







Outline

- Soal
- Hasil
- Report
- Kesimpulan

1. Soal





Task 5

```
    API Create
```

```
    Base url: <a href="https://regres.in">https://regres.in</a>
    Path url: /api/users
    Method: POST
    Header: application/json
    Request body:
        <a href="mailto:">k</a>
    "name": "morpheus",
            "job": "leader"
        <a href="mailto:">leader"</a>
    **
```

API Update

```
Base url: https://reqres.in
Path url: /api/users/2
Method: PUT
Header: application/json
Request body:
{
    "name": "morpheus",
    "job": "zion resident"
}
```

Task yang dikerjakan:

- Buat scenario test untuk integration test dari 2 API tersebut dan implementasikan ke dalam k6 serta tambahkan assertion dari tiap test yang dilakukan dalam 1 file test
- Buat scenario test untuk menguji performance dari 2
 API tersebut dengan total 1000 virtual user, 3500
 iterasi dan batas maksimum toleransi response APInya
 2 second serta tambahkan assertion dari tiap test
 yang dilakukan dalam 1 file test
- Generate report dalam bentuk HTML dari soal no. 2 untuk mendapatkan visualisasi dari performance test yang dilakukan





2. Hasil

https://drive.google.com/drive/folder s/1euAb3rD2OyEgGgmrqr_J_OVIO5jX qqE1?usp=sharing





3. Report

```
C:\Windows\system32\cmd.exe
D:\K6>k6 run check.js
  execution: local
     script: check.js
     output: -
  scenarios: (100.00%) 1 scenario, 1000 max VUs, 32s max duration (incl. graceful stop):
           * default: 3500 iterations shared among 1000 VUs (maxDuration: 2s, gracefulStop: 30s)
     Create Data
       √ response code was 201
     Update Data
```

Pada hasil terlihat bahwa pada Create Data hasil response sesuai dengan ekspetasi yaitu '201 OK'. Sedangkan, pada Update Data response code menjadi merah, dikarenakan pada test yang dilakukan sebelumnya di postman, hasil report pada update/PUT menunjukan response '404 not found' sedangkan response yang diinginkan adalah '200 OK'





```
checks...... 50.00% √ 999
                                             X 999
data received...... 12 MB 384 kB/s
data sent..... 644 kB 20 kB/s
dropped_iterations...... 2500 78.099426/s
group_duration..... avg=3.79s
                                        min=362.25ms med=2.65s
                                                              max=17.81s
                                                                         p(90)=8.38s
                                                                                     p(95)=9.56s
http_req_blocked..... avg=937.13ms_min=0s
                                                                                     p(95)=5.47s
                                                   med=0s
                                                              max=15.39s
                                                                         p(90)=3.49s
http req connecting...... avg=50.65ms min=0s
                                                              max=300.12ms p(90)=180.7ms p(95)=217.9ms
                                                   med=0s
http_req_duration..... avg=1.83s
                                       min=14.71ms med=1.17s
                                                              max=16.97s
                                                                         p(90)=4.65s
                                                                                     p(95)=6.03s
  { expected response:true }...: avg=1.37s
                                                              max=16.97s
                                                                         p(90)=3.44s
                                                                                     p(95)=4.61s
                                       min=14.71ms med=1.09s
http req failed...... 33.33% √ 999
                                             X 1998
http req receiving..... avg=421.24ms min=0s
                                                                                     p(95)=2.96s
                                                   med=2.96ms
                                                              max=9.44s
                                                                         p(90)=1.57s
http req sending..... avg=34.03µs min=0s
                                                   med=0s
                                                              max=1.93ms
                                                                         p(90)=0s
                                                                                     p(95)=393.05µs
http_req_tls_handshaking.....: avg=877.24ms min=0s
                                                                                     p(95)=5.21s
                                                   med=0s
                                                              max=15.13s
                                                                         p(90)=3.28s
http req waiting..... avg=1.41s min=12.62ms med=935.88ms max=16.97s
                                                                         p(90)=3.41s
                                                                                     p(95)=4.43s
```

Lalu pada hasil yang bertandakan X (silang), dikarenakan request untuk threshold yg diinginkan adalah sebagai berikut :

- http_req_failed: ['rate<0.01'], yang artinya https error harus kurang dari 1%. Sedangkan hasil
 menunjukan eror lebih dari 1% (rata-rata = 1.37s yang seharusnya rata-rata = 1.83s)
- http_req_duration: ['p(95)<200']. //95% permintaan memiliki waktu respons di bawah 200ms. Durasi yang diinginkan adalah 2s, tetapi ternyata hasil durasi yang diinginkan jauh lebih lama dari 2s.





4. Report with Github

```
C:\Windows\system32\cmd.exe
D:\K6>k6 run check2.js
 execution: local
   script: check2.js
   output: -
 scenarios: (100.00%) 1 scenario, 1000 max VUs, 32s max duration (incl. graceful stop):
        * default: 3500 iterations shared among 1000 VUs (maxDuration: 2s, gracefulStop: 30s)
INFO[0018] [k6-reporter v2.3.0] Generating HTML summary report source=console
running (17.4s), 0000/1000 VUs, 1000 complete and 0 interrupted iterations
```

Pada hasil terlihat tulisan (default x), dikarenakan pada PUT/Update Data response yang ditunjukkan adalah '404 not found', sedangkan expected nya itu '200 OK'





4. Kesimpulan

Berdasarkan hasil pengerjaan yang telah dilakukan, maka dapat diambil kesimpulan diantaranya yaitu :

- 1. Pada pembuatan scenario test dapat menentukan standar kualitas yang harus dipenuni (ekspetasi). Selain ekspetasi, dari scenario test ini juga dapat menentukan kekurangan pada produk software. Kekurangan ini bisa dikembangkan sesuai dengan standar yang telah ditetapkan.
- 2. Agar sistem pada software berjalan dengan optimal tanpa adanya masalah yang akan dialami, maka harus dilakukan pengujian/testing. Untuk menguji kelayakan sistem yang akan digunakan.

