Nama : Azhar Adi Dirgantara Setiawan

Npm : 06.2020.1.07277

* CLASS SEARCH

// Creating a Search interface

package rmi;

import java.rmi.\*;

public interface Search extends Remote

{

// Declaring the method prototype

public String query(String search) throws RemoteException;

}

***Antarmuka yang akan memberikan deskripsi metode yang dapat dipanggil oleh klien jarak jauh.***

* CLASS SEARCH QUERY

package rmi;

import java.rmi.\*;

import java.rmi.server.\*;

public class SearchQuery extends UnicastRemoteObject implements Search

{

SearchQuery() throws RemoteException

{

super();

}

// Implementation of the query interface

public String query(String search)

throws RemoteException

{

String result;

if (search.equals("Reflection in Java"))

result = "Found";

else

result = "Not Found";

return result;

}

}

***Untuk mengimplementasikan antarmuka jarak jauh, kelas harus diperluas ke kelas UnicastRemoteObject dari paket java.rmi.***

* CLASS SEARCH SERVER

// Java program for server application

package rmi;

import java.rmi.\*;

import java.rmi.registry.\*;

public class SearchServer

{

public static void main(String args[])

{

try

{

// Create an object of the interface

// implementation class

Search obj = new SearchQuery();

// rmiregistry within the server JVM with

// port number 1900

LocateRegistry.createRegistry(1900);

// Binds the remote object by the name

// geeksforgeeks

Naming.rebind("rmi://localhost:1900"+

"/geeksforgeeks",obj);

}

catch(Exception ae)

{

System.out.println(ae);

}

}

}

***Alat rmic digunakan untuk memanggil compiler rmi yang membuat objek stub dan skeleton. Prototipenya adalah nama kelas rmic. Untuk program di atas, perintah berikut harus dijalankan pada prompt perintah rmic SearchQuery***

* CLASS CLIENT REQUEST

package rmi;

import java.rmi.\*;

public class ClientRequest

{

public static void main(String args[])

{

String answer,value="Reflection in Java";

try

{

// lookup method to find reference of remote object

Search access =

(Search)Naming.lookup("rmi://localhost:1900"+

"/geeksforgeeks");

answer = access.query(value);

System.out.println("Article on " + value +

" " + answer+" at GeeksforGeeks");

}

catch(Exception ae)

{

System.out.println(ae);

}

}

}

***Program Server menggunakan metode createRegistry dari kelas LocateRegistry untuk membuat rmiregistry di dalam JVM server dengan nomor port yang diteruskan sebagai argument. Metode rebind dari kelas penamaan digunakan untuk mengikat objek jarak jauh ke nama baru***

**Hasil Run**

