	<b>FAKULTAS TEKNIK</b>		
	<b>UNIVERSITAS NEGERI YOGYAKARTA</b>		
	<b>LAB SHEET BASIS DATA</b>		
	Semester : 3	Accessing Data in Multiple Tables	200 menit
No. : LST/PTI/PTI 222/08	Revisi : 01	Tgl. : 16-12-2008	Hal. 1 dari 8 hal.

1. Kompetensi  
Memahami berbagai cara (perintah SQL) yang digunakan untuk mengakses data dari banyak tabel.
2. Sub Kompetensi  
Dapat menggunakan dan mengimplementasikan perintah SQL untuk memperoleh data atau informasi yang melibatkan banyak tabel.
3. Dasar Teori
  - 1) BASIC JOIN  
Mengambil data dari banyak tabel dimana nama-nama tabel yang terlibat ditulis semua dengan menggunakan koma sebagai pemisah antar tabel.  
Contoh :  
Diketahui permasalahan sebagai berikut :  
Menampilkan nama pelanggan yang pernah membeli Sapu (tampilkan nama dan tanggal pembeliannya).

Solusi :

Untuk memperoleh nama pelanggan, kita harus melibatkan tabel pelanggan karena nama pelanggan adanya di tabel tersebut. Sedangkan untuk menampilkan tanggal transaksi kita harus melibatkan tabel transaksi. Terakhir nama barang, dalam hal ini Sapu, kita perlu menggunakan tabel barang. Tabel pelanggan terhubung dengan tabel transaksi (lihat ERD dalam modul 3). Sedangkan untuk dapat mengakses tabel barang, tabel transaksi harus melalui tabel detail transaksi baru dapat terhubung dengan tabel barang. Jadi dalam perintah SQL kita harus melibatkan keempat tabel tersebut (pelanggan, transaksi, detail\_transaksi, barang).

Perintah SQL-nya (basic join) :

```
SELECT pelanggan.nama,transaksi.tgl_transaksi
FROM pelanggan,transaksi,detail_transaksi,barang
WHERE pelanggan.id_pelanggan=transaksi.id_pelanggan AND
transaksi.kode_transaksi=detail_transaksi.kode_transaksi AND
detail_transaksi.id_barang=barang.id_barang AND barang.nama='Sapu'
ORDER BY pelanggan.nama;
```


Perintah SQL di atas dapat disederhanakan dengan menggunakan alias untuk nama tabel sebagai berikut :

```
SELECT a.nama,b.tgl_transaksi
FROM pelanggan a,transaksi b,detail_transaksi c,barang d
WHERE a.id_pelanggan=b.id_pelanggan AND
b.kode_transaksi=c.kode_transaksi AND
c.id_barang=d.id_barang AND d.nama='Sapu' ORDER BY a.nama;
```

Catatan :

Basic Join ini sudah anda kenal pada praktikum-praktikum sebelumnya.

Dibuat oleh : ADI	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh :
----------------------	--	------------------

	<b>FAKULTAS TEKNIK</b>		
	<b>UNIVERSITAS NEGERI YOGYAKARTA</b>		
	<b>LAB SHEET BASIS DATA</b>		
	Semester : 3	Accessing Data in Multiple Tables	200 menit
No. : LST/PTI/PTI 222/08	Revisi : 01	Tgl. : 16-12-2008	Hal. 2 dari 8 hal.

2) INNER dan CROSS JOIN

Perintah SQL untuk permasalahan yang sama dengan di atas dengan menggunakan INNER JOIN adalah :

```
SELECT pelanggan.nama,transaksi.tgl_transaksi
FROM pelanggan INNER JOIN transaksi ON
pelanggan.id_pelanggan=transaksi.id_pelanggan
INNER JOIN detail_transaksi ON
transaksi.kode_transaksi=detail_transaksi.kode_transaksi
INNER JOIN barang ON detail_transaksi.id_barang=barang.id_barang
WHERE barang.nama='Sapu' ORDER BY pelanggan.nama;
```

Anda juga dapat menggunakan alias untuk nama tabel seperti halnya pada Basic Join :

```
SELECT p.nama,t.tgl_transaksi
FROM pelanggan p INNER JOIN transaksi t ON
p.id_pelanggan=t.id_pelanggan
INNER JOIN detail_transaksi d ON t.kode_transaksi=d.kode_transaksi
INNER JOIN barang b ON d.id_barang=b.id_barang
WHERE b.nama='Sapu' ORDER BY p.nama;
```

Atau untuk lebih efisien gunakan USING untuk menggantikan klausa ON seperti berikut ini :

```
SELECT p.nama,t.tgl_transaksi
FROM pelanggan p INNER JOIN transaksi t USING (id_pelanggan)
INNER JOIN detail_transaksi d USING (kode_transaksi)
INNER JOIN barang b USING (id_barang)
WHERE b.nama='Sapu' ORDER BY p.nama;
```


Atau hilangkan klausa INNER seperti berikut ini :

```
SELECT p.nama,t.tgl_transaksi
FROM pelanggan p JOIN transaksi t USING (id_pelanggan)
JOIN detail_transaksi d USING (kode_transaksi)
JOIN barang b USING (id_barang)
WHERE b.nama='Sapu' ORDER BY p.nama;
```

Klausa INNER JOIN dapat diganti dengan CROSS JOIN sehingga kedua perintah tersebut dapat saling menggantikan :

```
SELECT p.nama,t.tgl_transaksi
FROM pelanggan p CROSS JOIN transaksi t USING (id_pelanggan)
CROSS JOIN detail_transaksi d USING (kode_transaksi)
CROSS JOIN barang b USING (id_barang)
WHERE b.nama='Sapu' ORDER BY p.nama;
```

Dibuat oleh : ADI	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh :
----------------------	--	------------------

	<b>FAKULTAS TEKNIK</b>		
	<b>UNIVERSITAS NEGERI YOGYAKARTA</b>		
	<b>LAB SHEET BASIS DATA</b>		
	Semester : 3	Accessing Data in Multiple Tables	200 menit
No. : LST/PTI/PTI 222/08	Revisi : 01	Tgl. : 16-12-2008	Hal. 3 dari 8 hal.

3) LEFT OUTER JOIN dan RIGHT OUTER JOIN

Perhatikan dan diskusikan dengan teman anda hasil dari ketiga perintah SQL berikut ini dan bandingkan dengan BASIC JOIN, INNER JOIN, dan CROSS JOIN.

- a) `SELECT DISTINCT(p.nama),t.kode_transaksi  
FROM pelanggan p INNER JOIN transaksi t USING (id_pelanggan)  
ORDER BY t.kode_transaksi;`
- b) `SELECT DISTINCT(p.nama),t.kode_transaksi  
FROM pelanggan p LEFT OUTER JOIN transaksi t USING(id_pelanggan)  
ORDER BY t.kode_transaksi;`
- c) `SELECT DISTINCT(p.nama),t.kode_transaksi  
FROM transaksi t RIGHT OUTER JOIN pelanggan p USING  
(id_pelanggan)  
ORDER BY t.kode_transaksi;`

Perintah SQL berikut ini merupakan contoh aplikasi dari LEFT dan RIGHT OUTER JOIN :


- a) `SELECT DISTINCT(p.nama),t.kode_transaksi  
FROM pelanggan p LEFT OUTER JOIN transaksi t USING(id_pelanggan)  
WHERE t.kode_transaksi IS NULL ORDER BY t.kode_transaksi;`
- b) `SELECT DISTINCT(p.nama),t.kode_transaksi  
FROM transaksi t RIGHT OUTER JOIN pelanggan p USING  
(id_pelanggan)  
WHERE t.kode_transaksi IS NULL ORDER BY t.kode_transaksi;`
- c) `SELECT DISTINCT(k.nama),t.kode_transaksi  
FROM transaksi t RIGHT OUTER JOIN karyawan k USING (id_karyawan)  
WHERE t.kode_transaksi IS NULL ORDER BY t.kode_transaksi;`

4) NATURAL JOIN

Bandingkan perintah SQL dengan NATURAL JOIN berikut ini dengan BASIC JOIN, INNER JOIN, CROSS JOIN, LEFT OUTER JOIN, dan RIGHT OUTER JOIN.

- a) `SELECT DISTINCT(p.nama),t.kode_transaksi  
FROM pelanggan p NATURAL JOIN transaksi t  
ORDER BY t.kode_transaksi;`
- b) `SELECT DISTINCT(p.nama),t.kode_transaksi  
FROM pelanggan p NATURAL LEFT JOIN transaksi t  
ORDER BY t.kode_transaksi;`
- c) `SELECT DISTINCT(p.nama),t.kode_transaksi  
FROM pelanggan p NATURAL LEFT OUTER JOIN transaksi t  
ORDER BY t.kode_transaksi;`

Dibuat oleh : ADI	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh :
----------------------	--	------------------

	<b>FAKULTAS TEKNIK</b>		
	<b>UNIVERSITAS NEGERI YOGYAKARTA</b>		
	<b>LAB SHEET BASIS DATA</b>		
	Semester : 3	Accessing Data in Multiple Tables	200 menit
No. : LST/PTI/PTI 222/08	Revisi : 01	Tgl. : 16-12-2008	Hal. 4 dari 8 hal.

- d) `SELECT DISTINCT(p.nama),t.kode_transaksi  
FROM transaksi t NATURAL RIGHT JOIN pelanggan p  
ORDER BY t.kode_transaksi;`
- e) `SELECT DISTINCT(p.nama),t.kode_transaksi  
FROM transaksi t NATURAL RIGHT OUTER JOIN pelanggan p  
ORDER BY t.kode_transaksi;`
- 5) UPDATE dan DELETE
- Masukkan data baru pada tabel suplier sebagai berikut :  
`INSERT INTO suplier VALUES  
( 'S009','Dapur Ngepul','Jl. Dapurmu No.666','08122855666',NULL),  
( 'S010','Patmo','Jl. Perjuangan No.45','08122954545','45@yahoo.co.id');`
- Masukkan data baru pada tabel barang sebagai berikut :  
`INSERT INTO barang VALUES  
( 'B014','Wafer Tanggo','103','2500','5.0','2.5','1','S009'),  
( 'B015','Wafer Nissin','73','2400','4.0','2.5','1','S010');`
- Salah satu cara melakukan perubahan harga barang yang dipasok oleh suplier Patmo sebesar 5% adalah :  
`UPDATE barang b JOIN suplier s USING(id_suplier)  
SET b.harga=b.harga+(b.harga*0.05)  
WHERE s.nama='Patmo';`
- Berikut beberapa cara untuk melakukan penghapusan data suplier yang memasok makanan "wafer" adalah :
- a) Apakah perintah SQL ini bisa dieksekusi?  
`DELETE suplier FROM suplier s,barang b  
WHERE s.id_suplier=b.id_suplier AND b.nama REGEXP 'wafer';`
- b) Bagaimana dengan keempat perintah SQL berikut ini?
- `DELETE suplier FROM suplier,barang  
WHERE suplier.id_suplier=barang.id_suplier  
AND barang.nama REGEXP 'wafer';`
  - `DELETE s FROM suplier s,barang b  
WHERE s.id_suplier=b.id_suplier AND b.nama REGEXP 'wafer';`
  - `DELETE s FROM suplier s JOIN barang b USING (id_suplier)  
WHERE b.nama REGEXP 'wafer';`
  - `DELETE FROM s USING suplier s,barang b  
WHERE s.id_suplier=b.id_suplier AND b.nama REGEXP 'wafer';`

Dibuat oleh : ADI	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh :
----------------------	--	------------------



**FAKULTAS TEKNIK  
UNIVERSITAS NEGERI YOGYAKARTA  
LAB SHEET BASIS DATA**

Semester : 3	Accessing Data in Multiple Tables	200 menit
No. : LST/PTI/PTI 222/08	Revisi : 01	Tgl. : 16-12-2008 Hal. 5 dari 8 hal.

6) SUBQUERIES

Merupakan perintah SQL yang terdiri dari lebih dari satu perintah SQL dan digunakan untuk mengambil data dari lebih dari satu tabel. Subquery biasanya terdiri dari dua perintah SQL. Perintah SQL pertama disebut dengan perintah SQL utama dan perintah SQL kedua disebut subquery. Berikut contoh perintah SQL dengan subquery yang digunakan untuk menampilkan supplier yang memasok “sapu” (a) dan supplier yang memasok jenis barang “minuman” (b).

- a) `SELECT nama FROM supplier WHERE id_supplier= (SELECT id_supplier FROM barang WHERE nama='sapu');`
- b) `SELECT nama FROM supplier WHERE id_supplier<> (SELECT id_supplier FROM barang WHERE nama='sapu');`
- c) `SELECT DISTINCT(s.nama) FROM supplier s INNER JOIN barang b USING (id_supplier) WHERE b.id_jenis_brg= (SELECT id_jenis_brg FROM jenis_barang WHERE nama_jenis_brg='Minuman');`
- d) `SELECT DISTINCT(s.nama) FROM supplier s INNER JOIN barang b USING (id_supplier) WHERE b.id_jenis_brg<> (SELECT id_jenis_brg FROM jenis_barang WHERE nama_jenis_brg='Minuman');`

Bagaimana dengan perintah berikut ini?

`SELECT nama FROM supplier WHERE id_supplier= (SELECT id_supplier FROM barang WHERE id_jenis_brg=1);`

Seringkali subquery menggunakan operator IN, EXISTS, ANY, SOME, NOT, ALL dan atau operator pembandingan (=, >, <, <>, >=, <=) yang digunakan untuk dihubungkan kedua perintah SQL.

**IN dan NOT IN** digunakan untuk membandingkan data pada suatu tabel terhadap data pada tabel yang lain.

Contoh :

1. Menampilkan barang-barang yang dibeli oleh pelanggan.  
`SELECT nama FROM barang WHERE id_barang IN (SELECT id_barang FROM detail_transaksi);`

Bandingkan perintah SQL di atas dengan kedua perintah SQL berikut ini :

- a. `SELECT DISTINCT(b.nama) FROM barang b JOIN detail_transaksi d USING(id_barang);`
- b. `SELECT DISTINCT(b.nama) FROM barang b NATURAL JOIN detail_transaksi d;`


2. Menampilkan barang-barang yang belum pernah dibeli oleh pelanggan.

Masukan dahulu data barang berikut ini :

`INSERT INTO barang VALUES ('B014','Green Tea','180','2450','2.0','1','2','S003'), ('B015','Sikat','11','1500','4.0','0.0','4','S006');`

`SELECT nama FROM barang WHERE id_barang NOT IN (SELECT id_barang FROM detail_transaksi);`

Dibuat oleh : ADI	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh :
----------------------	--	------------------

	<b>FAKULTAS TEKNIK</b>		
	<b>UNIVERSITAS NEGERI YOGYAKARTA</b>		
	<b>LAB SHEET BASIS DATA</b>		
	Semester : 3	Accessing Data in Multiple Tables	200 menit
No. : LST/PTI/PTI 222/08	Revisi : 01	Tgl. : 16-12-2008	Hal. 6 dari 8 hal.

Bandingkan perintah SQL di atas dengan perintah SQL berikut ini :  
 SELECT DISTINCT(b.nama) FROM barang b LEFT OUTER JOIN  
 detail\_transaksi d  
 USING(id\_barang) WHERE d.id\_barang IS NULL;

**EXISTS dan NOT EXISTS** digunakan untuk mengecek atau mengetahui keberadaan suatu data dalam suatu tabel tertentu.

Contoh :

- a. Menampilkan barang-barang yang dibeli oleh pelanggan.  
 SELECT nama FROM barang WHERE EXISTS  
 (SELECT id\_barang FROM detail\_transaksi WHERE  
 barang.id\_barang=detail\_transaksi.id\_barang);
- b. Menampilkan barang-barang yang belum pernah dibeli oleh pelanggan.  
 SELECT nama FROM barang WHERE NOT EXISTS  
 (SELECT id\_barang FROM detail\_transaksi WHERE  
 barang.id\_barang=detail\_transaksi.id\_barang);

**ANY dan SOME** dapat digunakan bergantian. ANY dan SOME hampir sama dengan IN tetapi ANY dan SOME menggunakan operator perbandingan, sedangkan IN tidak.

Contoh :

- a. SELECT nama FROM barang WHERE persen\_laba > ANY  
 (SELECT b.persen\_laba FROM barang b NATURAL JOIN jenis\_barang j  
 WHERE nama\_jenis\_brg='elektronik');
- b. SELECT nama FROM barang WHERE persen\_laba > SOME  
 (SELECT b.persen\_laba FROM barang b NATURAL JOIN jenis\_barang j  
 WHERE nama\_jenis\_brg='elektronik');

Subquery di atas akan menghasilkan persen laba 7 dan 8. Perintah SQL di atas sama dengan perintah berikut ini :

SELECT nama FROM barang  
 WHERE persen\_laba > 7 OR persen\_laba > 8;

**ALL** digunakan untuk mengambil nilai dari perintah SQL utama yang cocok dengan semua nilai yang ada pada perintah Sub Query.

Contoh :

SELECT nama FROM barang WHERE persen\_laba > ALL  
 (SELECT b.persen\_laba FROM barang b NATURAL JOIN jenis\_barang j  
 WHERE nama\_jenis\_brg='elektronik');

Subquery di atas akan menghasilkan persen laba 7 dan 8. Perintah SQL di atas sama dengan perintah berikut ini :

SELECT nama FROM barang  
 WHERE persen\_laba > 7 AND persen\_laba > 8;

Dibuat oleh : ADI	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh :
----------------------	--	------------------



**FAKULTAS TEKNIK  
UNIVERSITAS NEGERI YOGYAKARTA  
LAB SHEET BASIS DATA**

Semester : 3	Accessing Data in Multiple Tables	200 menit
No. : LST/PTI/PTI 222/08	Revisi : 01	Tgl. : 16-12-2008 Hal. 7 dari 8 hal.

- 7) Sub Query juga bisa dimasukkan di dalam suatu perintah SQL, bukan diluar seperti pada contoh-contoh sebelumnya. Subquery yang seperti ini akan diproses terlebih dahulu, baru kemudian perintah SQL utamanya.

Contoh :

- a. Menampilkan data kapan terakhir kali "charles" melakukan transaksi pembelian.  

```
SELECT nama,id_pelanggan,
(SELECT MAX(tgl_transaksi) FROM transaksi
WHERE pelanggan.id_pelanggan=transaksi.id_pelanggan) 'Transaksi
terakhir'
FROM pelanggan WHERE nama='charles';
```
- b. Menampilkan banyak transaksi yang pernah dilakukan oleh pelanggan (bandingkan dengan contoh pada modul ke-7 untuk kasus yang sama, yaitu GROUP BY).  

```
SELECT nama,id_pelanggan,
(SELECT COUNT(id_pelanggan) FROM transaksi
WHERE pelanggan.id_pelanggan=transaksi.id_pelanggan) 'Banyak
transaksi' FROM pelanggan;
```

4. Alat / Instrument / Aparatus / Bahan

1. Komputer dengan sistem operasi Windows.
2. Perangkat lunak MySQL Browser.

5. Keselamatan Kerja

6. Langkah Kerja

1. Pastikan MySQL Server sudah terinstall di komputer.
2. Jalankan MySQL Browser untuk mengakses MySQL Server.

7. Bahan Diskusi

Buatlah perintah SQL untuk menampilkan :

1. Data pelanggan (nama, kode transaksi, tanggal transaksi, dan total pembayaran transaksinya) pada bulan Juli 2006.
2. Berapa rata-rata transaksi yang terjadi pada bulan Juli 2006?
3. Data pelanggan (nomor 1) yang transaksinya lebih kecil dari rata-rata transaksi pada bulan Juli 2006 (nomor 2).

8. Lampiran

Contoh tampilan atau hasil dari bahan diskusi yang diberikan. Hasilnya kemungkinan berbeda karena ada beberapa data yang dirubah, oleh karena itu lakukan pengecekan secara manual sesuai dengan data anda.

1.

nama	tgl_transaksi	kode_transaksi	total_pembayaran
Merry	2006-06-10 14:38:11	J013	27490.2
Charles	2006-06-02 19:16:01	J011	21001.2
Meiliana	2006-06-05 19:00:12	J012	16764.2
Charles	2006-06-12 13:10:52	J014	14651.85

Dibuat oleh : ADI	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh :
----------------------	--	------------------



**FAKULTAS TEKNIK  
UNIVERSITAS NEGERI YOGYAKARTA  
LAB SHEET BASIS DATA**

Semester : 3	Accessing Data in Multiple Tables	200 menit
No. : LST/PTI/PTI 222/08	Revisi : 01	Tgl. : 16-12-2008
Hal. 8 dari 8 hal.		

2.

Rata2
19976.8625

3.

nama	kode_transaksi	tgl_transaksi	total_pembayaran
Meiliana	J012	2006-06-05 19:00:12	16764.2
Charles	J014	2006-06-12 13:10:52	14651.85

Dibuat oleh :  
ADI

Dilarang memperbanyak sebagian atau seluruh isi dokumen  
tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta

Diperiksa oleh :