# MODUL IV

***QUERY* PADA TABEL**

* 1. **TUJUAN**
     1. Mahasiswa dapat melakukan *query* pada satu tabel
     2. Mahasiswa dapat melakukan *query* berdasarkan kondisi tertentu.
  2. **DASAR TEORI**
     1. ***Query* Pada Tabel Tunggal**

Sebelum membahas akan tentang *Query* pada table tunggal kita mengenal terlebih dahulu apa itu *Query. Query* adalah semacam kemampuan untuk menampilkan suatu data dari database dimana mengambil dari tabel-tabel yang ada dalam database., namun tabel tersebut tidak semua ditampilkan sesuai yang kita inginkan. Data apa yang ingin kita tampilkan. Misalnya, data pinjaman dengan buku pinjaman, maka nanti akan mengambil data dari tabel peminjam dan tabel buku.

Semua tipe data MySQL dapat diindex-kan. Menggunakan index pada kolom yang sesuai merupakan cara terbaik untuk meningkatkan performa dari operator SELECT. Nomor maksimal index setiap tabel dan panjang index terdefinisi berdasarkan penyimpanan (*Storage Engine*).

Perintah untuk memperbaiki dan Menambah *primary key*

|  |
| --- |
| ALTER TABLE NAMA\_TABLE ADD PRIMARY KEY(KOLOM1, KOLOM2, ...); |

Contoh :

|  |
| --- |
| alter table Checker add primary key (idchecker); |

Perintah Klausa SELECT, FROM, AND dan WHERE

Query Select :

|  |
| --- |
| SELECT \* FROM barang; |

Query WHERE :

|  |
| --- |
| SELECT \* FROM nama\_tabel WHERE nama\_kolom = nama\_kriteria; |

MySQL menyediakan *query* SELECT, LIKE untuk menampilkan tabel berdasarkan pencarian karakter sederhana. Format dasar struktur SELECT, LIKE adalah sebagai berikut:

|  |
| --- |
| SELECT nama\_kolom\_tampil FROM nama\_tabel WHERE nama\_kolom\_cari LIKE keyword\_pencarian |

* nama\_kolom\_tampil adalah nama dari kolom yang akan kita tampilkan, bisa semua kolom dalam tabel, atau hanya kolom tertentu saja.
* nama\_tabel adalah nama tabel dimana nama\_kolom\_tampil berada.
* nama\_kolom\_cari adalah kolom yang akan kita gunakan untuk pencarian.
* keyword\_pencarian merupakan kata kunci yang digunakan untuk pencarian. Cara penulisan kata kunci ini akan kita bahas lebih lanjut.

* + 1. ***Query* Pada Banyak Tabel**

MySQL dapat membuat index gabungan (maksudnya membuat index dengan *query* memanggil dua kolom dari sebuah data atau tabel).

1. **JOIN**

Join tabel adalah penggabungan tabel-tabel menggunakan *Query* yang dilakukan melalui kolom/*key* tertentu yang memiliki nilai terkait untuk mendapatkan satu set data dengan informasi lengkap. Lengkap disini artinya kolom data didapatkan dari kolom-kolom hasil join antar tabel tersebut. Join diperlukan karena perancangan tabel pada sistem transaksional kebanyakan dinormalisasi, salah satu alasannya untuk mengurangi redundansi.

Pengaksesan data atau pencarian data dengan menggunakan *Query* atau Join pada database perlu memperhatikan ketepatgunaan implementasi dari data itu sendiri serta waktu prosesnya. Ada banyak cara yang dapat dilakukan oleh database manajemen sistem dalam memproses dan menghasilkan jawaban sebuah *query*. Semua cara pada akhirnya akan menghasilkan jawaban (output) yang sama tetapi pasti mempunyai harga yang berbeda-beda, seperti misalnya total waktu yang diperlukan untuk menjalankan sebuah *query*.

|  |
| --- |
| SELECT nama\_tabel1.nama\_kolom, nama\_tabel2.nama\_kolom FROM nama\_tabel  INNER JOIN nama\_tabel ON nama\_tabel.nama\_kolom=nama\_tabel.nama\_kolom; |

Contoh :

|  |
| --- |
| SELECT Penjualan.id\_barang, Penjualan.id\_buyer, Barang.nama\_barang FROM Penjualan  INNER JOIN Barang ON Penjualan.id\_barang=Barang.id\_barang; |

1. **Clausa SELECT**

Salah satu *syntax* yang paling penting dalam SQL adalah select. Select berfungsi untuk menampilkan data-data pada tabel-tabel yang terdapat dalam database. Di sinilah terdapat banyak perbedaan antara SQL biasa dan SQL Oracle. Oracle banyak sekali meringkas *syntax* menjadi lebih sederhana walaupun sebenarnya oracle juga tetap memerima SQL strandar.

Perintah Select :

|  |
| --- |
| SELECT nama\_tabel1.nama\_kolom, nama\_tabel2.nama\_kolom FROM nama\_tabel1, nama\_tabel2; |

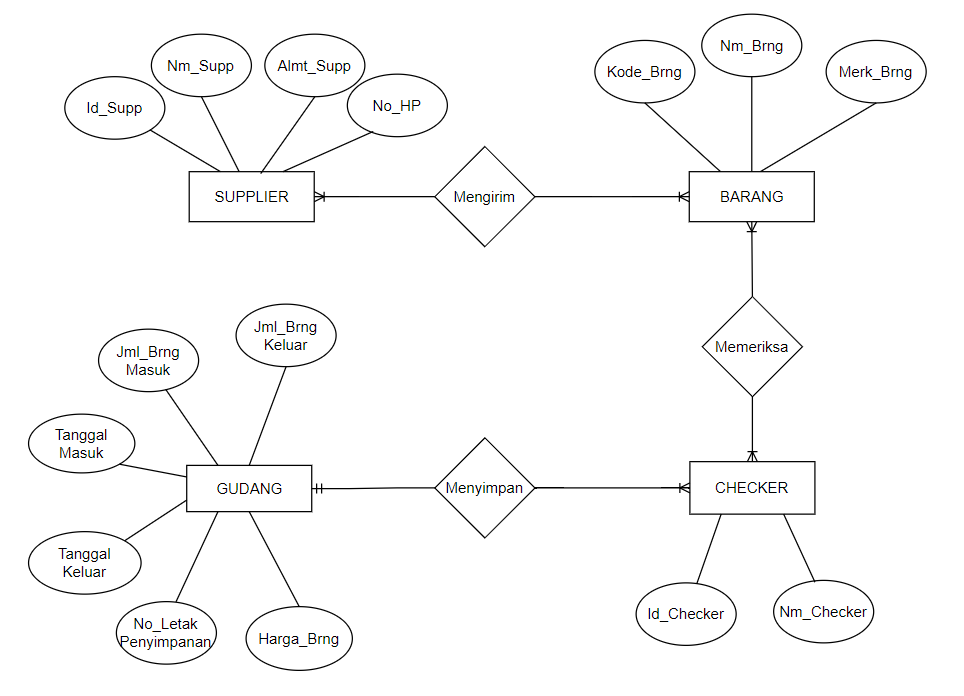
Contoh:

|  |
| --- |
| SELECT barang.id\_barang, penyuplai.nama\_penyuplai FROM barang, penyuplai; |

* 1. **STUDI KASUS**

**Daftar Persediaan Barang pada Gudang**

* 1. **DATA HASIL**
     1. **Entity Relationship Diagram**

****

Gambar 4.1 ERD (Entity Relationship Diagram)

* + 1. **Jobsheet**

|  |  |  |
| --- | --- | --- |
| 1 | Memasukkan *Primary Key* baru pada tabel database yang anda buat sebelumnya | alter table Checker add primary key (idchecker); |
|  |
| Pada gambar diatas merupakan perintah dalam sql untuk menambahkan *primary key* baru pada tabel yang sudah dibuat menggunakan *query* “alter table nama\_tabel add *primary key* (nama\_kolom);”. Pada gambar diatas menambahkan *primary key* pada tabel Checker dimana idchecker sebagai *primary key* pada tabel tersebut. Setelah berhasil maka selanjutnya kita dapat melihat deskripsi dari tabel menggunakan “desc namatabel” dan akan menampilkan semua data keterangan dari tabel tersebut. |
| 2 | Menggunakan variasi Select, from, and dan where pada tabel database yang telah anda buat untuk menampilkan data | select namachecker from Checker where idchecker = 'F1B020148' AND namachecker like '%wilad%'; |
|  |
| Pada gambar diatas merupakan perintah dalam sql untuk menampilkan data yang terdapat pada sebuah tabel dengan menggunakan beberapa variasi yaitu and dan where dengan *query* “select namachecker from Checker where idchecker = 'F1B020148' AND namachecker like '%wilad%';” pada gambar diatas berhasil menampilkan namachecker dengan berapa kondisi. |
| 3 | Menampilkan urutan data pada tabel database yang anda buat dengan semua data yang mengandung huruf “h” di dalamnya. | select namachecker from Checker where namachecker like '%h%'; |
|  |
| Pada gambar diatas merupakan perintah sql yang dimana untuk menampilkan data pada suatu tabel dengan kondisi namachecker mengandung huruf “H” dengan *query* “select namachecker from Checker where namachecker like '%h%';” ‘ |
| Menampilkan urutan data pada tabel database yang anda buat dengan semua data yang akhiran “A dan I” | select namachecker from Checker where namachecker like '%a' or namachecker like '%i'; |
|  |
| Pada gambar diatas merupakan perintah sql yang dimana untuk menampilkan data pada suatu tabel dengan kondisi namachecker mengandung akhiran huruf “a” dan “i” dengan *query* “select namachecker from Checker where namachecker like '%a' or namachecker like '%i'” |
| Menampilkan urutan data pada tabel database yang anda buat dengan semua data yang bukan akhiran “N” | select namachecker from Checker where namachecker not like '%n'; |
|  |
| Pada gambar di atas merupakan perintah SQL untuk menampilkan data pada table menggunakan *query* “select nama\_kolom from nama\_tabel where kondisi not like ’pola’;” di mana pada gambar di atas adalah menampilkan data dari kolom pelanggan dengan kondisi nama pelanggan yang bukan berakhiran huruf ‘n’ sehingga didapatkan hasil seperti pada gambar |
| 4 | Menggunakan sintax REGEXP untuk mencari data yang terdapat pada tabel database yang anda buat | select namachecker from Checker where namachecker regexp 'h'; |
|  |
| Pada gambar diatas merupakan perintah dalam sql yang dimana untuk menampilkan data dari suatu tabel dengan kondisi namachecker yang mengandung huruf “h” dengan menggunakan *query* “select namachecker from Checker where namachecker regexp 'h';” |

|  |  |  |
| --- | --- | --- |
| 1 | Melakukan *query* untuk menampilkan data dari suatu relasi | Select idchecker, namechecker from Checker limit3; |
|  |
| *query* Select idchecker, namechecker from Checker limit3; digunakan untuk dapat menampilkan data idchecker dan namachecker pada satu table yang dimana data diambil dari table “Checker”. Query limit pada perintah SQL diatas digunakan untuk dapat menampilkan jumlah data yang akan dikembalikan untuk menghindari dari perulangan pengembalian data dalam suatu table. |
| 2 | Melakukan *query* untuk menampilkan data dengan menggunakan kondisi tertentu | select idchecker,namachecker from Checker where idchecker like '%F1B020%'; |
|  |
| *Query* select idchecker,namachecker from Checker where idchecker like '%F1B020%'; digunakan untuk menampilkan data dengan kondisi tertentu, dimana kondisi pada *query* tersebut adalah where idchecker like '%F1B020%'; yang berarti yang akan di tampilkan adalah idchecker yang mengandung nilai “F1B020”. |
| 3 | Melakukan *query* dari 2 tabel dengan database yang sudang dibuat sebelumnya | select data.nama, matkul.namaMatkul from data, matkul limit4; |
|  |
| Query select data.nama, matkul.namaMatkul from data, matkul limit4; adalah untuk menggabungkan 2 data pada table yang berbeda . Dimana *query* diatas akan menampilka nama pada table data dan namaMatkul pada tabel matkul dan *query* limit pada perintah SQL diatas digunakan untuk menampilkan jumlah data yang akan dikembalikan sebanyak 4. |
| 4 | Melakukan *query* dari 3 tabel dengan databse yang sudah dibuat sebelumnya | select data.nama, kelas.namaRuangan, matkul.namaMatkul from data, kelas, matkul limit 4; |
|  |
| Query elect data.nama, kelas.namaRuangan, matkul.namaMatkul from data, kelas, matkul limit 4; adalah untuk menggabungkan 3 data pada 3 tabel yang berbeda, dimana data yang akan tampil adalah nama dari tabel data, nama Ruangan dari tabel kelas dan namaMatkul dari tabel matkul. *Query* limit pada perintah SQL diatas digunakan untuk dapat menampilkan jumlah data yang akan dikembalikan sebanyak 4. |

**4.5 Kesimpulan**

1. Melakukan *query* pada satu tabel menggunakan select untuk penentuan lokasi data yang ingin dilakukan *query*, agar mempermudah pencarian data. Terdapat beberapa perintah *query* yang dapat digunakan untuk mempermudah pencarian data dari sebuah database, untuk menggunakan perintah *query* pada sebuah tabel database, pada dasarnya dapat melakukan perintah *query*.

|  |
| --- |
| SELECT \* FROM nama\_tabel; |

1. Melakukan *query* dengan kondisi tertentu, untuk mempermudah pencarian data tanpa harus menampilkan keseluruhan isi tabel. Pencarian data menggunakan *query* dengan kondisi tertentu akan menampilkan data sesuai dengan yang ingin ditampilkan dengan dasar sintax

|  |
| --- |
| SELECT \* FROM nama\_tabel WHERE kondisi LIKE “%keyword%”; |

DAFTAR PUSTAKA

Anonim. 2022. *Modul Praktikum Basis Data.* Laboratorium Jaringan Dan Komputer. Jurusan Teknik Elektro Fakultas Teknik Universitas Mataram.

Fikry, Muhammad. 2019. *Basis Data*. Universitas Malikussaleh. Aceh. Unimal Press.

Jatnika, Hendra. 2012. Buku Pengantar Sistem Basis Data. Politenknik Piksi Ganesha. Bandung Jawa Barat.

Rosid, Muhammad Alfan dan Astutik, Ika Ratna Indra. 2020. *Buku Ajar Basis Data.* Universitas Muhamadiyah Sidoarjo.

Winiarti, Sri. Rosyda, Miftahurrahmah. 2020. Buku ajar matakuliah Basis Data. Laboratorium Teknik Informatika Universitas Ahmad Dahlan. Yogyakarta