MODUL PRAKTIKUM X

SELECT III

Tujuan:

- Mampu menampilkan data-data yang ada di tabel menggunakan perintah SQL SELECT
- Mampu memahami klausa SELECT dan kombinasinya

Materi:

- Group By
- Having
- Union
- Aggregate Function
- SubQuery

Persiapan

- Membaca buku literature, referensi atau dari sumber lain tentang penggunaan DML SELECT
- Modul Praktikum X

Landasan Teori

Aggregate Function

Ada lima fungsi aggregate yang bisa digunakan dalam daftar SELECT yang mana kelima ini sudah distandarkan dalam ISO yaitu COUNT, SUM, AVG, MIN, MAX

Aturan – aturan yang berlaku

- a. SUM dan AVG digunakan untuk field numeric
- b. COUNT, MIN, MAX untuk field numeric dan non numeric
- c. COUNT(*) berfungsi menghitung seluruh baris dalam table walaupun terdapat NULL atau duplikasi
- d. DISTINCT tidak berpengaruh terhadap operasi MIN atau MAX, tetapi berpengaruh pada SUM atau AVG
- e. Fungsi aggregate dapat digunakan dalam SELECT dan HAVING

Subquery

Subquery adalah suatu subselect yang digunakan dalam klausa WHERE dan HAVING disamping SELECT utama. Istilah lainnya adalah Nested Query. Subquery ini juga dapat digunakan dalam perintah INSERT, UPDATE, DELETE

Aturan yang berlaku

- 1. Klausa ORDER BY dapat tidak digunakan dalam subquery walaupun dapat digunakan dalam SELECT terluar
- 2. Subquery SELECT harus terdiri atas nama kolom tunggal atau expresi, kecuali untuk subquery yang menggunakan EXISTS
- 3. Berdasarkan default, nama kolom mengacu ke nama table pada klausa FROM dari subquery, selain itu dapat juga mengacu pada nama alias
- 4. Karena merupakan sebuah operasi perbandingan, maka harus dituliskan disebelah kanan
- 5. Tidak dapat digunakan sebagai operasi perbandingan dalam suatu expresi
- 6. EXISTS dan NOT EXISTS hanya untuk subquery dan menghasilkan hasil benar atau salah
- 7. EXIST dan NOT EXISTS benar jika dan hanya jika terdapat sedikitnya satu baris dalam table hasil yang dikembalikan oleh subquery, salah jika subquery mengembalikan nilai kosong

Group By

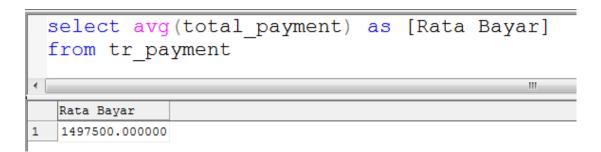
Seringkali informasi yang diinginkan dikelompok berdasarkan kolom-kolom terntu seperti untuk mengetahui frekuensi transaksi supplier dan customer per bulan. Atau untuk mengetahui jumlah siswa yang mengambil kursus, mengetahui berapa siswa yang absen berdasarkan jadwal yang diikuti, mengetahui pembayaran tiap bulan dan lain sebagainya. Untuk pengelompokan data bisa menggunakan klausa GROUP BY dan HAVING sebagai filtering untuk hasil pengelompokan data

Union

Union dipakai untuk menggabungkan dua buah query atau lebih. Syarat du buah query atau lebih bisa digabungkan adalah antara query yang satu dengan yang lainnya harus mempunyai jumlah kolom yang akan ditampilkan harus sama. Demikian juga dengan tipe data, masing – masing kolom yang dimapilkan pada query tersebut, satu sama lainnya harus sama juga

Langkah – Langkah Praktikum

- Gunakan Database yang telah dibuat pada minggu lalu, yaitu SBD_XXXX. XXXX merupakan NIM Masing – masing. Kalau belum ada silakan dibuat terlebih dahulu seperti langkah – langkah pada modul VIII
- 2. Menampilkan Data menggunakan fungsi Aggregate
 - a. Menampilkan rata rata pembayaran di table tr_payment



b. Menampilkan no regist_id yang paling besar, dan payment paling besar

```
select max(regist_id), max(total_payment)
from tr_payment

(No column name) (No column name)
1 reg010 2000000
```

c. Menampilkan banyaknya total pembayaran yang lebih dari 1500000

```
SELECT COUNT(*) AS banyaknya_pembayaran
FROM tr_payment
WHERE Total_payment > 1500000
```

d. Menampilkan jumlah ruangan yang terpakai pada hari senin dan selasa

```
SELECT COUNT (DISTINCT Room) AS Jumlah_ruangan
FROM tr_jadwal
WHERE day_ BETWEEN 'mon' AND 'tue'
```

e. Menampilkan banyaknya dan total pembayaran

f. Menampilkan minimal Pembayaran, maximal pembayaran, dan rata2 pembayaran

```
SELECT MIN(total_payment) AS min_payment,
MAX(total_payment) AS max_Payment,
AVG(total_payment) AS avg_payment
FROM tr_payment

min_payment max_Payment avg_payment
1 900000 2000000 1497500.0000000
```

3. Menggunakan GROUP BY untuk mengelompokan data

a. Menampilkan jumlah berapa kali masing masing program ID ada jadwalnya

```
SELECT program_id, COUNT (program_id) as myCount
FROM tr_jadwal
GROUP BY program_id

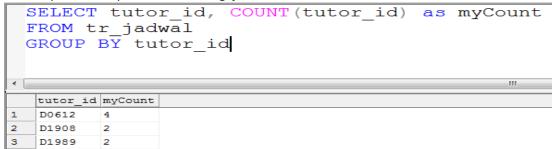
| program_id myCount |
| PC001 | 2 |
| PG001 | 4 |
| PP001 | 2 |
```

b. Menghitung berapa siswa laki dan perempuan

```
SELECT student_gender, COUNT(student_id) as myCount
FROM ms_student
GROUP BY student_gender

| student_gender myCount |
| Female | 6 |
| male | 4 |
```

c. Menampilkan berapa kali tutor mengajar



d. Menampilkan berapa kali tutor id mengajar di ruangan yang sama

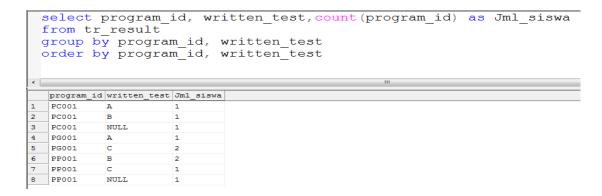
4. Penggunaan Having

a. Menampilkan program id, jumlah jadwal id yang jadwal id lebih dari satu

```
SELECT program_id, COUNT(jadwal_id) as Banyaknya_Jadwal
FROM tr_jadwal
GROUP BY program_id
HAVING COUNT (jadwal_id) > 1
```

b. Menampilkan jumlah siswa dari hasil written_test dan dikelompokkan berdasarkan program yang diikuti. Yang dimunculkan adalah jumlah siswa yang pesertanya lebih dari 1

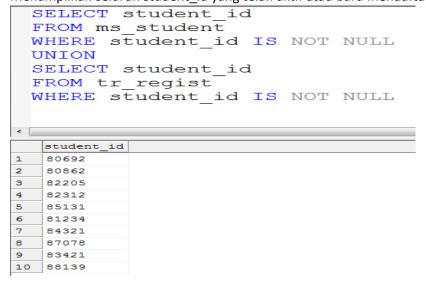
Untuk perbandingan, jalankan query dibawah. Query ini untuk menghitung jumlah siswa berdasarkan hasil wrriten_test dan program_id serta ditampilkan dalam bentuk terurut berdasarkan program_id dan written_id



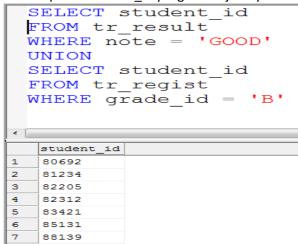
Kemudian, dari hasil query diatas, difilter lagi untuk menampilkan jumlah siswa yang lebih besar dari 1

5. Penggunaan Union

a. Menampilkan seluruh student_id yang telah aktif atau baru mendaftar



b. Menampilkan student_id yang hasil ujiannya 'GOOD' atau grade_id 'B'



c. Menampilkan data siswa dan tutor yang jenis kelaminnya Laki-laki

```
SELECT student id, student address, student gender
 FROM ms student
 WHERE student gender = 'Male'
 UNION
 SELECT tutor id, tutor address, tutor gender
 FROM ms tutor
 WHERE tutor gender = 'Male'
  student_id student_address student_gender
  81234 Slipi male
82205 jembatan tiga male
87078 Jeruk Purut male
                        male
2 82205
3
  D1507 -
           gang u male
serpong Male
  D1507 serpong
D1908 Haji rabu
                       Male
  D1989
           Harmoni
                        Male
```

- 6. Penggunaan Sub Query
 - a. Menampilkan jadwal id beserta id tutornya yang programnya PC001

```
SELECT jadwal_id, tutor_id
FROM tr_jadwal
WHERE program_id =

(SELECT program_id
FROM ms_program
WHERE program_id = 'PC001')
```

b. Menampilkan daftar pembayaran yang pembayarannya melebihi rata – rata pembayaran

c. Menampilkan student id, student name yang absensinya 'hadir' pada pertemuan 5

```
SELECT student id, student name
 FROM ms student
 WHERE student id IN(SELECT student id
           FROM trd absensi
           WHERE pertemuan 5 = 'hadir')
  student_id student_name
 80862 Stella clarissa
  82205
          Agustino
        Imelda putri
  82312
3
  83421
          Juliana
 84321
          Lily Annisa Clarissa
  85131
          Titis annisa astrini
  87078
         Kevin Pratama
```

d. Menampilkan payment_id yang total paymentnya lebih besar dari total payment dengan regist id reg001 dan reg003 dan reg004

```
SELECT *
FROM tr_payment
WHERE total_payment > all(SELECT total_payment
FROM tr_payment
WHERE regist_id IN('reg001','reg003', 'reg009'))|

| payment_id regist_id Total_payment Payment_date
| Polos reg005 2000000 2008-02-05 00:00:00.000
```

e. Menampilkan payment_id yang total paymentnya lebih besar dari total payment dengan regist id reg001 atau reg003 atau reg004

