DISTINCT bukan seluruh baris pelanggan. Ekspresi persediaan hanya dihitung ketika SELECT; tabel tidak diubah.

Langkah 2 — Rentang, daftar, pola, NULL dan logika

```
SELECT nama_produk, harga FROM produk_demo WHERE harga BETWEEN
12000 AND 15000 ORDER BY harga;
SELECT nama FROM pelanggan_demo WHERE kota IN ('Subang', 'Bandung')
ORDER BY id_pelanggan;
SELECT nama FROM pelanggan_demo WHERE nama LIKE 'A%' OR nama LIKE
'_ima' ORDER BY id_pelanggan;
SELECT nama FROM pelanggan_demo WHERE kota IS NULL;
SELECT nama FROM pelanggan_demo WHERE (kota = 'Subang' OR kota =
'Bandung') AND email IS NOT NULL;
SELECT nama FROM pelanggan_demo WHERE NOT (kota = 'Subang') OR
kota IS NULL ORDER BY id_pelanggan;
SELECT id_jual, tanggal FROM penjualan_demo
WHERE tanggal >= DATE '2026-10-01' AND tanggal < DATE '2026-10-04'
ORDER BY tanggal, id_jual;
SELECT 'diskon 10%' AS teks FROM dual WHERE 'diskon 10%' LIKE
'%!%' ESCAPE '!';
```

Rentang harga mencakup 12000 dan 15000. Query IS NULL memilih Citra. Pola terakhir mencari akhiran persen literal; ! adalah escape, bukan wildcard.

Langkah 3 — Dua alternatif pembatas hasil

```
SELECT id_produk, nama_produk, harga FROM produk_demo
ORDER BY harga DESC, id_produk ASC FETCH FIRST 2 ROWS ONLY;
SELECT id_produk, nama_produk, harga FROM (
  SELECT id_produk, nama_produk, harga FROM produk_demo ORDER BY
  harga DESC, id_produk ASC
) WHERE ROWNUM <= 2 ORDER BY harga DESC, id_produk ASC;
```

Kedua bentuk diprediksi memilih Kopi lalu Teh pada seed. Bentuk ROWNUM memisahkan pengurutan dari pembatasan; gunakan salah satu untuk tugas biasa.

Tugas / latihan

1. Pada XE lokal ATAU FreeSQL login, tulis lima query filter yang masing-masing memakai BETWEEN, IN, LIKE, IS NULL, dan kelompok AND/OR; lampirkan hasil nyata.

2. Bandingkan kota <> Subang dengan kota <> Subang OR kota IS NULL dan jelaskan pelanggan yang berbeda.
3. Tampilkan dua produk harga tertinggi memakai FETCH FIRST atau alternatif ROWNUM teratur; jelaskan pemutus seri ID.

Pertanyaan analisis

1. Mengapa DISTINCT nama,kota berbeda dari DISTINCT kota saja?
2. Jelaskan mengapa NOT pada perbandingan NULL tidak otomatis menghasilkan TRUE.

Pengumpulan

Tugas boleh dikerjakan di **Oracle XE lokal atau FreeSQL dengan login**; cukup pilih salah satu. Kumpulkan satu laporan melalui LMS/kanal yang ditetapkan dosen. Cantumkan Nama, NPM, pertemuan dan lingkungan yang dipakai. Sertakan kode SQL/PLSQL sebagai teks, output aktual atau pesan error, serta jawaban analisis.

Jika FreeSQL menyediakan tautan berbagi yang dapat diakses dosen, tautan boleh ditambahkan sebagai bukti pendamping, bukan pengganti hasil eksekusi yang terdokumentasi. Jangan membagikan password, token atau koneksi pribadi.

Hasil yang belum dieksekusi diberi label “prediksi”. Screenshot ilustrasi browser bukan bukti praktikum Oracle.

Rubrik penilaian

- Ketepatan konsep dan rancangan: **20%**
- Kebenaran query/program: **30%**
- Pengujian normal, batas dan gagal: **20%**
- Analisis serta penjelasan hasil: **20%**
- Identitas dan kerapian laporan: **10%**

Referensi resmi

- [Oracle Database 19c SQL Language Reference: SELECT](#)
- [Oracle Database 19c SQL Language Reference: Conditions](#)
- [Oracle Database 19c SQL Language Reference: Nulls](#)
- [Oracle Database 19c SQL Language Reference: ROWNUM-Pseudocolumn](#)

- [Oracle Database 19c SQL Language Reference: Pattern-matching-Conditions](#)

Panduan XE21c digunakan untuk konteks instalasi lokal; dokumentasi SQL/PLSQL untuk konsep bahasa. Versi FreeSQL/editor dapat berbeda, sehingga periksa fitur dan pesan aktual.
