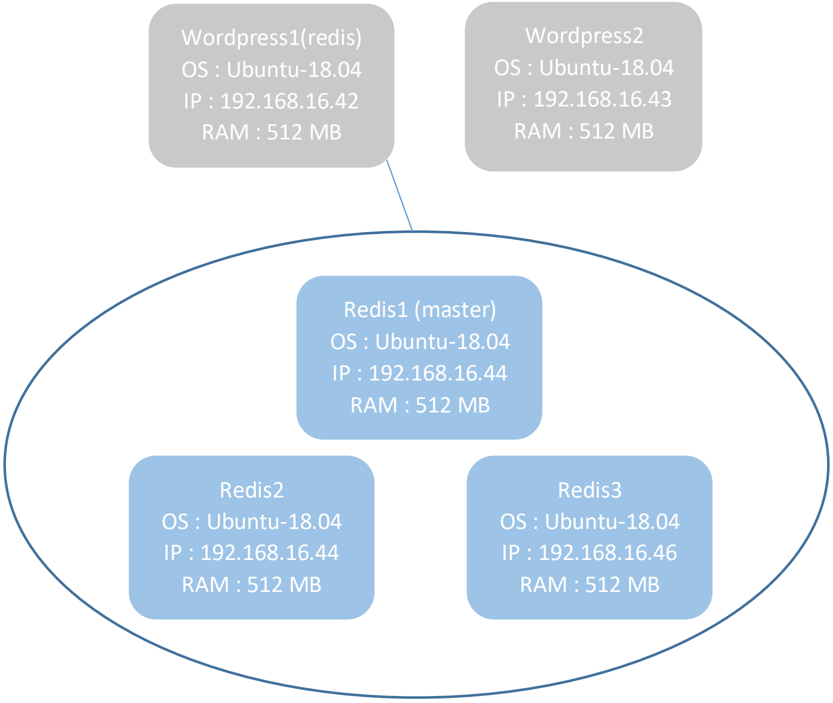
Laporan Redis Cluster

1. Implementasi Redis Cluster

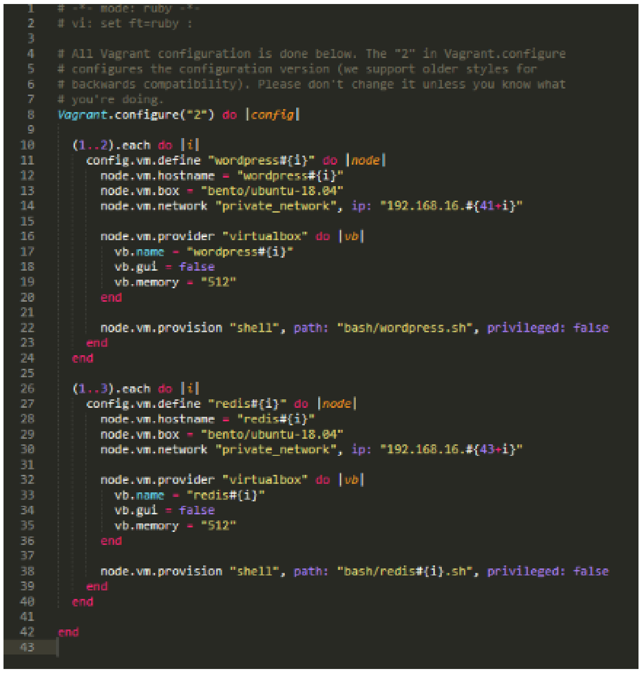
a. Arsitektur Server



b. Implementasi pada Vagrant

i. Membuat vagrant file : vagrant init

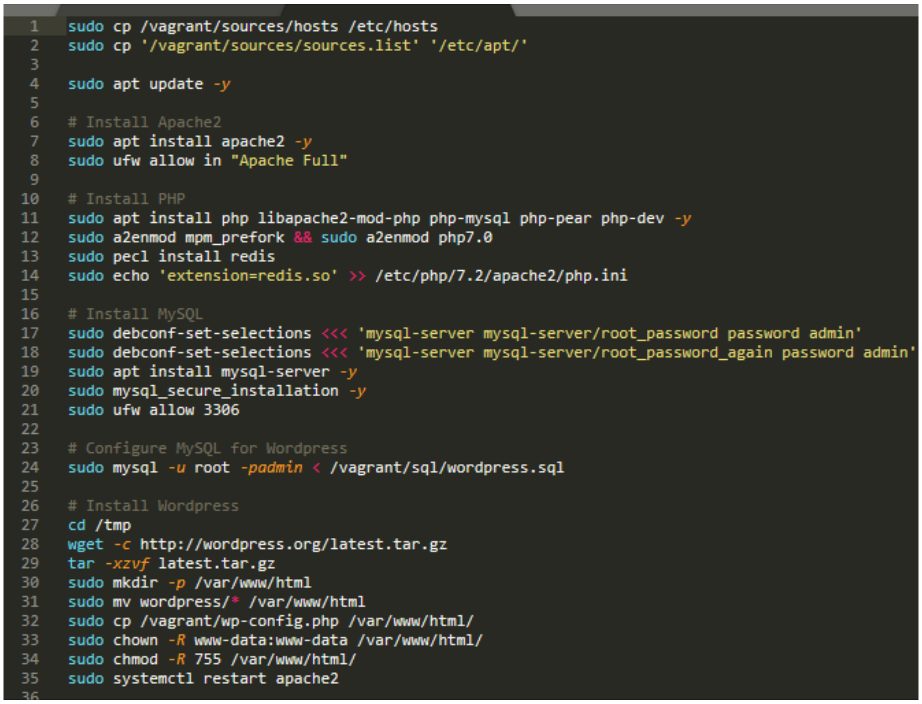
ii. Pada vagrantfile diatur seperti berikut.



```
# -*- mode: ruby -*-
# vi: set ft=ruby :

# All Vagrant configuration is done below. The "2" in Vagrant.configure
# configures the configuration version (we support older styles for
# backwards compatibility). Please don't change it unless you know what
# you're doing.
Vagrant.configure("2") do |config|

  (1..2).each do |i|
    config.vm.define "wordpress#{i}" do |node|
      node.vm.hostname = "wordpress#{i}"
      node.vm.box = "bento/ubuntu-18.04"
      node.vm.network "private_network", ip: "192.168.16.#{41+i}"

      node.vm.provider "virtualbox" do |vb|
        vb.name = "wordpress#{i}"
        vb.gui = false
        vb.memory = "512"
      end

      node.vm.provision "shell", path: "bash/wordpress.sh", privileged: false
    end
  end

  (1..3).each do |i|
    config.vm.define "redis#{i}" do |node|
      node.vm.hostname = "redis#{i}"
      node.vm.box = "bento/ubuntu-18.04"
      node.vm.network "private_network", ip: "192.168.16.#{43+i}"

      node.vm.provider "virtualbox" do |vb|
        vb.name = "redis#{i}"
        vb.gui = false
        vb.memory = "512"
      end

      node.vm.provision "shell", path: "bash/redis#{i}.sh", privileged: false
    end
  end

end
```

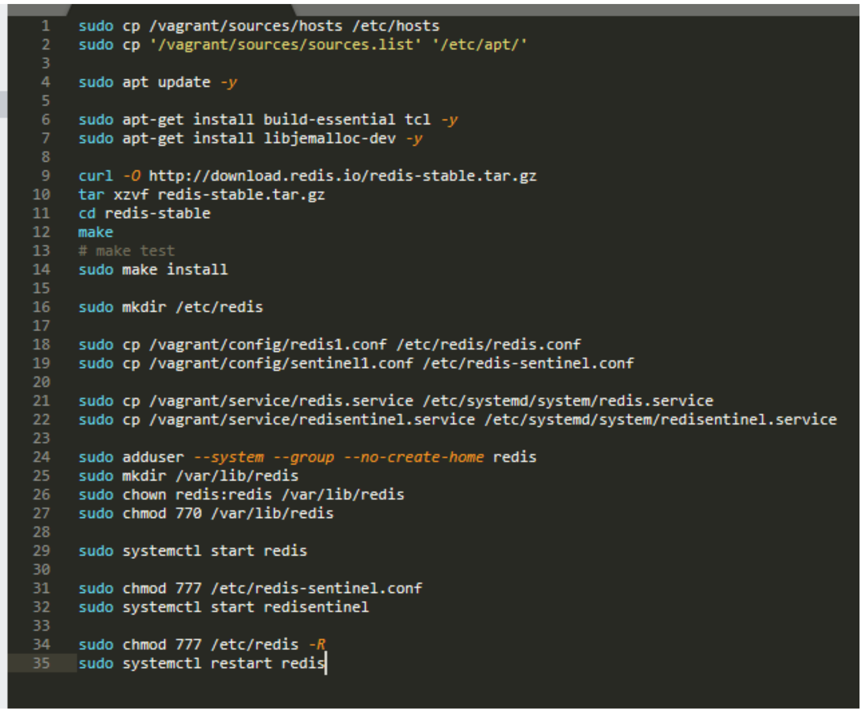
iii. Membuat provision script sebagai berikut.

1. Worpress.sh



```
sudo cp /vagrant/sources/hosts /etc/hosts
sudo cp '/vagrant/sources/sources.list' '/etc/apt/'

sudo apt update -y

# Install Apache2
sudo apt install apache2 -y
sudo ufw allow in "Apache Full"

# Install PHP
sudo apt install php libapache2-mod-php php-mysql php-pear php-dev -y
sudo a2enmod mpm_prefork && sudo a2enmod php7.0
sudo pecl install redis
sudo echo 'extension=redis.so' >> /etc/php/7.2/apache2/php.ini

# Install MySQL
sudo debconf-set-selections <<< 'mysql-server mysql-server/root_password password admin'
sudo debconf-set-selections <<< 'mysql-server mysql-server/root_password_again password admin'
sudo apt install mysql-server -y
sudo mysql_secure_installation -y
sudo ufw allow 3306

# Configure MySQL for Wordpress
sudo mysql -u root -padmin < /vagrant/sql/wordpress.sql

# Install Wordpress
cd /tmp
wget -c http://wordpress.org/latest.tar.gz
tar -xzvf latest.tar.gz
sudo mkdir -p /var/www/html
sudo mv wordpress/* /var/www/html
sudo cp /vagrant/wp-config.php /var/www/html/
sudo chown -R www-data:www-data /var/www/html/
sudo chmod -R 755 /var/www/html/
sudo systemctl restart apache2
```

2. Redis1.sh, redis2.sh dan redis3.sh

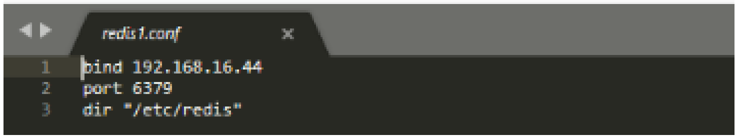


```
sudo cp /vagrant/sources/hosts /etc/hosts
sudo cp '/vagrant/sources/sources.list' '/etc/apt/'

sudo apt update -y

sudo apt-get install build-essential tcl -y
sudo apt-get install libjemalloc-dev -y

curl -O http://download.redis.io/redis-stable.tar.gz
tar xzvf redis-stable.tar.gz
cd redis-stable
make
# make test
sudo make install

sudo mkdir /etc/redis

sudo cp /vagrant/config/redis1.conf /etc/redis/redis.conf
sudo cp /vagrant/config/sentinel1.conf /etc/redis-sentinel.conf

sudo cp /vagrant/service/redis.service /etc/systemd/system/redis.service
sudo cp /vagrant/service/redisentinel.service /etc/systemd/system/redisentinel.service

sudo adduser --system --group --no-create-home redis
sudo mkdir /var/lib/redis
sudo chown redis:redis /var/lib/redis
sudo chmod 770 /var/lib/redis

sudo systemctl start redis

sudo chmod 777 /etc/redis-sentinel.conf
sudo systemctl start redisentinel

sudo chmod 777 /etc/redis -R
sudo systemctl restart redis
```

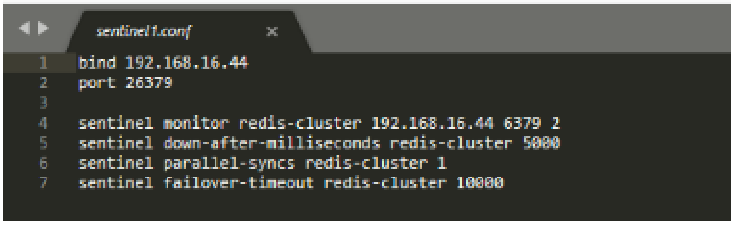
untuk redis2.sh dan redis3.sh ubah redis1.conf dan sentinel1.conf sesuai dengan indeks redisnya. Menjadi redis2.conf dan sentinel2.conf utk redis2 dan seterusnya.

iv. Membuat file konfigurasi

1. Redis1.conf dan sentinel1.conf (dari atas ke bawah)

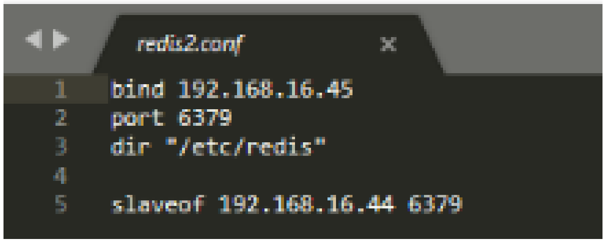


```
bind 192.168.16.44
port 6379
dir "/etc/redis"
```

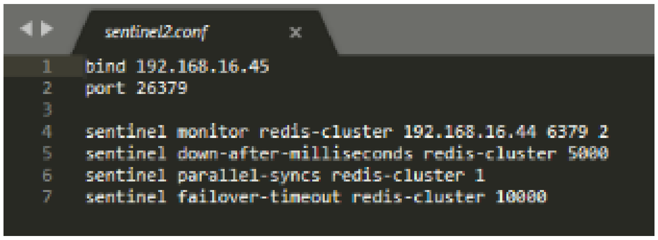


```
bind 192.168.16.44
port 26379

sentinel monitor redis-cluster 192.168.16.44 6379 2
sentinel down-after-milliseconds redis-cluster 5000
sentinel parallel-syncs redis-cluster 1
sentinel failover-timeout redis-cluster 10000
```

2. Redis2.conf dan sentinel2.conf (dari atas ke bawah)

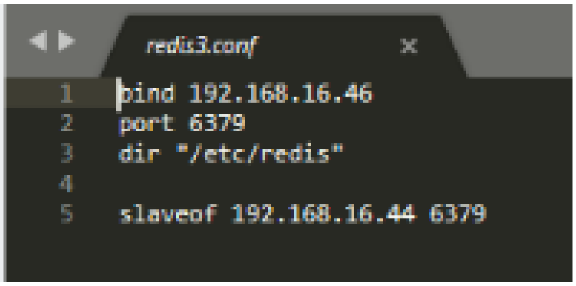


```
bind 192.168.16.45
port 6379
dir "/etc/redis"

slaveof 192.168.16.44 6379
```

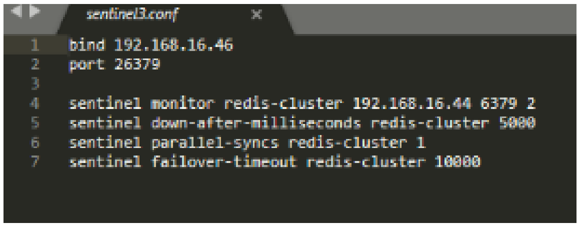


```
bind 192.168.16.45
port 26379

sentinel monitor redis-cluster 192.168.16.44 6379 2
sentinel down-after-milliseconds redis-cluster 5000
sentinel parallel-syncs redis-cluster 1
sentinel failover-timeout redis-cluster 10000
```

3. Redis3.conf dan sentinel3.conf (dari atas ke bawah)



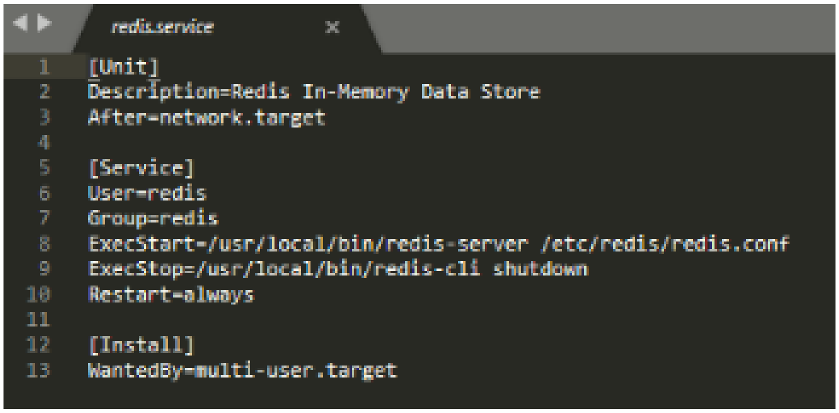
```
bind 192.168.16.46
port 6379
dir "/etc/redis"

slaveof 192.168.16.44 6379
```



```
bind 192.168.16.46
port 26379

sentinel monitor redis-cluster 192.168.16.44 6379 2
sentinel down-after-milliseconds redis-cluster 5000
sentinel parallel-syncs redis-cluster 1
sentinel failover-timeout redis-cluster 10000
```

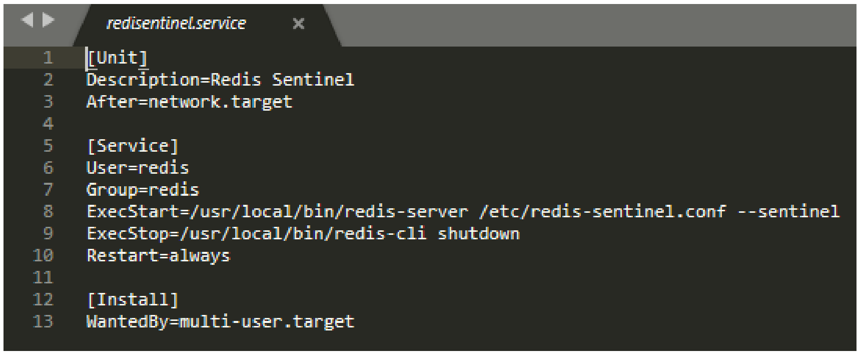
v. Membuat file konfigurasi tambahan

1. Redis.service



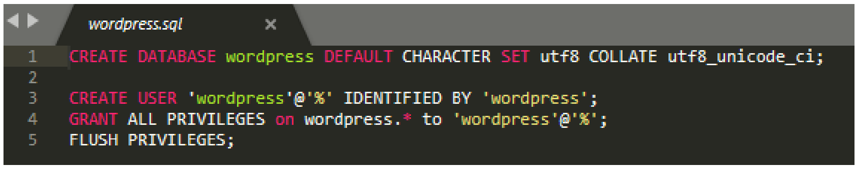
```
[Unit]
Description=Redis In-Memory Data Store
After=network.target

[Service]
User=redis
Group=redis
ExecStart=/usr/local/bin/redis-server /etc/redis/redis.conf
ExecStop=/usr/local/bin/redis-cli shutdown
Restart=always

[Install]
WantedBy=multi-user.target
```

2. Redisentinel.service



```
[Unit]
Description=Redis Sentinel
After=network.target

[Service]
User=redis
Group=redis
ExecStart=/usr/local/bin/redis-server /etc/redis-sentinel.conf --sentinel
ExecStop=/usr/local/bin/redis-cli shutdown
Restart=always

[Install]
WantedBy=multi-user.target
```

3. Wordpress.sql



```
CREATE DATABASE wordpress DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;

CREATE USER 'wordpress'@'%' IDENTIFIED BY 'wordpress';
GRANT ALL PRIVILEGES on wordpress.* to 'wordpress'@'%';
FLUSH PRIVILEGES;
```

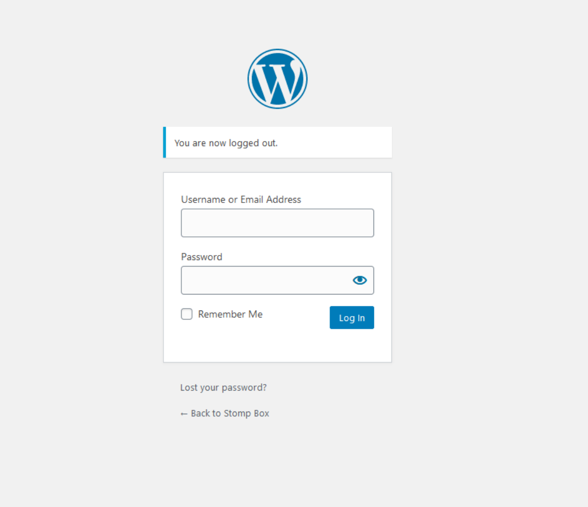
vi. Menjalankan vagrant : vagrant up

vii. Menjalanjan redis cluster pada server master : redis-cli –h 192.168.16.44

2. Menginstal Wordpress

a. Masuk pada halaman web 192.168.16.42/wp-admin

b. Setelah menjalankan instalasi maka akan muncul tampilan demikian.



You are now logged out.

c. Dilakuan hal yang sama untuk web 192.168.16.43

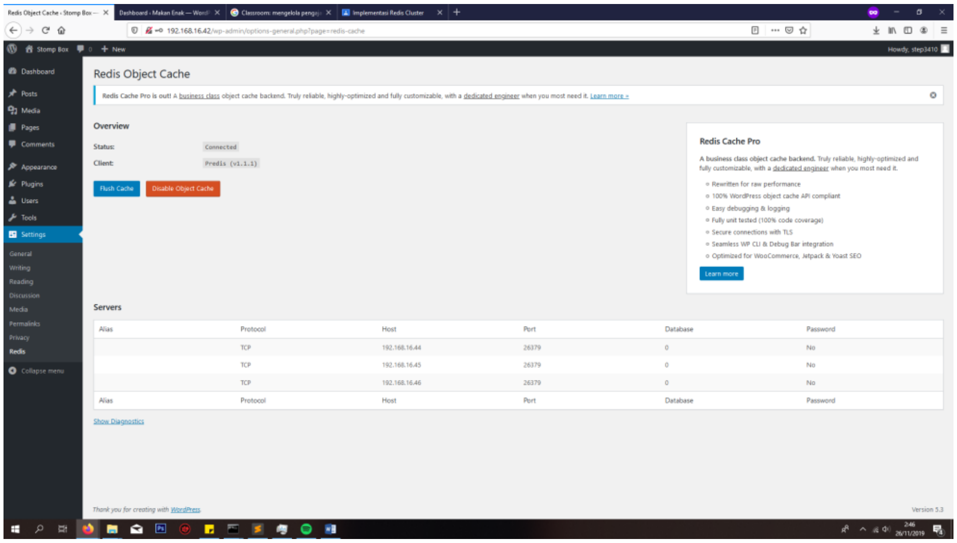
3. Menginstal Redis Object Chache ke Server Wordress 1 (192.168.16.42)

a. Masuk ke panel admin. Pilih Plugins lalu cari dan install Redis Object Cache

b. Tambahkans Konfigurasi pada `/var/www/html/wp-config.php` pada server `wordpress1`, kemudian tambahkan line berikut.

```
define('FS_METHOD', 'direct');
define('WP_REDIS_SENTINEL', 'redis-cluster');
define('WP_REDIS_SERVERS', ['tcp://192.168.16.23:26379',
'tcp://192.168.16.24:26379', 'tcp://192.168.16.25:26379']);
```

c. Aktifkan redis cache.



Redis Object Cache

Redis Cache Pro is out! A business class object cache backend. Truly reliable, highly-optimized and fully customizable, with a dedicated engineer when you most need it. Learn more »

Overview

Status: Connected

Client: Predis (v1.1.1)

Redis Cache Pro

A business class object cache backend. Truly reliable, highly-optimized and fully customizable, with a dedicated engineer when you most need it.

- Rewritten for raw performance
- 100% WordPress object cache API compliant
- Easy debugging & logging
- Fully unit tested (100% code coverage)
- Secure connections with TLS
- Seamless WP CLI & Debug Bar integration
- Optimized for WooCommerce, Jetpack & Yoast SEO

Servers

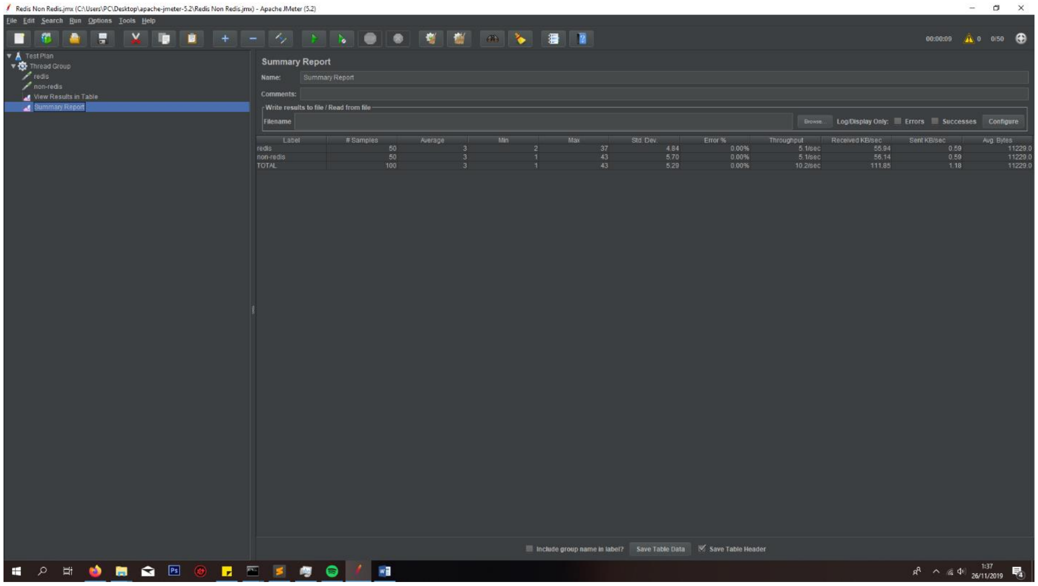
Alias	Protocol	Host	Port	Database	Password
	TCP	192.168.16.44	26379	0	No
	TCP	192.168.16.45	26379	0	No
	TCP	192.168.16.46	26379	0	No
Alias	Protocol	Host	Port	Database	Password

Thank you for creating with WordPress.

Version 5.3

4. Menguji dengan JMeter

a. Dengan 50 user


Redis Non Redis.jmx (C:\Users\PC\Desktop\apache-jmeter-5.2\Redis Non Redis.jmx) - Apache JMeter (5.2)
File Edit Search Run Options Tools Help
00:00:09 0 0/50
Test Plan
Thread Group
redis
non-redis
View Results in Table
Summary Report
Browse...
Configure
Save Table Data
1:37
26/11/2019

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

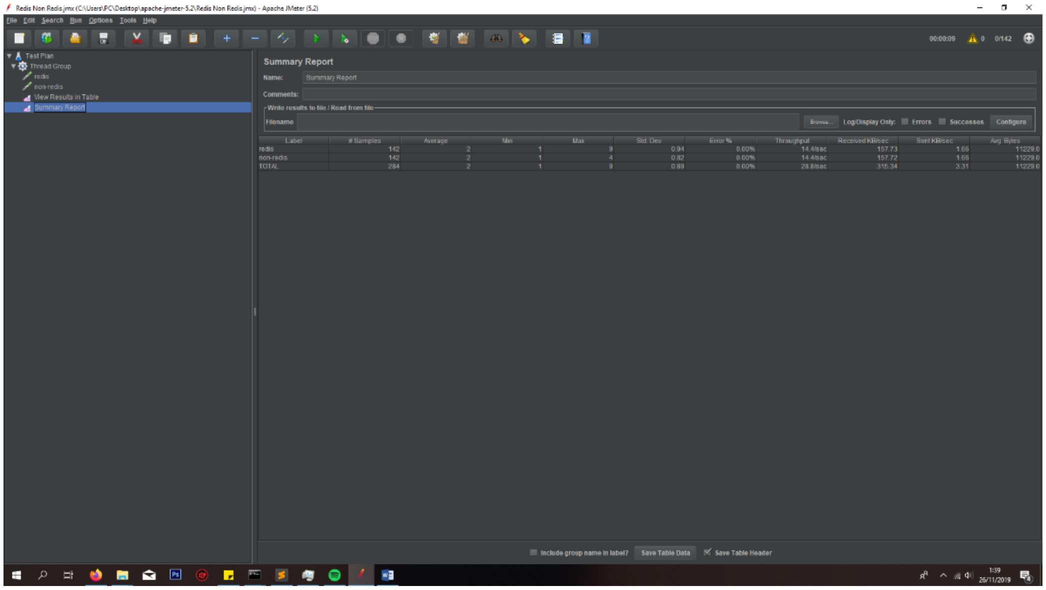
Filename

Log/Display Only: □ Errors □ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
redis	50	3	2	37	4.84	0.00%	5.1/sec	55.94	0.59	11229.0
non-redis	50	3	1	43	5.70	0.00%	5.1/sec	56.14	0.59	11229.0
TOTAL	100	3	1	43	5.29	0.00%	10.2/sec	111.85	1.18	11229.0

□ Include group name in label? ☑ Save Table Header

b. Dengan 142 User


Redis Non Redis.jmx (C:\Users\PC\Desktop\apache-jmeter-5.2\Redis Non Redis.jmx) - Apache JMeter (5.2)
File Edit Search Run Options Tools Help
00:00:09 0 0/142
Test Plan
Thread Group
redis
non-redis
View Results in Table
Summary Report
Browse...
Configure
Save Table Data
1:39
26/11/2019

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

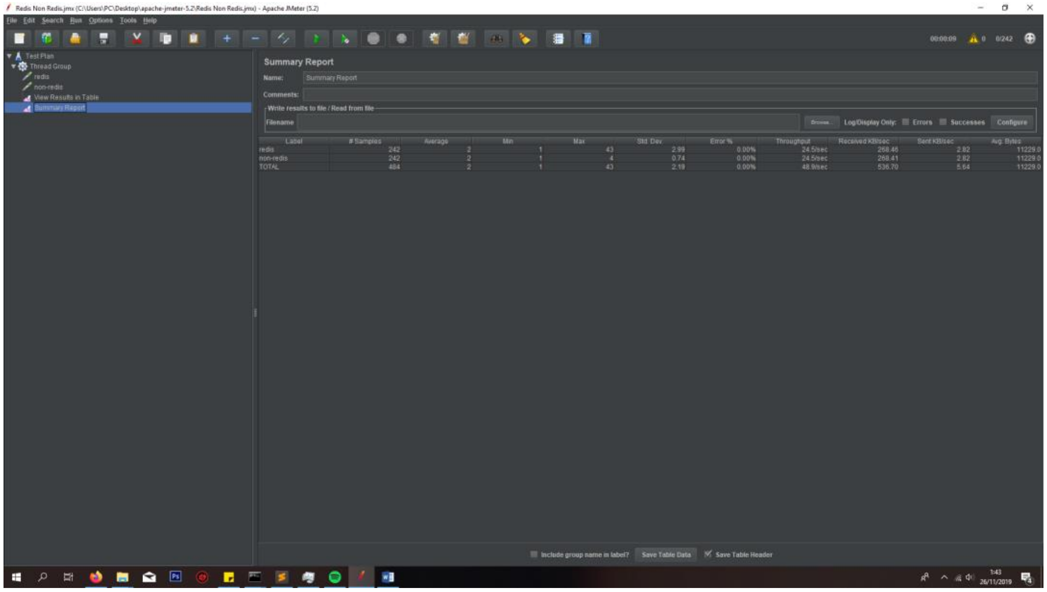
Filename

Log/Display Only: □ Errors □ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
redis	142	2	1	9	0.94	0.00%	14.4/sec	157.73	1.65	11229.0
non-redis	142	2	1	4	0.82	0.00%	14.4/sec	157.72	1.65	11229.0
TOTAL	284	2	1	9	0.89	0.00%	28.8/sec	315.34	3.31	11229.0

□ Include group name in label? ☑ Save Table Header

c. Dengan 242 User



Redis Non Redis.jmx (C:\Users\PC\Desktop\apache-jmeter-5.2\Redis Non Redis.jmx) - Apache JMeter (5.2)
File Edit Search Run Options Tools Help
00:00:09 ⚠ 0 0/242
Test Plan
Thread Group
redis
non-redis
View Results in Table
Summary Report
Browse...
Log/Display Only: □ Errors □ Successes
Configure
□ Include group name in label?
Save Table Data
☑ Save Table Header

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
redis	242	2	1	43	2.99	0.00%	24.5/sec	268.46	2.82	11229.0
non-redis	242	2	1	4	0.74	0.00%	24.5/sec	268.41	2.82	11229.0
TOTAL	484	2	1	43	2.19	0.00%	48.9/sec	536.70	5.64	11229.0

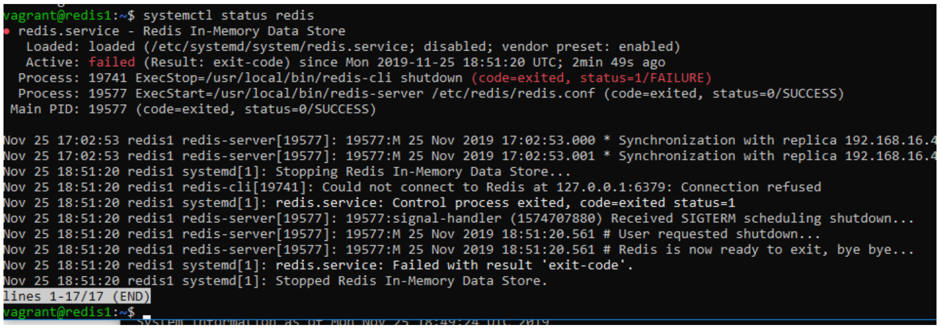
d. Dapat disimpulkan bahwa dengan menggunakan redis cendrung lebih lambar daripada tanpa redis. Hal ini dimungkinkan karena proses replikasi yang terjadi memperlambat waktu respon server.

5. Fail-Over Test

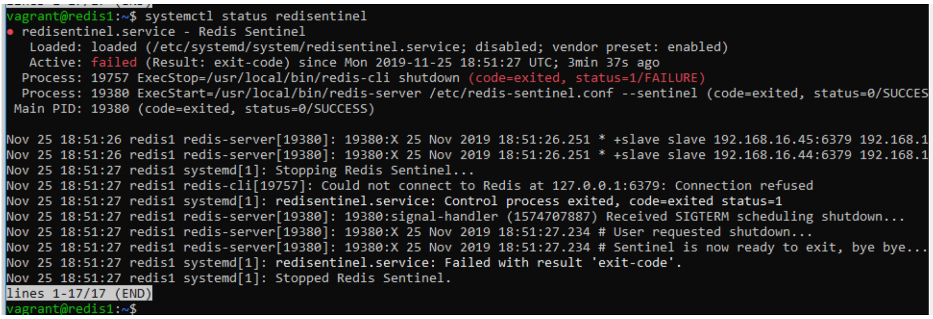
a. Matikan server master (redis1)

i. Systemctl stop redis

ii. Systemctl stop redisentinel

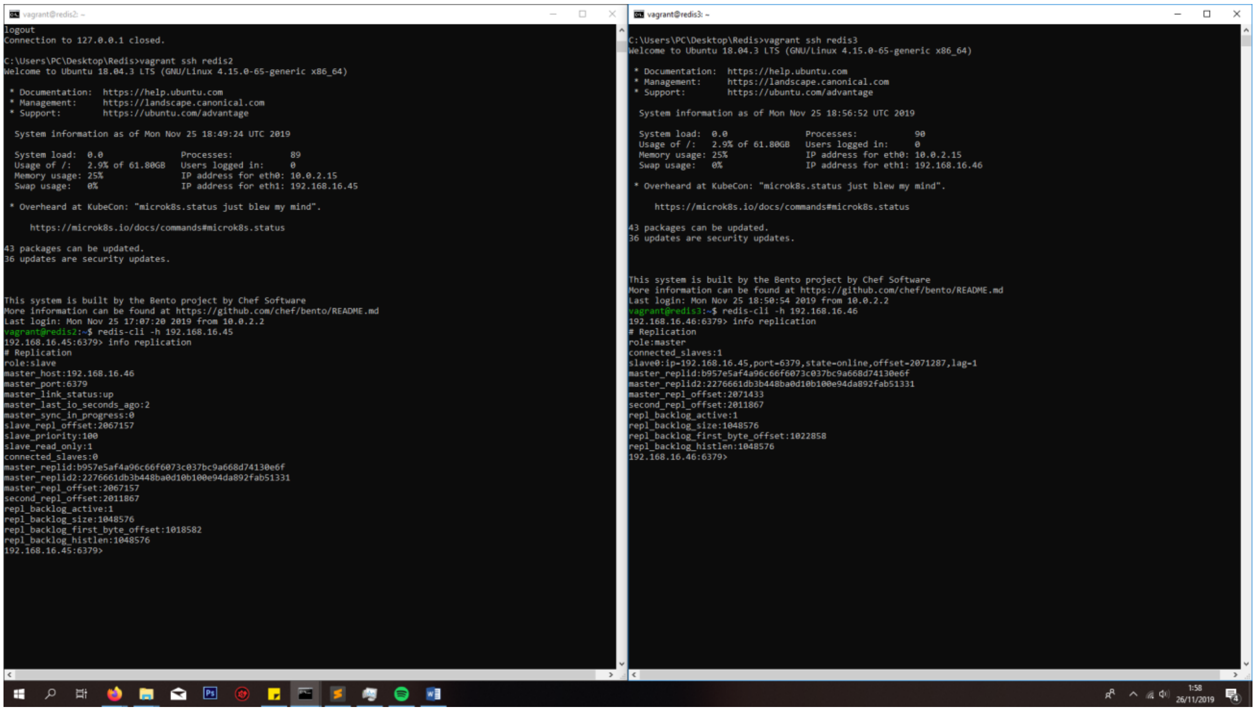


```
vagrant@redis1:~$ systemctl status redis
● redis.service - Redis In-Memory Data Store
   Loaded: loaded (/etc/systemd/system/redis.service; disabled; vendor preset: enabled)
   Active: failed (Result: exit-code) since Mon 2019-11-25 18:51:20 UTC; 2min 49s ago
  Process: 19741 ExecStop=/usr/local/bin/redis-cli shutdown (code=exited, status=1/FAILURE)
  Process: 19577 ExecStart=/usr/local/bin/redis-server /etc/redis/redis.conf (code=exited, status=0/SUCCESS)
 Main PID: 19577 (code=exited, status=0/SUCCESS)

Nov 25 17:02:53 redis1 redis-server[19577]: 19577:M 25 Nov 2019 17:02:53.000 * Synchronization with replica 192.168.16.4
Nov 25 17:02:53 redis1 redis-server[19577]: 19577:M 25 Nov 2019 17:02:53.001 * Synchronization with replica 192.168.16.4
Nov 25 18:51:20 redis1 systemd[1]: Stopping Redis In-Memory Data Store...
Nov 25 18:51:20 redis1 redis-cli[19741]: Could not connect to Redis at 127.0.0.1:6379: Connection refused
Nov 25 18:51:20 redis1 systemd[1]: redis.service: Control process exited, code=exited status=1
Nov 25 18:51:20 redis1 redis-server[19577]: 19577:signal-handler (1574707880) Received SIGTERM scheduling shutdown...
Nov 25 18:51:20 redis1 redis-server[19577]: 19577:M 25 Nov 2019 18:51:20.561 # User requested shutdown...
Nov 25 18:51:20 redis1 redis-server[19577]: 19577:M 25 Nov 2019 18:51:20.561 # Redis is now ready to exit, bye bye...
Nov 25 18:51:20 redis1 systemd[1]: redis.service: Failed with result 'exit-code'.
Nov 25 18:51:20 redis1 systemd[1]: Stopped Redis In-Memory Data Store.
lines 1-17/17 (END)
vagrant@redis1:~$
```



```
vagrant@redis1:~$ systemctl status redisentinel
● redisentinel.service - Redis Sentinel
   Loaded: loaded (/etc/systemd/system/redisentinel.service; disabled; vendor preset: enabled)
   Active: failed (Result: exit-code) since Mon 2019-11-25 18:51:27 UTC; 3min 37s ago
  Process: 19757 ExecStop=/usr/local/bin/redis-cli shutdown (code=exited, status=1/FAILURE)
  Process: 19380 ExecStart=/usr/local/bin/redis-server /etc/redis-sentinel.conf --sentinel (code=exited, status=0/SUCCES
 Main PID: 19380 (code=exited, status=0/SUCCESS)

Nov 25 18:51:26 redis1 redis-server[19380]: 19380:X 25 Nov 2019 18:51:26.251 * +slave slave 192.168.16.45:6379 192.168.1
Nov 25 18:51:26 redis1 redis-server[19380]: 19380:X 25 Nov 2019 18:51:26.251 * +slave slave 192.168.16.44:6379 192.168.1
Nov 25 18:51:27 redis1 systemd[1]: Stopping Redis Sentinel...
Nov 25 18:51:27 redis1 redis-cli[19757]: Could not connect to Redis at 127.0.0.1:6379: Connection refused
Nov 25 18:51:27 redis1 systemd[1]: redisentinel.service: Control process exited, code=exited status=1
Nov 25 18:51:27 redis1 redis-server[19380]: 19380:signal-handler (1574707887) Received SIGTERM scheduling shutdown...
Nov 25 18:51:27 redis1 redis-server[19380]: 19380:X 25 Nov 2019 18:51:27.234 # User requested shutdown...
Nov 25 18:51:27 redis1 redis-server[19380]: 19380:X 25 Nov 2019 18:51:27.234 # Sentinel is now ready to exit, bye bye...
Nov 25 18:51:27 redis1 systemd[1]: redisentinel.service: Failed with result 'exit-code'.
Nov 25 18:51:27 redis1 systemd[1]: Stopped Redis Sentinel.
lines 1-17/17 (END)
vagrant@redis1:~$
```

b. Cek pada redis2 dan redis3 yang mana menjadi master.



```
logout
Connection to 127.0.0.1 closed.

C:\Users\PC\Desktop\Redis>vagrant ssh redis2
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-65-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Nov 25 18:49:24 UTC 2019

  System load:  0.0               Processes:           89
  Usage of /:   2.9% of 61.80GB   Users logged in:     0
  Memory usage: 25%               IP address for eth0: 10.0.2.15
  Swap usage:   0%                IP address for eth1: 192.168.16.45

 * Overheard at KubeCon: "microk8s.status just blew my mind".

     https://microk8s.io/docs/commands#microk8s.status

43 packages can be updated.
36 updates are security updates.



This system is built by the Bento project by Chef Software
More information can be found at https://github.com/chef/bento/README.md
Last login: Mon Nov 25 17:07:20 2019 from 10.0.2.2
vagrant@redis2:~$ redis-cli -h 192.168.16.45
192.168.16.45:6379> info replication
# Replication
role:slave
master_host:192.168.16.46
master_port:6379
master_link_status:up
master_last_io_seconds_ago:2
master_sync_in_progress:0
slave_repl_offset:2067157
slave_priority:100
slave_read_only:1
connected_slaves:0
master_replid:b957e5af4a96c66f6073c037bc9a668d74130e6f
master_replid2:2276661db3b448ba0d10b100e94da892fab51331
master_repl_offset:2067157
second_repl_offset:2011867
repl_backlog_active:1
repl_backlog_size:1048576
repl_backlog_first_byte_offset:1018582
repl_backlog_histlen:1048576
192.168.16.45:6379>
```

```
C:\Users\PC\Desktop\Redis>vagrant ssh redis3
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-65-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Nov 25 18:56:52 UTC 2019

  System load:  0.0               Processes:           90
  Usage of /:   2.9% of 61.80GB   Users logged in:     0
  Memory usage: 25%               IP address for eth0: 10.0.2.15
  Swap usage:   0%                IP address for eth1: 192.168.16.46

 * Overheard at KubeCon: "microk8s.status just blew my mind".

     https://microk8s.io/docs/commands#microk8s.status

43 packages can be updated.
36 updates are security updates.



This system is built by the Bento project by Chef Software
More information can be found at https://github.com/chef/bento/README.md
Last login: Mon Nov 25 18:50:54 2019 from 10.0.2.2
vagrant@redis3:~$ redis-cli -h 192.168.16.46
192.168.16.46:6379> info replication
# Replication
role:master
connected_slaves:1
slave0:ip=192.168.16.45,port=6379,state=online,offset=2071287,lag=1
master_replid:b957e5af4a96c66f6073c037bc9a668d74130e6f
master_replid2:2276661db3b448ba0d10b100e94da892fab51331
master_repl_offset:2071433
second_repl_offset:2011867
repl_backlog_active:1
repl_backlog_size:1048576
repl_backlog_first_byte_offset:1022858
repl_backlog_histlen:1048576
192.168.16.46:6379>
```

c. Pada hasilnya dapat dilihat bahwa yang menjadi server adalah redis3 ketika master di-*shutdown*.