

**Load Balanced Web API dengan Multi-Container Docker (Backend,  
Load Balancer dan Redis Database)**



Disusun oleh:

Alayavaro Rachmadia	2410501095
Panji Anugerah Panengah	2410501105
Rakha Abyan Hertamtama	2410501089
Rizki Ramadhan	2410501091
Shidqi Athalla Arka Qafriyanto	2410501077

**Jalan RS. Fatmawati Raya, Pd. Labu, Kec. Cilandak, Kota Jakarta Selatan,**

**Daerah Khusus Ibukota Jakarta 12450**

**D3 Sistem Informasi**

## 1. Deskripsi Arsitektur

Project ini membangun sebuah sistem *multi-container* menggunakan Docker Compose yang terdiri dari empat service utama:

1. **Backend1 (Flask API instance 1)**
2. **Backend2 (Flask API instance 2)**
3. **Redis Database** untuk penyimpanan data (*total\_visits*)
4. **Nginx Reverse Proxy** berfungsi sebagai load balancer

Ketika pengguna mengakses <http://localhost:8080>, request akan masuk ke Nginx → didistribusikan secara bergantian (*round-robin*) ke backend1 atau backend2. Setiap instance backend mencatat jumlah kunjungan ke Redis, sehingga data tetap tersimpan meskipun container backend dihentikan atau dihapus.

## 2. Diagram Arsitektur

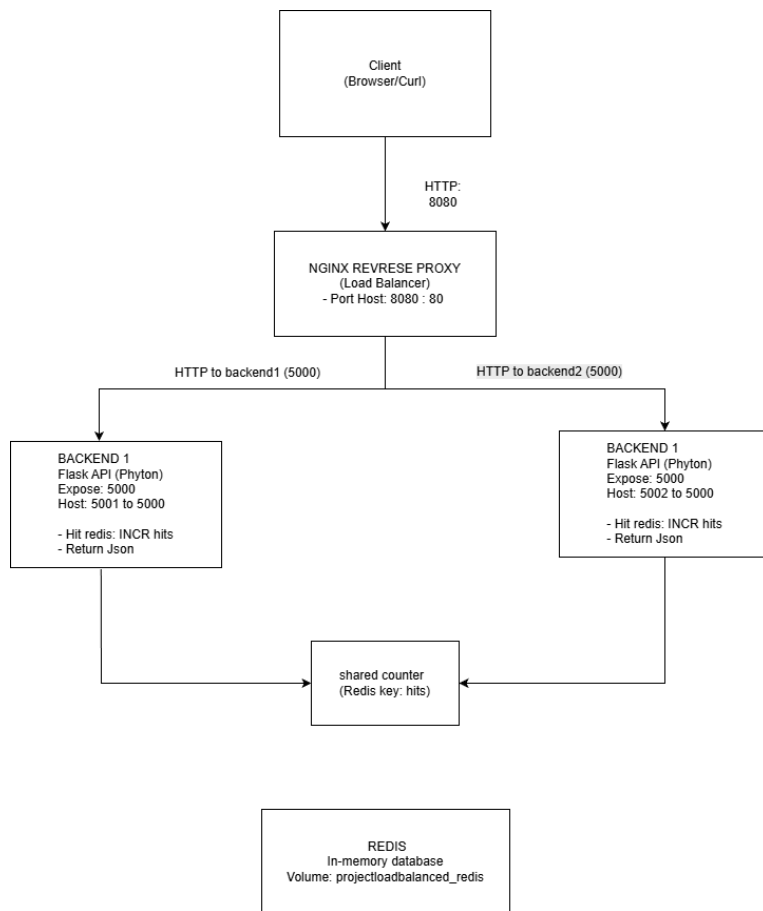


Diagram tersebut menunjukkan arsitektur Load Balanced Web API di mana client mengirim request ke Nginx (port 8080) sebagai load balancer. Nginx kemudian membagi request secara bergantian ke dua backend Flask API (Backend 1 dan Backend 2). Setiap backend mengakses Redis untuk menambah nilai counter (hits) yang digunakan bersama (shared). Hasilnya, beban kerja terbagi merata, performa lebih stabil, dan data tetap tersinkron antar backend.

### 3. Penjelasan Dockerfile

```
Dockerfile X
C: > ProjectLoadBalanced > Dockerfile > ...
1 FROM python:3.9-slim
2
3 WORKDIR /app
4
5 # Copy requirements dan install library
6 COPY requirements.txt .
7 RUN pip install --no-cache-dir -r requirements.txt
8
9 # Copy source code
10 COPY src/ .
11
12 EXPOSE 5000
13
14 CMD ["python", "app.py"]
15
```

Penjelasan:

- **Base image:** python:3.9-slim (ringan & stabil)
- **WORKDIR /app:** lokasi file aplikasi
- **COPY src/:** menyalin app.py
- **Install dependency:** Flask & Redis client
- **Expose 5000:** port API
- **CMD:** menjalankan Flask server

Dockerfile ini digunakan untuk membangun image backend1 dan backend2.

## 4. Penjelasan docker-compose.yml

```
Dockerfile X docker-compose.yml X
C: > ProjectLoadBalanced > docker-compose.yml
1  version: '3.8'
2
3  services:
4    >Run Service
5    redis:
6      image: redis:alpine
7      container_name: redis
8      ports:
9        - "6379:6379"
10     volumes:
11       - redis_data:/data
12
13     >Run Service
14     backend1:
15       build: ./src
16       container_name: backend1
17       ports:
18         - "5001:5000"
19       environment:
20         - REDIS_HOST=redis
21       depends_on:
22         - redis
23
24     >Run Service
25     backend2:
26       build: ./src
27       container_name: backend2
28       ports:
29         - "5002:5000"
30       environment:
31         - REDIS_HOST=redis
32       depends_on:
33         - redis
34
35     >Run Service
36     nginx:
37       image: nginx:alpine
38       container_name: nginx
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

## Penjelasan Service

Service	Fungsi
redis	database penyimpanan total kunjungan
backend1	instance API Flask pertama

**backend2** instance API Flask kedua

**nginx** reverse proxy + load balancer

## Port Mapping

- Backend1 → localhost:5001 → port internal 5000
- Backend2 → localhost:5002 → port internal 5000
- Redis → localhost:6379
- Nginx → localhost:8080

## Volume

- redis\_data:/data (persistent storage)

## Network

Semua service berada dalam 1 internal network bernama mynet.

## 5. Cara Menjalankan Project

Build Image Backend:

```
PS C:\ProjectLoadBalanced> docker build -t backend-image .
[+] Building 3.3s (11/11) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile               0.0s
=> => transferring dockerfile: 275B                               0.0s
=> [internal] load metadata for docker.io/library/python:3.9-slim 2.9s
=> [auth] library/python:pull token for registry-1.docker.io     0.0s
=> [internal] load .dockerignore                                  0.0s
=> => transferring context: 2B                                      0.0s
=> [1/5] FROM docker.io/library/python:3.9-slim@sha256:2d97f6910b16bd338d3060f261f53f144965f755599aaba1acda1e13cf1731b1b 0.0s
=> => resolve docker.io/library/python:3.9-slim@sha256:2d97f6910b16bd338d3060f261f53f144965f755599aaba1acda1e13cf1731b1b 0.0s
=> [internal] load build context                                  0.0s
=> => transferring context: 90B                                     0.0s
=> CACHED [2/5] WORKDIR /app                                       0.0s
=> CACHED [3/5] COPY requirements.txt .                             0.0s
=> CACHED [4/5] RUN pip install --no-cache-dir -r requirements.txt 0.0s
=> CACHED [5/5] COPY src/ .                                        0.0s
=> exporting to image                                              0.1s
=> => exporting layers                                              0.0s
=> => exporting manifest sha256:1abeb6834327d5ff622e70892aef1960f9cce76e6219deb965f6544a62fd7309 0.0s
=> => exporting config sha256:4fccfb993fb84e0cf3502db3c18a1c5ccae714949c4599d0940efb5bbc7106ab 0.0s
=> => exporting attestation manifest sha256:448a10b7a84ed0a02f65bf03dbc1403ddfdc07e9de71058b86b6180c6a6e31df 0.0s
=> => exporting manifest list sha256:4416b3e0f2dd9b3c47d3547ab098302254de889723ebd10328f92d12486d8075 0.0s
=> => naming to docker.io/library/backend-image:latest            0.0s
=> => unpacking to docker.io/library/backend-image:latest         0.0s
PS C:\ProjectLoadBalanced>
```

Docker Compose Up -d:

```
PS C:\ProjectLoadBalanced> docker compose up -d
time="2025-12-03T21:40:36+07:00" level=warning msg="C:\\ProjectLoadBalanced\\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 5/5
✓ Network projectloadbalanced_default Created 0.1s
✓ Container redis Started 0.8s
✓ Container backend1 Started 1.0s
✓ Container backend2 Started 1.0s
✓ Container nginx Started 1.2s
PS C:\ProjectLoadBalanced>
```

Cek Kontainer Proses:

```
PS C:\ProjectLoadBalanced> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
	NAMES				
de5e21e7597d	nginx:alpine	"/docker-entrypoint. ...."	32 seconds ago	Up 30 seconds	0.0.0.0:8080->80/tcp, [::]:80
80->80/tcp	nginx				
fe65b4364c13	projectloadbalanced-backend2	"python app.py"	32 seconds ago	Up 30 seconds	0.0.0.0:5002->5000/tcp, [::]:
5002->5000/tcp	backend2				
d6f53c5499a2	projectloadbalanced-backend1	"python app.py"	32 seconds ago	Up 30 seconds	0.0.0.0:5001->5000/tcp, [::]:
5001->5000/tcp	backend1				
8dec3b4ad075	redis:alpine	"docker-entrypoint.s..."	32 seconds ago	Up 31 seconds	0.0.0.0:6379->6379/tcp, [::]:
6379->6379/tcp	redis				

```
PS C:\ProjectLoadBalanced>
```

Log nginx:

```
PS C:\ProjectLoadBalanced> docker compose logs nginx
time="2025-12-03T21:41:43+07:00" level=warning msg="C:\\ProjectLoadBalanced\\docker-compose.yml: the attribute `version` is obsolete,
it will be ignored, please remove it to avoid potential confusion"
nginx | /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
nginx | /docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
nginx | /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
nginx | 10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
nginx | 10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
nginx | /docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
nginx | /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
nginx | /docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
nginx | /docker-entrypoint.sh: Configuration complete; ready for start up
PS C:\ProjectLoadBalanced>
```

Test Api Via Load Balancer:

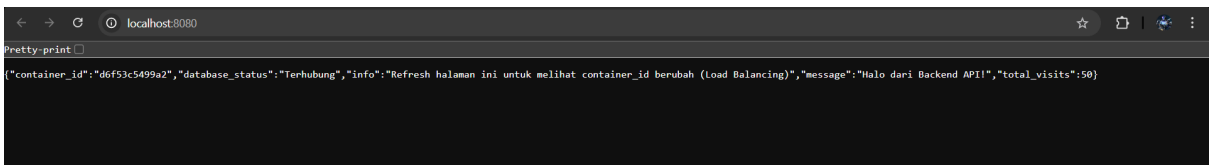
```
PS C:\ProjectLoadBalanced> curl http://localhost:8080

StatusCode      : 200
StatusDescription : OK
Content         : {"container_id":"d6f53c5499a2","database_status":"Terhubung","info":"Refresh halaman ini untuk melihat
container_id berubah (Load Balancing)","message":"Halo dari Backend API!","total_visits":46}

RawContent      : HTTP/1.1 200 OK
                  Connection: keep-alive
                  Content-Length: 196
                  Content-Type: application/json
                  Date: Wed, 03 Dec 2025 14:42:26 GMT
                  Server: nginx/1.29.3

Forms           : {"container_id":"d6f53c5499a2","database_stat...
Headers         : {[Connection, keep-alive], [Content-Length, 196], [Content-Type, application/json], [Date, Wed, 03 Dec 2025
                  14:42:26 GMT]...}
Images          : {}
InputFields     : {}
Links           : {}
ParsedHtml      : mshtml.HTMLDocumentClass
RawContentLength : 196
```

6. Screenshot Hasil Berjalan



```

backend2 | * Serving Flask app 'app'
backend2 | * Debug mode: off
backend2 | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
backend2 | * Running on all addresses (0.0.0.0)
backend2 | * Running on http://127.0.0.1:5000
backend2 | * Running on http://172.22.0.4:5000
backend2 | Press CTRL+C to quit
backend2 | 172.22.0.5 - - [03/Dec/2025 14:42:45] "GET / HTTP/1.0" 200 -
backend2 | 172.22.0.5 - - [03/Dec/2025 14:42:49] "GET / HTTP/1.0" 200 -
backend1 | * Serving Flask app 'app'
backend1 | * Debug mode: off
backend1 | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
backend1 | * Running on all addresses (0.0.0.0)
backend1 | * Running on http://127.0.0.1:5000
backend1 | * Running on http://172.22.0.3:5000
backend1 | Press CTRL+C to quit
backend1 | 172.22.0.5 - - [03/Dec/2025 14:42:26] "GET / HTTP/1.0" 200 -
backend1 | 172.22.0.5 - - [03/Dec/2025 14:42:48] "GET / HTTP/1.0" 200 -
backend1 | 172.22.0.5 - - [03/Dec/2025 14:42:50] "GET / HTTP/1.0" 200 -
redis | Starting Redis Server
redis | 1:C 03 Dec 2025 14:40:37.447 * o000o000o000o Redis is starting o000o000o000o
redis | 1:C 03 Dec 2025 14:40:37.447 * Redis version=8.4.0, bits=64, commit=00000000, modified=1, pid=1, just started
redis | 1:C 03 Dec 2025 14:40:37.447 * Configuration loaded
redis | 1:M 03 Dec 2025 14:40:37.448 * monotonic clock: POSIX clock_gettime
redis | 1:M 03 Dec 2025 14:40:37.516 * <ReJSON> version: 80400 git sha: unknown branch: unknown
redis | 1:M 03 Dec 2025 14:40:37.516 * <ReJSON> Exported RedisJSON_V1 API
redis | 1:M 03 Dec 2025 14:40:37.517 * <ReJSON> Exported RedisJSON_V2 API
redis | 1:M 03 Dec 2025 14:40:37.517 * <ReJSON> Exported RedisJSON_V3 API
redis | 1:M 03 Dec 2025 14:40:37.517 * <ReJSON> Exported RedisJSON_V4 API
redis | 1:M 03 Dec 2025 14:40:37.517 * <ReJSON> Exported RedisJSON_V5 API
redis | 1:M 03 Dec 2025 14:40:37.517 * <ReJSON> Exported RedisJSON_V6 API
redis | 1:M 03 Dec 2025 14:40:37.517 * <ReJSON> Enabled diskless replication
redis | 1:M 03 Dec 2025 14:40:37.517 * <ReJSON> Initialized shared string cache, thread safe: true.
redis | 1:M 03 Dec 2025 14:40:37.517 * Module 'ReJSON' loaded from /usr/local/lib/redis/modules//rejson.so
redis | 1:M 03 Dec 2025 14:40:37.517 * <search> Acquired RedisJSON_V6 API
redis | 1:M 03 Dec 2025 14:40:37.519 * Server initialized
redis | 1:M 03 Dec 2025 14:40:37.520 * <search> Loading event starts
redis | 1:M 03 Dec 2025 14:40:37.520 * <search> Changing workers threadpool size from 0 to 4
redis | 1:M 03 Dec 2025 14:40:37.520 * <search> Enabled workers threadpool of size 4
redis | 1:M 03 Dec 2025 14:40:37.521 * Loading RDB produced by version 8.4.0
redis | 1:M 03 Dec 2025 14:40:37.521 * RDB age 43 seconds
redis | 1:M 03 Dec 2025 14:40:37.521 * RDB memory usage when created 1.12 Mb
redis | 1:M 03 Dec 2025 14:40:37.521 * Done loading RDB, keys loaded: 1, keys expired: 0.
redis | 1:M 03 Dec 2025 14:40:37.521 * <search> Changing workers threadpool size from 4 to 0
redis | 1:M 03 Dec 2025 14:40:37.522 * <search> Disabled workers threadpool of size 4
redis | 1:M 03 Dec 2025 14:40:37.522 * <search> Loading event ends
redis | 1:M 03 Dec 2025 14:40:37.522 * DB loaded from disk: 0.002 seconds
redis | 1:M 03 Dec 2025 14:40:37.522 * Ready to accept connections tcp

```

Log di atas menunjukkan bahwa seluruh sistem **Nginx + Backend1 + Backend2 + Redis** telah berjalan dengan benar dan saling terhubung di dalam Docker Compose. Nginx berhasil memuat konfigurasi load balancer dan menerima request dari host (172.22.0.1), lalu meneruskannya bergantian ke Backend1 dan Backend2, terlihat dari log Flask yang menerima request bergantian.. Kedua backend berjalan pada alamat internal container (172.22.0.3 dan 172.22.0.4) serta merespons permintaan melalui port 5000 tanpa error. Redis juga berhasil berjalan dalam mode standalone dengan modul-modul tambahan aktif, menunjukkan bahwa database siap menerima dan menyimpan data secara persisten melalui volume yang telah dipasang.

## 7. Kendala dan Solusi

### Kendala 1 Nginx 502 Bad Gateway



- Penyebab: backend berjalan di port 3000
- Solusi: mengubah app.run(... port=5000) sesuai dengan compose & Nginx

## **Kendala 2 Load balancer tidak merespons**

Penyebab: salah upstream port di default.conf

Solusi: sesuaikan ke backend1:5000 dan backend2:5000

## **Kendala 3 Data tidak persistent**

Penyebab: Redis belum memakai volume

Solusi: menambahkan

volumes:

- redis\_data:/data
  - Data total\_visits menjadi tetap setelah restart.

## **Kendala 4 Data tidak persistent**

Request tidak bergantian ke Backend1/Backend2

Solusi: Memakai round\_robin default pada Nginx upstream.

