SQL (STRUCTURED QUERY LANGUAGE)

PESIAPAN MENGGUNAKAN SQL

SQL adalah sebuah bahasa yang dirancang khusus untuk dapat berkomunikasi dengan database. Dalam buku ini perintah yang banyak digunakan adalah "Pencarian Data", apakah itu menampilkan data, mengurutkan dan lain-lain. Satu hal yang menggembirakan adalah dalam menggunakan perintah SQL tidak ada keahlian khusus yang harus dimiliki oleh user, tapi yang harus dilakukan adalah ketelitian dan kejelian dalam menuliskan perintah SQL itu sendiri. Perintah yang paling dominan dalam pembahasan ini adalah SELECT, yaitu suatu perintah untuk mencari (menampilkan) data yang diinginkan dari tabel tertentu. Syntax umumnya adalah sebagai berikut :

```
SELECT nama_field FROM nama_tabel WHERE kondisi GROUP BY nama_field HAVING group_criteria ORDER BY nama_field
```

Di bawah ini kami ingatkan kembali struktur masing-masing tabel agar anda terhindar dari kesalahan ketika menuliskan syntax SQL.

Peringatan :

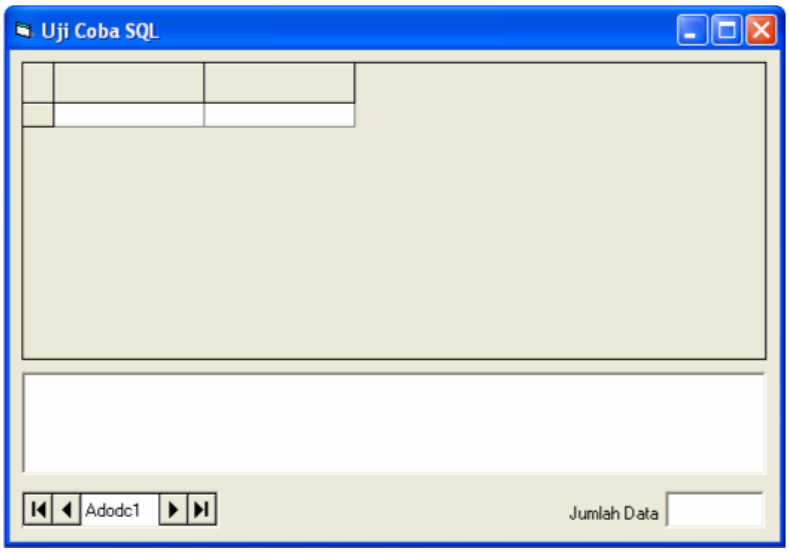
Jika Anda menuliskan perintah SQL dengan nama-nama field yang tidak sesuai dengan rancangan struktur tabel yang telah dijelaskan di atas, maka SQL (dalam program) akan memunculkan pesan kesalahan. Pesan kesalahan inipun dapat terjadi jika aturan main pada Properti Data Control tidak sesuai dengan yang seharusnya.

Dalam penggunaan perintah SQL usahakan Anda tidak memberi nama objek (file master database, tabel dan nama-nama field) dengan nama yang sama dengan keyword (kata kunci) dalam SQl. Di bawah ini terdapat tabel yang berisi beberapa kata kunci SQL.

CONTOH-CONTOH SYNTAX SQL

SYNTAX DASAR SQL

Setelah Anda memahami aturan main penggunaan SQL dalam Visual Basic, langkah selanjutnya adalah rancanglah sebuah form dengan bentuk seperti di bawah ini :



Koding :

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
On Error GoTo salah
KeyAscii = Asc(UCase(Chr(KeyAscii)))
If KeyAscii = 27 Then End
If KeyAscii = 13 Then
    Adodc1.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source= " & App.Path & "\DBmaster.mdb"
    Adodc1.RecordSource = Text1
    Adodc1.Refresh
    Text2 = Adodc1.Recordset.RecordCount
    Set DataGrid1.DataSource = Adodc1
    DataGrid1.Refresh
    If Adodc1.Recordset.EOF Then
        MsgBox "Data Tidak Ditemukan"
        Adodc1.Refresh
        Text1.SetFocus
    End If
End If
On Error GoTo 0
Exit Sub
salah:
MsgBox "Syntax SQL Salah..!"
End Sub
```

Setelah anda membuat rancangan form seperti gambar di atas, maka cobalah beberapa perintah SQL di bawah ini :

SELECT KodeBrg FROM Barang

Fungsi : menampilkan field KodeBrg yang ada pada tabel barang

Hasil :

KodeBrg
BRG001
BRG002
BRG003
BRG004
BRG005
BRG006

SELECT KodeBrg, NamaBrg FROM barang

Fungsi : menampilkan kolom KodeBrg dan NamaBrg yang ada pada tabel barang

Hasil :

KobeBrg	NamaBrg
BRG001	PROCESSOR P III
BRG002	KOMPUTER
BRG003	PRINTER
BRG004	MONITOR
BRG005	SPEAKER
BRG006	KEYBOARD SERIAL

SELECT KodeBrg, NamaBrg, Harga FROM barang

Fungsi : menampilkan field KodeBrg, NamaBrg dan harga yang ada pada tabel barang

Hasil :

KobeBrg	NamaBrg	Harga
BRG001	PROCESSOR P III	450000
BRG002	KOMPUTER	2500000
BRG003	PRINTER	550000
BRG004	MONITOR	700000

BRG005	SPEAKER	35000
BRG006	KEYBOARD SERIAL	35000

SELECT KodeBrg, NamaBrg, Harga, Jumlah FROM barang

Fungsi: menampilkan kolom KodeBrg, NamaBrg, Harga dan Jumlah (seluruh field) yang ada pada tabel barang

Hasil :

KobeBrg	NamaBrg	Harga	Jumlah
BRG001	PROCESSOR P III	450000	12
BRG002	KOMPUTER	2500000	5
BRG003	PRINTER	550000	10
BRG004	MONITOR	700000	3
BRG005	SPEAKER	35000	15
BRG006	KEYBOARD SERIAL	35000	10

Perintah di atas dirasakan terlalu panjang dan kemungkinan kesalahan penulisan nama field akan lebih besar, untuk menghasilkan data yang sama SQL memberikan perintah yang lebih singkat yaitu:

SELECT * FROM Barang

Tanda * merupakan sebuah perintah untuk menampilkan seluruh field.

Kesimpulan :

Berdasarkan beberapa perintah di atas yang telah dijelaskan Anda dapat mengambil satu atau beberapa field yang dibutuhkan, karena dalam kondisi tertentu sering kali kolom-kolom lain tidak dibutuhkan pada saat melakukan pencarian data yang diinginkan.

Hal lain yang perlu diperhatikan adalah penggunaan tanda * (asterisk). Pada satu sisi dengan cara tersebut memang mempercepat penulisan tapi di sisi lain hal seperti ini terkadang mengganggu konsentrasi, karena beberapa field yang tidak dijadikan sasaran pencarian akan tetap ditampilkan walaupun tidak diperlukan dan user akan cenderung melihat seluruh data yang ditampilkannya, padahal kolom-kolom tersebut tidak dibutuhkan.

MENGURUTKAN DATA DENGAN KLAUSA ORDER BY

Mengurutkan data merupakan hal yang cukup penting dalam database, karena dengan data yang terurut Anda dapat melakukan pencarian dengan cepat. Dalam SQL perintah yang digunakan untuk mengurutkan data adalah ORDER BY. Dalam perintah ini terdapat dua pilihan yang dapat digunakan yaitu ASC (singkatan dari ASCENDING) yang berfungsi untuk mengurutkan data dari yang terkecil hingga yang terbesar dan DESC (singkatan dari DESCENDING) yang berfungsi untuk mengurutkan data dari yang terbesar hingga yang terkecil.

SELECT * FROM Barang ORDER BY KodeBrg

Fungsi : menampilkan seluruh field di tabel barang yang diurutkan berdasarkan kode barang.

Pada perintah di atas SQL akan secara otomatis memberikan nilai ASC, yaitu mengurutkan data dari yang terkecil hingga yang terbesar. Dalam hal ini jika kode barang diawali dengan huruf, maka SQL akan mengurutkan data dari kode barang dengan awalan A (atau a), tapi jika kode barang itu bercampur dengan angka maka SQL akan mengurutkan data dari huruf berawalan A kemudian data yang diawali dengan angka 0 sampai 9.

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah
BRG001	PROCESSOR P III	450000	12
BRG002	KOMPUTER	2500000	5
BRG003	PRINTER	550000	10
BRG004	MONITOR	700000	3
BRG005	SPEAKER	35000	15
BRG006	KEYBOARD SERIAL	35000	10

SELECT * FROM Barang ORDER BY NamaBrg

Fungsi: menampilkan seluruh field di tabel barang yang diurutkan berdasarkan nama barang.

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah
BRG006	KEYBOARD SERIAL	35000	10
BRG002	KOMPUTER	2500000	5
BRG004	MONITOR	700000	3
BRG003	PRINTER	550000	10

BRG001	PROCESSOR P III	450000	12
BRG005	SPEAKER	35000	15

Perhatikanlah hasil tampilan di atas, dengan perintah tersebut maka pengurutan berdasarkan kode barang akan diabaikan karena SQL mengurutkannya berdasarkan nama barang dari yang terkecil, artinya dari nama barang yang diawali dengan huruf A (atau a) hingga huruf Z (atau z).

SELECT * FROM Barang ORDER BY Harga

Fungsi : menampilkan seluruh field di tabel barang yang diurutkan berdasarkan harga.

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah
BRG005	SPEAKER	35000	15
BRG006	KEYBOARD SERIAL	35000	10
BRG001	PROCESSOR P III	450000	12
BRG003	PRINTER	550000	10
BRG004	MONITOR	700000	3
BRG002	KOMPUTER	2500000	5

SELECT * FROM Barang ORDER BY Jumlah

Fungsi : menampilkan seluruh field di tabel barang yang diurutkan berdasarkan Jumlah

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah
BRG004	MONITOR	700000	3
BRG002	KOMPUTER	2500000	5
BRG003	PRINTER	550000	10
BRG006	KEYBOARD SERIAL	35000	10
BRG001	PROCESSOR P III	450000	12
BRG005	SPEAKER	35000	15

SELECT * FROM Barang ORDER BY KodeBrg Desc

Fungsi : menampilkan seluruh data (field) di tabel barang yang diurutkan berdasarkan kode barang diurutkan dari yang terbesar.

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah
BRG006	KEYBOARD SERIAL	35000	10
BRG005	SPEAKER	35000	15
BRG004	MONITOR	700000	3
BRG003	PRINTER	550000	10
BRG002	KOMPUTER	2500000	5
BRG001	PROCESSOR P III	450000	12

SELECT * FROM Barang ORDER BY KodeBrg, NamaBrg

Fungsi : menampilkan seluruh data (field) di tabel barang yang diurutkan berdasarkan (pertama) Kode barang dan (kedua) nama barang diurutkan dari yang terkecil.

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah
BRG001	PROCESSOR P III	450000	12
BRG002	KOMPUTER	2500000	5
BRG003	PRINTER	550000	10
BRG004	MONITOR	700000	3
BRG005	SPEAKER	35000	15
BRG006	KEYBOARD SERIAL	35000	10

Perintah di atas dilakukan jika pada nama barang terdapat huruf awal yang sama, maka SQL akan mengurutkan data berdasarkan nama barang yang diawali dengan huruf a atau A hingga huruf z atau Z. Cara lain yang dapat digunakan untuk melakukan hal yang sama adalah dengan menuliskan nilai kolomnya. Dalam hal ini KodeBrg dianggap kolom 1, NamaBrg kolom 2, Harga kolom 3 dan Jumlah kolom 4, jadi perintah SQL yang ditulis adalah :

SELECT * FROM Barang ORDER BY 1,2

Untuk hal yang sama tulislah perintah di bawah ini:

SELECT * FROM Barang ORDER BY 1

SELECT * FROM Barang ORDER BY 3

SELECT * FROM Barang ORDER BY 4 Desc

Catatan :

Klausa ORDER BY harus disimpan setelah klausa WHERE (jika ada), atau disimpan di posisi terakhir dari perintah SQL, karena jika penempatannya tidak benar maka akan muncul pesan kesalahan. Jika Anda menuliskan perintah ASC atau DESC maka perintah tersebut akan dilakukan pada nama field terdekat. Sebagai contoh :

SELECT * FROM Barang ORDER BY Harga ,Jumlah Desc

Fungsi : menampilkan seluruh kolom di tabel barang yang diurutkan pertama berdasarkan harga (secara otomatis bernilai asc) kemudian diurutkan berdasarkan jumlah secara descending.

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah
BRG005	SPEAKER	35000	15
BRG006	KEYBOARD SERIAL	35000	10
BRG001	PROCESSOR P III	450000	12
BRG003	PRINTER	550000	10
BRG004	MONITOR	700000	3
BRG002	KOMPUTER	2500000	5

Perhatikanlah hasil tampilan di atas, pada baris pertama dan kedua terdapat harga yang sama yaitu 35000, tapi pada kolom jumlah datanya berbeda yaitu 15 dan 10, karena pengurutan yang kedua berdasarkan jumlah secara descending, maka jumlah 15 disimpan di posisi paling atas sedangkan jumlah 10 disimpan di baris berikutnya. Pada dasarnya perintah di atas dapat diartikan “tampilkanlah

seluruh kolom di tabel barang dengan harga terurut dari yang terkecil kemudian jumlah terurut dari yang terbesar".

Dalam mengurutkan data tidak ada aturan baku bahwa menuliskan nilai kolom dalam ORDER BY harus dimulai dari yang terkecil atau yang terbesar, jadi Anda dapat dengan bebas menuliskan nomor kolom yang dibutuhkan untuk menemukan data yang dicari.

MENCARI DATA DENGAN KLAUSA WHERE

Mencari data yang diinginkan dengan tingkat keakuratan yang tinggi merupakan hal yang tidak kalah pentingnya dibandingkan dengan mengurutkan data. Dalam hal ini Anda akan menggunakan klausa WHERE. Klausa ini harus diikuti dengan nama field dan menuliskan kriteria yang diinginkan.

SELECT KodeBrg FROM Barang WHERE KodeBrg="BRG001"

Fungsi : menampilkan kolom KodeBrg dengan kode barang ="BRG001"

Hasil :

KodeBrg
BRG001

SELECT KodeBrg,NamaBrg FROM Barang WHERE KodeBrg="BRG003"

Fungsi : menampilkan kolom KodeBrg dan NamaBrg dengan kode barang ="BRG003"

Hasil :

KodeBrg	NamaBrg
BRG003	PRINTER

Dengan perintah di atas data yang ditampilkan hanya terbatas pada nama-nama field yang ditulis setelah pernyataan SELECT saja. Satu hal yang perlu diketahui bahwa menggunakan klausa WHERE ini tidak selamanya harus mencari data yang nama kolomnya disebut setelah pernyataan SELECT. Contoh :

SELECT KodeBrg,NamaBrg,Harga FROM Barang WHERE Jumlah>5

Fungsi : menampilkan kolom KodeBrg, NamaBrg dan Harga yang jumlahnya > 5

Hasil :

KodeBrg	NamaBrg	Harga

BRG001	PROCESSOR P III	450000
BRG003	PRINTER	550000
BRG005	SPEAKER	35000
BRG006	KEYBOARD SERIAL	35000

Perhatikanlah perintah di atas. Setelah pernyataan SELECT Anda menuliskan field kode barang, nama barang dan harga, padahal kriteria pencarian berdasarkan jumlah yang secara fisik tidak dituliskan setelah pernyataan SELECT.

SELECT * FROM Barang WHERE KodeBrg=”BRG006”

Fungsi : menampilkan seluruh kolom dengan kode barang =”BRG006”

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah
BRG006	KEYBOARD SERIAL	35000	10

SELECT * FROM Barang WHERE Harga >500000

Fungsi : menampilkan seluruh kolom dengan harga >500000

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah
BRG002	KOMPUTER	2500000	5
BRG003	PRINTER	550000	10
BRG004	MONITOR	700000	3

MENGGUNAKAN KLAUSA AND

Klausa AND ini digunakan untuk mencari dua data atau lebih yang diinginkan, selain itu Anda dapat menggunakan tambahan operator aritmatika sebagai pembanding untuk mencari data yang diinginkan.

Operator	Fungsi
>	Lebih besar
>=	Lebih besar sama dengan

<	Lebih kecil
<=	Lebih kecil sama dengan
<>	Tidak sama dengan

SELECT * FROM Barang WHERE Harga >500000 AND Harga <1000000

Fungsi : menampilkan seluruh kolom dengan harga >500000 dan harga <1000000

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah
BRG003	PRINTER	550000	10
BRG004	MONITOR	700000	3

SELECT * FROM Barang WHERE Harga>400000 AND Jumlah>5

Fungsi : menampilkan seluruh kolom dengan Harga > 400000 dan Jumlah > 5

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah
BRG001	PROCESSOR P III	450000	12
BRG003	PRINTER	550000	10

Kesimpulan :

Jika Anda menggunakan operator AND, maka setelah kata AND Anda harus menuliskan kembali nama field yang akan dijadikan pencarian kedua, ketiga dan seterusnya, jika setelah kata AND tidak ditulis kembali nama field yang dijadikan rujukan pencarian maka akan muncul pesan kesalahan. Penggunaan operator AND dapat dilakukan berulang-ulang (beberapa kali dalam satu baris) selama memungkinkan dan selama syntaxnya benar sesuai dengan aturan penulisan perintah SQL. Contoh :

SELECT * FROM Barang WHERE Harga>=50000 AND Harga<1000000 AND Jumlah>5 AND Jumlah<15

Fungsi : menampilkan seluruh kolom dengan Harga >=50000 dan Harga <1000000 dan Jumlah>5 dan Jumlah <15

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah

BRG001	PROCESSOR P III	450000	12
BRG003	PRINTER	550000	10

5 MENGGUNAKAN KLAUSA OR

SELECT * FROM Barang WHERE KodeBrg=”BRG001” OR Jumlah >5

Fungsi : menampilkan seluruh kolom dengan kode barang=”BRG001 atau Jumlah >5

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah
BRG001	PROCESSOR P III	450000	12
BRG003	PRINTER	550000	10
BRG005	SPEAKER	35000	15
BRG006	KEYBOARD SERIAL	35000	10

MENGGUNAKAN KLAUSA NOT

Perintah ini digunakan untuk mengabaikan (tidak ditampilkan) data yang dicari dengan klausa NOT

SELECT * FROM Barang WHERE NOT KodeBrg=”BRG001”

Fungsi : menampilkan seluruh kolom dengan kode barang bukan ”BRG001”

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah
BRG002	KOMPUTER	2500000	5
BRG003	PRINTER	550000	10
BRG004	MONITOR	700000	3
BRG005	SPEAKER	35000	15
BRG006	KEYBOARD SERIAL	35000	10

Penggunkaan klausa ini sama dengan perintah berikut :

SELECT * FROM Barang WHERE KodeBrg<>"BRG001"

Perbedaannya terletak pada penempatan penulisannya, jika menggunakan klausa NOT maka kata NOT ditulis setelah Klausa WHERE sebelum nama field, tapi jika menggunakan operator aritmatika maka tanda <> (tidak sama dengan) disimpan setelah nama field dan sebelum kriteria data yang dicari. Jika Anda menuliskan kata NOT sebelum klausa WHERE, misalnya :

SELECT * FROM Barang NOT WHERE KodeBrg='BRG001'

Maka hasilnya adalah :

KodeBrg	NamaBrg	Harga	Jumlah
BRG001	PROCESSOR P III	450000	12

MENGGUNAKAN KLAUSA IN

Perintah ini sama dengan operator OR yang artinya tampilkanlah data pertama atau data kedua, dan seterusnya. Contoh :

SELECT * FROM Barang WHERE KodeBrg IN('BRG001','BRG006')

Fungsi : menampilkan seluruh kolom dengan kode barang BRG001 dan BRG006

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah
BRG001	PROCESSOR P III	450000	12
BRG006	KEYBOARD SERIAL	35000	10

Perintah di atas sama dengan perintah di bawah ini :

SELECT * FROM Barang WHERE KodeBrg="BRG001" OR KodeBrg="BRG006"

SELECT * FROM Barang WHERE NOT KodeBrg IN('BRG001','BRG006')

Fungsi : menampilkan seluruh kolom yang kode barangnya bukan BRG001 dan BRG006 Hasil :

KodeBrg	NamaBrg	Harga	Jumlah
BRG002	KOMPUTER	2500000	5
BRG003	PRINTER	550000	10
BRG004	MONITOR	700000	3
BRG005	SPEAKER	35000	15

SELECT * FROM Barang WHERE NOT Jumlah IN(10)

Fungsi : menampilkan seluruh kolom yang jumlahnya bukan 10

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah
BRG001	PROCESSOR P III	450000	12
BRG002	KOMPUTER	2500000	5
BRG004	MONITOR	700000	3
BRG005	SPEAKER	35000	15

MENGGUNAKAN OPERATOR LIKE DAN WILDCARD (% _ [ ])

Operator ini digunakan untuk mencari data dengan menuliskan salah satu atau beberapa karakter yang diinginkan. Operator ini sangat bermanfaat dalam melakukan pencarian data dengan tingkat spesifikasi yang cukup tinggi, karena dengan Wildcard Anda dapat menemukan satu huruf yang diinginkan, tidak peduli apakah huruf tersebut terletak di tengah, di depan atau di belakang sebuah item data. Bahkan dengan Wildcard tersebut Anda dapat mengabaikan satu atau beberapa karakter yang tidak diinginkan.

SELECT * FROM Barang WHERE NamaBrg LIKE “%E%”

Fungsi : menampilkan seluruh kolom di tabel barang dengan nama barang yang mengandung huruf “E” dan huruf di depan atau di belakangnya apa saja.

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah

BRG001	PROCESSOR P III	450000	12
BRG002	KOMPUTER	2500000	5
BRG003	PRINTER	550000	10
BRG005	SPEAKER	35000	15
BRG006	KEYBOARD SERIAL	35000	10

Perhatikanlah perintah di atas dengan baik…!. Penulisan huruf E diapit dengan tanda * yang artinya mengabaikan huruf yang berada di depan dan di belakangnya, dengan demikian jika pada kolom nama barang terdapat data yang mengandung huruf E, maka data tersebut akan ditampilkan.

Penulisan klausa LIKE dalam Visual Basic boleh menggunakan tanda “” atau tanda ‘’, tetapi Anda tidak dibenarkan menuliskan dengan dua lambang yang berbeda yaitu di depan dengan tanda “ tetapi di belakang dengan tanda ‘ dan sebaliknya.

SELECT * FROM Barang WHERE NamaBrg LIKE ‘K%’

Fungsi: menampilkan seluruh kolom di tabel barang dengan nama barang yang huruf depannya “K” dan selanjutnya huruf apa saja.

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah
BRG002	KOMPUTER	2500000	5
BRG006	KEYBOARD SERIAL	35000	10

SELECT * FROM Barang WHERE NamaBrg LIKE ‘%R’

Fungsi : menampilkan seluruh kolom di tabel barang dengan nama barang yang huruf belakangnya “R” dan sebelumnya huruf apa saja.

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah
BRG002	KOMPUTER	2500000	5
BRG003	PRINTER	550000	10
BRG004	MONITOR	700000	3
BRG005	SPEAKER	35000	15

Kesimpulan :

Wildcard dengan tanda % dalam SQL digunakan untuk mengabaikan huruf apa saja yang ada di depan atau di belakangnya dengan jumlah huruf yang tidak terbatas.

SELECT * FROM Barang WHERE NamaBrg LIKE '_O*'

Fungsi : menampilkan seluruh kolom di tabel barang dengan nama barang yang huruf depannya (satu huruf) apa saja, huruf keduanya "O" dan huruf selanjutnya apa saja.

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah
BRG002	KOMPUTER	2500000	5
BRG004	MONITOR	700000	3

SELECT * FROM Barang WHERE NamaBrg LIKE '__M*'

Fungsi : menampilkan seluruh kolom di tabel barang dengan nama barang yang dua huruf depannya apa saja, huruf ketiganya "M" dan huruf selanjutnya apa saja.

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah
BRG002	KOMPUTER	2500000	5

Kesimpulan :

Tanda _ digunakan untuk mengabaikan sejumlah karakter. Jika Anda menulis tanda _ sebanyak dua buah berarti Anda mengabaikan dua huruf. Jadi pengabaian jumlah karakter tergantung pada berapa jumlah tanda _ yang ditulis dan di posisi mana tanda tersebut diletakkan.

SELECT * FROM Barang WHERE NamaBrg LIKE '[PK]%'

Fungsi : menampilkan seluruh kolom dimana nama depannya diawali dengan huruf "P" atau huruf "K" dan nama belakangnya apa saja.

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah
BRG001	PROCESSOR P III	450000	12
BRG002	KOMPUTER	2500000	5
BRG003	PRINTER	550000	10
BRG006	KEYBOARD SERIAL	35000	10

Kesimpulan :

Dengan menggunakan tanda [] berarti Anda mencari huruf dengan pilihan tergantung huruf apa yang dicantumkan dalam tanda kurung siku tersebut. pada contoh di atas, karena tanda [] diakhiri dengan tanda % yang diletakkan di luar, maka artinya Anda mencari huruf awal dengan pilihan huruf yang ada pada tanda [] dengan huruf berikutnya apa saja.

MENGGUNAKAN PERHITUNGAN MATEMATIS

Pada kesempatan ini akan dicoba menggunakan perintah matematis. Pada tabel barang tidak terdapat Total Harga sebagai hasil perkalian antara Harga dengan Jumlah, namun dengan perintah SQL dapat ditampilkan hasil perhitungan matematis tersebut.

SELECT KodeBrg,NamaBrg,Harga,Jumlah,Harga*Jumlah AS Total FROM Barang

*Fungsi : menampilkan seluruh kolom di tabel barang dimana HARGA * JUMLAH sebagai field (kolom) baru yaitu TOTAL*

Hasil :

KodeBrg	NamaBrg	Harga	Jumlah	Total
BRG001	PROCESSOR P III	450000	12	5400000
BRG002	KOMPUTER	2500000	5	12500000
BRG003	PRINTER	550000	10	5500000
BRG004	MONITOR	700000	3	2100000

BRG005	SPEAKER	35000	15	525000
BRG006	KEYBOARD SERIAL	35000	10	350000

SELECT *,Harga*Jumlah AS Total FROM Barang

Hasil :

Total	KodeBrg	NamaBrg	Harga	Jumlah
5400000	BRG001	PROCESSOR P III	450000	12
12500000	BRG002	KOMPUTER	2500000	5
5500000	BRG003	PRINTER	550000	10
2100000	BRG004	MONITOR	700000	3
525000	BRG005	SPEAKER	35000	15
350000	BRG006	KEYBOARD SERIAL	35000	10

Pada contoh di atas, penulisan tanda bintang dianggap sebuah perintah untuk menampilkan seluruh kolom, dengan demikian jika Anda menulis perintah SELECT *, harga * jumlah as total FROM barang, kolom total akan disimpan di posisi paling awal.

SELECT Harga,Jumlah,Harga*Jumlah AS Total FROM Barang WHERE Jumlah=10

Fungsi : menampilkan kolom harga, jumlah dan total sebagai field baru hasil kalkukasi yang jumlah barangnya =10

Hasil :

Harga	Jumlah	Total
550000	10	5500000
35000	10	350000

MENGELOMPOKAN DATA

SELECT Harga,Sum(Harga) AS Gabung FROM Barang GROUP BY Harga

Fungsi : menampilkan kolom Harga yang jumlahnya sama digabungkan disimpan dalam field (kolom) baru bernama Gabung

Hasil :

Harga	Gabung
35000	70000
450000	450000
550000	550000
700000	700000
2500000	2500000

Perintah di atas artinya mengelompokkan data Harga dimana jika ditemukan data (harga) yang sama, maka data tersebut dijumlahkan dan disimpan dalam sebuah field baru dengan nama Gabung. Perhatikanlah baris pertama tabel di atas. Karena data harga 35000 ada dua maka data tersebut dijumlahkan dan disimpan pada sebuah field baru dengan nama Gabung dengan nilai 70000.

SELECT Harga, Count(Harga) AS Jumlah FROM Barang GROUP BY Harga

Fungsi : kelompokan Harga dan hitung berapa jumlah harga yang sama dimana hasil perhitungannya harga yang sama tersebut disimpan pada field baru bernama Jumlah.

Hasil :

Harga	Jumlah
35000	2
450000	1
550000	1
700000	1
2500000	1

SELECT Jumlah,Sum(Jumlah) AS Gabung FROM Barang GROUP BY

Jumlah Hasil :

Jumlah	Gabung
3	3
5	5
10	20
12	12

15	15

Perintah di atas artinya mengelompokkan data Jumlah dimana jika ditemukan data (Jumlah) yang sama, maka data tersebut dijumlahkan dan disimpan dalam sebuah field baru dengan nama Gabung. Perhatikanlah baris ketiga tabel di atas. Karena data Jumlah 10 ada dua maka data tersebut dijumlahkan dan disimpan pada sebuah field baru dengan nama Gabung dengan nilai 20.

SELECT Count(*) AS JumlahData FROM Barang

Fungsi : menghitung jumlah data (baris) dari seluruh field. Data menampilkan angka 6 karena seluruh data ada 6 item (baris)

Hasil

JumlahData
6

SELECT Count(*) AS JumlahData FROM Barang WHERE Jumlah =10

Fungsi : Perintah ini digunakan untuk menghitung jumlah data dari seluruh field dengan jumlah=10. Data menampilkan angka 2 karena seluruh data (jumlah yang bernilai 10) ada 2 item.

Hasil

JumlahData
2

SELECT Max(Harga) AS Terbesar FROM Barang

Fungsi : menampilkan nilai terbesar dari kolom Harga yang ada di tabel barang dan disimpan dalam sebuah kolom baru dengan nama "Terbesar".

Hasil :

Terbesar
2500000

SELECT Min(Harga) AS Terkecil FROM Barang

Hasil :

Terkecil
35000

SELECT Max(Jumlah) AS Terbesar FROM Barang

Hasil :

Terbesar
15

SELECT Min(Jumlah) AS Terkecil FROM Barang

Hasil :

Terkecil
3

SELECT Count(*) AS Jml, Max(Harga) AS Terbesar, Min(Harga) AS Terkecil FROM Barang

Fungsi : Perintah di atas digunakan untuk menampilkan jumlah seluruh data yang disimpan dalam field baru yaitu "Jml", harga terbesar disimpan dalam field "Terbesar" dan harga terkecil pada field "Terkecil".

Hasil :

Jml	Terbesar	Terkecil
6	2500000	35000

MENGGUNAKAN DATA TANGGAL

Data tanggal dalam suatu database merupakan suatu hal yang sangat penting dan data tanggal ini disimpan dalam suatu field dengan suatu type tertentu yaitu DATE. Anda dapat memanipulasi data tanggal sedemikian rupa sesuai dengan data yang dibutuhkan. Untuk memulainya cobalah syntx-syntax di bawah ini.

SELECT * FROM Penjualan WHERE CDATE (Tanggal)="7/23/2008"

Fungsi : menampilkan data Penjualan tanggal 23 juli 2008.

Sebelum field Tanggal ditulis CDATE, hal ini dilakukan karena data tanggal yang diketik berupa karakter, maka Anda harus mengubah data tanggal yang ada pada tabel Penjualan dengan bentuk string.

Hasil :

FAKTUR	TANGGAL	TOTAL	KODEPLG
F01	7/23/2008	2400000	PLG01

Cara lain yang dapat dilakukan adalah mengubah bentuk karakter tanggal menjadi bentuk nilai tanggal yaitu dengan perintah DATEVALUE.

SELECT * FROM Penjualan WHERE Tanggal=DateValue('23/07/2008')

Perintah tersebut mempunyai arti sebaliknya dari perintah sebelumnya, pada perintah CDATE berarti mengubah bentuk data tanggal di dalam tabel Penjualan menjadi bentuk karakter, tetapi dengan perintah DATEVALUE berarti mengubah bentuk text SQL menjadi bentuk nilai tanggal karena data tanggal yang ada dalam tabel nilai default-nya adalah date dengan format dd/mm/yy.

Selain itu Anda pun dapat menuliskan perintah SQL di bawah ini untuk menghasilkan data yang sama :

SELECT * FROM Penjualan WHERE Tanggal =#07-23-2008#

SELECT * FROM Penjualan WHERE CDATE (Tanggal)>'7/23/2008’ AND CDATE (Tanggal) <'7/25/2008'

Fungsi : menampilkan data Penjualan tanggal lebih besar dari tanggal 23 juli dan lebih kecil dari tanggal 25 juli 2008.

Hasil :

FAKTUR	TANGGAL	TOTAL	KODEPLG
F02	7/24/2008	3020000	PLG02

MENGAMBIL DATA (HARI) TANGGAL

Dalam pengolahan data tanggal sering kali Anda menginginkan pencarian berdasarkan tanggal tertentu saja dengan mengabaikan data bulan dan tahun. Dalam hal ini SQL mampu melakukan pencarian tersebut yang secara spesifik Anda tuliskan kata DAY, yang artinya mengambil data berdasarkan tanggal tertentu. Contoh :

SELECT * FROM Penjualan WHERE CDATE (DAY (Tanggal))<=24

Fungsi : menampilkan data Penjualan dengan tanggal faktur lebih kecil sama dengan tanggal 24

Hasil :

FAKTUR	TANGGAL	TOTAL	KODEPLG
F01	7/23/2008	2400000	PLG01
F02	7/24/2008	3020000	PLG02

MENGAMBIL DATA BULAN

Hal yang sama dapat Anda lakukan untuk mencari data tanggal hanya berdasarkan bulannya saja. Dalam hal ini perintah SQL yang harus ditulis dengan menambahkan kata MONTH. Dengan cara seperti ini berarti Anda mengabaikan kondisi tanggal dan tahun. Contoh :

SELECT * FROM Penjualan WHERE CDATE (MONTH (Tanggal))=7

Fungsi : menampilkan data Penjualan yang dilakukan pada bulan Juli.

Hasil :

FAKTUR	TANGGAL	TOTAL	KODEPLG
F01	7/23/2008	2400000	PLG01
F02	7/24/2008	3020000	PLG02
F03	7/25/2008	1205000	PLG03

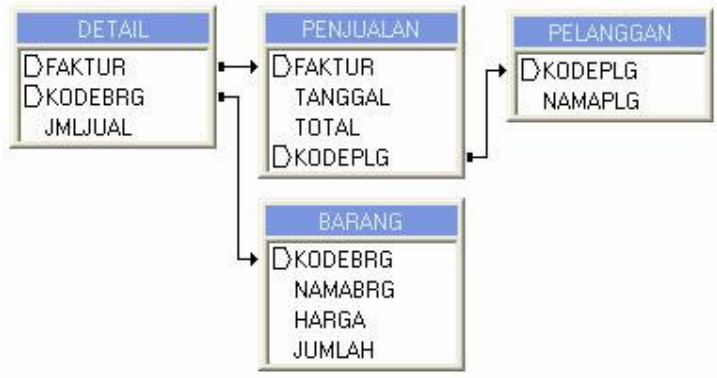
SELECT * FROM Penjualan WHERE CDATE (DAY (Tanggal))>24 AND CDATE (MONTH (Tanggal))=7

Fungsi : Tampilkanlah data Penjualan dengan tanggal faktur lebih besar dari tanggal 24 dan bulan transaksi sama dengan Juli.

Perintah di atas merupakan gabungan dari DAY dan MONTH, cara seperti ini dirasakan sangat penting untuk mencari suatu data tanggal berdasarkan kriteria yang diinginkan. Hal yang sama dapat juga Anda lakukan untuk mencari data tanggal berdasarkan tahunnya saja yaitu dengan menuliskan kata YEAR.

MENGGABUNGKAN TABEL (MATERI PENTING)

Dengan SQL Anda dapat mengambil data dari beberapa tabel sekaligus. Hal ini sangat berguna untuk membuat laporan yang mempunyai nilai informasi bagi pihak-pihak yang membutuhkan. Dalam buku ini hanya ada empat tabel yaitu tabel Barang, Pelanggan, Detail dan tabel Penjualan dan Anda harus mengingat kembali nama-nama field yang ada pada tiap tabel tersebut. Sebelum melakukan percobaan query banyak tabel dibawah ini terlihat gambar relasi tabelnya sebagai panduan untuk menuliskan syntax sql.



Cara terbaik untuk memahami query banyak tabel adalah dengan mencoba dua tabel terlebih dahulu yang mempunyai relasi, misalnya query tabel Barang dengan tabel Detail atau tabel Pelanggan dengan tabel Penjualan atau tabel Detail dengan tabel Penjualan. Setelah itu dilanjutkan dengan query tiga tabel yang telah mempunyai relasi, misalnya tabel Pelanggan, Penjualan dan Detail. Dan terakhir adalah melakukan query ke semua tabel.

Aturan dasar penulisan query banyak tabel adalah sebagai berikut:

1. Pemisah antara nama tabel yang satu dengan nama tabel yang lain adalah

   KOMA. Contoh : Select * From Barang,Detail,Penjualan,Pelanggan

2. Pengambilan nama field dari suatu tabel yang dituliskan di bagian depan adalah dengan tanda TITIK kemudian nama field.

   Contoh 1 : Select Barang.Namabrg, Barang.Harga, Barang.Jumlah From Barang

Contoh 2 : Select Barang.Namabrg,Barang.Harga,Detail.Jmljual From Barang,Detail

3. Setelah pernyataan select tidak boleh secara langsung menuliskan nama field yang berada pada dua tabel yang berbeda atau field yang menjadi kunci relasi.

   Contoh yang salah :

   Select Kodebrg, Namabrg, Jmljual From Barang,Detail

   Syntax ini akan menampilkan pesan error karena field kodebrg berada di dua tabel yang berbeda yaitu tabel barang dan tabel detail dan kodebrg menjadi field yang direlasikan.

   Adapun syntax yang benar adalah dengan menyebutkan nama tabel didepannya kemudian diikuti dengan tanda TITIK kemudian menuliskan nama fieldnya.

   Contoh yang benar :

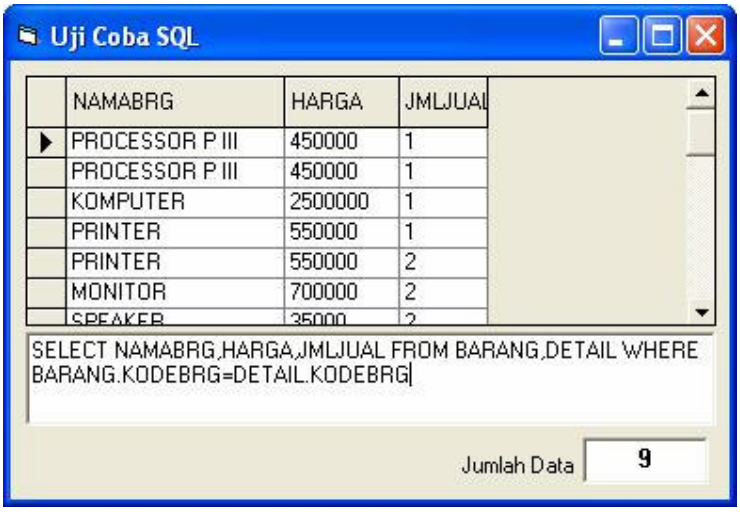
   Select Barang.Kodebrg, Barang.Namabrg, Detail.Kodebrg, Detail.Jmljual From Barang,Detail

4. Query yang optimal sebaiknya dengan menuliskan relasi pada masing-masing tabel setelah klausa where.

   Contoh Query Dua Tabel

   Select Namabrg, Harga, Jmljual From Barang, Detail
   Where Barang.Kodebrg=Detail.Kodebrg

   Hasil

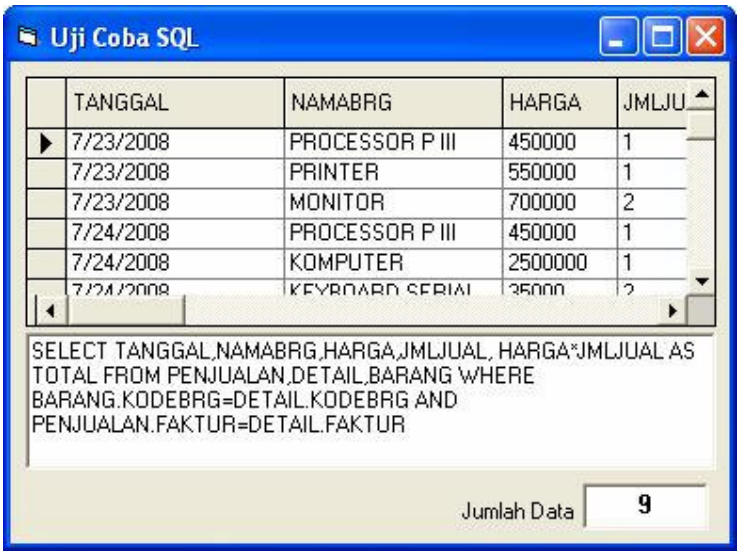


Perhatikan syntax setelah where, tertulis BARANG.KODEBRG=DETAIL.KODEBRG. hal ini dilakukan karena kedua field tersebut menjadi penghubung antara kedua tabel yang disebutkan dalam syntax sql. Hal yang sama sebaiknya anda lakukan pada query tiga tabel atau lebih.

Contoh Query Tiga Tabel:

Select Tanggal,Namabrg,Harga,Jmljual, Harga*Jmljual As Total From Penjualan,Detail,Barang Where Barang.Kodebrg=Detail.Kodebrg And Penjualan.Faktur=Detail.Faktur

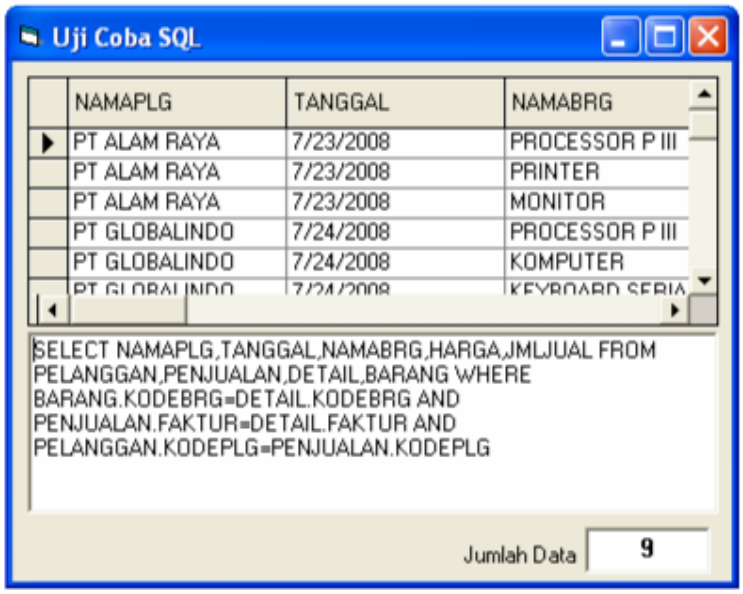
Hasil



Contoh Query Empat Tabel:

Select Namaplg,Tanggal,Namabrg,Harga,Jmljual From Pelanggan,Penjualan,Detail,Barang Where Barang.Kodebrg=Detail.Kodebrg And Penjualan.Faktur=Detail.Faktur And Pelanggan.Kodeplg=Penjualan.Kodeplg

Hasil



Data yang dihasilkan dari beberapa contoh query di atas tidak memberikan nilai informasi, karena datanya masih banyak yang redudan. Hal ini karena query tidak diikat dengan kriteria tertentu.

Untuk memulai query banyak tabel, terlebih dahulu harus dipertimbangkan informasi apa yang diinginkan. Bentuk informasi yang diperlukan biasanya berupa faktur yang berisi nama barang, harga, jumlah jual, dan total. Selanjutnya keempat data tersebut "diikat" dengan kriteria tertentu yaitu faktur dengan tanggal tertentu, faktur dengan nama pelanggan tertentu, faktur dengan nomor tertentu dan sejenisnya.

Setelah pertimbangan tersebut dilakukan maka cobalah untuk melatih menulis beberapa syntax sql di bawah ini.

Select * From Barang,Detail

Fungsi : menampilkan seluruh kolom di tabel barang dan tabel detail.

Dengan cara seperti ini akan terbentuk banyak kolom, tetapi data tidak memiliki nilai informasi apapun.

Select Namabrg,Harga,Jmljual From Barang,Detail Where Barang.Kodebrg=Detail.Kodebrg

Fungsi : menampilkan nama barang dari tabel barang, harga dari tabel barang, jmljual dari tabel detail dimana kode barang di tabel barang sama dengan kode barang di tabel detail

Data yang sama dapat pula dihasilkan dengan menuliskan syntax di bawah ini.

```
Select Barang.Namabrg,Barang.Harga,Detail.Jmljual From Barang,Detail
Where Barang.Kodebrg=Detail.Kodebrg
```

Data yang sama juga dapat dilakukan dengan menuliskan syntax di bawah ini.

```
Select B.Namabrg,B.Harga,D.Jmljual From BARANG AS B,DETAIL AS D
Where B.Kodebrg=D.Kodebrg
```

```
Select Dinstinct Tanggal,Namabrg,Harga,Jmljual, Harga*Jmljual As Total From
Penjualan,Detail Inner Join Barang On Detail.Kodebrg=Barang.Kodebrg
```

Fungsi : menampilkan kolom Tanggal,Harga,JmlJual dan Total (sebagai kolom baru) dari tabel Penjualan yang digabungkan ke dalam tabel Barang dimana kode Barang di tabel Penjualan sama dengan kode Barang di tabel Barang.

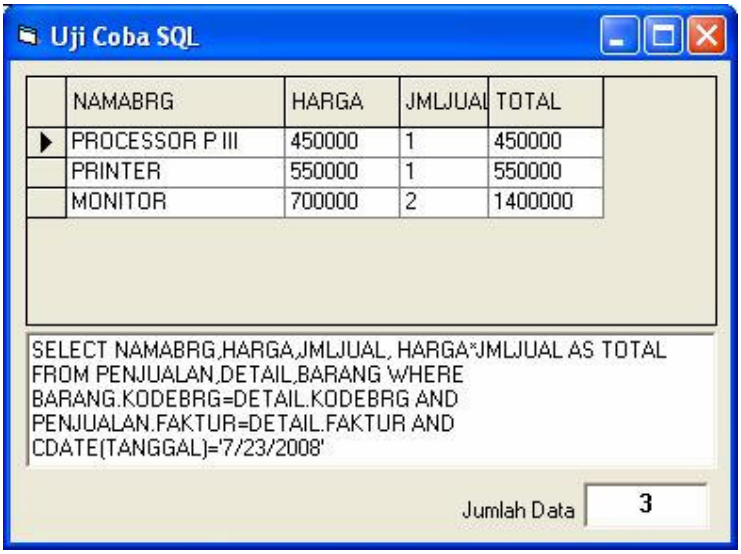
Hasil :

Sekarang cobalah untuk menampilkan hasil query dengan kriteria tertentu, misalnya dengan kriteria tanggal, pelanggan, nomor faktur dan sejenisnya.

```
Select Namabrg,Harga,Jmljual, Harga*Jmljual As Total From Penjualan,Detail,Barang
Where Barang.Kodebrg=Detail.Kodebrg And Penjualan.Faktur=Detail.Faktur And
Cdate(Tanggal)='7/23/2008'
```

Fungsi : tampilkan nama barang, harga, jmljual, total (sebagai field baru hasil perkalian antara harga dengan jmljual) dari tabel penjualan, detail, barang dimana kode barang di tabel barang sama dengan kode barang di tabel detail dan faktur di tabel penjualan sama dengan faktur di tabel detail dan tanggal di tabel penjualannya adalah tanggal 23 juli 2008

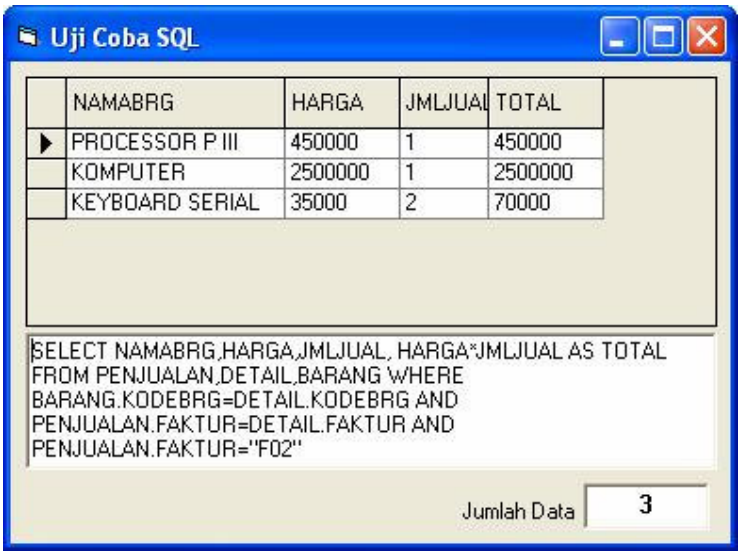
Hasil:



Select Namabrg,Harga,Jmljual, Harga*Jmljual As Total From Penjualan,Detail,Barang Where Barang.Kodebrg=Detail.Kodebrg And Penjualan.Faktur=Detail.Faktur And Penjualan.Faktur="F02"

Fungsi : tampilkan nama barang, harga, jmljual dan total (sebagai field baru hasil perkalian antara harga dengan jmljual) dari tabel penjualan, detail, barang dimana kode barang di tabel barang sama dengan kode barang di tabel detail dan faktur di tabel penjualan sama dengan faktur di tabel detail dan faktur di tabel penjualannya adalah F01

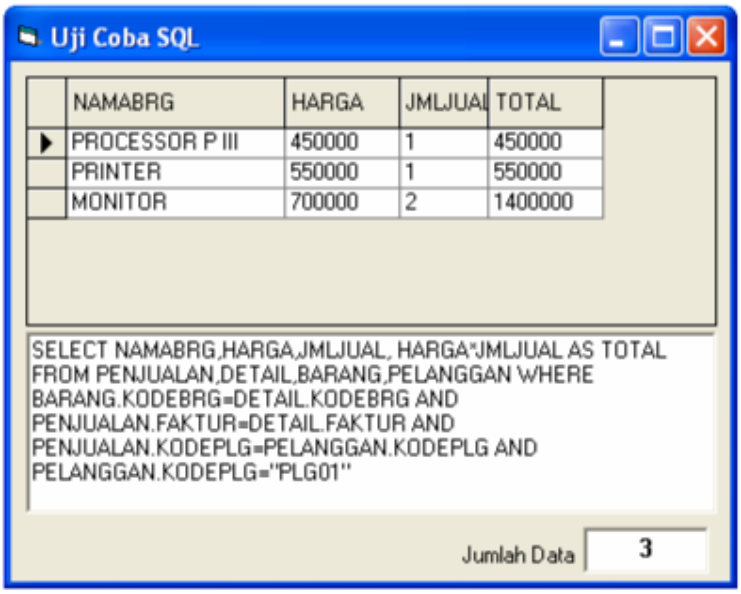
Hasil



Select Namabrg,Harga,Jmljual, Harga*Jmljual As Total From Penjualan,Detail,Barang,Pelanggan Where Barang.Kodebrg=Detail.Kodebrg And Penjualan.Faktur=Detail.Faktur And Penjualan.Kodeplg=Pelanggan.Kodeplg And Pelanggan.Kodeplg="Plg01"

Fungsi : tampilkan nama barang, harga, jmljual, total (sebagai field baru hasil perkalian antara harga dengan jmljual) dari tabel penjualan, detail, barang dan tabel pelanggan dimana kode barang di tabel barang sama dengan kode barang di tabel detail dan faktur di tabel penjualan sama dengan faktur di tabel detail dan kode pelanggan di tabel penjualan sama dengan kode pelanggan di tabel pelanggan dan kode pelangannya adalah "plg01"

Hasil :



Itulah beberapa contoh dasar query banyak tabel, selanjutnya anda tinggal menambahkan kriteria lain yang diperlukan dan dapat pula query dibuat secara spesifik dengan menambahkan klausa order by, in, not, like, wildcard dan sejenisnya.