

LAPORAN TUGAS BESAR
SISTEM PARALEL DAN TERDISTRIBUSI
“ Siaga COVID-19 ”

Disusun untuk memenuhi salah satu tugas mata kuliah Sistem Paralel dan Terdistribusi

Oleh :

I Wayan Adi Wahyudi	(1301194084)
Muhammad Ikram Kaer Sinapoy	(1301193341)
Mohammad Mirza Qusyairi	(1301194203)
Muhammad Sulthon Asramanggala	(1301194008)

IF-43-05



PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM
2021

DAFTAR ISI

1. Pendahuluan	3
A. Deskripsi Tugas Besar	3
B. Kontribusi Anggota Kelompok	3
2. Analisis	3
A. Pemilihan Solusi	3
B. Model Sistem	4
3. Perancangan	4
A. Arsitektur Sistem	4
B. Alur Proses Aplikasi	6
a. Client	6
b. Server	6
4. Implementasi	7
A. Source Code & Screen capture saat aplikasi dijalankan	7
B. Link Video Dokumentasi	7

1. Pendahuluan

A. Deskripsi Tugas Besar

Pada tugas besar kali ini kami diminta untuk membuat sebuah program "Siaga COVID-19" dan diminta untuk menerapkan paradigma sistem paralel ataupun terdistribusi. Program ini dibuat agar masyarakat dapat melaporkan kasus Covid yang terjadi di daerahnya (NIK pelapor, nama pelapor, nama terduga Covid, alamat terduga Covid, gejala yang dirasakan). Server akan memilah laporan mana yang valid atau tidak berdasarkan NIK yang sudah ada dalam database (csv/txt). Apabila valid, maka server akan merespon kepada client berupa waktu, nama dan jumlah orang yang akan melakukan penjemputan.

B. Kontribusi Anggota Kelompok

- a. I Wayan Adi Wahyudi : Server dan Client
- b. Muhammad Ikram Kaer Sinapoy : Laporan dan Testing
- c. Mohammad Mirza Qusyairi : Server dan Client
- d. Muhammad Sulthon Asramanggala : Laporan dan Testing

2. Analisis

A. Pemilihan Solusi

Untuk solusi yang kami terapkan dari tugas besar mengenai siaga covid 19 ini yaitu kami menerapkan konsep RPC. Teknik ini merupakan sebuah metode yang memungkinkan kita untuk mengakses sebuah prosedur yang berada di komputer lain. Untuk dapat melakukan ini sebuah server harus menyediakan layanan remote procedure. Pendekatan yang dilakukan adalah sebuah server membuka socket, lalu menunggu client yang meminta prosedur yang disediakan oleh server. Hal ini tentunya sangat berguna untuk siaga covid 19 dimana nantinya server akan memberikan informasi kepada client mengenai penjemputan pasien covid 19, karena sudah menerapkan sistem RPC nantinya proses penjemputan pasien yang terpapar covid 19 akan mudah dilakukan.

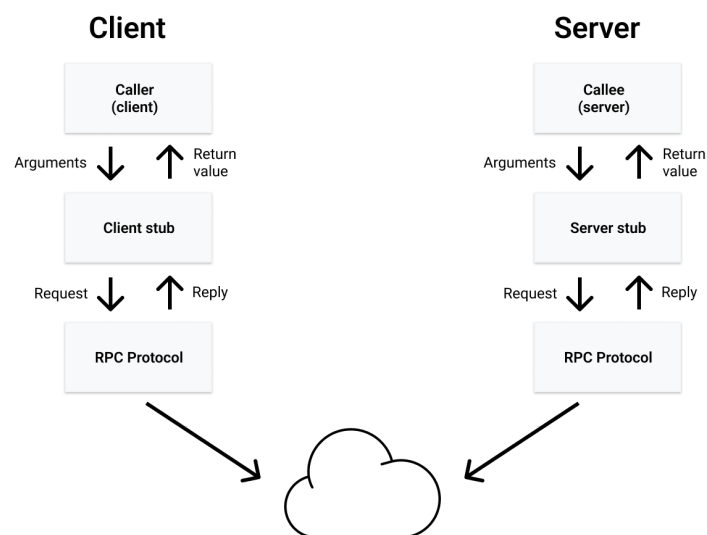
B. Model Sistem

Untuk model sistem yang dibangun kami menggunakan sistem client dan server. Client disini bertugas untuk melakukan input data pasien (NIK pelapor, nama pelapor, nama terduga Covid, alamat terduga Covid, gejala yang dirasakan) lalu server akan melakukan pengecekan terhadap database apakah sesuai dengan yang diinput oleh client jika sesuai server akan mereturn waktu penjemputan dan petugas yang mengantar jika data tidak terdapat dalam database server mereturn pesan gagal. RPC merupakan salah satu metode yang terdapat dalam sistem paralel dan distribusi. Metode ini merupakan suatu terobosan dalam sistem distribusi yang mana tujuan dari metode ini adalah membuat program sistem

distribusi dapat mirip dengan pemrograman pada umumnya (sumber: Materi 06-Remote Invocation). Dengan menggunakan RPC, pihak client dapat menggunakan prosedur-prosedur yang telah didaftarkan oleh server. Hal tersebut memperbolehkan client untuk mengolah data yang dimiliki menggunakan prosedur-prosedur yang terdapat pada sisi server.

3. Perancangan

A. Arsitektur Sistem

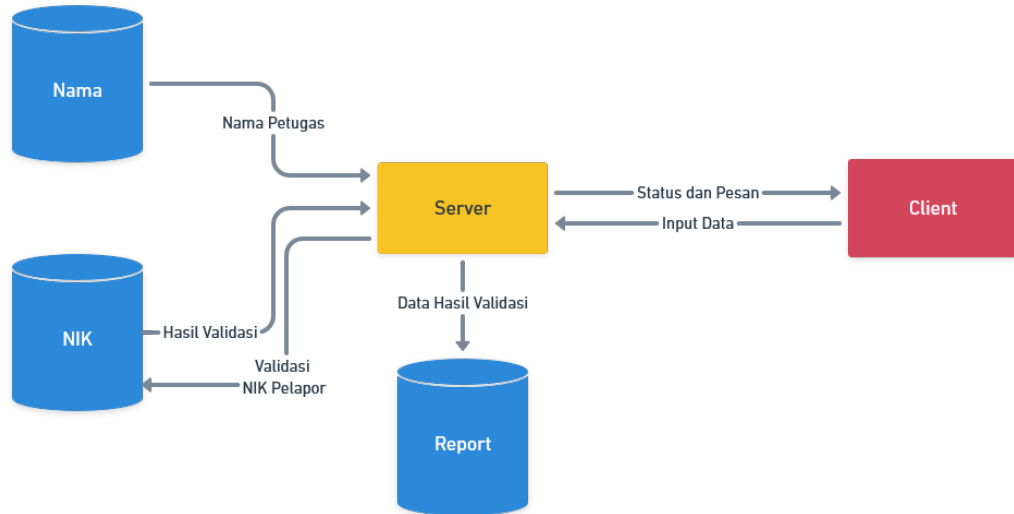


RPC didefinisikan menggunakan rutin yang terkandung di dalam protokol RPC. Tiap message dari panggilan akan disesuaikan dengan message balikan. Protokol ini juga mendukung adanya prosedur callback dan select subroutine pada sisi server. client dan Server client adalah komputer atau proses yang mengakses suatu servis/layanan atau resources dari proses atau komputer pada suatu jaringan. Server adalah komputer yang menyediakan servis/layanan dan resources, dan yang mengimplementasikan servis jaringan.

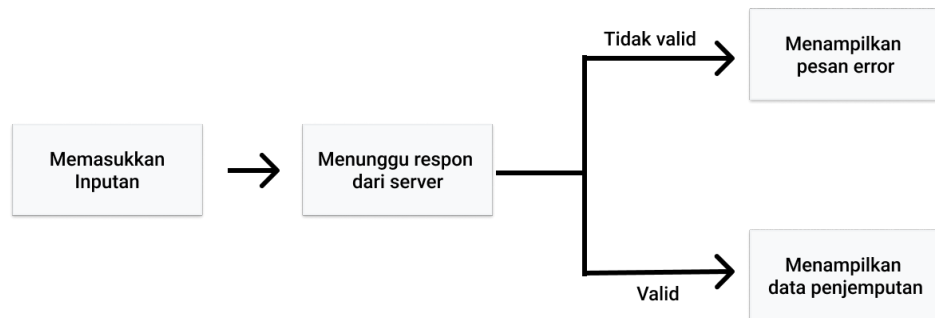
Tiap klien yang akan melakukan RPC pada suatu server di jaringan akan menerima balasan (reply) berupa hasil dari eksekusi program tersebut dengan mencocokkan message balasan untuk tiap call message yang diminta client. Client dapat mengirim message calls ke server dalam jumlah besar secara sequence (berurutan). Pada saat melakukan message calls ke server, klien tidak menunggu server untuk memberikan reply terhadap tiap messages yang dikirim, begitu pula dengan server yang tidak pernah mengirimkan messages reply. Fitur inilah yang banyak digunakan klien, karena arsitektur RPC didesain agar pada tiap call

message yang dikirimkan oleh klien harus ada proses menunggu balasan dari server.

B. Alur Proses Aplikasi

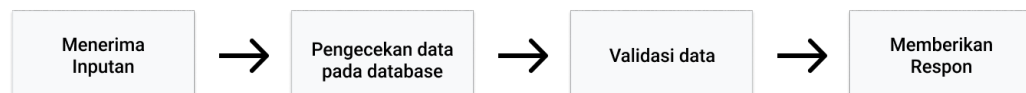


a. Client



Pada sisi Client, setelah program dijalankan, akan menginputkan data kemudian data tersebut dikirim ke server untuk dilakukan pengecekan terhadap data tersebut, kemudian setelah data tersebut dilakukan pengecekan, client akan menerima respon dari server jika data tersebut Valid maka akan menampilkan data penjemputan dan jika data tersebut Tidak Valid maka akan menampilkan pesan error.

b. Server



Pada sisi Server, setelah program dijalankan, akan menerima inputan data dari sisi client kemudian data tersebut akan dilakukan pengecekan pada database yang ada, setelah melakukan validasi pada data tersebut server akan mengirimkan respon Valid atau Tidak Valid dari data tersebut ke sisi Client.

4. Implementasi

A. Source Code & Screen capture saat aplikasi dijalankan

a. Source Code

https://github.com/adiwahyudi/siaga_covid-19

Server

```
server.py M X
server.py > ...
You, seconds ago | 2 authors (Mohammad Mirza Qusyairi and others)
#Server side
1 from xmlrpc.server import SimpleXMLRPCServer
2 from xmlrpc.server import SimpleXMLRPCRequestHandler
3 from datetime import timedelta, datetime
4 import pandas as pd
5 import numpy as np
6 import random
7
8
9 # Membaca dataset
10 df_nik = pd.read_csv('data_nik.csv')
11 df_penjemput = pd.read_csv('data_penjemput.csv')
12
13 # Ubah dari dataframe ke 1d list
14 df_nik = np.array(df_nik).ravel().tolist()
15 df_penjemput = np.array(df_penjemput).ravel().tolist()
16
17 def validate_nik(nik):
18     return int(nik) in df_nik
19
20 def list_pickup(kuantitas):
21     list_penjemput = []
22     for i in range(kuantitas):
23         list_penjemput.append(df_penjemput[random.randrange(i, 499)])
24     return list_penjemput
25
26 def time_pickup():
27     time = datetime.now() + timedelta(hours=4)
28     time_format = time.strftime("%d-%m-%Y %H:%M:%S")
29
30     return time_format
31
32 # Buat kelas requesthandler batasi pada path /RPC2 saja
33 class RequestHandler(SimpleXMLRPCRequestHandler):
34     rpc_paths = ('/RPC2')
35
36     def do_OPTIONS(self):
37         self.send_response(200)
38         self.end_headers()
39
40     # Add these headers to all responses
41     def end_headers(self):
42         self.send_header("Access-Control-Allow-Headers",
43             "Origin, X-Requested-With, Content-Type, User-Agent, Accept")
44         self.send_header("Access-Control-Allow-Origin", "*")
45         SimpleXMLRPCRequestHandler.end_headers(self)
```

you, seconds ago | 2 authors (Mohammad Mirza Qusyain and others)

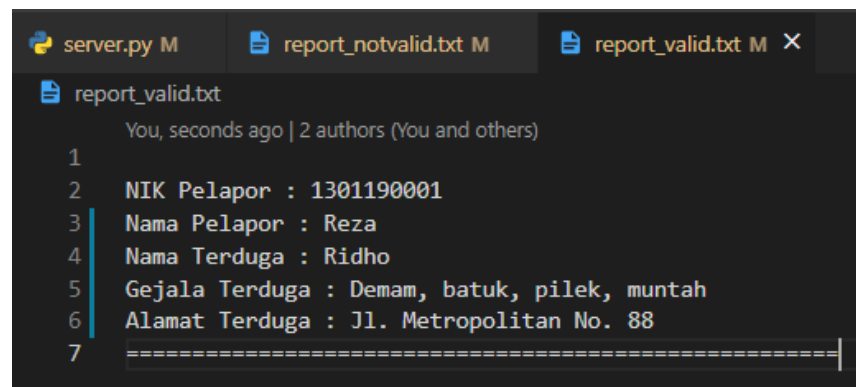
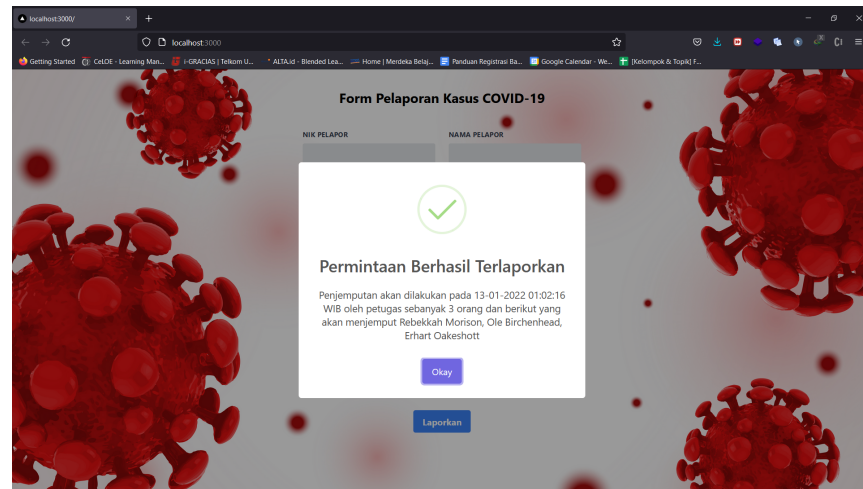
```
33 class RequestHandler(SimpleXMLRPCRequestHandler):
34     rpc_paths = ('/RPC2')
35
36     # Enable CORS
37     def do_OPTIONS(self):
38         self.send_response(200)
39         self.end_headers()
40
41     # Add these headers to all responses
42     def end_headers(self):
43         self.send_header("Access-Control-Allow-Headers",
44                         "Origin, X-Requested-With, Content-Type, User-Agent, Accept")
45         self.send_header("Access-Control-Allow-Origin", "*")
46         SimpleXMLRPCRequestHandler.end_headers(self)
47
```

```
server.py M X
server.py > RequestHandler
48 # Buat variabel port number dan ip server
49 PORT_NUM = 8008
50 IP_SERVER = '26.122.77.13'
51
52 # Buat server serta register fungsi
53 with SimpleXMLRPCServer((IP_SERVER, PORT_NUM), requestHandler = RequestHandler) as server:
54     server.register_introspection_functions()
55
56     def report(nik, nama_pelapor, nama_terduga, gejala_terduga, alamat_terduga):
57         response = dict()
58         if validate_nik(nik):
59             total_pickup = random.randint(3,6)
60             penjemput = list_pickup(total_pickup)
61             penjemput = ', '.join(penjemput)
62             time = time_pickup()
63
64             response['status'] = "Success"
65             response['message'] = "Penjemputan akan dilakukan pada " + str(time) + " WIB oleh petugas sebanyak " + str(total_pickup) + " orang dan berikut yang akan menjemput " + str(penjemput)
66
67             # Save to txt file
68             teks = "\nNIK Pelapor : {} \nNama Pelapor : {} \nNama Terduga : {} \nGejala Terduga : {} \nAlamat Terduga : {}".format(nik, nama_pelapor, nama_terduga, gejala_terduga, alamat_terduga)
69             file_bio = open("report_valid.txt", "a")
70             file_bio.write(teks)
71             file_bio.close()
72         else :
73             response['status'] = "Failed"
74             response['message'] = "NIK pelapor tidak valid"
75
76             # Save to txt file
77             teks = "\nNIK Pelapor : {} \nNama Pelapor : {} \nNama Terduga : {} \nGejala Terduga : {} \nAlamat Terduga : {}".format(nik, nama_pelapor, nama_terduga, gejala_terduga, alamat_terduga)
78             file_bio = open("report_notvalid.txt", "a")
79             file_bio.write(teks)
80             file_bio.close()
81
```

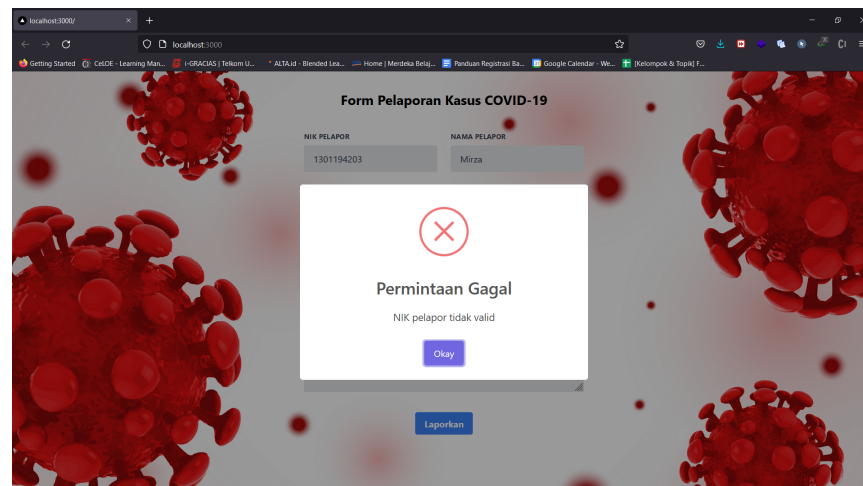
```
81
82     return response
83
84 server.register_function(report, 'report')
85 print("Server running on port", PORT_NUM)
86 server.serve_forever()
```

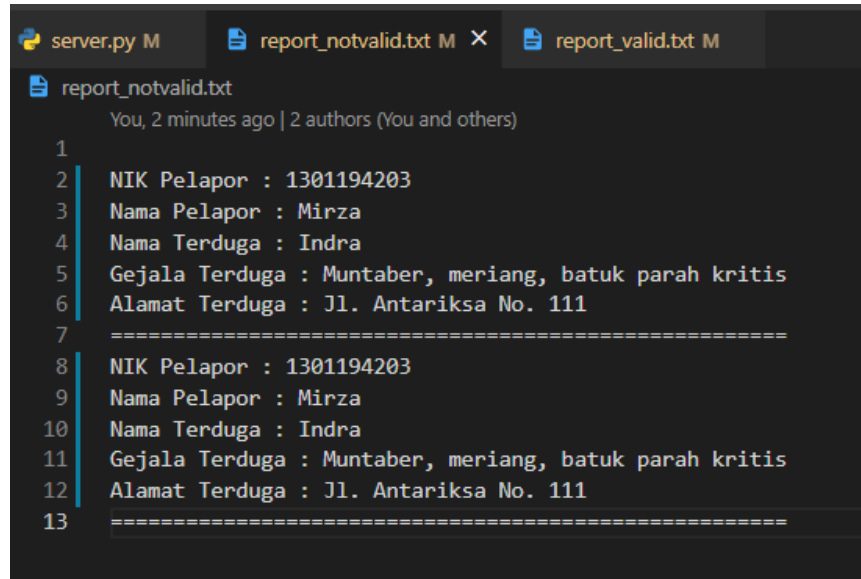
b. Screen Capture saat aplikasi dijalankan

Report Valid



Report Not Valid





The screenshot shows a code editor with three tabs: 'server.py M', 'report_notvalid.txt M', and 'report_valid.txt M'. The active tab is 'report_notvalid.txt', which displays the following text:

```
1  
2 NIK Pelapor : 1301194203  
3 Nama Pelapor : Mirza  
4 Nama Terduga : Indra  
5 Gejala Terduga : Muntaber, meriang, batuk parah kritis  
6 Alamat Terduga : Jl. Antariksa No. 111  
7 =====  
8 NIK Pelapor : 1301194203  
9 Nama Pelapor : Mirza  
10 Nama Terduga : Indra  
11 Gejala Terduga : Muntaber, meriang, batuk parah kritis  
12 Alamat Terduga : Jl. Antariksa No. 111  
13 =====
```

B. Link Video Dokumentasi

Presentasi :

https://drive.google.com/file/d/1jYu7qEAv35rFs-lIjp5e92i39zsywhpt/view?usp=s_haring

Demo : <https://youtu.be/KO4ogX90-Uc>