

Database Link Pada PostgreSQL

Mudafiq Riyan Pratama

van_der_dhaf32@yahoo.co.id

http://dhafiq-san.blogspot.com/

Lisensi Dokumen:

Copyright © 2003-2007 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Pendahuluan

Konsep utama dari sistem database terdistribusi adalah **database link**. **Database Link** adalah suatu koneksi antara dua database fisik yang memungkinkan client untuk mengakses datanya sebagai satu database logic. Database link merupakan sebuah pointer yang didefinisikan sebagai satu-arah jalur komunikasi dari server database kepada server database lainnya. Koneksi database link memungkinkan user-user lokal untuk mengakses data pada database remote. Agar koneksi ini berhasil, setiap database pada sistem distribusi harus memiliki global database name yang unik pada domain jaringan. Dan nama domain tersebut didaftarkan berdasarkan IP dari host.

Bagaimana cara penggunaan dblink pada postgresQL?? Silahkan baca artikel ini dengan teliti.....

Persiapan:

- Database PostgreSQL 8.3 Enterprise
- 2 host komputer
- Secangkir kopi panas ^_^

Berikut detail dari komputer yang bertindak sebagai **server**:

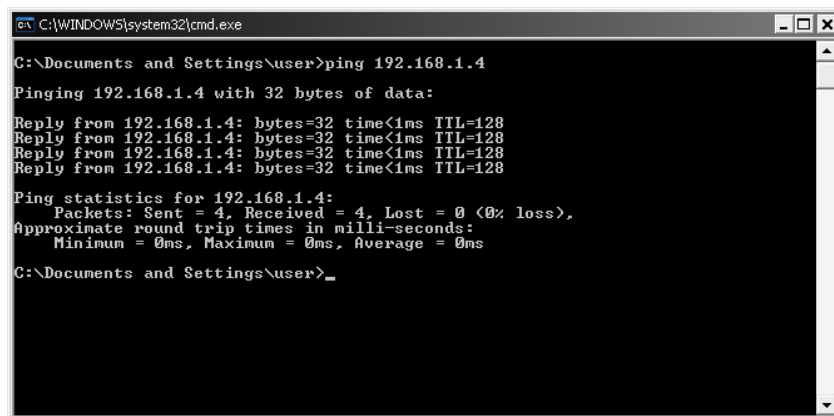
- IP : 192.168.1.4
- User : postgres
- Port : 5432
- Password : keuangan
- Database name : sagara
- Table name : pegawai

Berikut detail dari komputer yang bertindak sebagai **client**:

- IP : 192.168.1.9

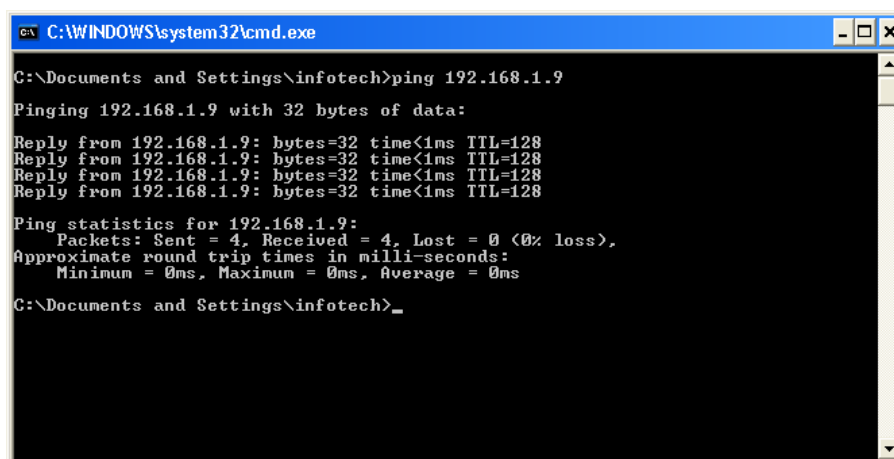
1. Tes koneksi antar host

Untuk langkah pertama, pastikan bahwa antara kedua host (komputer) telah terkoneksi.



```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\user>ping 192.168.1.4
Pinging 192.168.1.4 with 32 bytes of data:
Reply from 192.168.1.4: bytes=32 time<1ms TTL=128
Reply from 192.168.1.4: bytes=32 time<1ms TTL=128
Reply from 192.168.1.4: bytes=32 time<1ms TTL=128
Reply from 192.168.1.4: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.1.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\Documents and Settings\user>_
```

Gambar 1: hasil ping di komputer client ke IP server (192.168.1.4)



```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\infotech>ping 192.168.1.9
Pinging 192.168.1.9 with 32 bytes of data:
Reply from 192.168.1.9: bytes=32 time<1ms TTL=128
Reply from 192.168.1.9: bytes=32 time<1ms TTL=128
Reply from 192.168.1.9: bytes=32 time<1ms TTL=128
Reply from 192.168.1.9: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.1.9:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\Documents and Settings\infotech>_
```

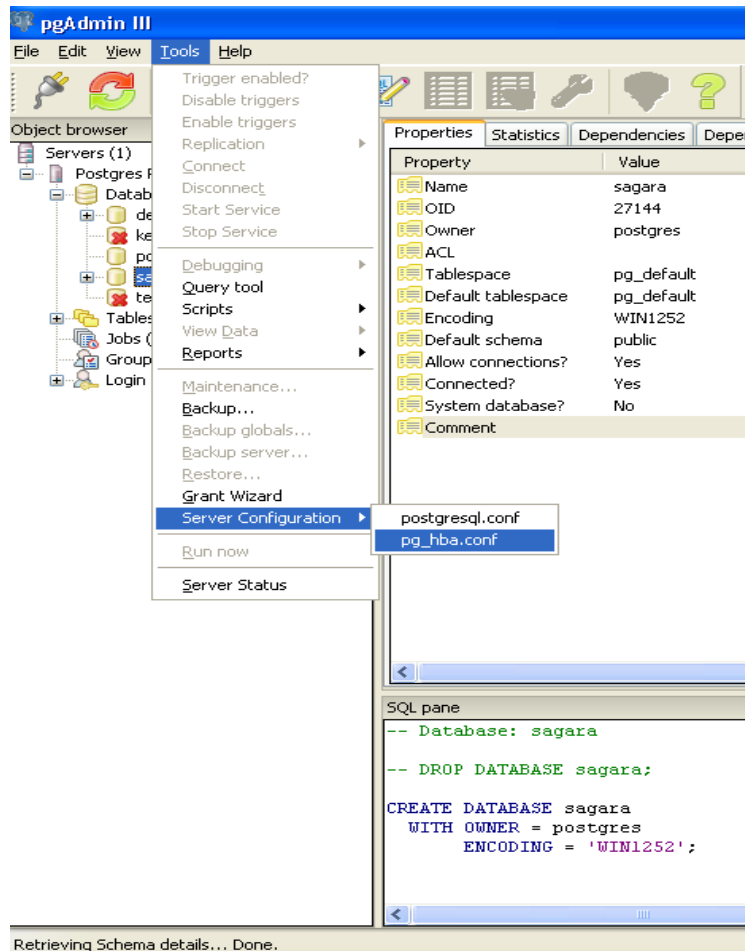
Gambar 2: hasil ping di komputer server ke IP client (192.168.1.9)

2. Register IP client

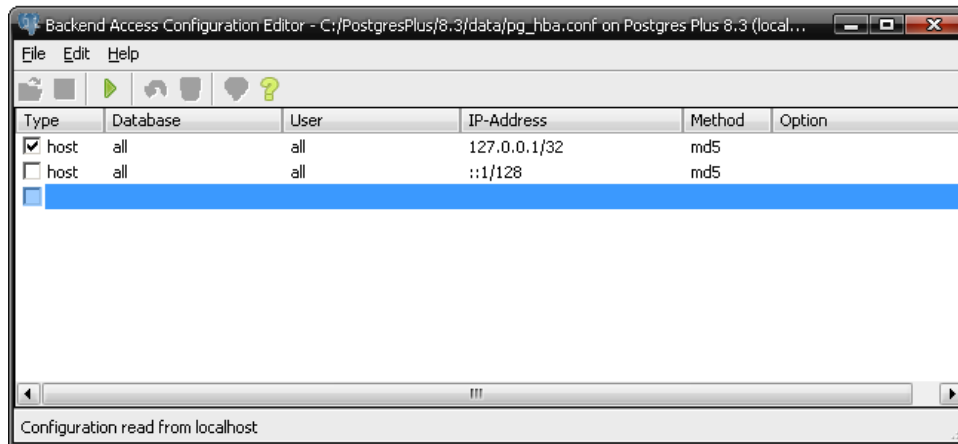
Mendaftarkan IP client di server postgre melalui **pg_hba.conf**. Dengan tujuan bahwa hanya IP client tersebut yang bisa mengakses database server.

Lakukan setting dengan cara:

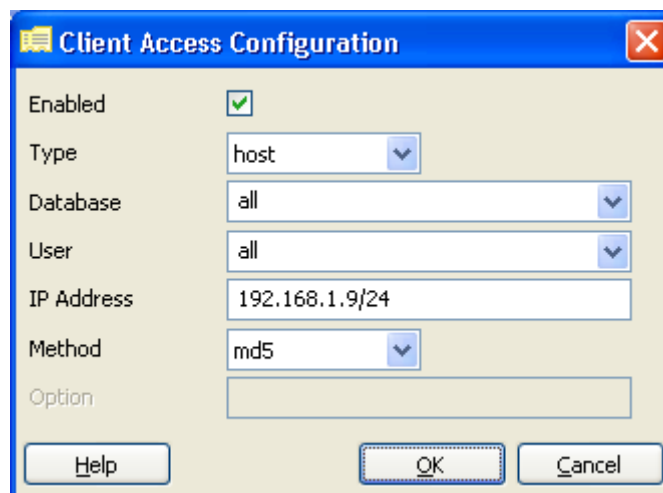
- Pada menu postgresQL yang ada di komputer server, pilih menu **Tools – Server Configuration – pg_hba.conf**



- Pada form pg_hba.conf, double klik sesuai gambar dibawah pada area kosong paling bawah sendiri



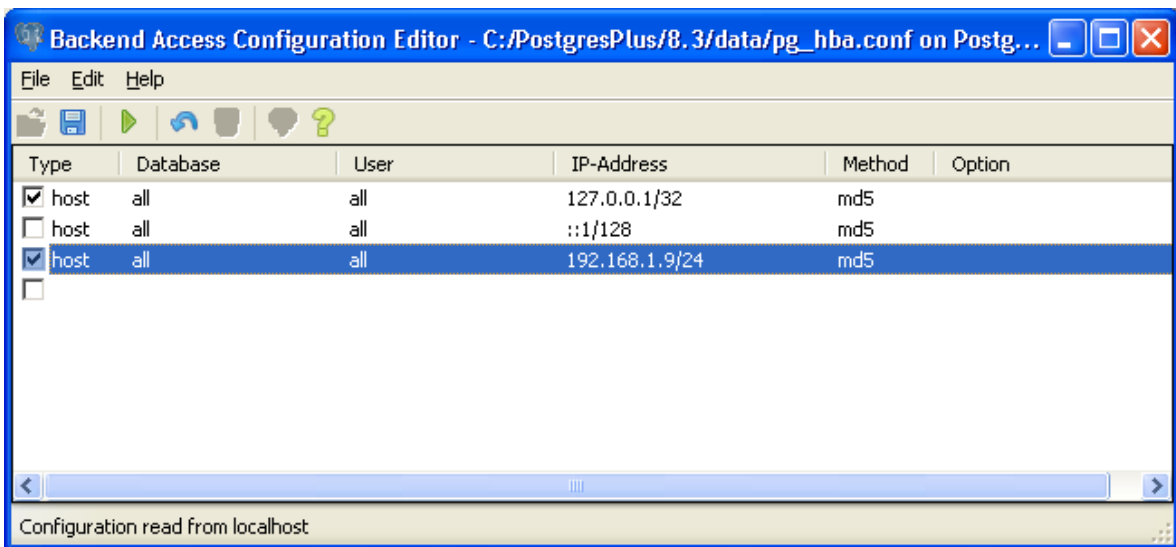
- Kemudian isikan seperti gambar dibawah ini. Setelah itu, tekan tombol OK



Mendaftarkan IP client pada postgre server, dengan tujuan agar hanya IP tersebut yang dapat mengakses data server.

- Type: host
- Database: all (memberikan hak akses pada client untuk semua database yang ada di server)
- User: all (untuk semua user dari database server dapat diakses oleh client)
- IP Address: 192.168.1.9/24 (adalah IP client yang dapat mengakses data server dengan subnet mask /24 atau sama dengan 255.255.255.0)
- Method: md5 (kami memakai method md5 untuk enkripsi data)

- Dan hasilnya adalah sebagai berikut. Setelah itu, tekan tombol icon save untuk menyimpan konfigurasi IP client.

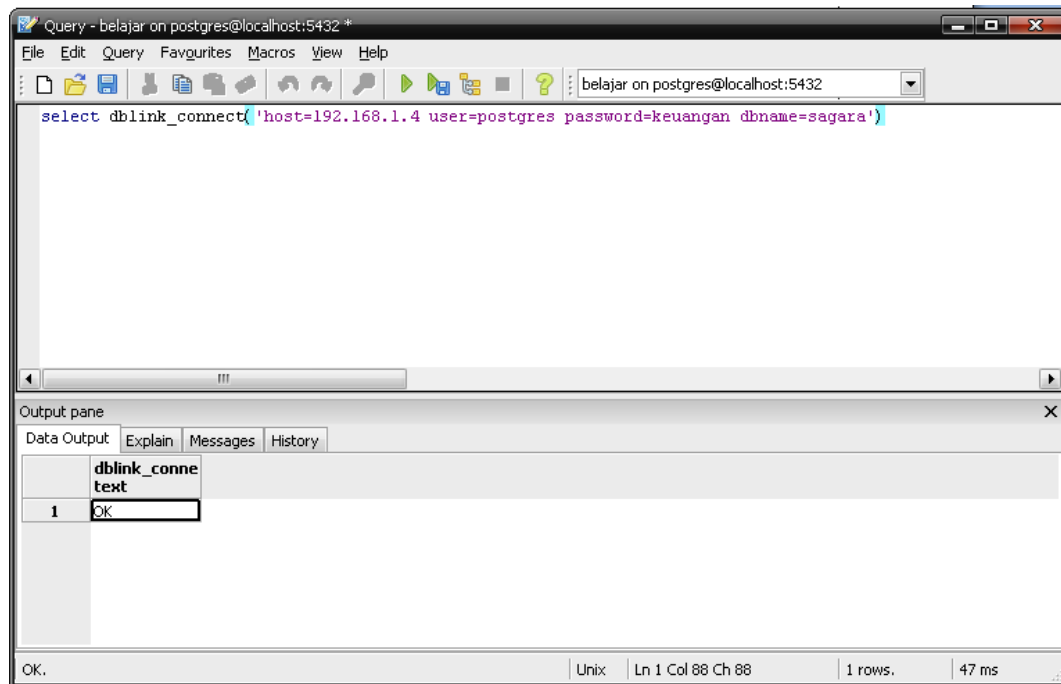


Tujuan dari konfigurasi di atas adalah untuk mendaftarkan IP mana saja yang bisa mengakses database server. Akan tetapi, jika semua IP client bisa mengakses database server, maka yang didaftarkan adalah IP dari server itu sendiri. Sehingga tidak ada security lagi jika IP server itu sendiri yang didaftarkan di postgresSQL milik server, karena semua host yang terkoneksi dengan server dapat mengakses database server.

3. Tes koneksi database

Tes koneksi melalui postgre client menggunakan script berikut:

```
select dblink_connect('host=192.168.1.4 user=postgres password=keuangan  
dbname=sagara')
```

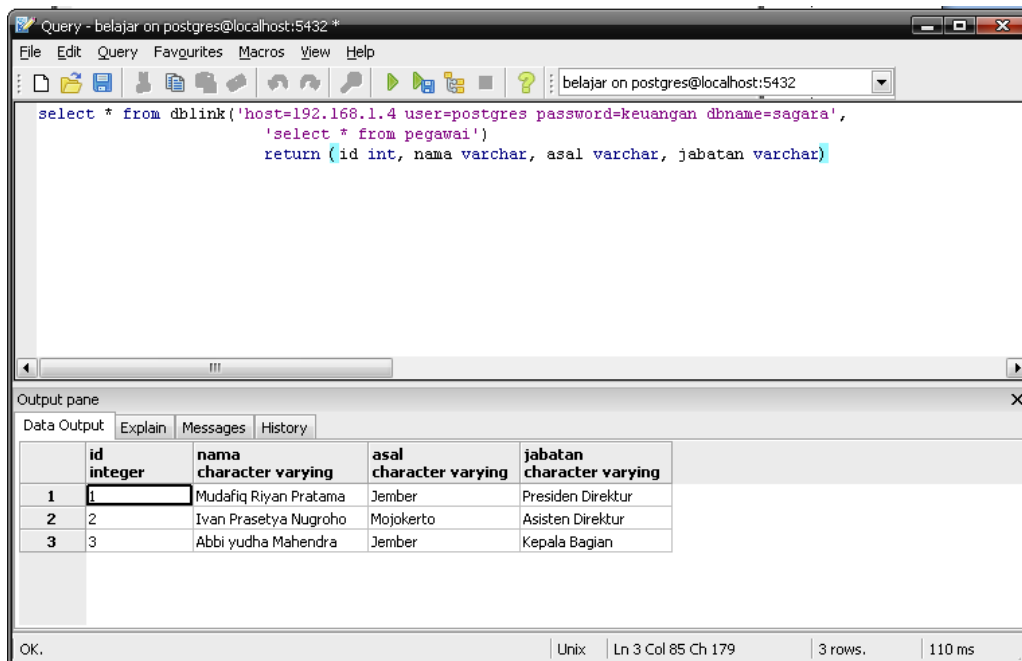


Pada keterangan di atas, dinyatakan bahwa **host=192.168.1.4** dengan **user=postgres** dan **password=keuangan** dan **dbname=sagara** berhasil terkoneksi.

4. Melihat isi data

Melihat isi database yang ada di postgresQL server. Menggunakan script SQL berikut:

```
select * from dblink('host=192.168.1.4 user=postgres password=keuangan dbname=sagara',  
'select * from pegawai')  
return (id int, nama varchar, asal varchar, jabatan varchar)
```

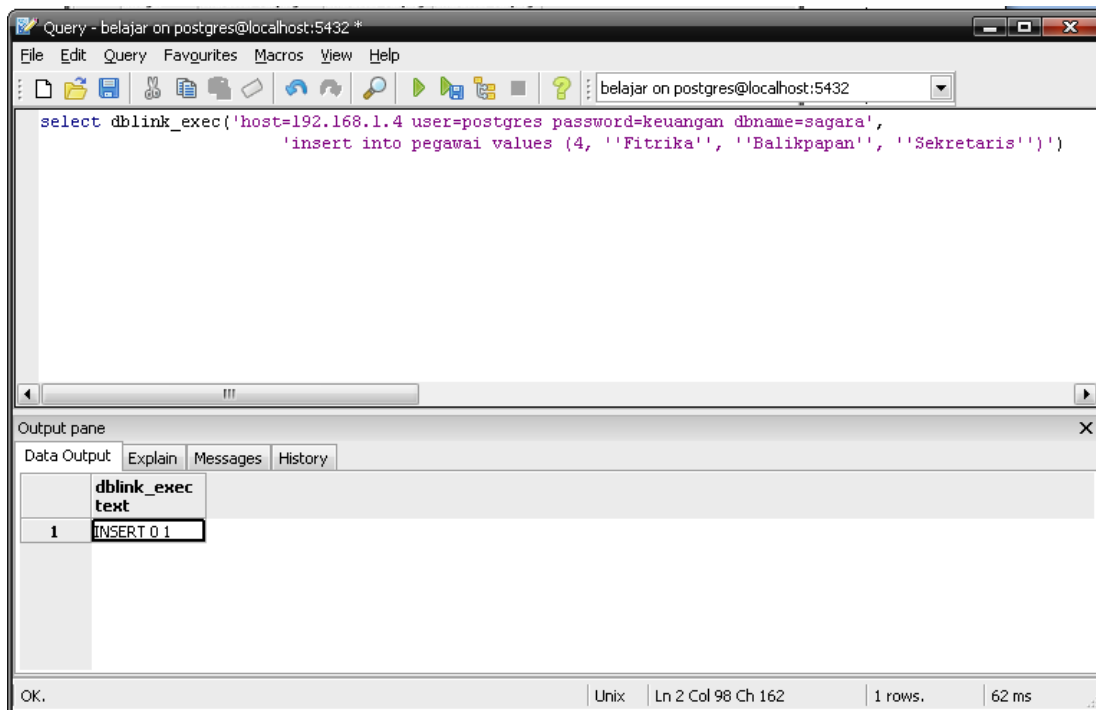


Data pegawai yang ada pada database server dapat kita lihat isinya melalui client.

5. Remote insert data

Memasukkan data melalui client. Menggunakan query berikut:

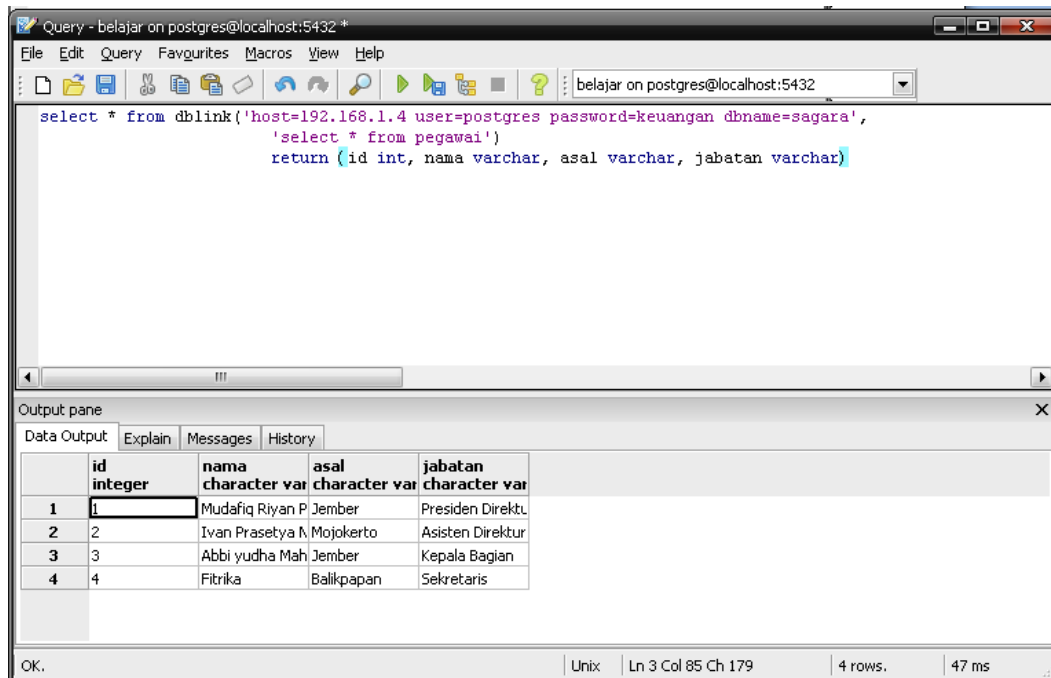
```
select dblink_exec('host=192.168.1.4 user=postgres password=keuangan  
dbname=sagara',  
'insert into pegawai values (4, "Fitrika", "Balikpapan", "Sekretaris")')
```



Kita berhasil memasukkan data baru pada database server melalui client.

Kita periksa data diserver apakah sudah berhasil memasukkan data baru atau belum?

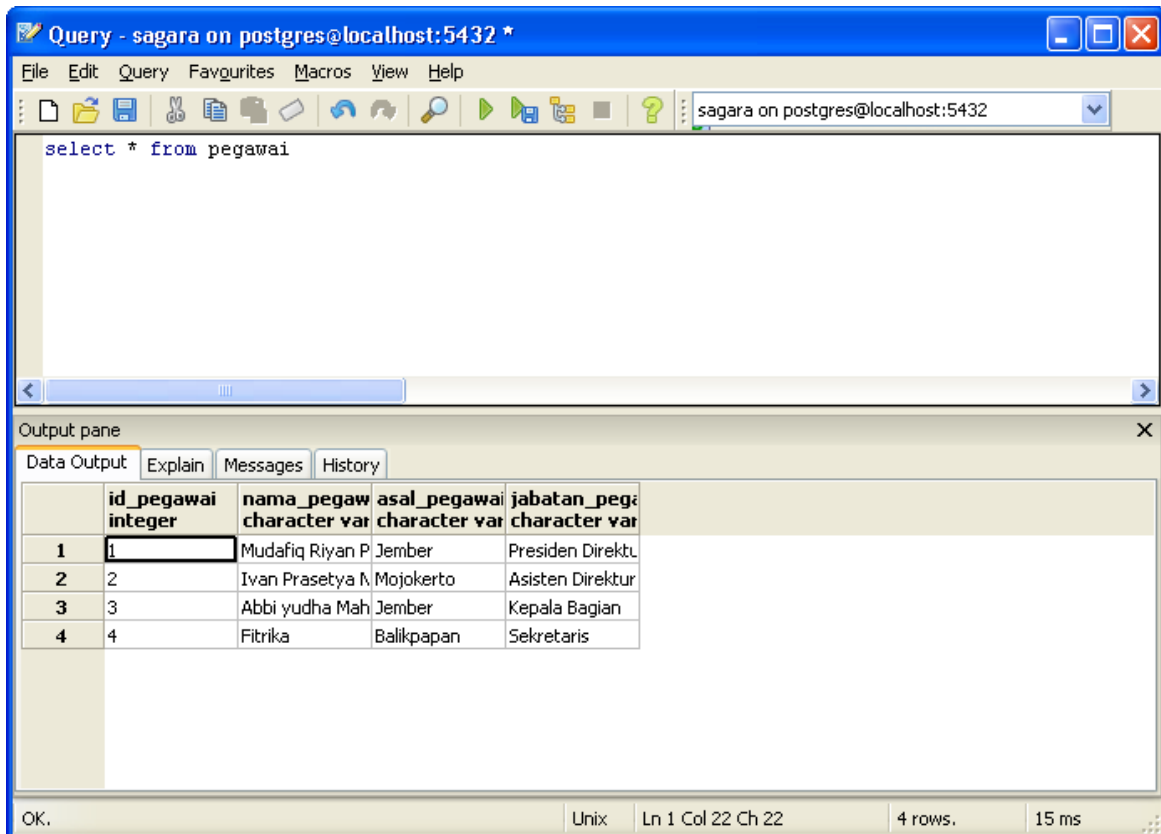
Diperiksa melalui client:



Setelah perintah select, terpadat fungsi return yang berfungsi untuk mengembalikan attribute dari kolom-kolom yang ada pada tabel pegawai, yang mana:

- **id_pegawai** di return dengan **id** yang attributenya integer
- **nama_pegawai** di return dengan **nama** yang attributenya varchar
- **asal_pegawai** di return dengan **asal** yang attributenya varchar
- **jabatan_pegawai** di return dengan **jabatan** yang attributenya varchar

Diperiksa melalui server:



The screenshot shows a PostgreSQL query tool window titled "Query - sagara on postgres@localhost:5432 *". The query editor contains the SQL statement: `select * from pegawai`. The output pane shows the results of the query in a table format. The table has four columns: `id_pegawai` (integer), `nama_pegawai` (character var), `asal_pegawai` (character var), and `jabatan_pegawai` (character var). The results are as follows:

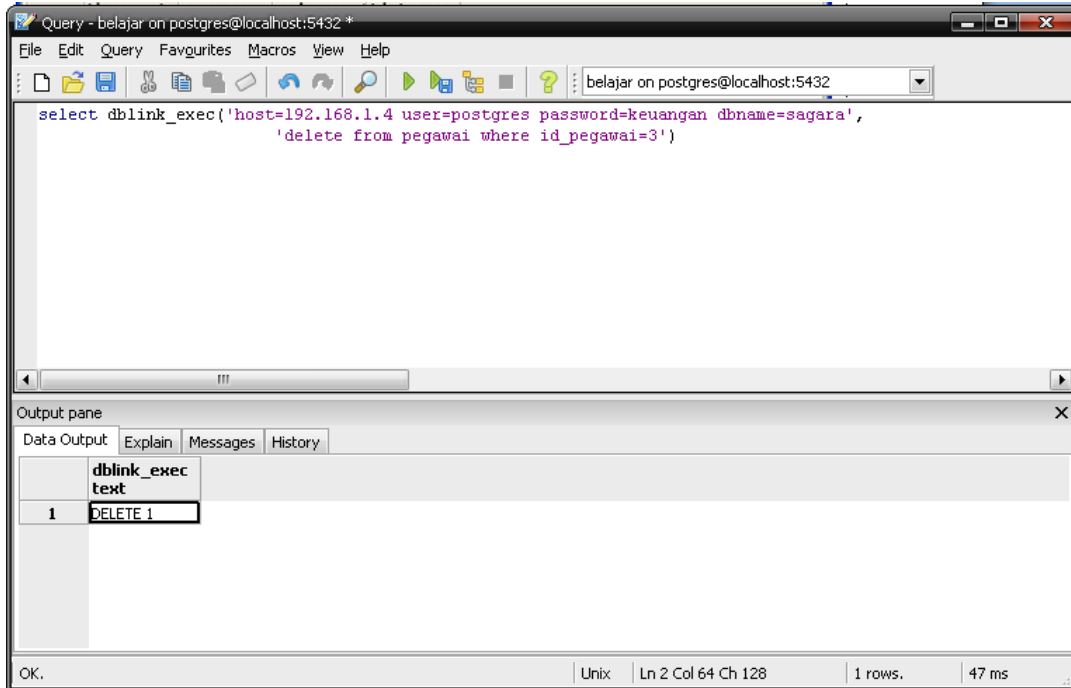
	<code>id_pegawai</code> integer	<code>nama_pegawai</code> character var	<code>asal_pegawai</code> character var	<code>jabatan_pegawai</code> character var
1	1	Mudafiq Riyan P.	Jember	Presiden Direktur
2	2	Ivan Prasetya M.	Mojokerto	Asisten Direktur
3	3	Abbi yudha Mah	Jember	Kepala Bagian
4	4	Fitrika	Balikpapan	Sekretaris

The status bar at the bottom indicates "OK.", "Unix", "Ln 1 Col 22 Ch 22", "4 rows.", and "15 ms".

Data berhasil masuk ke database server

6. Remote delete data

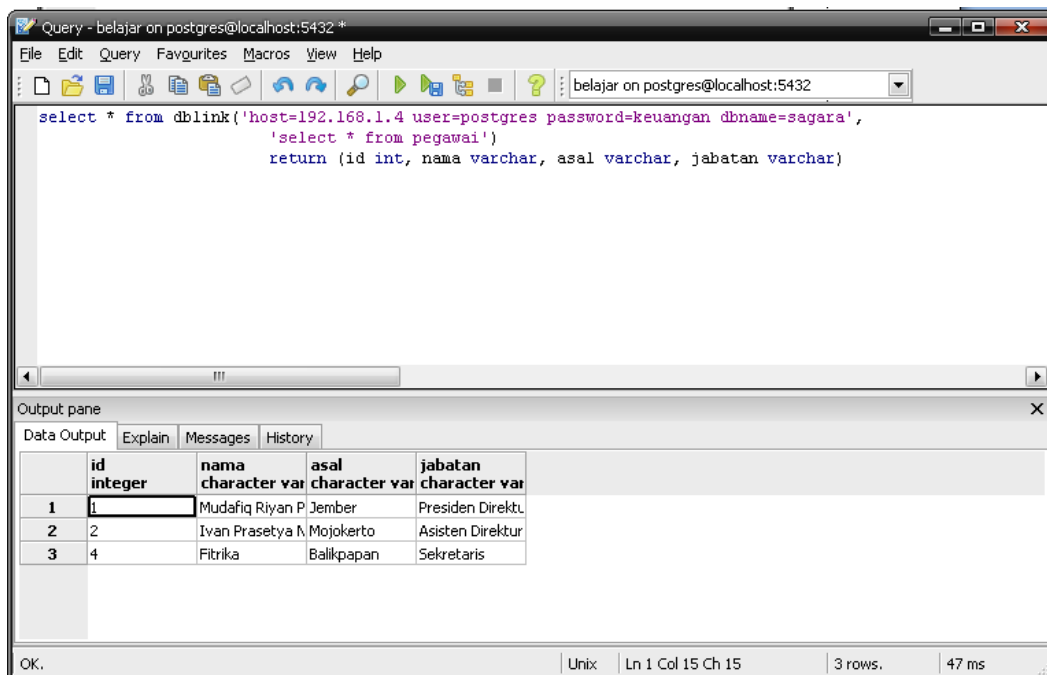
```
select dblink_exec('host=192.168.1.4 user=postgres password=keuangan  
dbname=sagara', 'delete from pegawai where id_pegawai=3')
```



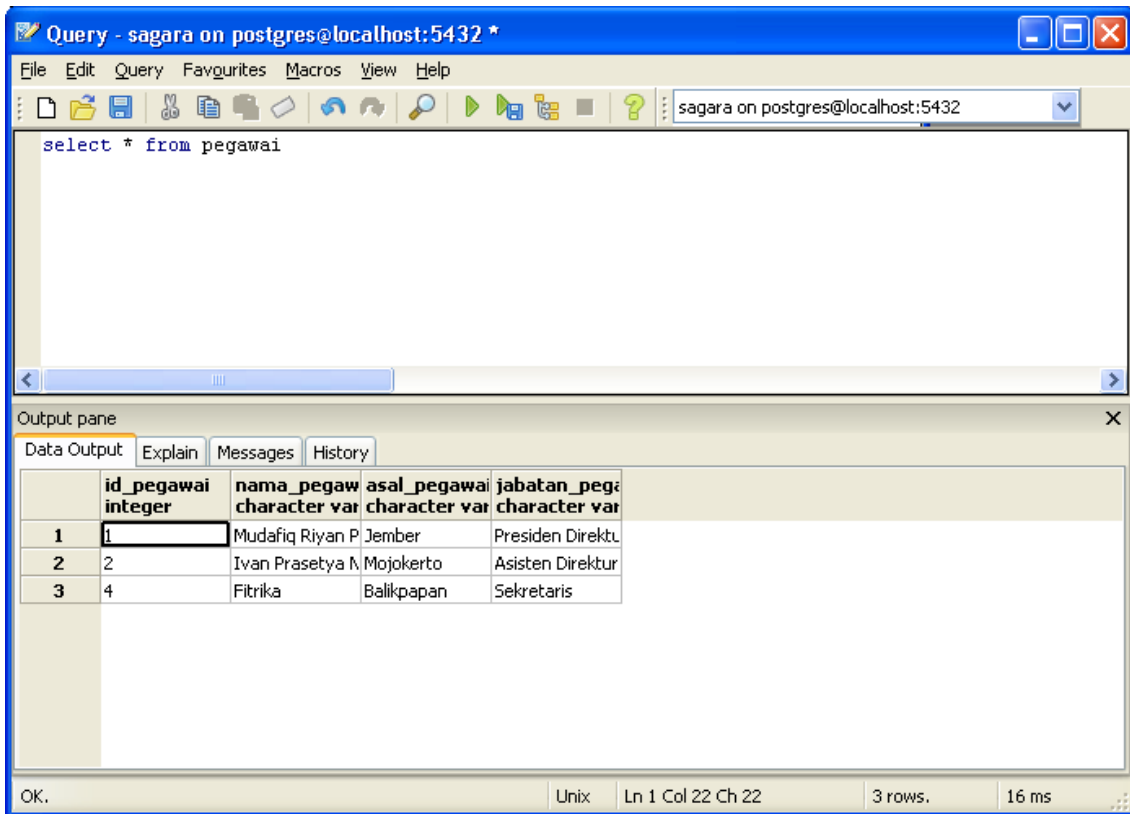
Data berhasil di delete melalui client

Kita periksa apakah udah terhapus apa belum?

Pengecekan melalui client:



Pengecekan melalui server:



The screenshot shows a PostgreSQL query tool window titled "Query - sagara on postgres@localhost:5432". The query editor contains the SQL statement: `select * from pegawai`. The output pane displays the results of the query in a table format. The table has four columns: `id_pegawai` (integer), `nama_pegaw` (character var), `asal_pegawai` (character var), and `jabatan_peg` (character var). The results are as follows:

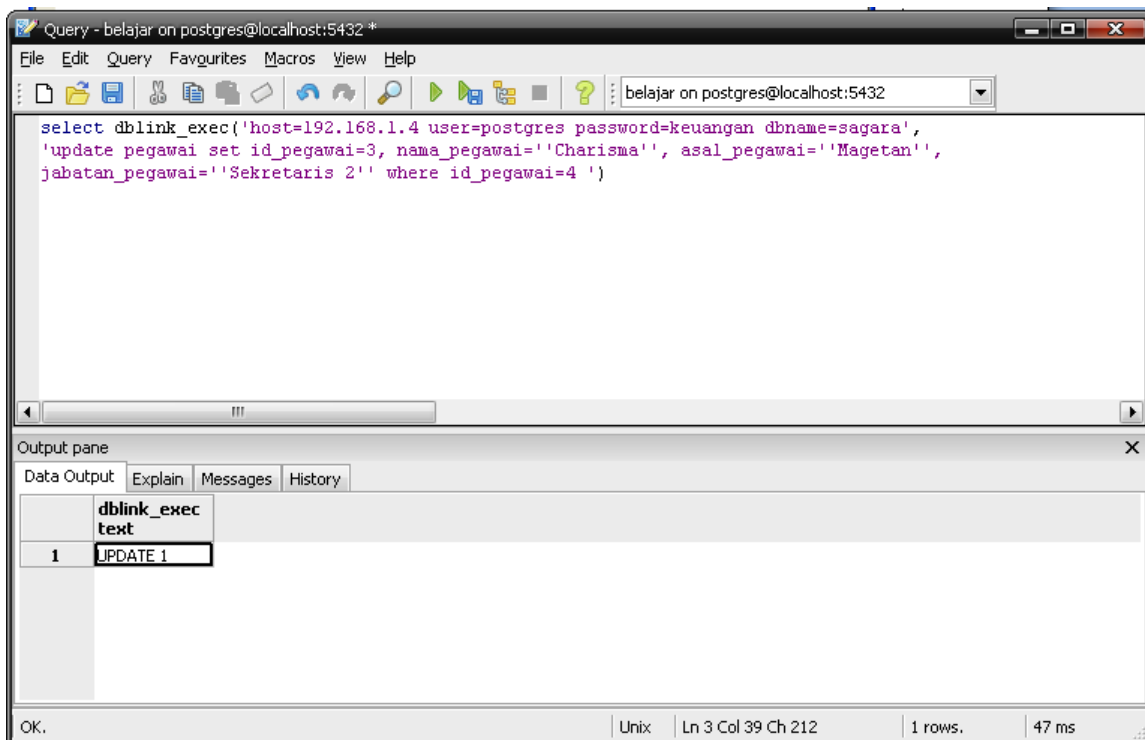
	<code>id_pegawai</code> integer	<code>nama_pegaw</code> character var	<code>asal_pegawai</code> character var	<code>jabatan_peg</code> character var
1	1	Mudafiq Riyan P	Jember	Presiden Direktu
2	2	Ivan Prasetya N	Mojokerto	Asisten Direktur
3	4	Fitrika	Balikpapan	Sekretaris

The status bar at the bottom indicates "OK.", "Unix", "Ln 1 Col 22 Ch 22", "3 rows.", and "16 ms".

7. Remote update data

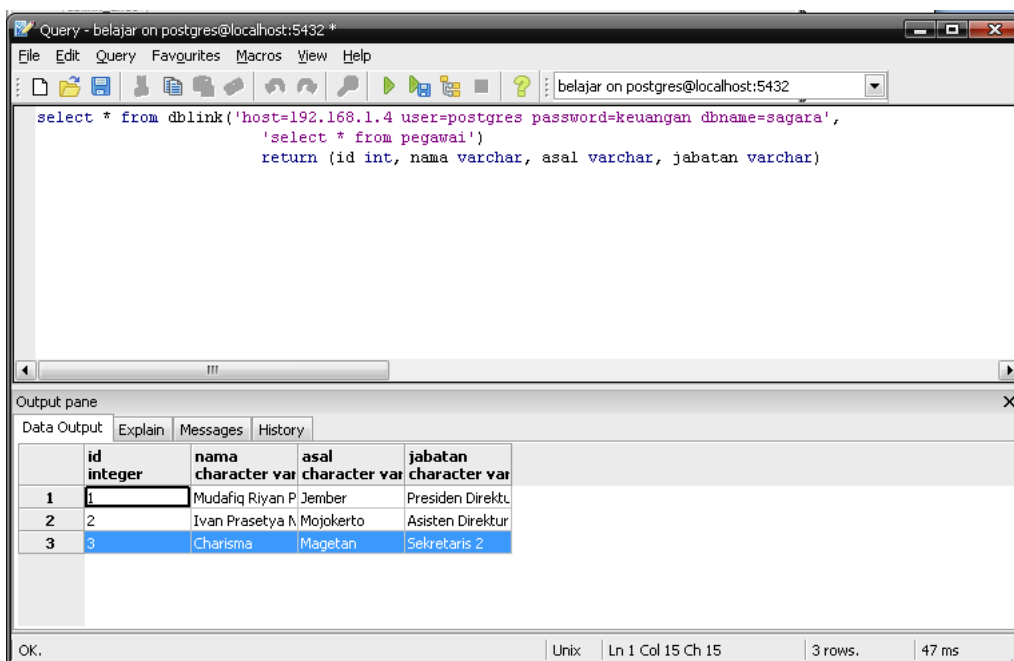
```
select dblink_exec('host=192.168.1.4 user=postgres password=keuangan  
dbname=sagara',  
'update pegawai set id_pegawai=3, nama_pegawai="Charisma",  
asal_pegawai="Magetan",  
jabatan_pegawai="Sekretaris 2" where id_pegawai=4 ')
```

Periksa data melalui client:

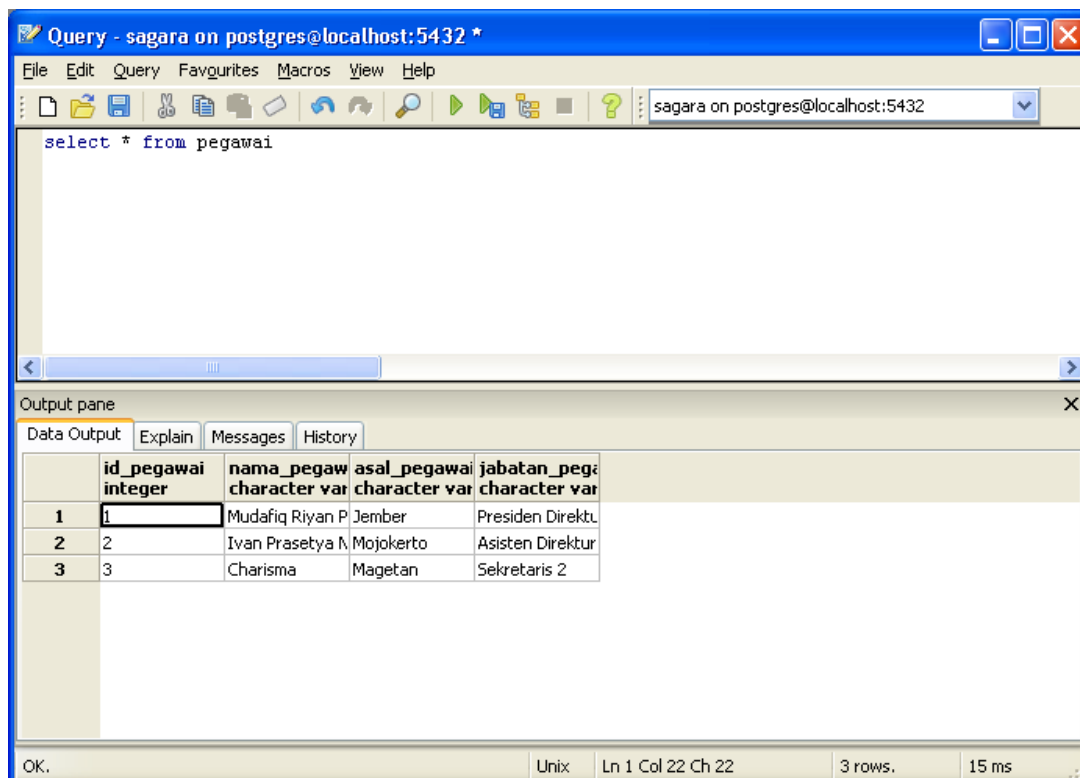


Proses meng-update data pada tabel pegawai yang memiliki id_pegawai=4. Data berhasil di update melalui client. Kemudian kita lakukan pengecekan:

Periksa data melalui client:



Periksa data langsung dari server:



=0=0=0=0=0=0=0=0= Selamat Mencoba =0=0=0=0=0=0=0=0=

Biografi Penulis



Mudafiq Riyan Pratama. Lahir di Jember pada tanggal 9 Mei 1989. Kediaman di Jember. Memulai pendidikan TK dan SD di Jenggawah. Kemudian menempuh SMP di SMPN 6 Jember yang kemudian dilanjutkan ke SMAN 2 Jember. Dan saat ini sedang menempuh kuliah S1 jurusan Teknik Informatika di Universitas Muhammadiyah Malang angkatan 2007.

Didunia maya, penulis lebih sering memakai nama **Dhafiq Sagara.**

YM : van_der_dhaf32

FB : ray_dafier@yahoo.co.id