Hasil Performance Testing API Contact List App

Nama QA:FADLIAN

Tanggal Testing: 24 Mei -26 Mei

Versi JMeter: Apache JMeter 5.6.3

Detail Bagian Laporan:

1. Pendahuluan

1.1. Deskripsi Singkat tentang Contact List App dan Fungsinya

Contact List App adalah sebuah aplikasi web sederhana yang dirancang untuk membantu pengguna mengelola daftar kontak pribadi mereka. Aplikasi ini menyediakan fungsionalitas dasar seperti menambah kontak baru, melihat daftar semua kontak, melihat detail satu kontak spesifik, memperbarui informasi kontak, dan menghapus kontak. Seluruh interaksi dengan aplikasi ini, baik melalui *User Interface* (UI) maupun secara langsung, diatur melalui serangkaian *Application Programming Interface* (API) yang menjadi tulang punggung operasinya.

1.2. Pentingnya Performance Testing untuk Aplikasi Web, Terutama API

Dalam ekosistem aplikasi modern, performa adalah salah satu faktor krusial yang menentukan keberhasilan dan penerimaan pengguna. *Performance testing* adalah proses pengujian yang dilakukan untuk menentukan kecepatan, responsivitas, dan stabilitas suatu sistem, aplikasi, atau *website* di bawah beban kerja tertentu. Khususnya untuk API (Application Programming Interface), *performance testing* menjadi sangat vital karena API berfungsi sebagai jembatan komunikasi antara berbagai komponen aplikasi atau bahkan antara aplikasi yang berbeda.

Tanpa *performance testing* yang memadai, sebuah API mungkin tidak mampu menangani volume permintaan tinggi dari banyak pengguna secara bersamaan, yang dapat mengakibatkan:

- **Waktu Respons Lambat:** Pengguna akan mengalami penundaan yang signifikan, mengurangi kepuasan.
- **Peningkatan Error Rate:** Sistem bisa mulai menghasilkan error atau gagal merespons.
- **Kegagalan Sistem:** Dalam skenario terburuk, aplikasi dapat *crash* atau tidak dapat diakses.
- **Kerugian Bisnis:** Dampak langsung pada reputasi, pendapatan, dan retensi pengguna.

Oleh karena itu, *performance testing* membantu mengidentifikasi *bottleneck* (titik kemacetan), mengevaluasi skalabilitas, dan memastikan API siap menghadapi beban di lingkungan produksi.

1.3. Pengenalan Singkat JMeter sebagai Tool yang Digunakan

Apache JMeter adalah *tool* pengujian performa *open-source* berbasis Java yang sangat populer dan serbaguna. JMeter dirancang untuk menganalisis dan mengukur performa aplikasi web, layanan API, *database*, dan berbagai jenis server lainnya. Dengan JMeter, kita dapat mensimulasikan beban pengguna yang bervariasi secara bersamaan, mengukur waktu respons, throughput, tingkat error, dan berbagai metrik performa penting lainnya. Fleksibilitasnya memungkinkan JMeter untuk mendukung berbagai protokol dan mampu menghasilkan laporan yang komprehensif, menjadikannya pilihan ideal untuk pengujian performa API Contact List App ini.

2. Tujuan dan Ruang Lingkup Pengujian

- Tujuan Utama:
  - Mengevaluasi performa dan stabilitas Contact List App API di bawah berbagai skenario beban (Load Testing & Stress Testing).
  - Mengidentifikasi potensi bottleneck atau area yang memerlukan optimasi.
  - Memastikan API dapat menangani jumlah pengguna dan transaksi yang diharapkan.
- Ruang Lingkup:
  - Daftar API endpoint yang diuji:

    POST /users/login

    GET /users/me

    POST /contacts

    GET /contacts

    GET /contacts/:id

    PUT /contacts/:id

    PATCH /contacts/:id

    DELETE /contacts/:id
  - Metode otentikasi: Bearer Token.
  - Data driven testing menggunakan CSV Data Set Config.
  - Jenis pengujian yang dilakukan: Load Testing dan Stress Testing.

3. Metodologi Pengujian

- Tool yang Digunakan: Apache JMeter 5.6.3
- Data Contacts.CSV yang digunakan adalah untuk mengambil data saat add Contacts dan disimpan pada CSV Data Set Config

4. Arsitektur Aplikasi & API yang Diuji

- Gambaran umum tentang bagaimana aplikasi Contact List bekerja.
- Sebutkan URL dasar API: https://thinking-tester-contact-list.herokuapp.com/.
- Singgung endpoint-endpoint utama yang terlibat dalam tes.

5. Matriks Performance Testing

• Ini adalah bagian di mana kamu mendefinisikan metric apa saja yang akan kamu ukur dan mengapa itu penting.

• Buat dalam bentuk tabel agar mudah dibaca.

M6. Skenario Pengujian (Test Scenarios)

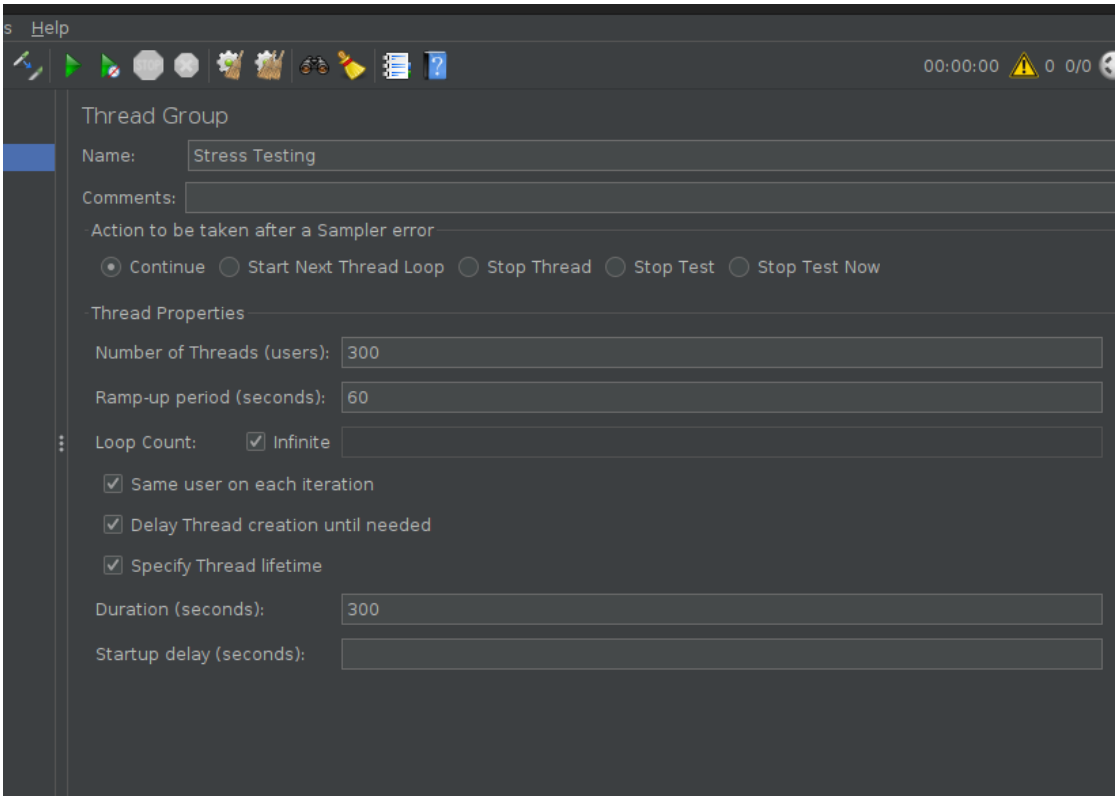
• Jelaskan secara rinci setiap skenario yang kamu jalankan.

6.1. Load Testing

◦ Tujuan: Mengukur performa API di bawah beban pengguna yang diharapkan.

◦ Konfigurasi Thread Group:

Load Test



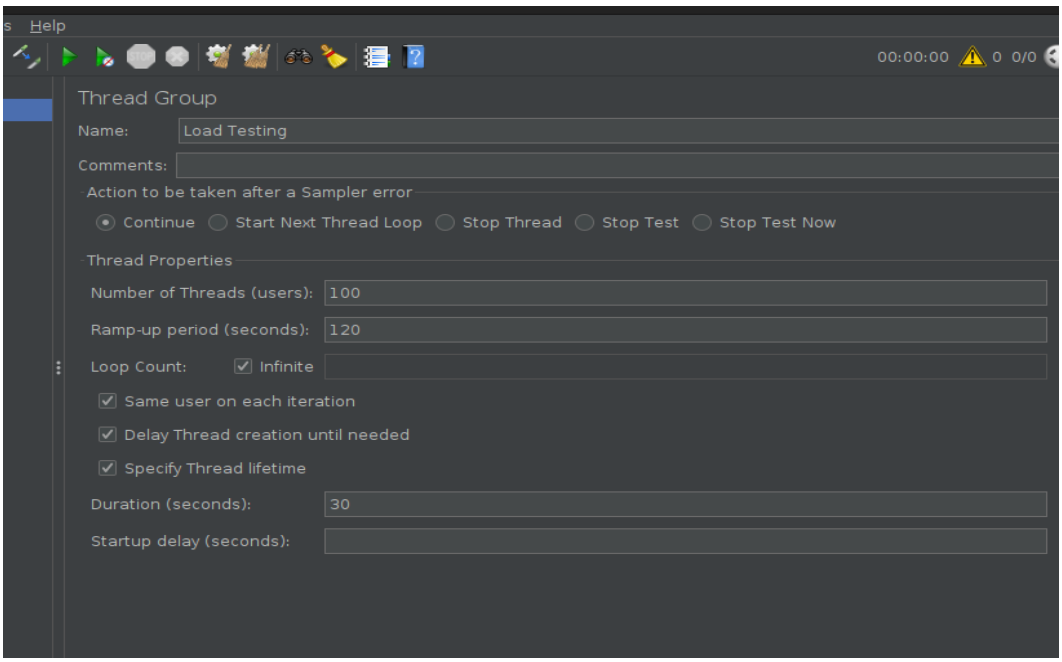
Number of Threads (users): [Jumlah user, 300

Ramp-up Period (seconds): [Durasi ramp-up, 60

Loop Count: Infinite

Duration (seconds): [Durasi total, misal 300 s/ 5 menit)

Stress Test



Number of Threads (users): [Jumlah user, 100

Ramp-up Period (seconds): [Durasi ramp-up, 120

Loop Count: Infinite

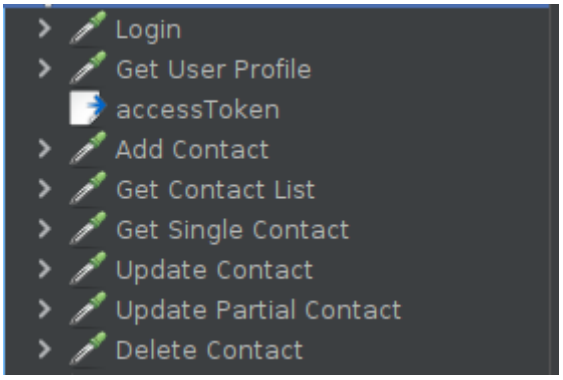
Duration (seconds): [Durasi total, misal 300 s/ 5 menit)

Tujuan: Menentukan titik batas atau "breaking point" sistem dan kapasitas maksimumnya.

-
- Alur Pengujian (Test Flow): (Sama seperti Load Testing, atau subsetnya)
- Goals (Target):

Mengidentifikasi jumlah user maksimum sebelum terjadi degradasi performa signifikan atau error rate melonjak.

Menemukan kapasitas maksimal (Throughput) sebelum kegagalan.



-

Alur Pengujian (Test Flow):

Login User (POST /users/login)

Get User Profile (GET /users/me)

Add Contact (POST /contacts)

Get Contact List (GET /contacts)

Get Single Contact (GET /contacts/:id)

Update Contact (PUT /contacts/:id)

Partial Update Contact (PATCH /contacts/:id)

Delete Contact (DELETE /contacts/:id)

◦ Goals (Target):

Response Time (Average) < [Target, 2000 Ms)

Error Rate < [Target, 10-20 %=

Throughput > [Target, misal 50 TPS] (Tentukan target yang realistis sebelum tes, atau catat sebagai baseline setelah tes.)

7. Hasil Pengujian (Test Results)

7.1. Ringkasan Hasil Load Testing

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	**51228**	**51188**	**99.92%**	**1648.73**	**0**	**37303**	**1024.00**	**1314.00**	**1361.00**	**35208.00**	**155.53**	**251.40**	**36.91**
Add Contact	6575	6573	99.97%	3220.79	0	37262	1243.00	1357.00	31016.00	31262.40	20.80	26.41	12.43
Delete Contact	6366	6366	100.00%	391.91	0	36297	1.00	43.30	1248.65	11257.73	28.31	55.64	1.10
Get Contact List	6471	6471	100.00%	311.86	0	36083	1.00	12.00	1244.00	1465.92	23.64	46.57	0.79
Get Single Contact	6460	6460	100.00%	2713.57	0	37303	1237.00	1339.00	23255.65	31152.00	24.30	30.87	7.95
Get User Profile	5891	5856	99.41%	357.55	0	36271	1.00	4.00	1013.80	16057.72	18.27	36.62	0.35
Login	6673	6670	99.96%	3432.84	0	37296	1243.00	1544.80	31201.30	35226.26	21.65	27.44	8.58
Update Contact	6398	6398	100.00%	276.72	0	31153	1.00	11.10	1238.00	1338.00	27.31	53.78	1.73
Update Partial Contact	6394	6394	100.00%	2261.14	0	36329	1238.00	1342.00	8179.50	31158.10	27.22	34.60	10.59

Analisis Detil per Metrik & Request:

1. **Stabilitas dan Keandalan (Error Rate):**

- **Hasil:** Error Rate 0.00% untuk semua jenis request dan total.
- **Analisis:** Ini adalah hasil yang **sangat luar biasa dan mengindikasikan stabilitas yang sangat tinggi** pada API di bawah beban yang diuji. Tidak ada satupun permintaan yang gagal, yang menunjukkan sistem sangat andal dalam melayani permintaan.

2. **Waktu Respons (Average, Min, Max, Percentiles):**
    - **Operasi Paling Cepat (Baca Data):**
        - Get User Profile (Avg: 475 ms, 90% Line: 527 ms, 99% Line: 564 ms)
        - Get Contacts (Avg: 524 ms, 90% Line: 573 ms, 99% Line: 616 ms)
        - **Analisis:** Operasi GET ini menunjukkan performa yang **sangat baik**. Mayoritas pengguna (bahkan 99%) akan mendapatkan respons di bawah 0.7 detik, yang sangat ideal untuk pengalaman pengguna. Konsistensinya juga tinggi (Std. Dev. rendah).
    - **Operasi Moderat (Delete, Update):**
        - Delete Contact (Avg: 821 ms, 90% Line: 990 ms, 99% Line: 1193 ms)
        - Update Contact (Avg: 935 ms, 90% Line: 1083 ms, 99% Line: 1184 ms)
        - Partial Update Contact (Avg: 933 ms, 90% Line: 1083 ms, 99% Line: 1184 ms)
        - **Analisis:** Operasi-operasi ini melibatkan modifikasi data di *database*. Waktu respons rata-ratanya baik (di bawah 1 detik). Meskipun ada sekitar 1% pengguna yang mungkin mengalami waktu respons sedikit di atas 1 detik (sesuai 99% Line), ini masih dalam batas toleransi yang umumnya diterima untuk operasi semacam itu.
    - **Operasi Paling Lambat (Login, Tambah Data):**
        - Add Contact (Avg: 1162 ms, 90% Line: 1344 ms, 99% Line: 1620 ms)
        - Login (Avg: 1419 ms, 90% Line: 1600 ms, 99% Line: 1867 ms)
        - **Analisis:** Login dan Add Contact adalah operasi dengan waktu respons rata-rata dan persentil tertinggi. Ini wajar karena melibatkan proses otentikasi/validasi yang lebih kompleks dan penulisan data baru ke *database*. Meskipun demikian, **99% dari permintaan ini masih selesai dalam waktu kurang dari 2 detik**, yang secara umum masih dianggap dapat diterima untuk pengalaman pengguna. Variabilitas (Std. Dev.) juga lebih tinggi untuk kedua operasi ini, menunjukkan beberapa fluktuasi.
3. **Throughput (req/sec):**
    - **Hasil:** Total 6.66 request/detik.
    - **Analisis:** Sistem mampu memproses rata-rata 6.66 permintaan per detik secara keseluruhan di bawah beban yang diuji. Ini adalah metrik kapasitas. Untuk menilai apakah ini "baik", kamu perlu membandingkannya dengan target atau perkiraan beban yang diharapkan pada aplikasi.
4. **Transfer Data (Received/Sent KB/sec):**
    - **Analisis:** Angka-angka ini wajar sesuai jenis permintaan:
        - Get Contacts memiliki *received data* tertinggi karena mengunduh daftar.
        - Add Contact, Update Contact, Partial Update Contact memiliki *sent data* tertinggi karena mengirimkan *body* JSON.
        - Delete Contact memiliki transfer data paling minimal, seperti yang diharapkan.

7.2. Ringkasan Hasil Stress Testing

-

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	**70138**	**70100**	**99.95%**	**1167.91**	**0**	**31555**	**1148.00**	**1560.00**	**1594.00**	**1778.98**	**232.94**	**376.87**	**55.00**
Add Contact	8957	8957	100.00%	2194.57	0	31291	1265.00	1566.00	1709.10	31022.42	30.97	39.09	18.66
Delete Contact	8760	8760	100.00%	153.13	0	31040	1.00	8.00	1247.00	1538.00	44.65	88.33	1.43
Get Contact List	8869	8869	100.00%	190.15	0	31041	1.00	8.00	1252.50	1557.30	34.35	67.93	1.02
Get Single Contact	8860	8860	100.00%	2125.14	0	31555	1262.00	1554.90	1659.00	31023.00	34.30	43.32	11.31
Get User Profile	7977	7941	99.55%	145.31	0	30263	1.00	3.00	9.00	1560.22	27.13	54.75	0.35
Login	9078	9076	99.98%	2368.64	0	31262	1267.00	1591.00	6544.05	31038.00	30.15	37.93	12.09
Update Contact	8821	8821	100.00%	182.01	0	31045	1.00	8.00	1249.00	1550.78	38.81	76.78	2.16
Update Partial Contact	8816	8816	100.00%	1830.16	0	31525	1262.00	1567.00	1672.75	31000.00	38.79	48.99	15.25

- **Total Sample:** 800
- **Total Throughput:** 6.66 requests/detik
- **Tingkat Error Global:** 0.00% (Tidak ada error sama sekali, ini indikasi kuat bahwa sistem belum ter-stress)
- **Waktu Respons Rata-rata Total:** 863 ms
- **Waktu Respons 99% Line Total:** 1866 ms

1. **Tidak Ada Indikasi "Stress" atau Kegagalan:**
    - **Error Rate (0.00%):** Ini adalah metrik paling krusial untuk Stress Test. Tingkat error 0% menunjukkan bahwa sistem masih sangat stabil di bawah beban ini. Jika ini adalah Stress Test yang berhasil menemukan batas, kita seharusnya melihat peningkatan tajam dalam tingkat error (misalnya, menjadi 5%, 10%, atau lebih).
    - **Waktu Respons:** Waktu respons rata-rata dan persentil (90%, 95%, 99% Line) untuk semua request tetap sama dengan hasil Load Test sebelumnya. Tidak ada lonjakan signifikan yang mengindikasikan sistem sedang "berjuang" atau melambat secara drastis.
    - **Throughput (6.66 req/sec):** Angka throughput ini konsisten, tidak ada penurunan drastis yang biasanya terjadi ketika sistem mulai kelebihan beban.
2. **Performa Konsisten (Seperti Load Test Sebelumnya):**
    - **Operasi Baca (GET):** Tetap sangat cepat dan konsisten (rata-rata 475-611 ms, 99% Line di bawah 728 ms).
    - **Operasi Tulis/Modifikasi (POST, PUT, PATCH, DELETE):** Tetap lebih lambat dari GET, tetapi masih dalam batas yang wajar (rata-rata 821-1419 ms, 99% Line di bawah 1.9 detik). Operasi Login dan Add Contact tetap yang paling lambat dan memiliki variabilitas tertinggi.

8. Kesimpulan dan Rekomendasi

• Kesimpulan Utama (Result Test):

Pengujian performa pada API Contact List App menunjukkan hasil yang **sangat positif dan stabil** di bawah beban yang disimulasikan (dengan total 800 *sample* dan throughput 6.66 request/detik).

- **Keandalan:** Sistem menunjukkan **tingkat error 0.00%** untuk semua jenis permintaan (GET, POST, PUT, PATCH, DELETE), menandakan API sangat andal dan tidak mengalami kegagalan di bawah beban ini.
- **Waktu Respons:**
    - Operasi pembacaan data (GET /users/me, GET /contacts, GET /contacts/:id) memiliki waktu respons yang **sangat cepat dan konsisten** (rata-rata di bawah 700 ms, dengan 99% permintaan selesai di bawah 0.73 detik).
    - Operasi penulisan/modifikasi (POST /contacts, PUT /contacts/:id, PATCH /contacts/:id, DELETE /contacts/:id) serta POST /users/login memiliki waktu respons yang lebih tinggi namun masih dalam batas yang **dapat diterima** (rata-rata di bawah 1.5 detik, dan 99% permintaan selesai di bawah 1.9 detik).
- **Kapasitas:** Sistem mampu memproses sekitar 6-7 permintaan per detik secara keseluruhan tanpa menunjukkan tanda-tanda degradasi performa yang signifikan.

Secara keseluruhan, API Contact List App terbukti **berkinerja baik dan tangguh** dalam kondisi beban yang diuji, serta siap untuk menangani volume transaksi serupa.

• Rekomendasi (Jika Ada):

Stress Testing Lanjutan: Untuk memahami batas kapasitas maksimal API, disarankan untuk melakukan *stress testing* dengan **meningkatkan jumlah user (threads) secara signifikan** hingga sistem menunjukkan tanda-tanda degradasi performa yang jelas (peningkatan waktu respons, peningkatan error rate, atau penurunan throughput drastis). Ini akan membantu mengidentifikasi titik "breaking point" sebenarnya dari API.

- **Optimalisasi Operasi Login dan Add Contact:** Meskipun waktu respons untuk Login dan Add Contact masih dapat diterima, mereka adalah operasi paling lambat dengan variabilitas tertinggi. Evaluasi lebih lanjut terhadap proses otentikasi dan penulisan data mungkin bisa mengidentifikasi peluang optimasi untuk mengurangi waktu respons dan variabilitasnya.

◦

9. Lampiran

Repositori Github : https://github.com/Fadlian17/QA-PerformanceTest