MODUL V

**JOIN**

* 1. **TUJUAN**
     1. Mahasiswa dapat melakukan *query* pada beberapa tabel menggunakan klausa JOIN.
     2. Mahasiswa dapat menggunakan klausa JOIN dengan benar.
  2. **LANDASAN TEORI**

1. **Jenis-Jenis JOIN**

Secara umum terdapat 5 jenis operasi untuk JOIN, yaitu:

1. **CROSS atau CARTESIAN JOIN**

CROSS JOIN digunakan untuk mendapatkan berbagai kemungkinan kombinasi. Hasil CROSS JOIN ialah kolom yang merupakan gabungan dari semua tabel yang di-JOIN dan baris merupakan semua kombinasi dari semua baris yang di-JOIN [1].

1. **INNER atau EQUI JOIN**

Dapat menampilkan data-data dari kedua tabel dengan men-JOIN-kan *foreign key* dari satu tabel dengan *primary key* dari tabel lainnya. EQUI adalah relasi yang dinyatakan dengan bentuk persamaan (=).

1. **INNER atau NON-EQUI JOIN**

NON-EQUI adalah relasi antara dua atau lebih tabel dimana nilai dari kolom-kolom yang dihubungkan tidak saling terhubung secara langsung (tidak ada hubungan *primarykey* dan *foreignkey*). Dalam NON-EQUI JOIN digunakan operator selain *equal* (=), seperti >, <, <>, >=, <= [3].

1. **OUTER JOIN**

OUTER JOIN digunakan untuk mendapatkan informasi dari suatu tabel yang datanya sesuai dan tidak sesuai. OUTER JOIN terbagi menjadi 3, yaitu [2] :

1. **LEFT OUTER JOIN**

Mencari informasi yang relasinya ada pada kedua sisi tabel dan yang relasi tabel pada sisi kiri tidak dijumpai pada tabel sisi kanan [2].

1. **RIGHT OUTER JOIN**

Mencari informasi yang relasi ada pada kedua sisi tabel dan relasi tabel yang ada pada sisi kanan tabel dan tidak dijumpai pada sisi kiri tabel.

1. **FULL OUTER JOIN**

Melakukan JOIN yang menghasilkan semua baris dari tabel kiri dan tabel kanan.Namun, saat ini FULL OUTER JOIN sudah tidak ada pada SQL.Maka untuk melakukan FULL OUTER JOIN dilakukan dengan menggabungkan LEFT OUTER JOIN dengan RIGHT OUTER JOIN.

1. **SELF JOIN**

SELF JOIN sama dengan JOIN lainnya, hanya melibatkan tabel yang sama.

1. ***Query* JOIN pada Beberapa Tabel**

Masing-masing operasi untuk JOIN, memiliki *query*-nya masing-masing, yaitu:

1. **CROSS atau Kartesian JOIN**

*Query* umum untuk CROSS JOIN adalah:

|  |
| --- |
| SELECT \*FROM nama\_tabel1, nama\_tabel2; |

1. **INNER atau EQUI JOIN**

*Query* umum untuk INNER JOIN:

|  |
| --- |
| SELECT \*FROM nm\_tabel1 INNER JOIN nm\_tabel2 ON tabel1.kolom1=tabel2.kolom2; |

|  |
| --- |
| SELECT \*FROM nm\_tabel1, nm\_tabel2 WHERE tabel1.kolom1= tabel2.kolom2; |

1. **INNER atau NON-EQUI JOIN**

*Query* umum untuk INNER atau NON-EQUI JOIN:

|  |
| --- |
| SELECT \*FROM nm\_tabel1 INNER JOIN nm\_kolom1 ON tabel1.kolom1 [>||<||<>||>=||<=] tabel2.kolom2; |

|  |
| --- |
| SELECT \*FROM nm\_tabel1, nm\_tabel2 WHERE tabel1.kolom1 [>||<||<>||>=||<=] tabel2.kolom2; |

1. **OUTER JOIN**
2. **LEFT OUTER JOIN**

|  |
| --- |
| SELECT \*FROM nm\_tabel1 LEFT JOIN nm\_tabel2 ON nm\_tabel1.nm\_kolom1=nm\_tabel2.nm\_kolom2; |

1. **RIGHT OUTER JOIN**

|  |
| --- |
| SELECT \*FROM nm\_tabel1 RIGHT JOIN nm\_tabel2 ON nm\_tabel1.nm\_kolom1=nm\_tabel2.nm\_kolom2; |

1. **FULL OUTER JOIN**

|  |
| --- |
| SELECT \*FROM nm\_tabel1 LEFT JOIN nm\_tabel2 ON nm\_tabel1.nm\_kolom1=nm\_tabel2.nm\_kolom2  UNION  SELECT \*FROM nm\_tabel1 RIGHT JOIN nm\_tabel2 ON nm\_tabel1.nm\_kolom1=nm\_tabel2.nm\_kolom2; |

1. **SELF JOIN**

*Query* umum untuk SELF JOIN :

|  |
| --- |
| SELECT alias1\_tabel1.kolom1, alias2\_tabel2.kolom2 FROM tabel as alias1, tabel AS alias2 WHERE alias1\_tabel.kolom1= alias2\_tabel1.kolom; |

* 1. **KASUS KELOMPOK**

Sekolah Dasar Internasional Wakanda (SBIW) akan merancang sebuah sistem informasi. Basis data yang dibentuk berdasarkan data-data berbagai entitas. Guru akan didata nama, kontak dan alamat sehingga terbentuk NUPTK. Lalu guru dibedaan menjadi guru honorer dan PNS yang memiliki NIP. Semua guru dapat mengajar mata pelajaran, dan guru pun menguasai tidak hanya satu saja mata pelajaran, namun terkadang mereka harus mengajar mata pelajaran yang tidak mereka pelajari, sehingga mata pelajaran dapat dikuasai dan diajar oleh banyak guru. Mata pelajaran hanya didata namanya saja, sehingga dibentuk kode mata pelajaran. Setiap mata pelajaran yang diajar guru akan didata pegisian kelas mana yang diajar. Sehingga guru dapat mengajar mata pelajaran yang sama di kelas yang berbeda, begitupun dikelas yang sama dapat mengajar mata pelajaran lain. Kelaspun harus di data nama kelasnya sehinggaa terbentuk kode kelas. Sebuah kelas dapat diwalikan oleh seorang guru non-honorer. Sebuah kelaspun memiliki banyak siswa, dimana dari mereka terdapat satu yang menjadi ketua kelas. Setiap siswa akan didata nama, alamay, nama wali, kontak wali, dan rangking mereka hingga dibentuk NIS. Setiap siswa dapat mempelajari banyak mata pelajaran, begitu pula sebaliknya setiap mata pelajaran dapat dipelajari banyak siswa. Setiap kegiatan belajar mengajar akan dilakukan pengambilan nilai.

|  |
| --- |
|  |

**Gambar 5.1** ERD setelah di normalisasi

* 1. **KASUS MODUL**

1. Membuat *query inner join*, *non equi join*, *cross join*, *natural join*, *left join*, dan *right join* pada 2 tabel.
2. Membuat *queryinnerjoin*, *non equi join*, *cross join*, *natural join*, *left join*, dan *right join* pada 3 tabel.
   1. **ANALISIS KASUS**

Berikut merupakan analisis dari hasil kasus modul dan kasus kelompok

* + 1. ***Join* dua tabel dengan kondisi**

Berikut merupakan penjelasan tentang *join* yang dilakukan terhadap dua tabel.

1. ***Inner Join***

|  |
| --- |
|  |

**Gambar 5.2** *Inner join* dua tabel

Pada **Gambar 5.2** duah buah tabel yaitu “guru” dan “mengajar”, dari kedua buah tabel tersebut ditampilkan kolom “nama\_guru” dan “kode\_mapel” yang diajar oleh guru-guru tersebut, “nama\_guru” dan “kode\_mapel” yang diajar diperoleh dari *query* “select nama\_guru,kode\_mapel from guru inner join mengajar on guru.nuptk=mengajar.nuptk;”, jadi bisa disimpulkan bahwa tabel tersebut merupakan relasi antara “guru” dengan “kode\_mapel” apa yang diajar.

1. ***Non equi Join***

|  |
| --- |
| c |

**Gambar 5.3** *Non equi join* dua tabel

Pada **Gambar 5.3** dua buah tabel yaitu “siswa” dan “kelas”, pada tabel tersebut ditampilkan “nama\_siswa”, “nis”, “nama\_wali”, dan “NIS\_ketuakelas” dimana pada tabel tersebut ada kondisi dimana “nama\_siswa” tidak boleh ada yang bernama “I Putu Angga Purnama Widiarta” untuk ditampilkan pada tabel atau dalam artian lain, tabel tersebut menampilkan data diluar dari nama yang ada pada kondisi tersebut. Hal ini merupakan hasil dari *query* “select nama\_siswa,nis,nama\_wali,NIS\_ketuakelas from siswa inner join kelas where nama\_siswa!=”I Putu Angga Purnama Widiarta” limit 5;”, fungsi *limit* pada *query* tersebut digunakan untuk membatasi data yang ingin ditampilkan.

1. ***Cross Join***

|  |
| --- |
|  |

**Gambar 5.4** *Cross join* dua tabel

Pada **Gambar 5.4** dua buah tabel yaitu “mata\_pelajaran” dan “mengajar”, “kode\_mapel” pada kedua tabel dimana pada “mata\_pelajaran”bernilai sama dengan “mengajar” dengan *query* “select \* from mata\_pelajaran cross join mengajar where kode\_mapel=5 limit 3;” ditambahkan dengan kondisi dimana “kode\_mapel” bernilai 5.

1. ***Natural Join***

|  |
| --- |
|  |

**Gambar 5.5** Natural *join* dua tabel

Pada **Gambar 5.5** dua buah tabel yaitu “guru” dan “mengajar”, “nama\_guru” pada kedua tabel dimana pada “guru”bernilai sama dengan “mengajar” dengan *query* “select \* from guru natural join mengajar where nama\_guru=”Imam Suryadi”;” ditambahkan dengan kondisi dimana “nama\_guru” bernama “Imam Suryadi”.

1. ***Left Join***

|  |
| --- |
|  |

**Gambar 5.6** *Left join* dua tabel

Pada **Gambar 5.6** dua buah tabel yaitu “guru” dan “mengajar”, “nuptk” pada kedua tabel dimana pada “guru”bernilai sama dengan “mengajar” dengan *query* “select \* from guru left join mengajar on guru.nuptk=mengajar.nuptk limit 6;”. Pada *query* tersebut ada suatu kondisi dimana ada seorang “guru” yang belum mengajar “mata\_pelajaran” apapun, kondisi tersebut dapat diketahui dikarenakan adanya penggunaan dari *left join* itu sendiri, hal ini dapat disimpulkan melalui “kode\_mapel” yang bernilai *null*.

1. ***Right* *Join***

|  |
| --- |
|  |

**Gambar 5.7** *Right join* dua tabel

Pada **Gambar 5.7** dua buah tabel yaitu “guru” dan “mengajar”, “nuptk” pada kedua tabel dimana pada “guru”bernilai sama dengan “mengajar” dengan *query* “select \* from tiket right join membeli on tiket.harga=membeli.id\_penonton limit 2;”. Pada tabel tersebut dapat dilihat adanya “kode\_mapel” yang belum memiliki pengajar, kondisi tersebut terjadi karena fungsi dari *right join* yang menampilkan ketidaksesuaian pada suatu tabel.

* + 1. ***Join* tiga tabel**

Berikut merupakan penjelasan tentang *join* yang dilakukan terhadap tiga tabel berbeda :

1. ***Inner join***

|  |
| --- |
|  |

**Gambar 5. 8** *Inner join* pada tiga tabel

Pada **Gambar 5.8** tiga buah tabel yaitu “guru”, “mengusai”, pada tabel tersebut ditampilkan “nuptk”, “nama\_guru”, “kode\_mapel”, dan “nilai”. Ada sebuah kondisi dimana “nilai” yang diminta untuk ditampilkan hanya “nilai” yang bernilai sama dengan “100”, *query* dari tabel tersebut adalah “select guru.nuptk, nama\_guru, menguasai.kode\_mapel,nilai from guru inner join menguasai inner join mempelajari where nilai=100 limit 3;”. *Limit* pada *query* diatas difungsikan untuk membatasi data yang dapat ditampilkan pada tabel yang dimana jumlah data yang dapat ditampilkan hanya maksimal 3 data saja.

1. ***Non equi Join***

|  |
| --- |
|  |

**Gambar 5.9** *Non equi join* pada tiga tabel

Pada **Gambar 5.9** tiga buah tabel yaitu “guru”, “menguasai”, dan “mempelajari”, dimana data pada tabel tersebut hanya menampilkan data dengan “nilai” dibawah “100”. Data-data pada tabel tersebut ditampilkan melalui *query* “select guru.nuptk, nama\_guru, menguasai.kode\_mapel,nilai from guru inner join menguasai inner join mempelajari where nilai<100 limit 3;”. *Limit* pada *query* diatas difungsikan untuk membatasi data yang dapat ditampilkan pada tabel yang dimana jumlah data yang dapat ditampilkan hanya maksimal 3 data saja.

1. ***Cross Join***

|  |
| --- |
|  |

**Gambar 5.10** *Cross join* pada tiga tabel

Pada **Gambar 5.10** tiga buah tabel yaitu “guru”, “membeli”, dan “penonton”, dimana pada tabel tersebut ditampilkan gabungan dari setiap kolom yang ada dari ketiga tabel diatas. Pada kolom “nilai”, “nilai” yang hanya bisa ditampilkan yaitu “nilai” yang sama dengan “100”, pada tabel ini tidak ada hubungan benar-benar saling berelasi, *cross join* hanya digunakan untuk menggabungkan kolom-kolom yang ada pada setiap tabel yang ada pada sebuah *database*.

1. ***Natural Join***

|  |
| --- |
|  |

**Gambar 5. 11** Natural *join* tiga tabel

Pada **Gambar 5.11** tiga buah tabel yaitu “guru”, “menguasai”, dan “mempelajari”, dimana pada tabel ini akan menampilkan “guru” yang dimana “mata\_pelajaran” yang dikuasai olehnya, siswa-siswinya mendapatkan nilai “100” hal ini terjadi dikarenakan fungsi dari *query* “select guru.nuptk,nama\_guru,kode\_mapel,nilai from guru natural join menguasai natural join mempelajari where nilai=100;”. Hal ini dapat terjadi dikarenakan ketiga tabel tersebut saling berhubungan antara satu dengan lainnya, fungsi dari *primary key* dan *foreign key* dapat kita buktikan melalui tampilan dari tabel diatas, dikarenakan kita dapat mengetahui “guru” yang “menguasai” suatu “mata\_pelajaran” dimana siswa-siswinya dapat memperoleh “nilai” yang sama dengan “100”.

1. ***Left Join***

|  |
| --- |
|  |

**Gambar 5. 12** *Left join* tiga tabel

Pada **Gambar 5.12** tiga buah tabel yaitu “guru”, “menguasai”, dan “mempelajari”, dimana “nuptk” pada “guru”bernilai sama dengan “menguasai” dan “kode\_mapel” pada “mempelajari”bernilai sama dengan “mempelajari” dengan *query* “select guru.nuptk,nama\_guru,menguasai.kode\_mapel,nilai from guru left join menguasai on guru.nuptk=menguasai.nuptk left join mempelajari on menguasai.kode\_mapel=mempelajari.kode\_mapel order by kode\_mapel limit3;”.

1. ***Right Join***

|  |
| --- |
|  |

**Gambar 5. 13** *Right join* tiga tabel

Pada **Gambar 5.13** tiga buah tabel yaitu “guru”, “menguasai”, dan “mempelajari”, dimana “nuptk” pada “guru”bernilai sama dengan “menguasai” dan “kode\_mapel” pada “mempelajari”bernilai sama dengan “mempelajari” dengan *query* “select guru.nuptk,nama\_guru,menguasai.kode\_mapel,nilai from guru right join menguasai on guru.nuptk=menguasai.nuptk right join mempelajari on menguasai.kode\_mapel=mempelajari.kode\_mapel order by kode\_mapel limit3;”.

* 1. **KESIMPULAN**

Berdasarkan praktikum yang telah dilakukan pada Modul 5 JOIN, dapat disimpulkan bahwa:

1. Terdapat 5 operasi dalam *join* yaitu *inner join*, *outer join*, *natural join*, *cross join*, dan *self join*.
2. *Inner join* dan *equi join* memiliki kesamaan dalam melakukan penampilan data. Perbedaannya hanya pada klausa penghubungnya, yaitu “inner join” pada *inner join* dan tanda koma “,” pada *equi join*. O*uter join* terdiri dari dua jenis *join*, yaitu *left join* dan *right join*. Pada *left join*, yang menjadi patokan adalah tabel sisi kiri, sedangkan pada *right join*, yang menjadi patokan adalah tabel sisi kanan. *Cross join* merupakan kombinasi perkalian dari semua baris yang ada dalam semua tabel yang digabungkan.*Natural join* hampir sama dengan *cross join*, hanya saja pada *natural join* diberikan kondisi sehingga baris hasilnya tidak banyak. *Self join*, tabel yang terlibat dalam penggabungan hanya satu. Penggabungan ini berguna jika ada dua kolom dalam tabel yang sama, namun ingin digunakan dua kali.