**LAPORAN LATIHAN 2**

Remote Invocation



**Dibuat oleh:**

Friskadini Ismayanti (1301198496)

Muhammad Faisal Amir (1301198497)

Ridho Maulana Cahyudi (1301198515)

**TELKOM UNIVERSITY**

**FAKULTAS INFORMATIKA**

**2020**

**Code Program**

config.py

|  |  |
| --- | --- |
| 1 |  |
| 2 | BASE\_CONFIG\_IP\_ADDRESS = **'127.0.0.1'** |
| 3 | BASE\_CONFIG\_PORT = 8000 |
| 4 | BASE\_CONFIG\_URL = **'http://127.0.0.1:8000'** |
| 5 | BASE\_CONFIG\_PATH = **'/RPC2'** |
| 6 |  |
| 7 | BASE\_VERSION\_SERVER = 3 |
| 8 | BASE\_VERSION\_CLIENT = 2 |

rpc\_client.py

|  |  |
| --- | --- |
| 1 | *# import library xmlrpc client karena akan digunakan rpc* |
| 2 | **import** xmlrpc.client |
| 3 |  |
| 4 | *# import config file* |
| 5 | **from** com.frogobox.config **import** \* |
| 6 |  |
| 7 | *# buat stub/skeleton (proxy) pada client* |
| 8 | rpcServer = xmlrpc.client.ServerProxy(BASE\_CONFIG\_URL) |
| 9 |  |
| 10 | *# versi dari client* |
| 11 | version\_client = BASE\_VERSION\_CLIENT |
| 12 |  |
| 13 | *# Menampilkan Versi Client* |
| 14 | print(**"Version Client \t: "** + str(version\_client)) |
| 15 | print(**"-- Checking Version From Server --"**) |
| 16 | print(**"Version Server \t: "** + str(rpcServer.get\_version\_update())) *# Menampilkan versi server* |
| 17 | print() |
| 18 | print(**"Result : "**) |
| 19 |  |
| 20 | print(**"----------------------------------"**) |
| 21 | **if** rpcServer.updated\_version(version\_client): *# Checking versi server dan client* |
| 22 | print(**"Latest Version Apps - No Need Update"**) |
| 23 | **else**: |
| 24 | inputUser = input(**"Apakah Ingin Memperbarui Version? (Y/N) "**) |
| 25 | **if** inputUser == **'y'**: |
| 26 | *# Update versi client* |
| 27 | version\_client = rpcServer.get\_version\_update() |
| 28 | print(**"Thanks For Update"**) |
| 29 | print(**"Version Client \t : "** + str(version\_client)) |
| 30 | **elif** inputUser == **'n'**: |
| 31 | *# Tidak mengupdate versi Client* |
| 32 | print(**"Your version not updated"**) |
| 33 | print(**"Version Client \t : "** + str(version\_client)) |
| 34 | **else**: |
| 35 | print(**"Updating Error"**) |
| 36 | print(**"----------------------------------"**) |

rpc\_server.py

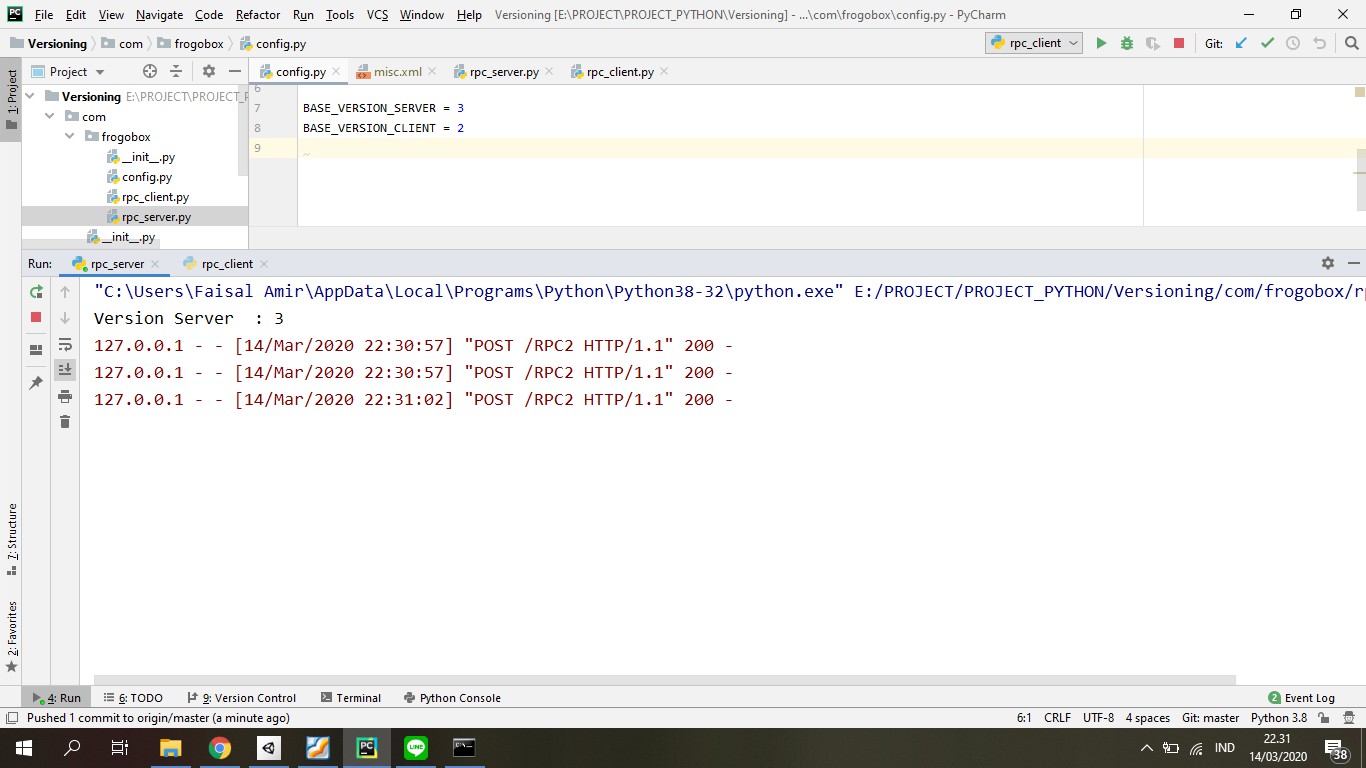
|  |  |
| --- | --- |
| 1 | *# gunakan xmlrpc bagian server* |
| 2 | *# import SimpleXMLRPCServer dan SimpleXMLRPCRequestHandler* |
| 3 | **from** xmlrpc.server **import** SimpleXMLRPCServer |
| 4 | **from** xmlrpc.server **import** SimpleXMLRPCRequestHandler |
| 5 |  |
| 6 | *# buat kelas requesthandler* |
| 7 | **from** com.frogobox.config **import** \* |
| 8 |  |
| 9 |  |
| 10 | **class** RequestHandler(SimpleXMLRPCRequestHandler): |
| 11 | *# batasi pada path /RPC2 saja* |
| 12 | rpc\_paths = (BASE\_CONFIG\_PATH) |
| 13 |  |
| 14 |  |
| 15 | *# Deklarasi versi server* |
| 16 | version\_server = BASE\_VERSION\_SERVER |
| 17 |  |
| 18 | *# Menampilkan versi server* |
| 19 | print(**"Version Server \t: "** + str(version\_server)) |
| 20 |  |
| 21 | *# buat server serta register fungsi register\_introspection\_functions()* |
| 22 | rpcServer = SimpleXMLRPCServer((BASE\_CONFIG\_IP\_ADDRESS, BASE\_CONFIG\_PORT), requestHandler=RequestHandler) |
| 23 | rpcServer.register\_introspection\_functions() |
| 24 |  |
| 25 |  |
| 26 | *# Fungsi untuk mengecek versi* |
| 27 | **def** updated\_version(version\_client): |
| 28 | **if** version\_client == version\_server: |
| 29 | **return True** |
| 30 | **else**: |
| 31 | **return False** |
| 32 |  |
| 33 |  |
| 34 | *# Fungsi mengembalikan nilai versi server* |
| 35 | **def** get\_version\_update(): |
| 36 | **return** version\_server |
| 37 |  |
| 38 |  |
| 39 | *# Registrasi fungsi* |
| 40 | rpcServer.register\_function(updated\_version, **'updated\_version'**) |
| 41 | rpcServer.register\_function(get\_version\_update, **'get\_version\_update'**) |
| 42 |  |
| 43 | *# Jalankan server* |
| 44 | rpcServer.serve\_forever() |

**ANALISIS CODE**

Analisis RPC Server

|  |  |
| --- | --- |
| CODE | FUNGSI (PENJELASAN) |
| **from** xmlrpc.server **import** SimpleXMLRPCServer | Untuk membuat server melintasi platform dengan Bahasa mandiri menggunakan protocol XML-RPC |
| **from** xmlrpc.server **import** SimpleXMLRPCRequestHandler | Untuk mengontrol permintaan baru |
| **from** com.frogobox.config **import** \* **class** RequestHandler(SimpleXMLRPCRequestHandler): | Untuk membuat kelas baru untuk requesthandler |
| **version\_server = BASE\_VERSION\_SERVER** | Membuat versi dari server |
| rpc\_paths = (BASE\_CONFIG\_PATH) | Untuk membatasi pada path |
| print("Version Server \t: " + str(version\_server)) | Menampilkan very dari server |
| rpcServer = SimpleXMLRPCServer((BASE\_CONFIG\_IP\_ADDRESS, BASE\_CONFIG\_PORT), requestHandler=RequestHandler) rpcServer.register\_introspection\_functions() | Untuk membuat server dan register fungsi introspection |
| def updated\_version(version\_client):  if version\_client == version\_server:  return True  else:  return False | Fungsi yang digunakan untuk melakukan pengecekan versi pada client dan server |
| **def get\_version\_update():**  **return version\_server** | Fungsi yang mendapatkan nilai dari server |
| **rpcServer.register\_function(updated\_version, 'updated\_version')**  **rpcServer.register\_function(get\_version\_update, 'get\_version\_update')** | Mendafarkan fungsi agar dapat di gunakan di client |
| rpcServer.serve\_forever() | Untuk membuat server dapat digunakan |

Screenshoot RPC SERVER



Analisis RPC Client

|  |  |
| --- | --- |
| CODE | FUNGSI (PENJELASAN) |
| **import** xmlrpc.client | Memanggil library untuk digunakan RPC |
| **from** com.frogobox.config **import** \*  rpcServer = xmlrpc.client.ServerProxy(BASE\_CONFIG\_URL) | Membuat proxy untuk client |
| version\_client = BASE\_VERSION\_CLIENT | Membuat versi dari client |
| print("Version Client \t: " + str(version\_client))  print("-- Checking Version From Server --")  print("Version Server \t: " + str(rpcServer.get\_version\_update()))  print()  print("Result : ") | Menampilkan versi dari client dan melakukan pengecekan dengan versi client |
| print("----------------------------------")  if rpcServer.updated\_version(version\_client):  print("Latest Version Apps - No Need Update")  else:  inputUser = input("Apakah Ingin Memperbarui Version? (Y/N) ")  if inputUser == 'y':  version\_client = rpcServer.get\_version\_update()  print("Thanks For Update")  print("Version Client \t : " + str(version\_client))  elif inputUser == 'n':  print("Your version not updated")  print("Version Client \t : " + str(version\_client))  else:  print("Updating Error")  print("----------------------------------") | Melakukan pengecekan dan di beri kondisi dimana versi server dan client sudah sama atau belum dan ketika belum akan melakukan perubahan versi dari client agar sama dengan versi yang dari server dan ketika sudah sama versi server dan client tidak akan melakukan pengecekan |

Screenshoot RPC Client

