**­**

Replication

Postgresql 9.4

Oleh: Andi Almafhum

2023

Dengan bdr

**DAFTAR ISI**

REPLIKASI DATABASE 2

INSTALL DAN KONFIGURASI BDR DAN POSTGRESQL 9.4 3

A. Install dan Konfigurasi pada Server Node 1 3

a. Install dan konfigurasi Postgresql 9.4 3

b. Pembuatan dan konfigurasi cluster Node 1 5

c. Install dan konfigurasi bdr 6

B. Install dan Konfigurasi pada Server Node 2 7

a. Install dan konfigurasi Postgresql 9.4 7

b. Pembuatan dan konfigurasi cluster Node 2 9

c. Install dan konfigurasi bdr 10

CLONE DARI SERVER NODE 1 KE SERVER NODE 2 11

A. Konfigurasi Server Node 1 11

B. Konfigurasi Server Node 2 11

PEMBUATAN SCRIPT UNTUK PENGUJIAN 13

**REPLIKASI DATABASE**

Replikasi adalah proses membuat dan menjaga salinan data dari satu sumber data (biasanya database) ke sumber data lainnya dengan tujuan untuk memastikan ketersediaan data, meningkatkan kinerja, dan menyediakan kemampuan pemulihan bencana. Pada simulasi kali ini kita akan menggunakan alamat ip seperti di bawah ini.

|  |  |  |
| --- | --- | --- |
| Server | IP | Nama node |
| Server Node 1 | 10.1.255.211 | Node 1 |
| Server Node 2 | 10.1.255.212 | Node 2 |

**INSTALL DAN KONFIGURASI BDR DAN POSTGRESQL 9.4**

Pada proses replikasi kali ini kita akan memanfaatkan postgresql 9.4 dan bdr pada linux berbasis ubuntu/debian. Berikut merupakan konfigurasinya.

**A. Install dan Konfigurasi pada Server Node 1**

Pada Server Node 1 kita perlu melakukan penginstalan serta konfigurasi pada database Postgresql 9.4, berikut merupakan tata caranya.

**a. Install dan Konfigurasi Postgresql 9.4**

Install Postgreql 9.4

|  |
| --- |
| sudo apt install wget ca-certificates -y  wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -  sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ $(lsb\_release -cs)-pgdg main" >> /etc/apt/sources.list.d/pgdg.list'  sudo apt install postgresql-9.4 postgresql-client-9.4 postgresql-contrib |

**Link suport postgresql supaya dapat di eksekusi**

|  |
| --- |
| **sudo ln -s /usr/lib/postgresql/9.4/bin/\* /usr/bin/** |

Konfigurasi postgresql

|  |
| --- |
| sudo -u postgres psql |

Buat password untuk user postgres.

|  |
| --- |
| ALTER USER postgres WITH PASSWORD 'postgres'; |

konfigurasi **/etc/postgresql/9.4/main/postgresql.conf**

**lakukan perubahan.**

|  |
| --- |
| **listen\_addresses = '\*'**  max\_connections = 100  shared\_preload\_libraries = 'bdr'  wal\_level = 'logical'  track\_commit\_timestamp = on  max\_wal\_senders = 10  max\_replication\_slots = 10  max\_worker\_processes = 10 |

**Setelah itu pergi ke /etc/postgresql/9.4/main/pg\_hba.conf tambahkan.**

|  |
| --- |
| **host replication all 10.1.255.211/32 trust**  **host replication all 10.1.255.212/32 trust**  **host all all 10.1.255.211/32 md5**  **host all all 10.1.255.212/32 md5**  **host all all 0.0.0.0/0 md5** |

**simpan kemudian restart postgresql.**

|  |
| --- |
| **systemctl restart postgresql** |

**b. Pembuatan dan konfigurasi cluster Node 1**

Cek kluster yang tersedia.

|  |
| --- |
| **sudo pg\_lsclusters** |

Membuat cluster dengan nama **‘rep’** yang jalan pada port **‘5433’**.

|  |
| --- |
| **sudo pg\_createcluster 9.4 rep --start --port 5433** |

konfigurasi **/etc/postgresql/9.4/main/postgresql.conf**

**lakukan perubahan.**

|  |
| --- |
| **listen\_addresses = '\*'**  max\_connections = 100  shared\_preload\_libraries = 'bdr'  wal\_level = 'logical'  track\_commit\_timestamp = on  max\_wal\_senders = 10  max\_replication\_slots = 10  max\_worker\_processes = 10 |

**Setelah itu pergi ke /etc/postgresql/9.4/main/pg\_hba.conf tambahkan.**

|  |
| --- |
| **host replication all 10.1.255.211/32 trust**  **host replication all 10.1.255.212/32 trust**  **host postgres all 10.1.255.211/32 md5**  **host postgres all 10.1.255.212/32 md5**  **host all all 0.0.0.0/0 md5** |
|  |

**simpan kemudian stop postgresql cluster tersebut.**

|  |
| --- |
| **sudo pg\_ctlcluster 9.4 rep stop** |

**stop postgresql cluster tersebut.**

|  |
| --- |
| **sudo pg\_ctlcluster 9.4 rep start** |

**Cek status postgresql cluster tersebut.**

|  |
| --- |
| **sudo pg\_ctlcluster 9.4 rep status**  **sudo pg\_lsclusters** |

**c. Install dan konfigurasi bdr**

Install bdr

|  |
| --- |
| **apt install postgresql-bdr-9.4 postgresql-bdr-9.4-bdr-plugin** |

Konfigurasi postgresql

|  |
| --- |
| sudo -u postgres psql -p 5433 |

**Beri password pada user postgres**

|  |
| --- |
| psql -c "ALTER USER postgres WITH PASSWORD 'postgres'; |

**Buat database baru, disini kita beri nama **'rep'****

|  |
| --- |
| CREATEDB rep; |

**Masuk ke dalam database **'rep'****

|  |
| --- |
| \c rep; |

**Inisiasi extension bdr**

|  |
| --- |
| CREATE EXTENSION btree\_gist;  CREATE EXTENSION bdr; |

**A. Install dan Konfigurasi pada Server Node 2**

Pada Server Node 1 kita perlu melakukan penginstalan serta konfigurasi pada database Postgresql 9.4, berikut merupakan tata caranya.

**a. Install dan Konfigurasi Postgresql 9.4**

Install Postgreql 9.4

|  |
| --- |
| sudo apt install wget ca-certificates -y  wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -  sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ $(lsb\_release -cs)-pgdg main" >> /etc/apt/sources.list.d/pgdg.list'  sudo apt install postgresql-9.4 postgresql-client-9.4 postgresql-contrib |

**Link suport postgresql supaya dapat di eksekusi**

|  |
| --- |
| **sudo ln -s /usr/lib/postgresql/9.4/bin/\* /usr/bin/** |

Konfigurasi postgresql

|  |
| --- |
| sudo -u postgres psql |

Buat password untuk user postgres.

|  |
| --- |
| ALTER USER postgres WITH PASSWORD 'postgres'; |

konfigurasi **/etc/postgresql/9.4/main/postgresql.conf**

**lakukan perubahan.**

|  |
| --- |
| **listen\_addresses = '\*'**  shared\_preload\_libraries = 'bdr'  wal\_level = 'logical'  track\_commit\_timestamp = on  max\_connections = 100  max\_wal\_senders = 10  max\_replication\_slots = 10  max\_worker\_processes = 10 |

**Setelah itu pergi ke /etc/postgresql/9.4/main/pg\_hba.conf tambahkan.**

|  |
| --- |
| **host replication all 10.1.255.211/32 trust**  **host replication all 10.1.255.212/32 trust**  **host postgres all 10.1.255.211/32 md5**  **host postgres all 10.1.255.212/32 md5**  **host postgres all 0.0.0.0/0 md5** |
|  |

**simpan kemudian restart postgresql.**

|  |
| --- |
| **systemctl restart postgresql** |

**b. Pembuatan dan konfigurasi cluster Node 2**

Cek kluster yang tersedia.

|  |
| --- |
| **sudo pg\_lsclusters** |

Membuat cluster dengan nama **‘rep’** yang jalan pada port **‘5433’**.

|  |
| --- |
| **sudo pg\_createcluster 9.4 rep --start --port 5433** |

konfigurasi **/etc/postgresql/9.4/main/postgresql.conf**

**lakukan perubahan.**

|  |
| --- |
| **listen\_addresses = '\*'**  max\_connections = 100  shared\_preload\_libraries = 'bdr'  wal\_level = 'logical'  track\_commit\_timestamp = on  max\_wal\_senders = 10  max\_replication\_slots = 10  max\_worker\_processes = 10 |

**Setelah itu pergi ke /etc/postgresql/9.4/main/pg\_hba.conf tambahkan.**

|  |
| --- |
| **host replication all 10.1.255.211/32 trust**  **host replication all 10.1.255.212/32 trust**  **host postgres all 10.1.255.211/32 md5**  **host postgres all 10.1.255.212/32 md5**  **host all all 0.0.0.0/0 md5** |
|  |

**simpan kemudian stop postgresql cluster tersebut.**

|  |
| --- |
| **sudo pg\_ctlcluster 9.4 rep stop** |

**stop postgresql cluster tersebut.**

|  |
| --- |
| **sudo pg\_ctlcluster 9.4 rep start** |

**Cek status postgresql cluster tersebut.**

|  |
| --- |
| **sudo pg\_ctlcluster 9.4 rep status**  **sudo pg\_lsclusters** |

**c. Install dan konfigurasi bdr**

Install bdr

|  |
| --- |
| **apt install postgresql-bdr-9.4 postgresql-bdr-9.4-bdr-plugin** |

Konfigurasi postgresql

|  |
| --- |
| sudo -u postgres psql -p 5433 |

**Beri password pada user postgres**

|  |
| --- |
| psql -c "ALTER USER postgres WITH PASSWORD 'postgres'; |

**Buat database baru, disini kita beri nama **'rep'****

|  |
| --- |
| CREATEDB rep; |

**Masuk ke dalam database **'rep'****

|  |
| --- |
| \c rep; |

**Inisiasi extension bdr**

|  |
| --- |
| CREATE EXTENSION btree\_gist;  CREATE EXTENSION bdr; |

**CLONE DARI SERVER NODE 1 KE SERVER NODE 2**

Proses ini akan melakukan clone dari Server Node 1 ke Server Node 2 secara upstream, sehingga data dapat live diantara dua server tersebut.

**A. Konfigurasi Server Node 1**

**masuk registrasi node-1**

|  |
| --- |
| sudo -u postgres psql rep -p 5433 |

**Lakukan registrasi koordinator**

|  |
| --- |
| \c rep  SELECT bdr.bdr\_group\_create(  local\_node\_name := 'node\_1',  node\_external\_dsn := 'host=10.1.255.211 port=5433 user=postgres dbname=rep password=postgres'  ); |

**Pengecekan koneksifitas.**

|  |
| --- |
| select \* from bdr.bdr\_nodes;  select \* from bdr.bdr\_connections; |

**B. Konfigurasi Server Node 2**

**masuk registrasi node-2**

|  |
| --- |
| sudo -u postgres psql rep -p 5433 |

**Lakukan registrasi koneksifitas**

|  |
| --- |
| \c rep  SELECT bdr.bdr\_group\_join(  local\_node\_name := 'node\_2',  node\_external\_dsn := 'host=10.1.255.212 port=5433 user=bdrsync dbname=rep password=postgres',  join\_using\_dsn := 'host=10.1.255.212 port=5433 user=postgres dbname=rep password=postgres'  ); |

**Pengecekan koneksifitas.**

|  |
| --- |
| select \* from bdr.bdr\_nodes;  select \* from bdr.bdr\_connections; |

Data dapat dinikmati secara upstream pada kedua sisi server, server pertama berguna untuk eksekutor.

**PEMBUATAN SCRIPT UNTUK PENGUJIAN**

Script ini berfungsi untuk melakukan pengujian terhadap sambungan yang ada. Script ini di buat menggunakan php 5.6 yang telah dikonfigurasi apachenya sehingga membuka /var/www yang merupakan penempatan localhost. Pada simulasi kali ini, kita akan menempatkan script ini pada /var/www Berikut merupakan tahapannya.

Konfigurasi postgresql

|  |
| --- |
| sudo -u postgres psql rep -p 5433 |

Buat password untuk user postgres.

|  |
| --- |
| ALTER USER postgres WITH PASSWORD ‘postgres’; |

Buat table

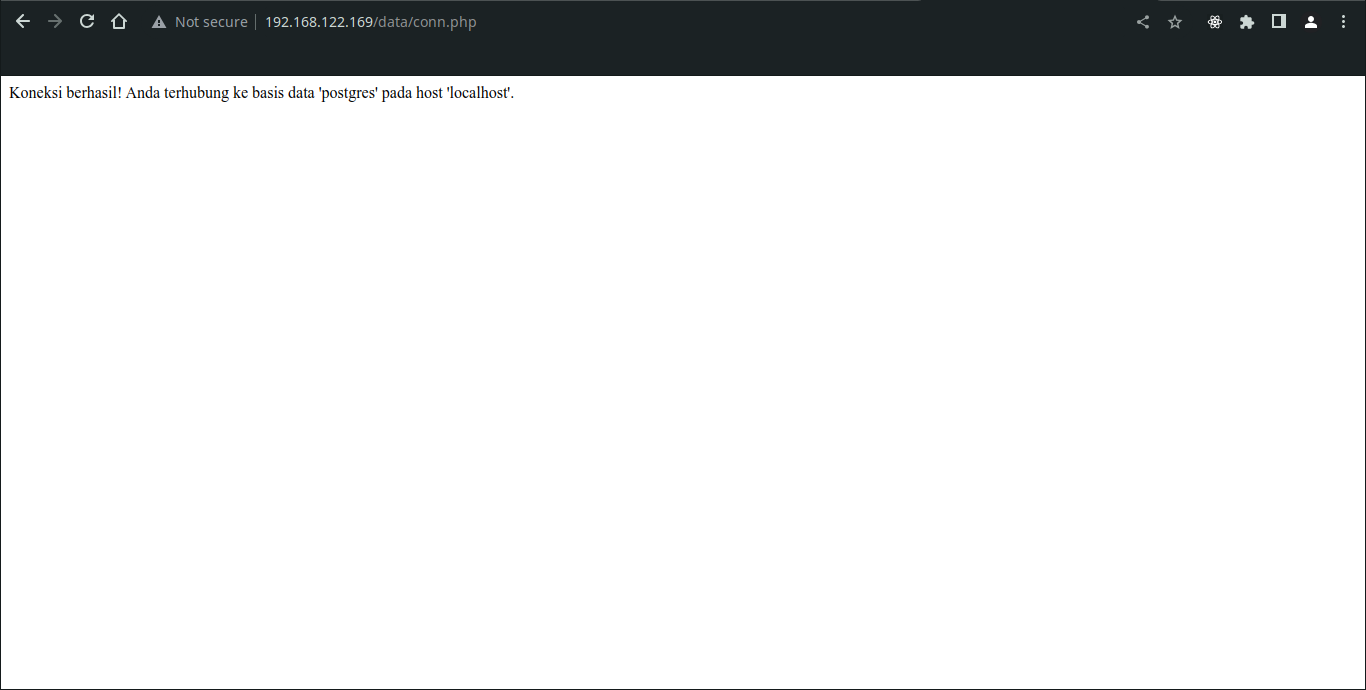
|  |
| --- |
| CREATE TABLE data\_diri (  id serial PRIMARY KEY,  nama varchar(50),  nama\_lengkap varchar(100),  username varchar(50),  password varchar(255),  tempat\_lahir varchar(50),  no\_telephon varchar(15),  tanggal\_lahir date,  biografi text,  create\_time timestamp DEFAULT current\_timestamp,  update\_time timestamp DEFAULT current\_timestamp  ); |

ate\_time timestamp DEFAULT current\_timestamp, timestamp DEFAULT current\_timestamp

Buat script php untuk melakukan testing konektifitas.

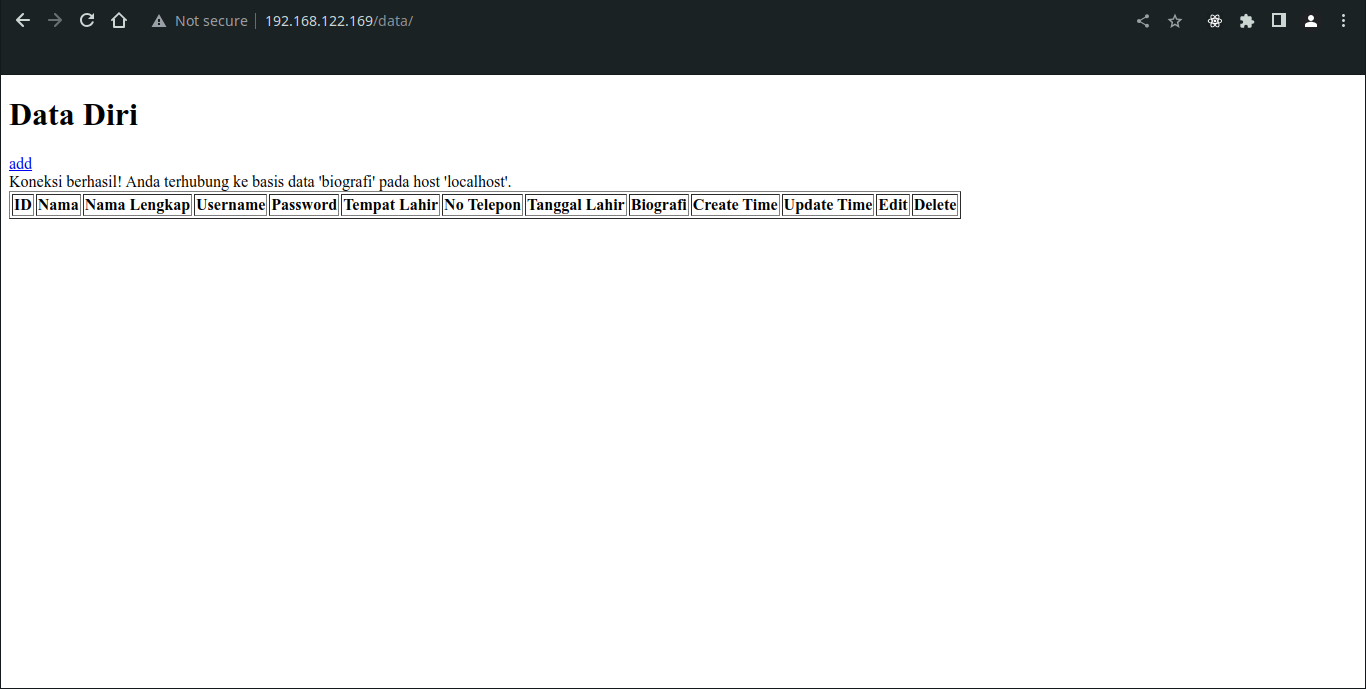
|  |
| --- |
| <?php  // Informasi koneksi PostgreSQL  $dbhost = 'localhost'; // Ganti dengan alamat host PostgreSQL Anda  $dbport = '5433'; // Ganti dengan port PostgreSQL Anda  $dbname = 'rep'; // Nama basis data  $dbuser = 'postgres'; // Nama pengguna PostgreSQL Anda  $dbpass = 'postgres'; // Kata sandi PostgreSQL Anda  // Membuat koneksi  $conn = pg\_connect("host=$dbhost port=$dbport dbname=$dbname user=$dbuser password=$dbpass");  // Memeriksa apakah koneksi berhasil  if (!$conn) {  die("Koneksi gagal: " . pg\_last\_error());  }  echo "Koneksi berhasil! Anda terhubung ke basis data '$dbname' pada host '$dbhost'."; |

Simpan dengan nama conn.php di /var/www. Jika sudah berhasil maka akan tampil seperti berikut.

Sekarang buat script untuk index.php yang berfungsi untuk menampilkan semua.

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Data Diri</title>  </head>  <body>  <h1>Data Diri</h1>  <div>  <a href = "form.php" >add</a>  </div>  <?php  // Sertakan berkas koneksi  include 'conn.php';  // Ambil data dari tabel 'data\_diri'  $sql = "SELECT \* FROM data\_diri";  $result = pg\_query($conn, $sql);  if (!$result) {  die("Gagal mengambil data: " . pg\_last\_error($conn));  }  // Tampilkan data dalam bentuk tabel HTML  echo '<table border="1">';  echo '<tr><th>ID</th><th>Nama</th><th>Nama Lengkap</th><th>Username</th><th>Password</th><th>Tempat Lahir</th><th>No Telepon</th><th>Tanggal Lahir</th><th>Biografi</th><th>Create Time</th><th>Update Time</th><th>Edit</th><th>Delete</th></tr>';  $no = 1;  while ($row = pg\_fetch\_assoc($result)) {  echo '<tr>';  // echo '<td>' . $row['id'] . '</td>';  echo '<td>' . $no++ . '</td>';  echo '<td>' . $row['nama'] . '</td>';  echo '<td>' . $row['nama\_lengkap'] . '</td>';  echo '<td>' . $row['username'] . '</td>';  echo '<td>' . $row['password'] . '</td>';  echo '<td>' . $row['tempat\_lahir'] . '</td>';  echo '<td>' . $row['no\_telephon'] . '</td>';  echo '<td>' . $row['tanggal\_lahir'] . '</td>';  echo '<td>' . $row['biografi'] . '</td>';  echo '<td>' . $row['create\_time'] . '</td>';  echo '<td>' . $row['update\_time'] . '</td>';  echo '<td><a href="edit.php?id=' . $row['id'] . '">Edit</a></td>'; // Tambahkan link ke halaman edit.php dengan parameter ID  echo '<td><a href="delete.php?id=' . $row['id'] . '">Delete</a></td>'; // Tambahkan link ke halaman delete.php dengan parameter ID  echo '</tr>';  }  echo '</table>';  ?>  </body>  </html> |

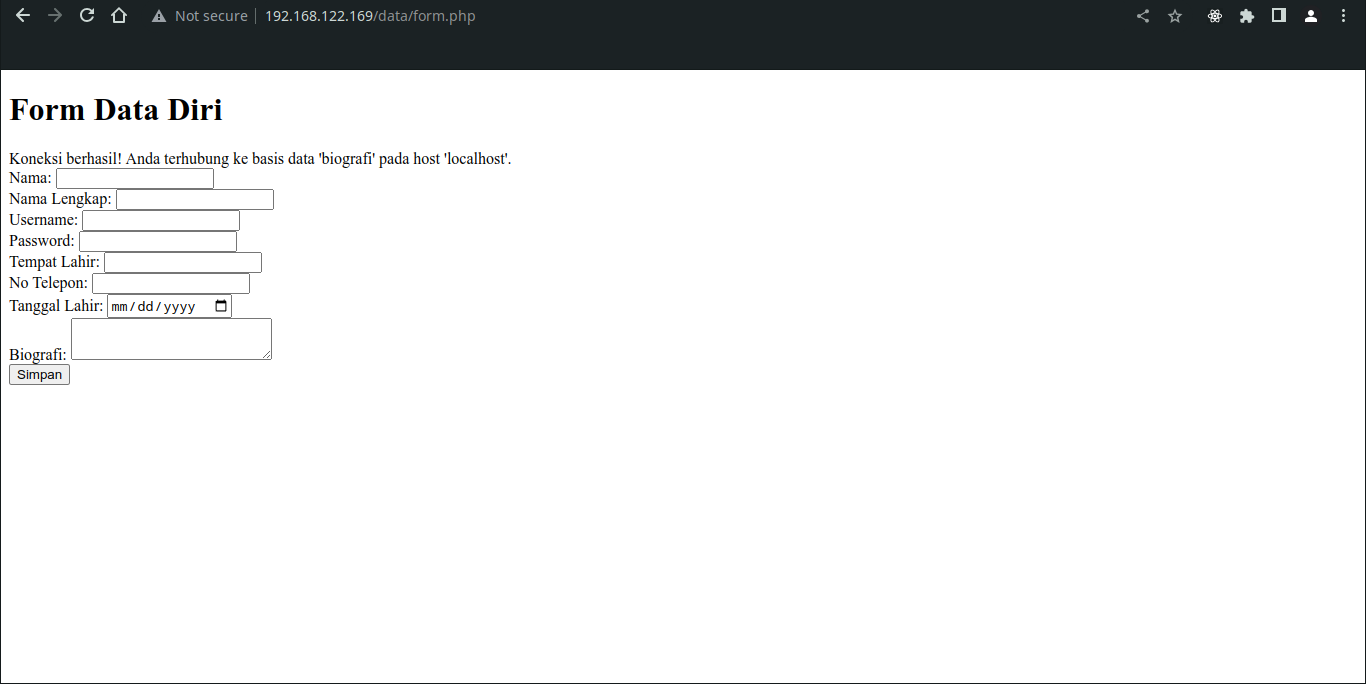
Jika sudah disimpan sebagai index.php maka di browser akan tampil seperti ini.



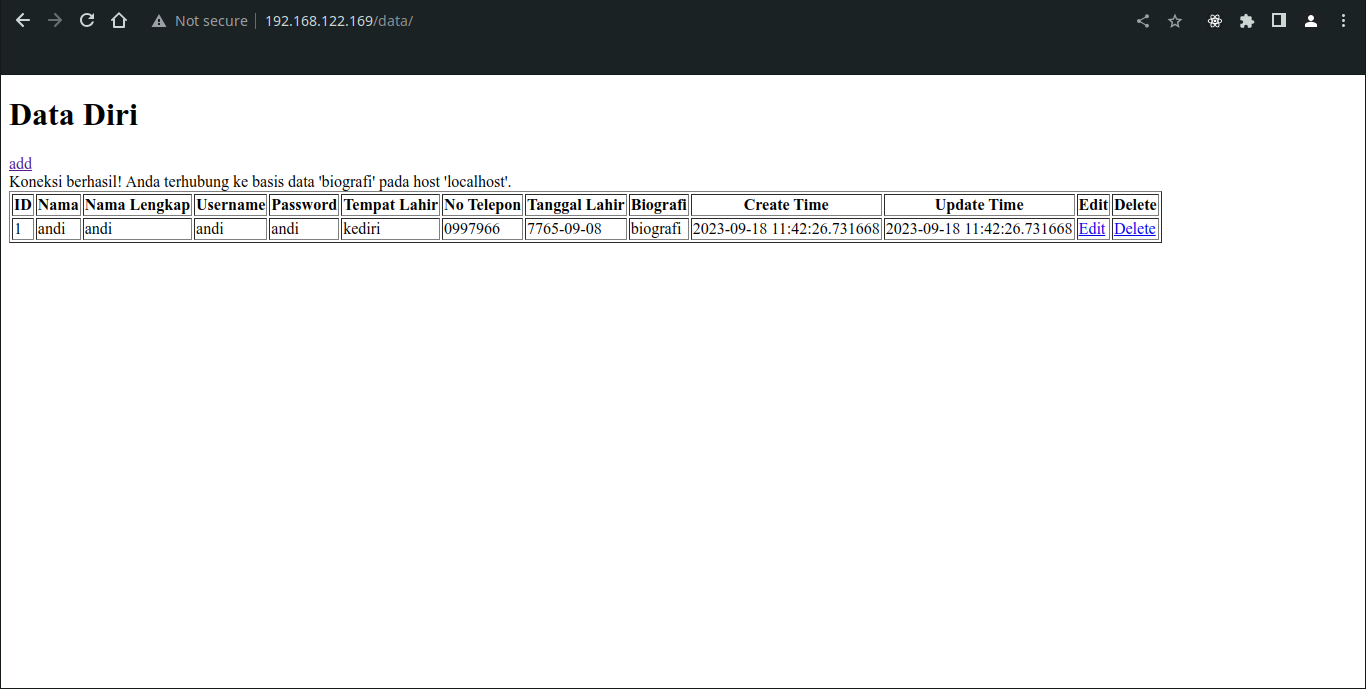
Sekarang kita akan membuat supaya dapat menambahkan data.

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Form Data Diri</title>  </head>  <body>  <h1>Form Data Diri</h1>    <?php  // Sertakan berkas koneksi  include 'conn.php';  // Proses data saat formulir dikirimkan  if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {  // Ambil data dari formulir  $nama = $\_POST['nama'];  $nama\_lengkap = $\_POST['nama\_lengkap'];  $username = $\_POST['username'];  $password = $\_POST['password'];  $tempat\_lahir = $\_POST['tempat\_lahir'];  $no\_telephon = $\_POST['no\_telephon'];  $tanggal\_lahir = $\_POST['tanggal\_lahir'];  $biografi = $\_POST['biografi'];  // Masukkan data ke dalam tabel 'data\_diri'  $sql = "INSERT INTO data\_diri (nama, nama\_lengkap, username, password, tempat\_lahir, no\_telephon, tanggal\_lahir, biografi)  VALUES ('$nama', '$nama\_lengkap', '$username', '$password', '$tempat\_lahir', '$no\_telephon', '$tanggal\_lahir', '$biografi')";  $result = pg\_query($conn, $sql);  if ($result) {  echo "Data berhasil disimpan.";  } else {  echo "Gagal menyimpan data: " . pg\_last\_error($conn);  }  }  ?>  <form method="POST" action="">  <label for="nama">Nama:</label>  <input type="text" name="nama" required><br>  <label for="nama\_lengkap">Nama Lengkap:</label>  <input type="text" name="nama\_lengkap" required><br>  <label for="username">Username:</label>  <input type="text" name="username" required><br>  <label for="password">Password:</label>  <input type="password" name="password" required><br>  <label for="tempat\_lahir">Tempat Lahir:</label>  <input type="text" name="tempat\_lahir" required><br>  <label for="no\_telephon">No Telepon:</label>  <input type="text" name="no\_telephon"><br>  <label for="tanggal\_lahir">Tanggal Lahir:</label>  <input type="date" name="tanggal\_lahir" required><br>  <label for="biografi">Biografi:</label>  <textarea name="biografi"></textarea><br>  <input type="submit" value="Simpan">  </form>  </body>  </html> |

Simpan /var/www/form.php



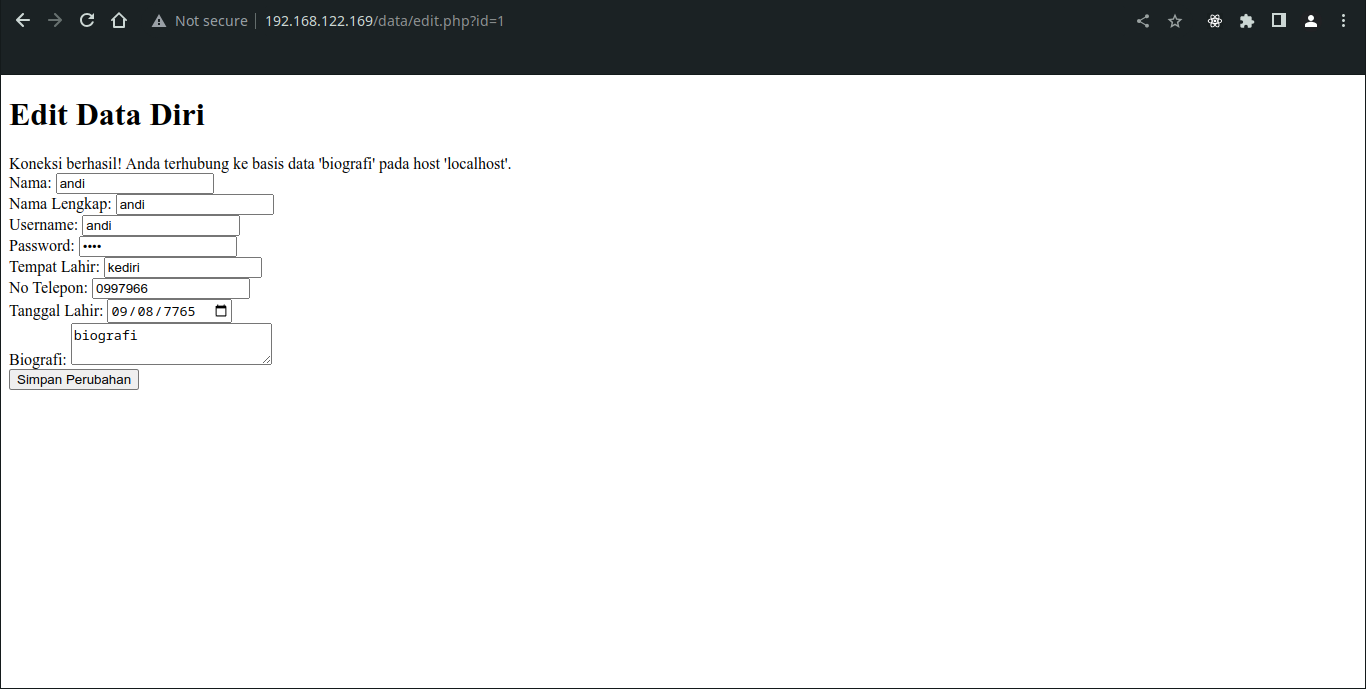
Isi form dan sesuaikan jika sudah maka kembali ke index.php. Maka akan tampil seperti ini.

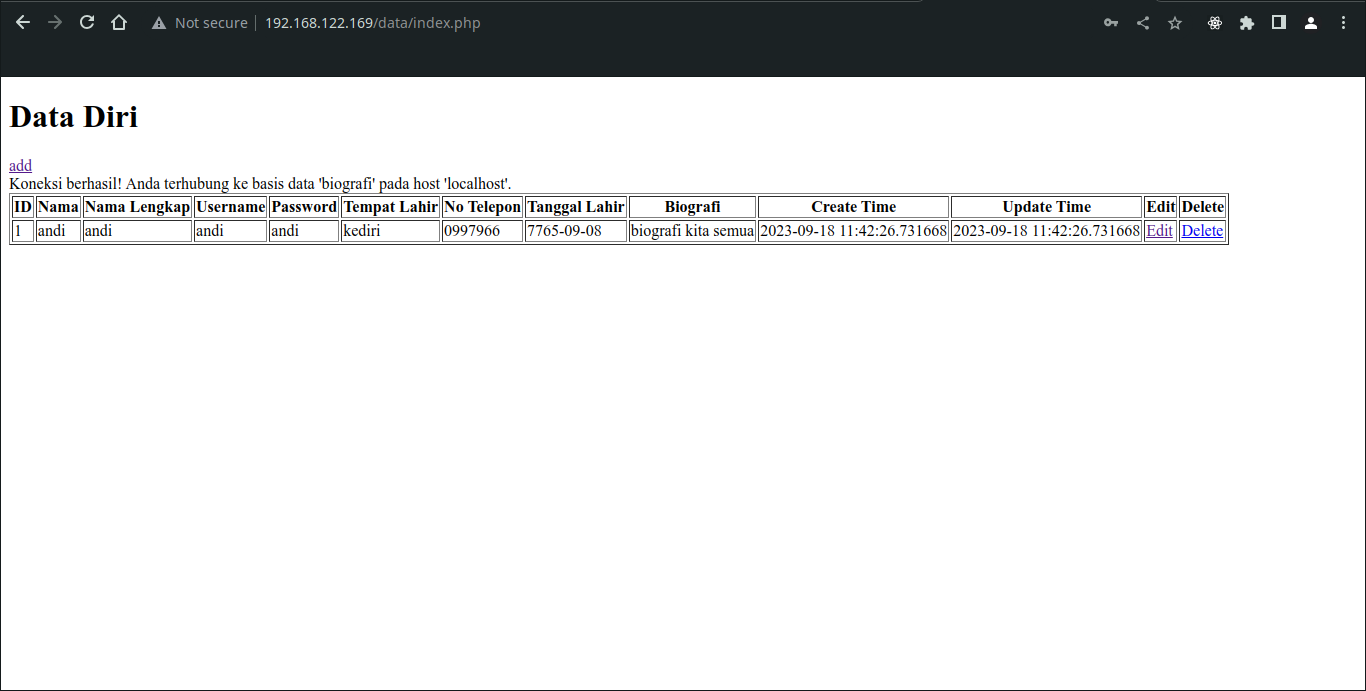


Berikutnya buat script edit

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Edit Data Diri</title>  </head>  <body>  <h1>Edit Data Diri</h1>    <?php  // Sertakan berkas koneksi  include 'conn.php';  // Pastikan parameter 'id' telah diterima melalui URL  if (isset($\_GET['id'])) {  // Ambil nilai 'id' dari URL  $id = $\_GET['id'];  // Proses saat formulir dikirimkan (perubahan data)  if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {  // Ambil data dari formulir  $nama = $\_POST['nama'];  $nama\_lengkap = $\_POST['nama\_lengkap'];  $username = $\_POST['username'];  $password = $\_POST['password'];  $tempat\_lahir = $\_POST['tempat\_lahir'];  $no\_telephon = $\_POST['no\_telephon'];  $tanggal\_lahir = $\_POST['tanggal\_lahir'];  $biografi = $\_POST['biografi'];  // Perintah SQL untuk mengupdate data berdasarkan 'id'  $sql = "UPDATE data\_diri  SET nama = '$nama',  nama\_lengkap = '$nama\_lengkap',  username = '$username',  password = '$password',  tempat\_lahir = '$tempat\_lahir',  no\_telephon = '$no\_telephon',  tanggal\_lahir = '$tanggal\_lahir',  biografi = '$biografi'  WHERE id = $id";  $result = pg\_query($conn, $sql);  if ($result) {  echo '<script>alert("Berhasil Melakukan Perubahan.");</script>';  echo '<script>window.location.href = "index.php";</script>'; // Redirect kembali ke halaman utama setelah pengeditan berhasil  } else {  die("Gagal melakukan perubahan: " . pg\_last\_error($conn));  }  }  // Ambil data yang akan diedit dari database  $query = "SELECT \* FROM data\_diri WHERE id = $id";  $result = pg\_query($conn, $query);  $data = pg\_fetch\_assoc($result);  } else {  echo "ID tidak ditemukan.";  }  ?>  <!-- Formulir untuk mengedit data -->  <form method="POST" action="">  <label for="nama">Nama:</label>  <input type="text" name="nama" value="<?php echo $data['nama']; ?>" required><br>  <label for="nama\_lengkap">Nama Lengkap:</label>  <input type="text" name="nama\_lengkap" value="<?php echo $data['nama\_lengkap']; ?>" required><br>  <label for="username">Username:</label>  <input type="text" name="username" value="<?php echo $data['username']; ?>" required><br>  <label for="password">Password:</label>  <input type="password" name="password" value="<?php echo $data['password']; ?>" required><br>  <label for="tempat\_lahir">Tempat Lahir:</label>  <input type="text" name="tempat\_lahir" value="<?php echo $data['tempat\_lahir']; ?>" required><br>  <label for="no\_telephon">No Telepon:</label>  <input type="text" name="no\_telephon" value="<?php echo $data['no\_telephon']; ?>"><br>  <label for="tanggal\_lahir">Tanggal Lahir:</label>  <input type="date" name="tanggal\_lahir" value="<?php echo $data['tanggal\_lahir']; ?>" required><br>  <label for="biografi">Biografi:</label>  <textarea name="biografi"><?php echo $data['biografi']; ?></textarea><br>  <input type="submit" value="Simpan Perubahan">  </form>  </body>  </html> |

Kemudian simpan /var/www/edit.php ketika kita klik edit maka akan masuk kedalam edit.php.

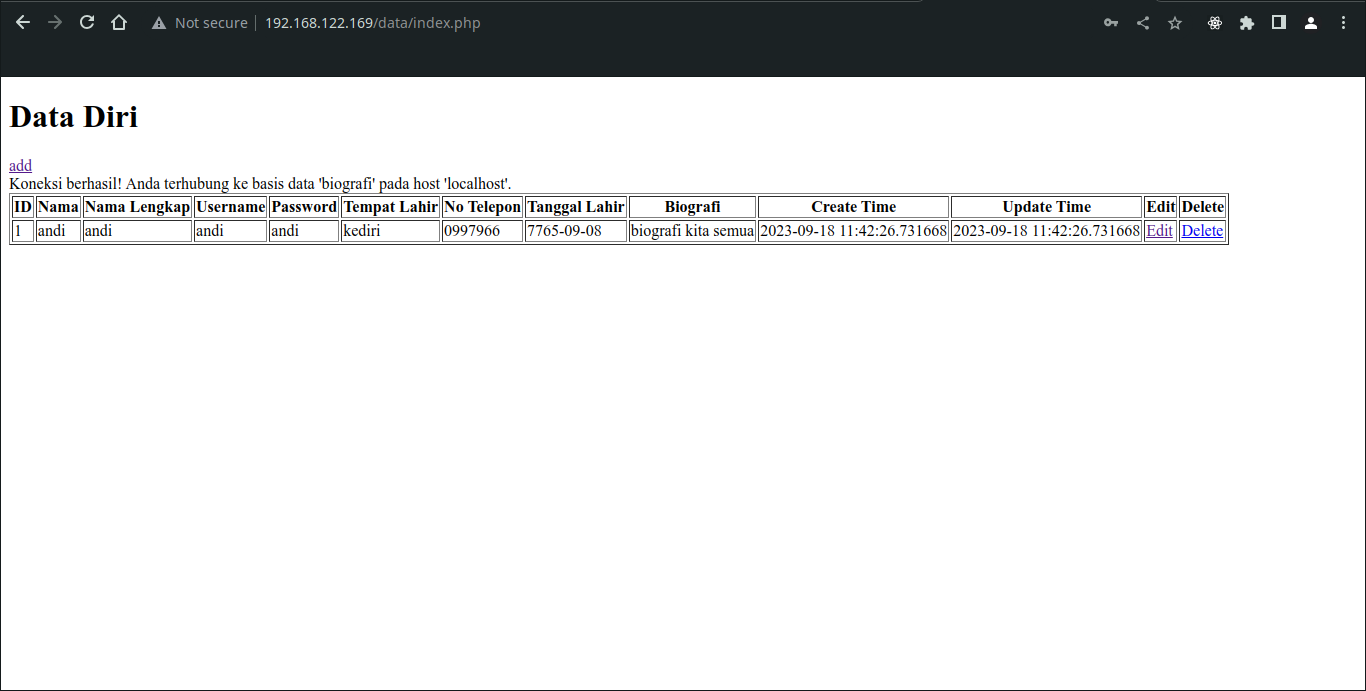


Lakukan perubahan kemudian simpan, setelah itu lihat pada index.php, maka peruahan akan ditampilkan.

Berikutnya buat skript delete.

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Delete Data</title>  </head>  <body>  <h1>Delete Data</h1>  <?php  // Sertakan berkas koneksi  include 'conn.php';  // Pastikan parameter 'id' telah diterima melalui URL  if (isset($\_GET['id'])) {  // Ambil nilai 'id' dari URL  $id = $\_GET['id'];  // Perintah SQL untuk menghapus data berdasarkan 'id'  $sql = "DELETE FROM data\_diri WHERE id = $id";  $result = pg\_query($conn, $sql);  if ($result) {  echo '<script>alert("Berhasil Menghapus Data.");</script>';  echo '<script>window.location.href = "index.php";</script>'; // Redirect kembali ke halaman utama setelah menghapus  } else {  die("Gagal menghapus data: " . pg\_last\_error($conn));  }  } else {  echo "ID tidak ditemukan.";  }  ?>  </body>  </html> |

Simpan di /var/www/delete.php

Setelah klik delete, maka data akan terhapus. Seperti dibawah ini.

