

## PERFORMANCE TEST REPORT UNTUK CONTACT LIST APP

### MATRIKS PERFORMANCE:

#### Load User

- **Normal Load:** Jumlah rata-rata users yang menggunakan app ini 10 users.
- **Heavy Load:** Jumlah maksimum users yang menggunakan app ini 200 users.

#### Response Time:

Level	Respons (Detik)	Deskripsi
Sangat Bagus	< 0.04	Sistem merespons sangat cepat, hampir seketika. User merasakan interaksi langsung.
Bagus	$\leq 0.04 - 2$	Sistem merespons dengan baik dan cepat. User merasakan sedikit penundaan, tetapi masih dalam batas toleransi
Cukup	2 - 6	Sistem merespons dengan cukup baik, tetapi users mungkin mengalami penundaan yang terlihat, yang dapat mengurangi kenyamanan.
Buruk	> 6	Sistem merespons lambat. User mungkin merasa frustrasi dan tidak puas.

## SKENARIO 1 : FITUR ACCOUNT

- Url : <https://thinking-tester-contact-list.herokuapp.com/>

Endpoint yang akan dites:

1. **POST** users/login
2. **POST** users
3. **GET** contacts
4. **PUT** contacts/
5. **PATCH** contacts/
6. **GET** contacts/
7. **DELETE** contacts/

Method	Path	Parameter	Body Request	Response Code
<b>POST</b>	<b>users/login</b>	Account ID	{ "email": "test2@fake.com", "password": "myNewPassword" }	200 OK: { "user": { "_id": "608b2db1add2691791c04c89", "firstName": "Updated", "lastName": "Username", "email": "test2@fake.com", "__v": 212 }, "token": "eyJhbGciOiJIUzI1NiIsInR5cGEiOiJpYXQiOjE2MDgyMWwzMDYyZmJiMjEzZTJhZDIhMjAiLCJpYXQiOiE2MTk3MzQ0NDI6L4CftGzYRmK04PJV6xKqmWWo9iOH2wlizEU8vk-L48LI" }
<b>POST</b>	<b>users</b>	Contact Data	{ "firstName": "John", "lastName": "Doe", "birthdate": "1970-01-01", "email": "jdoe@fake.com", "phone": "8005555555", "street1": "1 Main	201 CREATED: { "_id": "6085a221fcfc72405667c3d4", "firstName": "John", "lastName": "Doe", "birthdate": "1970-01-01", "email": "jdoe@fake.com", "phone": "8005555555", "street1": "1 Main St.",

			<pre> St.",   "street2": "Apartment A",   "city": "Anytown",   "stateProvince": "KS",   "postalCode": "12345",   "country": "USA" } </pre>	<pre> "street2": "Apartment A", "city": "Anytown", "stateProvince": "KS", "postalCode": "12345", "country": "USA", "owner": "6085a21efcfc72405667c3d4", "__v": 0 } </pre>
GET	contacts			<pre> 200 OK: [   {     "_id": "6085a221fcfc72405667c3d4",     "firstName": "John",     "lastName": "Doe",     "birthdate": "1970-01-01",     "email": "jdoe@fake.com",     "phone": "8005555555",     "street1": "1 Main St.",     "street2": "Apartment A",     "city": "Anytown",     "stateProvince": "KS",     "postalCode": "12345",     "country": "USA",     "owner": "6085a21efcfc72405667c3d4",     "__v": 0   } ] </pre>
PUT	contacts/	Contact ID	<pre> {   "firstName": "Amy",   "lastName": "Miller",   "birthdate": "1992- 02-02",   "email": "amiller@fake.com",   "phone": "8005554242",   "street1": "13 School St.",   "street2": "Apt. 5",   "city": "Washington",   "stateProvince": "QC",   "postalCode": "A1A1A1",   "country": "Canada" } </pre>	<pre> 200 OK: {   "_id": "6085a221fcfc72405667c3d4",   "firstName": "Amy",   "lastName": "Miller",   "birthdate": "1992-02-02",   "email": "amiller@fake.com",   "phone": "8005554242",   "street1": "13 School St.",   "street2": "Apt. 5",   "city": "Washington",   "stateProvince": "QC",   "postalCode": "A1A1A1",   "country": "Canada",   "owner": "6085a21efcfc72405667c3d4",   "__v": 0 } </pre>

<b>PATCH</b>	<b>contacts/</b>	Contact ID	{ "firstName": "Anna" }	200 OK: { "_id": "6085a221fcfc72405667c3d4", "firstName": "Anna", "lastName": "Miller", "birthdate": "1992-02-02", "email": "amiller@fake.com", "phone": "8005554242", "street1": "13 School St.", "street2": "Apt. 5", "city": "Washington", "stateProvince": "QC", "postalCode": "A1B2D4", "country": "Canada" "owner": "6085a21efcfc72405667c3d4", "__v": 0 }
<b>GET</b>	<b>contacts/</b>	Contact ID		200 OK: { "_id": "6085a221fcfc72405667c3d4", "firstName": "John", "lastName": "Doe", "birthdate": "1970-01-01", "email": "jdoe@fake.com", "phone": "8005555555", "street1": "1 Main St.", "street2": "Apartment A", "city": "Anytown", "stateProvince": "KS", "postalCode": "12345", "country": "USA", "owner": "6085a21efcfc72405667c3d4", "__v": 0 }
<b>DELETE</b>	<b>contacts/</b>	Contact ID		200 OK: Contact deleted

### 1.1. Test Plan Scenario – Load Testing

#### a. Tujuan:

Mengukur sejauh mana aplikasi dapat menangani beban user yang tinggi periode waktu tertentu.

#### b. Langkah-langkah:

Mensimulasikan request dimulai dari 10 user dan setting ramp-up period diawali dengan 2 detik.

**c. Specification:**

HOST\_URL: thinking-tester-contact-list.herokuapp.com

PATH:

- Path 1 : [POST] users/login
- Path 2 : [POST] users
- Path 3 : [GET] contacts
- Path 4 : [PUT] contacts/
- Path 5 : [PATCH] contacts/
- Path 6 : [GET] contacts/
- Path 7 : [DELETE] contacts/

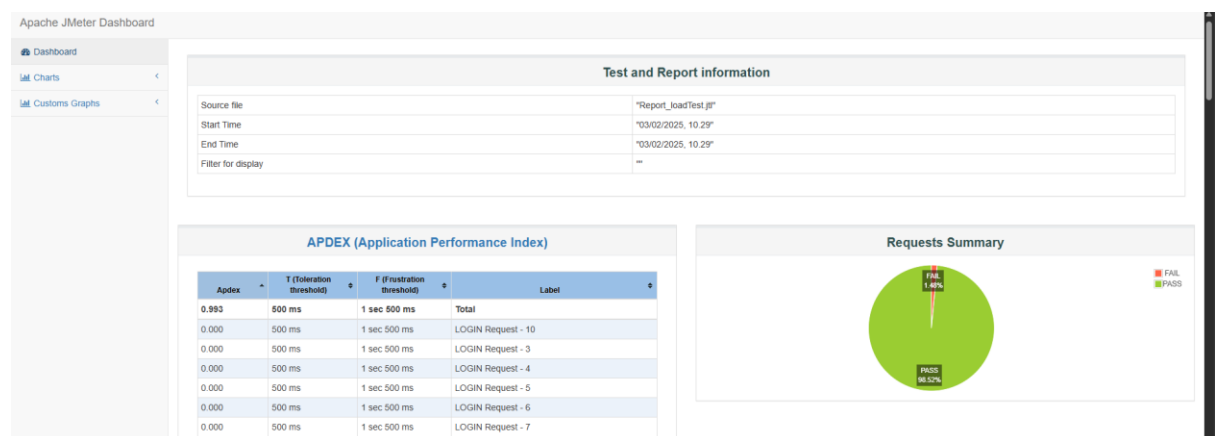
THREAD GROUP:

- Number of Thread (User) : 10
- Ramp-Up Period (in seconds) : 2
- Loop Count: 10

Deskripsi: Dari scenario diatas, maka setiap detik akan ada 5 thread dijalankan dan percobaan akan diulang sebanyak 10 kali pengulangan.

**d. Result:**

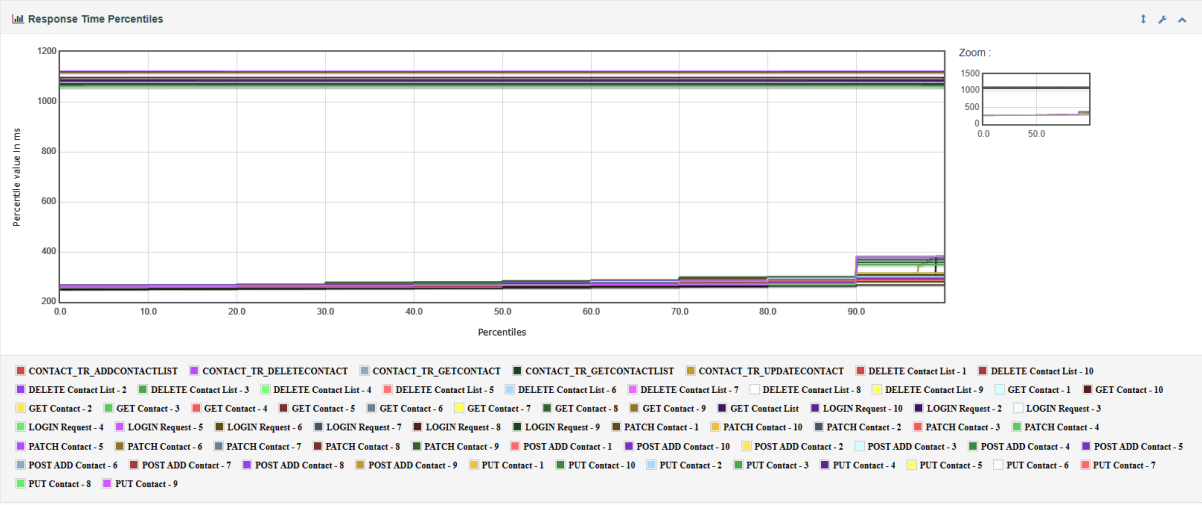
**Dashboard:**



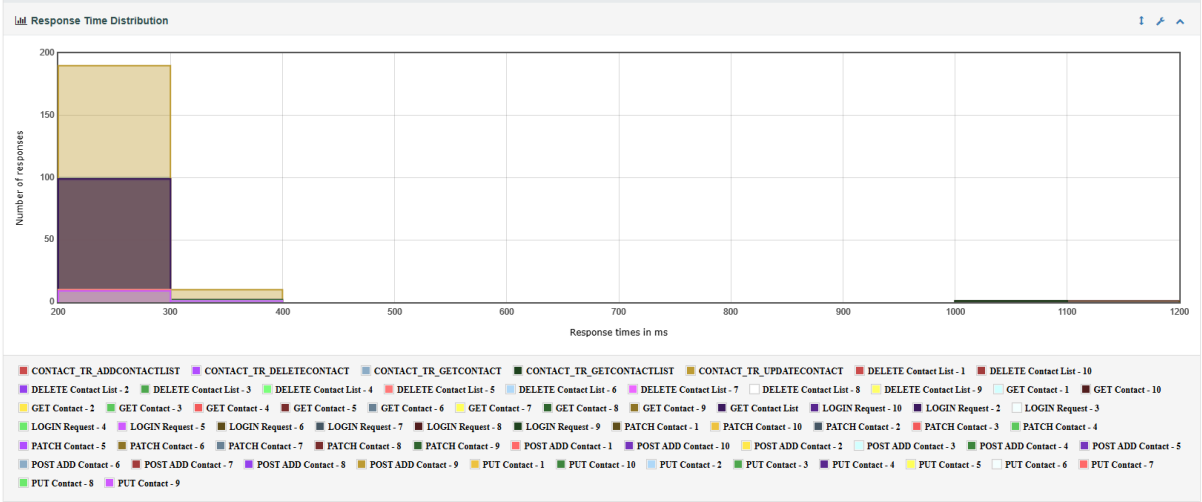
Statistics

Requests		Executions			Response Times (ms)							Throughput	Network (KB/sec)	
Label	~	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total		609	9	1.48%	283.21	256	1122	268.00	284.00	295.00	1071.80	32.13	48.16	17.35
CONTACT_TR_ADDCONTACTLIST		100	0	0.00%	272.14	262	315	269.00	284.00	290.75	314.81	5.99	6.24	4.39
CONTACT_TR_DELETECONTACT		100	0	0.00%	270.92	258	304	267.00	283.00	289.85	303.98	6.01	4.46	3.16
CONTACT_TR_GETCONTACT		100	0	0.00%	266.18	256	294	264.00	277.00	280.95	293.93	6.01	6.24	2.35
CONTACT_TR_GETCONTACTLIST		100	0	0.00%	269.26	257	383	266.50	280.90	287.95	382.07	6.00	24.62	2.20
CONTACT_TR_UPDATECONTACT		200	0	0.00%	274.25	256	383	269.00	287.00	300.90	380.99	11.80	12.33	7.34
DELETE Contact List - 1		10	0	0.00%	270.30	259	290	266.50	289.40	290.00	290.00	0.68	0.50	0.36
DELETE Contact List - 2		10	0	0.00%	271.00	265	302	267.50	299.10	302.00	302.00	0.67	0.50	0.35
DELETE Contact List - 3		10	0	0.00%	269.20	261	282	268.00	281.50	282.00	282.00	0.67	0.50	0.35
DELETE Contact List - 4		10	0	0.00%	272.20	265	290	268.00	289.10	290.00	290.00	0.67	0.50	0.35

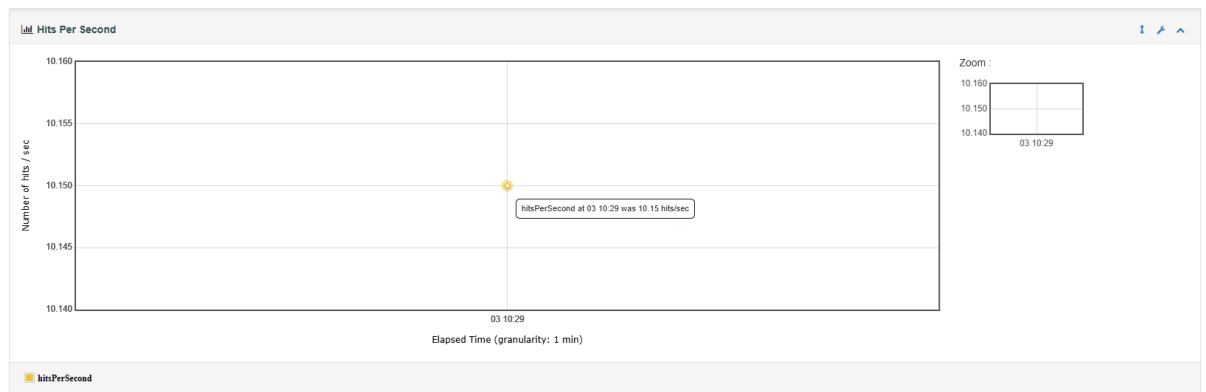
Response Time Percentiles:



Response Time Distribution:



## Throughput – Transaction Per Second:



### e. Kesimpulan dari hasil testing:

Dari hasil di atas kita dapat menarik kesimpulan bahwa dari semua thread yang dijalankan dalam waktu 2 detik (ramp-up), dibutuhkan waktu rata-rata eksekusi **283** milidetik (**0.28** detik) dengan waktu minimum yang dibutuhkan untuk mengeksekusi thread yaitu **155** milidetik (**0.15** detik) dan waktu maksimumnya **1122** milidetik (**1.12** detik). Dari throughput kita dapat melihat server mampu mengeksekusi 32 thread per time unit dan terdapat error sebanyak **1.48%**, dengan ini dapat disimpulkan bahwa server termasuk dalam kategori “**Bagus**” yang berarti server merespons dengan baik dan cepat sehingga user merasakan sedikit penundaan, tetapi masih dalam batas toleransi.

## 1.2. Test Plan Scenario – Stress Testing

### a. Tujuan:

Mengidentifikasi Batasan kapasitas dan mengevaluasi kinerja aplikasi di bawah tekanan ekstrem.

### b. Langkah-langkah:

Mensimulasikan tekanan penuh terhadap server dengan menjalankan 200 thread dalam ramp-up period 600 detik.

### c. Specification:

HOST\_URL: thinking-tester-contact-list.herokuapp.com

PATH:

- Path 1 : [POST] users/login
- Path 2 : [POST] users
- Path 3 : [GET] contacts
- Path 4 : [PUT] contacts/
- Path 5 : [PATCH] contacts/
- Path 6 : [GET] contacts/
- Path 7 : [DELETE] contacts/

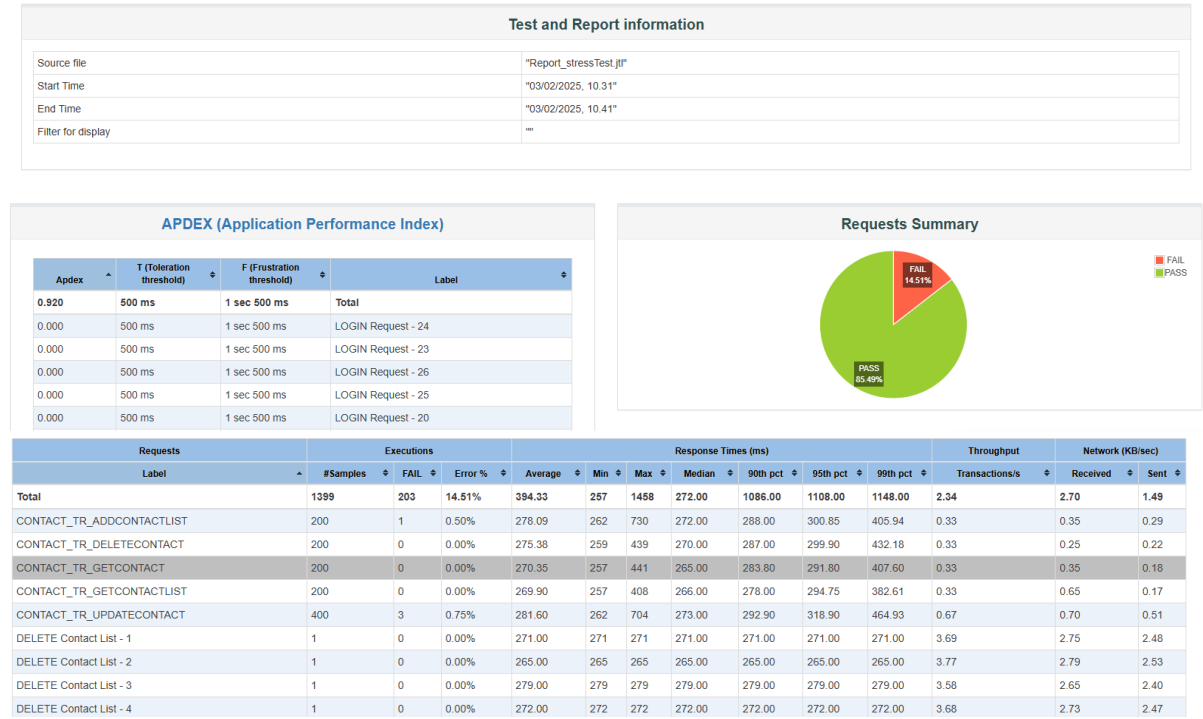
THREAD GROUP:

- Number of Thread (User) : 200
- Ramp-Up Period (in seconds) : 600
- Loop Count : 1

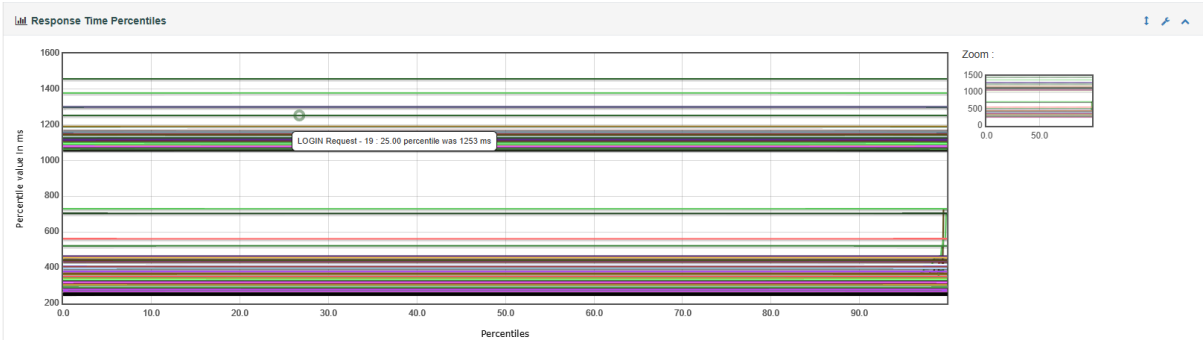
Deskripsi: Dari scenario diatas, maka thread akan dilakukan sebanyak 200 kali, dimana tiap thread dilakukan selama 3 detik sebelum berganti ke thread berikutnya.

d. Result:

Dashboard:

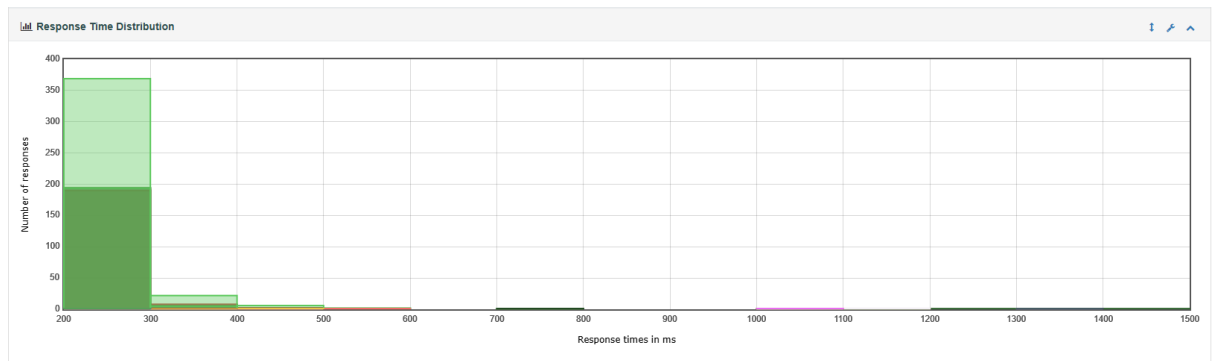


Response Time Percentiles:

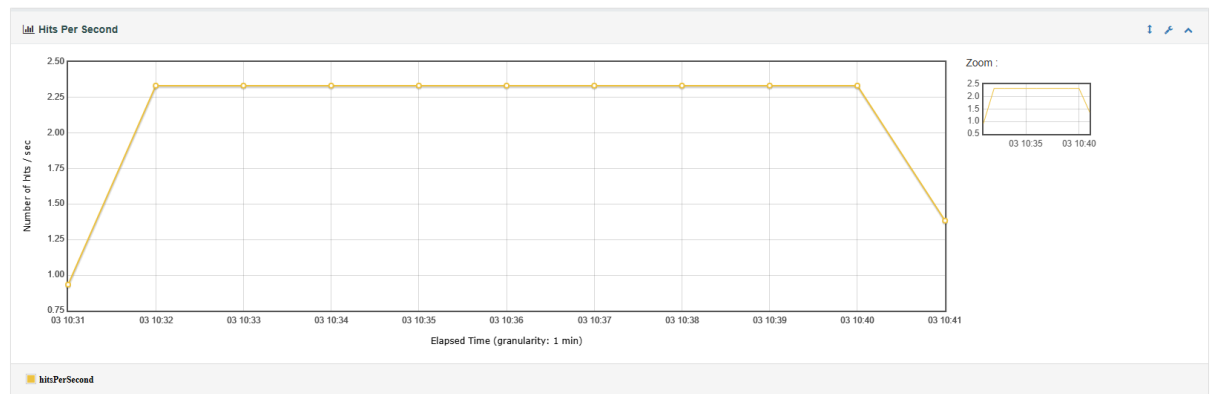


Response Time Distribution:





### Throughput – Transaction Per Second:



#### e. Kesimpulan dari hasil testing:

Dari hasil di atas kita dapat menarik kesimpulan bahwa dari semua thread yang dijalankan dalam waktu **600 detik** (ramp-up), dibutuhkan waktu rata-rata eksekusi **394 milidetik (0.39 detik)** dengan waktu minimum yang dibutuhkan untuk mengeksekusi thread yaitu **257 milidetik (0.25 detik)** dan waktu maksimumnya **1458 milidetik (1.14 detik)**. Dari throughput kita dapat melihat server mampu mengeksekusi **2.34 thread** per time unit dan terdapat error sebanyak **14.51%**, dengan ini dapat disimpulkan bahwa server termasuk dalam kategori “**Bagus**” yang berarti server merespons dengan baik dan cepat sehingga user merasakan sedikit penundaan, tetapi masih dalam batas toleransi.