



Institut Informatika & Bisnis
DARMAJAYA
Yayasan Alfian Husin



**Kampus
Merdeka**
INDONESIA JAYA

**MERDEKA
BELAJAR**

DATA SCIENCE DARMAJAYA
“YOUR BEST FUTURE IN DATA”

MEETING: [25 & 26]

STRUCTURE QUERY LANGUAGE (SQL) DATA MANIPULATION LANGUAGE (DML) SELECT

BY: HENDRA KURNIAWAN



STRUCTURE QUERY LANGUAGE (SQL) DATA MANIPULATION LANGUAGE (DML) INSERT, UPDATE, DELETE, SELECT

1. Introduction to SQL
2. Data Manipulation Language (DML)
3. Select Commands (SELECT, WHERE, AND, OR, NOT, DISTINCT, BETWEEN, KOLOM ALIAS, LIKE, SORTING, GROUPING, JOIN)



Learning Objectives

1. Students be able to comprehend and apply the principles of Data Manipulation Language (DML),
2. Students will gain the skills to select values in one or more tables.



DISTINCT

DISTINCT used to filter duplicate data/more than one row of the same displayed into just one row.

Example:

Select **Distinct** (KodeJurusan) From Jurusan



BETWEEN

BETWEEN, used to check a value between a given or specified range of values. BETWEEN is usually used to check values of the number type.

Example:

```
Select * from Sales Where TotalSales Between 100000 and 120000
```



ALIAS

COLUMN ALIAS, used to give another name to a column. The alias is written after the name of the column or expression that will be replaced or named by using the keyword 'AS' as the replace command. The writing rules are as follows:

**Select KodeJurusan as [Kode Jurusan], Jurusan as [Nama Jurusan]
from Jurusan**

(Note: If you want to have spaces in the column, then use the [] sign)



LIKE

LIKE, is used to search for data on a character by character basis according to search needs. By using the LIKE condition, any character can be searched, it can be one, two or all the characters in the string being searched. To get the desired character we can use the character WILD CARD. WILDCARD functions to represent the characters to be searched for according to the existing query. Using wildcard characters can make the LIKE operator/condition more flexible than using string comparison operators – or !=.



LIKE (Cont'd)

1. Character %

The % character is used to represent all unknown characters before, in the middle or after the character to be searched. The writing rules are as follows:

Select KodeJurusan from Jurusan where Jurusan like 'SI%' atau

Penjelasan: mencari kode Jurusan dengan karakter awal S

Select KodeJurusan from Jurusan where KodeJurusan like '%S' atau

Penjelasan: mencari kode Jurusan dengan karakter kedua S

Select KodeJurusan from Jurusan where KodeJurusan like '%S%'

Penjelasan: mencari kode Jurusan dengan karakter ditengah S



LIKE (Cont'd)

2. Character _

The _ character is used to represent a character to be searched for. The writing rules are as follows:

Select KodeJurusan from Jurusan where Jurusan like '_A%' or

Select KodeJurusan from Jurusan where Jurusan like 'A_%'



SORTING

SORTING, used to sort the data resulting from **SELECT**. The clause used for the sorting process is **ORDER BY**. Sorting can be done by ascending and descending. The writing rules are as follows:

```
Select KodeJurusan, Jurusan from Jurusan Order By Jurusan Asc
```

```
Select KodeJurusan, Jurusan from Jurusan Order By Jurusan Desc
```



GROUPING

GROUPING, used to group data based on the list of columns mentioned in the SELECT command. The writing rules are as follows:

```
Select KodeJurusan, Jurusan  
from Jurusan  
Group By Jurusan  
Order By Jurusan Asc
```



GROUPING

GROUPING, used to group data based on the list of columns mentioned in the SELECT command. The writing rules are as follows:

```
Select KodeJurusan, Jurusan  
from Jurusan  
Group By Jurusan  
Order By Jurusan Asc
```



HAVING

HAVING, is used to limit the results of grouping data based on the list of columns mentioned in the SELECT command. The writing rules are as follows:

Select *

From Penjualan

Group By NoInvoice

Order By NoInvoice Asc

Having TotalPenjualan>120000



JOIN

JOIN, is an operation used to get combined data from two or more tables. Joins are used to obtain more detailed data from tables that are related to each other. The join command is usually found/used in the Select. Join command has three types of operations, including:

1. Cross Join
2. Inner Join
3. Outer Join

CROSS JOIN

Cross Join produces a combination of all rows in the joined tables, both paired and unpaired. This join is not actually used. Example (there are 2 tables of departments and students)

Tabel Jurusan

<u>Kode Jurusan</u>	<u>Jurusan</u>
SI	<u>Sistem Informasi</u>
MI	<u>Manajemen Informatika</u>

Tabel Mahasiswa

<u>NPM</u>	<u>Kode Jurusan</u>
03050006	SI
03050036	MI

Select * From Jurusan Cross Join Mahasiswa → **hasilnya:**

<u>Kode Jurusan</u>	<u>Jurusan</u>	<u>NPM</u>	<u>Kode Jurusan</u>
SI	<u>Sistem Informasi</u>	03050006	SI
MI	<u>Manajemen Informatika</u>	03050006	SI
SI	<u>Sistem Informasi</u>	03050036	MI
MI	<u>Manajemen Informatika</u>	03050036	MI



INNER JOIN

Inner Join produces all rows with the same value contained in the joined tables. Writing Rules:

Select Jurusan.Jurusan, Mahasiswa.NPM From Jurusan

INNER JOIN Mahasiswa ON

Jurusan.KodeJurusan=Mahasiswa.KodeJurusan



INNER JOIN

Select Jurusan.Jurusan, Mahasiswa.NPM From Jurusan
INNER JOIN Mahasiswa ON
Jurusan.KodeJurusan=Mahasiswa.KodeJurusan

<u>Tabel Jurusan</u>		<u>Tabel Mahasiswa</u>	
<u>Kode Jurusan</u>	<u>Jurusan</u>	<u>NPM</u>	<u>Kode Jurusan</u>
SI	Sistem Informasi	03050006	SI
MI	Manajemen Informatika	03050036	MI
TI	Teknik Informatika		

Tabel Mengajar

<u>NPM</u>	<u>Ruang</u>
03050006	F.3.6

Hasilnya:

<u>Jurusan</u>	<u>NPM</u>
Sistem Informasi	03050006
Manajemen Informatika	03050036

INNER JOIN

Select Jurusan.Jurusan, Mahasiswa.NPM, Mengajar.Ruang
From Jurusan INNER JOIN Mahasiswa ON
Jurusan.KodeJurusan=Mahasiswa.KodeJurusan
INNER JOIN MENGAJAR ON Mahasiswa.NPM=Mengajar.NPM

<u>Tabel Jurusan</u>		<u>Tabel Mahasiswa</u>	
<u>Kode Jurusan</u>	<u>Jurusan</u>	<u>NPM</u>	<u>Kode Jurusan</u>
SI	Sistem Informasi	03050006	SI
MI	Manajemen Informatika	03050036	MI
TI	Teknik Informatika		

<u>Tabel Mengajar</u>	
<u>NPM</u>	<u>Ruang</u>
03050006	F.3.6

Hasilnya:

<u>Jurusan</u>	<u>NPM</u>	<u>Ruang</u>
Sistem Informasi	03050006	F.3.6

OUTER JOIN

the same as Inner Join, the difference is that the outer join displays data that has pairs or does not have pairs. Meanwhile, inner joins are only for couples. Outer joins are divided into 3 types, namely:

1. Left outer join, The data in the table on the left will be displayed all according to their pairs, if there are no pairs then the table on the right will be filled with null.
2. Right outer join, The data in the table on the right will be displayed all according to their pairs, if there are no pairs then the table on the left will be filled with null.
3. Full outer join, all data from the joined tables is displayed, whether it is a left outer join, right outer join, inner join.



LEFT OUTER JOIN

Select Jurusan.Jurusan, Mahasiswa.NPM From Jurusan
LEFT JOIN Mahasiswa ON Jurusan.KodeJurusan=Mahasiswa.KodeJurusan

<u>Tabel Jurusan</u>		<u>Tabel Mahasiswa</u>	
<u>Kode Jurusan</u>	<u>Jurusan</u>	<u>NPM</u>	<u>Kode Jurusan</u>
SI	<u>Sistem Informasi</u>	03050006	SI
MI	<u>Manajemen Informatika</u>	03050036	MI
TI	<u>Teknik Informatika</u>		

<u>Tabel Mengajar</u>	
<u>NPM</u>	<u>Ruang</u>
03050006	F.3.6

Hasilnya:

<u>Jurusan</u>	<u>NPM</u>
<u>Sistem Informasi</u>	03050006
<u>Manajemen Informatika</u>	03050036
<u>Teknik Informatika</u>	Null



RIGHT OUTER JOIN

Select Jurusan.Jurusan, Mahasiswa.NPM From Jurusan RIGHT JOIN Mahasiswa ON Jurusan.KodeJurusan=Mahasiswa.KodeJurusan

<u>Tabel Jurusan</u>		<u>Tabel Mahasiswa</u>	
<u>Kode Jurusan</u>	<u>Jurusan</u>	<u>NPM</u>	<u>Kode Jurusan</u>
SI	<u>Sistem Informasi</u>	03050006	SI
MI	<u>Manajemen Informatika</u>	03050036	MI
TI	<u>Teknik Informatika</u>		

Tabel Mengajar

<u>NPM</u>	<u>Ruang</u>
03050006	F.3.6

Hasilnya:

<u>Jurusan</u>	<u>NPM</u>
<u>Sistem Informasi</u>	03050006
<u>Manajemen Informatika</u>	03050036



FULL OUTER JOIN

Select Jurusan.Jurusan, Mahasiswa.NPM From Jurusan
FULL JOIN Mahasiswa ON Jurusan.KodeJurusan=Mahasiswa.KodeJurusan

<u>Tabel Jurusan</u>		<u>Tabel Mahasiswa</u>	
<u>Kode Jurusan</u>	<u>Jurusan</u>	<u>NPM</u>	<u>Kode Jurusan</u>
SI	<u>Sistem Informasi</u>	03050006	SI
MI	<u>Manajemen Informatika</u>	03050036	MI
TI	<u>Teknik Informatika</u>		

<u>Tabel Mengajar</u>	
<u>NPM</u>	<u>Ruang</u>
03050006	F.3.6

Hasilnya:

<u>Jurusan</u>	<u>NPM</u>
<u>Sistem Informasi</u>	03050006
<u>Manajemen Informatika</u>	03050036
<u>Teknik Informatika</u>	Null



CONCLUSION

- The SELECT-clause lists the attributes or functions to be retrieved
- The FROM-clause specifies all relations (or aliases) needed in the query but not those needed in nested queries
- The WHERE-clause specifies the conditions for selection and join of tuples from the relations specified in the FROM-clause
- GROUP BY specifies grouping attributes
- HAVING specifies a condition for selection of groups
- ORDER BY specifies an order for displaying the result of a query
- A query is evaluated by first applying the WHERE-clause, then GROUP BY and HAVING, and finally the SELECT-clause
- JOIN consists of CROSS JOIN, INNER JOIN, OUTER JOIN



REFERENCES

1. Connolly. T., Begg. Carolyn. 2015. Database System: A Pratical Approach to Design, Implementation, and Management. Sixth Edition. Global Edition. Pearson.



Institut Informatika & Bisnis
DARMAJAYA
Yayasan Alfian Husin



**Kampus
Merdeka**
INDONESIA JAYA

**MERDEKA
BELAJAR**

THANK YOU!!

DATA SCIENCE DARMAJAYA "YOUR BEST FUTURE IN DATA"